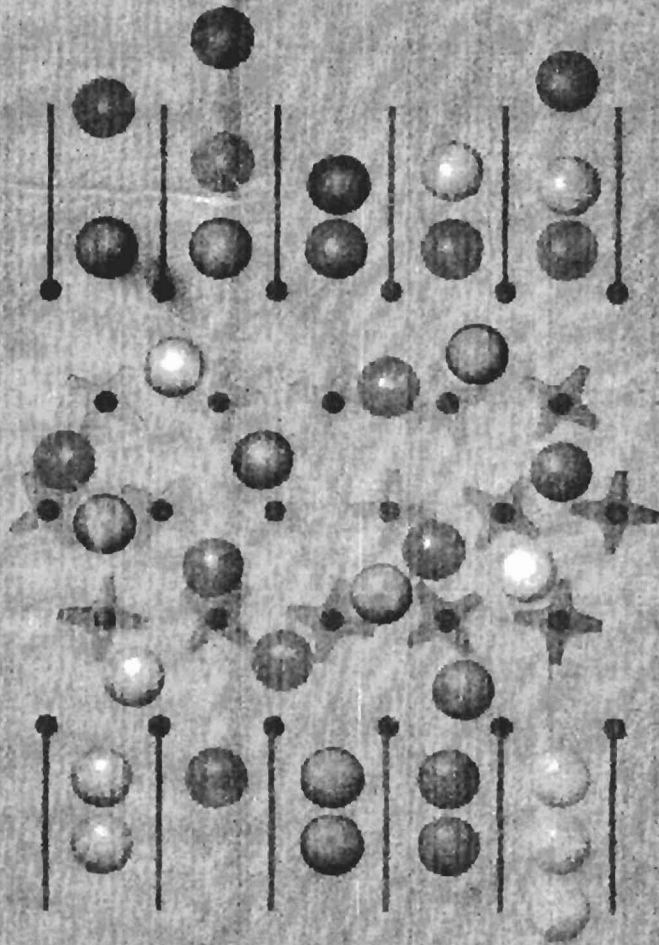


# International Data Acquisition Conference

on Event Building and Event Data Readout  
in Medium & High-Energy Physics Experiments



Fermi National Accelerator Laboratory  
Batavia, Illinois, USA

October 26 - 28, 1994

The quality of this scan reflects the quality of the original,  
paper copy, from which it was reproduced.

# Table of Contents

## Table of Contents

Conference Highlights	1
Conference Questionnaire	2
Results of Questionnaire	3
Video Tape Order Form	5
Table of Conference Video Tapes	6
Conference Speakers	10
Conference Presentations	
Session 1: Requirements & Current / Proposed Architectures	
S1-1 Introduction & Conference Goals	11
S1-2 CDF & D0 Data Acquisition Systems	12
S1-3 SLAC & KEK B Detector Data Acquisition Systems	13
S1-4 STAR Detector Data Acquisition System	14
S1-5 PHENIX Detector Data Acquisition System	15
S1-6 ATLAS, CMS & ALICE Detector Data Acquisition Systems	16
S1-7 Applications of Switching Networks and Meshes of Point-to-point Links in Massively Parallel Systems	17
S1-8 Applications of Switching Networks & Point to Point Links in Other Physics Applications	18
Session 2: Switching Tutorial & Standards ... Tutorials & Status	
S2-1 High-Speed Switching Networks	19
S2-2 ATM/SONET	20
S2-3 Fibre Channel	21
S2-4 SCI	22
Session 3: Current & Planned R&D Efforts	
S3-1 ATM Research Projects	23
S3-2 Fibre Channel Research Projects	24
S3-3 SCI Research Projects	25
S3-4 Other Research Projects	26
S3-5 Matrix of Projects and Standards	27
S3-6 New VME Standards For Physics Applications	28
Session 4: Integrated Circuits and Board Products	
S4-1 Overview of ATM Integrated Circuits & Board Products	29
S4-2 Overview of Fibre Channel Integrated Circuits & Board Products	30
S4-3 Overview of SCI Integrated Circuits & Board Products	31
S4-4 Design of SCI-Class Interconnects	32
Session 5: Switches	
S5-1 ATM Switches for Telecommunications Applications	33
S5-2 High-Performance Switching in the MAN and Public Network	34
S5-3 Fibre Channel Switches	35
S5-4 SCI Switches	36
S5-5 Overview of Optical Switches	37
S5-6 Prizma Switch	38
S5-7 Phoenix Switch & Bell Labs Switch Research	39
S5-8 Switches for Point to Point Links using OMI/HIC Technology	40
Session 6: Simulation Goals & Techniques	
S6-1 Requirements & Goals of Simulation	41
S6-2 Behavioral Simulation and High Level Modelling	42
S6-3 Review of SCI Simulation Results	43
S6-4 Review of ATM, Fibre Channel and Conical Network Simulation Results	44
Session 7: Software	
S7-1 Software Issues When Implementing An ATM Network	45
S7-2 Data Acquisition Software Design Issues	46
S7-3 Software Protocols for Event Builder Switching Networks	47

## Table of Contents

Session 8: System Design	
S8-1 A Scalable Fibre Channel Architecture for Event Building	48
S8-2 Pros and Cons: Commercial & Non-Commercial Switching Networks	49
S8-3 Event Data Flow Control Techniques	50
Poster Sessions	51
DAQ Simulation Library	52
A 155 Mbit/s VME to ATM interface with special features for event building applications based on ATM switching fabrics	53
Performance Simulations of Networks with Point-to-Point Links	54
Application of SCI in the STAR data acquisition system	55
Event Building Using an ATM Switching Network in the CLAS Detector at CEBAF	56
The Event Builder of the ZEUS Detector	57
The CLEO III Data Acquisition System	58
DART Data Acquisition System	59
Sloan Digital Sky Survey Data Acquisition System	60
The KLOE DAQ System	61
A Continuous Time Stamping Time Digitizer Architecture for HEP Applications	62
SCI in Data Acquisition Systems	63
The 3D-Flow System as Programmable Switch for Moving and Reducing Data in DAQ Applications	64
An SCI Video DRAM Memory Module	65
FASTBUS CHI-SCI Link	66
SWIPP - Switched Interconnection of Parallel Processors - A General Purpose Heterogeneous Multicomputer Optimized For Data Acquisition	67
Dual Port Memory	68
Performance Evaluation Tool for DAQ Computers (DAQBENCH)	69
Global Traffic Control System on High Speed Event Builder	70
Testing of the HP G-link Chip Set for an Event Builder Application	71
SCI with DSPs & RISC Processors for LHC 2nd Level Triggering	72
Vortex: A High Performance Parallel Processing Event Server with an ATM Interface	73
Prototype of an Event Building System Based on HiPPI	74
SyncC++, a Concurrent Language Based on C++	75
DSP based Data Acquisition Systems	76
Initial Experiences With a Network of INMOS C104 Packet Routing Switches	77
The G-2 Data Acquisition System	78
Fast Data Link & Modern RISC Processors for HEP Projects	79
List of Vendors	80
List of Attendees	81
The Conference Organizing Committee	92
The Local Organizing Committee	92
VME Standards for Physics Applications	93

## Conference HighLights

The First International Data Acquisition Conference was hosted by Fermilab, October 26-28th, 1994. A mixture of industry and laboratory speakers presented talks on a variety of subjects related to emerging standards for high speed data transport. European participation in the conference presentations was especially strong. The first day was devoted to data acquisition requirements at current and future detectors, with a review of research projects in data acquisition and tutorials on ATM, Fibre Channel and SCI. The second day covered integrated circuit, board and system level products useful in assembling large data acquisition networks. The final day included discussions of software, simulation and system design issues in future data acquisition systems.

In response to a questionnaire, attendees rated the conference at 4 out of a possible 5 in nearly every category, with almost unanimous agreement that the conference should be continued on an annual or bi-annual basis. The organizing committee would like to thank all attendees and everyone who contributed their effort to make this a successful conference at Fermilab.

## Conference Questionnaire

1. On the basis of the following general characteristics, how would you rate this conference?

	poor				excellent
Host Institute Fermilab	1	2	3	4	5
Poster/Vendor Selection of Posters	1	2	3	4	5
Oral Presentations Selection of topics	1	2	3	4	5
Selection of speakers	1	2	3	4	5
Panel Discussion Mix of panel members	1	2	3	4	5
Topics discussed	1	2	3	4	5
Accommodations (one hotel) Holiday Inn	1	2	3	4	5
Red Roof Inn	1	2	3	4	5
Social Events Dinner at Fermilab	1	2	3	4	5
Dinner at St. Charles Place	1	2	3	4	5
Wine & Cheese Reception	1	2	3	4	5
Conference Taxi Service	1	2	3	4	5

2. Did you have adequate time to visit the Poster/Vendor Area?      Yes    No

3. Was the Poster/Vendor area too isolated from the oral presentation area of the conference?      Yes    No

4. Should this conference be continued on an annual basis?      Yes    No

a. If yes, select the next host institute. \_\_\_\_\_

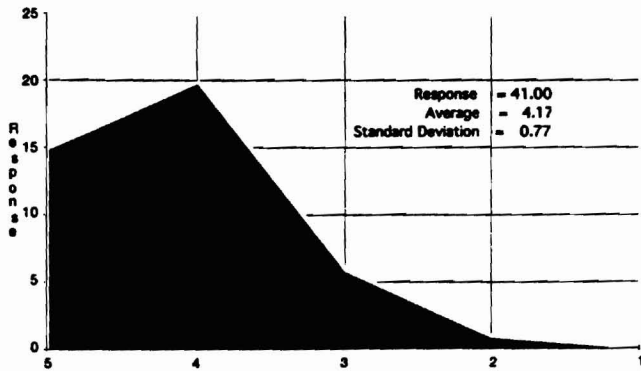
Comments: \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

# Results of Questionnaire

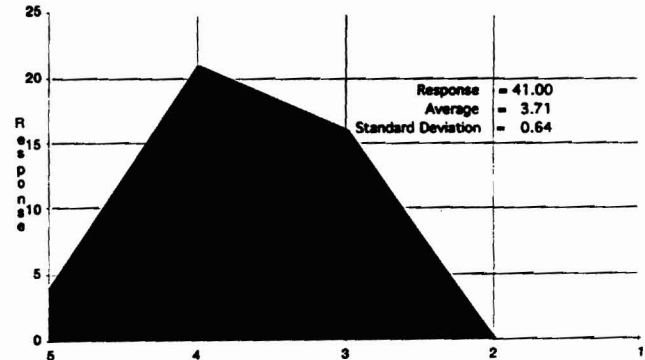
## Results of Questionnaire (202 Attendees) (42 Attendees Responded)

1. On the basis of the following general characteristics, how would you rate this conference?

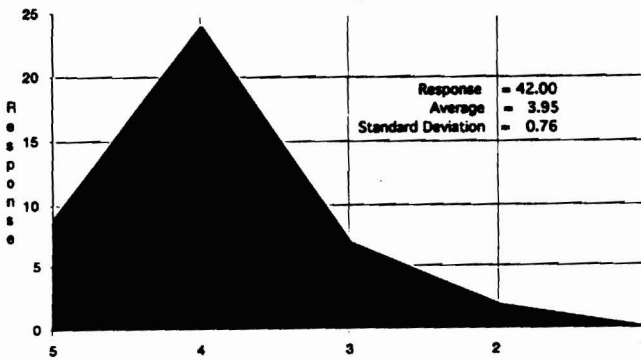
Host Institute



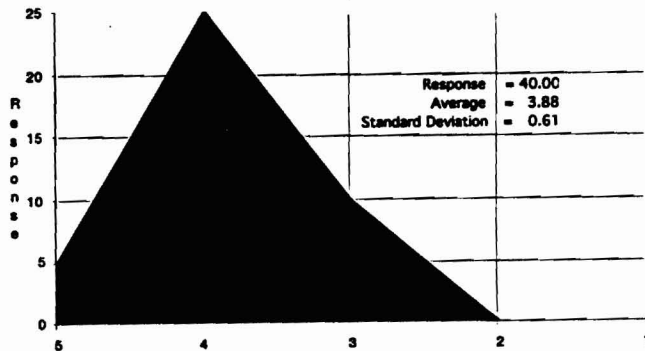
Selection of Posters



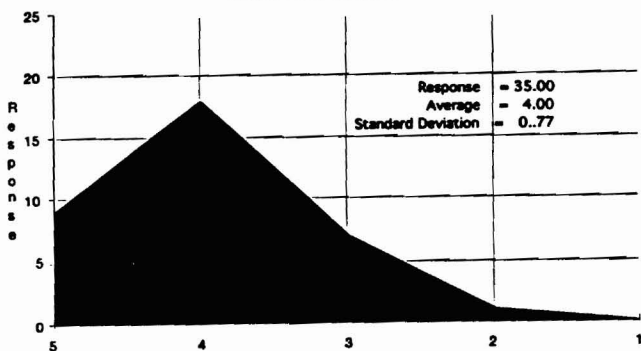
Oral Presentation  
Selection of Topics



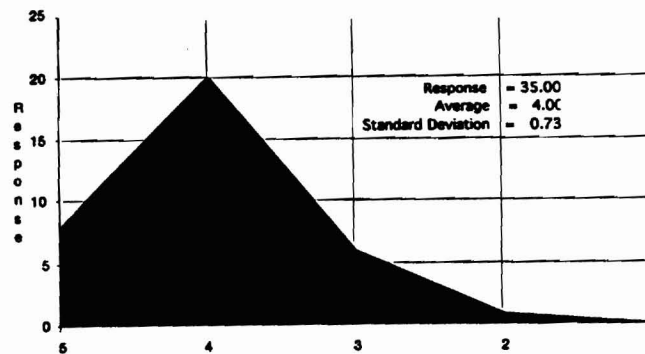
Oral Presentation  
Selection of Speaker



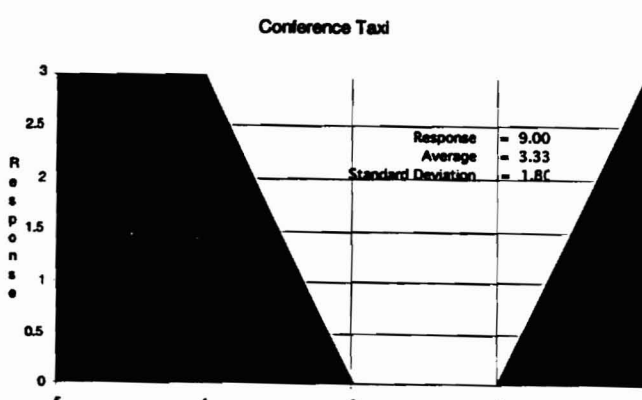
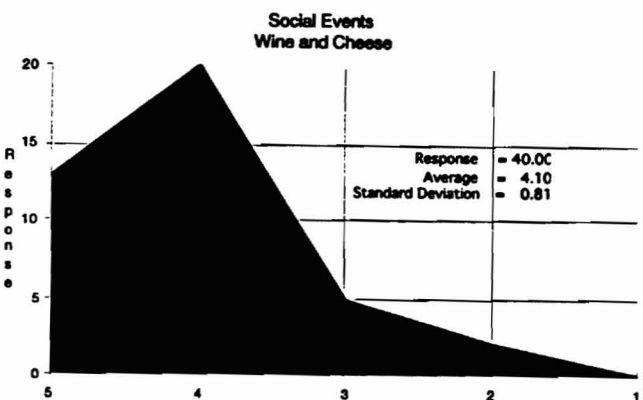
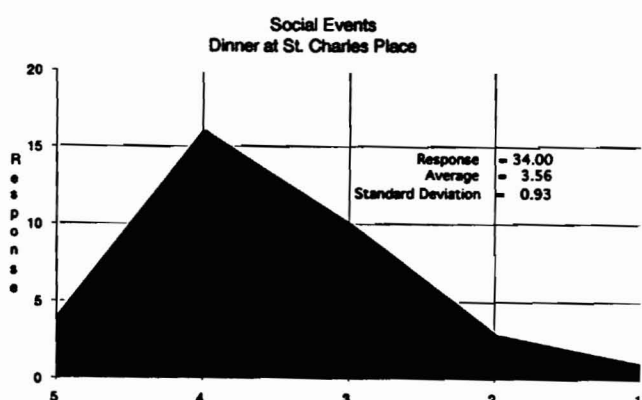
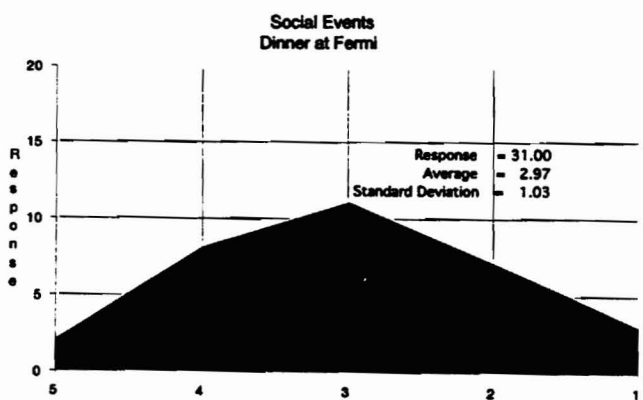
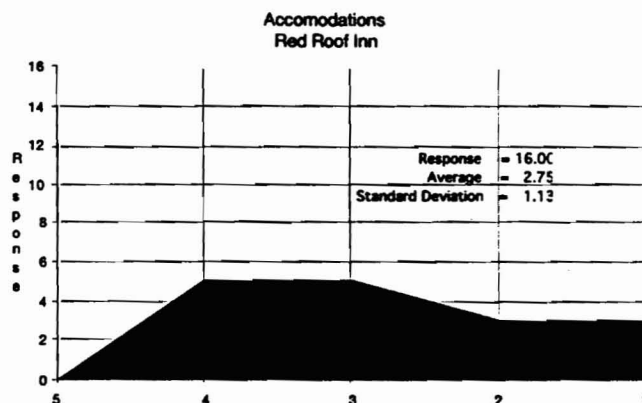
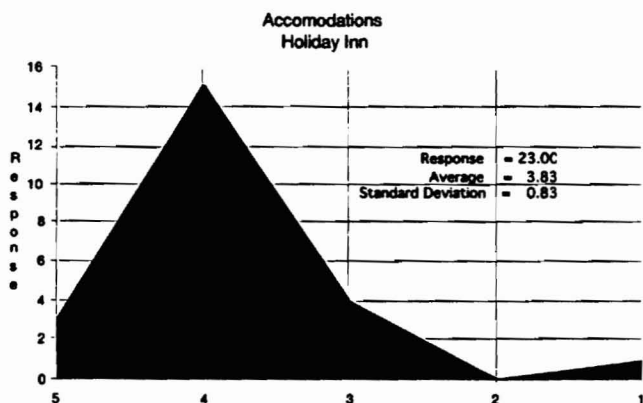
Panel Discussion  
Mix of Panel Members



Panel Discussions  
Topics Discussed



# Results of Questionnaire



- |   |    |    |
|---|----|----|
| 2. Did you have adequate time to visit the Poster/Vendor Area?                                | 37 | 5  |
| 3. Was the Poster/Vendor area too isolated from the oral presentation area of the conference? | 25 | 17 |
| 4. Should this conference be continued on an annual basis?                                    | 26 | 12 |

**Of the attendees responding w/NQ; 11 stated every other year.**

a. If yes, select the next host institute.

Europe 6, CERN 7



## Video Tape Order Form

The Data Acquisition Conference was recorded on 14 VHS video tapes as shown in the following tables. These tapes are available in NTSC (USA) format at a cost of \$10.00 per tape or \$140.00 for the full 14 tape set of the conference. They are also available in PAL (European) format at \$40.00 per tape or \$560.00 for the full set. To order, please enclose the following order form and make your check payable to:

**Fermilab  
PO Box 500  
Batavia, IL 60510**

Tape	Quantity	Price Each NTSC / PAL	Item Cost (Qty x Price Each)
Tape 1		\$10 / \$40	
Tape 2		\$10 / \$40	
Tape 3		\$10 / \$40	
Tape 4		\$10 / \$40	
Tape 5		\$10 / \$40	
Tape 6		\$10 / \$40	
Tape 7		\$10 / \$40	
Tape 8		\$10 / \$40	
Tape 9		\$10 / \$40	
Tape 10		\$10 / \$40	
Tape 11		\$10 / \$40	
Tape 12		\$10 / \$40	
Tape 13		\$10 / \$40	
Tape 14		\$10 / \$40	
Full Set (Tape 1 thru. Tape 14)		\$140 / \$560	
		Sub-Total	
Shipping and Handling (per Tape)			\$1.00
Total Order Cost			

## Table of Conference Video Tapes

<b>Tape 1</b>			
Session 1: Requirements & Current And Proposed Architectures			
Date	Time	Session	Title
10/26/94	8:30	S1-1	"Introduction & Conference Goals" (Joel Butler - Fermilab)
	8:45	S1-2	"CDF & D0 Data Acquisition Systems" (Paris Sphicas - MIT)
	9:05	S1-3	"SLAC & KEK B Detector Data Acquisition Systems" (Walt Innes - SLAC)
	9:25	S1-4	"STAR Detector Data Acquisition System" (Mike Levine - BNL)
	9:40	S1-5	"PHENIX Detector Data Acquisition System" (Cheng-Yi Chi - Nevis)
	9:55	S1-6	"ATLAS, CMS & ALICE Detector Data Acquisition Systems" (Livio Mapelli - CERN/LBL)

<b>Tape 2</b>			
Session 1: Requirements & Current And Proposed Architectures			
Date	Time	Session	Title
11/26/94	10:45	S1-7	"Applications of Switching Networks and Meshes of Point-to-point Links in Massively Parallel Systems" (Mark Fischler - Fermilab)
	11:00	S1-8	"Applications of Switching Networks & Point to Point Links in Other Physics Applications" (Marvin Johnson - Fermilab)
	11:15	S2-1	"High-Speed Switching Networks" (Don Peterson - Bell Labs)

<b>Tape 3</b>			
Session 2: Switching Tutorial & Standards ... Tutorials & Status			
Date	Time	Session	Title
10/26/94	13:30	S2-2	"ATM/SONET" (Jean-Yves LeBoudec - EPFL)
	14:10	S2-3	"Fibre Channel" (Roger Cummings - Storage Technology)

<b>Tape 4</b>			
Session 2: Switching Tutorial & Standards ... Tutorials & Status			
Date	Time	Session	Title
10/26/94	14:50	S2-4	"SCI" (Hans Muller - CERN)
	15:50	S3-1	"ATM Research Projects" (Jean-Pierre Dufey - CERN)
	16:22	S3-2	"Fibre Channel Research Projects" (Erik van der Bij - CERN)

## Table of Conference Video Tapes

<b>Tape 5</b>			
Session 3: Current & Planned R&D Efforts			
Date	Time	Session	Title
10/26/94	16:50	S3-3	"SCI Research Projects" (Fred Wickens - Rutherford)
	17:20	S3-4	"Other Research Projects" (Masa Nomachi - KEK)
	17:50	S3-5	"Matrix of Projects and Standards" (Robert McLaren - CERN)
10/27/94	8:30	S3-6	"New VME Standards For Physics Applications" (Robert Downing - University of Illinois)

<b>Tape 6</b>			
Session 4: Integrated Circuits and Board Products			
Date	Time	Session	Title
10/27/94	8:45	S4-1	"Overview of ATM Integrated Circuits & Board Products" (Lee Goldberg - Electronic Design Magazine)
	9:20	S4-2	"Overview of Fibre Channel Integrated Circuits & Board Products" (Murray Thompson - University of Wisconsin)

<b>Tape 7</b>			
Session 4: Integrated Circuits and Board Products			
Date	Time	Session	Title
10/27/94	9:55	S4-3	"Overview of SCI Integrated Circuits & Board Products" (Volker Lindenstruth - LBL)
	10:50	S4-4	"Design of SCI-Class Interconnects" (Wayne Nation - IBM)

<b>Tape 8</b>			
Session 5: Switches			
Date	Time	Session	Title
10/27/94	11:15	S5-1	"ATM Switches for Telecommunications Applications" (Ian Mahood - Alcatel)
	11:45	S5-2	"High-Performance Switching in the MAN and Public Network" (Barry Phillips - Adger Smythe Corp.)
	14:15	S5-3	"Fibre Channel Switches" (Clint Jurgens - AnCor Communications)
	14:45	S5-4	"SCI Switches" (Bin Wu - University of Oslo)

## Table of Conference Video Tapes

Tape 9			
Session 5: Switches			
Date	Time	Session	Title
10/27/94	15:15	S5-5	"Overview of Optical Switches" (Larry McAdams - Optivision)
	16:05	S5-6	"Prizma Switch" (Ton Engbersen - IBM Zurich)
	16:35	S5-7	"Phoenix Switch & Bell Labs Switch Research" (Andre Wiesel - EPFL)
	17:05	S5-8	"Switches for Point to Point Links using OMI/HIC Technology" (Ernst Kristiansen - SINTEF)

Tape 10			
Session 6: Simulation Goals & Techniques			
Date	Time	Session	Title
10/28/94	8:30	S6-1	"Requirements & Goals of Simulation" (Steve Tether - MIT)
	9:00	S6-2	"Behavioral Simulation and High Level Modelling" (Mike Haney - University of Illinois)
	9:30	S6-3	"Review of SCI Simulation Results" (Andre Bogaerts - CERN)

Tape 11			
Session 6: Simulation Goals & Techniques			
Date	Time	Session	Title
10/28/94	9:55	S6-4	"Review of ATM, Fibre Channel and Conical Network Simulation Results" (Irakli Mandjavidze - CERN/Saclay)

## Table of Conference Video Tapes

<b>Tape 12</b>			
Session 7: Software			
Date	Time	Session	Title
10/28/94	10:55	S7-1	"Software Issues When Implementing An ATM Network" (Henry Dardy - Naval Research Laboratory)
	11:20	S7-2	"Data Acquisition Software Design Issues" (Bob Russell - University of New Hampshire)
	11:50	S7-3	"Software Protocols for Event Builder Switching Networks" (Irakli Mandjavidze - CERN/Saclay)

<b>Tape 13</b>			
Session 8: System Design			
Date	Time	Session	Title
10/28/94	14:20	S8-1	"A Scalable Fibre Channel Architecture for Event Building" (Bill Greiman - LBL)
	14:50	S8-2	"Pros and Cons: Commercial & Non-Commercial Switching Networks" (Alexandro Marchioro - CERN)
	15:20	S8-3	"Event Data Flow Control Techniques" (Mark Bowden - Fermilab)

<b>Tape 14</b>			
Session 9: System Design Panel Session & Conference Wrap Up			
Date	Time	Session	Title
10/28/94	16:20	S9-1	Panel Discussion (System Modelling & Design) (Irwin Gaines - Fermilab)
	17:30	S9-2	Conference Wrap Up Talk (Sergio Cittolin - CERN)



## Conference Speakers

October 26, 1994



Shown in photo left to right, front to back:  
Robert McLaren (S3-5), Jean-Pierre Dufey (S3-1), Jean-Yves Dufey (S2-2), Joel Butler (S1-1), Roger Cummings (S2-3), Erik van der Bij (S3-2), Livio Mapelli (S1-6), Hans Muller (S2-4), Fred Wickens (S3-3), Walt Innes (S1-3), Marvin Johnson (S1-8)

Not Shown:  
Paris Sphicas (S1-2), Mike Levine (S1-4), Cheng-Yi Chi (S1-5), Mark Fischler (S1-7), Don Peterson (S2-1), Masa Nomachi (S3-4)

Shown in photo left to right, front to back:

Bary Phillips (S5-2), Clint Jurgens (S5-3), Larry McAdams (S5-5), Murray Thompson (S4-2), Lee Goldberg (S4-1), Bin Wu (S5-4), Wayne Nation (S4-4), Robert Downing (S3-6), Ernst Kristiansen (S5-8)

Not Shown:

Volker Lindenstruth (S4-3), Ian Mahood (S5-1), Ton Engbersen (S5-6), Andre Wiesel (S5-7)

October 27, 1994



Shown in photo left to right, front to back:

Mark Bowden (S8-2), Andre Bogaerts (S6-3), Bob Russel (S7-2), Bill Greiman (S8-1), Irakli Mandjavidze (S6-4, S7-3), Steve Tether (S6-1), Mike Haney (S6-2)

Not Shown:

Henry Dardy (S7-1), Alexandro Marchioro (S8-2)

October 28, 1994





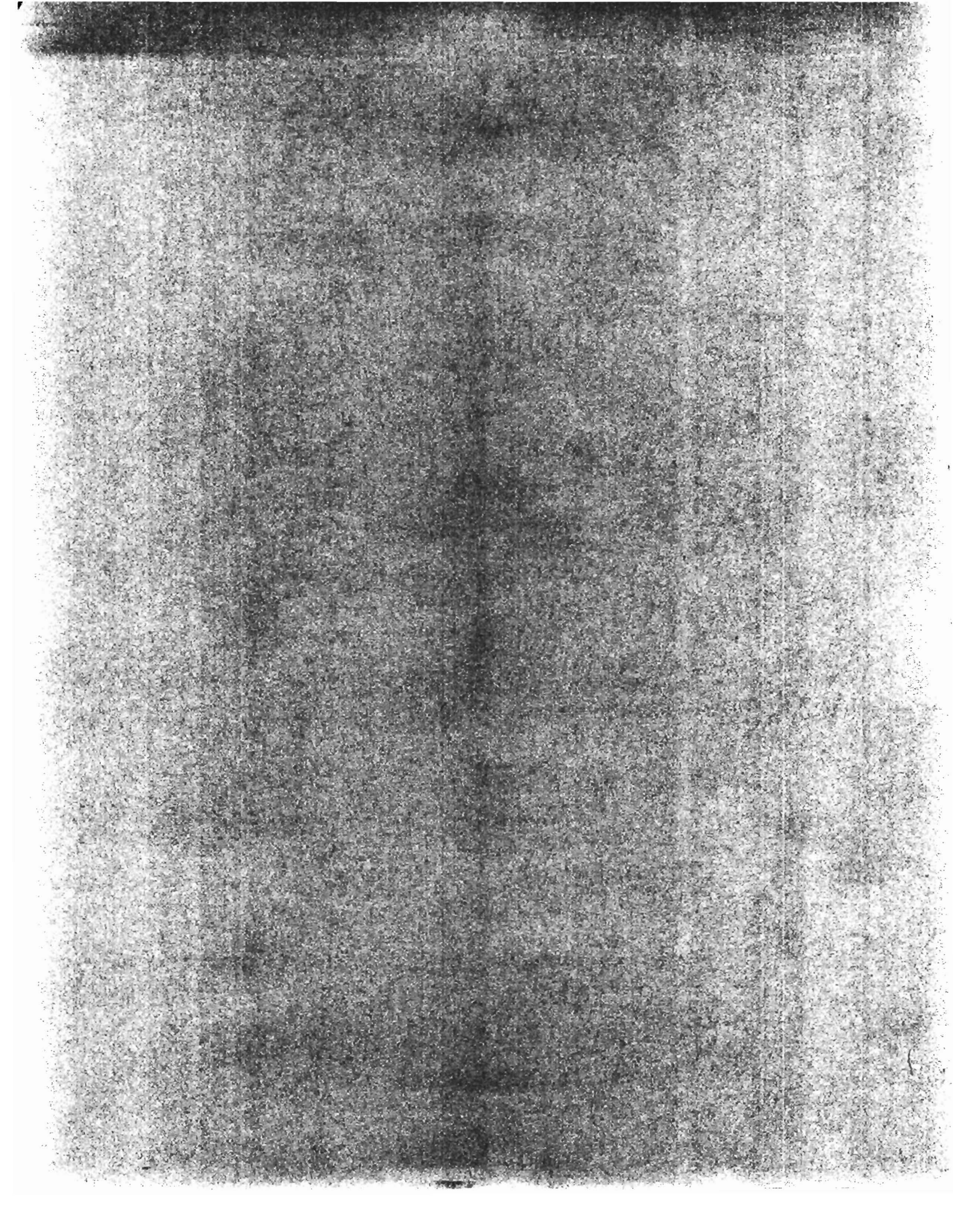


**S1-1**

**"Introduction & Conference Goals"**

**(Joel Butler - Fermilab)**

Conference schedule, access to terminals & other pertinent information. Overall goals of conference...network standards to be considered...list of current detectors/research projects investigating switching networks or rings.



## **International Data Acquisition Conference**

### **On Event Building & Event Data Readout**

#### **In Medium & High Energy Physics Experiments**

**October 26th - 28th, 1994**

**Fermilab**

**Batavia, Illinois**

### **Conference Overview**

- **No Parallel Sessions**
- **Oral Presentation Topics**
  - **System Requirements & Implementations**
  - **Tutorials ... Switching Networks & Network Standards**
    - **Data Acquisition R&D Activities**
    - **VME Standards Activities**
    - **Products ... ICs, Boards, Switches, Etc.**
    - **System Design ... Simulations**
    - **System Design ... Software**
  - **System Design ... Buffering, Queuing, Event Data Flow Control & Commercial/Non-Commercial Switching Network Comparison**
  - **System Design ... Panel Session**
  - **General Discussion With Attendees**
- **Conference Questionnaire:**
  - **Please fill out and return to Registration Desk by early Friday morning**

## Organizing Committee

Ed Barsotti	Fermilab
Mark Bowden	Fermilab
Sergio Cittolin	CERN
Robert Downing	University of Illinois
Jean-Pierre Dufey	CERN
Bill Haynes	Fermilab
Maribel Herrera	Fermilab
Marvin Johnson	Fermilab
Walter Knopf	Fermilab
Patrick LeDu	SACLAY
Livio Mapelli	CERN/LBL
Robert McLaren	CERN
Hans Muller	CERN
Masa Nomachi	KEK
Ruth Pordes	Fermilab
Paris Sphicas	MIT
Sonya Wright	Fermilab

### The Local Organizing Committee:

Elizabeth Brown	Fermilab
Denise Bumbar	Fermilab
John Elias	Fermilab
James Franzen	Fermilab
Cynthia Sazama	Fermilab
Colleen Yashikawa	Fermilab

## Tight Schedule

**Lights  
Questions  
Shut off!**

## Conference Schedule

- **Wednesday:**

- Morning Oral Sessions  
8:30 AM - 11:50AM
- Poster Presentations & Vendor Exhibits  
11:50AM - 1:30PM
- Afternoon Oral Sessions  
1:30PM - 6:05PM
- Wine & Cheese  
(With Poster Presentations & Vendor Exhibits)  
6:05PM - 7:30PM
- Dinner At Fermilab  
7:30PM

- **Thursday:**

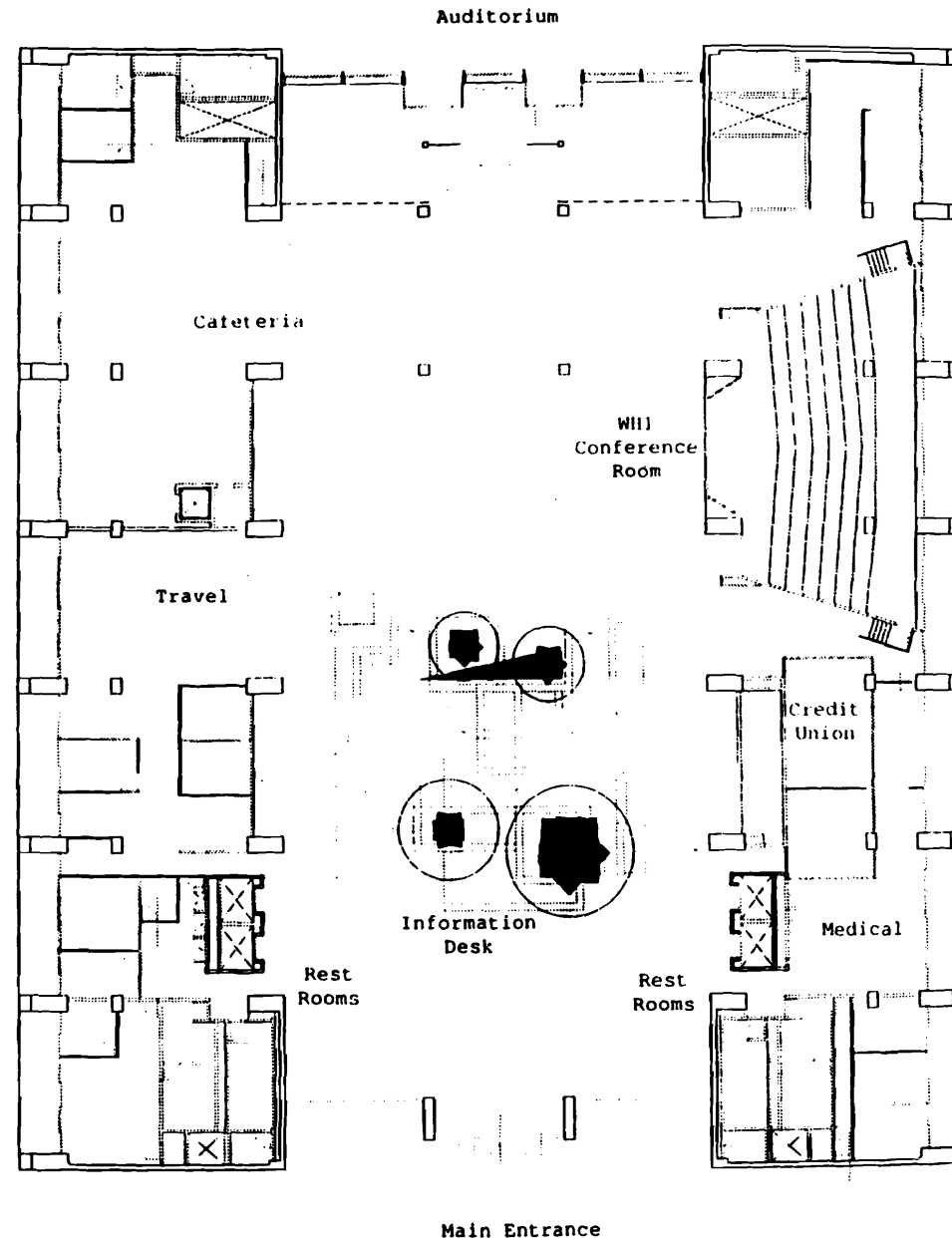
- Morning Oral Sessions  
8:30 AM - 12:20PM
- Poster Presentations & Vendor Exhibits  
12:15PM - 2:15PM
- Afternoon Oral Sessions  
2:15PM - 5:35PM
- Wine & Cheese  
(With Poster Presentations & Vendor Exhibits)  
5:35PM - 7:30PM
- Dinner At Galleon In St. Charles  
7:30PM

- **Friday:**

- Morning Oral Sessions  
8:30 AM - 12:15PM
- Poster Presentations & Vendor Exhibits  
12:20PM - 2:20PM
- Afternoon Oral Sessions  
(1 West Conference Room) (\*\*\*\*\* & overflow area\*\*\*\*\*)  
2:20PM - 5:45PM

# Friday Afternoon Meeting Room Change

Due to a scheduling conflict, we will meet for oral presentations in the 1 West Conference Room Friday Afternoon. This meeting room is located in Wilson Hall on the west side of the first floor, adjacent to the cafeteria. TV monitors will be setup in the cafeteria area for 1 West overflow Friday afternoon. All other oral presentations are as scheduled in Fermilab's auditorium.



### **Fermilab Cafeteria Hours:**

- **Breakfast: 07:30 - 10:15**
- **Lunch: 11:30 - 13:30**

### **Travel Department:**

- **Wilson Hall, east side of first floor**

### **Terminals:**

- **Wilson Hall, west side of Eight floor, near the elevators**

### **Transportation:**

- **Share a ride; see sign-up sheets at the Registration Desk**
- **Fermilab taxi service**
  - **Schedule (refer to the conference program)**

**For Assistance**

**Phone: 840-2915**

**FAX: 840-2783**

**Social Events:**

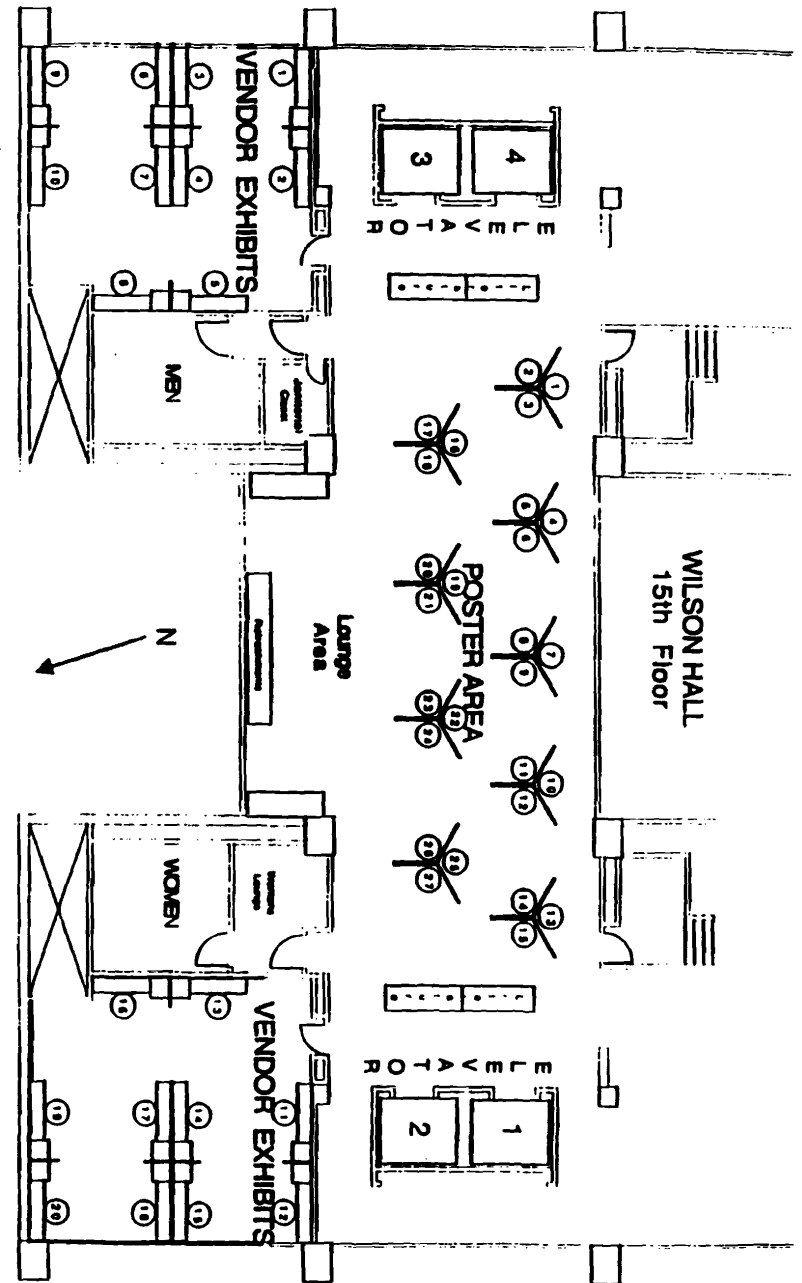
- **Wednesday:**
  - **Wine & Cheese, 15th floor of Wilson Hall immediately after the oral presentations  
(Poster presenters & vendors present)**
  - **Dinner, Fermilab cafeteria at 19:30**
  
- **Thursday:**
  - **Wine & Cheese, 15th floor of Wilson Hall immediately after the oral presentations  
(Poster presenters & vendors present)**
  - **Dinner, Galleon in St. Charles at 19:30**



## *Morning & Afternoon Breaks Wednesday, Thursday & Friday*

We ask that you NOT go to the poster/vendor area on the 15th floor during breaks. It takes far too long to get to and return from the poster/vendor area.

Poster presenters and vendors are not requested to be present on the 15th floor during breaks.



## Conference Proceedings

- Attendees only
- Old/new papers &/or transparencies
- Turn in old/new papers &/or transparencies at Registration Desk by early Friday AM

## Conference Questionnaire

1. On the basis of the following general characteristics, how would you rate this conference?

	poor				excellent
	1	2	3	4	5
Host Institute Fermilab	1	2	3	4	5
Poster/Vendor Selection of Posters	1	2	3	4	5
Oral Presentations Selection of topics	1	2	3	4	5
Selection of speakers	1	2	3	4	5
Panel Discussion Mix of panel members	1	2	3	4	5
Topics discussed	1	2	3	4	5
Accommodations (one hotel) Holiday Inn	1	2	3	4	5
Red Roof Inn	1	2	3	4	5
Social Events Dinner at Fermilab	1	2	3	4	5
Dinner at St. Charles Place	1	2	3	4	5
Wine & Cheese Reception	1	2	3	4	5
Conference Taxi Service	1	2	3	4	5

2. Did you have adequate time to visit the Poster/Vendor Area? Yes No

3. Was the Poster/Vendor area too isolated from the oral presentation area of the conference? Yes No

4. Should this conference be continued on an annual basis? Yes No  
a. If yes, select the next host institute. \_\_\_\_\_

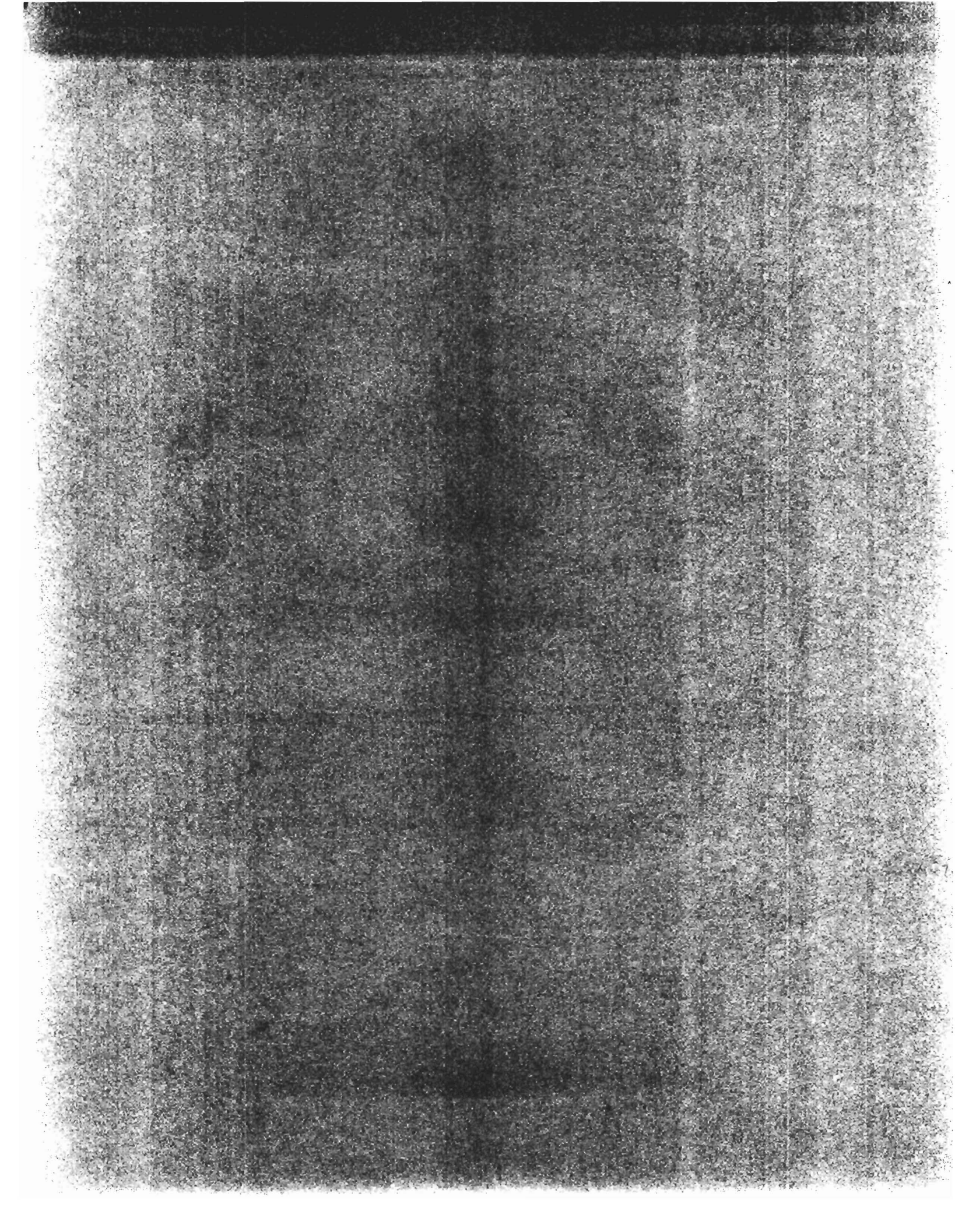
Comments: \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

**S1-2**

**"CDF & D0 Data Acquisition Systems"**

**(Paris Sphicas - MIT)**

Discussion of DAQ system requirements and architectures for CDF (experience using a small DAQ switch), CDF upgrades & D0.

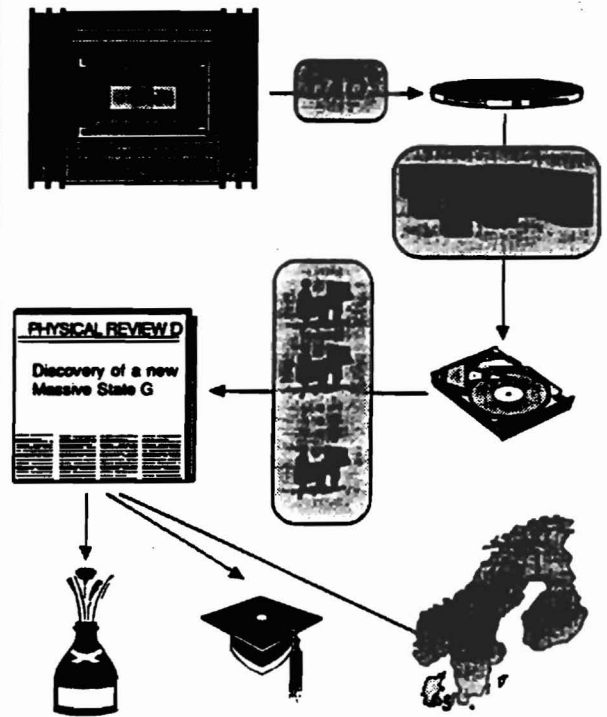


# CDF & D0 Data Acquisition Systems

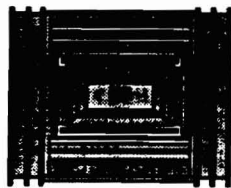
- Introduction, requirements
- History & Current Architecture
- Implementation
- Results
- Planned Upgrades
- Conclusion

Paris Sphicas/MIT  
Oct 26, 1994

## Introduction



## Requirements I



Interaction Rate:  
300 KHz



Level-1 Trigger:  
accept 0.3-5 kHz

Digiters, Front-End Memories



Level-1.5, 2 Trigger:  
accept 30-150 Hz

Digiters, Scanners, VBDs

To Processor Farm (Level 2/3)

## Requirements II

450 kBytes @ D0

250 kBytes @ CDF

Digiters, Scanners, VBDs

150 Hz @ D0

30 Hz @ CDF

Processor Farm (Level 2/3)

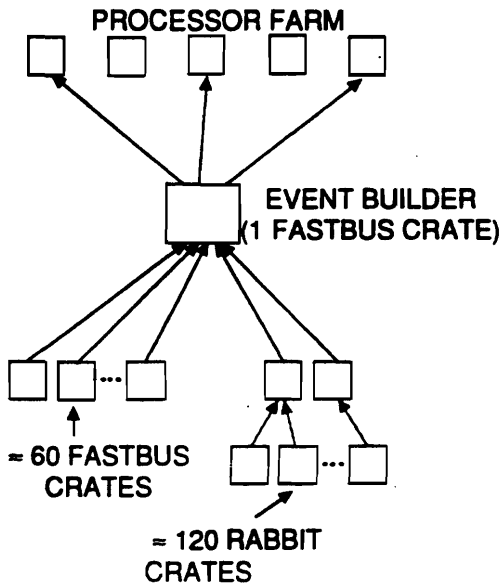
To Tape @  
~ 5 Hz

Monitors @  
= 1 Hz



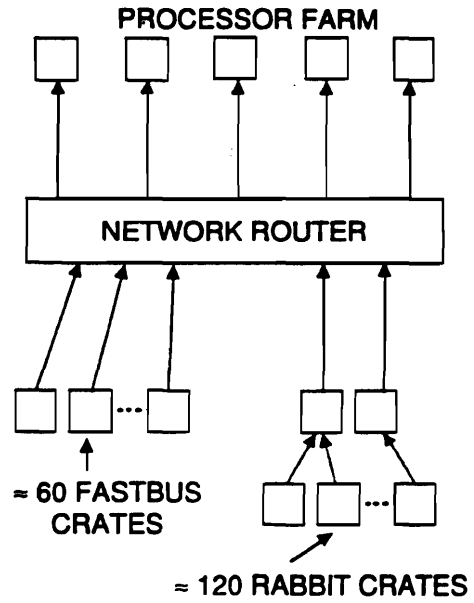
**History I: Old System (CDF)**

**Event Building Bottleneck**



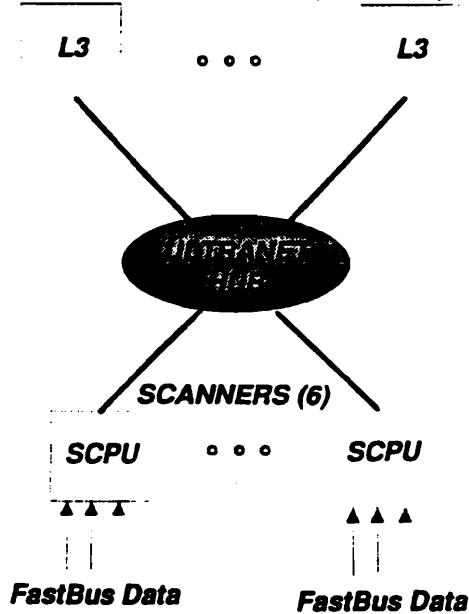
**History II: Option I (CDF)**

**Need a Large # of Ports**



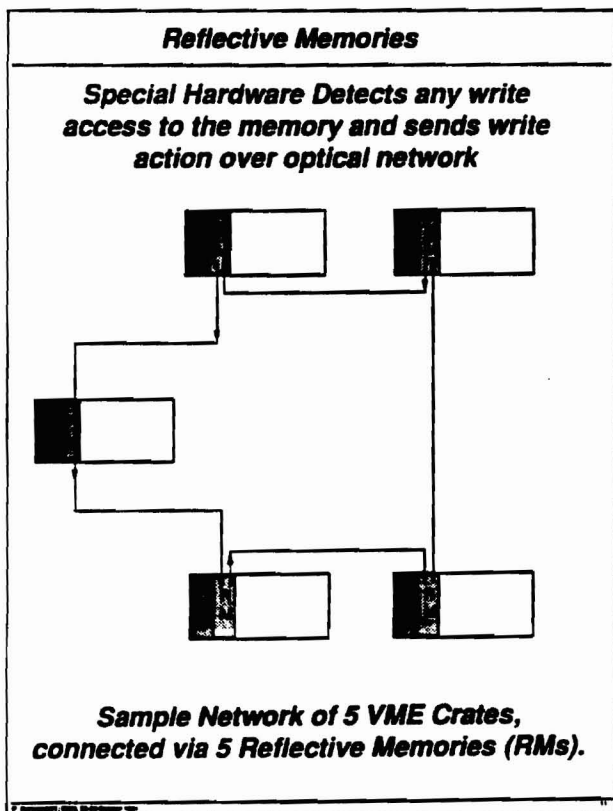
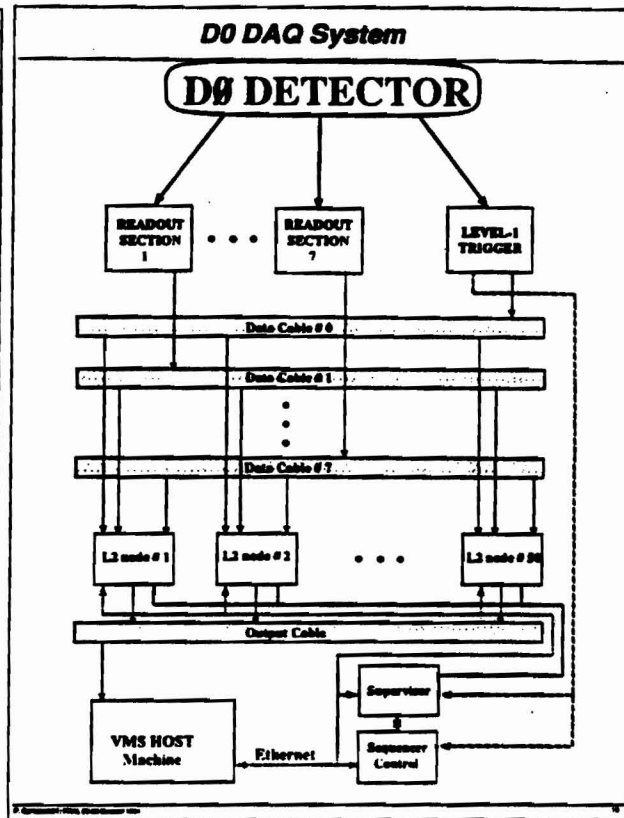
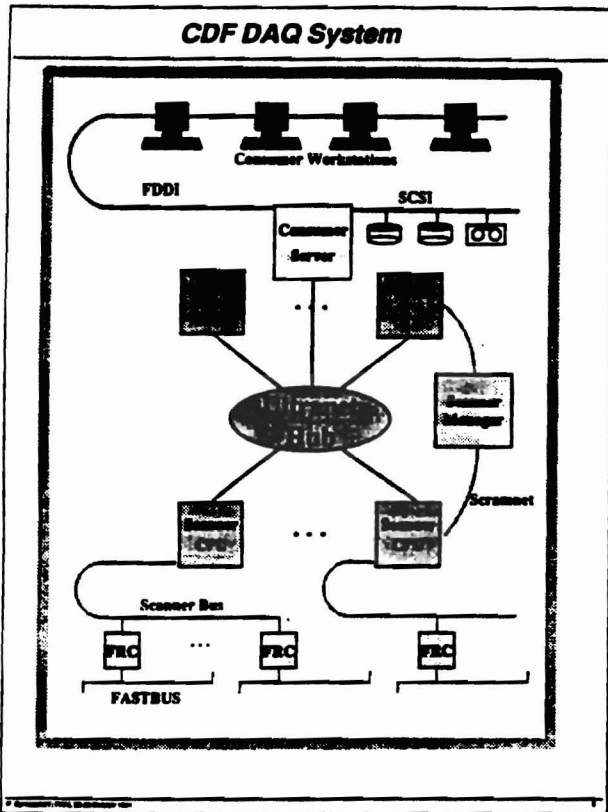
**History III: New Data Path**

**PROCESSOR FARM (8 NODES)**



**History IV: New Architecture**

- **Gather Fastbus Data into 6 VME crates**
- **Send Data to Processor Farm in unassembled form, let farm build the event.**
- **Use a commercial network for the data path between VME memories and SGI CPUs**
- **Use a centralized intelligence for Event Flow Control ⇒ Need fast control path, use Reflective Memories**



### Event Flow Control (CDF)

**CDF:**  
*Ultranet transfers data*  
*SCRAMNET transfers Control Information*

**D0:**  
*8 parallel cables transfer data*  
*Same cables (Token Ring) for Control*

**Link Performance:**

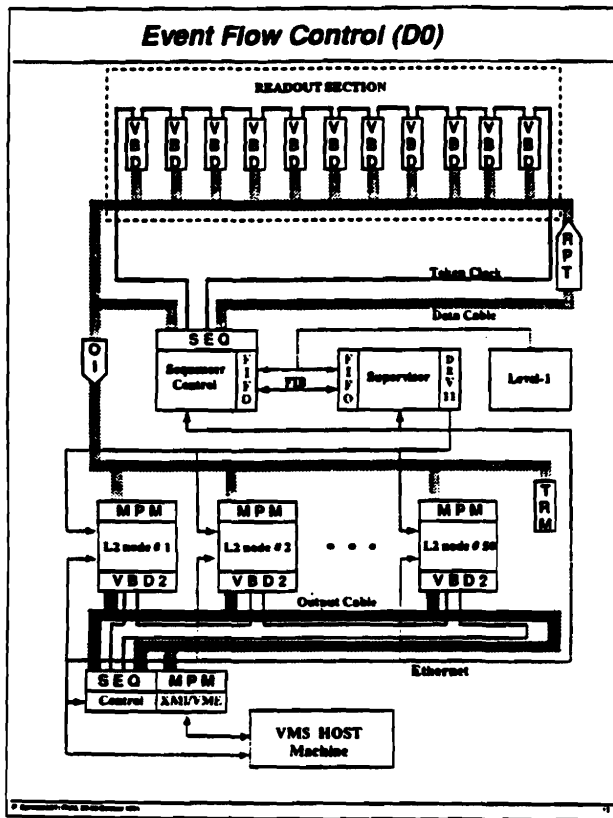
**CDF:**  
*point-to-point @ 13.5 MB/sec*

**D0:**  
*serial links, @ 40-48 MB/sec*

**Parallelism:**

**CDF:**  
*utilize "concurrent" event fragment sending*

**D0:**  
*none: links are fast enough*



### Level-3 Trigger (Processor Farm)

**CDF: (Level-3 Trigger, 50 Hz → 5 Hz)**  
**UNIX; SGI (4 X 4D/80 + 4X Challenge XL)**

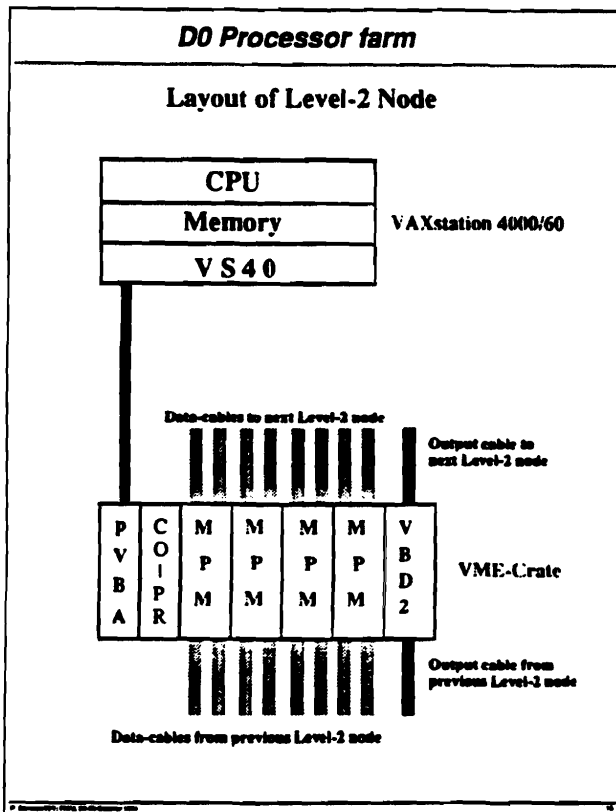
**D0: (Level-2 Trigger, 150 Hz → 5 Hz)**  
**Vaxeln; = 50 Vax 4000-60**

**Common Functions:**

- **Receives event fragments from SCPUs**
- **Builds events out of fragments**
- **Creates detector banks**
- **Runs (special) trigger algorithms based on offline reconstruction code**
- **Sends accepted events to Server**
- **No Central L3 Control**

**Differences:**

- **Event Building:**  
 6 → 1 @ CDF, 32 → 1 @ D0
- **Input Rate @ CDF: depends on L2**
- **Input Rate @ D0: constant at 150 Hz**



### Ultrahnet (CDF)

- **Proprietary high-performance network**
- **Point to Point Links via a central hub**
- **Serial Links into/out of hub: 250 Mbit/sec**
- **Maximum Speed: 1 Gbit/sec**
- **Host adapters for VME CPUs**
- **Software drivers for VxWorks, IRIX**
- **Peak VME transfer speed: 13 MB/sec**
- **TCP/IP protocol**
- **Standard Berkeley sockets**

Possible Data Paths

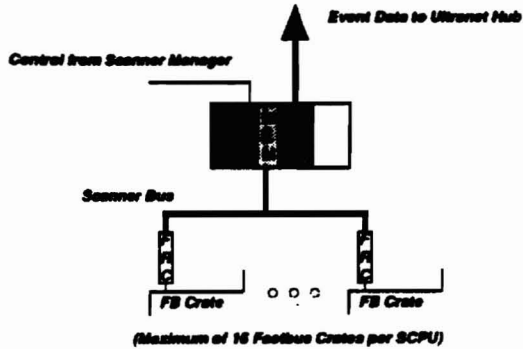
Link Adapter

4 links



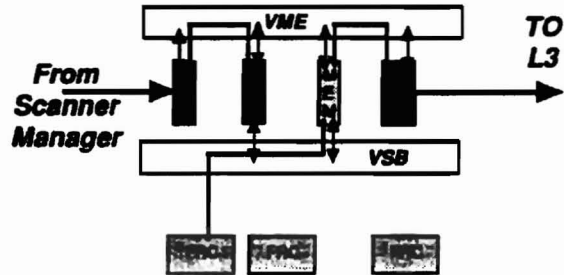
### Scanners I (CDF)

**FRC:** Fastbus Readout Controller  
**RM:** Reflective Memory  
**SCPU:** Scanner CPU  
**MEM:** Dual Ported Memory (VSB-VME) for event data  
**UHA:** Ultramet Host adapter



### Data and Control Flow (CDF)

———— Data Flow  
 ———— Control Information Flow



### Results

	CDF	DO
Max. Input Rate to L3	110	240
Actual Rate (Physics)	30	150
Limited by	L3 CPUs	L1.5 Trig

#### Example:

#### DO at low luminosity:

L1 accept: 600-700 Hz  
 L1.5 accept: 150 Hz  
 L2 accept: 3 Hz

#### DO at high luminosity:

L1 accept: 350 Hz  
 L1.5 accept: 150 Hz  
 L2 accept: 3 Hz

### Planned Upgrades (I)

(a) Both CDF & DO plan major upgrades to their trigger systems.

(b) Also associated changes to (parts of) the readout system

(c) No (major) changes to the event building schemes are anticipated

#### Way to Achieve (c):

DO: event size will decrease:  
 450 Kbytes → 130-130 KBytes

CDF: add more scanners for new detector elements, thus increasing the total throughput via Ultramet

## **Conclusion**

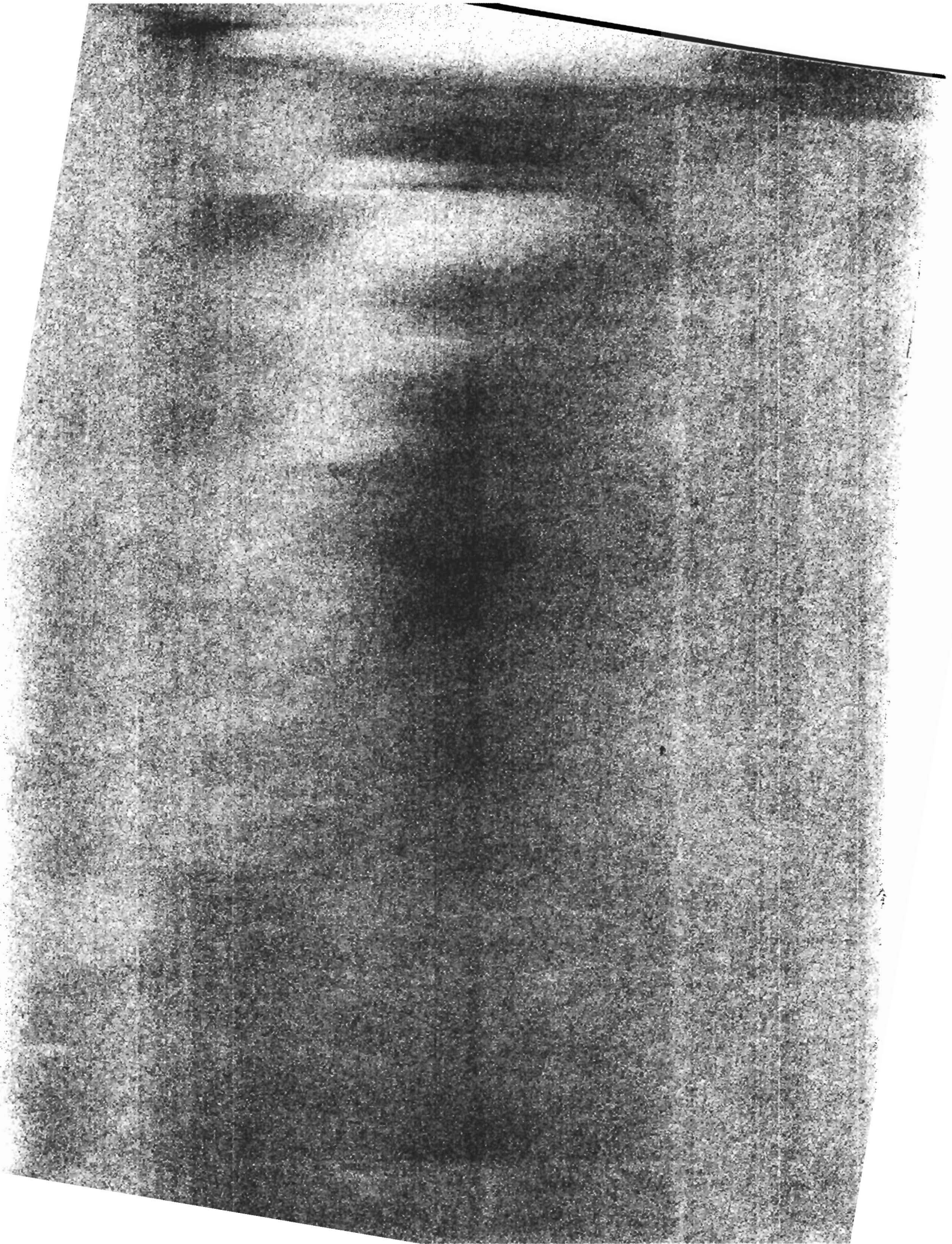
- **Problem solved by CDF&D0 via:**
  - **L1 Trigger: 300 kHz → ~ 1 kHz**
  - **CDF: L2 Trigger: → ~ 30 Hz**
  - **D0: L1.5 Trigger: → ~ 150 Hz**
  - **Processor Farm: → ~ 3-5 Hz**
  
- **Two ways to send the subevents to a single intelligence:**
  - **ULTRANET : (logical) switch**
  - **Parallel cables with high throughput**
  
- **Control is done via ring "networks":**
  - **Reflective Memories**
  - **Token Ring (special hardware)**
  
- **CDF & D0 are taking data, doing physics and publishing...**

**S1-3**

**SLAC & KEK B Detector Data Acquisition Systems"**

**(Walt Innes - SLAC)**

Discussion of DAQ system requirements and architectures for SLAC and KEK B detectors.





## **B Factory Data Acquisition, BELLE and BABAR**

*Walt INNES*

*SLAC*

*including contributions from*

*Masa NOMACHI*

*KEK*



## **DAQ Environment at PEP II**

- *Double ring asymmetric machine: 9 GeV on 3.1 GeV.*
- *High Luminosity -> Crossing Period: 4.2 ns  
For DAQ purposes this is continuous*
- *"Physics" Rate: 30 to 100 Hz*
- *Physics event size: 25 kilobytes*
- 

### **Backgrounds**

- *Synchrotron radiation is well controlled*
- *Non-local lost particle background is collimated*
- *Dominant background source is beam particles which  
interact with the residual gas between 3 and 50~m  
from the IP.*
- *EM showers of these lost particles cause occupancy.*
- *Electro-production causes triggers.*

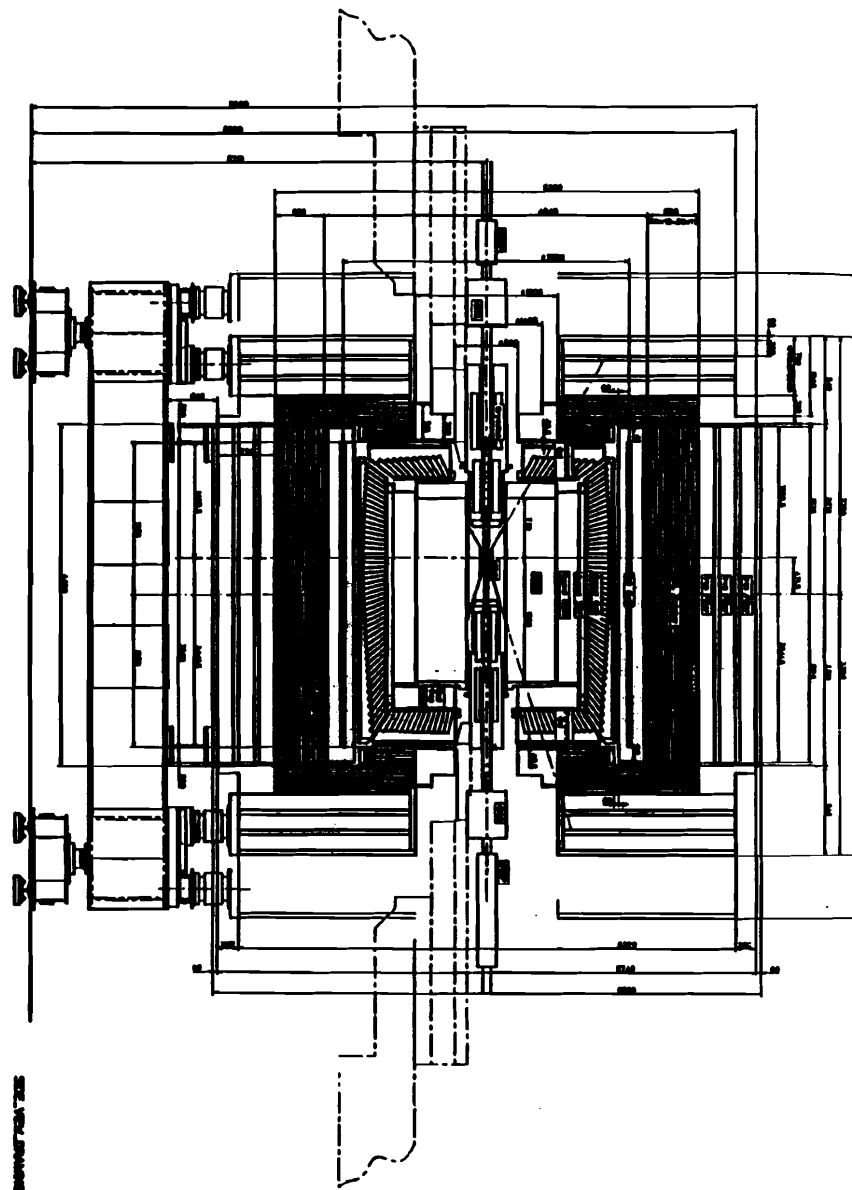
## BELLE DAQ Statistics

subsystem	No. of channels	words per event
Si vertex det.	100 k	2000
Drift ch.	10 k (A and T)	1500
Cherenkov c.*	2 k (A)	200
TOF c.	500 (A and T)	200
Csl cal.	10 k (A)	2000
KL/mu ch.	5 k (A and T)	300
Others		1000

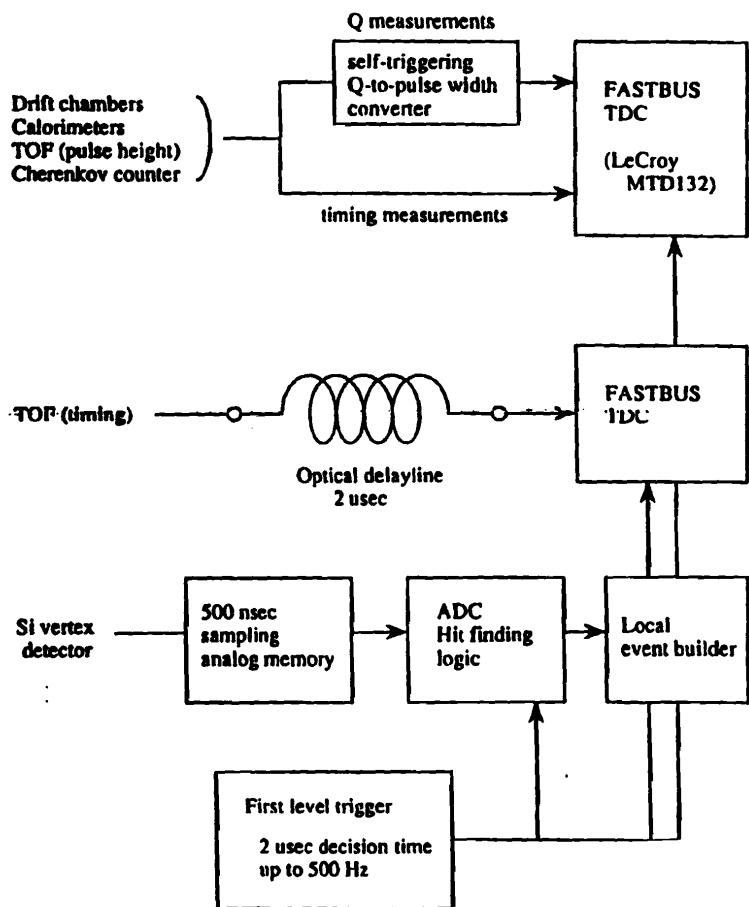
total approx. 30kB/ev.

\* Selection of PID device has not been done.

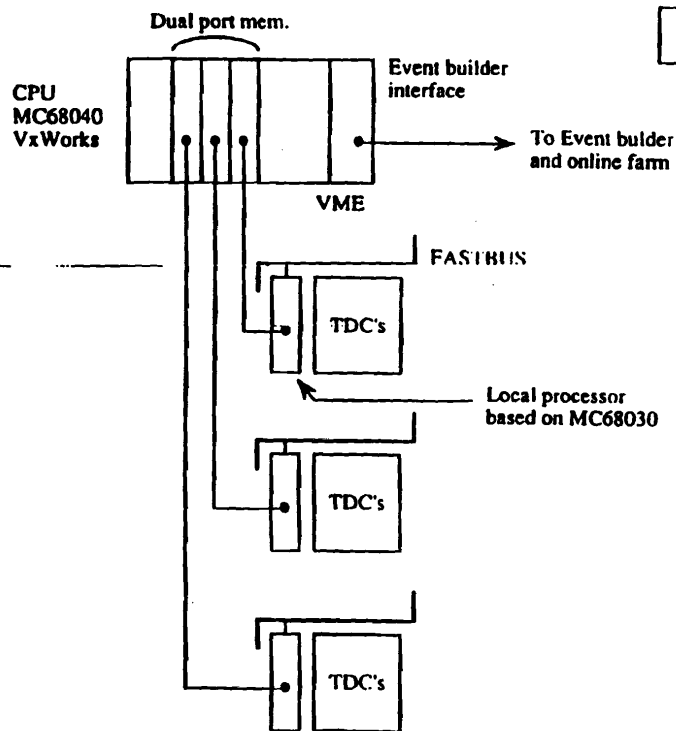
Beam cross rate	508 MHz
Max. beam current	1.1A (HER) + 2.6A (LER)
Max. luminosity	$1 \times 10^{34}$ /cm <sup>2</sup> /sec
Max. trigger rate (L1)	500 Hz
Max. data flow rate	15 MB/sec
Max. data rate on tape	10 MB/sec



### Digitization Scheme



### Readout Scheme



x 8



Question: Given that we must have a hardware trigger, should we make it sophisticated enough to do the complete level 1 trigger?

This would require good tower sums in the calorimeter, and a good Pt measurement for tracks in the DC. Fair z pointing would also be useful.

If there is a processor based level 1 trigger, how do the trigger primitives get to the processors? It could be done via dedicated path or via the same LAN as the DAQ.

Using the LAN is simpler, but requires a LAN which can handle a high rate of small transactions.



## Event Building

Question: Do we build an event for every Level 1 trigger?

The alternative is for the level 2 processor to ask for a small portion of the data and use that to refine the trigger decision.

- *Advantage: Lower load on the DAQ board processors which have to extract the features from the data.*
- *Disadvantage: A higher transaction rate on the LAN, less efficient use of the LAN, and a more complicated data flow architecture.*

Question: What LAN to use? This is bound up with the questions above.

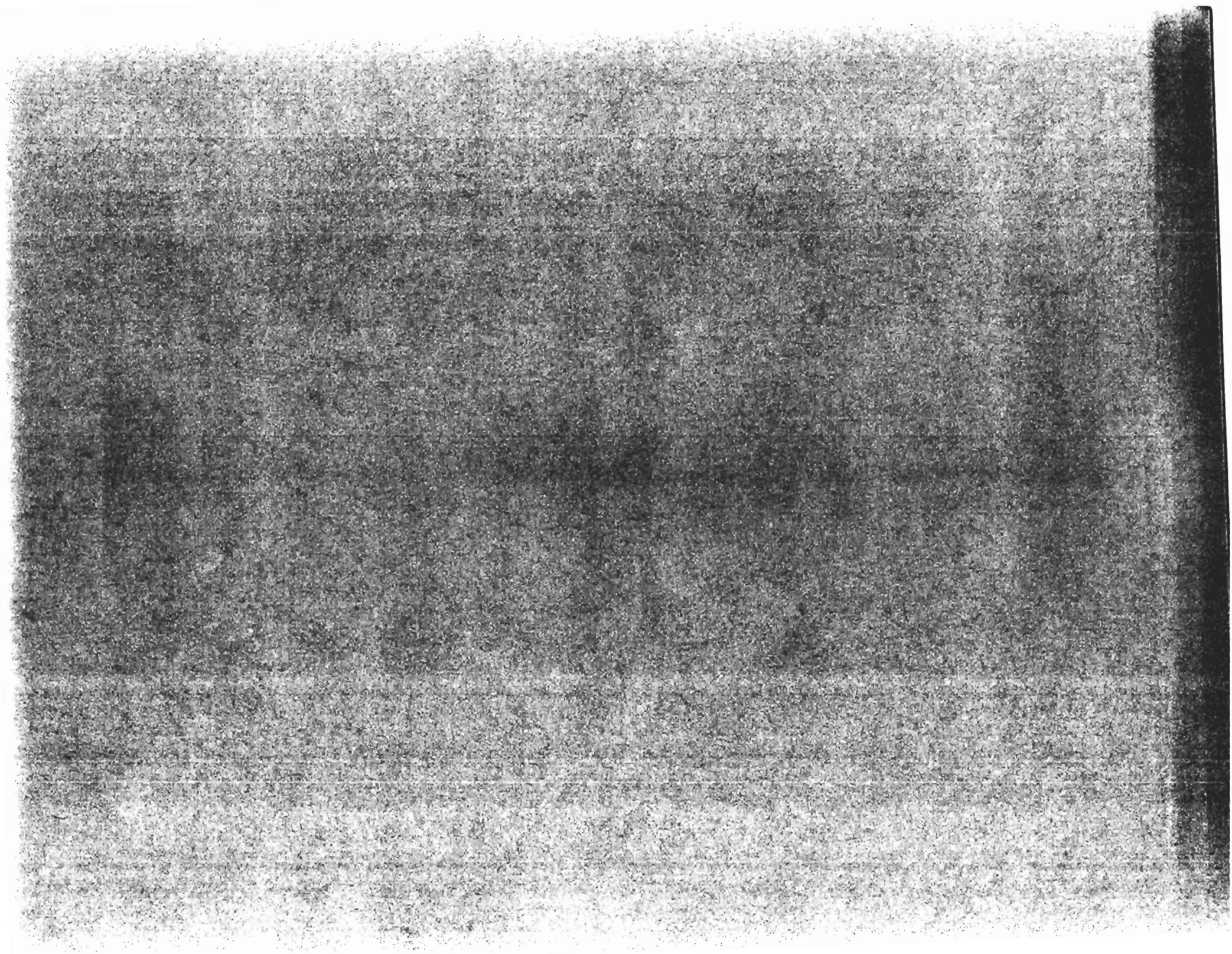


**S1-4**

**"STAR Detector Data Acquisition System"**

**(Mike Levine - BNL)**

**Discussion of DAQ system requirements and proposed architecture for STAR.**



# STAR Data Acquisition

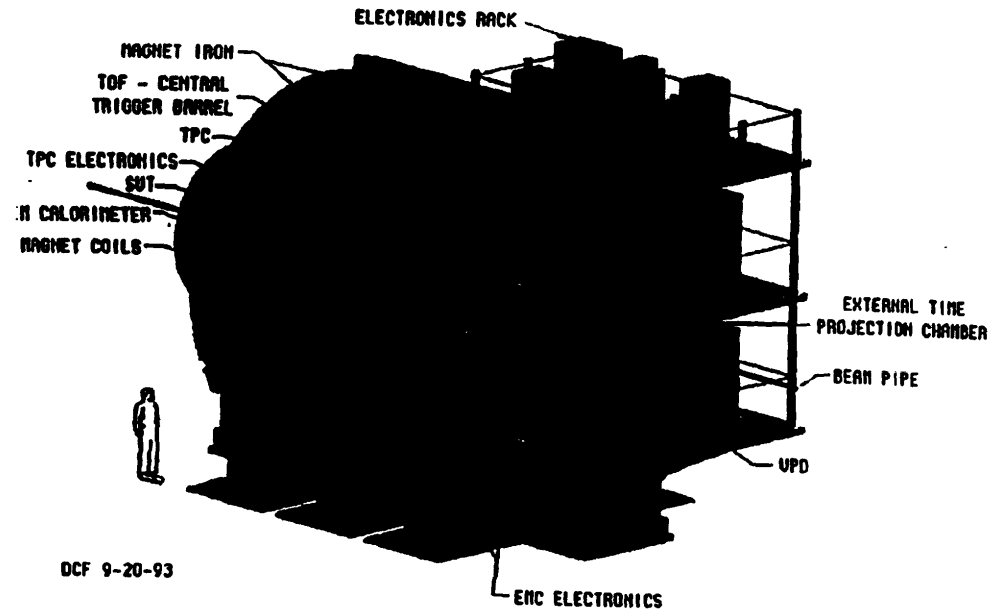
M. J. LeVine

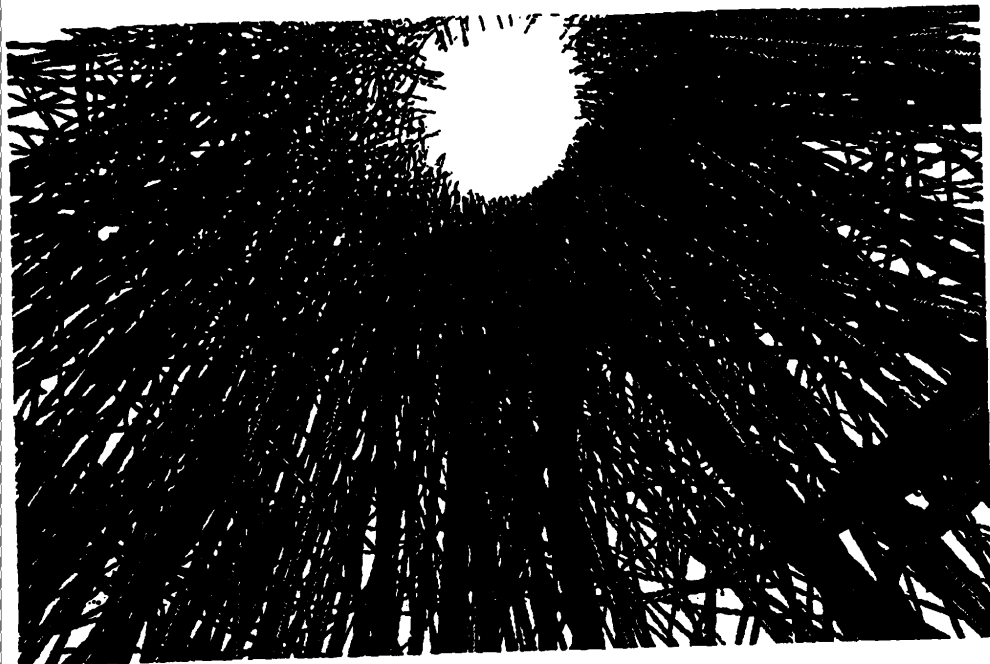
Brookhaven National Laboratory

## Overview

- the STAR detector
- event sizes/rates
- processing requirements
- architecture

## STAR DETECTOR





## STAR DETECTOR

Subdetector	# channels	status
TPC	140 000	funded
TRG	240	funded
SVT	103 000	R&D
EMC	32 000	future
XPTC	22 000	future
TOF	8 000	future

## Event size (TPC only)

# pads	140 000	
time samples/pad	1024	8 bits, compressed
total "pixels"	135 M	

### Occupancy

outer pad rows	5 %
inner pad rows	15%
average	7%

### After pedestal suppression

encoding overhead	~30%
9.5 MByte + overhead	13 MByte

## Event Rates

Beam crossings - 9 MHz  
Input - up to 1000/s (physics)  
Output - ~ 1/s (tape limited)

# Triggers

## Level 0

Generated by beam crossing

- output rate 9 MHz
- decision time 0

## Level 1

Uses preliminary information from TRG detectors

- input rate 9 MHz
- pipelined
- output rate < 2 kHz
- decision time ~ 1  $\mu$ sec
- opens TPC gating grid
- starts writing into SCAs

## Level 2

Uses preliminary information from TRG detectors

- input rate < 2kHz
- output rate < 100 Hz
- decision time < 10 msec
- causes digitization cycle to abort

## Level 3

Uses tracking information from TPC

- input rate < 100 Hz
- output rate < 1 Hz
- decision time < 40 msec
- causes event builder to discard/retain event

## Distribution on a TPC sector

### # receiver boards:

outer	4
inner	2

### # pads:

outer	3940
inner	1750

### Au-Au central collision:

# hits/receiver board (outer)	3.6K
# pixels/receiver board (outer)	43.0K
# centroids	1.2K
hit summary info (@20 bytes)	24.0K bytes

## Level 3 processing

**All processing following digitization**

**Pedestal (zero) suppression**

**"Quick" hit finding**

- isolate data belonging to one hit
- use centroid as space point
- fails where hits overlap
- choose TPC region where this works  
(outer 16 pad rows)

**Tracking based on hits found**

**Physics cuts based on tracks found**

## Dataflow scenario

**Background facts -**

Data produced: Typically 50 kB/ receiver board  
Assume 100 CPUs required  
to perform Level 3 in 10 msec

**Distributed Level 3**

Buffer few events on receiver boards  
(few X 50 KB each board)

Transport 4 receiver boards/sector to 4 CPUs/sector

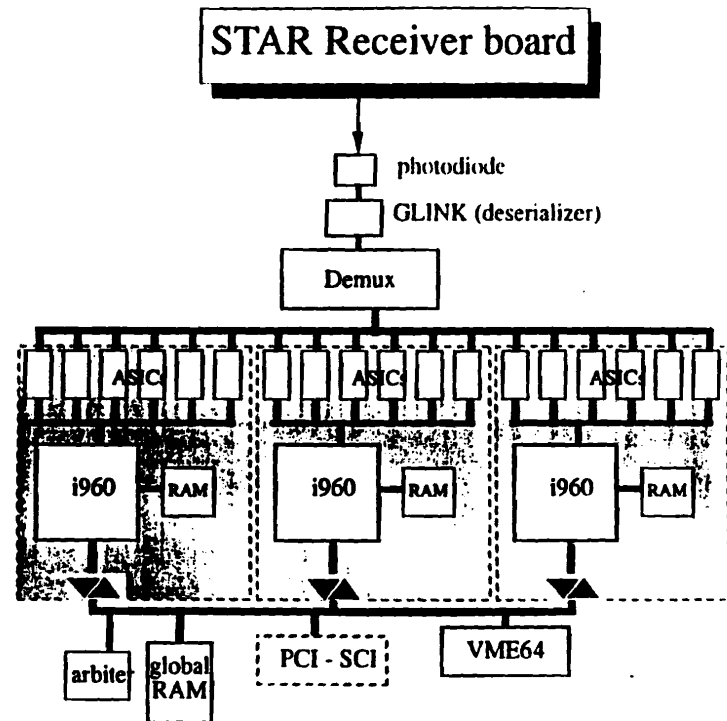
Transport remaining data only for accepted events

Perform level 3 decision in 100 CPUs—

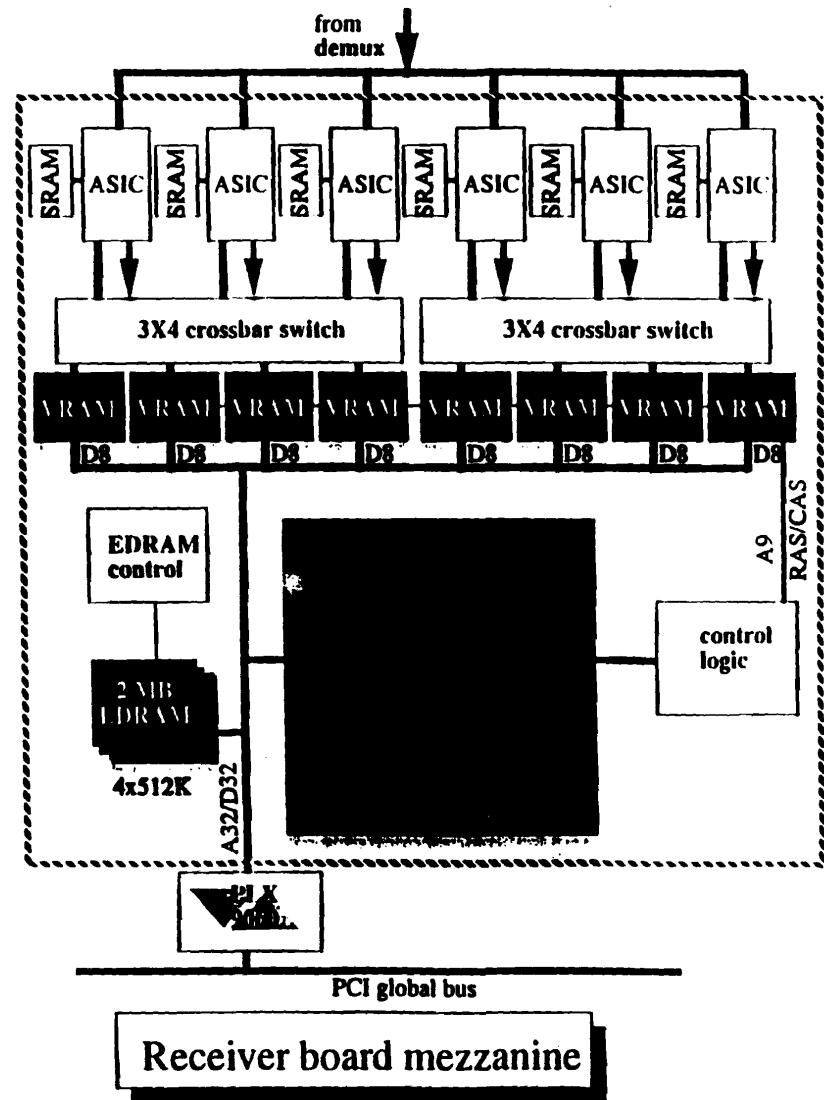
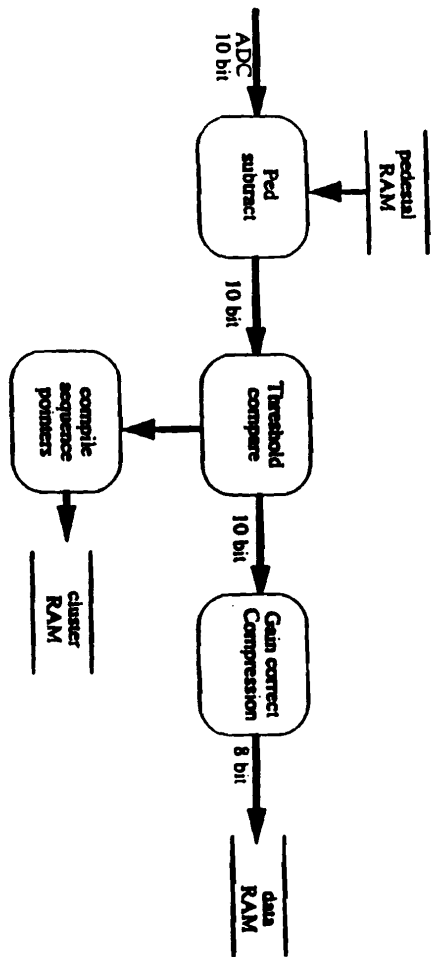
- RAM required - 1 GB (7 unsuppressed events)
- Aggregate bandwidth required - 240 MByte/s
- Local (CPU) bandwidth required - 10 MByte/s
- Sector bandwidth required - 10 MByte/s

## STAR Architecture (baseline)

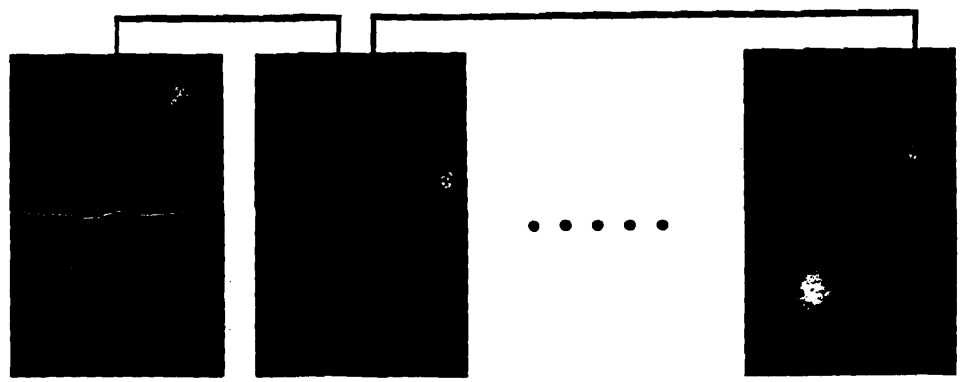
**TPC receiver board**      **PCI**  
**TPC sector**              **VME64 backplane**  
**TPC sector interconnect**    **SCI**



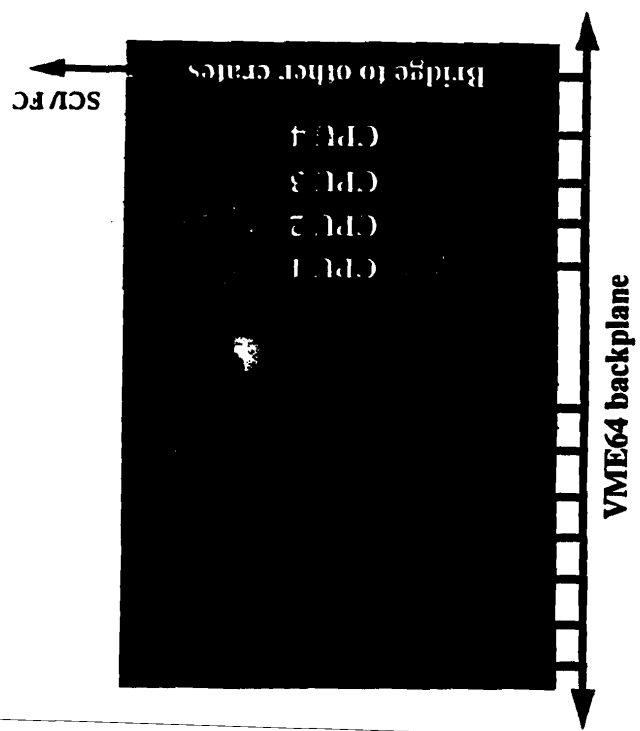




DAQ hardware organization



DAQ sector crate

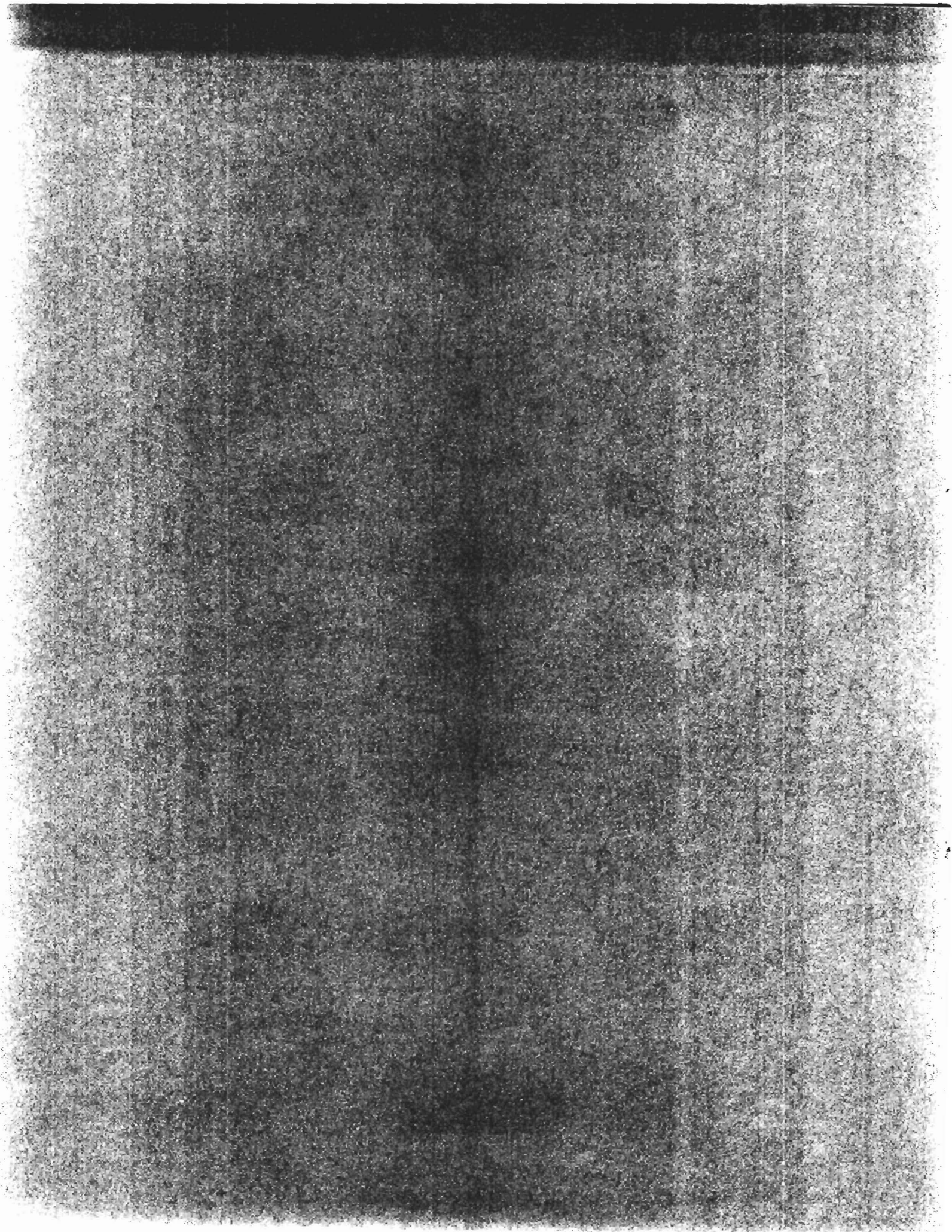


## **S1-5**

### **"PHENIX Detector Data Acquisition System"**

**(Cheng-Yi Chi - Nevis)**

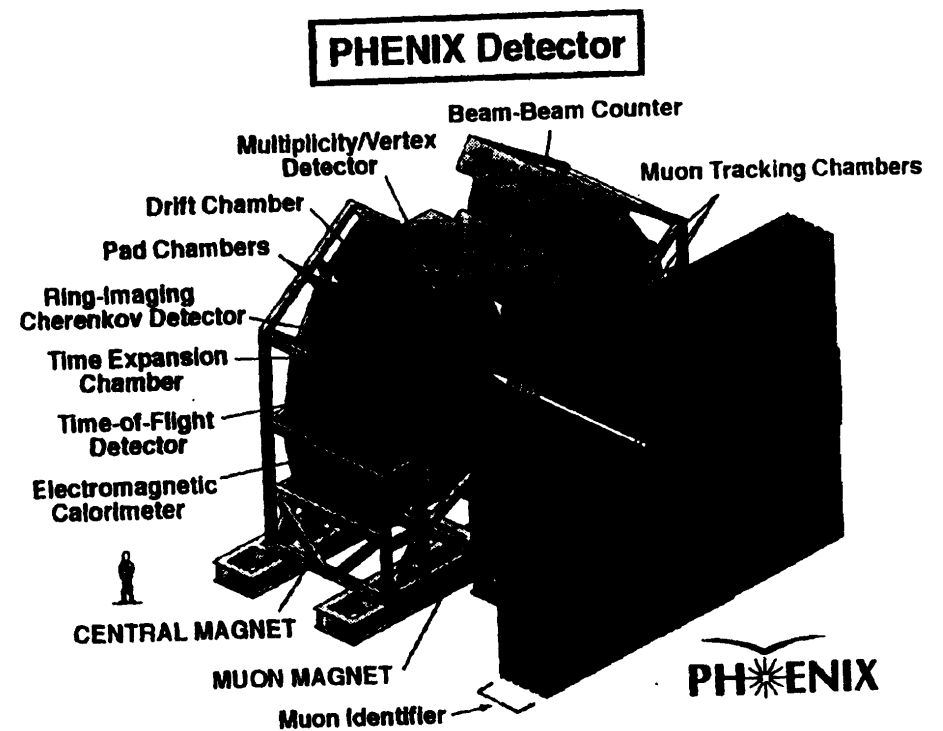
PHENIX is one of the two large experiments for Brookhaven National Laboratory's Relativistic Heavy Ion Collider (RHIC). RHIC produces a complex environment for the online system due to the fast beam clock, the high track density for heavy ion collisions and the large interaction rate in the case of proton collisions. To meet this challenge, the PHENIX online system is based on analog memory, flash ADC's and/or TDC's system at the front-end, DSP+glue logic for zero suppression and signal correction on digitized data at the data collection stage, and a high bandwidth event builder+trigger at higher level. The system is fully pipelined, data-driven and deadline-less.



# PHENIX DETECTOR DATA ACQUISITION SYSTEM

By Cheng-Yi Chi  
(Nevis Lab)  
(Columbia University)

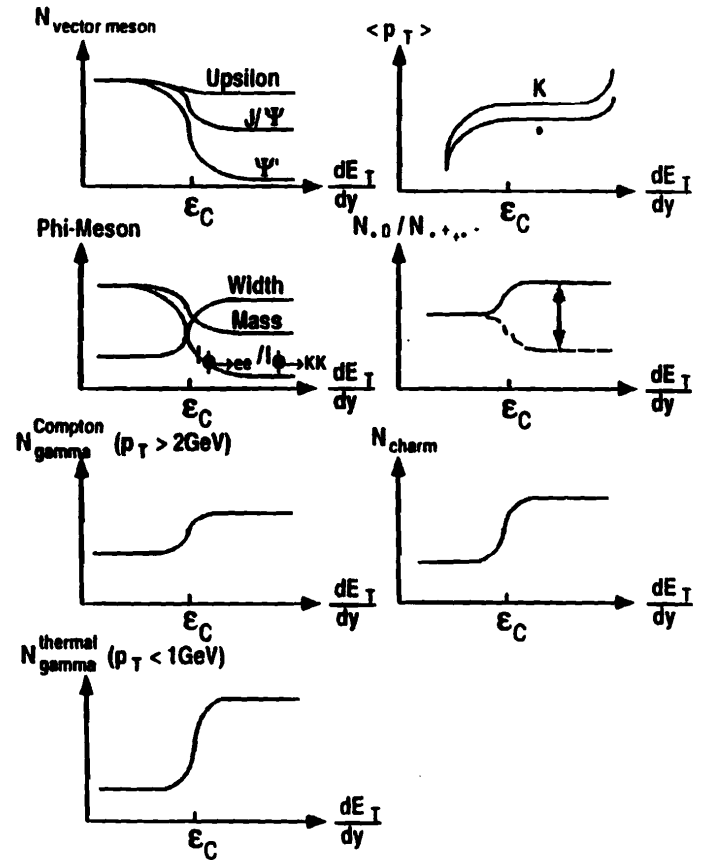
For PHENIX On-line Group



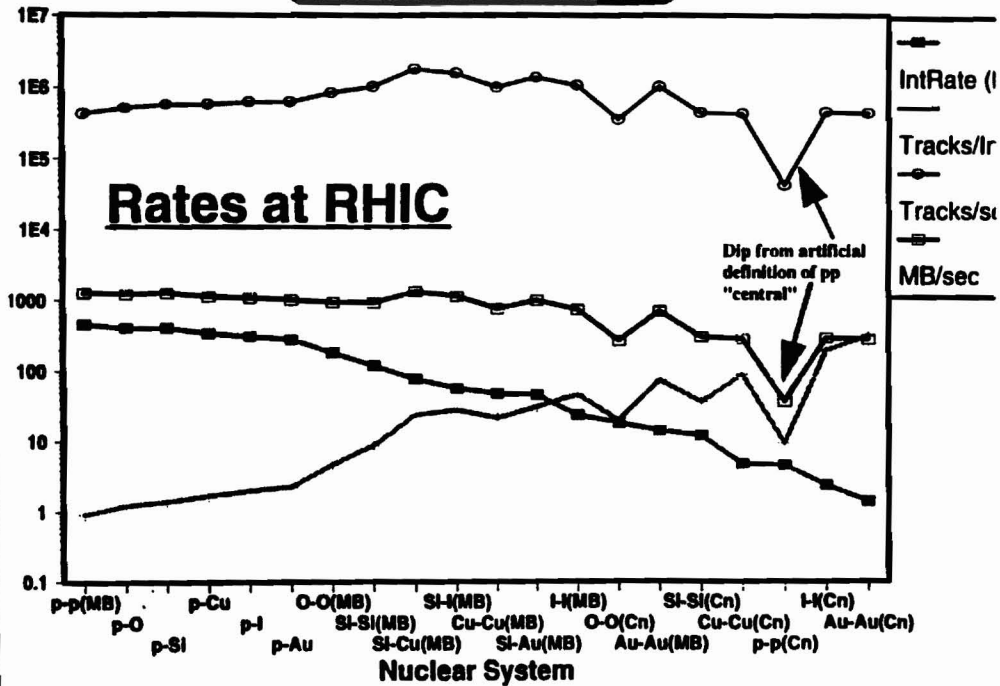
**PHENIX Collaboration**

Brasil: (3)		Russia: (102)	
U. Sao Paolo	3	IHEP-Protvino	28
		INR-Moscow	7
Canada: (9)		ITEP-Moscow	8
McGill U.	9	JINR-Dubna	23
		Kurchatov Inst.	9
China: (33)		PNPI - St. Petersburg	26
CIAE	13	Individual	1
IHEP-Beijing	8		
Inst. Mod. Phys.	8	Sweden: (8)	
Peking U.	4	Lund U.	8
Germany: (7)		U. S. A.: (151)	
U. Muenster	7	U. Alabama	4
		BNL	25
India: (6)		UC-Riverside	5
BARC-Bombay	6	Columbia U.	14
		Florida State	3
Japan: (45)		Georgia State	3
Hiroshima U.	8	Idaho NEL	4
INS, U. Tokyo	5	Iowa State/Ames Lab.	9
KEK	9	LLNL	10
Kyoto U.	2	LANL	16
Nagasaki	2	Lousiana State	4
Nat. Inst. Rad. Sci.	1	MIT	6
U. Tokyo	6	SUNY - Stony Brook	12
Tokyo U. Agr. Tech.	1	ORNL	17
U. Tsukuba	11	U. Tennessee	4
		Vanderbilt U.	4
		Yale U.	9
Korea: (12)		Individual	2
Chung-ang U.	1		
Korea U.	4		
Seoul Nat. U.	6		
Soong-Sil U.	1		
		<b>Total</b>	<b>376</b>

**Potential Signatures of Quark-Gluon Plasma**



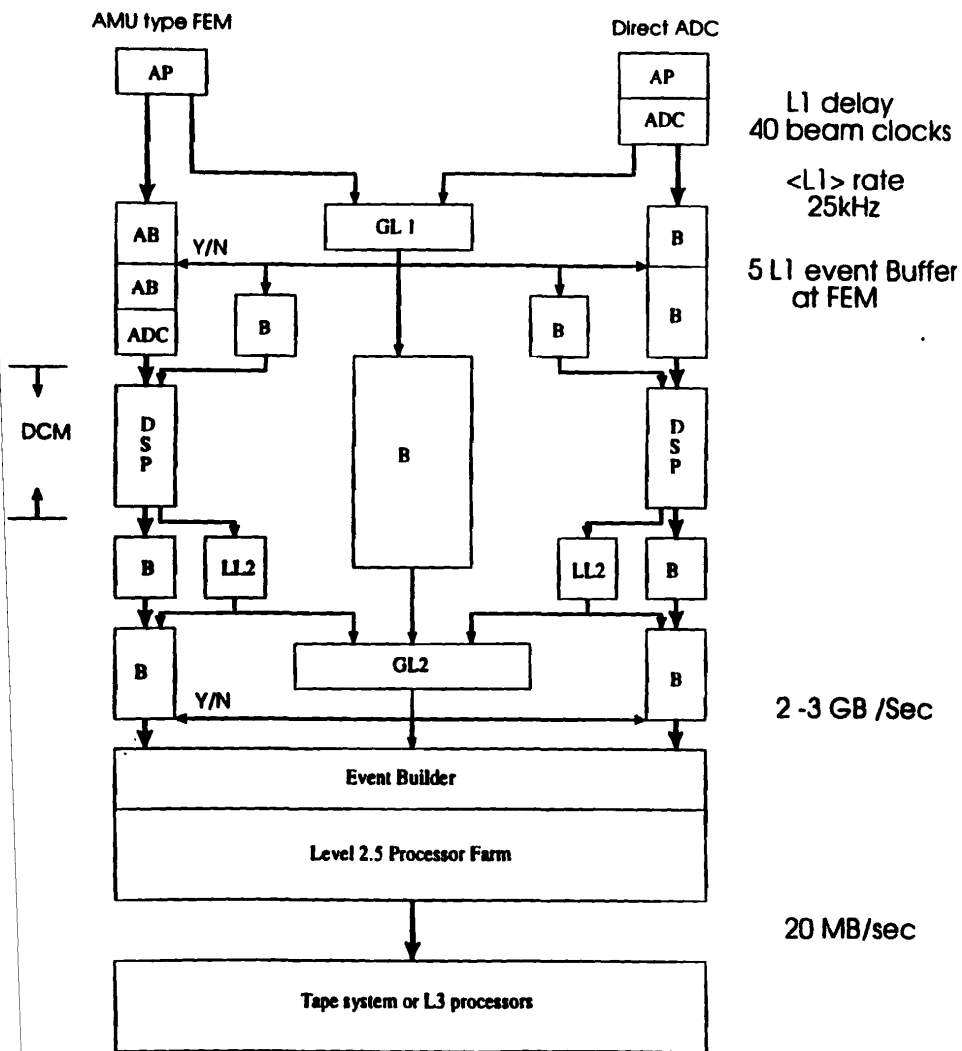
# PHENIX DAQ



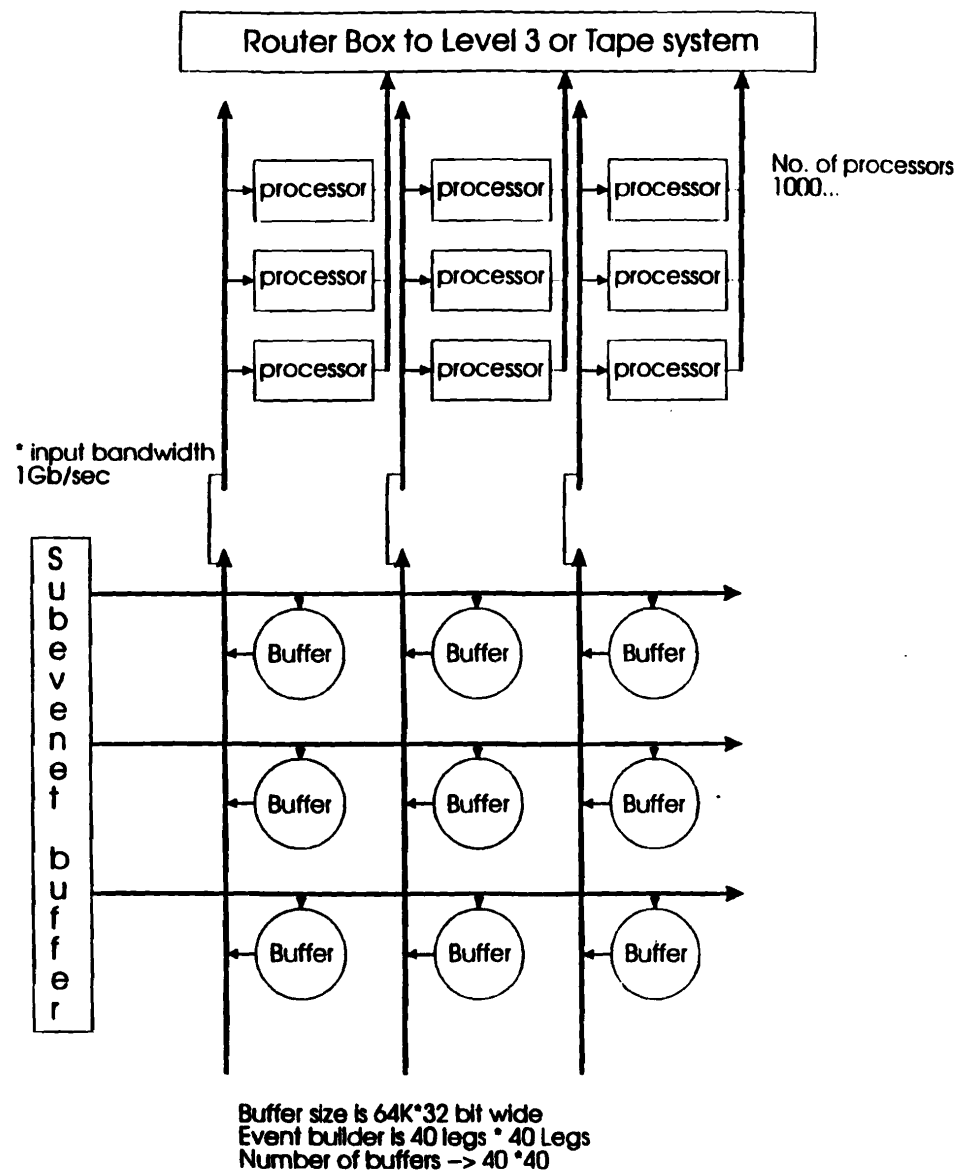
## Key Design Issues:

- (A) ~200K channel counts
- (B) 112ns beam Clock  
 Max. <L1> trigger rate = 25 KHz  
 ==> Similar to commercial/custom ADC/TDC Speed .  
 ==> Front-end system need to be Pipeline + Simultaneous R&W  
 ==> AMU type system need AX+B(cells) correction
- (C) System has to be I/O & Processing Efficient.  
 ==> Data Driven Principle (Data itself contain enough information to be processed )  
 ==> Control, Monitor, Main Data Flow are orthogonal to each others
- (D) Fully corrected data needed after Level 1 trigger.
- (E) Data taking at Year 1999...  
 ==> Base on the existing technologies and trends  
 Make system scaleable and easy upgraded.

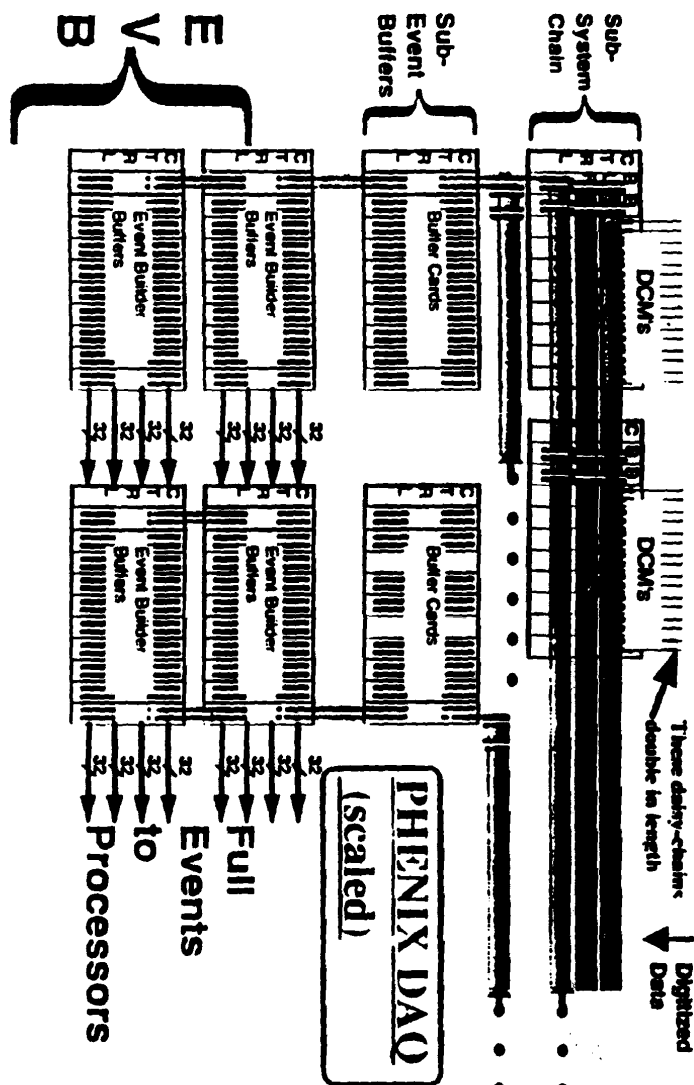
### PHENIX Online System Overview



### Event Builder







## Conclusions/Outlook....

- (1) PHENIX DAQ system's design is driven by the physics need.
- (2) The design of the system have the following characters:
  - (a) Fully pipelined, data driven
  - (b) Scalable
  - (c) Combining FPGA+Buffer+Processors(DSP) to achieve both Flexible and Efficient means of processing.
  - (d) Using both "ROI" and Processor Farm architecture to accommodate the complex RHIC env.
- (3) Processing needs at Data Collection Module level are driven by the detectors.  
Event Builder are driven by the Physics  
How one can keep EVB expandable and scaleable???
- (3) The design still in its preliminary stage. How one integrate/control/monitor the overall system will be one of the biggest challenges

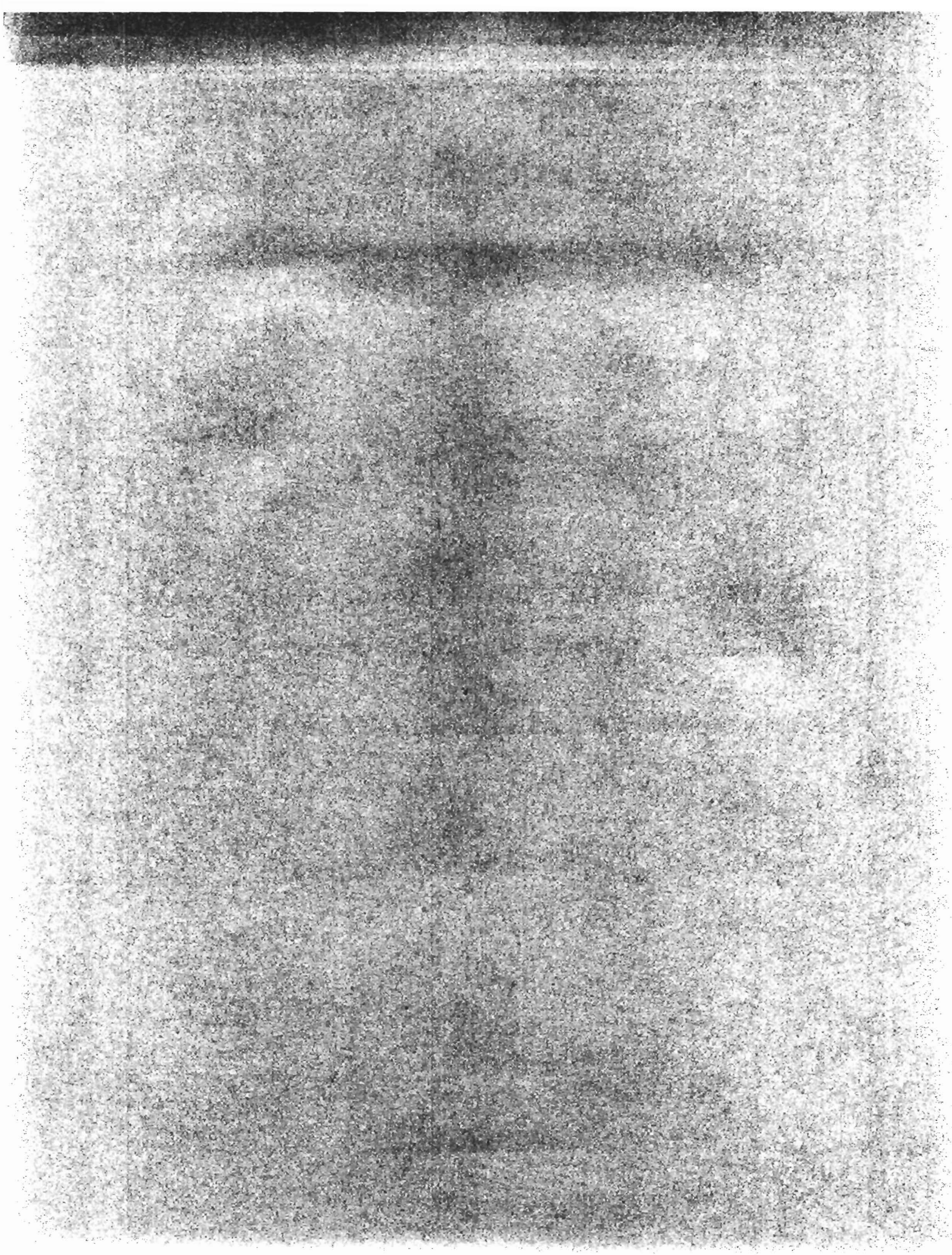


**S1-6**

**"ATLAS, CMS & ALICE Detector Data Acquisition Systems"**

**(Livio Mapelli - CERN/LBL)**

**Discussion of DAQ system requirements and proposed architectures of ATLAS, CMS and ALICE.**



# **Trigger and DAQ plans at the LHC**

## **ATLAS - CMS - ALICE**

*Livio Mapelli*  
CERN

- Requirements
- Trigger/DAQ Architectures
- Conclusions

*DAQ Conference - FNAL*  
*October 26-28, 1994*

## **Requirements**

- Physics - Accelerator
- Rates - Data Volume
- LHC Detectors

## Physics - Accelerator

---

### • Physics --> rejection

#### • p-p

small x-section + QCD background

--> rejection up to  $10^{13}$

( $10^9$  for the Z at the Tevatron)

#### • Pb-Pb

not significant

### • Accelerator --> readout structure

#### • p-p

14 TeV (mb's x-sect) -  $> 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  (40 MHz)

- 40 pile-up events / bunch xing (@ max lum)

--> pipelined front-end

#### • Pb-Pb

6.1 TeV/nucleon -  $> 10^{27} \text{ cm}^{-2} \text{ s}^{-1}$

interaction rate is much lower than bunch xing frequency

--> no need of pipelining

## Rates - Data Volume

---

### • Rates --> trigger

#### • p-p

event rate: 2 GHz

(or 40 MHz of 40 overlapping events)

prompt trigger rate:

dominated by QCD jets faking electrons in calorimeter

--> 40-50 kHz aggregate

#### • Pb-Pb

event rate: 4 kHz

prompt trigger rate (central collisions)

--> 50 Hz

### • Detectors --> data volumes

#### • p-p

selection of interesting events in bulk of QCD background

$O(10^7)$  electronics channels (excluding pixels)

multi-level trigger selection

--> Tbit/s L1 throughput

--> MByte/s recording

#### • Pb-Pb

has to cope with huge particle multiplicity

8000 charged tracks/event

only minimal trigger possible

--> Gbyte/s L1 throughput

--> Gbyte/s recording

# LHC Detectors

- p-p

## A Toroidal LHC Apparatus Compact Muon Solenoid

- Technical Proposal in preparation (Dec 94)

<u>No. Channels</u>	ATLAS	CMS
Pixel	$145 \cdot 10^6$	$80 \cdot 10^6$
Inner Tracker	$7.1 \cdot 10^6$	$16 \cdot 10^6$
Preshower+Calorimeters	$0.19 \cdot 10^6$	$0.76 \cdot 10^6$
Muons	$1.2 \cdot 10^6$	$10^6$

- Pb-Pb      A Large Ion Collider Experiment

- unique heavy-ion experiment at LHC

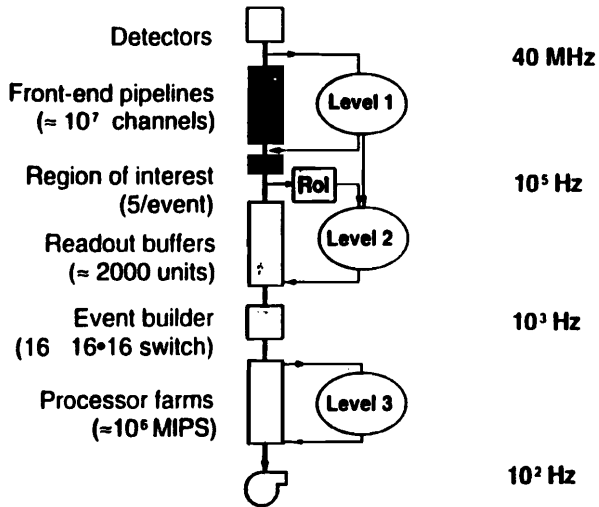
<u>No. Channels</u>	ALICE
TPC	$0.52 \cdot 10^6$
Inner Tracker	$9 \cdot 10^6$
Part Identification	$3.2 \cdot 10^3$
Calorimeter	$20 \cdot 10^3$

## Trigger/DAQ Architectures

- DAQ Logical Structures
- CMS
- Region-of-Interest
- ATLAS
- T/DAQ Components
- ALICE

# Data acquisition logical structures

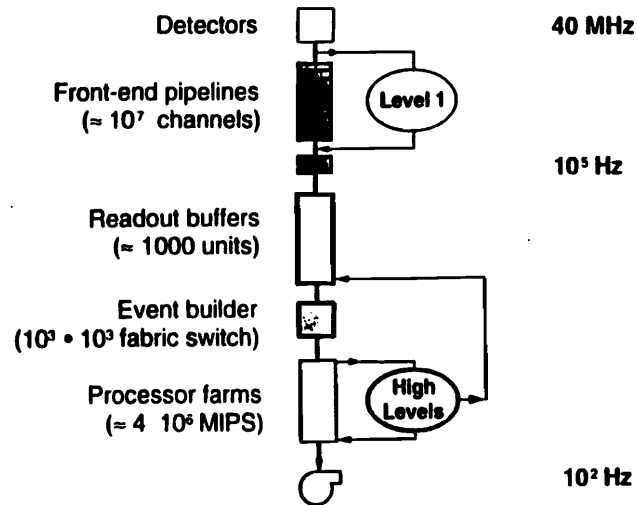
## ATLAS



## CMS

- CMS DAQ Parameters
- CMS DAQ
- T/DAQ Subsystems
- DAQ Main Units
- CMS Virtual L2

## CMS





## CMS DAQ parameters

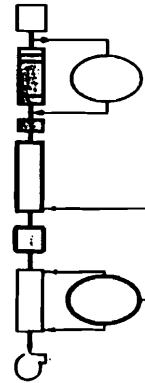
Number of channels and data volumes (at  $10^{34}$  luminosity)

Detector	No. Channels	Occupancy%	Event size (kB)
Pixel	80000000	.01	100
InnerTracker	16000000	3	700
Preshower	512000	10	50
Calorimeters	250000	10	50
Muons	1000000	.1	10
Trigger			10

Average event size	= 1 MB
Level-1 trigger rate	100 kHz
No. of Readout units (200-5000 Byte/event)	1000
Event builder (1000*1000 switch) bandwidth	= 500..1000 Gb/s (*)
Event filter computing power	= $5 \cdot 10^6$ MIPS
Data production	= Tbyte/day
No. of readout crates	= 300
No. of electronics boards	= 10000

(\*) In order to achieve the data acquisition figure of 100 kHz event rate after the level-1 trigger, the tracking data must not be moved into the readout network until the associated event has passed the test of the high trigger levels based on the information from the other detectors. This operation (called virtual level-2) is expected to reduce the event rate (for the tracker data) by at least one order of magnitude.

## CMS two physical levels

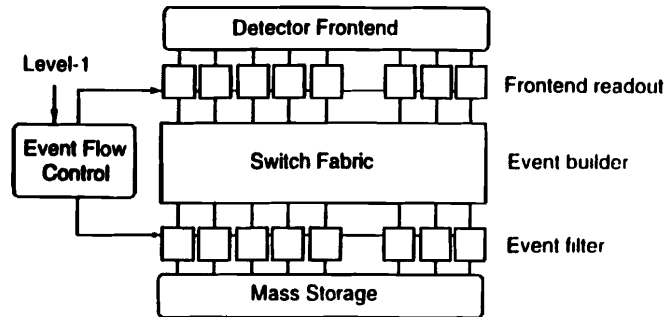


- Reduces the number of building blocks simpler design, easier maintenance and upgrades
- Simplifies the data flow
- Exploits the commercial components 'state of the art' memory, switch, CPU
- Upgrades and scales with the machine performances flexibility in logical redistribution of resources
- Makes full use of the computing power anyway needed for the off-line analysis

## Technology ansatz

- The CPU processing power increases by a factor 10 every 5 years (at constant cost)
- The memory density increases by a factor 4 every two years (at constant cost)
- The 90's are the data communication decade

## CMS data acquisition



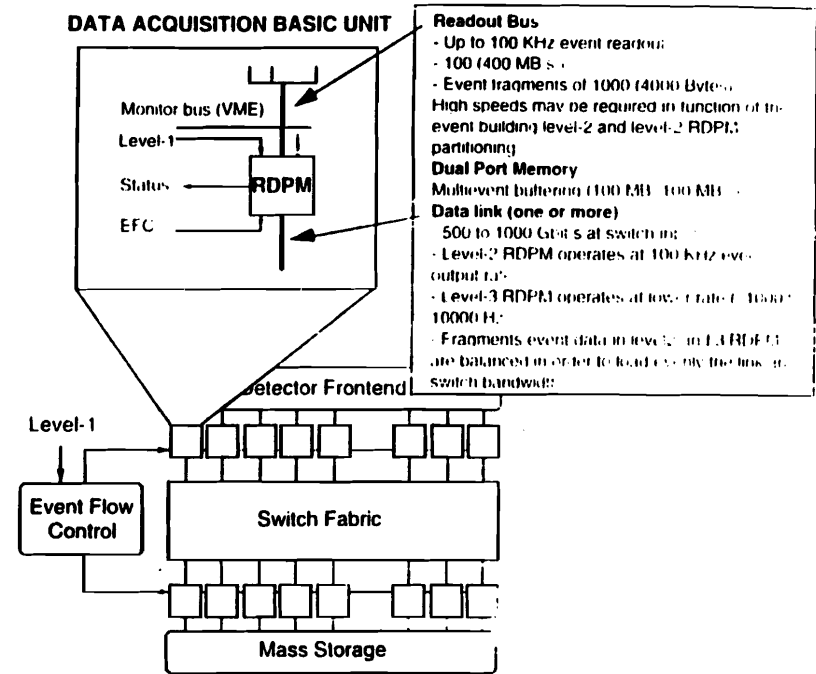
**Frontend readout.** The readout is performed by ( $\approx 1000$ ) data acquisition units composed of frontend driver boards and microprogrammable dual port memories driving a data link to a switch fabric. Each unit is able to handle an event rate of 100 kHz and has the capability of multievent buffering ( $\approx 100000$  event fragments).

In order to achieve the global data acquisition figure of 100 kHz event rate after the level-1 trigger, the tracking data are not moved into the readout network until the associated event has passed the test of the high trigger levels based on the information from the other detectors. This operation (called virtual level-2) is expected to reduce the event rate (for the tracker data) by at least one order of magnitude.

**Event builder.** Switch fabric network ( $\approx 1000 \times 1000$ ) capable to assembly event fragments from readout sources into farm memories.

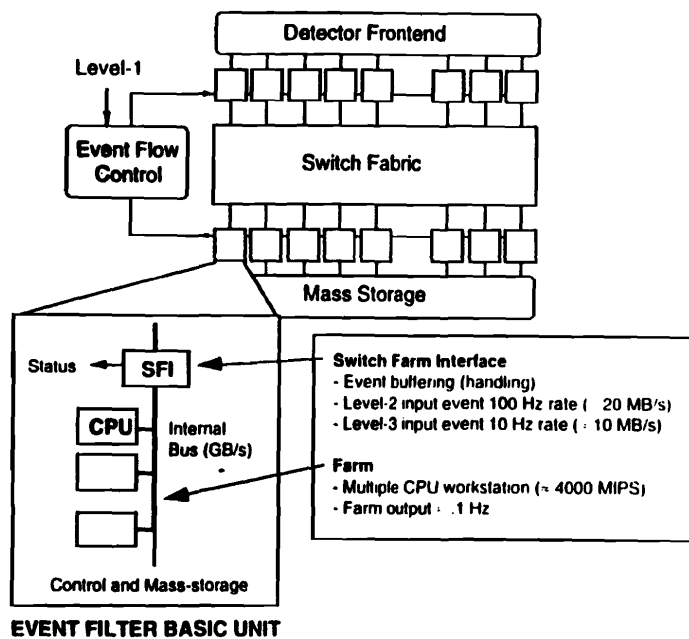
**Event filter.** The event filter performs the high level trigger tasks and event analysis. It consists of ( $\approx 1000$ ) processor farms each connected to a switch output. Events are assembled into each farm by a switch interface at a rate of about 100 Hz.

## Data acquisition basic unit:

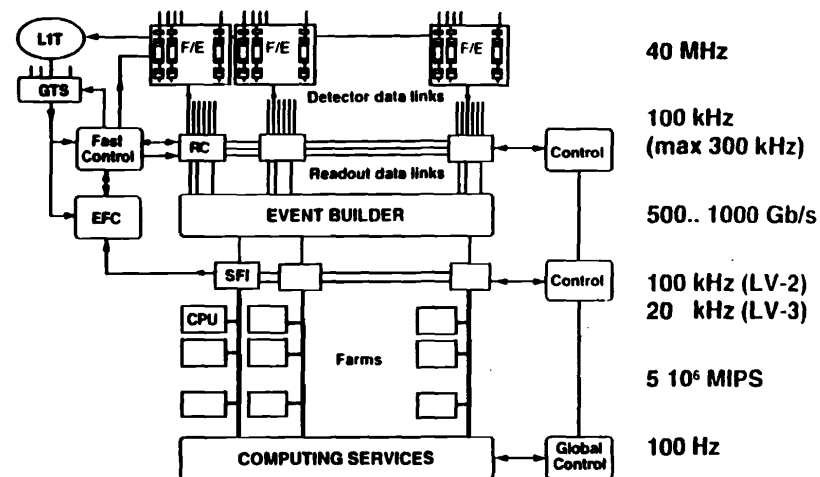


Frontend data acquisition, event data formatting, multi-event buffering  
 The readout is performed by a local interconnection between the DPM and one or more Frontend drivers. The bus is simple and can sustain few 100MB/s. A descriptor contains the data access information and the event number in the order of generation.  
 The output is driven by external control and the events are selected by event number and named by destination task number. The external control can hold clear, read and read a clear a given event;  
 Data are sent out to one or more output link;

## Event filter basic unit



## CMS trigger and data acquisition



Two readout levels:

- ≈ 1000 micro data acquisition units. Frontend readout, multievent buffering
- ≈ 1000•1000 switch fabric. Partial/full event assembly into processor farm
- ≈ 5•10<sup>6</sup> MIPS farms. High trigger levels based on commercial processors

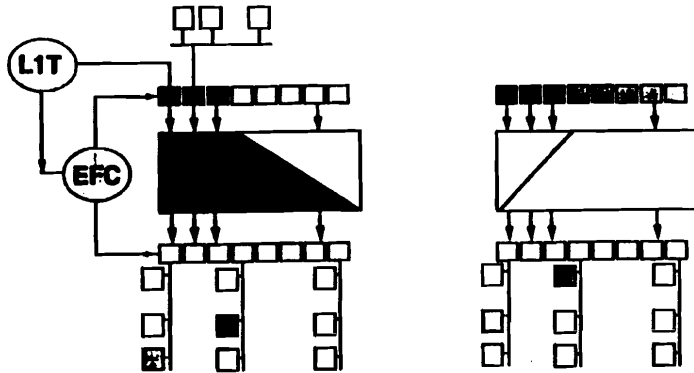
A multiprocessor system with fast input output.

When CPU and memory resources are available, a message is sent by the SFI unit to the Event Flow Controller.

Event are built either into the SFI memory or the CPU memory. The solution depends on the architecture of future computer servers.

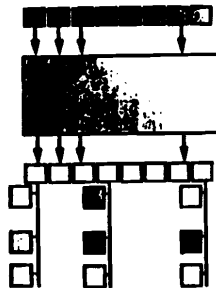
Selected event are archived by global service network facilities

## High trigger levels



1) The level-2 selection uses the calorimeter, muon and preshower data. The sub-events are built using a fraction of the switch bandwidth (e.g. 30%).

2) The rest of the event data is sent after the level-2 decision if the event is accepted

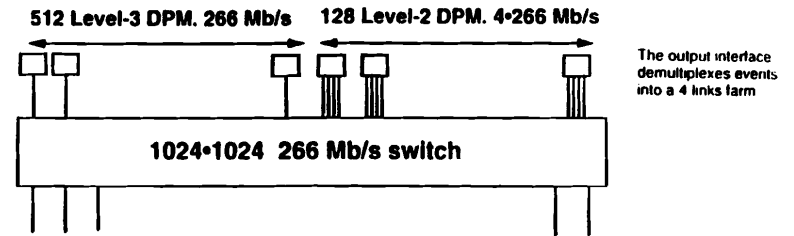


The two operations take place in parallel. The sharing between the level-2 data sources and the rest is such as to match the event builder bandwidth with the level-2 acceptance rate.

- Up to 100 kHz with virtual level-2 mode
- Up to 50 kHz reading the full event data

## CMS high level trigger simulation results

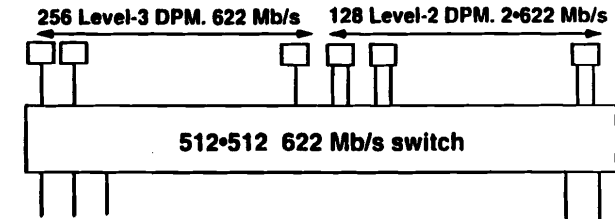
Level-2 and level-3 balanced data source (ATM 4\*4, RD-31 I, Mandjavez)



Level-1 event rate R = 100 kHz  
 Level-2 amount of data D = 80000 Bytes  
 Level-3 amount of data D = 700000 Bytes

Level-2 reduction factor F = 10  
 (- 600 Bytes/DPM, input = 60 MB/s, output 120 MB/s)  
 (- 1400 Bytes/DPM, input = 140 MB/s, output 30 MB/s)

LV1 rate (KHz)	Switch (n*n)	Link (Mb/s)	Load (%)	Latency (ms)	Mode
100	1024*1024	256	54	20..100	Shaping
100	1024*1024	640	22	10..100	Shaping
100	1024*1024	640	22	4..20	Flow Ctrl.



Level-1 event rate R = 100 kHz  
 Level-2 amount of data D = 80000 Bytes  
 Level-3 amount of data D = 700000 Bytes

Level-2 reduction factor F = 10  
 (- 600 Bytes/DPM, input = 60 MB/s, output 120 MB/s)  
 (- 2800 Bytes/DPM, input = 300 MB/s, output 60 MB/s)

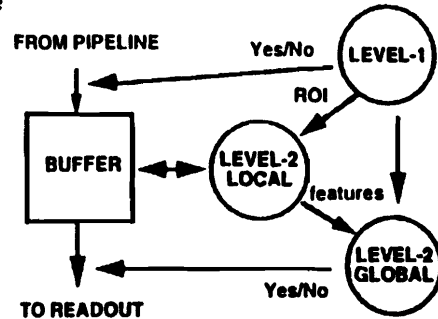
LV1 rate (KHz)	Switch (n*n)	Link (Mb/s)	Load (%)	Latency (ms)	Mode
100	512*512	640	43	8..30	Shaping
100	512*512	640	43	40	Flow Ctrl.

## Region-of-Interest

- Assume we can drive the L2 trigger with the L1 system
  - by passing 'Region-of-Interest' coordinates  
low threshold calo clusters and muon candidates

- Adopt 'Local/Global' scheme for L2 processing

- suggested by nature of event event selection
  - processing is mostly on 'pieces' of events
- local processors
  - extract ROI's 'features'
- global processors
  - event topology analysis



- Data reduction for ROI analysis

- No. of ROIs / event
  - ATLAS jet generation, 35 GeV threshold, full  $\eta$  coverage (6 units):  
No. 5 GeV (isolation 5 GeV) = 2.6 average, tail up to 7  
No. 6 GeV (no isolation) = 3.7 average  
--> Take 1 ROI /  $\eta$ -unit
- ROI size:
  - On calo (electrons):  $\Delta\eta \times \Delta\phi = 0.2 \times 0.1$  or  $0.1 \times 0.2$ .  
--> 0.1 % of total calo area
- In inner tracker detectors, ROI projection size increased by  $z_{\text{vtx}}$  spread
  - > 1-4% of total InDet area  
(ignoring un-matched detector granularity and non-optimised readout)
- ROI data volume
  - does not exceed a few percent of total data volume  
even assuming that all detector in full contribute to L2

## ATLAS

- ATLAS DAQ Parameters
- ATLAS T/DAQ Architecture
- ATLAS Triggering and Data Flow
- T/DAQ Elements - DAQ Crate
- T/DAQ Elements - T2 & T3

# ATLAS DAQ Parameters

## EVENT RATES

( $\odot 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ )

40 Mhz  
(20 events/b-c)

Level 1:  
hardwired  
processors

10-100 kHz

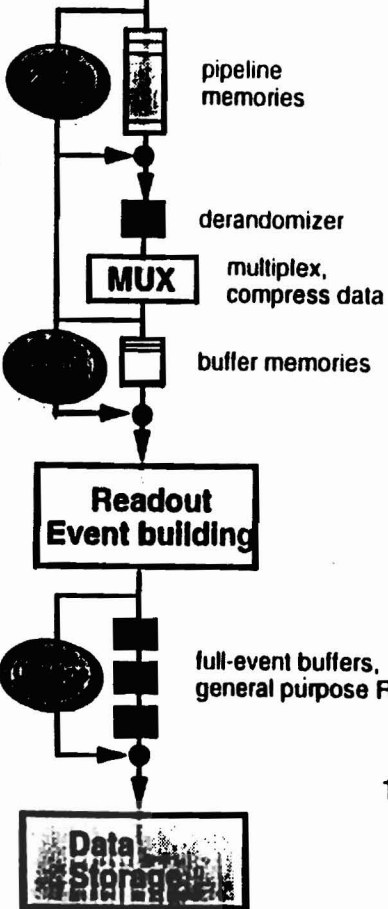
Level 2:  
local  
analysis

100-1000 Hz

Level 3:  
global  
analysis

10-100 Hz

10<sup>7</sup> detector  
channels



## DATA RATES

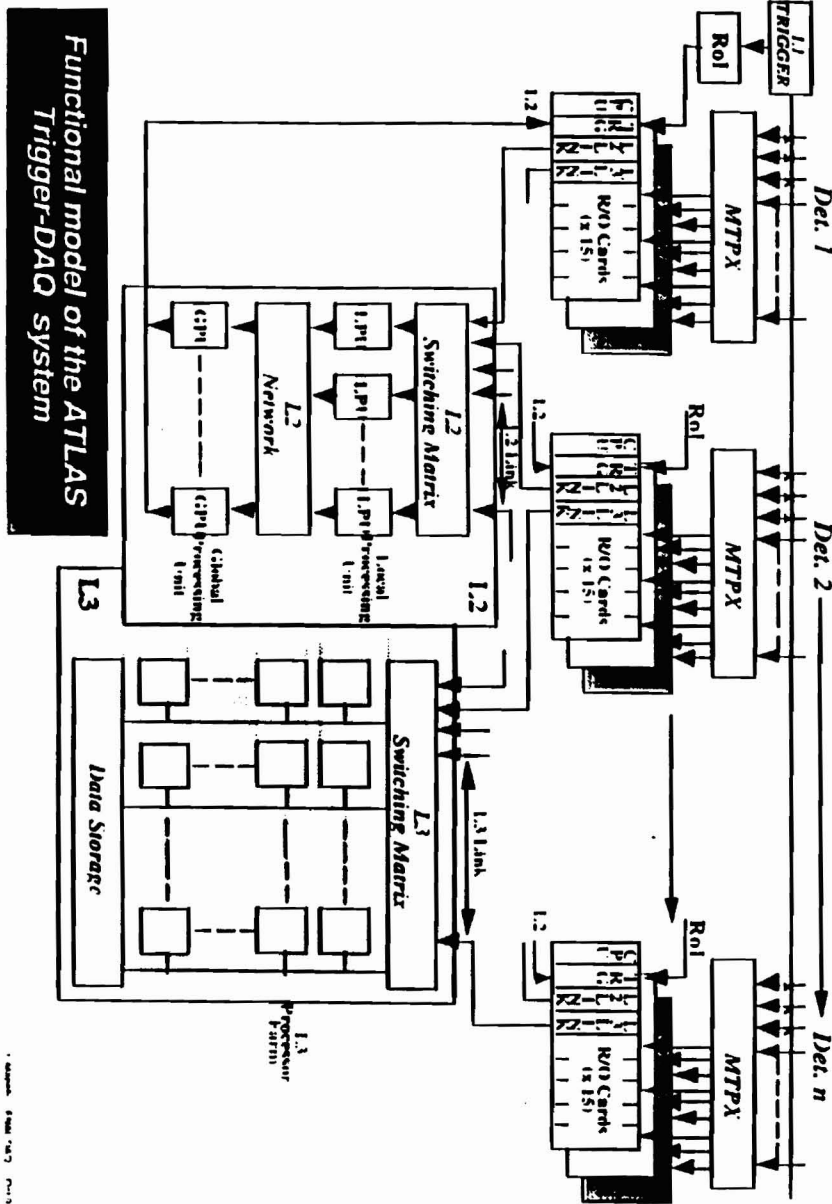
( $\odot 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ )

event  
~ 1 MB

~ 1 Tbits/s

1-10 GB/s

10-100 MB/s

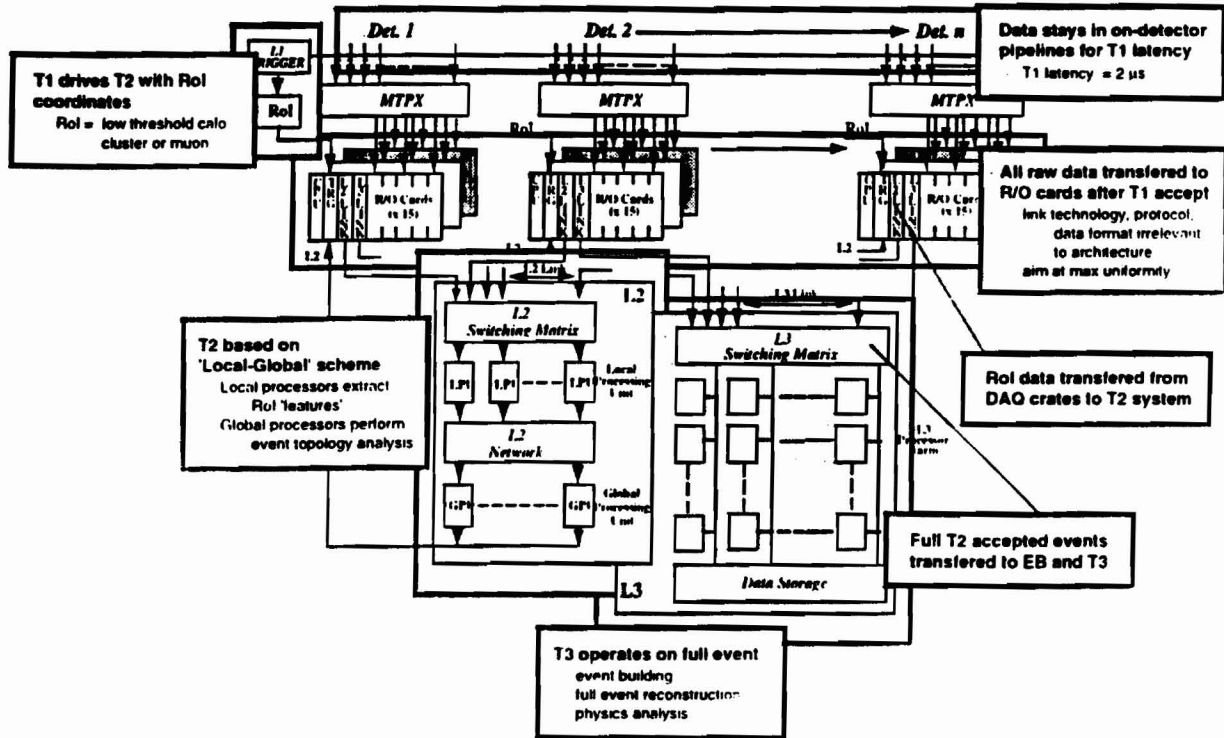


## ATLAS T/DAQ Architecture

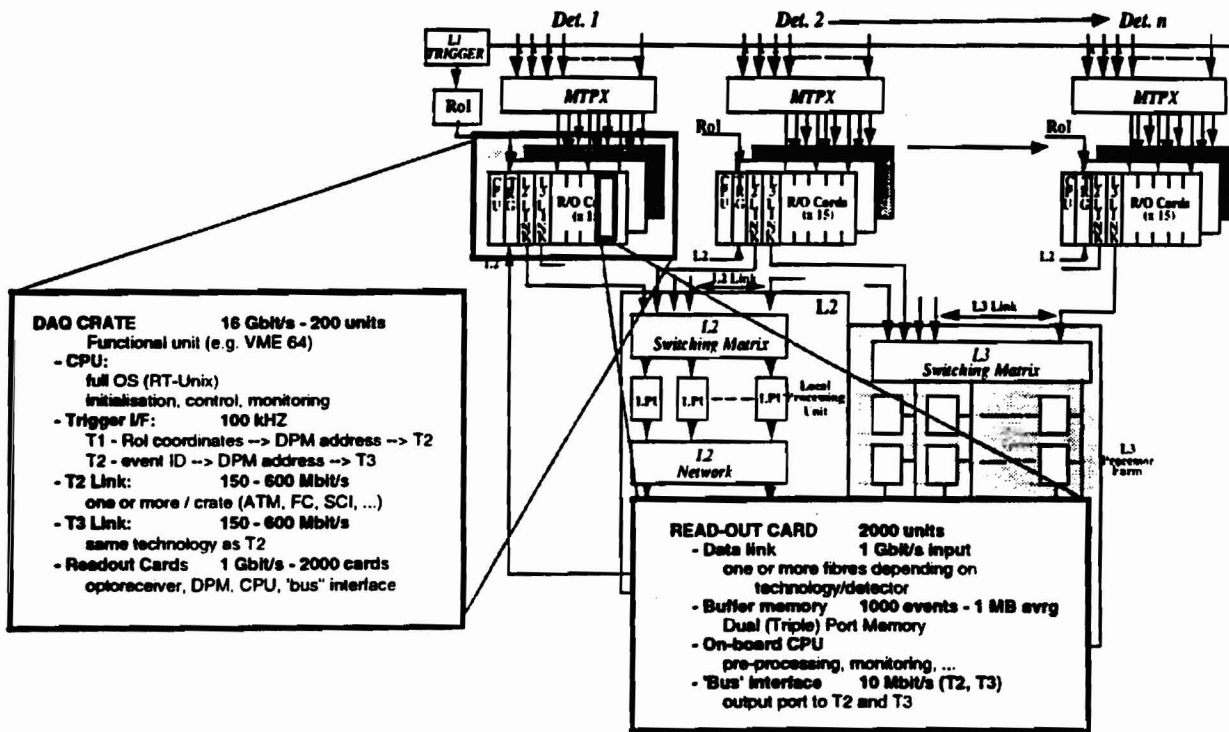
# ATLAS Triggering and Data Flow

## TRIGGERING

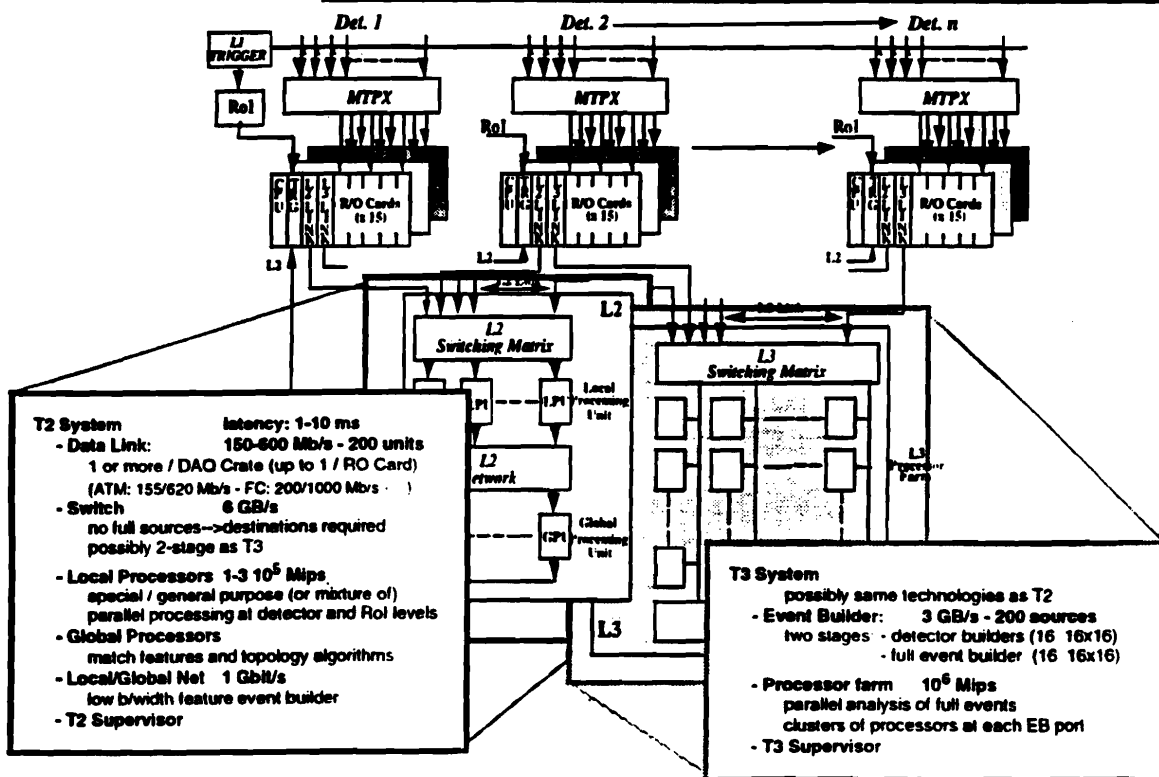
## READOUT SEQUENCE



## T/DAQ Elements - DAQ Crate



# T/DAQ Elements - T2 & T3



L. March - FINAL (1992) - Oct 94

## T/DAQ Components

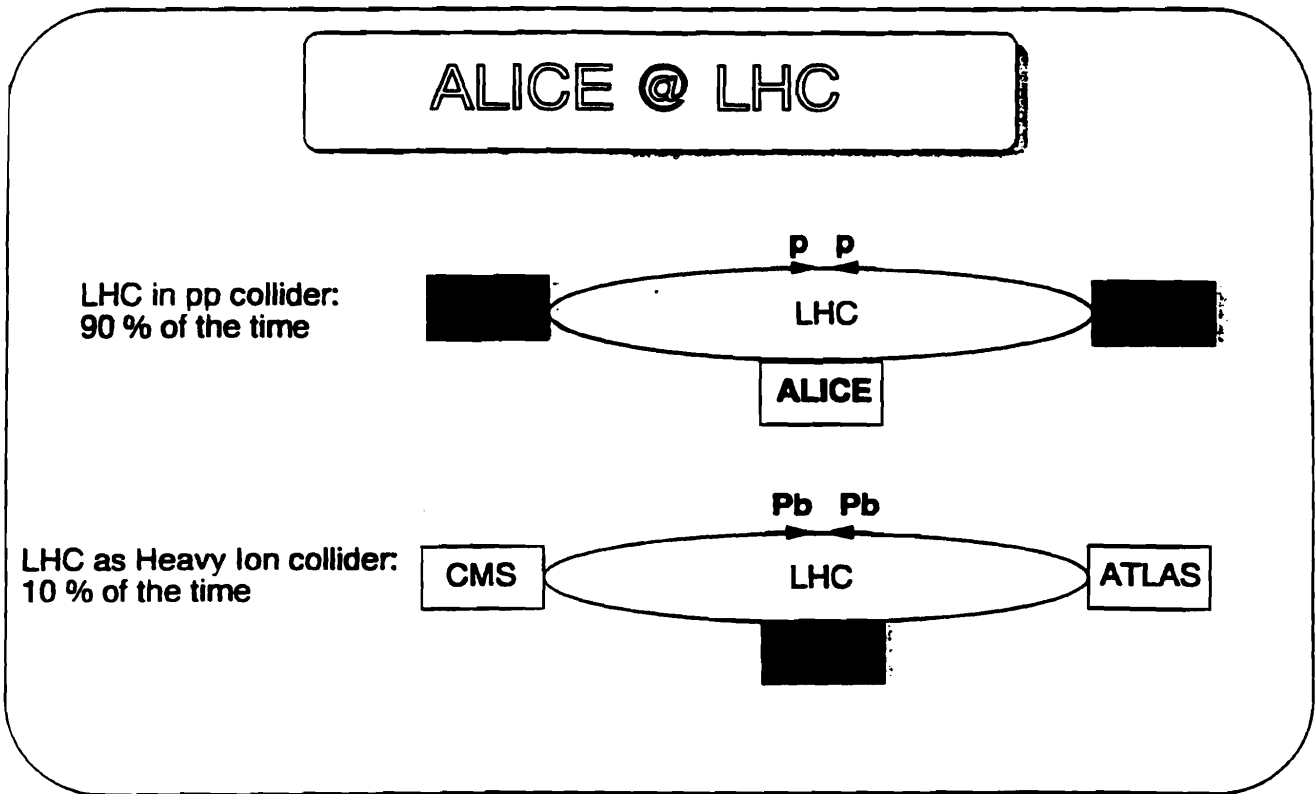
DAQ COMPONENTS	ATLAS	CMS
Average Event Size	1.5 MB	1 MB
Max T1 Rate	100 kHz	100 kHz
No. Readout Cards	2000	1000
No. Readout Crates	200	300
No. electronics boards		10000
Switching Fabric (T2+T3)	3 x (16x16) <i>16x</i>	1000 x 1000
Switch b/width (T2+T3)	100 Gbit/s	500-1000 Gbit/s
Processing power (T2+T3)	1.5·10 <sup>6</sup> Mips	5·10 <sup>6</sup> Mips
T2 InputRate	100 kHz	100 kHz
T3 InputRate	1 kHz	10 kHz

L. March - FINAL DAQ - Oct 94



# ALICE

- Data Volume
- Data Rate
- ALICE DAQ Architecture



## Data Volume

	Time Projection Chamber	Inner Tracking System	Particle IDentification (TOF-RICH)	Electro Magnetic CALorimeter	MB/ Event
# Channels	520 $10^3$	9 $10^6$	170-3200 $10^3$	20 $10^3$	
Time slices	$10^3$	5	1	1	
# bits	8	16-32	8	14	
Occupancy	5%				
Data volume	520 MB				520
Zero suppression	22 MB	1 MB	0.2-1.6 MB	0.035 MB	25
Cluster finding	13 MB				15
Partial tracking	2-7 MB				4-9

## Data rates

	LHC Heavy Ion Collider	LHC Proton collider
Event Rate	50 Hz	500 Hz
Event Size	15-25 MBytes	20 KBytes
Data volume / sec.	750-1250 MBytes/s	10 MBytes/s
Data volume / year	1 month 1000 TBytes	10 month 100 TBytes

## Conclusions

### Trigger/DAQ architectures for LHC experiments

Still need

finalisation of detector configuration

better understanding of requirements from detectors

Top-down designs starting

Need better view of suitable technologies

### Event Building

10 - 100 GB/s required

$10^2 - 10^3$  sources-to-destinations

Industry standard or HEP development?

Still no clear solution

### LHC experiments specify different requirements

Different architecture approaches

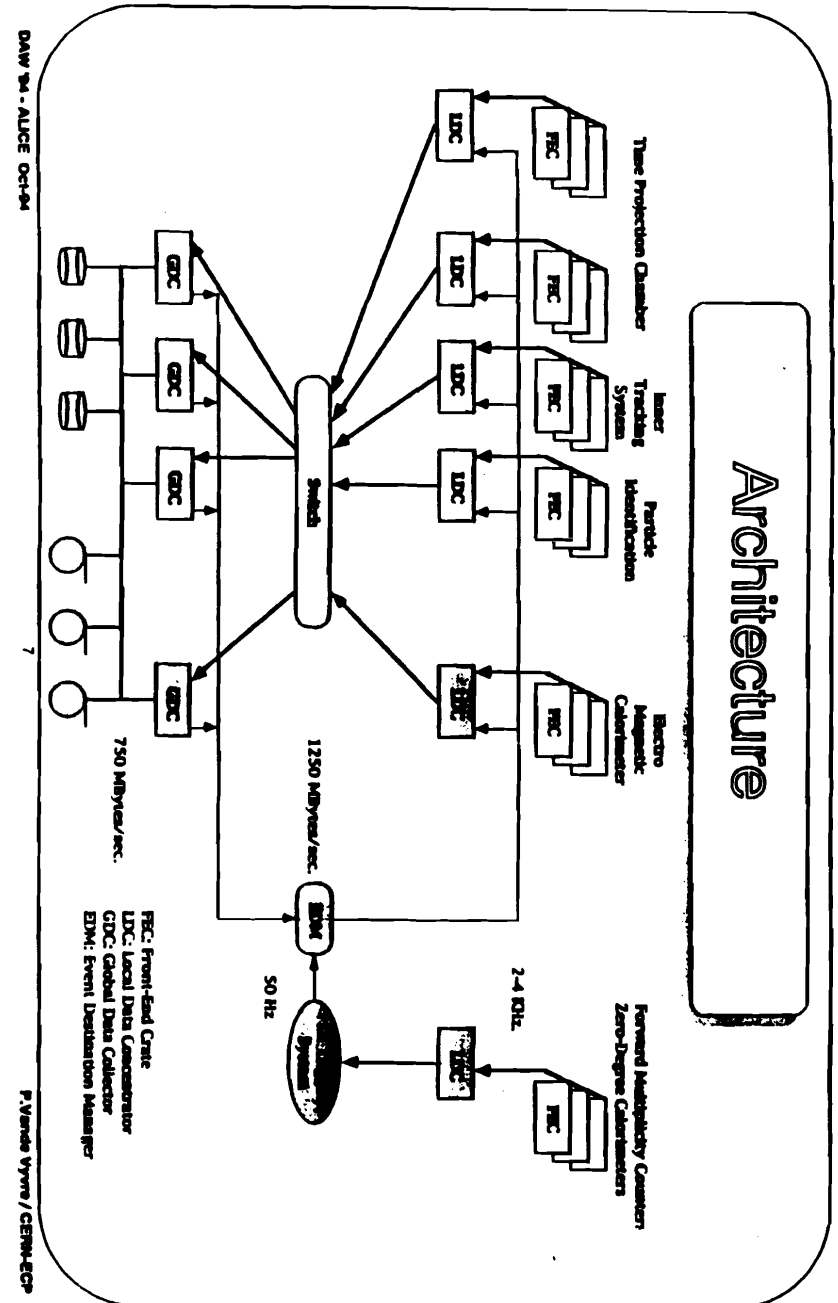
Different Event Building loads

Different complexity of control

Whether one or both approaches work

Technology solution can be common

L. Maier, F. Aul, C. A. C. G. M.



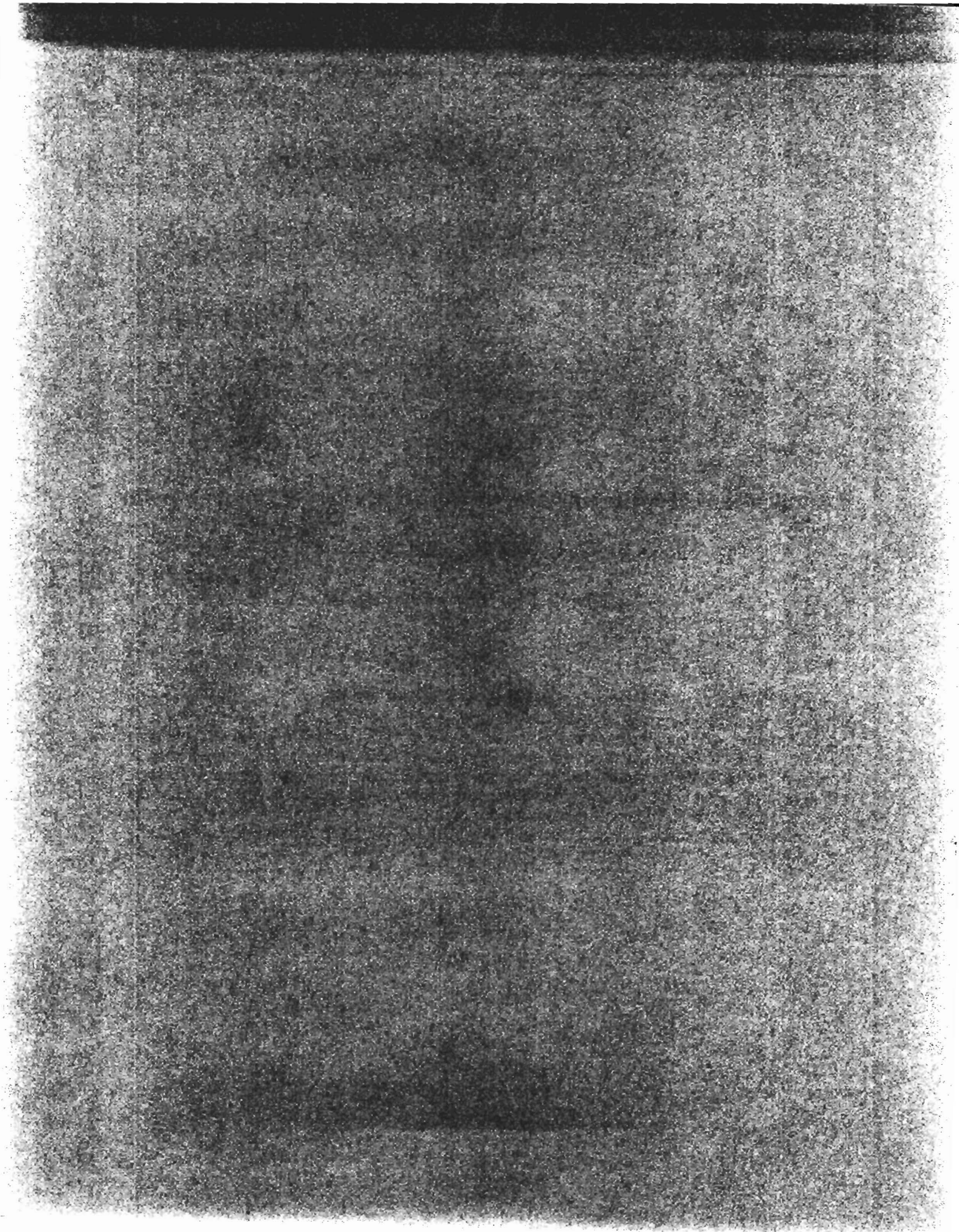


## **S1-7**

### **"Applications of Switching Networks and Meshes of Point-to-point Links in Massively Parallel Systems"**

**(Mark Fischler - Fermilab)**

Communications networks designed for massively parallel computers have many properties desirable in networks used in data acquisition, event building, and trigger processing. Supercomputer communication fabrics tend to have high bandwidths, good switching flexibility, low connection latency, and robustness against the failure of single components. An outline of various MPP communications strategies is presented. We discuss various classes of state-of-the-art switching and supercomputer communications fabrics — current products, concrete product plans, and conservative custom-built options. Custom designs are contrasted with commercial products that may be applicable off-the-shelf or with trivial adaptation effort.



Title: Applications of Switching Networks and Meshes of Point-to-point Links in Massively Parallel Systems

Presenter: Mark Fischler, Fermilab

Abstract:

Communications networks designed for massively parallel computers have many properties desirable in networks used in data acquisition, event building, and trigger processing. Supercomputer communication fabrics tend to have high bandwidths, good switching flexibility, low connection latency, and robustness against the failure of single components. An outline of various MPP communications strategies is presented. We discuss various classes of state-of-the-art switching and supercomputer communications fabrics - currently products, concrete product plans, and conservative custom-built options. Custom designs are contrasted with commercial products which may be applicable off-the-shelf or with trivial adaptation effort.

<For this transcript, I have put the material on transparencies in UPPER CASE. Some indented material may not be presented at talk due to time constraints.

Switching Networks - 1

We will discuss the concept of

USING MASSIVELY PARALLEL PROCESSORS

(or at least their Interprocessor Connection Fabric)

FOR SWITCHING AND/OR PROCESSING  
IN DAQ SYSTEMS

By Massively Parallel Processors (MPP) we mean collections which can go to thousands of CPU's, with some nature of tight interprocess communication. That communication "backbone" can be exploited in a DAQ system as a super switching network.

MPPs can be obtained commercially, and we will describe what is emerging there. There are also systems designed by smallish groups for specific types of applications, and some of these have sufficiently flexible switching to be of interest for DAQ.

This switching network can be of use in either an

	1	L2	build the event.
EVENT BUILDER	1	L2.5	TRIGGER
	01	L3	record the event.

By "level 2.5 we mean that the full event is available but maybe not built in one place; the output is a decision to discard the event or the assembled event to record.

Computer companies, driven by the requirements of some users, are making these backbones (sometimes called "switching fabrics")

HIGH BANDWIDTH  
 FLEXIBLE  
 RAPIDLY RECONFIGURABLE for complex data movement  
 and ROBUST because errors in an MPP system quickly make the entire system worthless.

WHY WOULD AN EXPERIMENT CONSIDER USING THESE SWITCHES IN THEIR DAQ?

ONLY IF YOU...

NEED BIG BANDWIDTH

AND (either)

COMPLEX | NEED  
 DATA | OR INTEGRATED  
 MOVEMENT | COMPUTING  
 PATTERN | POWER

## WHO MAKES THESE SYSTEMS

There are the vendors, and although I've listed just their current products, the trend is toward much better switch fabrics in intended products.

VENDORS -----	SPECIALIZED -----
INTEL (PARAGON)	COLUMBIA (Norman Christ)
CRAY (T3E)	FERMILAB (ACPMAPS)
CONVEX (EXEMPLAR)	IBM (GF11)
SGI (CHALLENGE)	ATI (ITALY)
IBM (SP2)	OCDFAN (JAPAN)
TMC (CM 5) (rest in peace)	SWITCHES DESIGNED FOR GENERAL DAQ USE (FNAL)

And there are groups creating dedicated systems -- the key area is Lattice QCD machines. Important examples are Norman Christ's at Columbia (always at or near the top in raw power), and our ACPMAPS at Fermilab, based on crossbar switches and emphasizing flexibility (although for the past few years we have been at the top in power as well).

There have also been efforts to create general switches to be used in many HEP experiment DAQ systems. For example, Ed Barsotti et. al. at FNAL developed such a switch system a couple of years ago. These have generally not caught on, perhaps because until recently DAQ switching needs were not as severe as they will be in the future.



To evaluate these fabrics for DAQ use, it is important to understand the strategies and models they employ:

STRATEGIES FOR COMMUNICATION  
AND  
PROGRAMMING MODELS

The connections can be nearest neighbor or global. A physical grid of neighbor connections can still logically be global if automatic routing is done in hardware.

Programming models supported by hardware are listed in order of increasing flexibility:

	NEIGHBOR -----	GLOBAL -----
LOCKSTEP	COLUMBIA	GF11
MESSAGES	IPSC (early)	PARAGON IBM SP2
REMOTE ACCESS	CM-2	CRAY T3D ACPMAPS
GLOBAL SHARED MEMORY	XXXXXXXXX (makes no sense)	SGI CHALLENGE CONVEX EXEMPLAR

By "lockstep", we mean all processors communicate at the same time  
(not quite as restrictive as completely single instruction stream SIMD).

The important feature of message passing is that the target has to be prepared to receive each message.

Remote access is the model chosen at FNAL for ACPMAPS. We base ours on crossbar switches. Intel and Cray use grids with sophisticated routing chips at each intersection -- the grid approach is a strong trend among vendors.

Finally, there is Global Shared Memory, which today is done via a shared bus or SCI ring (which is a limitation). But it won't be limited that way forever.

Systems with neighbor connections, or based on message passing or lockstep communication, are less suitable for use in a DAQ context.

Today's leading-edge and most of tomorrow's commercial system have some

COMMON FEATURES

HIGH BANDWIDTHS      because vendors have learnt that no programming tricks can overcome a net bandwidth deficiency  
100+ - SEVERAL HUNDRED MB/S  
today                    over each link, and these are bidirectional.

In practice, you get only half that bandwidth at most. Note that we always talk in MegaBYTES per second; "real men" don't use Megabits/second.

SOPHISTICATED ROUTING

- \* BASED ON ROUTING CHIPS: SMALL CROSSBAR + LOGIC to handle routing decisions and cache coherency issues.
- \* STRUCTURED AS GRID (in general, with variations)
- \* SIMPLE PACKET PROTOCOLS

ROBUSTNESS

- \* ERROR CHECKING AND/OR CORRECTION
- \* ALTERNATE ROUTING TO SURVIVE SINGLE COMPONENT FAILURES

VERY HIGH EXTERNAL "I/O" BANDWIDTH

- \* MULTIPLE INTERFACES FOR A VARIETY OF BUSES  
PCI, ATM ...

This I/O connectivity is crucial for DAQ applications unless you choose to directly mimic the internal switching protocol (which is probably unwise).

**WHAT TECHNOLOGY CAN WE GET FROM OTHERS THAN MPP VENDORS**

That is, other people's roll-their-own supercomputer projects

**WHAT IS THERE TO EXPLOIT?**

-----

**OPTICAL INTERCONNECTS**

For one thing, optical connections are emerging that allow you to solve tricky line-length and packaging issues

+++ DOIM

Dense Optical Interconnect Modules do hundreds or even thousands of megabytes per second by combining multiple fibers.

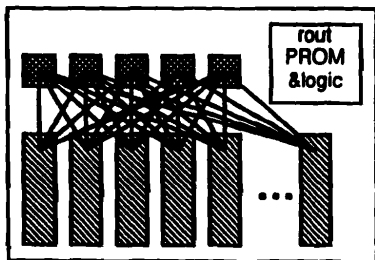
**OFF-THE-SHELF OR SEMI-CUSTOM CROSSBAR SWITCHES**

Although the coherency and packet switching logic in vendors' grid routing chips would be tough to duplicate, with off-the shelf parts or routine semi-custom logic one can create large simple crossbars --

+++ 64-WAY IS PRACTICAL TODAY

One example of how you would use these is in the active crate backplane of ACPNAPS, where processor or crate interconnect modules plug into a 16-slot backplane. You can then assemble many crates to form a system. Trace routing problems limit this geometry to about 16 modules per backplane.

(Picture: The backplane has 16 connectors, multiple crossbar chips, and a routing PROM plus logic.)



We illustrate what can be done along these lines today by a

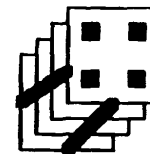
**POSSIBLE FUTURE CROSSBAR SWITCHING GEOMETRY**

developed by Don Husby at Fermilab.

The technical issues to be resolved are how to route the switching board, to avoid crosstalk problems with the high clock rates needed.

**ROUTING, CROSSTALK --- SMALL BOARDS WITH A FEW CROSSBAR CHIPS**

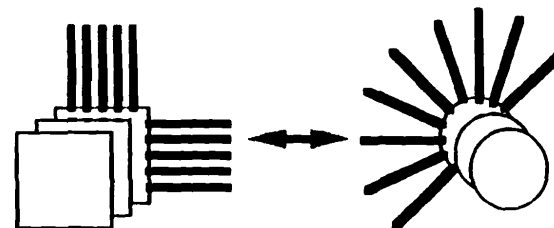
The solution is to produce small boards for short paths, with only a few crossbar chips on each to avoid trace routing nightmares. Of course, each small board can only handle a nybble or so, but these can be stacked to give 100 Megabytes/second or more on each path.



MODULES HAVE VERTICAL CONNECTION TO SIDES OF SEVERAL HORIZONTAL SWITCHING BOARDS

You can put together a system with an arbitrary number of such stacks, with as many links between them as are needed. Because there can be many paths between stacks, the routing logic should deal in terms of sets of links rather than specific links. This will minimize contention traffic jams.

A minor improvement is to use 64-sided or circular stacks of crossbar boards, to allow the plug-in modules extra chip height over most of their area.



WILL THESE SWITCH FABRICS BE AVAILABLE?

WITH COMPUTER -- YES                    SSSSSSSSSSS

Of course the vendors want to sell their product, but it may be expensive overkill.

WITHOUT (MASSIVE) CPU POWER -- OK        SOME THOUGHT

It will generally be possible to save money by configuring a low power, relatively low memory, high I/O and connectivity balance, if appropriate.

COMMERCIAL FABRICS ALONE  
-- SOME NO, OTHERS YES  
-- TAKE REAL EFFORT

Few if any will sell their routing fabric cheaply as a standard product. But collaborative innovative efforts are welcomed by some companies.

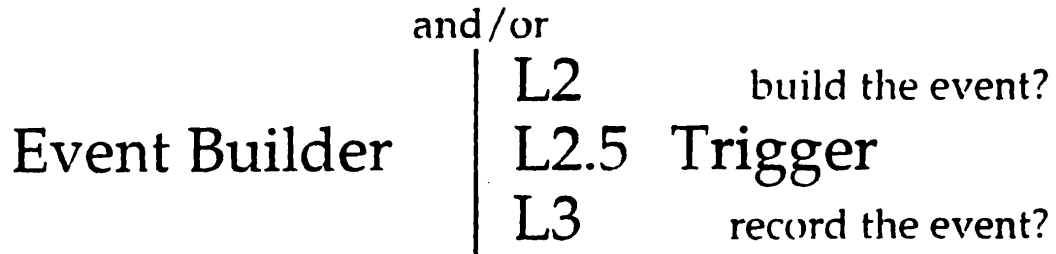
SPECIALIZED MPP INTERCONNECT HARDWARE  
-- IF SPECIFIC PROJECT MEETING NEEDS

This is practical if some project has switching technology that meets your needs. Then you can buy from a company that is manufacturing boards for that project, or replicate designs -- most projects will welcome that. Of course, to know what these projects are doing takes some attention.

-- CAN DESIGN YOUR OWN  
SHOULD BE LAST CHOICE

But many times there are no alternatives; you do what is needed to make the experiment fly.

# Using Massively Parallel Processors For Data Movement (Switching) and/or Processing in DAQ Systems



## SWITCHING FABRICS:

- HIGH BANDWIDTH
- FLEXIBLE
- RAPIDLY RECONFIGURABLE
- ROBUST

Why would an experiment consider using  
these switches in their DAQ system?

Only if you need


## BIG BANDWIDTH AND...

Complex  
Data  
Movement  
Pattern

Need  
Integrated  
Computing  
Power

## Who makes these systems?

### Vendors

Intel (Paragon)  
 Cray (T3D)  
 Convex (Exemplar)  
 SGI (Challenge)  
 IBM (SP2)  
 TMC  (CM-5)

### Specialized

Columbia (N. Christ's)  
 Fermilab (ACPMAPS)  
 IBM (GF11)  
 APE (Italy)  
 QCDPAX (Japan)  
 Switches designed for  
 general DAQ use (FNAL)

## Strategies for Communication and Programming Models

		<u>Global</u>
Lockstep		GF11
Messages		Paragon; IBM SP2
Remote Access		CRAY T3D; ACPMAPS
Global Shared Memory		SGI Challenge Convex Exemplar

# Common Features

## High Bandwidths

- 100+  $\Rightarrow$  several hundred Mbytes/sec

## Sophisticated Routing

- Based on routing chips: Small crossbar + logic
- Structured as grid (with variations)
- Simple packet protocols

## Robustness

- Error checking and/or correction
- Alternate routing to survive single failures

## Very high external "I/O" bandwidth

- Multiple interfaces for a variety of buses

(PCI, ATM, ...)

What technology can we get from projects other than MPP vendors?

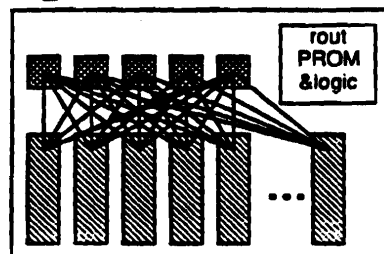
What will specialized systems be able to exploit?

## Optical Interconnects

$\Rightarrow$  DOIM (Dense Optical Interconnect Module)

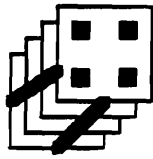
Off-the-Shelf }  
Semi-Custom } Crossbar Switches

$\Rightarrow$  64-way is practical today

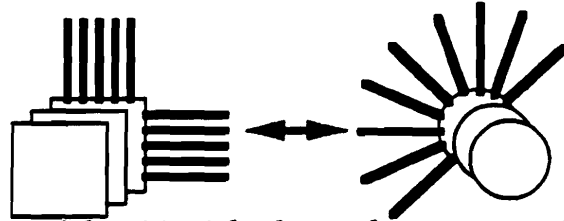


Possible future crossbar switching geometry:

Trace routing }  
Crosstalk } Small boards with a few crossbar switches



Modules have vertical connections to sides of several horizontal switching boards



Possibly 64-sided stacks to provide extra chip height on modules

Arbitrary number of links between pairs of stacks  
Routing logic in terms of SETS of interstack links

Will these switch fabrics be available?

With computer — YES                    \$ \$ \$ \$ \$

Without massive CPU power — OK  
But some thought needed

Commercial fabrics alone

- Some NO, others YES
- Requires real effort

Specialized MPP interconnect hardware

- If specific project (or its switching) meets your needs
- Can design these on your own

That should be the LAST choice

## How to decide

Do you need the  $\left\{ \begin{array}{l} \text{flexibility} \\ \text{compute power} \end{array} \right\}$  ?

Is there a  $\left\{ \begin{array}{l} \text{product} \\ \text{project} \end{array} \right\}$  that meets needs ?

---



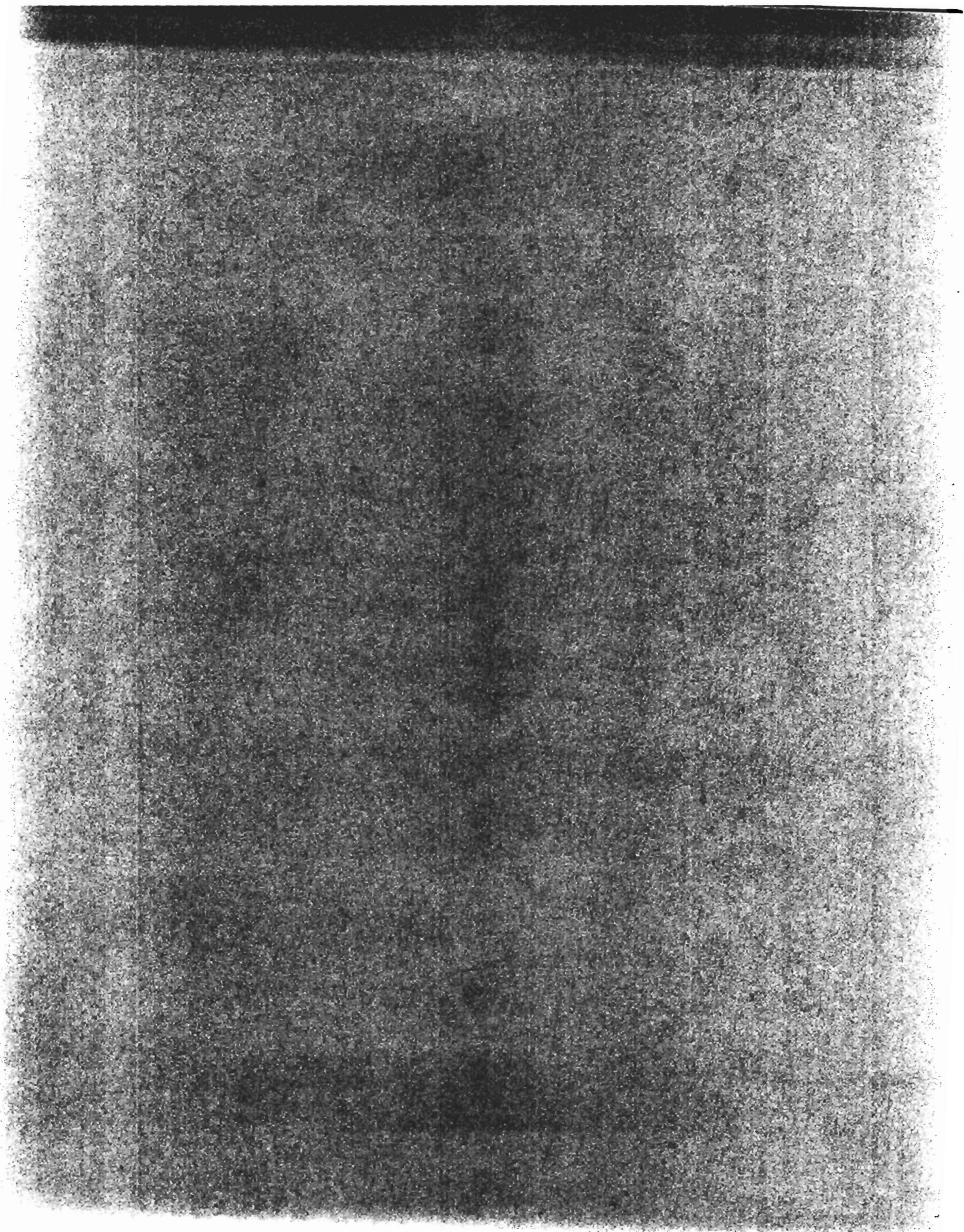
**S1-8**

**"Applications of Switching Networks & Point to Point**

**Links in Other Physics Applications**

**(Marvin Johnson - Fermilab)**

Use of switching networks in non-event-builder applications such as first level triggers, data distribution and concentration within front-ends.



## Use of Switches and Point to Point Networks for Intermediate Level Triggers

Marvin Johnson  
Fermilab

### Abstract

This paper discusses the use of switches and fiber optic links in intermediate level (Level 2) triggers.

### Introduction

The trend in high energy physics experiments is to look for ever rarer events which usually means ever higher rates. The amount of data that can be written to storage media is limited by both the physical media constraints (bulk, cost and so on) and the amount of labor available to analyze the data. The best solution to this problem is to be more selective in choosing data to record which requires more discriminating triggers. The advent of microprocessor farms for level 3 triggers and various types of event builders have significantly improved the overall performance of the trigger system. However, making decisions earlier reduces the demands on the data transfer system and usually increases the system bandwidth. This paper describes some current level 2 systems discusses some possible future developments.

Level 2 triggers are defined as the non dead timeless trigger between the dead timeless level 1 and the processor farm. Processing times are typically between 10 and 100  $\mu$ s.

Better triggers require more information which usually means using data from different parts of a subdetector or different detectors. In other words, they will use some type of event building. Several experiments already use these techniques. Section I discusses some existing trigger systems built in the last few years. Section II describes the use of passive optical splitters and cross bar switches in future experiments.

### I. Some Examples From Current Experiments

There have been several trigger systems built in Europe using INMOS transputers. Transputers have several built in communication ports so it is fairly easy to build systems that communicate locally between nearby sections of a detector as well as with an overall processor. Fig. 1 shows a diagram of a level 2 proportional wire chamber trigger system based on transputers from UA6 at CERN<sup>1</sup>. This device finds tracks in proportional wire chamber data. The Receiver Memory Hybrid (RMH) devices read out wire chamber data from the detector and send the data to the Parallel Crate Acquisition module (PCA) devices which are transputers. Two of the four I/O ports connect to adjacent transputers so that data from nearby wires can be combined to find lines and points. The other two ports are connected to the analysis network

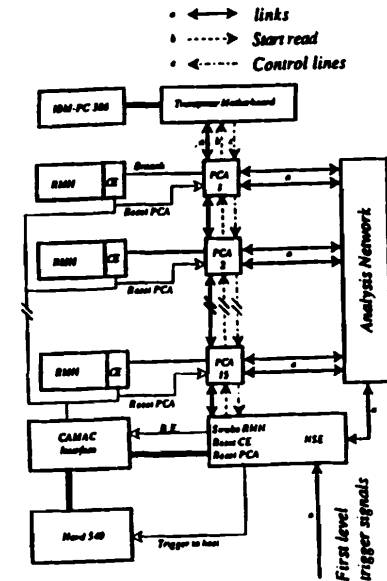
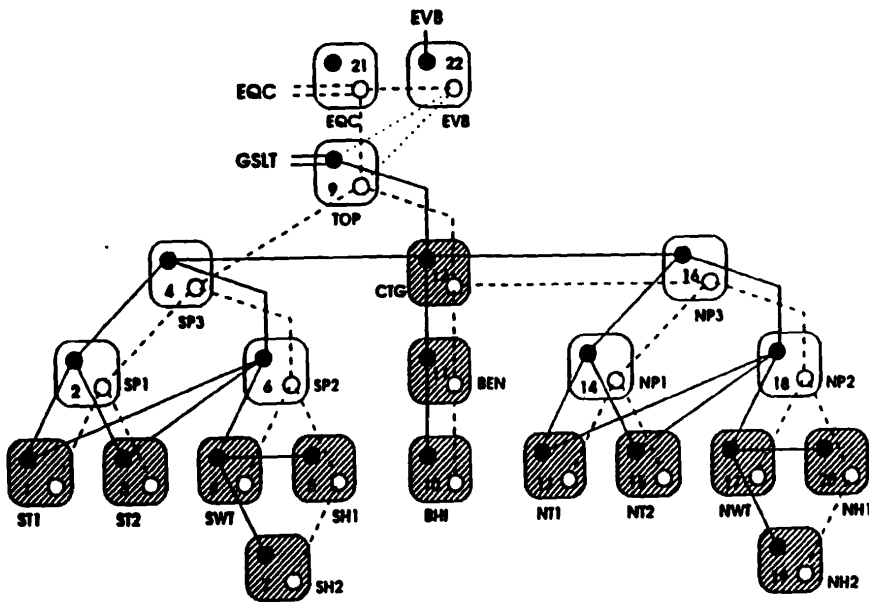


Fig 1

Fig. 2. The UA6 Transputer-based data acquisition and trigger system for multiwire proportional chambers.

### III. ANALYSIS NETWORK



2  
 Figure 2: The transputer network topology. Each box represents one 3-TP board, each circle one transputer. Solid lines are SLT subnetworks links, dashed lines are links belonging to EVB subnetwork. Boards which deliver data for the SLT algorithm are shaded. Numeric board addresses and mnemonic identifiers are shown.

which is also composed of transputers. The analysis network is essentially the level 2 trigger system. The PCA transputers are used to find lines and points in the wire chambers. These data points are then sent to the analysis network which then fits them into trajectories. This is event building on a local level which is not too different from many level 3 systems. The switch in this case is software in the transputers.

The second example is from the Zeus detector at HERA<sup>2</sup>. Fig. 2 shows a block diagram of this transputer based system. This one closely follows an event builder type of approach. All of the boxes in the figure include two transputers. The white ones are part of the general event building network for the detector and the black ones are dedicated to the second level trigger (SLT). Both systems are organized in a tree like structure and both employ a loose form of pipelining (each row is processing a different event but there is no time synchronization). Front end data is brought into the transputer board from the front ends. Data is shared between the event builder and the trigger by shared memory. Data is not transported to level 3 unless there is a positive second level trigger so data does not flow through the system in parallel. Rather, the front end buffers hold the data while it is sent through the various levels of the trigger system until a level 2 accept is generated. Each layer of the tree computes part of the second level trigger and sends it on the next layer. The switch is again software inside the transputer.

## II. Future Trends

The next example is a proposed second level impact parameter trigger for D0. It uses data from the silicon and fiber tracker detectors as well as different regions of the same detector. This system is similar to the Zeus system described above but differs in several important areas both in architecture and hardware implementation. The main architectural difference is that the normal DAQ system and the level 2 trigger system get the same data at the same time rather than retransmitting the data from the DAQ system.

The D0 system uses fiber optics to transmit the data from the front end to the DAQ system. The present design uses GLink fiber drivers from Hewlett Packard<sup>3</sup>. These devices take in a 16 bit word every 18.8 ns and send it out over a fiber optic cable (over 800 Mb/s on the cable). Trigger data is obtained by splitting the optical signals with a passive splitter into two identical parts with one going to the DAQ event builder and the second going to the trigger. Each receiver uses a separate GLink receiver. The GLink driver can synchronize any number of receivers as long as they receive an adequate sized signal. At the moment the driver can drive only two receivers. This system can accept additional level 1 triggers when level 2 is running so event data must be buffered before sending it to the level two processor. This buffering is done directly after the level 2 receivers. Figure 3 shows an overall block diagram of the system.

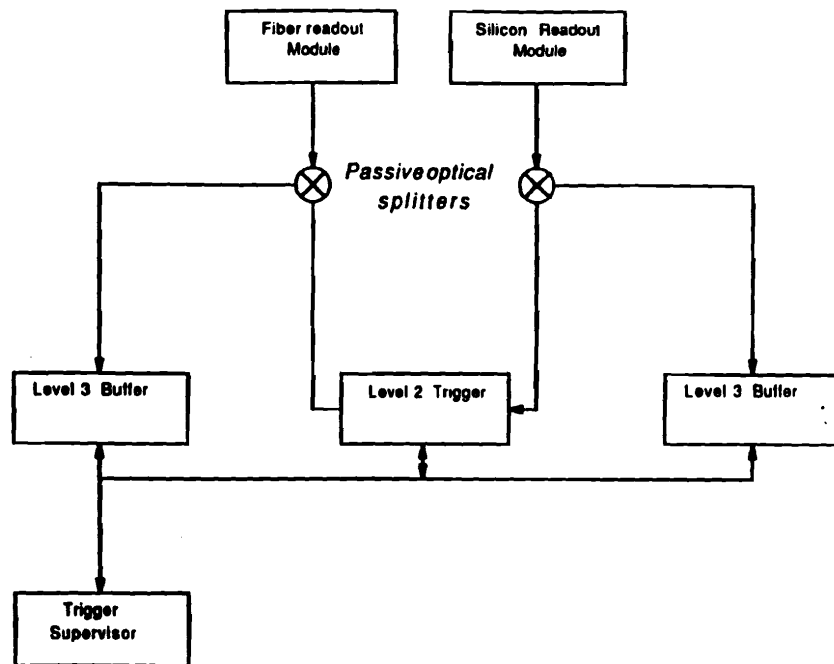


Fig. 3. Block diagram of a possible level 2 trigger system using fiber optic cables. Passive optical splitters are used to share the data between the normal processor farm buffer and the level 2 trigger system.

The D0 system uses programmable hardware in the form of field programmable gate arrays to do high speed data selection. Data from the fiber tracker is completely processed to determine roads in the silicon data. This data is then loaded into static RAM based gate arrays which then select any silicon hits associated with this road. Different sections of the silicon readout may contribute data to a given track. The data from these different sections are collected together via a cross bar switch in a manner identical to that of a switched based event builder. Impact parameters are reconstructed via digital signal processors. DSP's are used over general purpose computers because of their superior speed. Fig. 4 shows a more detailed block diagram.

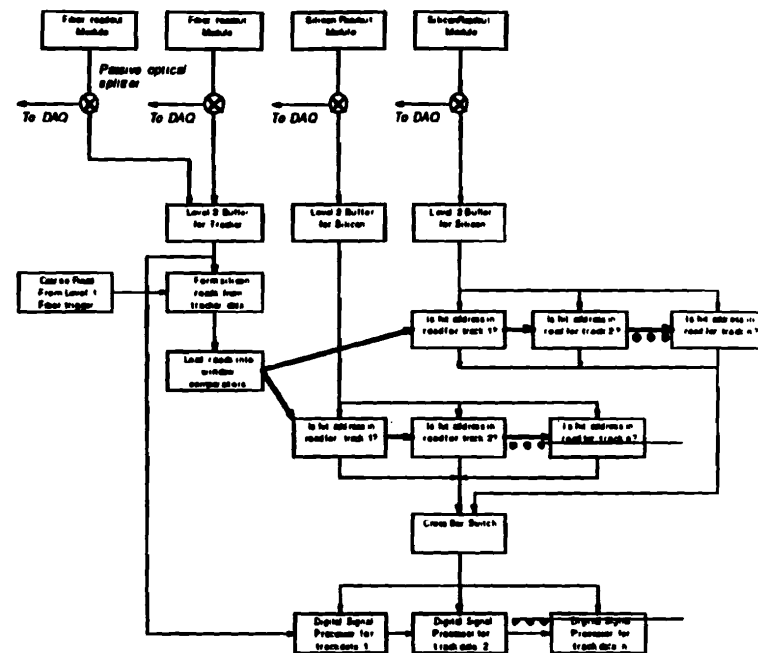


Fig. 4. This figure shows a more detailed block diagram of a possible silicon trigger system. Data from the fiber and silicon system is used to select hits in roads in the silicon data. Hits from different detector segments are combined into one DSP by a crossbar switch. The DSP then computes an impact parameter.

DSP's are very powerful computational engines. However, they work best on a specific type of calculation; namely the sum of scalar products. In order to get the best performance from DSP's, the algorithm should fit into this form. Punzi and Ristori have suggested a method of linearizing constraint equations. They do this by calculating the parameters at the center of every road and then using a Taylor series expansion to first order about the center of this road. The authors show that this method gives results that agree well with that obtained by a non linear least squares fit.

#### Summary

Fiber optics has sufficient speed and density to allow signals from many different detectors to be brought to a single printed circuit board for level 2 triggers. Splitting the optical signal with passive splitters allows data to go to the trigger system at the same time that it goes to the normal DAQ system. This simplifies both the DAQ and the trigger board.

The rapid progress in field programmable gate arrays now allows very complex logic to be embedded in a single integrated circuit. Many of these devices are also based on static RAM so that the trigger logic can be modified by downloading new equations without removing the boards.

The computation speed of DSP chips is increasing quickly. These devices give their best performance when evaluating scalar products so some algorithm development may be necessary to get the best performance from these devices.

---

<sup>1</sup> C. Comtat et al., A transputer-based second-level track trigger in the SpS Collider for the CERN UA 6 experiment, Proceedings of the Eighth Conference on Real-Time Applications in Nuclear, Particle and Plasma Physics, June 8-13, 1993, Vancouver, B.C. P 419.

<sup>2</sup> J. M. Pawlak and J. Milewski, The Design of the Zeus Backing Calorimeter Data Acquisition and Trigger System, Proceedings of the Eighth Conference on Real-Time Applications in Nuclear, Particle and Plasma Physics, June 8-13, 1993, Vancouver, B.C. P 450.

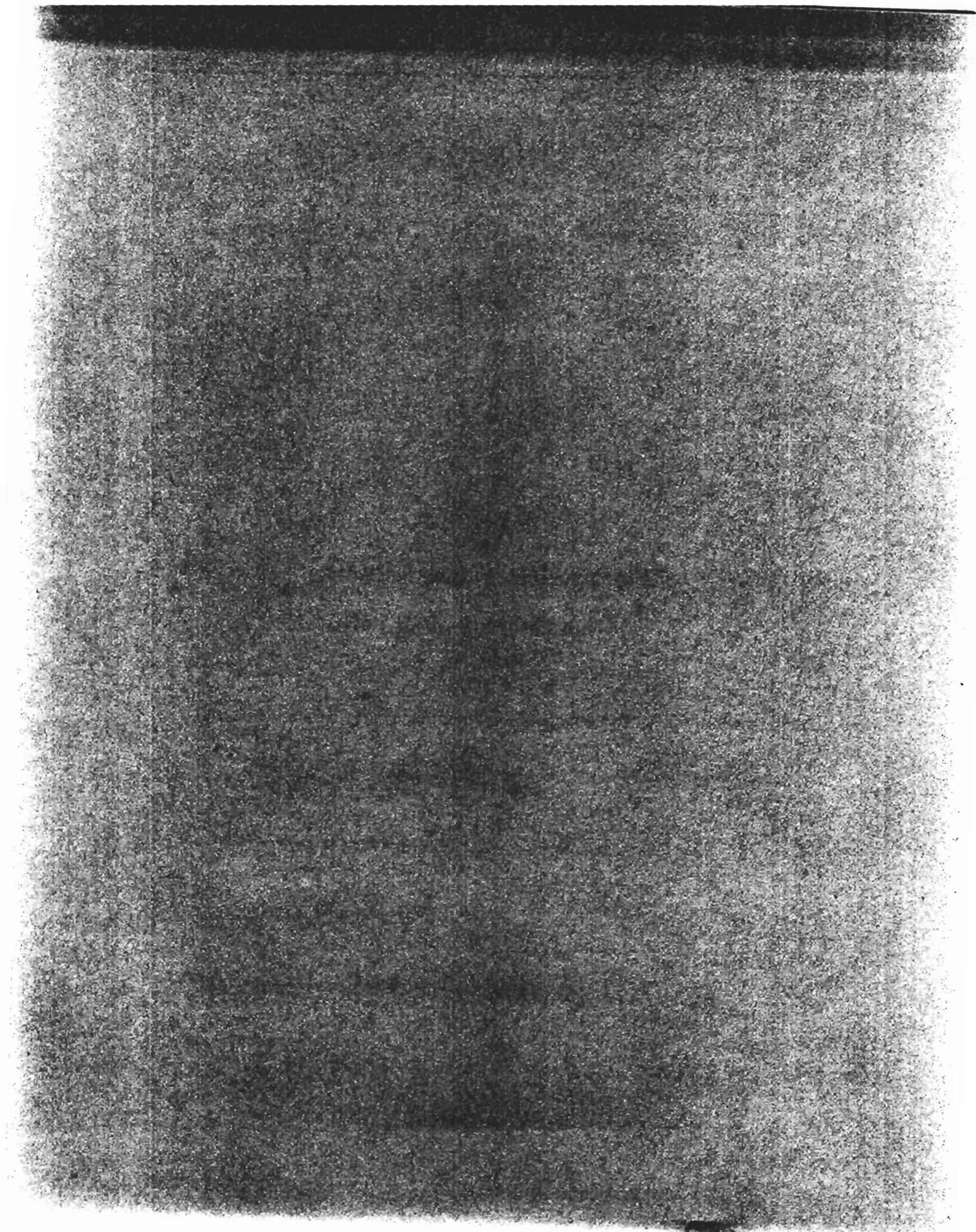
<sup>3</sup> HDMP-1000 Tx/Rx Pair, Hewlett Packard Co.

**S2-1**

**"High-Speed Switching Networks"**

**(Don Peterson - Bell Labs)**

**Tutorial on basic switching network design (fabrics, queuing, buffering, routing).**





# **Comparison of the ATM Switching Fabrics**

**Don Peterson**

**AT&T Bell Laboratories**

**Naperville, Illinois**



## **Agenda**

**Traffic Characteristics**

**Small Module Types**

**Small Module Performance**

**Large Module Types**

**Summary of Key Results**



## Traffic Characteristics Analysis

**ATM Switch Fabric should be capable of handling a wide range of traffic characteristics simultaneously:**

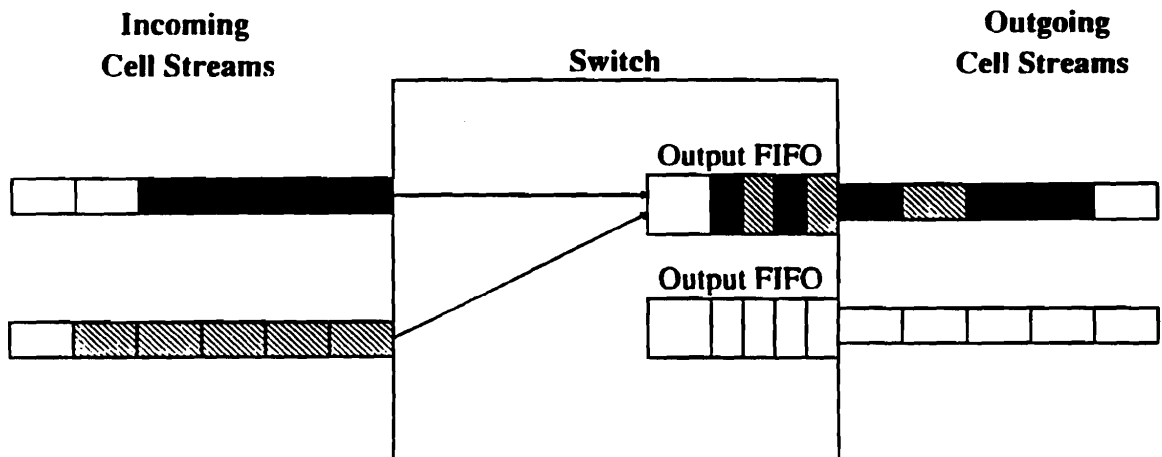
**constant bit rate traffic**

**bursty traffic**

**at high link utilization and low cell loss rates.**



## Bursty Traffic Example



**Given the Same Cell Loss Probability and Buffer Size,  
Increasing the Incoming Cell Burst Length Decreases Throughput**

Shading represents source of cell



## ***Traffic Characteristics Analysis***

---

**Burst Length: 20, 100, 200 cells**

**Link Utilization: > 80% w/o instability**

**Cell Loss Rate: < 10<sup>-10</sup>**



## ***Agenda***

---

**Traffic Characteristics**

**→ Small Module Types**

**Small Module Performance**

**Large Module Types**

**Summary of Key Results**



## **Small Module ATM Architecture Concepts**

**Self Routing -> Real Time Interpretation of Routing Info**

**Output Queuing - > Best Delay/Throughput Characteristics assuming infinite buffers**

**Result: Focus of ATM System Design is on Output Queuing Architectures that make Efficient Use of Buffering.**

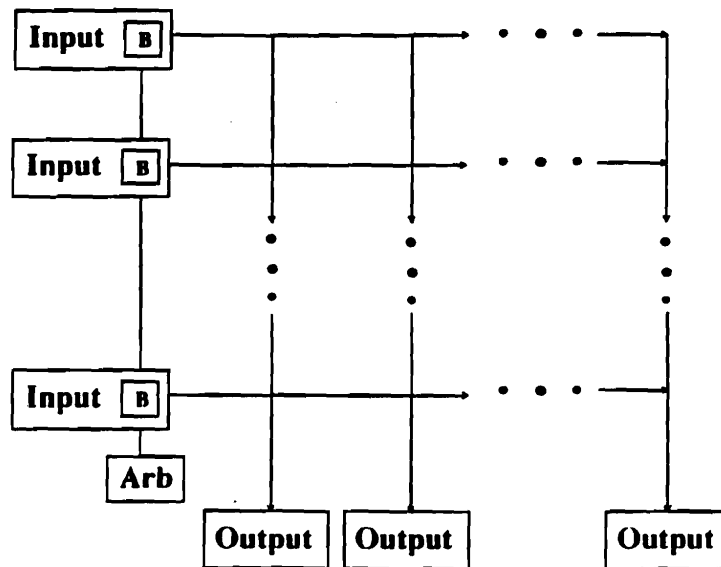


## **Small Module ATM Architecture Classification**

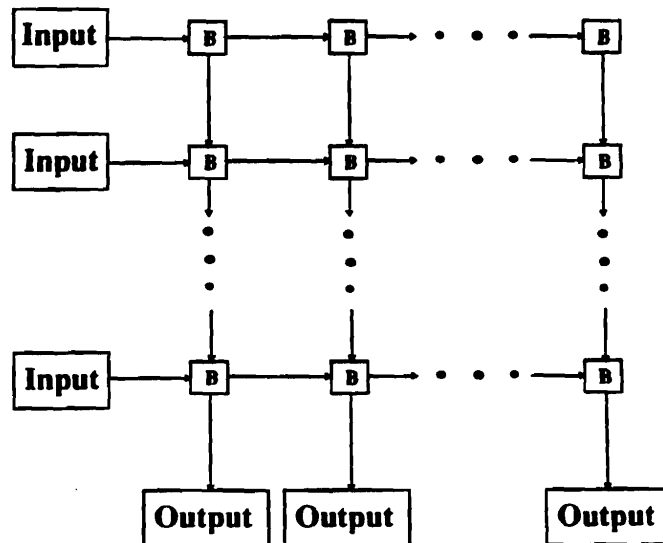
- 1) Input Buffering**
- 2) Cross Point Buffering**
- 3) Dedicated Output Buffering**
- 4) Shared Output Buffering**



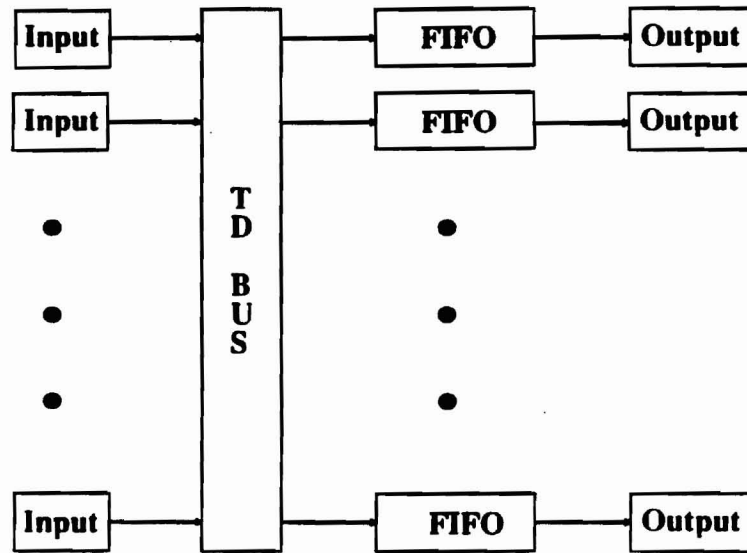
## Input Buffered Architecture



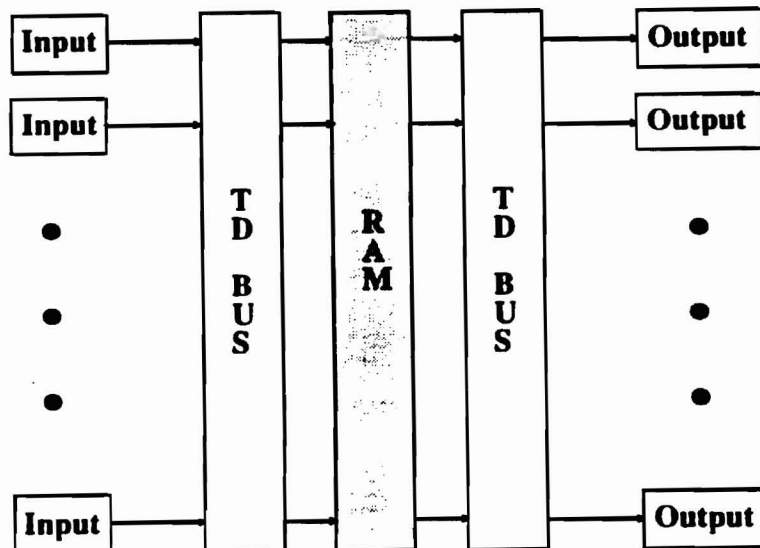
## Crosspoint Buffering Architecture



## Dedicated Output Buffering Architecture



## Shared Output Buffering Architecture



## **Agenda**

**Traffic Characteristics**

**Small Module Types**

**→ Small Module Performance**

**Large Module Types**

**Summary of Key Results**



## **8x8 Fabric Buffer Size Comparisons**

<b>Fabric Type</b>	<b>Shared Output Buffer</b>	<b>Shared Input Buffer</b>	<b>Dedicated Output Buffer</b>	<b>Cross Point Buffer</b>
<b>Buffer Size</b>	<b>650</b>	<b>1200</b>	<b>3520</b>	<b>8700</b>

**-10**  
**80% Load, Mean Burst Length of 5, 10 Cell Loss Rate**



## 8x8 Fabric Allowed Utilization Comparisons

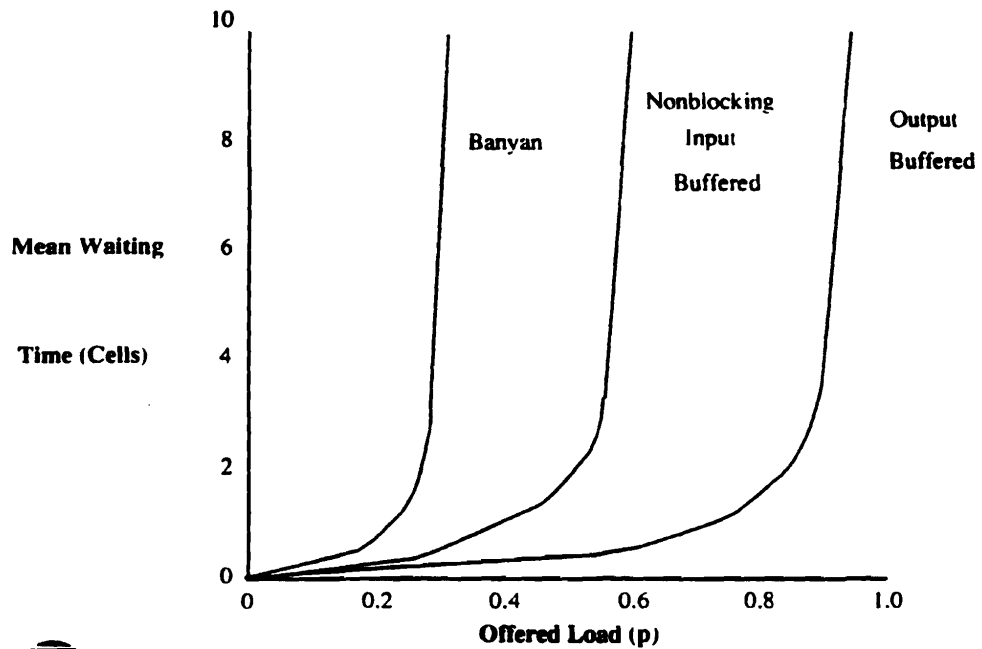
Fabric Type	Shared Output Buffer	Shared Input Buffer	Dedicated Output Buffer	Cross Point Buffer
<b>Allowed Utilization</b>	<b>88%</b>	<b>72%</b>	<b>45%</b>	<b>10%</b>

-10

Total Buffers 8000, Deterministic Burst Length of 100 Cells, 10<sup>-10</sup> Cell Loss Rate



## System Performance





## Agenda

Traffic Characteristics

Small Module Types

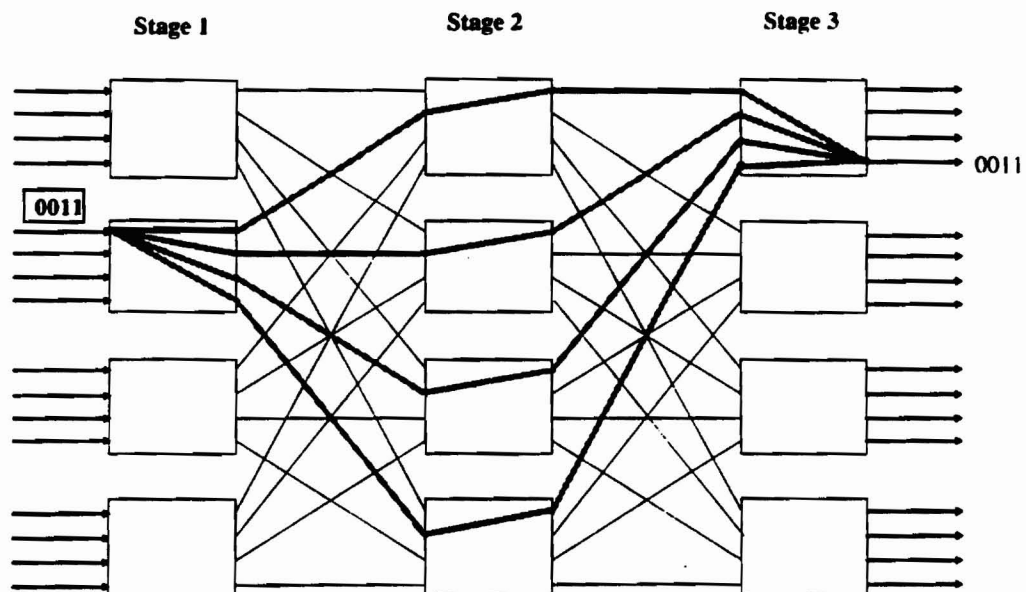
Small Module Performance

➔ Large Module Types

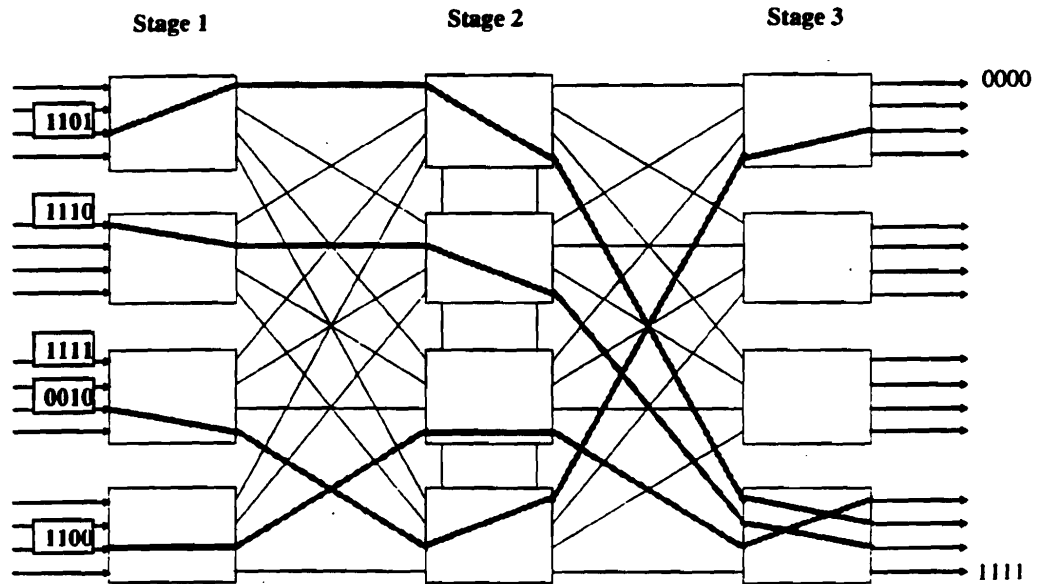
Summary of Key Results



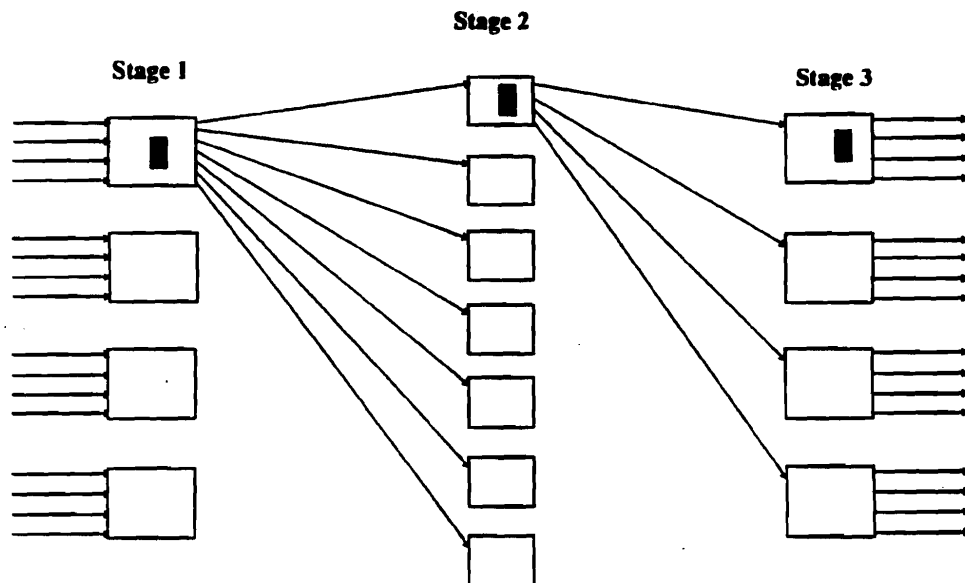
## Benes Alternate Routing Fabric Architecture



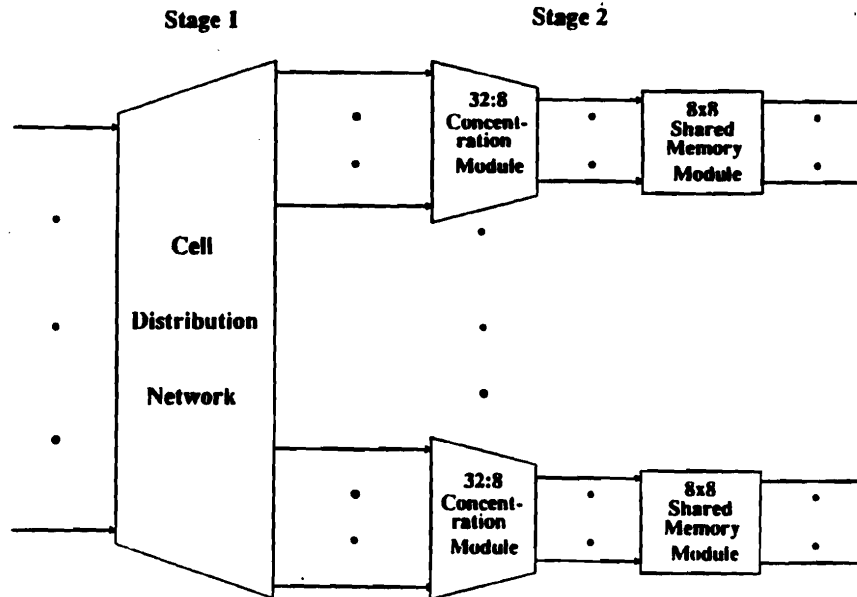
## Benes Blocking Scenario



## Clos Alternate Routing Fabric Architecture



## Large Shared Memory Growable Architecture



## Characteristics of the Growable Architecture

- 1) Ideal Performance for Small Module
- 2) Near Ideal Performance for the Large Fabric
- 3) Small Modules Reused in the Large Fabrics
- 4) Incremental Growth: 8x8, 32x32, 64x64 and larger
- 5) Stage Specialization



## ***Key Results***

---

**Shared Memory Architectures Achieve the Best Delay/  
Throughput Performance**

**Shared Memory Architectures Require the Least Buffering**

**Growable Architecture Achieves Near Ideal Performance for  
Large Fabrics**



## ***References***

---

K.Y. Eng and M.J. Karol, "High-Performance Techniques For Gigabit ATM Switching and Networking", IEEE ICC 1993 May 23-26, 1993.

K.Y. Eng, M.A. Pashan, R.A. Spanke, M.J. Karol and G.D. Martin, "A High-Performance Prototype 2.5 Gb/s ATM Switch for Broadband Applications", IEEE Globecom, December 6-9, 1992.

M. J. Karol and K.Y. Eng, "Performance of Hierarchical Multiplexing in ATM Switch Designs", ICC 1992, June 1992.

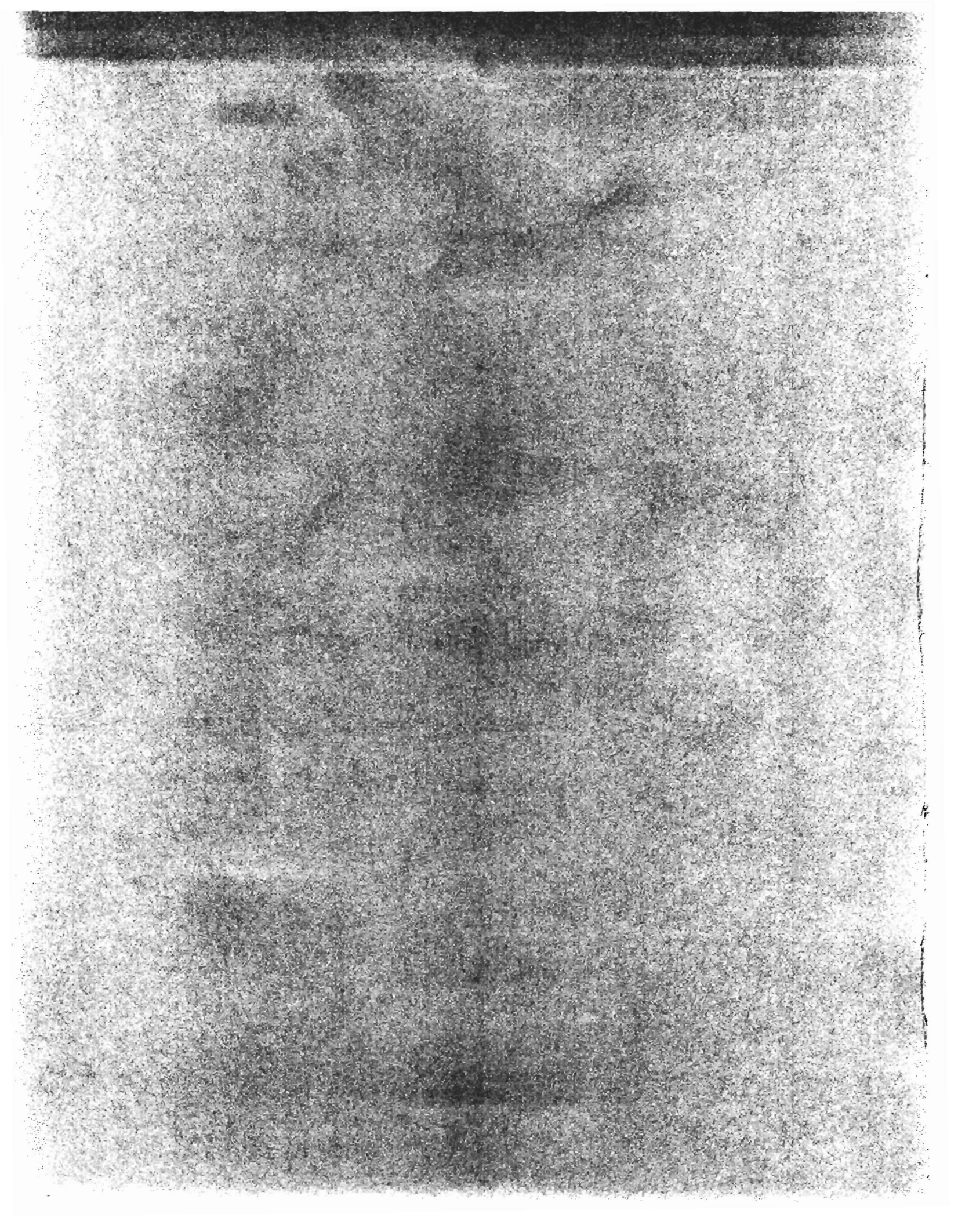
K.Y. Eng, M.J. Karol, and Y.S. Yeh, "A Growable (Packet) ATM Switch Architecture: Design Principles and Applications", IEEE Transactions Communications, February 1992.



**S2-2**

**"ATM/SONET"**

**(Jean-Yves LeBoudec - EPFL)**



---

# Aspects of ATM for Data Acquisition Systems

Jean-Yves Le Boudec  
Professor, EPFL, Lausanne, Switzerland  
FERMI-LAB Data Acquisition Conference, October 1994

---

## Abstract:

ATM is the technology chosen by International Telecommunication Union for the Broadband ISDN, but it has also been embraced by the computer networking industry as the next generation standard for both local and wide-area high speed networks. ATM network products are appearing on the market, with the promise of high volume production and the associated benefit of low cost and large feature sets.

The talk aims at providing a sufficient background in order to be able to decide whether it is worth investigating in the direction of ATM or not. The talk will explain what ATM is as a core concept, but will also introduce a number of neighbouring concepts that cannot be dissociated from ATM: physical layer and the role of SONET/SDH, Adaptation Layer, and the issue of cell loss.

---

## Contact:

Prof. Dr. Jean-Yves Le Boudec  
EPFL-LRC  
Labo Reseaux de Comm  
IN Ecublens  
1015 Lausanne, Switzerland

Tel +41 21 693 6631  
Fax +41 21 693 6610  
leboudec@di.epfl.ch

---

## 1. Asynchronous Transfer Mode : the industry choice for broadband and multimedia

ATM stands for "Asynchronous Transfer Mode". It is a standard defined by the ITU (International Telecommunication Union) for the broadband ISDN, namely, for integrated services networks at speeds above 2 Mb/s which are assumed to emerge in support of multimedia services. ATM defines a set of *interfaces* between network provider and attached equipment, but also defines to a very large extent the *technology* that is used for building the broadband, integrated services networks.

In the ITU's view, ATM is to be used as the strategic technology for broadband services, however, this would not necessarily be sufficient for making ATM a success, as is testified by the current situation in computer networking, where dominant technologies are not based on ITU standards. The key event was the perception by the majority of providers for local area network solutions that ATM would be the basis for their next generation of products. Beyond this, ATM has now become the basic strategy for almost all computer networking product vendors, as far as broadband and multimedia products are concerned, for both the local and the wide area [1].

ATM is intended to support traditional data equipment, as well as video, audio and multimedia sources.

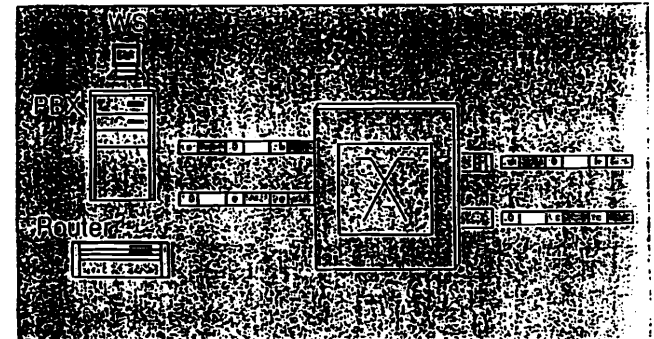


Figure 1. ATM is a fixed size, packet switching technology supported data, audio, video and multimedia sources.

In this paper, we highlight some aspects of ATM that, in our understanding, are relevant to support the decision whether to use ATM or not in a data acquisition system.

## 2. What is ATM ?

ATM is a packet switching technology, as are traditional computer networking technologies (Internet, public data services using X.25, proprietary computer networks using SNA, DECNET, etc.). However, it differs from those traditional packet switching technologies in two respects :

- ATM uses *fast packet switching* protocols
- ATM uses *fixed-size* packets, called *cells*.

### Fast Packet Switching

is an evolution from traditional packet switching that emerged in the eighties [2]. Traditional packet switching is based on analog lines and software processing of packets in the network nodes; at every network node, packets are processed extensively in order to perform such functions as correct transmission errors, limit the number of packets being sent in order to avoid congestion (flow control), or fragment packets into smaller size packets. This extensive processing is possible in software, but is not suited to hardware implementation, which would enable much higher network performance. On the other hand, it is less necessary where high quality digital links are used. These considerations paved the way to fast packet switching protocols, whose characteristics are :

- intermediate nodes perform no error correction or flow control (their essential function is thus reduced to understand a packet's address and forward it to the appropriate output port);
- all links in the network support the same maximum packet size, so no fragmentation is required at intermediate nodes;
- all error correction and flow control functions are performed in the end systems.

Another interesting feature of fast packet switching is that, since the network nodes do not perform the traditional data network functions, it is also well suited to support audio and video traffic, for which error recovery via retransmissions is not desirable.

Fast packet switching gave birth to the Frame Relay standards, a set of interfaces that support public data networks at speeds up to several Mb/s; fast packet switching is also implemented in the latest SNA versions called APPN/HIPR. ATM is, as mentioned earlier, also a fast packet switching technology [3].

### Fixed Size Cells

ATM differs from the other fast packet switching technologies mentioned above in that it uses fixed size packets, called cells. The motivation for fixed size cells comes from hardware considerations : implementing a cell switch is simpler than a variable length packet switch, and can support higher bit rates [4]. The standardized cell size (48 bytes of user information, plus 5 bytes of overhead) is a compromise between large cell sizes (64 bytes and above), supported by data overhead considerations, and small cell sizes (32 bytes and below), supported by the requirement to avoid excessive packetisation and other delays for voice services. Indeed, when voice is transmitted in packet form, the time required to build a packet grows linearly with the packet size and adds to the overall delay; voice services are very sensitive to delay because of echoes. The current cell size imposes a 5.75  $\mu$ sec delay per packetization for voice coded at 64 kb/s.

### Label Swapping

An ATM link carries a number of connections over the same physical link. Different connections are identified by a label, called Virtual Path Identifier / Virtual Channel Identifier (VPI/VCI). The VPI/VCI is 24 bits long at the user to network interface.

Figure 2 shows the basic operation of an ATM switch. The label in a cell from one input link is used to determine the correct output link, by consultation of the switching table. The label values are purely local to links, so two different connections on two different physical links may have the same VPI/VCI.

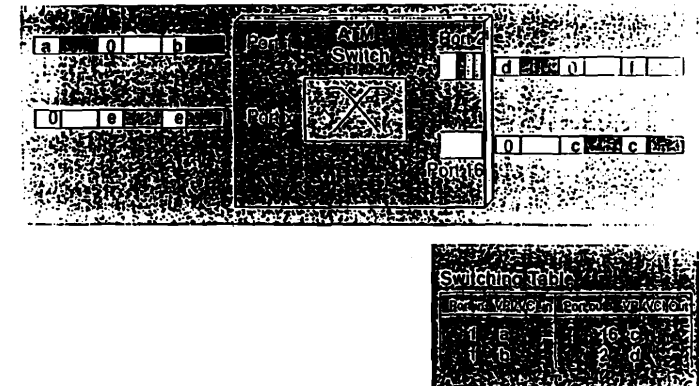


Figure 2. Label Swapping in an ATM switch



Therefore, the ATM switch needs, in general, to modify the label when transferring cells from input to output links (this is called "label swapping"). The switching table contains the new label value and is therefore also called swapping table. Note that figure 2 is a conceptual view and in many architecture the switching table is in reality distributed in a number of data structures located in input or output adapters. Before a connection can actually be used, the label swapping tables need to be configured (by a signalling or management system). This is the fundamentally connection oriented nature of ATM.

Label swapping is one of many possible ways of identifying different data flows in a network. It is used by computer network technologies such as APPN, but many other technologies use different concepts. With the RSVP protocol for instance, every packet carries the address of the destination (coded on 32 bits) plus a flow identifier that all together uniquely define the connection. This contrasts with label swapping where the label is a local identifier, that has no end-to-end significance. One overwhelming reason for selecting label swapping in the case of ATM is the small cell size that forbid the necessarily longer global addresses.

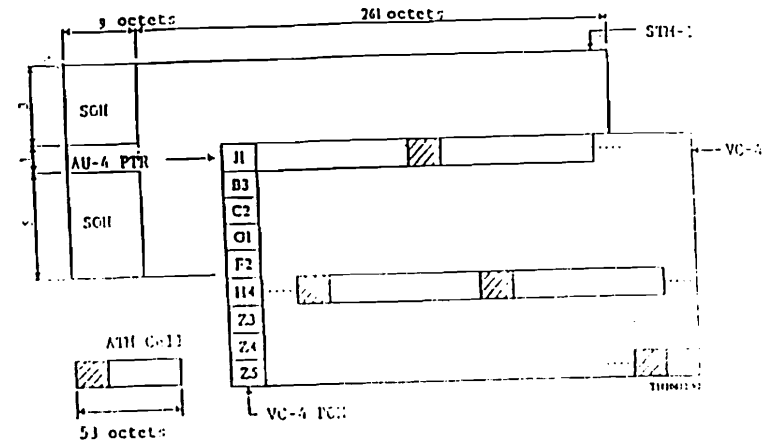


Figure 3. ATM over SDH

### Asynchronous Physical Layer

Some other physical layer systems, for short, local area links, take advantage of line coding technologies used for instance for FDDI. This is the case of the 100 Mb/s physical layer definition; it uses 4B/5B line coding, whereby non data symbols can be sent on the line. As shown on figure 4, when there is no cell to transmit, idle symbol pairs (JK) are sent. The beginning of a cell is marked by sending a TT pair. This physical layer is asynchronous in that the start of a cell can occur at any time.

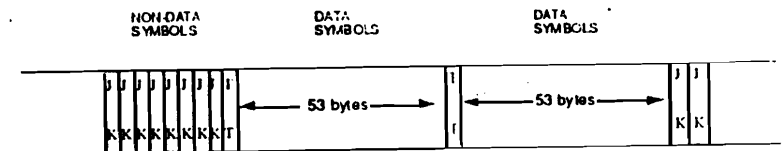


Figure 4. ATM over 100 Mb/s physical layer

### Cell based physical layer

Another possibility for detecting the cell boundaries on a bit transport mechanism is to use the header error code (HEC) for the ATM cell header. Every cell header carries 4 bytes of header information, plus one byte of cyclic redundancy check. As a result, a correct 5-byte header belongs to a specific linear code (the polynomial

## 3. The Physical Layer

Like any packet technology, ATM requires a bit transport mechanism to carry the cells over a physical medium. The ATM reference model defines a physical layer with a very large variety of options, that can be classified in one of three categories: framed, asynchronous, or cell based physical layer, depending on which method is used to recognize cell boundaries in the bit or byte stream. Bit rates vary from 1.5 Mb/s to 622 Mb/s.

### Framed Physical Layer

This is the case when an SDH, SONET 155 Mb/s or DS3 45 Mb/s byte transport mechanism is used. In such cases, the physical layer system offers a frame structure with a 125 µsec period. This frame structure can be used to identify cell positions. With DS3 systems, cell boundaries always occupy the same position inside the frame, so alignment on the frame boundary (a DS3 system function) is sufficient to be able to read the cells. With SDH/SONET systems, there is not an integer number of cells per 125 µsec frame; a pointer in the SDH/SONET path overhead is used instead to help determine cell boundaries (together with the self delineating method of the cell based physical layer below).

written from the 5 bytes must be a multiple of  $X^8 + X^2 + X + 1$ . This can be used to detect cell boundaries since it is very unlikely that a sequence of words in the ATM cell payload consistently carries 5-byte words that belong to the code (this is actually only true after scrambling the cell payload). This principle is applied for instance to ATM cell transport over the (little widespread) interface definition that uses the cell format as a framing structure (the "pure ATM interface").

## ATM and SDH

It appears from the above that an ATM network can actually use a large variety of interfaces, among which are SDH/SONET interfaces. Like in any packet switched network, it is quite possible (and usual) to mix different physical interfaces in the same system. There is no need for SDH in order to build an ATM system, even though it is likely that this will be one of the dominant types of interfaces. In any case, if SDH interfaces are used in a local, private environment most of the complex SDH operation and maintenance functions intended for public networks need not be implemented.

## 4. The Adaptation layer

ATM uses short, fixed size cells, whereas data protocols generate variable length, longer packets. The necessary adaptation is fortunately well defined in a now stable standard : the ATM Adaptation Layer 5 (AAL5). As illustrated in Figure 6, AAL5 takes as input a variable length message (up to 64 bytes), add 8 bytes of control and error checking pads the message to make it an integer number of 48 bytes, and transmits the resulting segments in ATM cells. In order to perform re-assembly at the receiving side, the last segment is differentiated by a bit set in the ATM cell header. This set of function is called segmentation and re-assembly (SAR). It is very efficient in that it uses extremely little overhead. AAL5 offers a connection service, with a one to one mapping with the underlying ATM connection.

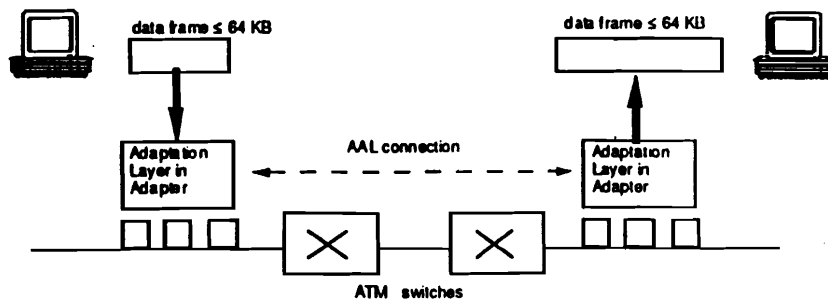


Figure 5 : ATM Adaptation Layer Connections support the transfer of large data blocks

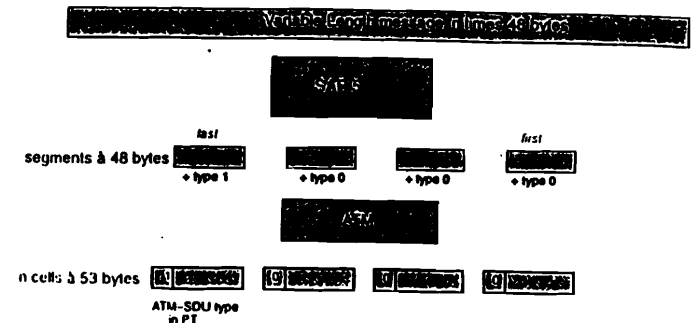


Figure 6 : Segmentation with AAL5

AAL5 is implemented in end-systems, for example on workstation adapters in programmable hardware or dedicated circuits. Segmentation and re-assembly are performed only at both ends of an AAL connection, not in the intermediate switching points, which handle only ATM cells headers without (in principle) having to know about the cell contents.

There exist other AAL types. AAL1 supports the emulation of digital circuit transport (for example DS1 at 1.5Mb/s or E1 at 2 Mb/s). AAL 3/4 is an alternative to AAL5 that uses more overhead (for instance 4 bytes out of 48 in every segment), but supports multiplexing of several AAL connections on one ATM connection. It is less widespread than AAL5.

The complete AAL is not only segmentation and re-assembly. Additional functions can be defined to make the AAL connection reliable (HDLC-like functions), or to make it emulate existing services such as the frame relay care service. It is worth noting that an ATM equipment, with AAL adapters and ATM switches, offers connections for transferring very large blocks of data (up to 64 Kbytes) even though the ATM building block (the cell) is smaller.

## 5. Cell losses

As any packet switching technology, ATM requires buffering cells at intermediate points, and potential cell losses may occur when a buffer overflow. ATM networks avoid cell losses by one of two methods.

- contract based connections,
- best effort connections.

Both types of connections can exist in the same network, and many networks today offer only contract based connections.

## Contract Based Connections

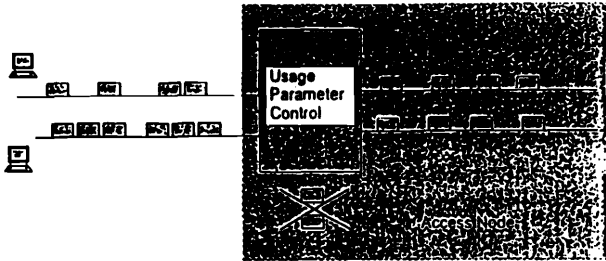


Figure 7 : Source Policing, or Usage Parameter Control

With this method, every connection comes with a *traffic contract*, negotiated at call establishment between the network and the user. The traffic contract specifies such things as the maximum peak rate, and maybe a maximum sustained rate and burst size. For example, a connection may be specified with a peak rate of 10Mb/s, a maximum sustained rate of 1Mb/s, and a burst size of 1 Mbyte (meaning that at most 1 Mbyte of data can be sent at a time at the full peak rate of 10 Mb/s). In reality, the parameters are defined in a slightly more complex way, using a formal definition, called the generic cell rate algorithm (figure 8).

This contract enables the network control software to decide whether the connection can be supported by the current state of the network. Based on offline buffer modelling studies, this guarantees that cell losses occur only rarely (usually with a probability less than  $10^{-9}$ ). In some cases, a network may offer several qualities of connections, with lower quality connections suffering from a higher cell loss probability ( $10^{-5}$ ). In all cases, the contract also guarantees the user a specified throughput, much like in a circuit-switched system.

This method requires that the connection behave, at worst, according to the network control's expectation. Since connection rates are limited only by the physical link rates, it is necessary for the network to implement a mechanism to enforce the contract. This is called Source Policing, or Usage Parameter Control (UPC). UPC is implemented at network boundaries, on the network side of the user-network interface. Cells that violate the contract (cells in excess) are discarded, or are marked with a lower priority using one bit in the cell header (cell loss priority bit). Cells with lower loss priority are discarded first in case of buffer saturation.

The combination of UPC and network control software is thus able to guarantee quasi-loss free operation. One key aspect of network control is the amount of capacity that need be allocated to every connection : for a connection with, say a

peak rate of 1 Mb/s, sustained rate of 10 Mb/s, the network will allocate a value lying somewhere between 1 and 10 Mb/s, depending on a number of parameters such as link buffer sizes, maximum burst duration for the connection, and the aggregate characteristics of the other existing connections [5].

## GCRA for Source Policing

$t$  : arrival time  
 $T_e$  : peak Interval  
 $\tau$  : tolerance  
 $t_{at}$  : theoretical arrival time

```
if ( t < tat -  $\tau$  )
    result=NONCONFORMANT;
else {
    tat = MAX ( t, tat ) + Te;
    result=CONFORMANT;
}
```

Figure 8. The generic cell rate algorithm

In cases where the peak rate is large (10% of link rate or more), it is however difficult to allocate significantly less than the peak rate; in other words, the statistical gain with this method is low for very bursty sources. This and other reasons motivated the introduction of best-effort connections.

## Best Effort Connections

This type of connection is not associated with a contract guaranteeing some throughput. In contrast, the actual throughput attainable on such a connection depends on the instantaneous states of the network. The ATM Forum calls this service "available bit rate (ABR)", which indeed means that best effort connections are intended to utilize the network capacity that is either unallocated, or allocated but unused.

The available capacity is thus shared between best effort users, dynamically, and without reservation. Of course, if nothing is done, buffer overflows are likely to occur as soon as the network is not extremely lightly loaded. Cell loss avoidance mechanisms are thus necessary.

Proprietary solutions by DEC (credit based) or IBM (backpressure based) exist for local area networks. They are based on hop-by-hop mechanisms, and allow a loss-free operation. A protocol between user and network regulates the admission of cells (credit or stop and go), and once a cell is admitted, the network will not discard it for reasons of buffer overflow. Inside the network, a buffer to buffer protocol (implemented in hardware) avoids cell losses, possibly at the expense of spreading congestion from a "hot spot" area back to the sources. Fairness among connections is guaranteed by implementing the protocol on a per-connection basis. Of course, if such protocols guarantee loss-free operation, there is, in contrast, no guarantee about the delay for individual cells to traverse the network.

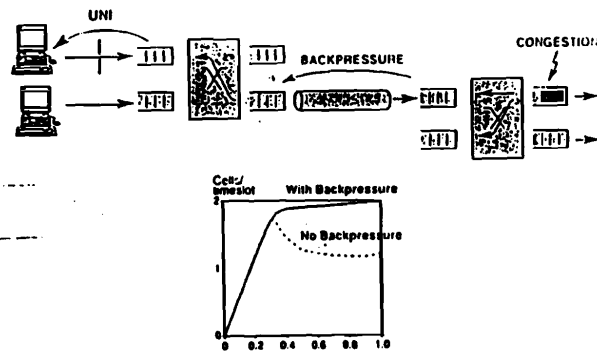


Figure 9 : Backpressure protocol supporting loss-free operation for the best-effort service

These solutions do not scale well to networks with very long links (they require large buffers). Solutions that use more dynamic buffer allocations are being researched. Alternative to the hop-by-hop solutions mentioned above are the end-to-end, or edge-to-edge solutions used for instance in Frame Relay or APPN/HPR : the ends of best effort connections sense the amount of traffic and the delay characteristics, and adjust their rate in reaction. Such solutions are believed to provide reasonably low loss probabilities under reasonable traffic assumptions.

For data acquisition systems, it is probably worth remembering at this point that local area ATM networks are able to provide loss-free best effort services.

- ATM uses fixed size, small cells, but AAL connections provided by quasi all commercial products provide a standard means for transferring blocks up to 64 Kbytes at high speeds (up to 600 Mb/s) and with complete networking solutions (namely, supporting various sizes and distances).
- SDH/SONET is suited to transport ATM cells but is not the only available transmission technology.
- Loss-free, best effort ATM services are available for the local area; quasi-loss free, contract based services exist for wide and local area.

ATM is an industry and services consensus, and even if some alternative technologies exist, the advanced standardization status of ATM, together with its intrinsic benefits and its complete network solution, will very likely guarantee high volume, if not low cost, development of components and solutions. ATM is appearing as the unchallenged next generation high speed local area network, and will be dominant in network backbones, both private and public. It is less clear, in contrast, whether ATM will also make the last step and penetrate the workstation world as Ethernet and Token Ring did.

## 7. How to know more

The interested reader should start by getting the "Frequently Asked Questions" (FAQ) list about ATM, available by FTP from [cell-relay.indiana.edu](ftp://cell-relay.indiana.edu). This FAQ contains an updated list of specific and tutorial documents. Various courses are also organized throughout the world ([cpit@di.epfl.ch](mailto:cpit@di.epfl.ch)).

## 6. The ATM perspective

In this very short overview, we tried to highlight a few features of ATM as of end of 1994 that are relevant to DAQs.

---

## 8. References

- [1] Jean-Yves Le Boudec, Erich Port, Hong Linh Truong  
"Flight of the FALCON"  
IEEE Comm. Magazine, vol. 31, No. 2, Feb. 1993, pp. 50-56
- [2] L.T. Wu, S.H. Lee and T.T. Lee  
"A packet Approach to Broadband Networking"  
ICC 1987, Seattle 1987
- [3] J. Gray and M. Peters  
"A Preview of APPN High Performance Routing"  
LAN Interconnection 1993, Raleigh
- [4] W. Denzel, A. Engbersen and I. Iliadis  
"A Flexible Share-Buffer Switch for ATM at Bb/s Rates"  
to appear in Computer Networks and ISDN Systems
- [5] L. Gün and R. Guérin  
"Bandwidth Management and Congestion Control Framework of the Broadband Network Architecture"  
Comp. Net. and ISDN, vol. 26 No 1, 1993
- [6] J.-Y. Le Boudec  
"The Asynchronous Transfer Mode : a Tutorial"  
Computer Networks and ISDN Systems, vol.24 No 4, 1992

**WHAT IS ATM ?**

## **Aspects of ATM**

**Jean-Yves Le Boudec**  
Professor, EPFL, Lausanne, Switzerland

FERMI-LAB Conference, October 1994

### **Abstract:**

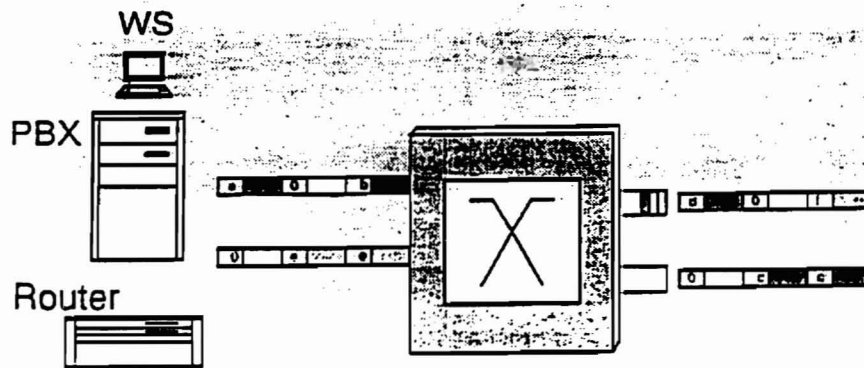
ATM is the technology chosen by International Telecommunication Union for the Broadband ISDN, but it has also been embraced by the computer networking industry as the next generation standard for both local and wide-area high speed networks. ATM network products are appearing on the market, with the promise of high volume production and the associated benefit of low cost and large feature sets.

The talk aims at providing a sufficient background in order to be able to decide whether it is worth investigating in the direction of ATM or not. The talk will explain what ATM is as a core concept, but will also introduce a number of neighbouring concepts that cannot be dissociated from ATM: physical layer and the role of SONET/SDH, Adaptation Layer, and the issue of cell loss.

### **Contact:**

Prof. Dr. Jean-Yves Le Boudec  
EPFL-LRC  
Labo Reseaux de Comm  
IN Ecublens  
1015 Lausanne, Switzerland

Tel +41 21 693 6631  
Fax +41 21 693 6610  
lehoudec@di.epfl.ch



What is ATM ?

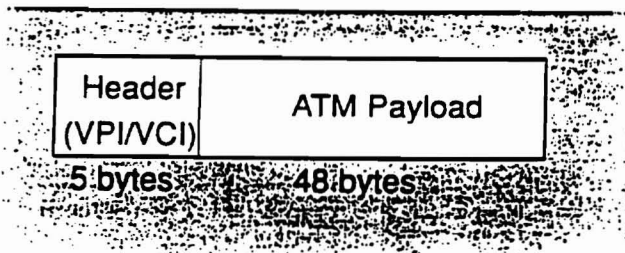
- technology
- service interface
- standard

For whom is ATM ?

- LAN
- private WAN
- public WAN

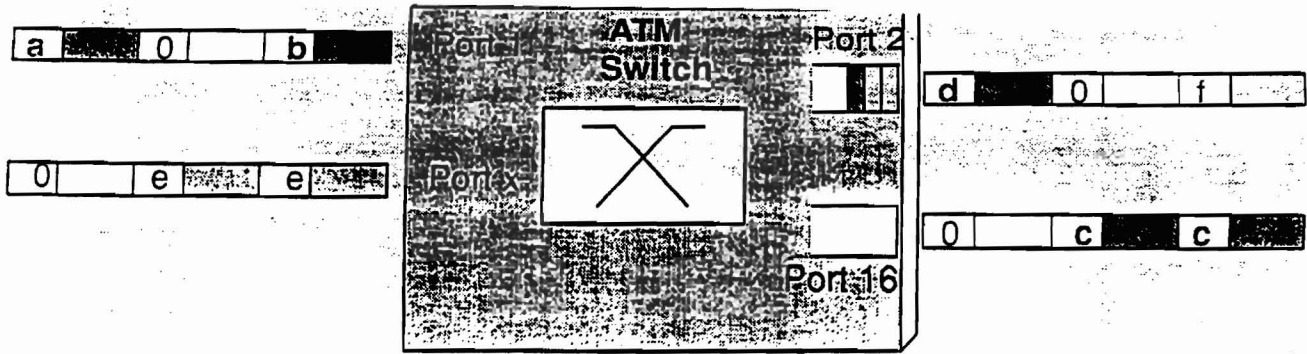
## ATM Cell

FL-2



1 cell = 53 bytes

- Fixed Size
- Packet (cell with VPI/VC)
- One Physical Channel at one interface
- All services
- Simple Protocols
- Losses not corrected

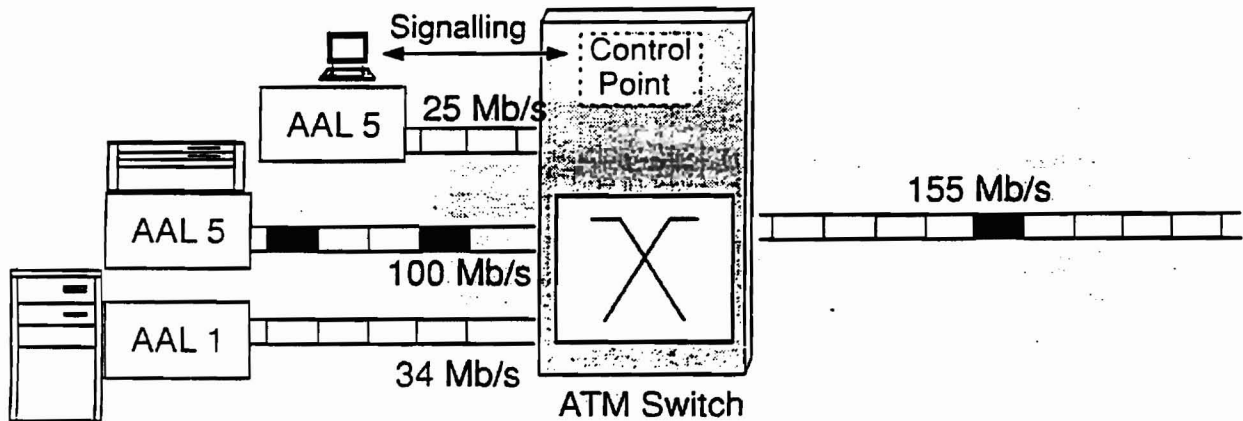


**Switching Table**

Port in	VPI/VCI in	Port out	VPI/VCI Out
1	a	16	c
1	b	2	d

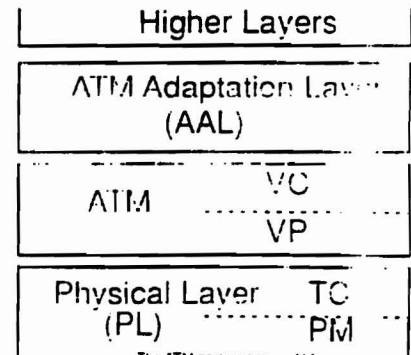
ATM = fixed size, packet switching

- hardware
- delay < 1 msec => all services
- label swapping
- connection setup



ATM is standard

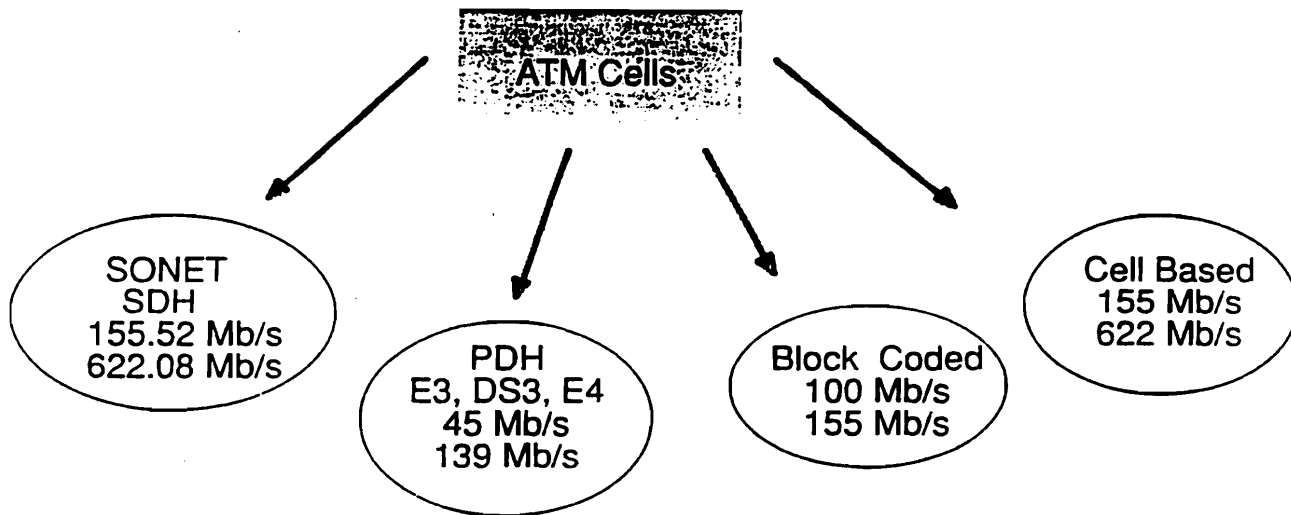
- ATM links
- services
- management
- control
  - just beginning
- ATM forum, TSS





# PHYSICAL LAYER

## Physical Layer

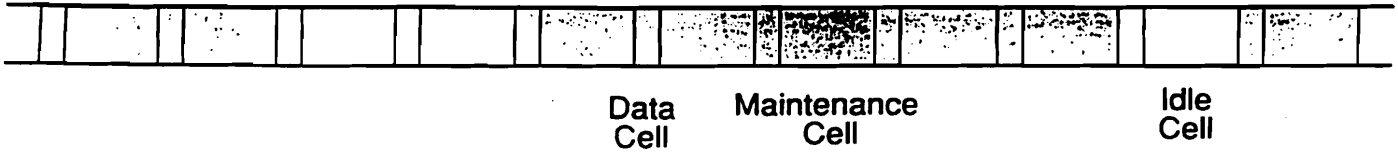


## Unframed Physical Layer

Transmission System

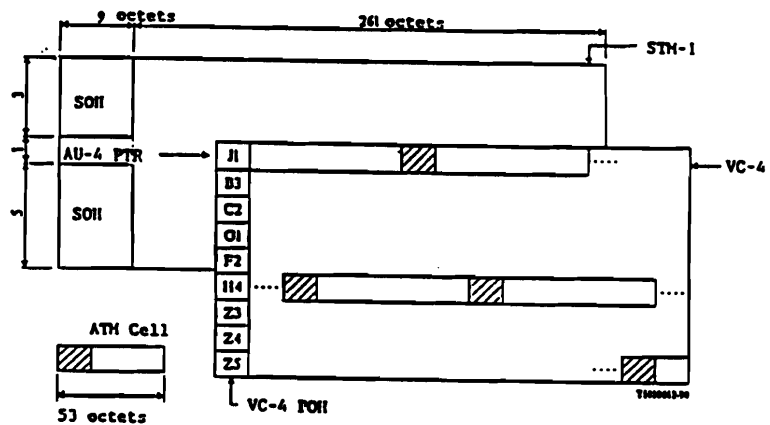


Bit or Byte Stream

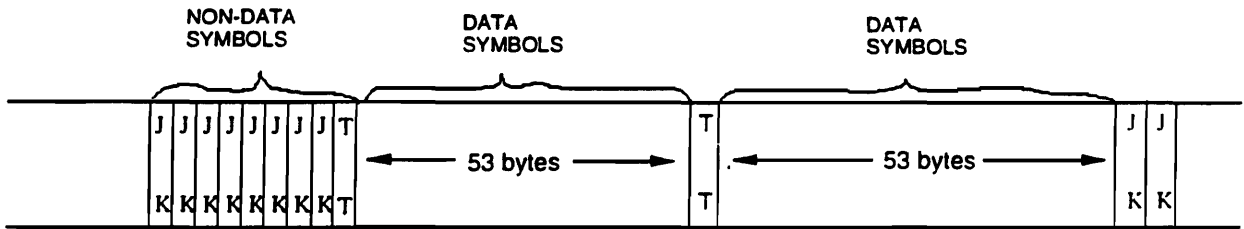


- Cell Synchronization on Header
- Idle cells for slip or stuffing
- Different links not bit synchronized
- PDH, CMI, ...

## Framed Physical Layer



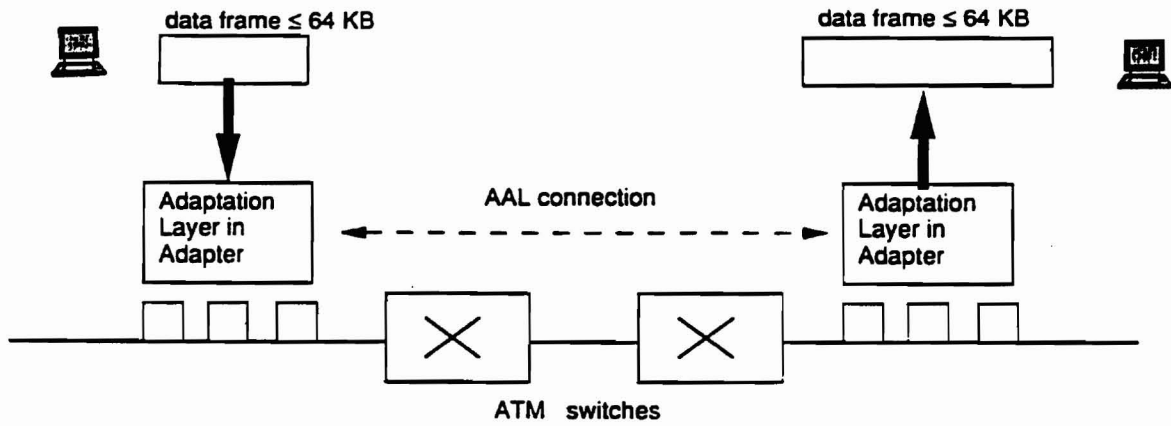
- SDH
- Block Coded (ATM forum)
- DS-3 (ATM forum)



- FDDI - PMD at 100 Mb/s;
- MUNI at 25.6 Mb/s is similar;

## ADAPTATION LAYER

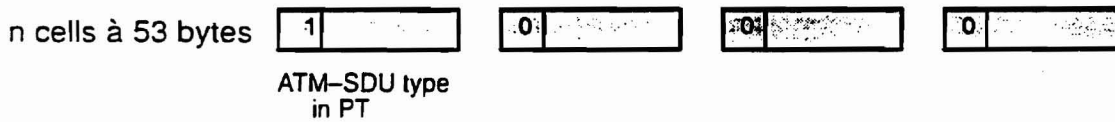
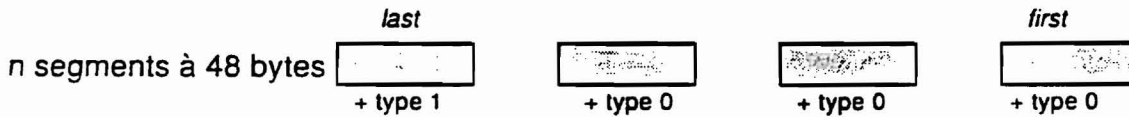
# Adaptation Layer for Data



- part of ATM standards
- hardware adapters
- available today
- performed only at end-systems

## SAR 5

Variable Length message, n times 48 bytes



## **CELL LOSSES**

FL-11

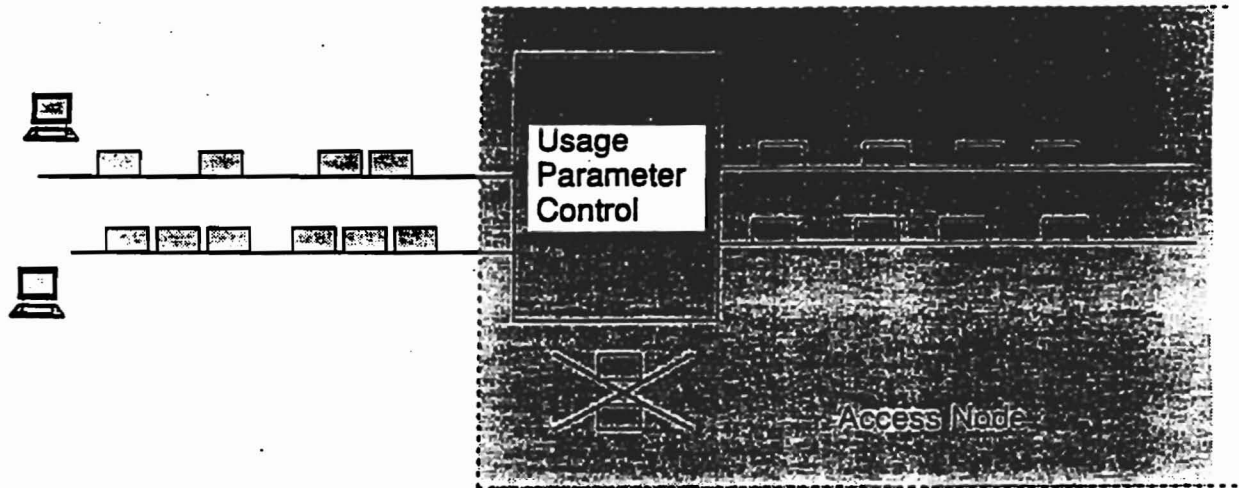
## **CELL LOSS AVOIDANCE**

ATM network avoid cell losses by one of two ways

- contract based connections
- best-effort connections

Both can exist in parallel on one network

## CONTRACT BASED CONNECTIONS



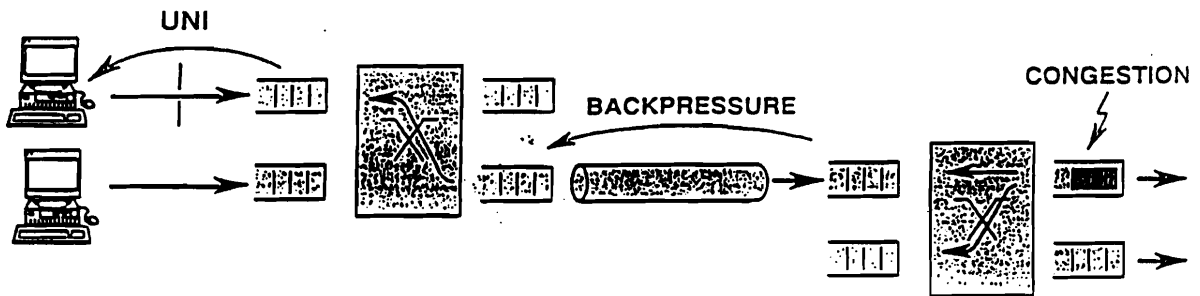
- contract at connection setup
  - maximum peak rate
  - maximum burst size
- enforced by source policing
- network control software guarantees quasi-absence of buffer overflows
- guaranteed throughput to user

FL-13

## BEST EFFORT CONNECTIONS

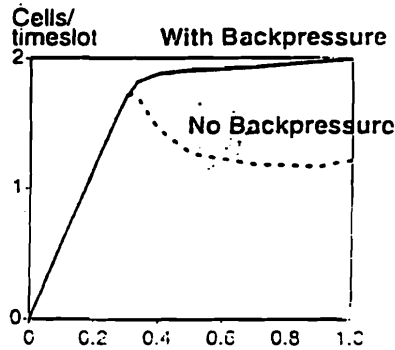
- no guaranteed throughput - best effort
- capacity shared dynamically without reservation
- flow control being specified at ATM-Forum for loss avoidance local area
  - products based on hop-by-hop backpressure or credit are loss-free

# Backpressure



## Desired Features

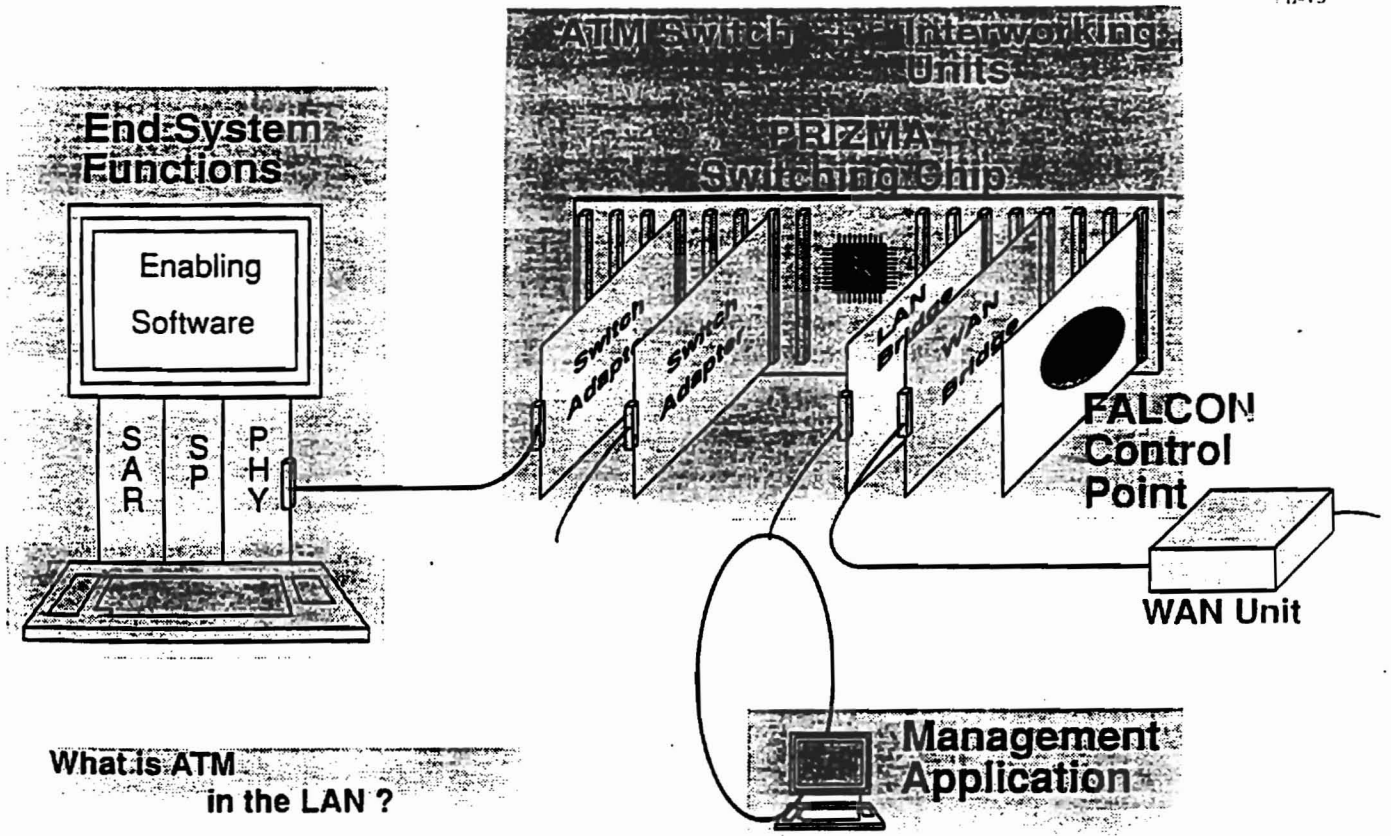
- No buffer over-runs
- No buffer under-runs
- Fairness
- No deadlocks
- Robustness
- Reserved-bandwidth traffic not affected
- VC-level granularity



## Implementation Issues

- Flow control signal in GFC field or private cell
- UNI: Implementable with SARA chipset
- Complexity of flow control and scheduling at VC level

## PRODUCTS



What is ATM in the LAN ?

## CONCLUSION



# CONCLUSION

c

- ATM the consensus on high-speed packet switching
- hardware and large scale benefits
- "ATM" is cell switching + adaptation layer + physical layer + signalling
- loss-free, best effort local areas exist  
quasi loss-free, contract base wide and local area
- SDH / SONET one of several bit transport mechanisms suited to ATM
  - ATM does not require a uniform bit transport mechanism

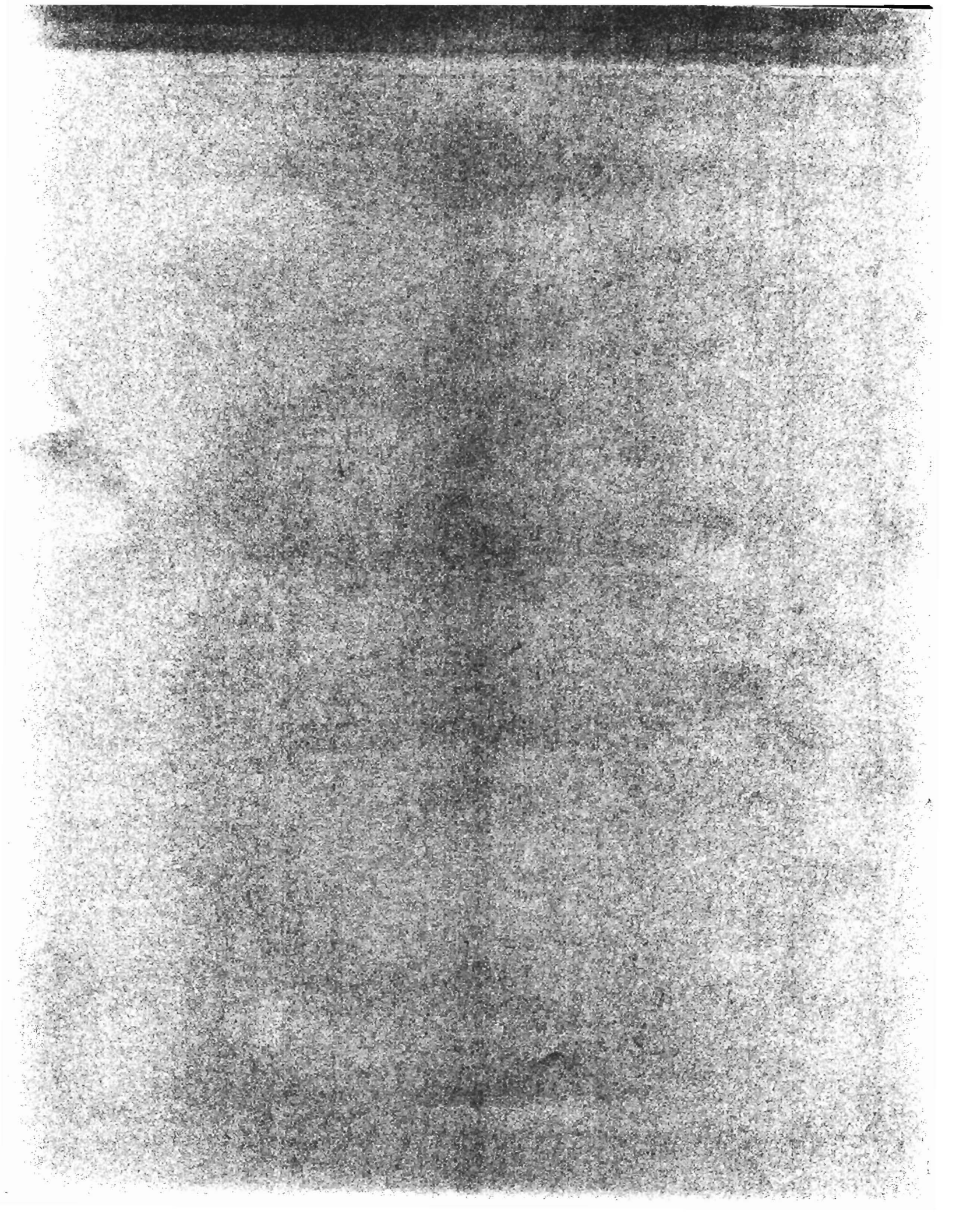


## **S2-3**

### **"Fibre Channel"**

**(Roger Cummings - Storage Technology)**

**This talk will provide an introduction to the Fibre Channel (FC) interface as defined by Technical Committee X3T11. It will describe the development of FC, the architecture of FC, and the FC definition with specific reference to data acquisition applications. An update of the status of the sixteen FC standardization projects will be provided, as will a description of the future enhancements planned for FC. The activities of industry organizations related to FC will be described, and the talk will close with an overview of the status of FC within the industry in general and the workstation segment in particular.**



**StorageTek**

**INTERNATIONAL DATA ACQUISITION CONFERENCE**

**FIBRE CHANNEL INTRODUCTION**

**Roger Cummings**  
**Chair, ANSI TC X3T11 (IPI, HIPPI, FC)**  
**Senior Advisory Engineer, StorageTek**

**October 26, 1994**

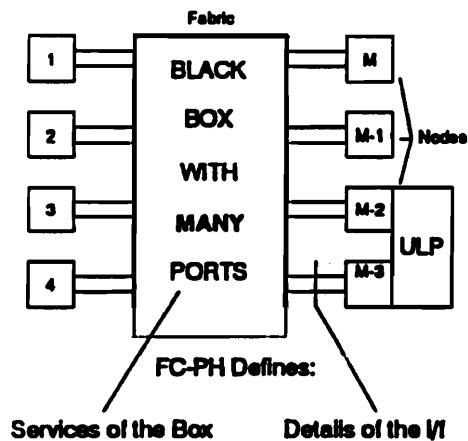
**TOPICS**

- **Architecture and details of Fibre Channel**
- **Where did Fibre Channel come from?**
- **Data Acquisition applications of Fibre Channel**
- **Summary**

## FIBRE CHANNEL ARCHITECTURE

- Composed of two entities:
  - Functional definition of a Fabric - an active, intelligent interconnection mechanism:
  - Complete definition of interface between Fabric and user equipments (Nodes)
- Functional model ensures that Fabric internals are transparent to Nodes
  - Change system cost/performance by changing only Fabric - no change to Nodes

## FIBRE CHANNEL ARCHITECTURE

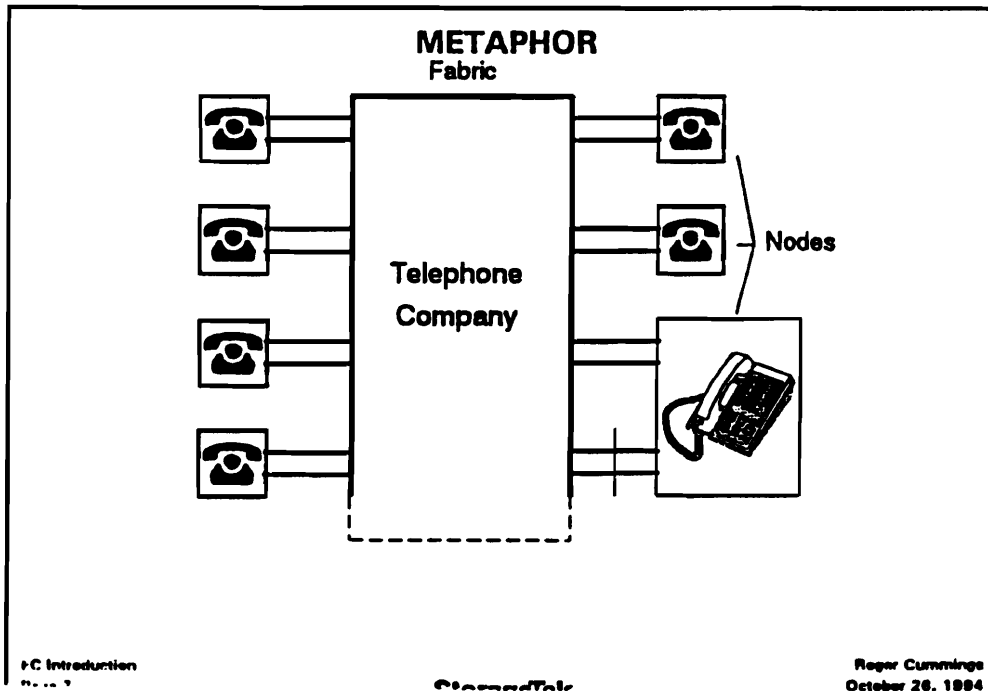


## FIBRE CHANNEL FABRICS

- Supports many different topologies, not just switches:
  - Arbitrated Loop for low-end cost-effectiveness (virtual Fabric)
  - Switched Fabrics for high system bandwidths (many parallel transfers) and excellent expandability
  - Many other application-specific Fabric possibilities
- Key to the architecture is that Fabric provides management, control and monitoring functions
  - Nodes are just data sinks and sources

## FABRIC OVERVIEW

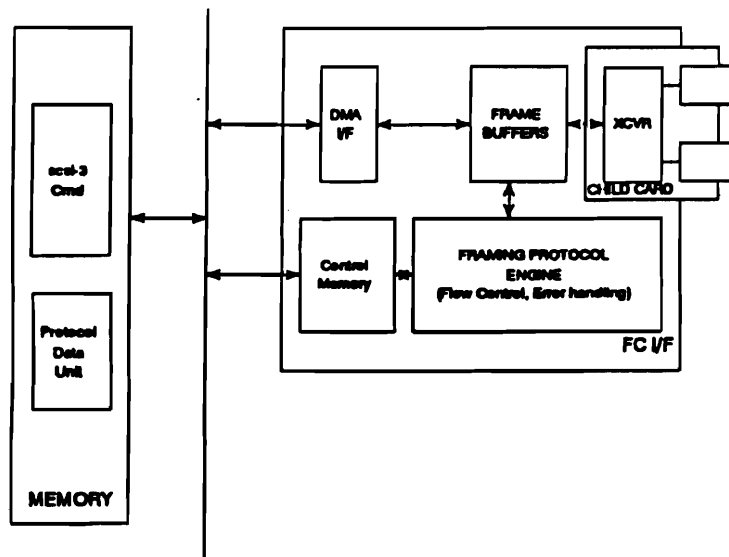
- Fabric provides three classes of service:
  - Class 1      Dedicated Connection and Ports (i.e wire)  
Guaranteed delivery, end-to-end flow control
  - Class 2      Multiplexed, switched, Many to Many Ports  
Guaranteed delivery with buffer-to-buffer & end-to-end flow control
  - Class 3      Datagrams  
"Ship and Pray", delivery not guaranteed
- "What goes in must come out" (down to frame level)
- Single-level address domain (24 bit address)



- WHY THIS APPROACH?**
- **Separates management and control from "users":**
    - Many protocols to be carried on FC were designed for short cables only, not network management
    - Locates management in Fabric where information is available
    - Avoids the "chain effect" (only as strong as weakest link)
  - **Facilitates putting Node interface functions in hardware:**
    - Single implementation of segmentation/reassembly, flow control & error recovery for all protocols
    - Leverages capabilities of VLSI for cheap interfaces
    - Allows high utilization of gigabit and faster links
- FC Introduction  
Page 8
- StorageTek
- Roger Cummings  
October 26, 1994



## INTERFACE ARCHITECTURE



## INTERFACE ARCHITECTURE

- FC provides a transparent delivery service:
  - Protocols define "data blocks" in memory
  - FC interface card fetches data from memory, disassembles data into a Sequence of frames
  - Sequence transmitted thru Fabric to destination
  - Destination interface card reassembles frames and recreates the original data block in the destination machines memory
- All flow control, error handling and most levels of recovery performed by the interface cards, transparent to the software

## INTERFACE DETAILS

- Four data rates supported - 133, 266, 531 & 1062.5 megabaud (data carrying capacity of 12.5, 25 50 & 100 megabytes/s)
- Optical variants use led & laser transmitters, multimode and single mode fibre:
  - Shortwave laser variants most popular & cost effective (2km@266, 1km@531, 0.5km@ 1062.5 megabaud)
- Electrical variants use ECL transmitters, video coax, subminiature coax (all rates) & STP (266, 531 only)
- Philosophy is that FC will run over what's available

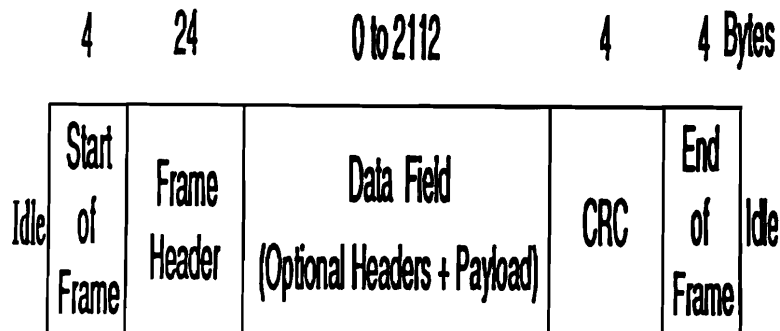
## COMPATIBILITY

- Proliferation of variants causes little incompatibility in the field:
  - All parts of an FC subsystems do not have to operate @ same rate, use the same technology
  - Only F\_Port has to match N\_Port technology
- De-facto child card scheme simplifies transceiver interchange in the field:
  - Available at all rates and with all technologies
  - Gigabit Link Module spec in public domain
  - All high-speed serial paths restricted to module
  - Simple parallel TTL-level interface to rest of interface card

## INTERFACE DETAILS

- Coding scheme is 8B/10B (same as ESCON, IBM patent generally available for FC):
  - One Special character used in first byte of four byte structure gives simple byte and word alignment mechanism
  - Simple encode and decode in parallel logic
- Framing protocol uses variable-length frame structure with fixed format 24 byte header:
  - Specifically designed for implementation in hardware
  - Each frame is self-describing

## FC FRAME FORMAT

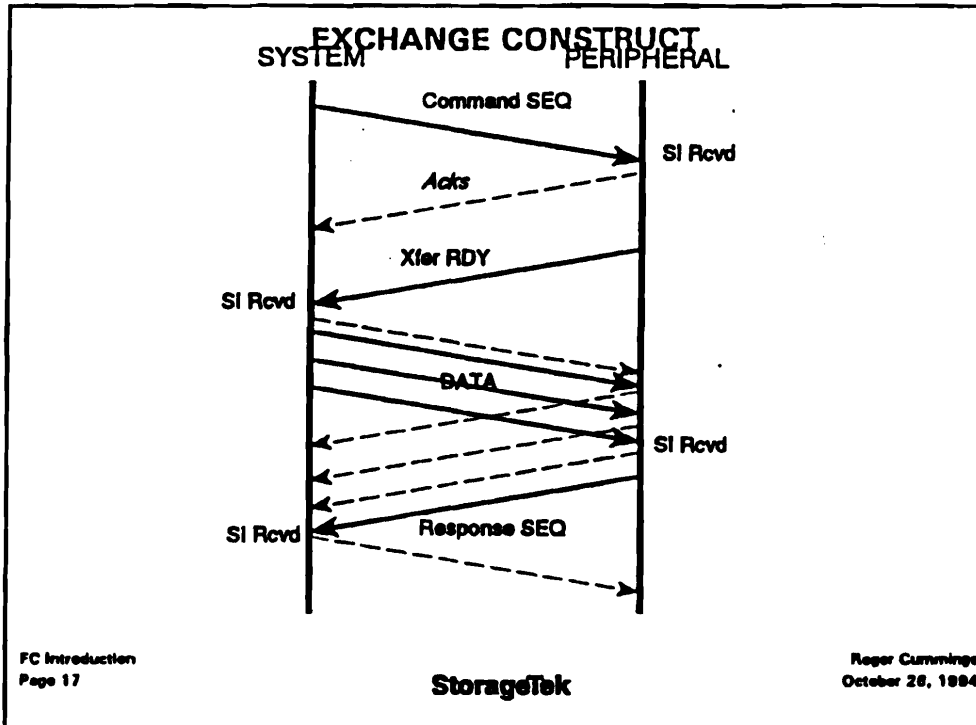


## INTERFACE DETAILS

- Powerful look-ahead flow control for high-performance long distance operation
- Class of Operation controlled by Start of Frame only
- Fixed format 24 byte header includes:
  - 24 bit source and destination addresses
  - Type of protocol encapsulated in data field
  - Identifiers for Sequence and Exchange Constructs
  - Optional gather and scatter support
  - Optional extensions for security applications, 64 bit addresses
- Data field size holds 2K of data plus its own 64 bit header

## EXCHANGE CONSTRUCT

- Peripheral command sets have half duplex command flows:
  - Use identifiers to relate received info to previous transmissions
  - Each protocol uses different details
- Exchange provides generic construct to support protocols designed for bidirectional but half duplex bus schemes:
  - Explicit passing of permission to transmit data
  - Facilitates use of existing drivers with new FC interfaces
  - Helps ensure distance-insensitive operation
  - Direct index into control structures at each end
  - Allows complete Operations to be performed in hardware
  - Single, optional interrupt upon completion, or chaining



- ### MULTIPLEXING
- **Frames, Sequences and Exchanges can all be multiplexed:**
    - **Frames from one Sequences interleaved with Frames of other Sequences**
    - **Multiple Exchanges active to a Node**
    - **Oriented to protocols that support deep command queues**
    - **Integrated with the flow control scheme**
    - **Controlled in real time by hardware**
    - **Leverages the abilities of multi-threaded DMA**
- FC Introduction  
Page 18
- StorageTek**
- Roger Cummings  
October 28, 1994

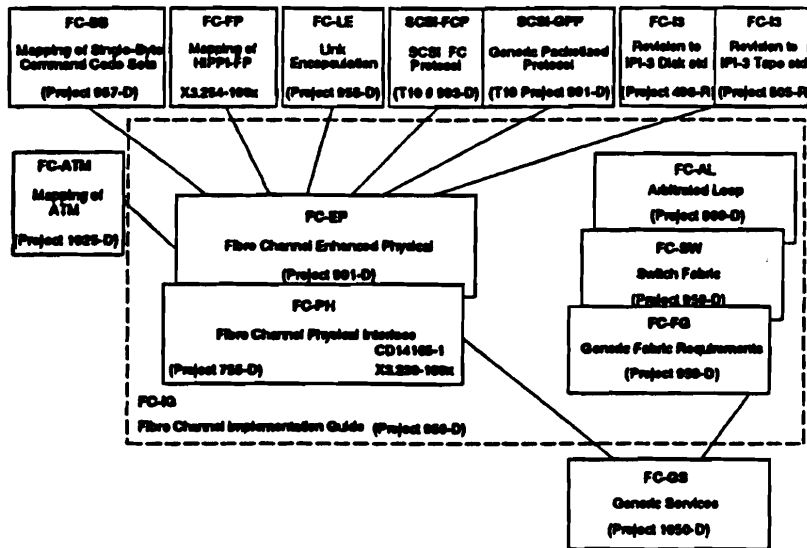
## FC SUMMARY

- Designed from the beginning to achieve high-performance by hardware-intensive means
- Integrated flow control and multiplexing to achieve huge system bandwidths even in widely distributed systems
- Supports a large variety of existing and future protocols and types (e.g. peripheral command sets, network protocols) with a single infrastructure
- Scalable in both performance and cost/performance over a wide range, expandable almost without limit

## WHERE DID FC COME FROM ?

- ANSI Technical Committee X3T11 began work on FC in 1988, development of actual standards in 1991:
- First FC standard (FC Physical and Signaling Interface, FC-PH) close to publication
- Fifteen other FC-related projects presently underway, in three groups:
  - Basic definitions (FC-PH, FC-EP, FC-IG)
  - Fabric definitions (Generic, Switched, Loop, Services)
  - Protocol mappings (SCSI, IPI-3, Block Mux, 802.2 LLC, AAL5)
  - "Foreign" transports (HIPPI, ATM etc.)

## FC STANDARDS PROJECTS.



## FC FUTURES

- Starting work on FC-EP (Enhanced Physical):
  - 2 & 4 gigabit physicals - s/w laser mm and single mode
  - Twinax cables to 1 gigabaud
  - Fractional bandwidth allocation (virtual connections)
  - Hunt Groups, Striping
  - Class 3 Broadcast & Multicast (unreliable)
  - 256 levels of priority
- Higher speed physicals still in 1995!
- More protocols and types of protocols (do not have to be done by X3T11), generic guide in process

## FC APPLICATIONS

- **Scalability of cost/performance and system size, Class 1 and Class 2 operation plus the number of protocols supported leads to a wide variety of applications using the same infrastructure**
- **Mass storage applications range from an interface to a 3½ inch disk drive to large disk farms**
- **Network applications range from terminal networks to router and gateway interconnects to backbones to supernets**
- **Many more areas - real-time, simulation, avionics, conferencing, imaging, broadcast studio, etc.**

## ADVANTAGES OF FC FOR DATA ACQUISITION

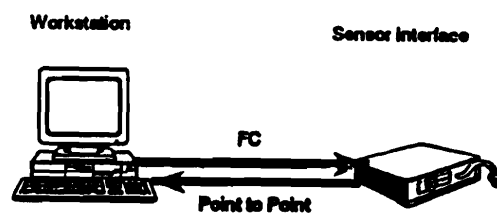
- **Hardware-only first level sensor interface multiplexing:**
  - **Leverage existing components - low cost, high performance**
- **Class 1,2 & 3 allow very small to very large data blocks to be efficiently transferred using the same infrastructure:**
  - **Change the mix of Classes between experiments based on latency and throughput requirements**
  - **Obtain high bandwidth utilization numbers in all situations**
  - **Class 1 & 2 provide lossless, deterministic operation**
  - **Support all types of protocol - network, channel etc.**
  - **Easy to support special and vendor unique protocols**



## ADVANTAGES OF FC FOR DATA ACQUISITION

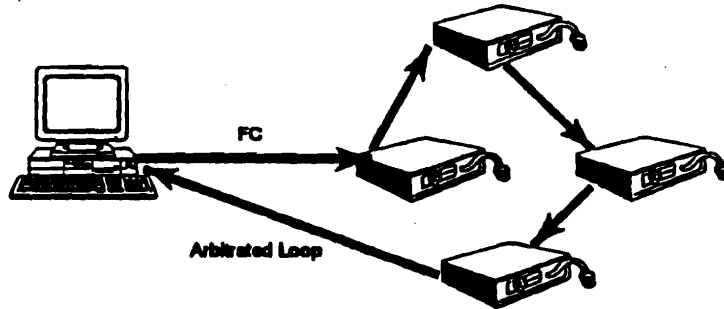
- FC designed from beginning to coexist with other technologies:
  - Mappings to/from ATM & HIPPI exist
  - SCI recently requested, type codes allocated for both SCI & Futurebus
- FC allows systems which are cost-effective, scalable and expandable across a wide range of data rates and system sizes
- Even possible to design a special switch:
  - Meet FC-PH Fabric model
  - Remain compatible with COTS FC-PH equipment

## DATA ACQUISITION EXPANDABILITY (1)



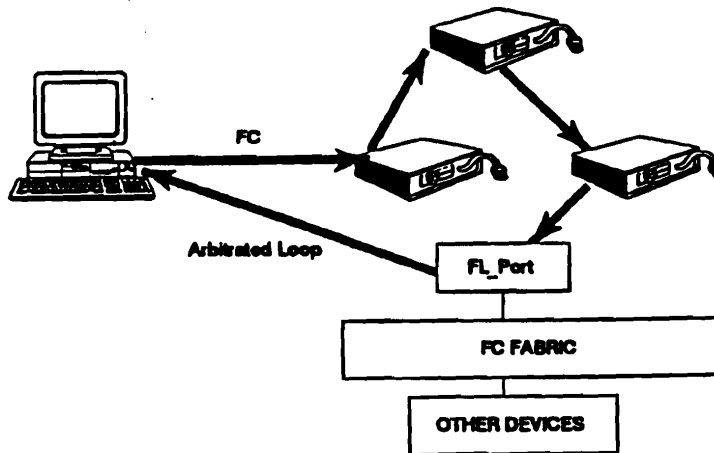
## DATA ACQUISITION EXPANDABILITY (2)

Up to 128 sensor interfaces

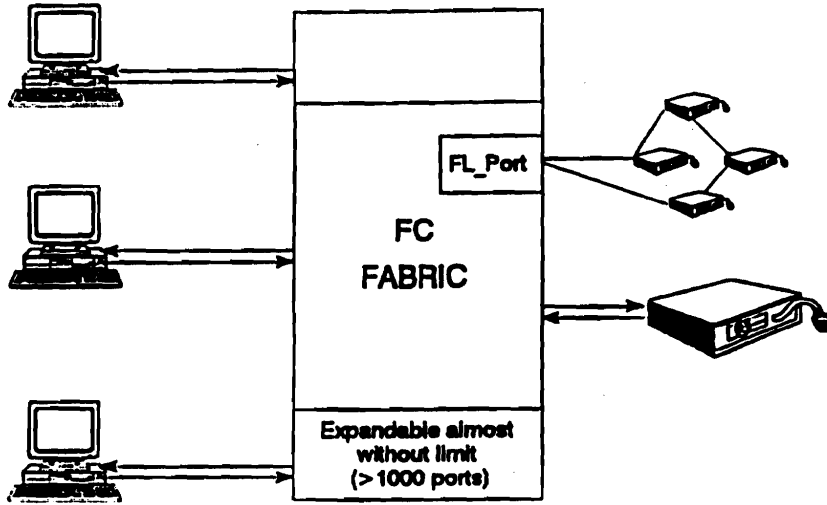


## DATA ACQUISITION EXPANDABILITY (3a)

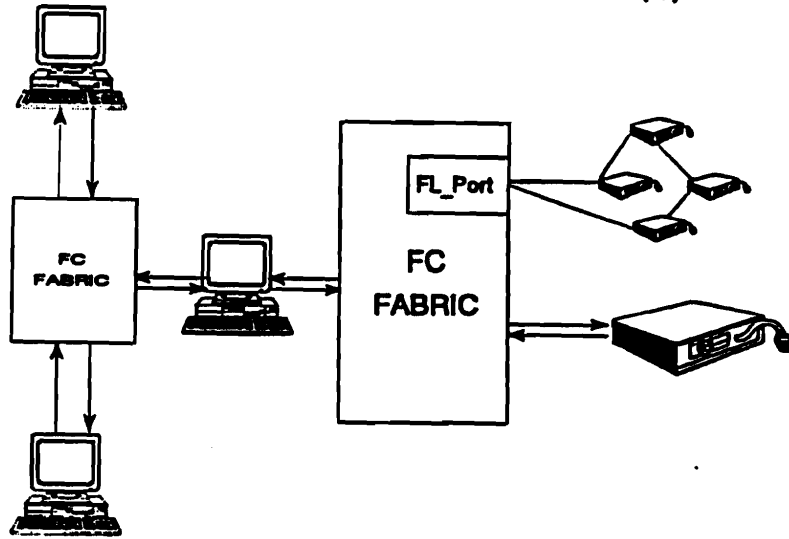
Up to 128 sensor interfaces



### DATA ACQUISITION EXPANDABILITY (3b)



### DATA ACQUISITION EXPANDABILITY (4)



## SUMMARY

- Fibre Channel allows small low-cost systems to be expanded seamlessly to very large systems:
  - Hardware focus lead to good utilization even @ high data rates
  - Excellent interconnection flexibility with Loops and Switches
  - Supports a wide range of traffic types with the three classes
  - Deterministic operation in Class 1 & 2
- Fibre Channel architecture allows special equipment to be used seamlessly with existing COTS products:
  - 15 sources of FC Silicon (and GaAs)
  - FC interface cards for VME, EISA, MCA etc. already exist

## SUMMARY

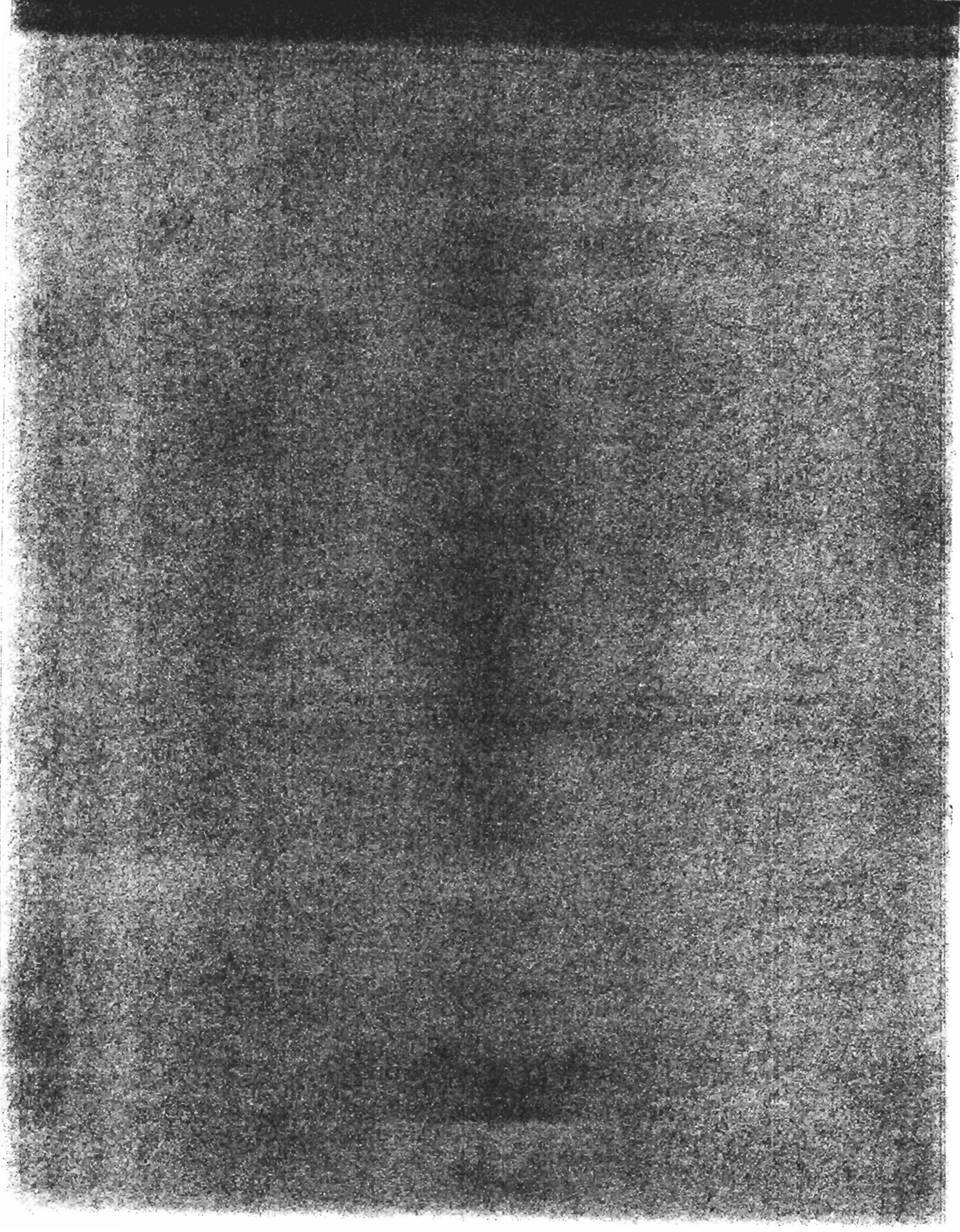
- Fibre Channel has been adopted by a wide spectrum of computer industry:
  - Significant product announcements already made (workstations, disk arrays etc.) with many more to come
  - Commitment to FC as interface of choice by HP, Sun, IBM
  - "The next generation disk array"
  - Supercomputer support as well!
- Fibre Channel was designed from the beginning to interoperate with other technologies such as HIPPI, ATM etc.:
  - Allows the best technology to be applied to the task in hand

S2-4

"SCI"

(Hans Muller - CERN)

Status of standard. Introduction to SCI (DAQ related items especially). Summary of growth of products. One minute at most on cache coherency.

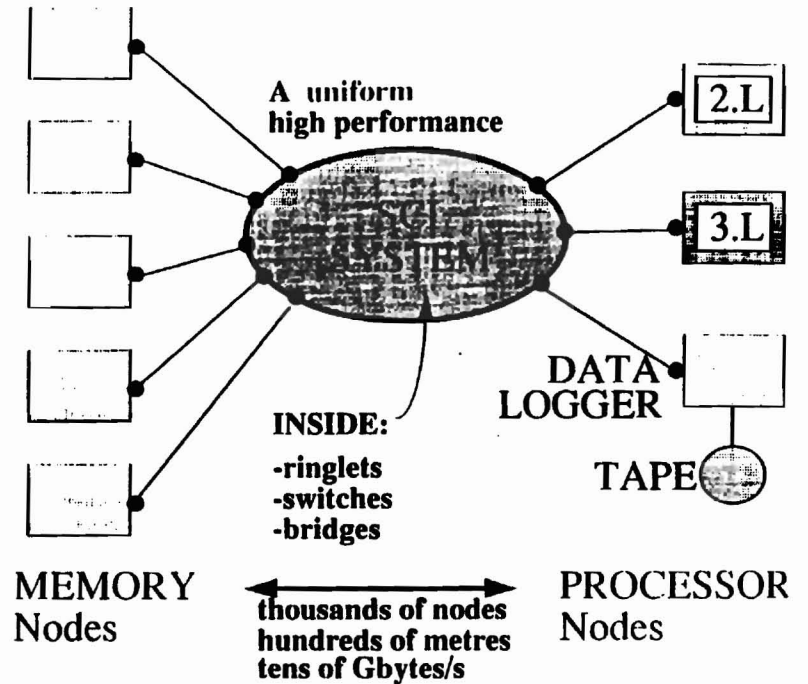


# Scalable Coherent Interface Applications to DAQ

DAQ Conference FNAL October 26-28

Hans Müller  
CERN/ECP-EDA  
RD24

# SCI for HEP experiments



## Pro's: (\* a packet of reasons)

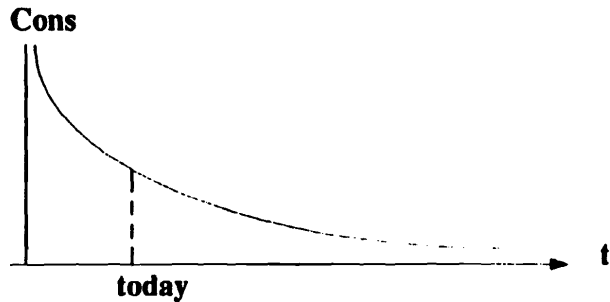
- High bandwidth ( up to 1 Gbyte/s per node)
- Sharing of address space is system wide (64 bit)
- See front-end memory nodes like local memory
- Avoid data copying, use caches
- Fully symmetric communication
- Packets protected via CRC, Retry packets
- Mix inversely directed data streams
- store&forward routing, low latencies
- Copper and Fiber optics
- Price per node falling towards 100\$ US
- guaranteed delivery of packets*

**Con's: (\* a shrinking # of reasons)**

SCI's marketplace is very small today  
 ...but SCI solves a general problem of  
**MPP: Memory sharing and low latency**  
 Most of SCI is in R&D Labs.. cooking takes time  
 Physical layers need work....ISO-IEC, WG 15  
 Fiber links for 1 Gbyte/s needs initiatives.. there are  
 Support chips ( Cache, Memory Controller)  
 ... still missing (?)

SCI bridges: currently only SBUS .. wait for PCI

SCI switches: still haven't seen one ( ..N month ?)



**Special requirements of HEP Experiments:**

Crate/node power loss: keep ringlets alive  
 Cabling requirements : "harsh environment"  
 N->1 event data synchronisation

**Status of the standard**

**S**calable  
**C**ache coherent  
**I**nterface

**ANSI/IEEE approved  
 Standard 1596-1992**

+ newly approved ISO/IEC 13961  
 WG15: SCI in harsh environments

- Substandards:  
 P1596.1 Bridge Architectures  
 P1596.2 KiloProcessor Extensions  
**P1596.3 Low Voltage Differential**  
 P1596.4 RAMlink  
 P1596.5 SCI Data Formats  
 P1596.6 SCI/Real Time

Related Standards + Technologies  
 IEEE 1212 CSR Architecture  
 P1394 Serialbus  
 Quickring (LVDS) by National Semiconductors

Not only a performance issue, also long lifetime + technology independence!

DAQ will need cached data access, whether coherency is needed is not clear. However, if needed the bit fields are there in each SCI packet.

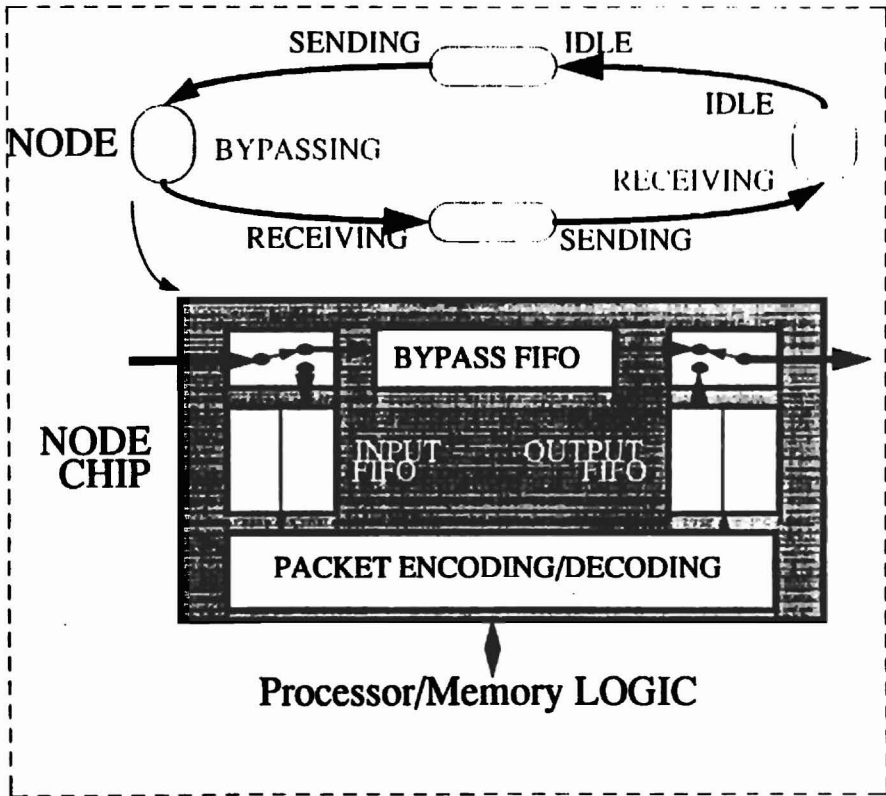
Processors done: R3000, 68040\*, Sparc\*, PA-7100\*  
 DSP: C40 in VME  
 coming: pPC60x\*  
 Memories done: Dual Port Ram\*  
 bidirectional FiFo  
 Sbus RAM\*  
 DRAM\*  
 coming: cache-coherent  
 Video RAM  
 16 Mbyte VME-Ram  
 CMS FDPM  
 Buses /Bdidges done: VME\*, SBUS\*, TurboChannel  
 coming: PCI bus\* (!)  
 ATM\*, Fastbus\*

\* are commercial products, all other research item



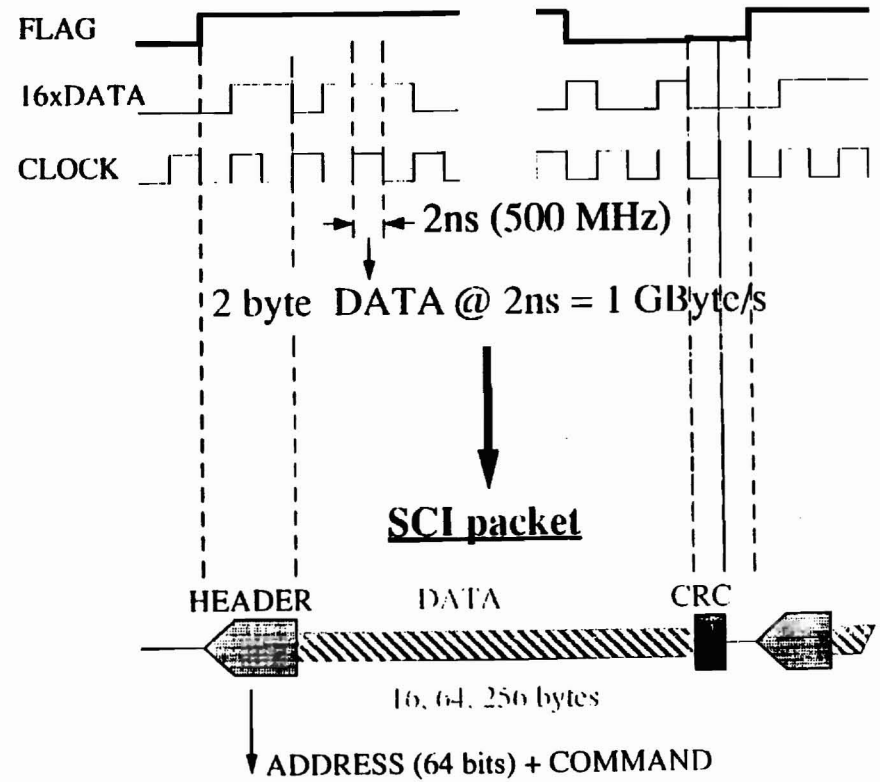
SCI gives us:  
*computer-bus-like services  
 over distance and between many nodes  
 A standard with built-in scalability.*

**An SCI Ringlet example:**



Ringlets are implemented via SCI cables or Fiber Optics  
 Ringlets on average can take bandwidth of 2 Nodes ( up 2 Gbyte/s)  
 Receiving nodes may be intermediate nodes to further ringlets  
 Send buffers are kept till echo from next node is returned

**SCI packet encoding on 16 bit wide links**

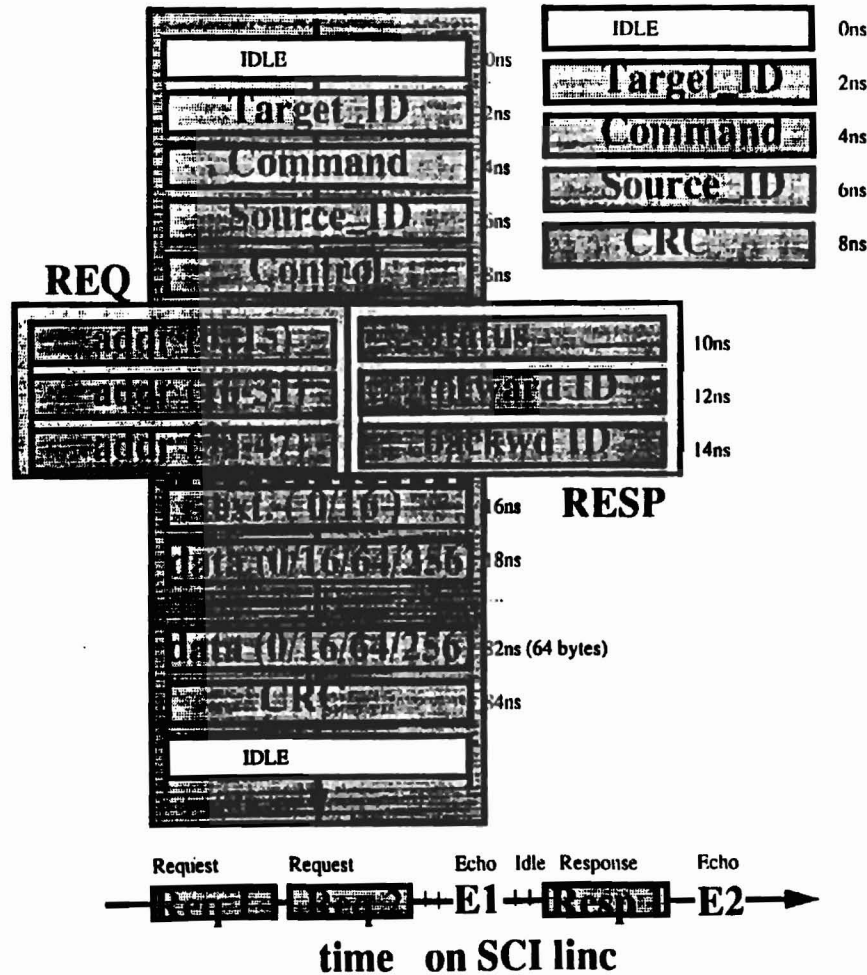


**18-DE-500 physical standard, ECL -> 1Gbyte/s**

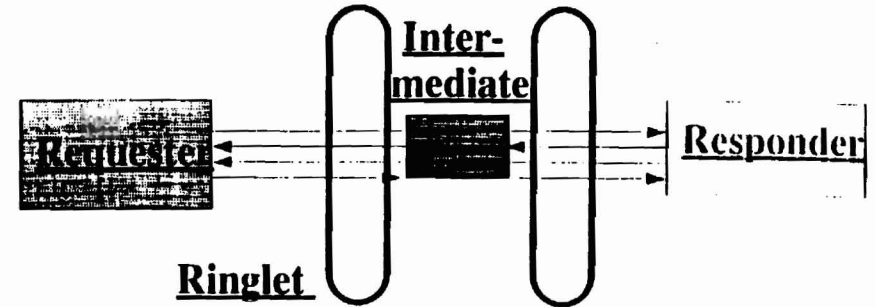
- GaAS nodechips Dolphin, ECL -> 500 Mbyte/s
- GaAS nodechips Fujitsu, ECL -> 700 Mbyte/s
- Vitesse Datapump ,2\*2 switch LVDS -> 1Gbyte/s
- IBM BiCMOS Lincchip, LVDS\* -> 1Gbyte/s
- LSI Logic CMOS nodechip -> 125 Mbyte/s
- Linc Controller .... -> 200 Mbyte/s

**SCI packet:  
a sequence of 16 bit SYMBOLS**

**Request /Response packet      Echo packet**

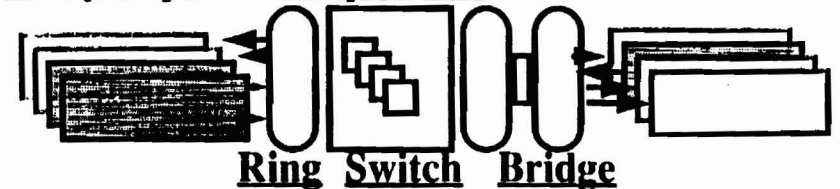


**SCI split Transaction mechanism**



- Request -> Response: between any requester and responder (latency is mostly responder property, min 1 us)
- Echo Subactions: between pairs of intermediate SCI nodes ( depends on link distance and speed ).  
Echos for buffer propagation  
Echos carry retry information.

**many requestor/responders:**



**Split transactions allow for bus-like connection, however simultaneous ( no WAIT signal )**

**Measured/reported SCI latencies**

CERN GaAS ( 500 Mbyte/s chip )	latency (ns)
DMA, dmove64 write	560
RIO R3000 MIPS firmware IF for "dataless" dmove64	1 700
RIO R3000 MIPS firmware IF for dmove64 "real data" write+read	21 300

Apple ATG GaAS ( 500 Mbyte/s chips )	latency (ns)
Inter Quadra 68040 memory-memory	5000-7000

Convex GaAS ( 700 Mbyte/s chips )	latency (ns)
memory access (best local - worst remote)	500- 18000

CERN CMOS ( 125 Mbyte/s chip )	latency (ns)
DMA, dmove64 write on CES' SC18224	2800
Sbus memory-memory CERN	4000

RAL/Manchester( 125 Mbyte/s chips )	latency (ns)
DBV44-SCI interface, between user buses	2500

**Conclusions:**

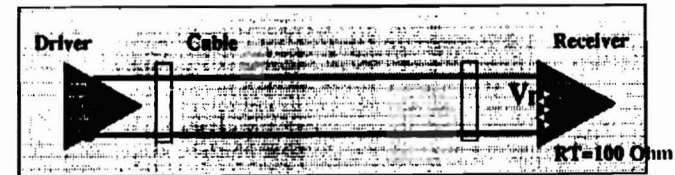
- \* GaAS 1/2 Gbyte/s speed Chips approach already 1  $\mu$ s latencies
- \* Extrapolation from 1/8 Gbyte/s CMOS chip to coming LVDS ( 1/5 .. 1 Gbyte/s) chips: 1  $\mu$ s = normal  
 → SCI latencies are really bus-like

**LVDS Low Voltage Signalling**  
**P1596.3**

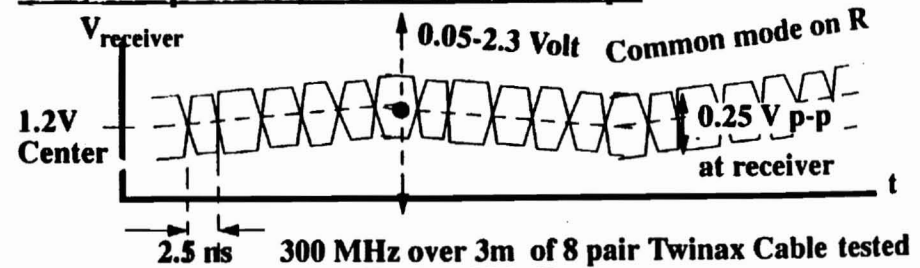
New line driver/receiver standard for high speed with much interest in Telecommunication Industry.

Compromise: low Voltage and high noise immunity

- \* lower voltage swing than PECL/ECL: differential signals centered around 1.2 Volt (GTL)
- \* constant current ( 3.2 mA) with  $Z_{out}$  matched to line
- \* Receiver Common Mode range between 0 and 2 Volt
- \* termination  $R_T$  on receiver device around 100 OHM
- \* Power dissipation in termination much reduced as compared to ECL-> no external termination resistors



**Test Chip 1.2 u CMOS at 400 Mbps**



**LVS Chips:**

- \* IBM's 500MHz BiCMOS: LVDS at 1.0 Volt
- \* Vitesse's 500MHz DATAPUMP uses LVDS
- \* Dolphin's next LC uses LVDS
- \* National: New TTL-LVDS Quad Rx/Tx chip
- \* RAMLINK uses LVDS

CABLES + CONNECTORS + signals

Metrol :

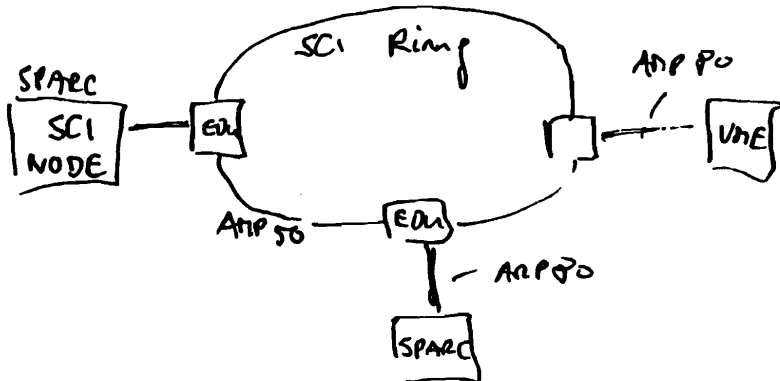
500MB/s 3M Gore tex twisted  
( expensive + fragile )

125MB/s 3meter AT&T  
( fragile , unreliable )

→ Metrol experience bad

AMPLITUDE 50 + 80

New Dolphin kind. cabling



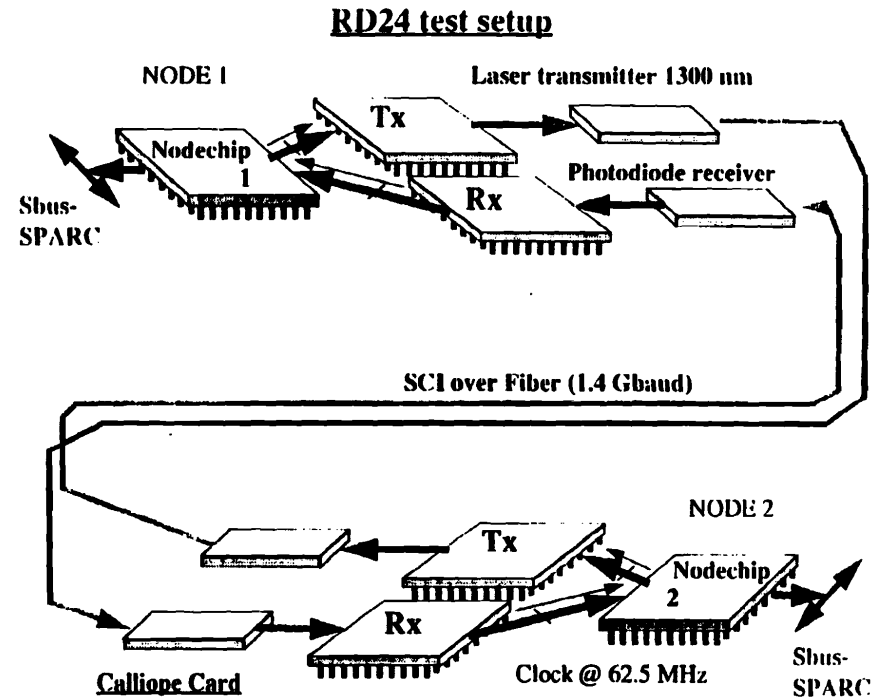
To be tested ....

ISO-IEC WG15 : 'Physical layers of SCI applications in harsh environment'

→ Haus@sunshine.cern.ch

SCI over Optical fiber transmission interconnect

Use of HP full duplex Glink Chips HDMP-1002/1004

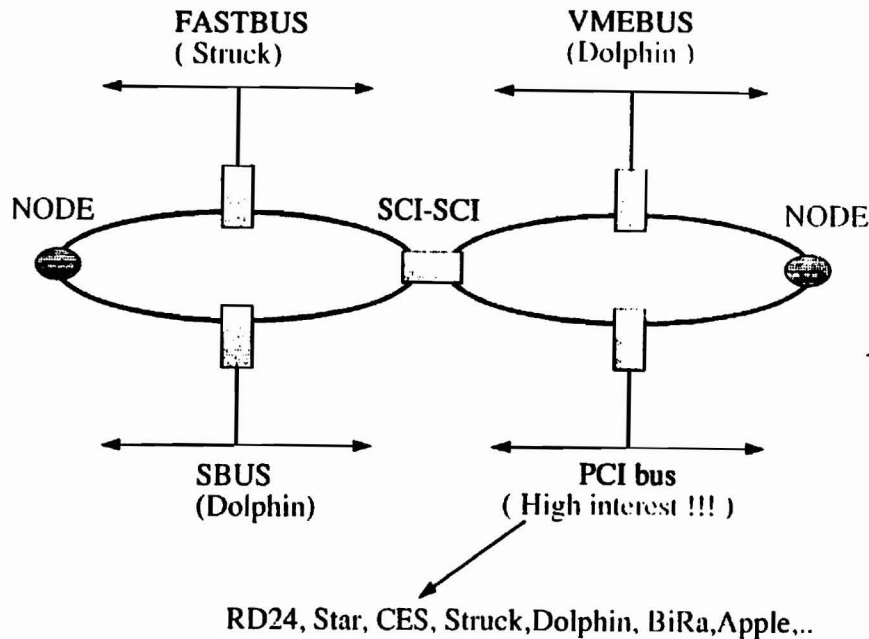


SCI-FI mode in HP Glink parts:

- 17 bit parallel ( 16 data 1 Flag ) -> serial 20 bit frames ( CIMT coding )
- PLL locks on user supplied frame rate clock ( also low speed applications )
- 65 MHz 16bit parallel in -> 1040 Mbit/s serial bit rate ( 1300 Mbaud )
- Extended operation rates, chips typically work up at 2 Gbit/s, future 3 Gbit
- Cost 117 USD for 1 @ 10

Optical modules: BT&D and Finishar use Glink with Laser optics

## SCI BRIDGES and Adapters



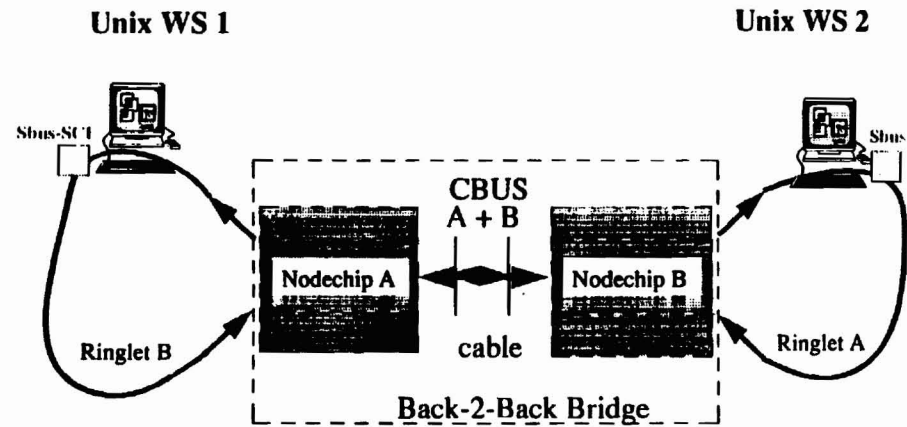
### Exists:

**SBUS-SCI: SCI development kit (Dolphin) includes Unix drivers. RD24 shared memory tests on Sparcstations**

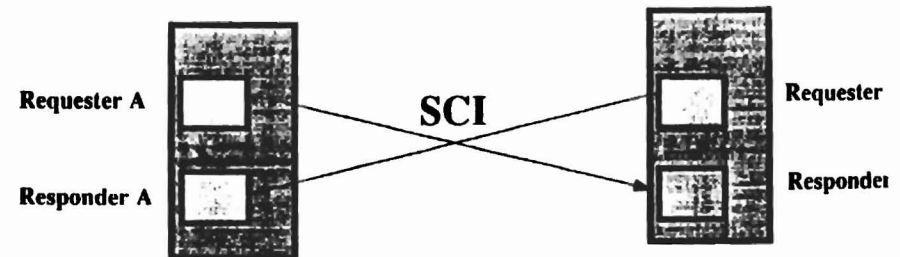
**VMEbus: Expected early 95 uses Cypress VIC chip and LSI Logic nodechip**

## First Two ringlet bridge with transparent memory access:

### Use of CMOS Nodechip bridge feature

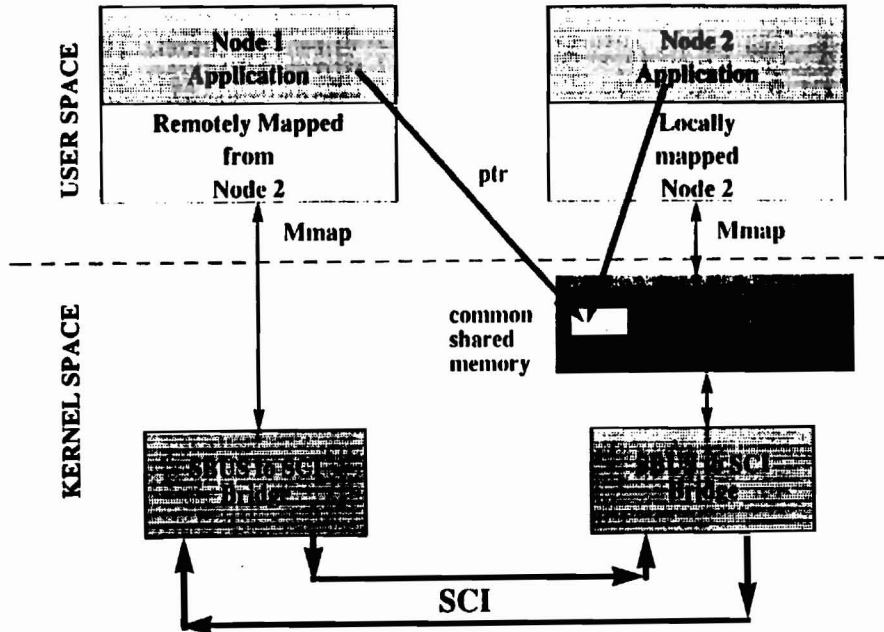


### tests using Sparcstation/SBUS-SCI cards



All transactions pass the bridge error-free  
transfer latency is 1.8 microseconds

**Tested: SHARED MEMORY via SBUS**



Example is applicable to N nodes and any distance

**TRANSPARENT ACCESS**

**NODE 1**

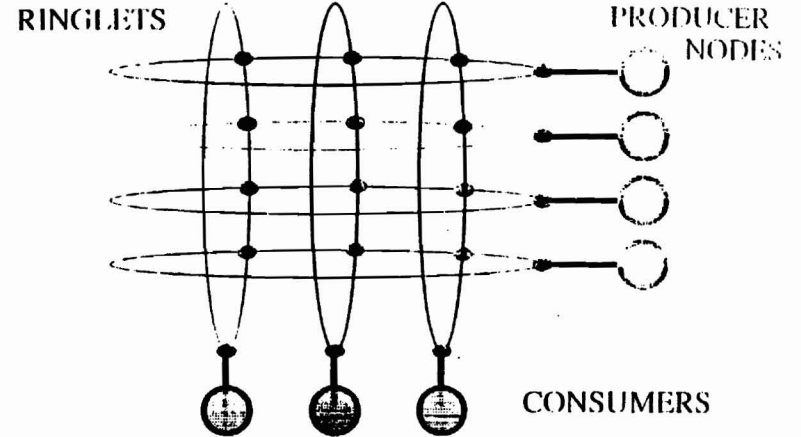
```
fd = open("/dev/sci1", O_REMOTE_MAP)
ioctl(fd, CONNECT, NodeId)
ptr = mmap(fd, 1024);
*ptr = 123;
close(fd);
```

**NODE 2**

```
fd = open("/dev/sci1", O_LOCAL_MAP)
ioctl(fd, CONNECT, NodeId)
ptr = mmap(fd, 1024);
printf("data is: %d", *ptr);
close(fd);
```

**DAQ - ARCHITECTURES USING SCI**

**A.**



Note however that:

Ringlets do not scale  
The number of bridge-nodes  
is large ( N \* M)

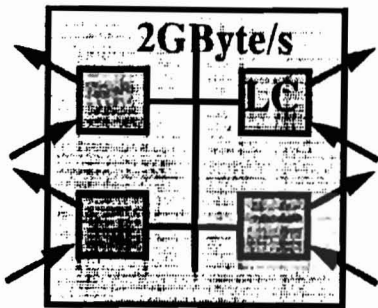
**1000.000 ! for a 1000\*1000 EB !**

For large architectures, use SCI switches:

Multistage switches scale  
The number of switches  
is smaller:  $N * \log_2(M)/2$

## B.) Large Switch ARCHITECTURES

### 4 way SWITCH:

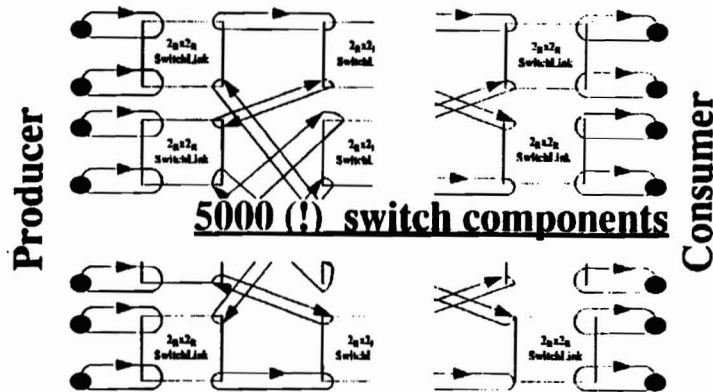


Four 1Gbyte/s Line chips with 2 Gbyte/s internal bus:  
4-way switch with 1.4 Gbyte/overall throughput.

1 Gbyte/s SCI

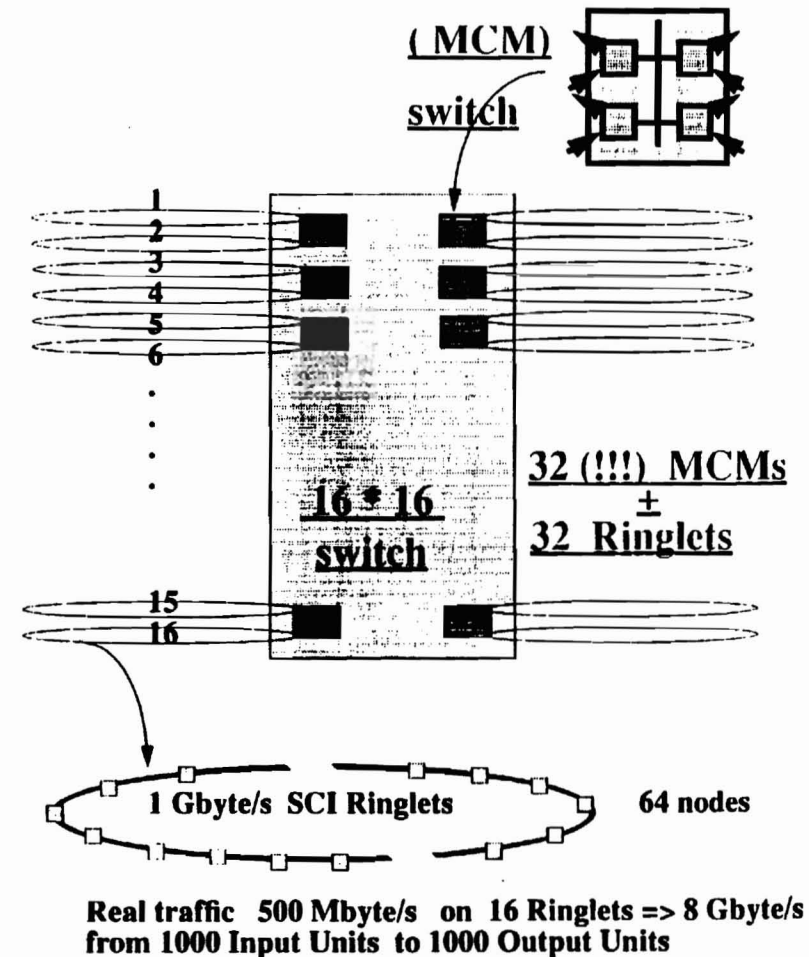
Eureka TOPSCI project:  
GaAS 4 way switch MCM

### 1000\*1000 Superswitch :



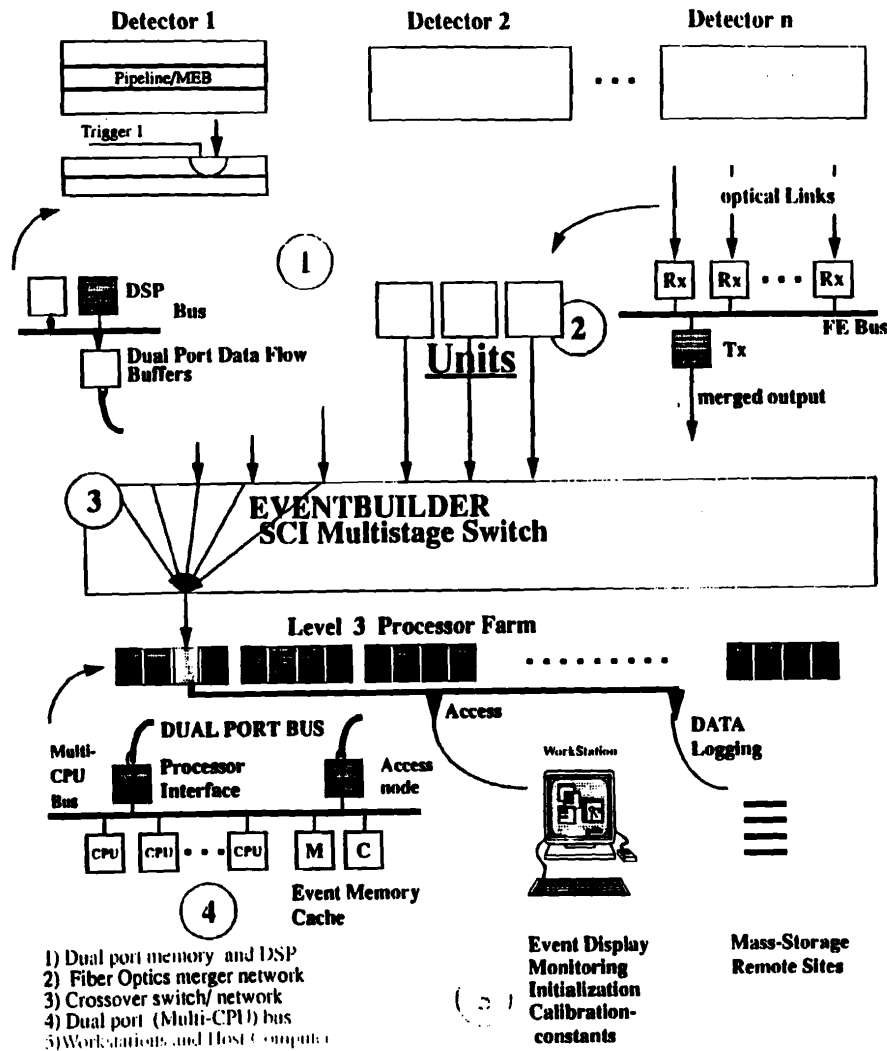
SCI retry mechanism for traffic derandomisation  
Intermediate buffers inside each LC port: event pipelining  
Supports both data driven and cached readout: large bandwidth savings  
No inherent packet losses.  
Scalability up to tens of Gbytes/s has been simulated

## C.) Realistic large Eventbuilder



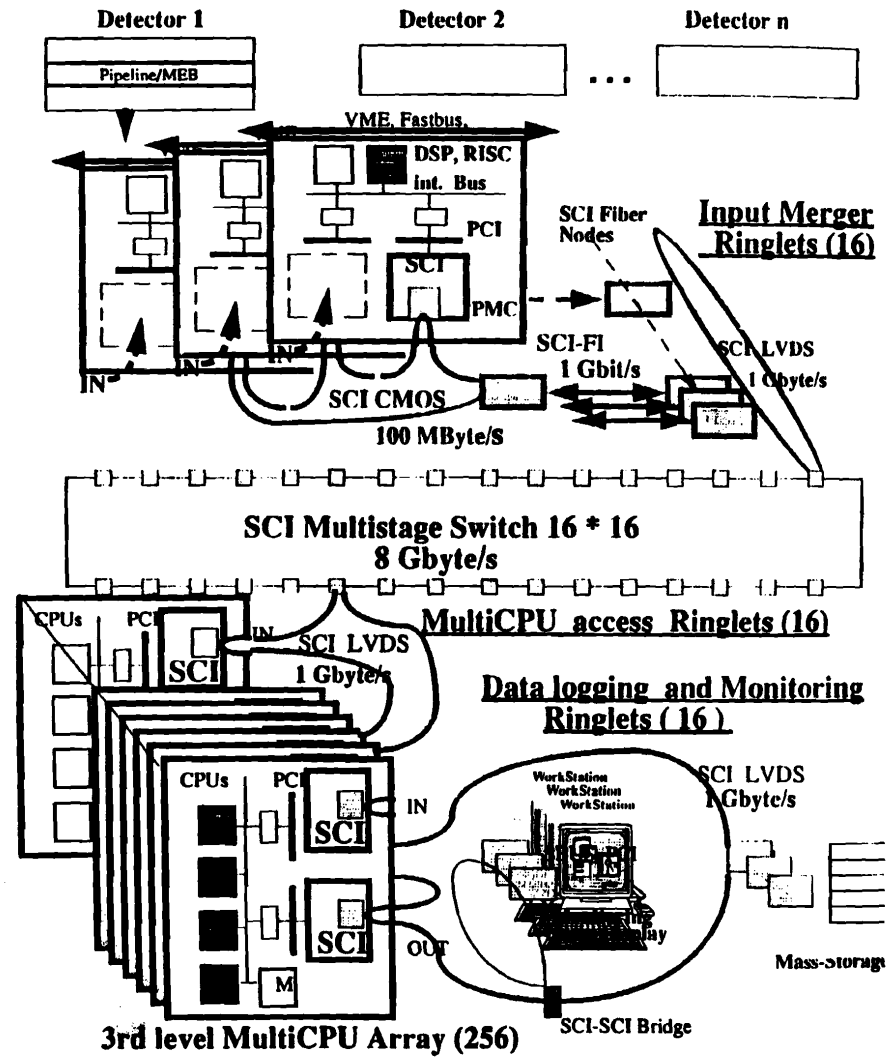
Other options: 16 bridges to low cost/speed CMOS nodes

# SCI for a uniform Data Acquisition system



- 1) Dual port memory and DSP
- 2) Fiber Optics merger network
- 3) Crossover switch/network
- 4) Dual port (Multi-CPU) bus
- 5) Workstations and Host Computer

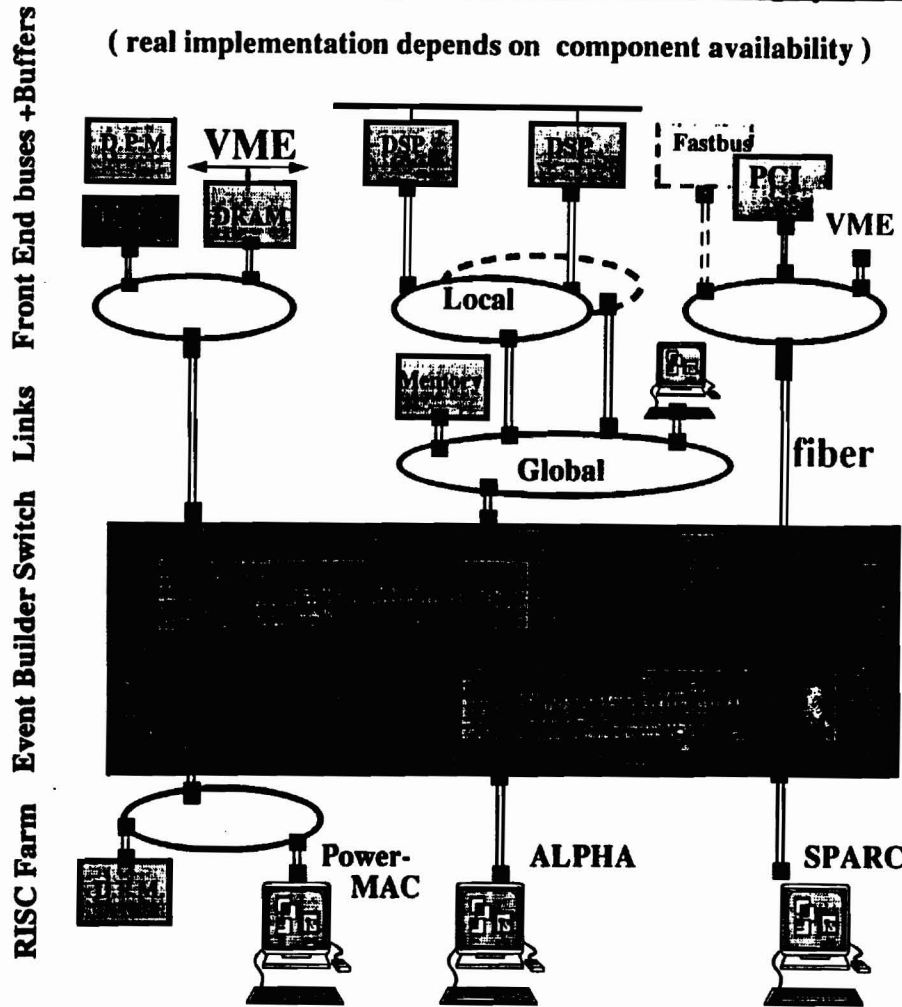
# LHC Data Acquisition System in SCI





# R&D for a heterogeneous mini SCI DAQ system

( real implementation depends on component availability )

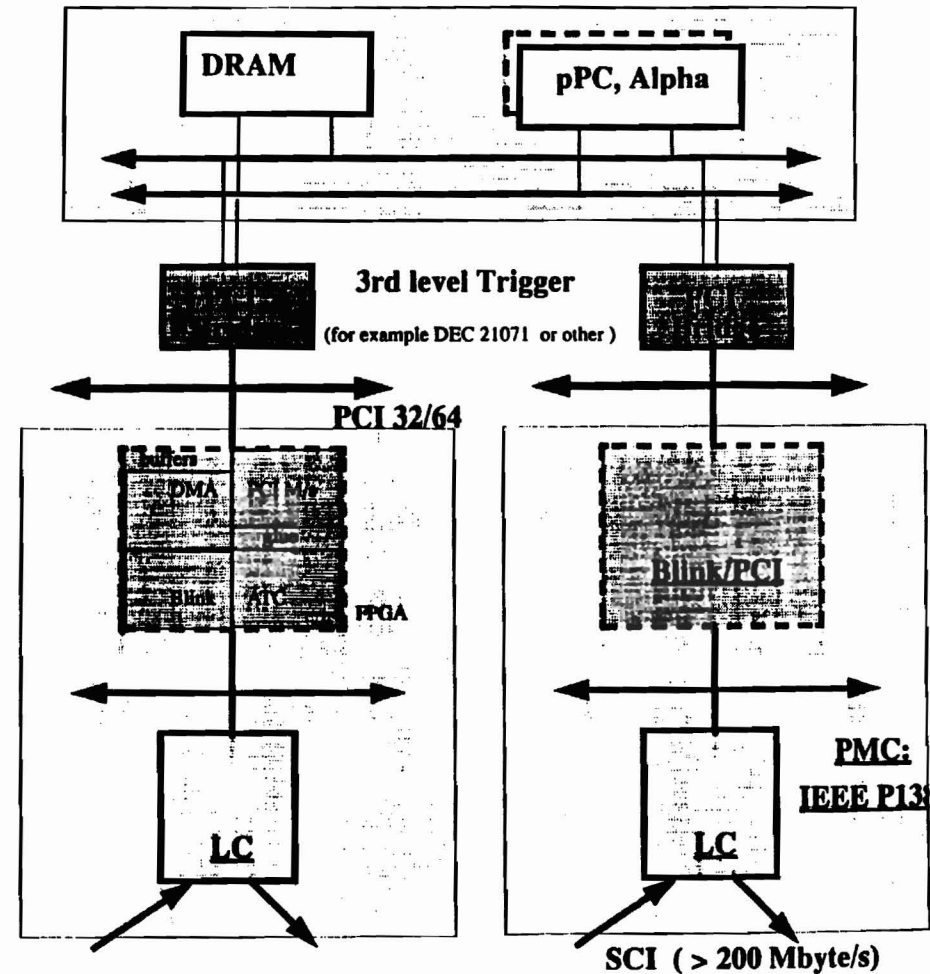


**CMS like:**  
data driven event memories +  
high speed event builder to  
massive RISC farm  
test transparent CPU access to  
data over EB switch

**ATLAS like:**  
decomposed local and  
global data network  
with  
intermediate processors  
before EB

**ALICE like**  
data  
concentrator  
of mixed FI  
buses,  
passive EB

# PCI and PMC : The new local bus standard and possibilities for Dual Port Architectures



Data Mainstream in

Data Mainstream out

## SCI Bridge to a CPU

### Transparency:

Map CPU's bus transactions to SCI transactions:

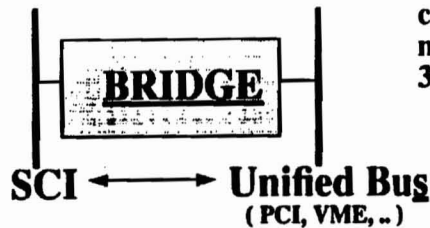
### Performance:

Optionally add DMA (ETA\*)

→ user doesn't need to know

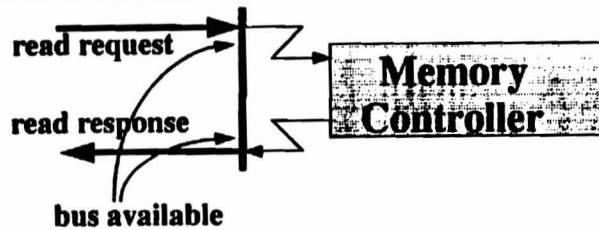
→ programmer needs to know

### Resolve bus incompatibilities

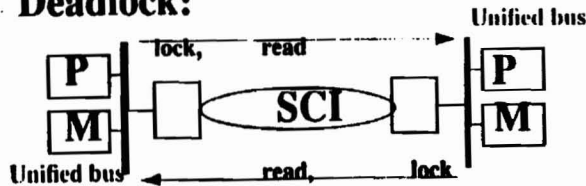


Split/Uniform, cachebursts, mpu primitives, 32/64 bit address

### SCI: Split Transactions



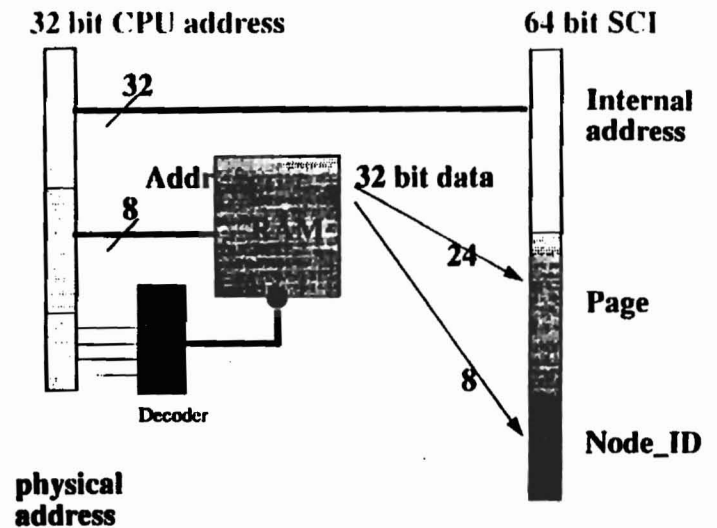
### Deadlock:



Solution: Use Release-Retry

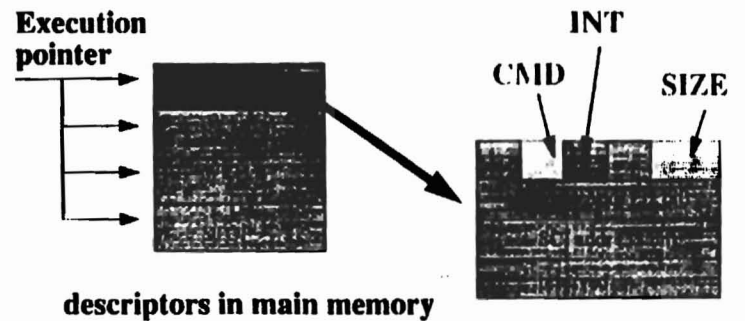
\* Extended Transaction Unit

## 32->64 bit address translation



physical address

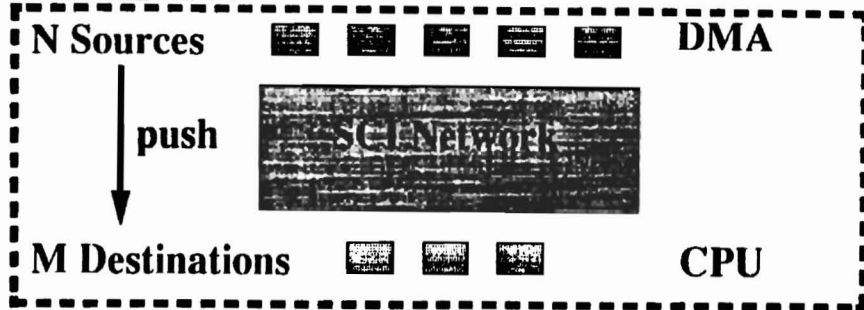
### DMA Extended Transaction Unit



CMD = Read, Write, STOP  
 SIZE=1-65536, 0=NOP  
 INT= Interrupt on normal completion

### Choices ARCHITECTURES

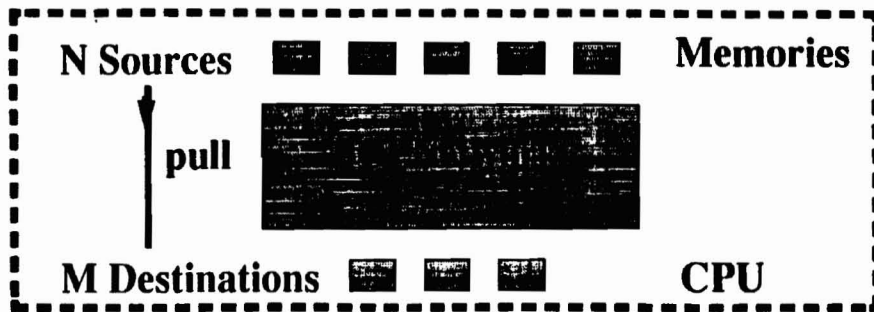
#### Data driven:



- how do sources know where to send
- how do destinations know when last arrived

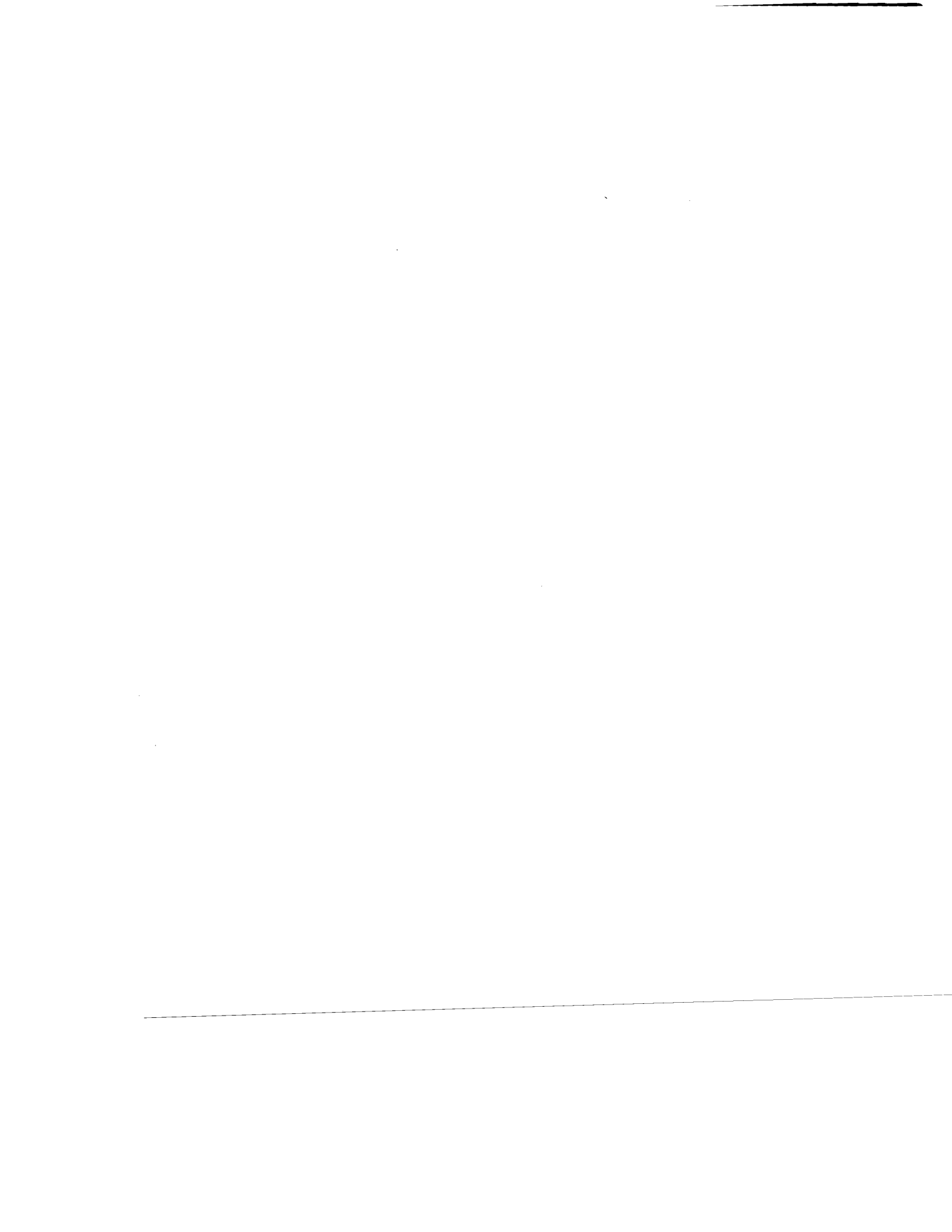
**SCI possibilities:** *SCI common clock, shared counter can use fetch&add  
SCI interrupts, SCI broadcasts, coherent protocols..*

#### CPU readout:



- SCI caching to reduce for access latenc

**Which is better: SCI can do both with same HW**

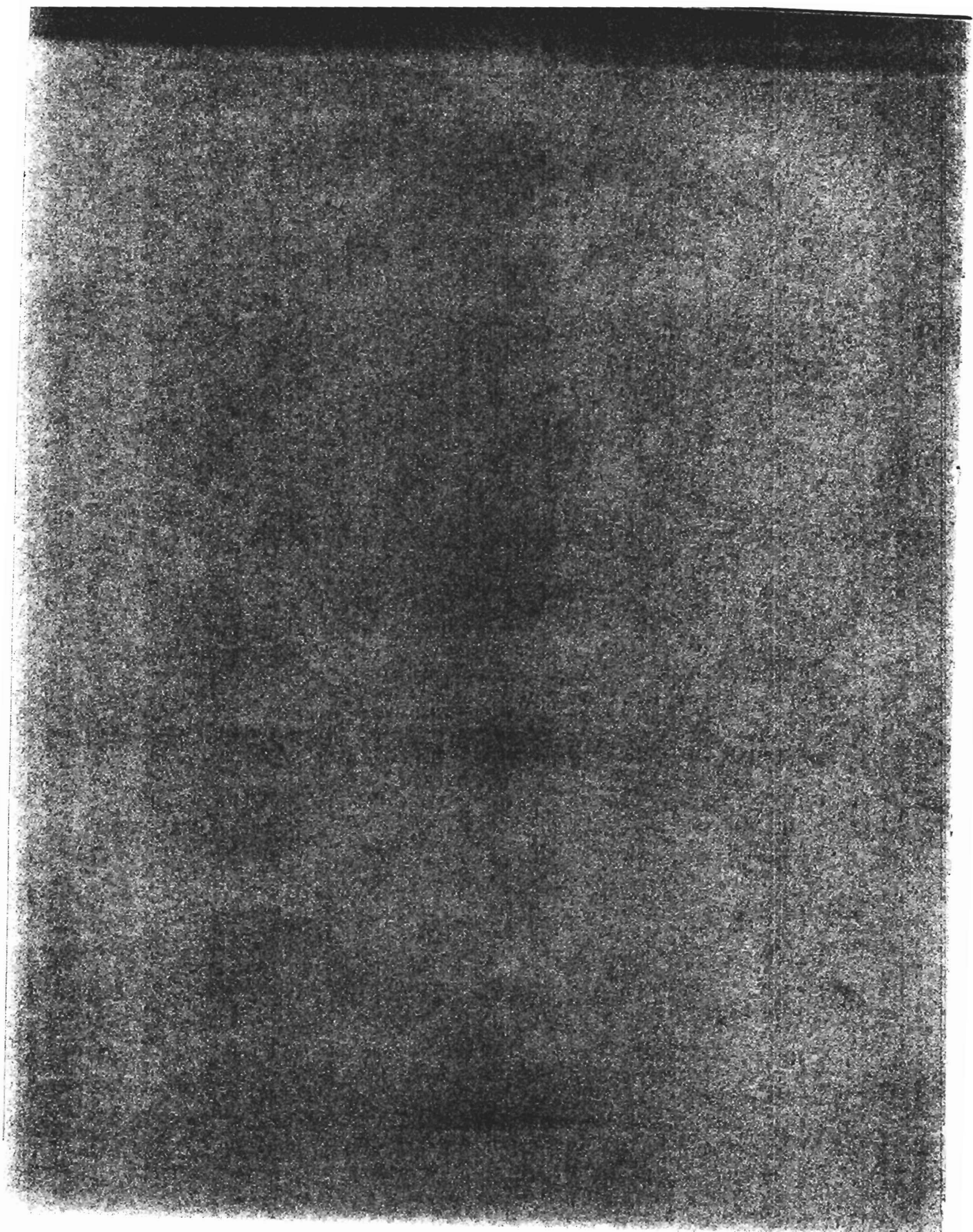


## S3-1

### "ATM Research Projects"

(Jean-Pierre Dufey - CERN)

The presentation will review the European research projects which contribute to evaluate the applicability of ATM for high energy physics data acquisition systems, in particular for the future high data rate experiments for LHC. The various architectures and the different switching network technologies that are being considered when simulating event builders will be reviewed. The ongoing and planned efforts to develop ATM adaptation boards, data generators and event-builder demonstrators will be described.



## Overview of some research projects in Europe

### ATM Research Projects

Jean-Pierre Dufey

(dufey@sunvlsi.cern.ch)

CERN

*LAY-OUT:*

*Overview of some research projects in Europe*

*The CERN RD31 project:*

*Modelling*

*Event-builder demonstrators*

*ATM Adapter*

*Software development --> I. Mandjavidze*

#### *Industry projects:*

ALCATEL: MPSR switch --> I. Mahood

AT&T & EPFL Lausanne:  
Phoenix switch --> A. Wiesel

IBM: Prizma switch --> T. Engbersen

#### *Physics research labs:*

CERN: RD31 --> this presentation

CERN: CN division --> " "

UPPSALA: WASA expt. --> " "

ESRF, Grenoble, France --> " "

INFN, Frascati, Italy

## The Parallel data acquisition system for WASA

**CERN: CN division:**

(B. Carpenter, D. Davids (danny@dxcoms.cern.ch), J. Joosten)

Application of ATM to the CERN computer network.

Provides useful information and help for:

- News about standards (ATM Forum)
- Contacts with industry.
- Evaluation of products.
- Some benchmark measurements (Netcom, IBM, HP)

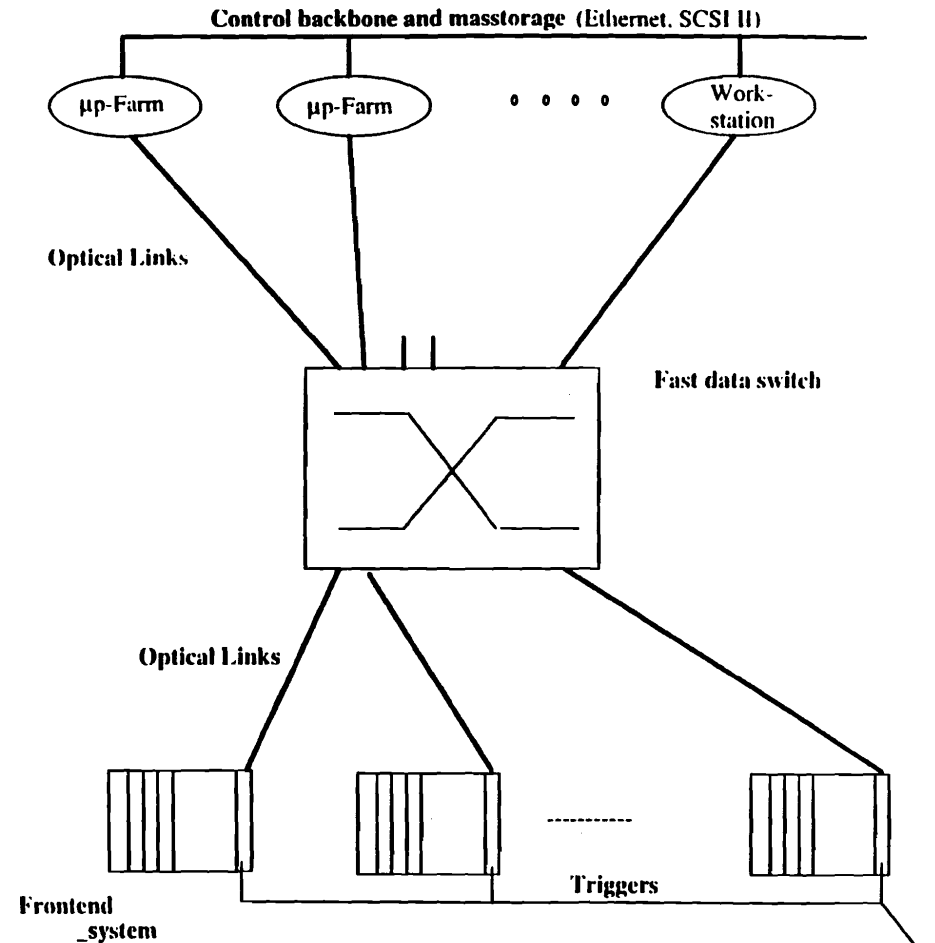
**UPPSALA: WASA expt.:**

(L. Gustafsson, lrg@tsl.uu.se)

~~Point-to-point~~ Connection using ATM over SONET OC-3 (multimode fibers) and multicast of the data using an 8 X 8 switch.

Collaboration with RD31 for the development of a VME-ATM interface.

Plans to port the design to PCI .





**ESRF, Grenoble, France: (Synchrotron radiation accelerator)**

B. Lebayle (lebayle@esrf.fr) et al.

Use of ATM for fast data transfer between the experiments ('beam lines') and a computer center with:

- fast data storage devices
- high-end graphics servers
- computing servers
- DAT tapes.

Detectors: VME/VXI based.

ATM interfaces: from FORE, Driver under LYNXOS

ATM switches: 2 @ 12 ports each (-> 16)

Status:

- in operation since Dec 1993
- 4 beam lines connected (up to 60 in the future)
- 100 Mbit/s to be upgraded to 155 Mbit/s, OC 3

**INFN, Frascati, Italy**

(P. Mateuzzi, D. Salomini et al.)

Event builder based on Gigaswitch + FDDI --> ATM (Digital).

## RD-31

# NEBULAS: A high performance data-driven event building architecture based on an asynchronous self-routing packet-switching network

M. Costa, J-P. Dufey, M. Letheren, I. Mandjavidze, A. Marchioro, C. Paillard  
*CERN, Geneva*

K. Agehed, S. Hultberg, T. Lazrak, Th. Lindblad, C. Lindsey, H. Teuhunen  
*The Royal Institute of Technology, Stockholm*

L. Gustafsson  
*Institute of Radiation Sciences, University of Uppsala, Uppsala*

D. Calvet, K. Djidi, P. Leduc  
*CEN DPhPE SACLAV*

M. De Prycker, B. Pauwels, G. Petit, H. Verhille  
*Alcatel Bell Telephone, Antwerp*

M. Benard  
*Hewlett Packard*

### Collaborating Institutes:

P. Sphicas, S. Tether  
*MIT*

A. Manabe, M. Nomachi  
*National Laboratory for High Energy Physics (KEK), Japan*

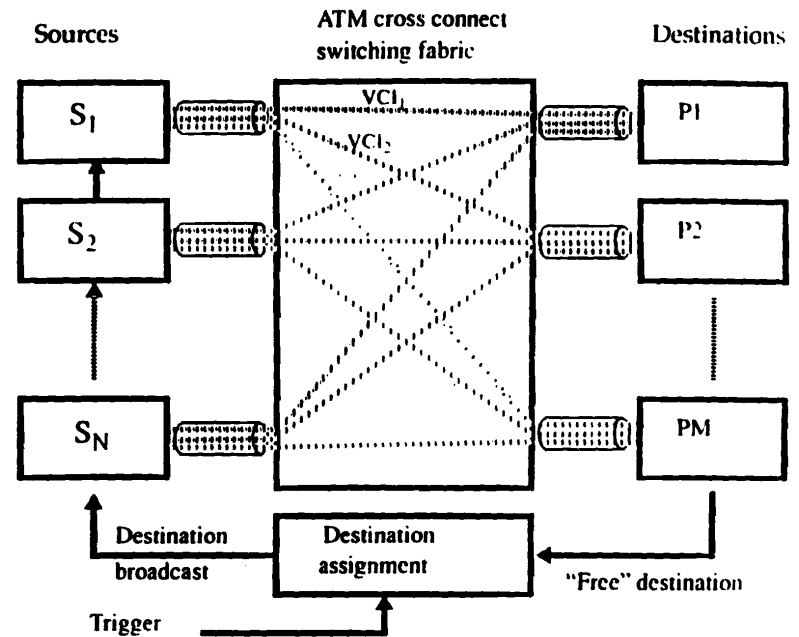
E. Barsotti, W. Knopf, D. Walsh  
*Fermilab, Batavia, USA*

M. Haney, T. Brandys  
*University of Illinois at Urbana-Champaign, USA*

## RD-31 Research Directions

- Modelling of switching fabrics and event builder architectures. (ATM, Fiber Channel, custom made 'conical').
- Development of adapters and data generators (ATM).
- Development of small demonstrator event builders (ATM).
- Development of drivers and DAQ protocols software .

## Generic ATM-based Event Builder



- $N \times M$  *semi-permanent virtual connections*.
- For each event, the destination assignment logic broadcasts the identity of the destination.
- Sources segment their event data into cells with appropriate VCI labels.
- Cells self-route through the switch to the destinations, which re-assemble the event data from the incoming cells.

## Why do we need modelling ?

To study the behaviour of switching network architectures

and to understand system design tradeoffs:

- Required size and speed of switch fabric,
- Event-building latencies
- Buffer occupancies -->Dimensioning of buffers
- Cell loss probabilities
- etc ...

...under various conditions of traffic, depending on:

- L1, L2 trigger rates and data movement strategies,
- Event size distribution,
- Distribution of data amongst sources.

## An event builder model is more than a switch model.

Steps to develop an event builder model based on a particular switch technology:

- Create a detailed model of the switch (if you manage to convince the manufacturer to give you enough details).
- Validate the model by comparison with the manufacturer's data. (same remark as above).
- Complement the model with the event-builder components:
  - a configurable traffic generator,
  - distribution of data amongst sources,
  - a control of the event building in the destinations,
  - a destination assignment scheme,
  - buffer management and monitoring in sources and destinations,
  - processing in destinations,
  - Traffic shaping.
- Possibly develop several models by different persons and with different languages
- Validate the results by a "theoretical" approach if possible (not easy !!!)

## Traffic shaping or internal flow control ? or a combination of both ???

This is the subject of hot debates  
within RD31 these days !

### Traffic shaping:

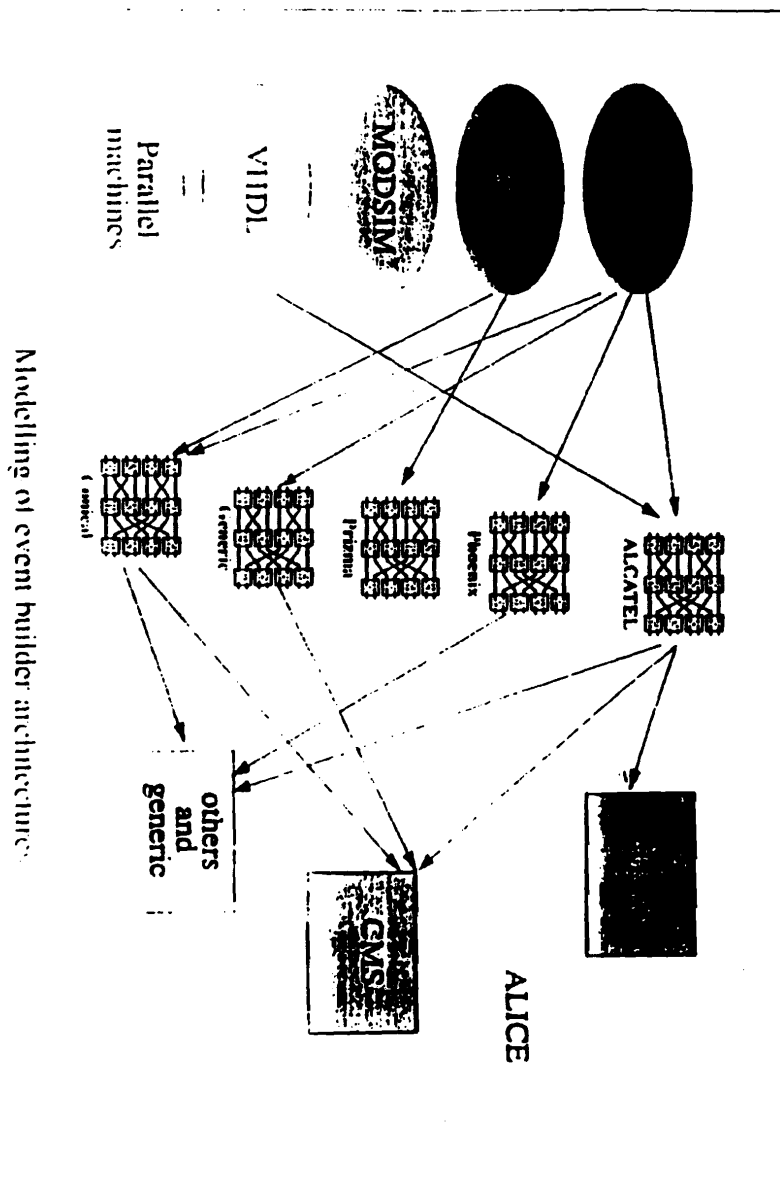
Principle of traffic shaping for event-builder applications:

- The traffic on a VC must not exceed, on average,  $1/S$  of the nominal bandwidth ( $S = \text{Number of sources}$ )
- The traffic from all sources towards a given destination must be skewed in time.

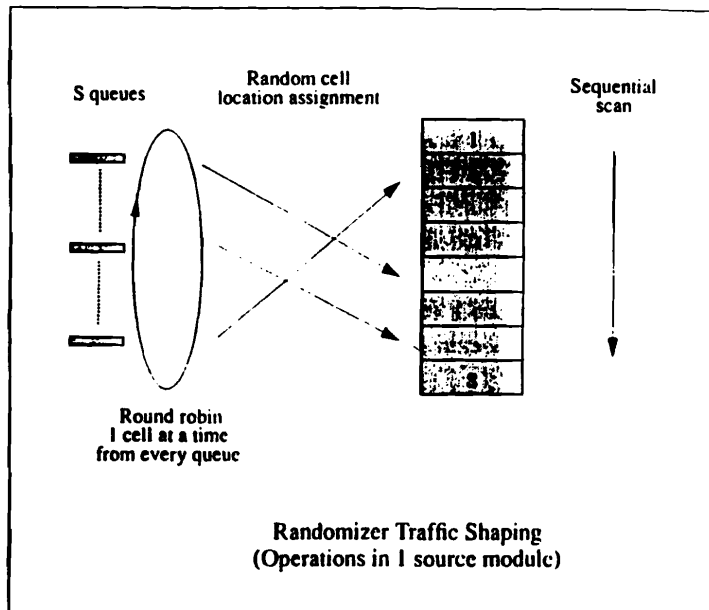
Characteristics:

- It gives good scalability characteristics
- It is necessary for switches without internal flow control.
- When applied to square switches with internal flow control, it helps to reach higher loads.

Two traffic shaping methods are proposed:

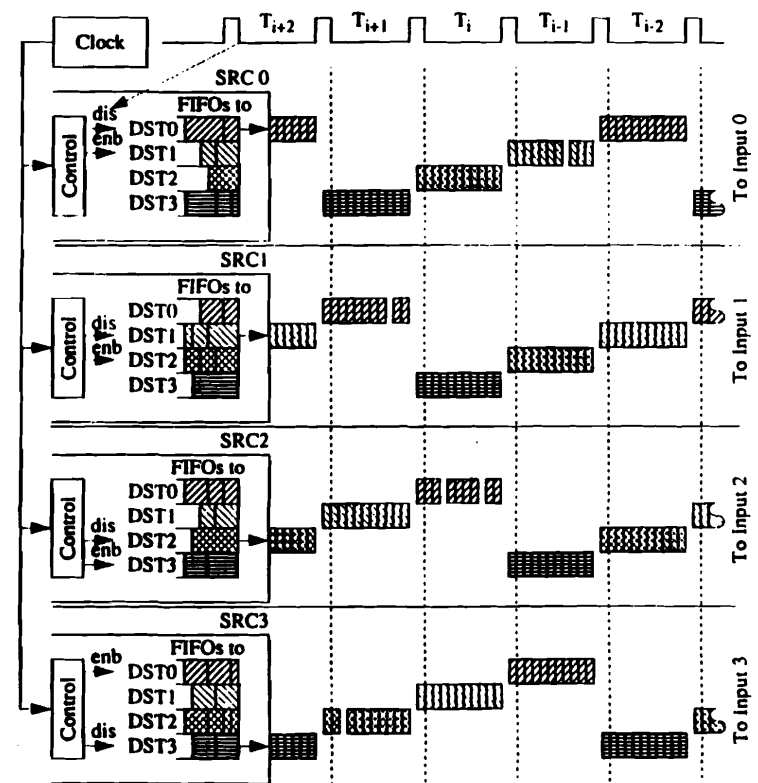


## Randomizer traffic shaping:



- random injection of cells on the network.
- requires special hardware in the source ATM interface.

## "The True Barrel Shifter":



- 'slow' synchronization of the sources.
- May be possible by software, with an external interrupt. (software switching overhead  $\sim 10$  usec)

## Event-Builder Demonstrator

### *Internal flow control:*

- It guarantees zero data loss inside the switch,
- It shows, under particular circumstances, much lower latencies than traffic shaping,
- The event builder system is simpler (no traffic shaping).

But it is not a panacea ...

- It is highly dependant on the traffic conditions,
- It is highly sensitive to traffic fluctuations.

--> to be used under low(?) loads.

### *Combination of traffic shaping and internal flow control ?*

- Advantages of traffic shaping and very low data loss probability.
- Could we apply a less strict traffic shaping ?

### *An event-builder demonstrator is useful to test:*

- Higher level software protocols,
- Some results from the modelling (e.g. EB Latency, throughput),
- Traffic shaping methods,
- Cell losses,
- Commercial adapters.
- The interoperability between the interfaces and the switch,

It can also be used as a real time simulator of heavy traffic with event-builder characteristics.

### *Building blocks:*

- *An ATM switch*
- *Source modules:* ATM full function adaptors or 'Data generators'
- *Destination modules:* ATM full function adaptors or 'sinks'
- *An ATM/SONET protocol test equipment*

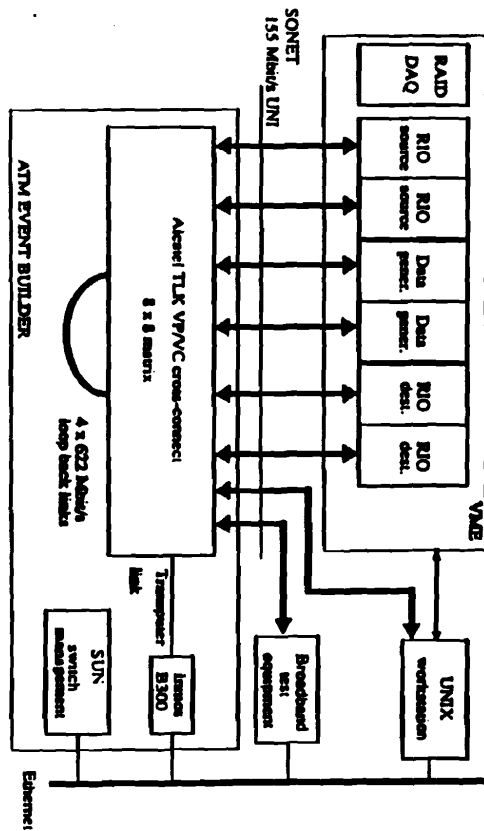
Two demonstrators will be set up, including:

- a telecom switch (no internal flow control)
- a switch with internal flow control.

# VME - ATM Adapter

## Goals:

- Gain experience with the ATM technology and standards,
- Check if and how the functionalities needed for event-building can be implemented,
- Check if and how it is possible to reach sustained maximum rate.
- Implement the additional hardware required for the 'Randomizer' traffic shaping technique,
- Develop software protocol layers.



VME-based event builder demonstrator

# VME - ATM Adapter (ctnd)

**Specifications:** (see poster by L. Gustafsson et al.)

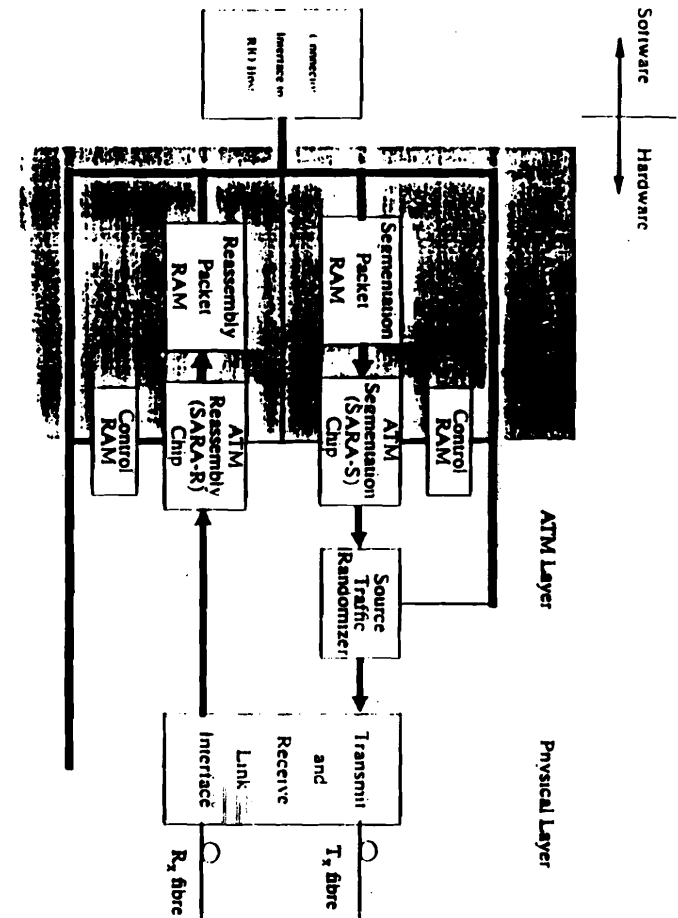
- 155 Mbit/s OC-3 SONET/SDH, multimode optical fibers,
- SARA (alias FRED) (SAR chipsets) for ATM and AAL5,
- SUNI for SONET framing,
- Implemented as daughter board on a CES RIO.

**Current Status:**

- Loop-back tests successful,
- Interoperability with HP test system,
- Proceeding gently towards nominal bandwidth ...

**Future plans:**

- Interworking tests with the ALCATEL switch,
- Produce several modules for the demonstrators,
- PCI interface,
- ...





## Data generators

22

### Requirements:

- Send pre-loaded data packets, on trigger,
- Data packets could contain meaningful data,
- Memory must be sufficient to contain a 'significant' amount of event fragments (depends on application),
- The effect of traffic shaping must be emulated.
- Half-duplex,

### Implementation:

- Data stored in memory are already segmented into ATM cells with headers (no hardware segmentation is required)
- Only the hardware of the physical layer is required,
- Multi-buffers for traffic shaping are simulated by mixing cells belonging to different VCI's. Empty cells can be inserted if necessary,

### Advantages:

- cheaper and more compact than ATM adapters,
- simpler control: just load and go.

### RD31: Distribution of activities

	CERN	SACLAY	KTH	UPPSALA	MIT
Modelling	All types ATM Conical	Flow control. MODSIM ATLAS L2 traffic patterns	Multi process-sors systems		generic for CMS
Adapter	Development, PCB. Tests	Evaluation of commercial adapters	Randomizer	Design, PCI interf. Use in real DAQ	
Demonstrator	ALCATEL switch	Int. flow control switch			
Data generator					
Software	Tests of adapter. High level protocols	Evaluation of low level protocols in commercial adapters			

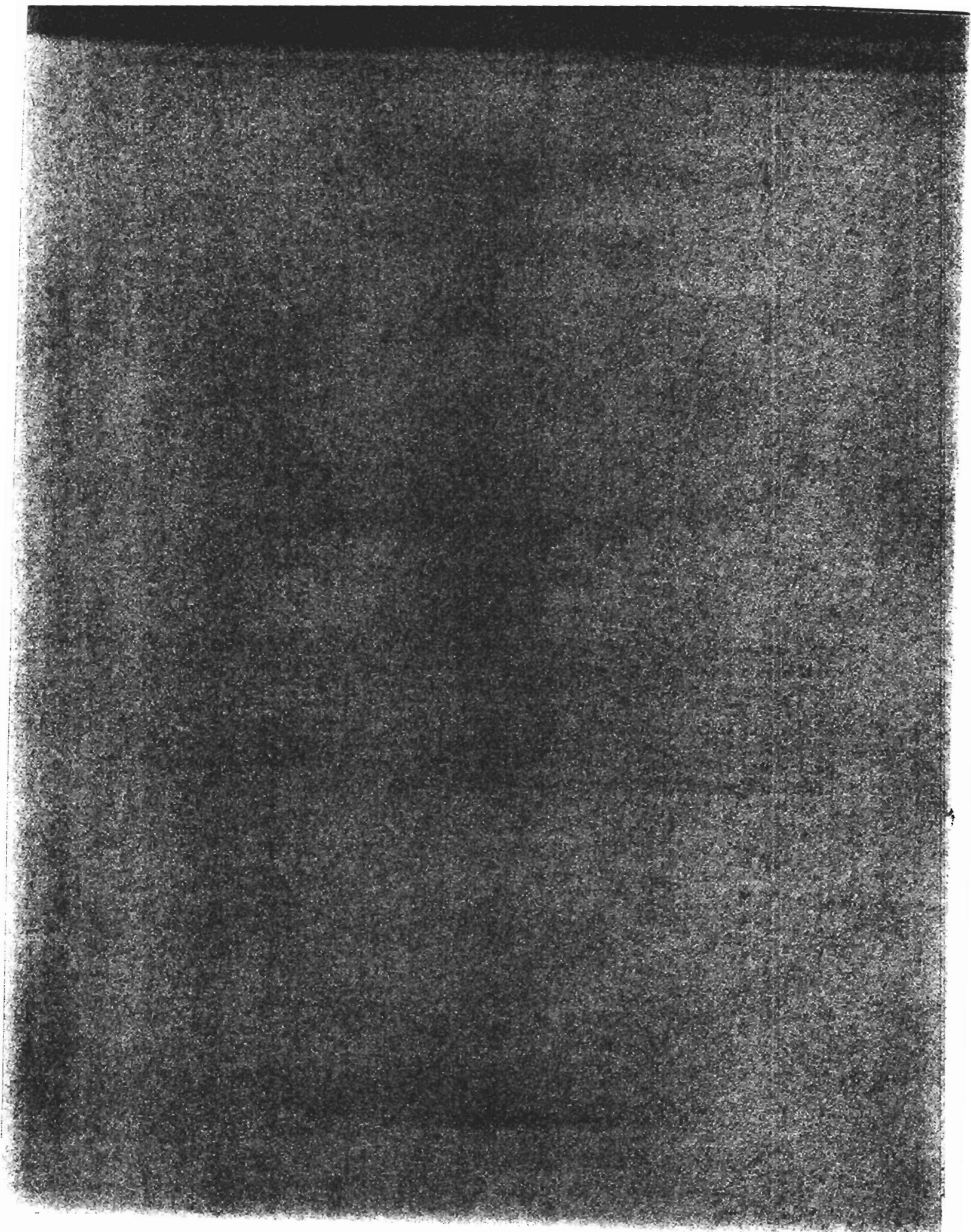


S3-2

**"Fibre Channel Research Projects"**

**(Erik van der Bij - CERN)**

Fibre Channel seems to be an ideal candidate for use in data acquisition systems as components, switches and interfaces are readily available. Already now Fibre Channel components have been used in optical links in the NA48 DAQ system. Other projects use the full Fibre Channel protocol, up to layers such as TCP/IP. Several LHC experiments and Euroball are testing and building Fibre Channel boards. The presentation will describe the different DAQ research projects that investigate the use of Fibre Channel. Up to date information can be found on <http://www.cern.ch/HSI/fcs/applie/applie.htm>, which is part of CERN's High Speed Interconnect pages <http://www.cern.ch/HSI>.





CERN

# Fibre Channel Research Projects

*Erik van der Bij*

CERN  
European Laboratory for Particle Physics  
Geneva, Switzerland  
E-mail: Erik.van\_der\_Bij@cern.ch

*Erik van der Bij*

*division ECP EDUIDQ*



CERN

# Fibre Channel Research Projects


CMS

ALICE

ATLAS

Euroball

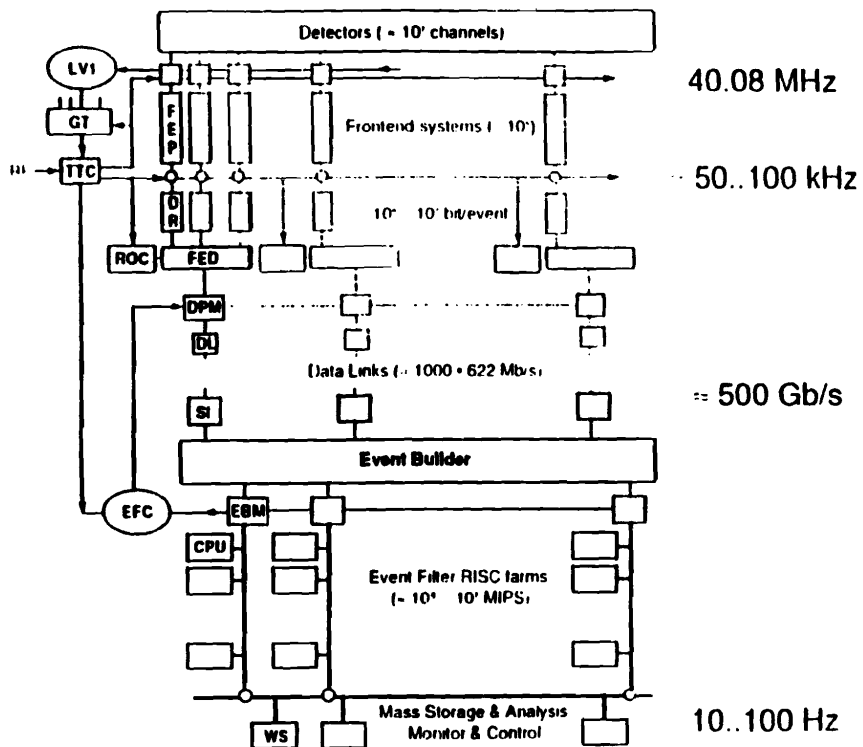
Other projects

- 
- NA48
  - CN CORE
  - RD31
  - STAR

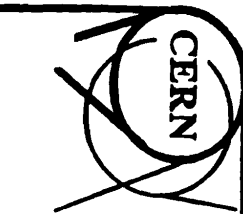
*Erik van der Bij*

*division ECP EDUIDQ*

# CMS data acquisition



- |            |                             |            |                      |
|------------|-----------------------------|------------|----------------------|
| <b>LV1</b> | Level 1 Trigger             | <b>DPM</b> | Dual Port Memory     |
| <b>GT</b>  | Global Trigger              | <b>DL</b>  | Data Link            |
| <b>TTC</b> | Timing, Trigger and Control | <b>SI</b>  | Switch Interface     |
| <b>FEP</b> | FrontEnd Pipeline           | <b>EBM</b> | Event Buffer Memory  |
| <b>DR</b>  | DeRandomizer                | <b>EFC</b> | Event Flow Control   |
| <b>ROC</b> | ReadOut Controller          | <b>WS</b>  | Work Station Cluster |
| <b>FED</b> | FrontEnd Drivers            |            |                      |



# CMS Fibre Channel



- DPM** Dual port memory FCS output (RD12)
- Fabric for Event Building
- Workstation Interfaces
- Sun SPARCStorage
- KFKI** FCS tester (KFKI/CERN/HP)

Erk van der Bij

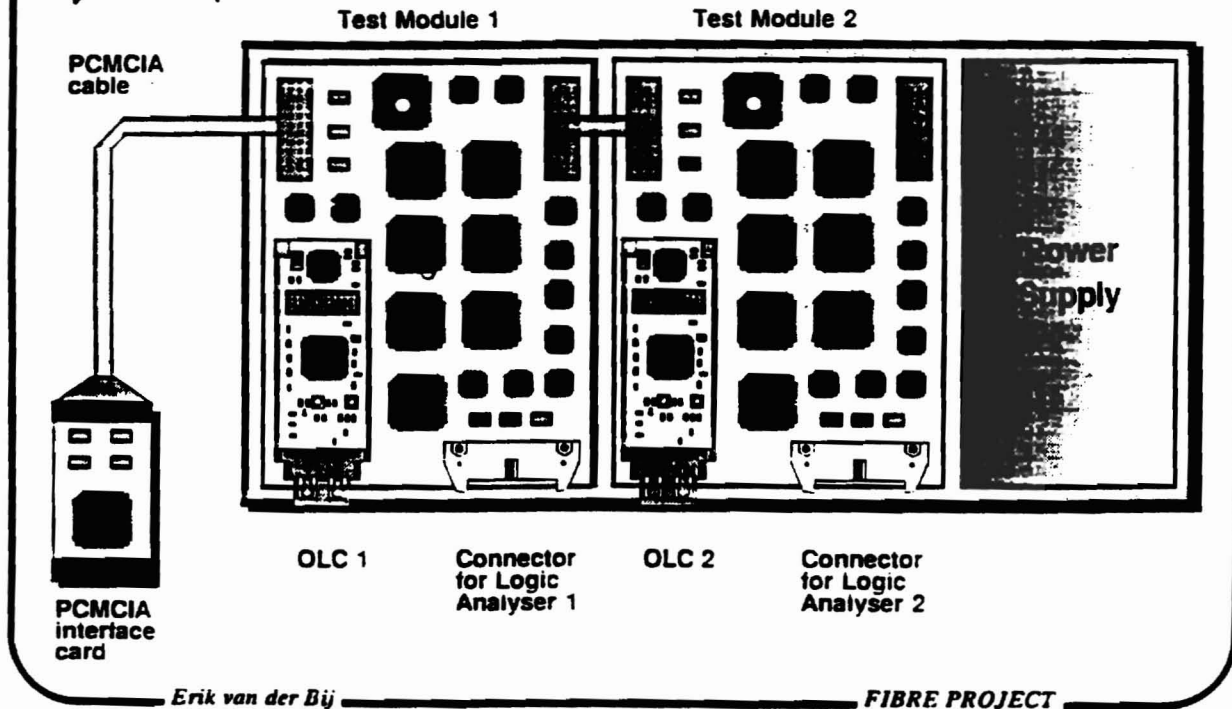
division ECP EDUDD

CERN

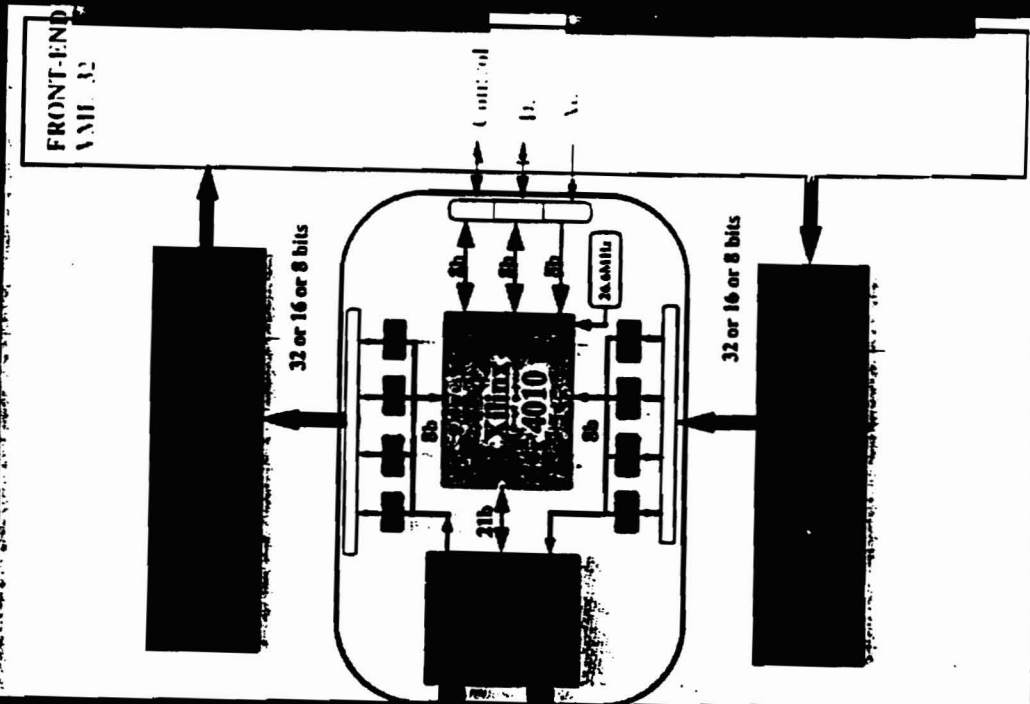
KFKI

hp

# Fibre Channel Tester



## Fibre Channel Data Pusher



# Frame Editor

Field name	Synthetic value	Numeric Value	Word 0	Byte 3	Byte 2	Byte 1	Byte 0
SOF	SOFc1	K28.5, D21.5, D23.0	00701	00	00	00	00
R_CTL	VD_UNSQL_CTRL	01000010	00	00	00	00	00
D_ID	AS_BEFORE		00	00	00	00	00
S_ID	AS_BEFORE		00	00	00	00	00
TYPE	IPI3_SLAVE	00010010	00	00	00	00	00
F_CTL	Manually (24 bits)	01100111, 00000000	00000000	00000000	00000000	00000000	00000000
SEQ_ID	Manually (1 byte)	00000100	00000000	00000000	00000000	00000000	00000000
DF_CTL	Expi.Netw.Asso.16 B	01110001	00000000	00000000	00000000	00000000	00000000
SEQ_CNT	INCREMENT		00000000	00000000	00000000	00000000	00000000
QX_ID	Manually (2 bytes)	10, 0	00	00	00	00	00
RX_ID	Manually (2 bytes)	0, C	00	00	00	00	00
PARAM		00000000, 00000000	00	00	00	00	00
EXP_HEAD	Manually (16 bytes)		00	00	00	00	00
NW_HEAD	Manually (16 bytes)		00	00	00	00	00
ASS_HEAD	Manually (16 bytes)		00	00	00	00	00
DEV_HEAD	Manually (16, 32 or 64)		00	00	00	00	00
LS_COMM			00	00	00	00	00
payload							
CRC	CALCULATE		00	00	00	00	00
EOF	EOFt	K28.5, D21.5, D23.0	00	00	00	00	00

Erik van der Bij

FIBRE PROJECT

# Dual Line Monitor Display

Working mode: Through Full

Defined Channels:

- Channel 1 -> Module 1
- <- Channel 2
- Module 2 <- Channel 2
- Channel 1 ->

Channel 1:

- 22007 ns Frame
- 22012 ns
- 22099 ns Idles
- 22102 ns
- 22334 ns Idles
- 22335 ns

Channel 2:

- 22171 ns R\_EDY
- 22176 ns
- 22245 ns Idles
- 22249 ns

Start of frame (Class 2):

- R\_ctl 04 (Unsolicited Data)
- Ports : 000037 -> 000017
- Type 09 (SCSI GPT)
- Seq 02, 00004

Start of frame (Class 2):

- R\_ctl C0 (ACK 1)
- Ports : 000017 -> 000037
- Type 0
- Seq 02, 00004

Erik van der Bij

FIBRE PROJECT

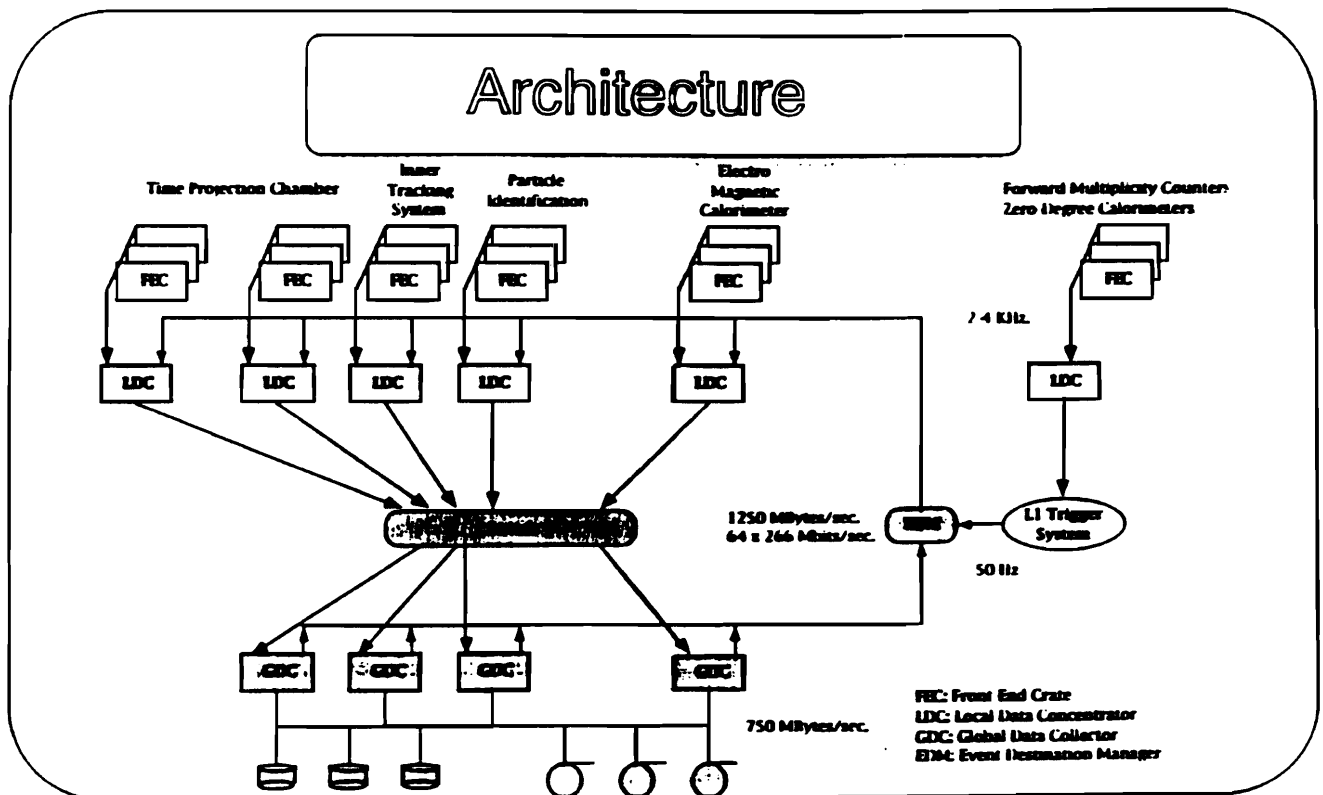


# ALICE Fibre Channel

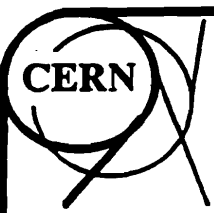
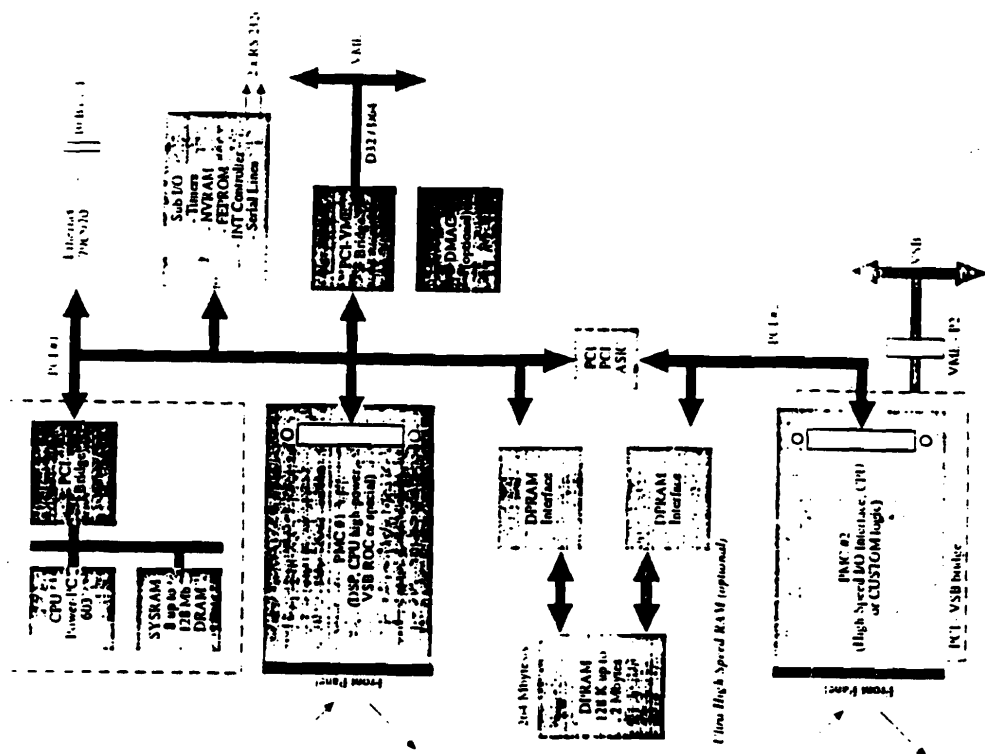
- Dual port memory FCS output (RD12)
- Fabric for Event Building
- VME and PCI FCS interface (CERN/ECP)
- Workstation Interfaces
- FCS tester (KFKI/CERN/HP)
- Storage

Erik van der Bij

division ECP EDU/DQ



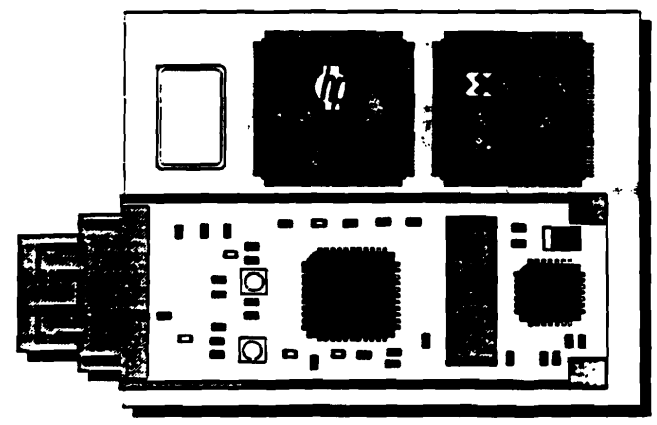
# RIO2 8060 General Block Diagram



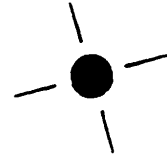
## VME/PCI/FCS interface

- CERN/ECP : Hardware
- LBL : Software for VME platforms
- KFKI/RMKI : Software for Windows platforms

PCI/FCS interface for Level 2 and Level 3 trigger processor module.  
 First used on a RIO II with LynxOS from CES, other PCI platforms later.



# ATLAS Fibre Channel



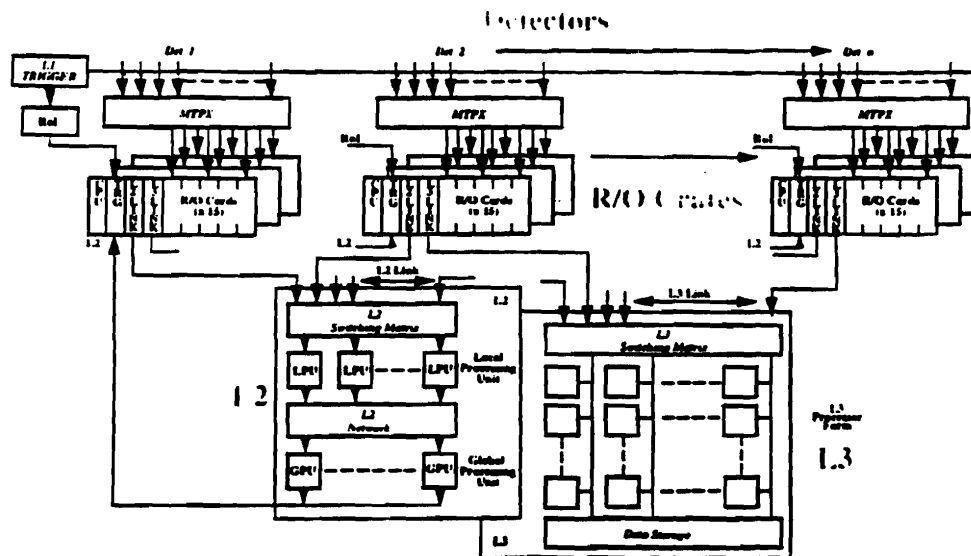
## Experience and Simulations with HIPPI (RD13 & STAR)

- Fabric for Event Building
- VME and PCI FCS interface (CERN/ECP)
- FCS tester (KFKI/CERN/HP)

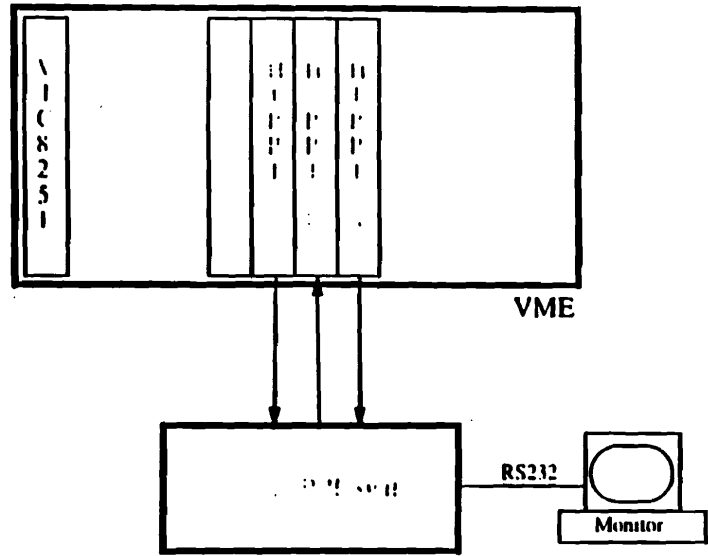
Erik van der Bij

division ECP EDUIDQ

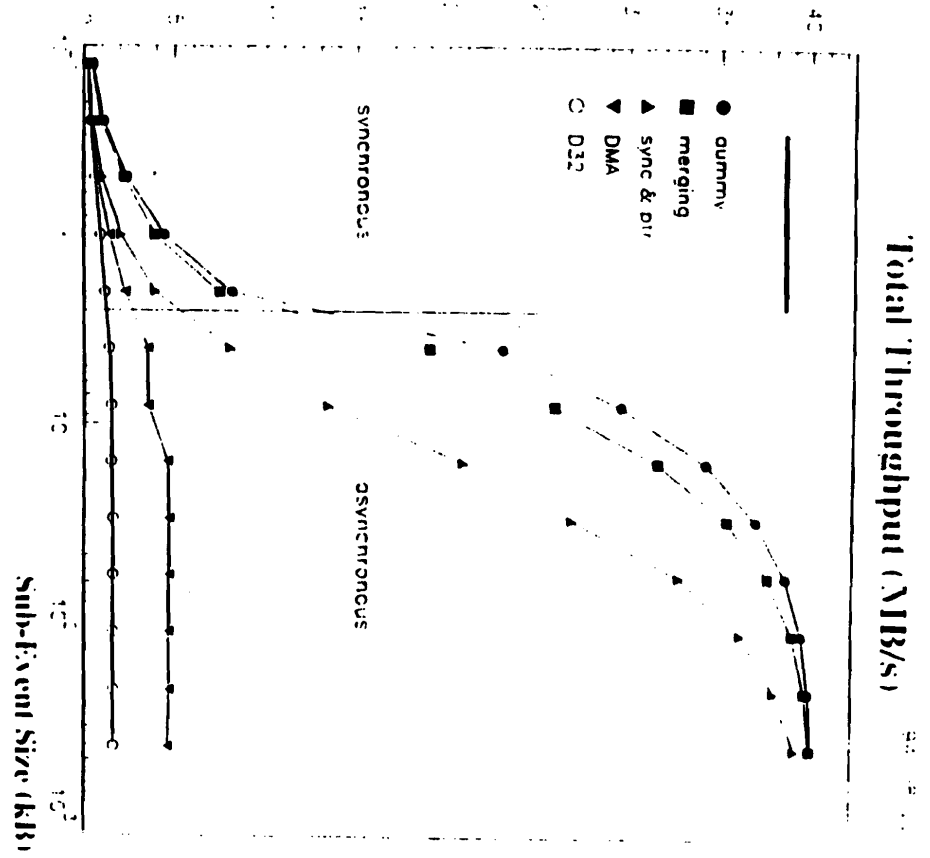
## Functional Model of ATLAS DAQ



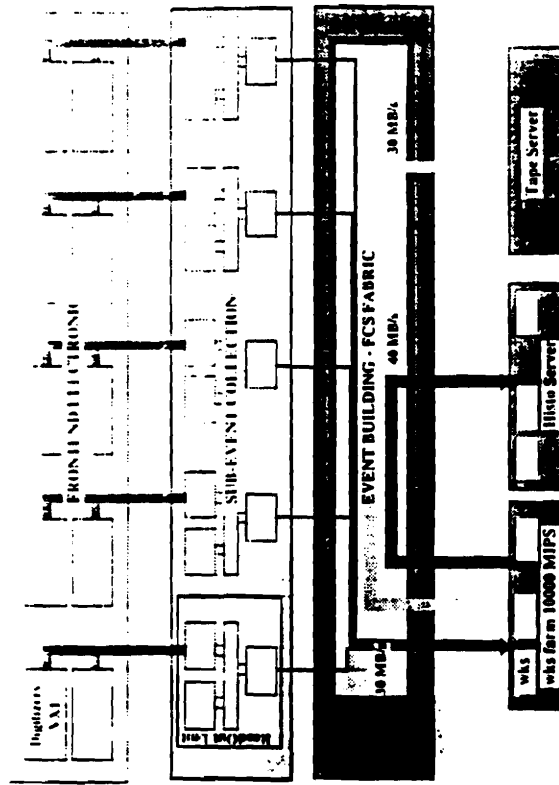
## EB Prototype



- VIC 8251: VME arbiter
- R3000, 32 MByte DRAM, EP/LX = real-time UNIX
- R3051, 4 MByte DRAM, HiPPI interface
- 8x8 switch  
switching delay < 1us  
logical addressing: 12 bits of CCI word => output port(s)  
camp on: Src requests are held in fifo



## Euroball DAQ



Frontend Electronic based on Finogam I/II electronic + new cluster electronics

Event Collections based on in home developed Readout Units

Event Building based on switching network; the events are automatically built in the target processors. Technology today available:

- HPP1, FCS, ATN

On-line data processing based on a 10000 MIPS workstation farm

The same system architecture can be used for the off-line data analysis

CERN

## Euroball Fibre Channel



..... Readout module

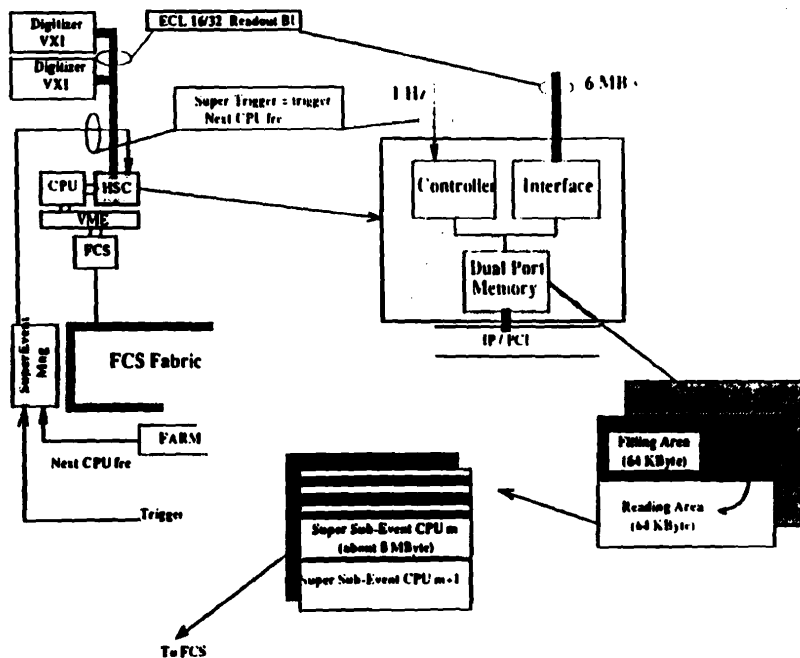
- VME and PCI FCS interface (CERN/ECP)
- Fabric for Event Building and Distribution
- Workstation Interfaces



..... TCP/IP and Direct Channel measurements (RD11 & Euroball)

- FCS tester (KFKI/CERN/HP)
- Storage

## IP based Readout Unit

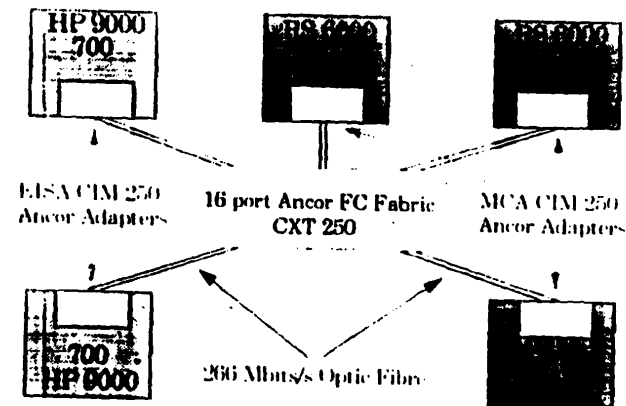


The readout unit is based in this case on two VME boards.

- the main CPU where, by means its standard IP (Industry Pack) interface, is located also the d132 readout interface. Such interface has the same functionality of the VME based one, but fits a 9x9 mezzanine board. It provides 500 Kbyte of dual port memory, the controller and the logic to drive the d132
- the interface to the switch

A second development step is foreseen to design a PCI based readout interface. In this case the interface to the switch should be based on PCI too, compacting in this way the full readout unit into a single VME board. PCI based workstation should be also considered in alternative to the VME based CPU.

## Experiment with IBMs and HPs connected around a Fibre Channel Fabric



Communication tests have been performed:

- \* under TCP/IP protocol:
  - Between IBM workstations,
  - Between HP workstations,
  - Between IBM and HP workstations.
- \* under Direct Channel:
  - Between IBM workstations.

## Experimental results

Table 1: Communication speed (MBytes/sec.)

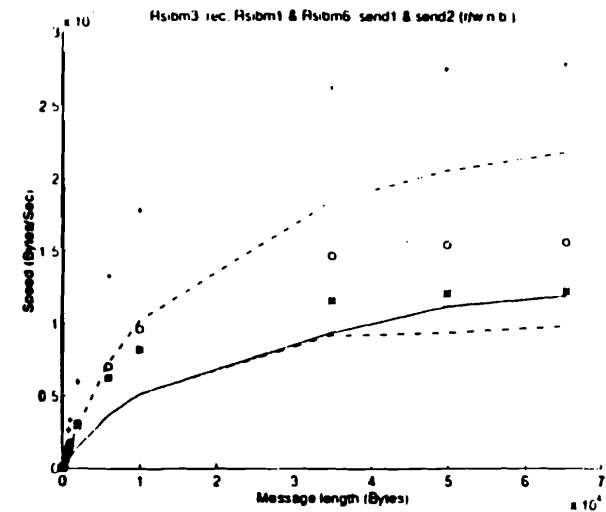
Direct Channel	64 Bytes	1 KBytes	64 KBytes
IBM-IBM	0.11	1.7	15.7

Table 2: Communication time (milli sec.)

Direct Channel	64 Bytes	1 KBytes	64 KBytes
IBM-IBM	0.58	0.595	4.19

IBM-IBM : from RS/6000-C10 (power pc601 at 80MHz) to RS/6000-590 (power2 at 66 MHz)

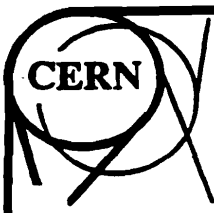
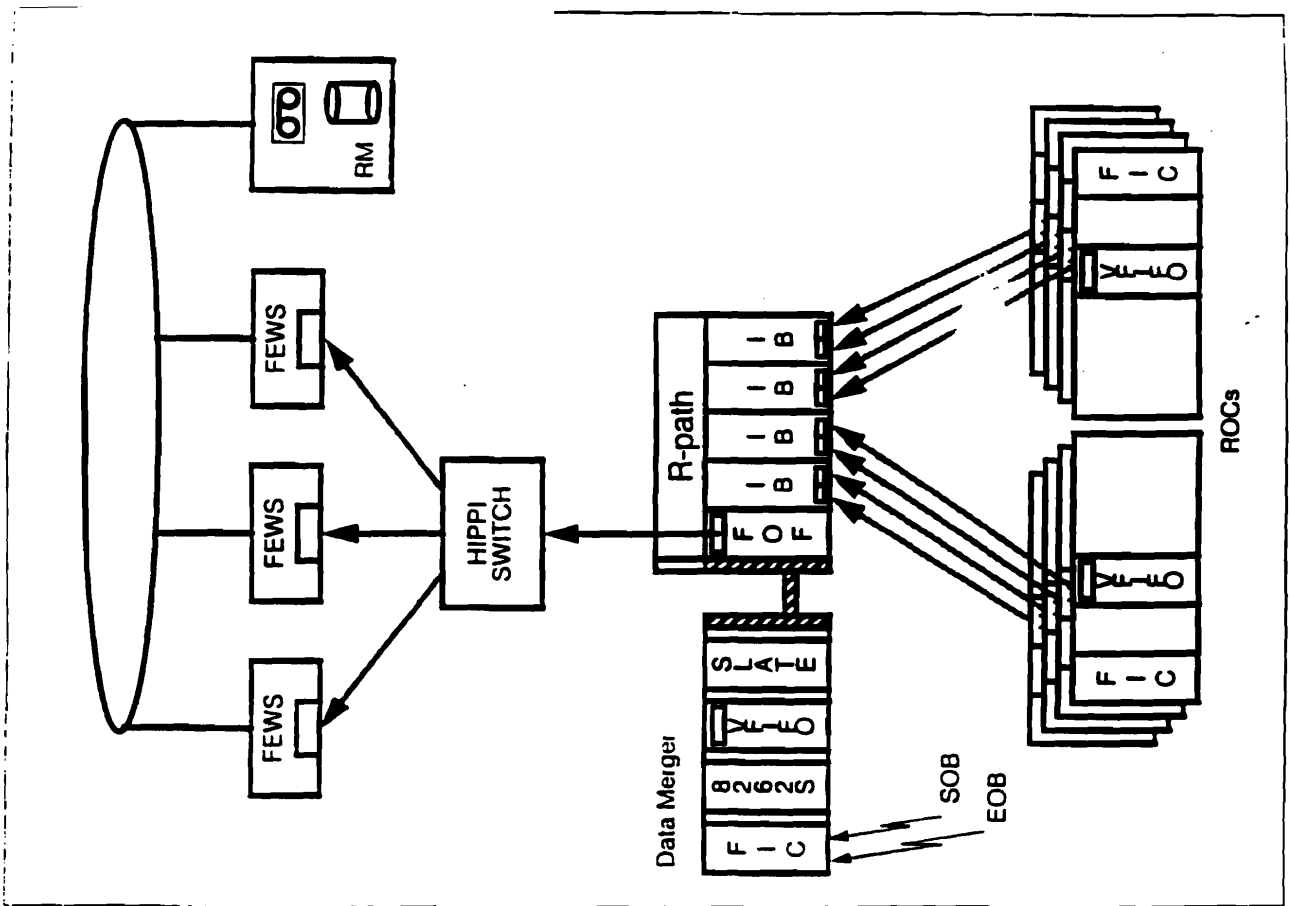
## Communications between three IBM machines (Direct Channel protocol)



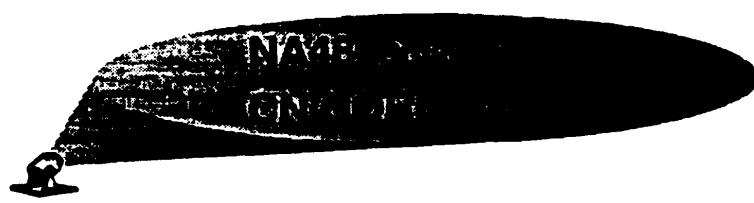
dashdot line : performance curve of Sender1 (RS/6000 250),  
 star line : theoretical performance curve of Sender1,  
 solid line : performance curve of Sender2 (RS/6000 C10),  
 circle line : theoretical performance curve of Sender2,  
 dashed line : performance curve of Receiver (RS/6000 590),  
 plus line : theoretical performance curve of Receiver,  
 dotted line : Maximum speed of an optic fibre.

Overhead : 680 microsec.

Speed of 21.8 MBytes/sec for transmitting 64 KByte messages.



## Other Projects using Fibre Channel



- RD31 (Switch simulation)
- STAR (Ancor VME, Ancor fabric, simulation)



# Conclusions

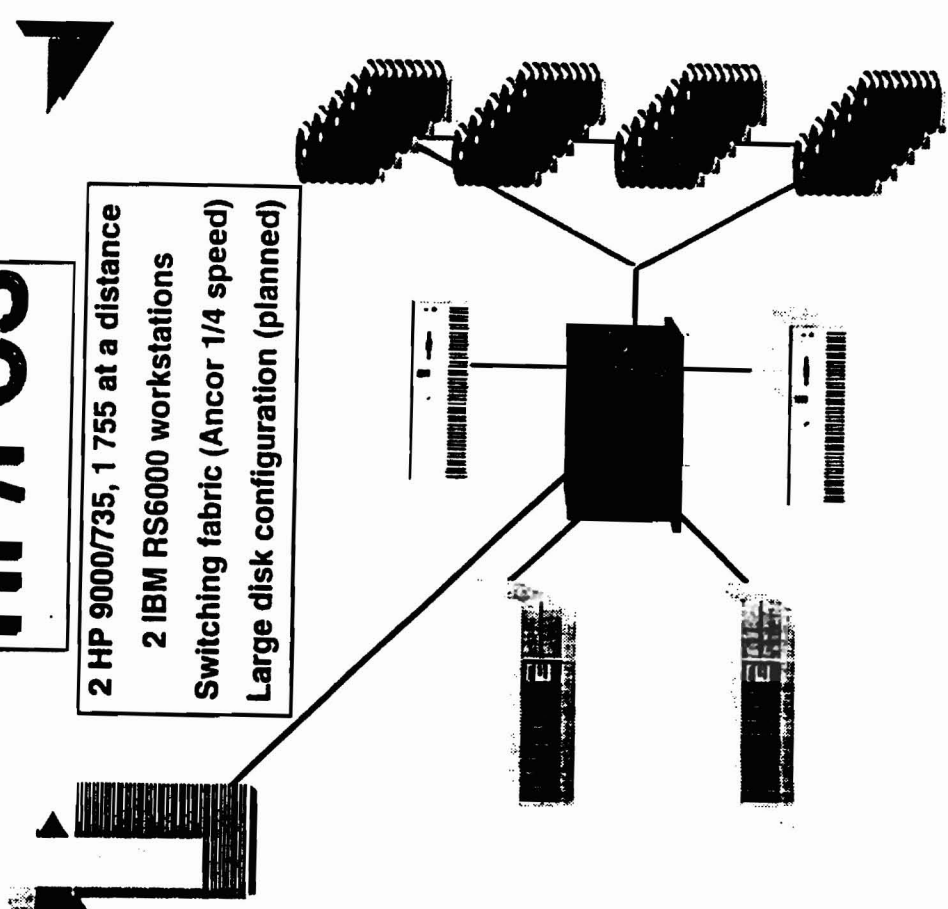
- ◇ All major experiments use or study Fibre Channel
- ◇ Commercial solutions available for many parts
  - Workstation interfaces
  - Fabrics
  - Disk and tape storage
- ◇ Hardware is developed for special DAQ requirements
  - Dual Port Memory output
  - Fibre Channel testers
- ◇ Hands-on experience is quickly growing
  - Workstation programming TCP/IP, Direct Channel
  - Knowledge of protocols
  - Measurement results
  - Simulation models

Erik van der Bij

division ECP EDUIDQ

## HP/FCS

2 HP 9000/735, 1 755 at a distance  
 2 IBM RS6000 workstations  
 Switching fabric (Ancor 1/4 speed)  
 Large disk configuration (planned)



Full speed switch later (YE94 ?)  
 Disk connected in an arbitrated loop.  
 (Either directly to switch or through a server)  
 Throughput of close to 100 MB/s.

August 1994



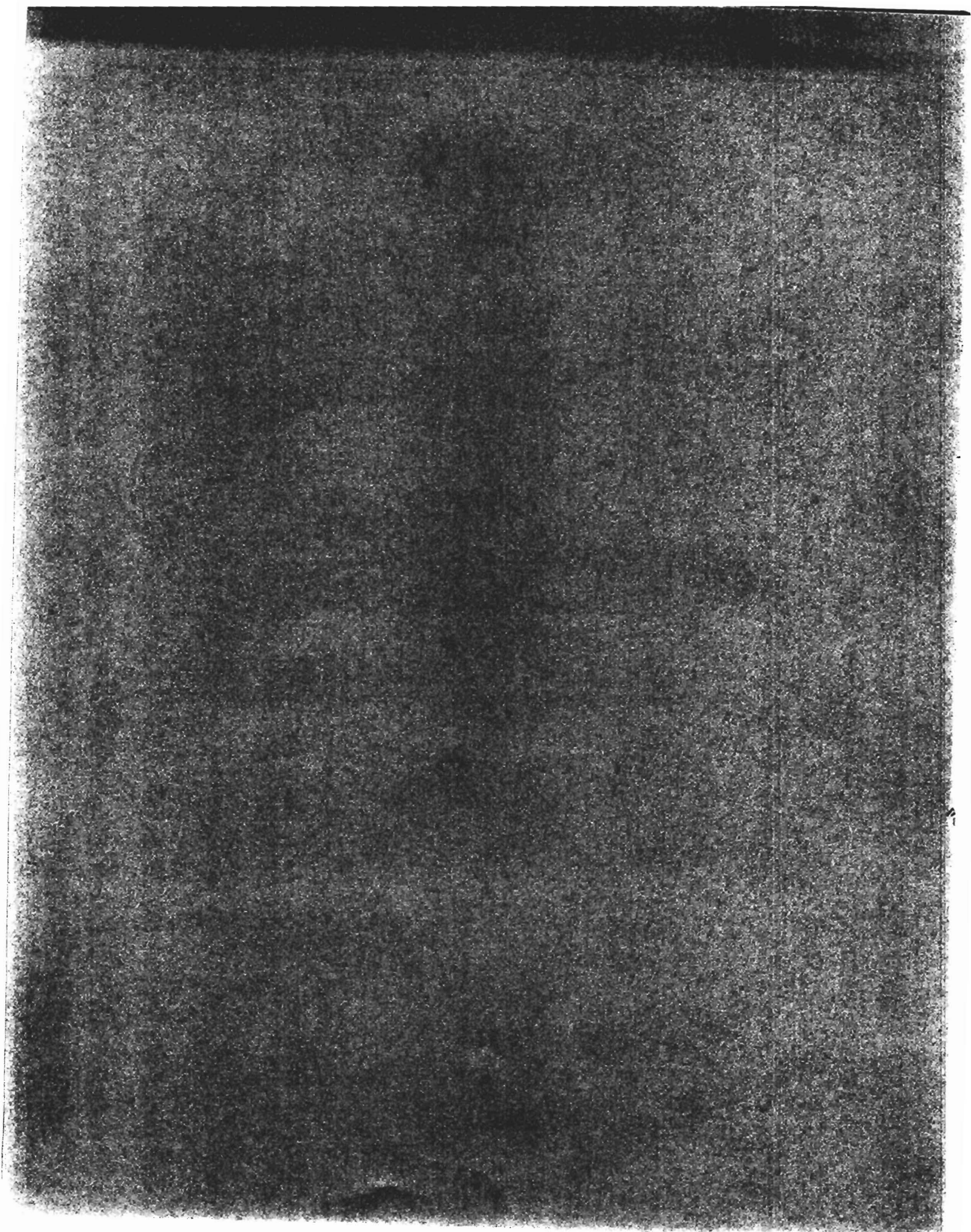


**S3-3**

**"SCI Research Projects"**

**(Fred Wickens - Rutherford)**

Discussion of goals and progress in RD-24 & other international SCI R&D projects including STAR & NA49. Review of known commercial SCI research projects with possible DDO applications.



## A Review of SCI Projects

F.J.Wickens

Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, OX11 0QX, UK.

### INTRODUCTION

The advent of the Scalable Coherent Interface standard (IEEE 1592)[1] and the development of components to support it has been accompanied by strong interest in several communities, especially: computer companies; computer scientists; and particle physicists working towards the next generation of experiments. For commercial reasons the first group have tended to do their developments behind closed doors, the second group have been particularly interested in the possibilities for using the coherent shared memory supported by the standard, but it is the third group who have done much of the openly published work studying the potential for this standard to be used for transporting data between memories and processors and between different existing bus standards. This review will concentrate on the work of this last group, but with some references to various commercial developments elsewhere.

### THE SCI STANDARD

The SCI standard provides for very high performance interconnects (GByte/s) between processors and memories, through networks of uni-directional point-to-point links, which allow bus like services. Typically SCI nodes would be organised into small rings, with bridges and switches between rings. Since all SCI links can transfer data concurrently and transactions are split into separate request and response phases, there are no arbitration bottlenecks. Within a SCI network there is a 64 bit SCI address space, of which 16 bits are used for node addressing. A range of transactions, both coherent and non-coherent, are covered by the standard. Over short distances (up to a few metres) the point to point links use cables with 18 parallel signals - 16 for data, plus clock and flag bits. For longer distances serial fibre optic links are foreseen.

### AVAILABLE COMPONENTS

The first vital ingredient for the R&D was the availability of chips and board level products to support the standard. For the HEP community this has been driven by the Norwegian company Dolphin who were responsible for the design of the first NodeChip(TM)<sup>1</sup>, initially in GaAs and later in the cheaper (but slower) CMOS form. The GaAs NodeChips were produced by Viesses[2], but Dolphin also marketed them on a VME based development board which contained not only the NodeChip and connectors for the input and output links, but also boot-logic, power converters, node clock and a connection for a parallel to serial converter for serial links. In 1994 the CMOS NodeChips[3], manufactured by LSI, became

available and Dolphin again marketed them on a VME based development board - and most importantly this was plug compatible with the earlier GaAs version of this board. In addition to the bare chips and development boards, Dolphin also produced a board using the CMOS NodeChip which plugs into the SBus of a SparcStation to give SCI connectivity between workstations. [4]

In parallel to the chip and board level products there has been a related development of cable assemblies for the interconnection of the SCI nodes. It is with these tools that most of the R&D has been done.

### THE PROJECTS

#### RD-24

Much of this review is centred around the CERN RD-24 project. This project was started to investigate the possibilities of using SCI for future experiments of the LHC era. However, over the last 2-3 years it has attracted many parties interested in SCI, both from HEP institutes and companies, many of whom have a significant part of their SCI work outside of RD-24. Thus in addition to the many valuable sub-projects within RD-24, it has played an equally important role as a co-ordinating forum for most of the groups actively working on SCI with relevance to HEP.

In 1993 RD-24 demonstrated a 2 node ring, using the GaAs NodeChips running at 500 MByte/s, one node being driven by a R3000 in a CES RIO module and the other by a 68040 in a CES FIC. For full details see the RD24 Status report - 1993 [5]

Subsequent to those tests the card used with the FIC has been further developed and now includes a DMA engine and with the CMOS node chip is now available as a commercial product from CES. [6]

Within RD-24 design work continues, at Protvino, to enable this SCI card to be used with an existing VME Dual Port Memory, i.e. without the FIC. In this way it is planned to develop more general SCI access with DMA from DPM's.

In 1994 when the CMOS components became available RD-24 demonstrated rings, with up to 4 nodes, using the prototype of the CES card mentioned above together with Dolphin SBus to SCI adaptor cards. In addition they demonstrated an SCI-SCI bridge, using a pair of modified Dolphin development cards, allowing two SCI rings to be interconnected. For these demonstrations the SCI links were running at 125 MByte/s, for fuller details see RD24 Status report - 1994 [7]

Using the SBus to SCI adaptor cards RD-24 were also able to demonstrate applications on different SparcStations transparently sharing a memory region in one of the nodes.

For the future RD-24 is now working towards building a heterogeneous mini SCI DAQ system (Figure 1) which can be configured according to the architectural preferences of each of the main LHC experiments (i.e. ATLAS, CMS and ALICE). The following sections describe the developments of many SCI components, both within RD-24 and elsewhere, which would need to complete all of the options for this system. The one vital ingredient not described below is the SCI switches and for these the reader is referred to the paper by Bin Wu from this conference [8].

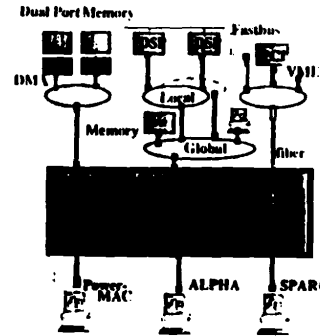


Figure 1 A heterogeneous mini SCI DAQ system

#### Bridges to Various Buses

Groups from Manchester University and RAL, have been working on prototyping exercises for the ATLAS level 2 trigger system and have produced several SCI interfaces [9].

The group from Manchester University developed their own custom SCI daughter card on a 6U Eurocard with a CMOS NodeChip, clock, control logic and 4 FIFO's - for separate input and output response and request queues, to provide deadlock free operation. Input to the board is via a 32 bit bus which with some simple interface logic can be connected to various common buses.

Ancillary boards have been produced to drive this input from VME. The SCI to VME interface, thus formed, was used both with an embedded VME controller and through a memory mapped interface into a DEC Alpha system.

Another ancillary board was produced to drive the daughter card from the global bus of a Texas Instruments TMS320C40 digital signal processor. This used the dBEX32 connection of C40 units from Loughborough Sound Images, as in their DBV42 & DBV44 modules [10]. This choice of connection to a C40 is also being pursued by another group from Valencia who are developing a more general purpose C40-SCI interface.

A fourth node was provided by RAL, who together with INFN (Rome and Lecce) have been extending earlier RD-24 work to produce an SCI-TurboChannel interface. This uses a Dolphin CMOS NodeChip development board driven by a CES RIO module using the R3000 processing in the RIO as a protocol converter.

With this equipment a four node SCI ring was first successfully operated in the ATLAS test beam line at CERN during September-October 1994 to pass detector data from a network of C40 processors to RISC processors in a prototype architecture for ATLAS level 2 triggering. This is believed the first time that 'live' detector data has been passed round an SCI ring at a beam line.

Work is now underway to produce further boards to allow the Manchester daughter cards to be driven from the PCI bus of DEC Alpha processors to give a more direct connection to the RISC processors.

Another project by Dolphin is for a VME-SCI bridge, which would support mapped transfers between VME64 and the 64 bit SCI address space, thus allowing transparent transfers between VME segments via SCI.

The University of Oslo, in collaboration with Struck, have a project to develop a Fastbus SCI bridge, connecting to the CERN Host Interface FB master[11]

SINTEF in Norway are working on the design of an SCI HIC (Heterogeneous InterConnect) interface, which could provide an alternative physical layer for the SCI packets [12].

Because of its adoption by RISC processor and VME board manufacturers the PCI bus standard, especially in the more recently agreed mezzanine card format (PMC), is now receiving a lot of interest from many groups. In particular RD24 and the STAR collaboration [13], together with CES, Apple and BiRa have all expressed interest in developing SCI-PCI interfaces. One aspect of this work is a small prototyping exercise within the STAR collaboration which plans to allow a priority scheme for the requests and responses by using DPM's instead of the more usual FIFO's for the queues. In addition they plan to handle the management of the Unified PCI transactions vs the Split SCI transactions using FPGAs.

Other projects outside the HEP community are:

- Norwegian Telecom are developing an ATM-SCI interface, this would map ATM channels to separate buffers within the SCI address space of a system. So

<sup>1</sup>NodeChip is a trademark of Dolphin Interconnect Solutions.

that channels would go directly to the appropriate physical memory and processor.

- Apple are developing an interface to bridge from the PowerPC bus of a PowerMac to SCI

### Memories

Another important ingredient for DAQ systems are memory modules. CIEMAT in Madrid are developing a SCI 16 MByte DRAM in a VME module which will support non-coherent read and write and move operations.

The University of Oslo is developing a VideoRAM which will support all of the request commands defined for the present NodeChips, including cache coherency [14]

### Serial SCI

Within RD-24 initial tests to use the Dolphin SCI connection to Gigalink chips to provide parallel to serial conversion were limited to local loopback. SL Division in CERN is now taking these tests further, connecting the Dolphin cards to modules from Lasertron, Finisar and BT&D to implement long distance shared memories coupled via fibre optics. It is planned to use these for accelerator controls.

### FUTURE COMPONENTS

The direction and rate of progress in the medium term will be strongly influenced by the availability of new components and board level products. Already some of the projects are planning to use the next generation of Node chips such as the Line Controller [15]. It is also very likely that more future chips will use the Low Voltage Differential Signalling (LVDS) technology which promises high speed, combined with low power and low costs. Already IBM have demonstrated SCI links, using this technology with BiCMOS chips, running at 1 GByte/s [16].

### CONCLUSIONS

SCI cannot yet claim to be a mature technology, but interfaces have been demonstrated to many of the components used in the latest and planned experiments. With more SCI products becoming available, and the broad sweep of projects investigating SCI for the DAQ environment it promises to satisfy many of the most demanding needs in the coming round of experiments.

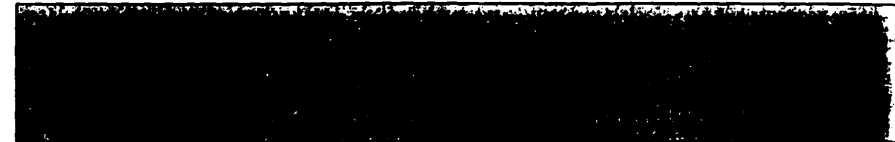
### ACKNOWLEDGEMENTS

The author would like to thank the many SCI co-workers who supplied information, but especially Hans Muller for the use of much prepared material and many useful discussions. Also special thanks go to Volker Lindenstruth and Andre Bogaerts who have clarified many details of SCI. Finally thanks are due to the UK colleagues from RAL, and the University of Manchester who have shared the joys and pain of working on such new technology.

### REFERENCES

1. "SCI, Scalable Coherent Interface". IEEE standard 1596-1992
2. DST501A GaAs NodeChip, Dolphin, Oslo, Norway.
3. L6400 SCI NodeChip Technical Manual, LSI Logic Corporation
4. Shus to SCI Adapter, Dolphin Interconnect Solutions AS, Oslo, Norway.
5. RD24 Status Report 1993, CERN/DRDC/93-20
6. Product SCI 8225, Creative Electronic Systems (CES), Geneva, Switzerland.
7. RD24 Status Report 1994, CERN/DRDC/94-23
8. Bin Wu, SCI Switches, This conference
9. P Clarke et al. SCI with DSPs and RISC processors for LHC 2nd Level Triggering. Poster paper submitted to this conference
9. DBV42 & DBV44 Technical Reference Manuals, Loughborough, Sound Images Ltd, Loughborough, UK.
10. B Skaali, Fastbus CIB SCI link. Poster paper submitted to the conference
11. E Kristiansen, Switches for Point to Point Links using OMI III Technology. This conference
11. V Lindenstruth, STAR SCI Backbone. Poster paper submitted to this conference
12. B Skaali, An SCI VideoRAM Memory Module. Poster paper submitted to this conference
13. V Lindenstruth, Overview of SCI Integrated Circuits and Board Products. This conference
14. W Nation, Design of SCI-Class Interconnects. This conference

SCI Nodechip + Bridge chip availability



SCI  
Research  
Projects

IDAC

Fermilab, Oct 26th, 1994

Fred Wickens

Rutherford Appleton Laboratory

used in RD24 1st phase:

1993 GaAS Nodechip Fujitsu/Convex  
1993 CMOS CCMC\* Convex  
667 Mbyte/s GaAs (not available  
to RD24)

1994 CMOS Nodechip LSI Logic: L64601  
125 Mbyte/s low cost/speed 4 Watt  
available in quantities

Scheduled for 1994

used  
in RD24  
2nd phase

200-250 Mbyte/s Line Controller  
overall performance gain ~ 3 \* L64601

to be used  
in coming phase

1 Gbyte/s Node LVDS  
1 Gbyte/s 2-way switch

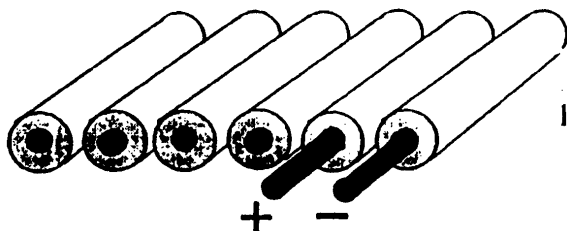
Scheduled for 1995

Cache and Memory Controller CMC  
4-way switch

New: IBM SCI chip within IBM project AS40

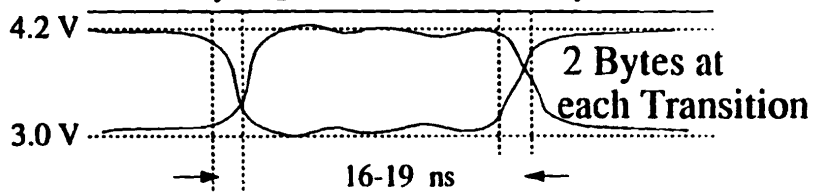
# LSI Logic's L 64601 CMOS Nodechip

DIFFERENTIAL Pseudo-ECL (ECL + 5Volts)

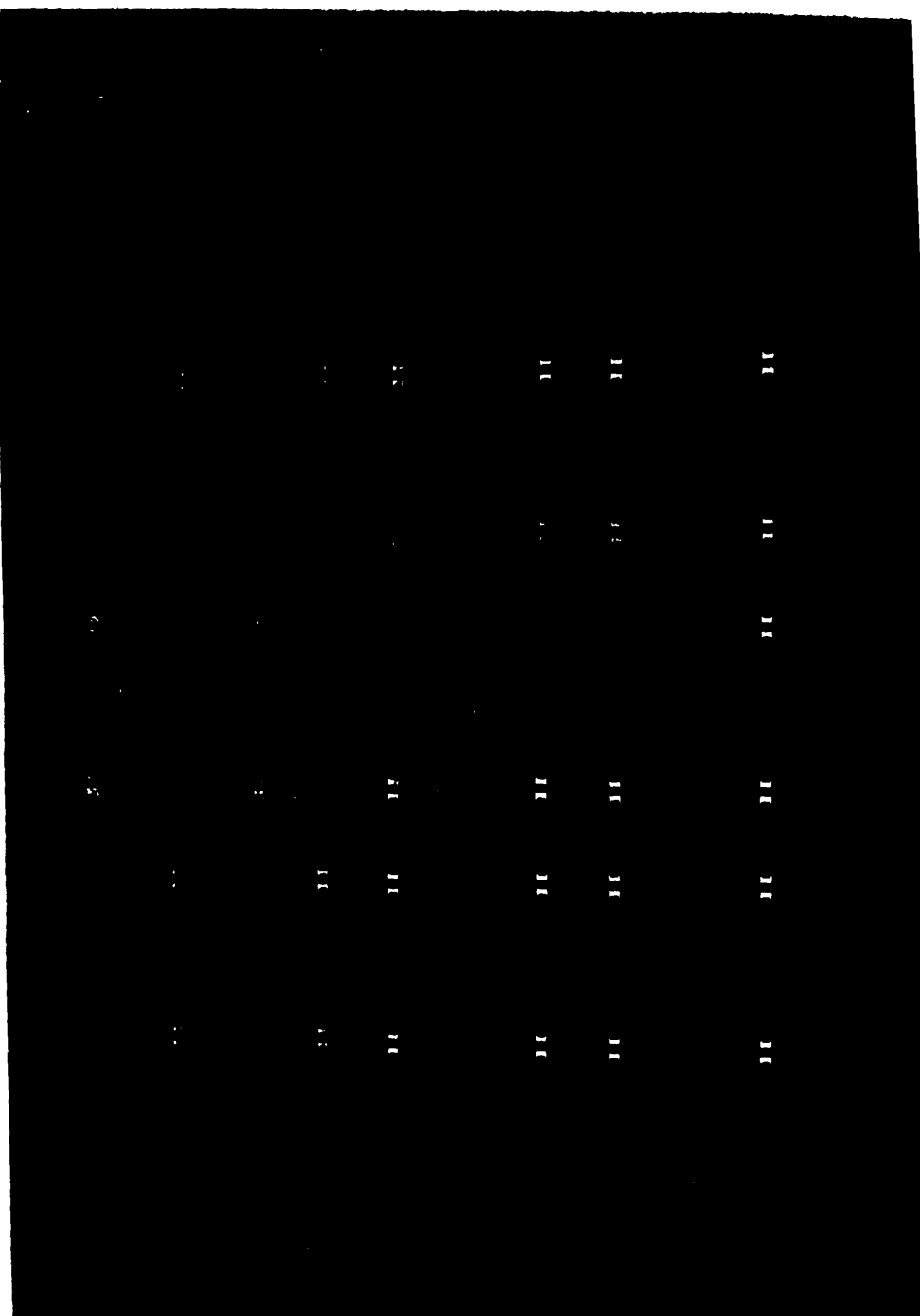
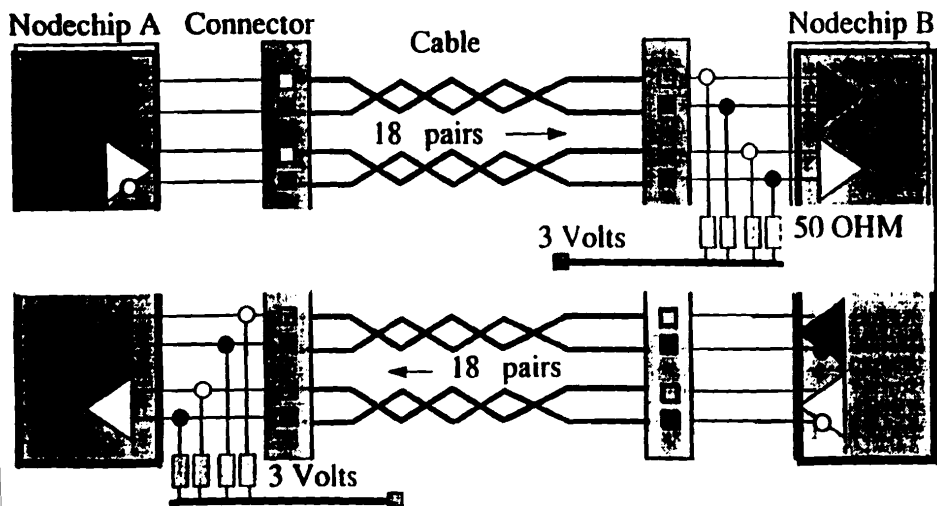


16 DATA + 1 Clock + 1 FLAG  
-> 18 pairs

2 bytes @ 62.5 MHz => 125 Mbyte/s



Chip directly drives/receives SCI, 50 OHM termination







## RD24 COLLABORATION

### *Application of the Scalable Coherent Interface to Data Acquisition at LHC*

Status July 1994

A. Bogaerts<sup>1</sup>, R. Keyser, H. Müller<sup>1</sup>, G. Magnas, P. Poitang, D. Samyn, P. Werner  
CERN, Geneva, Switzerland

B. Skali, E.H. Kristiansen<sup>2</sup>, H. Golparian, J. Wikne, B. Wu  
University of Oslo, Department of Physics, Norway

S. Gjessing  
University of Oslo, Department of Informatics, Norway

S. Falciano, F. Cesaroni, G. Modici  
INFN Sezione di Roma and University of Roma, La Sapienza, Italy

P. Cusi, M. Panico  
INFN Sezione di Lecce, Italy

A. Sytin, A. Ivanov, A. Ekimov  
IHEP, Protvino, Russia

E. Sanchez-Peris, V. Gonzalez-Millan, J.M. Lopez-Amengual, A. Sebastia, J. Ferrer-Prado  
IFIC, Valencia, Spain

F.J. Wickens, D.R. Boscill, R.W. Husley, J.L. Leake, R.P. Middleton  
Rutherford Appleton Laboratory, Didcot, UK

R. Hughes-Jones, S. Kolva, R. Marshall, D. Mercer  
University of Manchester, UK

V. Lindenstruh

Lawrence Berkeley Laboratory, Berkeley, CA, USA

K. Lachson, S.E. Johansen, H. Kohnmann, E. Rongved  
Dolphin Interconnect Solutions A.S., Oslo, Norway

A. Guglielmi, A. Pastore  
Digital Equipment Corporation (DEC), Joint Project at CERN

F.H. Worm, J. Bovier, A. Lounis  
Creative Electronic Systems (CES), Geneva, Switzerland

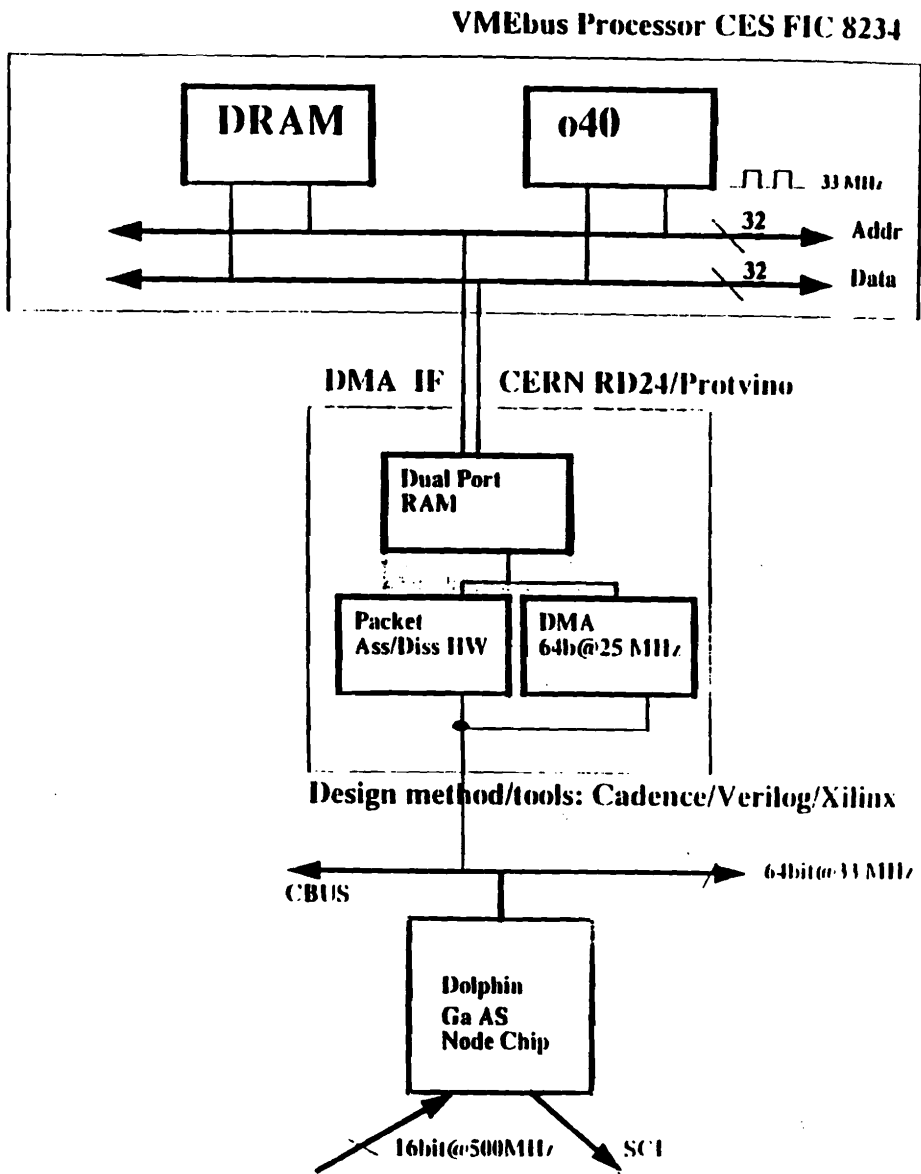
R. Hon, D. North, G. Stone  
Apple Computer, Inc. Cupertino USA

E. Peres  
Thomson-TCS Semiconducteurs Specifiques, Orsay, France

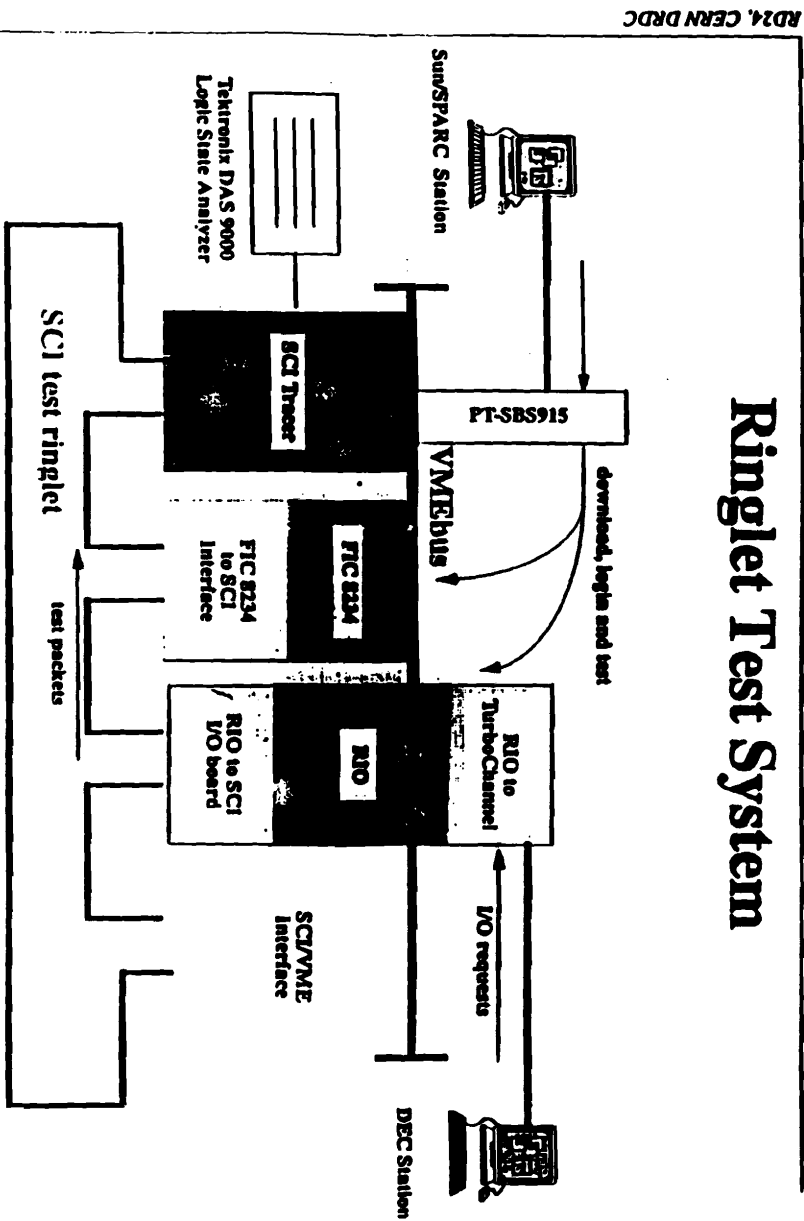
---

1. joint spokesmen  
2. SINTEF, OSLO, NORWAY

# RD24/CERN-Protvino (1993)

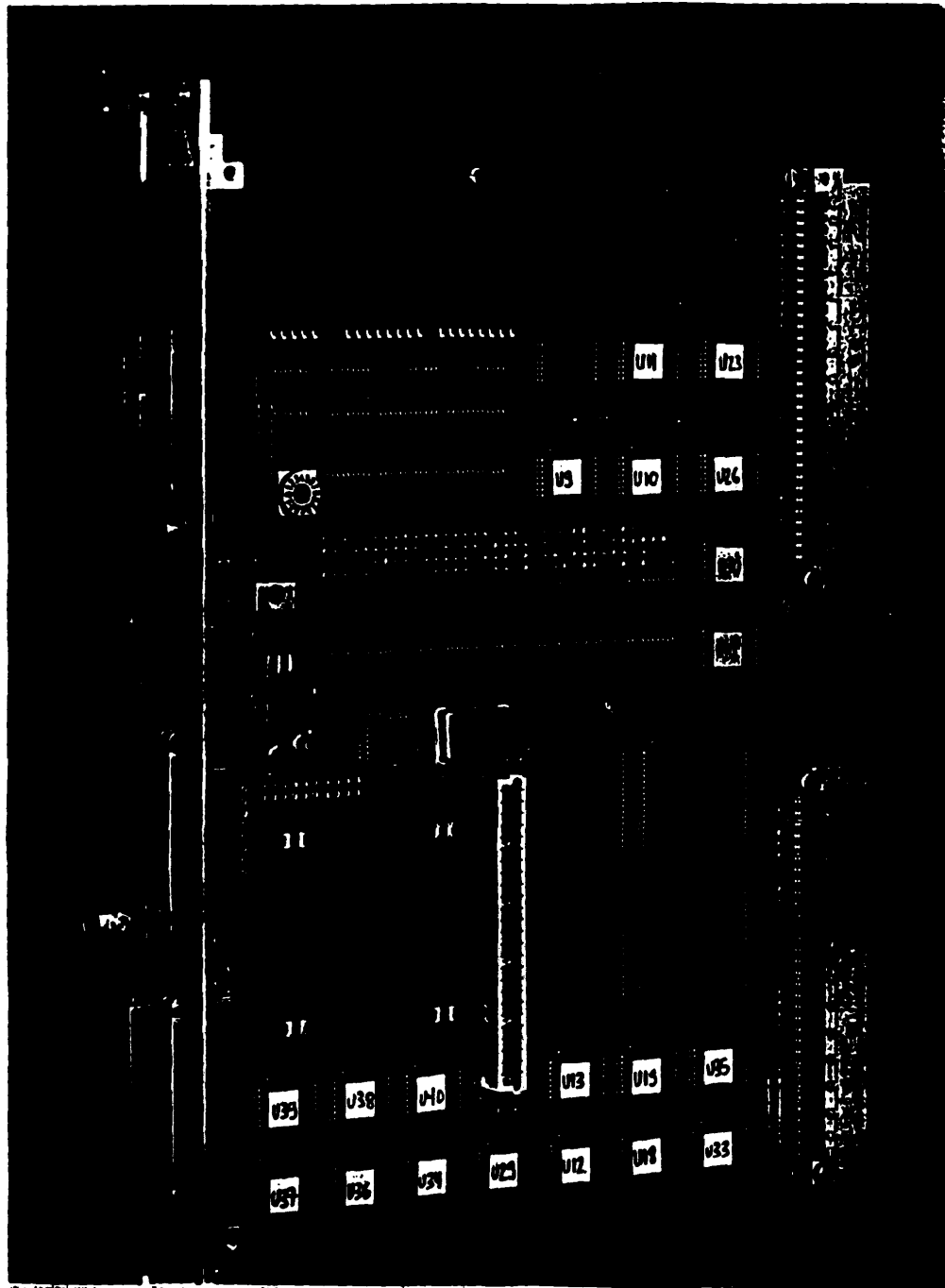


# Ringlet Test System

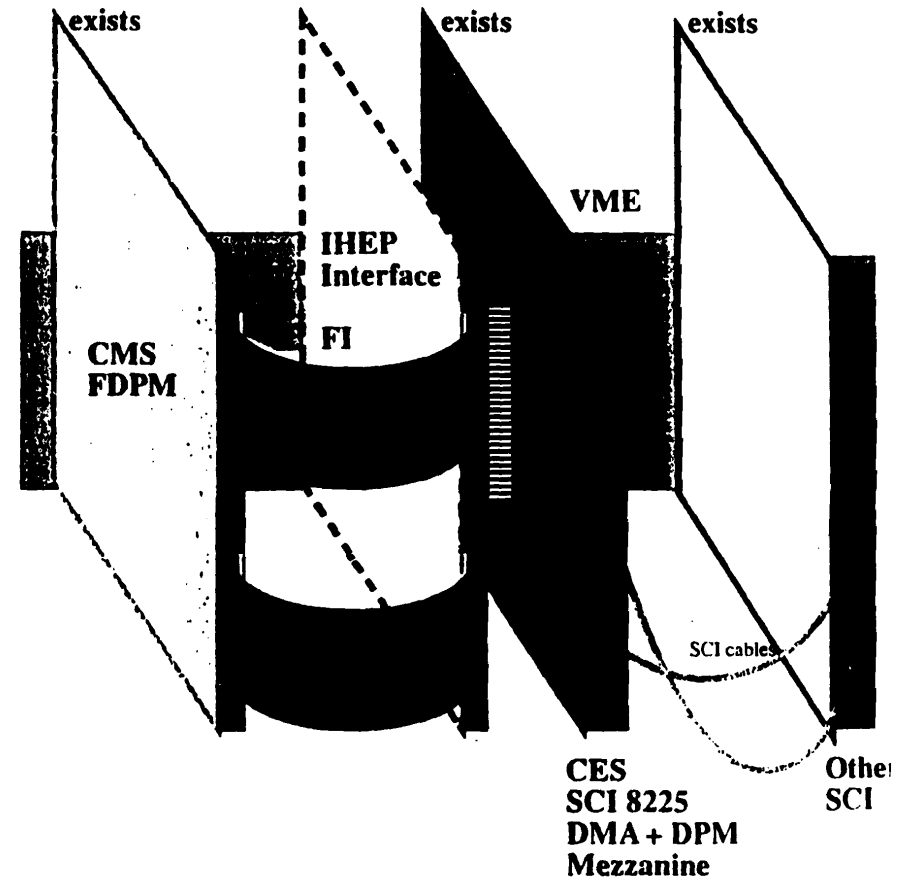


RD24, CERN DRDC

26 May 1993



SCI port for FDPM ( IHEP Protvino, Russia)



EI: Fast Dual Port SCI Interface to FDPM ( RD24/IHEP)

Registers: byte count, SCI address

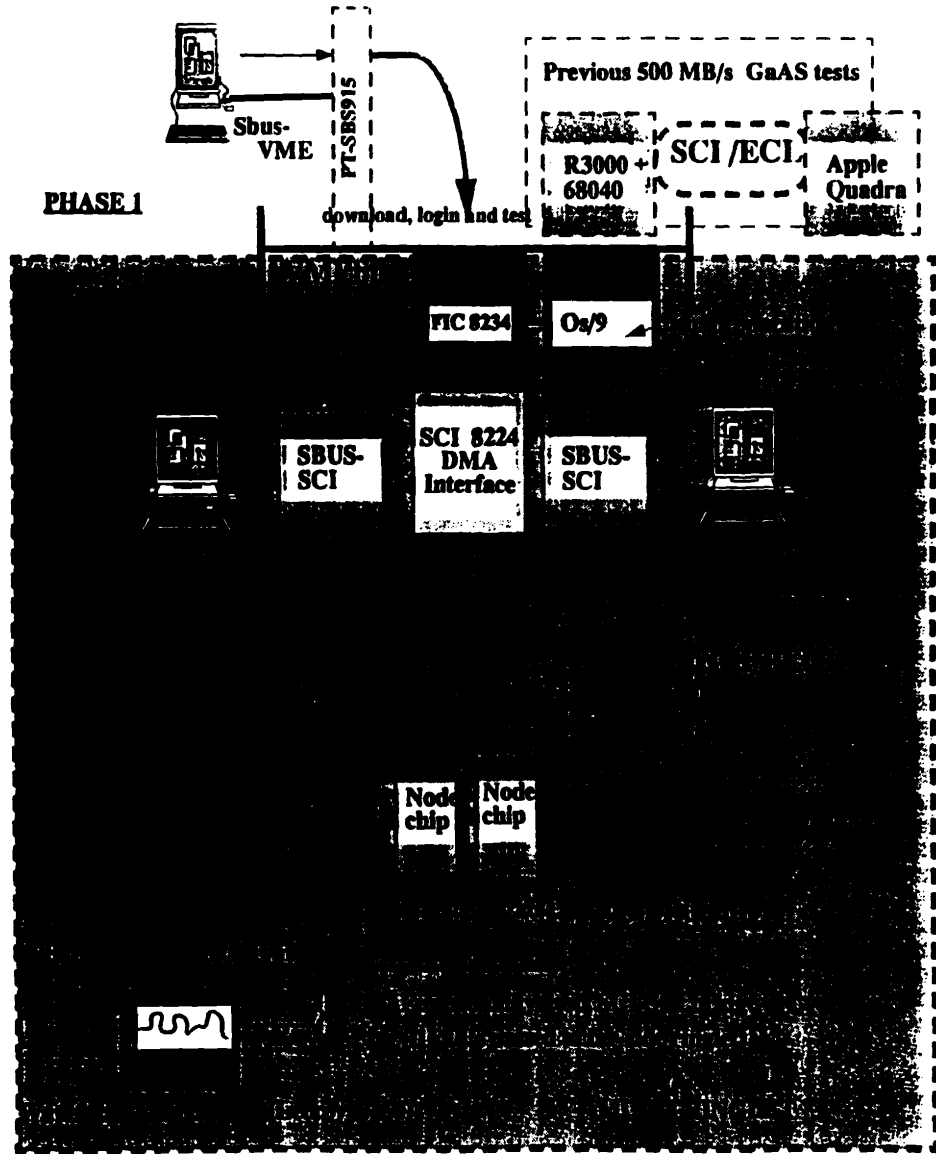
Commands: DMA Start-Stop

Transfer modes: LOAD FDPM->DPM

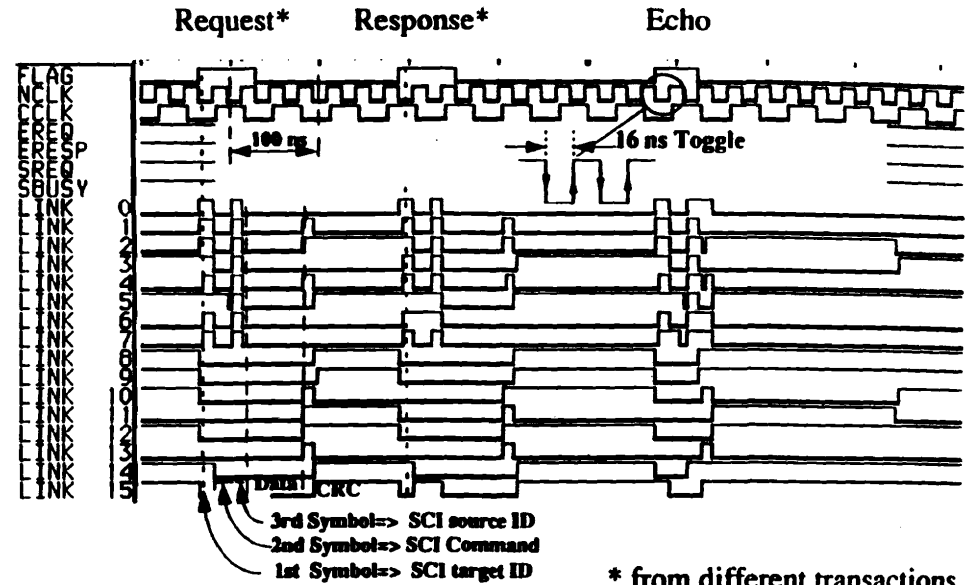
Transfer DPM-> SCI

Design Tools: Cadence Concept, Verilog at CERN + IHEP

### RD24's SCI laboratory



### Footprints of first SCI packets on CMOS nodechips (April 94)

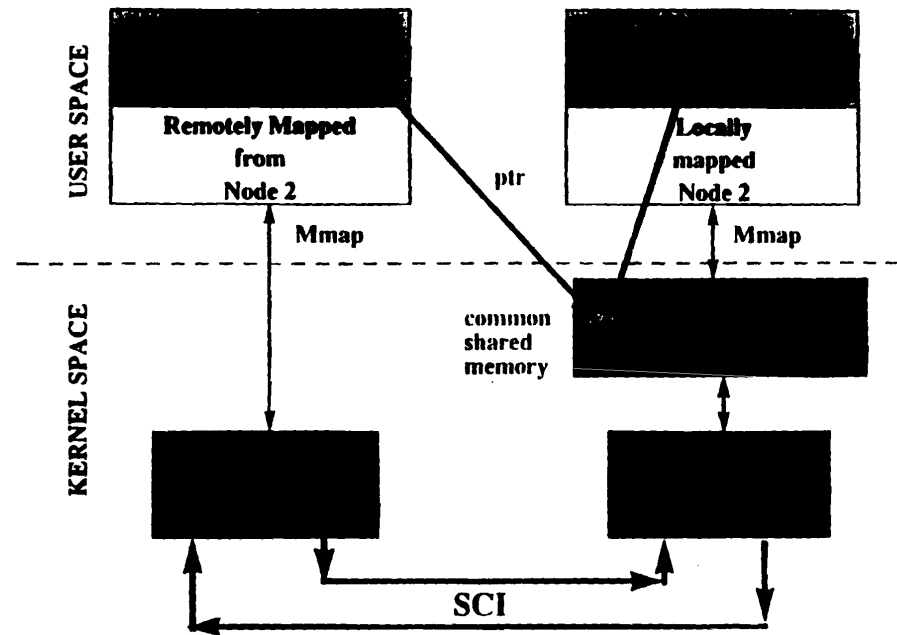


### RD24 Laboratory performance on L64601 systems

- 20 Mbytes on hardware DMA (29 max achievable)
- 10 Mbytes on DMA using Unix driver
- 20 Mbytes in transparent memory copy loop (4byte)
- 10 Mbytes in 8 byte operation

3 node SCI ringlet test  
in north area ATLAS test beam

## SHARED MEMORY



Example is applicable to N nodes and any distance

## TRANSPARENT ACCESS

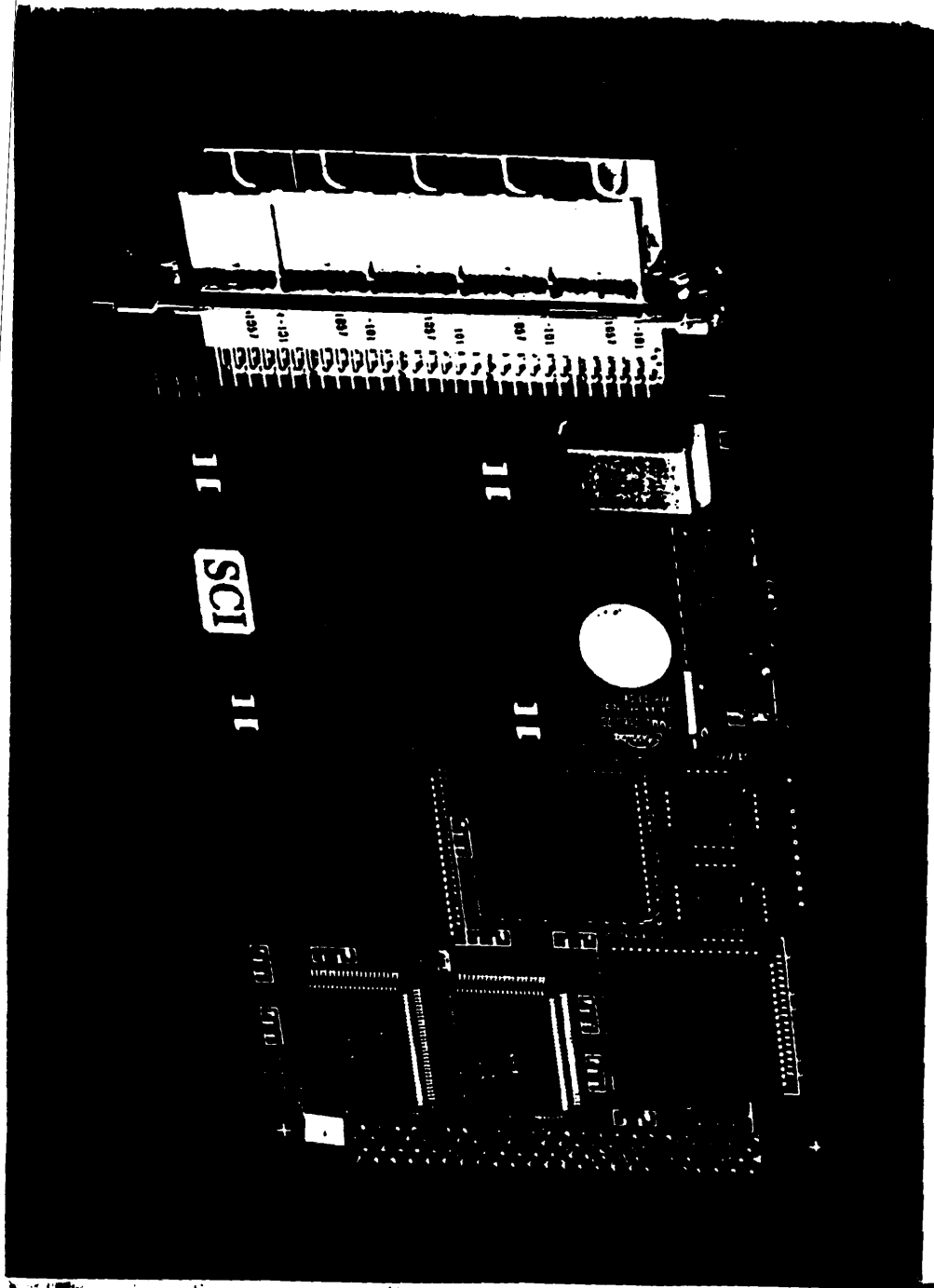
### NODE 1

```
fd = open("/dev/mem", O_RDWR | MAP_SHARED);
ioctl(fd, _IOWR(0, 0), 0);
ptr = mmap(0, 1024);
write(ptr, "1234");
close(fd);
```

### NODE 2

```
fd = open("/dev/mem", O_RDWR | MAP_SHARED);
ioctl(fd, _IOWR(0, 0), 0);
ptr = mmap(0, 1024);
read(ptr, "1234", 4);
close(fd);
```

Tested OK by RD24. Shared memory at 1 Mbyte/s data rate and 10 Mbytes w DMA



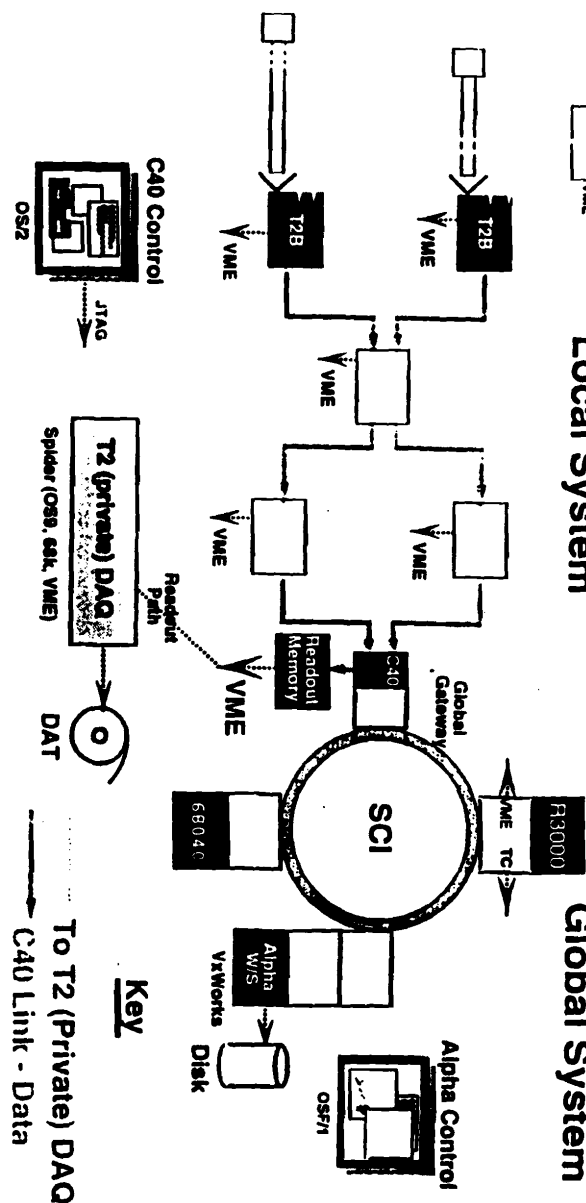
RDE  
Electronics



Test Beam Setup

Local System

Global System



RD6 & Atlas DAQ

T2B = T2 Buffer (C40)

LINK = Concentrate & Mesh (C40)

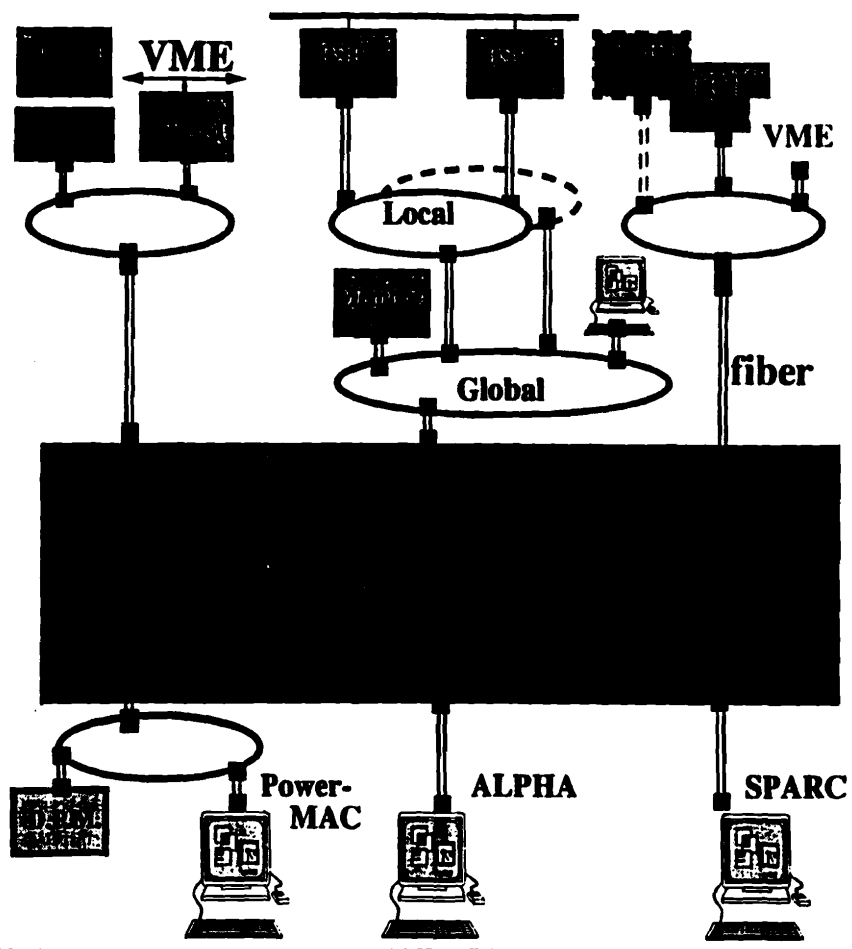
FEX = Feature Extractor (C40)

General Bus In

**Next step: heterogeneous mini SCI DAQ system**

(real implementation depends on component availability)

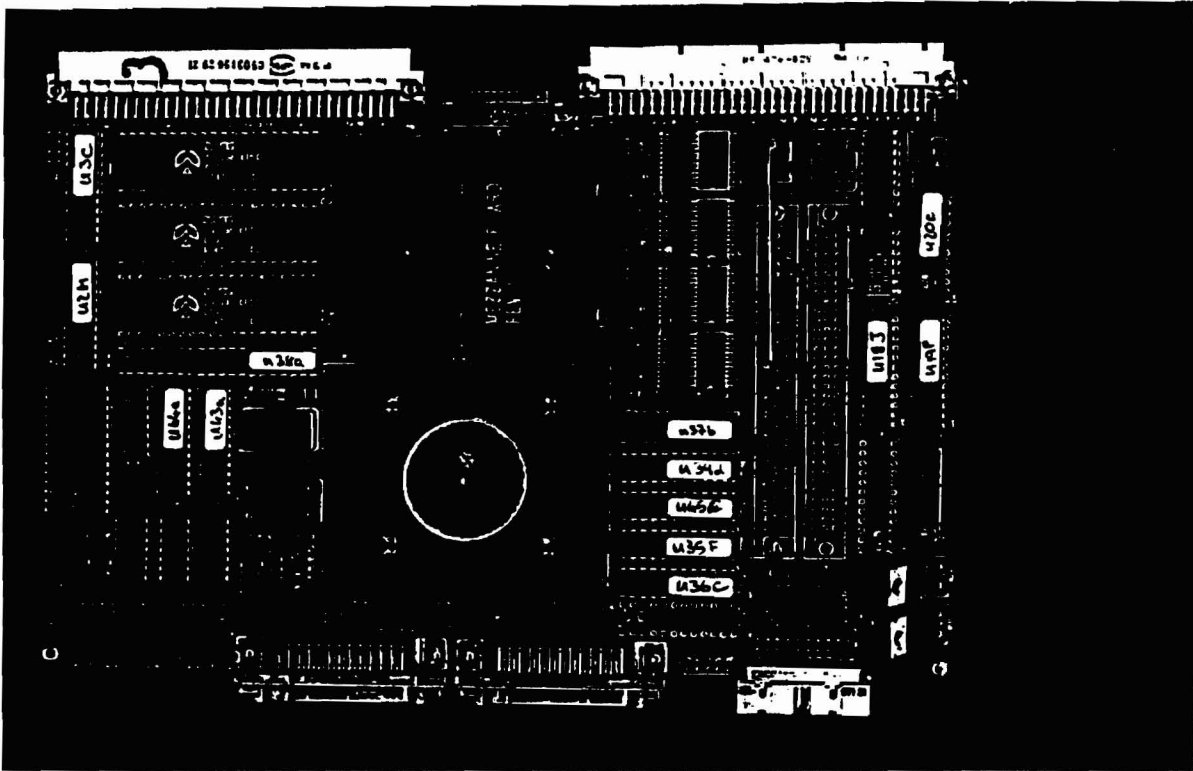
RISC Farm Event Builder Switch Links Front End buses + Buffers



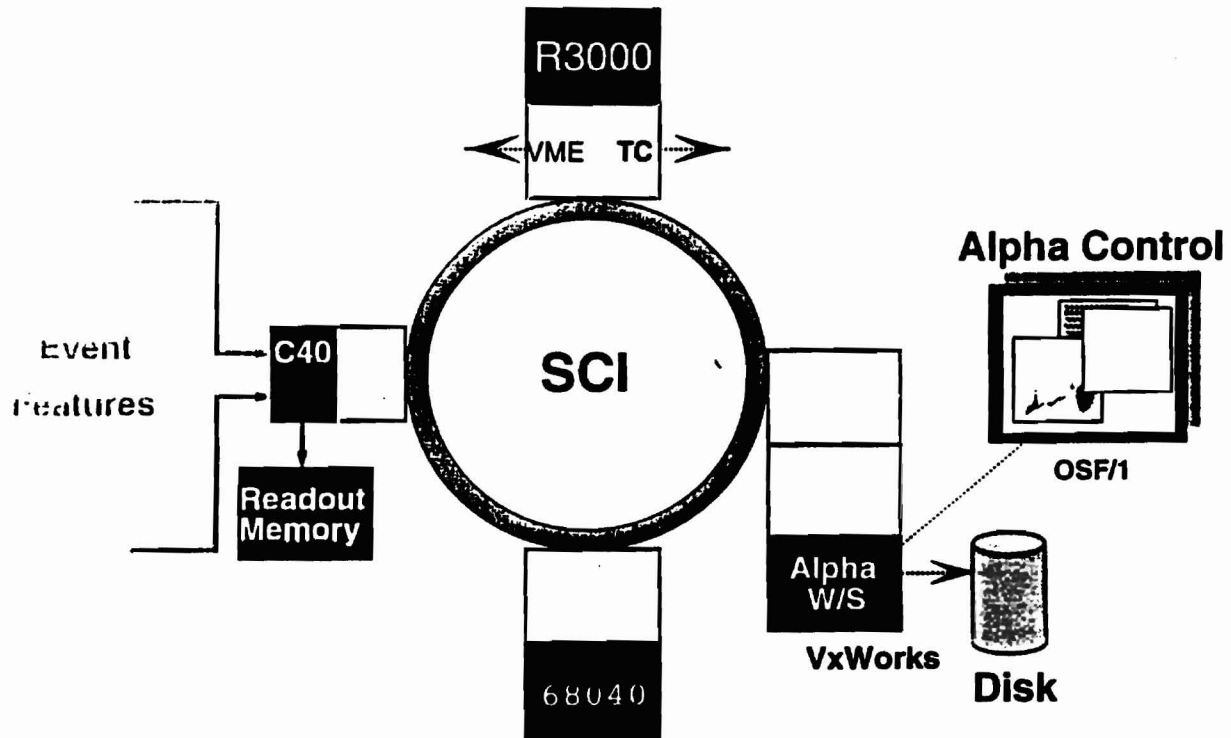
**CMS like:**  
data driven event memories +  
high speed event builder to  
massive RISC farm  
test transparent CPU access to  
data over EB switch

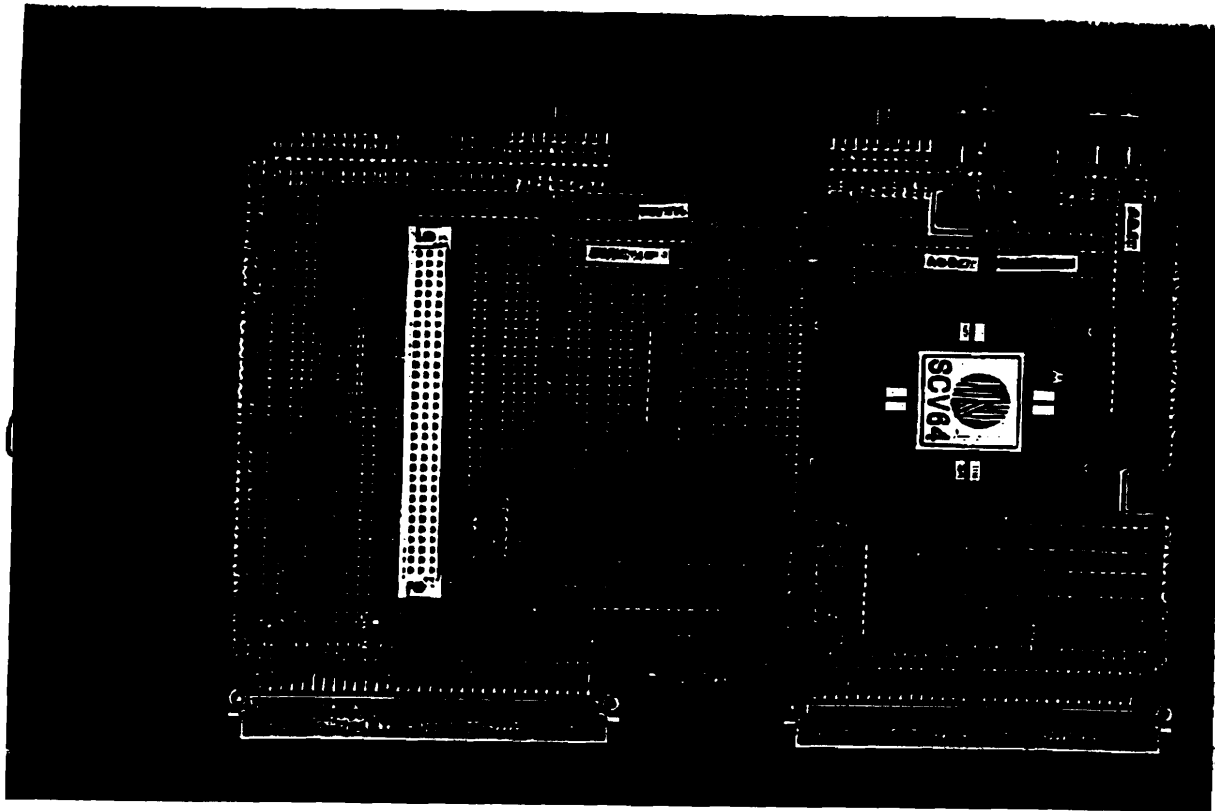
**ATLAS like:**  
decomposed local and  
global data network  
with  
intermediate processors  
before EB

**ALICE like:**  
data  
concentrator of mixed FE  
buses,  
passive EB

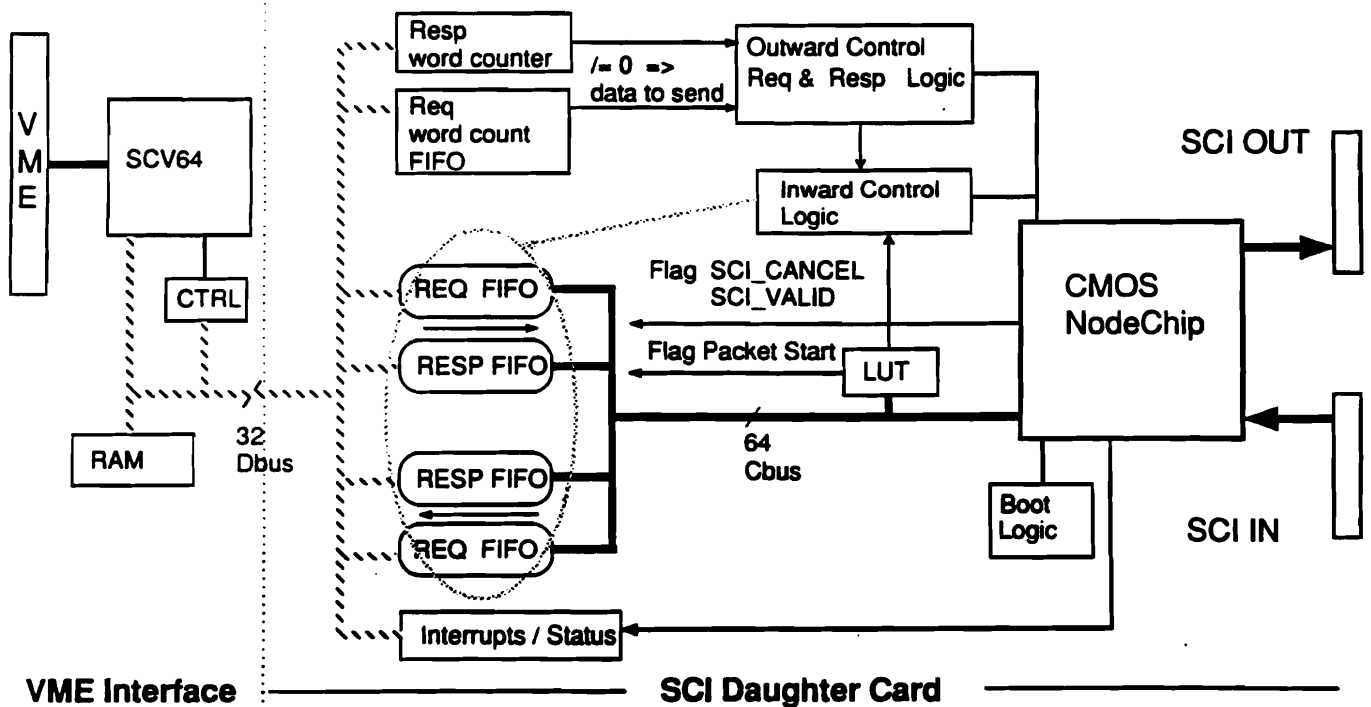


# Prototyping for ATLAS T2 Global System





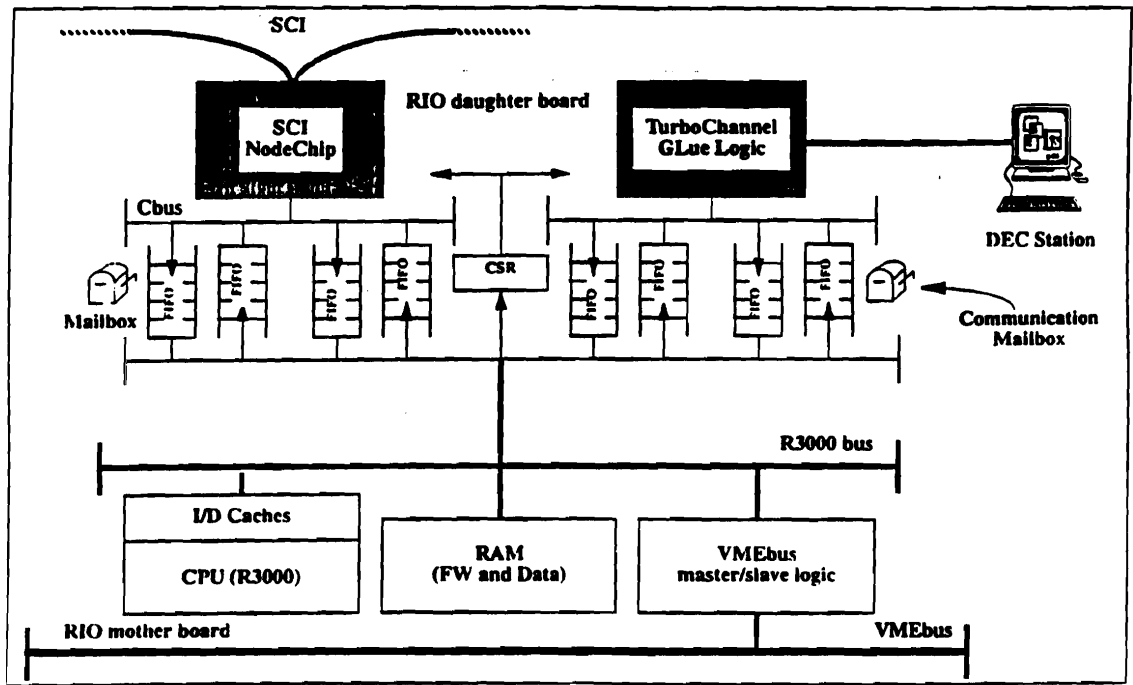
Block Diagram of the SCI Daughter Card and VME Interface



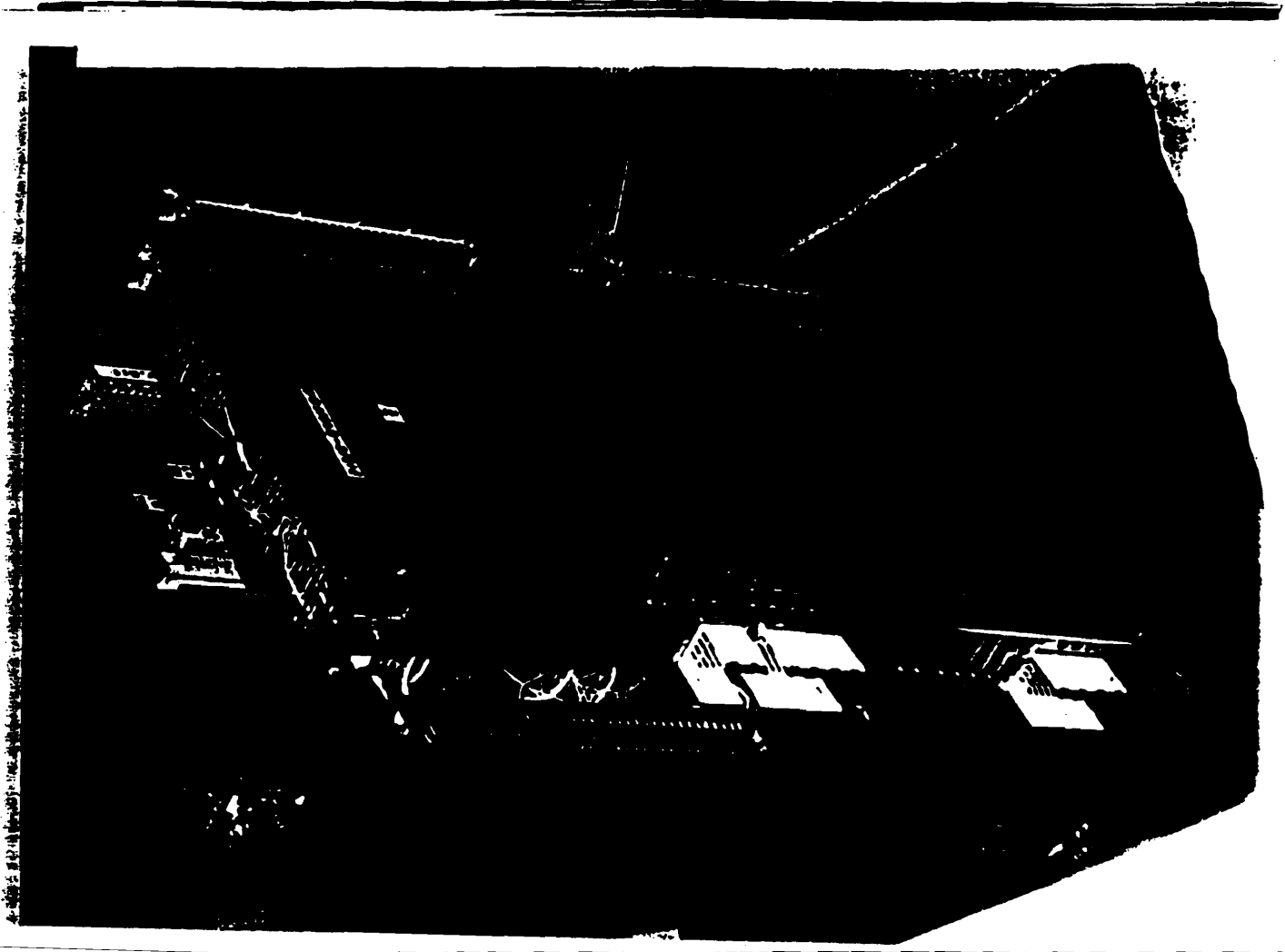


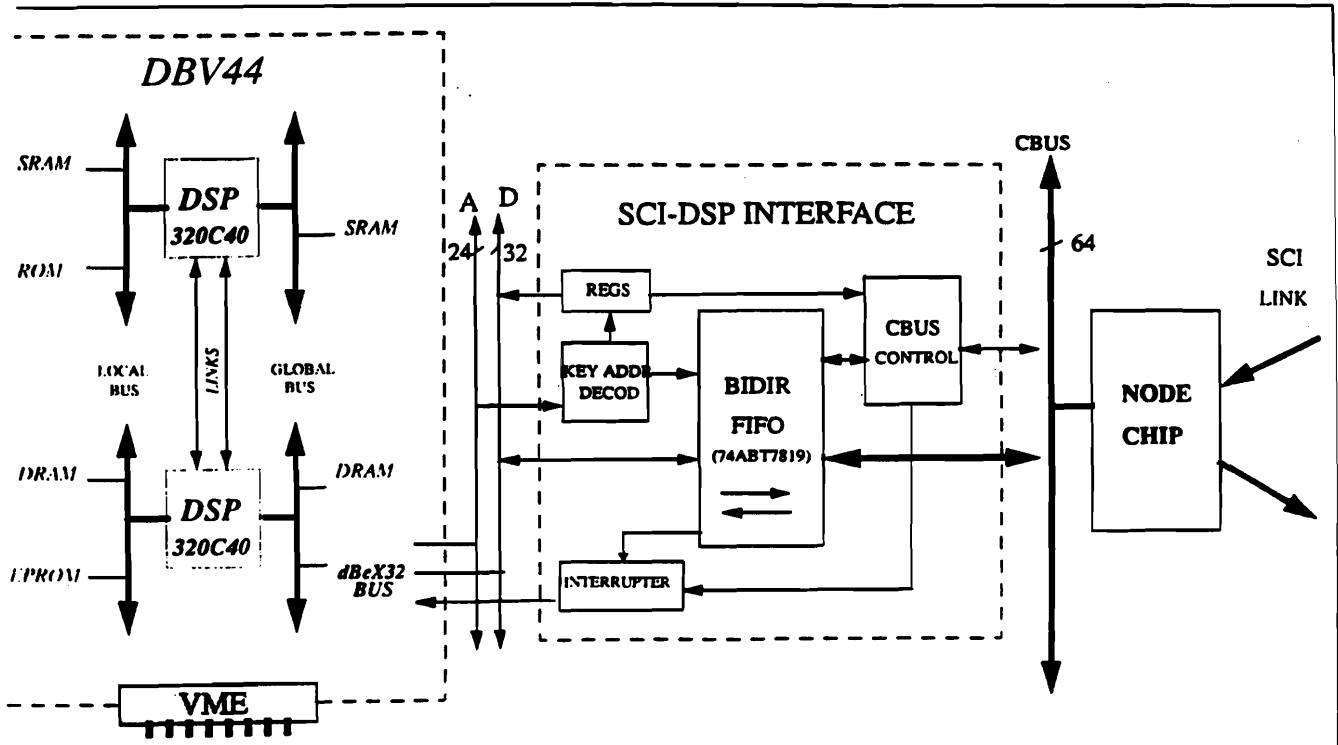
# MIPS R3000 Interface

RD34, CERN DRDC



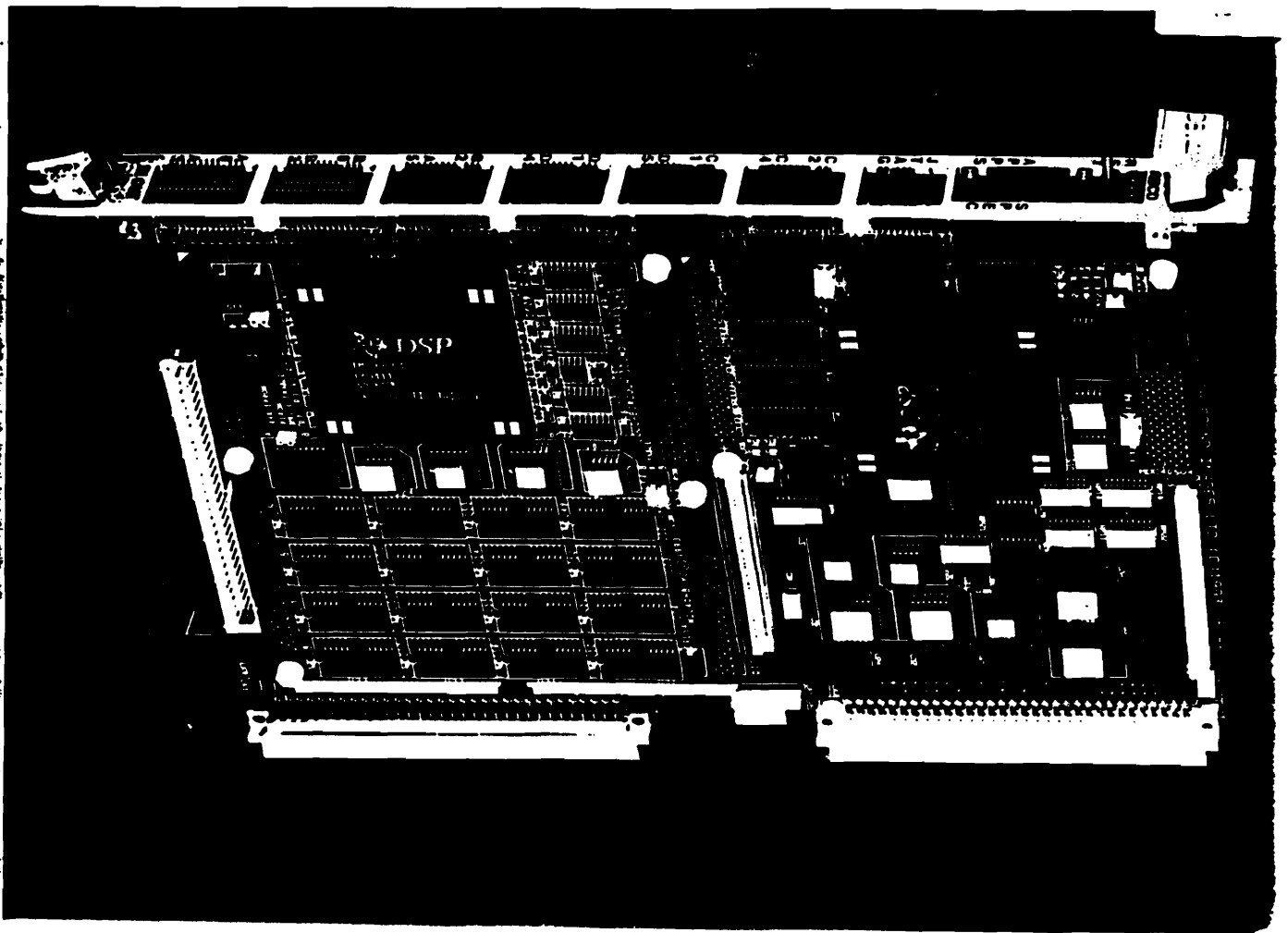
26 May 1993





*Allows Outgoing & Incoming Requests/Responses (Moves, Writes, Reads...)*

*R024 C40-SCI Interface*





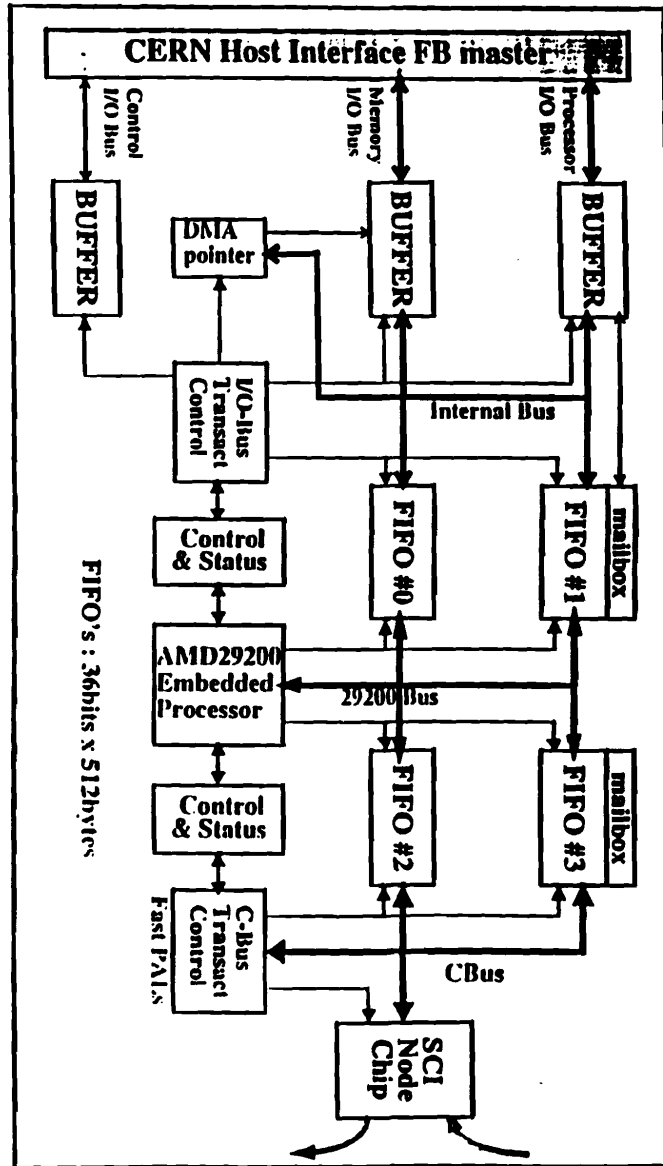
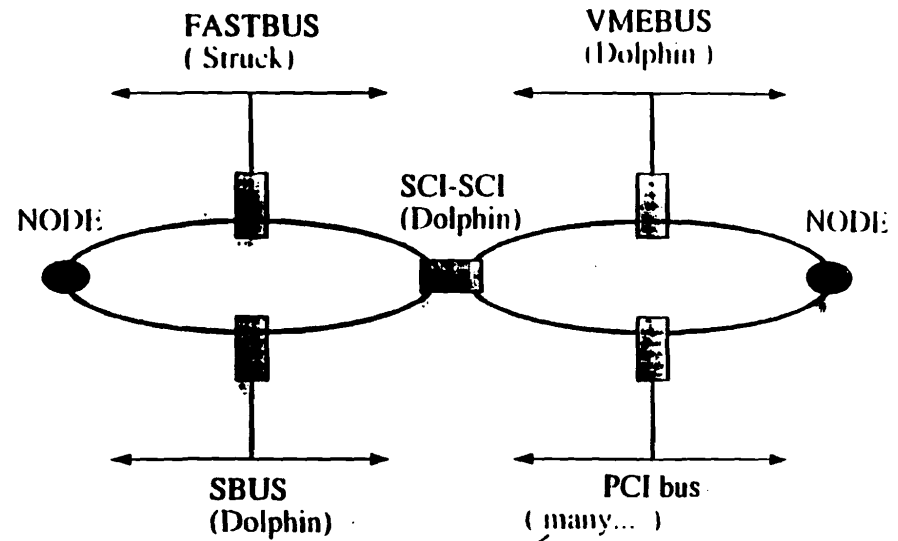


Figure 1: Block Diagram CH1/SCI link

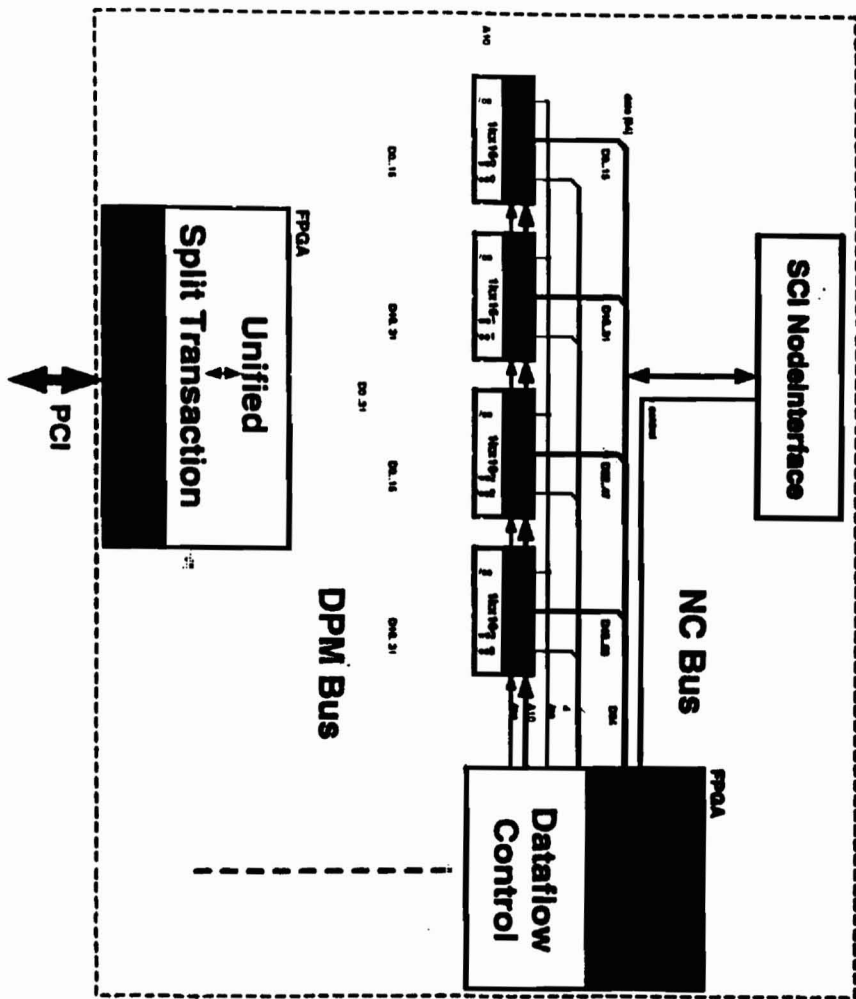
## SCI BRIDGES



RD24, Star, CES, Dolphin, BiRa, Apple...  
RD24 will try to coordinate at FNAL.  
DAQ conference

**SBUS-SCI: most advanced SCI development kit from Dolphin, includes Unix drivers. RD24 shared memory tests on Sparcstations**

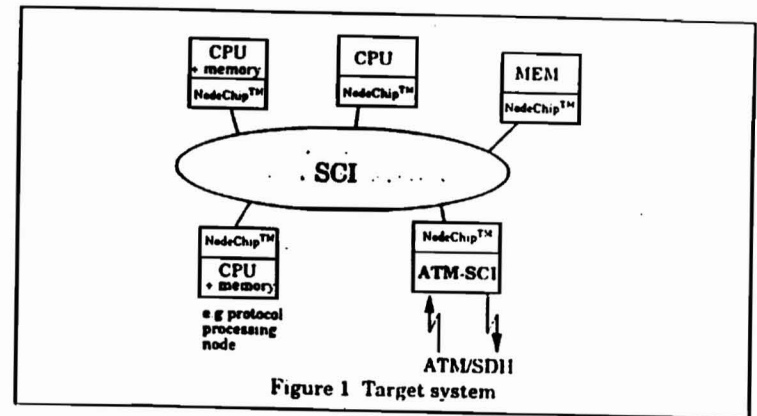
**VMEbus: Expected early 95 uses Cypress VIC chip and LSI Logic nodechip**



## The ATM-SCI: An ATM network interface for SCI-based multi-computer systems

Øivind Kure, Norwegian Telecom Research

The ATM-SCI is designed to provide ATM access for an SCI based multi-computer system. The interface makes it feasible to use SCI as a cluster interconnect and at the same time maintain ATM connectivity.

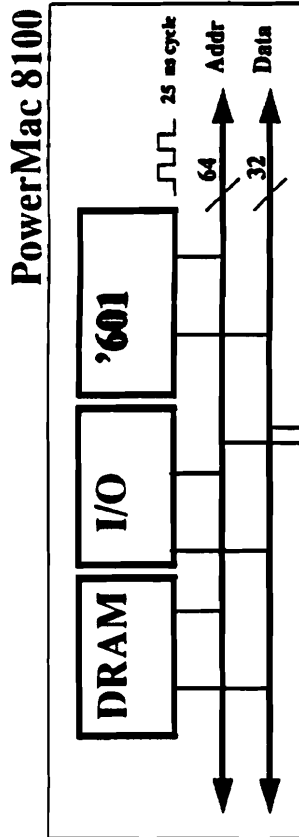


The premier advantage of the interface is the flexibility and the fine granularity of the control of the data streams. It is designed to support protocol stacks with multiplexing at the lowest level. Each ATM channel is mapped to a separate buffer area. This ensures no interference between the data streams, and processor and buffer resources can be assigned depending on the service requirements of the data stream. An extension of this mechanism is to map the ATM channels directly to special devices on the SCI interconnect.

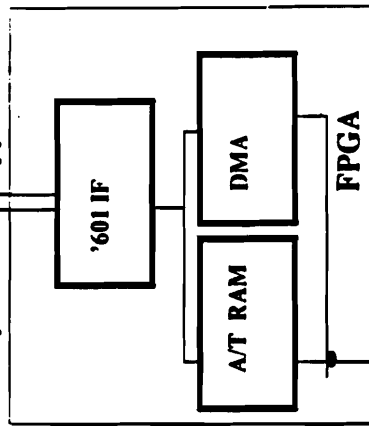
The ATM-SCI can either read or transfer data directly to the host (DMA) or let the hosts perform the transfer (programmed I/O). In order to meet the requirements from different applications, the mode, DMA or programmed I/O, is chosen on a per channel basis.

The interface uses two embedded Spare processors for the management of the communication streams, one for the transmit path and one for the receive path. The computational power of the embedded CPUs allows for additional functionality to be added to the interface, if necessary.

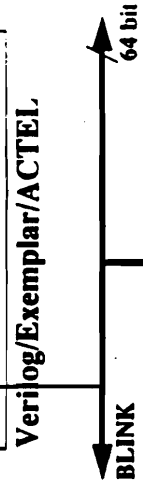
# Processor Interface Apple/RD24



Transparent IF Apple ATG



Map '601 to equivalent SCI transactions



CMOS Linc Controller

LC

LC: A new SCI chip factor 3 improved

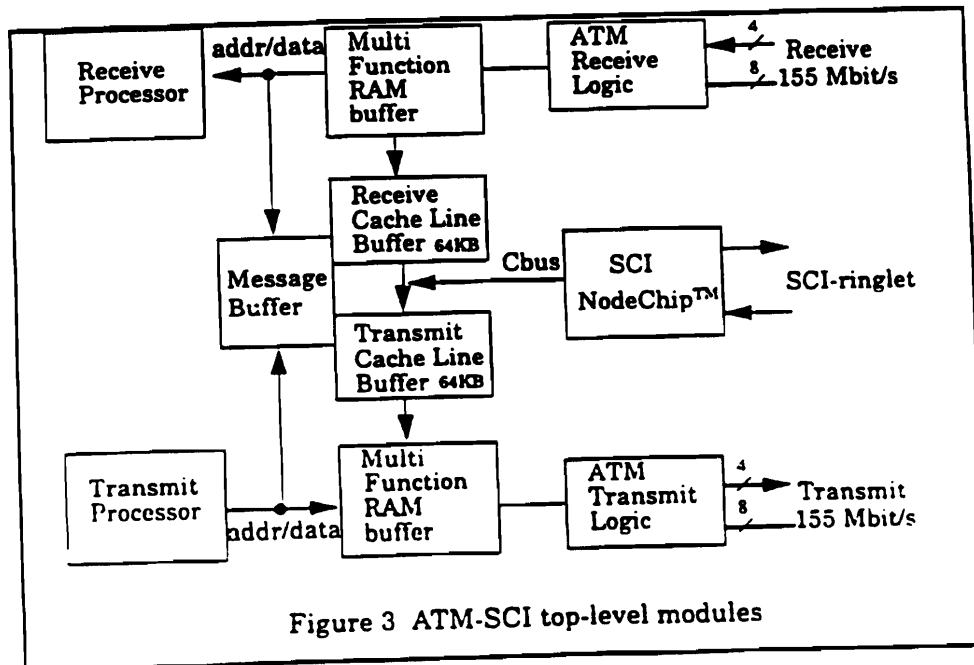


Figure 3 ATM-SCI top-level modules

## SCI Components + Products

**Sbus-SCI transparent card for SPARC Workstations (Dolphin)**

**DMA-SCI adapter from VMEbus 68040 processor (CES)**

**SCI-SCI bridge (Dolphin)**

**SCSI-like SCI cables and Extension boxes (Dolphin + CES)**  
Commercial NOW

**Exemplar Multiprocessor MPP, CONVEX** Now

**Fiber Optics SCI adapter using Lasertron, Finisar, BT&D modules**

**ATM over SCI Interface (Dolphin + Norwegian Telecom)**

**VME-SCI bridge (Dolphin)** Summer 1994

**PowerMac-SCI bridge card (Apple ATG)**

**SCI-PCI local bus bridge (Dolphin, STAR)**

**Fastbus Interface (Univ. Oslo, Struck)** End 1994

## RD24 Research Projects

**C40-SCI Interface (Univ. Manchester + RAL)**

**ALPHA-PCI-SCI adapter (Univ. Manchester, DEC)**

**TurboChannel Interface (RAL + INFN + DEC)**

**16 Mbyte VME DRAM node (CIEMAT Madrid)**

**Cache coherent Videoram memory (Univ. Oslo Physics)**

**Long distance shared memory over fiber optics (CERN SL)**

**SCI DMA component for 32/64 bit buses (CAE design at IHEP)**



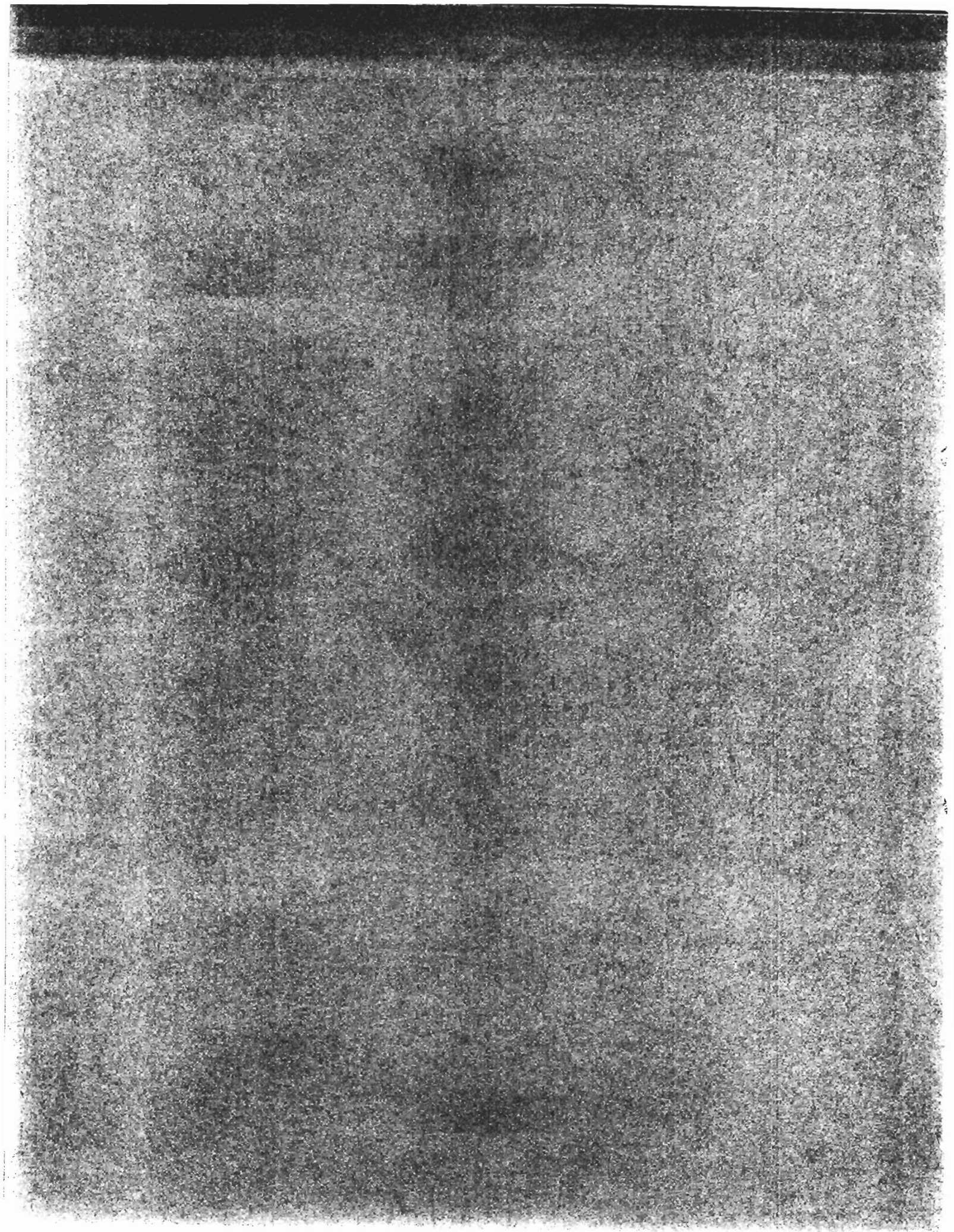


**S3-4**

**"Other Research Projects"**

**(Masa Nomachi - KEK)**

Review of some past and present R&D on switching networks and point to point data links for data acquisition applications such as the past Scalable DAQ project and current switching network test bed project at Fermilab, Fibre Channel R&D at LBL & in-house switching network R&D at KEK and Fermilab.



## Other Research Projects (Fermilab/LBL/KEK)

M.Nomachi  
National Laboratory for High Energy Physics

### Introduction

A functionality of an event builder is to compose a complete event data from event fragments coming from many front-end sub systems. A switch type event builder has been proposed as an extension of classical "N" to "one" event builder. The research projects we have been working are based on this architecture.

A classical event builder collects data from several front-end subsystems through data links (figure 1). In order to avoid a bottle-neck on the event composer node, parallel path is introduced.(figure 2) We are standing on this architecture.

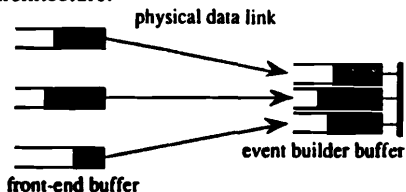


figure 1

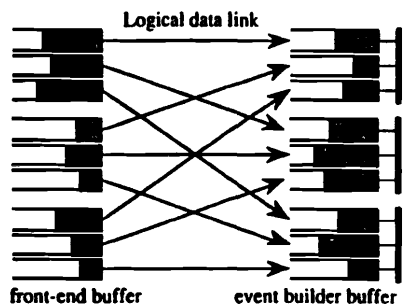


figure 2

Each physical data link is shared by several data links. The number of connections between front-end to event composer node increases very rapidly as a function of number of nodes. It will be very difficult to have

such a large number of physical data link. A switch type event builder is very efficient for large size system.

### Fermilab SPOA project

Fermilab scalable open architecture data acquisition system (SPOA) project is a pioneer work on the development of switch type event builder. [1] An 8 by 8 event builder was demonstrated. Each data source can transmit the 20MB/sec data.

An event builder architecture which fermilab introduced is based on logical permanent data link architecture. Each physical data link is shared by several logical data link by time sharing. The key component on this architecture is the module which handle this time sharing. A time slot interchange (TSI) module was developed.(figure 3) Each logical data link has constant time slot for physical data transmission. A packet of data which is transferred during a time slot is constant. A packet boundary can be free from an event boundary. Logically, data is transferred continuously. Therefore, it is not necessary to be taken care the packet boundary.

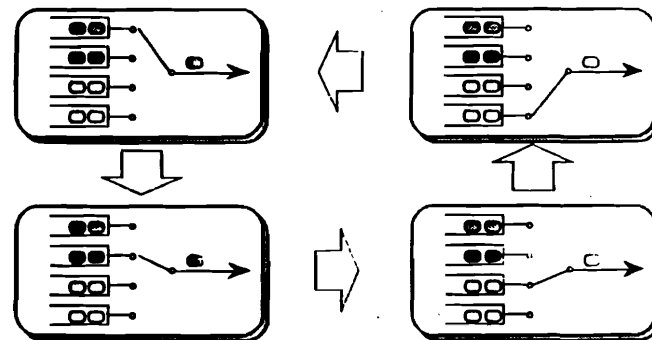


figure 3

A switch is operated as a barrel shifter mode. It is necessary to synchronize the switching for all physical data link. In order to reduce the switching overhead, FIFOs are placed before and after the synchronous switch. Switching frequency of 20KHz is achieved.

The demonstration system shows very good scalability up to 160MB/sec throughput with eight 20MB/sec data link.[2] Memory usage is very low for the event size of 200KB and packet size of 1KB. The measured usage of 1MB memory is shown in figure 4.

SPOADAQ project has successfully demonstrated a switch type event builder. It presented permanent data link architecture.

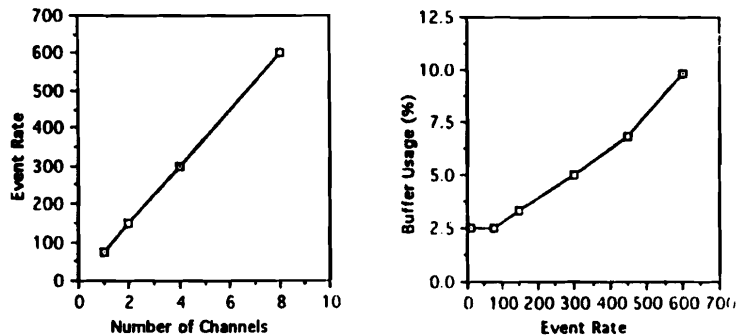


figure 4

#### SDC event builder

An event builder R&D for the SDC data acquisition system has been done by Fermilab/LBL/KEK and other Universities.[3] Based on the experience on the Fermilab's R&D, we have studied the possibility of using a commercial network interface as an Input/Output TSI module. Fiber channel interface has been mainly tested

Details of Fiber channel is described by many references [4]. Fiber channel is good for circuit switch. It has very high performance for high speed data transfer but maximum switching frequency may not be so high compared to ATM [5]. In order to obtain high throughput, a large packet size may be necessary. It requires large amount of memory and causes larger latency.

LBL proposed two stage event builder architecture.[6] It will reduce the required amount of memory and the latency. However, the cost of intermediate nodes must be taken into account.

LBL proposed a R&D project to test the Fiber channel based event builder. A functionality of TSI is cared by a commercial CPU board. Data movement is done by a hardware DMA which is controlled by the CPU.

#### Data link R&D

High speed serial data link has been studied for SDC data acquisition system. A speed of 1Gbps is required for the following applications. 1) Trigger signal readout/distribution. 2) Data readout data link. 3) Event builder data link. We have been tested a G-link chip set from Hewlett-Packard.[7] It is developed for serial HIPPI. We have also tested cheap electric to optical and optical to electric devices. Giga-bit data link could be used in our applications.

LBL developed a bit-error tester for testing the data link. KEK and Fermilab collaborated for testing the Re-lock time measurement of the G-

link chip set.[8] Re-lock time less than 30  $\mu$ sec will be short enough for event builder data link.

#### KEK transparent switch

KEK has studied an event builder based on the permanent logical data link architecture.[9] As one of the conclusion of our R&D works, KEK is proposing a global traffic control system. A traffic of data in our application is very coherent traffic. Most efficient control system for such a coherent traffic is a global traffic control system. Fermilab's SPOA system is a sort of global traffic control system. The details are shown in the reference.[10]

Traffic of data is controlled by traffic control signal. Switch is not necessary to care the contents of the data to configure the switch connection. Therefore, passive switch can be used. KEK has developed a high speed ECL switch.[11] It can handle up to 3Gbps signal.

Analytical calculation was done for the global traffic control system. There is no contention if the traffic is controlled properly. The traffic on each logical data link can be independent from the others. It makes the analysis simple. The results are shown in the reference 12. One of important results is queue length calculation as a function of the packet size. Figure 5 shows number of event fragment in the input queue as a function packet size. The calculation is done for several traffic intensities (k).

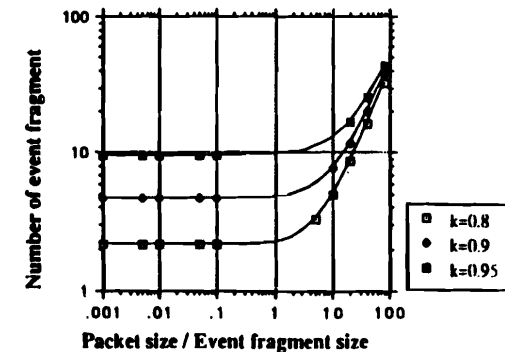
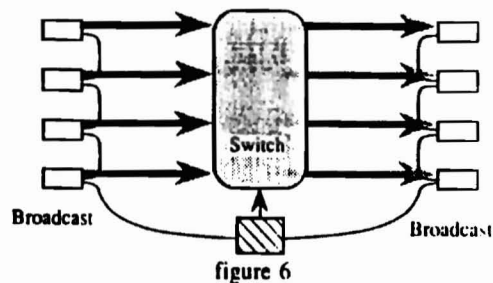


figure 5

It shows that a smaller packet is better but it is not necessary to be less than the event fragment size. Or, in other words, switching frequency is not necessary to be higher than the event rate. Our target experiments have a few KHz event rate. Therefore a switching frequency of 1KHz will be high

enough. We may be able to handle 1KHz with a commercial processor and 30  $\mu$ sec of re-lock time can be negligible compared to the switching interval.

We have proposed the event builder based on these R&D works. (figure 6) In order to handle high speed data transfer, VME-bus is not good enough. We propose to use high speed dual port memory on commercial CPU boards. It may reduce the R&D costs. We may have to develop the interface module on local buses such as PCI instead of on VME.



### **Conclusion**

We have studied event builder based on permanent logical data link architecture. Fermilab demonstrated switch type event builder successfully. As a result of our studies, KEK is proposing a global traffic control system. It will be an efficient and simple flow control system on permanent logical data link architecture.

### **References**

- [1] Ed. Barsotti et al., A Proposed Scalable Parallel Open Architecture Data Acquisition System for Low to High Rate Experiments, Test Beam and All SSC Detectors. IEEE Trans. NS, NS-37 No3.(1990)
- [2] D.Black, M.Bawden et al., Results From a Data Acquisition System Prototype Project Using a Switch-Based Event Builder.1991 IEEE Nucl. Science Symposium
- [3] Solenoidal Detector Collaboration, Technical Design Report. April 1992, SDC-92-201,SSCL-SR-1215
- [4] R.Cummings,Fibre Channel. International Data Acquisition Conference. October 1994.
- [5] J.Y.LeBoudec.ATM/SONET. International Data Acquisition Conference, October 1994.
- [6] B.Greiman, A scalable Fibre Channel Architecture for Event Building. International Data Acquisition Conference. October 1994.

- [7] Hewlett-Packard,Gigabit Rate Transmit Receive Chip Set Technical Data
- [8] O.Sasaki and J.Andresen, Testing Of Re-Lock Time Of HP G-link Chip Set, SDC-93-598, KEK-93-145.
- [9] M.Nomachi, O.Sasaki, H.Fujii, and T.K.Ohsaka,A large scale Switch Type Event Builder,CHEP92,188.
- [10] M.Tairadate et.al., Global Traffic Control System on High Speed Event Builder using Transparent Switches, International Data Acquisition Conference, October 1994.
- [11] O.Sasaki et al., A VME Barrel Shifter System for Event Reconstruction from Up to 3Gbps Signal Trains, IEEE Trans. NS, NS-40 (1993) 603.
- [12] M.Nomachi, Event Builder Queue Occupancy, SDC-93-566

# Other Research Projects (Fermilab/LBL/KEK)

KEK  
NOMACHI. Masaharu

Goal of this presentation

Re-view the R&D works

contents

Introduction

Fermilab : scalable DAQ

R&D for SDC event builder

LBL/KEK/Fermilab

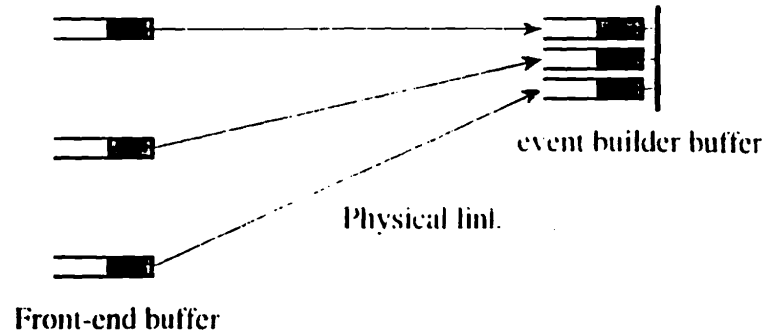
KEK : global traffic control system

Summary

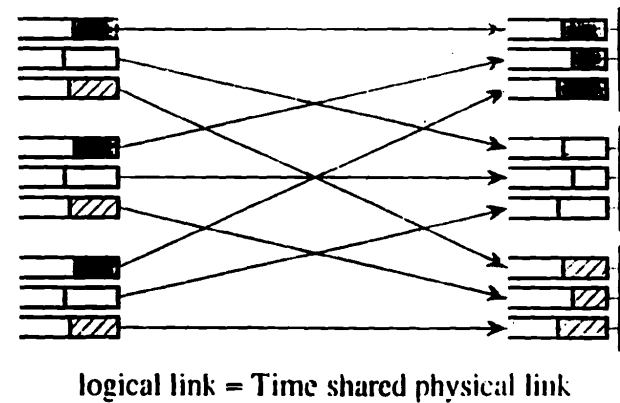
## Event builder

model : logical permanent link model

## Classical event builder



## Parallel event builder



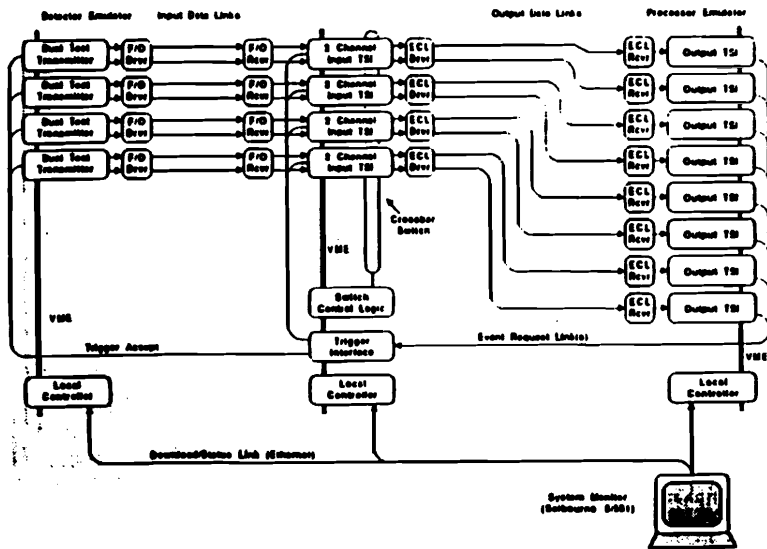
**Fermilab**  
**Scalable Parallel Open Architecture**  
**Data Acquisition System**

A pioneer work on the development of switch type event builder

A switch type event builder is demonstrated with barrel shifter.

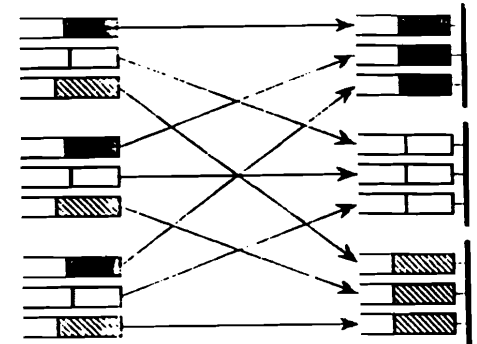
8 x 8 barrel shifting mode  
 20MB/sec data link

A study on various switching networks  
 The Verilog simulator was used



daqconf 26-oct-94, M.Nomachi

Model

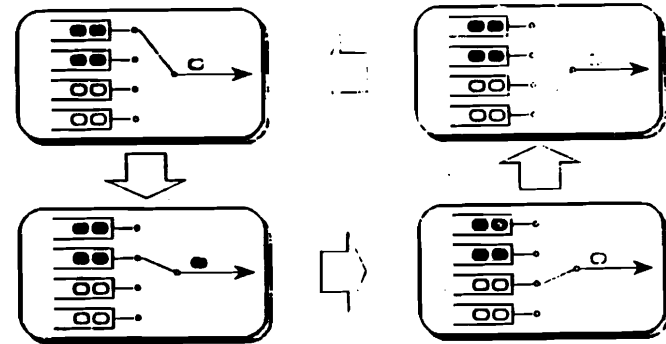


Logical permanent link

A model for traffic shaping/controlling

TSI (Time Slot Interchange)

Time sharing of the physical link



Each setup has a constant "time slot"

Fixed packet size.  
 Packet boundary is free from event boundary.

daqconf 26-oct-94, M.Nomachi

## Time Multiplexed Switch

Crossbar switch

used as a barrel shifter

8 bit wide synchronous switch

Back-plane switch

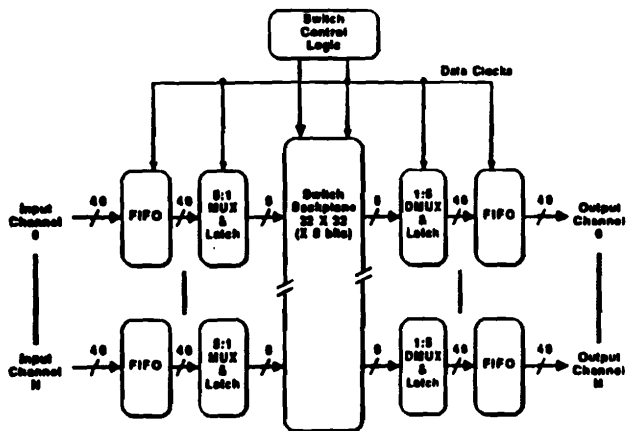
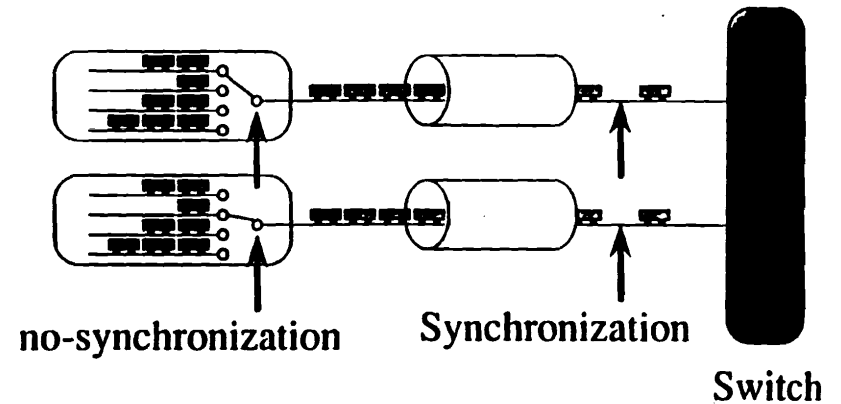


Figure 22 Time Multiplexed Switch

## Flow control

Predetermined order

Broadcast the signal for synchronization



## pipe-lined synchronization

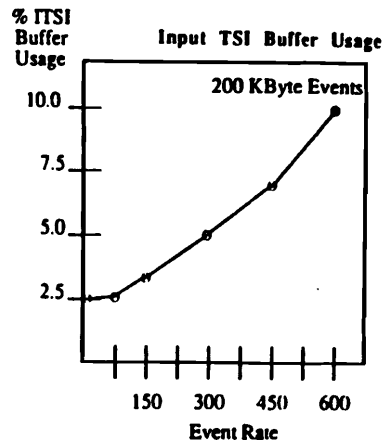
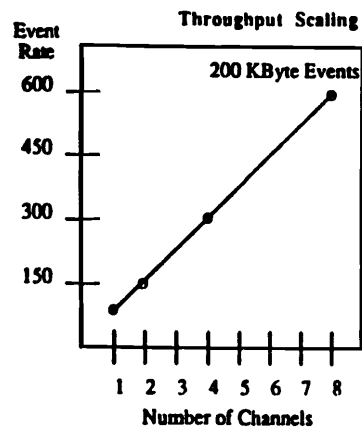
Reduce the switching over head

Switching frequency  
Minimum event size

20 KHz  
1 KB



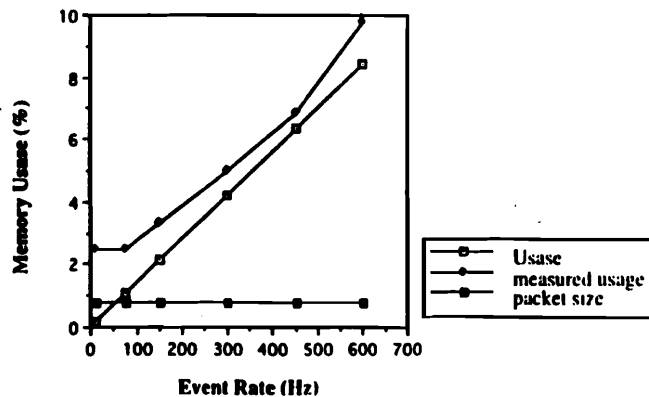
## Results



160 MB/sec throughput  
Good scalability

Event size = 200KB  
Link speed = 20MB/sec  
Packet size = 1KB

## Analysis on single queue (SDC note 93-566)



Global traffic control system

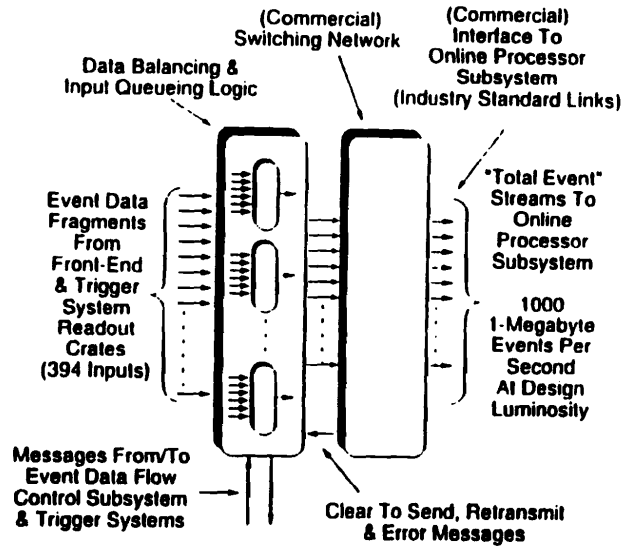
## Summary of Fermilab SPOA DAQ

A pioneer work on switch type event builder

Logical permanent link model  
Idea of time slot interchange

Functionarity of TSI is very complicated.  
possibility of using a commercial module

# SDC event builder



## Baseline design

commercial switching network

## possible developments

HIPPI

FC

ATM (too early to use for SDC)

Fiberchannel R&D has been done at LBL and Fermilab/KEK.

# Fiber channel R&D (at LBL)

## Fiber channel

1 Gbps / 250 Mbps

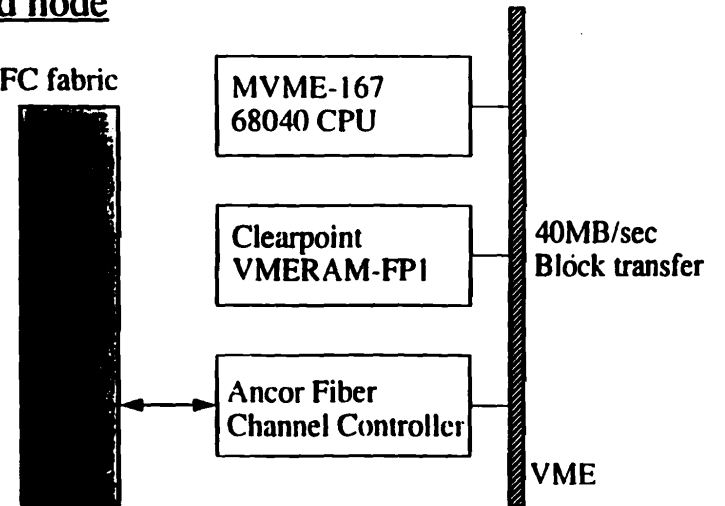
Good for large packet

Connection type network  
(connectionless is also available)

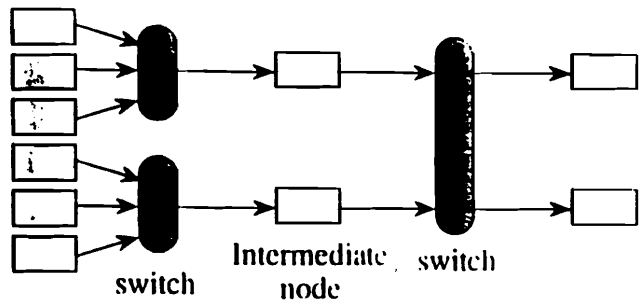
? Connection time for a large number of simultaneous connection requests. ?

## Test Bed node

ANCOR FC fabric



## Architecture

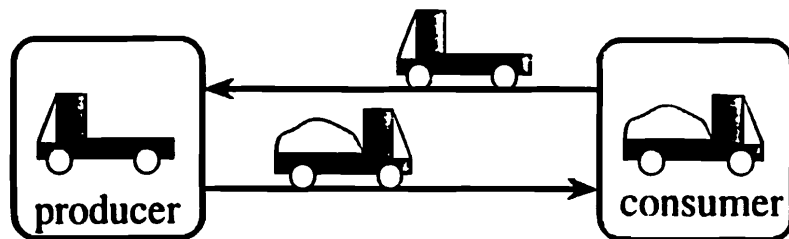


Two stage switch

Single path flow control

Event building node with a DMA engine

## Single path flow control



Allocate fixed number of containers

Allocate buffer for maximum number of containers

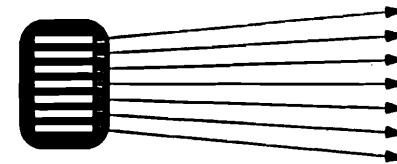
## Two stage switch

case: Packet size > event fragment size

Queue length is proportional to packet size

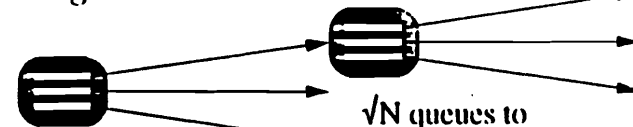
Reduce the latency  
Reduce the number of queue

## *single stage switch*



"N" queues to "N" destinations

## *two stage switch*



$\sqrt{N}$  queues to  
 $\sqrt{N}$  second stage

$\sqrt{N}$  queues to  
 $\sqrt{N}$  destinations

- + Large packet size can be used
- + Large switch is not needed
- Cost of intermediate nodes must be taken into account

## KEK transparent switch

Based on Fermilab's event builder

Reduce the complexity

use simple switch

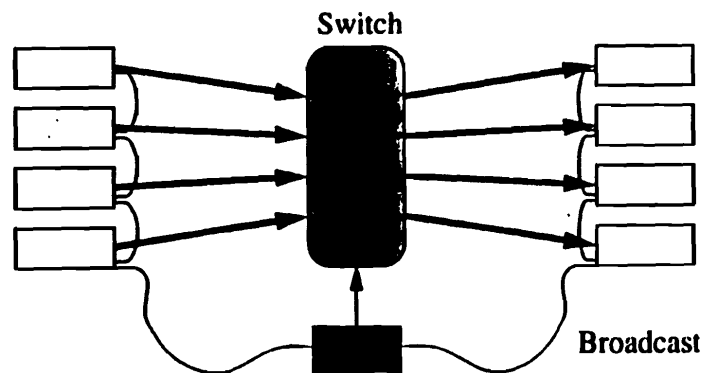
synchronize at input buffer

Lower switching frequency

1k Hz switching

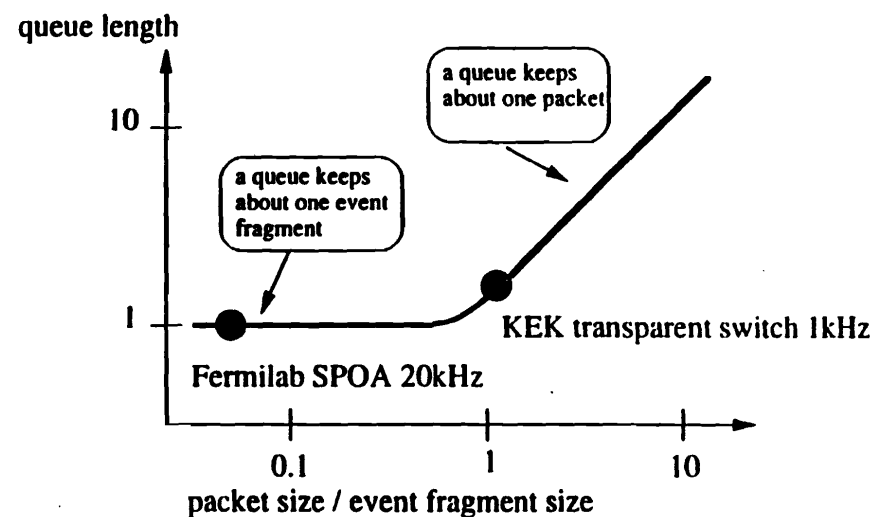
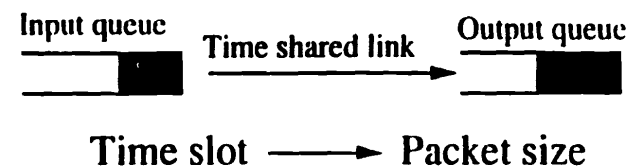
Global traffic control

Broadcast the configuration  
identification.



## Analysis of global traffic control system

SDC note 93-566  
Poster at DAQ conf.



**Packet size  $\approx$  event fragment size**

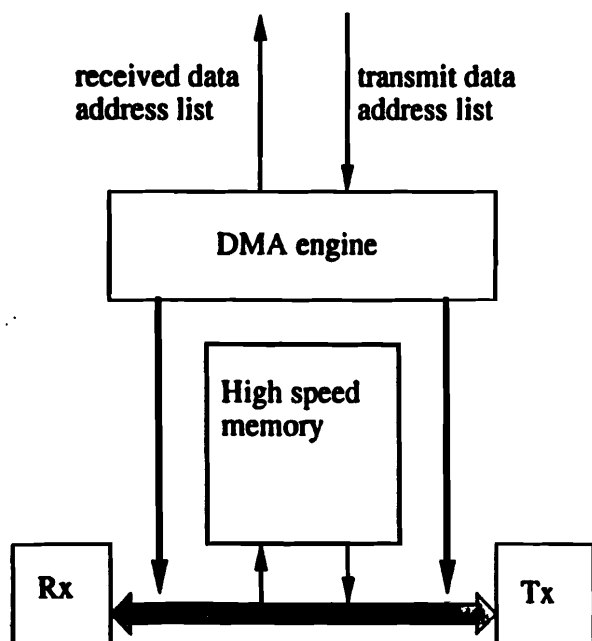
It is an enough small packet size.

## Event Assembling using DMA engine

Fermilab : Dual port memory (Video RAM)



DMA controller on CPU board  
+ High speed Memory



daqconf 26-oct-94, M.Nomachi

## Fiber-optics

### G-link

1 Gbps Chip-set from HP  
developed for serial HIPPI

### Trigger signal / Trigger data link

One 16 bit word = SSC beam crossing  
(Wisconsin / Fermilab)

### Data readout

100MB/sec data readout  
Bit-error tester is developed

(LBL / Fermilab)

### Event builder data link

Re-lock time has been measured  
30 $\mu$ sec is enough short  
for the application

(KEK / Fermilab)

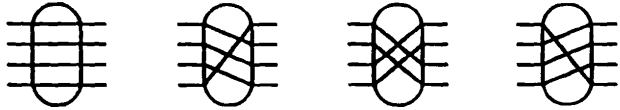
### Conclusion

We can use 1Gbps data link  
Further cost reduction is necessary

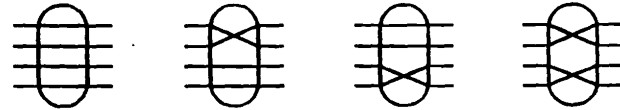
daqconf 26-oct-94, M.Nomachi

## High speed ECL switch

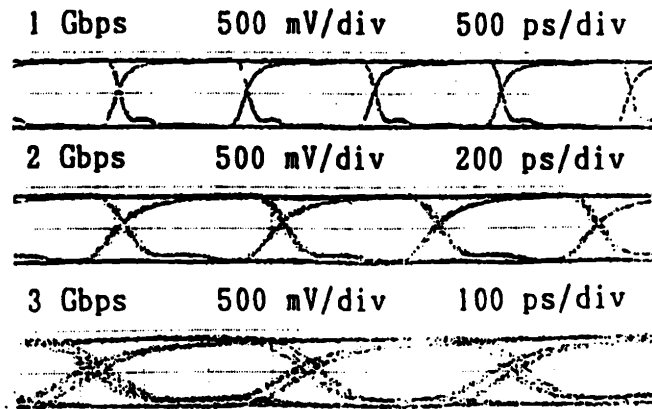
4 x 4 barrel shifter



Dual 2 x 2 switch



Eye pattern



Cascade connection capability

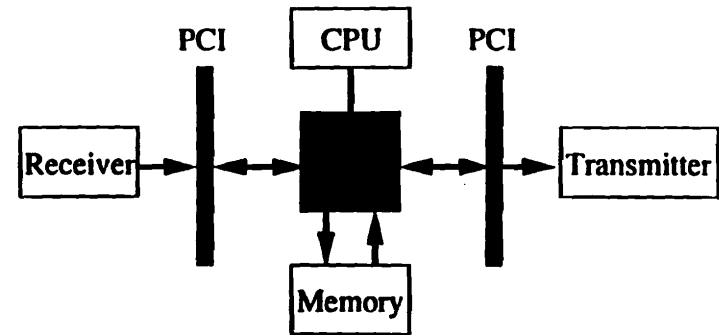
1024 x 1024 switch needs 5 cascade connection

It works at more than 12 cascade connection for 1Gbps signal

daqconf 26-oct-94, M.Nomachi

## Dual port memory *key component in TSI*

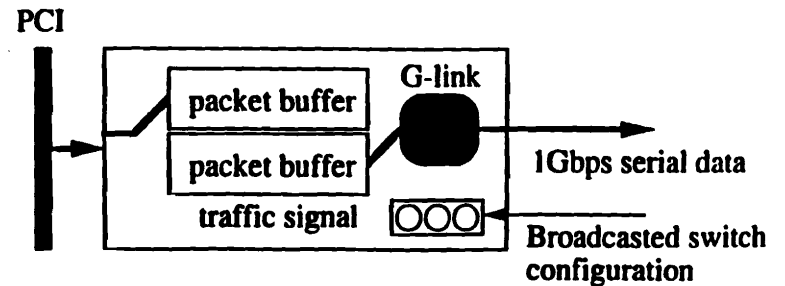
- 1) video RAM (Fermilab)
- 2) DMA controller + High speed memory module
- 3) Dual port memory on CPU board.



PCI

standard inter-chip interface.  
up to 133 MB/sec.

PCI module



daqconf 26-oct-94, M.Nomachi

# Summary

## What we did.

R&D on connection type switch

## What we learned.

How to control the traffic

Global traffic control system has  
been established  
(It is predictable)

Importance of Input/Output buffer

PCI interface may have important role.



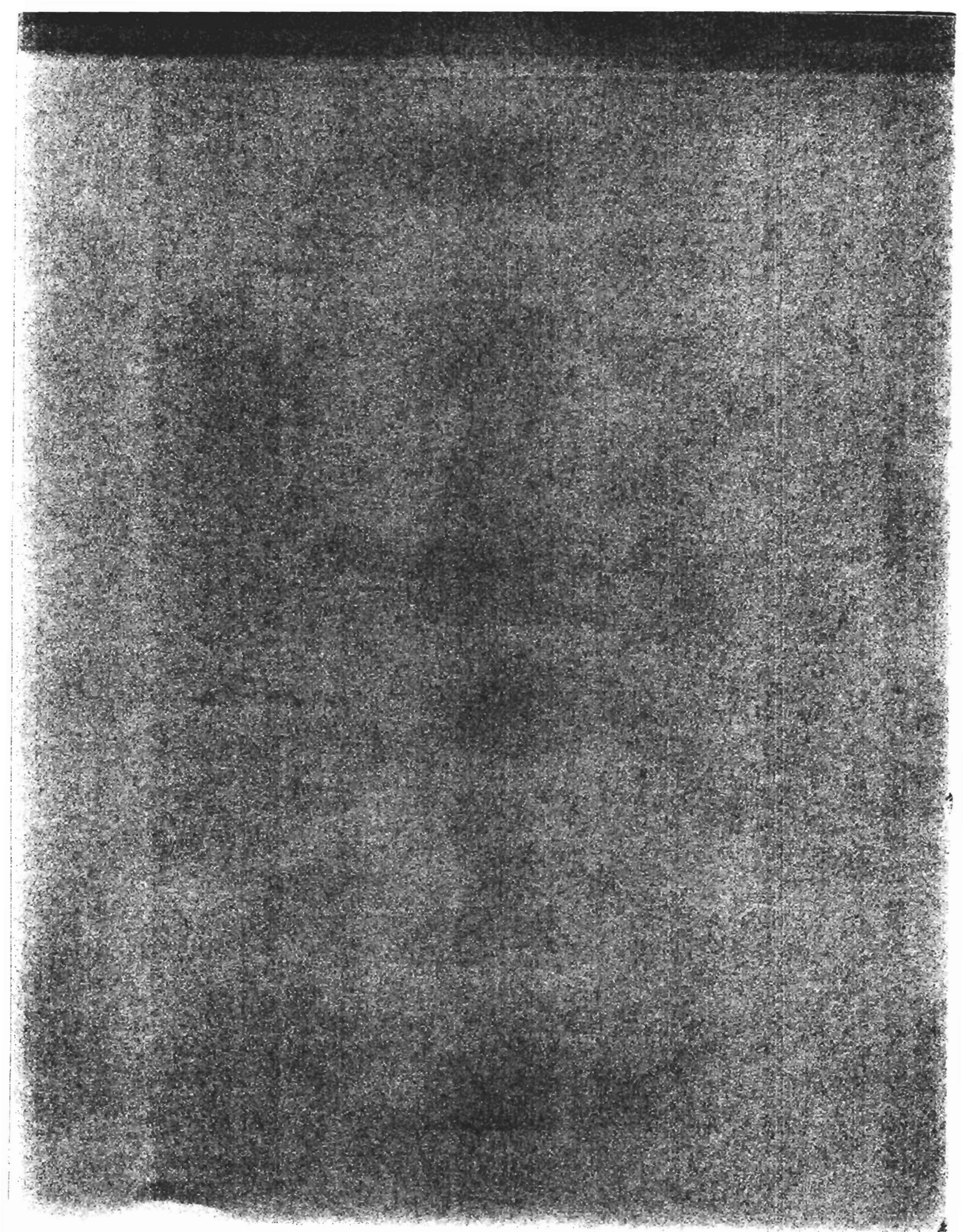


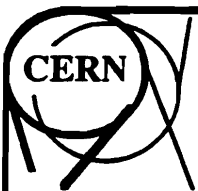
**S3-5**

**"Matrix of Projects and Standards"**

**(Robert McLaren - CERN)**

A short presentation linking the numerous research projects to the standards being evaluated, along with contact information.



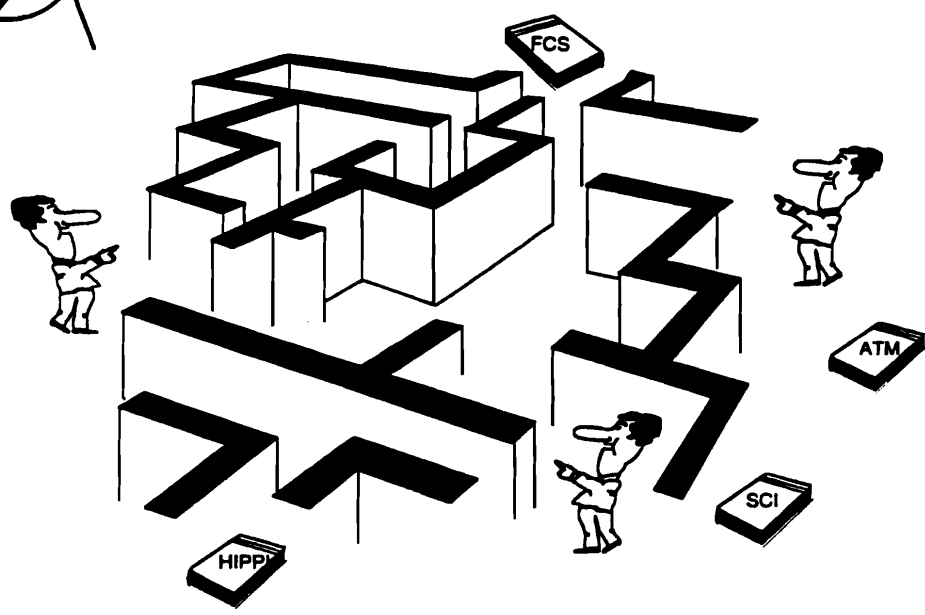
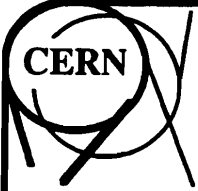


# A MATRIX OF PROJECTS AND STANDARDS

presented by Robert McLaren at the  
International Data Acquisition Conference  
Fermi National Accelerator Laboratory , Batavia, Illinois  
October 26-28, 1994

*Robert.McLaren@cern.ch*

*ECP Division*



*Robert.McLaren@cern.ch*

*ECP Division*

## • The World Wide Web

- **Advantages**

- Vast amount of information
- Easy to browse
- Maintained by the information provider

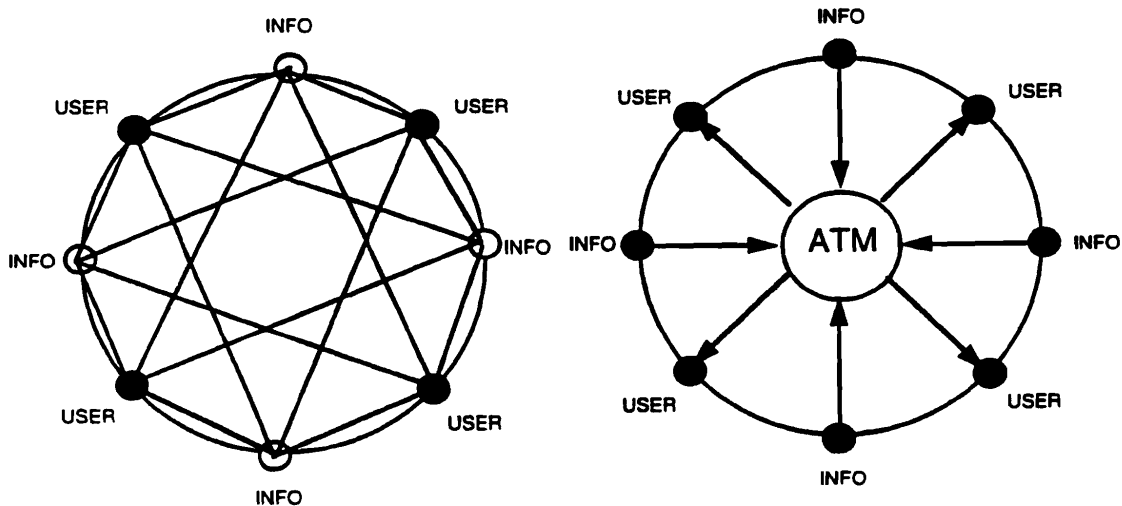
- **Problems**

- Tendency to "wander the Web"
- Not easy to find specific information

- **Solution**

- Cluster information

## An Information Hub

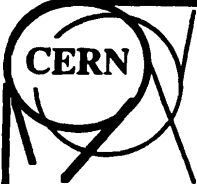


## The High Speed Interconnect Matrix

- **Aim**
  - Centralise information on technologies
  - Centralise information on projects
  - Classify by function in DAQ system
- **Organisation**
  - Set up Technology / DAQ component matrix
  - Programme managers (HIPPI, FCS, ATM, SCI )
    - » Edit the news
    - » Receive information on new products, filter for DAQ
    - » Enter brief note with a pointer to the full article
    - » Close contacts with the Forums

## Technologies

- **News**
- **General introduction**
- **Specifications**
- **Applications**
- **Components**



# Data Acquisition Functions

DATA GENERATION

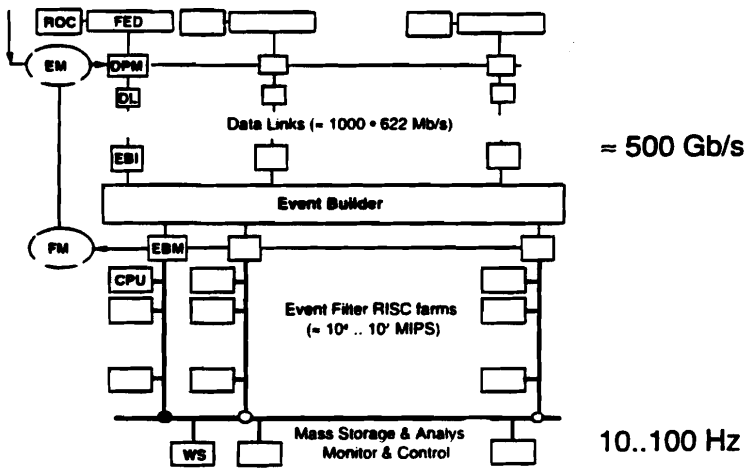
TESTING

READ-OUT

EVENT BUILDING

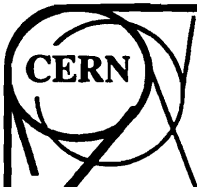
PROCESSING

STORAGE



Robert.McLaren@cern.ch

ECP Division



**Try it out**

**The URL is <http://www.cern.ch/HSI/>**

**and please send us comments**

Robert.McLaren@cern.ch

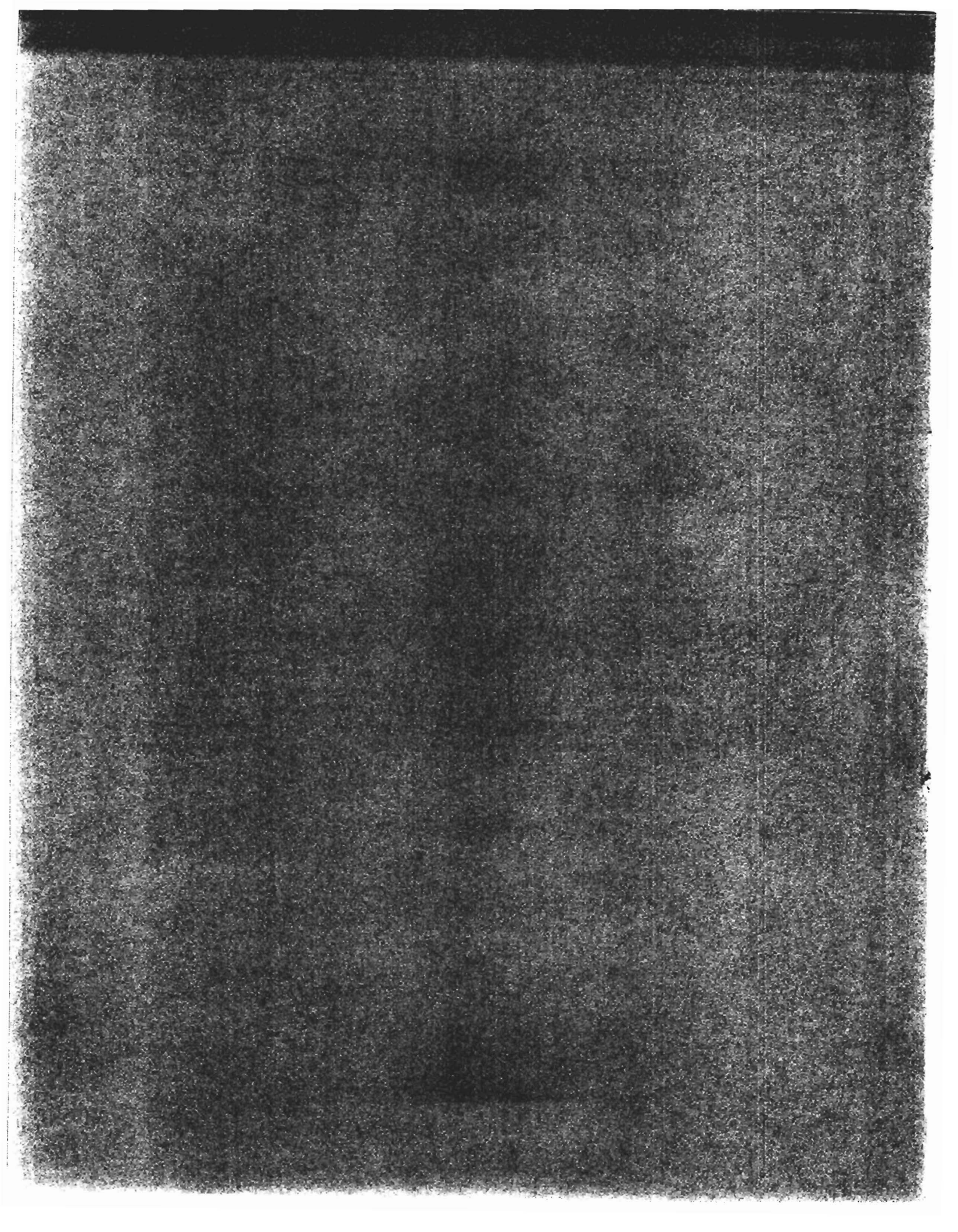
ECP Division

## S3-6

### **"New VME Standards For Physics Applications"**

**(Robert Downing - University of Illinois)**

The VME Standards Organization (VSO), the VME International Trade Association (VITA) and ESONE, CERN, Fermilab & NIM are in the process of forming a Physics Interest Group under VITA with a charter to develop VME hardware and software standards for physics applications compatible to the VME standards that allow features such as special voltages, higher power, geographical addressing, etc. while maintaining compatibility and interchangeability with the base VME standards. Hopefully, this effort will significantly reduce the need for implementers to design their own in-house packaging and bus systems.





# VME for Physics

RWD - U of IL 1

FINAL 27 Oct 1984

## Short History of VME

- 1979 VERSAbus
  - Motorola 68K bus
- 1981 VME Revision A
  - Versa Module Eurocard
  - Motorola, Mostek, Signetics
- 1982 VME Revision B
  - VMEbus Manufacturers' Group Publication
  - Eurocard Based:
    - Connectors: DIN 41612, IEC 603-2
    - Boards: IEEE 1101
    - Racks: DIN 41494, IEC 297-3

RWD - U of IL 2

FINAL 27 Oct 1984

### Short History of VME, cont.

- 1982, October VME Rev. B to IEC
  - IEC SC47B
- 1983, March VME Rev. B to IEEE
  - IEEE P1014
- 1983, Dec. 1985, Feb. - release
  - VME Mnts' Group VME Rev. C
  - IEEE P1014 draft 1.0
  - IEC IEC 821 BUS
- 1984 VITA Formed
  - VME Industry Trade Association

### Short History of VME, cont.

- 1985, August VME Rev. C.1
  - IEEE P1014 draft 1.2
- 1987 VME Rev. C.3
  - IEEE 1014-1987
- 1988 VFEA Formed
  - VME Futurebus+ Extended Architecture
- 1991, January VME64 Started
  - IEEE P1014R RevD
- 1992, September VME Rev. D
  - VSO VITA Standards Organization

## **Short History of VME, cont.**

---

- **1993**            **VME64**
  - VSO P1
  - ANSI Canvas Standards Sponsor
  - Latest Draft Rev. 1.8, 4 January 1994
- **1994, May**    **VME64 Extensions, draft 0.4**
  - VSO P1.x

## **European VME Efforts**

---

- **CERN VMEbus Steering Committee - VSC**
  - Under CERN's User Group for Microprocessor Support - UGMS
  - Formed VSC in 1992
  - Chaired by Chris Parkman, CERN
- **Workshop at CERN 15 March 1994**
- **Member of VITA**
- **ESONE Working Group**

## US VME Efforts

- Formed group under NIM in mid 1994
  - Louis Costrell, NIST - NIM Secretary
  - Chaired by Ed Barsotti, FNAL
  - Preliminary Meeting June 21 with Wayne Fischer
  - First formal meeting September 7-8
- Interest stirred by VME64 Extensions document
- Membership in VITA soon
- Form Special Interest Group in VITA

RWD - U of M 7

FNAL 27 Oct 1994

## VME for Physics

- Minuses:
  - Noisy Bus
    - Lack of ground pins in connector
    - Integrators on certain lines
  - Missing voltages for ECL
  - Card too small for front end
  - No Geographical Addressing
  - Only one error response

RWD - U of M 8

FNAL 27 Oct 1994

## VME for Physics

---

- **Pluses:**
  - Single Board Computers
  - Large Industrial Base
  - Single Board Computers
  - O/S's for SBC's

## Experimental Use Problems

---

- **Many incompatible systems**
  - One experiment has 14 versions of "VME"
- **Analog noise trouble**
  - Bus halted during signal acquisition
- **Incompatible larger boards**
- **Incompatible Protocols**

## VME64 Extension

- Many Grounds Added
- Geographical Addressing
- Additional Power
- More User I/O Pins
- Additional Connector
- Test Bus for Diagnostics and Maintenance

RWD - U of M 11

FNAL 27 Oct 1984

## VME Physics Interests

- Power
  - +/- 15 volts
  - -5.2, -2 volts
  - Analog Ground
  - DC-DC Converters - 48 volts
  - Ground
- Mechanical
  - 9U format
  - Cooling, boards and rear I/O
  - Board Keying
  - Mixing of 6U with 9U

RWD - U of M 12

FNAL 27 Oct 1984

## **VME Physics Interests**

---

- **Protocol**
  - **Unknown Data Lengths**
  - **Geographical Addressing**
  - **Pipeline Block Transfers (SSBT)**
  - **Buses for Front End**
- **Other**
  - **Standard Software**
  - **Live Insertion**
  - **Fusing**
  - **"in between" Power Connectors**

## **Goals for VME-P**

---

- **Recommended Practices Document under VITA**
- **Maximize Industrial Support for Physics Research use of VME**
- **Influence VME64 Extensions Document**
- **Participate with VITA to be on the inside of future developments**

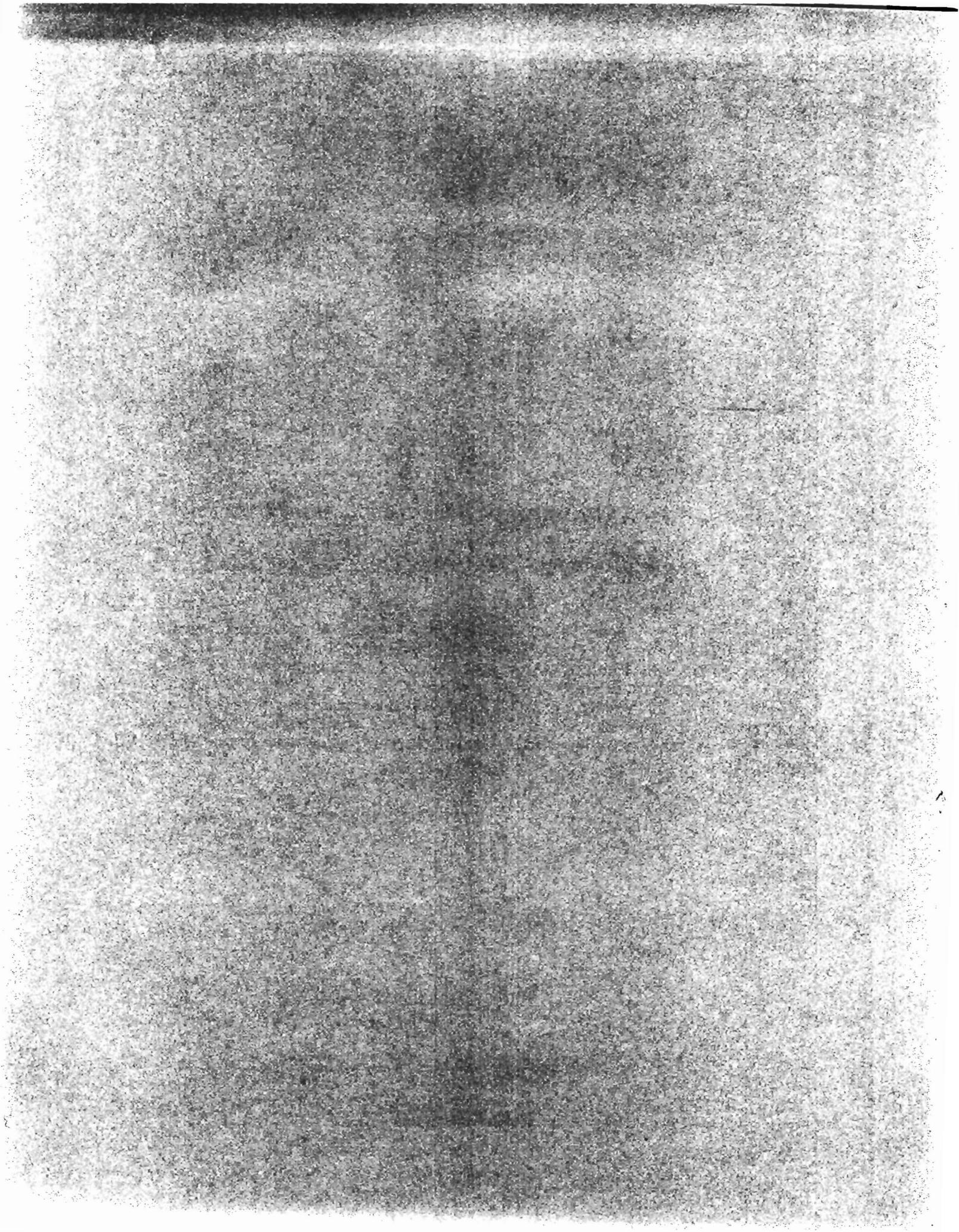
**S4-1**

**"Overview of ATM Integrated Circuits & Board Products"**

**(Lee Goldberg - Electronic Design Magazine)**

This talk will focus on the practical aspects of getting a workstation or other equipment onto an ATM network. A survey of manufacturers will be presented along with some commentary about who's really delivering and who's got problems. Card architecture and its affect on performance will also be discussed. Attendees will receive a list of ATM adapter cards that are copies of the ATM silicon listing published in the 4/94 Electronic Design magazine as well as advance copies of a listing of ATM switching devices that will appear in the 12/16 edition. Within the limited range of his knowledge, Lee will cheerfully answer questions on other ATM related topics.







# ATM Network Interface Cards: A Brief Overview

Presented by Lee Goldberg - Communications Editor  
Electronic Design Magazine



## A "NIC" Performs Several Functions

- Transfers ATM packets between the host system and the network
- Performs SAR functions on data
- Provides framing or other PHY-layer format processing (SONET, CAP-16 etc...)
- Transmits and receives data across physical medium (UTP copper, fiber, etc...)
- Buffers host system from raw data stream
- Provides some level of traffic control between host and network

**"Your mileage may vary" - A card's sustained transfer rate may be as little as 1/3 line speed. The common bottlenecks show up in:**

- Host system bus bandwidth
- Data transfer schemes
- Card architecture
- Driver software

**Raw bandwidth is the first of many factors**

**ISA - 8.3 Mbytes/s**

**MicroChannel - 20 Mbytes/s**

**EISA - 33Mbytes/s**

**SBus - up to 100 MBytes/s**

**PCI - up to 320 Mbytes/s**

**ELECTRONIC**  
**DESIGN**

## **Raw bandwidth shrinks rapidly in the face of system overhead**

**EISA - approx 15 Mbytes/s (almost 1/2 duplex @ 155 M)**

**SBus - 25-30 MBytes/s**

**PCI - 40-50 Mbytes/s - (burst to 80 M)**

**Approx 40 Mbytes/s required to support full duplex transfers at the full 155 Mbits/s line speed**



- **Byte-by-byte transfers involve more overhead**
- **Generally, DMA block transfers in the maximum block size afford the most efficient data transfer**
- **For SBus systems**
  - **32 or 64-byte maximum size xfr block**
  - **5 clock cycles to initiate transfer**
  - **1 clock per 32-bit word thereafter**
- **Bus mastering is the preferred transfer method on the PCI bus - variable burst size also helps**



## Some factors affecting card performance

- Buffering
- SAR control structures
- Host bus interface



### SAR control structures

- A SAR must stay informed about how it must treat data from each VC it supports:
  - VC identification
  - rate and flow control parameters
  - AAL 3/4 processing parameters
- Channel Parameter data may be stored in buffer RAM or in separate RAM
- The SAR function needs to access its control channel as little as possible unless it is on a separate interface from the cell data path



### Buffering - expensive but worth it

- Allows block transfers
- Initial latency of one cell time
- Most cards buffer on a per-VC basis
- requires 2 PDUs per VC to avoid cell loss with each PDU running 1-to-64 Kbytes
- Significant SRAM (1/2 to 2 MB) required to support many VCs
- A good NIC has dynamic buffer allocation to cope with variable length PDUs



### Host bus interface

- For PCI bus - look at whether card supports bus mastering
- Check word width and block size that card supports



- Watch for excessive latency and overhead when using PC-based machines running Windows NT. You need to directly tie application to the card using manufacturer's APIs to cure this.
- Until UNI 3.1 spec arrives, not all cards will be able to drive all switches - check with vendors for compatability with your existing hardware
- SVCs are in the near future, for now, PVCS must be manually set up until UNI 3.1 arrives.
- Most hardware shipped today will support UNI 3.1 when it arrives. Only software updates will be required. Check with vendors to be sure.

## COMMERCIALLY AVAILABLE ATM SWITCHING ICs

Vendor	Device	Functions and features	Price, availability, and comments
AT&T Microelectronics Dept. AL50040200 Allentown, PA (800) 372-2447 xP21 (610) 712-4106 fax	T7650 2-by-2 crosspoint switch	Cascadable and self-routing crosspoint architecture. 320-Mbit/s bandwidth/port. Crosspoint-buffered switch matrix using back-pressure flow control.	\$25 each in lots of 10,000; available now.
	T7652 ATM layer-interface	Works directly with T7650 switch. Performs address translation and policing on virtual channels. Extracts and inserts OA&M signaling cells. Supplies traffic statistics to host interface. 32-cell FIFO buffer with 4 internal priority levels.	\$35 each in lots of 10,000; available now. Uses internal 32-bit bus architecture. Requires network termination.
Fujitsu Microelectronics San Jose, CA (408) 526-8515 Attn: Betsy Taub	MB86680 4-by-4 self-routing switch	Cascadable and self-routing crosspoint architecture. 155-Mbit/s bandwidth/port. Non-blocking architecture lets up to 5 cells arrive simultaneously. Programmable threshold sets congestion-notification flag. Supports multicasting without copying cells.	\$70 each in lots of 10,000; available now. Self-routing feature uses packet tagging. May be cascaded in matrix, delta and Clos configurations up to 32 by 32.
	MB86683 network-termination controller	Implements transmission convergence for SDH/Sonet at 100 and 155 Mbit/s. Supports PHY-layer OA&M. Gathers OA&M statistics for host processor.	\$50 each in lots of 10,000; available now.
	MB86686 adaption-layer controller	155-Mbit/s I/O bandwidth. Implements AAL 3, 4, and 5 functions. Performs cell tagging and detagging for routing within switch matrix.	\$75 each in lots of 10,000; available now.
	MB86689 address-translation controller	Uses 1024-entry CAM. Full 28-bit comparison with optional masking. Supports multiple matches for multicasting. Cascadable for extended addressing range.	\$40 each in lots of 10,000; available now.
IBM Microelectronics Research Triangle Park, NC (800) 426-3333	Przma 16-by-16 Switch-on-a-Chip	400-Mbit/s bandwidth/port. Novel shared-packet buffer uses rotating shift registers to eliminate blocking. Cascadable for higher speed and/or matrix size. Supports efficient multicasting.	Marketing plans under development; available now to selected equipment manufacturers.
Integrated Telecom Technology (IGT) Gaithersburg, MD (301) 990-9890 (301) 990-9893 fax	WAC-188A 8-by-8 switch	Shared-buffer architecture with dynamically allocated 32-cell output-buffer pool. Programmable buffer-level congestion control. Buffering supports up to 5 priority levels. 155-Mbit/s bandwidth/port. Efficient multicasting minimizes traffic. Cascadable for higher speed and/or matrix size.	\$76 each in lots of 25,000; sampling now; production in the first quarter 1995.
	WAC-187A routing table and switch buffer	Performs address translation up to 4096 channels. Uses external SRAM for input buffering of 512, 3027, or 7168 cells. Internal output buffer. Supports OA&M cell insertion and extraction. Host interface supports OA&M functions. Programmable congestion-control functions on a per-channel basis.	\$61 each in lots of 25,000; sampling now; production in the first quarter 1995.
	WAC-186A UPC/OAM processor	Monitors incoming cells for violations of negotiated-rate and service parameters. Selectively discards or tags cells for discarding. Supports monitoring of cell rates and delays on per-channel basis. Performs OA&M monitoring and alarm functions. Performs OA&M loopback and continuity tests.	\$71 each in lots of 25,000; sampling first quarter of 1995; production in the second quarter of 1995.
Music Semiconductors Colorado Springs, Co (719) 570-1555	MUSC 1480 LANCAM address translator	Provides VPI/VCI address translation. Uses 1-kbit CAM. Supports ATM switching up to 200 Mbit/s. Faster version available on request. Sophisticated search structure aids multicasting. Can perform cell tagging for self-routing fabrics and limited cell policing for good addresses.	\$28 each in lots of 10,000; available now. Development kit, which includes VL or ISA-bus card and development software, costs \$185.



## COMMERCIALLY AVAILABLE ATM SWITCHING ICs

Vendor	Device	Functions and features	Price, availability, and comments
Transwitch Corp. Shelton, CT (203) 929-8810 (203) 926-9453 fax	Cubit TXC05801 CellBus switch	Shared-bus architecture simplifies design. Good aggregate bandwidth supports up to eight 155-Mbit/s ports. A 4-cell input buffer and a 100+ cell output buffer. Self-contained address translation. Efficient broadcasting and multicasting characteristics.	\$30 each in lots of 10,000; available in the second quarter of 1995. Framing and termination chips also available. Development boards facilitate product development and testing.
TriQuint Semiconductor Beaverton, OR (503) 644-3535 (503) 644-3198 fax Attn: Dave Drummond	TQ8016 16-by-16 crosspoint switch	1.3-Gbit/s per port capacity. Very low latency, latency variation, crosstalk and jitter. Excellent broadcasting and multicasting characteristics. Crosspoint architecture easily expands into matrix, delta, Clos, or other configuration.	\$163 each in lots of 1000; available now. Custom impedance-controlled package.
	TQ8032 32-by-32 crosspoint switch	800-Mbit/s per port capacity. Very low latency, latency variation, crosstalk and jitter. Excellent broadcasting and multicasting characteristics. Crosspoint architecture easily expands into matrix, delta, Clos, or other configuration.	\$406 each in lots of 1000; available now. Custom-controlled impedance package.
	TQ8015/8017 16-by-16 crosspoint switch	1.2-Gbit/s per-port capacity. 8017 switch has PECL I/O for Fibre Channel. Very low latency, latency variation, crosstalk and jitter. Excellent broadcasting and multicasting characteristics. Crosspoint architecture easily expands into matrix, delta, Clos, or other configurations.	\$100 each in lots of 10,000; sampling now; production in the first quarter of 1995. Industry-standard packaging for lower cost.
VLSI Technology San Jose, CA (408) 434-3000 (408) 434-7931 fax	Custom ASICs No standard products at this time	Company has been active in building ASICs specifically for ATM switching. Customers include Lightstream Corp. FlexArray technology cuts development time. Functional standard cells include T1/E1/Sonet/SDH and ATM processing logic.	Call for details.

*ATM = asynchronous transfer mode; CAM = contents-addressable memory; OA&M = operations, administration and maintenance; VCI = virtual-channel interface; VPI = virtual-path interface.*

<b>SysKonnct, Inc.</b> 1922 Zanker Road San Jose, CA 95112  Sales 800-752-3334  Donna Elmore V.P. Marketing 408-437-3840	SK-NET ATM Sbus	Sbus	155 Mbps SONET, MMF	Solaris, SunOS	65-70 Mbps	NEC, FORE, Synoptics	TCP/IP, RFC 1577, UNI 3.0 SVC, AAL5	Now	\$1,895
	SK-NET ATM Sbus	Sbus	155 Mbps SONET, UTP-5	Solaris, SunOS	65-70 Mbps	NEC, FORE, Synoptics	TCP/IP, RFC 1577, UNI 3.0, SVC, AAL5	10/94	\$1,395
	SK-NET ATM EISA	EISA	155 Mbps SONET, MMF	NetWare, Windows NT, IRIX, HP-UX	NA	NEC, FORE, Synoptics	TCP/IP, RFC 1577, UNI 3.0, SVC, AAL5	4Q94	\$1,995
	SK-NET ATM EISA	EISA	155 Mbps SONET, UTP-5	NetWare, Windows NT, IRIX HP-UX	NA	NEC, FORE, Synoptics	TCP/IP, RFC 1577, UNI 3.0, SVC, AAL5	4Q94	\$1,695
	SK-NET ATM PCI	PCI	155 Mbps SONET, MMF, UTP-5	Windows NT, NetWare and DOS	NA	NEC, FORE, Synoptics	NA	1H95	NA
<b>Transcell Systems, Inc.</b> 3000 Scott Blvd. Suite 111 Santa Clara, CA 95054  Mahesh Veerjna Sales/Marketing Contact 408-958-5353	PCI-100F	PCI	155 Mbps SONET, MMF	DOS, Windows, NT, Netware, UNIX	NA	NA	UNI 3.1 SVC-1Q95, PVC, AAL3/4/5, ODI, NDIS, TCP/IP, IPX/SPX	4Q94	\$1,495
	PCI-100C	PCI	155 Mbps SONET, UTP-5	DOS, Windows, NT, Netware, UNIX	NA	NA	UNI 3.1 SVC-1Q95, PVC, AAL3/4/5, ODI, NDIS, TCP/IP, IPX/SPX	4Q94	\$1,295
	VLB-100F	VESA	155 Mbps SONET, MMF	DOS, Windows, NT, Netware, UNIX	NA	NA	UNI 3.1 SVC-1Q95, PVC, AAL3/4/5, ODI, NDIS, TCP/IP, IPX/SPX	4Q94	\$1,295
	VLB-100C	VESA	155 Mbps SONET, UTP-5	DOS, Windows, NT, Netware, UNIX	NA	NA	UNI 3.1 SVC-1Q95, PVC, AAL3/4/5, ODI, NDIS, TCP/IP, IPX/SPX	4Q94	\$1,095
<b>WhiteTree Network Tech</b> Wayne Marston 415-855-0871	NA	PCI	25 Mbps, UTP-3	Netware, NT, Windows	NA	NA	AAL 5, LE	2Q95	NA
	NA	Sbus	25 Mbps, UTP-3	SunOS, Solaris	NA	NA	AAL 5, LE	2Q95	NA
<b>ZellNet, Inc.</b> 2255 Martin Suite F Santa Clara, CA 95050  Jim Horn 408-562-1800	ZN1221 PCI	PCI	155 Mbps SONET, Fiber - 2 Km	Windows NT, Netware, NDIS, ODI	NA	NA	UNI 3.0/3.1, SNMP, SVC, AAL5, IP/ATM, ILMI, LE, Signalling	4Q94	\$1,095
	ZN1225 PCI	PCI	155 Mbps SONET, UTP5	Windows NT, Netware, NDIS, ODI	NA	NA	UNI 3.0/3.1, SNMP, SVC, AAL5, IP/ATM, ILMI, LE, Signalling	4Q94	\$995
	ZN1228 PCI	PCI	100 Mbps TAXI, MMF	Windows NT, Netware, NDIS, ODI	NA	NA	UNI 3.0/3.1, SNMP, SVC, AAL5, IP/ATM, ILMI, LE, Signalling	NA	\$1,095
	ZN1211 Sbus	Sbus	155 Mbps SONET, Fiber - 2 Km	Solaris/Sun OS	NA	NA	UNI 3.0/3.1, SNMP, SVC, AAL5, IP/ATM, ILMI, LE, Signalling	4Q94	\$995
	ZN1215 Sbus	Sbus	155 Mbps SONET, UTP5	Solaris/Sun OS	NA	NA	UNI 3.0/3.1, SNMP, SVC, AAL5, IP/ATM, ILMI, LE, Signalling	4Q94	\$895
	ZN1218 Sbus	Sbus	100 Mbps TAXI, MMF	Solaris/Sun OS	NA	NA	UNI 3.0/3.1, SNMP, SVC, AAL5, IP/ATM, ILMI, LE, Signalling	NA	\$995
KEY: MMF=Multimode Fiber; SMF=Single Mode Fiber; UTP=Unshielded Twisted Pair; LE=Lan Emulation; MC=Microchannel Bus; HW=Hardware; SW=Software									

Company	Product	Bus	Data Rate & Physical Interface	Platform & OS Support	Sustained Data Rate	ATM Switch Support	Features	Avail.	Price
<b>Madge Networks, Inc.</b> 800-876-2343	NA	EISA	155 SONET, MMF	Netware	NA	NA	LE, AAL 5	NA	NA
<b>National Semiconductor</b> 2900 Semiconductor Dr. P.O. Box 58090 Santa Clara, CA 95052 Mark Wotter ATM Product Manager 408-721-6099	NA	EISA	155 Mbps SONET, MMF, UTP, 4B/5B MMF, DS3 coax	Netware, NT, Unxware	NA	NA	AAL 3.4 & 5, LE	Not in Prod	\$2,500
<b>Newbridge Networks Inc.</b> <b>VIVID</b> 693 Herndon Parkway Herndon, VA 22070-5241  Donna Cowan 703-708-8904	VIVID Sbus	Sbus	155 Mbps SONET, MMF, UTP-5	Sun Sparc Station, Sparc Center, Solaris 2.3, Sun O/S 4.1.3	ATM Layer: 148 Mbps UDP: 100 Mbps	VIVID, Newbridge FORE	SNMP, SVC, UNI 3.0/3.1, AAL-5, IP, ATM API	Now	\$1,995
	VIVID Elsa	EISA	155 Mbps SONET, MMF, UTP-5	Novell 4.0, Windows, Windows NT, HP-UX, IRIX 5.2	ATM Layer: 148 Mbps UDP: 100 Mbps	VIVID, Newbridge, FORE	SNMP, SVC, UNI 3.0/3.1, AAL-5, IP, ATM API	10/94	\$1,995
	VIVID GIO	GIO	155 Mbps SONET, MMF, UTP-5	for Indy; IRIX 5.2	ATM Layer: 148 Mbps UDP: 100 Mbps	VIVID, Newbridge, FORE ASX-100	SNMP, SVC, UNI 3.0/3.1, AAL-5, IP, ATM API	12/94	\$1,995
	VIVID VME	VME	155 Mbps SONET, MMF, UTP-5	for SGI Challenge; IRIX 5.2	ATM Layer: 148 Mbps UDP: 100 Mbps	VIVID, Newbridge, FORE	SNMP, SVC, UNI 3.0/3.1, AAL-5, IP, ATM API	2/95	\$3,500
	VIVID PCI	PCI	155 Mbps SONET, MMF, UTP-5	PC: Novell/Windows/Windows NT	ATM Layer: 148 Mbps UDP: 100 Mbps	VIVID, Newbridge, FORE ASX-100	SNMP, SVC, UNI 3.0/3.1, AAL-5, IP, ATM API	2/95	\$1,995
<b>Okcom USA Inc.</b> Dallas, TX  Max Jensen 214-423-7580	NA	EISA	155 Mbps SONET, MMF	Netware, NT	NA	NA	AAL 5, LE	6/94	NA
<b>SMC</b> Hauppauge, NY 800-SMC-4YDU	SMC PC Adapters		155 Mbps	NA	NA	NA	UNI 3.1/4.0, Q.2931, PVC, SVC, Q.2931, AAL1&5, LE 802.3, 802.5 & Classical IP.	2Q95	NA
<b>Sun Microsystems, Inc.</b> 2550 Garcia Ave. Mt. View, CA 94043 Andi Uberoi Product Line Manager 415-336-0703	SUN ATM 155	Sbus	155 Mbps MMF	Sun Workstations, Solaris	NA	FORE Synoptics Newbridge Entel	SVC, Q.93B, AAL5 In HW, AAL1 In SW, Drivers for Solaris	1/95	\$1,295
	SUN ATM 155	Sbus	155 Mbps UTP-5	Sun Workstations, Solaris	NA	FORE, Synoptics, Newbridge, Entel	SVC, Q.93B, AAL5 In HW, AAL1 In SW, Solaris drivers	1/95	\$995

# **THE ATM REPORT**

**YES!** Start my FREE 3-month subscription to THE ATM REPORT today! I will be under no obligation to subscribe.

**YES!** I want to save \$180.00. Begin my subscription today, so I can receive a 20% discount off the regular subscription price and get three extra months free. (Offer good for new subscribers only.)

**Special Show Offer: \$318 for 15 issues**

**Regular Subscription Price: \$398 for 12 issues**

Name \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City, State, ZIP+4 \_\_\_\_\_

Phone number/FAX number \_\_\_\_\_

**FREE**

**\$180 Value**

**FOR SHOW ATTENDEES**

Just mail this completed card to us to receive your free 3-month subscription to:

**THE ATM REPORT**  
815-B Rockville Pike #240  
Rockville, MD 20852  
TEL: (301) 816-7858  
FAX: (301) 816-3021

**FMI**



**S4-2**

**"Overview of Fibre Channel Integrated Circuits & Board  
Products"**

**(Murray Thompson - University of Wisconsin)**

A Comparison of the latest Fibre Channel ASICs and modules will be presented. The comparison will include the bandwidths, functionality, costs and availability of the Fibre Channel devices.



# Fibre Channel ASICs and Modules

Murray A. Thompson

High Bandwidth Communications Group

Physics Dept.  
University of Wisconsin-Madison  
thompson@WISHPA.physics.wisc.edu  
608 262 8509

Fibre Channel is Growing!

$$a e^{bt}$$

Every exponential with a positive  
exponent is initially small!



FC-0 is the Fibre Channel Physical Layer which is the actual bidirectional point to point serial data channel. It is sometimes divided into two sublayers

1. "Interface" - This has the Transmitters and Receivers. It includes the Parallel to Serial and Serial to Parallel conversions
2. "Media" - A variety of optical fibers and even coaxial cables

FC-0 does NOT include the 8b/10b encoding or decoding (FC-1), Ordered Sets or Framing

**FC-0 Components First Table**

Vendor	Model	Device	Bandwidth	higher /lower	Avail
AT&T	ATT1409A	Optical Link Card	266		Now
AT&T	ATT1408N	Op Trans	266		93Q1
AT&T	ATT1238A	Xmtr mod	1062	?	Now
AT&T	ATT1318A	Rcvr mod	1062	?	Now
AMCC	S2032	Parallel to Serial	1062		now
AMCC	S2033	Serial to Parallel	1062		now
HP	HDMP-1512	Trans	1062	20 TTL /PECL	?
HP	HDMP-1514	Rec-PECL	1062	20 TTL /PECL	?
HP	GLM	FC-0 Trans/Rec	1062	20 TTL PECL	Now
AMD	8b/10b Taxi	?	?	?	?
AT&T/NCR	NCR85C266	Xmtr/Rec IC	266		Now
Finisar	?	short wave laser-rec	1062	?	?
Force	2556T	Op Trans	1062		Now
Force	2556R	Op Rec	1062		Now
Force	2666T	Op Trans	1062		Now
Force	2666R	Op Trans	1062		Now

**FC-0 Components Second Table**

Vendor	Model	Device	Bandwidth	higher /lower	Avail
Force	2667T	Op Trans	2124	?	Now
Force	2667R	Op Trans	2124	?	Now
Force	2581T	Op Trans	266	?	Now
Force	2581R	Op Trans	266	?	Now
Force	2684T	Op Trans	1062	?	93Q4
Force	2684R	Op Trans	1062	?	93Q4
Fujikura	FC Serial	?	531	?	?
Fujikura	FC Laser	?	531	?	?
Fujikura	FC Parallel data link	?	531	?	?
HP Comp	Op HOLC-0266	short wave laser card	266	?	Now
HP Comp	Op HDMP-1514/12	?	1062	?	94Q4
HP Comp	Op HDMP-?	daughter Card	?	?	94Q4
HP Comp	Op HFBR-5301/2	Op Transceiver	266	?	94Q2
IBM AS/400	OLC-266	FC-0	266	10 bit /light	Now
IBM AS/400	OLM-266	FC-0	266	10 bit /light	?
IBM AS/400	OLC-531	FC-0	531	10 bit /light	Now
IBM AS/400	OLM-531	FC-0	531	10 bit /light	Now
IBM AS/400	OLM-1063	FC-0	1062	10 bit /light	?

**FC-0 Components Third Table**

Vendor	Model	Device	Bandwidth	higher /lower	Avail
SGS-Thomson	IMSSC101	Link-Parallel Conv	266	?	?
Siemens Fib Op Comp	RX-266T2E long wave REC	266	?	?	
Siemens Fib Op Comp	TX-266T2E long wave trans	266	?	?	
TriQuint	GA9040A	Optic Mod	1062	? /light	Now
TriQuint	GA9101 /GA9102A	Xmt/Rec	266	? /?	Now
TriQuint	FC-266	FC-0 IC	266	? /?	Now
TriQuint	FC-200	Xmt/Rec	266	? /?	Now
TriQuint	GA9301 /GA9302	FC Xmt/Rec	1062	? /?	?
TriQuint	PC-1000	FC Xmt/Rec	1062	? /?	?

FC-1 is the Fibre Channel Physical Layer which uses the FC-0 bidirectional point to point serial data channel. It includes the 8b/10b encoding and decoding and passes 10b data in parallel (or 20 bit or 40 bit) to the FC-0. FC-1 includes

1. Ordered Sets (RDY etc)
2. IDLE pattern

FC-1 does NOT include the framing

**FC-1 Components**

Vendor	Model	Device	Bandwidth	higher /lower	Avail
Cypress	CY7B923 Hotlink	Trans	1062	16 TTL /PECL	now
Cypress	CY7B933 Hotlink	Rec	1062	16 TTL /PECL	now
AT&T	ATTDA202A	Xmt	?	?	?
AT&T	ATTDA203A	Rec	?	?	?
AT&T	ATTDA204A	F0+F1 Rec			
AT&T	ATTDA205A	F0+F1 Trans			
AT&T	ATTDA208A	Rec	?	?	93 Oct
AT&T	ATTDA210A	Trans	?	?	?
AMCC	S2039	Par to Ser Trans	1062		
AMCC	S2040	Ser to Par Rec	1062		
HP	HFBR-5301	?	133 pair	?	?
HP	HFBR-5302	?	266 pair	?	?
Motorola	?	?	?	?	dropped
Vitesse	VSC7105	FC Xmt	1062	?	Now
Vitesse	VSC7106	FC Rec	1062	?	Now
Vitesse	VSC7105	Endec	1062	?	94Jun

FC-2 is the Fibre Channel Physical Layer which uses the FC-1 8b/10b links, ordered sets and IDLEs. FC-2 includes

1. Framing
2. Classes of service
3. Fabric
4. Sequences
5. Exchanges

FC-2 can normally communicate directly to the Host computer's Memory.

### FC-2 Components

Vendor	Model	Device	Bandwidth	higher /lower	Avail	
Interphase	5026	HP-PB Adapter	1062		95Q3	?
Interphase	4526	PMC Adapter	1062		95Q3	?
Interphase	5526	PCI Adapter	1062		95Q3	?
HP	Tachyon	ASIC	1062	Mem /20 TTL	95Q1	Class 1+2+3
Ancor	VHSCI 4	ASIC	1062	Mem /20 TTL	Now	Class 1 (2)
Emulex	Chipset		1062	TTL /PECL		Class 1 + 2
AT&T	ATDF200	Adap Eval Card	1062	?	93Oct	

### FC Adaptor and Other Devices

Vendor	Model	Device	Bandwidth	higher /lower	Avail
Seagate	Barracuda	4GB Disk	1062	Disk /Serial Cu	95Q1
AMCC	2025	Crosspoint IC	1062		95Q1
Ancor	EISA250	adaptor	266		Now
Ancor	MCA250	Adaptor	266		Now
Ancor	VME/64 250	Adaptor	266		Now
Ancor	HiPPI 250	Gateway	266		Now
Ancor	FCS 250	Legacy Router	266		Now
Emulex		PCI Adapter	1062		
Augment	AL303	S-bus Adaptor	?	?	94Q1
Augment	AL301	Nibus Adaptor	?	?	94Q1
Cypress	CY9266-F/C	HotLink Eval Card	266	?	?
IBM RISC	?	MCA Adaptor	266	?	?
Interphase	?	Sbus Adaptor	1062	?	94Q1
Jaycor	?	SBus Adaptor	1062	?	94Q1

### FC Fabrics

Vendor	Model	Device	Bandwidth	higher /lower	Avail	Classes
Ancor	CXT 250D1	8,16 Port Clos	266		Now	1+(2)
Ancor	CXT 250DM	... 64 port Clos	266		Now	1+(2)
HP CNO	HPFC Sw	16 Port un blocking	266			1+2+3
IBM Risc	?	FC Fabric for RS/6000	?	?	?	
SGS-Thomson	IMSC104	32 way dynamic Routing	266	?	?	

### FC Vendors and Contacts

Vendor	Contact Person	phone	Fax
AMCC	John Mazzaferro	619 535 4274	
AMD	Jim Kubinec	408 987 2302	Fax 408 749 2800
AMP	Charles Brill	717 561 6198	717 561 6179
Ancor	Clint Jurgens	612 932 4000	612 932 4037
AT&T Microelectronics	Jay Sherfey	215 439 5726	
AT&T/NCR Microelectronics			
Augment Systems	Harry Ives	617 275 4461	
Cypress	Ed Silva	408 943 2693	
Emulex	Charles Bazzar	714 513 8154	
Finisar	Jerry Rawls	415 364 2722	415 364 3011
Force	David Goff or Wendell Hensley	703 382 0462	
Fujikura	Mike Morando	408 988 7406	
	Tom Robinson	803 433 5322	
GEC Plessey	Phil Welsh	408 439 6046	408 438 5576
HP (CNO)	Kumar Malavalli	416 490 3331	
HP Opt Com	Ron Whitetree	408 435 4283	408 435 6506
HP Inf Networks	Don Wilson	408 447 2388	
IBM RISC Sys	Jim Silva	512 838 3700	
IBM AS/400	Steve Sibley	507 253 2943	
Interphase	Ernest Godsey	214 919 9122	
Jaycor	Teri Parish	619 535 3174	
Netstar			
Radway	Yony Talgam	972 3 645 8515	972 3 645 6585
SGS Thomson	Forrest Crowell	714 957 6018	714 957 3281
Siemens Fib Opt	Sheito Van Doorn	201 890 1606	
Sun Microsystems	Bob Williamsen	415 336 5335	
Triquant	Sundh Sanghavi	408 982 0900	
Vitesse	Howey Chun	408 730 3648	
	Brett Butler	805 398 7108	

## Anchor Communications Fibre Channel Product Summary

	CXT 1000 D1	MainStage Software	SBus 266	PCI 266	Sbus 1000	PCI 1000	VME 1000
	Switch		Adapters				
Description and Function	8 or 16 Port Fibre Channel Fabric	Downstream Switching Architecture Providing upto 3000 Ports	Workstation Fibre Channel Adapter	PC or Workstation Fibre Channel Adapter	Workstation Fibre Channel Adapter	PC or Workstation Fibre Channel Adapter	Workstation VME Crate Adapter
FC-0	1062.5 Mbps	266 Mbps	266 Mbps	266 Mbps	1062.5 Mbps	1062.5 Mbps	1062.5 Mbps
Half or Full Duplex	Full All Classes	Full All Classes	Half-Class 1 Full-Class 2&3	Half-Class 1 Full-Class 2&3	Half-Class 1 Full-Class 2&3	Half-Class 1 Full-Class 2&3	Half-Class 1 Full-Class 2&3
Frame Size Supported	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B
Availability	Q2, 1995	December, 1994	Q1, 1995	Q1, 1995	Q2, 1995	Q2, 1995	Q3
Profile	N/A	N/A	Link Encapsulation and Direct Channel	Link Encapsulation and Direct Channel	Link Encapsulation and Direct Channel	Link Encapsulation and Direct Channel	Link Encapsulation and Direct Channel
Interface	FCS 3.0 & 4.2	FCS 3.0 & 4.2	FCS 3.0 & 4.2 SBus	FCS 3.0 & 4.2 PCI	FCS 3.0 & 4.2 SBus	FCS 3.0 & 4.2 PCI	FCS 3.0 & 4.2 PCI
Incompatibility TCP/IP SPX/IPX	FCS 3.0 or 4.2	FCS 3.0 or 4.2	Solaris 2.4 SUN OS 4.1	Solaris TBD	Solaris 2.4 SUN OS 4.1	Solaris 2.4 TBD	IR, VxWorks (planned)
Sustained I/W Demonstrated	Class1:100 MB/s Class2, 3: 20 MB/s Per Port	Class1: 25 MB/s Class2, 3: 5 MB/s Per Port	Host Dependent TBD	Host Dependent TBD	Host Dependent TBD	Host Dependent TBD	Host Dependent TBD
Classes Supported	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix
Advanced Loop Support	NO	NO	NO	NO	NO	NO	NO
Price	Not Available	Provided Upon Request	Not Available	Not Available	Not Available	Not Available	Not Available

Point of Contact: Rob Davis (612)932-4000, rob@anchor.com

October 6, 1994

## Anchor Communications Fibre Channel Product Summary

	CXT 250 D1	CXT 250 DM	CXTWatch	EISA 250	MCA 250	VME/64 250	HIPPI 250	FCS 250 Router	VHSCI
	Switches			Adapters				Legacy I/F	ASIC
Description and Function	8 or 16 Port Fibre Channel Fabric	16, 32, 48, or 64 Port Fibre Channel Fabric	Fibre Channel Network Management S/W	PC or Workstation Fibre Channel Adapter	PC or Workstation Fibre Channel Adapter	VME Crate or Workstation Fibre Channel Adapter	HIPPI to Fibre Channel Gateway	Fibre Channel to Ethernet, Token Ring, FDDI, or ATM IP, IPX Router	FC-1 and FC-2 ASIC
FC-0	266 Mbps	266 Mbps	266 Mbps	266 Mbps	266 Mbps	266 Mbps	266 Mbps	266 Mbps	All
Half or Full Duplex	Full All Classes	Full All Classes	N/A	Half-Class 1 Full-Class 2&3	Half-Class 1 Full-Class 2&3	Half-Class 1 Full-Class 2&3	Full	N/A	Half-Class 1 Full-Class 2&3
Frame Size Supported	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	N/A	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B	Class1:0-2112B Class2&3:0-128B
Availability	In Production	In Production	In Production	In Production	In Production	November 1994	In Production	In Production	In Production
Profile	N/A	N/A	SNMP	Link Encapsulation and Direct Channel	Link Encapsulation and Direct Channel	Link Encapsulation and Direct Channel	Link Encapsulation and Direct Channel	Link Encapsulation	N/A
Interface	FCS 3.0 & 4.2	FCS 3.0 & 4.2	RS-232 and LAN	FCS 3.0 & 4.2 EISA	FCS 3.0 & 4.2 MCA	FCS 3.0 & 4.2 VME	FCS 3.0 & 4.2 HIPPI	FCS 3.0 & 4.2 Ethernet, Token Ring, FDDI, ATM (future)	FCS 3.0 & 4.2
Incompatibility TCP/IP SPX/IPX See Note 1	FCS 3.0 or 4.2	FCS 3.0 or 4.2	SNMP	PC: Novell SGI-IRIX	PC: Novell IBM AIX	SGI-IRIX VME Crate VxWorks (planned) IRIX	HIPPI Direct IP, Direct Channel	IP, IPX	N/A
Sustained I/W Demonstrated	Class1: 25 MB/s Class2, 3: 5 MB/s Per Port	Class1: 25 MB/s Class2, 3: 5 MB/s Per Port	N/A	Host Dependent TCP/IP: 10-12 MB/s IPX: 5-16 MB/s Direct Channel: 14 MB/s	Host Dependent TCP/IP: 6-15 MB/s IPX: 5-16 MB/s Direct Channel: 10-23 MB/s		20 MB/s	Performance measurement in process. No reliable data at this time.	100% (1994)
Classes Supported	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	N/A	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix	Class 1, Class 2, Class 3, & Intermix
Advanced Loop Support	NO	NO	N/A	NO	NO	NO	NO	NO	NO
Price	Provided Upon Request	Provided Upon Request	Provided Upon Request	Provided Upon Request	Provided Upon Request	Provided Upon Request	Provided Upon Request	Provided Upon Request	Provided Upon Request

Note 1: Anchor Communications has demonstrated a heterogeneous TCP/IP network with SUN, HP, IBM, and SGI workstations plus a link to Ethernet and Token Ring via the Anchor FCS 250 Router. We have also demonstrated a mix of different platform running Novell Network.

Point of Contact: Rob Davis (612)932-4000, rob@anchor.com

October 6, 1994



**AT&T Microelectronics**

**FCA Member**

Mbps	PROD.#	TYPE	PROD DESCRIPTION	AVAILABLE
1, 2	ATT1409A	O	Optical Link Card/LED-ST Connector	Now
1, 2	ATT1408N	O	Optical Transceiver/LED-SC Connector	Q4 93
G	ATT1238A	O	Xmtr module	Now
G	ATT1318A	O	Rcvr module	Now
G	ATTDA204AHG1		Xmtr/Rcvr w/ENDEC	Now
G	ATTDA205AHG			
G	ATTDA208AHG1		Xmtr/Rcvr	10/93
G	ATTDA210AHG			
S	ATTDA202AHG1		Xmtr/Rcvr	***
G	ATTDA203AHG			
G	ATTDF200	N	Adapter Evaluation Card	10/93
G	ATTDF201	I	IC Evaluation Card	8/93

\*\*\* To get additional information on these products, please feel free to contact the following AT&T representative(s): Jay Sherley (215) 439-5726

**AT&T/NCR Microelectronics**

Mbps	PROD.#	TYPE	PROD DESCRIPTION	AVAILABLE
1, 2	NCR85C266	I	Xmtr/Rcvr	Now

**Augment Systems, Inc.**

Mbps	PROD.#	TYPE	PROD DESCRIPTION	AVAILABLE
2	AL303	N	S Bus Adapter/Controller	Q1 1994
2	AL301	N	NUBUS Adapter/Controller	Q1 1994

\*\*\* To get additional information on these products, please feel free to contact the following Emulex representative(s): Harry Ives (617) 275-4461

**Cypress Semiconductor**

**FCA Member**

Mbps	PROD.#	TYPE	PROD DESCRIPTION	AVAILABLE
2	CY7B923/ CY7B933	I	Hotlink Xmtr/Rcvr & Server w/Encoder & BIS ***	***
2	CY9266-F/C	:	Hotlink evaluation board for fibre or copper	***

\*\*\* To get additional information on these products, please feel free to contact the following Cypress representative(s): Ed Silva (408) 943-2693

**Emulex**

**FCA Member**

Mbps	PROD.#	TYPE	PROD DESCRIPTION	AVAILABLE
1, 2, 5, G	**prodno	N	Adapter/Controller	Q2 1994

\*\*\* To get additional information on these products, please feel free to contact the following Emulex representative(s): Charles Bazzar (714) 513-8154

**Finisar**

Mbps	PROD.#	TYPE	PROD DESCRIPTION	AVAILABLE
1, 2, 5, G	**prodno	O	Shortwave laser Xmtr/Rcvrs	***
1, 2, 5, G	GLA-3000	I	Tester	***

\*\*\* To get additional information on these products, please feel free to contact the following Finisar representative(s): Jerry Rawls (415) 364-2722 or Fax: (415) 364-3041

**Force, Inc.**

Mbps	PROD.#	TYPE	PROD DESCRIPTION	AVAILABLE
1.2, 5, G	2556T	O	1.5 Gbps FO Transmitter	Now
1.2, 5, G	2556R	O	1.5 Gbps FO Receiver	Now
1.2, 5, G	2666T	O	1.3 Gbps FO Transmitter	Now
1.2, 5, G	2666R	O	1.3 Gbps FO Receiver	Now
1.2, 5, G	2667T	O	2.5 Gbps FO Transmitter	Now
1.2, 5, G	2667R	O	2.5 Gbps FO Receiver	Now
1.2	2581T	O	270 Mbps FO Transmitter	Now
1.2	2581R	O	270 Mbps FO Transmitter	Now
1.2, 5, G	2644T	O	1.1 Gbps SW FO Transmitter	Q4 93
1.2, 5, G	2644R	O	1.1 Gbps SW FO Receiver	Q4 93
1.2, 5	2706T	O	Militarized, Low Profile FO Transmitter	Q4 93
1.2, 5	2706R	O	Militarized, Low Profile FO Receiver	Q4 93

\*\*\* To get additional information on these products, please feel free to contact the following Force Representative(s): David Goff (VP Eng.) or Wendell Henster (VP Marketing) at (703) 382-0462

**Fujikura Technology America**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
5	FLC-531-200-0	O	Fibre Channel serial/OFC module	***
5	FCS-531-0	I	Fibre Channel Laser/PD Submodule	***
5	FCP-531-200-0		Fibre Channel parallel data link	***
	C015999		SC connectors for multimode	***
	C031904/C016047		SC connectors for singlemode	***
	C031901		Duplex SC connector clip	***
	C021450		Duplex SC bulkhead adaptor	***
	C024562		SC Duplex receptacle	***
	C001928		SC connector cleaning cartridge	***
	C015261		SC connector assembly kit	***
			SC Polishing machine	***
	SF-PC-ZR-SF-PC-ZR-PZR-0002		Multimode Duplex SC cable (2 meter)	***
	SF-PC-ZR-SF-PC-ZR-PZQ-0002		Singlemode Duplex SC cable (2 meter)	***

\*\*\* To get additional information on these products, please feel free to contact the following Fujikura representative(s):  
 Milan Morando (408) 988-7408 Tom Robinson (803) 433-3322

**GEC Messer Semiconductor**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
1	P1480	I	High Speed Address Filter (CAM)	Now

\*\*\* To get additional information on these products, please feel free to contact the following GEC Messer representative(s):  
 Phil Welch (408) 439-6046 or Fax: (408) 438-5576

**Hewlett Packard Canadian Networks Operations**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2, G	**prodno	F	Class 1, 2 Switch	***

\*\*\* To get additional information on these products, please feel free to contact the following Caeser representative(s):  
 Kumar Malivalli (416) 490-3331

**Hewlett Packard Optical Components Division**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2	HOLC-0266	O	Shortwave laser daughter card	Now
5, G	HDMP-1514/12	I	Tx/Rx Clip Pair	Q4-1994
2, G	TBD	C	Daughter Card	Q4-1994
1, 2	HPBR-4501/2	C	Optical Transceiver	Q2-1994

\*\*\* To get additional information on these products, please feel free to contact the following Hewlett Packard representative(s):  
 Ron Whitte: (408) 435-2123 or Fax: (408) 435-6506

**FCA Member**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2	**prodno	N	EISA Adapter for S700	Q1 1994
G	**prodno	N	Adapter for S700 & S800s	Q1 1993

\*\*\* To get additional information on these products, please feel free to contact the following Hewlett Packard representative(s):  
 Don Wilson (408) 447-2388

**IBM Rise Systems/6000 Divisions**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2, G	**prodno	N	MCA Host Adapter for RS/6000	***
2, G	**prodno	F	Switch for RS/6000	***

\*\*\* To get additional information on these products, please feel free to contact the following IBM representative(s):  
 Jeff Silva (312) 638-3700

**IBM AS/400 Division**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2	CLC-264	O	FC-0/byte to Light, 10 bit data path	Now
2	CLM-266	O	FC-0/byte to Light, 10 bit data path	***
3	CLC-531	O	FC-0/byte to Light, 10 bit data path	Now
3	CLM-531	O	FC-0/byte to Light, 10 bit data path	***
G	CLM-1063	O	FC-0/byte to Light, 20 bit data path	***

\*\*\* To get additional information on these products, please feel free to contact the following IBM representative(s):  
 Steve Sibley (507) 253-2943

**Interphase Corporation**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2, G	**prodno	N	SBus Adapter (Fiber & Copper)	Q1 1994

\*\*\* To get additional information on these products, please feel free to contact the following Interphase representative(s):  
 Ernest Godwin (214) 919-9122

**Jaycor, Inc.**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2, G	**prodno	N	SBus Adapter	Q1 94

\*\*\* To get additional information on these products, please feel free to contact the following Jaycor representative(s):  
 Ten Parsh (619) 535-3174

**Hewlett Packard Informative Networks Division**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2	**prodno	N	EISA Adapter for S700	Q1 1994
G	**prodno	N	Adapter for S700 & S800s	Q1 1993

\*\*\* To get additional information on these products, please feel free to contact the following Hewlett Packard representative(s):  
 Don Wilson (408) 447-2388

**FCA Member**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2, G	**prodno	N	SBus Adapter (Fiber & Copper)	Q1 1994

\*\*\* To get additional information on these products, please feel free to contact the following Interphase representative(s):  
 Ernest Godwin (214) 919-9122

**FCA Member**

Model	PROD.#	TYPE	PROD. DESCRIPTION	AVAILABLE
2, G	**prodno	N	SBus Adapter	Q1 94

\*\*\* To get additional information on these products, please feel free to contact the following Jaycor representative(s):  
 Ten Parsh (619) 535-3174



**Nestor Incorporated**

<u>Mfrs</u>	<u>PROD.#</u>	<u>TYPE</u>	<u>PROD. DESCRIPTION</u>	<u>AVAILABLE</u>
G	**prodno	F	16 Port Switch/Router	1994

\*\*\* To get additional information on these products, please feel free to contact the following Nestor representative(s): N/A

**RADWAY International Ltd.**

FCA Member

<u>Mfrs</u>	<u>PROD.#</u>	<u>TYPE</u>	<u>PROD. DESCRIPTION</u>	<u>AVAILABLE</u>
2.5.G	FCCS-1004	N	4 slot modular Fibre-Channel router with Ethernet, Token-Ring and FDDI interfaces	***
2.5.G	FCCS-1012	N	12 slot modular Fibre-Channel router with Ethernet, Token-Ring and FDDI interfaces	***
2.5.G	FCIA-501	N	Fibre-Channel protocol (FCP) to fast & wide SCSI-II converter	***

\*\*\* To get additional information on these products, please feel free to contact the following RADWAY representative(s): Yeav Talgam +972-3 645-8515 or Fax: +972-3 645-8585

**SGS-Thomson**

<u>Mfrs</u>	<u>PROD.#</u>	<u>TYPE</u>	<u>PROD. DESCRIPTION</u>	<u>AVAILABLE</u>
1, 2	DMSC101	I	Serial DS-Link to Parallel converter	***
1, 2	DMSC104	F	Serial DS-Link 32-way dynamic routing crossover	***

\*\*\* To get additional information on these products, please feel free to contact the following SGS Thomson representative(s): Forrest Crowell (714) 957-6018 or Fax: (714) 957-3281

**Siemens Fibre Optic Components**

FCA Member

<u>Mfrs</u>	<u>PROD.#</u>	<u>TYPE</u>	<u>PROD. DESCRIPTION</u>	<u>AVAILABLE</u>
1, 2	RX-266T2E	O	Longwave LED Receiver	***
1, 2	TX-266T2E	O	Longwave LED Transmitter	***

\*\*\* To get additional information on these products, please feel free to contact the following Siemens representative(s): Sholto Van Doorn (201) 890-1606

**Sun Microsystems**

FCA Member

<u>Mfrs</u>	<u>PROD.#</u>	<u>TYPE</u>	<u>PROD. DESCRIPTION</u>	<u>AVAILABLE</u>
TBD	TBD	TBD	TBD	TBD

\*\*\* To get additional information on these products, please feel free to contact the following Sun representative(s): Bob Williamsen (415) 336-5335

**TriQuint Semiconductor**

FCA Member

<u>Mfrs</u>	<u>PROD.#</u>	<u>TYPE</u>	<u>PROD. DESCRIPTION</u>	<u>AVAILABLE</u>
1, G	GA9040A	O	Optical module (multi-mode fiber)	Now
:	GA9101/GA91021	I	Xmtr/Recv (FC / ATM / ESCON)	Now
:	FC-266	I	FC Xmtr/Recv / ENDEC	Now
:	FC-200	I	ESCON Xmtr/Recv / ENDEC	Now
:	FCDS-266FL	O/I	Fibre Channel development system with fiber optic interface card	Now
:	FCDS-265C	O/I	Fibre Channel development system with coax interface card	Now
:	FCC-266FL	N	Fibre Channel fiber optic interface card	Now
:	FCC-265C	N	Fibre Channel coax interface card	Now
G	GA9301/GA93021	I	Fibre Channel Xmtr/Recv	Q4 93
G	PC-1000	I	Fibre Channel Xmtr/Recv / ENDEC	Q4 93
G	FCDS-1000FL	O/I	Fibre Channel Development system with Fiber Optic interface card	Q4 1993
G	FCC-1000FI	I	Fibre Channel fiber optic interface card	Q4 1993

\*\*\* To get additional information on these products, please feel free to contact the following TriQuint representative(s): Sunil Sanghani (408) 982-0900

**Vitesse Semiconductor**

<u>Mfrs</u>	<u>PROD.#</u>	<u>TYPE</u>	<u>PROD. DESCRIPTION</u>	<u>AVAILABLE</u>
G	VSC710:	I	Fibre Channel Xmtr	Now
G	VSC710o	I	Fibre Channel Recv	Now
G	VSC710?	I	Endec	Q4 94
G	VSC7100EXER	O/I	Fibre Channel exerciser/tester	Now
G	VSC7100EVAL	O/I	Fibre Channel development system with coax interface and optional optical interface	Now

\*\*\* To get additional information on these products, please feel free to contact the following Vitesse representative(s): Howey Chin (408) 730-3048 or Brent Butler (805) 388-7468

Every exponential with a positive  
exponent is initially small!

$$a e^{bt}$$

Fibre Channel is Growing!

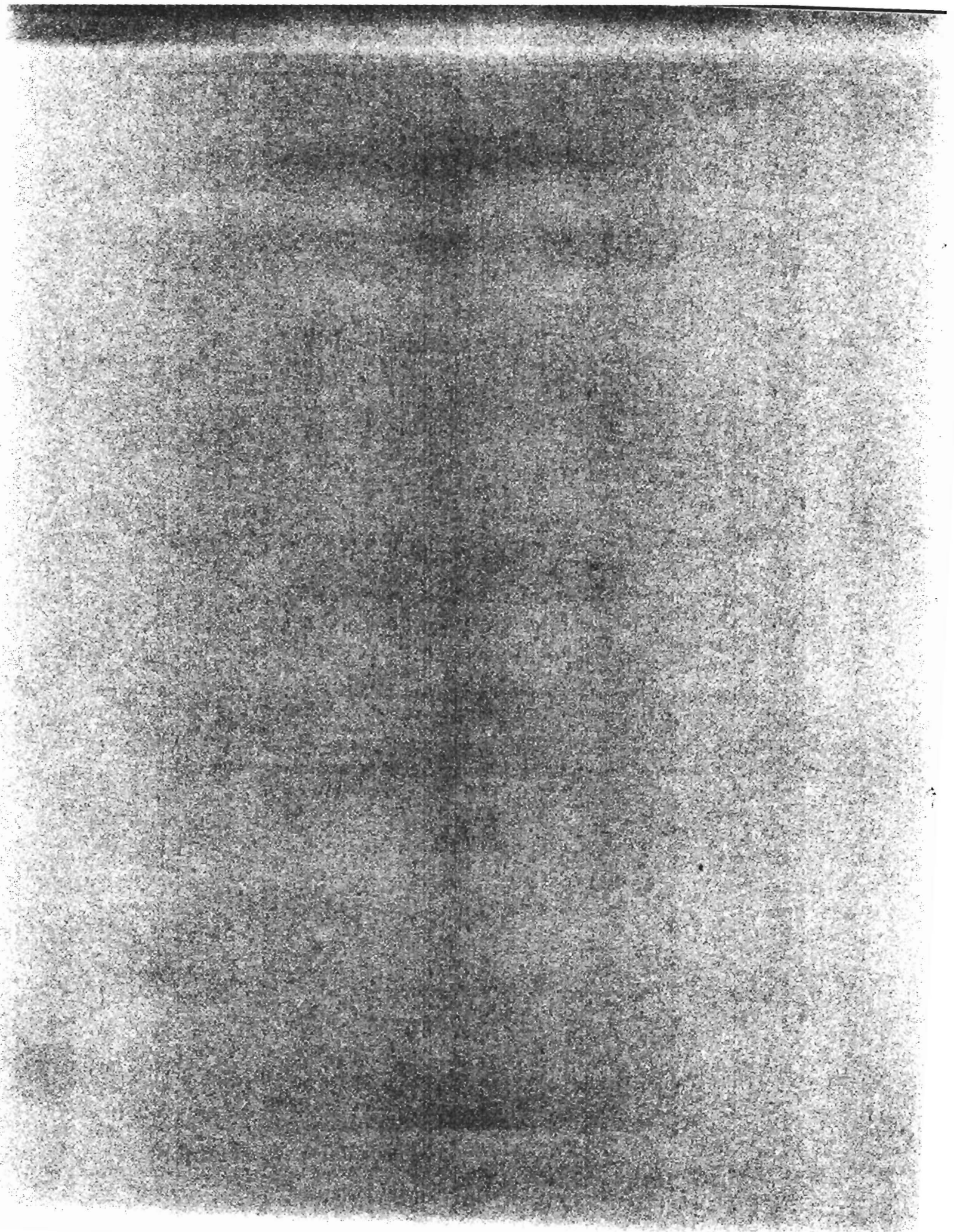
file=fctables.tex

### **S4-3**

## **"Overview of SCI Integrated Circuits & Board Products"**

**(Volker Lindenstruth - LBL)**

It will be presented which SCI chips are currently available and what their performance figures are. The features and availability of SCI board products and interfaces will be outlined together with their related software. Features of SCI network simulation software will be discussed. Typical problems related to interfacing between unified transaction busses and split transaction busses like SCI will be discussed using a concrete implementation as an example.



# Overview of SCI Integrated Circuits & Board Products

Volker Lindenstruth  
Lawrence Berkeley Laboratory  
1 Cyclotron Rd. M/S 50D Berkeley, CA 94720  
lindenstruth@lbl.gov

## Abstract

This paper describes SCI node chips and board products. It will describe both today's existing devices and SCI hardware that will be available within the next few months.

The Dolphin SBUS board will be used as an example to discuss the performance of both the CMOS NodeChip and the SBUS board itself.

## 1 Introduction

SCI is an approved IEEE standard [1] defining a high speed (1 GByte/sec) split transaction network. It merges the shared memory concept with a bus-like architecture. SCI supports transparent read and write transactions to or from any node in the whole system of up to 65000 nodes based on a 64-bit address. Many different nodes may share cached copies while the network ensures cache coherence throughout the network.

This paper will outline what SCI hardware is available now or about to become available in the near future. Due to the number of different devices it is not possible to go into any detail but just give a rough overview. The interested reader should use the references to follow up on any of the mentioned devices.

## 2 SCI Chips / NodeChips

SCI NodeChips are node interface chips that implement the link level protocol and a back-end bus interface. Figure 1 shows a sketch of a generic SCI NodeChip. All node chips have in common that there is a high-speed link part that handles the physical SCI packet stripping and forwarding and a back-end part that includes request and response queues. This back-end bus runs at a different clock rate than the SCI link.

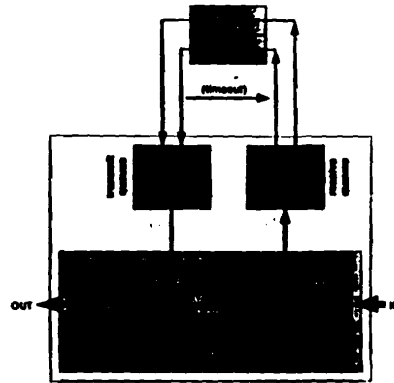


Figure 1: The functional blocks of an SCI NodeChip.

One important general feature of SCI NodeChips is that the link part implements a mini-router since it strips packets from the SCI link or forwards them to the next node depending on the target ID of the packet. This feature allows building small systems without any bridge or switch hardware by just connecting all nodes to one SCI ring. This implementation, however, is not recommended for large systems since the effective maximum throughput on a single SCI ring depends on the node to node data flow and saturates for typical scenarios at about 1.4GBytes/sec assuming 1GByte/sec link speed [2].

The SCI technology breakthrough was achieved in April 1993 when the first SCI packets were sent between two nodes using the Dolphin [3] GaAs NodeChip DST 501A. The GaAs NodeChip runs at 500MB/sec link speed. First tests at RD24, CERN [4] revealed a DMA node-to-node transfer rate of 114MBytes/sec. The design of the GaAs NodeChip was adopted by Convex and modified to be used in their high performance computers [5]. The next step

of the Dolphin/Convex collaboration was the Convex CMOS based cache coherent memory controller (CCMC) that interfaces with the Fujitsu/Convex GaAs NodeChip.

The next generation NodeChip developed by Dolphin was targeted for low cost, and consequently CMOS based. It supports, like the GaAs implementation, the transaction level of the SCI cache coherence protocol. Its link speed is 125MBytes/sec. One important new feature of the CMOS NodeChip is that it supports bridging functionality. Two NodeChips can be connected back-to-back without extra glue logic to form an SCI-SCI bridge. The link part of this chip also supports direct interfacing to the HP GigaLink Chips [6] allowing optical transmission of SCI packets. The 4 watt CMOS NodeChip has been available in quantity since May 1994 from LSI logic (L64601)

There are two other SCI NodeChips in the queue. Both are expected early in 1995. One SCI NodeChip is the Dolphin LinkController. It is a CMOS device and supports link speeds of 200MBytes/sec minimum. Dolphin chose a different conceptual approach for the design of the LinkController. The SCI transaction level (read, write) and subaction level (echoes, request and response packets) were split. The Dolphin LinkController supports only the SCI subaction level. This freed a lot of real estate on the chip since the SCI cache coherence-related logic no longer had to be implemented. This real estate was used to implement three receive and three send buffers. The back end bus (B-Link) of the LinkController was implemented as a multimaster packet-mode bus, allowing several LinkControllers to communicate to form a complex switch.

The other SCI NodeChip that is about to become available is the Unisys/Vitesse Datapump. The Datapump is a GaAs chip running at full SCI speed of 1 GigaByte/sec. It supports the SCI subaction level as well. The SCI link input and output signals of the Datapump are LVDS [7] compliant. There are four receive and four send queues implemented on the Datapump, two each for request and response packets. Due to the high internal clock rate of 500MHz the Datapump implements very low latency for both packet forwarding through the bypass FIFO at the link part and packet forwarding through the receive/send queues.

## 3 SCI Boards and Interfaces

With the availability of SCI interface chips, many projects were started to design SCI boards and interfaces. It is impossible to discuss them all in detail. Therefore I will show a list with brief descriptions and references to acquire detailed literature. Please note that I do not claim this list to be complete.

- NodeChip Tester [3] available  
Plug-in board for CMOS NodeChip supplying pods for HP logic analyzer
- LinkProbe [3][8] available  
VME-based SCI link-level tracer. It implements a Tracer Control Language allowing one to define complex filters.
- SCI Prototyping Board [3] available  
This VME board implements the NodeChip with the link connectors. The only use of the VME connectors is to supply the board with power.
- SBUS Interface [3] available  
Sun SBUS interface supporting both transparent read/write and DMA transactions. The SunOs driver supports shared memory, pipe scenarios and TCP/IP links through SCI.  
The SBUS interface is also available as a packet-mode evaluation board allowing one to create SCI packets under processor control to evaluate, for example, the cache coherence protocol.
- VME Bridge [3] available Q1 95  
VME64 interface supporting both transparent crate to crate read/write transactions and DMA block move transactions through the SCI back bone.
- ATM-SCI Interface [3][9] announced  
Interface supports direct AAL5 to shared memory transactions and vice versa.
- Apple Quadra Interface [10] available  
The 68040 Apple Quadra interface maps the 16 byte cache line on SCI rsb16 and wsb16 transactions. These are the only transactions supported.
- PowerPC Interface [10] announced  
The Apple PowerPC 601-SCI interface supports both DMA and transparent SCI transactions. It implements the PowerPC crit-

ical hexlet first transactions and the Apple patented mechanism supporting the RISC LoadLocked/StoreConditional transactions over an arbitrary interconnect.

- SCI/HIC Interface [11] announced  
HIC is a new proposed IEEE Standard P1355 for Heterogeneous Interconnect (HIC) (Low Cost, Low Latency, Scalable Serial Interconnect for parallel system construction). The Verilog model of the interface is designed.
- SCI-CHI [12][8] announced  
The SCI - CERN Host Interface allows connecting Fastbus systems to the SCI network.
- SCI VideoRAM [8] announced  
The SCI VideoRAM memory is an implementation of a SCI cache coherent memory using VideoRAM technology.
- SCI DRAM [13] announced  
The SCI DRAM is a VME dual ported memory allowing read/write transactions from both VME and SCI.
- MC68040 DPM [4][14][15] available  
This is a mezzanine board for the CES 8224 68040 VME processor. It implements a dual ported memory as buffer for the SCI transactions and a DMA controller.
- SCI DSP Bridge [4] announced  
This interface is designed as a general DSP-SCI interface implementing a key addressing scheme to trigger SCI requests. It is implemented first using the TMS320C40 as its basis. However it is generic enough to be easily adapted to other DSP architectures.

#### 4 A Concrete Example

The Dolphin SCI SBUS board will be used as an example in this section to outline some implementation and performance aspects.

One very interesting and unique feature of an SCI network is its ability to have a shared memory somewhere in the system that is transparently accessible by any node in the system. Within the framework of a UNIX operating system that allows paging and swapping of memory regions this is a much more compli-

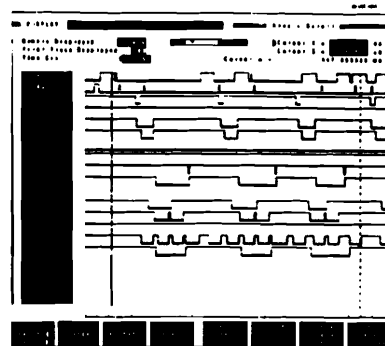


Figure 2: Logic analyzer trace SCI SBUS board receiving shared memory moves simultaneously from two other SCI nodes.

cated task than it may appear. First of all, a shared memory has to be created and locked in memory ensuring that a transparent read coming at any time from a remote SCI node does not hit a page fault. In case of SBUS, physical I/O is performed in virtual DMA address space. Consequently, the appropriate page table entries have to be set up correctly before any remote node could access the shared memory. These functions are supported by the kernel at the driver level. Therefore it was required to support the SBUS board with a driver even for just creating and mapping a shared memory. It is not possible to simply force a shared memory to a fixed address and use this address by definition. Consequently a mail-box scheme had to be implemented that allowed remote nodes to request the base address of a given shared memory, identified by a token.

Figure 2 shows a logic analyzer trace of an SBUS board receiving multiple DMA moves. The setup consisted of three nodes, where two nodes were simultaneously writing to the system with the analyzer connected. This many-to-one scenario is typical for data acquisition systems. The SCI transaction used here was dmove64. The following critical latencies were observed with respect to the CMOS NodeChip running at 62.5 MHz:

Latency through bypass FIFO:	224 ns
Latency IFLAG to SREQ:	800 ns
CBUS dmove64 transaction time:	768 ns
(NodeChip low CBUS priority)	

These numbers indicate a theoretical dmove receiving bandwidth of about 30MB/sec assuming no latency write acknowledges.

However in order to complete the transaction the SBUS board has to become SBUS master and write the 64-byte packet to the requested virtual DMA address. This can be clearly seen in Figure 2. It takes the SBUS board about 2.2µs (with respect to the leading edge of SBSY) to complete the request.

Figure 2 shows another interesting feature – the behavior of the NodeChip if two packets arrive close to each other. The third dmove64 packet (IFLAG) arrives while the second packet is still being transferred at the CBUS side. However it is obviously buffered resulting in the third CBUS transaction. This allows the SBUS board to be able to receive data. Packets arriving at a higher speed would be busy-retried by the NodeChip. The second and third SBSY cycles are 3.2 µs apart. This corresponds to a maximum receiving data rate of 20MB/sec.

#### Acknowledgements

I want to thank all mentioned groups and especially Knut Aines, Hans Mueller, and Bernhard Skaali for their excellent support. Most transparencies presented are courtesy of Knut, Hans, and Bernhard.

My work is supported by the German Humboldt program.

#### References

- [1] SCI - Scalable Coherent Interface IEEE std 1596
- [2] Simulations with SCI as a data carrier in data acquisition systems  
E.H. Kristiansen Proceedings of the IEEE RT93 Vancouver
- [3] Dolphin Interconnect Solutions Inc, US sales office: 5201 Great America Parkway, Suite 320 Santa Clara, CA 94054, knal@netcom.com
- [4] RD24 Collaboration CERN  
joint spokespersons:  
Hans Mueller (Hans@sunshine.cern.ch)  
Andre Bogaerts (bogaerts@sunsciab.cern.ch)
- [5] Press Release Dolphin/Convex May 26, 1993
- [6] Hewlett Packard  
HDMP-1002/1004 or HDMP 1012/1014
- [7] Low Voltage Differential Signals IEEE 1596.3
- [8] Bernhard Skaali, Bin Wu  
Department of Physics, University of Oslo, N-0316 Oslo 3, Norway, t.b.skaali@fys.uio.no
- [9] Øivind Kure  
Norwegian Telecom Research, P.O. Box 83, N-2007 KJELLER, Norway, olvind@brage.nta.no
- [10] Donald N. North, Glen D. Stone  
Apple Advanced Technology Group,  
1 Infinite Loop, Cupertino, CA 95014  
north@apple.com, stone@apple.com
- [11] Ola Tørudbakken, E.H. Kristiansen  
SINTEF Instrumentation, Forskningsvn 1,  
P.O. Box 124, N-0314 Blindern Norway  
Ola.Torudbakken@si.sintef.no
- [12] STRUCK Holger Oelschlaeger  
hoe@struck.de
- [13] CIEMAT University Madrid, Particle Physics Div.  
Luciano Romero, romerol@dec.cimat.es
- [14] CES  
70 Route du Pont-Butin, CH-1213 Petit Lancy  
lounis@lancy.ces.ch
- [15] IHEP Institute for High Energy Physics, Russia  
anatoli@ts1.ihep.su

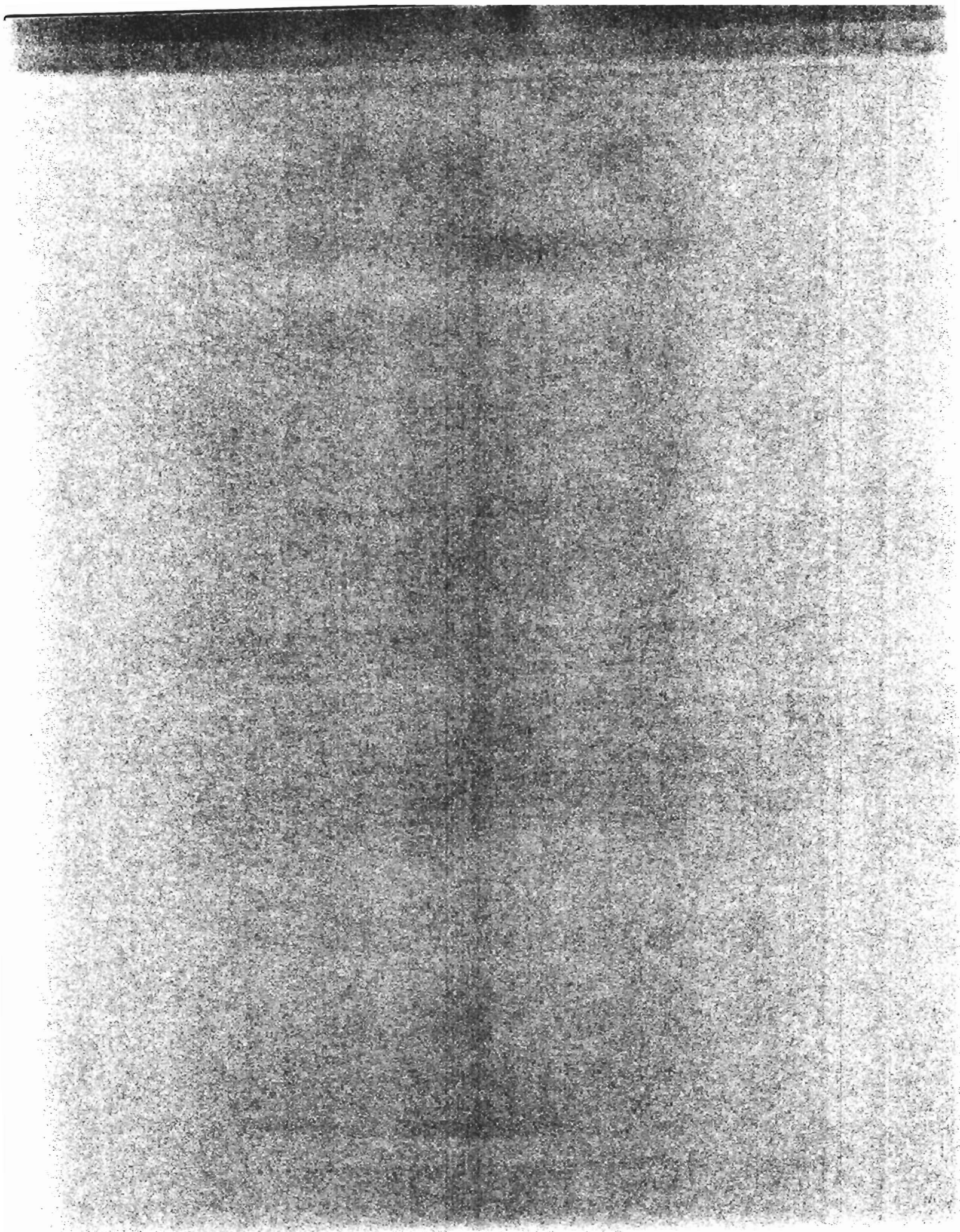
**S4-4**

**"Design of SCI-Class Interconnects"**

**(Wayne Nation - IBM)**

The development of high-bandwidth, low-latency, interconnect links will impact the structure of future computers. The ability to send more data, faster, over longer distances increases the feasibility of coupling more processors, memories, and I/O in system designs.

As advanced product technology, IBM has developed several testbeds for the evaluation of SCI-class interconnects. The first is a BiCMOS link chip that functions at 500 MHz and supports 2-byte-wide, 1 GByte/second SCI (Scalable Coherent Interface) signaling technology. The chip is used for the evaluation of high-speed logic, module, card, connector, and cable evaluation. A parallel fiber testbed has been built to study emerging parallel fiber technologies. The parallel fiber testbed uses the BiCMOS link chip as packet generator for a parallel fiber link. Also discussed are potential uses of these technologies in systems.





# Design of SCI-Class Interconnects

Wayne Nation  
nation@vnet.ibm.com

AS/400 Processor Architecture and Technology  
IBM Systems Architecture and Technology Division  
Rochester, Minnesota

Fermilab in Batavia, Illinois  
International Data Acquisition Conference  
October 26-28, 1994

**IBM**

1

## Agenda

- Problem definition and background
- SCI-Link chip description
- Chip status and applications
- Serial fiber products
- Further work

**IBM**

2

## Introduction

- **Problems:**
  - Higher bandwidth at low latency needed
  - Higher I/O bandwidth to more I/O devices needed
  - Multi-drop busses are approaching physical limits
- **Solution Approach:**
  - Use point-to-point asynchronous interconnect
  - Build packets to produce logical multi-drop busses where needed

**IBM**

3

## Introduction

### Background:

In 1990, the AS/400 Division established a team to evaluate and develop a high-speed interconnect

Chose IEEE P1596 SCI (Scalable Coherent Interface) as a starting point for evaluation

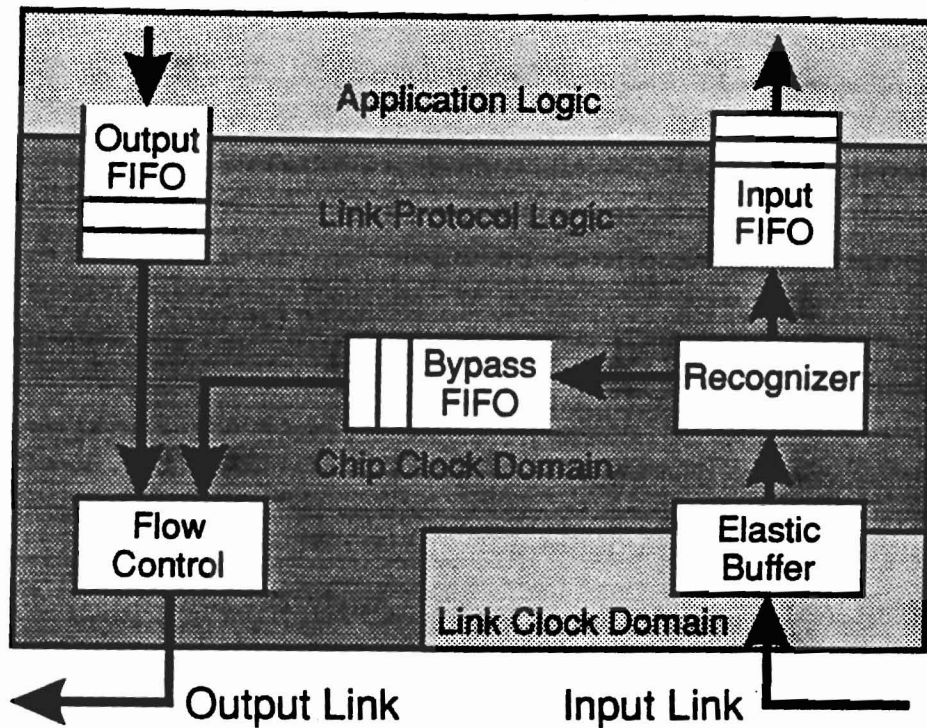
### Reasons for selection:

- Supported high-bandwidth low-latency link
  - 500 Mbit/sec/bit with 16-bit data, 1 flag bit, 1 clock bit
- Scalable to moderate distance (8 meters)
- Standard
- Relatively complete

**IBM**

4

## SCI-Link Architecture



IBM

5

## SCI-Link Architecture

Modification to SCI link protocol for data integrity:

- 32-bit CRC
  - Meet commercial system data integrity requirements
  - More suitable for use in fiber technologies
- Hardware detection/retry of lost/corrupted packets
  - Avoid software invocation on detected link errors
- Duplicate packet suppression
  - Avoid data integrity failure
- End-to-end packet acknowledgment
  - Tolerate failures in redundant topologies

IBM

6

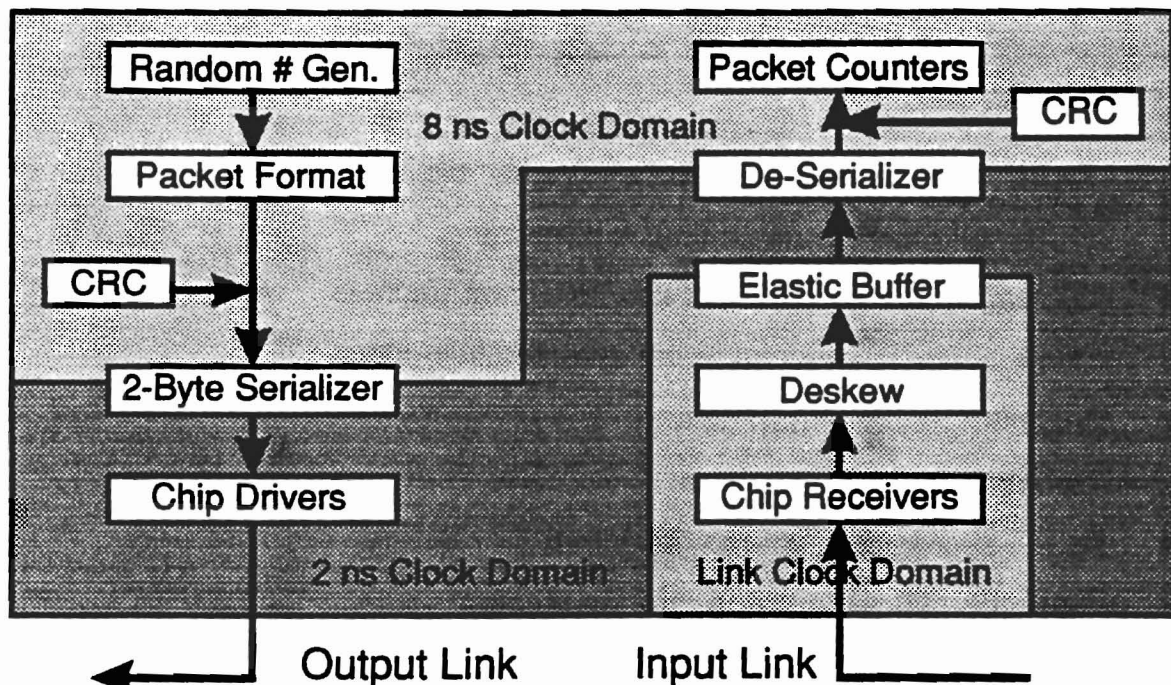
## SCI-Link Chip Description

- Chip objective: demonstrate design feasibility of 500 MHz
- Essential portion of SCI-Link architecture implemented
- Necessary 500 MHz function included:
  - Driver/Receiver logic
  - Elastic buffer
  - Bit-to-bit deskew logic
  - Packet aligner
- Ancillary function included:
  - Random packet-data generator
  - Statistics counters
  - Packet framing logic
  - Analyzer ports
  - 32-bit CRC

**IBM**

7

## SCI-Link Chip Description



**IBM**

8

## SCI-Link Testbed Description

- Chip:

Technology	0.8 micron BiCMOS
Die Size	12.7 mm x 12.7 mm
Voltage	3.6 Volts
Package	32 mm Ceramic SBC (BGA)
Chip Area	15%
Watts (500 MHz logic)	3 Watts
- Card: Standard AS/400 9x11" card, 6 signal, 4 power planes
- Cable: 5 meter cable, 50-pin connector (AMP)

**IBM**

9

## SCI-Link Chip Status

- Tape-out 6/93 / Power-on 1/94 / Debug complete 3/94
- Correct operation as slow as 125 MHz
- 500 MHz box-to-box (separate power and oscillators)
  - Error-free transmission of 8- to 512-byte packets on 5-meter cable
  - Functional bit-to-bit deskew
  - Functional elastic buffer
  - Passed Class A FCC EMC product-level testing

**IBM**

10

## SCI-Link Media Evaluation

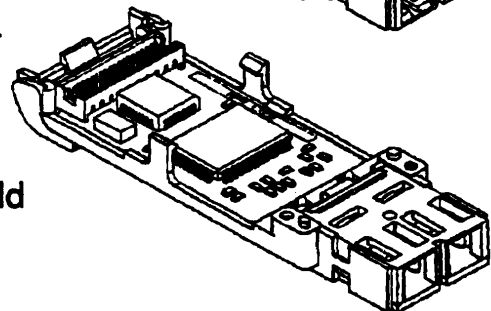
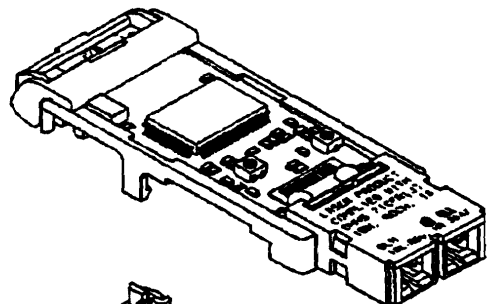
- Testbed for 500 MHz logic, card, copper cable/connector evaluation
- Testbed for emerging parallel fiber technologies
- OETC (ARPA-funded consortium): laser parallel fiber transceivers
  - 500 Mbit/sec/signal
  - Up to 100 meters
- JITNEY (NIST-funded consortium): LED parallel fiber transceivers
  - 500 Mbit/sec/signal
  - Up to 30 meters
- SCI-Link chip is basis for a testbed
  - Sophisticated packet generator
  - Bit error rate checker

**IBM**

11

## Serial Fiber Media Products

- 1063 Mb/s Optical Link Module (OLM)
- Supports FC, ATM, etc.
- Class 1 laser safety certification
- UL and CSA approved
- Average failure rate < 0.02% per KHour
- High integration, low power
- ~200,000 220/266 Mb/s OLCs in the field



**IBM**

12

## Further Work

- CMOS designs underway
- Lower-cost, lower-speed designs underway
- Architectural evaluation and definition continues
  - System coupling - alleviating the multi-drop bus bottleneck
  - I/O coupling - scaling I/O subsystem connectivity and performance
- Evaluating emerging parallel fiber technologies

**IBM**

13

## Conclusions

- SCI link protocol as starting point; modifications where needed
- Demonstrated 500 MHz link operation achievable in silicon
- Constructed testbed for evaluation of:
  - High-speed logic
  - Card designs
  - Connectors and cables
- Continued use of testbed for parallel fiber study
- Modifications continue to evolve
- Important enabling technology for future products

**IBM**

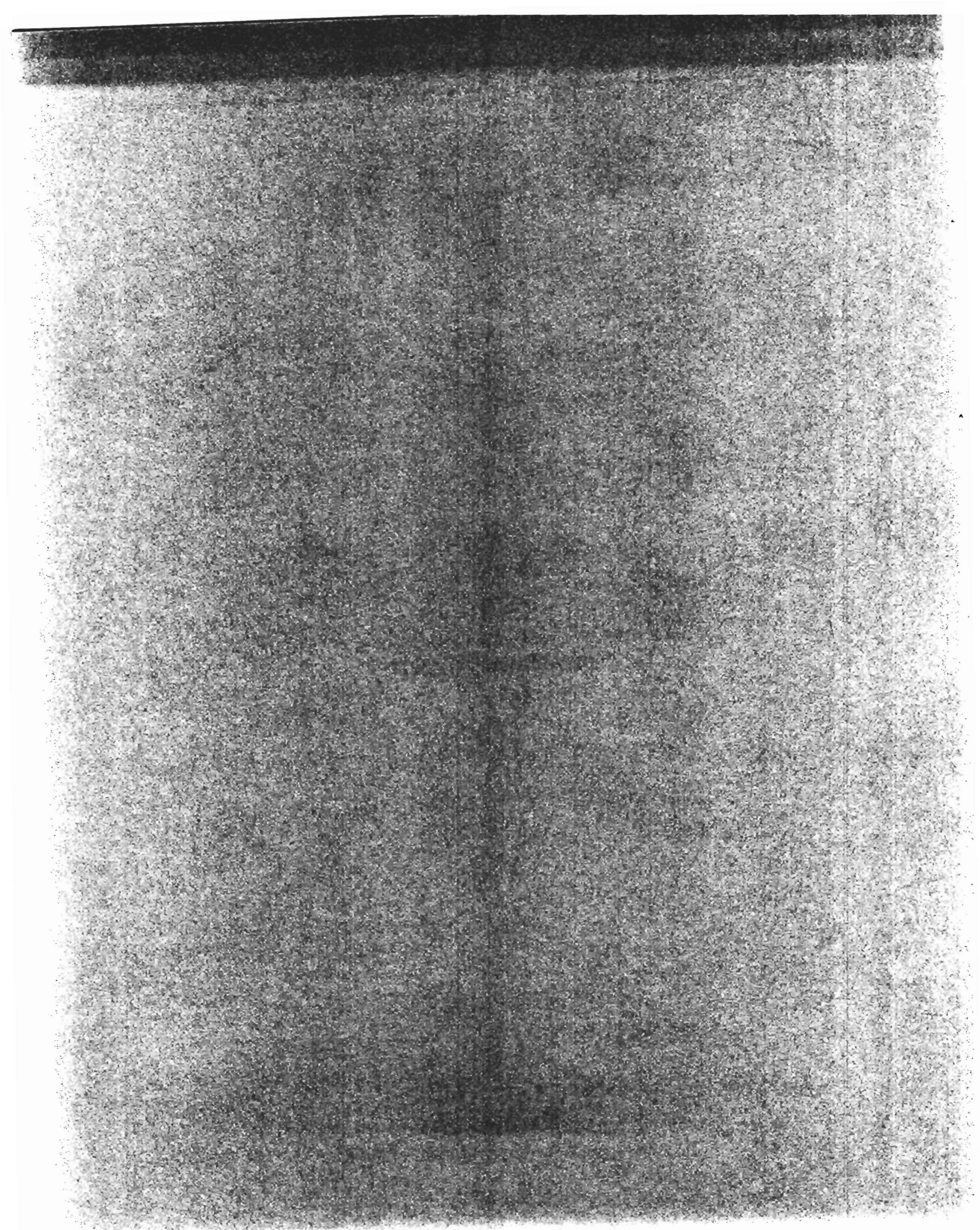
14





**S5-1**

**"ATM Switches for Telecommunications Applications"  
(Ian Mahood - Alcatel)**



# ATM Switches for Telecommunications Applications

Ian Mahood  
Alcatel Network Systems

Fermilabs

October 94



## ATM Switching Architecture

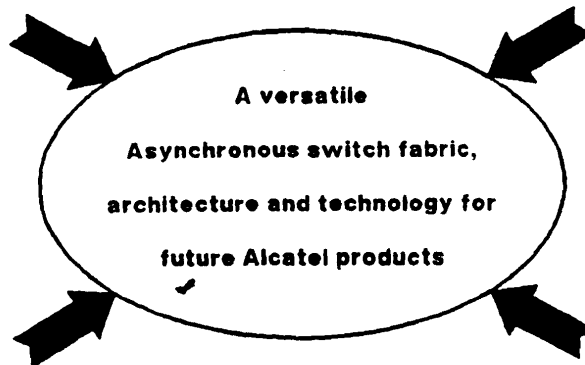
### Relevant Requirements and Objectives

#### INDEPENDENT FROM OUTSIDE ENVIRONMENT

- External Links
- External Protocols
- External Services

#### GENERIC SWITCHING DEVICES

- Large capacity matrix
- Generic module design
- Reduced set of chips and boards



#### CAPACITY RANGE FROM SMALL TO LARGE

- Expandable up to 16K links at 0.8 Erlangs
- Very high connection throughput

#### FUTURE SAFE TRAFFIC PERFORMANCE

- Very short connection set-up delay
- Virtually non blocking
- Very low cell loss ratio ( $10^{*-10}$ )
- Fault tolerant to failures in the switch

## Alcatel ATM Products

The Alcatel ATM Products have been designated as the

### Alcatel 1000 family

There are 2 primary members:

#### The Alcatel 1000AX

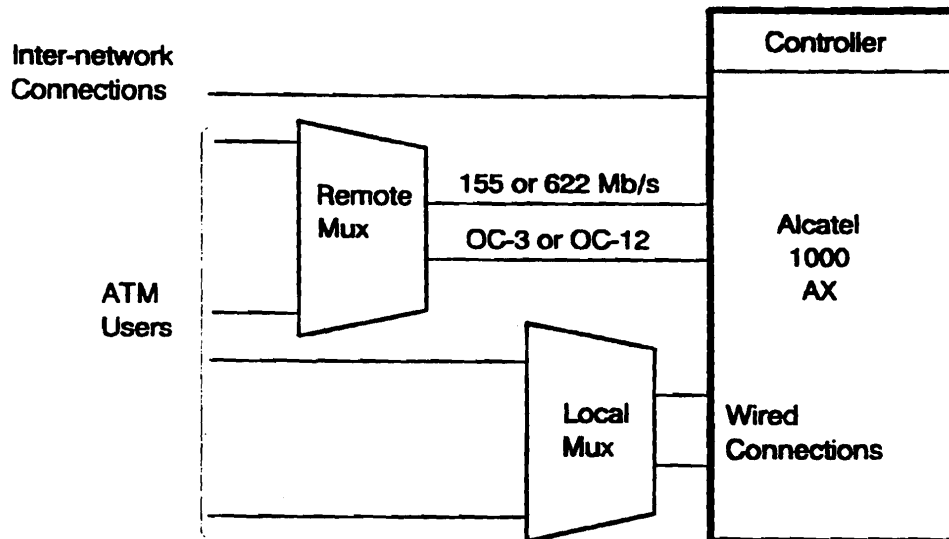
This is what is known as an ATM Crossconnect, i.e. it does not perform real-time switching functions. The 1000AX will be used as a high level "Tandem" or "Transit" backbone switch in the ATM network. Other likely applications include video switching in a "Video Dial Tone" environment, where near real-time connection meets the service requirement.

#### The Alcatel 1000/S12 and /E10

This is designed as a very large capacity real-time capable switch which can be installed as a growth addition to the existing families of voice switches, the S12 and E10 systems.



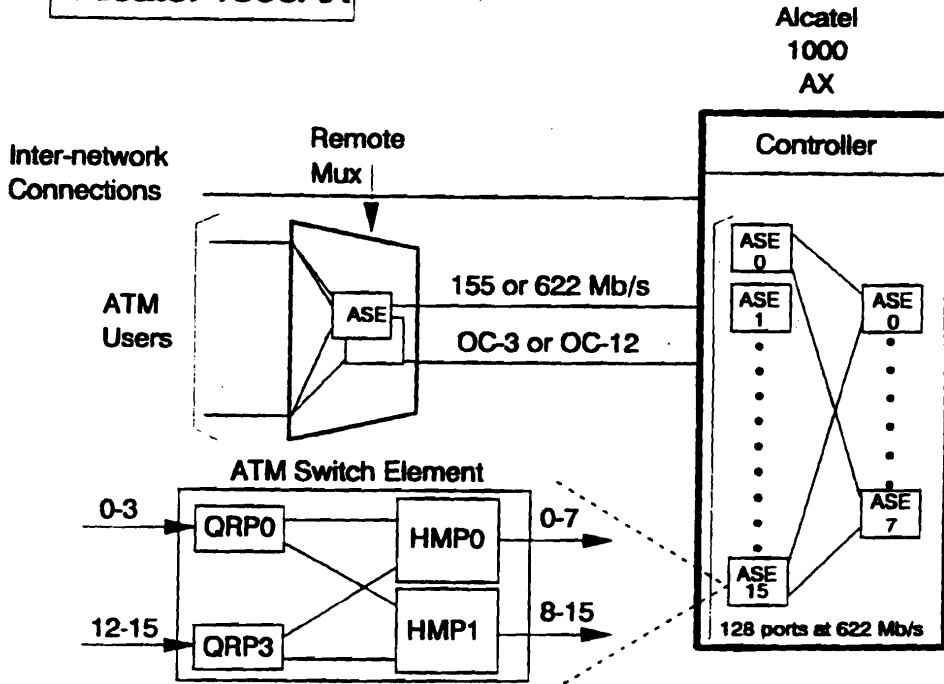
## Alcatel 1000AX



System Layout



# Alcatel 1000AX



System Structure

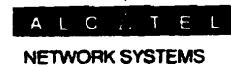
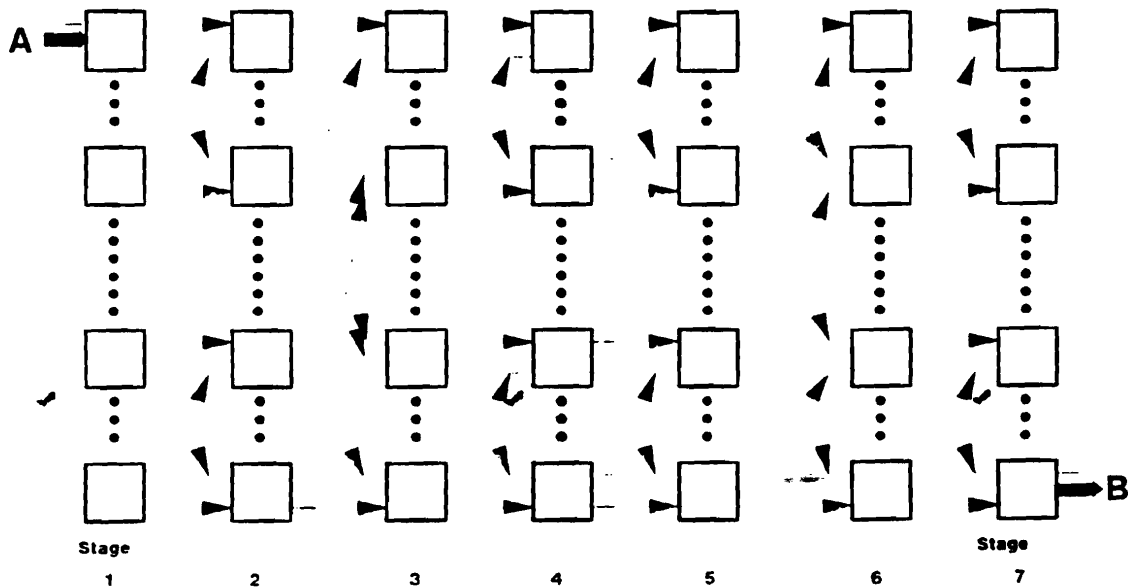


## Alcatel 1000

## MPSR Switching Principles

### A seven stage MPSR Switching Network (unfolded)

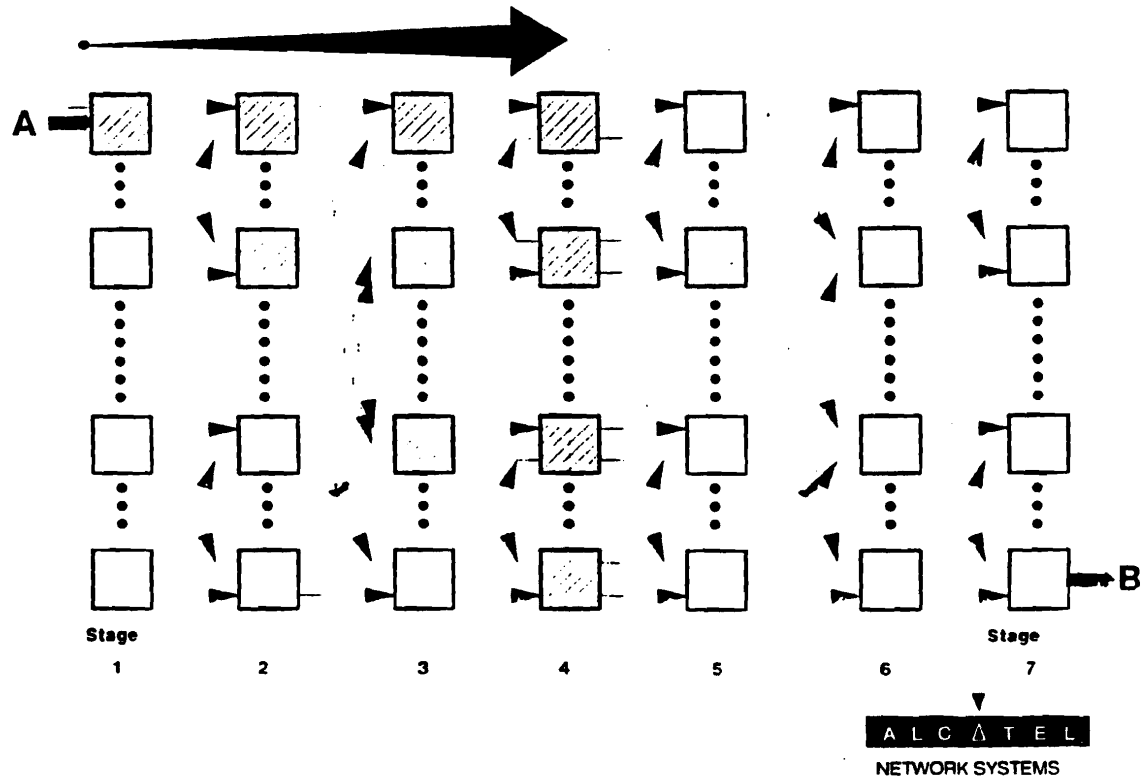
The self routing tag appended to a cell at input A contains its destination port address B



Alcatel 1000

MPSR Switching Principles

Cell Transfer from A to B: Randomized Distribution

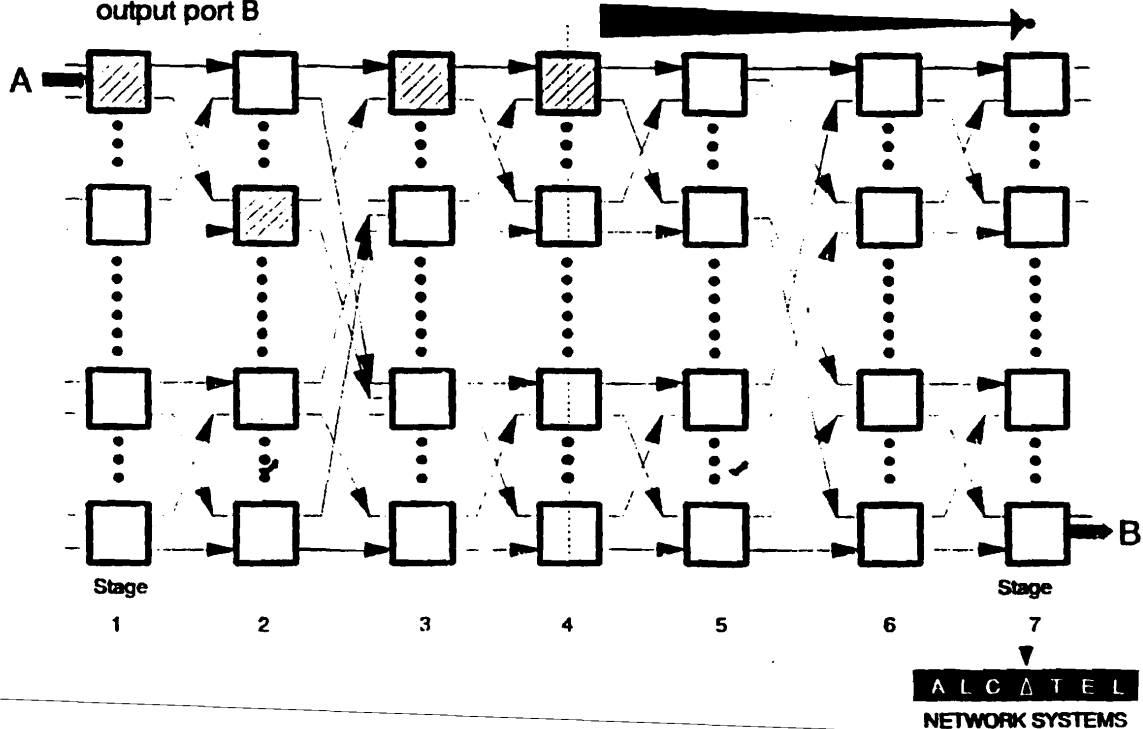


Alcatel 1000

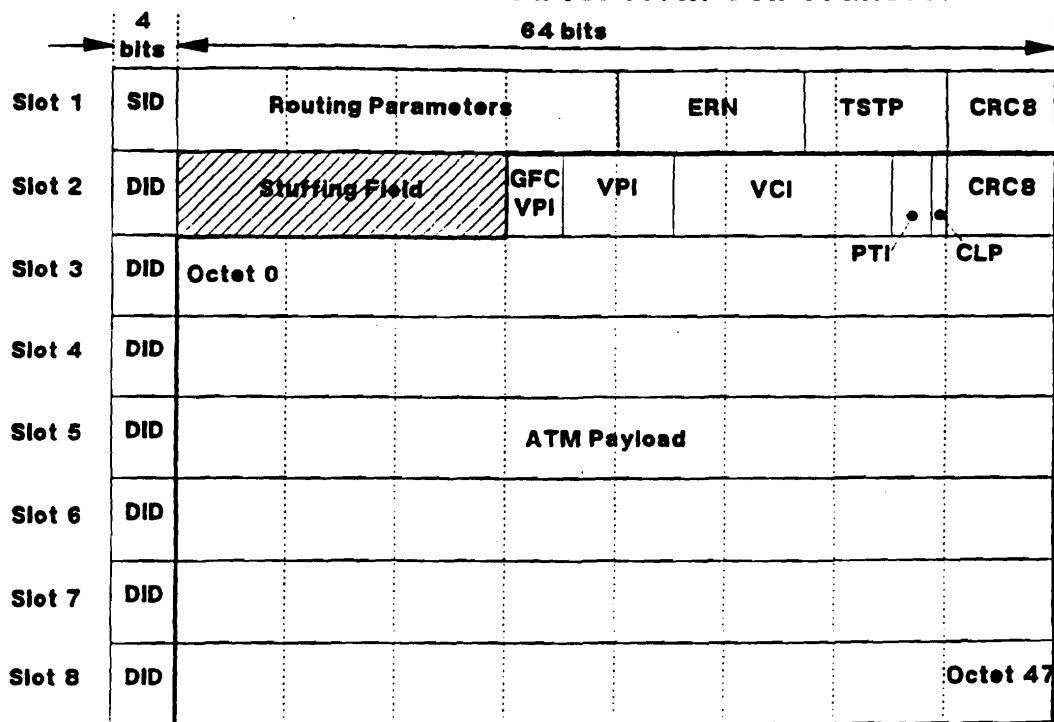
MPSR Switching Principles

Cell Transfer from A to B: Multipath Routing

From Randomly selected matrix in Stage 4, multipath routing to switch output port B



## Multislot Cell Format used for ATM Cell Transfer



DID

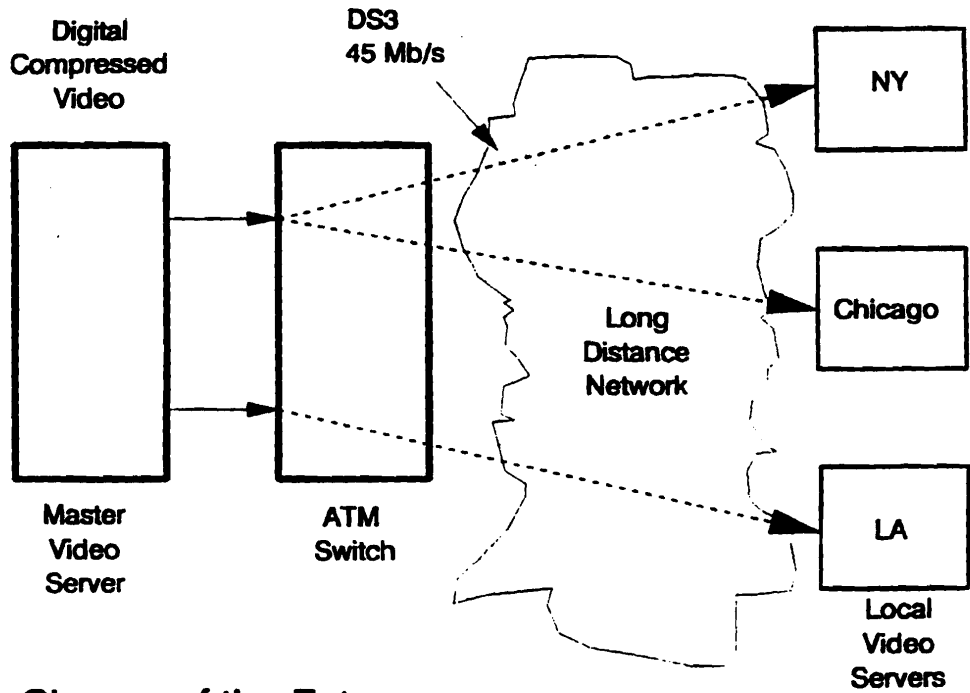
Alcatel 1000.

**SELF ROUTING OF INDIVIDUAL CELLS**

- \* No connection path needs to be selected, reserved, activated or released (connectionless operation)
- \* Considerable simplification of connection control and bandwidth management within the switch
- \* Significant increase in connection handling performance: short set-up times, high connection rate, no internal blocking

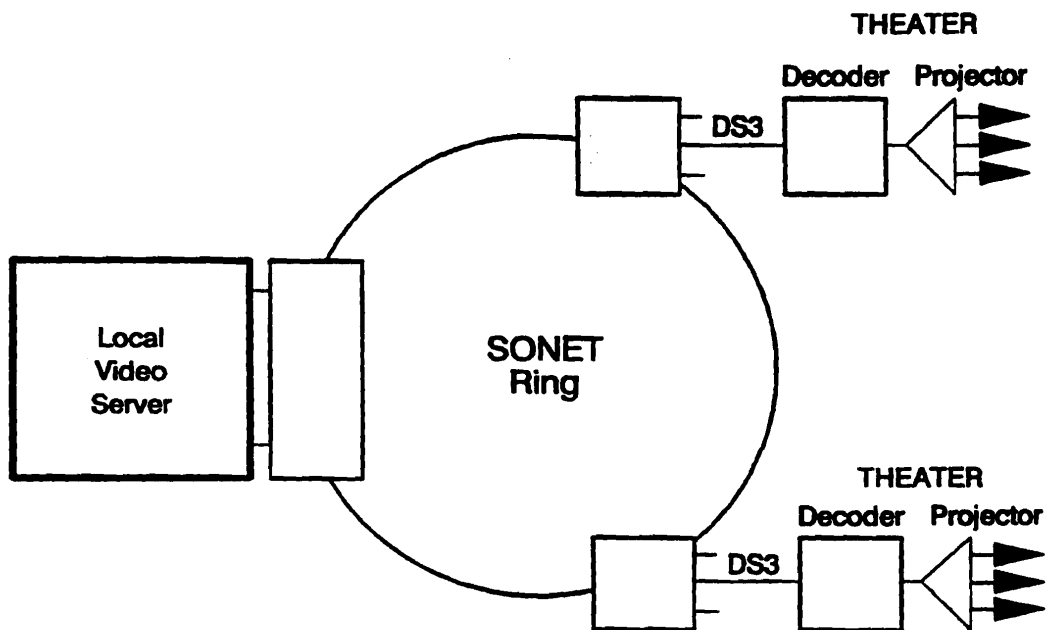
**MULTIPATH SELF ROUTING OF CELLS**

- \* Built in fault tolerance through path redundancy
- \* Full decoupling of internal transfer rate from external link bit rates and service cell rates
- \* Reduced sensitivity to user service mix and burstiness
- \* Increased throughput performance



**Cinema of the Future**

Movie Distribution

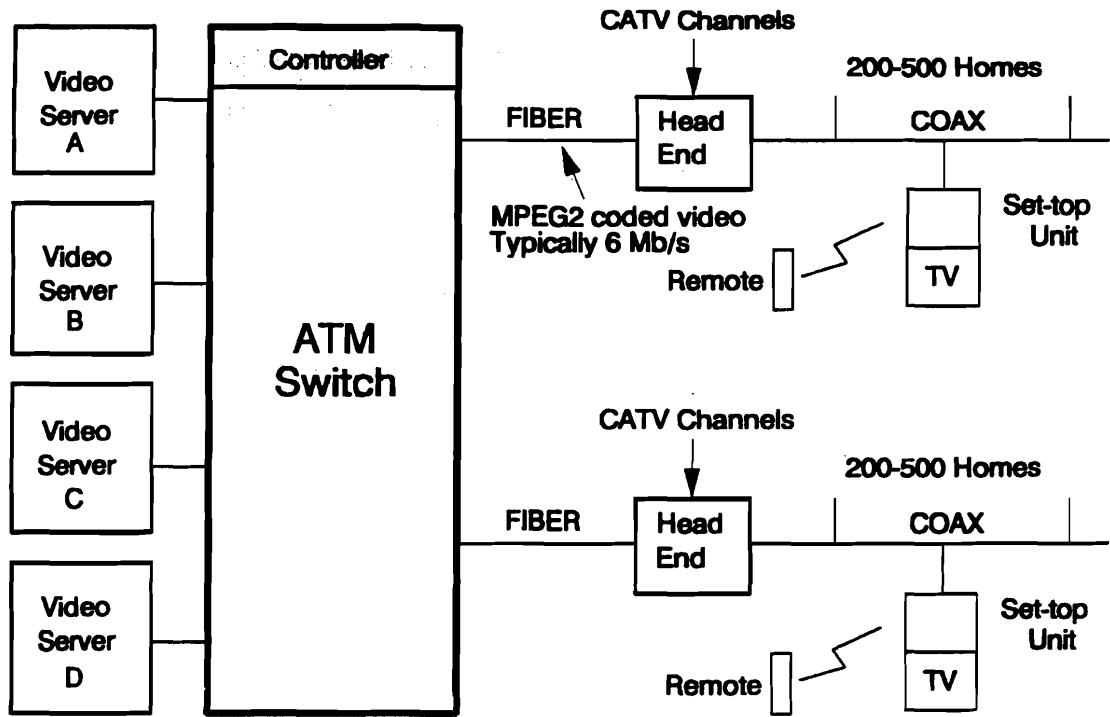


**Cinema of the Future**

Local Distribution







**Video on Demand Application**





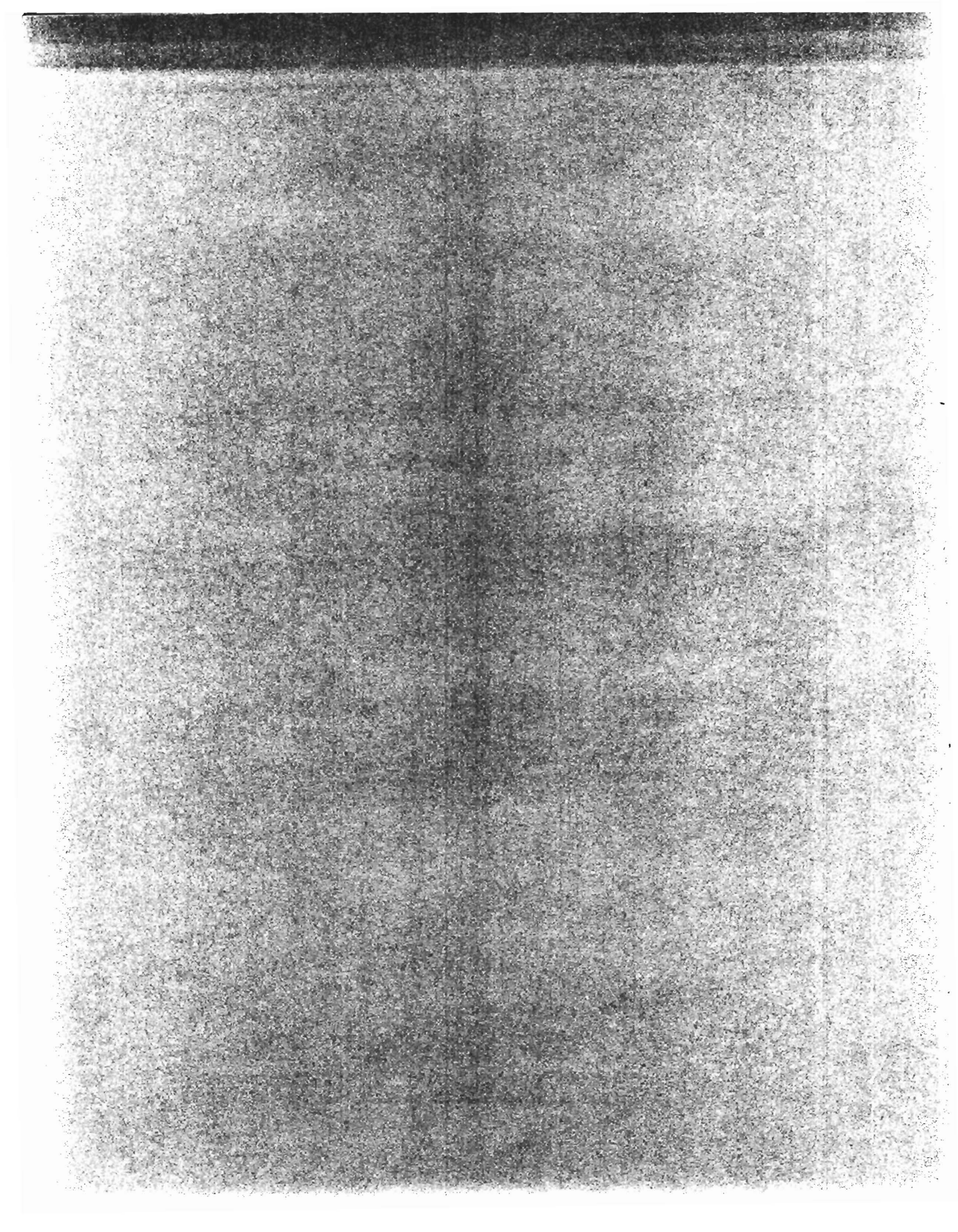
## S5-2

### "High-Performance Switching in the MAN and Public Network"

(Barry Phillips - Adger Smythe Corp.)

With 155-622 Mbps performance today, and up to 10 Gbps in the future, Asynchronous Transfer Mode (ATM) is the connection fabric of choice for multiprocessor architectures that span the metropolitan and wide-area public network. While the Synchronous Optical Network Digital Signal Hierarchy (SONET) physical layer guarantees enormous bandwidth (155.52 Mbps to 2.488 Gbps) and inter-vendor compatibility for broadband MANs and WANs, the performance of the ATM switch (communication latencies) is the limiting factor in the performance of distributed databases, real-time data fusion and supercomputer access applications.

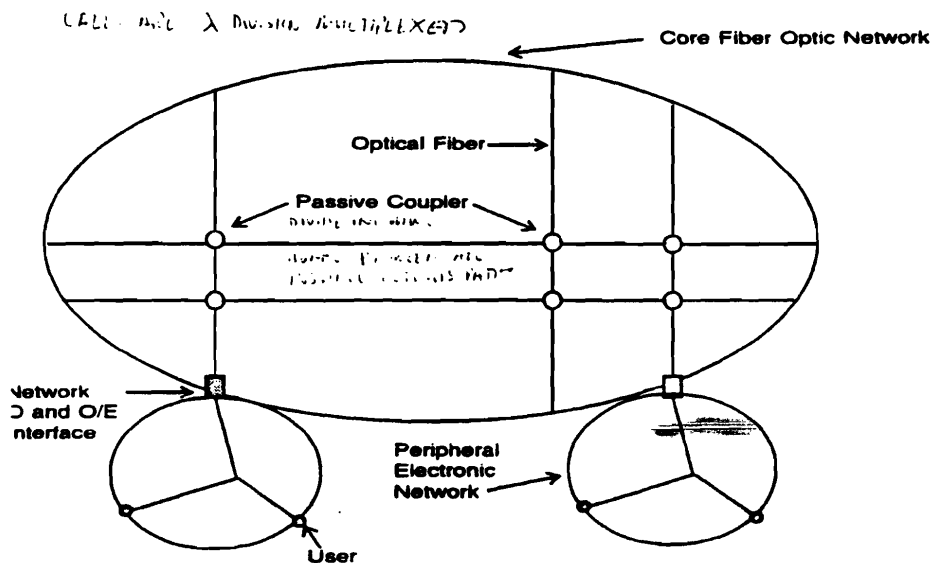
This talk discusses performance bottlenecks in the ATM MAN/WAN switches. A feature comparison table of ATM WAN switches from 16 vendors (e.g., Alcatel, AT&T, Fujitsu, General Datacomm, NEC, NET, Newbridge, Stratacom, TRW) is presented. Issues such as switch architecture and throughput (latency), and the high speed I/O expansion capabilities (e.g., OC-3, future fiber interfaces) are compared. A review of the emerging high-speed public (155 Mbps and above) ATM/SONET services is presented. A bibliography of gigabit networking sources is given.



## Purpose of Most ATM Switches

- Conserve bandwidth
- Allocate core amount of bandwidth within carrier network
- Provision multiple services
- Make profit

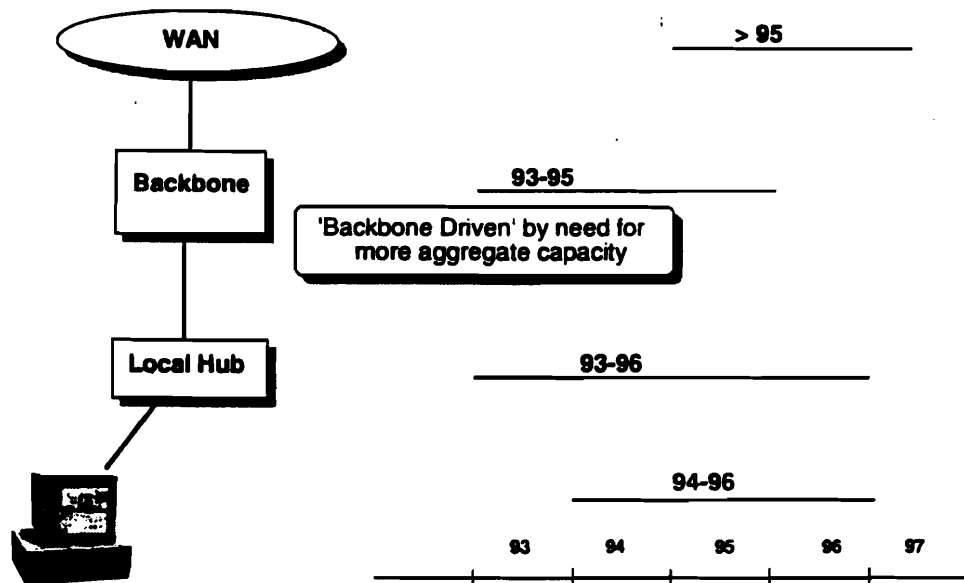
## Optical Network Without Switches



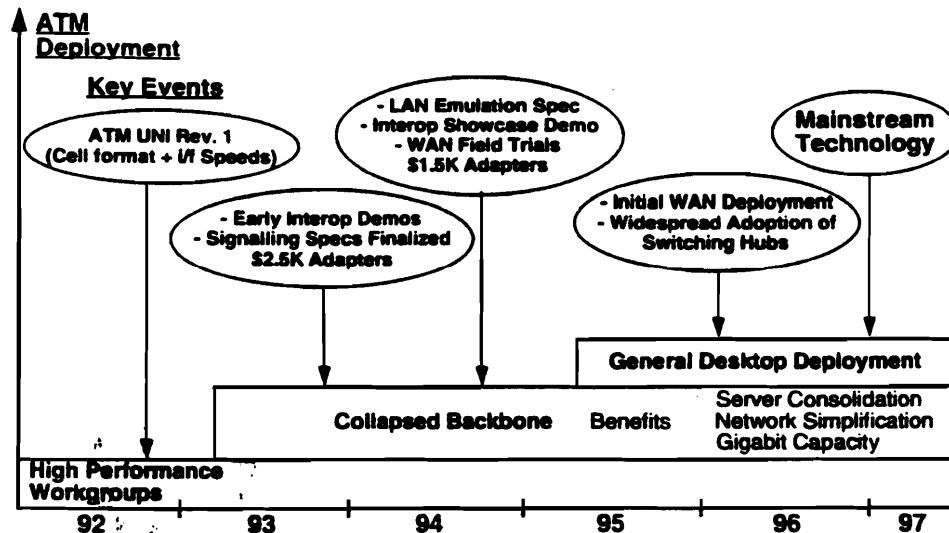
Broadcast at  
1 Tbps = 1000 Gbps

\* ALL CALLS REQUIREMENT OVER ALL POSSIBLE ROUTES BETWEEN SOURCE & DEST.

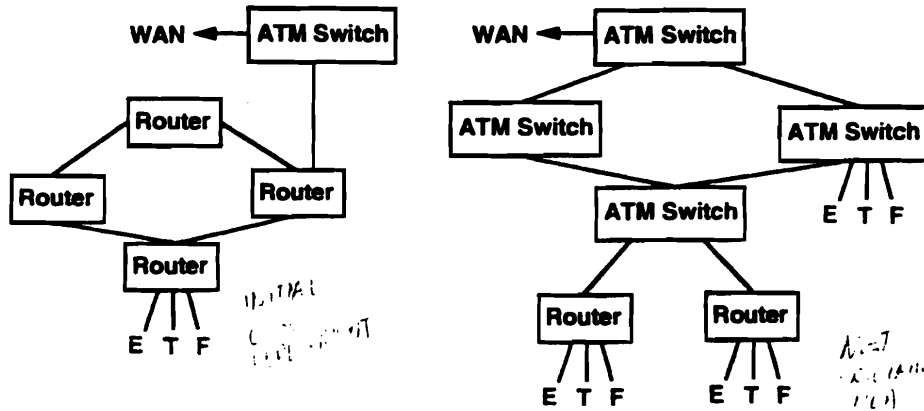
# ATM Evolutionary Scenario



# ATM Evolutionary Path



## Router Bigot vs. Switch Bigot

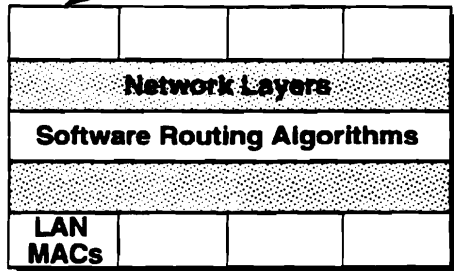


This is common

## ATM Impact on Routers

Today through 1996

Initially Replace LAN MAC with ATM MAC

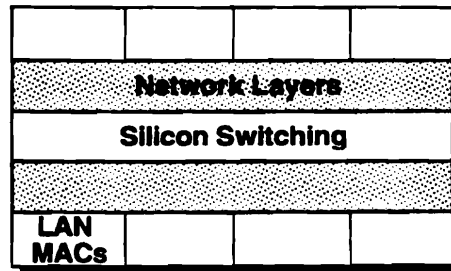


Software Switching

(CISC -> RISC -> Bitslice)

1996 and Beyond

Handwritten note: *Handwritten note: This is the goal*

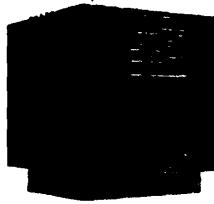


Hardware Switching

(Separate route selection from switching)

## Main WAN Switch/Access Contenders:

- Stratacom/AT&T
  - Newbridge/RBOCs
  - Alcatel/Sprint
  - NET/Cisco
  - NEC/Witel
- General Datacomm
  - Digital Link W/ATM
  - Northern Telecom
  - Cascade Communications
  - Hughes
  - TRW/Sprint
  - Fujitsu



WAN Switches

Vendor	Product	Switch Architecture	Switching Throughput	T1 (DS-1) ATM	T3 (DS-3) ATM
Alcatel Data Networks	1100 HSS	16 x 16 Matrix Nonblocking	1.2 and 10 Gbit/s options	Planned	Yes
Ascom Timeplex	STS-50				
AT&T Network Systems					
Cascade Communications	B-STDx 9000	Bus-based	1.2 Gbit/s		Yes
Digital Link	W/ATM Gateway	Bus-based			
DSC Communications	MegaHub BSS	16 slot switch, 16 buses	6 Gbit/s per module	No	Yes
	MPAX 200	Dual slotted bus structure			
	MPAX 300	Dual Slotted bus structure			
Fujitsu Network Switching					
General DataComm Inc.	APEX	16 x 16 Matrix Nonblocking	3.2 Gbit/s, 6.4 Gbit/s options	Yes	Yes
Hughes Network Systems ATM Enterprise Switch					
			2.5 Gbit/s		Yes
Lightstream	Model 2010	Matrix Nonblocking	2 Gbit/s	No	Yes
Motorola Codex	6950 SoftCell	Bus-Based Modular	1.0 Gbit/s	Yes	No
NEC		8x8 to 16x16 TDM Bus	2.5 - 10 Gbit/s		
NET					
Newbridge Networks					
	36150	4 x 4 broadcast cascading to 16 x 16			
Northern Telecom					
	Passport	FrameCell shared buses	1.6 Gbit/s	Yes Q4 1991	Yes Q4 1991
	Gateway	Matrix Nonblocking	1.2 Gbit/s	Yes	Yes
Stratacom					
	BPX	Matrix cross point switch	9.6 Gbit/s blocking	Yes	Yes
TRW					
		Bus-based	3.2-12 Gbit/s		Yes



WAN Switches

Other ATM Interfaces	AAL Types	Frame Relay Interfaces
100, 155 Mbit/s over fiber	3, 4	56/64 T1
OC-3	1, 3, 4, 5	Sub-T1 to DS-3, Channelized T1 Frame Relay
E1, E2, E3, OC-3	1, 5	T1, T3
100, 155 Mbit/s over fiber, 155 Mbit/s over STP	5	
E3	5	Frame relay, frame relay to X.25, HDLC and SNA/SDLC at DS-0 to E3 56/64 T1 frame relay
100, 140 Mbit/s over fiber	1, 5	
OC3/STM3	1, 5	Yes
100, 155 Mbit/s UNI over fiber	1, 3, 4, 5	
6 Mbit/s Japanese	1, 3, 4, 5	56, 64 T1, T3 frame relay
OC 3/12		

Page 2

WAN Switches

LAN Interfaces	SMDS Support	Other Interfaces	Max Ports	Network Management Architecture
	Yes		29 ATM	
	Yes	PPP	14 ATM	
Ethernet				Uni-View NMS via x.25
Ethernet, HSSI, X.21, RS449		T1, E1 circuit emulation	32 ATM, 64 Ethernet	
		T1/E1, T3/E3, OC-3	16 ATM (DS-3 or OC3) or 128 T1 71 T1/E1 or 18 T3/T3	
Ethernet, FDDI, Token-Ring		T1		
V.35, V.11, PPP, Planned Ethernet, FDDI, Token Ring, Yes		DS-1, E1, DS-3, E3, X.25, DS-3 isochronous T1		End-to-end proactive 8 ATM (OC-3), 24 ATM (DS-3) or 32 ATM (DS-1)
			36 T3/E3	

Page 3

Congestion Management	Buffering	ATM SVC Support	Price & Availability
Prioritization, open loop	n/a	No	\$50,000-100,000/1Q 1994
Prioritization, open loop	52 Kbytes per port	No	\$30,000-180,000/4Q 1993
Usage Parameter Control/Connection Admission Control		No	
	64 Kbytes per port	No	\$32,500-125,000/Now
	Up to 512 Kbytes per port		\$60,000-90,000/3Q 1994
Lightstream Feedback and Rate Control	Up to 2.5 Mbytes per port	No	\$25,000-50,000/1Q 1994
Closed loop			
		External route server option	
	56-Kbyte input 52-Kbyte output		\$175,000-400,000-Now
	1.24 Mbytes/port	Yes proprietary	\$75,000-250,000/4Q 1993

## ATM Switches

## ATM Switch Challenges:

- Coverage must extend across Globe
- Customer needs new equipment to access service.  
Customers will want compatible ATM LANs
- Must compete with dedicated line costs?
- Carrier compatibility issues: ATM systems from different manufacturers don't interoperate, especially SVCs and traffic issues.

## Questions for ATM Switch Vendors

- Does it provide advanced traffic policing and possess the ability to prioritize CBR over regular traffic?
    - Conform with ATM Forum V3.0 Spec
  - Adaptation capability for frame relay, SNA/SDLC, Ethernet and circuit emulation
    - SNMP-based network management: both in-band and out-of-band
  - How many shelves to support 32 OC-3s?
  - Redundancy for power supplies, switching fabric, common logic
- Does it possess switched virtual circuits based on Q39B signalling?

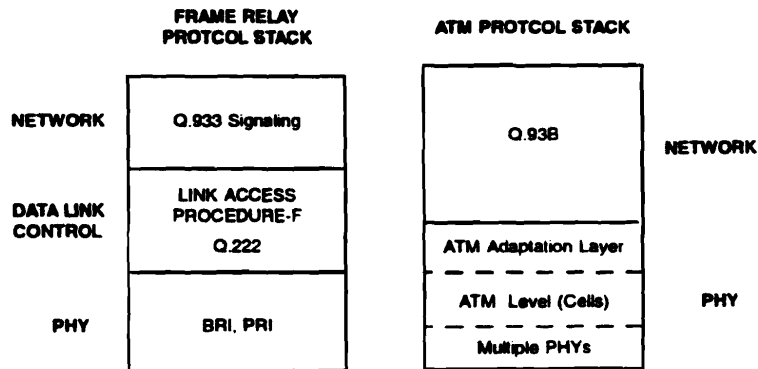
## Routing Today

- PVC
- Vp routing on a call by call basis
- Need to build SVC into switch
  - SVC= Free for all for all

## Waiting for Switched Virtual Circuits

- Any-to-any connection never materialized for frame relay; many LANs over one pre-routed connection possible
- Switch makers dropped development; hopped to ATM product
  - Sprint/Fore have SVCs planned for LAN/WAN nets
- Without connection management the switch is a cell DACS (digital cross connect)
- Manually reconfigure VPI/VCI makes re-routing slow and inflexible
- ATM Forum working standards - needs global implementation

## ATM vs. Frame Relay



Data Link Connection Identifier (DLCI) is local: ability to mux multiple logical apps. onto single link.

Flow Control via Forward Explicit Congestion Notification (FECN) Backwards Explicit Congestion Notification (BECN) and Discard Eligibility (DE) bits

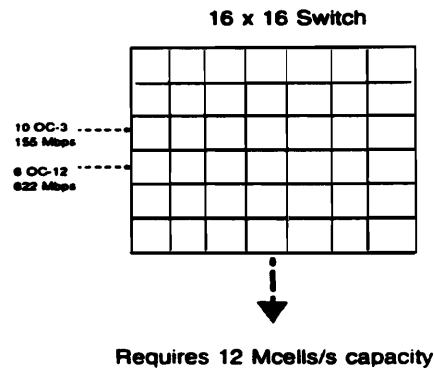
Connection-Oriented: Packet muxing based on identification of individual channels based on Virtual Circuit (VC) number. ATM uses addressing that is of local significance for a particular interface only.

Switch/service provider have burden of traffic shaping and bandwidth management.

IF CARRIES TO THE ATM (WHICH THEY DID) IN FRAME RELAY - CONNECTIONS  
 ARE AVAILABLE IN ALL SIZES

## Future Switch Fabric Requirements

- **Switch design challenge is how to switch hundreds of millions of cells arriving per node**



## ATM Switch I/O Modules

- **Copper Trunks: 56/64K, T1, T3**
- **Optical: OC-3, OC-12 (future)**

**How many shelves of equipment  
dowa it take to support 32 OC-3's?**

## **ATM Switch Profile: Stratacom**

- **9.6 Gbit/s Matrix Crosspoint Switch; Next generation switch fabric scales to x Mbit/s aggregate capacity**
- **Closed Loop Bandwidth Manager assures CIRs**
- **Multiband: 56 Kbit, Nx64, T1, T3, 6 Mbit/s (Japan )for Frame Relay and ATM**
- **Automatically routes PVCs**

## **ATM Switch Profile: Stratacom Admission Control**

- **Feedback Control**
- **Advanced Queueing**
  - per Vc queueing (isolate sources of congestion and direct corrective action to Vc)
- **Opticlass Queue Service Class**
  - Proportional Queueing (32 service subclasses)
  - Selectable queue service algorithms

## **AT&T Engineering + Stratacom Switch = Network:**

- Experiments/users don't want busy signals
- AT&T Credit Manager allocates 'tokens'
- Network engineered to give out credits fast enough to meet Committed Information Rates (CIRs)
- Works with Stratacom Queue Service Parameters (Min/Max bandwidth credit accumulation rates, service algorithm, maximum queue depth, CLP high/low, EFCN threshold, service priority)

### **Stratacom's Service Algorithms:**

- Always serve, always OK to serve, guaranteed, min. transmit rate, delay limited

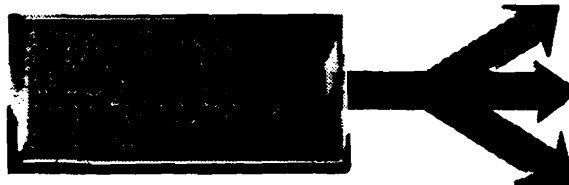
## **ATM Switch Profile: Northern Telecom Passport**

- Preventive and Reactive Controls
  - Network Engineering
- Load Sharing on Multiple Paths
  - Traffic Enforce/Shaping
    - Call Admission
    - Queuing Policy
    - Buffer Allocation
    - Traffic Rerouting
    - Rate Adaption
- Explicit Congestion Notification/Discard (priority-based)
- Expanded frame relay notion of FECN, BECN
- Monitor network, CIRs, buffers, queueing trends

## Impact of Open Loop Congestion Management:

- Trick or Treak!
- If congested, segments of larger frames can be discarded at intermediate nodes
- Other segments may continue to consume bandwidth as frames march onward
- Re-tries necessary to deliver entire message
- Carrier may not be engineered to handle your burst rate - discards.

## ATM Switch Conclusions:



- Users want ATM LAN/WAN link
- SVCs Coming
- Carriers will provide most robust digital services at frame relay type cost



## **ATM Bibliography**

***Broadband: Business Services, Technologies, and Strategic Impact,***  
David Wright, Artech House, 1993, ISBN 0-89006-589-6

***Gigabit Networking,*** Craig Partridge, Addison-Wesley, 1993

***Distributed Multimedia Through Broadband Communications Services,***  
Daniel Minoli and Robert Keinath, Bellcore  
Artech House, 1993, ISBN 0-89006-689-2

***Broadband Networks: ATM & Frame Relay,*** Steven A. Taylor, Consultant,  
Distributed Networking Associates, (910) 292-4444, [taylor@uncecs.edu](mailto:taylor@uncecs.edu)

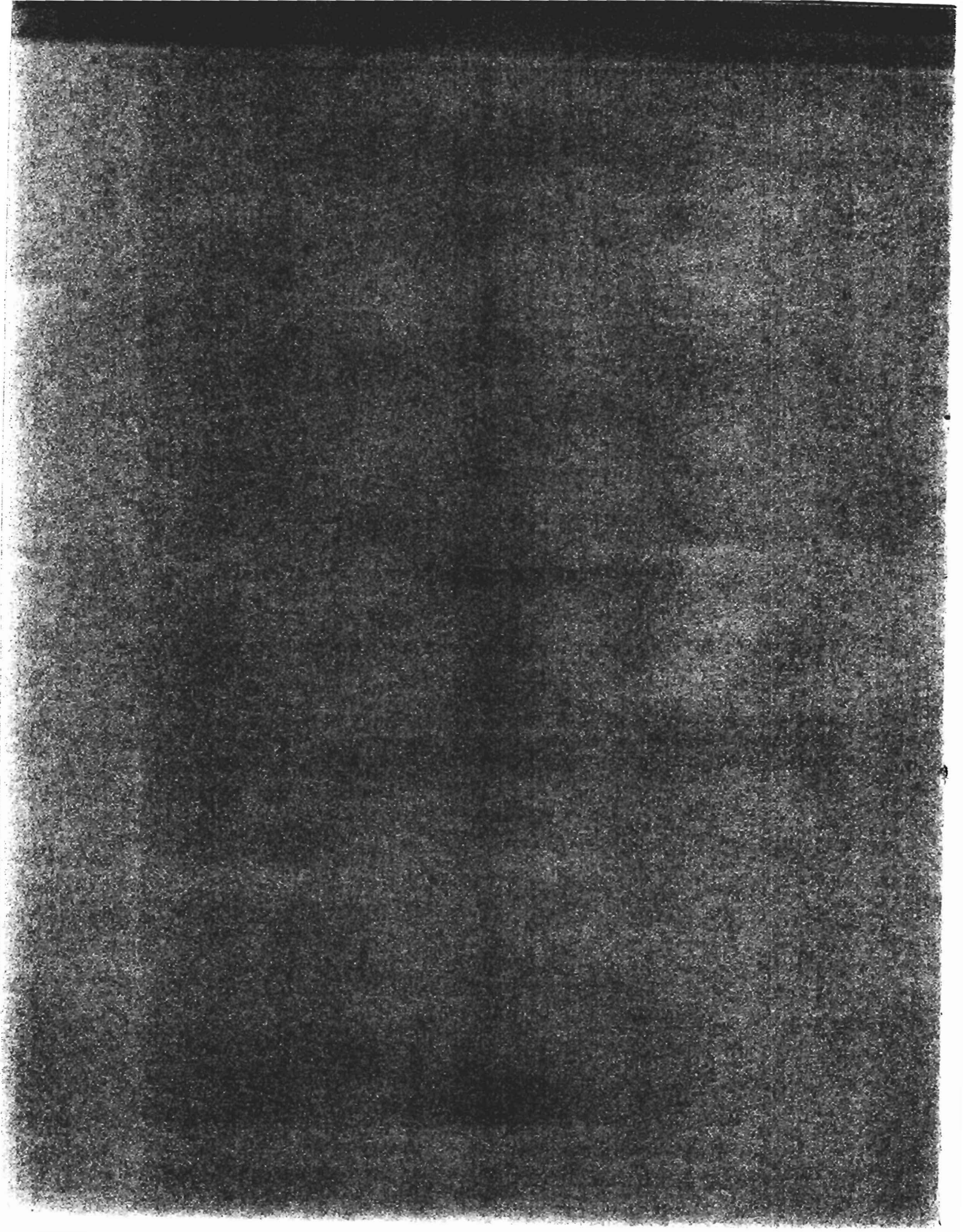


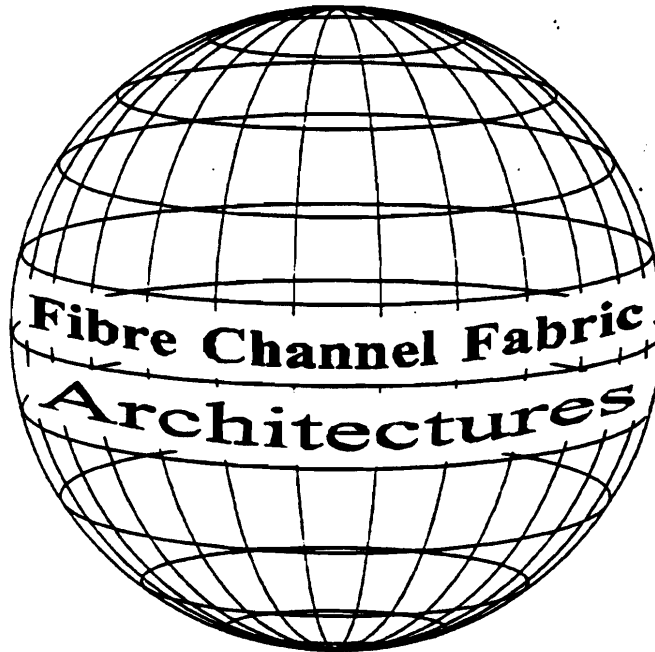
**S5-3**

**"Fibre Channel Switches"**

**(Clint Jurgens - ArCor Communications)**

The world of high-energy physics is demanding very high-bandwidth, low-latency networks for data acquisition, parallel processing, visualization, backbone, and storage networks. Fibre Channel is the leading candidate to meet this need. Fibre Channel is unique in its ability to deliver data that simulation shows will be delivered without an unbuffered send mechanism in Fibre Channel and in the switching architecture to ensure that requests are delivered as addressed. In addition, switching architectures to provide Fibre Channel Connection and Connectionless service independently or concurrently are available.



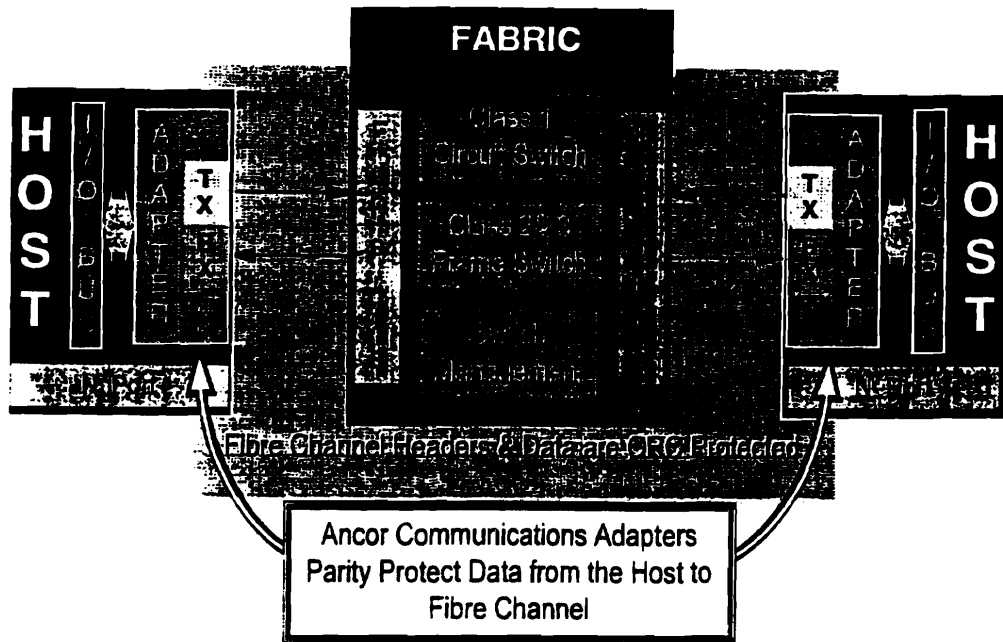


## Requirements

- Optimum Delivery of Short and Long Messages
  - ⊖ Low-latency
  - ⊖ High-bandwidth
- Scalable
- Non-blocking
- Gigabit fiber optic communications
- Support data acquisition environment
- Standard based solution
- Reliable, robust design
- Absolute protection from data corruption

# System Architecture

Ancor Communications



Copyright 1994 Ancor Communications. All Rights Reserved

10/28/94

# Reliable Data Transfer

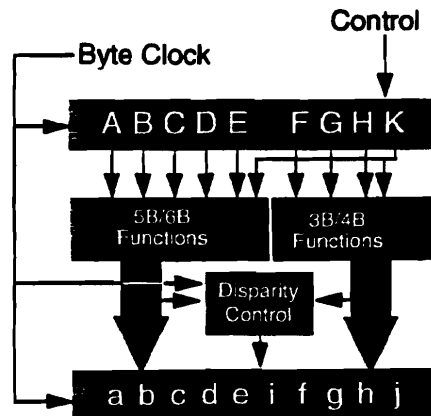
Ancor Communications



- 8B/10B encoding
- Unique 4 byte delimiters and ordered sets
- 32 bit CRC on data and header
- **NO** undetected errors

## FC-1

8B / 10B from IBM improves the Transmission Characteristics of Data



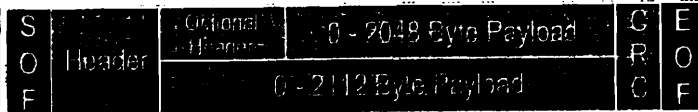
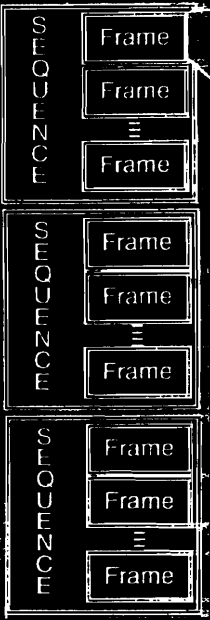
Copyright 1994 Ancor Communications. All Rights Reserved

10/28/94

# Delimiter Control Switching

ANCOR Communications

E  
X  
C  
H  
A  
N  
G  
E



## Frame-delimiter ordered sets

- ⊙ SOF Connect Class 1 (SOFc1) EOF Normal (EOFn)
- ⊙ SOF Initiate Class 1 (SOFi1) EOF Terminate (EOFt)
- ⊙ SOF Normal Class 1 (SOFn1) EOF Disconnect Terminate (EOFdt)
- ⊙ SOF Initiate Class 2 (SOFi2) EOF Abort (EOFa)
- ⊙ SOF Normal Class 2 (SOFn2) EOF Normal Invalid (EOFni)
- ⊙ SOF Initiate Class 3 (SOFi3) EOF Disconnect Terminate Invalid (EOFdti)
- ⊙ SOF Normal Class 3 (SOFn3)
- ⊙ SOF fabric (SOFf)

## Result in negative Running Disparity

## SOF, EOF used to control functions of Fabric

10/26/94

Copyright 1994 © Anzor Communications. All Rights Reserved

# Data Integrity Through the Fabric

ANCOR Communications

## Class 1 Connections

- ⊙ Only valid SOFc1 Frames make connections
- ⊙ Any other problem that prevents a connection results in error message to originator of SOFc1
- ⊙ Sender cannot transmit until it completes connection with destination

## Class 1 Data

- ⊙ Switch passes Class 1 frames unchanged
- ⊙ If switch should modify a frame it will cause a CRC error at the destination port
- ⊙ If a port has an abort condition (example: loss of sync due to cable removal), Fibre Channel primitive signals provide proper take down and re-link

## Class 2 and 3 Frames

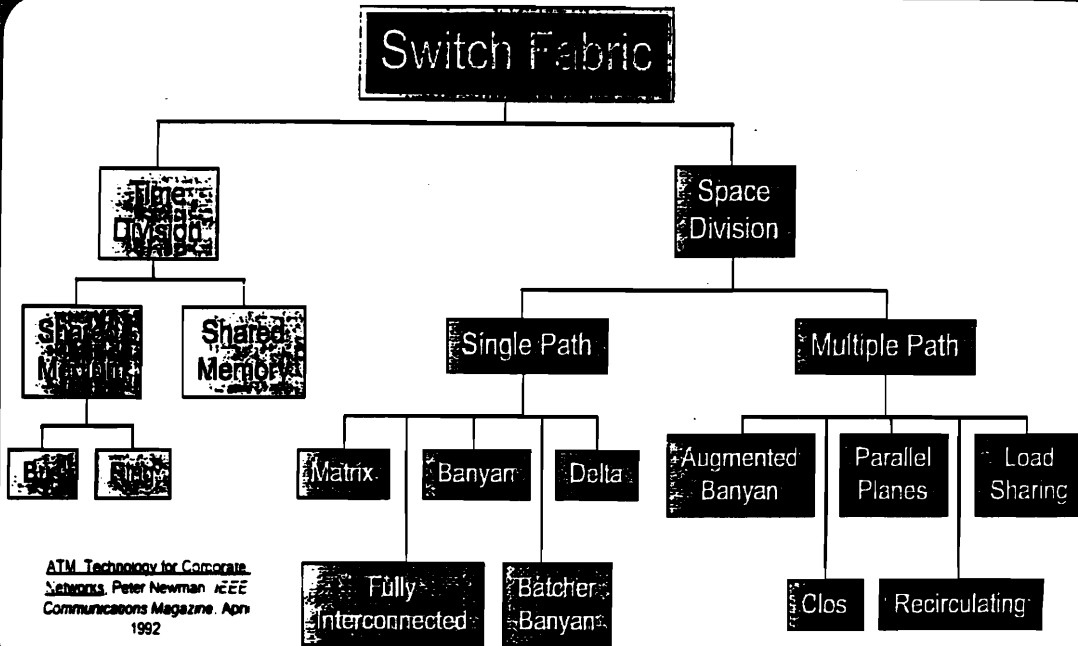
- ⊙ Only valid frames are delivered to destination address
- ⊙ Frames received in error are dropped
  - ✓ 8B/10B error
  - ✓ CRC error
  - ✓ Framing error
- ⊙ Undeliverable Class 2 Frames Result in Error Message to Originator
- ⊙ Undeliverable Class 3 Frames are Dropped

10/26/94

Copyright 1994 © Anzor Communications. All Rights Reserved

# Switch Fabric Classification

ANCOR Communications

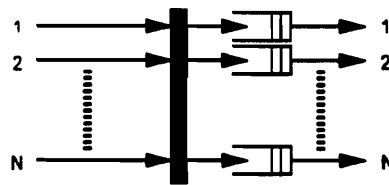


10/28/94

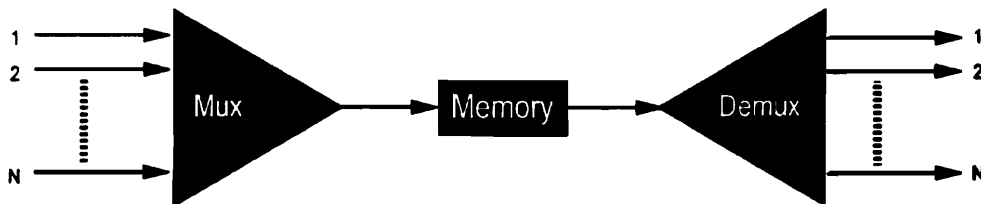
Copyright 1994 © Ancor Communications. All Rights Reserved

# Time Division Fabrics

ANCOR Communications



**Shared Medium**



**Shared Memory**

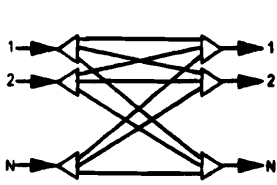
10/28/94

Copyright 1994 Ancor Communications. All Rights Reserved

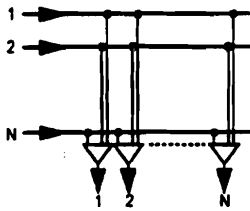


# Single-Path Networks

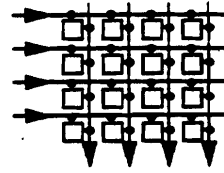
ANGOR Communications



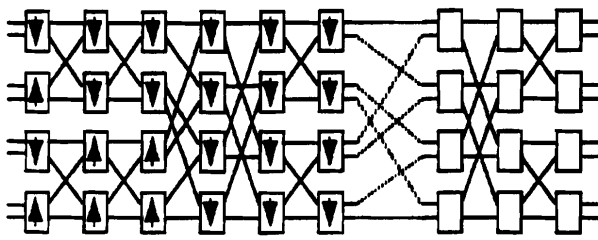
Fully Interconnected



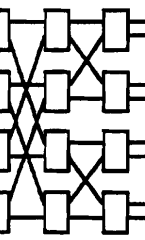
Fully Interconnected



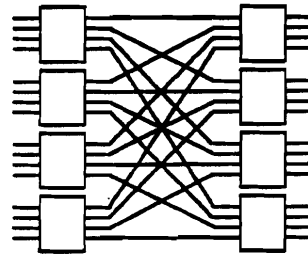
Matrix



Batcher



Banyan



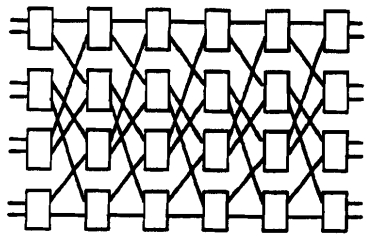
Delta

10/26/84

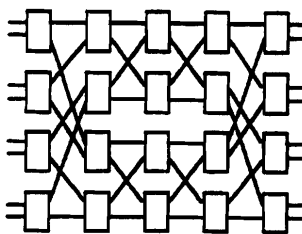
Copyright 1984 © Anacor Communications. All Rights Reserved

# Multiple-Path Networks

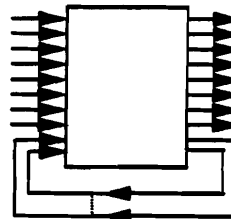
ANGOR Communications



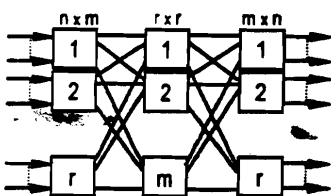
Augmented Banyan



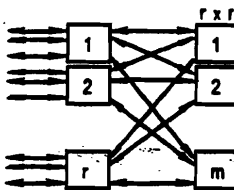
Benes



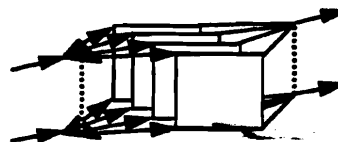
Recirculation



Two-stage Clos



Folded Clos

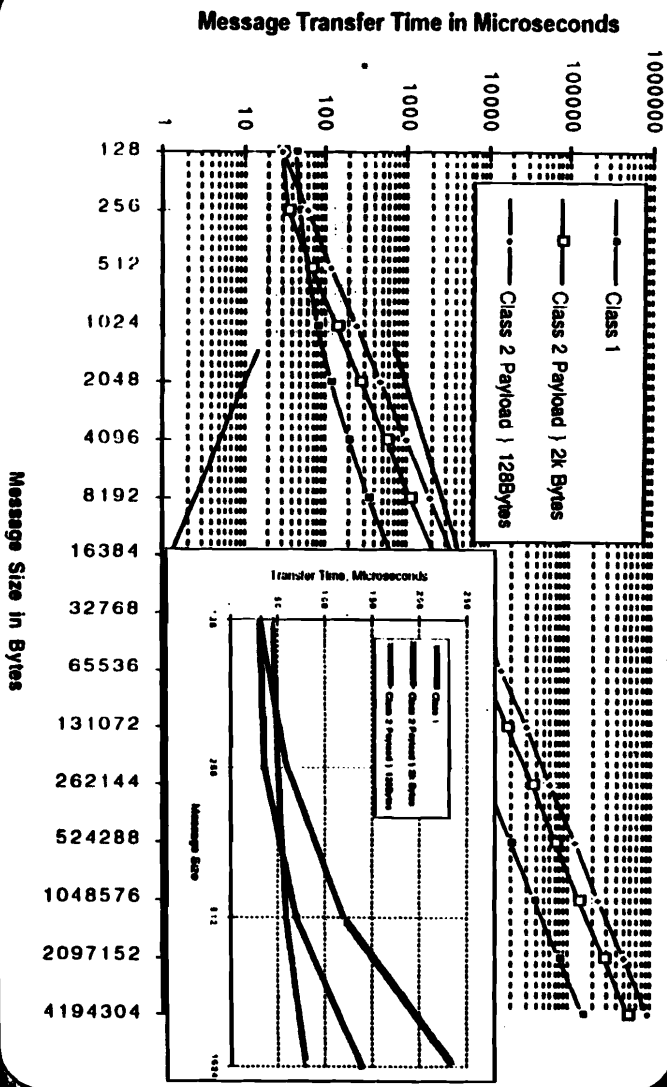


Switch Planes in Parallel

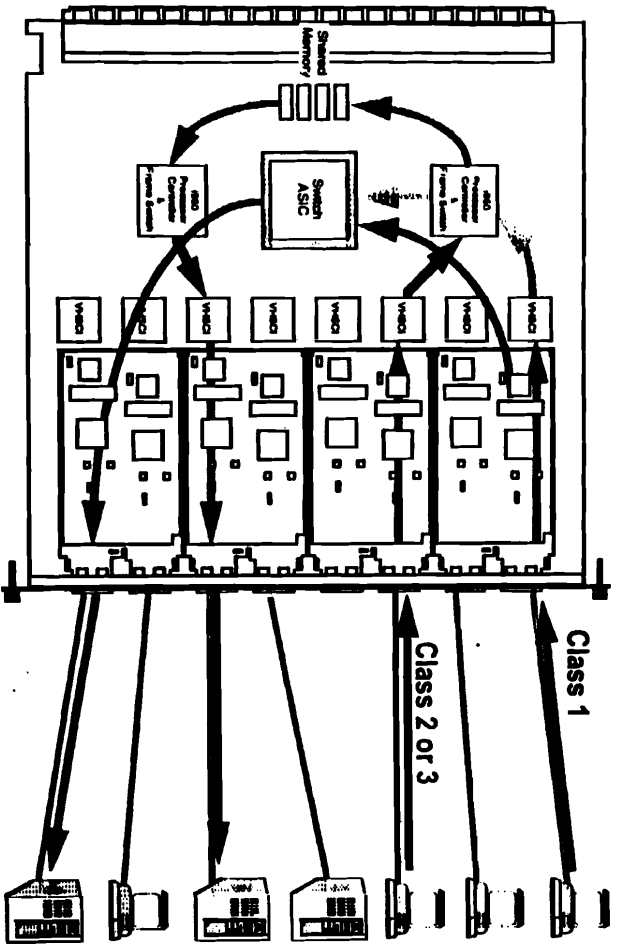
10/26/84

Copyright 1984 © Anacor Communications. All Rights Reserved

# Class 1 for Long Messages & Class 2 for Short Messages



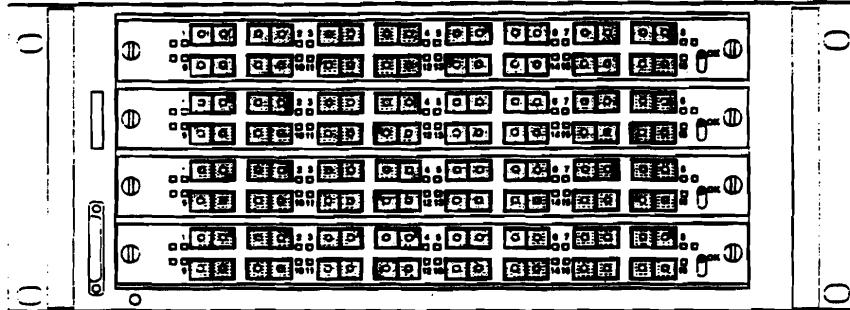
# Two Dimensional Switching



# Modular and Fixed Size Chassis

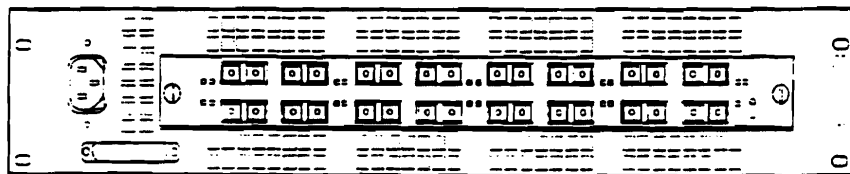
ANCOR Communications

- ✓ External -48Vdc Power
- ✓ Redundant Power Input
- ✓ Side-to-Side Air Flow
- ✓ 700 Watts
- ✓ 45 Pounds
- ✓ Rack Mount
- ✓ Hot Swap Modules



CXT 250 DM

- ✓ Internal Universal Power Supply
- ✓ Back-to-Front Air Flow
- ✓ 170 Watts
- ✓ 28 Pounds
- ✓ Rack mount or table top



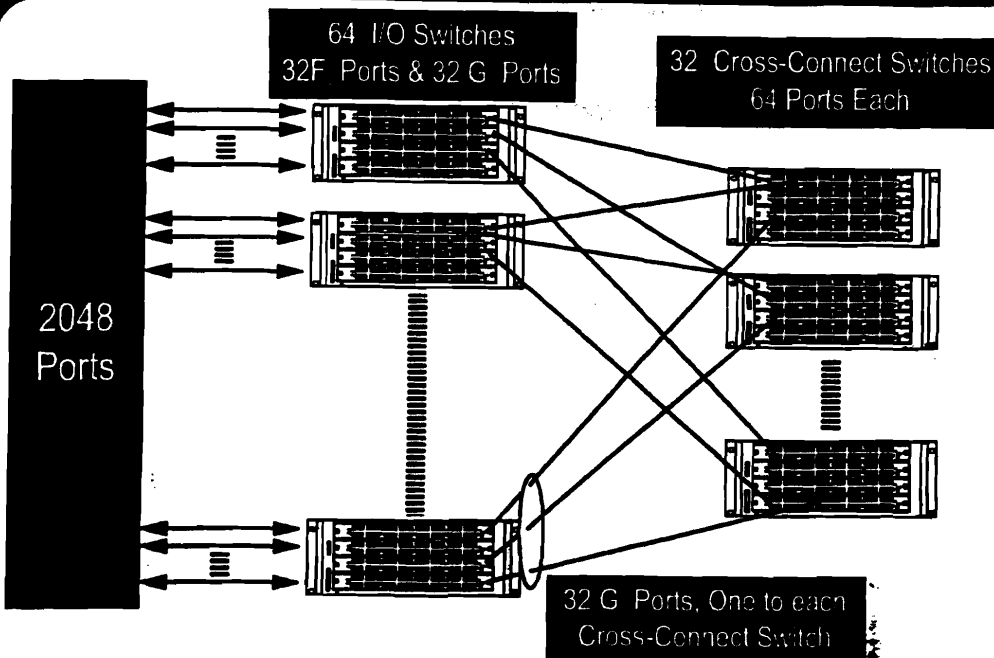
CXT 250 D1

10/28/94

Copyright 1994 © Anacor Communications. All Rights Reserved

# Non-blocking 2048 Port Fabric

ANCOR Communications



10/28/94

Copyright 1994 © Anacor Communications. All Rights Reserved

# Conclusion

ANCOR Communications

- Scalable, non-blocking architecture from Ancor Communications
  - 266 Mbps shipping today
  - 16 Port 1.0625 Gbps ship in 1995
  - 16 & 64 Port 1.0625 Gbps ship in 1996
- Assured Data Integrity
- Optimized for High-bandwidth, Low-latency communications
- Ideal for Data Acquisition

10/20/94

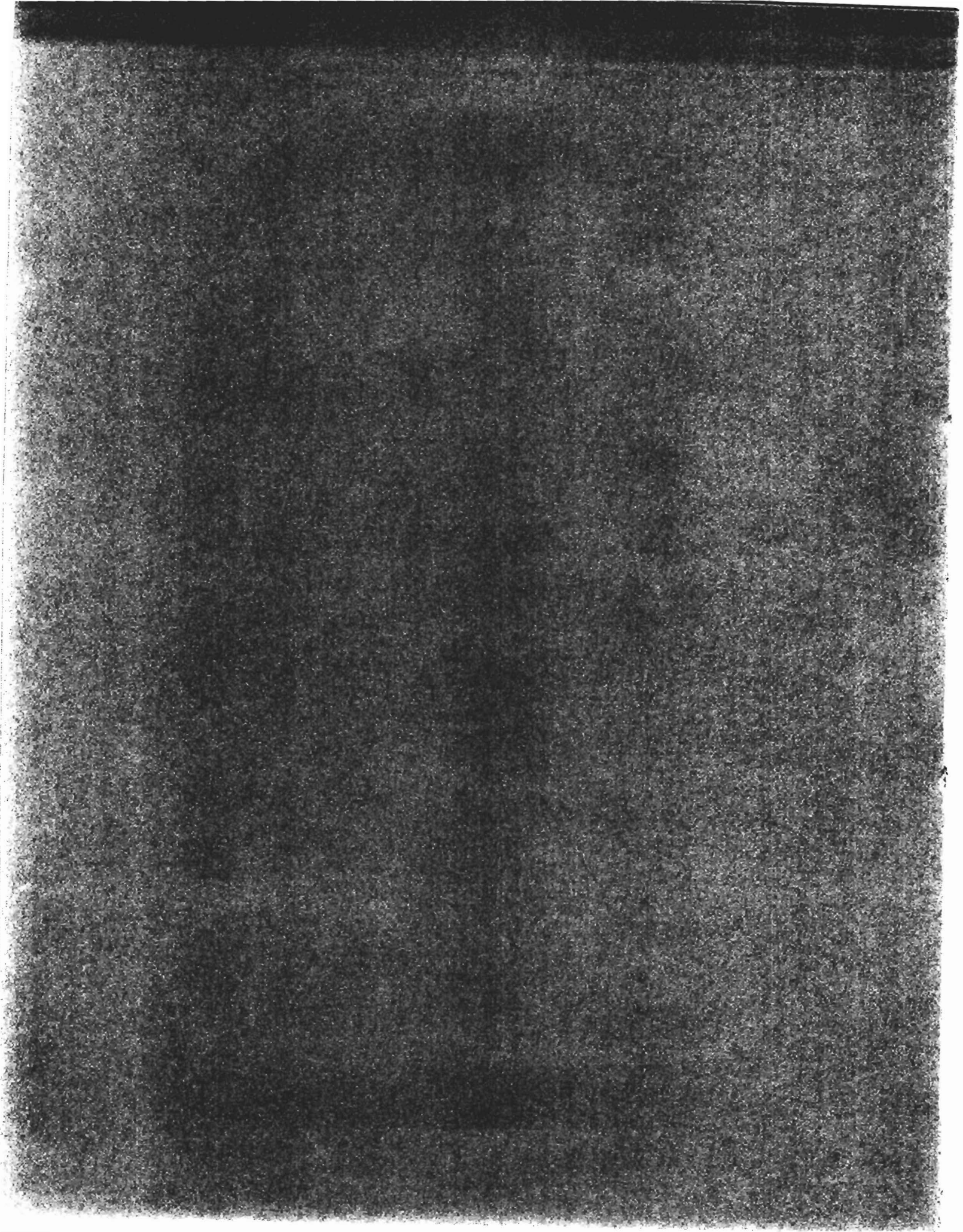
Copyright 1994 © Ancor Communications. All Rights Reserved

**S5-4**

**"SCI Switches"**

**(Bin Wu - University of Oslo)**

A system built from a single SCI ring has limited performance and does not scale. It is therefore necessary to connect many rings through SCI switches to achieve high throughput and low latency. Models of commercial switch designs from several companies will be presented. We will also discuss different system topologies using SCI switches, with special emphasis on applications for HEP Data Acquisition Systems.



## SCI Switches

Bin Wu  
 Department of Physics  
 University of Oslo  
 0316 Oslo, Norway  
 Email: bin.wu@fys.uio.no

### Abstract

A system built from a single SCI ring has limited performance and does not scale. It is therefore necessary to connect many rings through SCI switches to achieve high throughput and low latency. Models of commercial switch designs from several companies will be presented. We will also discuss different system topologies using SCI switches, with special emphasis on applications for HEP Data Acquisition Systems.

### 1 Introduction

The approved IEEE standard 1596-1992 - The Scalable Coherent Interface (SCI) provides computer-bus-like services in a distributed environment. It uses point-to-point unidirectional links to connect up to 64 K nodes [1].

A single SCI ring system is known for its simplicity. However, it has limited performance and does not scale [2][6]. A ring structure is also sensitive to hardware failures. SCI switch is a key component in building up large SCI-based processor architectures. The SCI standard IEEE std. 1596 does not directly specify an SCI switch or bridge. A wide variety of mechanisms are possible [7][8].

In the next section, the general switch models will be introduced. The switch models provide a direct connection between any two rings that are connected to the switch. Information on commercial SCI switch products from several companies will be presented in section 3. We will discuss different system topologies using SCI switches in section 4, with special focus on Data Acquisition systems. Finally, section 5 summarizes the paper.

### 2 General SCI Switch Models

One of the SCI N-switch<sup>1</sup> models is shown in Figure 1.

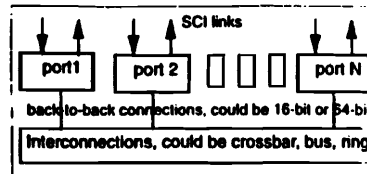


Fig 1. A general block diagram of SCI N-switch

with a number of SCI ports connected. The structure of each port could be implemented as in Figure 2. We expect

<sup>1</sup> N-switch is another terminology in for NxN SCI switch. We will use both in this paper.

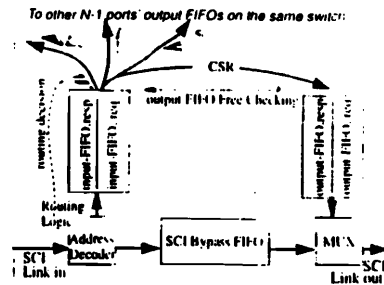


Fig 2. One of the N ports of an SCI N-switch

that an SCI switch port would have minor difference from an SCI node. But no cache coherent logic is needed. Several switch ports are connected back-to-back with possible interconnections as bus, ring, or crossbar. The Address Decoder would be more complicated than the Address Decoder in an SCI node interface since it has to decode a range of address, and probably uses information other than packet's target address, like the transactionid.

The complexity and size of such a port make the implementation of a switch on one single chip difficult when the number of ports becomes large. Another switch model bases on circuit switching concept, aiming at using less queue space, thus reduces hardware on each switch port design (Figure 3). However, this model has not been well studied for SCI, specially on how to solve the congestion problem. None of the current products uses this approach. We will focus on the first model in this paper.

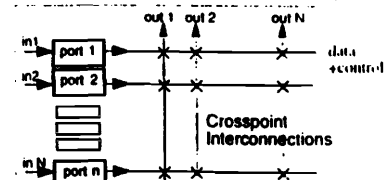


Fig 3. A general block diagram of NxN SCI Switch

Some of the important issues in switch design have been addressed in simulations, and the results are helpful. For example, with four packets deep queues, the high

utilization of SCI and performance can be guaranteed [7]. The speed of routing logic, when under certain threshold, will not influence system throughput, depending on the speed of backside connections [9]. How to choose the interconnects is discussed in [8].

### 2.1 2x2 switch model

To connect two rings, an SCI ring-to-ring switch will be needed. The SCI specification proposes several different topologies that can be built up with simple 2x2 switches. These SCI switches have two inputs and two outputs. Data will go unidirectionally into the inputs and come out from the outputs. The input and output links here are parallel version of SCI, i.e. 16 bits data path, 1 bit clock and 1 bit flag. A detailed block diagram of the 2x2 switch is shown in Figure 4. It is in principle two SCI node chip compatible ports are connected back-to-back.

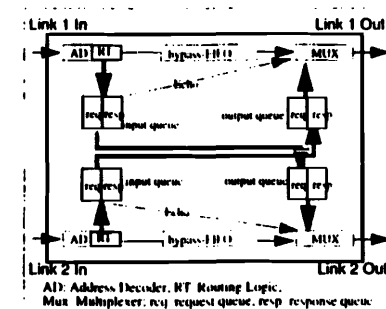


Fig 4. A model of SCI 2-switch

### 2.2 4x4 switch model

A small number of rings connected by SCI 2x2 switches can offer quite high performance, but large SCI-based systems containing thousands of nodes with many 2x2 switches will not be able to use the potential bandwidth [7]. A long path between nodes leading through switches also introduces long latencies. It is also more costly to build a system (for example, 4x4 switch) by using 2x2 switches as in Figure 5. a, b, where eight 2x2 ports are used. The true 4x4 switch model could be derived from the general model in Figure 1. 2.

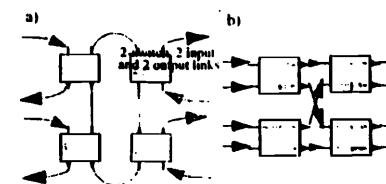


Fig 5. pseudo 4x4 switch built of 2x2 switches

### 2.3 8x8 switch and larger

With the development of new technology, large switches such as 8x8 switch, 16x16 switch or even 32x32 switch

elements will finally become possible to realize. In principle, the more ports one can have on a chip-set, the easier one can form a large system. However, small switches will still be needed to make the best cost/performance trade-offs. Connecting small switches to big switches and then using them in place of the big ones provides flexibility, though may be with some sacrifice of performance and cost.

### 3 SCI Switch Products

Several commercial SCI switch products from companies around the world are either available now or will be available in one year's time.

#### 3.1 Unisys two-by-two switch

Unisys' two-by-two switch uses the same principle as the 2x2 switch model. The input queue of one port is merged with the output queue of the other (Figure 6). In reality, it could be perceived as two back-to-back Datapumps with some additional logic implemented on the same die [3]. So the chip will leverage on the technology, design, and experience achieved from the Datapump effort, thus parallel Low Voltage Differential Signal (LVDS, IEEE P1596.3) at 500 MHz is used and the chip is implemented in GaAs technology. The switch uses a static routing scheme where the routing decision is based on the targetid and the transactionid of the received packet.

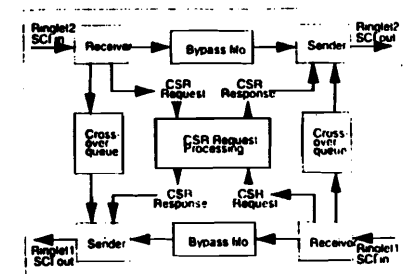


Fig 6. The two-by-two SCI switch block diagram [3]

#### 3.2 Unisys 16x16 switch

The prototype of Unisys Government System Group's SCI switch consists of a switch fabric with 16 ports.

The switch fabric is constructed by replicating the use of a single switch chip in a multistage configuration. Using a banyan-like multistage network will have most of the characteristics of a big crossbar switch, but in a cheap way. Known blocking characteristics of former multistage switch configurations are controlled/minimized by using a patented chip wiring scheme and routing algorithm.

Same as the Unisys two-by-two switch, parallel LVDS is used, and the chips will be implemented in GaAs technology. Bandwidth of the switch fabric is scalable to 16 Gbytes/s. Unisys is currently developing three different, but complementary SCI interconnect topologies, namely single ring, mesh and multistage system.

Detailed descriptions of the switch and fabric are still considered proprietary.

### 3.3 TOPSCI switch

Thomson TCS, France and Dolphin Interconnect Solutions, Norway, are the leading partner of the TOPSCI Eureka project EU R34 to develop and produce SCI technology for interconnecting high performance computer systems, and in particular to develop a high performance SCI switch. CERN and the University of Oslo (RD24 project) are associate partners.

Extensive simulations have been carried out at CERN and the University of Oslo to assess the performance for different architectures. The conclusion points to the convenience of using Dolphin Link Controllers (LC) with a B-Link as the back-to-back internal bus interconnection. The switch will be realized in a Silicon multi-layer Multi-Chip Module (MCM) substrate with four GaAs chips and line termination resistors on a Silicon chip. The routing algorithm will be based on table lookup for flexibility and compatibility (with other SCI switches) reasons.

The target fabrication process of the switch chips is: Vitesse's 0.6 micron II-GaAs III and the design will be carried out using FX-200K Sea of Gates. Alternative interconnection schemes have been evaluated, including micro-bumping, wire-bonding and TAB. Substrate evaluations have been carried out by TCS including 4-layer interconnections with micro-bumping and wire-bonding of the chips.

Based on TCS' preliminary interconnection design rules, Dolphin, SINTEF-Oslo and TCS carried out complete electrical simulations of different interconnection schemes on the silicon substrate. The effect of the substrate resistivity on attenuation, characteristic impedance and crosstalk was evaluated.

A hermetic cooling tower is to put on top of the ceramic package, thus avoiding potential reliability problems coming from liquid-semiconductor interactions. The liquid is inside an exchanger, and the heat transport is assisted by its phase transition.

### 3.4 Dolphin LC switch

The CMOS version of TOPSCI switch is under development by Dolphin Interconnect Solutions. The new switch will use the new Dolphin LC chip with B-Link as the back-end bus on a PCB. The LC switch will be implemented in 2x2 and 4x4 versions, with the possibility of sandwiching two or more PCBs to a larger switch through B-Link.

A mask-based routing algorithm will effectively link many nodes into a hierarchical system. Fault tolerance feature of the switch is emphasized. The switch will be packaged in a special designed box with own power supply. The SCI LinkScope (Tracer, a product from Dolphin) could be easily connected for analysis and debug purpose.

### 3.5 CERN simple 2x2 switch

A simple SCI-SCI bridge (Figure 7) has been built from two nodechips connected back-to-back using a special cable between the two Cbuses (the back-end bus of the Dolphin nodechip). A bridge is possible because the CMOS nodechips from LSI Logic, Ca. can be initialized to recognize a range of 16 SCI destination identifiers whose packets should be passed to the Cbus instead of retransmitting them to the ring. The Cbus maintains the SCI packet structure. If two Cbuses are connected with

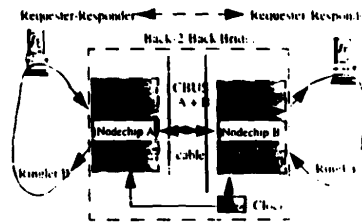


Fig 7 A ringlet bridge prototype[5]

request and response lines crossed, these packets re-appear on the far-side SCI link.

On April 6, 1994, RD24 project at CERN has for the first time successfully transmitted SCI packets without error between two SPARC workstations over such an SCI-SCI bridge[5].

### 3.6 HIC switch for SCI

IEEE P 1355, Heterogeneous Interconnect (HIC) is based on technical developments of highly integrated, low power interconnect technology implemented in high volume commodity VLSI processes. Aspects of the baseline for this standard have their origins in work on parallel transporter-based systems which has taken place in a number of ESPRIT projects. SCI to HIC interface is under development of SINTEF, Oslo, which will use HIC as the transport layer of SCI. Based on the current existing 32x32 switch, C104 from INMOS, the system will be looked like in Figure 8. Each of the HIC here is serial, runs at 100

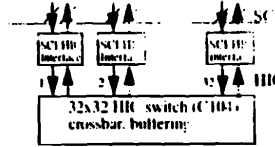


Fig 8 An SCI system based on a single HIC switch

Mbits. For higher link speed, HIC also provides possibility of using 1 Gbit/s link, which is an existing product from BELL Serial Link Technology in France. University of Paris is developing an 8x8 HIC switch which also runs at 1 Gbit/s per link.

### 3.7 Others

Convex switch supports SCI-based supercomputer. The Convex Exemplar which they are shipping today uses a two-level interconnect system consisting of up to 16 "nodes", each of which contains up to 8 PA-RISC processors, memories, and I/O.

The first level interconnect (intra-node) is a coherent crossbar connecting 8 processors, 4 memories, and an intelligent I/O subsystem.

The second level interconnect (inter-node) consists of four SCI rings, each of which interconnects one of the four memory controllers from each of the 16 nodes.

Table 1. SCI Switches

Switch	Vendor	Size	Tech.	Speed	Routing	Link width	Use	Prototype
Urays 2x2 switch	UNISYS	2x2	GaAs, LVDS	2 Gbyte/s	masking	16	UNISYS machine	June 1995
Urays 16x16 switch	UNISYS	16x16	GaAs, LVDS	16 Gbyte/s		16	UNISYS General	Early 1995
TOPSCI switch	Thomson Dolphin	4x4	GaAs, LVDS	4 Gbyte/s	table lookup	16	General DAQ	End 1995
Dolphin LC switch	Dolphin	4x4	CMOS	800 Mbyte/s	masking	16	General	End 1994
Simple 2x2 switch	CERN/Dolphin	2x2	CMOS	200Mbyte/s	Simple masking	16	DAQ	May 1994
HIC switch (C104)	INMOS	32x32	CMOS	32x100 Mbit/s	interval	1	Transporter SCI ATM FC	end 1993
HIC switch	Univ. Paris	8x8	CMOS	8 x 1 Gbit/s	interval	1	SCI ATM FC	end 1995

The current system does not include an "SCI switch" that routes data between SCI ringlets. The routing is performed by the crossbar prior to entering the "SCI domain".

### 3.8 Summary

Table 1 provides a comparison between different SCI switches.

There might be some more secret SCI switch projects going on somewhere around the world.

## 4 SCI-based Networks

SCI switches could be used to connect many topologies, depending on the routing algorithms they support, the size of the switch and the back-end link variation, etc. The typical ones are mesh, cube and multistage networks. SCI based multistage network is seen at CERN as a candidate to cope with the requirements of future very high rate and large scale Data Acquisition Systems (Figure 9), because

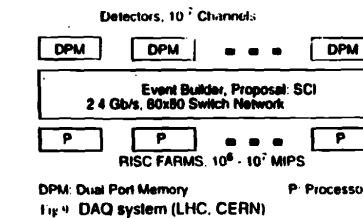


Fig 9 DAQ system (LHC, CERN)

multistage networks are specially suited for topologies with memories on one side of the network and processors on the other[10]. The scalability of multistage network has already been shown in [4], so larger systems can be expected to have scalable performance.

### 4.1 4-switch based 8x8x8 system

Figure 10 is an example of multistage systems that adapts with SCI. We found we must develop a new

terminology to describe such systems. Instead of calling it 8x8 or 16x16, which neither fits, we call it a 8x8x8 multistage system. This system is suitable for data communication between the two sides, which is the case when processors sit on one side and memories on the other. This data flow uniquely from one side to the other. The reverse path in a ring structure can be used for response packets. It is also true that in some cases data flow in both directions, which makes good use of all links. Data flow between the nodes on the same side is also possible, though it makes routing more difficult and the deadlock issue must be taken care of[9].

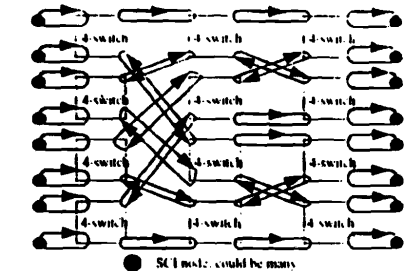


Fig 10 8x8x8 system built of 4-switches

### 4.2 2-switch based 8x8 unidirectional system

Using SCI in conventional multistage network is possible, see Figure 11. As indicated in the figure, a number of active nodes (processors and memories) can be distributed along the path. However, for the same reason we have mentioned before, a ring with many nodes will not scale and has limited performance. When the traffic is uniformly moving packets, the benefit of such a topology is obvious, all the links will be fully used. Nevertheless, it requires deep queues and several outstanding requests to saturate the system due to the echo-issue, i.e. echoes will have to pass a



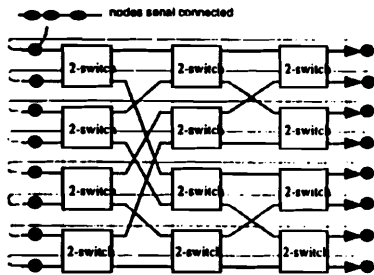


Fig 11. 6x8 unidirectional system built of 2-switches

long way before being routed back to recognize the successes of the send packets.

## 5 Summary

This paper gives a short overview over general SCI switch models, and summarizes several commercial switch designs. Some of the critical features of the commercial SCI switches are not revealed since they are still considered proprietary. Two topologies of using SCI switches are presented. More could be found from the literature list in this paper. As it is shown in table 1, several high speed SCI switches will be available in 1995, which will vastly overcome the current SCI single ring systems in performance and make SCI-based systems more robust.

## Acknowledgements

I am grateful to Andre Bogaerts, Hans Muller, Bernhard Skaali, Ernst. H. Kristiansen, Stein Ojessing, Ernesto Perea, Kaare Lochsen, Inge Birkeli, Ralph Lachenmaier, Mike Chastain, Larry Andersen and Khan Kibria for the help that assisted this work.

The author is supported by the Norwegian Research Council.

## References

- [1] IEEE Std. 1596-1992, "The Scalable Coherent Interface"
- [2] J. Bothner, T. Hulaas, "Topologies for SCI-based systems with up to a few hundred nodes", Thesis for the degree Candidatus Scientiarum, University of Oslo, Norway, 1993
- [3] J. Buggert, V. Desai, L. Herzberg and Khan Kibria, "The Unisys Datapump and Switch", Proceedings the 1st International Workshop on SCI-based High Performance Low-Cost Computing, Santa Clara, Ca. Aug. 1994
- [4] "RD24 Status Report, Application of the Scalable Coherent Interface to Data Acquisition at LHC", CERN/DRDC 93-20, May 1993
- [5] RD24 Status Report, CERN/DRDC 94, May 1994
- [6] S.Scott, J.Goodman, M.Vernon, "Performance of the SCI Ring", Proceedings IEEE ISCA 92, Queensland, May 1992
- [7] B. Wu, A. Bogaerts, R. Divia, E. Kristiansen, H. Muller, E. Perea, B. Skaali, "Distributed SCI-based Data Acquisition Systems constructed from SCI bridges and SCI switches", 10th Int'l Symp. of Problems of Modular Information Systems and Networks, St.Petersburg, 13-18 Sept. 1993
- [8] B. Wu and A. Bogaerts, "Several Details of SCI Switch Models", University of Oslo/CERN Internal Report, version 0.5, 15 Nov. 1993.
- [9] B. Wu, A. Bogaerts, E. Kristiansen, B. Skaali, "Several Approaches of Routing Algorithm for SCI-based Multistage Networks", Proceedings the 1st International Workshop on SCI-based High Performance Low-Cost Computing, Santa Clara, Ca. Aug. 1994
- [10] B. Wu, A. Bogaerts, E. Kristiansen, H. Muller, E. Perea, B. Skaali, "Applications of the Scalable Coherent Interface in Multistage Networks", Proceedings the IEEE TENCON, Singapore, Aug. 1994

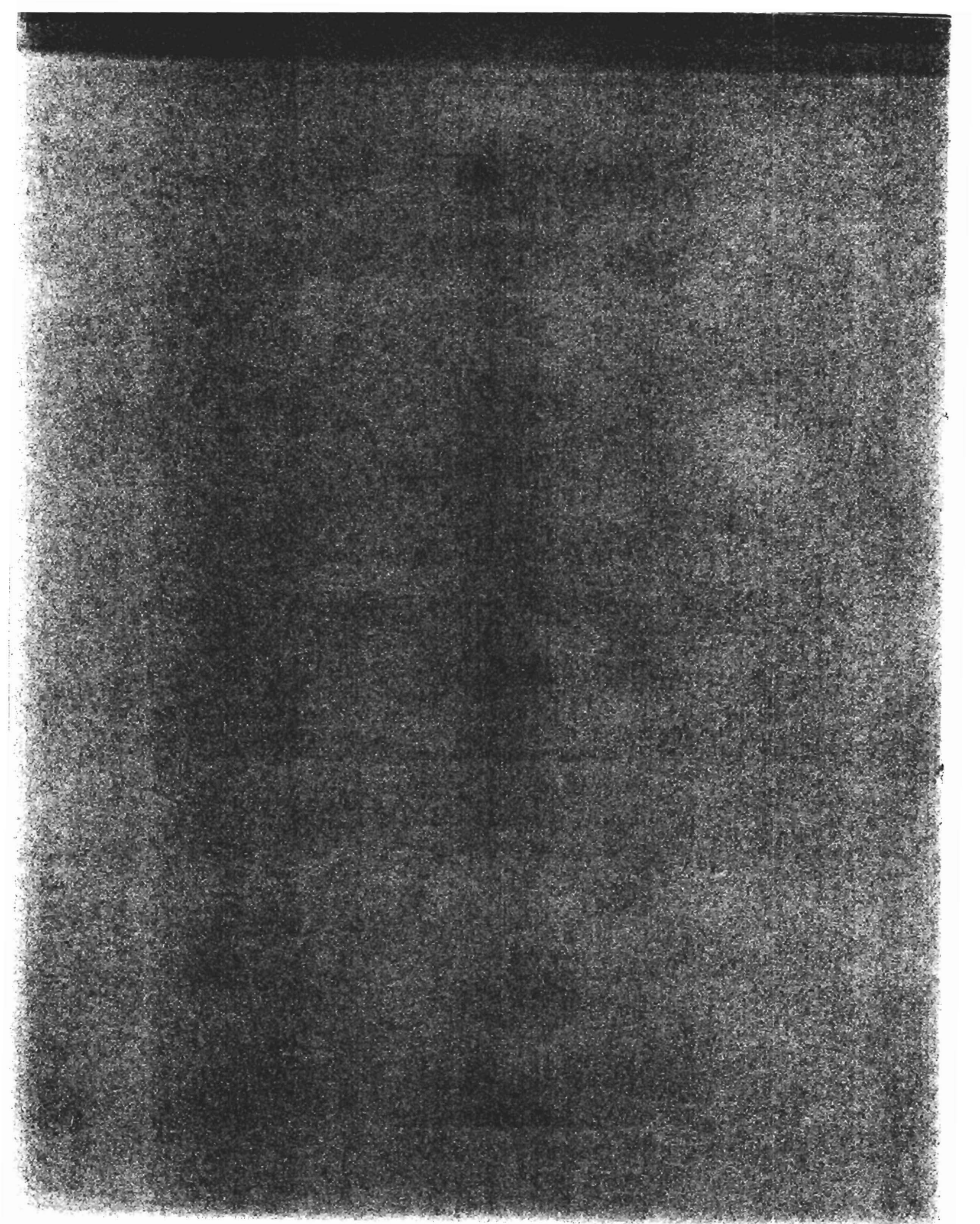


**S5-5**

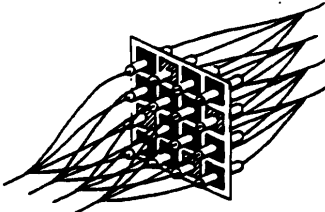
**"Overview of Optical Switches"**

**(Larry McAdams - Optivision)**

An overview of optical switching technologies will be presented. The application of optical switches to "event building" will be discussed. Two examples of optical switches will be described: an acousto-optic barrel-shift switch and a crossbar switch based on semiconductor optical amplifiers. Performance, functionality, and scalability of each of these designs will be discussed. Data collection applications and switch control methods will be covered.



# Overview of Optical Switches



Larry R. McAdams  
Charles H. Chalfant

Optivision, Inc  
4009 Miranda Ave  
Palo Alto, CA 94304

Data Acquisition Conference  
October 27, 1994 Fermilab



## Outline of Talk

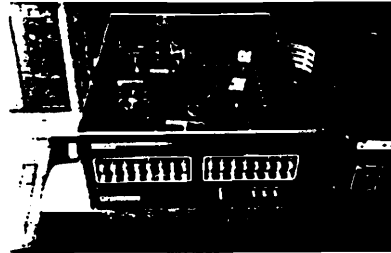
- Survey of Optical Switching Elements
- Acousto-optic Barrel Shifter
- SOA-based Optical Crossbar
- Optical Switching for Data Collection Applications

Data Acquisition Conference  
October 27, 1994 Fermilab



## Performance

- Size: 16 x 16
- Fiber: SM In, MM out
- Ave. loss: -26.0 dB
- Loss Variation:  $\pm 2.5$  dB
- Polarization dependence:  $< 1.0$  dB
- Operating Wavelength: 1285-1320 nm
- Switching time: 1  $\mu$ sec

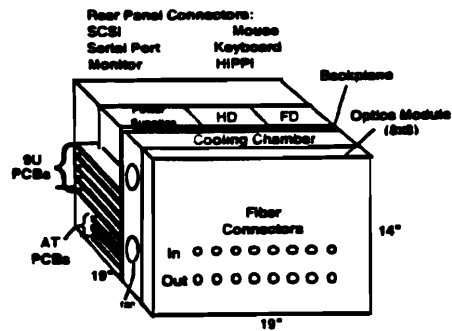
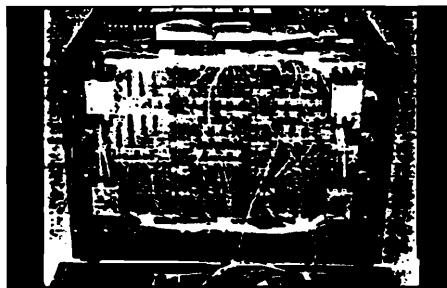


Data Acquisition Conference  
October 27, 1994 Fermilab

OPTIVISION  
INCORPORATED

10

## SOA-based Optical Crossbar Switch

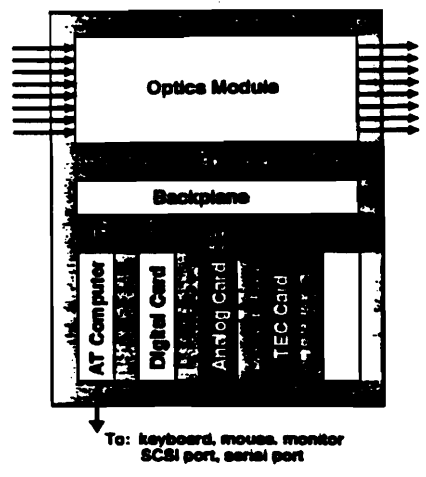


- Modular design
  - Supports both in-band and out-of-band control options
  - Supports connectivity up to 8 x 8

Data Acquisition Conference  
October 27, 1994 Fermilab

OPTIVISION  
INCORPORATED

## Out-of-band Control of Switch

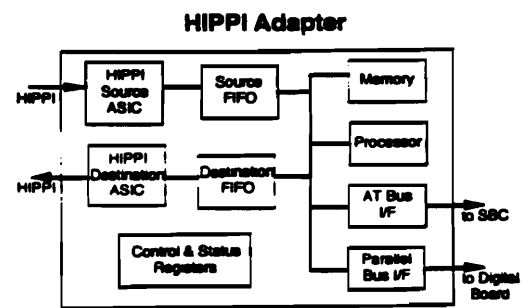
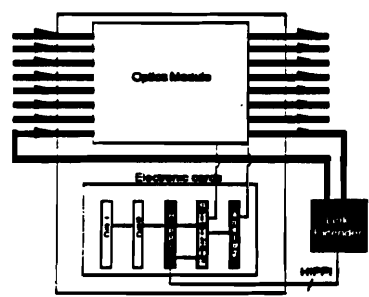


- Optics module contains optics, thermoelectric coolers, and front end electronics
- AT computer supports a graphical I/F, OA&M, and remote comm.
- Digital card demultiplexes AT commands into ECL control signals for individual SOAs
- Analog card generates precision CW drive currents for individual SOAs
- Thermoelectric controller card regulates temperature of optics module

Data Acquisition Conference  
October 27, 1994 Fermilab



## In-band Control Configuration with Polling Receiver

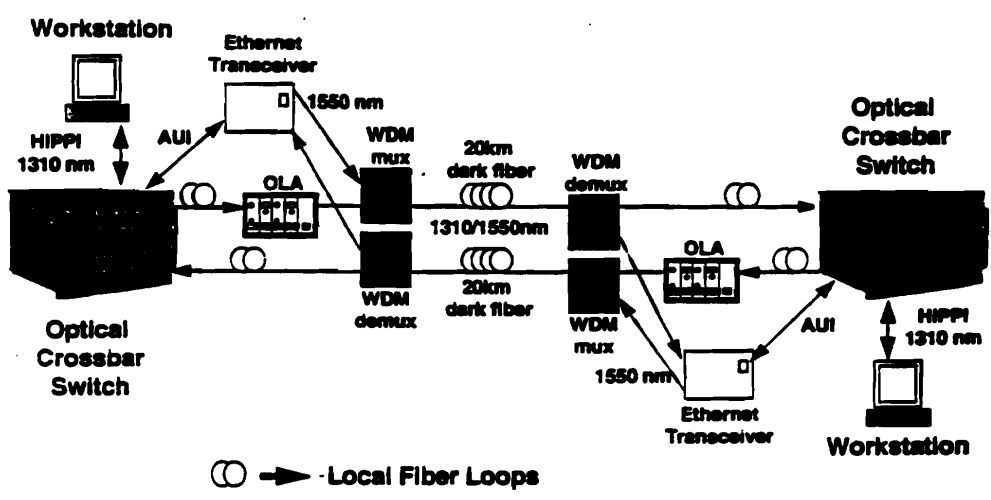


- Initial demonstration based on HPPPI protocol
- Round robin polling using multicast function of MVM architecture

Data Acquisition Conference  
October 27, 1994 Fermilab



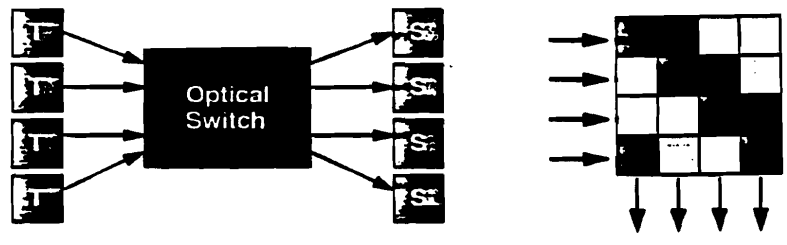
# TBONE Testbed Environment



Data Acquisition Conference  
October 27, 1994 Fermilab



# Data Collection with Optical Switches



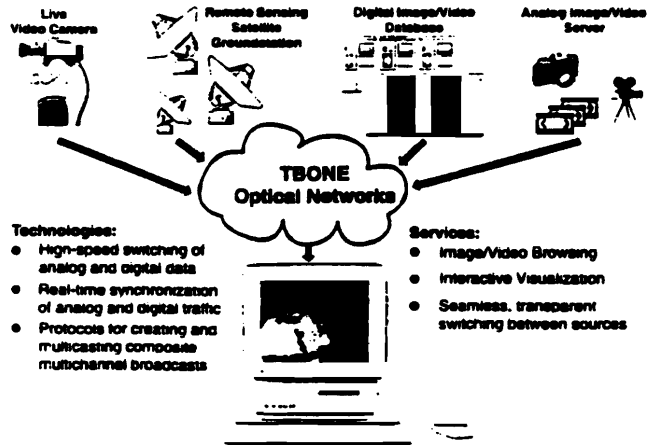
- Optical Switching provides high bandwidth, low error rate, circuit switched connections
- In-band or out-of-band switch control possible
- Buffering for flow control and contention resolution must be done at end points
- The barrel shifter supports the minimum required connectivity for "event building" (N states)
- The crossbar switch supports arbitrary connectivity (NI states)

Data Acquisition Conference  
October 27, 1994 Fermilab





## INTERACTIVE ELECTRONIC GLOBE APPLICATION





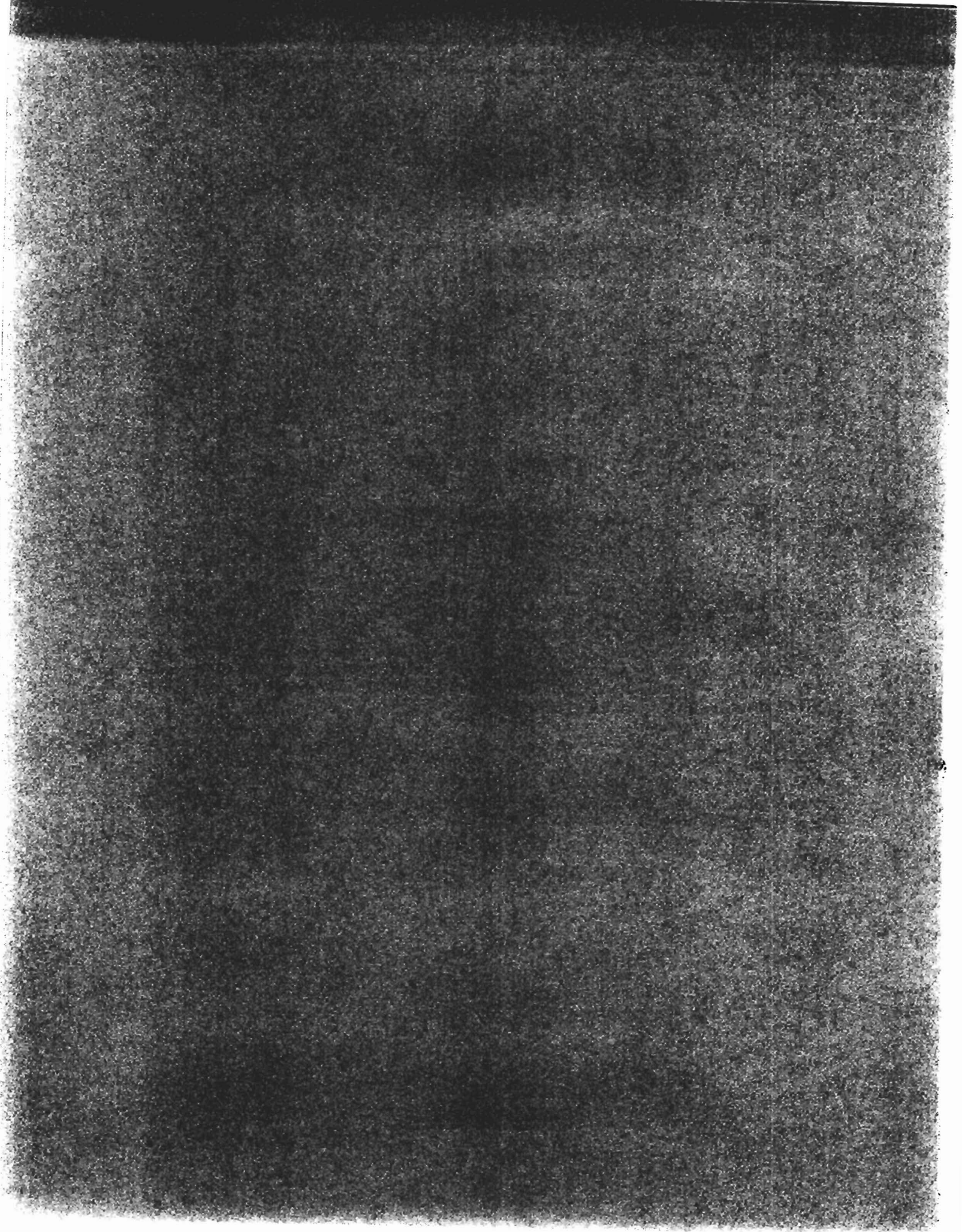
**S5-6**

**"Prizma Switch"**

**(Ton Engbersen - IBM Zurich)**

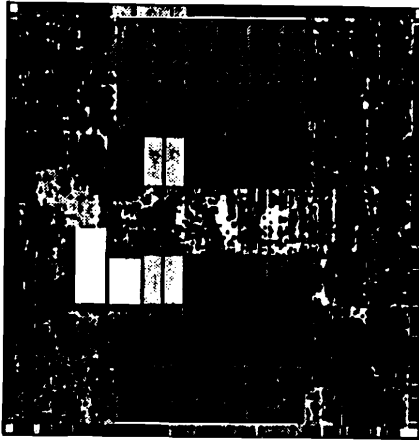
A wide range of evolving multimedia applications integrating voice, video, and data have introduced the demand for new high speed, high bandwidth networks. These networks must also support the different and dynamic bandwidth needs of these applications in a unified manner. ATM (Asynchronous Transfer Mode) has been conceived as an appropriate network technology to address the variety of needs of these applications. ATM's streams supporting fast switching on demand are key elements in providing necessary dynamic bandwidth allocation capabilities within ATM networks.

PRIZMA is an integrated switch chip developed at the IBM Research Laboratory in Zurich, Switzerland. The Laboratory's extensive research and advanced technology expertise in switching systems has led to the development of a state-of-the-art, innovative, high performance, modular, extendible switch chip, particular suited to ATM, but also other packet formats can be supported.



# "Switch-on-a-chip"

## IBM's ATM Switching Technology



**IBM**

### Switch-on-a-chip - IBM's ATM Switching Technology

#### The Road to ATM

A wide range of evolving multimedia applications integrating voice, video, and data have introduced the demand for new high speed, high bandwidth networks. These networks must also support the different and dynamic bandwidth needs of these applications in a unified manner. ATM (Asynchronous Transfer Mode) has been conceived as an appropriate network technology to address the variety of needs of these applications. ATM's structure supporting fast switching on demand are key elements in providing necessary dynamic bandwidth allocation capabilities within ATM networks.

#### Switch-on-a-chip

Switch-on-a-chip is an integrated switch chip developed at the IBM Research Laboratory in Zurich, Switzerland. The Laboratory's extensive research and advanced technology activities in switching systems has led to the development of a state-of-the-art, innovative, high performance, modular, extendable ATM switch chip. The Switch-on-a-chip is unparalleled in today's market place.

#### Characteristics of Switch-on-a-chip include:

- 16 input ports
- 16 output ports
- 300-400 Mbit/s per port
- Built-in support for modular growth in number of ports
- Built-in support for modular growth in port speed
- Built-in support for modular growth in aggregate throughput
- Built-in support for automatic load-sharing
- Self-routing switch element
- Dynamically Shared-output Buffered element
- Built-in multicast and broadcast
- Aggregate data rate 6.4 Gbit/s per module
- 2.4 Million transistors on 15mm chip
- 472 I/O pins

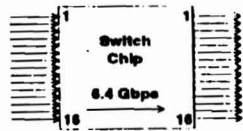


Figure 1. Switch-on-a-chip.

### Description

A maximum internal cell length of 64 bytes has been chosen. Each of the 16 input and output ports has a maximum speed of 400 Mb/s which results in a total aggregate data rate of 6.4 Gb/s per single chip. A unique feature of the switching element is its scalability.

- Switch systems with a larger number of ports can be built from the basic module as single or multi stage self routing network
- The system port speed can be increased by connecting several modules in parallel.
- Better throughput for bursty traffic environments (LAN's) can be achieved by increasing the size of the internal packet buffer also using paralleled modules

Because of its modular architecture, Switch-on-a-chip is the ideal basis for a wide range of products with different price- and performance demands.

### Basic Switching System

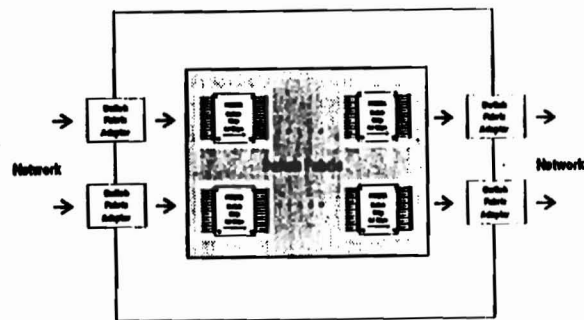


Figure 2. Basic Switching System

A typical switching system is shown in Figure 2 and consists of two elements, the Switch Fabric Adapter (SFA) and the switch fabric formed by the switch chips. In the SFA's, typically the network-dependent processing is done. For example, in an ATM-network, the SFA's provide the conversion of the ATM label to the switch fabric-internal routing mechanism, and provide the new label for the outgoing ATM link. We will call the ATM-cell with its prefixed internal-switch-routing header a packet in the sequel. Note that it is not necessary that Switch-on-a-chip works with ATM cells: any data-frame segmented such that it fits the internal 64-byte maximum size (including the header) will be handled.

### Basic Switch Fabric Element: Switch-on-a-chip Routing

An ideal switch element will route packets without loss and with minimum transit delay, while preserving the order of packet arrival. Most switch architectures proposed for ATM employ the concept of self-routing: logic in the switch inspects the internal-switch-routing header, and routes the packet to the appropriate output. Internally provisions are made such that the complete input traffic can be stored in the shared packet buffer.

### Output Buffering

Due to the statistical nature of the incoming traffic it is possible that several packets contend for the same output port at the same time. Therefore some packets may be temporarily queued to resolve this output port contention. Because of performance it is optimal to buffer packets at the switch module outputs in FIFO order. The ideal packet switch has unlimited buffering capacity and therefore never has to reject a packet because of buffer limitations. In real life the size of the internal packet buffer is limited and means are required to use the available buffer space efficiently. This is achieved by dynamically sharing the limited buffer space among all outputs while maintaining logically separate output queues. The output buffering and routing concepts employed result in the logical switch architecture shown in Figure 3.

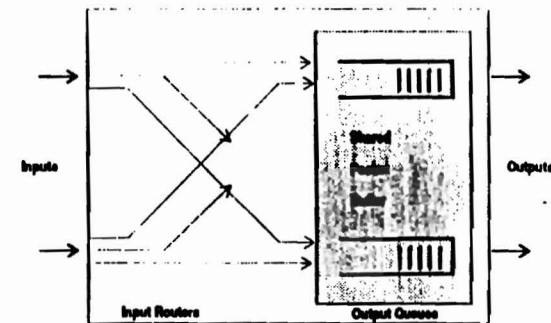


Figure 3. Logic switch element architecture

### Limitations of conventional switch implementation.

Conventional implementations typically use a single or dual port memory (see Figure 4) Internally a time-division multiplex technique is used to obtain access to the shared packet buffer.

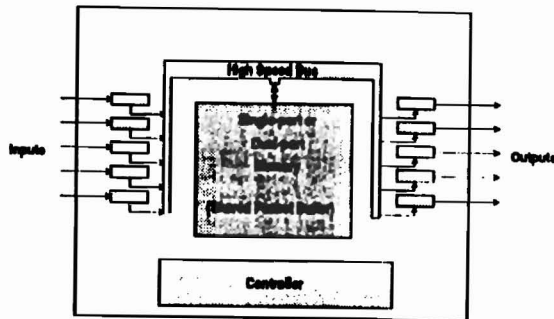


Figure 4. Conventional shared memory implementation

The number of ports, the memory and the port speed determine the required width of the internal high-speed bus: if the port speed is  $V$  and the number of ports is  $2N$  ( $N$  input- and  $N$  output ports) the required width is  $2NVZ$ . As the bit rate and/or the number of ports is increased, the bandwidth required increases rapidly: A switch with 16 ports, 1 Gigabit/port requires a memory cycle time of 13.25 nanoseconds (single-port memory) and an internal width of the "High Speed Bus" (ref. Figure 4) of 424 bits (note: this is the size of an ATM cell: 53\*8 bits!). For ATM-applications, there is no relief in increasing the bus-width: there are simply no more bits in an ATM cell which can make use of the additional bus wires. Consequently, any increase in speed and/or ports has a dramatic effect on the required memory cycle time. As future improvements in silicon technology are expected mainly in density-gain and not in factors of speed-gain, no relief of this memory cycle time problem is in sight.

### Switch-on-a-chip: novel switch implementation.

The architecture of Switch-on-a-chip is based on the separation of the data and the control flows. The bulk of packet data is fed through the data section only. It consists of the global shared packet buffer and fully parallel I/O routing trees in order to avoid the bottleneck of a shared medium for the data stream.

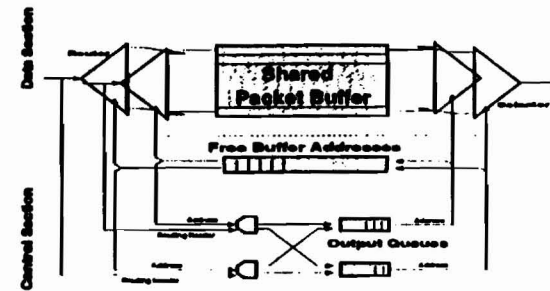


Figure 5. Basic Switch-on-a-chip architecture

The data section is controlled by a control section such that the required switching function is achieved (Figure 5).

The control section only receives a pointer and a copy of the first byte of an incoming packet: the first byte is used to route this pointer into the appropriate output queue. After this, the control section deletes the copy of the first byte. This mechanism works with short units: in the order of one or two bytes. This allows for a fully parallel implementation in the control section, thereby reducing the time needed to actually perform the routing function to approximately 12-15 clock cycles for a 16 by 16 switch element.

Separation of control- and data completely removes any interference, and both sections can be optimized for the function they carry out. As there is no multiplexing of data (in the data section) the cycle-time of the internals of the data section is equal to the system-level cycle-time (e.g., 50MHz for a byte-wide data path implementing a 400Mbps switch).

#### Data Section

The data section contains the shared packet buffer which is built from a set of shift-registers, in which the packet is serially shifted in- and out when it arrives, respectively leaves the chip. Each shift-register holds exactly one packet. At the input side up to 16 incoming packets can be routed simultaneously into free shift registers via 16 routers, one for each input. An empty shift register address for each of these routers is provided in advance by the control section. This shift register address is only renewed when an incoming packet on the respective input has 'consumed' it. At the output side up to 16 outgoing packets can be transmitted simultaneously from selected shift registers via 16 selectors, one for each output.

### Control Section

The control section consists of 17 control queues, namely one output queue per switch output and a single free queue. From the latter, addresses pointing to empty shift registers are de-queued and fed to the input routers in the data section. As soon as a new packet has been received, the shift register address from that input router is entered into the output queue indicated by the routing tag in the packet header and a new empty shift register address from the free queue is fetched. The output queues behave as FIFO's and contain only shift register addresses of packets ready for transmission via the corresponding output port. The addresses are sequentially de-queued and fed to the output selectors in the data section. After a successful packet transmission the addresses are returned to the free queue.

### Switch-on-a-chip: Scalability

The robustness of the Switch-on-a-chip architecture provides for

1. Increasing the number of ports
2. Increasing the port speed
3. Increasing the aggregate throughput
4. Hardware-assisted automatic load sharing

#### Increasing the number of ports - Port Expansion

Switches with a larger number of ports than the basic switch module can be realized by connecting several Switch-on-a-chip modules in parallel for a single stage, or cascading them for a multi stage system. Switch-on-a-chip has built-in logic to allow address-filtering at the input and activation of an output for supporting single stage expansion. For multi stage expansion every stage requires a different routing-tag in general.

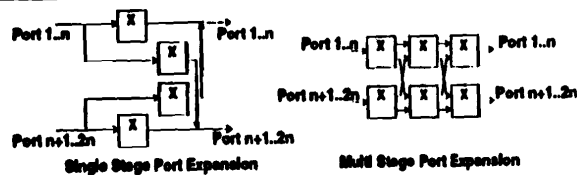
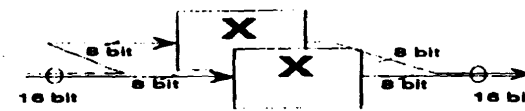


Figure 6. Switch-on-a-chip port expansion

Switch-on-a-chip has a built-in look-up table which allows routing-header bytes of the arriving packets to be shuffled. This allows multi stage routing without customizing the individual stages. While multi stage networks grow according to a logarithmic law, single stage networks grow with a square law, but have less delay. A high port count on a single chip as Switch-on-a-chip is a great advantage because it requires significant less chips for larger switches. (32 port single-stage switch requires 4 Switch-on-a-chip chips, compared to 64 chips if there would be only 4 input- and 4 output ports).

### Increasing the port speed - Speed Expansion

A unique feature of Switch-on-a-chip is to expand the actual speed of the switch ports by using multiple Switch-on-a-chip chips in parallel. Instead of an 8-bit wide port, the switch ports become then 16, or more bits wide, and a doubling, tripling etc. of the port speed is achieved.



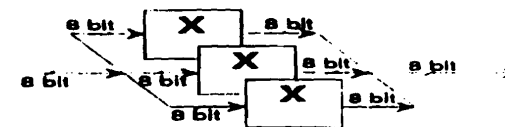
### Speed Expansion

Figure 7. Switch-on-a-chip speed expansion

Switch-on-a-chip has built-in the hardware support to build such switch systems easily. This is the preferred method to build, e.g., switches for port speeds of 622 Mbit/s and 2.488 Gbit/s. Assuming a 400 Mbit/s port speed for a single switch module, the paralleling of only two modules would be sufficient to build a 622 Mbit/s switch.

### Increasing the aggregate throughput - Performance Expansion

The aggregate throughput of a packet switch is given by the product of all ports and the port speed. The result obtained, however, has to be multiplied with a factor less than 1 to account for the fact that there is only a limited amount of output-buffer available.



### Performance Expansion

Figure 8. Switch-on-a-chip performance expansion

The actual value of this factor is a function of two variables: the internal buffer memory and the traffic characteristics: more bursty traffic will reduce the factor. Clearly, to anticipate future needs, a means of increasing the internal buffer memory, without re-designing a chip, is needed. Switch-on-a-chip has the hardware control built-in to allow cascading the internal buffer memory of multiple Switch-on-a-chip chips, such that the system behaves as if it were one chip with increased buffer memory. Control signals between the Switch-on-a-chip modules guarantee proper packet sequence.



### Automatic Load Sharing - Link Paralleling

At the system level, it is often required to support access to a high-speed backbone from multiple lower-speed links. A typical example is the access to a 622 Mbps ATM link to carry the non-local traffic of multiple (more than 4) Mbps ATM access links.

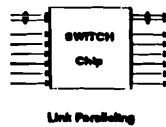


Figure 9. Switch-on-a-chip Link Paralleling.

The usual means of addressing this problem is that multiple lower speed ports are multiplexed together into the 622 Mbps link. This, however, requires careful bandwidth management, and rearranging streams when large bandwidth requests must be accommodated. Switch-on-a-chip has a built-in feature, dubbed *link paralleling*, which manages the bandwidth on such links fully with hardware. I.e., 2 or 4 physical Switch-on-a-chip ports can be combined to support a double- or quadruple-speed link, without software to control which connection is allocated to a physical Switch-on-a-chip output port.

### Combining the expansion modes - The scope of the Switch-on-a-chip application space.

The four mentioned expansion methods can be combined freely to design a switch-fabric. The port- and performance expansion methods require the external manipulation of Switch-on-a-chip's control signals to provide maximum flexibility in functionality: e.g. port expansion can also be used to support multiple priorities.

In Figure 10 on page 10 this richness of the Switch-on-a-chip application space is shown. (link paralleling is treated as a special case of speed expansion).

### Multicast Support

Switch-on-a-chip supports to build high speed communication networks which support applications that feature a heterogeneous mix of voice, data and video traffic. Typically such systems require the capability of handling multipoint connections for services such as video distribution and teleconferencing. Switch-on-a-chip provides a flexible multicast capability: it is possible to send a copy of a packet to all (broadcast) or only a subset (multicast) of the switch module's output ports. In order to conserve buffer memory, only one packet storage location is used, from which multiple copies are sent. The activation of a multicast connection is done through the packet routing header, and a dynamically programmable table internal to the Switch-on-a-chip module.

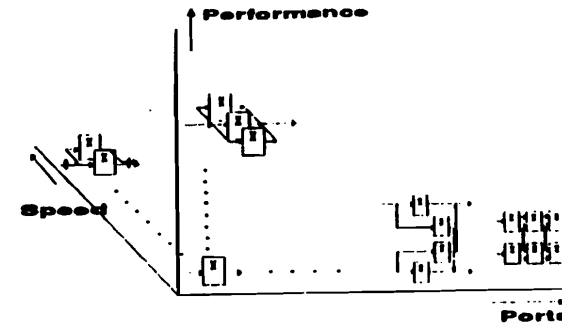


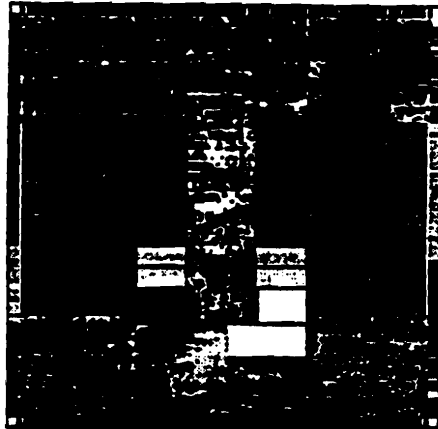
Figure 10. Scope of the Switch-on-a-chip application space.

### References.

1. W.E. Denzel, A.P.J. Engbersen, I. Iliadis and G. Karlsson, "A Highly Modular Packet Switch for Gb/s Rates", in *Proc. of ISS'92*, Yokohama, Japan, A8.3, 1992
2. W.E. Denzel, A.P.J. Engbersen, I. Iliadis, "A flexible Shared-Buffer Switch for ATM at Gb/s Rates", accepted by *Computer Networks & ISDN Systems*.

Dr. A.P.J. Engbersen  
Mgr. High-Speed Networks  
IBM Research Division  
Zurich Laboratory  
Saumerstrasse 4  
CH - 8803 Ruschlikon  
Switzerland  
Email: apj@zurich.ibm.com

**"Switch-on-a-chip"  
IBM's ATM Switching Technology**



IBM  
Research  
Division  
Zurich  
Laboratory

---

## **Broadband-ISDN / ATM Switching**

### **PROBLEM**

Each 155 Mb/s line may carry 350 000 ATM Cells/sec.

### **REQUIREMENT**

ATM switch must handle Megacells/s or Gigacells/s

### **DIRECTIONS**

High throughput requirement dictates:

Simple, efficient buffer management

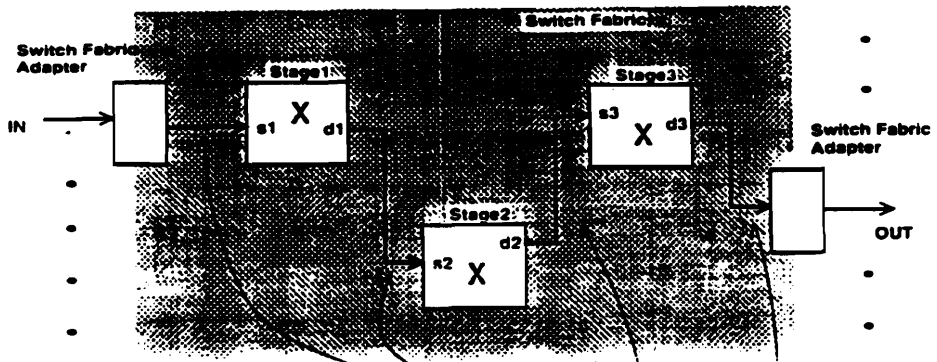
Switching in VLSI hardware

Topologies with high degree of parallelism

Self-Routing

Typically cell-oriented

# Packet Self-Routing Concept



User Data | Hdr

c | d3 | d2 | d1

User Info | c | d3 | d2 | d1

User Info | s1 | c | d3 | d2

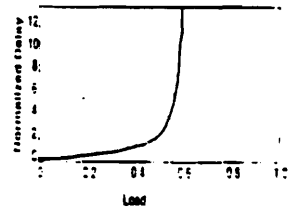
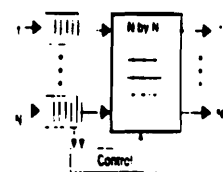
User Info | s2 | s1 | c | d3

User Info | s3 | s2 | s1 | c

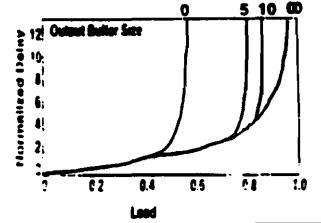
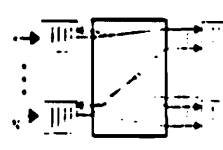
IBM Research Division - Zurich Laboratory

## Performance of Different Switch Topologies

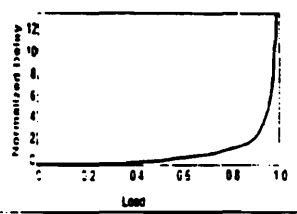
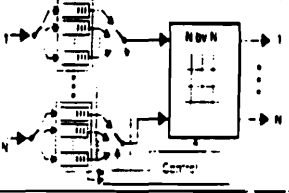
Crossbar with Input Queueing



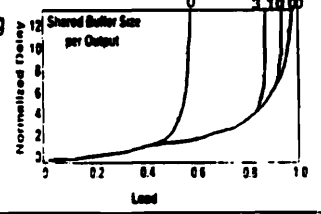
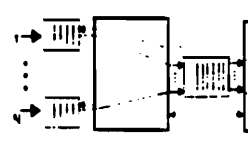
Self-route with Finite Output Queueing



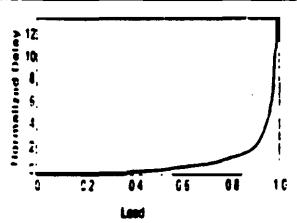
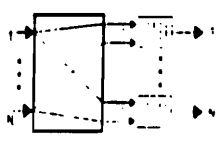
Crossbar with Multiple Input Queueing



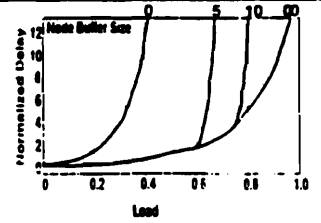
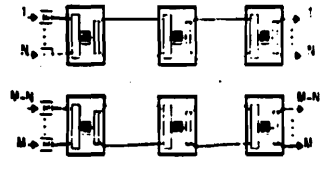
Self-route with Shared Output Queueing



Self-route with Output Queueing



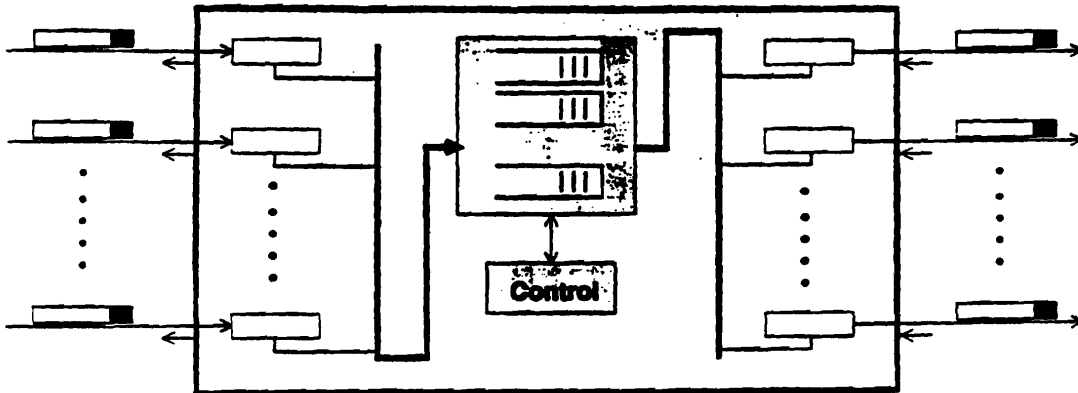
Multistage Queueing



IBM Research Division - Zurich Laboratory

# Conventional Switch Design

First Route, then Store

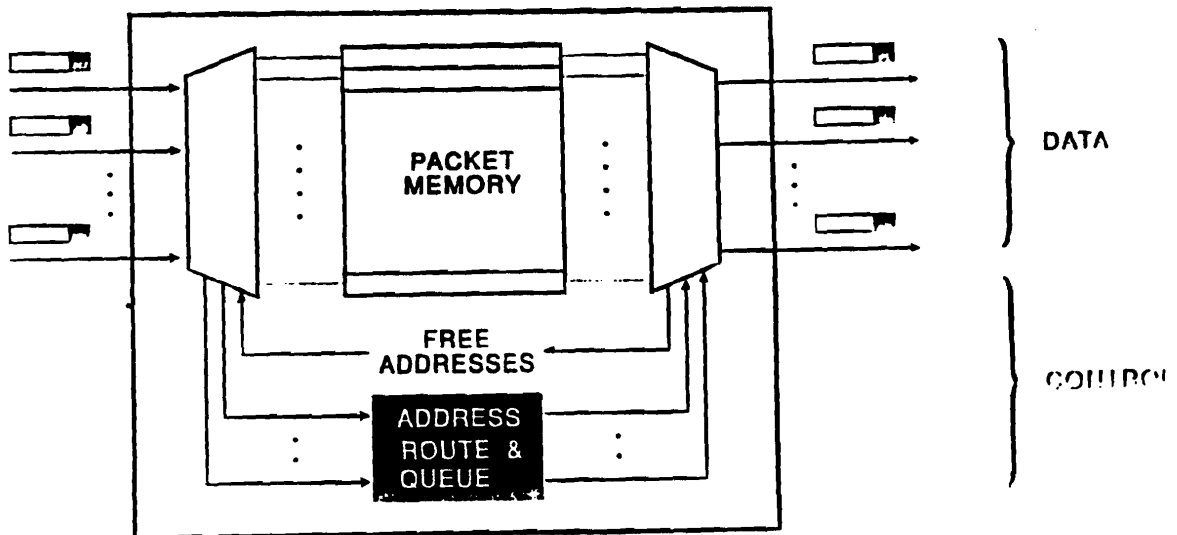


- "Bus on a Chip"
- Bus: several hundred wires
- Does not scale to required performance levels

IBM Research Division - Zurich Laboratory

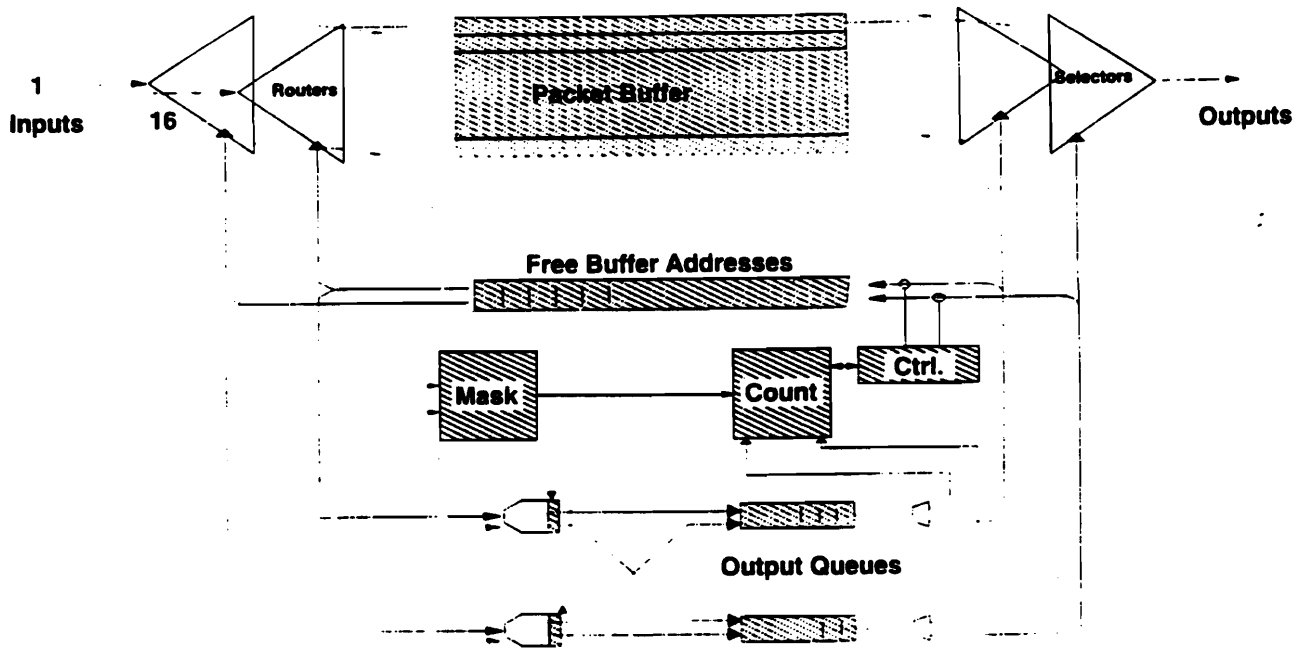
# PRIZMA ARCHITECTURE

Store while Route : Full Parallelism !



- Separation of data & control
- Control speed fraction of data speed

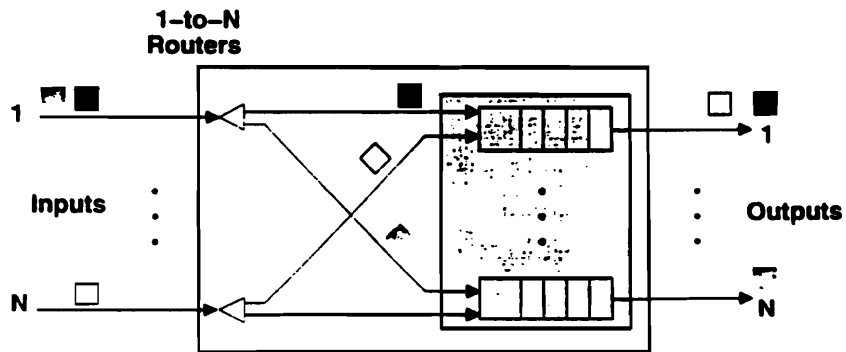
# PRIZMA Basic Operation



IBM Research Division - Zurich Laboratory

PRIZMA Basic Operation

# PRIZMA - Switch Basic Structure

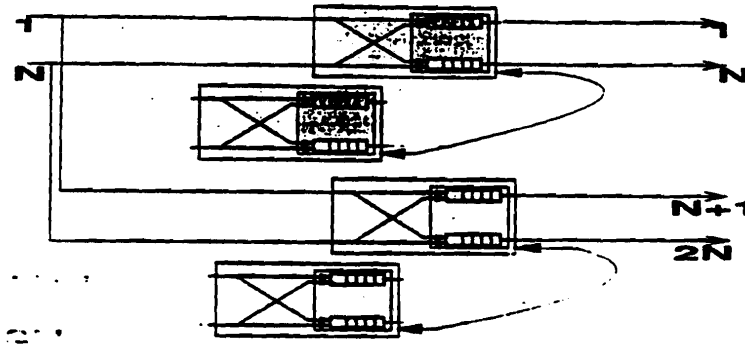


- Non-Blocking
- Queueing to resolve output contention
- Self-Routing

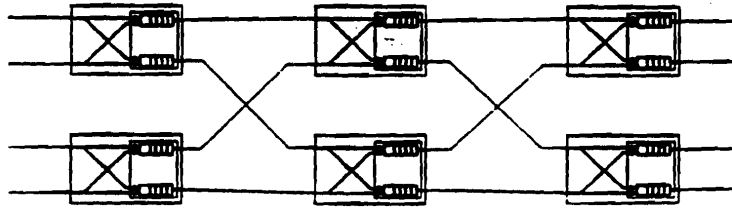
IBM Research Division - Zurich Laboratory

PRIZMA Basic Structure

# PRIZMA - Switch Port Expansion



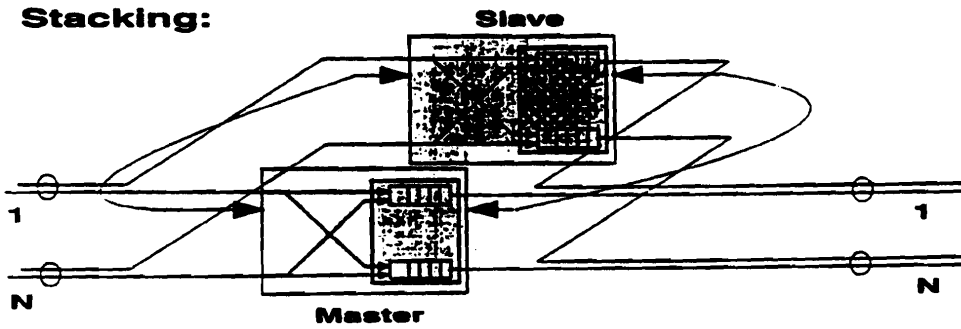
Single Stage: growth with square law



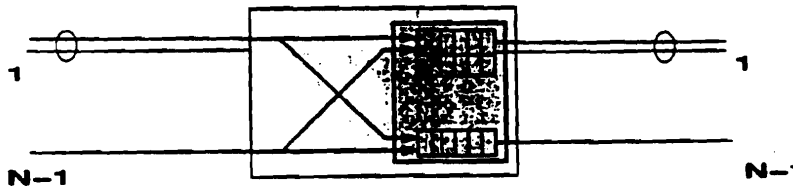
Multi Stage: growth with  $n \cdot \log n$  law

# PRIZMA - Switch Speed Expansion

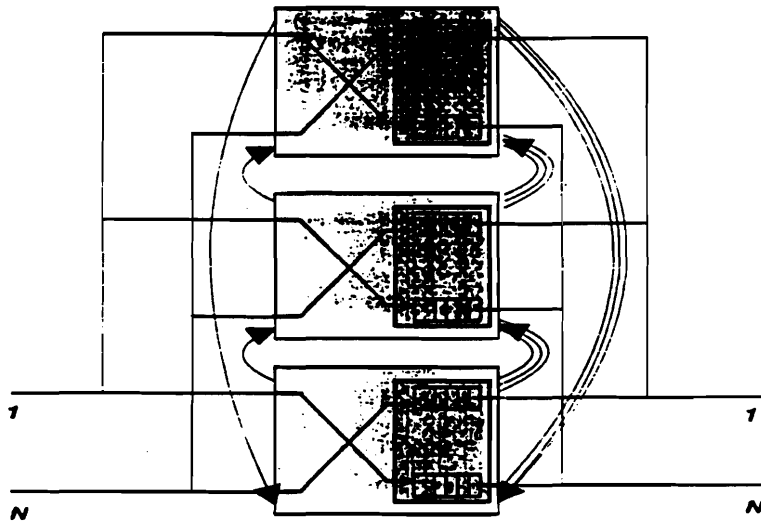
Stacking:



Link Paralleling:



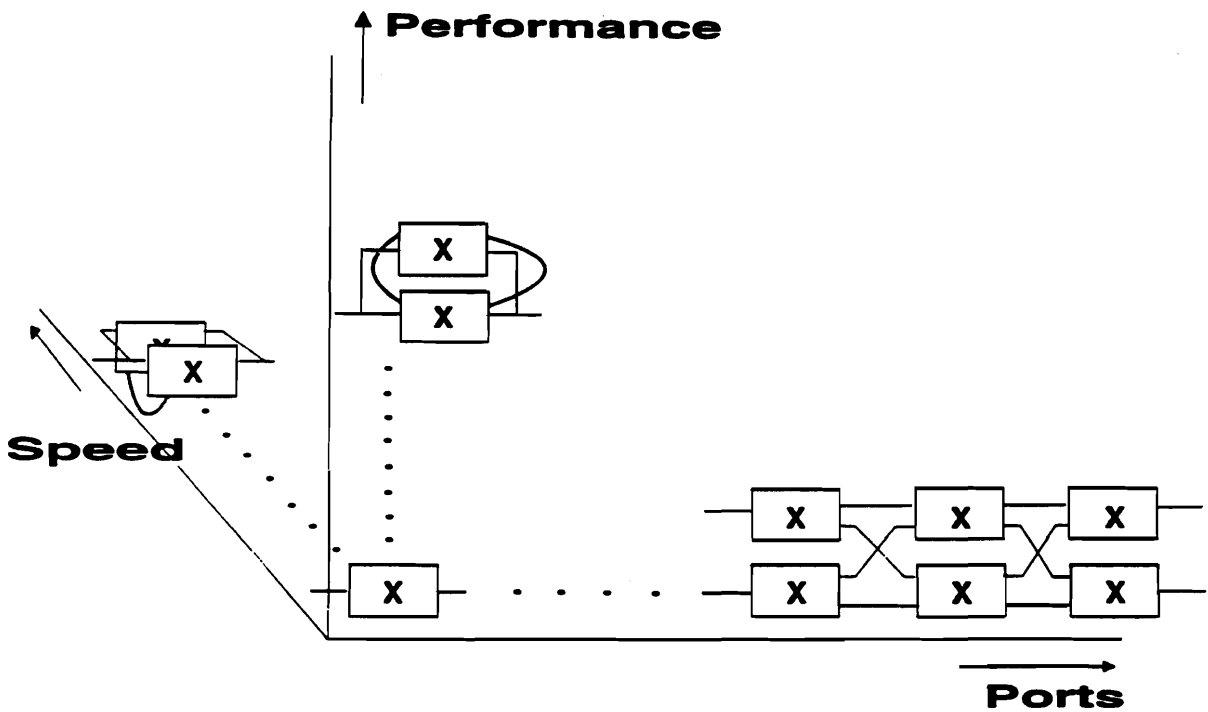
# PRIZMA - Switch Performance Expansion



IBM Research Division - Zurich Laboratory

prizma@zurich.ibm.com

# PRIZMA Switch Application Coverage

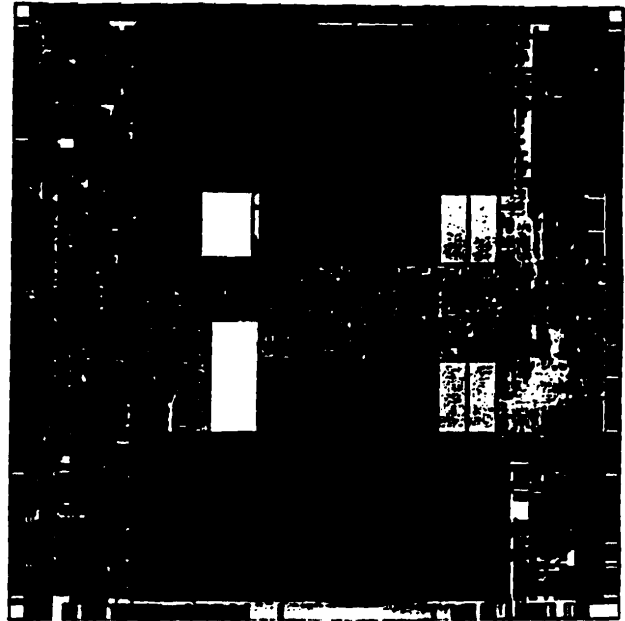
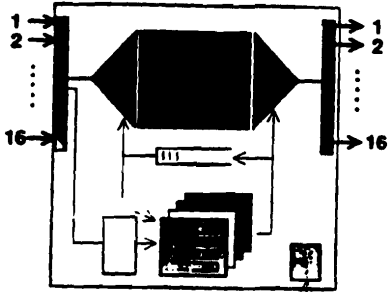


IBM Research Division - Zurich Laboratory

prizma@zurich.ibm.com

# PRIZMA Chip Implementation

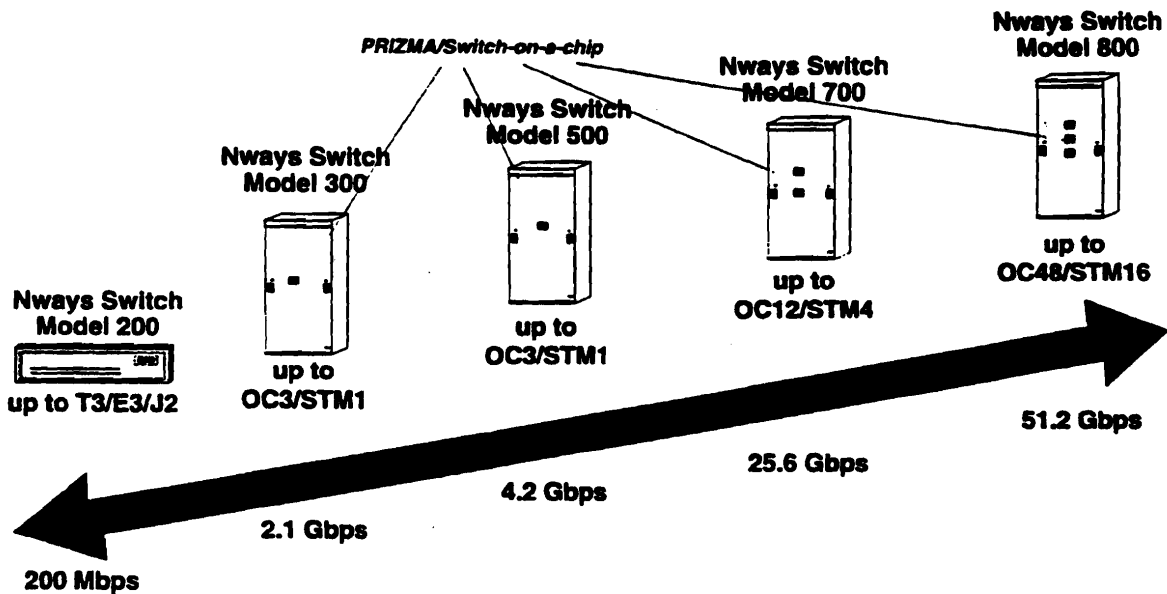
6.4 Gbps / 15Mcells/s



- 16 by 16
- 400 Mbps/port
- 128 cell locations
- 472 I/O's
- 2.4 Million transistors

IBM Research Division - Zurich Laboratory

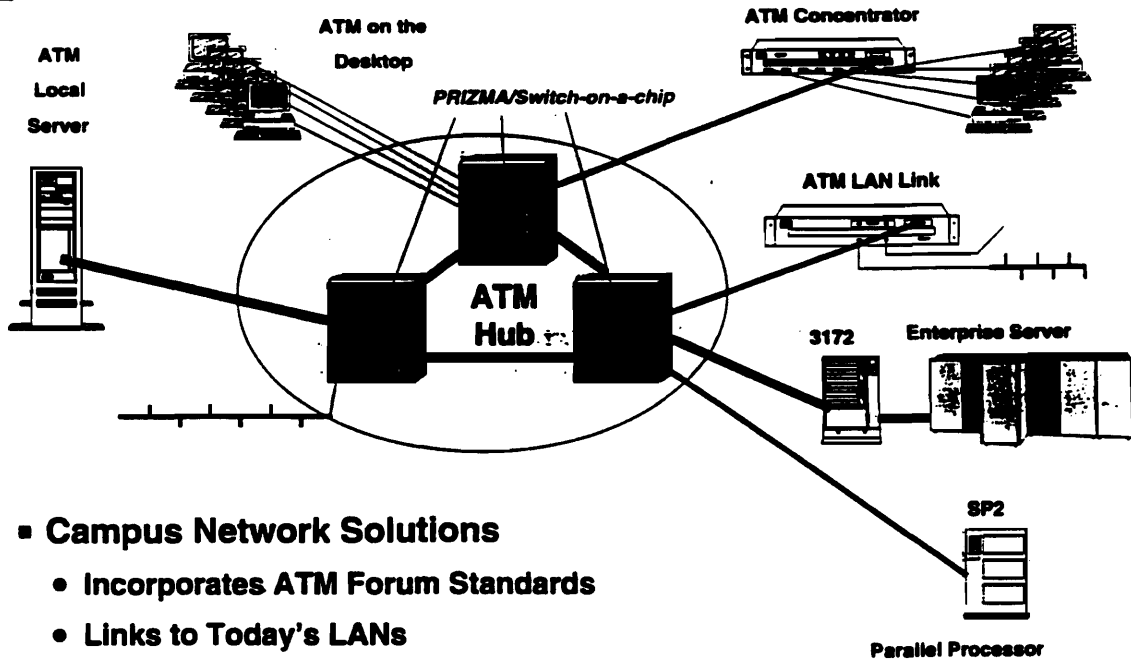
## Nways BroadBand Switch



Same Access Services : CES, Frame Relay, Voice/Fax, ATM, SMDs/CBDS, X.25, HSSI/DXI, HDLC, ISDN, LAN Routing/Bridging, ...

Common Technology : BBNS, Hardware (ATM Switch, Trunk/Port Adapter), Software, Network Mgt., Node Operations





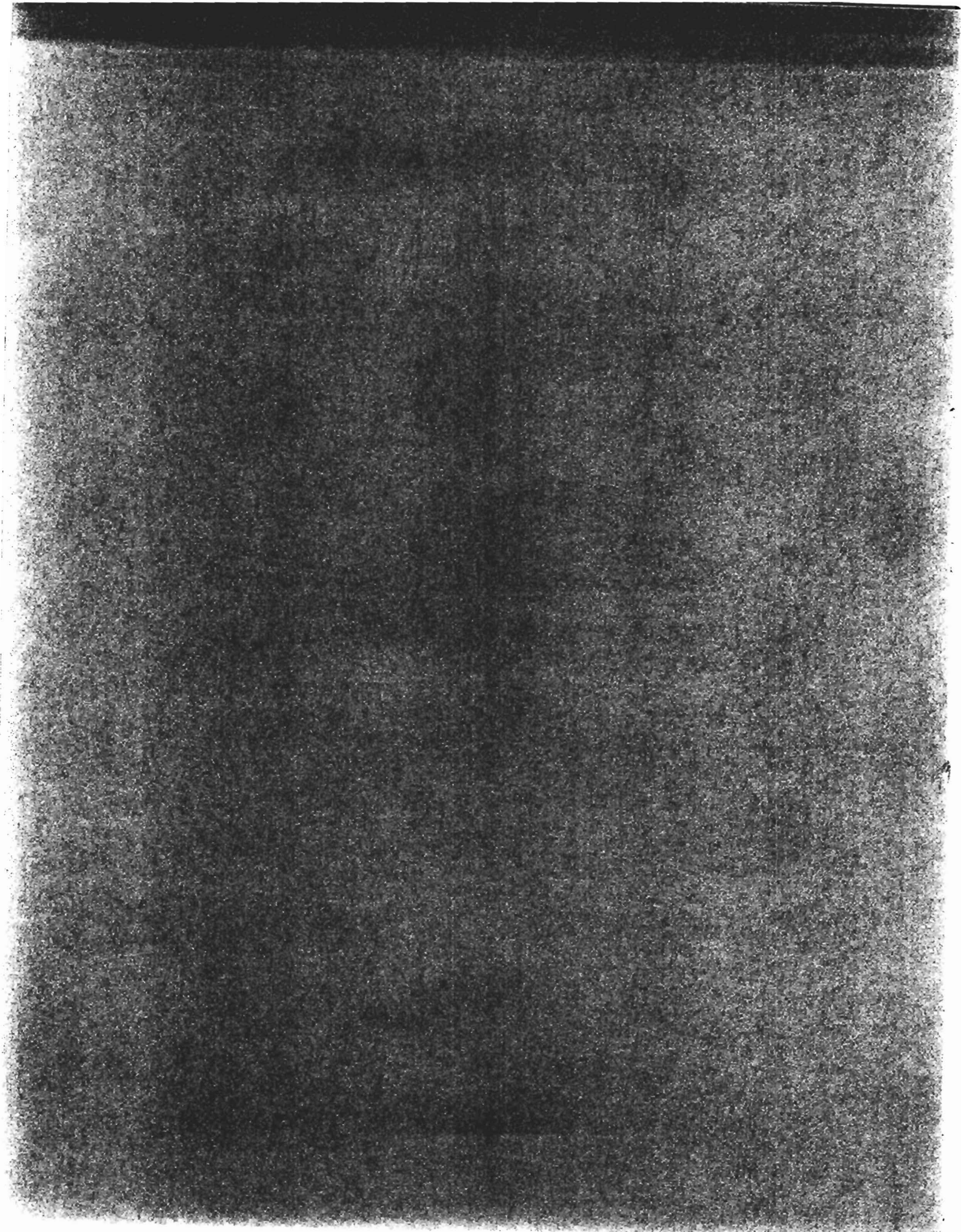
- **Campus Network Solutions**
  - Incorporates ATM Forum Standards
  - Links to Today's LANs
  - Supports Existing LAN Applications
  - Provides Virtual LAN Support



**S5-7**

**"Phoenix Switch & Bell Labs Switch Research"**

**(Andre Wiesel - EPFL)**



**Phoenix Switch &  
Bell Labs Switch Research**



**André Wiesel**

**Swiss Federal Institute of Technology  
Computer Networking Laboratory(LTI)  
Prof. C. Petitpierre  
1015 Lausanne**

**Tel : +41 21 693 47 13  
Fax: +41 21 693 66 00  
e-mail : wiesel@ltsun.epfl.ch  
http://diwww.epfl.ch/w3lti**

**Phoenix Switch &  
Bell Labs Switch Research**



### The PHOENIX Chip

- 2 by 2 switching node.
- Broad range of application in broadband switching.
- Well suited for ATM.
- Developed by the LTI in collaboration with the AT&T's Bell Labs.

**Phoenix Switch &  
Bell Labs Switch Research**



### Topics

- The PHOENIX Chip
- The ATM Layer Interface (ALI) Chip
- The ATM Switch
- The ATM Host
- Simulations

**Phoenix Switch &  
Bell Labs Switch Research**



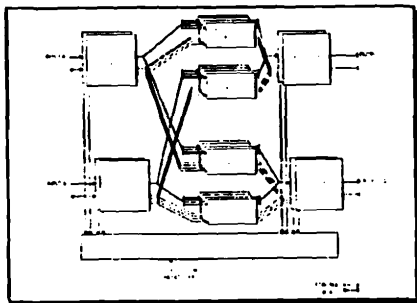
### PHOENIX Chip Features

- Self routing.
- 8 bit wide parallel data port
- 320 Mbits/s per port @40 MHz
- Synchronous protocol with packet flow control.

### PHOENIX Chip Features (cont'd)

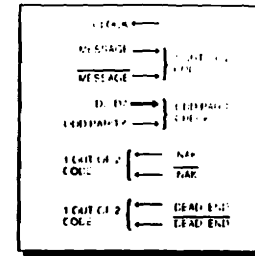
- Variable packets from 10 to 81 octets.
- Fault tolerant error detection + adaptative routing)
- Four priorities.
- 8 Kbytes FIFO buffers.

### PHOENIX Building Block



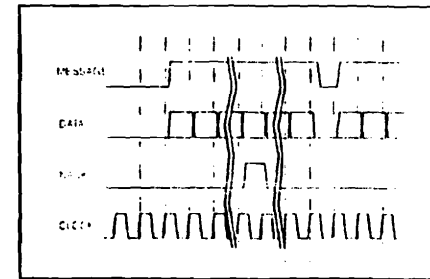
### PHOENIX Protocol

- Signals on a link:



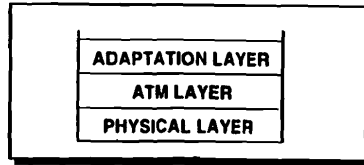
### PHOENIX Protocol (cont'd)

- Link Protocol :



### The ALI Chip

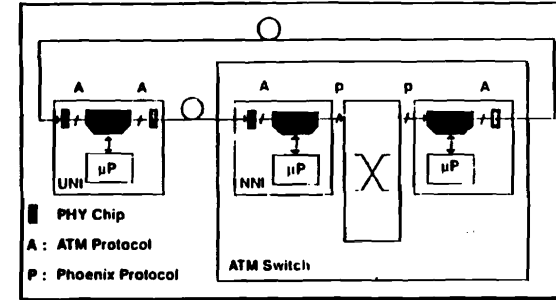
- ALI stands for  $\Delta$ TM Layer Interface



- Applications :
  - Network Node Interface (NNI)
  - User Network Interface (UNI)

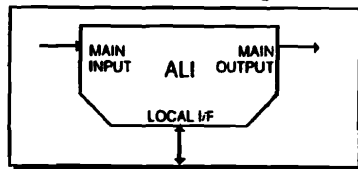
9

### The ALI Chip



11

### The ALI Chip



- Main Input & Output supports :
  - ATM Protocol
  - PHOENIX Protocol
  - AT&T's T7650 (Phoenix) chip
  - UTOPIA Chips

10

### ALI Chip Features

- ATM Header translation
- Removal or addition of Phoenix headers (1 to 4 bytes)
- 4 priorities
- Policing of input channels ( leaky bucket algorithm )
- Internal look-up table for up to 1023 channels

12

Phoenix Switch &  
Bell Labs Switch Research

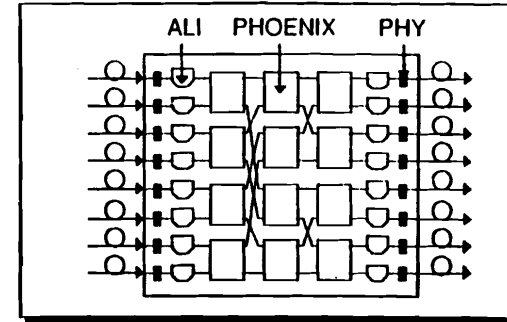
ALI Chip Features (cont'd)

- 400 Mbit/s data rate at input and output
- Flow control in Phoenix mode
- 50 MHz local bus
- Statistics gathering
- Independent clocks for the three ports

13

Phoenix Switch &  
Bell Labs Switch Research

The ATM Switch

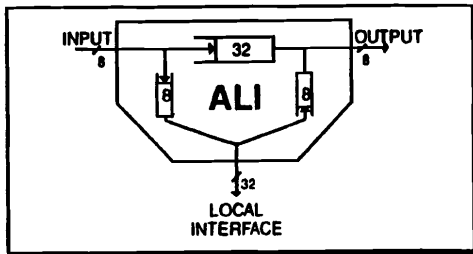


14

Phoenix Switch &  
Bell Labs Switch Research

ALI Chip Features (cont'd)

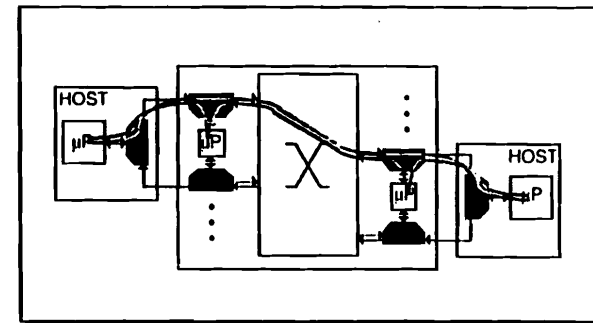
- Internal fifos for rate matching



14

Phoenix Switch &  
Bell Labs Switch Research

The ATM Switch (cont'd)



15



### Phoenix Switch & Bell Labs Switch Research



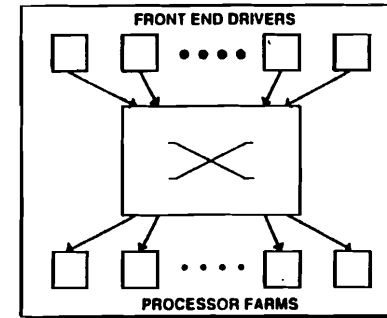
## The Processing Elements

- PHOENIX and/or ATM interface with PCI Local Bus Controller
- RIO II based processing element
- SyncC++ native
- Planned for february 1995

### Phoenix Switch & Bell Labs Switch Research



## Event Building Simulations

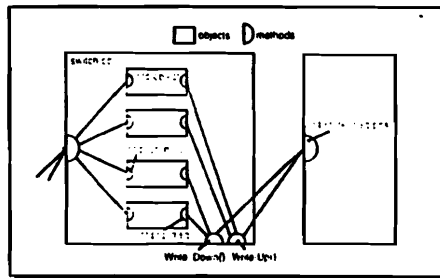


### Phoenix Switch & Bell Labs Switch Research



## PHOENIX C++ Model

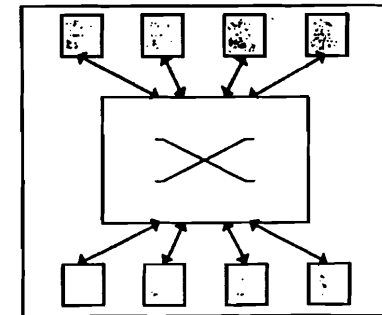
- The switch is described in pure C++.



### Phoenix Switch & Bell Labs Switch Research



## Event Building Simulations (cont'd)



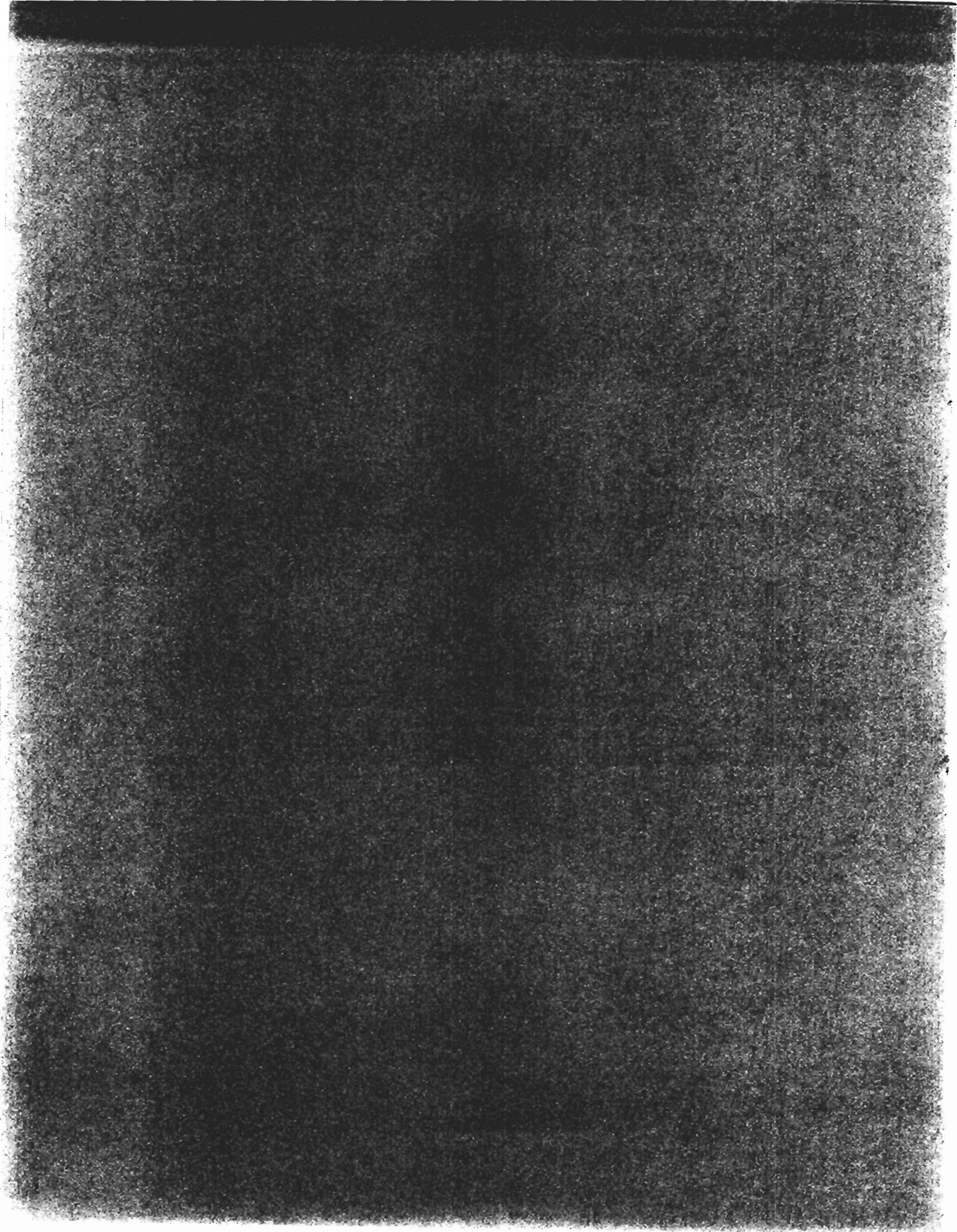


S5-8

**"Switches for Point to Point Links using OMBHC Technology"**

**(Ernst Kristiansen - SINTEF)**

OMBHC is a point to point link technology under development in project funded by the European Union. The OMBHC technology has a flexible packet format and supports flow control, which makes OMBHC suitable as an alternative physical layer for other higher level protocols, like SCI and ATM. Advanced routing chips with a worm-hole routing mechanism, and these routing chips are useful building blocks for larger switches for point to point links. The OMBHC technology is being standardized in the IETF P333 working group.



# Switches for Point-to-Point Links using OMI/HIC Technology

Ernst H. Kristiansen, Geir Horn, Svein Linde

SINTEF Instrumentation, Forskningsveien 1, P.O. Box 121 Blindern, 0311 Oslo, Norway.

## Abstract

OMI/HIC is a point-to-point link technology under development. It has a flexible packet format and supports flow control, which makes OMI/HIC usable as an alternative physical layer for other higher level protocols, like SCI and ATM. Routing chips with advanced routing strategies are made, and these routing chips are useful building blocks for larger switches for point to point links.

## 1. INTRODUCTION

Up to now Data Acquisition (DAQ) systems have used high speed buses between equipment connected to the same backplane and lower speed connections to more distant equipments. The performance of the traditional backplane buses are limited by signal transmission delays and the impedance mismatch across a backplane. The signal quality is affected by reflections caused by multiple connectors, as well as by load variation. A backplane bus can only transmit one symbol at a time between all connected nodes and therefore easily becomes a bottleneck in multiprocessor systems. The use of complex communication protocols in software to avoid corruption or loss of data have often reduced the potential performance of the system.

Point-to-point connection is a very promising technology that could be applied to both in and between multiprocessor systems. Point-to-point links avoid the "one-at-a-time" limitation of the shared bus. A large number of requests can potentially be outstanding at the same time in systems based on point-to-point links, which is a basic requirement for high performance parallel systems. In addition, point-to-point links also reduce the non-ideal-transmission-line problem. Thus, the clock rate can be much higher for point-to-point links than for buses.

Almost all the new protocols for interconnection networks are based on point-to-point links. Commonly known examples are ATM<sup>1</sup>, Fibre Channel for communication

purposes, and for multiprocessor systems, SCI [1]. The European Union funds the ESPRIT<sup>2</sup> OMI/HIC project forming the basis for the IEEE P1375 working group founded in July 1993 to develop a standard for Heterogeneous Interconnect (HIC). Both SCI and OMI/HIC data links are based on the assumption that the links should be high performance and be able to support more than the requirements inside or between data systems. The current performance of SCI, running over meters of shielded flat cable at 1 Gigabyte per second, is a clear evidence that point-to-point data links will provide the system builder with a new opportunity to reach a higher performance level.

This paper will in section two discuss some general aspects of point-to-point technology and the various standards available. Some background on OMI/HIC link technology is presented in section three, before we present the OMI/HIC router in section four. Finally, section five shows how an OMI/HIC network may work as a transport layer and form an interconnection for systems based on the other high level protocols.

## II. POINT-TO-POINT TECHNOLOGY

### A. Common characteristics

When using point-to-point links, it is in general not possible to provide a direct physical connection between every pair of devices, and the data must be routed through intermediate nodes. For a successful data transmission to take place, one must then either configure all intermediate nodes or one must make the data self-routing by prepending a header containing the address information to the data. Thus, the data links we focus on have some characteristics in common:

- Point-to-point connection between only two devices.
- Data sent in packets with header information. Some packet outlines are shown in figure 2.1
- The data link standards can be split in layers: *physical layers*, *packet layer* and at the top an *application layer*.

<sup>1</sup> Scalable Coherent Interface.

<sup>2</sup> European Strategic Programme for Research and Development in Information Technology.

<sup>3</sup> Open Multiprocessor systems Initiative - High performance Heterogeneous Interprocessor Communication.

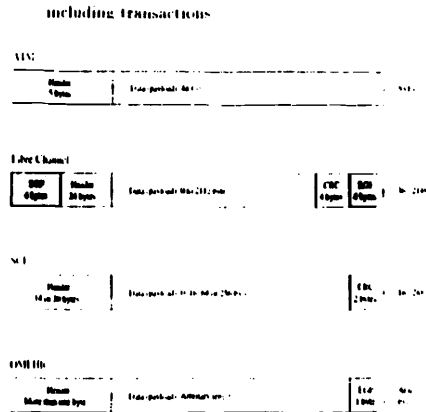


Figure 2.1. Packet formats for some of the point-to-point protocols. All packets have a header and a data part. Some protocols include a Cycle Redundancy Check (CRC) in each packet. The OMI/HIC protocol has an End-of-Packet (EOP) byte while the Fibre Channel has both Start-of-Frame (SOF) and End-of-Frame (EOF) bytes.

### B. Different technologies

If you want to standardize on one point-to-point link for all application fields, you will need to sacrifice on something, because there is no point-to-point link that is the best and obvious choice to cover the whole range of applications in computer systems. We will here only briefly discuss and outline the intended field of applications for the various point-to-point technologies; for a more thorough treatment and available hardware consult [2].

- **ATM** is intended to be used in Wide Area Networks (WAN) and is a telecommunication standard for point-to-point links with speed currently maximum at 155.52 Mbit/s. The standard has defined several ATM Adaptation Layers for high level support. The physical layer is, however, not yet standardized and use of other physical media will not conflict with the standard. Media usable to make efficient switches and large systems will be of interest when these problems should be solved.
- **Fibre Channel** was developed for memory-to-memory and Local Area Networks (LAN) applications. Transfer speeds are defined from 132 Mbaud up to 1062.5 Mbaud on either fiber or coax cable at distances ranging from less than 100 m up to 10 km.
- **SCI** was originally designed for substituting buses inside computers, and it therefore has bus-like functions. Protocols for read, write and atomic operations are supported in hardware together with a CRC in

each packet, which automatically will be retried if errors occur. SCI has protocols to handle each coherence through a directory based scheme with doubly linked lists. It uses unidirectional lines and it is possible to configure nodes efficiently in rings. Forward progress is guaranteed in the rings, and there are buffer capacity to both send and receive packets at full speed in every node. SCI is defined up to a speed of 1 Gbyte/s on a parallel interface with 18 signals: 16 data, 1 flag and 1 clock.

- **HIC**-links were intended for message passing in multiprocessor systems and are discussed in section three.

### C. Changing transport protocol

When bridging packets between different standards, two options exist:

- The packets can be collected to build up complete messages and sent through the network. This is normal in gateways for WAN.
- The packets could simply be put into another format and use this format as a transport layer. This new format could be the final one or the packet could be changed back to its original format at the destination.

We will focus on the second feature, which is possible for packets following SCI, OMI/HIC, ATM and Fibre Channel. All these packets have routing information in the packet header, which easily could be used in another network, see figure 2.2.

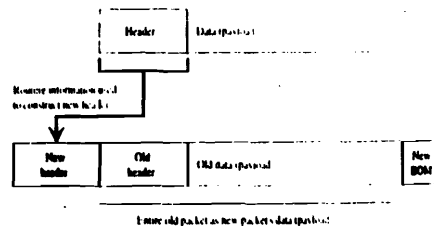


Figure 2.2. Embedding a packet from one protocol into a packet of another for transport. The routing information from the original header is used to determine the routing information in the header of the transport protocol. The transport protocol may also require some End-of-Message (EOM) bytes, such as CRC, EOP or EOF, to be appended.

Limitations are caused by the fact that it is simple to put a shorter packet into a longer one, but not opposite. ATM has the shortest packet of all. Thus, it is easiest to put ATM packets into SCI packets or Fibre Channel packets than the other way around. It may be possible to put SCI packets into Fibre Channel packets, but that may not be a good choice since the Fibre Channel protocol is much more complicated than the SCI protocol. However,

<sup>\*</sup>This work is supported in part by the ESPRIT 894: OMI/Macramé project.

<sup>1</sup>Asynchronous Transfer Mode: Packet format chosen by the CCITT as basis for the Broadband Integrated Services Digital Network (B-ISDN).

OMI/HIC has not defined a fixed packet length, and can for that reason be the transport layer for all the others. At the moment there are projects in Europe making use of OMI/HIC as transport layer for SCI, ATM and Fibre-Channel.

Figure 2.3 gives an illustration of how physical layers are usable to transport packets from different packet layer protocols.

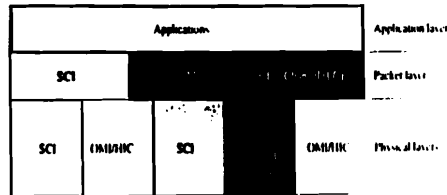


Figure 2.3. Physical layers that are usable to transport packets from different packet layer protocols.

### III. HIC LINK TECHNOLOGY

The OMI/HIC technology is optimized towards relatively local communication. Care is taken to ensure the maximum of compatibility with the protocols used on copper and optic media. With copper intended to be used between chips, copper or optic between boards, and optica between cabinets and rooms according to the design of each particular application. To allow integration of a large number of links on a single chip the protocols are so designed that the links can be implemented using a relatively small amount of silicon area. This makes construction of large packet router chips possible.

Instead of trying to design links with highest possible speed, the emphasis have been on excellent performance to cost ratio and simplified system engineering. The following links have been developed:

- 200 Mbit/sec Data-Strobe (DS) links [3]
- 1 Gbit/sec or 3Gbit/s High Speed (HS) links [4]

There is also a special fiber optic version of the DS links called TS<sup>2</sup> links. Table 3.1 summarize the various OMI/HIC links.

Only serial bidirectional links are specified as these are the simplest forms of point-to-point links. This relaxes the performance requirement on the individual links, since engineering of a large number of serial links are easier. It also demands less wires, has reduced or no skew constraints and easier clocking.

The links support flow control for the cases when a packet may be unable to proceed because the required output is already in use. Data continues to flow until all buffers along the packet path are filled, then the flow of data is stalled. This in contrast to protocols which may

discard incoming data, an approach which increases the protocol complexity and may degenerate the system to the state where most connections are carrying packets to their point of destruction. Buffering the incoming packets is not an alternative as it may increase system cost and make the hardware dependent on the chosen packet length.

The OMI/HIC links use a credit-based flow control algorithm. A flow control character is sent from the receiver to the sender for each  $n$  characters of credit, free characters, in the receive character buffer. Once the sender has transmitted a further  $n$  characters it waits until it receives another flow control character. A consequence is that a link must provide a minimum receive buffer of  $n$  characters at either side. The provision of more than  $n$  characters of buffering ensures that in practice the next flow control character is received by the sending side before the previous batch of  $n$  characters has been fully transmitted, so the flow control does not restrict the maximum bandwidth of the link. The value of  $n$  is specific to each link type and is shown in table 3.1.

### IV. HIC ROUTING TECHNOLOGY

OMI/HIC links may be implemented using a small area of silicon, thus allowing integration of a large number of links on a single chip. The router now available, ST C101 from Lumes [5], has integrated a 32-way crossbar plus area for 32 links with buffers and packet routing logic on a single CMOS chip. Maximizing the valency of a router is beneficial as the router's overall throughput is determined by the number of links operating concurrently. The use of high valency routers also reduces the number of stages in a network, which again reduces the network latency and eases the engineering.

Packet routing chips may be combined in a variety of ways to construct packet switches and networks. Some links on some packet router chips are used to connect to the nodes which supply and consume the packets; the other links are used to connect the packet router chips to each other in a network.

The ST C101 packet routing chip supports advanced techniques such as adaptive routing and header deletion. Adaptive routing is used when a set of consecutively numbered links have been grouped so that a packet routed to any link in the set would be sent down any free link of the set. This improves network performance in terms of both latency and throughput.

The routers are designed to do *internal routing* [6] whereby each output link is assigned an interval, or a range, of labels against which the packet header is compared to determine the output link. The number of header bytes decoded by the router can be configured to either one or two bytes, but the number of destination nodes that can be reached are in either case limited to 256 or 65535. For this reason each output link from the router can be set up to do header deletion, revealing a possible second header for further routing. In this way, one may route a packet through several sub-networks as shown in

Table 3.1  
THE VARIOUS OMI/HIC LINKS

	DS-Link	HS-Link	HS-Link
Maximum data rate (Mbaud)	200	1000	3000
Minimum data rate (Mbaud)	10	700	2000
Power consumption at maximum data rate (mW)	100	300	600
Technology	CMOS 5V 0.8 micron	CMOS 3.3V 0.5 micron	BICMOS 3.3V 0.5 micron
Transmission type	meter range 10's of meters 100's of meters	Single ended Differential 1S Multimode fiber	Single ended Monomode fiber Differential Monomode fiber
Silicon area (square mm)	0.2	1	2
Number of wires per bidirectional link	1	2	4
Maximum number of links per chip	2	12	16
Flow control character value	5	12	2

Figure 4.1

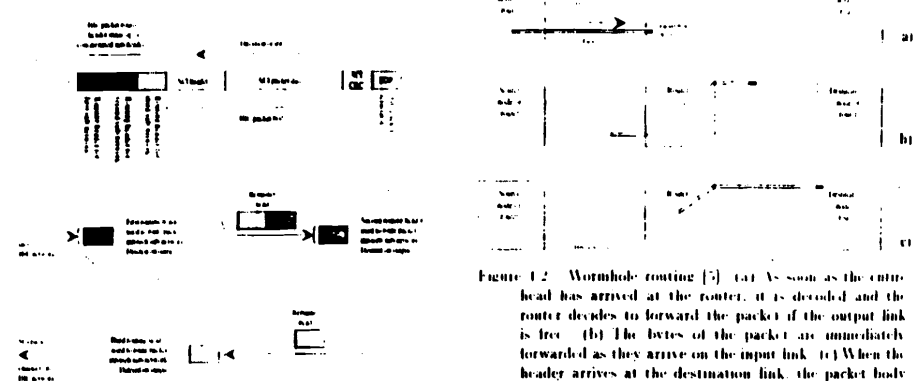


Figure 4.1. Header deletion supports routing of packets through an hierarchical composition of networks. Here OMI/HIC is used to transport an SCI packet.

The router chip supports *wormhole routing* [7]. This means that the router forwards the packet as soon as the output link is determined from the addressing information in the packet header provided that the output link is free. When passing through the network, the packet header creates a temporary path through which the data flows. As the end of the packet is pulled through, the path vanishes. Thus a packet may be active in many links, routers and nodes simultaneously. The packet header may even be received by the destination node before the whole packet has been transmitted by the source. An illustration of this concept is given in figure 4.2.

Figure 4.2. Wormhole routing [7]. (a) As soon as the entire head has arrived at the router, it is decoded and the router decides to forward the packet if the output link is free. (b) The bytes of the packet are immediately forwarded as they arrive on the input link. (c) When the header arrives at the destination link, the packet body may be active in many other entities in the network. The links not used by the packet are free to be used by other packets.

The router may also be configured to implement a two phase routing algorithm known as *randomized routing* [8]. The packet is then first sent to a random intermediate destination which forwards the packet to its final destination. This routing technique may maximize network capacity and minimize network delay under conditions of heavy load.

### V. SYSTEM INTERCONNECT

Within a network, a router is a key element to interconnect a wide range of nodes together. Improved system bandwidth and lower latency may be achieved in

<sup>2</sup> Three-of-Six encoding

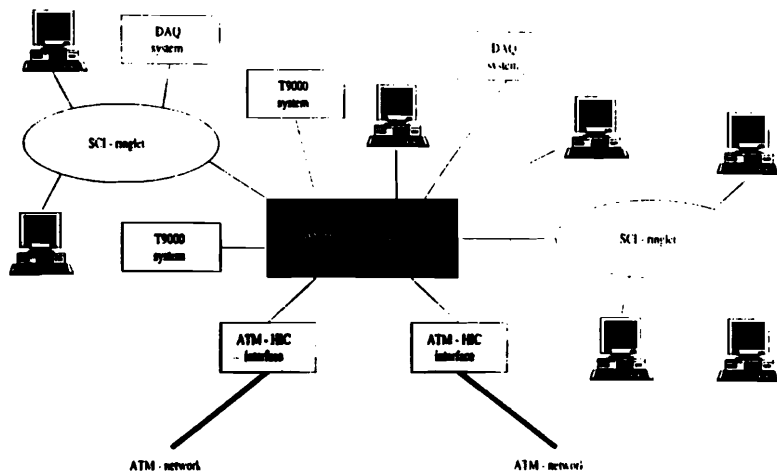


Figure 5.1. An example of connecting systems together using more than one interconnect standard

large systems containing thousands of nodes using routers. Different network protocols do in general require different types and architectures of switches. However, OMI/HIC's serial links with flow control and wormhole routing could be alternative physical layers for SCI, Fibre Channel and ATM in applications where high connectivity between a high number of nodes is needed. The flexible format also creates a possibility to share a OMI/HIC switch between SCI, Fibre Channel and ATM. A system of this kind is illustrated in figure 5.1.

The latency through the OMI/HIC network will in general, add very little to the time taken to transmit a single packet through the interconnect. However, it is possible for congestion to occur as a result of several packets competing for the same destination. The OMI/HIC network's flow control will result in packets being delayed, rather than being discarded, which greatly ease the management function.

The effects of a delay can be minimized by ensuring sufficient buffering at the point of competition for the maximum number of packets which can compete in this way. If the traffic has a predicible or random distribution, then nearly optimal performance can be achieved by using much less buffering. The effects of the delay can also be greatly reduced by using multiple physical links to appreciated units. One may even use multiple networks, which can also provide a very high degree of fault tolerance.

## VI. CONCLUSION

This paper introduced the OMI/HIC technology and showed why it may be useful as a transport layer for other high level protocols currently available. The OMI/HIC

standard provides higher connectivity and has enabled construction of an advanced efficient routing chip which again make an easier and more scalable switch and network design possible. Various point-to-point standards can also be combined to optimize the application and use of equipment.

## REFERENCES

- [1] IEEE, *The Scalable Coherent Interface*, 1992, Standard 1596.
- [2] Ernst H. Kristiansen, Bin Wu, and John W. Rothner, "High speed point-to-point datalinks for use in and between multiprocessor systems", in *Proc. International Conference on Electronics and Information Technology*, Aug 1994, pp. 228-231. Conference held in Beijing, China, August 2-5.
- [3] M. D. May, P. W. Thompson, and P. H. Welch, *Networks, Routers and Transports: Function, Performance, and Applications*. INMOS Limited, UK, 1991. ISBN 90-5199-129-0. Available from anonymous ftp to inmos.co.uk on the directory inmos/info/commo/book.
- [4] Bull SA Serial Link Technology, France, *Bullit Data sheet v1.1*, 1991. OMI/HIC and OMI/Macramé project confidential.
- [5] INMOS Limited, "ISIS C104 packet routing switch preliminary data", Tech. Rep., INMOS Limited, UK, 1991. Available by anonymous ftp to inmos.co.uk in the directory inmos/info/commo/datanhosts on the file C104-0, pg 2.
- [6] J. van Leeuwen and B. B. Tan, "Interval routing", *The Computer Journal*, vol. 30, no. 4, pp. 296-307, 1987.
- [7] Lionel M. Ni and Philip K. McKinley, "A survey of wormhole-routing techniques in direct networks", *Computer*, pp. 62-76, Feb 1991.
- [8] L. G. Valiant, "A scheme for fast parallel communication", *SIAM Journal on computation*, vol. 11, no. 2, pp. 350-361, May 1982.



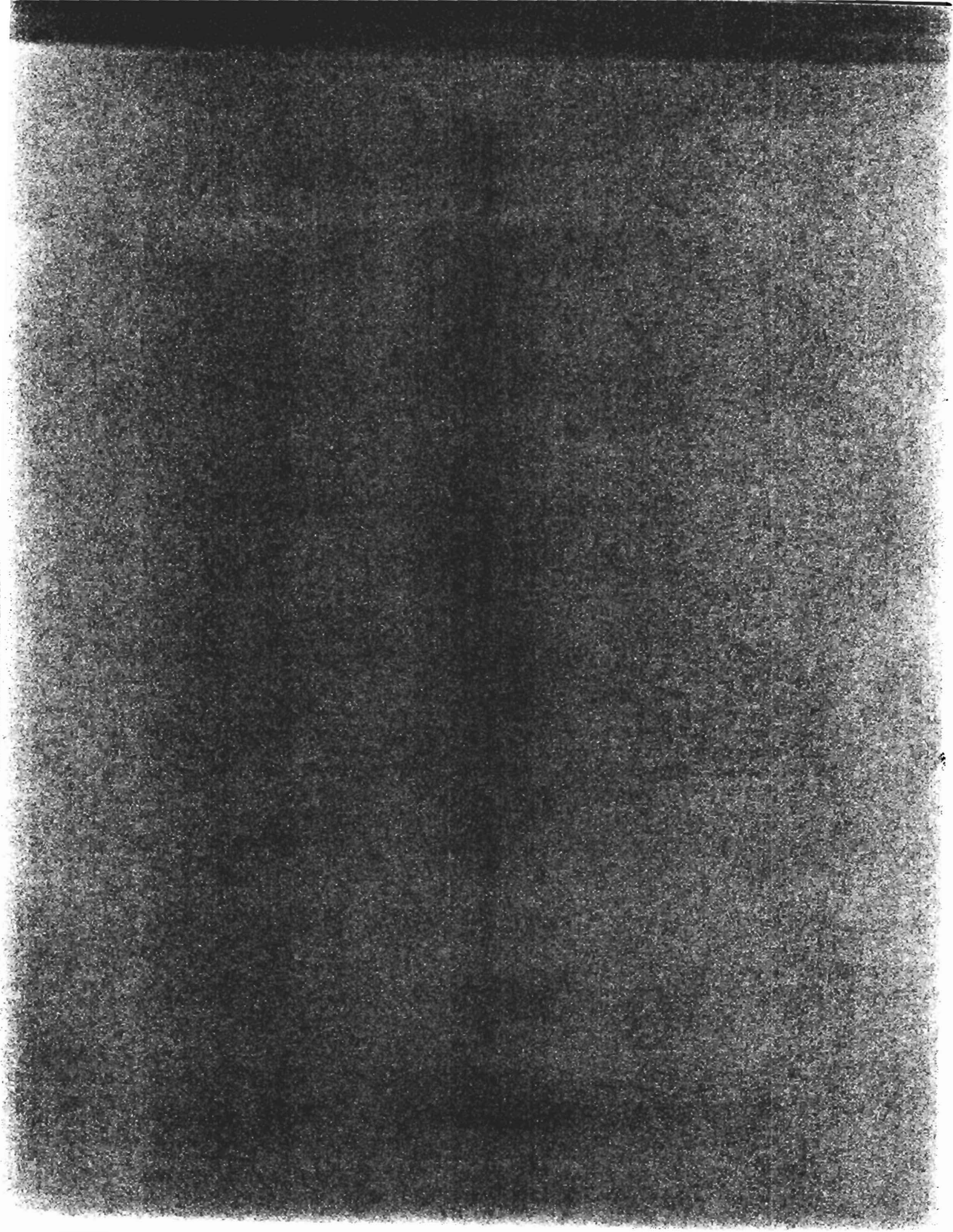


## **S6-1**

### **"Requirements & Goals of Simulation"**

**(Steve Tether - MIT)**

A DAQ simulation need only reproduce the sizes and movements of the data "atoms" in each component. Examples are front-end memories (event fragments, trigger intervals, windows), switching fabrics (cells, internal queue sizes, traffic shaping, cell loss rates), processing flows (events, event-building times, execution times), and transmission links (any size data, speed). This kind of model can be used to calculate total throughput, required memory sizes, and the number of events lost due to congestion. It can't debug DAQ components or simulate MTBF. In simulating a large system, even this abstractly, algorithms that scale well must be used.



## DAQ Simulation Requirements & Goals – 1

---

### Notes of a simulation watcher

A neophyte “simmer” shares what he's learned from

- Talking to experienced “simmers”
- Examining code
- Reading books and articles
- Net surfing

## DAQ Simulation Requirements & Goals – 2

---

### What's the simulation supposed to tell you?

Languages and software are available to handle everything from nuclei to earthquakes. What you leave out is as important as what you put in.

At least one thing is clear: use discrete simulation techniques. Most of a DAQ system is digital and isn't easily described by differential equations.

Some important questions at the level of a complete DAQ system are about bulk data flow unrelated to physics as such:

- Total throughput
- Load balance
- Lost or misrouted data

Others involve some actual inspection of the data

- Rejection factors at each level
- Finding regions of interest

Typical inputs to a global DAQ simulation

At this level the tendency is to avoid consideration of physics; that's done in separate programs which generate input for this one.

Typical inputs:

- Event-size distribution
- Time distribution of Level 1 triggers
- Transmission rates
- "Black box" behavior of subsystems
- Scale (numbers of subsystems)
- Topology
- Decision-time distributions
- Resource-management algorithms

Usually left out:

- Power consumption and heat dissipation
- Reliability
- Scale (physical sizes, cable delays)
- Data transmission protocols
- Voltage levels, signal rise/fall times

Data atoms

It helps to have some general guide to tell you what to keep in the simulation and what to omit.

One way: look for "atoms" of data. That is, pieces that change state together in response to some stimulus.

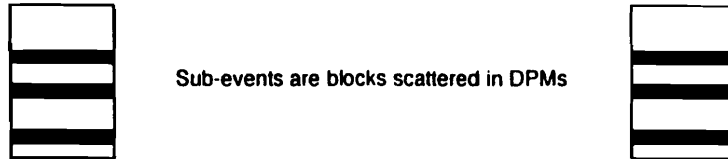
Example 1. A dual-port memory may keep a sub-event in several blocks, but all blocks are read out or freed by a single command.

Example 2. Data sent through an ATM switching fabric is composed of many cells, each of which moves more or less independently. ATM guarantees cell indivisibility.

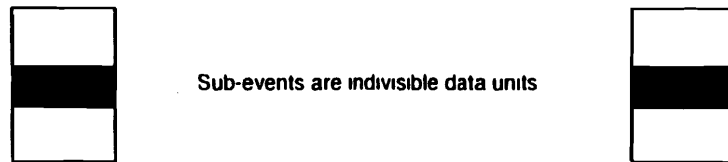
Example 3. A collection of interchangeable data items all in one place may be represented as an item count + tag.

Atoms are created and destroyed at subsystem boundaries.

Progressive abstraction: an event in multiple DPMs



Sub-events are blocks scattered in DPMs



Sub-events are indivisible data units



Sub-events are byte counts + tags



Counts+tags held in a single object

Monitoring

A simulation with no output isn't any use.

There are always limits on resources:

- Computing power
- Memory
- Bandwidth

In some places the limits are generous. in others strict.

Monitor usage at suitable intervals. Histograms, etc.

Consider an "ideal" mode with unlimited resources everywhere; good for early trials when limits may not be well known.

Keep subtotals; processor 83 should not be handling 70% of Level 3 decisions.

Apply conservation laws where possible.

Documentation

Have some.

Pace of development can be furious; hard to keep documentation up to date.

Systems such as CWEB let you generate both compilable code and T<sub>E</sub>X documents from the same set of source files.

FTP CWEB from labrea.stanford.edu, /pub/cweb/, for C/C++.

For language-independent WEB packages, FTP from src.doc.ic.ac.uk, /0-Most-Packages/TeX/uk-tex/web/

- funnelweb/
- noweb/
- spiderweb/

Time

How you handle time is crucial.

Distinguish between sim-time and CPU/real time. The two are only loosely related.

Two general methods for advancing sim-time:

- Time-driven
- Event-driven

Time-driven simulations add a fixed increment to the clock, then check if anything needs to be done. Can be inefficient.

Event-driven simulations keep track of when events are due; they set the clock forward to the time of the next one.

Event-driven requires non-trivial data structure. Luckily, good freeware implementations are available.

Keep sim-time clock precision as low as possible. Use one of the computer's built-in data types if you can.

Scaling

DAQ systems are getting bigger. That slows simulation in two ways:

- More happening at any given sim-time
- More internal bookkeeping (longer queues, etc.)

Use algorithms and data structures that scale well. Avoid searching linear lists, for example.

Try to find places where many physical sub-systems can be represented by a single entity in the program.

Use profiling tools to see where CPU time is spent.

There are many types of code tweaking not performed by compilers.

Last resorts:

- Break the simulation into smaller pieces
- Extrapolate

More scaling

The simulation may use many pseudo-random number generators that should be independent. Their states must be easily controllable for both reproducibility and statistical validity.

Generators based on Knuth's Algorithm M or similar to CERNLIB's RANECU are probably best.

Generators using only simple linear sequences may be inadequate (depends on how they're used).

Graphical simulation packages may have trouble scaling up. Duplication of sub-drawings isn't enough because interconnections have to be made.

Parallel processing

Workstations nowadays have expansion slots for more processors. It's tempting to try a multithreaded simulation.

However ...

- Decomposition is tricky (inter-task communication)
- Programming support is anything but standard
- It may interfere with standard packages such as X

For long runs you can use a multiprocessor in the same way you use a set of independent workstations: run multiple copies of the simulation.

You may want to keep a parallel programming package on hand if it offers superior debugging tools that can be used on single-threaded programs.

Ease of use and portability

Of course it should be easy to use!

Standard X tool kits such as XView<sup>Tc/Tk</sup> and Motif can help a lot.

Try not to make the control panel look like something from NASA.

The GNU C/C++ compiler seems to be everywhere.

There are well-known free packages available via anonymous FTP or (some) on CD-ROM.

Generic class libraries:

- LEDA
- GNU libg++
- NIHCL

Simulation libraries:

- SimPack
- Awesime



Where to look for more

**Books**

Bentley, Jon

"Writing Efficient Programs"

Prentice-Hall, 1982

ISBN 0-13-970244-X

Bentley, Jon

"Programming Pearls"

Addison-Wesley, 1986

ISBN 0-201-10331-1

Bentley, Jon

"More Programming Pearls"

Addison-Wesley, 1988

ISBN 0-201-11889-0

Fishwick, Paul

"Simulation Model Design and Execution: Building Digital Worlds"

Prentice-Hall, Nov. 1994?

Not yet printed, but the first chapter is online via WWW: <http://www.cis.ufl.edu/~fishwick>

Knuth, Donald

"The Art of Computer Programming"

Vols. 1,2,3

Addison-Wesley, 1975

ISBN 0-201-03809-9, 0-201-03802-1, 0-201-03803-X

Knuth, Donald and Levy, Silvio

"The CWEB System of Structured Documentation"

Addison-Wesley, 1994

ISBN 0-201-57569-8

Koenig, Andrew

"C Traps and Pitfalls"

Addison-Wesley, 1989

ISBN 0-201-17928

Perry, Douglas

"VHDL", 2nd ed.

McGraw-Hill, 1993

ISBN 0-07-049434-7

Prycker, Martin de

"Asynchronous Transfer Mode", 2nd ed.

Ellis Horwood, 1993

ISBN 0-13-178542-7

Sedgewick, Robert

"Algorithms"

Addison-Wesley, 1984

ISBN 0-201-06672-6

Wirth, Niklaus and Reiser, Martin

"Programming in Oberon"

Addison-Wesley, 1992 (ACM Press)

ISBN 0-201-56543-9

**Articles**

Buhr, Peter and Strooboscher, Richard

" $\mu$ C++ Annotated Reference Manual"

1993

See under "Free Software"

Christiansen et al.

"NEBULAS - A High Performance Data-Driven Event-Building Architecture

based on an Asynchronous Self-Routing Packet-Switching Network"

CERN/DRDC/92-14

CERN/DRDC/92-47

CERN/DRDC/93-55

Mandjavidze, I.

"Modeling and Performance Evaluation Activities for Event-Builders

based on ATM Switches"

CERN/RD31/TN/94-??

## Where to look for more

### Books

Bentley, Jon

"Writing Efficient Programs"

Prentice-Hall, 1982

ISBN 0-13-970244-X

Bentley, Jon

"Programming Pearls"

Addison-Wesley, 1986

ISBN 0-201-10331-1

Bentley, Jon

"More Programming Pearls"

Addison-Wesley, 1988

ISBN 0-201-11889-0

Fishwick, Paul

"Simulation Model Design and Execution: Building Digital Worlds"

Prentice-Hall, Nov. 1994?

Not yet printed, but the first chapter is online via WWW: <http://www.cis.ufl.edu/~fishwick>

Knuth, Donald

"The Art of Computer Programming"

Vols. 1,2,3

Addison-Wesley, 1975

ISBN 0-201-03809-9, 0-201-03802-1, 0-201-03803-X

Knuth, Donald and Levy, Silvio

"The CWEB System of Structured Documentation"

Addison-Wesley, 1994

ISBN 0-201-57569-8

Koenig, Andrew

"C Traps and Pitfalls"

Addison-Wesley, 1989

ISBN 0-201-17928

Perry, Douglas

"VHDL", 2nd ed.

McGraw-Hill, 1993

ISBN 0-07-049434-7

Prycker, Martin de

"Asynchronous Transfer Mode", 2nd ed.

Ellis Horwood, 1993

ISBN 0-13-178542-7

Sedgewick, Robert

"Algorithms"

Addison-Wesley, 1984

ISBN 0-201-06672-6

Wirth, Niklaus and Reiser, Martin

"Programming in Oberon"

Addison-Wesley, 1992 (ACM Press)

ISBN 0-201-56543-9

### Articles

Buhr, Peter and Strooboscher, Richard

" $\mu$ C++ Annotated Reference Manual"

1993

See under "Free Software"

Christiansen et al.

"NEBULAS - A High Performance Data-Driven Event-Building Architecture based on an Asynchronous Self-Routing Packet-Switching Network"

CERN/DRDC/92-14

CERN/DRDC/92-47

CERN/DRDC/93-55

Mandjavidze, I.

"Modeling and Performance Evaluation Activities for Event-Builders based on ATM Switches"

CERN/RD31/TN/94-??

## DAQ Simulation Requirements & Goals – 15

---

Mandjavidze, I.

"Modeling of the CMS Virtual Level 2"  
CERN/RD31/TN/94-??

Näher, Stefan

"LEDA Manual"

MPI-I-93-109 (Max-Planck-Institut für Informatik)

1993

See under "Free Software"

### Free software

Prime-Time Freeware for UNIX (commercial collections of freeware on CD-ROMs)

May be found in the computer section of bookstores

To contact PTF (Sunnyvale, CA, USA)

by phone: (408) 433-9662

by e-mail: [ptf@cicl.com](mailto:ptf@cicl.com)

For FTP access I give below the name of the machine and the directory

Awesime (class library for process-oriented discrete event simulation)

[gatekeeper.dec.com, /pub/misc/](ftp://gatekeeper.dec.com/pub/misc/)

GNU (Free Software Foundation)

[gcc/g++](ftp://gcc/g++) (C/C++ compiler)

[libg++](ftp://libg++) (general-purpose class library)

[prep.ai.mit.edu, /pub/gnu/](ftp://prep.ai.mit.edu/pub/gnu/)

LEDA (class library emphasizing combinatorial computing)

[ftp.mpi-sb.mpg.de, /pub/LEDA/](ftp://ftp.mpi-sb.mpg.de/pub/LEDA/)

$\mu$ C++ (front-end for adding parallel computing to C++)

[plg.uwaterloo.ca, /pub/uSystem/](ftp://plg.uwaterloo.ca/pub/uSystem/)

NIHCL (general-purpose class library, said not to work too well with g++)

[atw.nih.gov, /pub/](ftp://atw.nih.gov/pub/)

## DAQ Simulation Requirements & Goals – 16

---

Oberon (compiler and development system)

[neptune.ethz.ch, /pub/Oberon/](ftp://neptune.ethz.ch/pub/Oberon/)

SimPack (C library of simulation routines)

[ftp.cis.ufl.edu, /pub/simdigest/tools/](ftp://ftp.cis.ufl.edu/pub/simdigest/tools/)

### Newsgroups

Each of these newsgroups has a list of Frequently Asked Questions (and answers) stored on [rtfm.mit.edu](http://rtfm.mit.edu). To look at `comp.lang.c`, for example, go to the directory `/pub/usenet-by-hierarchy/comp/lang/c`

`comp.dcom.cell-relay` (ATM)

`comp.dcom.frame-relay`

`comp.lang.c`

`comp.lang.c++`

`comp.lang.fortran`

`comp.lang.misc`

`comp.lang.oberon`

`comp.lang.vhdl`

`comp.parallel`

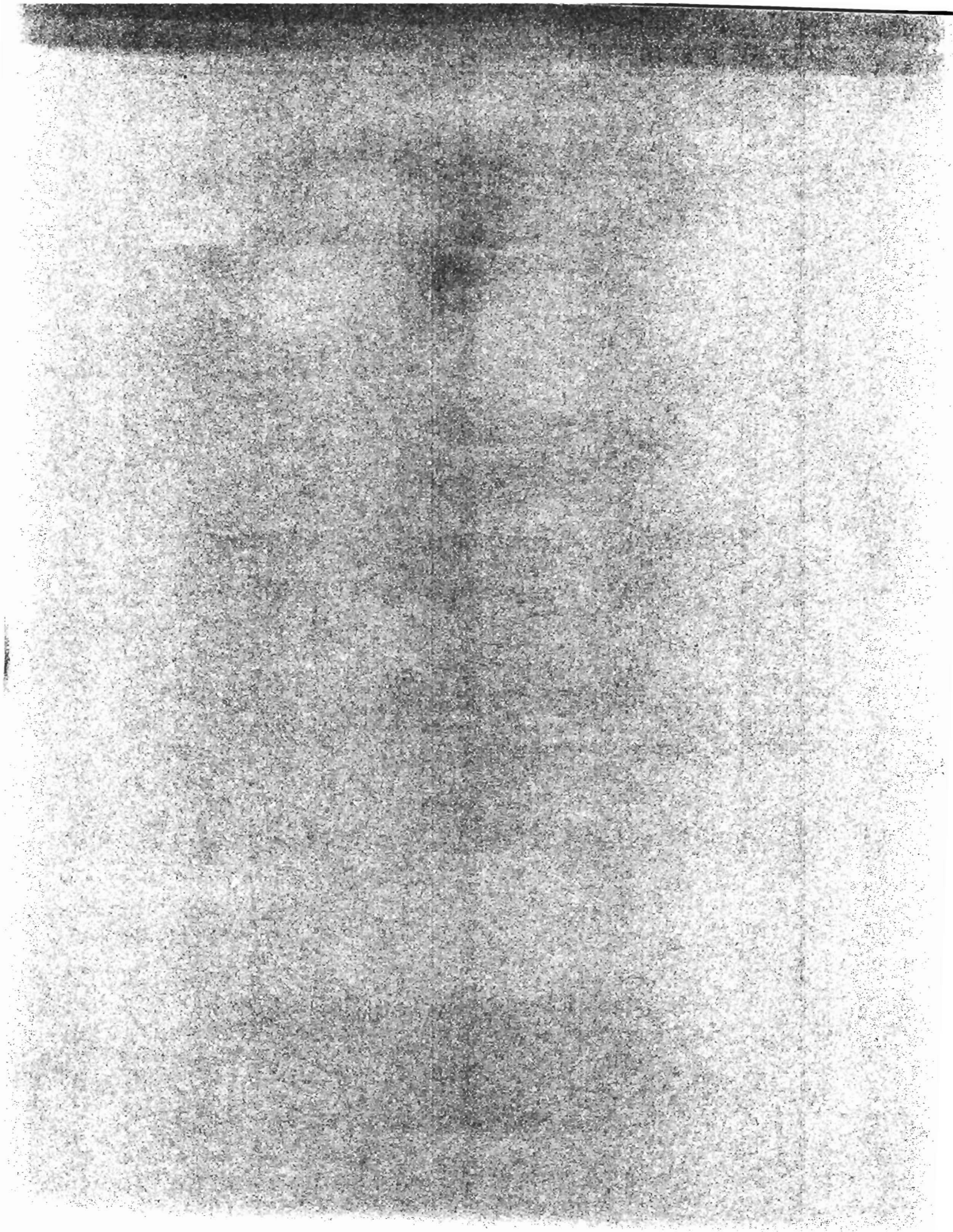
`comp.simulation`

**S6-2**

**"Behavioral Simulation and High Level Modelling"**

**(Mike Haney - University of Illinois)**

Commercial and academic languages and tools for electronic system design automation. Behavioral modeling, multilevel modeling, simulation, validation, verification, and synthesis issues. Examples of the use of these tools in industry. Emphasis on what these tools deliver, in contrast to the objectives of the user.



## Behavioral Simulation and High Level Modeling

Mike Haney  
High Energy Physics, University of Illinois  
Urbana, IL 61801  
m-haney@uiuc.edu

### Abstract

Commercial and academic languages and tools  
for electronic system design automation.

Behavioral modeling, multilevel modeling, simulation,  
validation, verification, and synthesis issues.

Examples of the use of these tools in industry.

Emphasis on what these tools deliver,  
in contrast to the objectives of the user.

### What?

Behavioral modeling focusses on what, not how;  
emphasis on function rather than implementation

Many levels of abstraction  
(see fig)

Behavioral modeling is relative:  
systems vs algorithms  
algorithms vs RTL  
RTL vs logic

Every model is "behavioral"  
with respect to some lower level  
(including reality!)

Mixed level modeling:  
Combining a "high level" model of this  
with a "low(er) level" model of that

Best of both (all) worlds:  
Faster than complete low level simulation  
Focusses on what matters

E.g. module on a "virtual testbench"

What (continued)

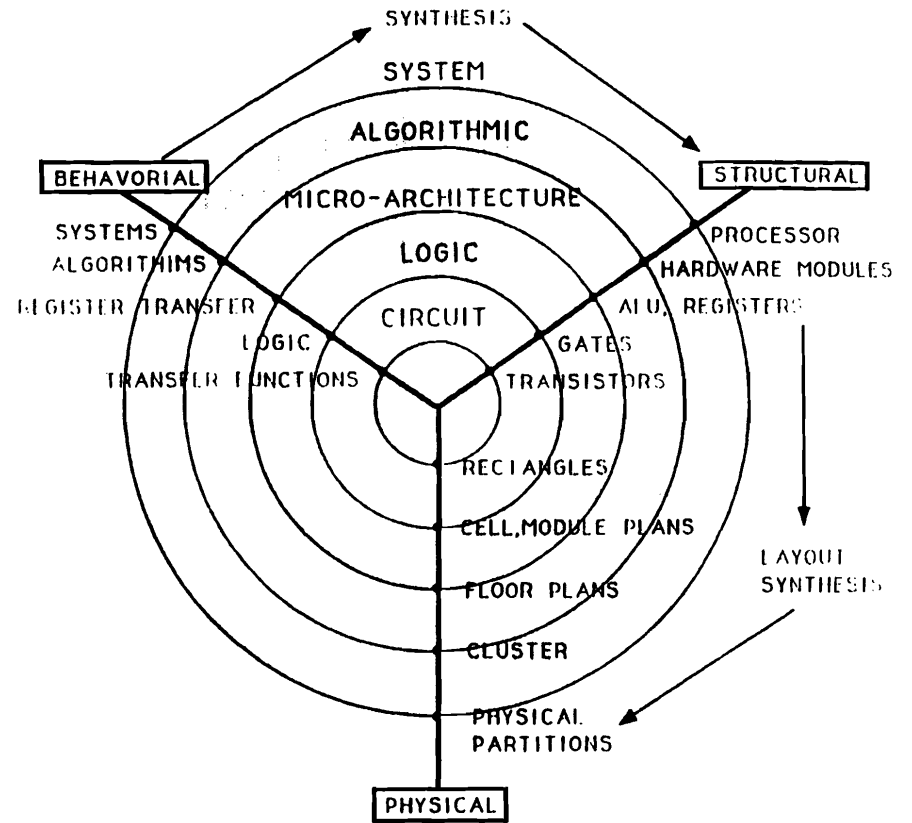
**Simulation**

continuous (analog)  
 linear algebraic methods  
 numerical integration  
 "accurate" but expensive

synchronous (cycle based)  
 levelized compiled code can be fast (10x)  
 timing must be analyzed later

discrete event  
 better when cause and effect  
 have a "large" delay  
 timing can be implicit  
 (no need for unit delay)

discrete task  
 groups of events  
 with static temporal relationships  
 local causality horizon  
 well-defined boundaries



## Why?

to better understand the problem

Motorola used trace-driven simulation to model/analyze PowerPC performance, to let the compiler optimization group study the code that their product produced

to codify the specification

the SCI "spec" is available as an executable C program

to "explore the design space"

Motorola used trace-driven simulation to model/analyze PowerPC performance, to examine bus/cache trade-offs

to prepare for "the real thing" - testing strategies

Motorola created a family of PowerPC simulators (C++):  
architecture (testing application programs)

timing (learned that "error handling" depended on the user)

function (timing model adapted to Verilog)

IBM Haifa Research Lab; PCI Checker.  
Takes simulation traces; looks for protocol violations. (yaron@vnet.ibm.com)

## Why (continued)

to get better leverage on the design process

Fiat Central Research reduced design cycle time by 30% by mixing schematics with VHDL.  
(did not want to "lose" time learning VHDL up front)

SGI started with 78 PALs. Added 3 times more features to the spec.  
=> 18 FPGA's (13 new) by 6 engineers in 8 months (Verilog + Synopsis)



## Examples from High Energy Physics

- Schurecht, et al., IEEE NSS, Nov. '91  
CDF data acquisition system, in Verilog
- Booth, et al., IEEE NSS, Nov. '91  
general barrel-shift event builder,  
in Verilog + DataViews + Nexpert
- Streets, et al., FERMILAB-Conf-92/43  
Experience with Modsim II
- Bogaerts, et al., IEEE Trans. Nuc. Sci., April '92  
SCI-based data acquisition architecture for LHC,  
in Modsim II
- Shu, et al., IEEE Trans. Nuc. Sci., April '92  
flexible modeling software for pulsed data  
acquisition, in FORTRAN
- Hughes, et al., IEEE Trans. Nuc. Sci., April '92  
SDC data collection chip, and trees of chips,  
in VHDL
- Milner, et al., IEEE Trans. Nuc. Sci., April '92  
SDC data acquisition, in Modsim II
- Angstadt, et al., IEEE Trans. Nuc. Sci., Aug. '92  
DZero data acquisition, in RESQ
- Dean & Hancy, IEEE Trans. Nuc. Sci., Aug. '92  
FASTBUS virtual environment, in VHDL.

## Examples from High Energy Physics (continued)

- Kristiansen, et al., IEEE Trans. Nuc. Sci., Feb., '94  
SCI architectures, in C++
- Wang, et al., IEEE Trans. Nuc. Sci., Feb. '94  
SDC data acquisition, in Modsim II + DataViews
- Letheren, et al., IEEE Trans. Nuc. Sci., Feb. '94  
ATM event builder, in  $\mu$ C++

## How?

### VHDL

IEEE standard 1076-1987, 1076-93(?)  
standard, but few libraries

more expressive, more flexible,  
tighter (against programming error)

countless vendors, products

The VHDL Handbook, Coelho,  
Kluwer Academic Press, 1989

VI - VHDL International

### Verilog

IEEE 1364 committee  
libraries, but not standard (yet)

shorter, faster, more models available,  
easier to learn

Cadence Design Systems,  
and many (secondary) vendors

The Verilog Hardware Description Language,  
Thomas & Moorby,  
Kluwer Academic Press, 1992

OVI - Open Verilog International

### Others...

```
entity find is
  port (x : in bit_vector(3 downto 0);
        index : out bit_vector(3 downto 0));
end find;

architecture find of find is
  type int_array is array (0 to 7) of bit_vector(3 downto 0);
  signal list : int_array := ("1000", "0111", "0110", "0101",
                             "0100", "0001", "0010", "0011");
begin
  sort : process
    variable i, j, inter : integer := 0;
    variable low, high, mid, found : integer := 0;
    variable temp : bit_vector(3 downto 0);
    variable sorted : bit := 0;
  begin
    if (sorted = '0') then
      i := 0;
      while (i < 8) loop
        j := i + 1;
        while (j < 8) loop
          if bits_to_int(list(j)) < bits_to_int(list(i)) then
            temp := list(j); list(j) <= list(i); list(i) <= temp;
            wait for 0 ns;
          end if;
          j := j + 1;
        end loop;
        i := i + 1;
      end loop;
      sorted := '1';
    end if;

    index <= "1111";
    low := 0;
    high := 8;

    found := 0;
    while ((low < high) and (found = 0)) loop
      mid := (low + high) / 2;
      if (x = list(mid)) then found := 1; end if;
      if (x > list(mid)) then low := mid + 1; end if;
      if (x < list(mid)) then high := mid; end if;
    end loop;

    if (found = 1) then index <= int_to_bits(mid); end if;
    wait on x;
  end process;
end find;
```

Figure 2. Behavioral specification of the Find in VHDL.

VHDL (asic & eda, July '94)

Leapfrog - Cadence Design Systems, San Jose, CA  
(408)-943-1234

Voyager - IKOS Systems, Cupertino, CA  
(408)-255-4567

AdvanSIM 1076 - Intergraph Electronics, Huntsville, AL  
(800)-VERIBEST

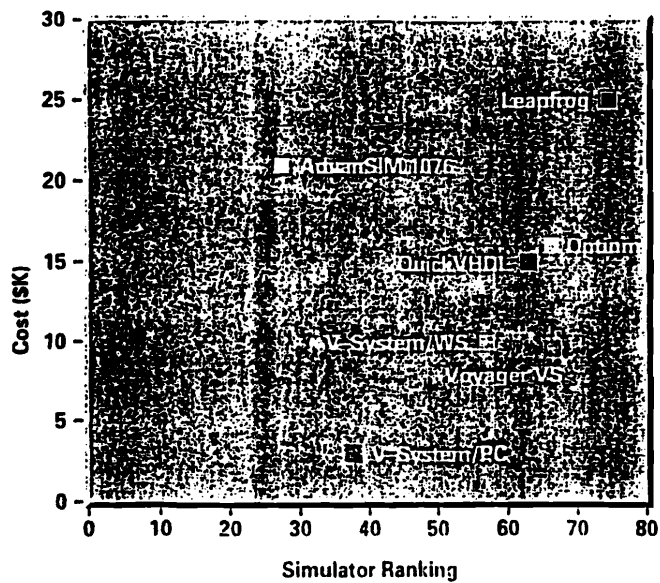
QuickVHDL - Mentor Graphics, Wilsonville, OR  
(508)-685-7000

V-System - Model Technology, Beaverton, OR  
(503)-690-6838

Optimum - Vantage Analysis Systems, Fremont, CA  
(510)-970-1600

Cadence Design Systems, San Jose, CA  
(408)-943-1234

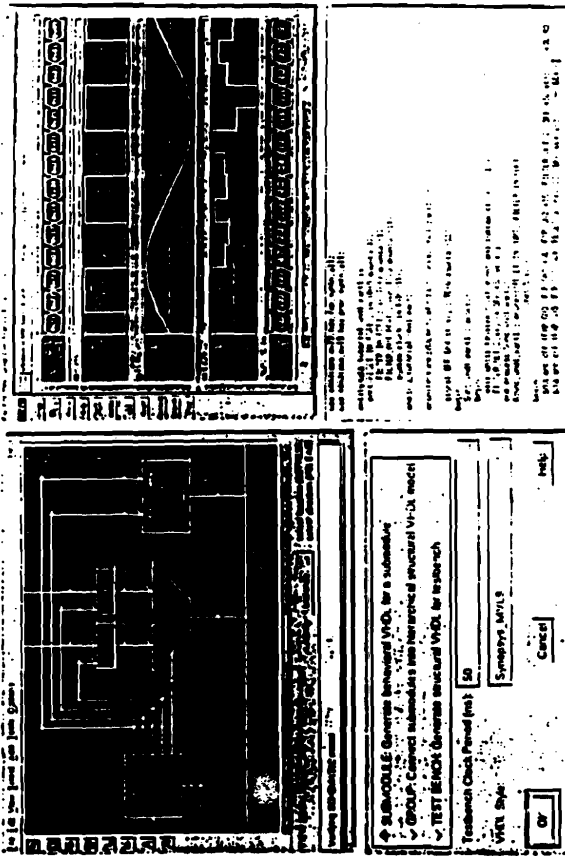
**Cost vs. utility**



Source: Seva Technologies, Inc.

Figure 7. The most cost-effective simulators appear in the lower right-hand corner of this chart.

**HIGH-PERFORMANCE VHDL SIMULATION**



**Typical Windows in System Design Architect**

As shown above, sophisticated digital architectures can be easily entered and built. SDA's intuitive graphical system SPA then generates VHDL automatically. System Design Architect provides multiple design views, comprehensive high level analysis, and support for virtual control panels and virtual instruments.

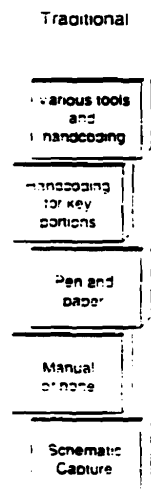
# Hardware Design System™ (HDS™)

Offers block diagram entry, fixed point simulation and analysis, architectural level design, VHDL output, and interface to synthesis tools

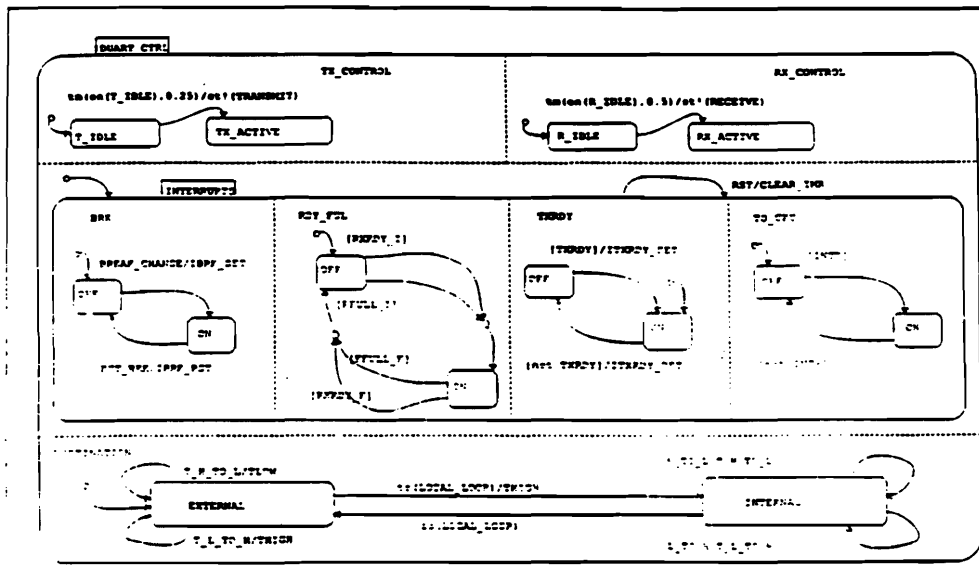
12/1

**System Level Design:  
HDS vs. Traditional Method**

Design Level	Activity	Methodology
System Level	DSP Algorithm Development	SPW
System Level	Fixed Point Analysis and Simulation	HDS Analyzer
System Level	Architectural Design and Simulation	HDS Architect
System Level	VHDL Code	HDS VHDL Link
System Level	Logic Design	Synthesis Tool



Modern hardware design consists of many levels and activities. As this figure shows, the traditional method involves a series of independent steps, some of which can take significant time if they are performed at all. But HDS offers a methodical, integrated way to take a design from a high-level diagram through synthesis—including automatic VHDL generation. This makes HDS as valuable to designers of general hardware systems as it is to DSP designers exploring the effects of limited precision on complex DSP algorithms. Highly sophisticated yet easy to use, HDS frees hardware designers from the tedium of hand coding and the complexities of schematic capture systems.



**TOP-LEVEL STATECHART** specifying the behavior of the Dual Asynchronous Receiver/Transmitter (DUART). This chip is used primarily for digital transmission and reception of bytes over serial lines.

Statecharts extend state transition diagrams with hierarchy, concurrency, and broadcasting. States and transitions are described with boxes and arrows. Dotted lines indicate concurrent states. The syntax for transition labels is Event(Condition)/Action. Actions can be used to broadcast information to other parts of the model.

**Generate VHDL directly from the model:**

**STRUCTURAL**

```

architecture structure of MAIN is
  component DUART
    port
      ( qdir_LINE_STATUS      in   integer;
        qdirM1_6              in   integer;
        qdirSR_1              in   integer;
    );
  begin
    MSCP1: DUART
      port map ( qdir_LINE_STATUS, qdirM1_6, qdirSR_1,
    );
  end
end structure;

```

**BEHAVIORAL**

```

procedure exec_st_HOLDING_REG is
begin
  case st_HOLDING_REG isin is
    when st_LOADED =>
      if rdirr_event then
        inst_LOADED << false;
        assign to dirSR_2(1);
        st_HOLDING_REG isin << st_EMPTY;
      end if;
    when st_EMPTY =>
      if oewRITE_SUP'event and dirCP_2 = 1 then
        assign to dirSR_2(0);
        assign to dirTER(dirDS_BUFFER);
        inst_LOADED << true;
        st_HOLDING_REG isin << st_LOADED;
      end if;
    end case;
  end exec_st_HOLDING_REG;

procedure exec_st_ENABLED is
begin
  exec_st_HOLDING_REG;
  exec_st_TX_SHIFT_REG;
end exec_st_ENABLED;

```

Portions of the DUART VHDL.

The generated behavioral VHDL entities are interconnected via structural VHDL constructs.

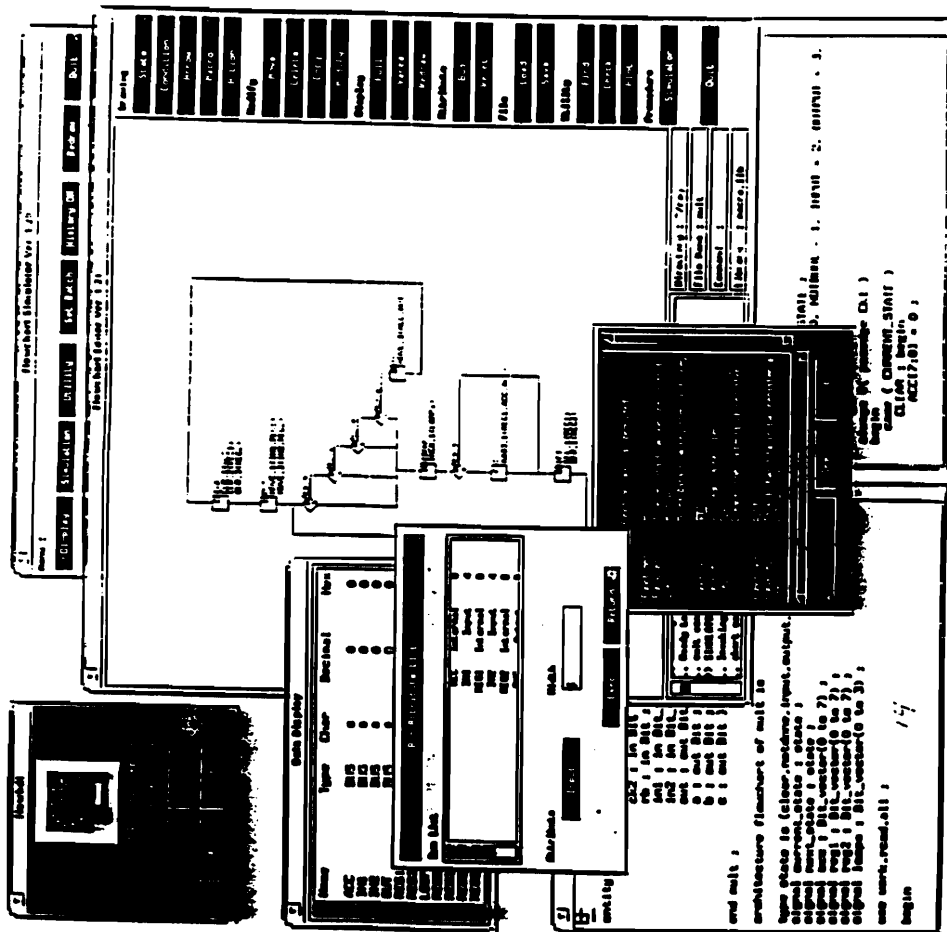
Knowledge Based Silicon Corp., Columbia, SC  
(803)-779-2504

## flowHDL™

Because a picture is worth a thousand lines of HDL code.

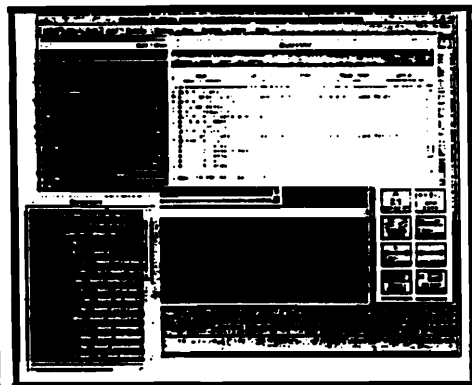
Introducing the flowHDL Training and Evaluation Kit.

An entry-level toolset for the ASIC designer who is looking for help in making the transition to top-down HDL-based design. Available at a modest price for the Sun SPARC workstation.



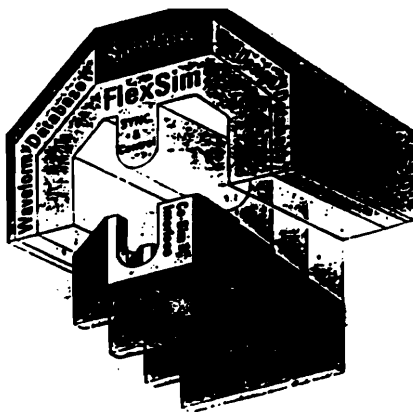
## FlexSim

Expand your simulation environment with the FlexSim™ simulation backplane. FlexSim is the foundation technology of the new Continuum™ mixed-signal simulator and QuickVHDL ProSystem, mixed QuickSim II™ and QuickVHDL co-simulation. FlexSim enables two or more simulation kernels to analyze a design while presenting you with a common design and debug environment. FlexSim partitions the design upon invocation and hand-sets it to the proper simulation kernel. SimView™ acts as the common user interface, sending commands to and displaying results across kernels. FlexSim delivers a unique combination of performance and ease of integration. Well documented APIs provide full access to design partitioning, user interface, and compilation functions. For more information, available through our Web site:



## QuickVHDL ProSystem

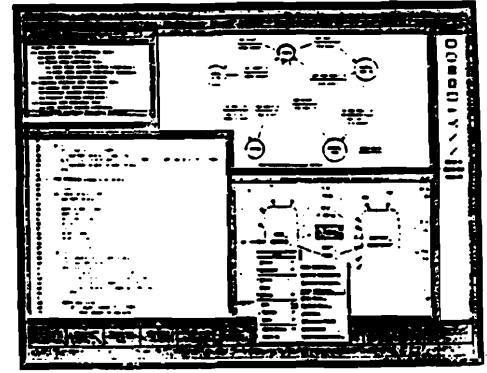
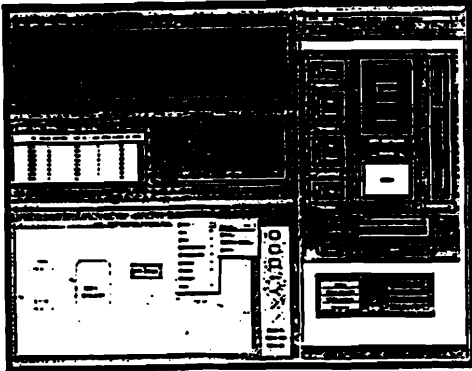
Boost your VHDL simulation performance with QuickVHDL ProSystem. The high-speed simulation toolset includes a Low-Level Controlled Code simulation engine for fast, 100 percent VHDL simulation. With its VHDL Designer Extractor, you can enter, export, and simulate designs created with Design Architect™ or System Architect™. Its System Configuration ensures that AutoCAD™ VHDL can be used and simulated. The System Configuration Co-simulation Interface lets you use QuickSim II™ to simulate VHDL models together. QuickVHDL ProSystem lets you take advantage of existing Mentor Graphics™ AMP libraries within a VHDL design environment. And, QuickVHDL ProSystem allows you to simulate VHDL models within Mentor Graphics' board-level design.



Mentor Graphics, Wilsonville, OR  
(508)-685-7006

## System Architect

Reduce your ASIC and FPGA design cycle times by 30-40 percent while increasing product quality with System Architect. System Architect lets you work with an abstract and understandable view of your design—a view you can't get from tools based on textual design alone. System Architect provides a highly flexible Data Flow Diagram editor to describe the hierarchy and architecture of your design. You can describe Finite State Machines as classical transition diagrams or matrices and infinite state machines to ensure correct behavior. Using automatically generated templates, you can describe algorithmic and datapath behavior in VHDL. Built-in checks ensure design consistency. Automatic VHDL generation, optimized for synthesis, enables rapid response to design and verification changes.



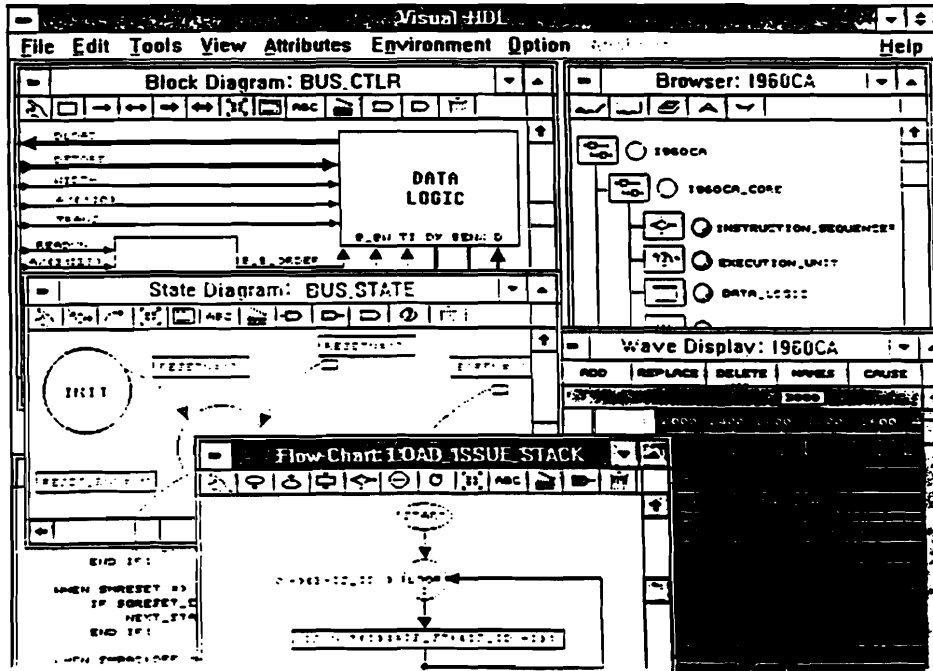
## VHDL Architecture Station

Get VHDL simulation and architecture visualization and simulation in one integrated package with VHDL Architecture Station. The training System Architect and QuickVHDL™ components of the System Architect flow provide consistent capture and debug of VHDL designs at the design level. Using graphical, automated application development, System Architect lets you build a complex system architecture, generate graphical templates, produce EDA data flow diagrams, and generate VHDL code, and automatically generate VHDL. Built-in data flow diagrams and graphical source changes result in automatic regeneration of the VHDL. With VHDL Architecture Station, you can probe and modify VHDL simulations through its graphical analysis and re-entrance capabilities, and simulation through VHDL simulation.

Mentor Graphics, Wilsonville, OR  
(508)-685-7000

# Visual HDL

Shaping the new wave of executable specifications

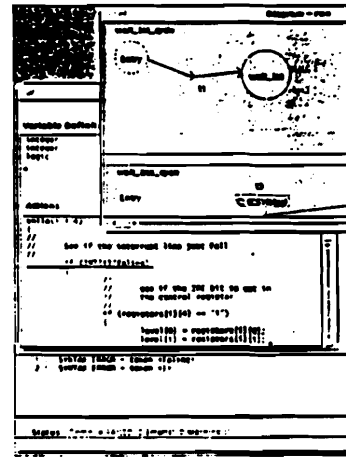
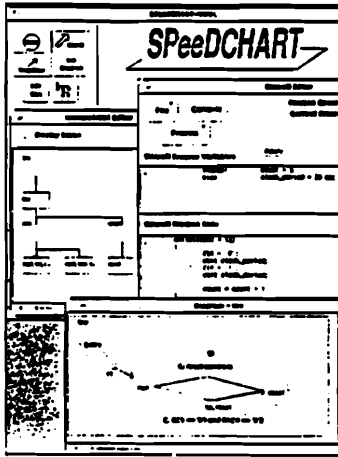


SEF Technologies, Sunnyvale CA  
(408)-737-2880

17a

## Key Points:

- Written in C++
- Based on XView
- Uses Floating License
- Runs on Unix Workstations
- Based on the well known Finite State Machine Theory
- Implements concurrent and hierarchical State Diagrams
- Condition/Action language with powerful Behavioral constructs
- Handles both synchronous and asynchronous modes
- Support of control logic and datapath design



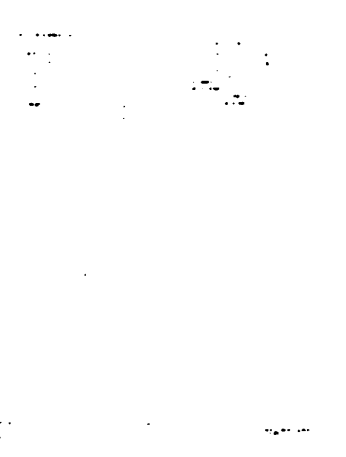
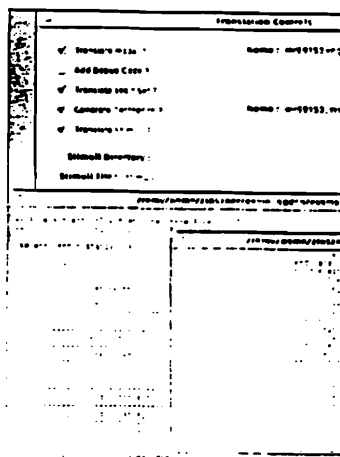
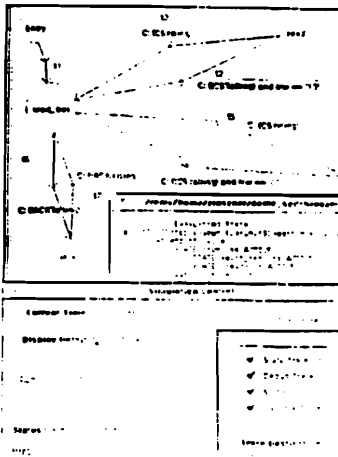
## Editing

- User friendly Graphical editing of the State Machine Diagrams
- Window based Variable and Condition/Action Editor
- Stimuli Editor
- Hierarchy Editor

## Analysing

- Analysis of Design Consistency
- Syntax checking
- Fast interactive error location
- Incremental compilation

## Your New HDL Design Capture Tool for:



## Simulation

- Built-in simulator
- Graphical simulator
- Extensive debugging facilities
- Data-base simulator

## Generating HDL code

- IEEE 1076 VHDL and Verilog HDL code generation for external simulation
- Generation of "Ready to use" test-bench including translated stimuli
- IEEE 1164 or User Definable VHDL Logic Set
- HDL synthesisable code targeted to commercial Synthesis tools

## Documentation

- Direct output to printer or file in the form of printouts
- Selection of size and position of the pages
- Text file output
- User definable printer



The Synopsys Design Flow:

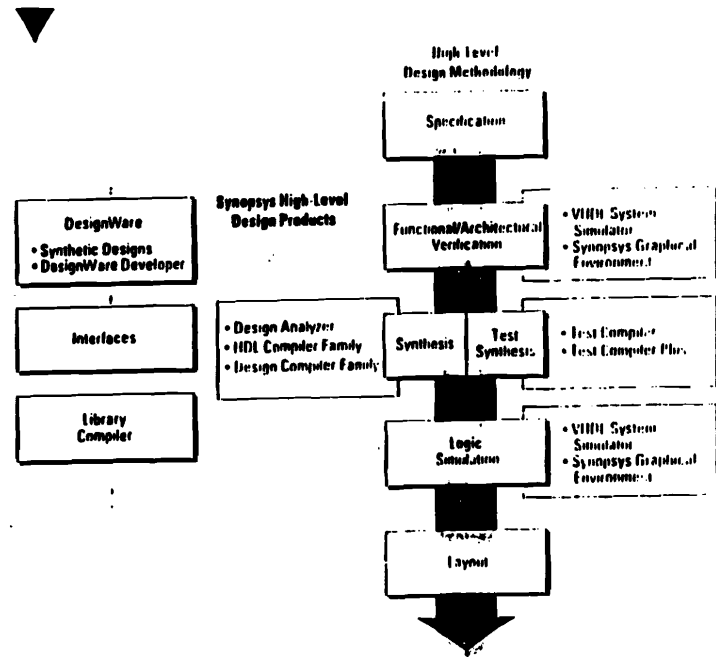
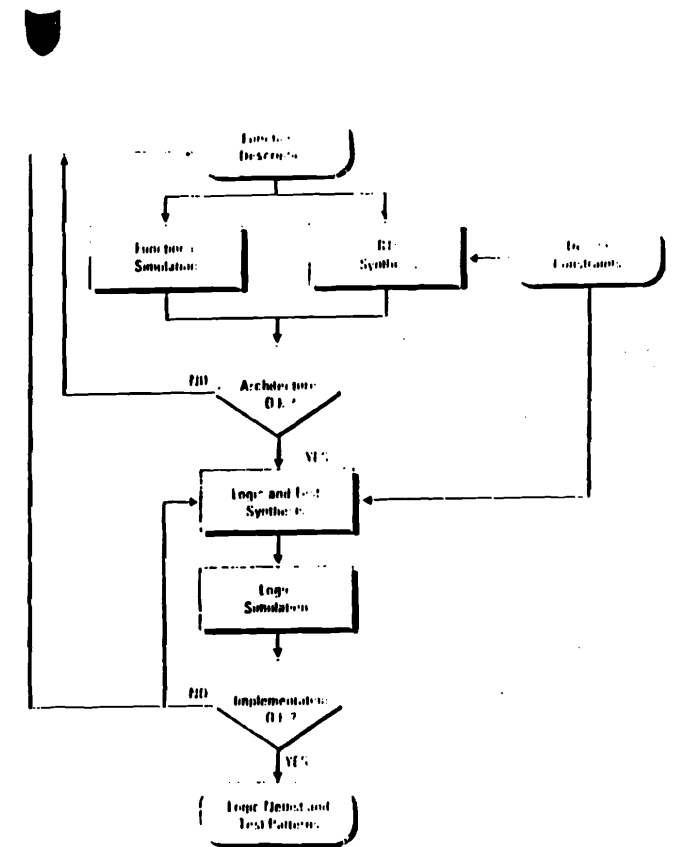
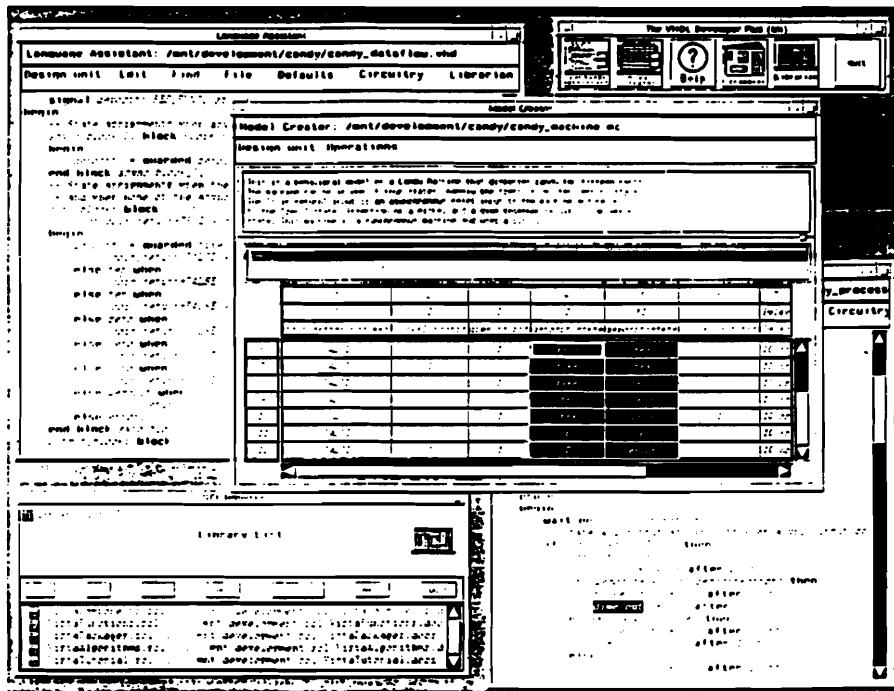


FIGURE 1  
 Synopsys EDA design flow



The VHDL Developer Plus supports quick creation of VHDL behavioral models.

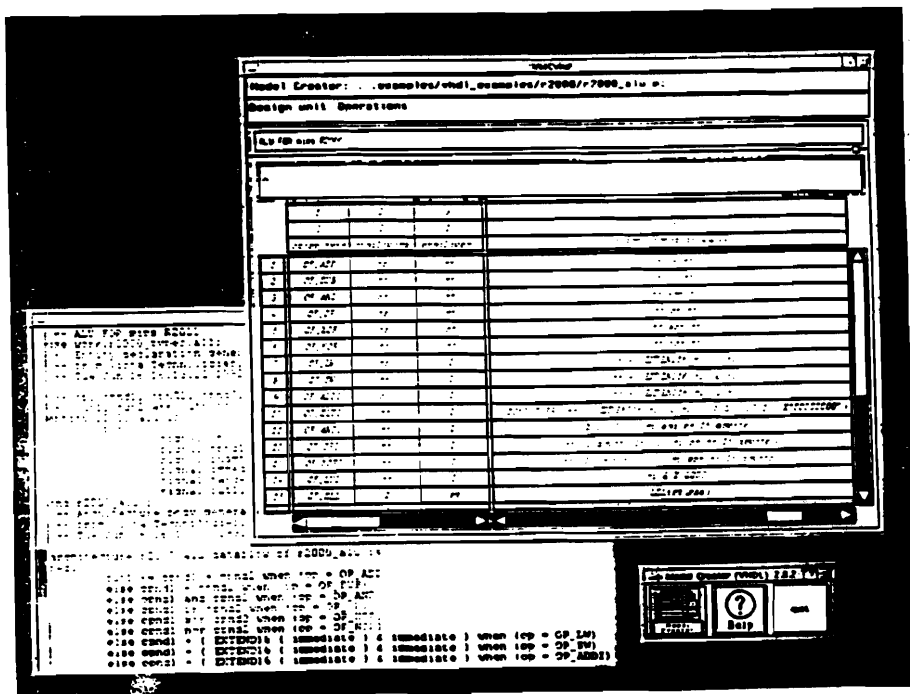
Visa Technologies, Inc. Schaumburg, IL.  
(708)-706-9300. info@vistatech.com



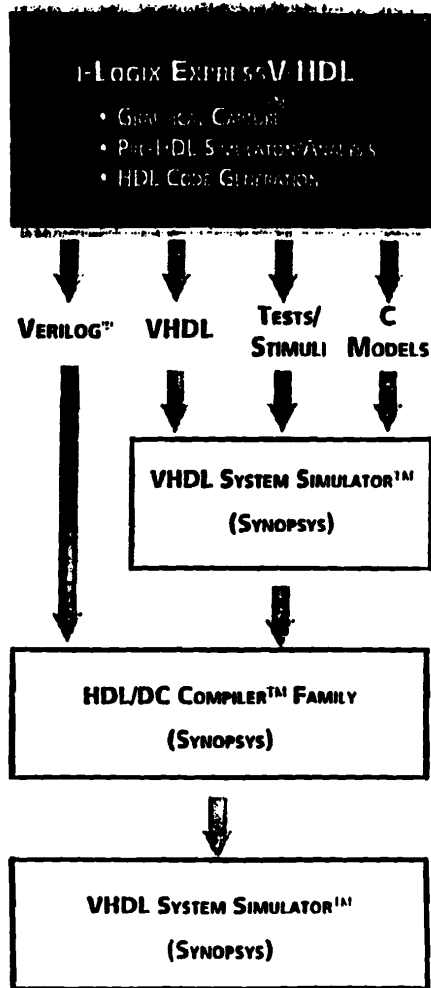
### VISTA MODEL CREATOR™ for VHDL

The Vista Model Creator for VHDL supports generation of combinational function and state-machine models.

Visa Technologies, Inc. Schaumburg, IL.  
(708)-706-9300. info@vistatech.com

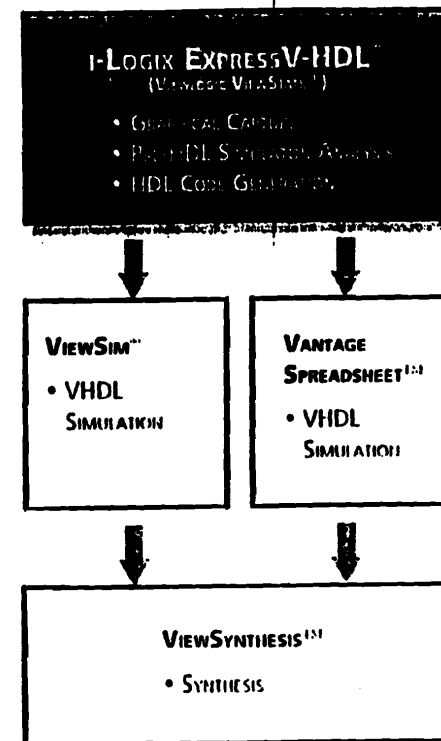


Synopsis, Mountain View, CA  
(415)-962-5000



18c

Viewlogic Systems, Marlboro, MA  
(508)-480-0881 x226

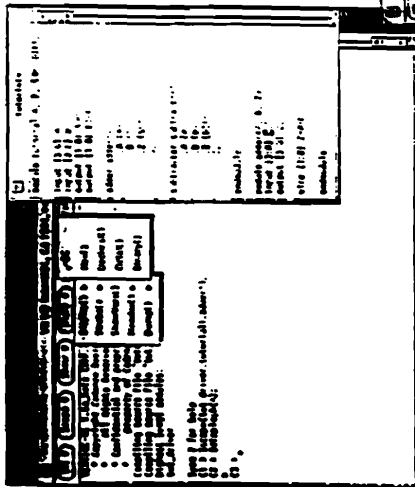


19

interHDL Inc., Sunnyvale CA  
(408)-749-8775

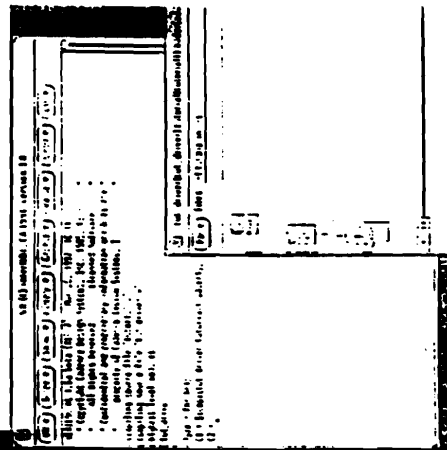
# VLTool

## Tools for Debugging Verilog



Reduces repetitive typing.

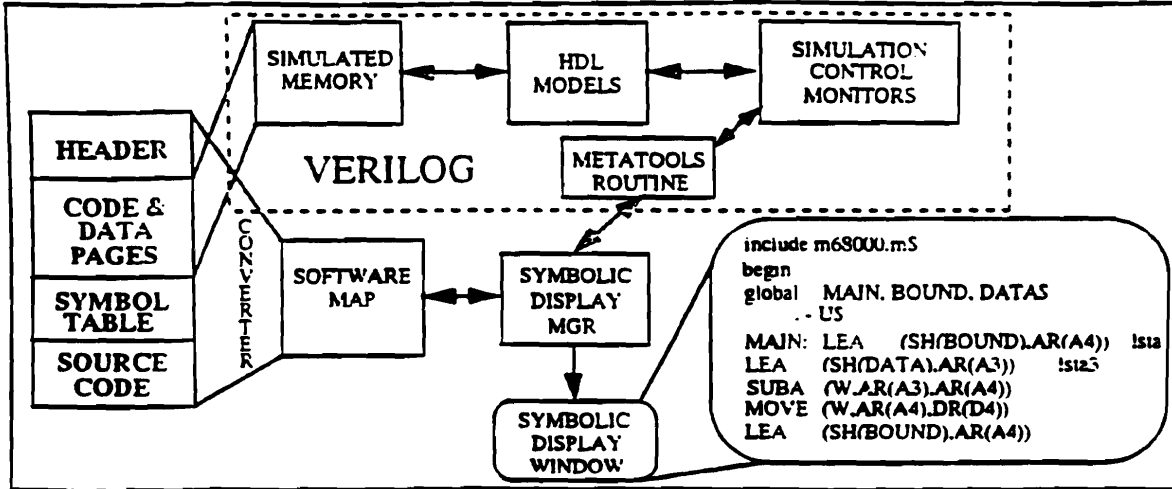
User defined commands.



Fast traversal of design.

Most commands implemented.

### MetaTools / Verilog Interface



PC-Based Verilog (asic & eda, April '94)

VeriBest - Intergraph Electronics, Huntsville, AL  
(800)-VERIBEST

FinSim - Fintronic USA, Menlo Park, CA  
(415)-325-4474

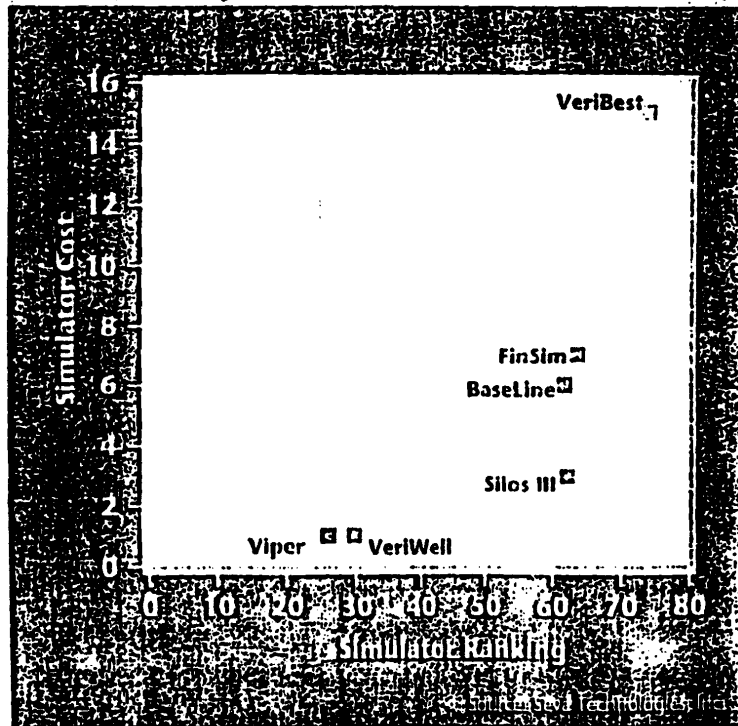
BaselLine - Frontline Design Automation, San Jose, CA  
(408)-456-0222

Silos III - Simucad, Union City, CA  
(510)-487-9700

VeriWell - Wellspring Solutions, Sutton, MA  
(800)-VERIWELL

Viper - interHDL, Inc., Sunnyvale, CA  
(408)-749-8775

### Cost vs. utility



```

module testTR;
reg Nickel, Dime, Select, Return;
wire enable, change;
popS ps (Nickel, Dime, Select, Return, enable, change);
Initial
begin
    (Nickel, Dime, Select, Return) = 0;
    Hit_Return;
    Deposit_Nickel;
    Deposit_Dime;
    Hit_Select;
    Deposit_Dime;
    Deposit_Dime;
    Hit_Select;
    #10 $stop;
end

always @(enable)
    if (enable)
        $display("Selector Enabled"); else
        $display("Selector Disabled");

always @(posedge change)
    $display("Change Received");

task Hit_Return;
begin
    #10 Return = 1;
    $display("Coin Return");
    #10 Return = 0;
end
endtask

task Hit_Select;
begin
    #10 Select = 1;
    $display("Selector Hit");
    #10 Select = 0;
end
endtask

task Deposit_Nickel;
begin
    #10 Nickel = 1;
    $display("Nickel Deposited");
    #10 Nickel = 0;
end
endtask

task Deposit_Dime;
begin
    #10 Dime = 1;
    $display("Dime Deposited");
    #10 Dime = 0;
end
endtask
endmodule
    
```

Veritools, Los Altos, CA  
(415)-941-5050

## VERITOOLS DATA SHEET

# Verilint™ 1.1

## *Verilog HDL Design Purifier*

Verilint is developed by InterHDL

### WHY

Verilint is a productivity enhancement tool for designers who use Verilog HDL. Verilint cuts development time by identifying coding errors which otherwise would be carried unrecognized into simulation and synthesis. Verilint saves time and money by providing a fast and inexpensive way to develop and check a design without invoking a simulator or a synthesizer. Verilint implements

### HOW

Verilint checks Verilog designs for syntax errors, semantic errors, and questionable constructs. At the same time, it performs a thorough rifting of the design and posts warnings regarding coding practices that may lead to problems in simulation and synthesis. These checks are user configurable. A project team may define its coding style by turning appropriate checks

### WHEN

Use Verilint during the design creation phase prior to simulation and synthesis. Verilint is very fast and helps you fix design errors without running a Verilog simulator or a synthesizer. It is a very efficient and economic way for each member of a project to check a design and its creation time. Use Verilint before place & route at design release time to check a netlist for logic design rule violations

Other - UDL/I

(if VHDL = description, then UDL/I = synthesis...)

Fintronic USA, Menlo Park, CA  
(415)-325-4474

## UDL/I® Features

### Synthesis Oriented

- Precise mapping to core subset
- Unique data type representing Integer, Bitstrings, Four-valued logic, and arrays of four-valued logic
- Supports clock and reset
- Syntax for finite-state machine descriptions

### Powerful Modeling Constructs

- Implicit conversion and resolution functions
- Functional (in table format) description of primitives
- Hierarchical structural descriptions
- Generic standard functions (variable number of ports, generic delays, etc.)
- Path delays
- Fanout description

### Potential for fast simulation

- Simple data type
- Rich set of predefined functions
- Support for synchronous assignment

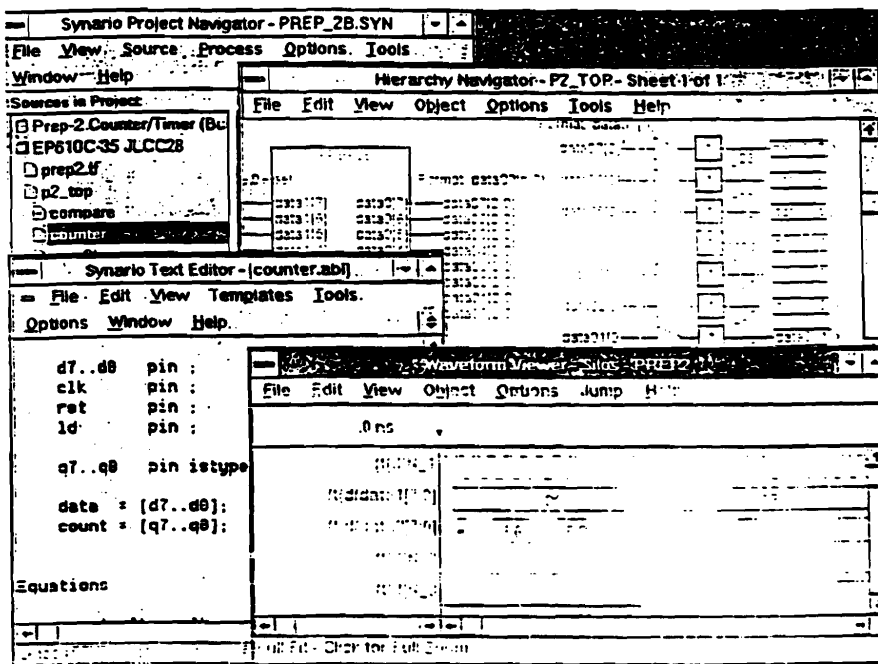


Figure 1. Synario project navigator, hierarchy navigator, text editor, and waveform viewer.

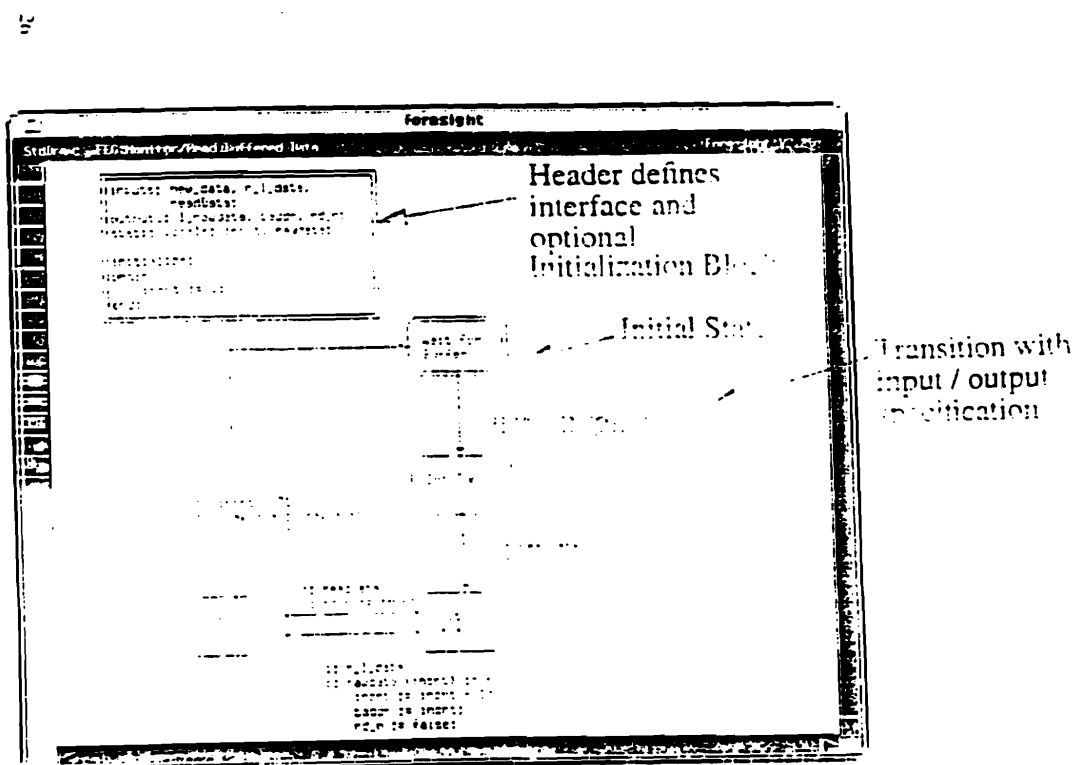


Figure 5. Foresight State Transition Diagram

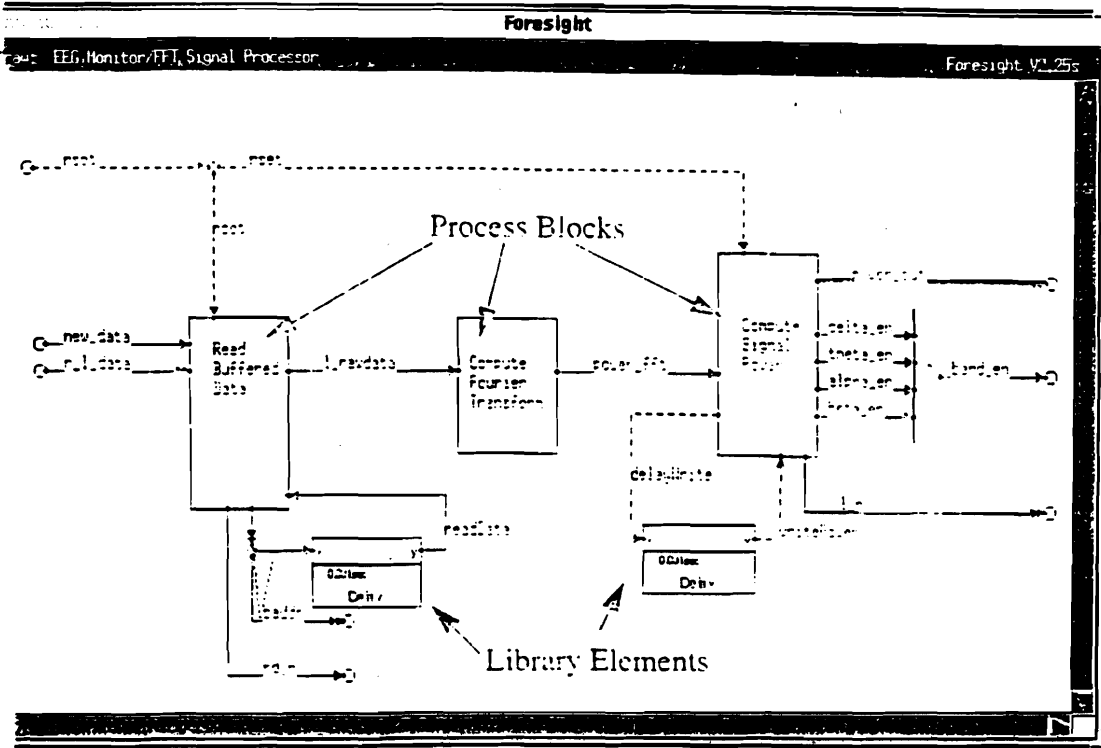


Figure 2. Foresight Data Flow Diagram Example

```

Foresight
MiniSpec: EEG Monitor/Compute Fourier Transform
Foresight V2.25c

inputs: l_rawdats;      -- array of 32 8-bit sampled data points
outputs: power_fft;    -- array of 16; power in each discrete frequency 0 - 15 Hz
locals: hz, real_var, imag_var, g, x;
static locals: pi;

Initialize:
Begin
  pi := 3.14152;      -- constant
End;

Procedure:
Begin
  for hz IN 0..15 loop -- compute power in each discrete frequency
    real_var := 0.0;
    imag_var := 0.0;
    for x IN 0..31 loop -- sum real and imag components over all samples
      g := 2.0 * pi * float(hz) * float(x) / 32.0;
      real_var := real_var + (float(l_rawdats(x)) * cos(g) / 32.0);
      imag_var := imag_var + (float(l_rawdats(x)) * sin(g) / 32.0);
    end loop;
    power_fft(hz) := 2.0 * sqrt(imag_var * imag_var + real_var * real_var);
  end loop;
End;

```

Header defines interface

Optional Initialization Block used to set initial conditions; executed at simulation startup

Procedure section executed each time mini-spec "fires"

Figure 5. Foresight Mini-specification



Academic Languages:

ADLIB	AHPI
BCI. (Base Conlan)	Cascade CDL
Chippe's HDL	CLASP
Conlan	DDL
DRI.	DSL
Ella	HardwareC
HSL	Ideal
IMBSL	ISP
ISPS	MIMOLA
SETI	Silage (DSP)
Simpact.	SpecCharts
STRUDEL	Torle_C
Wislan	VERS
YASL	Zeus

(IEEE Design & Test, June+Sept. '92, especially)

Corporate Languages

BLISS (DEC)	MAINSAIL (Intel)
RESQ (IBM)	TI-HDL (VHDL strawman)
V (IBM)	YLL (IBM)

Standard languages with simulation kernels

Ada	C (e.g. Qsim, Simpact)
C++ (e.g. µC++)	LISP
Modula-2 (Zeus)	Pascal
Smalltalk	

Dead Languages

SCALD	N.2
-------	-----

29

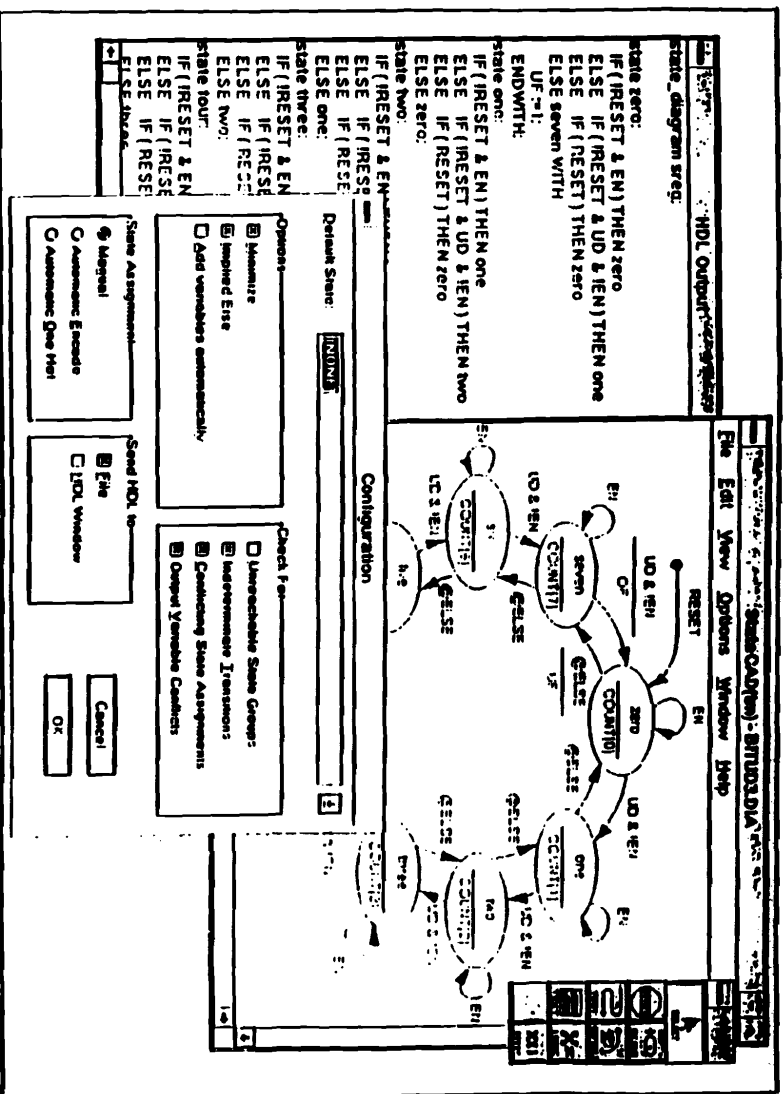


Figure 1. StateCAD state diagram, HDL output, and configuration menus.

Visual Software Solutions, Coral Springs, FL  
(800)-208-1051

Other-Other



## Analog Behavioral Modeling (continued)

VHDL-A (1076.1 analog extension)  
manual expected next quarter, ballot next year.  
(ASIC Design, Aug 1994)

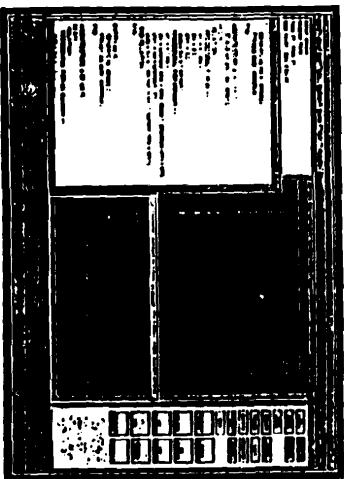
Verilog-A  
Cadence's SpectreHDL simulator accepts  
Verilog-A (analog) and VHDL-A, as well as  
Spice, for mixed level simulation  
(Computer Design, Aug. '94)

Silos Behavioral Language (SBL) for analog.  
Simucad

DIABLO, Intergraph

MIHDL (microwave)  
a very different beast  
behavior, function, structure and geometry  
on equal footing

232



### HDL-A

Develop comprehensive analog models for complex devices and systems with HDL-A, the first analog HDL based on the proposed VHDL extensions. HDL-A is a full analog behavioral modeling language that will work without an in-depth understanding of Spice. And, it can be used in conjunction with VHDL. The effort you need to expend to learn and use the language is minimal. You can also use HDL-A to develop models for systems that aren't purely electronic, such as electro-mechanical motors, actuators, transducers, and sensors, as well as optical and thermal models. HDL-A is a compiled system, offering the highest performance available in an analog behavioral language. It is tightly integrated into the environment through Accusim II and Design Architect and uses the same highlighting and other used in Mentor Graphics' digital VHDL products.

Mentor Graphics, Wilsonville, OR  
(508)-685-7000

The rest of the story...

The trap: implementation vs function  
Too much how, not enough what  
leads to decisions instead of choices

Early warning signs:  
when you say “bit” or “wire”  
instead of variable or link

Simulation can be large  
32-64 MByte RAM  
avoid swapping !!!  
100 MBytes disk  
traces, reports, partials

Simulation can be slow  
hours, days

Parallel simulation is still a research topic

The (rest of the) rest of the story...

Warmup...

most simulations are initially nonstationary  
(some never are stationary!)

How do you know when  
the statistics of the simulation are valid?

fixed initialization time, based on experience

fitting a distribution to an expected metric

design-of-experiment & power analysis...

It is typically better to run one long simulation,  
and “fold” results,  
than to run many independent copies...

(Pawlikowski, ACM Comp. Surveys, June '90)

## Then what? Synthesis?

synthesis can be chaotic  
strong sensitivity to initial conditions...

RTL synthesis makes gates out of equations  
limited scope optimization

Behavioral synthesis interprets algorithms, data flow  
schedules, allocates  
cross-register optimization

Synthesis generally  
focuses on data path  
assuming single-clock control (FSM)

Large design space requires boundaries  
to make synthesis computationally tractable.  
That means target architecture choices;  
that means local minima, not global...

## Synthesis (continued)

### Works (VHDL):

concurrent assignment  
processes  
package  
procedures and functions  
enumeration  
variables  
arrays  
component instantiation  
function/operator overloading

(Computer Design, Aug. '94)

### Doesn't:

assertion statements  
floating point (4.3 bits?)  
File I/O (ASIC/disk?)  
Configurations  
(one choice please)  
Scheduling delays  
Transport delays  
(effect, not cause)

**Rule #1 in VHDL synthesis: it won't happen  
when you think it will...**

(assume nothing about timing;  
use handshake signals,  
or synthesize/back-annotate/analyze)

(11 other rules: IEEE Design and Test, March '91)

## Synthesis "biggies":

Behavioral Compiler - Synopsis (VHDL/Verilog)

DSP Station/Mistral2 - Mentor (DSP focus;  
Mistral is the synthesis tool; uses DFL,  
a custom data flow language  
derived from Silage (UC Berkeley) )

BooleDozer - IBM/Altium  
(VHDL attributes for annotation)

Lambda - Viewlogic (interactive environment;  
rule-driven proof-of-correctness rather than  
comparing "before" and "after" HDL)

ArchGen - CAE+Plus (icon flowchart  
with C modules; output in VHDL/Verilog  
for further (external) synthesis))

(asic & eda, Aug. '94)

## Verification

Will it work?

static timing analysis

vs dynamic timing simulation

formal verification:

did the synthesis translation screw up?

test vectors:

did the ASIC foundry screw up?

## Validation

Will it do what you want it to do?

test vectors are not operational vectors

test passed = nothing broken

≠ it works in the system

(common ASIC complaint)

Behavioral simulation can guide you toward validation:

behavioral fault modeling

understanding the problem

side-by-side comparison, concept and product

**Odd bits:**

emacs macros (smart edit modes)  
grinddef macros (pretty printing)  
for VHDL, Verilog (public domain)

free (1000 line max) Verilog system from Wellspring

VHDL modeling guidelines  
European Space Agency  
psi@wd.estec.esa.nl

VITAL - VHDL Initiative Toward ASIC Libraries  
addressing the “no libraries” complaint of VHDL.  
“Sign-off quality” simulation  
vital@vhdl.org

**Beware of companies that “buy” their solutions:**

Intergraph - still merging Daisy/Cadnetix/Intergraph  
(2+ years...)

Viewlogic - “best of class” tools,  
but users complain of interoperability conflicts

Cadence - one of the more aggressive “land grabbers”

**To probe further:**

comp.lang.vhdl  
comp.lang.verilog  
especially the FAQs (vhdl, verilog)  
comp.lsi  
comp.lsi.cad  
comp.simulation

IEEE Design & Test  
IEEE Trans. Computer-Aided Design  
ACM Trans. Modeling and Computer Simulation  
Simulation (SCS)  
CACM  
IEEE Computer

Design Automation Conference (DAC)

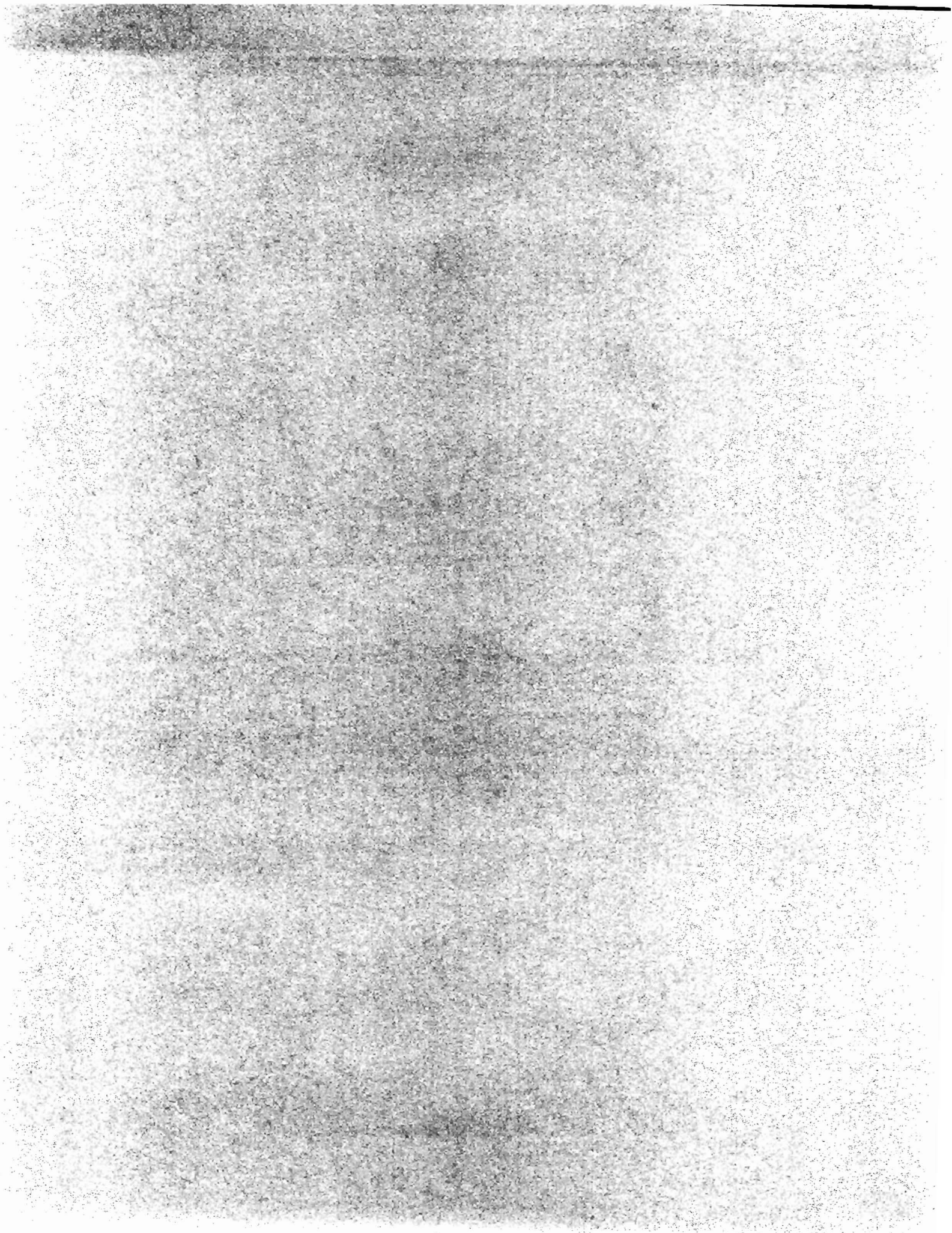
## **S6-3**

### **"Review of SCI Simulation Results"**

**(Andre Bogaerts - CERN)**

Various aspects of SCI have been modelled. This includes SCI Protocol Specification and Verification (IEEE C code), SCI interface and chip design by manufacturers using Verilog or VHDL, and the behavior of SCI Systems using C++, Simula or MODSIM. The RD24 Collaboration at CERN has developed SCILab (in MODSIM) to model Data Acquisition Systems for LHC. SCILab is also used in the TOPSCI Project to simulate the behavior of SCI switches. Recently, a major effort has been made to model the ATLAS Data Acquisition and Trigger System. Models of SCI and ATM based networks have been incorporated to evaluate and compare the behavior of these technologies in the same environment.





## **Review of SCI Simulation Results**

André Bogaerts

CERN, 1211 Geneva 23 Switzerland

Bin Wu

Physics Department, University of Oslo, POB 1048, Blindern, 0316 Oslo, Norway

Data Acquisition Conference

26-28 October 1991

Fermilab

### **Useful addresses:**

André Bogaerts: [bogaerts@dxcern.cern.ch](mailto:bogaerts@dxcern.cern.ch)

Bin Wu: [bin.wu@fys.uio.no](mailto:bin.wu@fys.uio.no)

anonymous ftp server of RD24 (SCI):  
[sunsci.cern.ch](http://sunsci.cern.ch) (directories sci and simulation)

e-mail reflector (ATLAS Simulations): [simulation@sunsci.cern.ch](mailto:simulation@sunsci.cern.ch)

WWW access to RD24:

<http://www1.cern.ch/RD24>

or use CERN Home Page, select "Research and Developments", "RD24"

## **Table of Content**

**Overview of SCI Simulations**

**SCI Simulations with SCILab**

**Switch Simulations**

**Modelling of the ATLAS DAQ and Trigger System**

**List of Publications**

## SCILab

### Behavioural Simulation of SCI Networks

- simulates transmission of SCI packets over a network (rings, switches) and handles contention
- models SCI protocols: packet transmission (request, response, echo), transactions (through "agents"), bandwidth allocation ("go bit"), queue acceptance ("packet busying") and cache coherency (based on IEEE C code)
- simulates SCI node chip internal delays, pipelining and interconnect backends
- connections, link delays, switches, memories, processors
- higher level objects (e.g. ATLAS Trigger Processors, Front End Memories) are derived from basic SCI nodes
- investigate the influence of SCI protocols, link speed and interface design under varying loading conditions for different topologies
- determine critical parameters such as SCI link traffic, throughput and latencies
- initially aimed at large HEP data Acquisition Systems but also suitable for general SCI networks consisting of rings interconnected by switches
- used for studying simple rings, switch design (TOPSCI project), switch based Data Acquisition Systems (64 X 64 multi-stage switch - 25 Gbytes/s)
- results have been compared with other independent simulations
- part of SIMDAQ (simulation of the ATLAS 2nd Level Trigger)
- new developments (started): modelling of AHCs, re-integration of IEEE C-code for cache coherency
- future work: models of specific commercial boards, SCI software comparison with measurements of Laboratory test benches

References (<http://www1.cern.ch/RD24/>)

SCILab Cookbook (Users Guide)

SCI Simulations with SCILab ("two pager")

SCI Publications (references to other documents)

Distribution:

SCILab is CERN copy-righted (contact [bogaerts@dxcern.cern.ch](mailto:bogaerts@dxcern.cern.ch))

### Table 1: Overview of SCI Simulation Work

Group/Location	Main Interest	Simulated Systems	Tools, Methods	e-mail Contact
Univ. of Oslo (Dept. of Informatics)	Cache Coherency, MP, RamLink	(Coherent) Rings, MP	SIMULA, C, MIPS simulator, IEEE C-code	<a href="mailto:gjesang@iti.uio.no">gjesang@iti.uio.no</a>
Oslo (SINTEF/Univ. of Oslo/Dolphin)	Performance (throughput, latency)	Rings, Networks, OM/HIC Switches	C++	<a href="mailto:Ernst.Kraussman@si.sintef.no">Ernst.Kraussman@si.sintef.no</a>
RD24 (CERN, Univ. of Oslo)	Performance, HEP Data Acquisition	Rings, Networks, Switches, DAQ	MOOSIM II, IEEE C-code	<a href="mailto:bogaerts@dxcern.cern.ch">bogaerts@dxcern.cern.ch</a>
Univ. of Edinburgh	(Coherent) Shared Memory, MP	(Coherent) Rings	SPLASH, C-code	<a href="mailto:tho@ics.edinburgh.ac.uk">tho@ics.edinburgh.ac.uk</a>
IEEE/Apple Corp	Protocol Specification and Verification, Coherency	(Coherent) Rings, MP	IEEE C-code, SUN FWP	<a href="mailto:dvt@apple.com">dvt@apple.com</a>
Univ. of Wisconsin	Rings, Cache Coherency, MP	Rings	C-code, amaran	<a href="mailto:sls@omulus.cray.com">sls@omulus.cray.com</a>
USCLA (Los Angeles)	Coherent MP	Rings	trac, drcos	<a href="mailto:harris@paris.usc.edu">harris@paris.usc.edu</a>
UCSD (San Diego)	DSP applications	Rings	MOOSIM II	<a href="mailto:riceman@ucsd.edu">riceman@ucsd.edu</a>
Univ. of Waikato (New Zealand)	SCI Systems	Subset of SCI Standard	Time Warp	<a href="mailto:DF-WIT@hamilton.mwa.cri.nz">DF-WIT@hamilton.mwa.cri.nz</a>

## SCILab Components

### SCILab has been tested on SUN/SPARC

- core implemented in MODSIM II<sup>®</sup> (release 1.9), a commercial object-oriented system for discrete event simulation from CACI Products Company, La Jolla, Ca. SIMOBJECT and COMNET III are under investigation
- cache coherency code from IEEE (compiled with GNU C compiler)
- input/output pre/post processors based on UNIX scripts using sed, awk and cc
- switch configurations and routing tables generated by TopoEngine which is a C program
- PAW is used for data presentation
- filters (UNIX scripts) exist for other graphics
- input based on ASCII configuration files (transformed by the cc)
- output in ASCII files suitable for PAW and many other graphics packages
- SCIMP (SCI Modelling Program) can be used as a stand alone program to model a large variety of SCI networks using built-in models of parameterized SCI nodes
- SCI code (in the form of MODSIM Object libraries) may also be linked into other simulation environments (e.g. the simulation of the ATLAS DAQ and Trigger System, SIMDAQ)

## Basic Model of an SCI Node

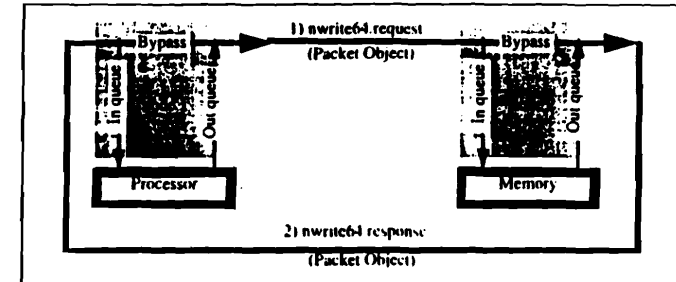


FIGURE 1. A simple 2-node SCI system with 1 processor and 1 memory

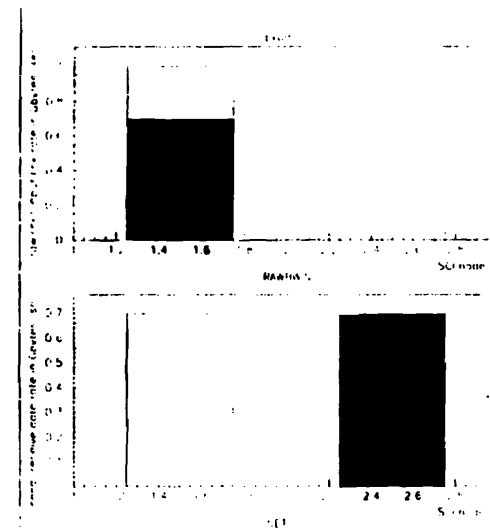
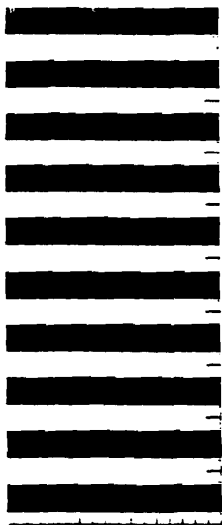
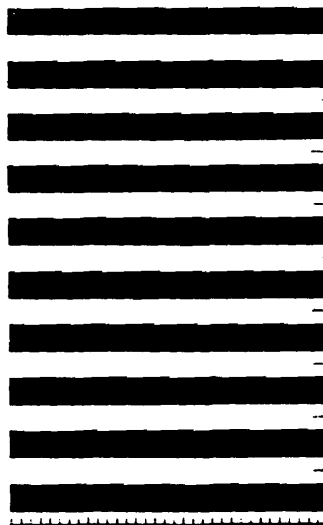
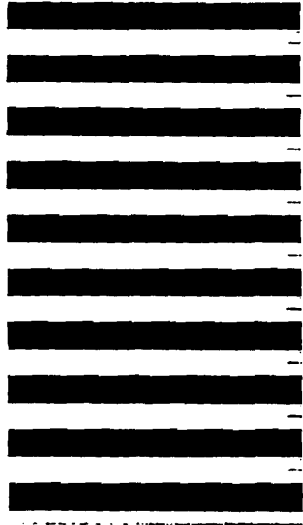


FIGURE 2. Plot of example0.out with PAW

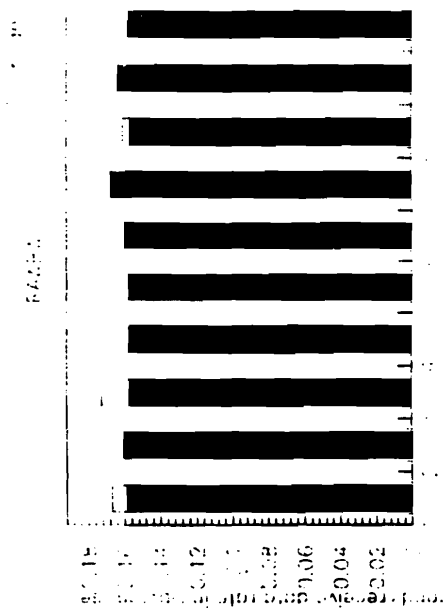
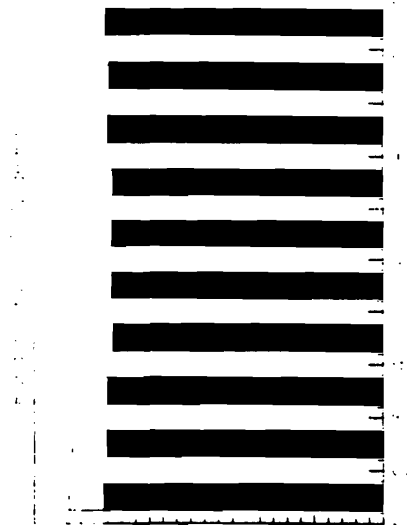
**SCI like Behaviour**



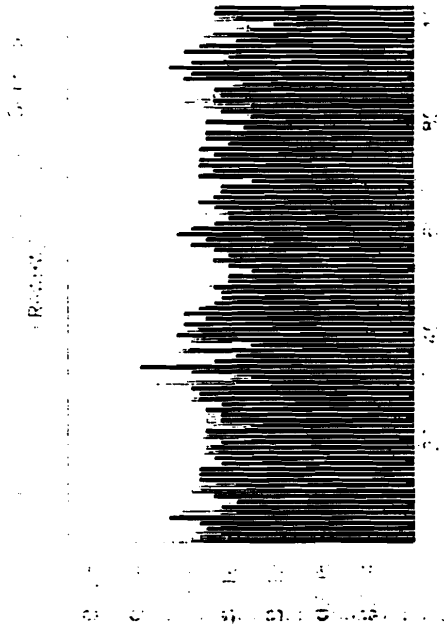
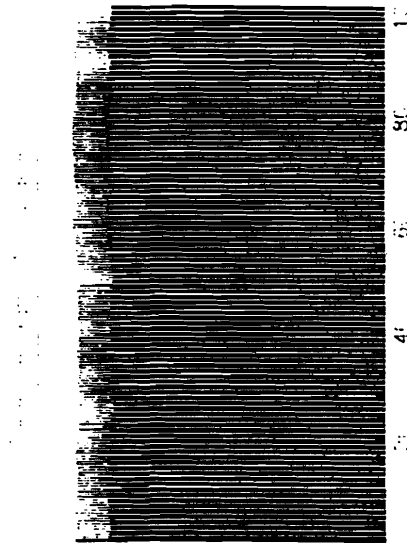
**Bus like Behaviour**



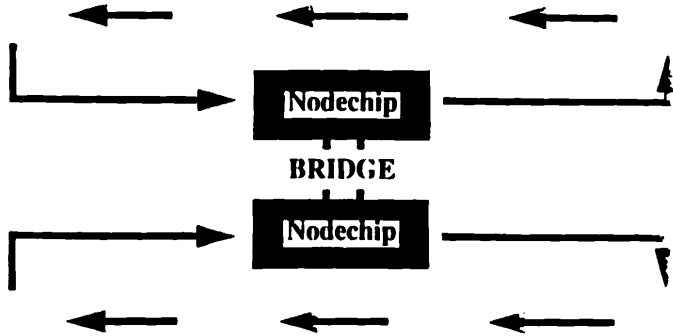
### Random Traffic - 10 Nodes



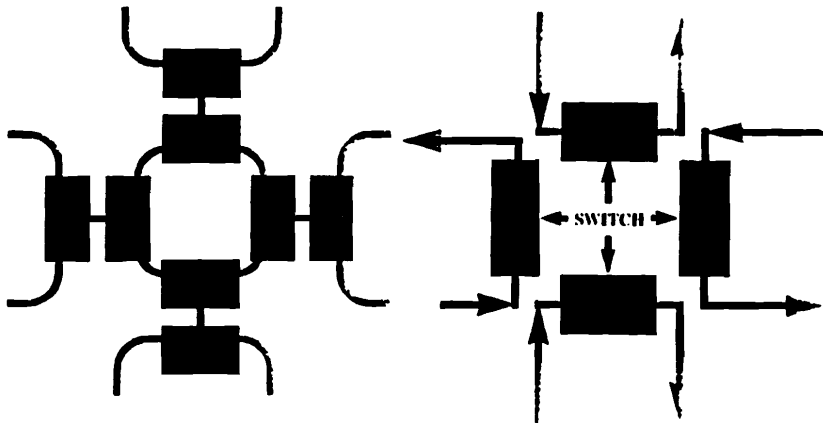
### Random Traffic - 100 Nodes



## Bridges and Switches



An SCI bridge constructed from 2 Nodechips "back to back" interconnects two rings.

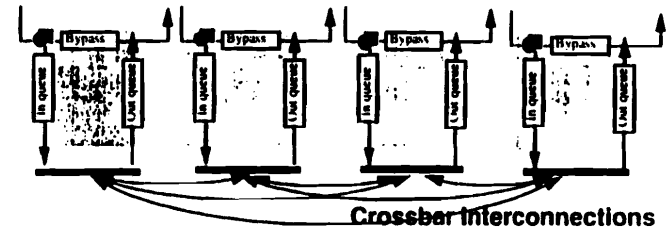


A pseudo switch constructed from 4 bridges.

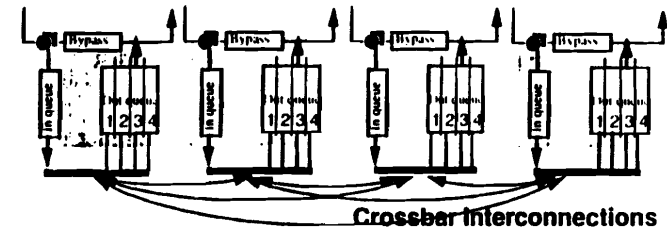
A real switch has better performance

## Internal Switch Architectures

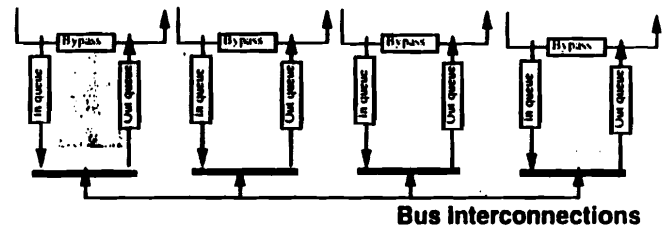
Which switch is best ?



a) Switch Object, with typeid = 1, CrossSwitch Model



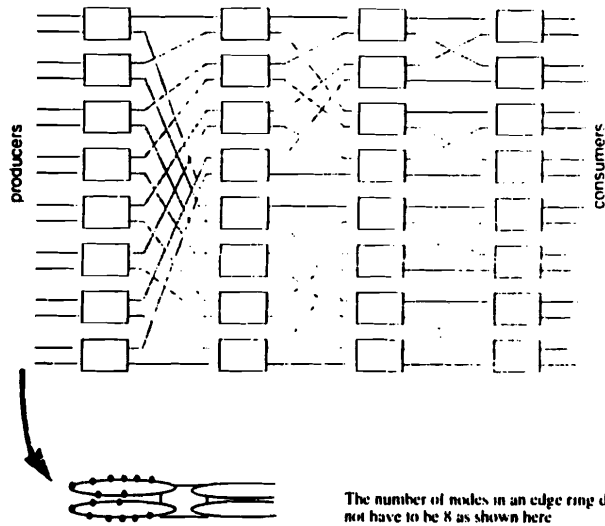
b) Switch Object, with typeid = 4, SwitchLink Model



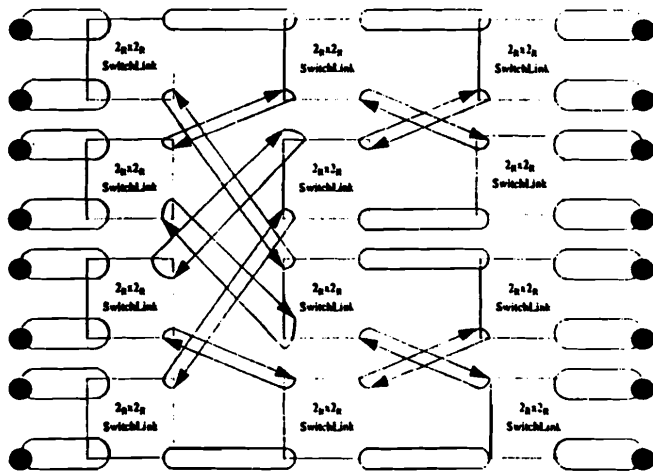
c) Switch Object with bus at back side, with typeid = 5

FIGURE 1. The CrossSwitch model, the SwitchLink model and the bus-based switch model

## Multistage Crossbar Switch (Banyan)



### 16<sub>R</sub>x16<sub>R</sub> crossbar switch constructed from 32 2<sub>R</sub>x2<sub>R</sub> chips



### 8<sub>R</sub>x8<sub>R</sub> SCI crossbar switch constructed from 12 2<sub>R</sub>x2<sub>R</sub> chips

## 2-D Mesh

### Topology and Deadlock-free Routing

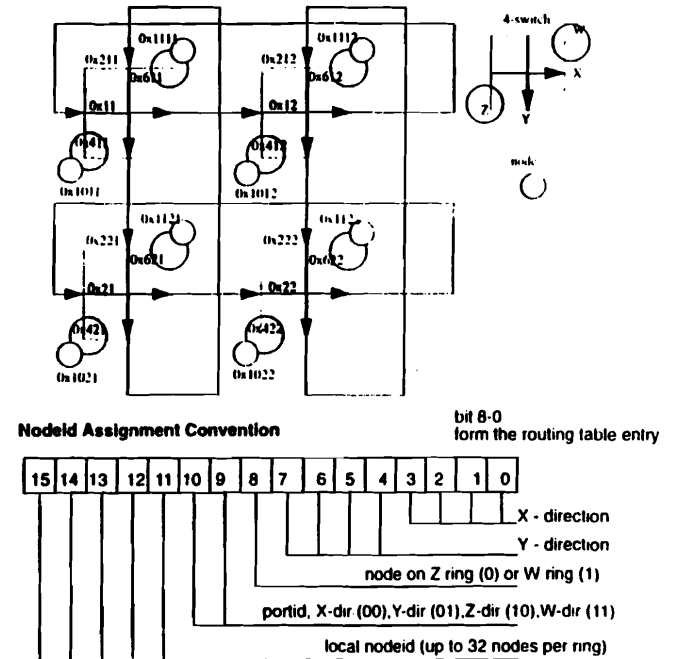
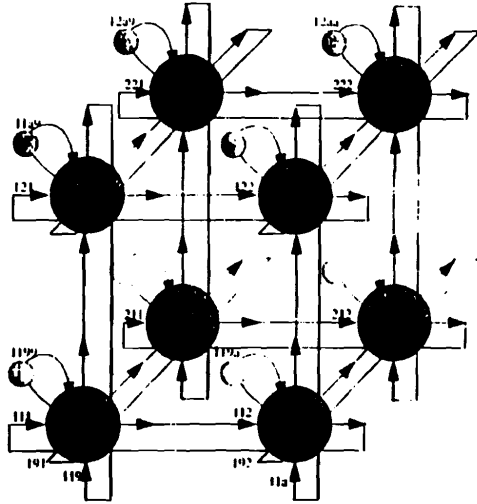


FIGURE 1. 2-d mesh system configuration and nodeid assignment convention

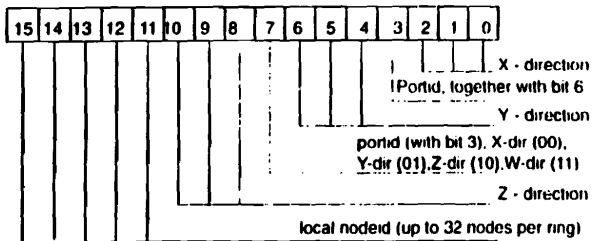


# 3-D Mesh

## Topology and Deadlock-free Routing



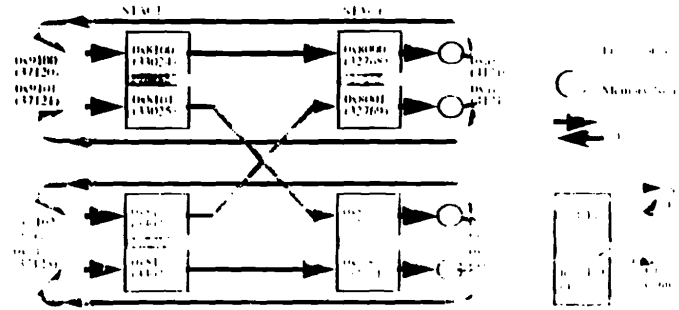
### Nodeid Assignment Convention



bit 10, 9, 8, 6, 5, 4, 2, 1, 0  
form the routing table entry

FIGURE 1. 3-d mesh system configuration and nodeid assignment convention

# Unidirectional Multistage Networks



### Nodeid Assignment Convention

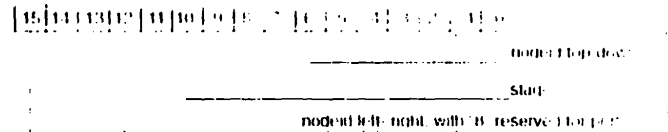
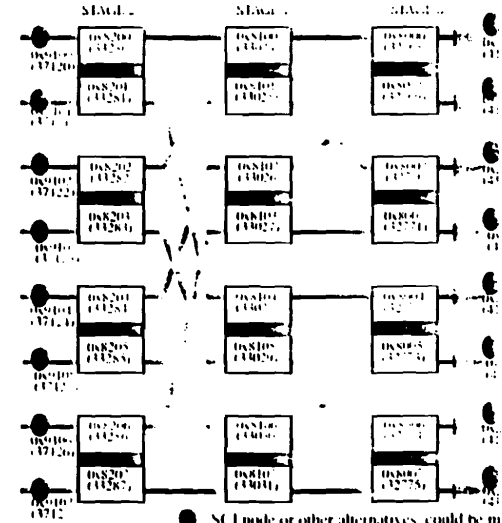


FIGURE 1. A two-stage unidirectional multistage network with self-routed and nodeid assignment convention



● SCI node or other alternatives, could be main

FIGURE 2. A three-stage 8x8 unidirectional multistage system interconnected by

# Scaling of Multistage Crossbar Switches

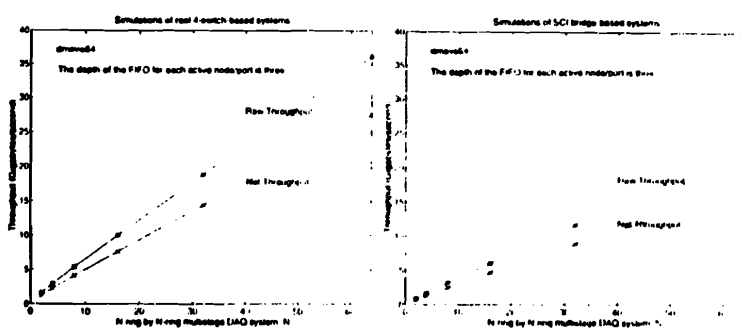


FIGURE 1. a, b. Test of throughput scalability of real 4-switch-based system and bridge-based system.

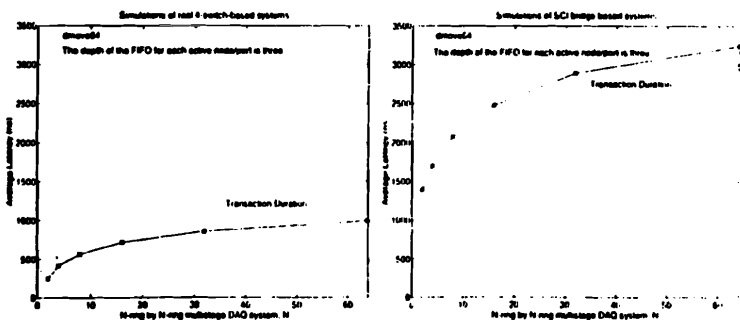


FIGURE 2. a, b. Latency of real 4-switch-based system and bridge-based system.

## Conclusion:

Throughput of a (composite) multistage crossbar switching network scales with the number of elementary switch elements and is insensitive to the internal architecture of its constituents.

# How much Internal Buffer Memory ?

## Optimal Queue Size

Table 1. Finding the optimal queue size in a 4-switch, dmove64 operation

# queues	Blink speed=1 GByte/s, Store-and-forward	Blink speed=1 GByte/s, Virtual-cut through	Blink speed=2 GByte/s, Store-and-forward	Blink speed=2 GByte/s, Virtual-cut through
1-queue <sup>a</sup>	0.67/0.51/2.04/970 <sup>b</sup>	0.78/1.60/1.84/745	0.92/0.70/1.69/687	1.14/1.87/1.17/478
2-queues	0.92/0.70/1.68/1071	0.94/1.71/1.66/970	1.64/1.25/0.56/521	1.77/1.35/0.29/416
3-queues	0.95/0.72/1.63/1381	0.95/0.72/1.64/1304	1.85/1.41/0.19/583	1.88/1.43/0.13/517
4-queues	0.96/0.73/1.62/1731	0.96/0.73/1.64/1651	1.89/1.44/0.12/703	1.89/1.44/0.11/646

a. 1-queue means one-packet-deep input-queue and one output-queue. n-queues means n-packet-deep input-queue and n output-queue  
 b. system's raw throughput(Gbyte/s)/net throughput(Gbyte/s)/crry. throughput(Gbyte/s)/average la.tency(ns)

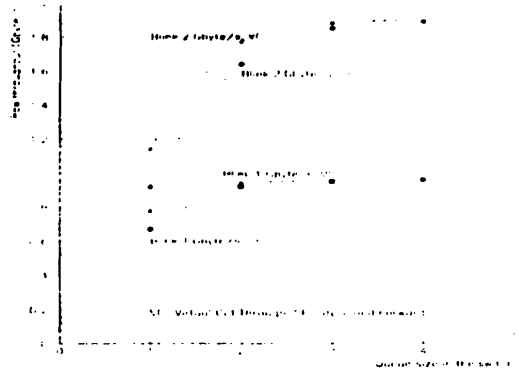


FIGURE 1. The queue size in the switch versus raw throughput of the system

# The Effect of Blink on the Throughput

## Store and Forward

Table 1. Performance of a 4-switch with Blink speed of 1, 2, 3, 4 Gbyte/s, Store-and-forward, dmove64.

# FIFOs	Blink speed=1 GByte/s, Store- and-forward	Blink speed=2 GByte/s, Store- and-forward	Blink speed=3 GByte/s, Store- and-forward	Blink speed=4 GByte/s, Store- and-forward
1-queue <sup>a</sup>	0.67/0.51/2.04/970 <sup>b</sup>	0.92/0.70/1.69/687	1.05/0.80/1.33/567	1.09/0.83/1.26/539
2-queues	0.92/0.70/1.68/1071	1.64/1.25/0.56/521	1.94/1.48/1.48/582	2.11/1.61/1.06/505
3-queues	0.95/0.72/1.63/1381	1.85/1.41/0.19/583	2.38/1.82/0.98/673	2.60/1.98/0.46/545
4-queues	0.96/0.73/1.62/1731	1.89/1.44/0.12/703	2.59/1.97/0.78/819	2.81/2.14/0.21/612

a. 1-queue means one-packet-deep input-queue and one output-queue. n-queues means n-packet-deep input-queue and n output-queue

b. system's raw throughput(Gbyte/s)/net throughput(Gbyte/s)/ctx. throughput(Gbyte/s)/average latency(ns)

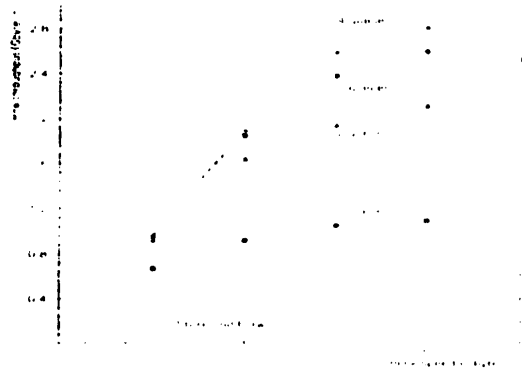


FIGURE 1. The effect of Blink speed on system throughput, store-and-forward

# The Effect of Blink on the Throughput

## Virtual cut through

Table 1. Performance of a 4-switch with Blink speed of 1, 2, 3, 4 Gbyte/s, Virtual cut through, dmove64

# FIFOs	Blink speed=1 GByte/s, Virtual- cut through	Blink speed=2 GByte/s, Virtual- cut through	Blink speed=3 GByte/s, Virtual- cut through	Blink speed=4 GByte/s, Virtual- cut through
1-queue <sup>a</sup>	0.78/0.60/1.84/745 <sup>b</sup>	1.14/0.87/1.17/435	1.27/0.97/0.94/419	1.34/1.01/0.89/398
2-queues	0.94/0.71/1.66/970	1.77/1.35/0.29/416	2.13/1.62/0.05/462	2.33/1.78/0.57/372
3-queues	0.94/0.71/1.66/970	1.88/1.44/0.13/517	2.49/1.90/0.72/585	2.69/2.05/0.28/452
4-queues	0.95/0.73/1.64/1651	1.89/1.44/0.11/646	2.64/2.01/0.68/761	2.85/2.17/0.12/514

a. 1-queue means one-packet-deep input-queue and one output-queue. n-queues means n-packet-deep input-queue and n output-queue

b. system's raw throughput(Gbyte/s)/net throughput(Gbyte/s)/ctx. throughput(Gbyte/s)/average latency(ns)

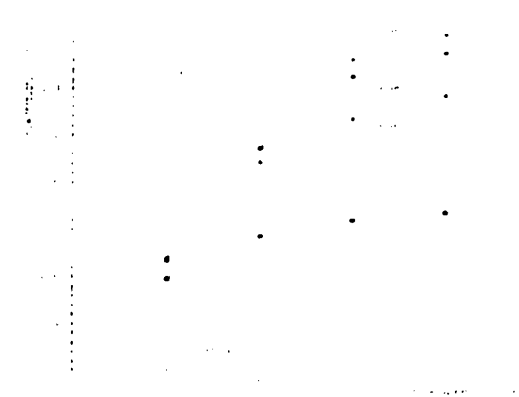


FIGURE 1. The effect of Blink speed on system throughput, virtual cut-through

## Effect of Pipelining on Latency

1. for 1 Gbyte/s Blink: 432/368/320/272 ns for a BPASS value of 0, 1, 2 or 3 respectively
2. for 2 Gbyte/s Blink: 348/300/264/228 ns for a BPASS value of 0, 1, 2 or 3 respectively

The latency is measured from the time a packet is in the sender's output queue till the whole packet is in the receiver's input queue for a dmove transaction.

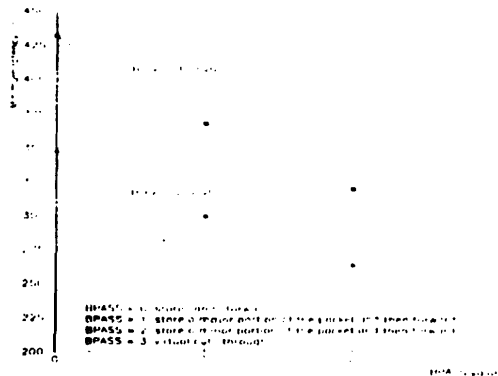


FIGURE 1. BPASS value versus minimum latency

### Conclusion

Virtual cut-through technique is specially effective when the size of the queue is one. The throughput can be 20% higher than store-and-forward.

The latency figures for various BPASS values show that virtual cut-through is the best. If we cascade a number of switches in a large system, this could be a big gain.

## Effect of Routing Delay on Throughput

All simulations so far use 2 ns routing delay needed for decoding the incoming packets and making routing decision. Two nanoseconds may be achievable with routing based on bit masking. This may be longer with table lookup. We only run simulation with BPASS = 11 in virtual-cut-through technique.

Table 1. The influence of routing delay on system performance, dmove64, BPASS=11

RD	2ns	4ns	8ns	16ns	32ns	64ns	128ns
1-queue, 1 Gbyte/s	0.790/60 1.84/745 <sup>a</sup>	0.790/60 1.80/735	0.790/60 1.74/736	0.780/59 1.63/753	0.720/55 1.45/802	0.700/53 0.87/870	0.610/47 0.75/936
1-queue, 2 Gbyte/s	1.140/87 1.17/478	1.140/87 1.12/476	1.140/87 1.07/481	1.100/84 1.01/499	1.020/78 0.93/540	0.920/70 0.73/598	0.750/57 0.55/736
2-queue, 1 Gbyte/s	0.930/71 3.25/1420	0.930/71 3.22/1316	0.930/71 3.21/1315	0.930/71 3.19/1314	0.930/71 3.11/1308	0.930/71 2.51/1314	0.900/67 1.82/1311
2-queue, 2 Gbyte/s	1.72/131 1.94/661	1.72/131 1.92/659	1.72/131 1.88/655	1.71/130 1.80/656	1.67/128 1.69/664	1.61/127 1.44/697	1.45/113 0.95/761

a. system's raw throughput(Gbyte/s)/net throughput(Gbyte/s)/cpu throughput(Gbyte/s)/average latency(ns)

FIGURE 1. Routing delay effect on system throughput

### Conclusion

The routing delay will not affect system throughput when it is reasonably low. Especially, for a delay of 2 ns to 16 ns, the system throughput does not vary much.

## Throughput for dmove64, nwrite64, nread64

### dmove/nwrite/nread

The write, read operations can cause different system behavior than the move operation. We compare the dmove64, nwrite64 and nread64 with BPASS = 11 in virtual-cut through technique.

Table 1. Simulation of the switch with different packet types and Blink speed of 1 Gbyte/s

# FIFOs	dmove64	nwrite64	nread64
1-queue <sup>a</sup>	0.79/0.60/1.83/745 <sup>b</sup>	0.79/0.48/2.57/2835	0.78/0.47/2.68/5943
2-queues	0.93/0.71/3.24/1320	0.90/0.56/3.48/5252	0.90/0.56/3.37/6220
3-queues	0.95/0.73/3.30/1973	0.93/0.58/3.45/6554	0.92/0.57/3.52/8084
4-queues	0.95/0.73/3.32/2658	0.93/0.58/3.45/7676	0.93/0.58/3.58/9283

a. 1-queue means one-packet-deep input-queue and one output-queue. n-queues means n-packet-deep input-queue and n output-queue.  
 b. system's raw throughput(Gbyte/s)/net throughput(Gbyte/s)/retry through put(Gbyte/s/average latency/s)

Table 2. Simulation of the switch with different packet types and Blink speed of 2 Gbyte/s

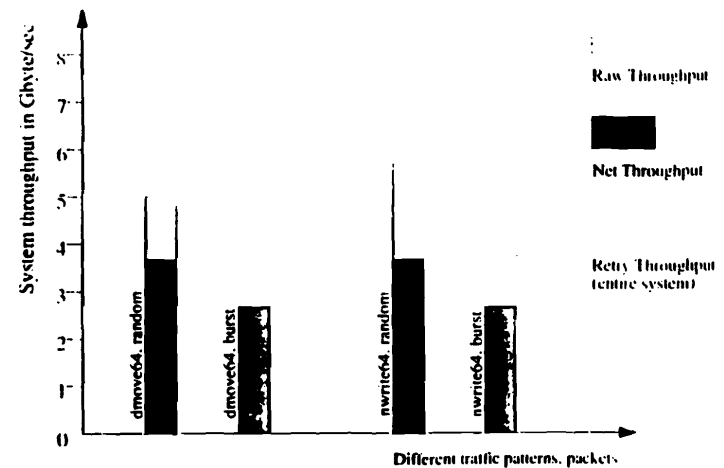
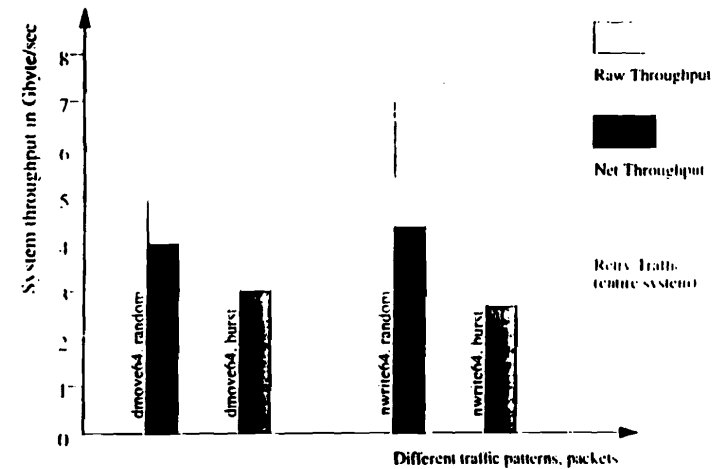
# FIFOs	dmove64	nwrite64	nread64
1-queue	1.14/0.87/1.17/478	1.27/0.78/1.69/2372	1.27/0.78/1.73/3013
2-queues	1.71/1.30/1.94/661	1.70/1.04/2.19/2869	1.67/1.02/2.29/5057
3-queues	1.82/1.39/2.05/964	1.81/1.12/2.12/3317	1.79/1.09/2.18/5351
4-queues	1.87/1.42/2.07/1276	1.85/1.14/2.09/4722	1.84/1.12/2.10/6191

## Conclusion

It is quite clear that the system performance is limited by Blink speed. Thus, whether the traffic is dmove64 or nwrite64 does not influence system's raw throughput. The latency of nwrite and nread is longer due to the response time which is also included.

## Random vs. Bursty Switch Traffic

Burst size is 15 packets of 64 bytes (960 bytes) [cf. "Bursty Traffic", Bin Wu]



## SCI Switches Summary and Conclusion

- Several switch designs have been investigated (internal SCI ring, bus, true cross-bar). Internal bus is acceptable only if it is several times faster than an SCI link.
- Different topologies have been studied: 2-d mesh, 3-d mesh, unidirectional multistage network and banyan switches, the latter having good characteristics for event builders.
- Throughput of multi-stage cross-bar switches scales linearly with their size. A  $N_R \times N_R$  switch ( $N_R$  input Rings,  $N_R$  output Rings) based on a  $2_R \times 2_R$  elementary switch chip requires  $(N/2) \log_2 N$  chips
- Optimum depth for each of the 4 fits (request/response in/out) of each switch port is 2-3 packets (each packet is 64 bytes of data + 16 bytes header)
- Internal switch capacity must be several times the capacity of a single SCI link.
- Internal pipelining improves latency and throughput.
- Routing latency has little influence on throughput
- Throughput difference between random/burst traffic patterns is not dramatic ( ~ 20%, depends on internal buffering)
- Two switches in preparation: CMOS ( $2_R \times 2_R$  on one board, SCI link = 200 Mbytes/s, Blink = 400 Mbytes/s) and GaAs ("TOPSCI Eureka project",  $2_R \times 2_R$  on Si substrate, SCI link = 1 GBytes/s, Blink = 2 Gbytes/s)
- Table below shows size/performance/cost of TOPSCI and CMOS switches (performance derived from simulation, for CMOS scale performance by 1/5). Use of high-performance switches is probably cost-effective.

Table 1. Size, throughput and cost of a multistage crossbar switch based on high-performance GaAs (SCI link = 1Gbytes/s, Blink = 2Gbytes/s) or CMOS (SCI link = 200 Mbytes/s, Blink = 400 Mbytes/s)

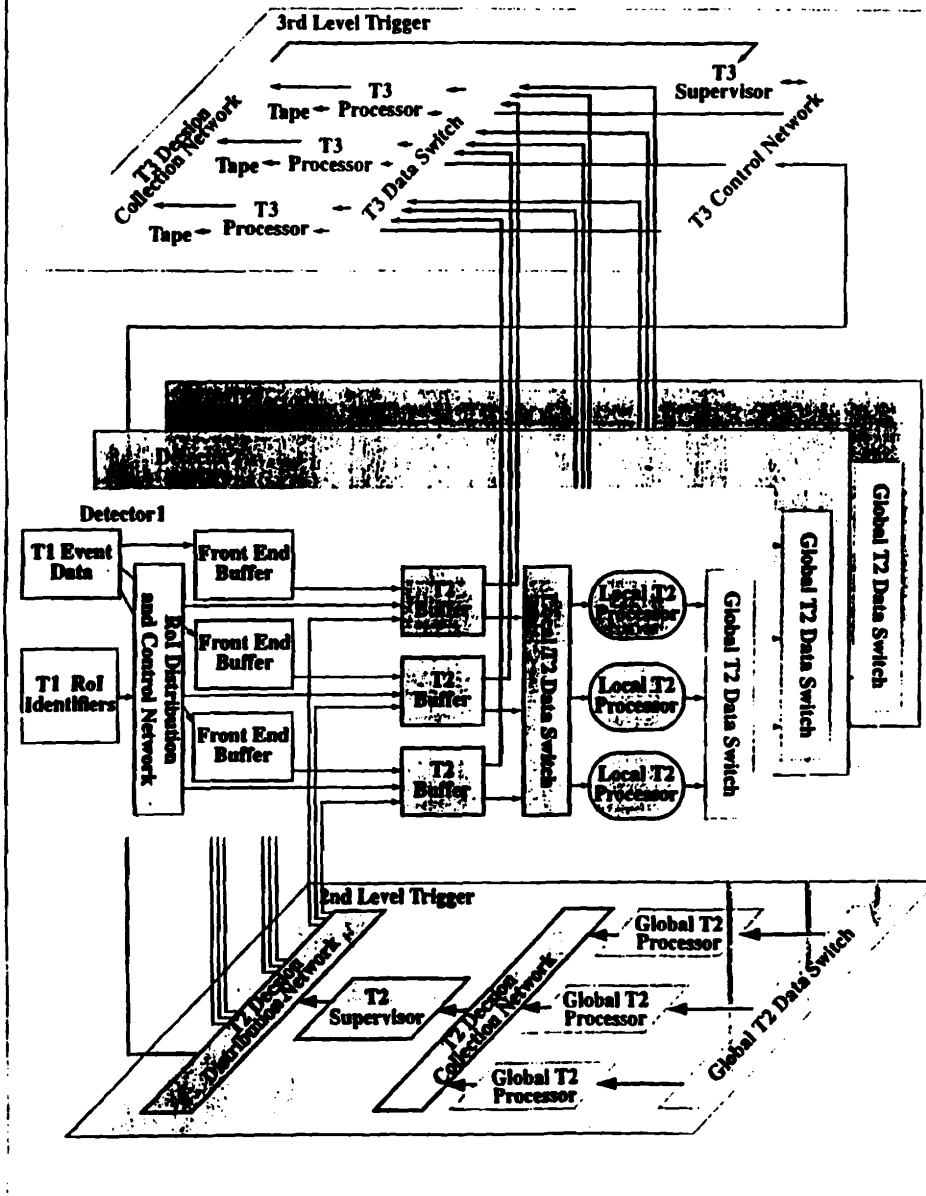
switch size (# rings)	Throughput (GBytes/s)	Cost (in # $2_R \times 2_R$ modules)
2x2	1 / 0.2	1
4x4	2 / 0.4	4
8x8	4 / 0.8	12
16x16	8 / 1.6	32
32x32	16 / 3.2	80
64x64	32 / 6.4	192

## SIMDAQ

### Model of the ATLAS Data Acquisition and Trigger System

- discrete event simulation of ATLAS DAQ and Trigger System written in MODSIM II
- simulates DAQ/Trigger Processors, Memories, Networks, Switches
- reads results from Physics Simulations to generate data and hit patterns in ATLAS detectors
- contains a model of the partitioning of the readout electronics to convert hit patterns (in  $\eta/\phi$  space) into # bytes associated with electronics channels
- allows a choice for models of Networks: generic (abstract), ATM and SCI. Others (C101 and Fiber Channel are in preparation)
- based on work by the ATLAS DAQ and Trigger Working Group, SIMDAQ has been implemented over a period of ~ 5 months with participation from many institutes. Special thanks to:
  1. data from Physics Simulations, Event Server: Jed Carter, Reiner Hauser, Christian Hortnagl
  2. detector electronics, partitioning: Patrick Ledu, Rudy Bock
  3. generic model: Stephen Hunt, Krys Korcyl, Ralph Spiwoks, Kamel Djidi
  4. T2 algorithms: Iosif Legrand
  5. ATM models: Denis Calvet
  6. SCI models: Hui Li, Rubina Chaudry, Bin Wu, Bernhard Skaali
  7. output, histogramming: Christian Hortnagl
  8. T2 architectures: Nick Ellis, John Strong, Livio Mapelli
  9. coordinators: Frank Harris, Andre Bogaerts

Overview of ATLAS DAQ and Trigger System



## ATLAS T2 Architectures

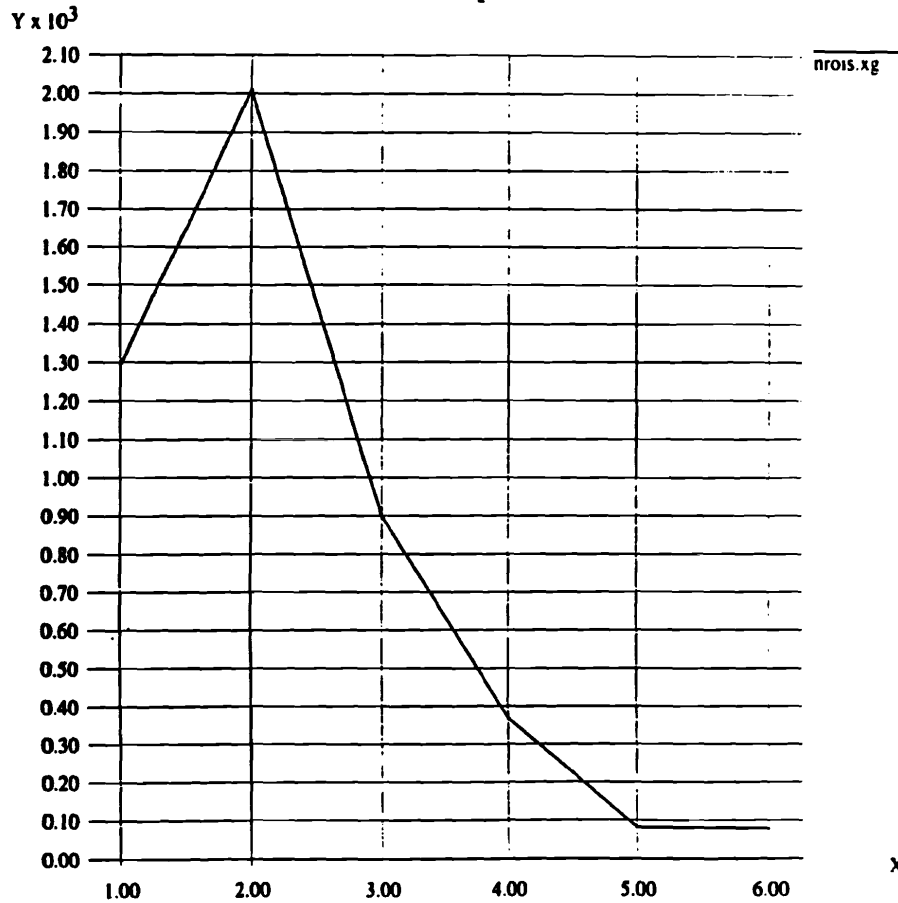
### Preliminary Configurations and Results

- SIMDAQ allows the study of DAQ and Trigger Architectures in terms of number and partitioning of buffers, processors, networks, switches, synchronisation protocols, timing of algorithms
- results are Trigger decision latencies, buffer occupancies, load on the network, size (and cost estimate) of the system
- evaluation of different technologies
- status:
  1. ATLAS EMC 512 buffers or 32 super-buffers
  1. ATLAS HAC 128 buffers or 8 super-buffers
  1. # Local T2 processors: 64, 128 or 256 for timing of 150, 300 or 600  $\mu$ s
  1. # Global T2 Processors: 64 for timing of 200  $\mu$ s
- comparison of generic, ATM and SCI switch network
- results:
  1. latency of T2 decision
  1. life time of events
  1. T2 buffer occupancy

21/10/94

Number of ROI's per  
event for the file  
tdump..51.data

X Graph



21/10/94

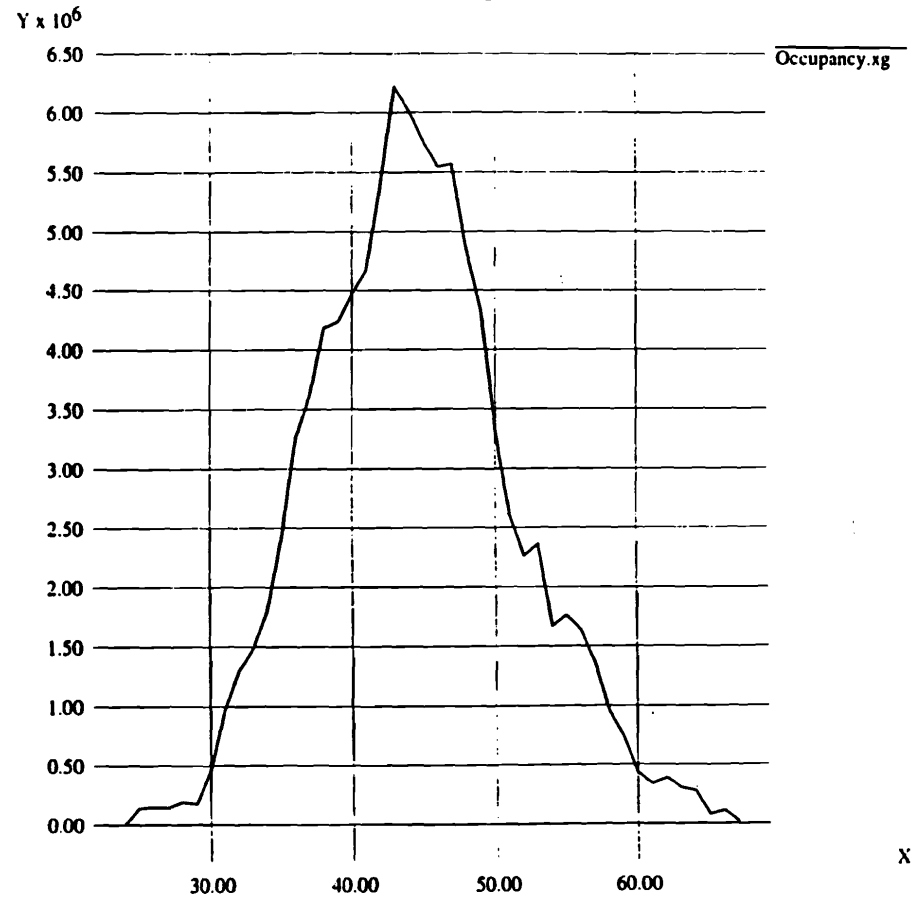
50ms run reset @ 1ms  
tdump 51.data  
Generic at Local & Global  
Crate EMC & MAC - front  
Occupancy of T2 Buffers.

FEX:  
150μs

L = 64 / p

G = 64

X Graph



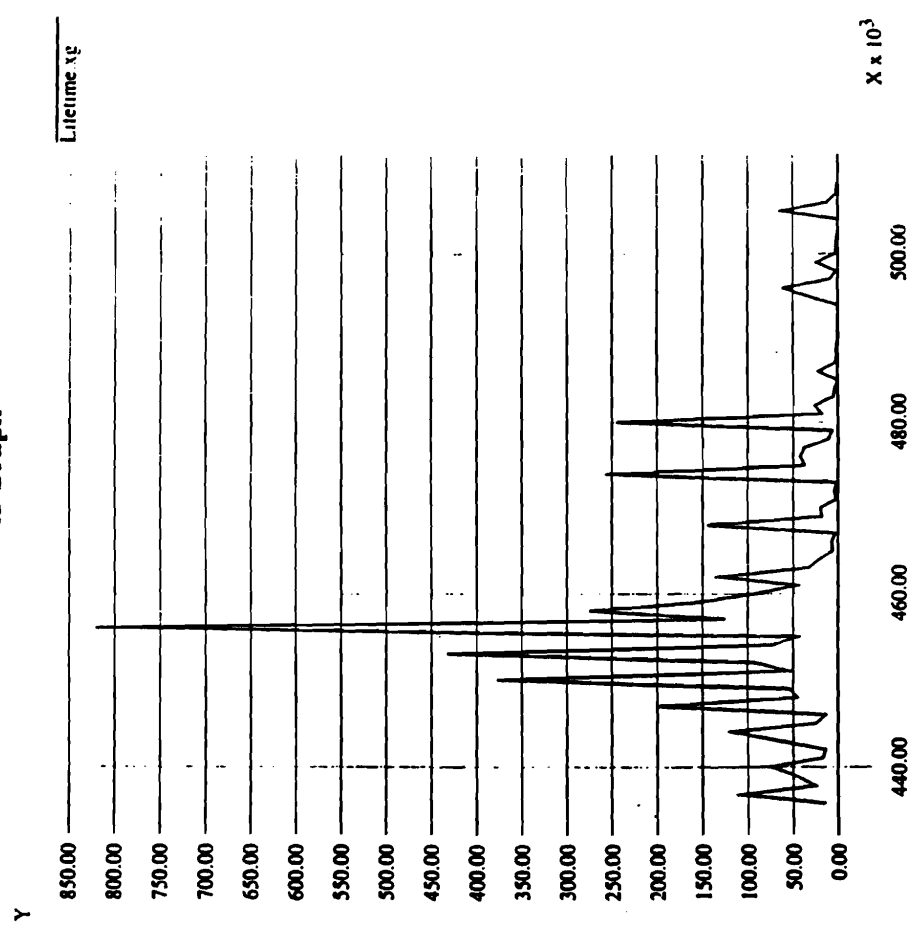


11/10/94

50ms run result @ 1ms  
tdump... 51 data  
Generic at Corel & Global  
EMC & MAC - Hudson  
Occupancy of tvents.

FEX  
150µs  
L = 64  
G = 64

### X Graph

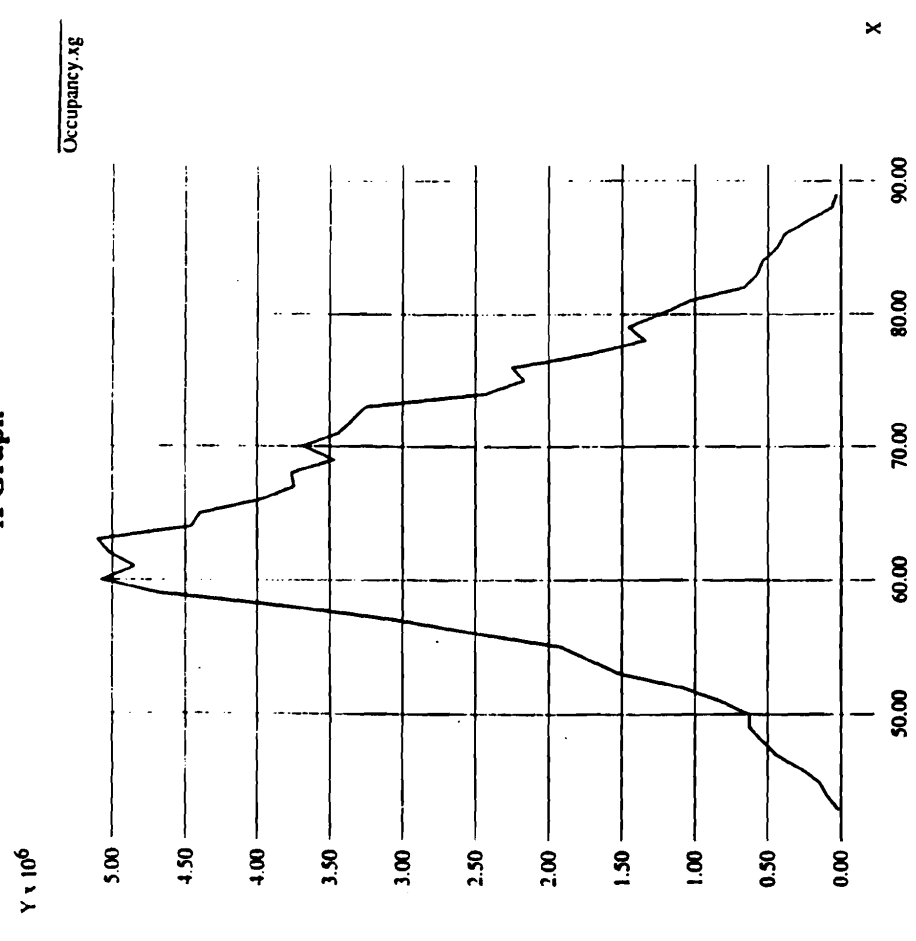


71/10/94

50ms run Result @ 1ms  
tdump... 51 data  
ATM Alcatel 622 Local  
Generic Global  
Corel MAC & EMC - Hudson  
Buffer Occupancy

FEX:  
150µs  
L = 64  
G = 64

### X Graph

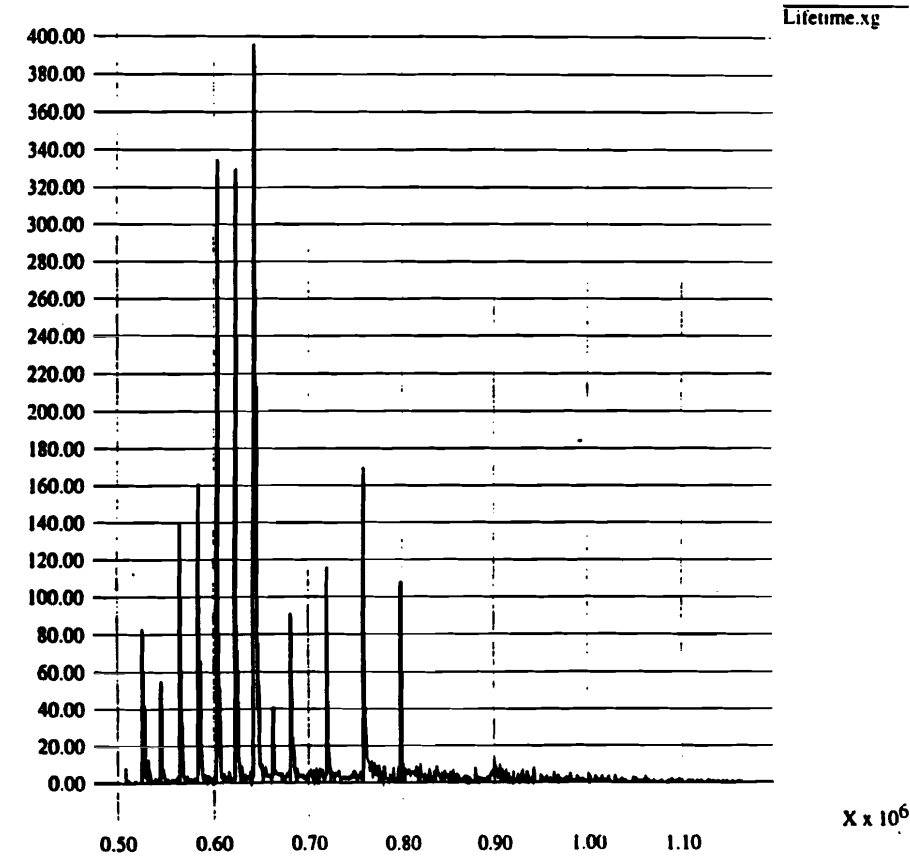


21/10/94

50ms run Reset @ 1ms  
+dump 51 data  
ATMAcatel622 Local  
Generic Global  
Create MAC & EMC - fasttran  
Lifetime of Events

FEX:  
150ms  
L = 64 / ?  
G = 64

X Graph

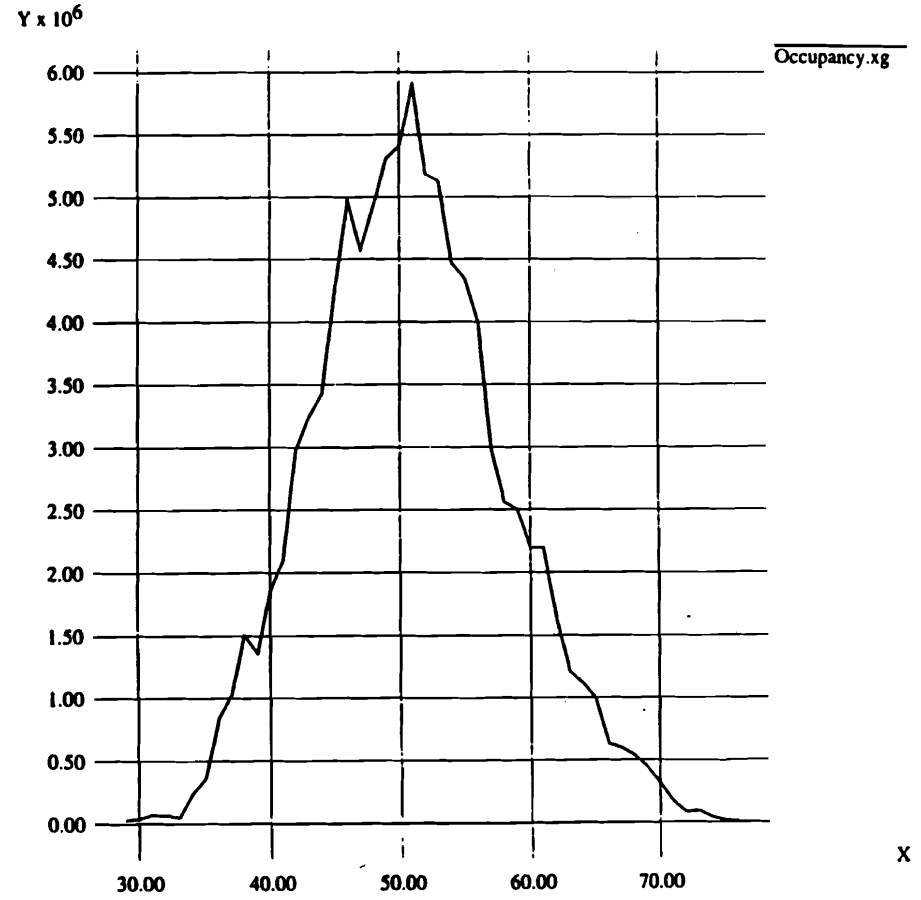


21/10/94

50ms run Reset @ 1ms  
+dump 51 data  
Generic Local  
ATMAcatel622 Global  
Create EMC & MAC - fasttran  
Buffer Occupancy

FEX  
150ms  
L = 64 / ?  
G = 64

X Graph



7/10/94

50ms run reset @ 1ms

tdump... 51 data

Generic Local

ATMAlcatel 622 Global

EMC & MAC - fast run

Lifetime of Events

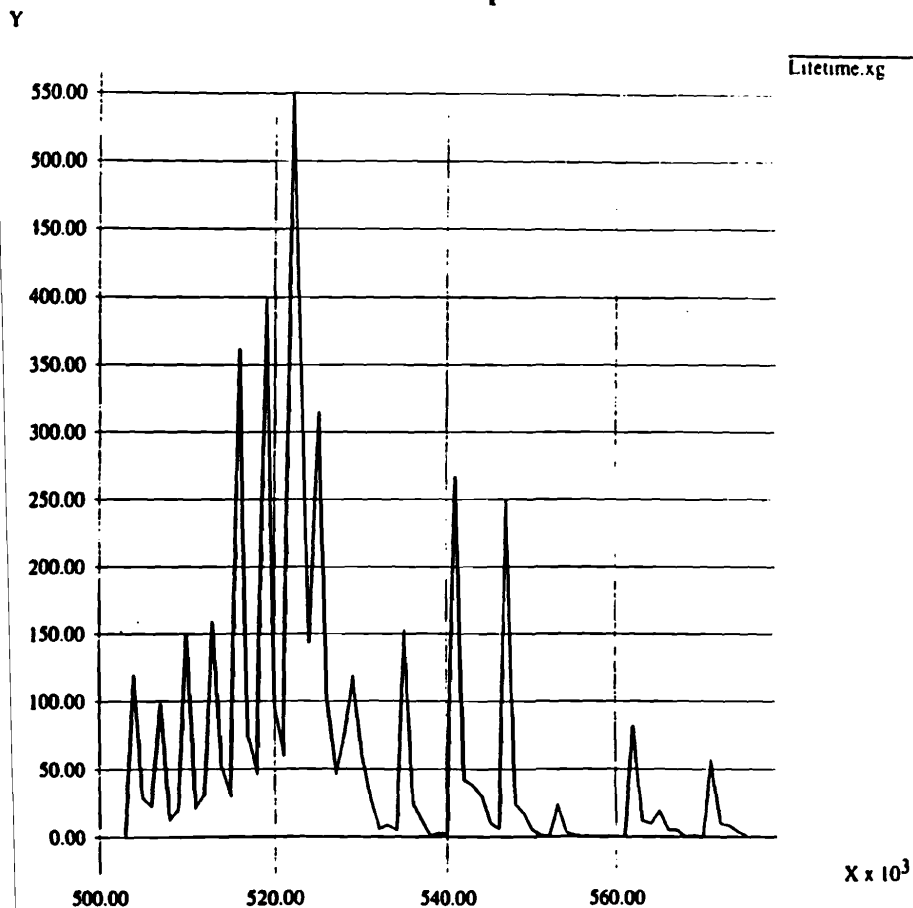
FEX

150μs

L = 64 / 17

γ = 64

X Graph



## ATLAS T2 WITH SCI

### Implementation of the ATLAS T2 Buffers, Local and Global Processors with SCI Switches

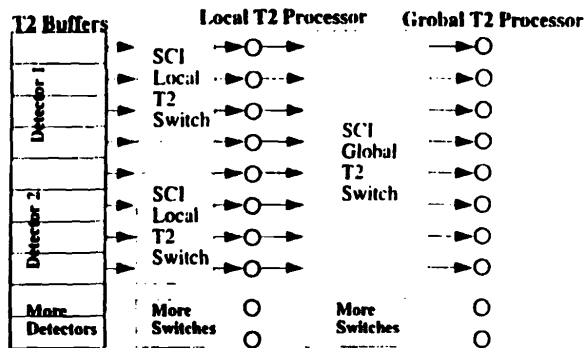


FIGURE 1. Generic SCI simulation model

# ATLAS T2 Architectures

## First Model of the ATLAS EMC T2 Implementation with SCI

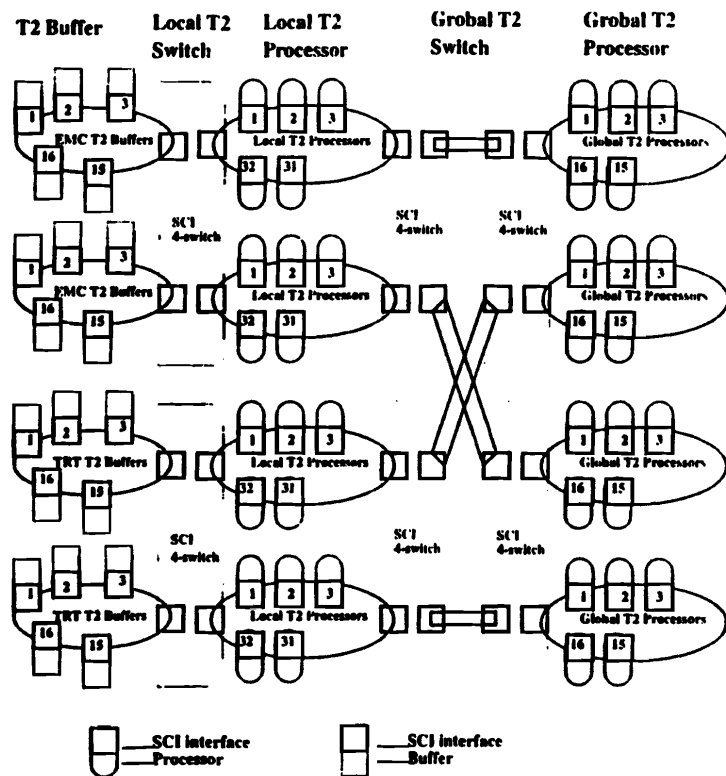


FIGURE 1. The first ATLAS simulation with full SCI setup

## SCI Publications

### 1. SCI Modelling

1.1 SINTEF/Dolphin/University of Oslo  
contact: Ernst.Kristiansen@si.sintef.no

1. John Bothner, Ernst Kristiansen, "Simulator report for SCI systems with HIC router", OMI-HIC ESPRIT Project T252, September 1994  
availability: OMI-HIC Project Confidential
2. John Bothner, T. Hulaas, "Topologies for SCI-based systems with up to a few hundred nodes", Thesis, University of Oslo, March 1993  
availability: ih.uio.no under pub/sci/Topologies\_Thesis.PS
3. J.W. Bothner and T.I. Hulaas, "Various interconnects for SCI-based systems", Proceedings of Open Bus Systems'91, Paris, pp. 197-202, 1991  
availability: fidibus.uio.no. under /incoming/SCI
4. E.H. Kristiansen, J.W. Bothner, T.I. Hulaas, E. Rongved, B. Staudl, "Simulations with SCI as a data carrier in data acquisition systems", 8th Conference on Real-Time Computer Applications in Nuclear, Particle and Plasma Physics, Vancouver, Canada, June, 1993
5. E.H. Kristiansen, J.W. Bothner and T.I. Hulaas, "Behavior of Scalable Coherent Interface in Larger Systems", Proceedings CAMAC-92, Warsaw, 29 September 1992
6. B. Wu, E. Kristiansen, "Some of the Perspectives on the Scalable Coherent Interface", to appear in the Int'l Conf. on Electronics and Information Technology, Beijing, 2-5 August, 1994  
availability: fidibus.uio.no. under /incoming/SCI
7. E. Kristiansen, B. Wu, J. Bothner, "High Speed Point-to-point Datalinks for use in and between Multiprocessor Systems", to appear in the Int'l Conf. on Electronics and Information Technology, Beijing, 2-5 August, 1994  
availability: fidibus.uio.no. under /incoming/SCI

### 1.2 University of Oslo (dept. of Informatics)

contact: gjessing@ifi.uio.no

8. T. H. Ramberg, "A SCI simulator that runs real MIPS executables", Presented at the 1994 European SCI Workshop, Univ. of Oslo, Sept. 1994  
availability: e-mail to gjessing@ifi.uio.no

9. H. Brhmi and B. Wu, "Initial Studies of SCI LAN topologies for local area clustering", First Int. Workshop on SCI based-high Performance low Cost Computing, Santa Clara Univ., Aug. 1994  
availability: SC1z/L (e-mail dbg@sunrise.scu.edu)

### 1.3 RD24 Collaboration (CERN, Univ. of Oslo Phys. Dept.)

contact: bin.wu@fys.uio.no, bogaerts@dxcern.cern.ch

10. Andre Bogaerts and Bin Wu, "SCI Simulations with SCILab", Two Pager, European SCI Workshop, Oslo, Sept. 1994  
availability: fidibus.uio.no. under /incoming/SCI

11. Andre Bogaerts and Bin Wu, "The SCILab Cook Book", CERN Internal Note, July, 1993

availability: sunsci.cern.ch under simulation/DOC

WWW: <http://www1.cern.ch/RD24>

12. *Andre Bogaerts, Roberto Divia, Hans Muller, J.F. Renard*. "SCI based Data Acquisition Architectures". IEEE Transactions on Nuclear Science, Vol. 39, No. 2, April 1992
13. *Bin Wu, Andre Bogaerts, Ernst Kristiansen, Hans Muller, Ernesto Perea, Bernhard Skaali*. "Applications of the Scalable Coherent Interface in Multistage Networks". IEEE TENCON'94, "Frontiers of Computer Technology", Aug. 22-26, 1994, Singapore  
availability: fidibus.uio.no, under /incoming/SCI
14. *B. Wu, A. Bogaerts, E. Kristiansen, B. Skaali*. "A Study of Routing Algorithms for SCI-based Multistage Networks". Proceedings of the First International Workshop on SCI-based High-Performance Low Cost Computing, also technical report UIO/PHYS/94-06, University of Oslo, Norway, March 1994  
availability: fidibus.uio.no, under /incoming/SCI
15. *Bin Wu and Andre Bogaerts*. "Several Details of SCI Switch Models". Univ. of Oslo/CERN Internal Report, Version 0.5, Nov 15, 1993  
availability: fidibus.uio.no, under /incoming/SCI
16. *Bin Wu, Andre Bogaerts, Roberto Divia, Ernst Kristiansen, Hans Muller, Bernhard Skaali*. "Constructing Large Scale SCI-based Processing Systems by Switch Elements". technical report UIO/PHYS/93-12, University of Oslo, Norway, May 1993  
availability: fidibus.uio.no, under /incoming/SCI
17. *B. Wu, A. Bogaerts, R. Divia, E. Kristiansen, H. Muller, E. Perea, B. Skaali*. "Distributed SCI-based Data Acquisition Systems constructed from SCI bridges and SCI switches". The 10th Int'l Symp. on Problems of Modular Information Systems and Networks, St. Petersburg, Russia, Sept. 13-18, 1993, also as technical report UIO/PHYS/94-02, University of Oslo, Norway, Jan. 1994  
availability: fidibus.uio.no, under /incoming/SCI
18. *B. Wu*. "Initialization of models and routing tables in an SCI system". Internal Note, March 13, 1994  
availability: fidibus.uio.no, under /incoming/SCI
19. *B. Wu and A. Bogaerts*. "A summary of simulation results for SCI switches and SCI systems". Internal Note, March 13, 1994  
availability: TOPSCI Project Confidential
20. *B. Wu*. "Routing for Link Controller". TOPSCI Internal Note, March 21, 1994  
availability: TOPSCI Project Confidential
21. *B. Wu*. "Simulations of SCI with Link Controller and Blink". TOPSCI Internal Note, Oct., 1994  
availability: TOPSCI Project Confidential
22. *B. Wu, A. Bogaerts, A. Bogacris*. "SCILab - A Simulation Environment for the Scalable Coherent Interface". to appeared in Proceedings MASCOTS'95, Durham, NC, 16-18 Jan. 1995  
availability: fidibus.uio.no, under /incoming/SCI

#### 1.4 IEEE

contact: [dvj@apple.com](mailto:dvj@apple.com)

23. "IEEE Standard for SCI, IEEE New York, August 1993. A diskette with the C-code is included (but contact [dvj@apple.com](mailto:dvj@apple.com) for latest updates)

#### 1.5 University of Edinburgh

contact: [rh@dcs.edinburgh.ac.uk](mailto:rh@dcs.edinburgh.ac.uk)

24. *R. Hessel and N. Topham*. "The Performance of SCI Memory Hierarchies". Proc. of the Int. Workshop on Support for Large Scale Shared memory Architectures, Cancun, Mexico, 1994
25. *R. Hessel*. "A Quantitative Performance Evaluation of SCI Memory Hierarchies", PhD Thesis, Univ. of Edinburgh, 1994

#### 1.6 University of Wisconsin

contact: [sls@romulus.cray.com](mailto:sls@romulus.cray.com)

26. *R.E. Johnson and J.R. Goodman*. "Synthesizing General Topologies from Rings". Proceedings of the ICPP, August, 1992

27. *S. Scott, J.R. Goodman and M. Vernon*. "Performance of the SCI Ring". Proceedings of the 19th Int. Symposium on Computer Architecture, May 1992

28. *Ross Johnson*. "Extending the Scalable Coherent Interface for Large-scale Shared-memory Multiprocessors". PhD thesis, Univ. of Wisconsin-Madison, USA, 1993

#### 1.7 University of California San Diego

contact: [rfellman@ucsd.edu](mailto:rfellman@ucsd.edu)

29. *D. Pucker and R. Fellman*. "An SCI Simulator with Traffic Flow Animation". First Int. Workshop on SCI based High Performance low Cost Computing, Santa Clara Univ., Aug. 1994

availability: SCI71, (e-mail [dhg@sunrise.scu.edu](mailto:dhg@sunrise.scu.edu))

#### 1.8 University of Southern California Los Angeles

contact: [barroso@paris.usc.edu](mailto:barroso@paris.usc.edu)

30. *L. Barroso and M. Dubois*. "Performance Evaluation of the Slotted Ring Multiprocessor". To appear in the IEEE Transactions on Computers

availability: usc.edu under pub/CENG

## 2. Other SCI Related Publications

### 2.1 RD24 Collaboration, CERN

contact: [bogaerts@dxcern.cern.ch](mailto:bogaerts@dxcern.cern.ch)

[Hans@sunshine.cern.ch](mailto:Hans@sunshine.cern.ch)

31. *RD24 Collaboration*. "RD24 Status Report, Application of the Scalable Coherent Interface to Data Acquisition at LHC". CERN/DRDC 94-23, Status Report May 1994

availability: [sunsci.cern.ch](http://sunsci.cern.ch) under sci/RD24\_Info/

32. *RD24 Collaboration*. "RD24 Status Report, Application of the Scalable Coherent Interface to Data Acquisition at LHC". CERN/DRDC 93-20, 5 May 1993

availability: [sunsci.cern.ch](http://sunsci.cern.ch) under sci/RD24\_Info/

33. *H. Müller et al.*. "First Experience with the Scalable Coherent Interface", RT93, Vancouver, Canada, June 1993, IEEE Trans, Vol 41, No 1, Feb. 1994, or preprint CERN/ECP 93-15

availability: [sunsci.cern.ch](http://sunsci.cern.ch) under sci/Talks+Papers

34. *R. Keyser and G. Mugnai*. "SCI for Accelerator Controls: Status Report". CERN-SI/Note 94-32(CO)

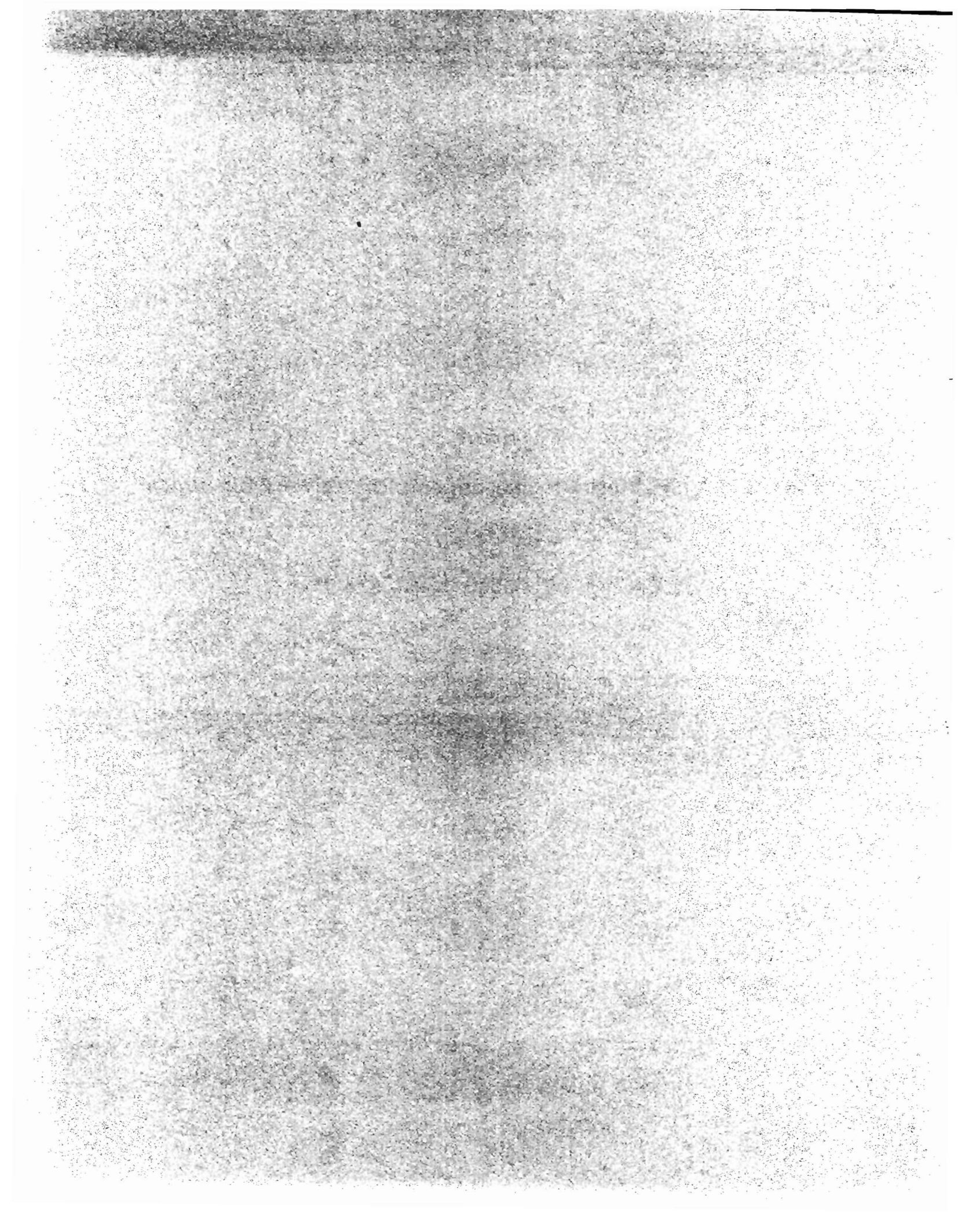
availability: e-mail to [keyser@dxcern.cern.ch](mailto:keyser@dxcern.cern.ch)

**S6-4**

**"Review of ATM, Fibre Channel and Conical Network Simulation  
Results"**

**(Irakli Mandjavidze - CERN/Saclay)**

Commercial and Non-Commercial switching network architectures have been proposed for applications in High Energy Physics data acquisition systems. Contention in any types of switching fabrics are inevitable under the traffic patterns generated by HEP experiments, which may result in data loss or long event-building latencies. Traffic Shaping versus Flow Control contention resolution techniques have been studied by means of simulation. Results of these generic studies are presented.



# Review of ATM, Fibre Channel and Conical Network Simulations (will not be presented)

## Congestion Control Techniques Flow Control and Traffic Shaping

I. Mandjavidze  
RD31, CERN/ECP  
mandjavi@sunvisi.cern.ch

Review of ATM, Fiber Channel and Conical Network Simulations

I. Mandjavidze, CERN

International Data Acquisition Conference on Event building and Data Readout

FNAL, 26-28 October, 1994

### OUTLINE

- 1) Candidates for an event builder network
- 2) Pathological traffic pattern
- 3) Congestion control techniques
  - \* Flow Control
  - \* Traffic Shaping
- 4) Generic event builder model
- 5) Simulation results
  - \* Flow Control
  - \* Traffic Shaping
- 6) Discussion

Review of ATM, Fiber Channel and Conical Network Simulations

I. Mandjavidze, CERN



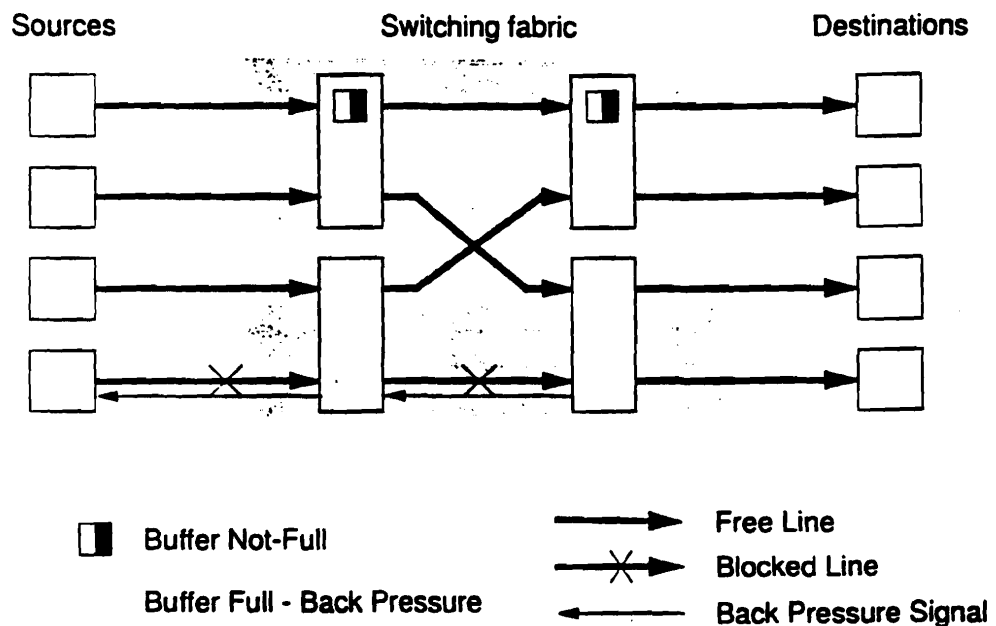
## Congestion Control Techniques

### Two types of fabric architectures

- \* *Link Level Hardware Flow Control*
  - Some ATM (LAN)
  - Custom made
  - Fibre Channel
  
- \* *No Link Level Hardware Flow Control*
  - Some ATM (Telecom)

## Congestion Control

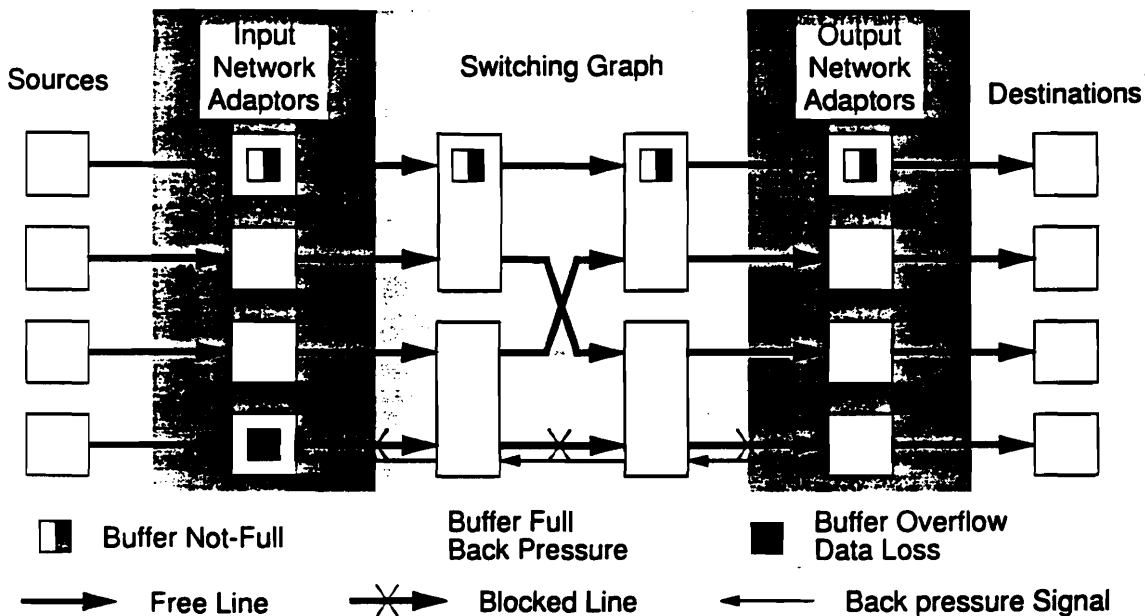
### Link level hardware flow control - Fibre Channel, Custom made



### Congestion Control

#### Internal Link Level Hardware Flow Control - ATM (LAN)

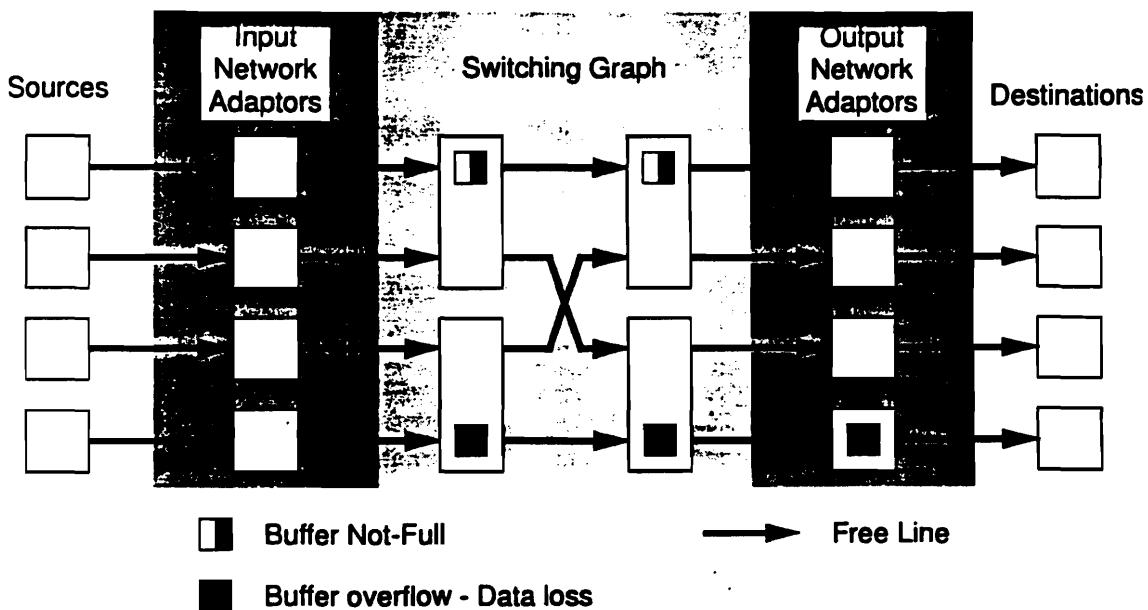
##### Generic ATM Switching Fabric



### Congestion Control

#### No link level hardware flow control - ATM (Telecom)

##### Generic ATM Switching Fabric

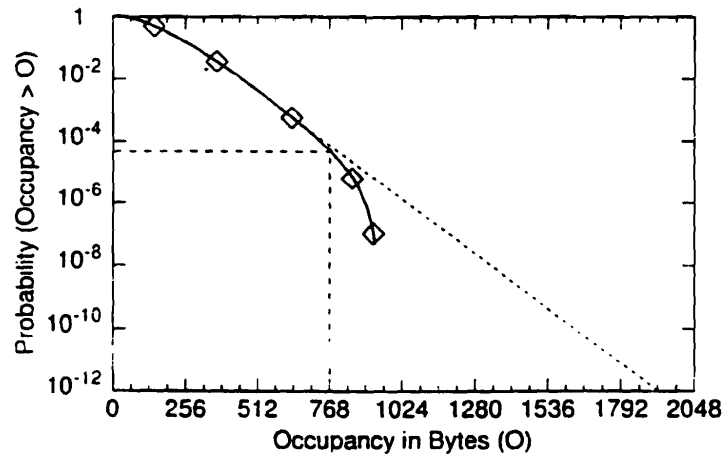


## Cell Loss Probability

*Alcatel switching fabric*  
*Switching element buffer size*  
 Random (telecom) traffic

256x256 I/O @ 155MBit/s  
 2 KByte  
 80% Load

Buffer occupancy of a switching element



Review of ATM, Fiber Channel and Conical Network Simulations

I. Mandjavidze, CERN

International Data Acquisition Conference on Event building and Data Readout

FNAL, 26-28 October, 1994

## Congestion Control

### Traffic Shaping

**Basic principles of traffic shaping are:**

- \* Rate Control - sum over all input bandwidths towards an output port does not exceed the available bandwidth of the output port - **Easy to Implement**
- \* Break the *instantaneous* time correlation of cell streams traveling to an output port - **Not Trivial**

**Studied Traffic Shaping Schemes:**

- \* Cell Based Barrel Shifter
- \* True Barrel Shifter
- \* Randomizer

Review of ATM, Fiber Channel and Conical Network Simulations

I. Mandjavidze, CERN

## A Generic Event-Builder Modeling Tool

### Switching element technology

**Size:** 2x2, 4x4 / 4x2, 8x4  
**Operation mode:**  
 \* Flow control - no buffer sharing (resembles AT&T/Phoenix)  
 \* Flow control - buffer sharing (resembles IBM/PRIZMA)  
 \* No flow control - buffer sharing (resembles ALCATEL/ISE)  
**Buffer size:** variable  
**Variable link speeds:** 160, 320, 640, 1280, 2560 Mbit/s  
**Transmission data unit:** A 64 Byte long cell carries 56 byte user payload

### Network Technology

**Topology:** Banyan  
**Variable size**  
 \* Square: from 8x8 to 1024x1024  
 \* Conic: from 16x4 to 8192x1024

### Traffic shaping modes:

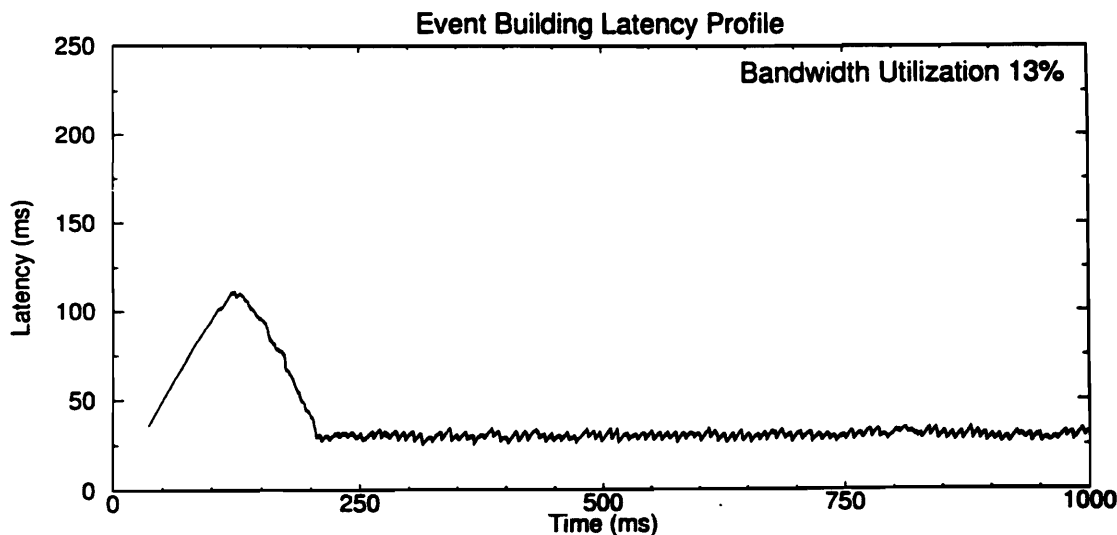
- \* No traffic shaping
- \* Cell based barrel shifter
- \* True barrel shifter
- \* Randomizer

### Simulation Language: $\mu$ C++

Simulation results have been compared with models written in C++ and Modsim

## Simulation Results - Flow Control

<u>Configuration</u>		<u>Input Conditions</u>	
<i>Event Builder</i>	1024x1024 @ 640Mbit/s	<i>Event Size</i>	1Mbyte
<i>Switching Element</i>	4x4 with 16KByte Memory	<i>Trigger Rate</i>	7.5KHz
<i>Back Pressure</i>	up to Sources		



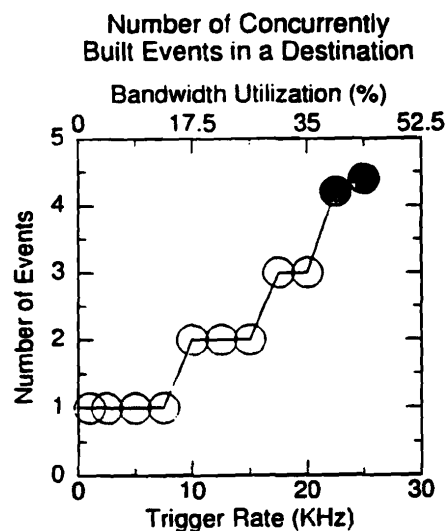
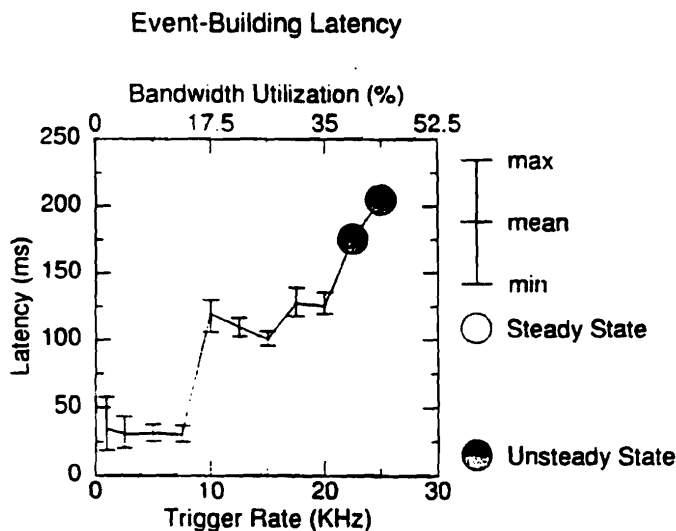
### Load Dependency

#### Configuration

Event Builder 1024x1024 @ 640Mbit/s  
 Switching Element 4x4 with 16KByte Memory  
 Back Pressure up to Sources

#### Input Conditions

Event Size 1Mbyte



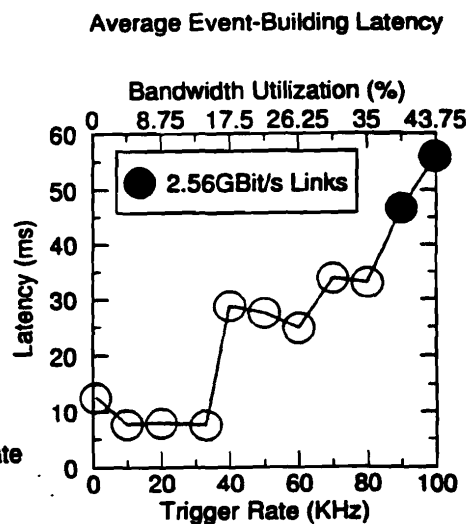
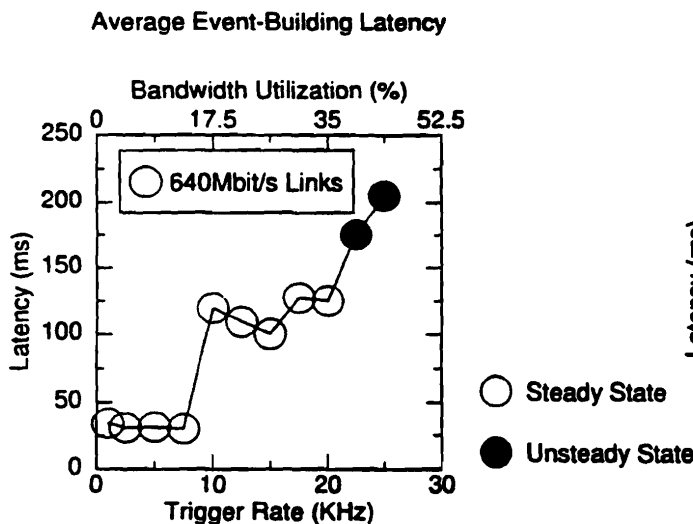
### Load Dependency

#### Configuration

Event builder 1024x1024  
 Switching element 4x4 with 16KByte Memory  
 Back pressure up to Sources

#### Input Conditions

Event size 1Mbyte



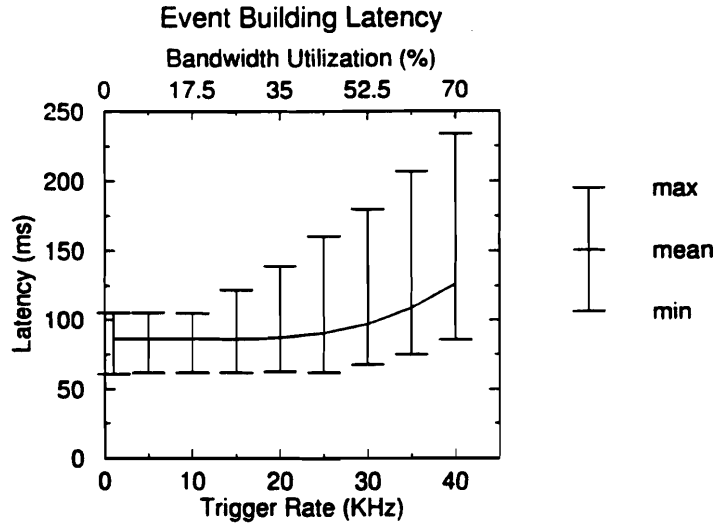
### Simulation Results - Traffic Shaping

#### Configuration

Event builder 1024x1024 @ 640 Mbit/s  
 Switching element 4x4 with 16KByte memory  
 Traffic shaping Randomizer

#### Input Conditions

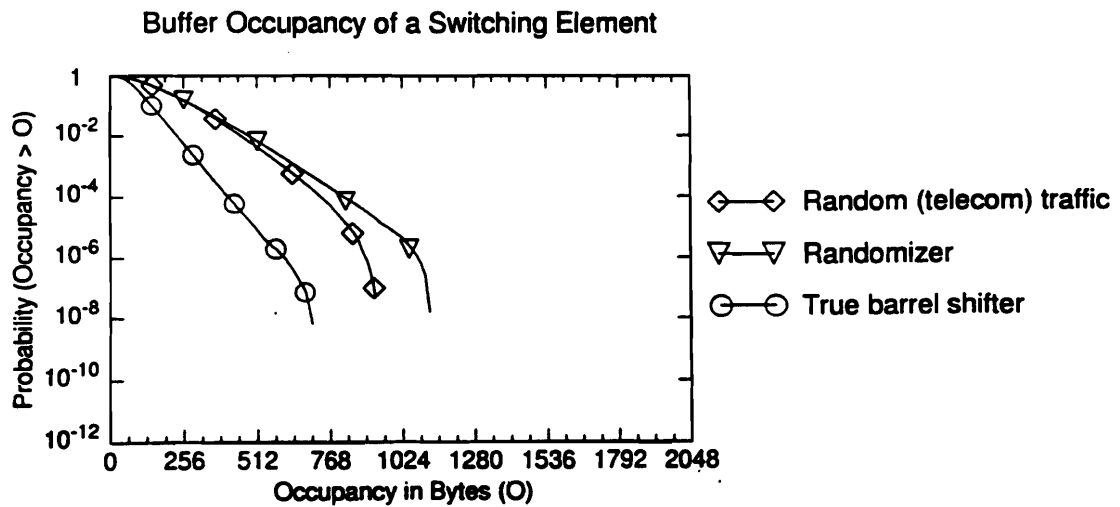
Event size 1 Mbyte



### Cell Loss Probability

Alcatel switching fabric  
 Switching element buffer size  
 Bandwidth utilization

256x256 I/O @ 155Mbit/s  
 2KByte  
 80%



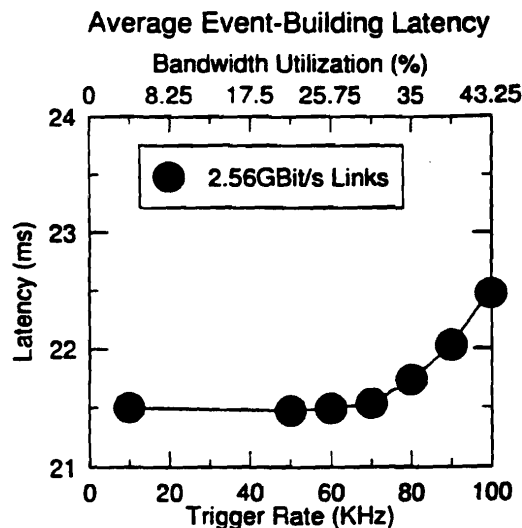
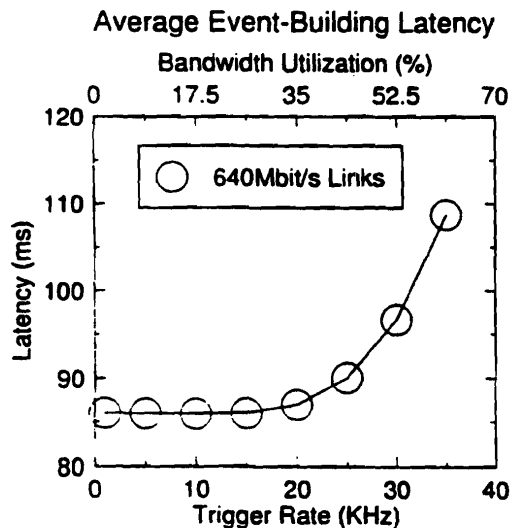
### Load Dependency

#### Configuration

Event builder 1024x1024  
 Switching element 4x4 with 16KByte Memory  
 Traffic shaping Randomizer

#### Input Conditions

Event size 1Mbyte



### Scalability

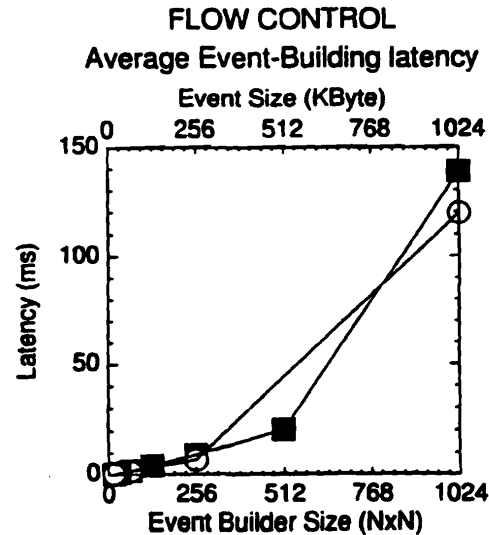
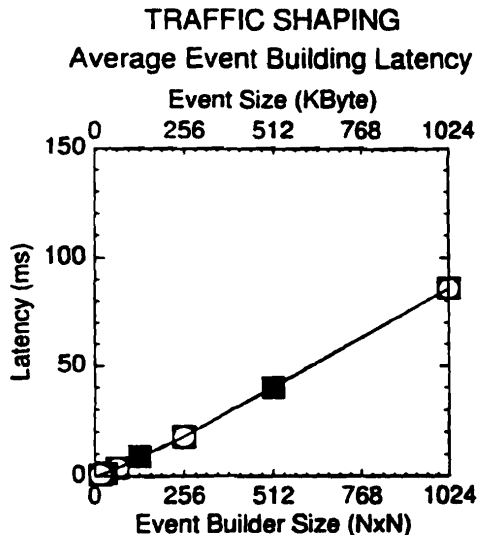
#### Configuration

I/O Rate 640Mbit/s  
 Switching Element 16KByte Memory

#### Input Conditions

Event Fragment Size 1KByte  
 Trigger Rate 10KHz

Bandwidth Utilization 17%



■ Network built from 2x2 switching elements  
 ○ Network built from 4x4 switching elements

## Summary

### 1) The behaviour of systems with source traffic shaping technique is predictable

- \* Good scaling
- \* High throughput
- \* But well suitable only for square and "Near-to-Square" systems

### 2) The behaviour of systems with flow control is difficult to understand and requires more study

- \* For Small Systems (up to 256x256) shorter Event-Building Latencies can be achieved (important for L2)
- \* Performance depends on the actual architecture of an event-building cross-connect. Probably, better results can be achieved with more advanced architectures
- \* But sensitive to traffic fluctuations and destination assignment schemes

### 3) For LARGE event builder systems non of the studied network architectures are "Buy-and-Use" solutions:

- \* ATM Fabrics - In order to achieve acceptable data loss probabilities will require traffic shaping
- \* Fibre Channel Fabrics (*prediction*) - Cascading several nodes in order to form a large cross-connect may significantly decrease throughput of the event builder system, if traffic shaping will not be used

Review of ATM, Fiber Channel and Conical Network Simulations

I. Mandjevidze, CERN

International Data Acquisition Conference on Event building and Data Readout

FNAL, 26-28 October, 1994

## Future Work

### 1) Behavior of the event builder systems with flow control should be understood better

### 2) Fibre Channel fabrics as an event builder cross-connect have to be studied:

- \* Scalability of the system when several Fibre Channel nodes are forming a large fabric
- \* Which service class has to be used?
- \* Is traffic shaping necessary?

### 3) Developments in technologies have to be followed up (ATM, Fibre Channel)

### 4) From generic event builder towards specific DAQ architecture

- \* Realistic input parameters

Review of ATM, Fiber Channel and Conical Network Simulations

I. Mandjevidze, CERN





**S7-1**

**“Software Issues When Implementing An ATM Network”**

**(Henry Dardy - Naval Research Laboratory)**



# Prototyping ATM Functionality

..... to Enable the Global Grid



*Dr. Henry D. Dardy*  
Center for Computational Science  
INFORMATION TECHNOLOGY DIVISION  
Naval Research Laboratory  
Washington, D.C. 20375-5000

## GLOBAL GRID: What is it ?

- A concept for aggressive worldwide distribution and processing of data that employs distributed resources . . .

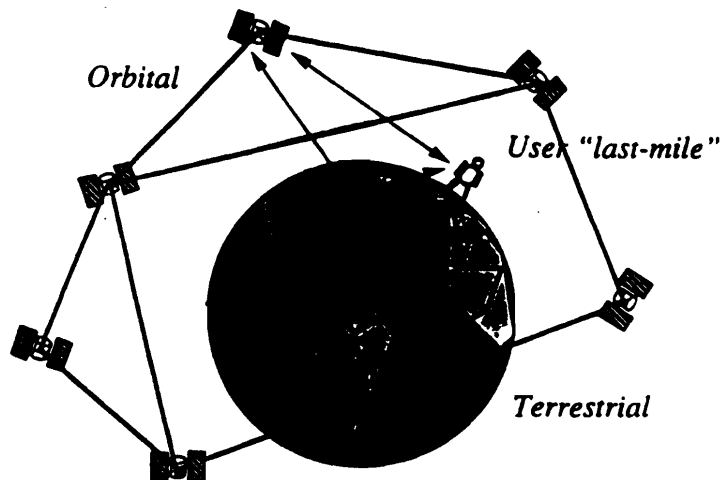
- *Data Bases*
- *High Performance Computers*
- *High Performance Workstations*
- *Multimedia Communications*

. . . Connected by commercial and government communications links . . . .

- *Backbone of high capacity fiber optic trunks*
- *Supplemented and integrated with COMSATS, RPV's and other communication links*
- *All use commercial telecommunications standards based on Broadband-ISDN technology (i.e., SONET and ATM)*



# One GLOBAL GRID Three Kinds of Interconnects



- *Grid to Grid (uplink/downlink)*
- *User to Terrestrial Grid ("Last Mile Problem")*
- *User to Orbital Grid -- explosion of activity*

## *GLOBAL GRID: What is being done ?*

- *Coordinated planning and production by government*
  - *Input to national level deliberations on telecommunications policy*
- *Critical experiments and demonstrations*
  - *Very high data rate switching and control*
  - *Encryption for packet communications*



## NRL CCS's View into the 21st Century . . .

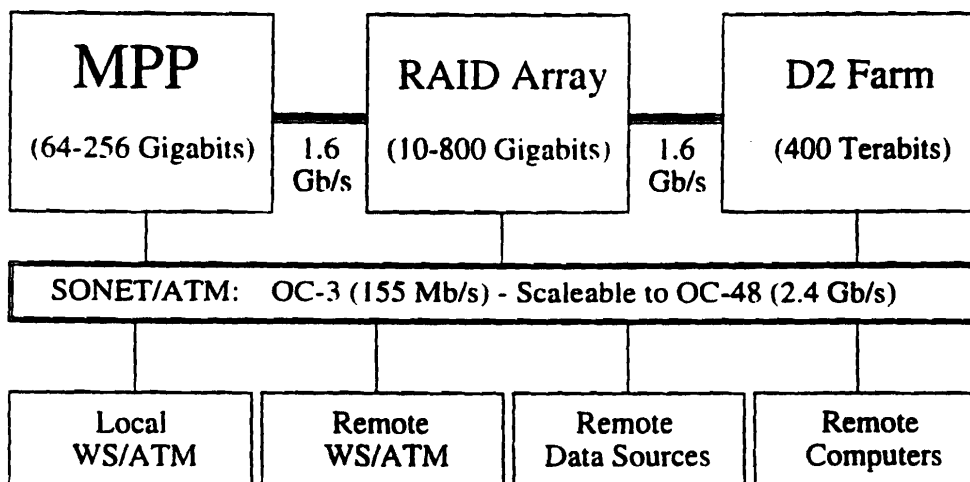
Future applications will be built with a transparent, scalable information framework that encompasses

- LANGUAGES . . . *optimized parallel Fortran and C; applications specific pre-compilers*
- PROGRAMMING TOOLS . . . *parallel debuggers, performance monitors*
- FILE SYSTEMS/ARCHIVES . . . *distributed data repositories that are remotely accessible*
- NETWORKS . . . *gigabit and beyond. BISDN, and wide-area networks*
- IMAGERY . . . *2D/3D lifelike video, visualization and virtual reality*



*. . . we need to evolve an information infrastructure to remain competitive; it must be capable of supporting our needs and NOT be overwhelmed by the rapid growth of available information.*

## Scaleable Architecture Test Configuration . . .



- Model for:**
- DIA - MLS - Global Intelligence
  - Labs - Thrusts - CRDA's (*Environmental, Reference IR, etc.*)
  - DISA - Network - Command & Control
  - NSA - Encrypted networks
  - Intelligence
  - Synthetic Environment
  - Simulation (*SIMNET*)

## **Building the Defense Information Infrastructure**

Prototype a "network-centric" Information Infrastructure for transparent, ubiquitous access to "globally remote" resources as if "local"

- *Wide area communications and computing via a **Global Grid**; SVC signalling*
- *Based on client/server and peer-to-peer paradigms*
- *Policy-based dynamic routing with authentication*
- *Information caching and hiding inherent in system*
- *Wide-area, on-time information retrieval*

## ***PRAGMATIC VIEW (at the fringe)***

- **Based on ATM**
- **Based on AFS**
- **Based on transparency of files**
- **Wide area**



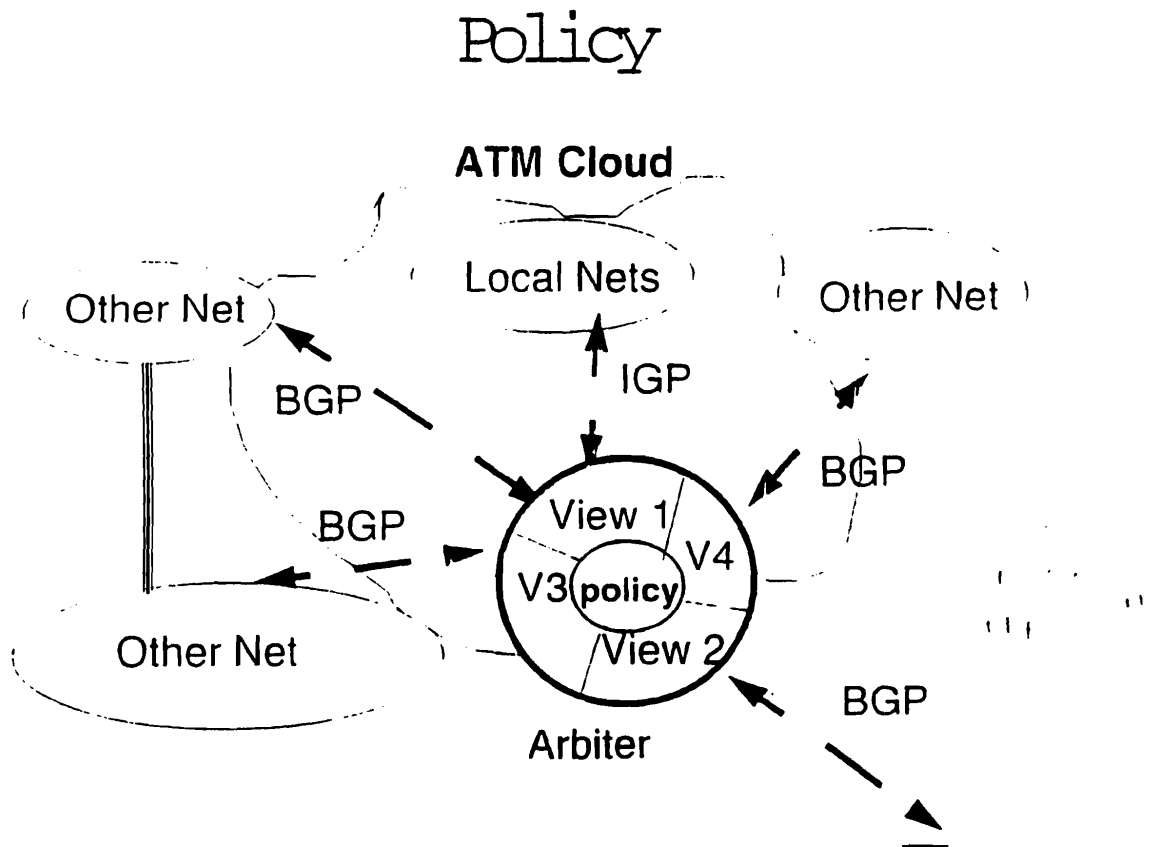
## ↑ DRIVING APPLICATIONS

### *The Information Infrastructure Services*

- Security and authentication
- Software distribution, recovery and backup
- Intelligent agents (knobots), nameservices
- License services, certificates, timestamps
- Caching / replication / cache & carry
- Databases / archives / voice-video repositories

### *B-ISDN Wide Area Coms with ATM*

- SVC signaling
- QOS, flow and congestion control
- Bandwidth Management
- Dynamic routing, addressing, state





**VINCE**

**Protocol**

**Development**

**Environment**

- An environment for wide area network experimentation and protocol development
- Early development and adoption of ATM standards
- Integration mechanism for differing protocols, valuable as an integration platform for disparate protocol families
- Tool for testbed experimentation
- Mechanism for experimentation in host network architectures, protocol engine on hosts and switches
- Prototype enabling services, encryption, interoperability
- Simple and fast protocol processing abstractions
- Freely distributable code - *Mach-like license*
  - *No encumberances to public use*



**SCHEDULER**

Core Services:

route, address, connection, path, signaling, switch, port

<i>Signaling protocols</i>	<i>Management protocols</i>	<i>Fabric controllers</i>	<i>Others</i>
SPANS	SKIP	FORE ASX	Routing
Q.93B	SNMPv1	simulation	Broadcast
- UNI V3.0	- ILM1	host	Multicast
PNNI proto	- AtomMIB	credit switch	QOS
			Scheduling
			API(sockets)
			User Documents

**Software Architecture:**

Vendor Independent Network Control Entity (VINCE)

*...an integration framework for switching hardware, host interfaces, experimental and standards track protocols*  
*... includes the UC Berkeley Tcl and Tk toolkit for development and virtualization of graphical user interfaces (GUI's)*



ILMI	Q93B	SPANS	ASX	host	sroute
SNMP		ATM Core			
Protocol		Skip	Address		
ELIB					

timer events / memory layout

## Elib



- Developed a strong portability library which requires only timing events and page-level allocation from the host system
- Enhanced memory management system to integrate buffer pool facility, allowing fast interrupt level processing of incoming packets
- Includes command dispatcher closely integrated into C calling semantics, string handling, and scheduling facilities

## PROTOCOL LIBRARY



- Lightweight, no-copy STREAMS like protocol stack system
- In combination with Elib, strongly isolates the task of protocol development from the operational environment
- Allows protocol processing stacks to span multiple address and scheduling spaces through the use of bridges
- *allocate\_down* feature allows bridges and hardware interfaces to transparently enforce alignment and padding requirements

**RECENT**

**and**

**ONGOING**

**DEVELOPMENT**

**WORK**

**for**

**VINCE 0.8**

- Enhanced kernel scheduling system to allow protocols running under VINCE to remain oblivious to kernel scheduling, priority, and memory management

- Ported entire environment to run on standalone i960; developing SBA-200 and PTAI2 interfaces

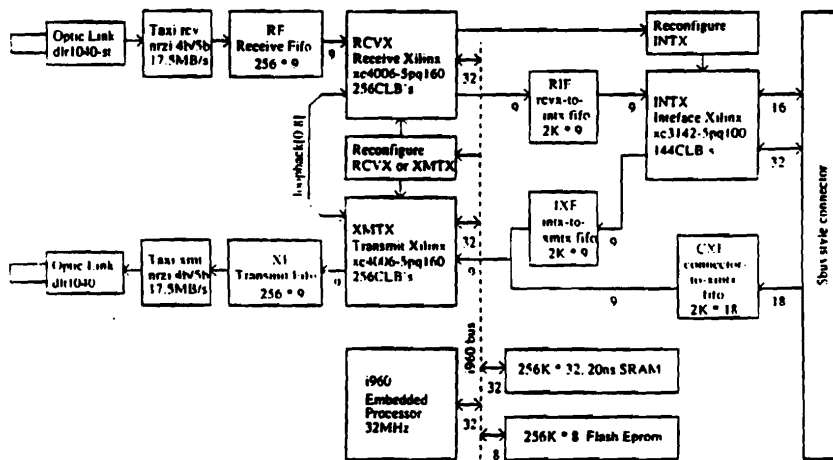
- Providing fast scheduled zero-copy interface across user-kernel boundary

- Changed *ATMCORE* to allow for multipoint connections and simplified call state processing

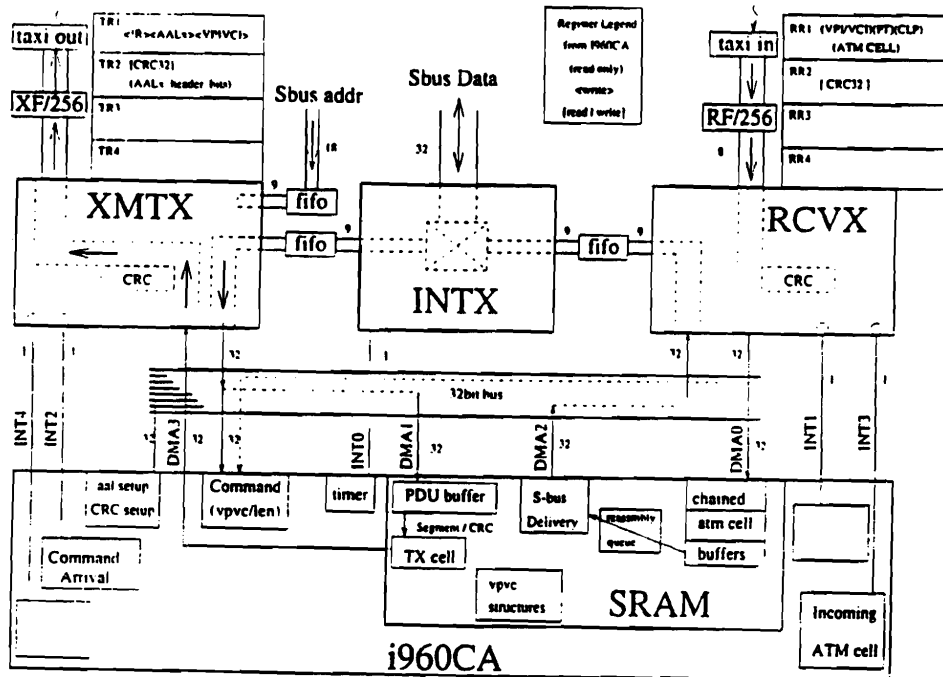
- Fully integrated SNMPv1 agent for use by the ILM1 and AtomMIB



*PTAI2 - Programmable TAXI-ATM Interface, V2*  
**COMMON MODULE CONCEPT**



PTAI2 - Programmable TAXI-ATM Interface, V2  
 COMMON MODULE CONCEPT, SBus Implementation



*The true ability to login to your home directory from anywhere - with all the benefits of AFS regardless of whether you are connected over SLIP or ATM !*

**AFS**

- Uniform access methods for files as a replacement for more conventional methods such as FTP or Gopher
- Uniform security over file accesses
- Take full advantage of technology advancements from high speed BISDN ATM networks, wireless and leading edge mass storage technology



**Andrew  
File  
System  
Integration  
with  
ATM**

- Work on path MTU discovery
- Caching algorithms for mass storage integration
- Profiling the cache manager shows 50% of time spent in local processing rather than data transfer
- Work done to understand modifications required to support multi-homing
- Plan evolution of intermediate servers and new client caching strategies



**Multi-homed  
Servers  
and  
Clients**

- Use name resolution instead of IP addresses
- Profile cache manager to find and fix bottlenecks
  - Integrate mass storage system with AFS directly for access to very large files (i.e., terabyte data stores)
- Resolve issues that will come up for database servers with the use of multi-homed machines
- If multiple paths exist between client and server, bias the code to use the path with best performance
- Prepare for the IPv6 address format changes
- Add options to bypass client cache if necessary
  - prevents read once only apps from overwriting cache
  - improves performance by avoiding cache data structure manipulation



**FUTURE**

**Development**

**Work**

- Examine kernel interrupt processing and scheduling issues; provide improvements
- Further integration of experimental IP protocols
- Provide generic BSD socket interface matching specification of Craig Partridge
- Provide early implementation and experience for the ATM Forum PNNI WG
- Provide extensive architecture and interface documentation for user community
- Deploy in production-like setting
  - *Washington Area ATDNET*
- Port to other switch fabrics (i.e., GTE, etc.)



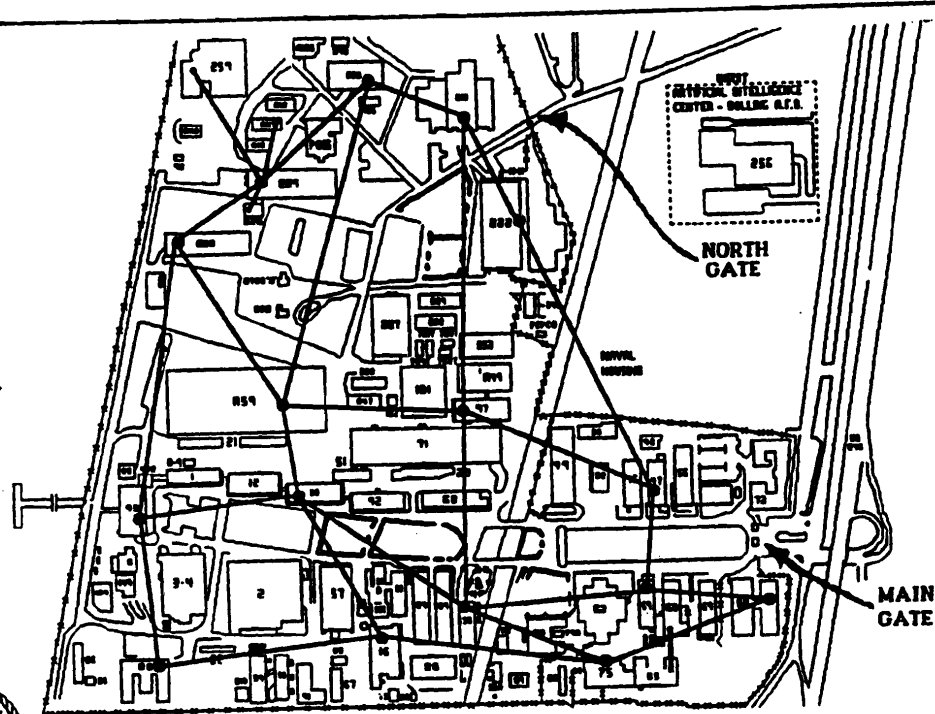
NRL

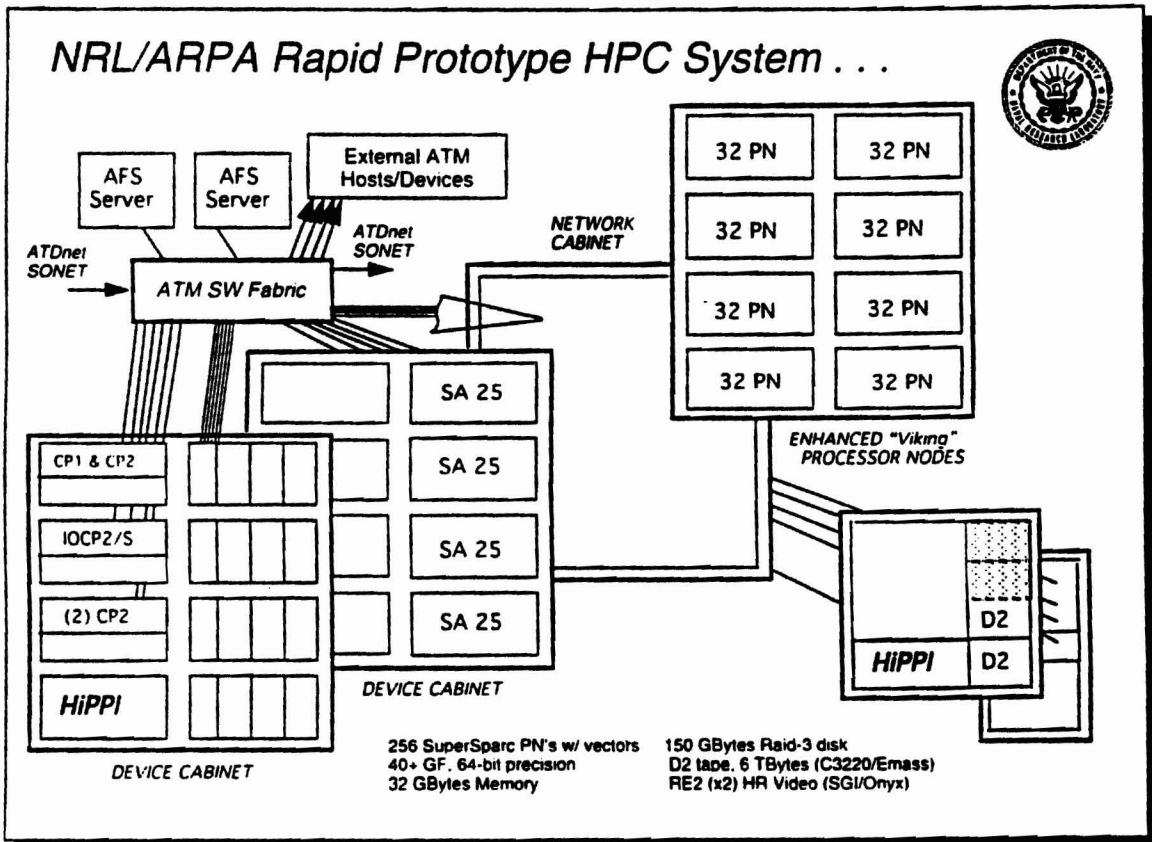
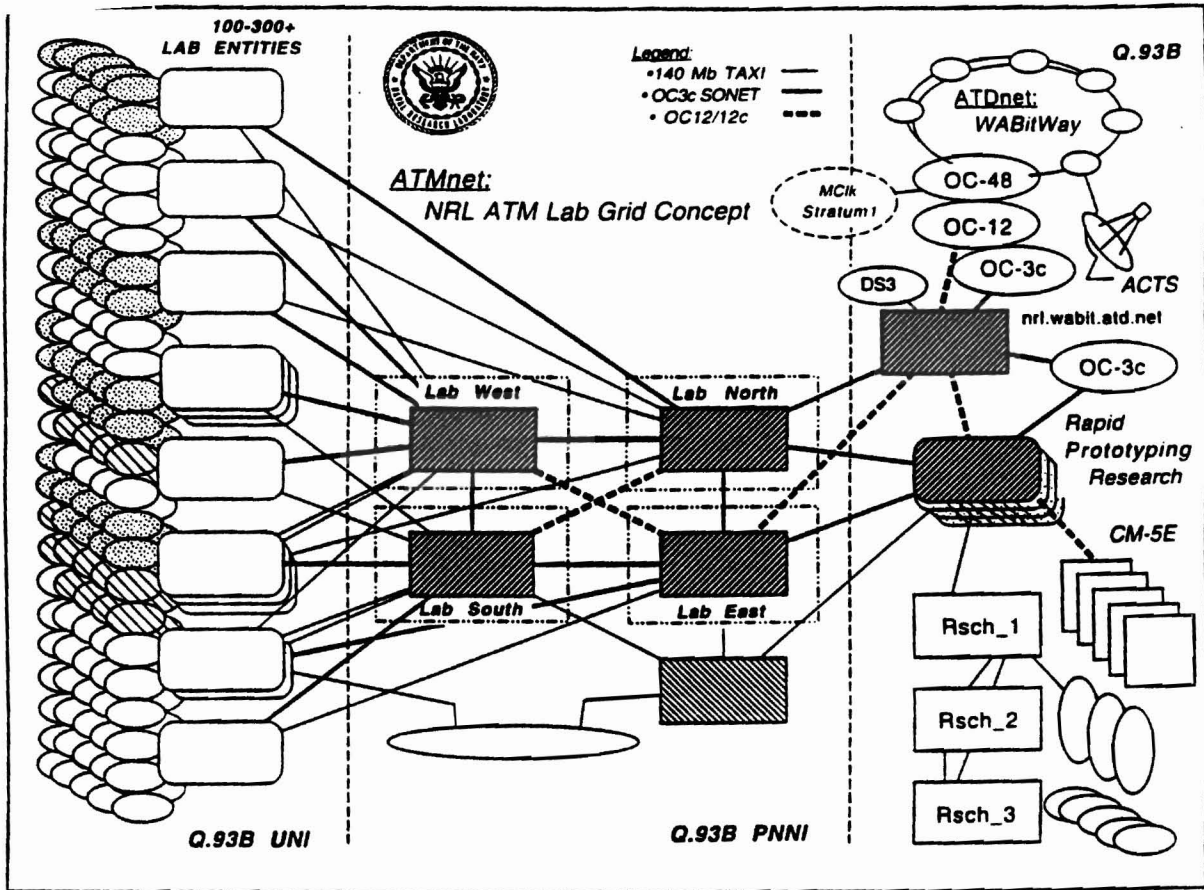
Campus

Fiber

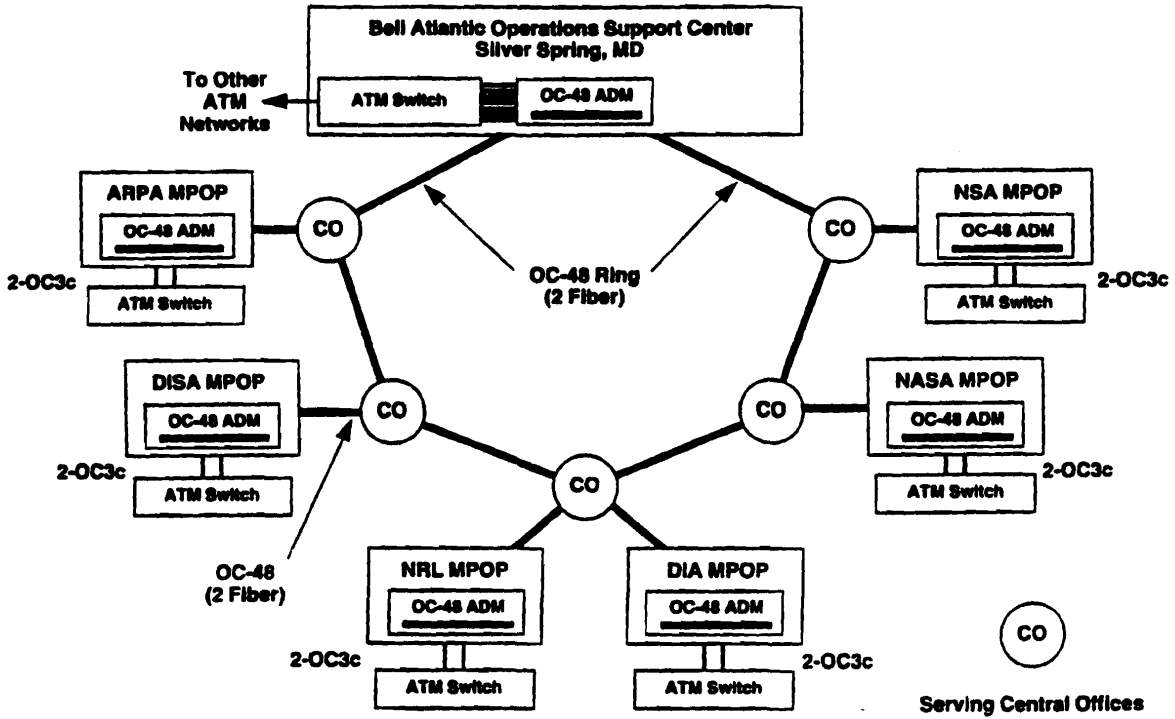
Grid

Topology





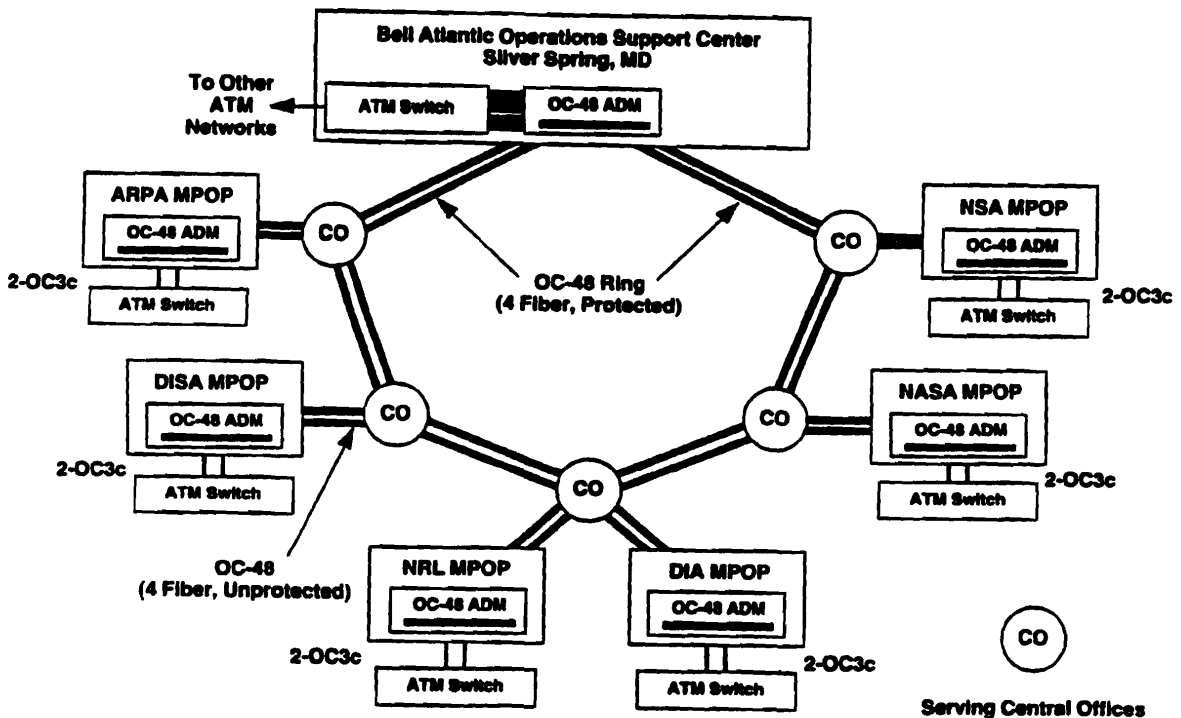
# ATDnet Initial Configuration (May 1994)



die:atdnet\_p.ppt

6/ 9/94

# ATDnet Final Configuration (June 95)

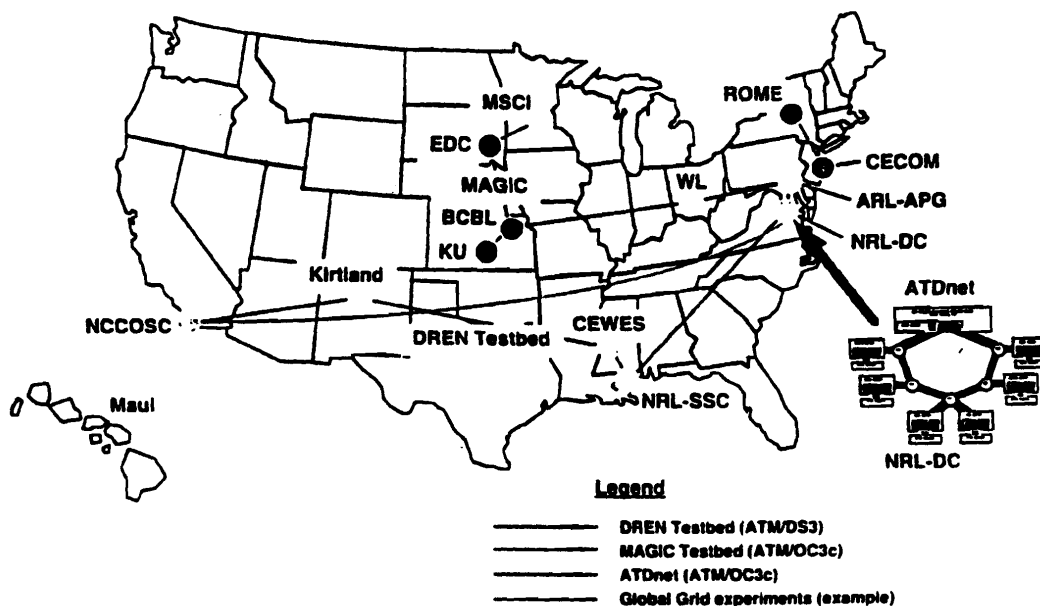


die:atdnet\_p.ppt

6/ 9/94



## DREN Testbed - Phase 1



6/28/94

### DRIVING APPLICATIONS...

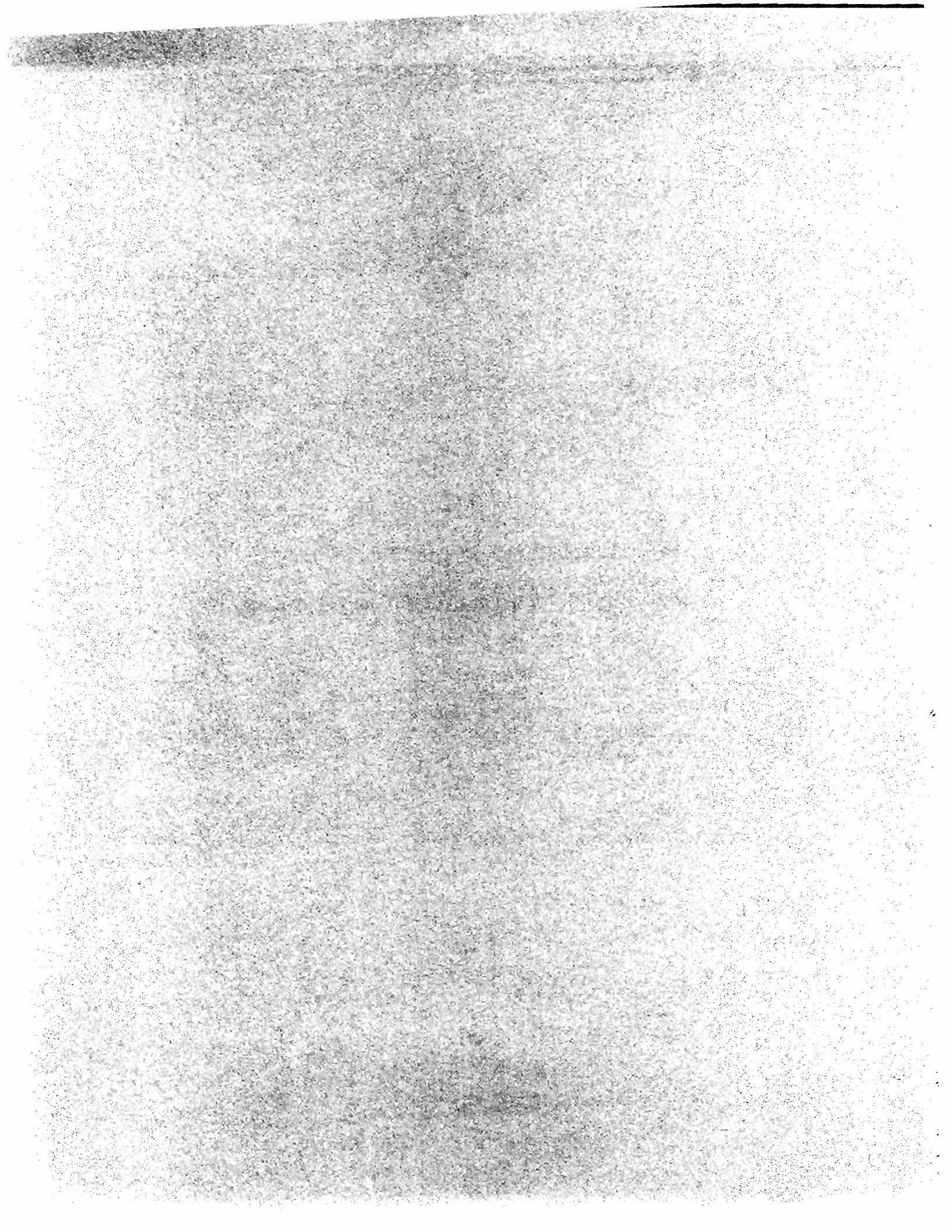
- *Distributed, wide area collaborative computing with OnTime delivery of information*
- *Digital video with dynamic resolution control and high resolution imagery (HDTV)*
- *Information repositories, archives, and multimedia databases*
- *Real time sensing, instrumentation: globally synchronized time and spatial (GPS) resolution*
- *Collaboration technology: virtualization, wide area simulation, mission rehearsal, training*
- *Virtual corporations, electronic commerce, and health care*
- *Entertainment*

**S7-2**

**"Data Acquisition Software Design Issues"**

**(Bob Russell - University of New Hampshire)**

A perspective on the role of open software standards such as POSIX on the design and implementation of DAQ systems. Consideration of how these standards affect the portability, interoperability, conformance and performance of software at all levels of a complex system.



## Data Acquisition Software Design Issues

Robert D. Russell  
Computer Science Department  
University of New Hampshire  
Durham, NH 03824 USA  
rdr@unh.edu

### Abstract

A perspective on the role of open software standards, such as POSIX, on the design and implementation of DAQ systems. Consideration of how these standards effect the portability, interoperability, conformance and performance of software at all levels of a complex system

### 1. Introduction

The DAQ systems currently under consideration for HEP are large, complex systems that must evolve and change over a period of many years. They are designed and built by large collaborations of groups geographically scattered all over the world, each using different hardware and software platforms to develop small pieces of the total system that must integrate and perform flawlessly when the experiment finally gets beam time. During its development, the technology, experimental layout, and requirements of the system will all change unpredictably and perhaps drastically. The question, therefore, is how can it all possibly be accomplished? Although there is no simple answer, we can look at some of the issues involved and try to get an overall picture of what design strategies might be followed.

### 2. Technological Developments

We begin by noting that large data acquisition collaborations share many of the technical and managerial problems of any large software system, and that a lot can be learned from how these problems are being addressed by the software industry as a whole. This is not to degrade the special problems related to the realtime nature of a DAQ system, but merely to put them into a more global context, as indeed the producers of most commercial realtime systems are doing. Table 1 lists some of the influences of the last few years that have had significant impact on the design and implementation of modern DAQ systems.

Underlying many of these factors is the overwhelming influence of standards and standards bodies. Almost all software systems on the market today claim to adhere to one or more standards, at least in part. And most vendors are committed to conform to developing standards after they are approved. But except for language standards, which began in the 1960's with the first standards for Fortran and Cobol, this universal embrace of standards for just about every aspect of software was unknown as recently as a decade ago. This is particularly noticeable in the area of operating system standards, where POSIX, "the Portable Operating System Interface for Computer Environments", has been embraced by virtually every operating system vendor.

Table 1. Technological trends influencing the design of DAQ software.

The emergence of C as a standard language for realtime systems.
The emergence of Object Oriented Methodology as a paradigm for programming.
The emergence of useful CASE tools with demonstrable benefit.
The emergence of Unix as the dominant OS in the workstation market.
The emergence of distributed computing as a paradigm for computing.
The emergence of practical examples of transparent distributed computing.
The emergence of X-windows as a standard user interface.
The emergence of POSIX as a standard operating system interface.
The emergence of open standards as a primary motivating force for vendors.
The emergence of industry consortia as a standard modus operandi for vendors.
The emergence of commodity software as a goal of most vendors.
The emergence of free software as a de facto implementation of many standards.

### 3. Standardization

Just as the early language standards attempted to define the common functionality that a programmer could expect to find from a system claiming to be "Fortran" or "Cobol", so the motivation for the standardization effort leading to POSIX, which began with the *usr/group* standard in 1984 [11], was to bring order to the chaos which had developed around the various versions of Unix that were proliferating at that time. The final POSIX 1003.1-1990 Standard, ISO/IEC 9945-1:1990 [4], very much reflects these origins. Developed under the auspices of the IEEE Technical Committee on Operating Systems and Application Environments (TCOS), it is a "minimal" standard in that most controversial issues were simply avoided in order to achieve the consensus necessary for approval. It is a "permissive" or "compromise" standard in that all remaining controversy was resolved by either designating several conflicting behaviors as "standard", or by declaring any possible behavior as "implementation defined" or more simply "unspecified" or "undefined". Even using these blatant excuses to avoid controversy, the POSIX effort took four years to be approved as a national standard in 1988, and two more to achieve international standard status in 1990.

Many unresolved issues uncovered during the POSIX 1003.1 standardization effort were pushed off into other TCOS working groups, so that by this time there are more than 20 POSIX 1003 project groups, some of them dealing with more than one subproject. Table 2 gives a recent list of these projects. Perhaps equally astonishing is that POSIX represents just one part of the bigger explosion in software standards that began in the 1980's and continues today. Examples include the multitude of networking standards, the user interface standards, the language standards, and the huge number of software engineering standards (over 250 by some counts).

#### 3.1. Issues in Standardization

There are a number of important meta-developments in this flurry of POSIX standardization activity which are having significant consequences for individual standards:

**Table 2. POSIX 1003 Standardization Projects.**

1003.0	Guide to OSE
1003.1-1990	Part 1: System API [C Language]
1003.1a	System API Extensions [C Language]
1003.1LIS	Part 1: System API [Language Independent]
1003.2-1992	Part 2: Shell and Utilities
1003.2a	Part 2: User Portability Extensions
1003.2b	Part 2: Shell and Utilities, ISO Revision
1003.3-1991	Test Methods for Measuring Conformance to POSIX
1003.3.1	Test Methods for Measuring Conformance to POSIX 1003.1
1003.3.2	Test Methods for Measuring Conformance to POSIX 1003.2
1003.4	Part 1: Realtime & Related System API
1003.4a	Threads Interface
1003.4b	Part 1: Realtime System API Extensions
1003.5-1992	1003.1 Ada Language Interfaces
1003.5a	Ada Language Interface Extensions
1003.5b	Ada Language Interface Realtime Extensions
1003.6	Security Interface
1003.7	Part 3: System Administration Interface
1003.7.1	Part 3 Amendment: Print Administration
1003.7.2	Part 3 Amendment: Software Administration
1003.7.3	Part 3 Amendment: User Administration
1003.8	Part 1: Network Transparent File Access
1003.9-1992	1003.1 Fortran-77 Language Interfaces
1003.10	Supercomputing AEP
1003.11	Transaction Processing AEP
1003.12	Part 1: Protocol Independent Network Interfaces
1003.13	Realtime AEP
1003.14	Multiprocessing AEP
1003.15	Part 1 Amendment: Supercomputing Batch Environment
1003.15a	Part 2 Amendment: Supercomputing Batch Environment
1003.16	1003.1LIS C Language Interfaces
1003.17	Directory and Name Services API
1003.18	Platform Environment Profile for multiuser timesharing
1003.19	1003.1LIS Fortran-90 Language Interfaces
1003.20	1003.1LIS ADA Language Interfaces
1003.21	Part 1 Amendment: Realtime Distributed Systems Communications
1003.22	Guide to OSE Security Framework

- (1) The evolution by all parts of POSIX from a historical Unix standard to a more general interface standard that is independent of any underlying operating system. Consequently, many non-Unix-based operating systems have committed to POSIX 1003.1 conformance, including VAX/VMS, OS/2 and Windows NT [12]. Most realtime

operating systems have already moved in this direction, including QNX, OS/9 and OS/9000, VxWorks, RTMX, pSOS+, and LynxOS [12]. According to their stated intentions, most realtime vendors can be expected to conform with 1003.4 as well, now that it has been approved.

- (2) The requirement by the International Standards Organization (ISO) that international interface standards be stated in a manner that is independent of the programming language. This has led to a revision of the current 1003.1-1990 POSIX standard, which is defined in terms of the C language, into the 1003.1LIS (Language Independent Standard), and has given rise to language binding projects for C (1003.16), Fortran (1003.9 and 1003.19), and Ada (1003.5 and 1003.20).
- (3) The realization that testing for conformance to a standard is a necessary part of the certification process, and that development of test procedures is an important part of the standardization process. POSIX 1003.3 [3] was developed and approved as a standard for developing test methods to measure conformance for other parts of the POSIX standard, but the state-of-the-art in test development has not advanced enough to enable test methods to be developed at the same pace as the standards they are intended to test. Therefore, all requirements that test methods be written before a standard could be submitted for approval were dropped. However, the need to advance the state-of-the-art of test method generation has become even more acute because of these difficulties [6].
- (4) The realization that no single standard covers all the topics needed for an application or class of applications to be portable. This led to the concept of an "Open System Environment (OSE)", which is a set of standards and specifications for interfaces, services, and data formats that together accomplish the various forms of portability. The POSIX 1003.0 Guide gives a reference model for a general OSE shown in Figure 1. An "Applications Environment Profile (AEP)" is a comprehensive subset of an OSE that includes appropriate choices for optional features of standards and specifications to support a particular class of applications. Finally, the "POSIX Environment Platform profile (PEP)", which was originally called a "Traditional Interactive Multiuser System (TIMS)", is a generalized AEP from which other AEPs can be derived. It was intended to define a complete, traditional Unix environment. Table 3 gives a list of current examples of POSIX AEP and PEP standards projects.

**Table 3. POSIX Profiles Projects.**

1003.10	Supercomputing AEP
1003.11	Transaction Processing AEP
1003.13	Realtime AEP
1003.14	Multiprocessing AEP
1003.15	Supercomputing Batch AEP
1003.18	timesharing PEP

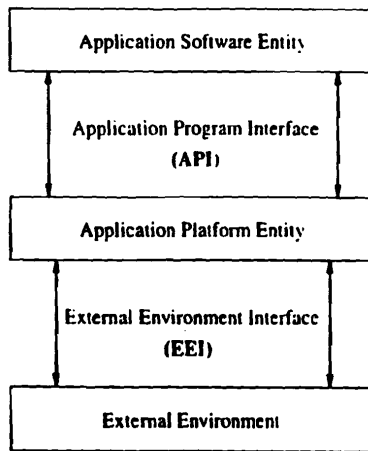


Figure 1. POSIX Open System Environment (OSE) Reference Model

### 3.2. Problems with Standardization

There are many criticisms of the rapid move toward POSIX standardization, of which only three will be dealt with here:

- (1) Not all standards have been finalized, and the approval process often drags on for years. What is a designer to do in the meantime? One answer is to build "thin interfaces" to the draft standards, an approach taken by the Virtual Operating System (VOS) developed at CERN [9]. This interface provides many of the realtime facilities included in POSIX 1003.4, but was defined, developed and put into use years before that standard became finalized. VOS defined facilities that are conceptually close to those being standardized without requiring the specific details, so that when the final standard is published, the "thin interface" can be rewritten and in most instances made simpler. The benefit to a DAQ based on VOS is that it could be implemented and put into use without waiting for the standard, yet once implementations of the standard become available, the DAQ can utilize them almost immediately because VOS has isolated any recoding to just its "thin interface".
- (2) Except for 1003.12, Protocol Independent Network Interfaces, and 1003.21, Realtime Distributed Systems Communications, all POSIX standards deal with a single platform. Yet the wave of the future is distributed computing, involving interaction between many different platforms communicating via networks, high-speed buses, etc. Other standards, such as the OSI standards and the X/Open model, are a necessary part of a total system design. Industry consortia have also developed more general models of distributed computing, such as OSF's DCE and UI's ATLAS.

- (3) There is the persistent fear that standardization has been misunderstood, "overhyped", and will inhibit innovation [5]. This is true in the sense that by conforming to an API standard vendors no longer will be developing different user interfaces to access files, semaphores, shared memory, etc. However, since it is the interface that has been standardized, developers are still free to come up with innovative implementations behind the interface. Vendors are also developing total environments around the standards that allow users to design, model, simulate, and generate code for their systems. This is the new "value" added by vendor innovations to the standards. It is a nice idea, provided one is aware of the traps, such as becoming "hooked" on one vendor by using that vendor's non-standard features indiscriminately. But because of the large and relatively stable base that standards offer a vendor, users should find many more products and "second sources" with higher quality and lower prices, plus improved interoperability and portability between products from different vendors.

### 4. Portability

There are many nuances to the term "portability" as used in the POSIX standard and in the earlier language standards. Primarily it has been taken to mean the ability to move source code from one implementation to another with minimal rewriting required to get it running on the new system. This is often called "source code portability". There are other important aspects of portability which are not addressed in these standards, but which have been dealt with elsewhere.

"Application portability" is the ability to move a complete program from one platform to another with minimal effort required to have it perform identically on the new system. This is broader than just source code portability, because it requires similar (ideally identical) execution semantics on the new system, which in turn requires identical interpretation of the meaning of the source code, and similar (again, ideally identical) functionality provided by the system facilities utilized by the source code. Application portability has been the primary goal of POSIX Application Programming Interfaces (APIs) and related standards.

"User portability" is the ability to move a user from one platform to another with minimal learning or retraining required to allow him or her to use the new system productively. X-windows, with the Motif or OPENLOOK Graphical User Interface (GUI) on top of it, has been the primary standard dealing with this aspect of portability [14].

"Information portability" is the ability to move information from one place to another with minimal (ideally no) conversion required for it to be useful in the new place. This has been the primary concern of the networking standards, notably ISO's Open Systems Interconnection (OSI) reference model and protocol suites, and the Internet development effort. The term "interoperability" is more commonly used to denote "information portability", perhaps because it more succinctly focuses on the key concern in moving information around: that all participants handling the information must agree on the ground rules in order to work together.

### 5. Interoperability

In order to accomplish the goal of interoperability, networking standards introduced the idea of a reference model composed of a hierarchy of functional layers, with well defined

interfaces between neighboring layers on the same system, and protocols for communicating between corresponding layers on separate systems. These three concepts: layering, interfaces, and protocols, are the primary design concepts we have to structure complex software and hardware systems.

### 5.1. Layering

"Layering" is the primary tool for dealing with complexity: breaking a complex problem down along functional lines, and assigning one function, or a set of closely related functions, to each layer. Variations of this basic paradigm have arisen under various names, such as "structured programming" and "levels of abstraction". Indeed, one of the major attractions of object oriented technology is that it provides a mechanism that not only encourages layered designs but helps to enforce design decisions as to which functions belong to which layers. The OSI reference model consists of 7 layers, many of which are further subdivided into sub-layers. The POSIX 1003.0 guide defines an OSE reference model consisting of three layers and two interfaces, as shown in Figure 1.

### 5.2. Interfaces

An "interface" is the meeting point between two layers in a hierarchy. Perhaps the most important design issue in large software systems is the correct specification of clean functional interfaces between the components. It is no accident that the POSIX 1003.1 standard is known as the "System Applications Program Interface Standard", because it attempts to define an interface that will hide implementation details of the operating system from the functional effect a user application program expects of the implementation. Although one may argue that POSIX 1003.1 does not completely achieve this goal, one can also argue that any failing in this respect is due to its role as the definition of a historical Unix system. Clearly the plethora of POSIX 1003.x standards spawned from the initial POSIX effort have focused almost exclusively on the goal of defining interfaces.

### 5.3. Protocols

A "protocol" is the set of rules and data formats by which information flows between two or more entities at corresponding layers in a hierarchy. The task of precisely defining a protocol invariably requires a correspondingly precise definition of a model of the participants in the information exchange. This in turn lends itself especially well to simulation and more formal techniques for verifying the correctness and completeness of the design. Protocols also lend themselves to development of test suites to verify conformance to a standard, because they permit the testing method to observe information flow without interfering with the entities participating in the flow.

### 6. Realtime Standards

The POSIX 1003.4 "Realtime System API" standard consists of a number of operating system facilities that are traditionally associated with realtime programming. The topics covered by this standard are shown in Table 4.

Table 4. POSIX 1003.4 Realtime Extensions.

Binary semaphores
Process memory locking
Shared memory
Priority scheduling
Asynchronous event notification
High resolution timers
Interprocess communication
Synchronized I/O
Asynchronous I/O
Realtime files
Performance metrics

Unlike the POSIX 1003.1 standard, which was essentially a codification of existing Unix facilities, a number of the realtime facilities in POSIX 1003.4 did not exist in Unix and were invented by the standards committee. This is more the approach taken by ISO in developing the protocols associated with its OSI model and is more like an exercise in design than in standardization. Perhaps for this reason, the standard underwent considerable discussion and revision (14 drafts) that dragged out for a lengthy period of time. In the process, two subsidiary projects were spawned: POSIX 1003.4a to consider a threads interface, and POSIX 1003.4b to consider additional extensions that could not be resolved in time for approval under 1003.4 itself. Because there are no prior implementations of some of these realtime features, it is difficult to foresee the problems, interpretations, and side effects that will arise when they are actually implemented and utilized in practice.

At the current time, draft 14 of POSIX 1003.4 has been approved by the IEEE Standards Board and is awaiting publication and eventual approval by ISO. The other parts of 1003.4, as well as the realtime AEP (1003.13), are still under development and may not be approved for some time.

### 6.1. Performance

Of significant importance in the POSIX 1003.4 standard is the inclusion of performance metrics. This is the only POSIX standard that explicitly deals with performance issues, all the others defining conformance in terms devoid of any reference to the time taken to perform a particular function. A particularly nasty area is the interaction between signals and the realtime extensions, particularly threads. For example, since 1003.1 is specified without regard to performance, there is no effective constraint on system overhead during context switches, during interrupt servicing, during critical section lockouts, etc. The approach of many historical Unix implementations, to simply delay delivery of signals generated during a system call, cannot be used in a realtime operating system.

The performance metrics included in POSIX 1003.4 require a conformant system to specify a maximum time to perform one operation of a single system call, such as posting a semaphore, or waiting on a semaphore, in one or more well defined situations. There is also the requirement that performance metrics be supplied retroactively for functions defined in

other parts of the POSIX standard, such as 1003.1, since the performance of many of these functions may be crucial to a realtime application.

It is perhaps the role of the realtime AEP (1003.13) to specify more clearly the need for performance metrics that apply to the entire operating system, not just individual functions. There is also a need for a set of well-defined benchmarks that go beyond tests verifying that a feature does or does not conform to the standard. These benchmarks need to characterize classes of applications that utilize combinations of features, as specified in an AEP. This is the only reliable way that performance of different systems can be compared [10].

## 6.2. Conformance

For all the POSIX standards, it has been left to national organizations to develop test suites and approve testing laboratories, accreditation criteria, validation bodies and certification procedures. In the U.S., this is handled by the National Institute of Standards and Technology (NIST). NIST's Computer Systems Laboratory (CSL) is the official POSIX validation body in the U.S., and NIST's Federal Information Processing Standard (FIPS) 151-2 [7] defines the testing required for certification of conformance to the POSIX 1003.1-1990 standard.

## 7. Utilizing Standards In DAQ Design

It is clear that the proliferation of standards at all levels in the software picture will have a significant influence on the design of data acquisition systems. If nothing else, it has already had a profound influence on the design of realtime operating systems and their vendors [2][13]. Gone are the days of small, stand-alone kernels designed with little regard for other kernels or development systems. Today, virtually every realtime vendor has committed to both POSIX 1003.1 and 1003.4, plus has provided networking (typically TCP/IP) and graphical user interfaces (typically X-windows). In addition, realtime operating systems are frequently bundled with compilers for standard languages (typically C, C++, Ada), integrated visual debuggers, and other facilities that constitute a complete development environment. It is rare to find a vendor that provides systems for just a single hardware platform — portability between different platforms has become almost as important for the vendors (in terms of providing a bigger market for software as a commodity) as it has for the users (in terms of providing a bigger choice of products). Undoubtedly, the existence of POSIX has provided a DAQ system designer with more choice. Vendors are more likely to implement an approved standard than to undertake the risk and cost of designing their own equivalent system. Reduced risk and cost also makes it more likely vendors will make long term commitments to supporting a standard product across various platforms, to the obvious benefit of the users.

Consequently, the factors going into the choice of a realtime operating system have less to do with kernel-type functionality (they all provide POSIX, networking, graphical user interfaces, etc.) and more to do with how well the many different standard facilities are integrated and supported by the vendor. Performance is still an issue, of course, but it now depends less on raw processing speed since newer, faster platforms are constantly coming to market (it is estimated that processor speeds have been doubling every 18 months to two years for quite some time). Therefore, the ability to simply move easily to the newer, faster technology (i.e., to be portable) is often the cheapest solution to a performance limitation. Perhaps even more

important may be the need to integrate older existing technology with newer technology in a seamless manner, a goal requiring a layered system designed for interoperability.

Since vendors are providing facilities defined by standards, system designers need to know what those facilities are and how they can be incorporated into the designs. For starts, the general guidelines for open systems, in particular the POSIX 1003.0 Guide to OSE, and the X/Open XPG, should be required reading for the DAQ design team. Designers should also follow the guidelines utilized by the standard reference models when designing their own system: the primary design emphasis should be on developing a reference model for the data acquisition system, subdividing it into functional layers with clean interfaces that can be distributed between entities communicating with efficient protocols. Models should be developed for the layers, and test suites developed along with them to verify that the different entities will interoperate. Mandatory use of object oriented languages, such as C++, will help to enforce the modularity of the layers.

It is essential that the design team be kept small and remain with the project from its inception through to its actual use in experiments, so that at least a few collaborators will always have a "total picture" of all aspects of the system. The designers must ensure the conceptual integrity of the system, and try to convey this coherently to all collaborators by providing accurate, up-to-date documentation accessible via the World-Wide Web (WWW). Given the hypertext nature of WWW, links should be provided directly into all parts of the development effort, including design documents, models, simulations, even actual code modules for purposes of sharing and review.

One of the lessons vividly demonstrated by the layered models used in networking is that the layers provide a clean framework for dealing with changing technology. Higher layers and their protocols (i.e., TCP and IP for example) can remain essentially unchanged while lower layers and their protocols are completely reimplemented to conform to newer technologies (i.e., ethernet, token ring, FDDI, ATM, etc.). And when eventually the assumptions embodied in the higher layers become out of date, they too can be reimplemented, thereby providing significant flexibility to a data acquisition system whose lifetime is to extend over more than one or two years.

A good example of the advantages of layered design is WWW [1], which is also an instance of a highly useful application of distributed computing accomplished in a transparent manner. WWW can serve as an example of how the pieces of a DAQ might be designed, developed and tested in a geographically disperse manner. Analogous with WWW, the DAQ design could include a control layer that identifies objects in a universal manner and provides a general protocol for communication between them. During development and test, the objects would be geographically dispersed and communication would be over the Internet. Although the data rates would not be adequate, the interoperability of the parts could be validated. When the parts are brought together for the actual experiment, the Internet layers would be replaced by higher-performance communications, but the majority of the integration would have already been accomplished. This design approach also lends itself to modeling and simulation, because of its emphasis on layering, interfaces, and protocols.



## 8. Conclusion

This paper has described the role of open software standards, particularly POSIX, on the design and implementation of DAQ systems. A number of issues were raised, and suggestions given as to how to cope with them in building a DAQ system. A much more detailed discussion of open standards, along with their history, is given in [8].

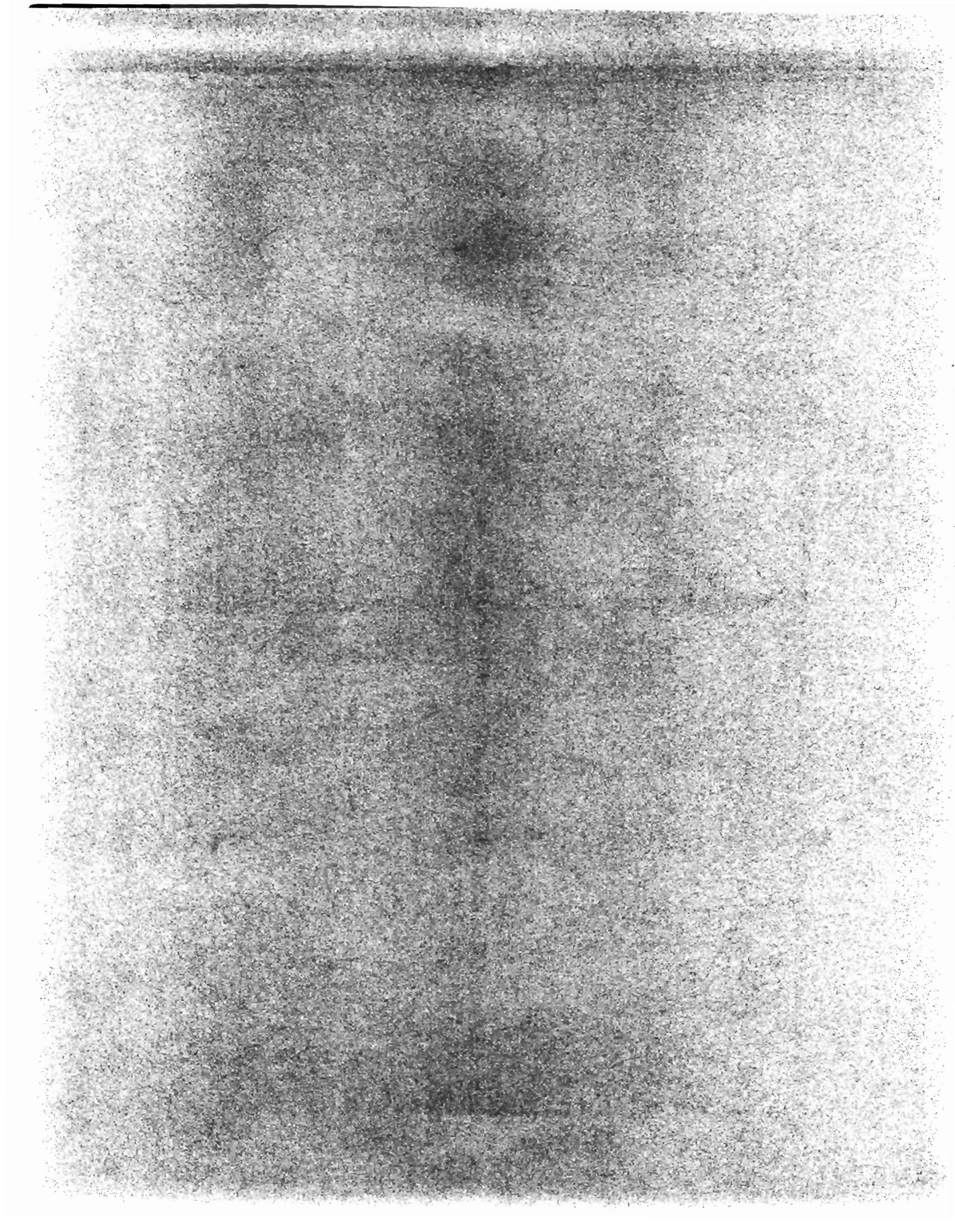
## 9. References

- [1] Berners-Lee, Tim, et. al., "The World-Wide Web", *Communications of the ACM* (Aug 1994) vol 37 no 8 pp 76-82.
- [2] Child, Jeffrey, "Realtime OSs diversify to attract", *Computer Design* (Aug 1994) pp 101-108.
- [3] IEEE, *Test Methods for Measuring Conformance to POSIX*, IEEE, New York, NY (1991). IEEE Std 1003.3-1991.
- [4] IEEE, *Portable Operating System Interface (POSIX) — Part 1: System Application Program interface (API) [C Language]*, IEEE, New York, NY (1990). IEEE Std 1003.1-1990. ISO/IEC 9945-1:1990.
- [5] Joseph, Moses, "Realtime POSIX: Boon or bunk?", *Computer Design* (Sep 1994) pp 155-160.
- [6] Leathrum, James F., and Liburdy, Kathleen A., "Impact of the role of testing in POSIX is expanding", *IEEE Computer* (Nov 1993) pp 81-82.
- [7] NIST, *FIPS 151-2 POSIX*, National Institute of Standards and Technology, Gaithersburg, MD (1991).
- [8] Quarterman, John S., and Wilhelm, Susanne, *UNIX, POSIX and Open Systems*, Addison-Wesley Publishing Company, Inc., Reading, MA (1993).
- [9] Russell, R. D., and Mornacchi, G., "VOS, a Virtual Operating System", CERN (Jul 1990).
- [10] Singh, Inder, "Realtime Benchmarking", *Computer Design* (Feb 1994) pp 125-130.
- [11] /usr/group, *1984 /usr/group Standard*, Uniform, Santa Clara, CA (1984).
- [12] Williams, Tom, "POSIX realtime may be a long time coming", *Computer Design* (Jul 1993) pp 38-40.
- [13] Williams, Tom, "Realtime OSs seek more functions and standards", *Computer Design* (Jul 1993) pp 99-105.
- [14] X/Open, *X/Open Portability Guide, Issue 3*, Prentice-Hall, Englewood Cliffs, NH (1989).

**S7-3**

**"Software Protocols for Event Builder Switching Networks"**

**(Irakli Mandjavidze - CERN/Saclay)**



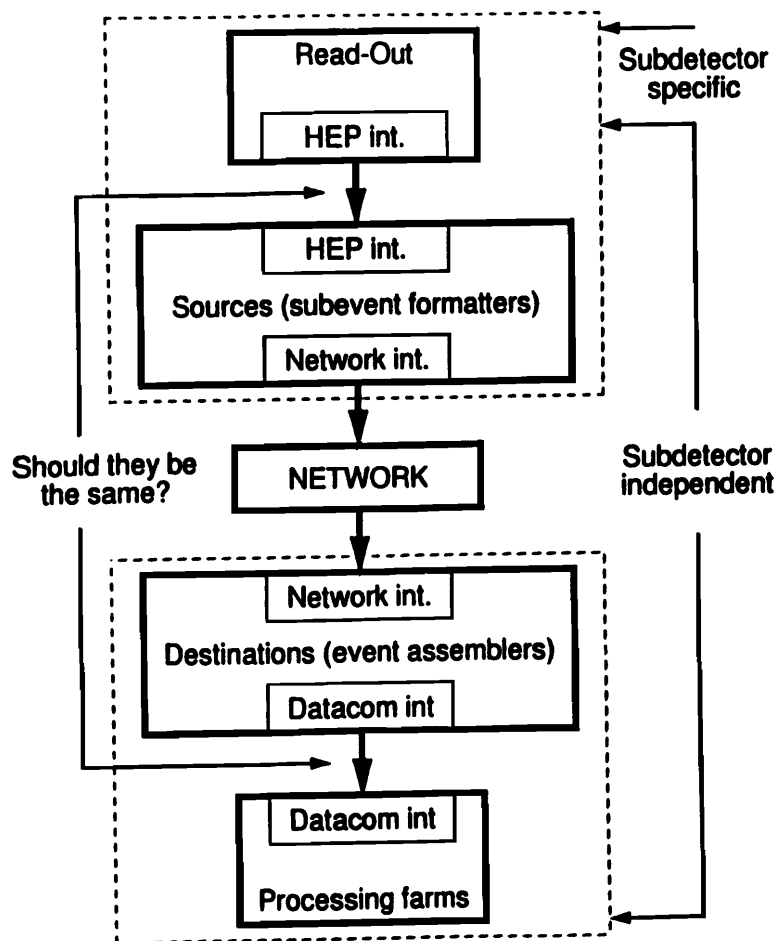
## Software Protocols for Event Builder Switching Networks

I. Mandjavidze  
RD31, CERN/ECP  
mandjavi@sunvisi.cern.ch

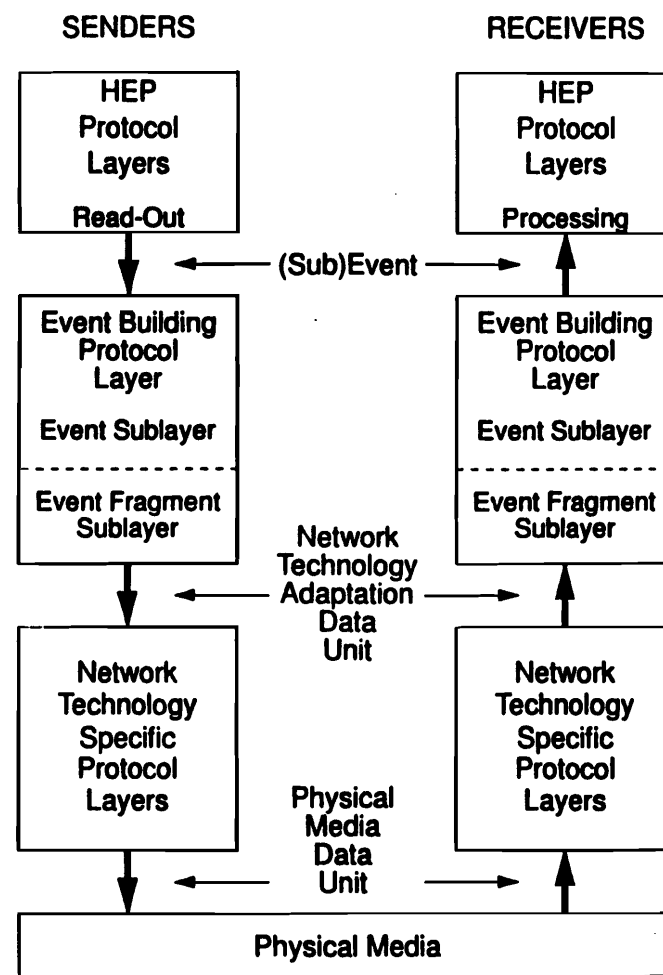
### OUTLINE

- 1) An Event Builder System
- 2) A Layered Structure of HEP Data Flow Protocol Model
- 3) An Event Fragment Sublayer
- 4) Event Building Schemes
  - \* Known Sources
  - \* Empty Records
  - \* Compete on Next
  - \* Time Out
  - \* Table of Comparison
- 5) A Demonstrator System
- 6) A Software Structure for Sources and Destinations
- 7) Discussion

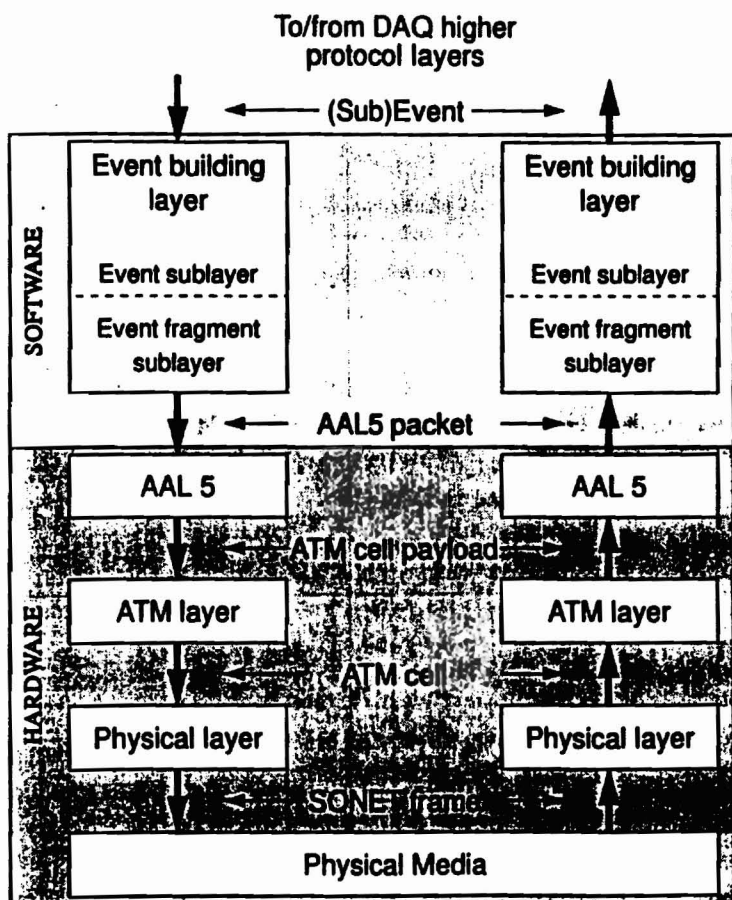
### An Event Builder System



### Layered Structure of HEP Data Flow Protocol Model

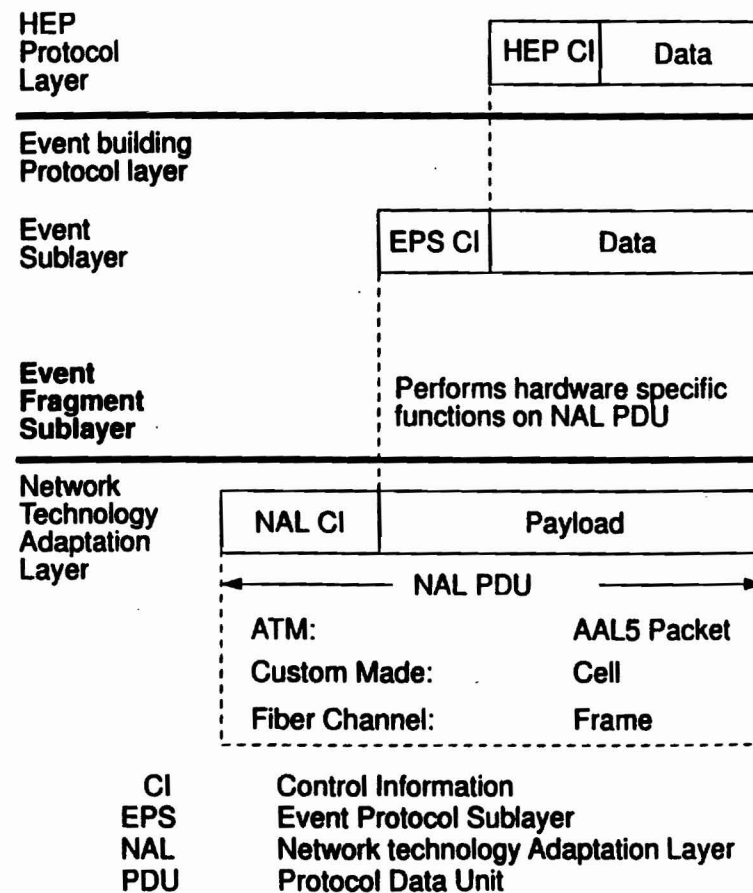


### Data Flow Protocol Model in the case of ATM technology



### Event Fragment Sublayer

Provides independence of Event Sublayer from Hardware



### Event Fragment Sublayer

More Functionality

AAL5 packet size is up to 64KByte

Event fragment size can be bigger

- \* ALICE
- \* Calibration events for ATLAS and CMS

Therefore event fragment sublayer should provide

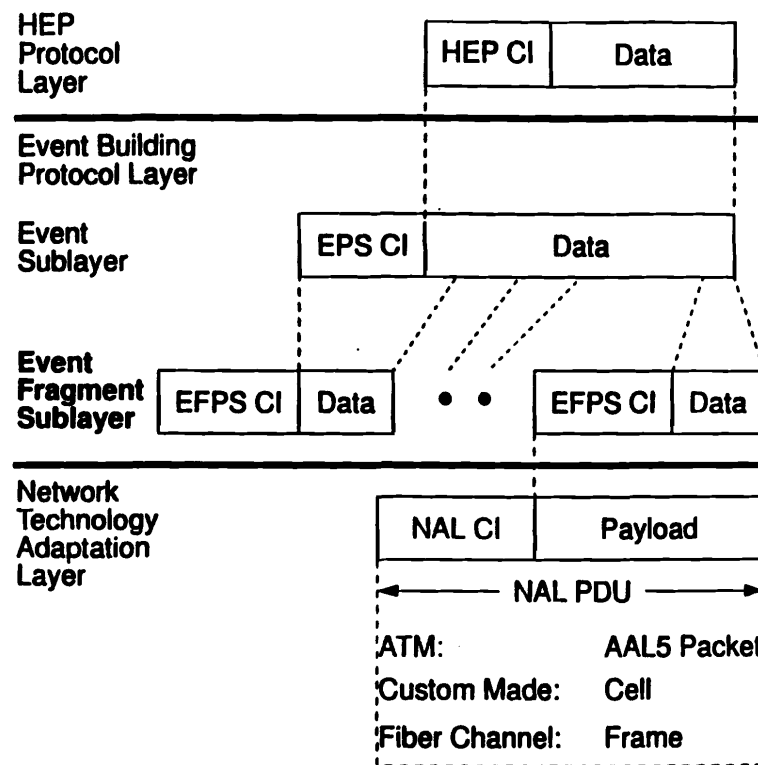
Sender

Segmentation of event fragments into AAL5 packets

Receiver

Reassembly of event fragments from AAL5 packets

### Event Fragment Sublayer



CI	Control Information
EPS	Event Protocol Sublayer
EPFS	Event Fragment Protocol Sublayer
NAL	Network technology Adaptation Layer
PDU	Protocol Data Unit

## Event Sublayer

### Source

Prepare event fragment data for event building in destinations

### Destination

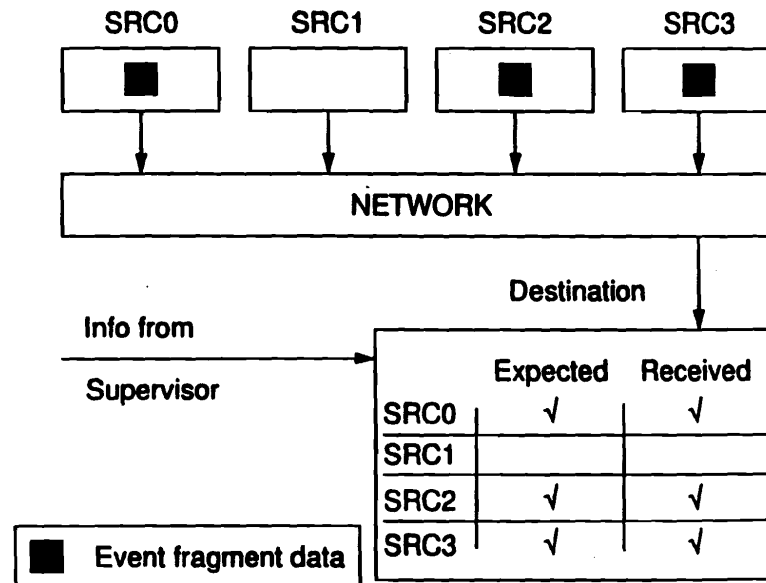
Perform actual event-fragment/event association

### Event Building Schemes

- Known Sources
- Empty Records
- Compete on Next
- Time Out

## Event Building Scheme: Known Sources

Sources participating in the event building process are known



Event is built when data from all participant sources are received

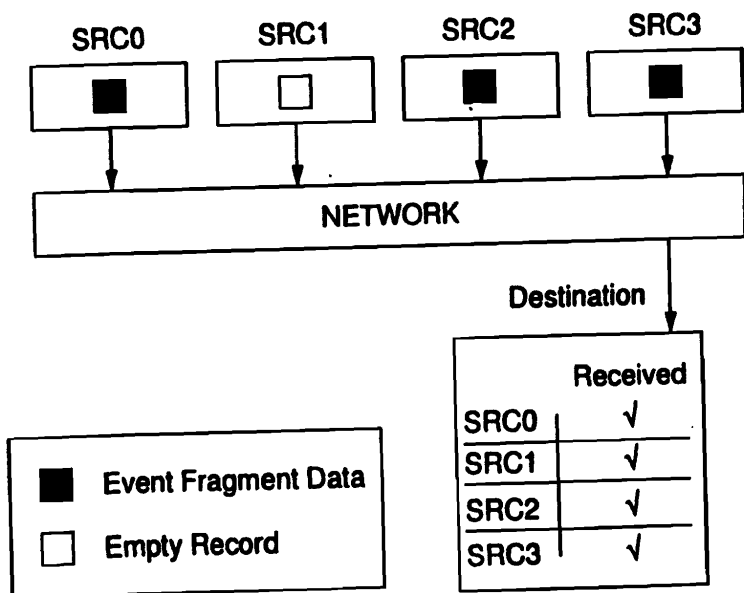
Possible use: ROI building within the ATLAS L2



### Event Building Scheme: Empty Records

Event building system is large  
Sources participating in the event building process are unknown

Empty records assist event building process



Event is built when data from all sources are received

Possible Use: CMS L2 event-building

### Event Building Scheme: Complete on Next

If event building latency is not important

Events assigned successively to the same destination

	Event # A	Event # I	Event # Z
SRC0	■	■	■
SRC1	■		■
SRC2	■	■	■
SRC3	■	■	■

The Destination

Event Reassembly Tables								
Event # A			Event # I			Event # Z		
Received			Received			Received		
SRC0	✓		SRC0	✓		SRC0	✓	
SRC1	✓		SRC1			SRC1	✓	
SRC2	✓		SRC2	✓		SRC2	✓	
SRC3	✓		SRC3	✓		SRC3	✓	

Currently received event assists in previous event reassembly

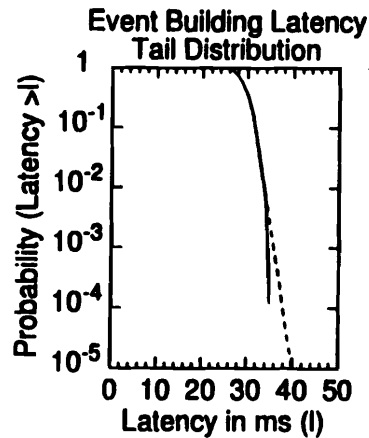
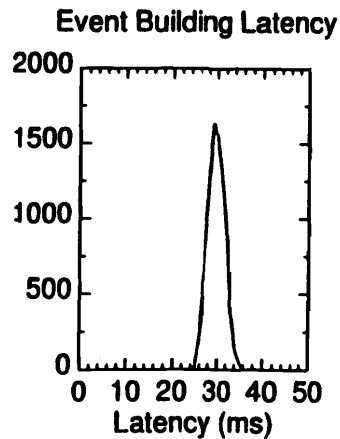
Possible Use: L3 Event Building

### Event Building Scheme: Time-Out

Event building latency is not important  
or  
Event building latency distribution is narrow

Event Builder	1024x1024
Event Size	1 Mbyte
Trigger Rate	7.5 KHz

min	24
mean	30.22
max	36
sigma	1.86



Event is assumed to be built after predefined "Time Out" interval

Possible Use: L3 Event-Building

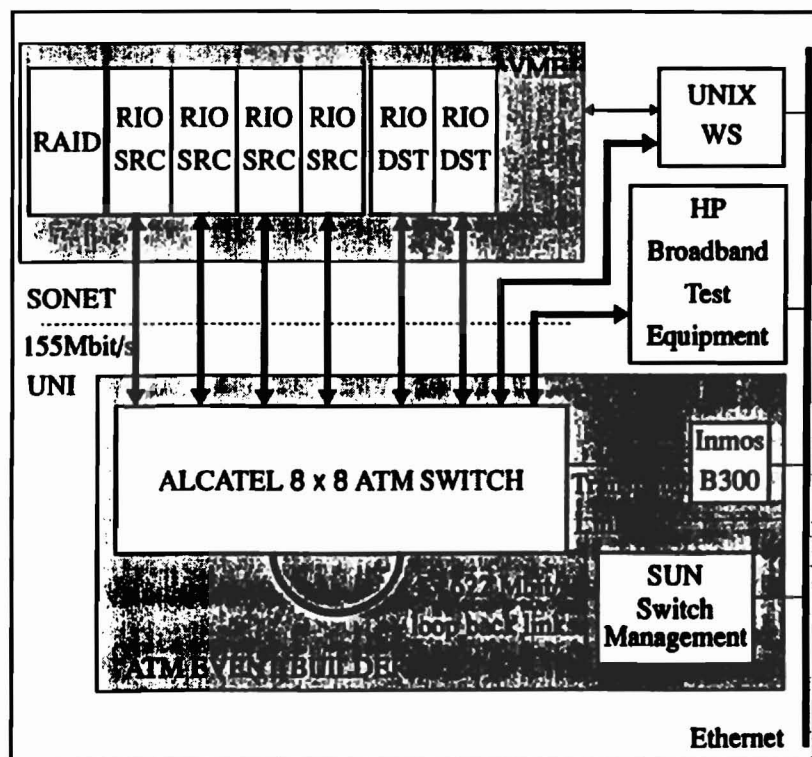
NOTE: "Time Out" protection is necessary in any case to recover from failures (e.g. dead source)

### Event Building Schemes

Table of Comparison

Scheme	Viable		Implementation Complexity	
	L2	L3	Hardware	Software
Known Sources	OK	No	Very high	Easy
Empty Records	OK	OK	Traffic Overhead	Software Overhead
Complete on Next	Limited	Limited	Easy	Easy
Time-Out (obligatory)	No	OK	Easy	Easy

## VME-based Event Builder Demonstrator for protocol development and evaluation



## Source/Destination Software Structure

### Event building protocol layer

- *Event Protocol Sublayer*
- *Exception handling*

### Network interface layer

- *AAL5, ATM and Physical layer initialization*
- *AAL5 Packet Segmentation/Reassembly control*
- *AAL5, ATM and Physical layer exception handling*
- *AAL5, ATM and Physical layer statistics*
- *Traffic Shaping (source)*
- *Event Fragment Protocol Sublayer*

### Hardware specific layer

- *AAL5 and ATM layer interface library*
- *Physical layer interface library*
- *Traffic Shaping interface library (source)*

## **Goals of software protocol developments**

- \* **Working event builder demonstrator system**
- \* **Network technology independent event protocol sublayer**
- \* **Study of various event building schemes**
- \* **Optimization of software overhead in sources and destinations**
- \* **Implementation of traffic shaping schemes**
- \* **Support for exception handling and error recovery**

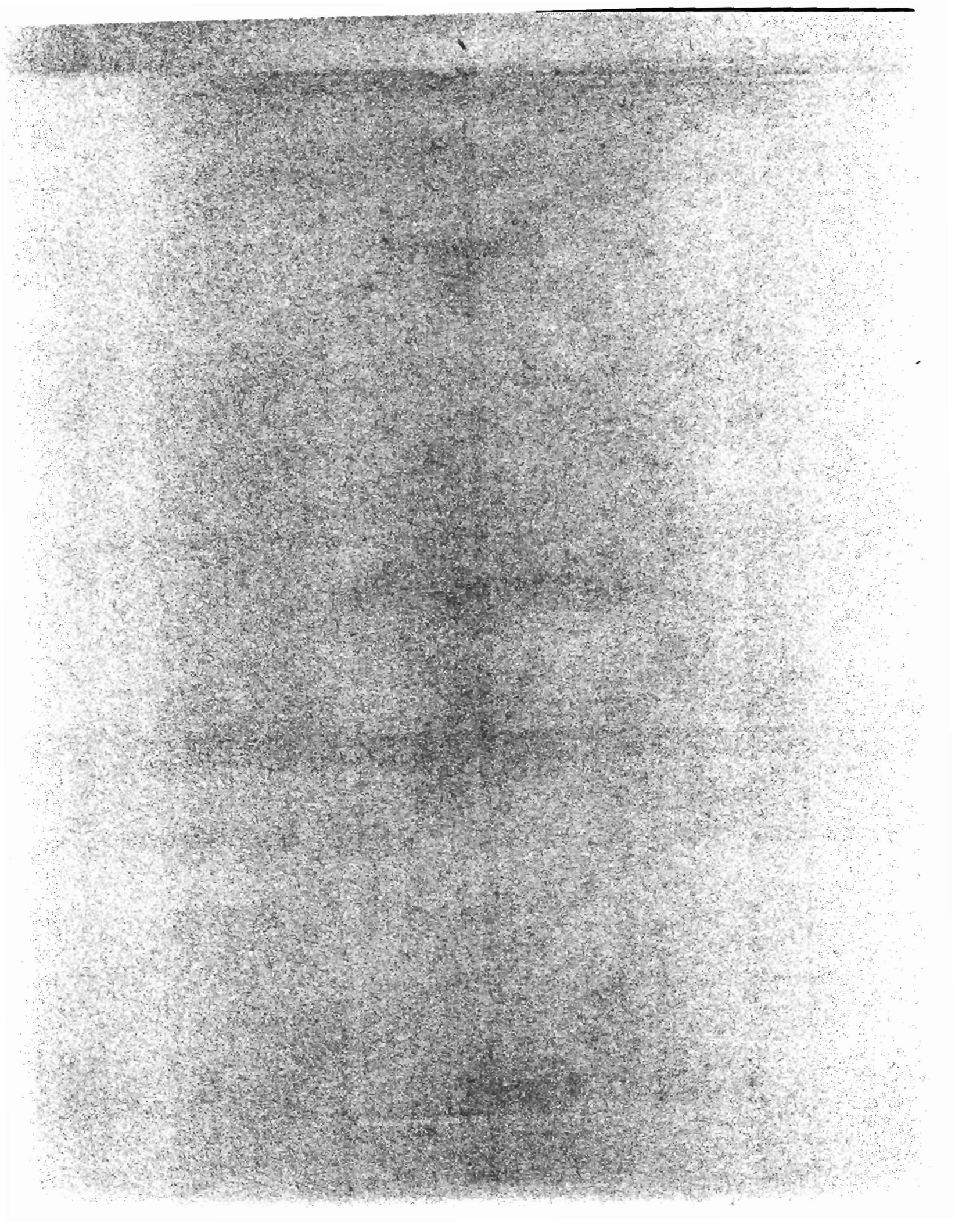


## **S8-1**

### **"A Scalable Fibre Channel Architecture for Event Building"**

**(Bill Greiman - LBL)**

For large event builders two stages of switches are used to provide event building at the subdetector and subfarm level. A modsim model has been developed. This model has been validated with measurements on a small HIPPI switch at RD13 in CERN. The model is used to simulate performance of a large event builder suitable for an LHC experiment such as ATLAS. Results include buffer sizes, queue lengths and latency as a function of event rate.



## Software Protocols for Event Builder Switching Networks

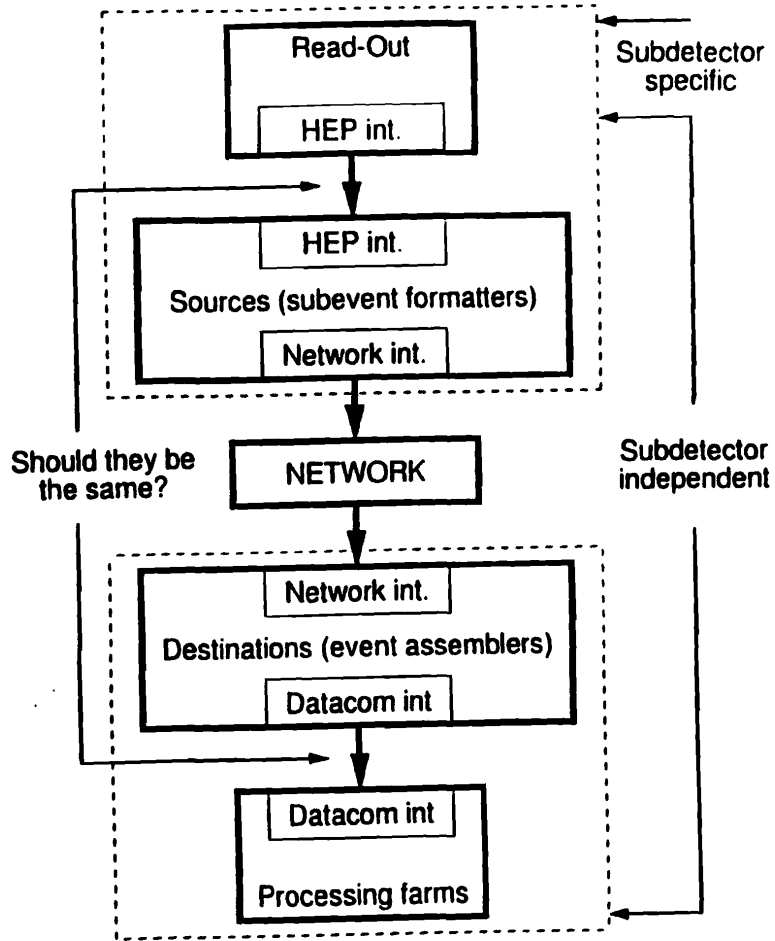
I. Mandjavidze  
RD31, CERN/ECP  
mandjavi@sunvlsi.cern.ch

### OUTLINE

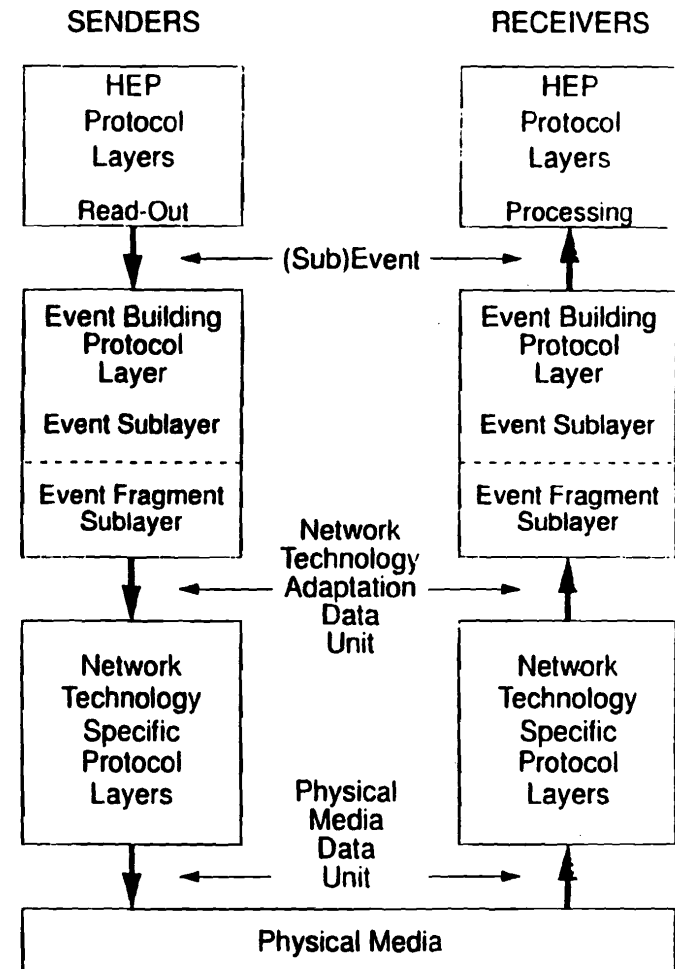
- 1) An Event Builder System
- 2) A Layered Structure of HEP Data Flow Protocol Model
- 3) An Event Fragment Sublayer
- 4) Event Building Schemes
  - \* Known Sources
  - \* Empty Records
  - \* Compete on Next
  - \* Time Out
  - \* Table of Comparison
- 5) A Demonstrator System
- 6) A Software Structure for Sources and Destinations
- 7) Discussion



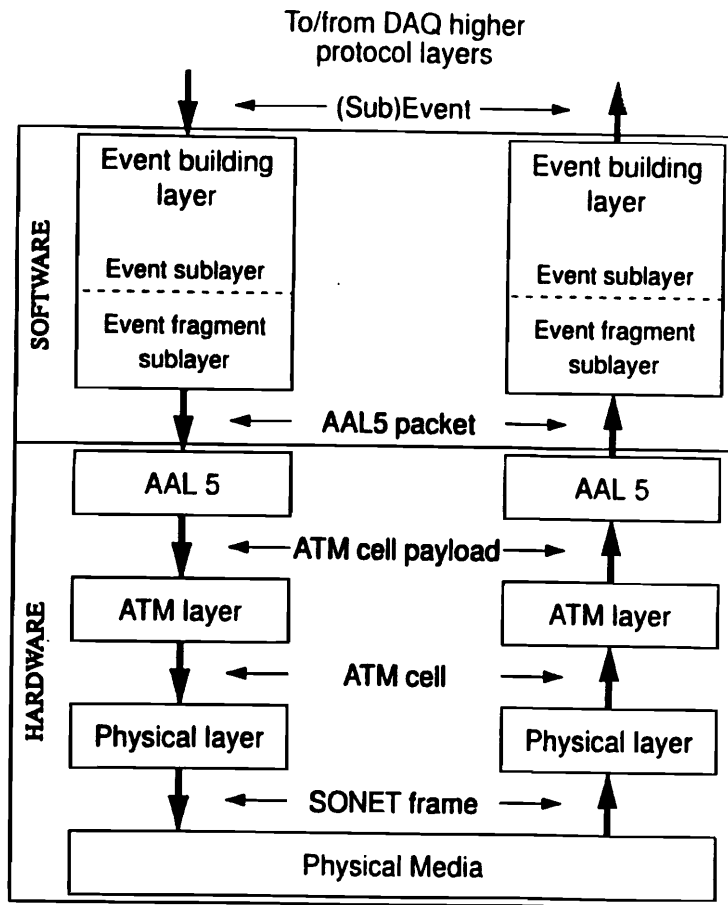
### An Event Builder System



### Layered Structure of HEP Data Flow Protocol Model

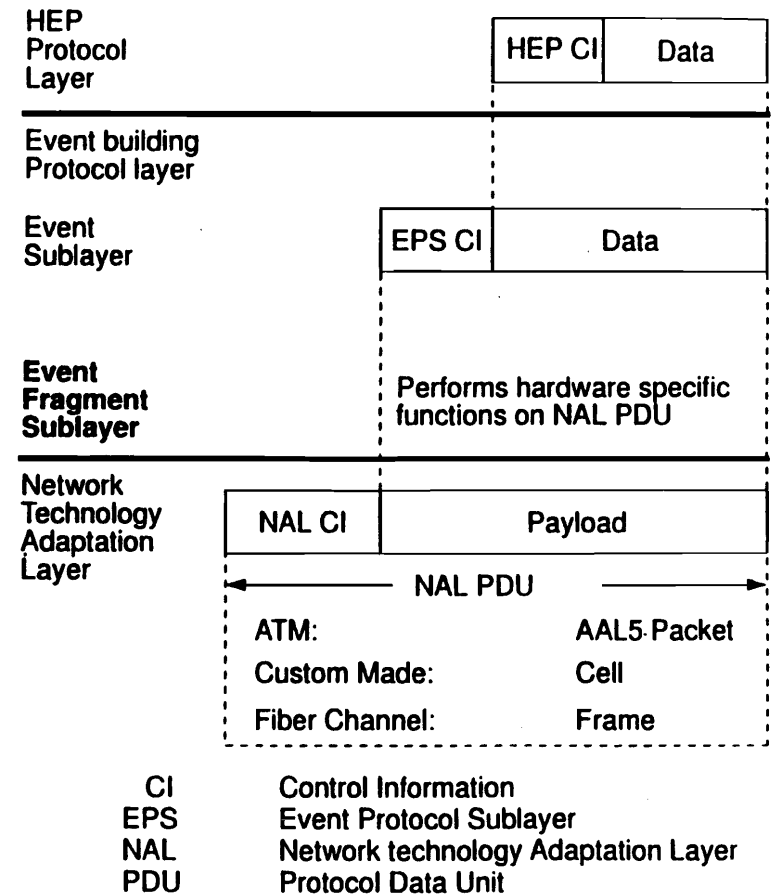


### Data Flow Protocol Model in the case of ATM technology



### Event Fragment Sublayer

Provides independence of Event Sublayer from Hardware



### Event Fragment Sublayer

More Functionality

AAL5 packet size is up to 64KByte

Event fragment size can be bigger

- ALICE
- Calibration events for ATLAS and CMS

Therefore event fragment sublayer should provide

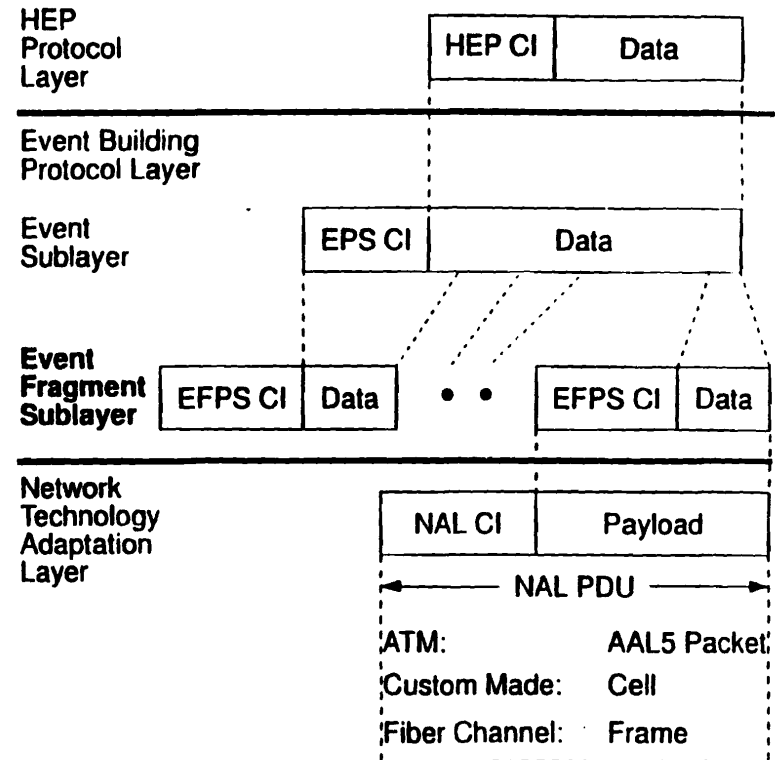
Sender

Segmentation of event fragments into AAL5 packets

Receiver

Reassembly of event fragments from AAL5 packets

### Event Fragment Sublayer



CI	Control Information
EPS	Event Protocol Sublayer
EPFS	Event Fragment Protocol Sublayer
NAL	Network technology Adaptation Layer
PDU	Protocol Data Unit

## Event Sublayer

### Source

Prepare event fragment data for event building in destinations

### Destination

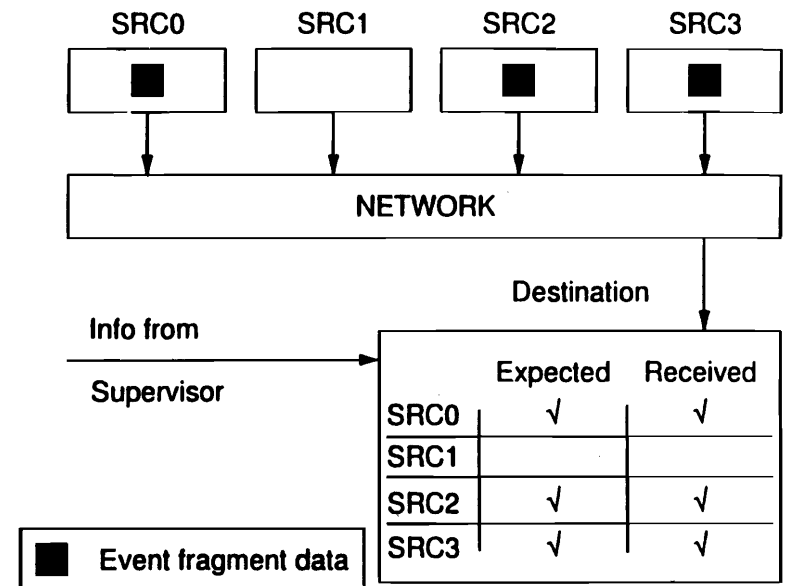
Perform actual event-fragment/event association

### Event Building Schemes

- \* Known Sources
- \* Empty Records
- \* Compete on Next
- \* Time Out

## Event Building Scheme: Known Sources

Sources participating in the event building process are known



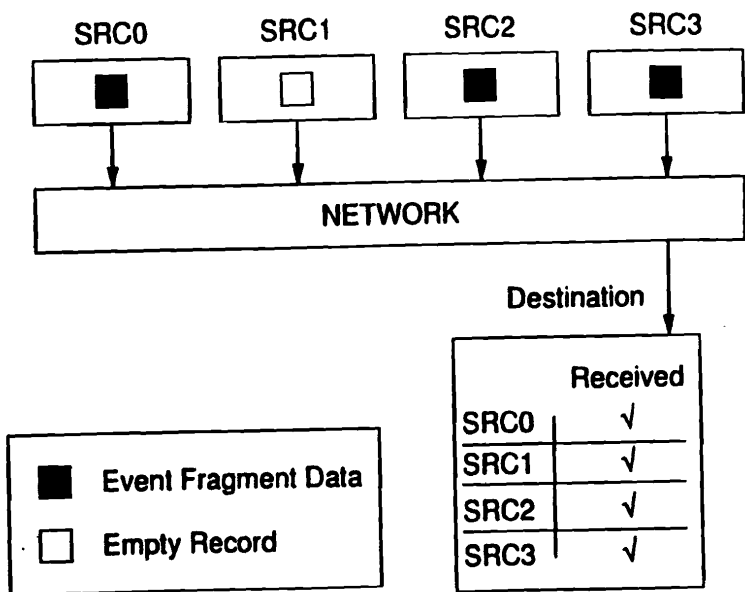
Event is built when data from all participant sources are received

Possible use: ROI building within the ATLAS L2

### Event Building Scheme: Empty Records

Event building system is large  
Sources participating in the event building process are unknown

Empty records assist event building process



Event is built when data from all sources are received

Possible Use: CMS L2 event-building

### Event Building Scheme: Complete on Next

If event building latency is not important

Events assigned successively  
to the same destination

	Event # A	Event # I	Event # Z
SRC0	■	■	■
SRC1	■		■
SRC2	■	■	■
SRC3	■	■	■

The Destination

Event Reassembly Tables								
Event # A		Event # I		Event # Z				
Received		Received		Received				
SRC0	✓	SRC0	✓	SRC0	✓			
SRC1	✓	SRC1	✓	SRC1	✓			
SRC2	✓	SRC2	✓	SRC2	✓			
SRC3	✓	SRC3	✓	SRC3	✓			

Currently received event assists in previous event reassembly

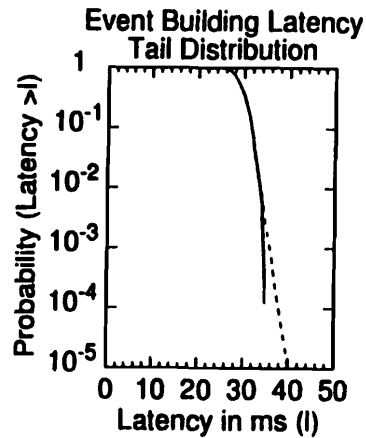
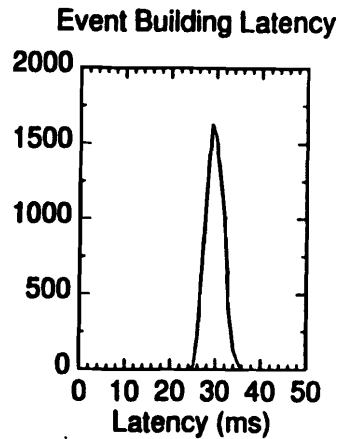
Possible Use: L3 Event Building

### Event Building Scheme: Time-Out

Event building latency is not important  
or  
Event building latency distribution is narrow

Event Builder	1024x1024
Event Size	1 Mbyte
Trigger Rate	7.5 KHz

min	24
mean	30.22
max	36
sigma	1.86



Event is assumed to be built after predefined "Time Out" Interval

Possible Use: L3 Event-Building

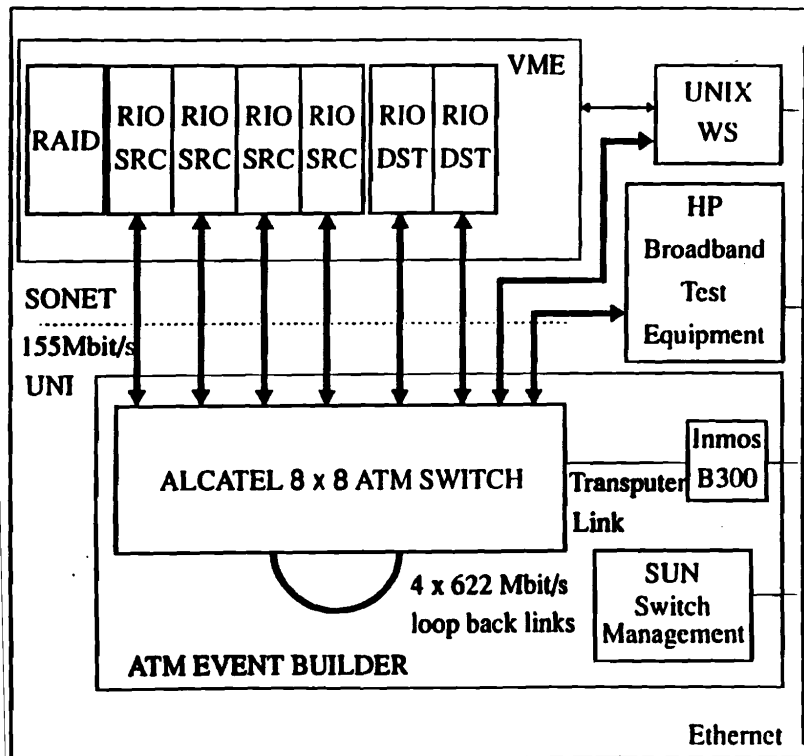
NOTE: "Time Out" protection is necessary in any case to recover from failures (e.g. dead source)

### Event Building Schemes

Table of Comparison

Scheme	Viable		Implementation Complexity	
	L2	L3	Hardware	Software
Known Sources	OK	No	Very high	Easy
Empty Records	OK	OK	Traffic Overhead	Software Overhead
Complete on Next	Limited	Limited	Easy	Easy
Time-Out (obligatory)	No	OK	Easy	Easy

## VME-based Event Builder Demonstrator for protocol development and evaluation



## Source/Destination Software Structure

### Event building protocol layer

- *Event Protocol Sublayer*
- *Exception handling*

### Network interface layer

- *AAL5, ATM and Physical layer initialization*
- *AAL5 Packet Segmentation/Reassembly control*
- *AAL5, ATM and Physical layer exception handling*
- *AAL5, ATM and Physical layer statistics*
- *Traffic Shaping (source)*
- *Event Fragment Protocol Sublayer*

### Hardware specific layer

- *AAL5 and ATM layer interface library*
- *Physical layer interface library*
- *Traffic Shaping interface library (source)*

## **Goals of software protocol developments**

- \* Working event builder demonstrator system
- \* Network technology independent event protocol sublayer
- \* Study of various event building schemes
- \* Optimization of software overhead in sources and destinations
- \* Implementation of traffic shaping schemes
- \* Support for exception handling and error recovery



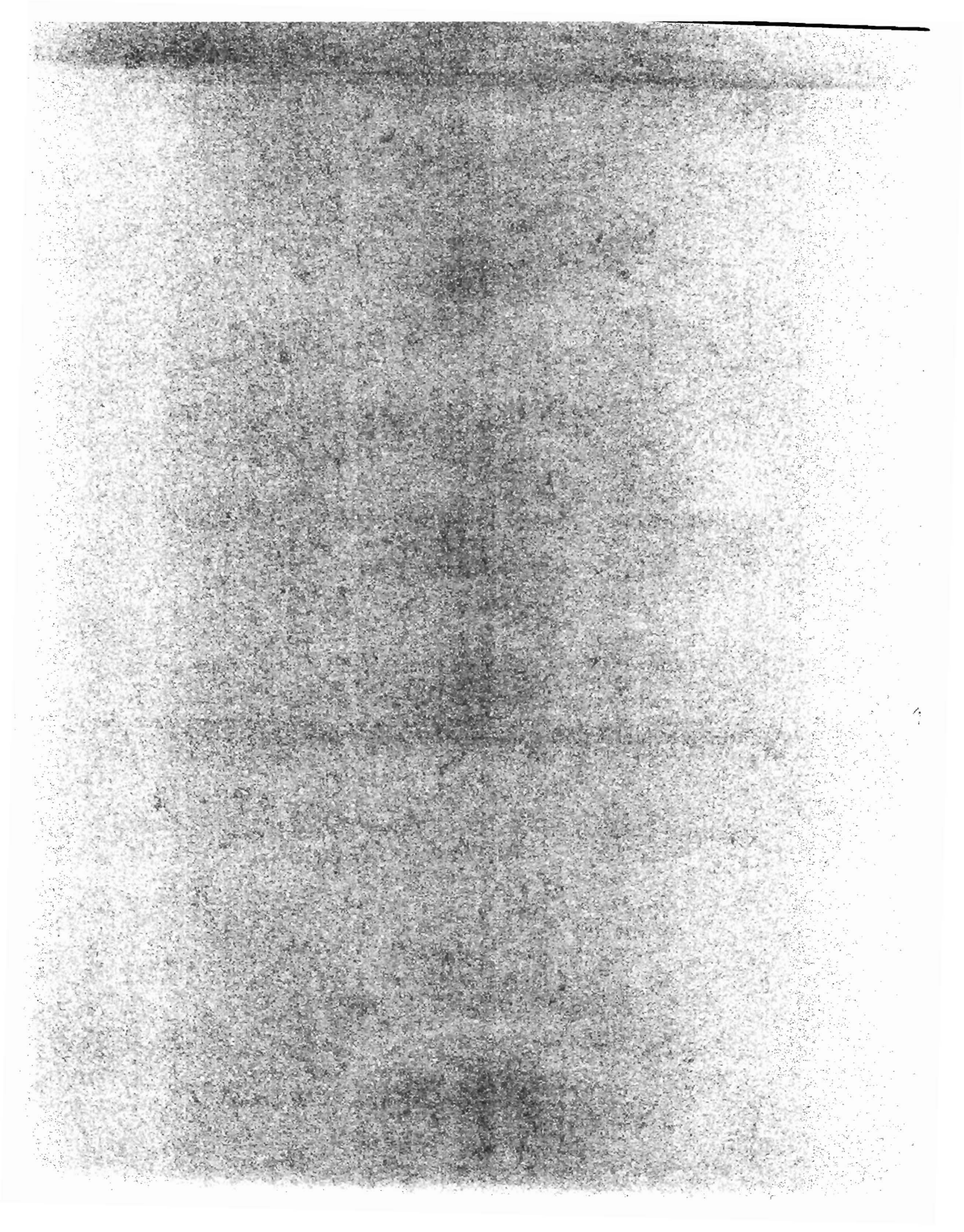


**S8-1**

**"A Scalable Fibre Channel Architecture for Event Building"**

**(Bill Greiman - LBL)**

For large event builders two stages of switches are used to provide event building at the subdetector and subfarm level. A modish model has been developed. This model has been validated with measurements on a small HIPPI switch at RD13 in CERN. The model is used to simulate performance of a large event builder suitable for an LHC experiment such as ATLAS. Results include buffer sizes, queue lengths and latency as a function of event rate.



# Design and Simulation of Fibre Channel Based Event Builders

W. Greiman

Lawrence Berkeley Laboratory, Berkeley, CA, 94720, USA

L. Mapelli, G. Mornacchi and R. Spiwoks  
CERN, Geneva, Switzerland

A model for event builders based on Fibre Channel and HiPPI switches is described. A simulation program for this model is implemented in MODSIM II. The model and program are verified using measured data from a small HiPPI event builder. Performance of an event builder using a single 256 x 256 port switch is simulated. A two stage architecture for event building is described and simulation is carried out for a large two stage event builder. Results of the single and two stage event builders are compared.

## INTRODUCTION

A number of proposals [1,2,3] have been made to use commercial communications switches for event building in DAQ systems for future high-energy physics experiments. These proposals are based on ATM or Fibre Channel protocols.

The most common architecture consists of a single large switch connecting readout crates to farm event builder nodes. Requests for events are sent to all readout crates and all fragments of a given event are routed to a single destination.

These communications protocols were not designed for this type of application. It has been clear that problems, such as cell loss, could occur with the ATM protocol. Much work [4] has been done by the RD31 project to understand use of ATM for event building. This paper studies architectures and performance of Fibre Channel based event builders.

## AN EVENT BUILDER MODEL

The model for one stage of a Fibre Channel event builder is shown in figure 1. The model has *srcCount* input nodes. These nodes collect data from a previous stage of DAQ. These

nodes could be readout crates and the data could be from level two in a detector like ATLAS for LHC. The input nodes send all fragments for a given event to a single destination node.

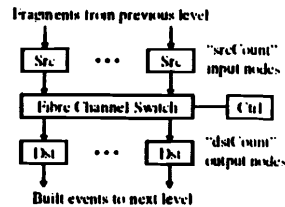


Figure 1: Block diagram of a Fibre Channel event builder

There are *dstCount* output nodes. These nodes collect all fragments for a given event and send this "built" event to the next stage of DAQ.

The routing of events to destination nodes may be dynamic or static. In the case of dynamic routing, the control node accepts requests for events from destination nodes. The control node sends the destination for each event to all source nodes. It is assumed that this control function is efficient and does not impact performance. For example it could be implemented by broadcast messages.

A predetermined algorithm is used for static routing and no control messages are required to implement it. The example of round robin destinations will be discussed below.

It is assumed that data is transferred by Fibre Channel class one connections. Source nodes use "camp on" mode which means they block if a given destination is busy receiving data from another source. This model can also be used for HiPPI switches. It is assumed that error free delivery of data is provided by the Fibre Channel protocol.

After a connection is established, the transfer time for a message is assumed to be a linear function of the message size.

$$\text{time} = \text{deadTime} + \text{size}/\text{linkSpeed}$$

The parameters for this function are a per message overhead or *deadTime* in usec and a *linkSpeed* in MB/sec which is the asymptotic speed for large messages.

## A MODSIM IMPLEMENTATION

A queuing model for the Fibre Channel event builder has been developed. This model has been implemented as MODSIM objects. The model with the main objects is shown in figure 2.

The EventGenObject is the control object for the simulation. It controls how many events are generated, their size and interarrival time distribution. Exponential and constant distributions have been implemented for interarrival time and size. This object also implements the destination algorithm for event fragments. Two algorithms have been implemented for destinations. Dynamic load balancing has been implemented by drawing destination nodes from a random distribution. Static load balancing is implemented by a round robin algorithm that cycles through destination nodes.

The SourceNodeObj represents buffers in source nodes. It maintains event fragment queues and collects much of the performance data.

The DstNodeObject implements the switch contention algorithm and the link model.

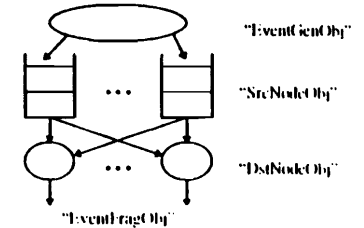


Figure 2: A queuing model of event building

Four contention algorithms have been implemented. They are fifo, priority, random and round robin. The fifo algorithm selects the source node with the message that has been queued for the longest time for a given destination node. The priority algorithm selects the node with the lowest node ID that has a message queued for a given destination. The random algorithm draws a random source node with a message queued for a given destination. The round robin algorithm is implemented by each destination node. Each destination node scans source nodes in a circular fashion looking for messages with its destination.

The EventFragObj contains routing data and information about the event that is used to gather performance statistics.

## VERIFICATION OF THE MODEL

The model was verified by using measurements of HiPPI data performed by R. Spiwoks [5]. Measurements of throughput vs. message size have been performed with two sources sending messages to a single destination. The measurements have been done using three software algorithms. The link parameters,

deadTime and linkSpeed have been determined for messages of four KB and greater. The linkSpeed parameter is 40.5 MB/sec for each of the datasets. The deadTime parameter has values of 75, 113 and 401 usec for the respective datasets.

A MODSIM simulation was performed using these parameters. The results are shown in figure 3.

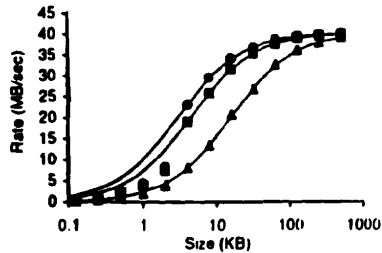


Figure 3: measured data vs. simulation

Agreement for messages four KB and larger is very good. The measurements for smaller messages had an additional software overhead and so a good fit is not expected.

#### SINGLE STAGE 256x256 PORT MODEL

The model has been used to simulate a large single stage event builder. This model has 256 source nodes and 256 destination nodes. The mean event size is two MB and the interarrival time distribution for events is exponential. The mean event fragment size is eight KB with an exponential size distribution. The maximum link speed is 40 MB/sec with a message overhead of 100 usec. The switch contention algorithm is FIFO. The destination algorithm is random to simulate dynamic load balancing. 5000 events have been simulated for each event rate.

Throughput vs. offered load is shown in figure 4. This event builder has a maximum throughput of about 1200 event/sec. The link efficiency is 24% at this rate.

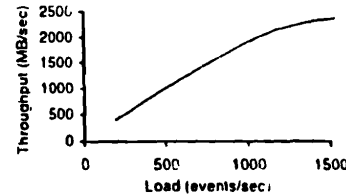


Figure 4: throughput vs. load for a 256 x 256 port single stage event builder

In addition to link efficiency, there are several other potential problems with this architecture. The cost and availability of large Fibre Channel switches is still an open question. The use of a single large switch presents system development and integration problems. This architecture requires large buffers and complex control in the readout crates. None of these problems are fatal. The performance of a large single stage event builder depends strongly on the distribution of event fragment sizes. This study is based on an artificial exponential distribution.

#### TWO STAGE ARCHITECTURE

An architecture for a two stage event builder based on smaller switches is shown in figure 5. The first stage corresponds to subdetectors. One or more subevent builders are associated with each subdetector. The second stage consists of subfarms.

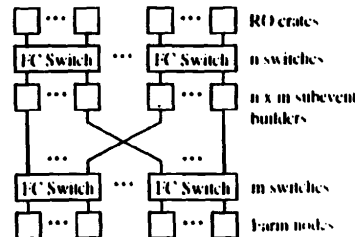


Figure 5: two stage event builder architecture

The first stage is assumed to use a static algorithm to route events to subfarms. The fraction of events sent to each subfarm is

determined by the total processing power of the subfarm. A round robin algorithm is used in the simulation.

Subfarms use a dynamic load balancing algorithm. Subfarm processors request events from a subfarm control node which forwards requests to subfarm source nodes which are subdetector subevent builder nodes.

If necessary, dynamic load balancing between subfarms could be achieved by sending complete events between subfarms over a single small switch that connects subfarm controllers.

#### A TWO STAGE EVENT BUILDER

Simulation of a two stage event builder of the same scale as the single stage event builder above has been carried out. Each stage of this event builder consists of 16 switches each having 16 x 16 ports. The values used in the previous simulation are used here for the following: link speed, link overhead, mean event size, event interarrival time distribution, mean event fragment size and size distribution.

The destination algorithm for the first stage is round robin. The algorithm for the second stage chooses destination nodes from a random distribution. A total of 10,000 events have been simulated at each event rate.

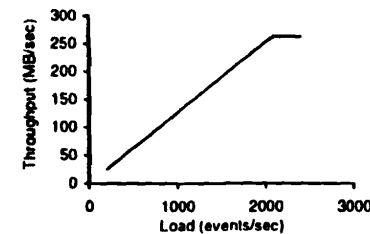


Figure 6: throughput vs. load for the first stage

The performance of one of the first stage 16 x 16 event builder is shown in figure 6. The maximum event rate is 2100 for a link efficiency

of 41%. This is about 75% better than the single stage event builder.

#### PERFORMANCE FACTORS

The improved performance of the two stage event builder over a single stage event builder is due to two factors, switch size and destination algorithm. These plus other performance factors are illustrated in figure 7.

The first factor is switch size. Large switches are not able to handle contention as well as small switches for the access pattern presented by event building. The curve labeled Rand in figure 7 has the same assumptions as the 256 x 256 switch. The maximum event rate is 1700 or 42% greater than the large switch.

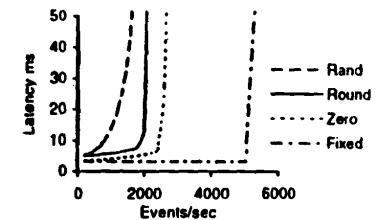


Figure 7: performance factors for a 16 x 16 switch

The second factor is the destination distribution. The curve labeled Round has the round robin distribution used in the first stage of the two stage model. The round robin distribution decreases destination contention since it has a more uniform time interval between events with a given destination.

Other factors that affect performance are link overhead, fragment size distribution and interarrival time distribution. The curve labeled Zero has the same assumptions as the Round curve except that the link deadTime parameter is zero. Eliminating dead time improves the performance, but not by the 50% expected. The final curve labeled Fixed is the same as the Zero curve but with fixed distributions for fragment size and interarrival time. This model

achieves 100% link efficiency. This shows how much performance depends on the fragment size and interarrival time distributions.

## SECOND STAGE PERFORMANCE

The performance of the second stage has been simulated with the following differences from the first stage. The destination algorithm is assumed to be random to simulate dynamic load balancing in a subfarm. The event fragment size is assumed to be 16 times as large and the event rate is scaled down by a factor of 16.

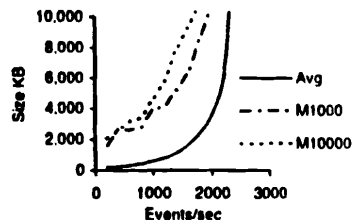


Figure 8: buffer size for a second stage source node

The result of this simulation is shown in figure 8. This figure shows the amount of buffer required as a function of event rate for a subevent builder node. The curve labeled Avg is the average buffer size for a source node. The curves M1000 and M10000 illustrate the variation of this size. The curve M1000 is the maximum buffer size for any source node during the first 1000 events. M10000 is the maximum during the entire simulation of 10,000 events. This shows that buffers must be much larger than the average size to prevent detector dead time due to inadequate buffering. Once again, performance is strongly dependent on the distribution of fragment sizes and interarrival times.

## SUMMARY

A queuing model has been developed for Fibre Channel and HIPPI based event builders. A MODSIM II simulation program for this model has been developed. This program has been

verified using measured HIPPI data. A large single stage event builder has been simulated. The single stage architecture is shown to have low link efficiency. A number of additional problems have been pointed out for this architecture.

An architecture for a two stage event builder has been developed. Simulation of this architecture shows that it is 75% more efficient than the single stage version.

Additional work needs to be done with more realistic event distributions from physics simulations to refine these results. A careful cost vs. performance analysis needs to be done for each architecture.

## REFERENCES

- [1] Vicky White, "Future Data Acquisition Architectures", Proc 8th Conf. on Computing in High Energy Physics, p 65-81, Santa Fe, NM, 1990.
- [2] W. Greiman, S. Loken and C. McParland, "Use of Commercial Gigabit Data Switches for SSC and LHC Event Builders", CHEP92, Ancey, France, pp. 184-187, September 1992.
- [3] W. Bozzoli et al, "High Performance Event Distribution Using HIPPI", CHEP92, Ancey, France, pp. 192-195, September 1992.
- [4] J. Christiansen et al., "NEBULAS: A high performance data-driven event building architecture based on an asynchronous self-routing packet-switching network", CERN / DRDC / 93-55 RD-31 Status Report, 22 December 1993.
- [5] R. Spiwoks, "Prototype of an Event Building System based on HIPPI", International Data Acquisition Conf. These proceedings, FNAL, October 26-28, 1994.

# Design and Simulation of Fibre Channel Based Event Builders

W. Greiman  
LBL

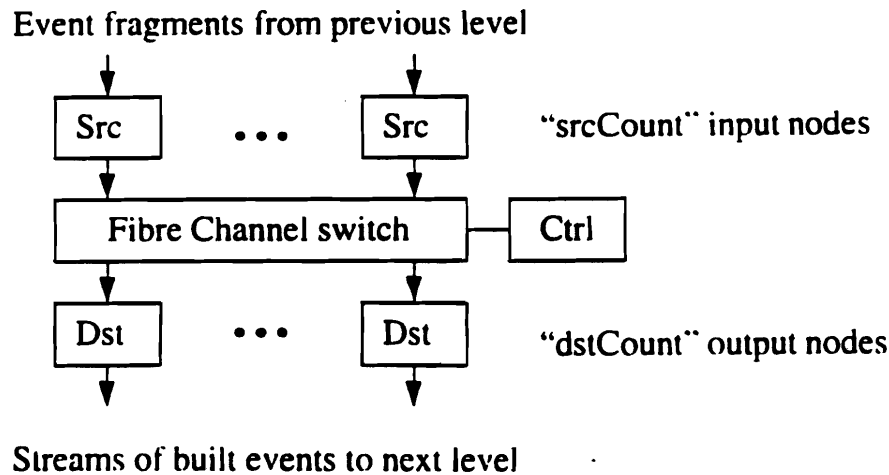
L. Mapelli, G. Mornacchi and R. Spiwoks  
CERN

October 1994

## Outline of Presentation

- Modsim model for Fibre Channel and HiPPI event building
- Validation of model using RD13 HiPPI measurements
- Simulation results for 256 x 256 port switch
- Architecture of a two stage Fibre Channel event builder
- Example of a two stage LHC class event builder
- Simulation results for the LHC class example
- Future work
- Summary

# Model for Fibre Channel Event Builders



11/11/04

FNAL DAQ 3

## Model Assumptions

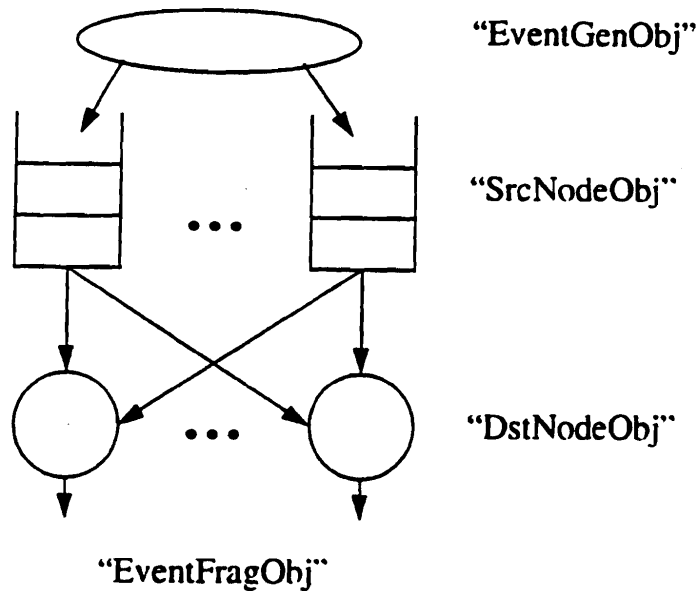
- Data transferred by Fibre Channel class one connections
  - Contention resolved by switch in “camp on” mode
  - Model can be used for HiPPI switches
- All fragments for a given event are sent to one destination node
  - Destination node is determined by the flow control algorithm
  - Error free delivery by Fibre Channel protocol
- Link transfer time is a linear function of transfer size

11/11/04

FNAL DAQ 4



## Queuing Model Implemented by MODSIM Objects



10/18/93

FNAL DAQ 5

### Role of Objects

- “EventFragObj” contains event description and routing data
- “EventGenObj” is the control object for the simulation
  - Event interarrival time and fragment size distributions: Exp. Fixed
  - Control mode for event destinations: Random. RoundRobin
- “SrcNodeObj” maintains fragment queues and performance statistics
- “DstNodeObj” implements the link model and switch contention
  - Link model based on max link rate and dead time
  - Contention models: Fifo. Priority, Random. RoundRobin

10/18/93

FNAL DAQ 6

## Summary of Program Arguments

Arguments can be on the command line or in an argument file

- a Interarrival time distribution (Exp, Fixed)
- c Contention algorithm (Fifo, Priority, Random, RoundRobin)
- d Link dead time (REAL usec)
- e Event fragment size distribution (Exp, Fixed)
- f Name of argument file (UNIX filename)
- i Number of input nodes (INTEGER)
- l Maximum link speed (REAL MB/sec)
- n Number of events to simulate (INTEGER)
- o Number of output nodes (INTEGER)
- r Mean event rate (REAL events/sec)
- s Mean event fragment size (REAL bytes)

10/18/94

FNAL DAQ 7

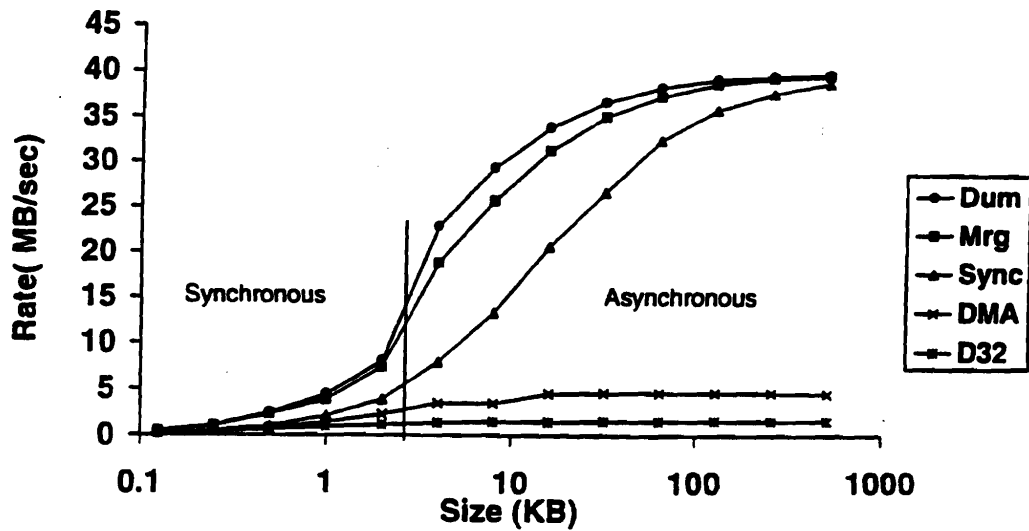
## Validation of the MODSIM Model

- Measurements of HiPPI performance at RD13 by R. Spiwoks
  - HiPPI switch with two sources sending to one destination
  - Throughput as a function of message size
- Determine linear fit to link performance
- Perform MODSIM simulation and compare results to measurements
- Additional measurements needed

10/18/94

FNAL DAQ 8

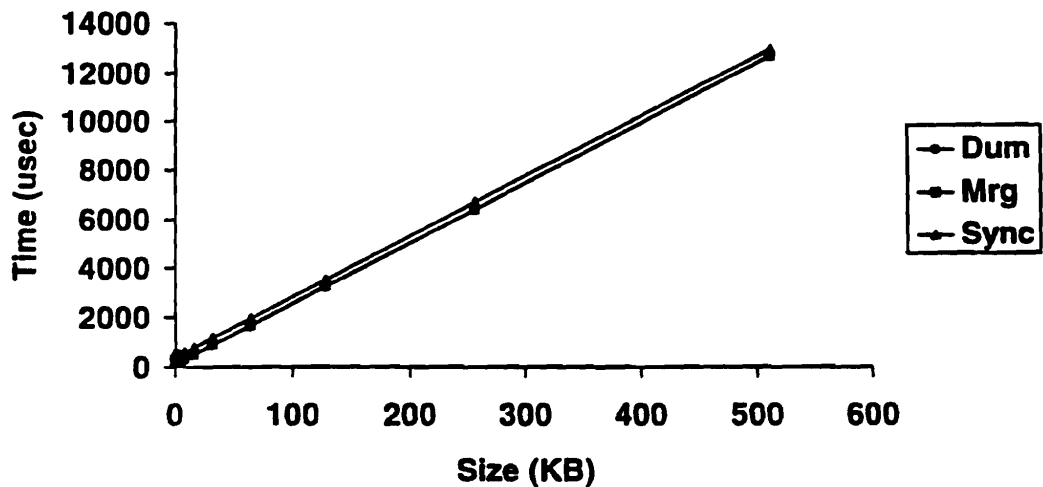
## HiPPI Performance by R. Spiwoks



10/18/94

FNAL DAQ9

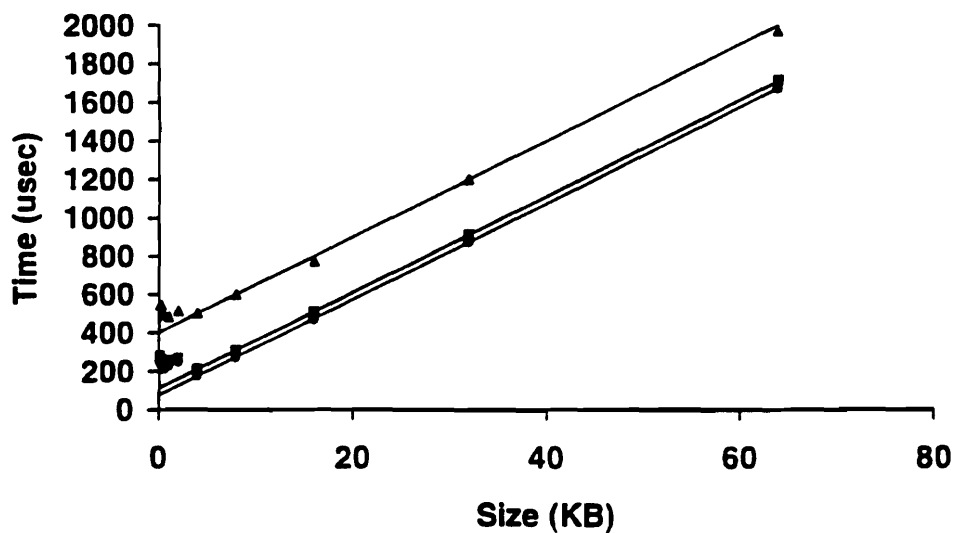
## Time per Message for HiPPI Data (R. Spiwoks)



10/18/94

FNAL DAQ 10

## Linear Fit to HiPPI Data



10/18/94

FNAL DAQ 11

## Parameters for Linear Fit to HiPPI Data

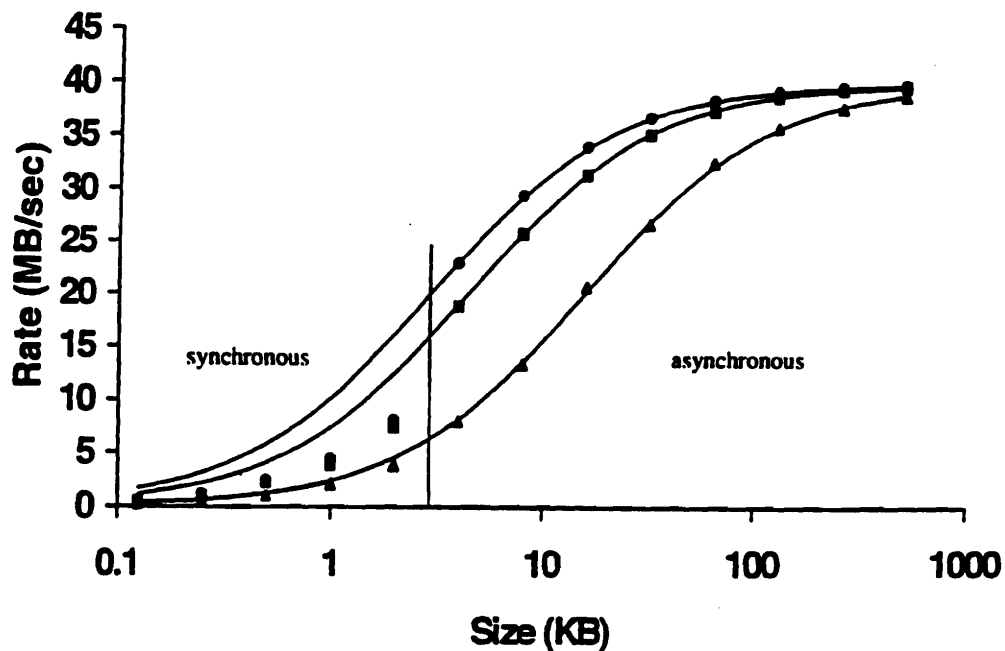
$$\text{time} = \text{deadTime} + \text{size}/\text{linkSpeed}$$

	Dummy	Merge	Sync
Link Speed (l) MB/sec	40.44	40.46	40.55
Dead Time (d) usec	75.00	112.77	401.35

10/18/94

FNAL DAQ 12

## Modsim Model vs HiPPI Data by R. Spiwoks



10/18/94

FNAL DAQ 13

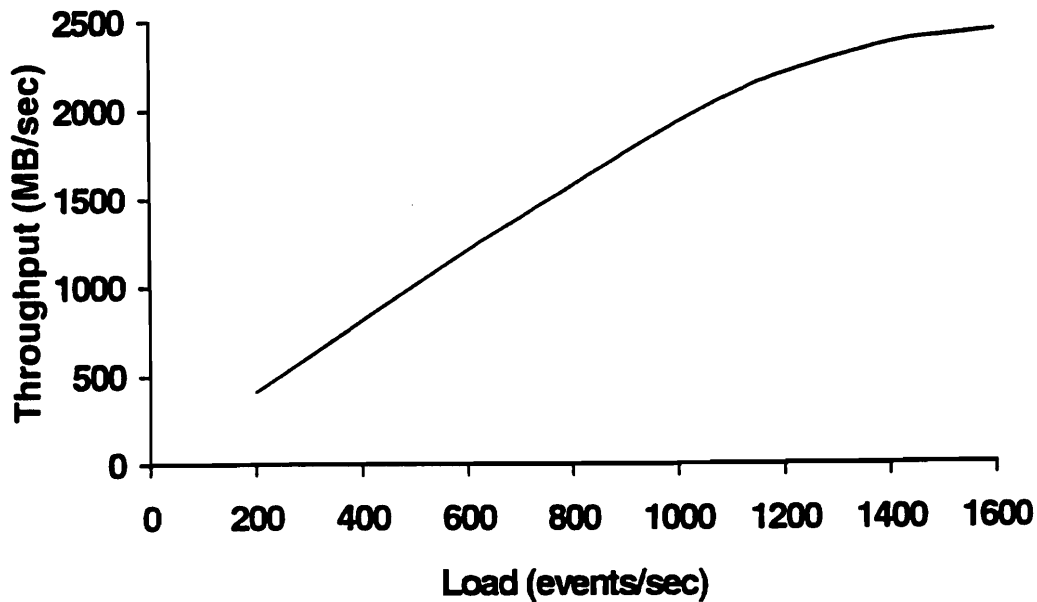
## Simulation of a 256 x 256 Event Builder

- Event builder has 256 source nodes and 256 destination nodes
- Mean event size 2MB, exponential interarrival time distribution
- Mean fragment size 8 KB with exponential size distribution
- Max link speed is 40 MB/sec with 100 usec dead time
- 256 links at 40 MB/sec each implies upper bound of 5000 events/sec
- 5000 events simulated at each event rate
- Destination node for each event from uniform random distribution

10/18/94

FNAL DAQ 14

## Throughput vs Offered Load - 256 x 256 Ports



10/18/94

FNAL DAQ 15

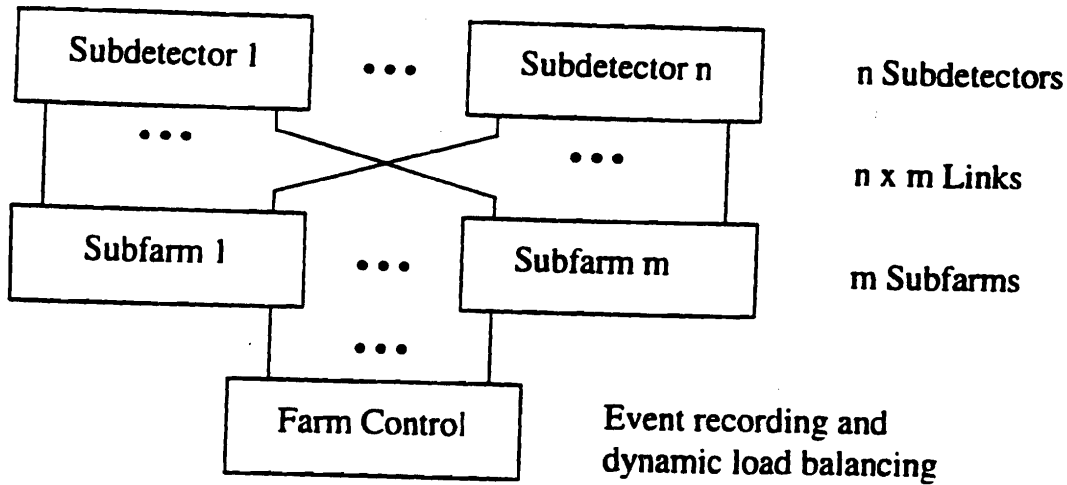
## Comments on 256 x 256 Event Builder

- Very low link efficiency, less than 25%
- Question of cost and availability of suitable Fibre Channel switch
- Use of single large switch presents system development and integration problems
- Large buffers and complex control required in readout crates
- Above are not "fatal"
- Performance depends strongly on distributions of event sizes

10/18/94

FNAL DAQ 16

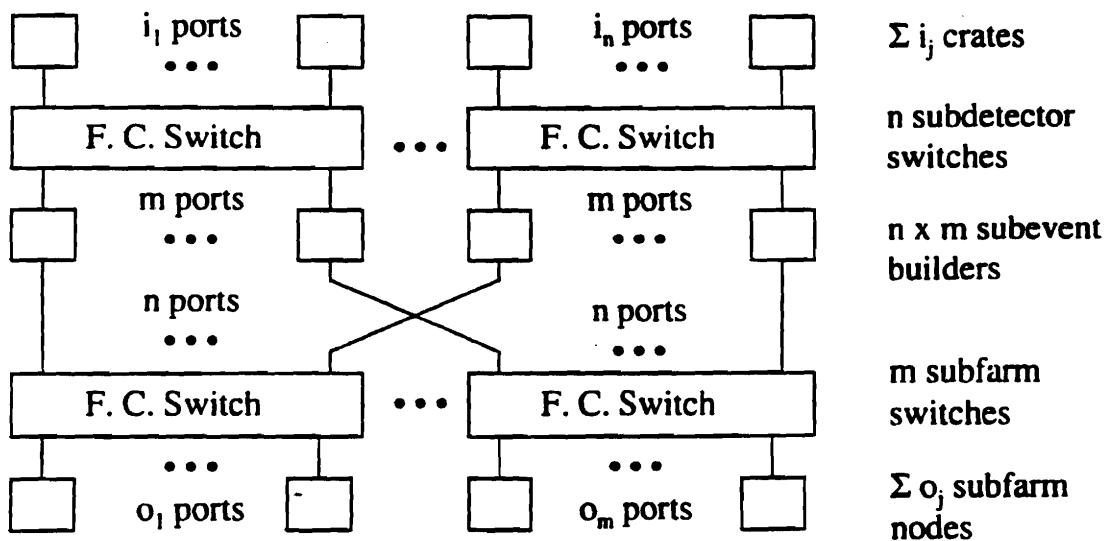
## Two Stage Event Builder Architecture



10/18/94

FNAL DAQ 17

## Fibre Channel Switch Architecture



10/18/94

FNAL DAQ 18

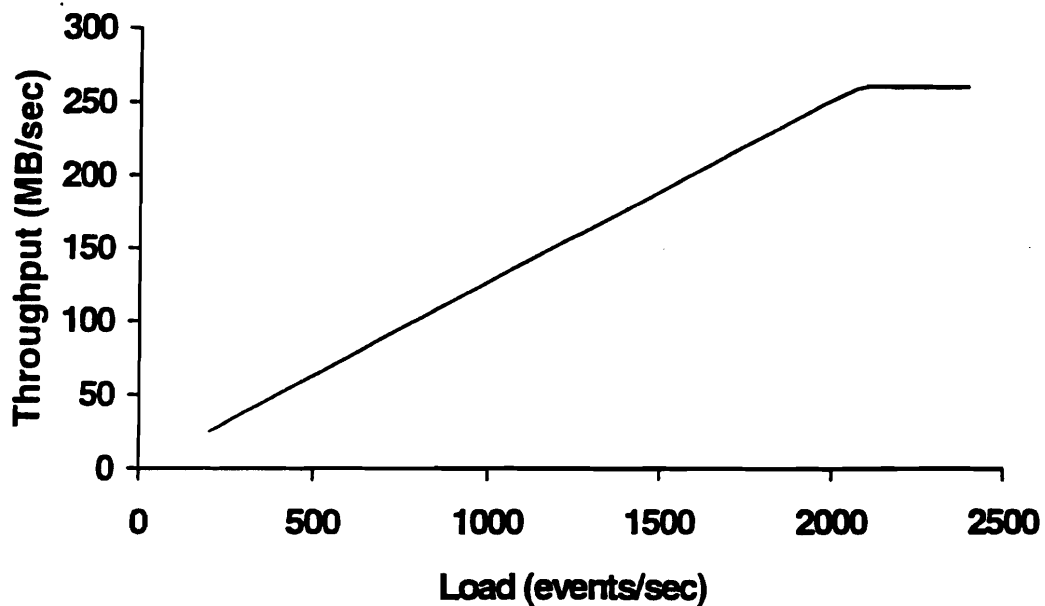
## A 256 x 256 Port Two Stage Event Builder

- Each stage consists of 16 switches each with 16 x 16 ports
  - Mean event size 2MB, exponential interarrival time distribution
  - Stage one mean fragment size 8 KB with exponential size distribution
  - Max link speed is 40 MB/sec with 100 usec dead time
  - 256 links at 40 MB/sec each implies upper bound of 5000 events/sec
  - 10000 events simulated at each event rate
- 
- Destination node for events in first stage is round robin
  - Destination node for second stage is from uniform random distribution
  - Stage two mean fragment size 128 KB, exponential size distribution

10/18/94

FNAL DAQ 19

### Throughput vs Offered Load - First Stage Switch

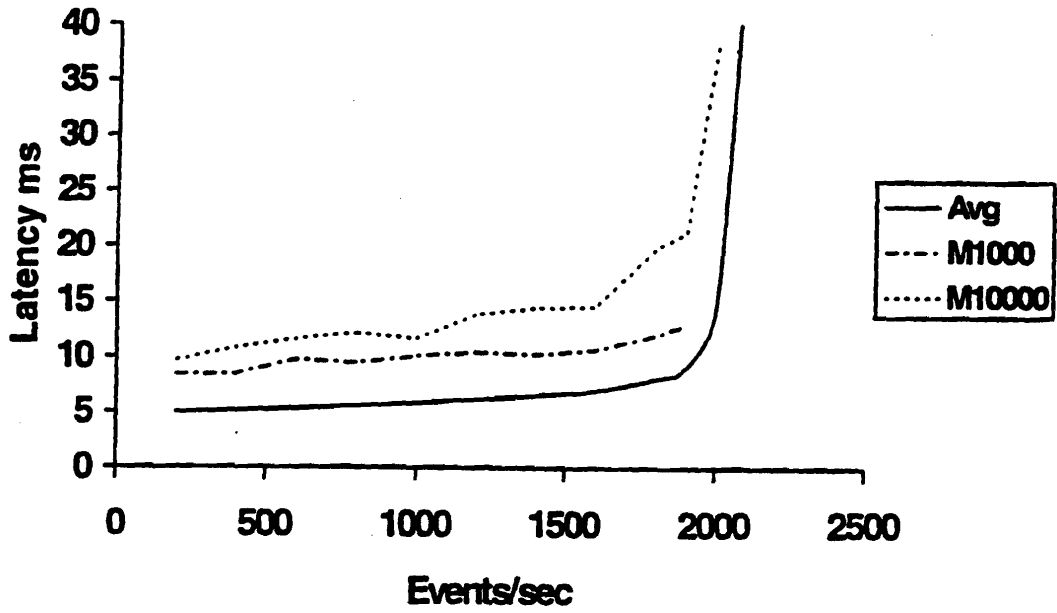


10/18/94

FNAL DAQ 20



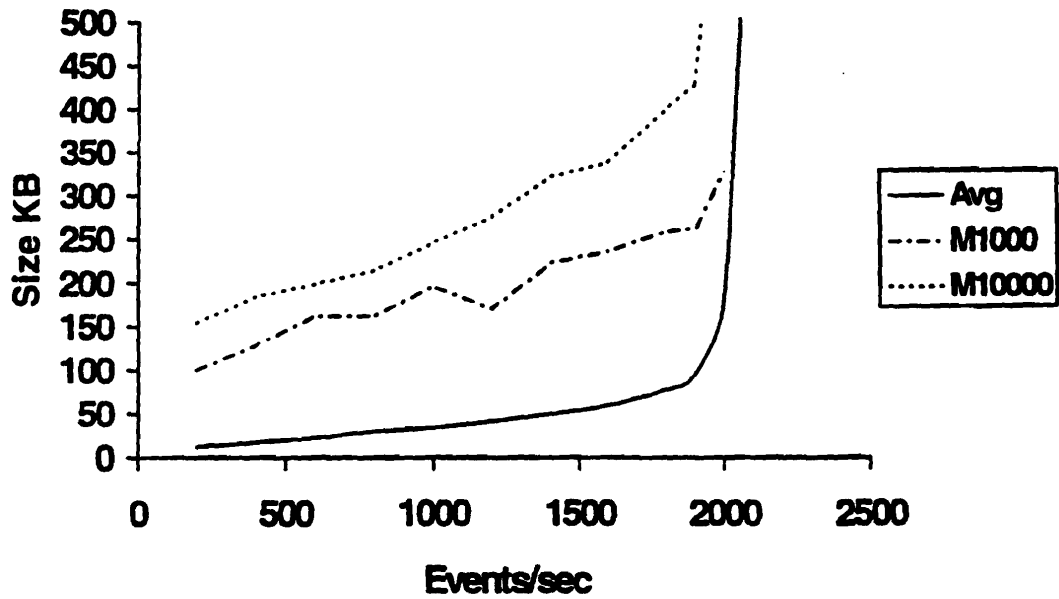
## Latency - Crate to Subevent Builder



10/18/94

FNAL DAQ 21

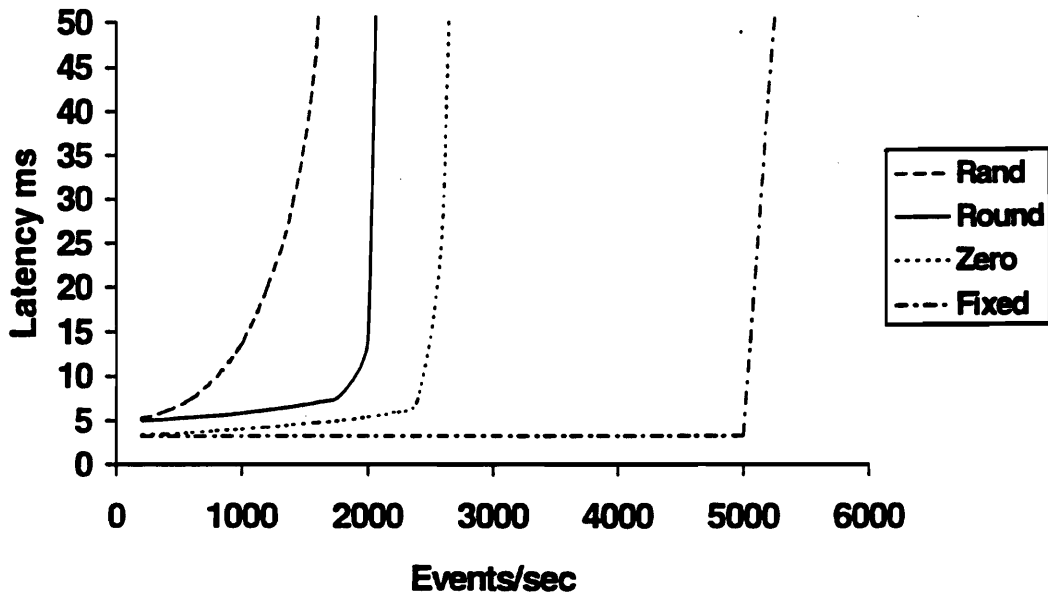
## Crate Buffer Size



10/18/94

FNAL DAQ 22

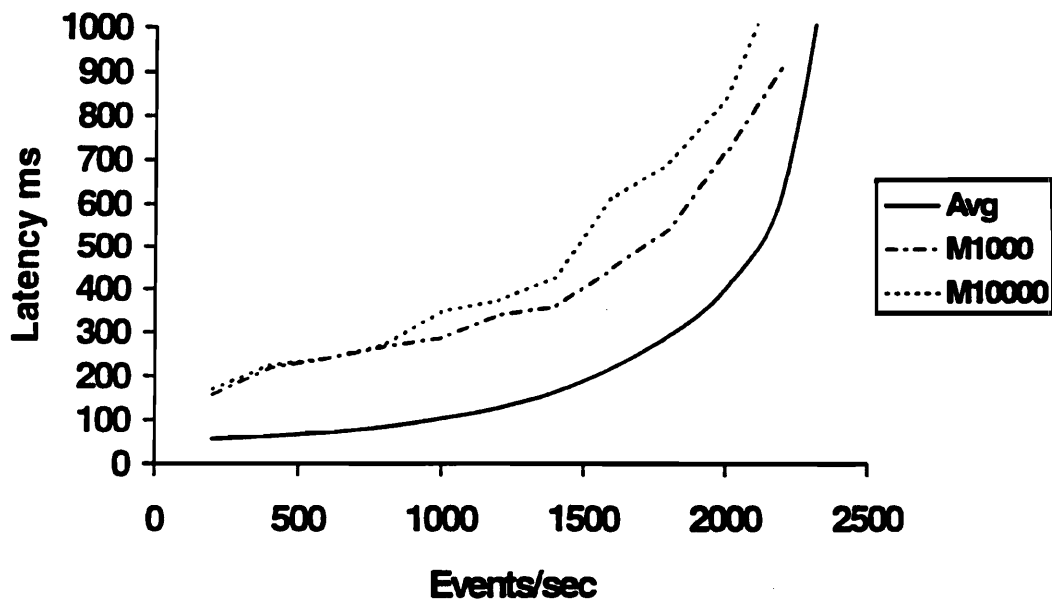
## Latency vs Parameters - 16 x 16 Event Builder



10/18/94

FNAL DAQ 23

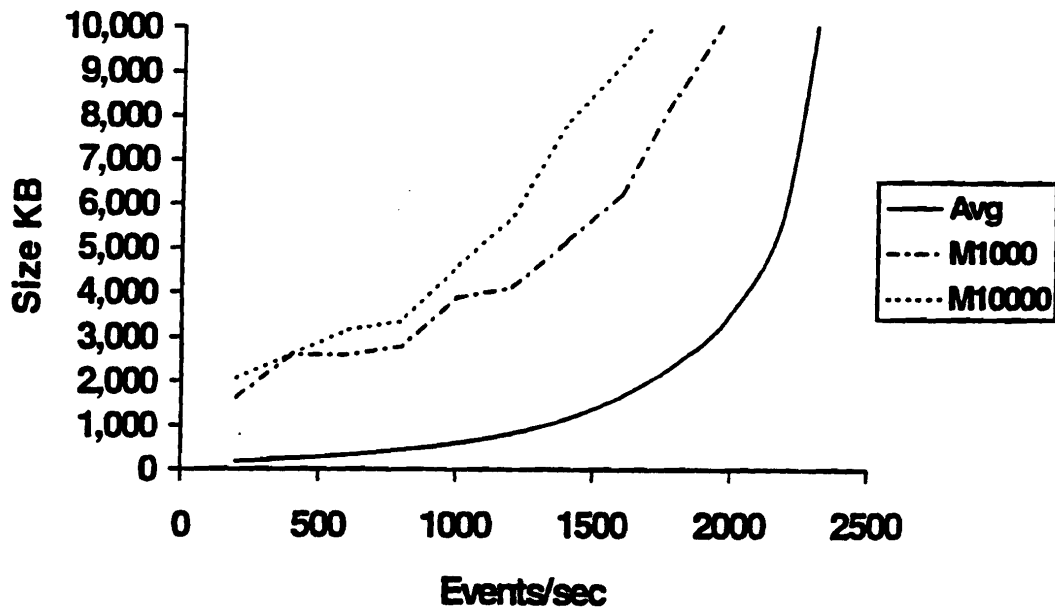
## Latency - Subevent Builder to Subfarm



10/18/94

FNAL DAQ 24

## Subevent Builder Buffer Size



10/18/94

FNAL DAQ 25

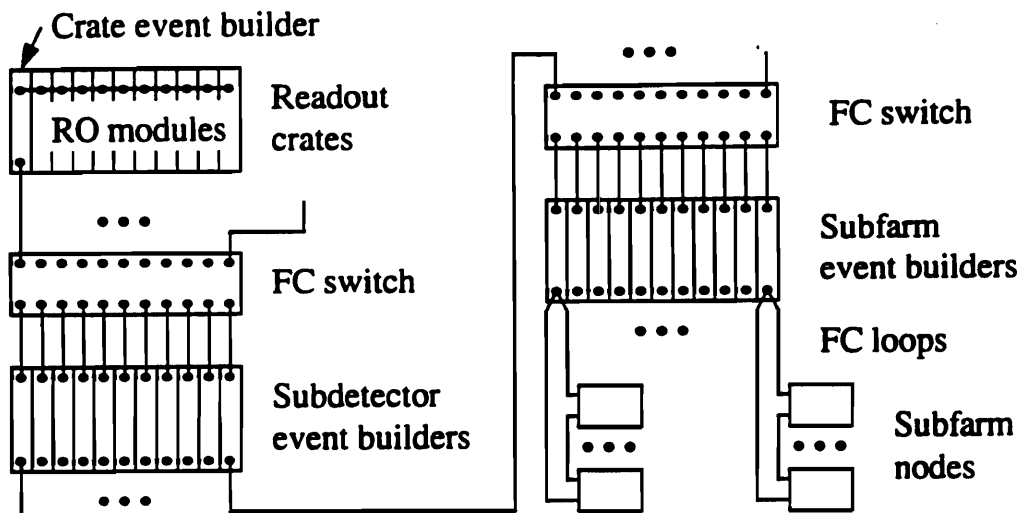
## Future Work

- Verify and develop model with additional lab measurements
- Recode prototype MODSIM program for production use
- Accept fragment arrival time and size from physics simulation
- Write time and size file for input to second stage of event building
- Improved output statistics
- Trace of simulation for debug
- Better user interface
- User documentation
  
- Hardware: develop event builder node

10/11/94

FNAL DAQ 26

## Need Event Builder Node - NOT Interface



10/18/94

FNAL DAQ 27

## Summary

- A queuing model has been developed for HiPPI and Fibre Channel event builders
- A MODSIM simulation program for this model has been implemented
- Initial verification has been done using measured HiPPI data
- A single stage 256 x 256 event builder has been simulated
- An architecture for a two stage event builder has been developed
- A LHC class example of this architecture has been simulated
- A two stage event builder is about 70% more efficient than a single stage event builder

10/18/94

FNAL DAQ 28

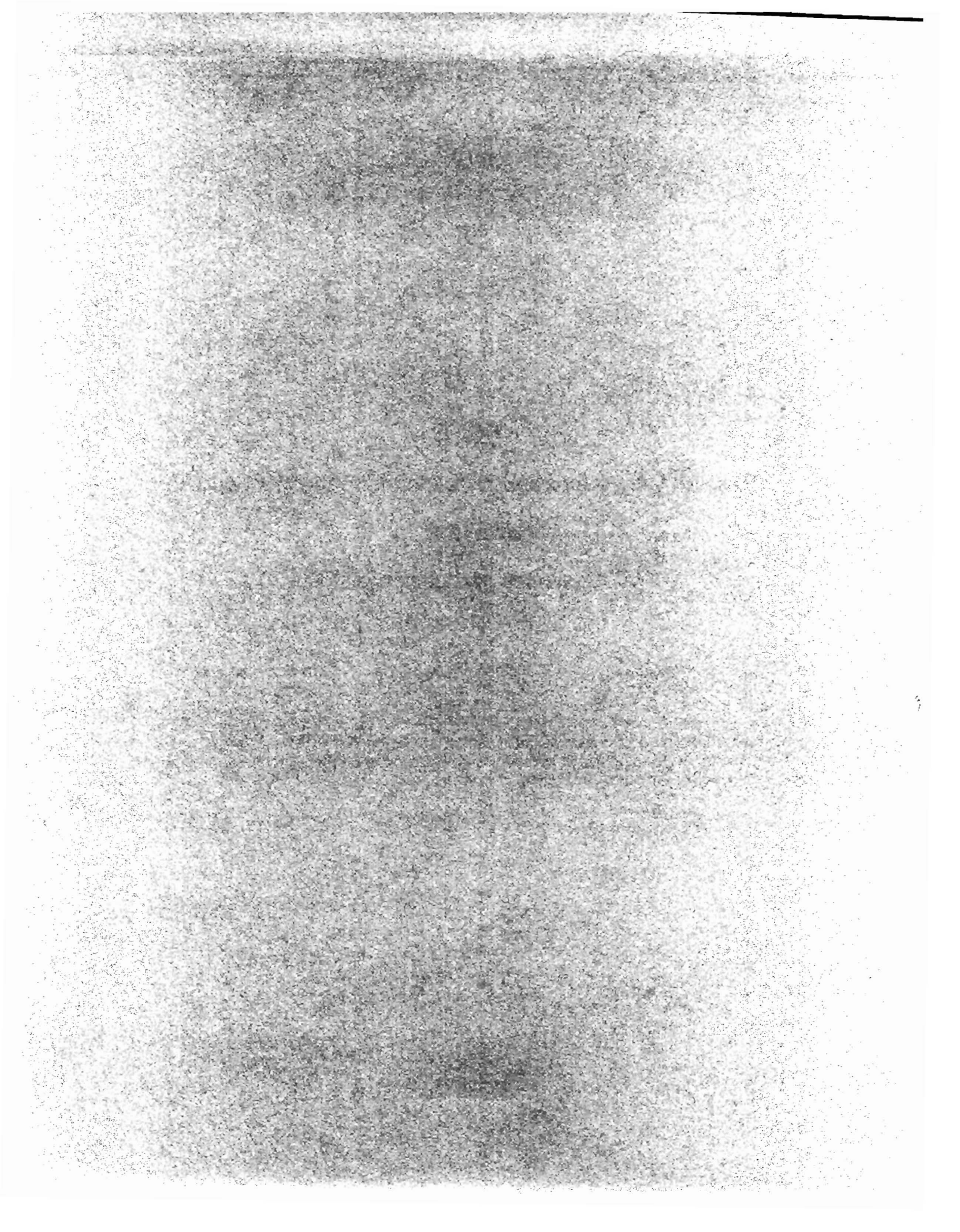


**S8-2**

**"Pros and Cons: Commercial & Non-Commercial Switching  
Networks"**

**(Alexandro Marchioro - CERN)**

Several packet switching network architectures have been proposed as alternatives to conventional bus-based read-out architecture for applications in High Energy Physics data acquisition systems. A hypothetical packet switching network called "Nebulas" optimized for HEP data acquisition systems is introduced and compared to commercial fabrics adapted for such usage. Benefits of the commercial solution are compared to advantages of an ad-hoc solution. It is shown that performance and cost might need to be more precisely defined before deciding for one or the other approach.



## Pros and Cons Commercial and Non-Commercial Switching Networks

A. Marchioro  
CERN / ECP-MIC

A. Marchioro 10/19/94

### What is wrong about the talk's title ?

- ◆ A complete data acquisition system is much more than a switching fabric

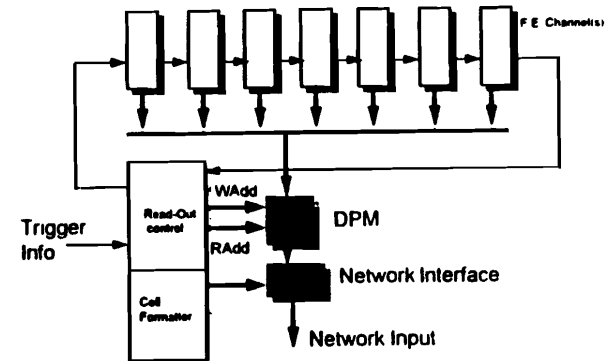
... However, it is important to keep in mind that the port controllers represent a vast majority of the cost of any switch. [...] Because switching logic design is intellectually more challenging than designing port controllers, lots of papers have been published about switch design but very few about port controllers

(C. Partridge, "Gigabit Networking", 1994)

## Pros and Cons Commercial and Non-Commercial DAQ Systems

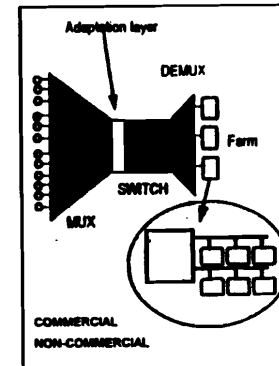
A. Marchioro 10/19/94

### Front-end subevent (packet) assembly



A. Marchioro 10/19/94

### Commercial subsystems approach

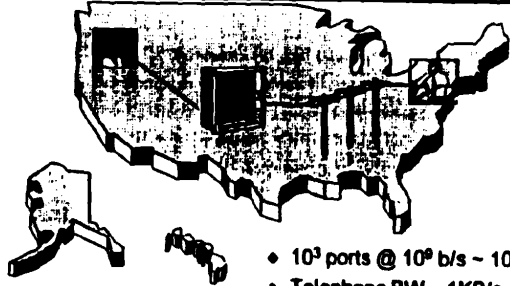


A. Marchioro 10/19/94

- ◆ ATM, Fiber Channel
- ◆ Commercially available sub-systems (components)
- ◆ Need adaptation layer in input
- ◆ Event assembly (packet re-ordering) is still necessary in output
- ◆ Needs input MUX with very high output speed



### How big is a 1K by 1K switch @1Gb/sec ?



- ◆  $10^3$  ports @  $10^9$  b/s ~  $10^{12}$  b/s
- ◆ Telephone BW ~ 1KB/s
- $10^8$  simultaneous phone conversations

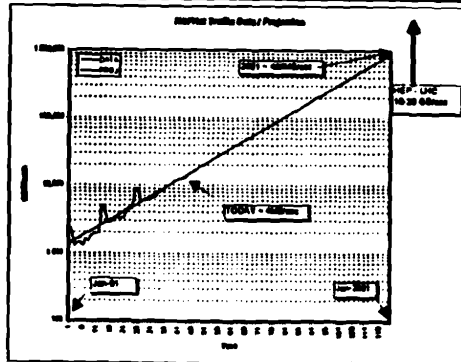
A. Marchetti 10/10/94

### Problems with square switches

- ◆ Switch is only used for event building and not for event collection
- ◆ Requires front end multiplexing stage
  - The fabric might well be scaleable with time but what about the adapters ?
- ◆ Not designed for data acquisition systems
  - Designed mainly for LANs or WANs
  - No need for bi-directional links
  - Not well adapted to M x N setups
- ◆ Protocols
  - ATM from telecommunications
  - Fiber Channel: for computer-to-computer links and disk I/O
  - High-speed video distribution (one to many)

A. Marchetti 10/10/94

### Really commercial ?



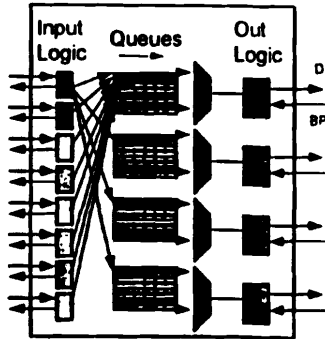
A. Marchetti 10/10/94

### Commercial vs. In house

	Commercial (ATM Telecom)	In house (Physics DAQ)
Optimized for	Telecom	Physics DAQ
Off the shelf	Yes, but only part	No
Protocol	Standard	In house
Standard	Yes, but not for ev. building	Hopefully! How about SCSI (FC)?
Computer Int	Yes, but what about the adapters ?	Only if planned
Scaleable	Yes, but what about the adapters ?	

A. Marchetti 10/10/94

## NEBULAS: Switch Organization



- 16 KB buffer in switching node (32\*8\*64 bytes)
- 80 MB/s port, to match BW to computer
- Packet format - 64 byte fixed length  
 $DST_{16}, SRC_{16}, Cont_{16}, Event_{16}, Payload_{32}, CRC_{16}$
- Non-head-of-line-blocking protocol
- Speed
  - 640 Mb/sec on 8 bit bus
  - 0.8  $\mu$ s / packet

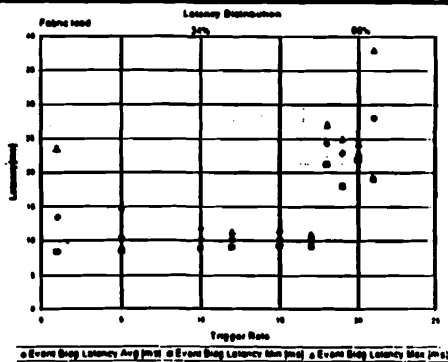
A. Marchese 16/10/94

## What to look for ?

- We need a unified approach to data acquisition:  
*"Connect your channels to the RO system, like your workstation to Ethernet"*
- Conic Banyan Fabric compares favourably with ATM switch and it does full event building
- No decision for the switching fabric itself should be taken until ( $t_0 - 5$ ) years
  - I/O interfaces and protocols could be done in a fabric independent way before
- The DAQ architecture (requirements) should instead be understood pretty soon
  - understand traffic patterns
  - different detector's requirements

A. Marchese 16/10/94

## Modeling results



A. Marchese 16/10/94

# Pros and Cons of Commercial and Non-Commercial Switching Networks

I. Mandjavidze, A. Marchioro / CERN - ECI\*

## ABSTRACT

An hypothetical packet switching network called Nebulus optimized for HEP data acquisition systems is introduced and compared to commercial fabrics adapted for such usage. Benefits of the commercial solution are compared to advantages of an ad-hoc solution. It is shown that performance and cost might need to be more precisely defined before deciding for one or the other approach.

## INTRODUCTION

Packet switching networks have been proposed as alternatives to conventional read-out architecture for applications in High Energy Physics data acquisition systems. This initiative of engineers around the HEP environment, occurs at the same time as a similar move in the computer and in the telecommunication industry. High-speed, relatively short distance interconnections across computers and between computers and shared peripherals are demanding the introduction of high performance switching fabrics from data-communication manufacturers. long-distance interconnection for telephony and digital television broadcasting also requires switching elements capable of routing packet based traffic efficiently across many I/O ports. It would certainly be appealing to be able to use some of these technologies made available by industry to build read-out systems for HEP. This paper discusses the matching of commercial architectures to the technical requirements of an optimized read-out system for LHC class experiments.

## WHAT IS A DATA ACQUISITION SYSTEM?

A typical LHC experiment will have to connect many million channels of front end-electronics to one (or few) thousand on-line-computers. Aggregate bandwidth requirements are expected to be in the  $10^{10}$  -  $10^{12}$  b/sec range (or  $10^7$  -  $10^9$  b/sec \* port). Not only do the event data from the different sources have to be collected together in one destination (this action is normally performed by a multiplexer) but also data of different events have to be directed to different destinations, requiring a routing capability, which in general could be provided by a fully interconnected matrix switch (crossbar). In the past such systems were normally built using a hierarchy of processors [Ref. 1], in which the multiplexing function was provided by a shared bus (CARLYC, Fast-bus etc.), mastered by one processor or hard-

wired data mover. All these architectures normally were of the pull type (ie. data were always read by controllers organized in a hierarchical arrangement).

The LHC front end electronics will very likely be organized in a similar manner (ie. electronic boards mounted on the detector and/or in electronics barracks) will house the front-end channels. It is reasonable to assume that the number of front end boards will exceed the number of available computer ports on the ALPH experiment at LEP about 100. Fastbus cards are read-out to a few on-line-computers. This demands a read-out architecture where multiplexing and routing from now on called event building, are somehow combined. While it is normal for commercial switches to have a square aspect ratio (same number of inputs and outputs), an LHC experiment will instead require a cone aspect ratio with M inputs and N outputs ( $M > N$ ).

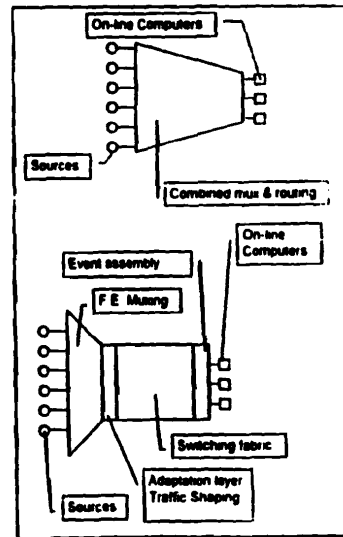


Figure 1. Generic read-out architectures

Such a system can basically be built as shown in FIG. 1. The first option shows an optimized fabric with a conical aspect connecting directly the front end-sources to the on-line-computers. Such an arrangement which will be the model for the Nebulus architecture, does not need any traffic adaptation between the

front-end electronics and the fabric. The second option uses a traditional data multiplexing arrangement in front of a square switch, where clearly the switch performs mainly the event building function (ie. routing and multiplexing) and the cone the input multiplexing (ie. that part of the multiplexing function needed to concentrate the data streams sufficiently to feed the inputs of the square switch). Notice that such a block diagram poses no restriction on the type of switch used (ie. different technologies such as ATM and Fiber Channel could in principle be used. Some of the commercial switches could also in principle be configured to work in an MSN arrangement like in the ATM case).

## WHY ARE DMQ TRAFFIC PATTERNS PATHOLOGICAL?

While traditional high performance computer buses were optimized primarily for speed with perhaps little attention to the specific type of traffic that could have been encountered in particular applications, today's commercial packet switching networks are conceived and highly optimized for the particular mix of traffic which is expected in real applications. As an example, the ATM hardware and software protocols were conceived from the beginning as a universal mechanism to support many different classes of services (real-time-voice, video etc.) and therefore it is a best compromise solution to cover a wide range of applications. The primary concern here was that packets be delivered in the shortest possible delay and in sequence, as it would be very inconvenient talking to your friend on the phone and get his/her words mixed up and in the wrong sequence. (Of course data protocols between computers are smart enough not to demand sequencing of packets by the network user (for example the TCP/IP protocol), and they instead are more interested in data throughput rather than short latency. In addition, such telecommunication equipment normally has many thousand (K) ports, ie. switches are very large. In Telecom applications a low packet loss probability is also acceptable, because the physical medium is anyway likely to introduce some losses on its side, due to noise in long distance connections. Finally, it is very likely that the type of interconnections established between users would always be of the one-to-one or one-to-many type).

The interconnection between computers and computers and shared peripherals (like the Fiber Channel standard) demands instead high throughput and very robust data transfer protocols. It must be bi-directional and is normally transferring large chunks of data in an bursty fashion, ie. relatively long setup times are acceptable to establish a connection between two subscribers. The number of ports of such a switch is also normally more modest than the one used in telecommunications

connections between subscribers is always one-to-one or one-to-many for a given transmission.

Event building requires sending data from many sources to the same destination concurrently. Such a peculiar traffic pattern (many to one) can not be sustained by ATM networks designed for Telecom and trucks must be introduced [Ref. 2]. In particular, as some ATM switches do not have a link-level flow-control back-pressure capability, a highly correlated traffic pattern is extremely dangerous even at very low aggregate network utilization (ie. 1). Breaking the time-correlation between packets traveling to the same destination can be achieved by either introducing a global control on the input sources or more simply by randomizing the injection times of data packets in the fabric at the source level. These techniques are commonly referred to as traffic shaping. Although these techniques could be considered as viable at low and medium network speed, this becomes a prohibitively complex operation at high link rates, as very short time is available to select an appropriate cell from the source buffers (for instance at 2.4 Gb/sec, only about 200 ns are available to pick up an appropriate cell from the source buffers in a pseudo-random manner).

	ATM	IP	SD	Nebulus
Isosceded	Yes	No	No	Yes
Bi-directional	Yes	Yes	Yes	Yes
Need no congestion	Yes	Yes	Yes	Yes
Traffic shaping	Yes	Yes	Yes	Yes
Precedence	Yes	Yes	Yes	Yes
Packet size	53-9232	variable	variable	variable
Interconnection type	one to one	one to many	one to many	one to many
Industry support	Yes	Yes	Yes	Yes

Table 1. Main characteristics of different types of switching network.

In addition to the adaptation hardware in input, one also needs to provide intelligence at the output of the fabric capable of assembling the incoming events in one complete event structure to be passed to the on-line computer for analysis. This is because the fabric will deliver packets from a given source in sequence but will mix packets from different sources belonging to the same event. The event assembly engine could also prepare data structures optimized for the on-line filtering by constructing tables pointers etc. It is conceivable that such a piece of electronics could be built more economically with a dedicated simple interface processor (or hardware controller) rather than using the on-line computer itself for the 10<sup>9</sup> intensive task of assembling an event.

All such adaptation layers are special to our application and must be designed and built outside the commercially available subsystems.

## NEBULAS

Assuming that one had complete freedom in building an optimized switching fabric, we could optimize all design parameters, and a proposal such as the one illustrated here could be conceived. Front end cards could be located either on the detector itself, and therefore optical connections would be used to connect to the switching fabric or in conventional electronic crates not too far from the switching fabric, from which copper cables or optical fibers could be used.

First of all, one would match the number of inputs to the number of front end-cards directly, avoiding the need for a separate multiplexing stage, that indeed looks very much like a traditional data acquisition system in front of a switch. The number of outputs will match the number of available computers. In addition, the outputs would provide no more bandwidth than a single computer can absorb. The network would provide directly both routing and multiplexing functions in one common architecture. In this example, we will assume a 8192 to 1024 network. Such a network can easily be scaled to any other configuration up to 32768 to 1024 ports. It is assumed that the connectivity will that of a Banyan network.

The key element of the network, i.e. the switching element, could be optimized also. As a reference, we will choose an 8 to 4 switch as illustrated in Fig. 2. This switch contains 32 queues, i.e. 4 sets of 8 queues, one set for each output port. Each queue is in turn 8 cells deep. Each output port controller (4 per switch) selects in a round-robin fashion from the 8 input queues the next data-cell to be sent. Such a switching element is rather ambitious to design in ASIC using today's technology (requires more than 16 KB of internal memory) but could certainly be designed with a submicron CMOS technology available to HEP in a few years.

The studies done until now assume a fixed packet length of 64 bytes and no special packet prioritizing mechanism in the switch. If the fabric has to combine both Level-2 and Level-3 traffic, it is conceivable, if the necessity can be shown by modeling, to introduce a priority scheme in which the Level-2 cells are given priority in the internal queues.

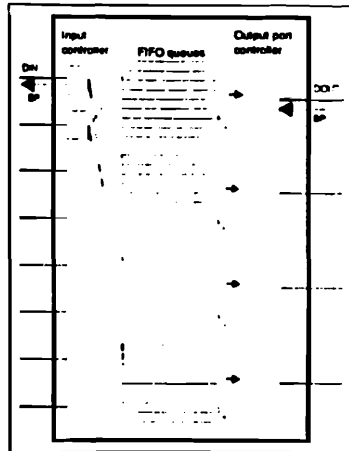


Figure 2. The studied 8x4 switching element for Nebulas

As the availability of high speed links in the future and the date of an LHC experiment are highly speculative today, performance simulation results will be given for the example of a 622 Mb/s link between nodes. (This bit-rate actually assumes using commercial 10-12 fiber-optic components at 622 Mb/s, where necessary.) The entire Nebulas fabric also works at the same speed, i.e. inside and at the boundary ports, unlike some of the commercial switches that need to boost their internal speed to sustain a given I/O rate.

Housing 16 switching elements on a card, an entire 8192 to 1024 fabric could be built on roughly 250 boards, i.e. about 20 large crates of electronics, the major problem being the interconnections between the boards. This space is much smaller than the one needed to house the 1024 on-line computers (even in pizza boxes).

A very simple protocol for transferring data packets could be used. For instance, each packet could just contain a 16-bit destination field, corresponding to the destination computer number and no higher level data structure is necessary for the fabric. Events at the destination could be built using a simple 'time-out' protocol and by a fabric mechanism that reports (eventually with delay) packets which have got confused in the fabric.

A scheme supporting automatic re-routing of cells through the network in case of faulty switching elements or connections has also been proposed, but will not be discussed in detail in this paper.

## NEBULAS PERFORMANCE

The Nebulas network has been studied extensively by using two fully independent simulation programs. Several network sizes ranging from 10x4 up to 1024x256 have been simulated. Currently the complete network of 8192x1024 ports can not be simulated easily on available computer equipment, as it requires more than one week of simulation time for each configuration.

Some work is underway in order to parallelize the code to be able to run the full network by the simulator on a multi-processor machine.

The results below are shown with an event generator having an exponentially distributed inter trigger delay and a number of data cells per source distributed also according to an exponentially decreasing distribution.

It must be noted that such a distribution is highly unrealistic in reality where different subdetectors will provide vastly different chunks of data. This will be taken into account once a better model of the event generator will be available to the authors.

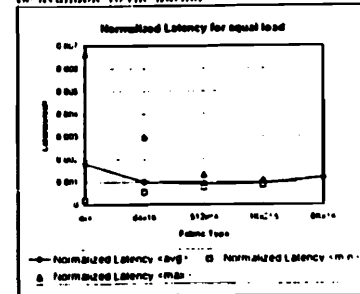


Fig. 3. Normalized latency as function of fabric type

The response of the network to different traffic characteristics are shown below. Fig. 3 shows the normalized event building latency (i.e. the total event building latency divided by the number of cells in the event) for different network fabrics all loaded at about 50%. Notice that the data point for the 8Kx1K fabric has been attained for only a 32% fabric load, due to limitations in simulation time. This shows that the average event building latency is independent of the fabric size and allows us to foresee that the 8192x1024 network will also behave properly at least up to a 50% load.

Fig. 4 shows the event building latency for a 512x64 network loaded at 50% as a function of the event characteristics. By modifying concurrently the event rate and size (to give approximately a constant aggregate bandwidth), it is shown that the average normalized event building latency does not depend on this characteristic of the traffic.

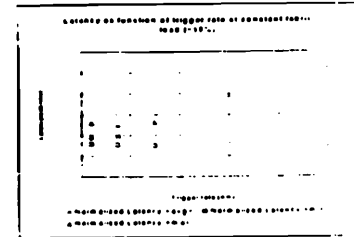


Fig. 4. Latency as function of traffic type in a 512x64 fabric (trigger rate  $\propto$  event size = constant).

Fig. 5 shows finally the event building latency for a 1024x256 fabric as a function of the load event rate for a constant event size of 670 cells. The fabric behaves very well up to a 60% load. Very strong non-linearities start above this load and it would probably not be safe to attempt to exceed this average load value in a real application. It should be noted that the event building latency under these circumstances is only about 20% larger than the best theoretical minimum value (i.e. the event building time for one single event going completely undisturbed through the fabric).

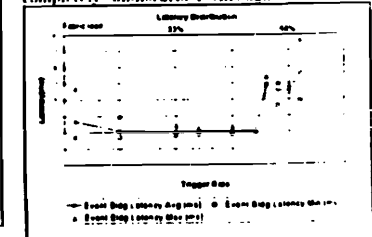


Figure 5. Latency as function of fabric load.

Some experiments proposed for LHC are studying the possibility to merge in the same network data for Trigger Level 3 read-out and data from partial events to allow computation in the on-line computers of the Level 2 trigger. Such traffic has been modeled in our Nebulas simulation under the assumptions summarized in Table 2.

	1k to 256	2k to 1k
L1 trigger rate	100 KHz	100 KHz
L1 trigger size	10 Kbit	10 Kbit
L2 data rate	4 KHz	4 KHz
L2 data size	4 Kbit	4 Kbit
Network load for L2 data	1%	1%
Network load for L1 data	20%	20%
L2 data sources	127	127
L1 data sources	127	127

Table 2. Summary of mixed traffic simulation in Nebulas

For the reasons explained above, only the 1024 to 256 case has actually been simulated and the results are summarized in Figure 6.

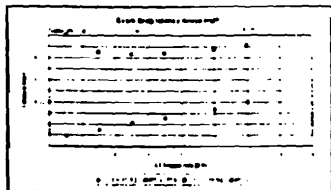


Figure 6. Latency as function of load for mixed L2L3 traffic.

From this, one can conclude that the network can withstand quite acceptably also the mixed traffic pattern. Such mixed traffic poses instead considerable problems in any fabric requiring traffic shaping as the latency of Level-2 events will be considerably increased.

#### LIMITATIONS

The results reported in this paper refer to a highly hypothetical traffic input. As mentioned above, all sources are assumed to provide data following an exponential decreasing distribution around a given average. When averaged over a large number of sources, total event sizes become distributed around a very narrow peak. A real experiment will instead clearly have different trigger types and related event sizes.

Preliminary simulation modeling results for these more realistic cases have shown that latency can be degraded seriously whenever large switching fabrics are used above 15% of their total bandwidth. This would indicate that indeed all switching fabrics (with and without flow-control) would need to be equipped with some sort of traffic shaping control, if one wants to use them at very high load. The trade-off of a higher bandwidth switching fabric used at relatively low load and without a traffic shaping stage compared to a slower fabric complemented with a relatively complex traffic shaping stage still has to be investigated more clearly.

#### DISCUSSION

Of course most engineers do not have infinite freedom when building a system. Cost compliance to standards for interfacing, availability of off-the-shelf spare parts, long term maintainability, software support are only some of the most important constraints he/she has to live with. All these factors are normally weighted against a potential performance improvement or cost reduction that can be achieved by introducing an ad-hoc solution.

But what is performance? Several users can define performance in different terms. For instance performance for telecommunication industry is related to low latency, while this might not be very important for physics traffic. Capability of sustaining mixed traffic (L2 and

L3 data) could be more important for LHC experiments. Performance could also be defined in terms of capability to support a given traffic load to use efficiently the fabric. Several ways of defining performance could demand optimization for a non-consistent set of parameters.

In addition, the definition of real costs still needs to be investigated, as complex adaptation layers between commercial sub-systems and HEP modules may turn out to be very expensive and as difficult to maintain as an optimized fabric itself.

Unfortunately, no commercial system matches exactly the needs for an HEP data acquisition system. Traffic shaping components must be used in front of a Telecom ATM switch. In addition, an input multiplexing stage must be provided in front of any square switch. Front end data sources provide data in very raw formats, and intelligence is needed to build data structure to be handled by standard protocols. Software protocols must also be adapted and some of the additional features of a commercial telecommunication system such as high redundancy and bi-directional links have to be paid for, and probably will hardly be used. As mentioned above, some event assembling engine is necessary between the network and the on-line computer, especially when one expects to use very high speed links. This is true also for the case of any ad-hoc solution.

The addition of all these adaptation layers to a commercial fabric makes clear that the total cost of an in-house-built solution may be competitive. Commercial ATM systems are just being deployed in the field, and cost projections in the LHC era still contain a large uncertainty factor.

Finally, it is conceivable to assume that the lifetime of an LHC experiment will be long enough that at least one upgrade program will be undertaken after a few years of operation. As computers will become faster, the fabric itself may become the bottleneck of the system. This factor clearly favors a commercial solution. On the other hand, the front-end multiplexers and the adaptation layers will have to be upgraded too, and those will definitively not be purchased from industry.

#### ACKNOWLEDGMENTS

We gratefully acknowledge the contribution of the entire R131 collaboration, mainly M. Letharen and J. P. Dufey who have constructively criticized our paper.

#### REFERENCES

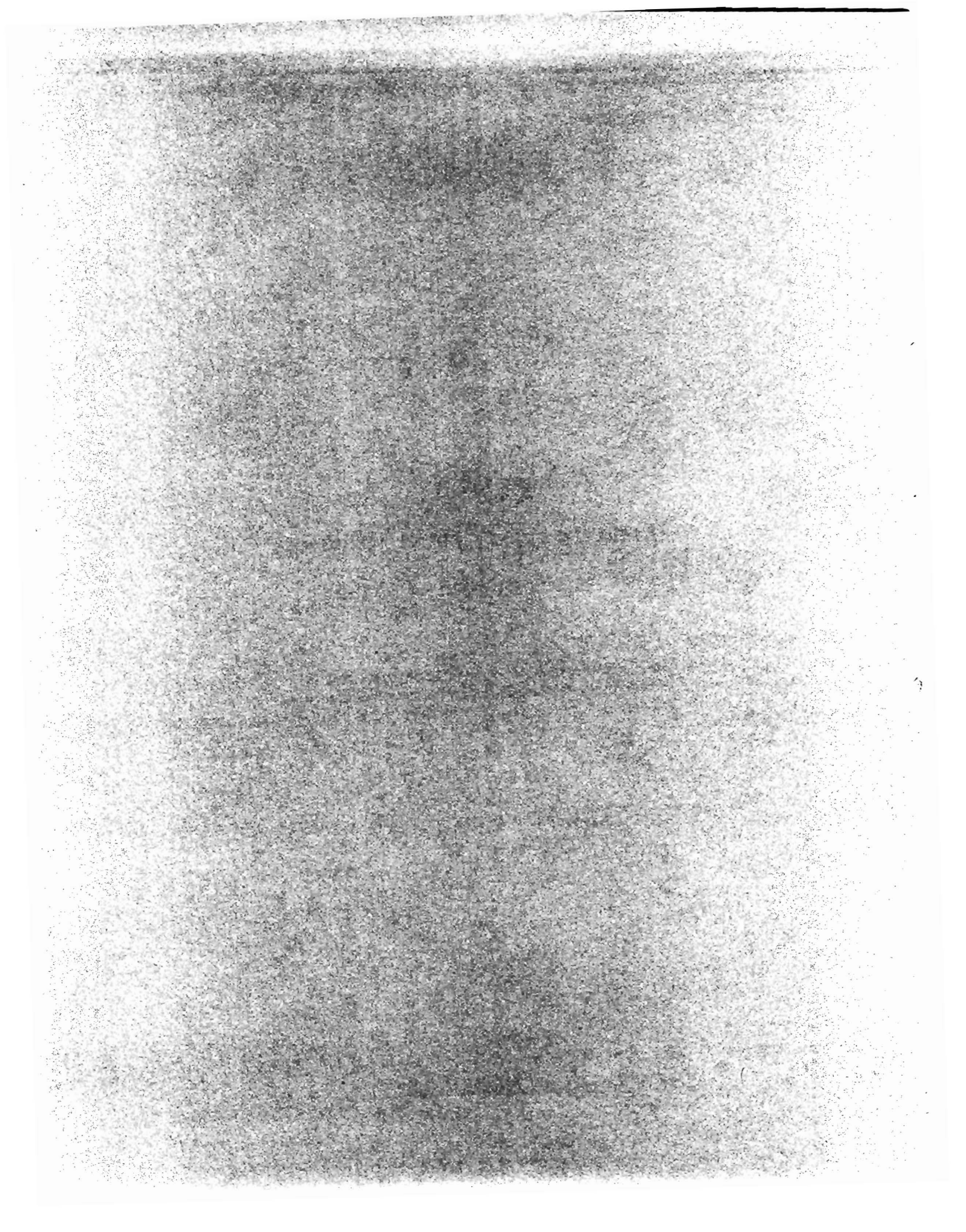
- [1] W. von Ruden et al. "The ALEPH Data Acquisition System", IEEE Trans. on Nucl. Sci., vol. 36, Oct. 1989, p. 1441.
- [2] J. Christiansen et al. "R131 Status Report", CERN/DRDC 83-55.

**S8-3**

**"Event Data Flow Control Techniques"**

**(Mark Bowden - Fermilab)**

An examination of various approaches to event data flow control in a large data acquisition system. Centralized vs. distributed control and the impact of virtual triggers on the processors, switching network and data flow controller. Interfaces between the data acquisition system and the trigger/front-ends.

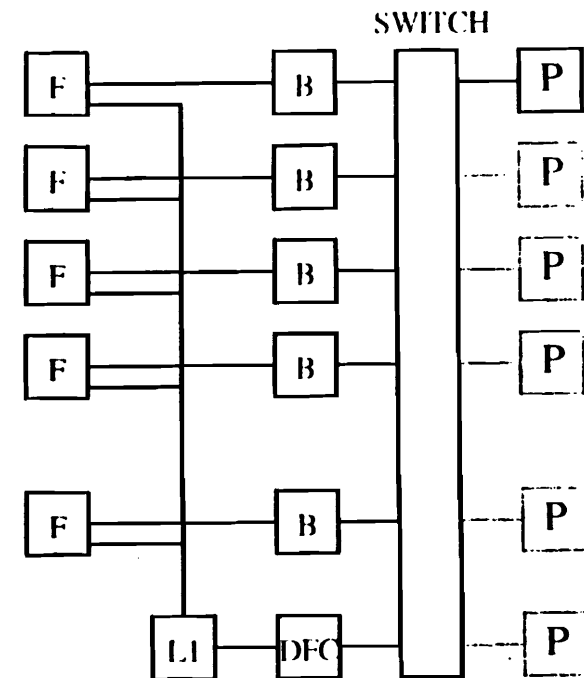


# Data Flow Control Techniques

October 28, 1994

M. Bowden

## Distributed Control (switch based architecture)





## Data Flow Control

- Independent processor driven readout
- Loosely coupled (msec message response times)

pipelined and buffered

context switching

data access times similar to disk access times

- Message based control

Event Request (Processor --> DFC)

Event Assign (DFC --> Processor)

Data Request(s) (Processor --> DPMs)\*

Data Return (DPMs --> Processor)

\* switch should support multicast

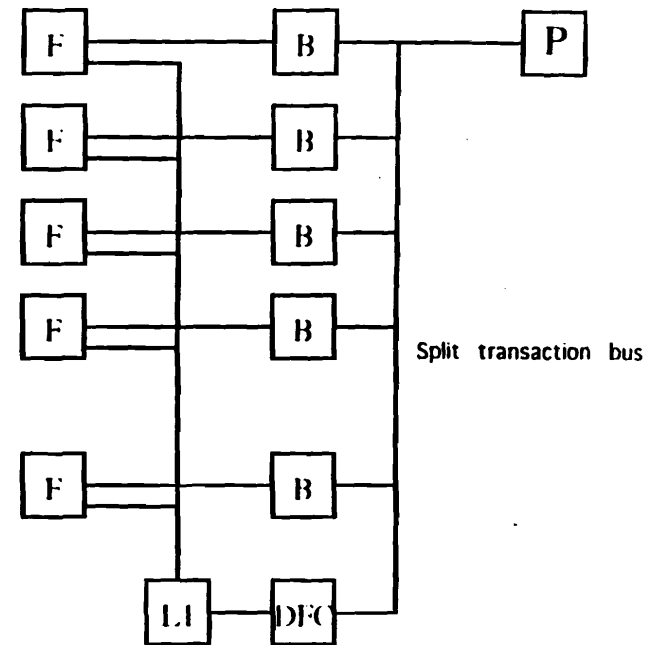
- DFC functions

Event assignment

Adaptive trigger rate control

Some load balancing

## Distributed Control (processor view of switch)

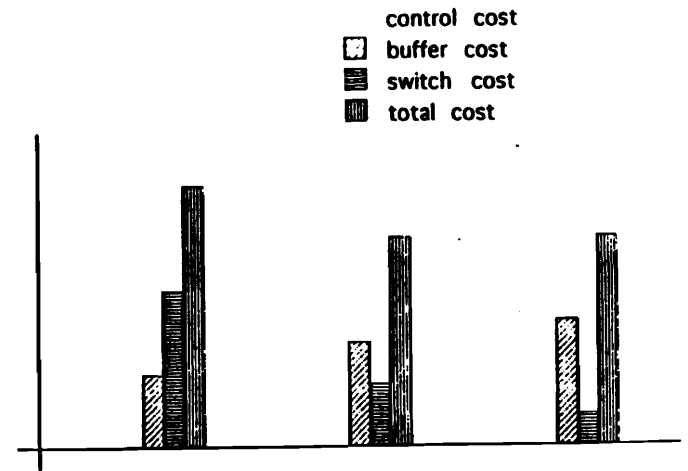


## Readout Modes

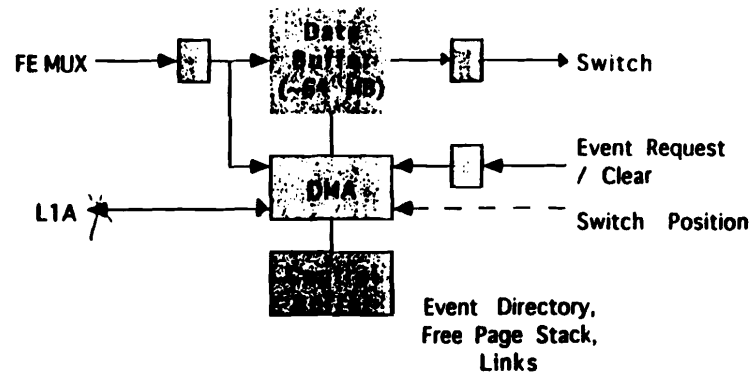
- ① Full readout mode.  
All data sent to processor following L1 accept.
- ② Partial readout by subsystem.  
All data involved in L2 trigger decision sent to processor following L1 accept. Remaining data sent following L2 accept.
- ③ Partial readout by "region of interest".  
L1 trigger summary sent to processor following L1 accept. Additional data sent per processor request. Remaining data sent following L2 accept.

## Readout Modes

	full readout	partial readout (by subsystem)	partial readout (by region of interest)
buffer size	10 MB	40 MB	80 MB
control bandwidth	5 MB/s	10 MB/s	40 MB/s
data bandwidth	50 GB/s	20 GB/s	10 GB/s



## Dual Port Memory (input buffer)



- **Page based data buffer ( to support all readout modes)**

Event fragments stored in data buffer

Fragment pointers stored in control buffer

Indexed by L1 accept number (virtual L2)

- **Data requests and clears by L1 number**

- **Additional DPM functions**

Traffic shaping

Asynchronous - randomization

Synchronous - switch position

## Switch Configuration Control

- **Very large event based applications**

100 KHz L1 accept rate

Virtual L2

1000 X 1000 switch

100 million event fragments with 2000000000 events per second in 100 buffers in 10000 fragments in subbuffer

Can't possibly be done in software based centralized switch controller.

- **Alternatives**

(1) Event concatenation (e.g., 1000 event fragments in each packet).

Requires large buffers.

Restricts partial readout mode.

(2) Self-routing switch (no central routing control).

(3) Predetermined configuration of switch (no central routing control).

## Characteristics of Event Builder Traffic

- Event building traffic is correlated (N to 1).
- High data latency is guaranteed.
- The average data rate from each source is known (or can be measured).
- The average data rate to each destination is known (or can be set).
  
- Switches are normally designed for random traffic.

Two choices.....

- ① Randomize the data
- ② Derandomize the switch

## Goals & Features

standard network applications

vs.

event builder applications

random, variable bandwidth traffic

correlated traffic

low latency (small buffers)

high latency (large buffers)

full interconnection, bidirectional ports

input to input and output to output connections not required, unidirectional, forward bandwidth is 1000 X reverse bandwidth,

## Switch Efficiency

- An event building switch must have a combination of input buffering or internal buffering of at least one event fragment for each virtual path in the system.

Example: 1000 channel switch, 1 MByte events

--> 1 Million virtual paths, ~ 1 KByte per event fragment

--> 1 GByte minimum total buffer size

- Large internal buffers are not available in most commercial switches. Buffers must be provided externally, at sources (input buffering).
- Various studies indicate that a switch with simple input buffering is < 40% efficient for random traffic due to HOL blocking.
- Without back pressure, data rate is set even lower to avoid cell loss.

## Switch Efficiency

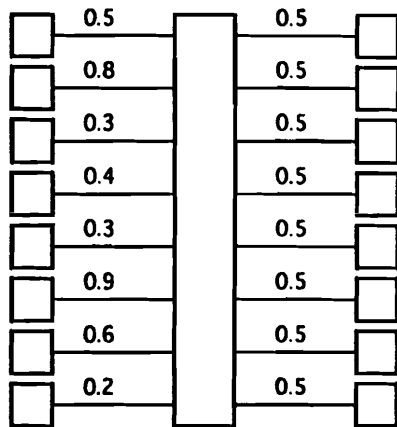
- IF
  - ① input buffering is sufficient to eliminate burst trafficand
  - ② the average data rate from each source is constant over timeand
  - ③ all outputs are equally loaded,

THEN

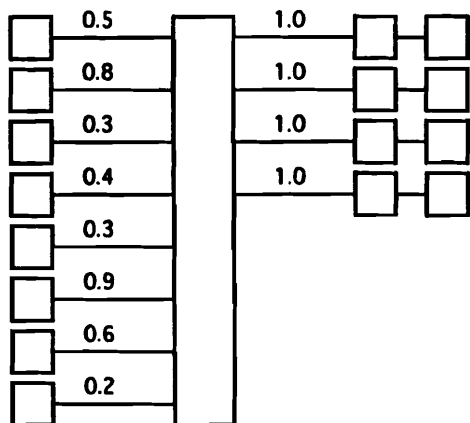
the traffic on all virtual connections is constant and the switch efficiency can be very high.

- Event building application meets these criteria

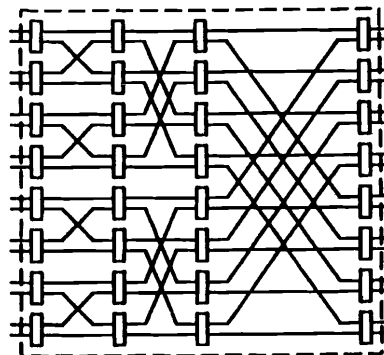
## Data Balancing



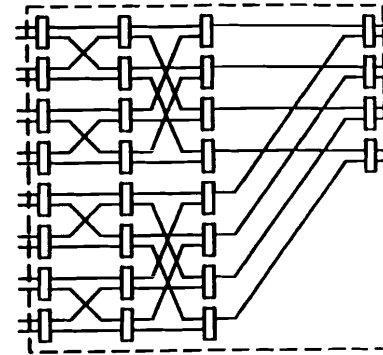
Outputs can be combined to increase output link utilization.



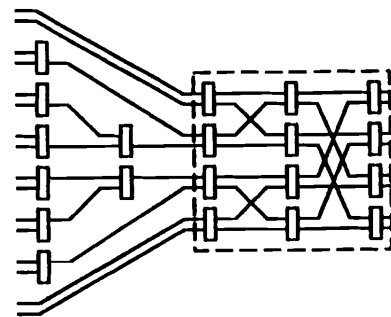
## Data Balancing



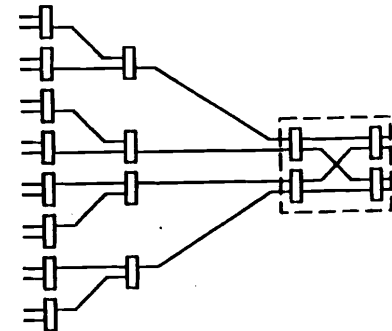
1) generic fully connected switch.



2) reducing number of outputs does not greatly reduce number of switch elements, even if output stage is depopulated.



3) multiplexing inputs to reduce switch size (as in partial readout by subsystem)



4) multiplexing inputs to reduce switch size (as in partial readout by region of interest)

### Typical Failure/Error Rates (from GEM)

component	failure rate	number	MTTF
front-end IC's	$10^{-11}$	200,000	20 days
rad hard LED	$1000^{-11}$	10,000	4 days (230 with redundancy)
high speed links	$1000^{-11}$	1,000	40 days
event builder			100 days
BER	$10^{-12}$	$10^{12}$	1 second
	$10^{-18}$	$10^{12}$	10 days
soft errors	$10^{-14}$	$3 * 10^{10}$	1 hour

### Conclusions (data flow control)

- Loosely coupled, message based
- Transmission latency not important  
(high latency is guaranteed by the application)
- Switch configuration latency is important
- Independent processor driven readout  
No central buffer manager or farm manager  
DFC assigns event numbers only

## **Conclusions**

(switch architecture)

- **Event building traffic is bursty but not random**

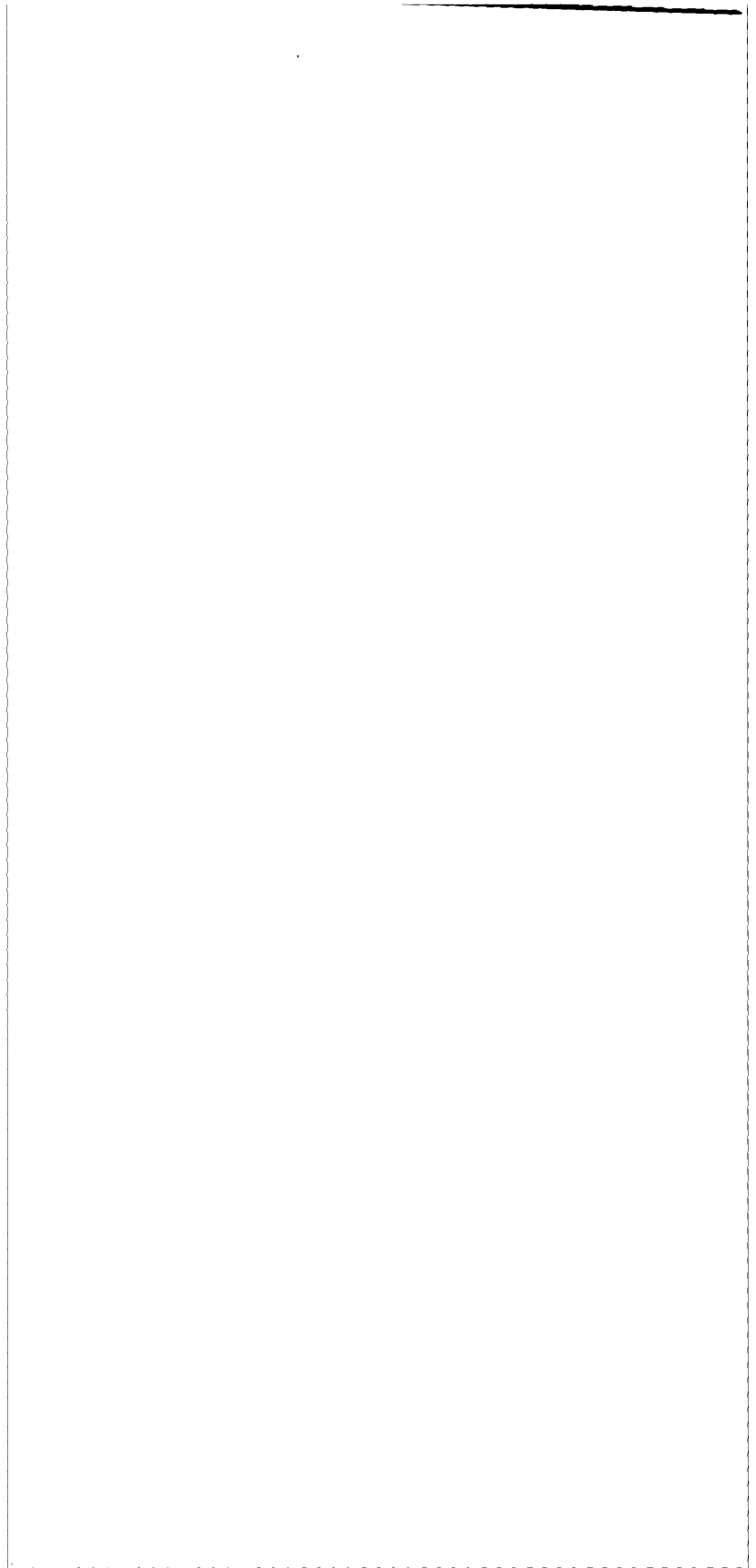
General-purpose switch architectures attempt to optimize for random traffic.

- **With sufficient buffering, traffic can be randomized.**
- **With sufficient buffering, traffic can also be derandomized.**
- **Using a commercial or non-commercial switch in a synchronous mode can result in higher efficiency, without blocking and without back-pressure.**
- **Either external traffic shaping or back-pressure is required in an event building application.**

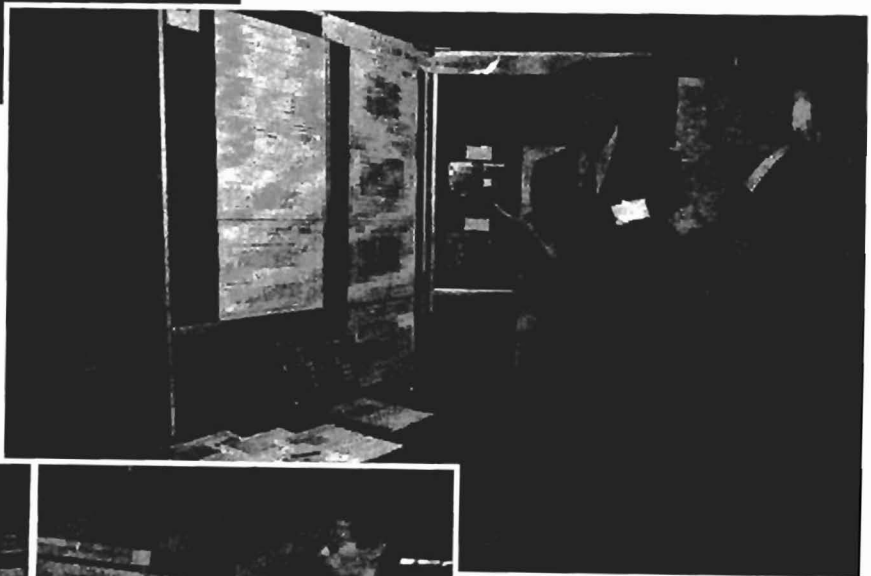
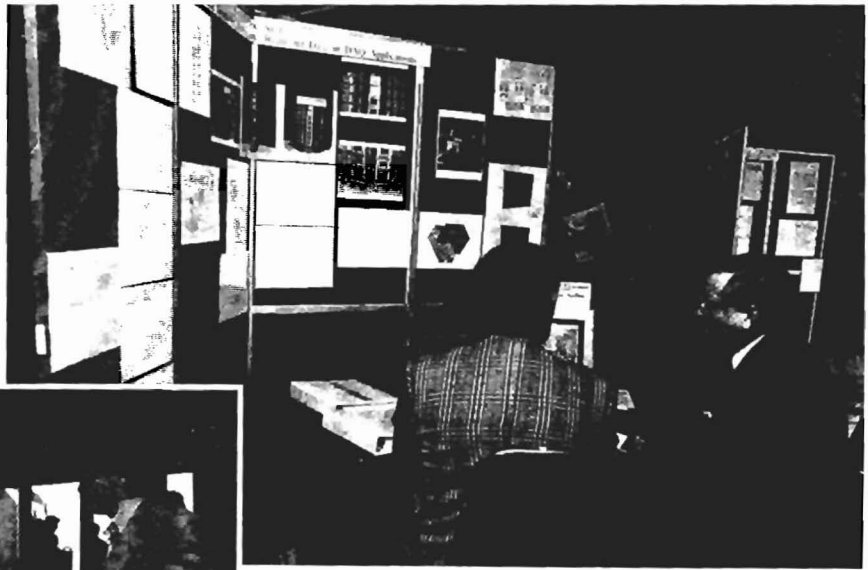
Not feasible to implement switch with sufficient internal buffering.

- **Cost difference may be 10X for general-purpose switch, compared to passive synchronous switch.**





Poster Sessions



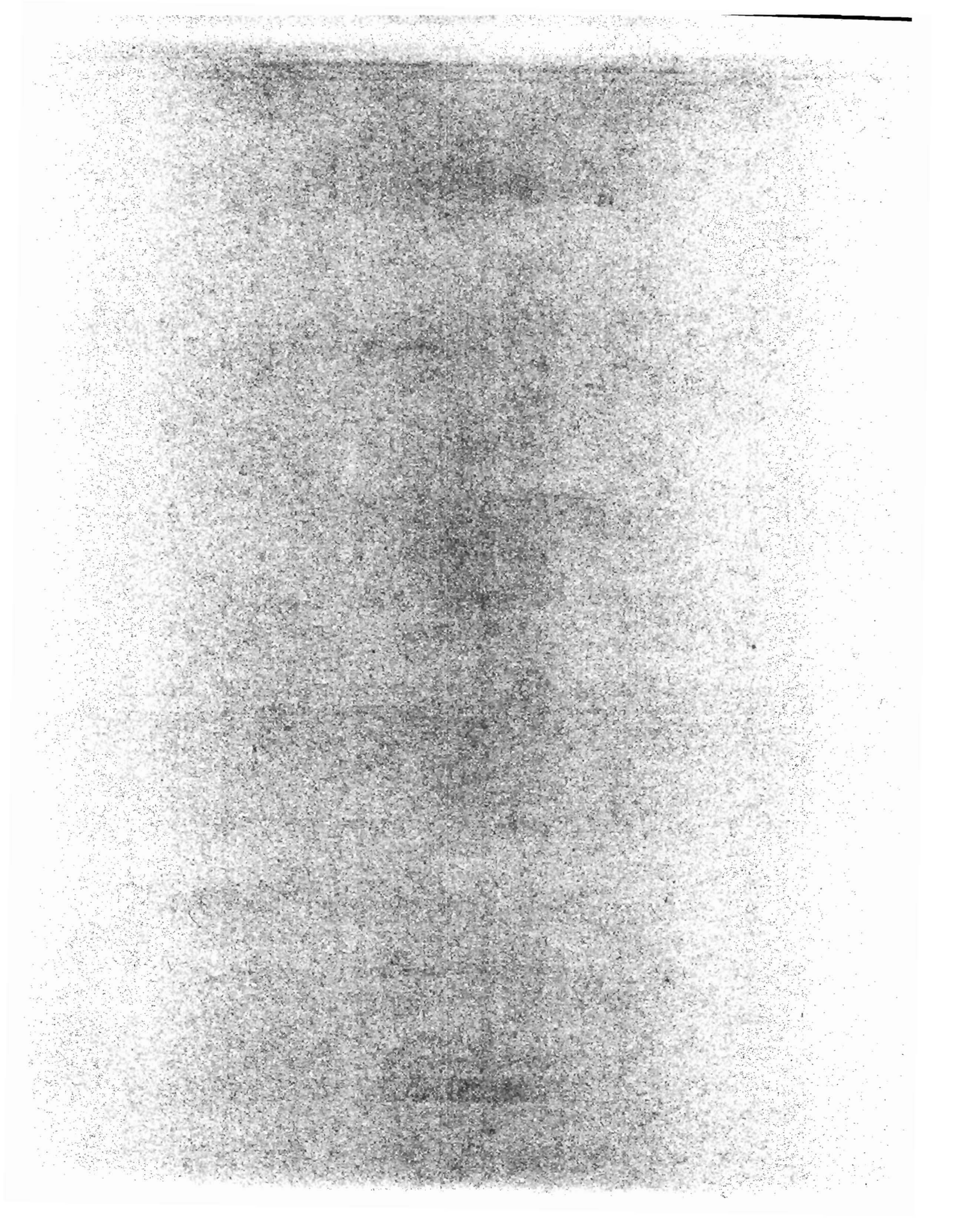


**DAQ Simulation Library**

**Ralf Spiwoke**

**Cern**

The poster shall show what the DSL is, and how it is used. It will explain the main idea behind the DSL, the graphical user interface as well as an example and its results.



# DAQ Simulation Library (DSL)

S. Buono, I. Gaponenko<sup>1</sup>, R. Jones, V. Kozlov<sup>1</sup>, L. Mapelli<sup>2</sup>,  
G. Mornacchi, D. Prigent, I. Soloviev<sup>3</sup>, R. Spiwox<sup>4</sup>  
*CERN, Geneva, Switzerland*

G. Ambrosini, G. Fumagalli, G. Polesello  
*Dipartimento di Fisica dell'Universita e Sezione INFN di Pavia, Italy*

P.Y. Duval, A. Le Van Suu  
*Centre de Physique des Particules de Marseille, IN2P3, France*

K. Djidi, M. Huet  
*Departement de Physique Nucleaire - STEN, C.E. Saclay, France*

## ABSTRACT

The RD13 project was approved in April 1991 for the development of a scalable data taking system suitable to host various LHC studies [1]. One of its goals is to use simulations as a tool for understanding, evaluating, and constructing different configurations of such data acquisition (DAQ) systems. The RD13 project has developed a modelling framework for this purpose. It is based on MODSIM II [2], an object-oriented discrete-event simulation language. A library of DAQ components allows to describe a variety of DAQ architectures and different hardware options in a modular and scalable way. A graphical user interface (GUI) is used to do easy configuration, initialization and on-line monitoring of the simulation program. A tracing facility is used to do flexible off-line analysis of a trace file written at run-time.

## I. Introduction

The DAQ systems for a detector at a future collider like LHC will have to cope with unprecedented high data rates (~10 GByte/s), parallelism (100 to 1000 processors) and new technologies (e.g. SCI, ATM) [3]. Simulation of different architectures, algorithms and hardware

- 
1. On leave from the Budker Institute of Nuclear Science, Novosibirsk, Russia.
  2. Spokesperson.
  3. On leave from the Petersburg Nuclear Physics Institute, St. Petersburg, Russia.
  4. Also at University of Dortmund, Dortmund, Germany.

components can be used to predict data throughput, the memory space and cpu power required and to find bottlenecks before such a system will be actually constructed. Therefore one needs a modelling framework with a high level of description and a clear mapping between the system to be built and the system to be modelled. The framework has to be modular and scalable to allow simulations of the different configurations from simple systems up to full DAQ systems for big detectors.

The modelling framework presented in this paper is written in MODSIM II [2] which is an object-oriented language for discrete event simulation and has its own graphics library.

The modelling framework itself consists of a library of generic objects for the DAQ simulation (DSL, DAQ Simulation Library), a graphical user interface (GUI) and a tracing facility for off-line analysis of the simulation results. The code is managed by CVS [4] and a makefile is used to build the binaries.

The package has been developed in the RD13 project and a working version is available [5]. It has been used for small applications and is used for event building studies and is being considered for DAQ simulations by the ATLAS collaboration [6].

## II. The DAQ Simulation Library

The DAQ Simulation Library consists of generic objects to describe any kind of DAQ system. The basic elements are:

- **Items** are information carrying data accumulations that are passed in a DAQ system, e.g. event data, trigger signals.
- **Processes** are the active objects in a DAQ system passing items and acting on them, e.g. read-out or recording process.
- **Resources** are the limiting factors the processes have to compete for in order to fulfill their task, e.g. cpu, buffer, transfer media.
- **Control** elements are abstract objects controlling the processes and carrying information on the data flow, e.g. timers, allocation algorithms.

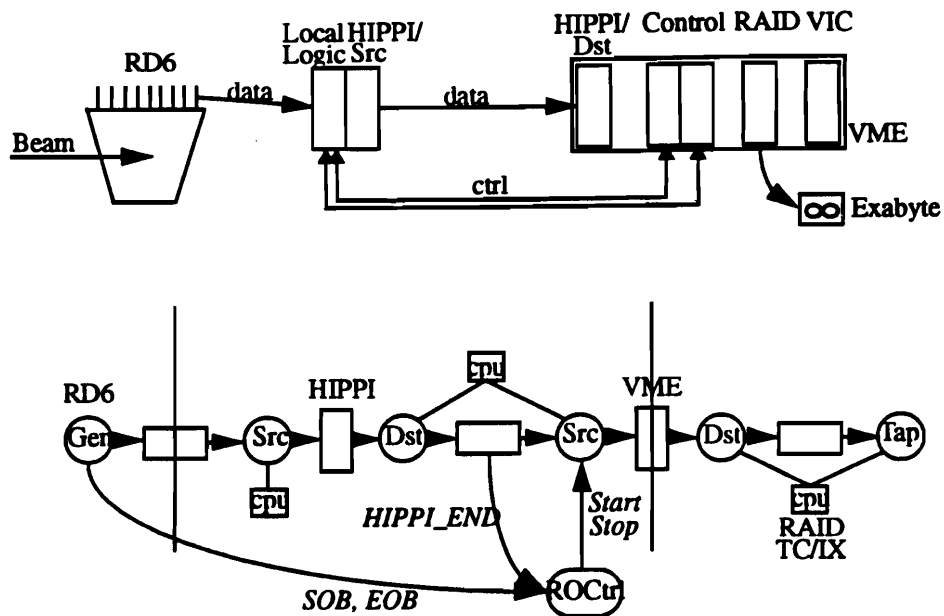
The main idea of the DSL is to use the smallest indivisible (“atomic”) processes that can then be used to build up any DAQ system. A dozen “atomic” processes have been defined and make the core of the DSL.

The DSL has a generic level consisting of objects for a generic description of DAQ systems, and a user level where inheritance is used to combine the generic objects with user dependent features. Thus the DSL contains the possibility to refine the objects and to include hardware dependent features.

As an example of an application of the DSL the readout of the combined RD6/RD13 testbeam in November 1993 has been simulated [7]. This setup consisted of a single chain of data flow using a HIPPI link and had a total data rate of 1.5 MByte/s (FIGURE 1.). This example was

used as a proof of principle: it showed the easy mapping between reality and simulation and the consistency between the values measured and the values simulated. The simulation could then be used for changes of parameters and extensions of the setup.

**FIGURE 1. The RD6/RD13 Testbeam Setup (Hardware & Model)**



### III. The Graphical User Interface

A graphical user interface [8] based on the graphical objects in MODSIM II is used to easily configure the simulation model, to initialize each object and to monitor parameters on-line. The GUI has three windows:

- the **library window** displays the objects of the DSL.
- the **configuration window** is a canvas on which the configuration to be simulated is built.
- the **display window** monitors parameters while running the program.

Additional features are available

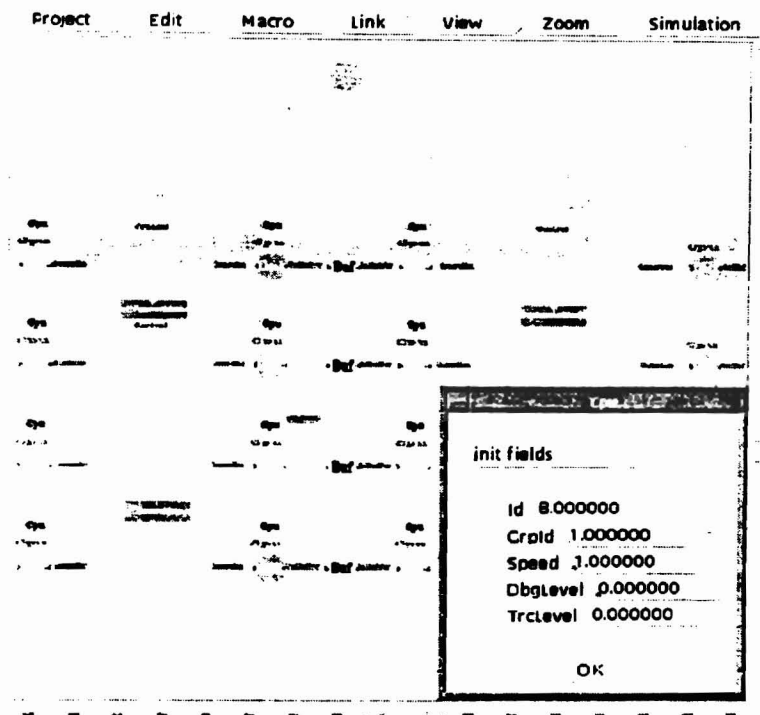
- for saving and reloading whole configurations and their initialization values.
- for grouping of objects (very useful for copying parts of the configuration).
- for organizing views in a hierarchial way (very useful for complex configurations).

While the GUI can be used to build a configuration and to debug it, there is also a fast version available which can be used to run the program without graphics, thus increasing the performance for time consuming simulations.



An example for the configuration Window is shown in FIGURE 2. This shows part of a configuration of a DAQ system and shows also the input window for the parameters of one of the DAQ objects.

FIGURE 2. The Configuration Window with an example



## IV. The Tracing Facility

The tracing facility is a tool that allows each single "atomic" process to report on its activity by writing a trace record in a file. This facility can be switched on and off for each individual process. The format of the trace record can be extended by the user.

The trace file can have binary or ascii format and can be processed off-line (i.e. after running the simulation) by a tool which is implemented as a C program. This tool can:

- reproduce each individual trace record.
- produce general statistics, e.g. number of events generated, size of the events, etc.
- produce statistics on each type of trace record, e.g. event generation frequency, buffer usage over time, etc.
- can order the records on an event-by-event basis, e.g. latencies, lifetime of event, etc.

The results of the various analysis are written in ntuple format and can be visualized with the help of PAW [9].

## V. Conclusions

The DSL (together with the GUI and the tracing facility) is a high-level description language for simulations of DAQ systems. It can be used for simulation of any kind of DAQ system and has the possibility to include lower level hardware simulations. The GUI allows an easy configuration, initialization and on-line monitoring of a simulation program. The tracing facility allows a highly flexible analysis of the output.

The part interfacing from the detector simulations to the DAQ simulations delivering the information on size and distribution of the data in the front-end buffers (the physics interface) is already foreseen, but not yet implemented.

The DSL has been successfully used for simple examples. In the RD13 project it is used for studies of the event building, the event distribution and the interfaces to the upstream and downstream parts of the DAQ system. It is being discussed for use in simulations of functional models of the whole ATLAS DAQ including read-out, Level-2 triggering, event building and Level-3 triggering.

A version of the software is publicly available [5] and documentation can be found on WWW [10].

## VI. References

- [1] L. Mapelli et al., A Scalable Data Taking System at a Testbeam for LHC, CERN/DRDC 90-64, CERN-DRDC 94-24.
- [2] CACI Products Company, MODSIM II The Language for Object-Oriented Programming, La Jolla, California, 1991.
- [3] L. Mapelli, The challenge of Triggering and Data Acquisition at Supercollider Experiments, NIM A315 (1992) 460.
- [4] P. Cederqvist, Version Management with CVS (GNU), 1993.
- [5] anonymous ftp from sunsci.cern.ch, directory simulation/dsl.
- [6] The ATLAS Collaboration, Letter of Intent for a General Purpose pp Experiment at the LHC, CERN/LHCC 92-4.
- [7] R. Spiwoks, RD13 Technical Note 97, Modelling of the RD6/RD13 Testbeam Setup, February 1994.
- [8] K. Djidi, RD13 Technical Note 107, A General Graphical User Interface with MODSIM II, February 1994.
- [9] R. Brun et al., PAW - Physics Analysis Workstation, CERN Program Library Q121, Geneva, 1991.
- [10] URL: <http://rd13doc/welcome.html>.

---

---

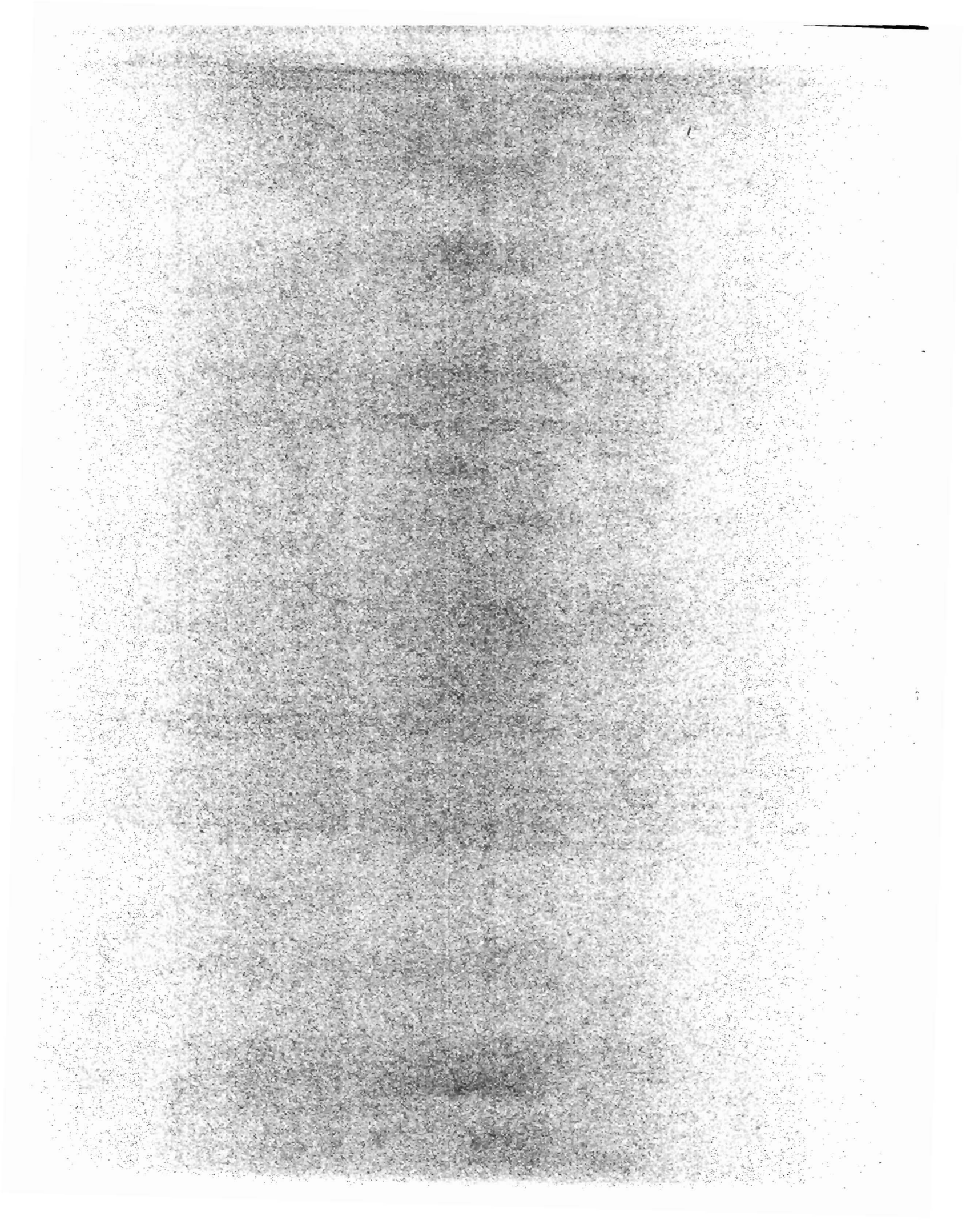
**A 155 Mbit/s VME to ATM interface with special features for event building applications based on ATM switching fabrics.**

**Leif Gustafsson**

**Uppsala University**

In the framework of the CERN RD31 project, a 155 Mbit/s SONET VME-ATM interface is being developed as part of a program to evaluate ATM-based event builders for future high-rate HEP experiments. The design uses commercial chipsets for high-performance implementation of the ATM layer and the ATM adaption layer (AAL5) of the standard B-ISDN protocol. The interface is built on a commercial VME board containing a 20 MHz RISC processor that runs firmware to support the protocol chipsets and to implement the higher layers of the data acquisition protocol. The physical layer is based on a commercial chip supporting the Synchronous Optical Network (SONET) protocol at 155 Mbit/s over multimode fibre.

Normally congestion will occur in an ATM switching fabric subjected to the "many-to-one" traffic patterns that characterize event building, and this can result in cells being discarded. The design therefore includes a hardware sub-system to support a specific "traffic shaping" scheme that has been proposed as a means of averting this problem. It acts by randomizing the time at which cells are injected into the switching fabric.



# A 155 Mbit/s VME to ATM interface with special features for event building applications based on ATM switching fabrics

*The RD31 collaboration*

*L. Gustafsson<sup>1</sup>, M. Costa<sup>2</sup>, J.-P. Dufey<sup>2</sup>, T. Lazraq<sup>3</sup>, M. Letheren<sup>2</sup>,  
A. Manabe<sup>4</sup>, I. Mandjavidze<sup>5</sup>, C. Paillard<sup>2</sup>*

*Presented at the International Data Acquisition Conference,  
Fermilab, Batavia, 26-28 October 1994*

- 1 Institute of Radiation Sciences, Uppsala university, Uppsala, Sweden
- 2 CERN, Geneva, Switzerland
- 3 Royal Institute of Technology, Stockholm, Sweden
- 4 visitor at CERN on leave of absence from KEK, Tsukuba, Japan
- 5 visitor at CERN on leave of absence from the Academy of Science, Tbilisi, Georgia

## Abstract

In the framework of the CERN RD31 project, a 155 Mbit/s SONET VME-ATM interface is being developed as part of a programme to evaluate ATM-based event builders for future high-rate HEP experiments. The design uses commercial chipsets for high-performance implementation of the ATM layer and the ATM adaptation layer (AAL5) of the standard B-ISDN protocol. The interface is built on a commercial VME board containing a 25 MHz RISC processor that runs software to support the protocol chipsets and to implement the higher layers of the event building protocol. The physical layer is based on a commercial chip supporting the Synchronous Optical Network (SONET) protocol at 155 Mbit/s. The physical medium interface uses relatively cheap LED-based optoelectronic transceivers and multimode fibre.

Normally congestion will occur in an ATM switching fabric subjected to the "many-to-one" traffic patterns that characterize event building, and this can result in cells being discarded. The design therefore includes a hardware sub-system to support a specific "traffic shaping" scheme that has been proposed as a means of averting this problem. It acts by randomizing the time at which cells are injected into the switching fabric

## 1. Introduction

We are developing, in the framework of the RD31 project [1], a VME-ATM interface as part of the implementation of an event builder demonstrator. The reasons for custom development are the following:

- in order to gain experience with ATM technology and to check if and how the functionality needed for event building can be implemented using

commercially available chipsets designed for building ATM host interfaces.

- to check if and how sustained high data transfer rates, as close as possible to the maximum available with the 155 Mbit/s bit-rate, can be reached.
- to include the hardware necessary to implement the Randomizer traffic shaping scheme [1], which is specific to the event building problem.

## 2. Demonstrator system

Figure 1 shows the demonstrator system currently being assembled. The switching fabric is a prototype 8 x 8 multi-path self-routing architecture [2,3] provided by Alcatel Bell Telephone.

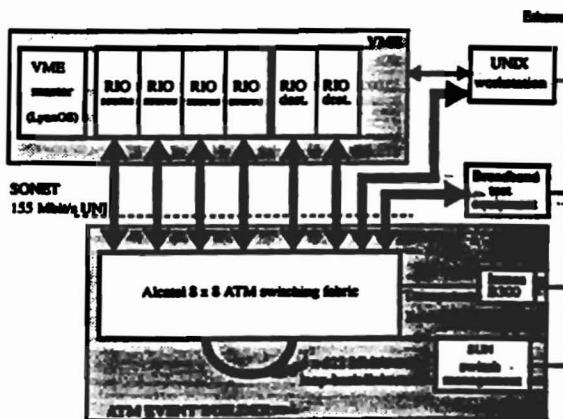


Fig. 1 The layout of the event builder demonstrator system.

The switch has been delivered together with embedded operations and management software (transputer based), and an operator interface which runs on a SUN workstation and communicates with the embedded software via an ethernet to transputer-link bridge. A Hewlett Packard broadband test system [4] is used for ATM protocol validation at the physical and ATM layers and also allows performance measurements and comprehensive error stressing. The functioning of the 8 x 8 Alcatel switch has been successfully validated using the HP test system.

The Alcatel switch supports the 155 Mbit/s SONET [5] User-Network-Interface (UNI) standard, and it will be used, together with the VME-ATM source and destination modules described below, to test data acquisition protocols and traffic shaping techniques. In the future, commercial workstations supporting the SONET standard can be incorporated into the demonstrator event builder, and alternative SONET-compliant switching architectures can be evaluated.

### 3. Event builder protocol stack

Figure 2 shows our proposal for a dataflow protocol stack to be implemented in the source and destination modules of a parallel event builder based on a switching network. In the case of the VME-ATM interface the upper layers are implemented in software, while the lower layers correspond to layers of the B-ISDN standard [6] and they are implemented in commercial chipsets.

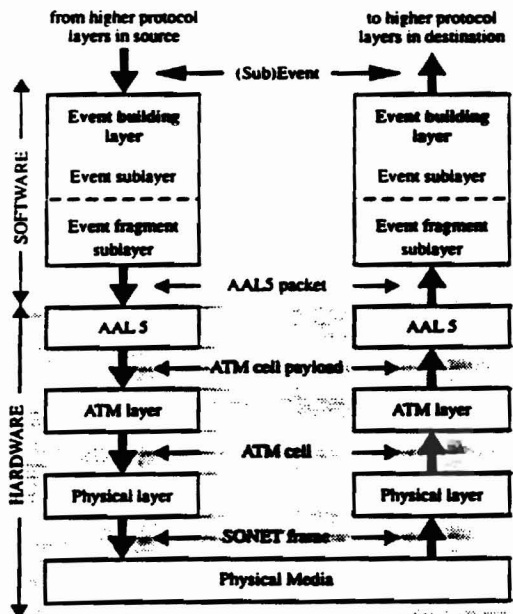


Fig. 2 The data acquisition dataflow protocol stack.

In the data-driven event builder, the sources have the task of sending event fragments to the destination. Each source must collect the data (or poll for their arrival) in a memory and identify the VC over which the data will be sent to the destination. In the destination, event fragments from different sources have to be linked together to form a built event. Some method must be applied to recognize when all fragments of an event have been received, and missing fragments must be flagged. This global scheme of assembling the fragments constitutes the *event building layer* of the dataflow protocol stack.

The underlying layers of the protocol stack are dependant on the switching fabric architecture and technology. We consider here only the case of an ATM self-routing packet switching architecture. We have selected the standard *ATM adaptation layer (AAL)* protocol called AAL5 (one of several standardized AAL protocols [6]) to implement the method by which variable length data packets, with a maximum length of 64 kByte, are segmented into (and reassembled from) sequences of fixed-length ATM cells. The next lower level of the protocol stack, namely the *ATM layer*, handles the functions associated with the routing of cells through the switching fabric. The *physical layer* specifies how the cells are to be framed and transported over some physical medium (in our case fibre optic links).

The event building layer is split into two sublayers; the *event sublayer* implements those functions that are independent of the underlying hardware, while the *event fragment sublayer* is necessary to adapt the requirements of the event sublayer to the services provided by any AAL hardware that may be selected. For example, if the event fragments can be larger than 64 kByte (expected for the ALICE experiment [7]), one of the tasks of the event fragment sublayer would be to segment the event fragments into several AAL5 packets and to recombine them in the destination.

### 4. VME-ATM interface hardware

We are developing a VME-ATM interface module to act as source and destination modules in the event builder demonstrator system described in section 2. This interface is implemented on a commercial VME RISC IO (RIO) module [8]. This module includes a 25 MHz RISC processor which will run the software implementing the higher layers of the protocol stack. The lower layers will be implemented in hardware in the form of a daughter board that plugs into the RIO mother-board and will communicate with the processor via the system bus. The architecture of the ATM adapter

hardware is shown in figure 3. During the prototyping phase we actually have three separate hardware plug-in modules. One implements the B-ISDN AAL5 and ATM protocol layers, one implements the SONET physical layer, and the third is an optional randomizer module that includes special hardware [9] to perform the traffic shaping required for event building over telecommunications switches.

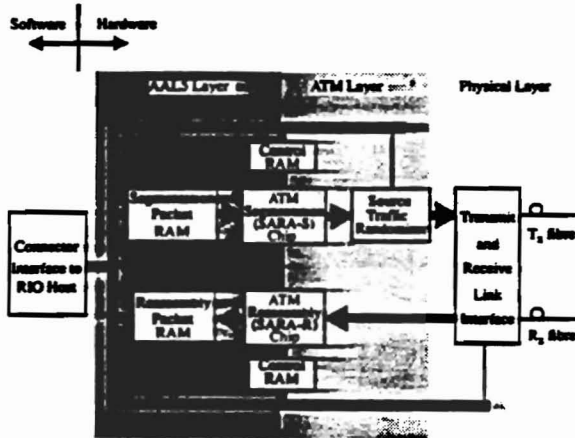


Fig. 3 Block diagram of the interface hardware supporting the AAL, ATM and Physical layers of the B-ISDN protocol.

#### 4.1 AAL and ATM layer module

A commercial chip set [10] performs in hardware the segmentation and reassembly of data packets, in the AAL5 format (up to 64 kByte long), into/from ATM cells. These segmentation and reassembly (SARA) chips require two dual-ported memories each. The first one, the packet memory, stores the actual data packet to be transmitted (or that has been received and reassembled). In order to sustain the full 155 Mbit/s rate, this memory is accessed by the SARA via a 32-bit port, and 12 memory accesses are required per ATM cell. The second port is also 32-bit wide and connects to the host's system bus. The port arbitration logic assigns equal priority to both ports. Currently we transfer data between VME bus and the packet memory using programmed I/O. Some improvements to the current design are required in order to be able to support block transfer mode between VME address space and the packet memory.

The second type of memory contains packet descriptors that point to the location of AAL5 packets in the packet memory and specify their length, the virtual connection index (VCI) and its associated traffic metering parameters. These control memories are

accessed from the host port and the SARA port via 16-bit datapaths. The segmentation chip implements sophisticated procedures to segment the packet when multiple VCIs are concurrently active, therefore the descriptors are arranged in linked lists and require a complex management function, which is performed by the segmentation chip itself. We measured that for every ATM cell generated and passed to the physical layer not less than 23 control memory accesses are required for this management.

Each SARA chip can support up to 64k different VCIs, and can simultaneously segment/reassemble 8k packets, which is sufficient to construct very large event builders. The current design uses 512 kByte packet memories and 256 kByte control memories.

#### 4.2 Physical layer module

For compatibility with the Alcatel switching fabric we chose to use the 155 Mbit/s physical layer interface defined in the SONET standard. The physical interface, on the sender side, has to add the ATM cells into a SONET bit-frame; conversely the receiver has to retrieve ATM cells from SONET frames. On the receiver side the clock and data have to be recovered from the NRZ encoded serial input stream.

The SONET framing and data-link error detection functions are implemented with a commercial SONET User Network Interface (SUNI) chip [11]. Clock and data recovery are performed by a commercial clock recovery chip [12]. The full-duplex, short-haul link consists of a pair of multimode fibres driven by a relatively cheap, LED-based optoelectronic transceiver [13].

A 16-bit wide datapath between the ATM layer module and the physical layer module is sufficient to sustain traffic at 155 Mbit/s.

#### 4.3 Traffic shaping hardware

Source traffic shaping can be used to control congestion within the switching fabric by regulating the bandwidth assignment to virtual connections, and by modulating the time at which cells are injected into the switch. Figure 4 shows the principle of the randomizer traffic shaping hardware developed for event building applications. Modeling studies have shown that this technique results in favourable scaling characteristics [1] (e.g. linear growth of event building latency is observed as a function of network size at constant load factor, etc.).

Each source module must maintain one logical FIFO queue per destination. The SARA segmentation chip services the packet queues in round robin, picking one



cell from the head of each packet queue in each round robin cycle. Rate metering is effectively imposed by SARA applying a programmable delay between each service cycle.

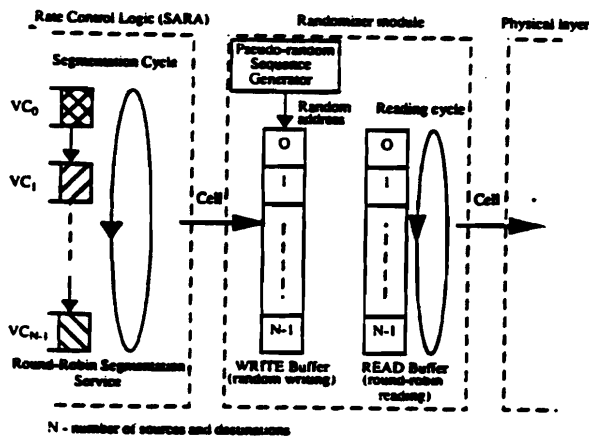


Fig. 4 The principle of operation of the randomizer traffic shaping hardware.

The randomization of a cell injection time, which breaks the correlation between traffic from different sources and therefore minimizes congestion inside the fabric, is performed by the randomizer module [9]. The randomizer contains two cell buffer memories (a "write" buffer and a "read" buffer). It operates by writing the ATM cells sent out by SARA during a segmentation cycle into pseudo-random locations in the write buffer. During the next segmentation cycle the write and read buffers are switched. The cells from the read buffer are always readout by scanning the memory sequentially, thus effectively adding a random delay to the injection time of cells on a given VC. The algorithm guarantees that cell sequencing within each VC is preserved.

## 5. VME-ATM interface software

Figure 5 shows the structure of the interface software. It is divided into 3 layers, each of which has a *dataflow protocol* plane and a *control and management* plane. The *event building* layer implements the dataflow and control functions associated with event building that are independent of the communication protocol used. The *network interface* layer implements dataflow and control functions specific to the communication technology. The *I/O specific* layer provides a library of functions to access the hardware. More details on the software are given in reference [14].

For example, the software in the event building

layer's dataflow plane will be used to evaluate various event building schemes, such as event building by "time-out", by "notification of zero-data" [1], etc. The time-out method assumes that all source data has been received after a predefined delay; this method will always be needed in order to recover from failed hardware (e.g. source modules). The notification method requires all sources to send a data message, even if the source has no data; event building is completed when a message has been received from all sources.

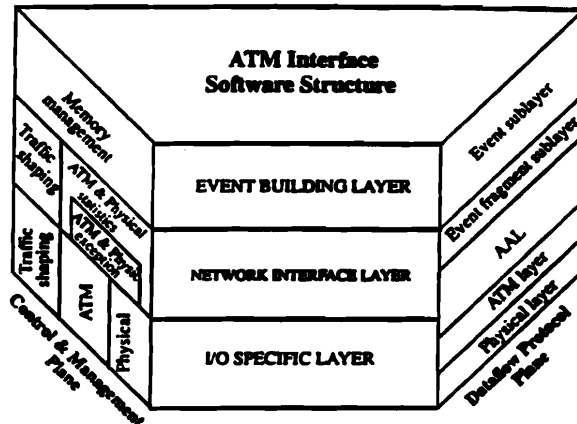


Fig. 5 Structure of the VME-ATM interface software.

Examples of the functionality to be implemented in the control and management plane are, for the case of the network interface layer, interface hardware initialization, hardware exception handling, performance monitoring and compilation of statistics, etc.

A software traffic shaping method [15] will also be evaluated, and this will involve software in the control and management plane of the network interface and I/O specific layers.

## 6. Status and performance

The interface has been tested successfully in full loop-back mode, including SONET framing and optical fibre. We have also tested with success the interoperability, at the ATM and physical layers, with the HP broadband test system. The randomizer printed circuit board module has been constructed and its control logic is implemented in a XILINX field-programmable gate array [16]. Stand-alone testing of the randomizer module is underway.

Currently we achieve 50 Mbit/s transfer rate between VME bus and the packet memories using programmed

I/O. In loop-back mode, when packet data are transferred between packet memories (but not to the VME bus), we achieve a sustained data transfer rate of 95 Mbit/s. Further optimization of the design is required in order to sustain the full bandwidth offered by the 155 Mbit/s bit-rate of the fibre optic transmission standard.

This development is proving to be very useful to familiarize us with the implementation of ATM technology and has shown which are the critical points requiring improvement in order to be able to sustain traffic at the full bandwidth offered by the 155 Mbit/s bit-rate. The next step will be to test interoperability with the commercial switching fabric in the demonstrator system, followed by integration of the randomizer module and the implementation and evaluation of the higher-level (software) layers of the event builder protocol.

## References

- [1] J. Christiansen et al., NEBULAS - A high performance data-driven event building architecture based on an asynchronous self-routing packet-switching network. CERN / DRDC 92-14, CERN / DRDC 92-47 and CERN / DRDC 93-55.
- [2] M. Henrion and D. Boettle, Alcatel ATM Switch Fabric and its Properties, Electrical Communications, Vol. 64, No. 2/3, (1990), pp.156-165, available from the Alcatel company.
- [3] M. Henrion et al., Technology, Distributed Control and Performance of a Multipath Self-Routing Switch, Proc. XIV Intl. Switching Symposium, Yokohama, Japan, Oct. 1992, Vol 2, pp. 2-6.
- [4] Hewlett Packard, Broadband Series Test System, 1994.
- [5] ANSI T1.105-1991, Digital hierarchy - Optical interface rates and formats specifications (SONET); ANSI T1E1.2/93-020R3, B-ISDN customer installation interfaces: physical layer specification.
- [6] International Telegraph and Telephone Consultative Committee, ITU, Geneva; recommendations I.150, I.211, I.311, I.321, I.327, I.361, I.362, I.363, I.413, I.432, I.610.
- [7] N. Antoniu et al., A large Ion Collider Experiment at the CERN Large Hadron Collider, CERN / LHCC / 93-16, March 1993.
- [8] Creative Electronic Systems SA, Geneva, RIO 8260 and MIO 8261 RISC I/O processors - user's manual, version 1.1 (March 1993).
- [9] T. Lazraq et al., ATM traffic shaping in event building applications, RD-31 note 94-09.
- [10] Transwitch Corp., Shelton, Connecticut, USA, SARA chipset, Technical Manual, version 2.0, Oct. 1992.
- [11] PMC-Sierra Inc., the PMCS345 Saturn user network interface manual (May 1993).
- [12] Analog Devices Inc., the AD802 clock recovery and data retiming phase-locked loop.
- [13] Hewlett Packard, HFBR-5205 ATM/SONET OC-3 transceivers.
- [14] M. Costa, ATM-VME interface software description, RD31 note 94-08.
- [15] I. Mandjavidze, A new traffic shaping scheme: the true barrel shifter, RD-31 note 94-03.
- [16] The Programmable Logic Data Book, Xilinx Inc., San Jose, California, 1994.

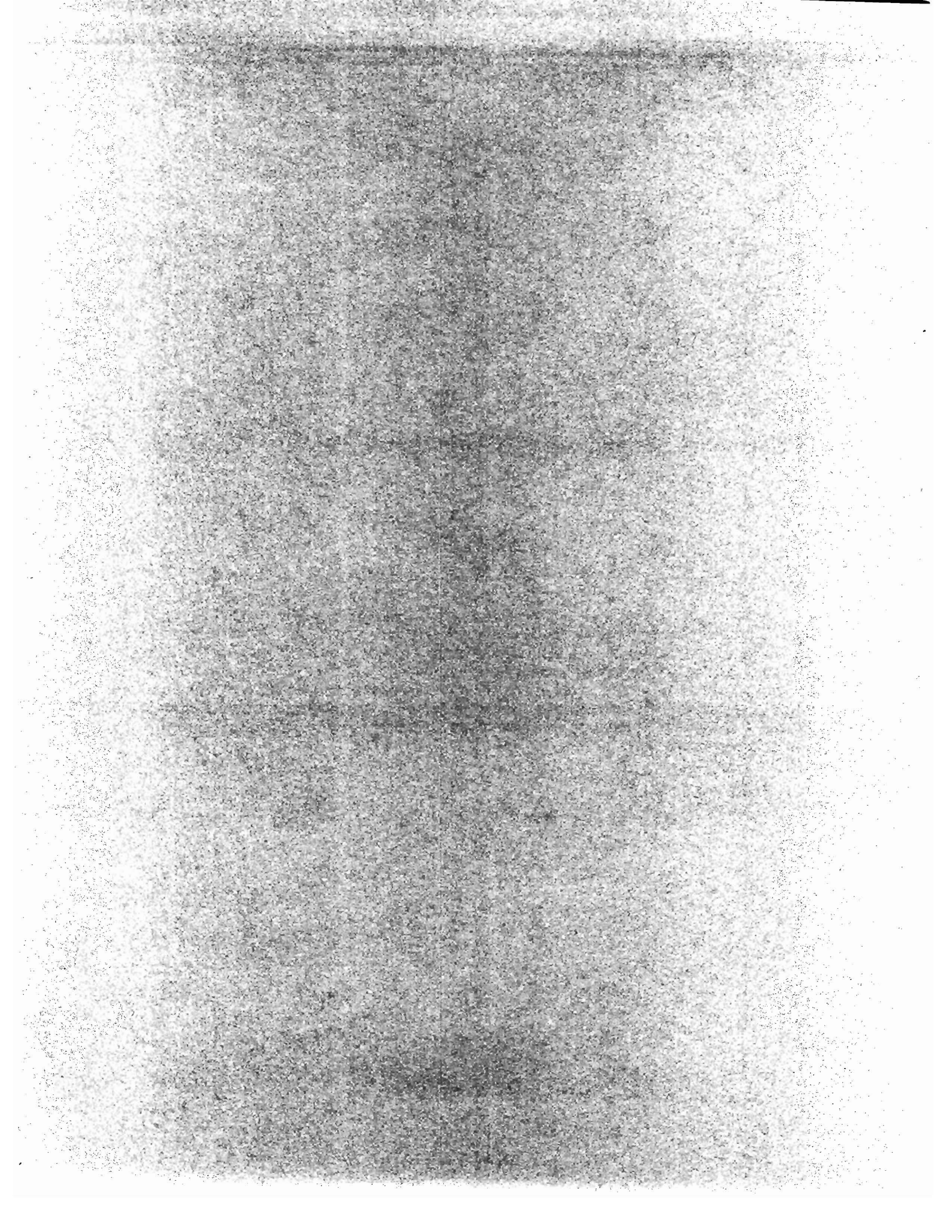


## **Performance Simulations of Networks with Point-to-Point Links**

**Geir Horn**

**SINTEF**

As you probably know, we already have an SCI simulator written in C++. Currently we are writing a new simulator for simulation of system behavior in DAQ and parallel computer systems. The poster will show some of our experiences with the old simulator and why we decided to write a new from scratch, based on what we learned from the old simulator. We will also present some overall design issues relating to the modular modeling in our new simulator. Last we will focus on which statistics that can be gathered from the simulation and how it may be used.



# Performance Simulations of Networks with Point-to-Point Links \*

Geir Horn, Svein Linge, Ernst H. Kristiansen

SINTEF Instrumentation, Forskningsveien 1, P.O. Box 124 Blindern, 0314 Oslo, Norway.

Øystein Gran Larsen

University of Oslo, Department of Informatics, P.O. Box 1080 Blindern 0316 Oslo, Norway.

## Abstract

Point-to-point link technology such as SCI<sup>1</sup> will become an alternative to backplane busses in DAQ systems in the near future. We show by simulation how the serial HIC<sup>2</sup> technology may be used to route SCI packets from ringlet to ringlet in a small topology. Design aspects of a HIC network simulator under development is also presented.

## I. INTRODUCTION

The large DAQ systems in high energy physics must provide high speed interconnections between a large number of front end data acquisition units and a processor farm. This requires an intermediate crossover network to build many event fragments from the DAQ front end units into single events for a CPU of the processor farm. The system is illustrated in figure 1.1.

The demanding communication needs of a DAQ system may be met using SCI [1] to provide high-speed point-to-point transmission on parallel links over relatively short distances. The serial HIC technology may enable construction of large low-cost low-latency interconnection networks used to carry and route packets of any size [2].

There are several ways one may analyze a DAQ topology based on point-to-point links prior to realizing it in hardware. One option is to use queueing theory [3], another is using simulation. It is our opinion that the amount of detailed information extracted from a simulation environment can potentially be greater than provided by queueing theory as it is often difficult to foresee and analyze the characteristics of a large topology. For this reason we have developed a simulator capable of simulating interconnected SCI ringlets forming topologies with up to a few hundred nodes [4, 5].

In section two we outline the use of a variant of our SCI simulator; extended to include an  $8 \times 8$  HIC crossbar switch and simulate routing of SCI packets from one SCI ringlet to another [6]. We are currently developing a new simulator for HIC networks based on our experience with

\*This work is supported in part by the ESPRIT 8603 OMI/Macramé project.

<sup>1</sup>Scalable Coherent Interface

<sup>2</sup>Heterogeneous Interprocessor Communication

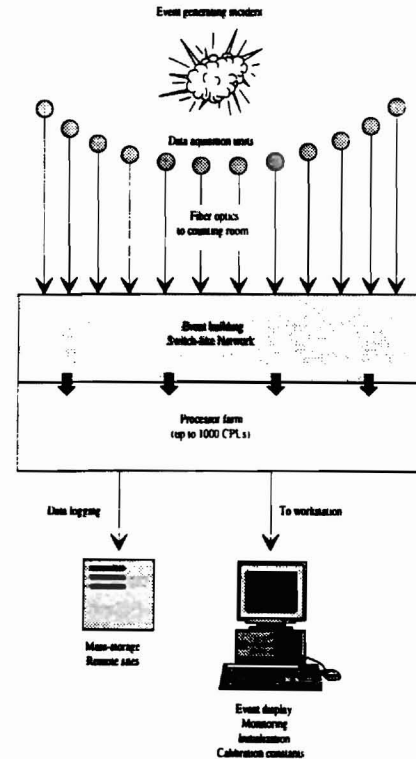


Figure 1.1. The structure of large Data Acquisition (DAQ) systems in high energy physics [7].

the present simulator, and some preliminary aspects of the forthcoming simulator are discussed in section three.

## II. ROUTING SCI PACKETS THROUGH A HIC ROUTER

### A. The simulated topology

The version of our current simulator to be presented here has three basic elements, the SCI-node, the SCI-to-HIC bridge and a HIC router. Several SCI-nodes may be connected on each ringlet, the ringlets may be interconnected using one SCI-to-HIC bridge on each ringlet and the HIC crossbar switch.

To guarantee that the entire network will not deadlock; provided that the HIC network itself does not deadlock, one needs two bidirectional HIC links on each SCI-to-HIC bridge: One for the request packets and one for response

packets. Further, we have only modelled a small  $8 \times 8$  HIC router. Thus a maximum number of four SCI ringlets can be attached to one router.

To illustrate what can be expected when routing SCI packets from ringlet to ringlet using a network consisting of a single HIC router, we have simulated the topology shown in figure 2.1.

### B. Simulation parameters and measurements

In addition to the network topology, the user must also specify:

- The number of SCI-nodes on each SCI-ringlet.
- The maximum number of outstanding requests in each SCI-node, i.e. the number of outstanding transactions.
- An SCI-node is allowed to send out a new request as soon as the response to a previous outstanding request has been received. The time it will wait from receiving the response to generation of a new request is drawn at random according to the uniform probability distribution in the range  $[20 \text{ ns}, t_{\text{delay}}]$ .
- Wire delay between two neighboring SCI-nodes, measured from output link to input link.
- Bypass delay measured from input link to output link in an idle SCI-node
- Bridge delay measured from input link on the SCI-ringlet to the output link on the HIC network side.
- Routing delay of the packet through the HIC switch.
- Time taken to put one packet into a buffer.
- Load factors such as locality. That is the fraction of packets sent to SCI-nodes on the same ringlet as the sending node.

Given these parameters, the simulator measures

- Effective system throughput. From the total byte count of all received packets we subtract the overhead due to packet headers. Then the total number of data bytes received is normalized to the length of the simulation, leaving us with the effective total throughput for the system in bytes per second.
- Latency. Latency is measured from the time the sender puts the packet into its output buffer until the packet is received completely in its destination's input buffer. This is then averaged for all packets sent during the simulation.

The HIC network simulator under development will in addition provide the possibility to detect entities in the network called hot-spots, i.e. having extraordinary large traffic compared to its peers.

### C. Simulation results

We simulated the topology of figure 2.1 as an example on how simulation may be used to answer general questions like

- How much of the theoretical network capacity is realized?

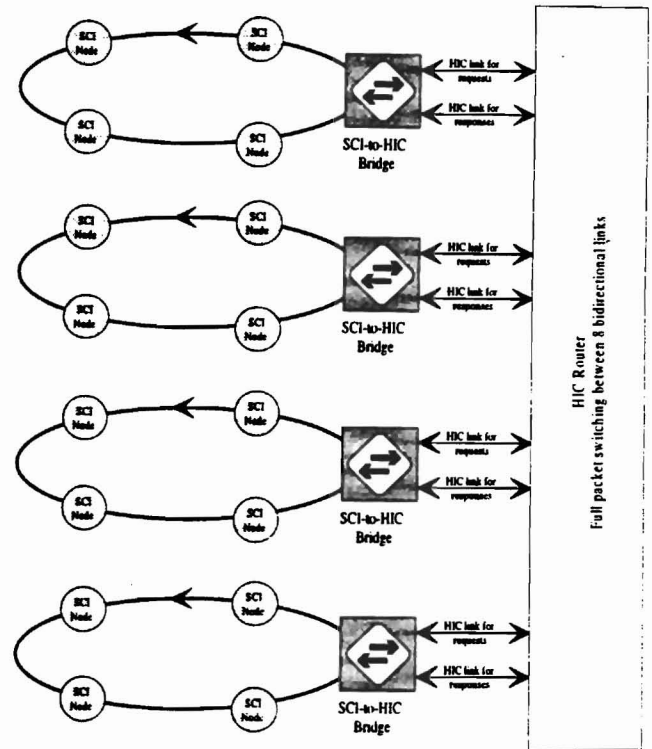


Figure 2.1. The simulated sample topology. Four SCI-nodes attached to each ringlet. The arrows on the ringlets indicate direction of packet flow. Four SCI ringlets are attached to one HIC router.

- When does the system saturate?
- How is the load distributed among the different hardware components?
- How much of a hardware component's theoretical capacity is used?

All SCI packets were taken to be 80 bytes, 16 bytes header and 64 bytes data. The delay time limit was taken to be  $t_{\text{delay}} = 14000$  nanoseconds. The destination SCI-node for a packet was selected at random among the other 15 SCI-nodes in the topology; i.e. a locality of  $3/15$  or 20%. When the SCI packet arrived at the SCI-to-HIC bridge for routing through the crossbar, the bridge prepended a one byte HIC routing header and appended a one byte HIC end-of-packet. These HIC bytes were stripped off again at the bridge on the destination node's ringlet.

The GaAs version of SCI was simulated with high speed (HS) HIC links operating at 1 Gbit/s. The minimum theoretical latency for this topology was found to be  $L_{\text{min}} = 1436$  nanoseconds, and the maximum aggregated bandwidth for the  $8 \times 8$  crossbar, assuming no conflicts over the outputs, is  $R_{\text{max}} = 526.72$  megabytes per second (Mb/s) [6]. As all SCI nodes are equally likely to be the destination of a packet, a fraction  $12/15$  of all packets sent from a node will be sent to nodes on the other rings. The maximum theoretical throughput of the system will be limited by  $R_{\text{max}}$ , and have an expected value of  $T_{\text{max}} = R_{\text{max}} \cdot 15/12 = 658.40$  Mb/s.

In figure 2.2 we have plotted the observed latency against the observed effective system throughput. We have also drawn a least square fit of the curve

$$T(L|\alpha, \beta, \gamma) = \alpha\sqrt{1 - e^{-\beta(L-\gamma)}} \quad L \geq L_{\min} \quad (1)$$

to the data yielding parameter values  $\alpha = 397.88$  Mb/s,  $\beta = 3.06 \cdot 10^{-3}/\text{ns}$  and  $\gamma = 1375.37$  ns.

From figure 2.2 and the fitted curve we observe a clear pattern of saturation: The throughput of the network increases up to a certain level and then flattens out due to a lot of contention in the router. The largest throughput observed was 407.68 Mb/s, thus the fraction of the bandwidth achieved for the network is  $407.68 \text{ Mb/s} / T_{\max} = 407.68 / 658.40 = 0.62$  or 62%. Compensating the largest throughput for the locality, we find that the maximum observed bandwidth for the router is  $12/15 \cdot 407.68 \text{ Mb/s} = 326.30 \text{ Mb/s}$ . Hence, the utilization of the router is  $326.30 \text{ Mb/s} / R_{\max} = 326.30 / 526.72 = 0.62$ , or 62%. This value compares well with the results of others [8, 6]. All the SCI-nodes experienced the same load, as did the bridges.

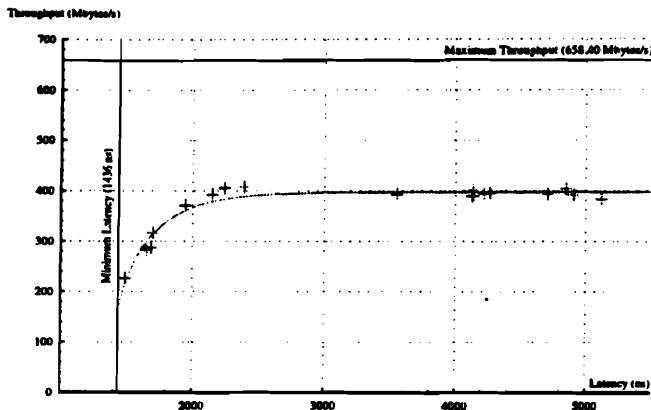


Figure 2.2. Simulation results: Each data point represents a different loading condition. Indicated are also the theoretical asymptotes  $L_{\min}$  and  $T_{\max}$  and the approximating curve given by equation (1).

However, as the complexity of the topology grows, it becomes virtually impossible to undertake a theoretical analysis. Further, the theoretical results may be misleading, or only able to give too wide performance limits, as this simulation showed.

### III. BUILDING A HIC NETWORK SIMULATOR

The burden of one simulation run on a complex topology is significant. When one wishes to simulate for many load factors for each topology, or many different topologies at the same time it is convenient to run simulations on a large number of computers simultaneously. This makes commercial simulation environments with licensing such as MODSIM II<sup>3</sup> too costly. Because of this a standard

<sup>3</sup>Trademark from CACI Products Company, La Jolla, CA. The syntax of the simulation language is derived from Modula 2, hence

computer programming language, C++, was chosen for all our simulator development as it was regarded superior to other object-oriented tools due to its speed and to its availability.

#### A. Event-oriented approach

Our current simulator takes the interval-oriented approach to time advancement [9]; its simulation loop shown in figure 3.1. The simulator's clock is advanced in regular increments of two nanoseconds, and for each increment of the clock, every entity in the topology is given the opportunity to execute. At a clock tick where most of the entities in the network are passive and will do nothing, this approach wastes computer resources by performing excessive context switches. Hence, *fast* simulators cannot use this method.

```

while global clock ≤ maximum simulation time do
  begin
    for each entity in the topology do
      begin
        Give the entity the opportunity to do something
      end
    Increase global clock by 2 nanoseconds
  end

```

Figure 3.1. Interval-oriented approach: Time is advanced in equal steps. For each time step all entities in the simulator are given the opportunity to execute.

A well established alternative approach is *event-oriented* [9], a variant of which we employ. Assigned to each entity in the topology there is an activation time giving the next time an entity will become active. At this time an *activation event* occurs for that entity. We maintain a data structure of the pending activation events sorted on ascending activation times. Conceptually this priority queue data structure can be thought of as a linked list of entities as illustrated in figure 3.2.



Figure 3.2. Event-oriented approach: Conceptual illustration of the list of pending activation events.

We keep a pointer to the first entity in this queue, *CurrentObject*. For this entity, we call a dedicated “do something” function, ask the object to settle for a new activation time, and reinsert it in the queue. The first operation is to *run* an entity while the latter ones are to *reschedule* it. The value of the simulator's clock is set to the activation time of the current object before running that entity. In this way, the clock increment is not a regular value, but allows the simulator's clock to jump to the next time an activity is scheduled to take place.

the name.



Concurrency is simulated by having two or more activation events with equal activation times in the queue. Such concurrent elements are always executed in first in, first out ordering. If the CurrentObject is dependent on actions performed by another concurrent entity, which has not yet run, it has to reschedule with activation time now, and becomes the last object of the ones with now as activation time. When an object is unable to provide a new activation time, it is taken out of the data structure and kept in another unsorted structure, which holds the passive objects. If the entity later goes active, it is put back into the sorted queue. This simulation loop algorithm is given in figure 3.3

```

while CurrentObject points to something do
begin
  with CurrentObject do
  begin
    Update global clock to object's activation time
    Run the object
    Reschedule the object
  end
  Update CurrentObject to next object
end

```

Figure 3.3. The simulation loop to be used in the new simulator. After execution the object reschedules to its new activation time and updates the CurrentObject.

Past experience indicate that as much as 40% of total simulation time may be consumed maintaining the pending activation event queue [10]. Further, the data structure selected for this queue critically influences the execution time [11, 10]. The simple linear list of figure 3.2 is inefficient for all but the very smallest event set sizes. For the efficient structures the cost of one iteration of the simulation loop, is typically proportional to  $\log_2(n)$ , where  $n$  is the number of entities in the activation event set [11].

Although there are many comparisons of such data structures, no single structure is found to perform overall best [12, 11, 10, 13]. Further, the performance is to a certain degree found to be dependent on the environment in which the data structure is used. To find the data structure best suited for our simulator, we intend to implement the three most promising structures and test their performances empirically. The selection of structures to test is based on literature and personal communication [14, 15, 16]: the splay tree [17, 18], the calendar queue [19] and Henriksen's algorithm [20, 21].

### B. Modularity

In contrast to the current simulator the new simulator will fully take advantage of hierarchal modelling, i.e. we are using the inheritance concept in an object oriented programming language such as C++ [22]. An object in C++ is called a *class*. A *base class* is a class passing some or all of its functionality to a *derived class*, and the derived class is said to inherit the base class. Thus one may create a

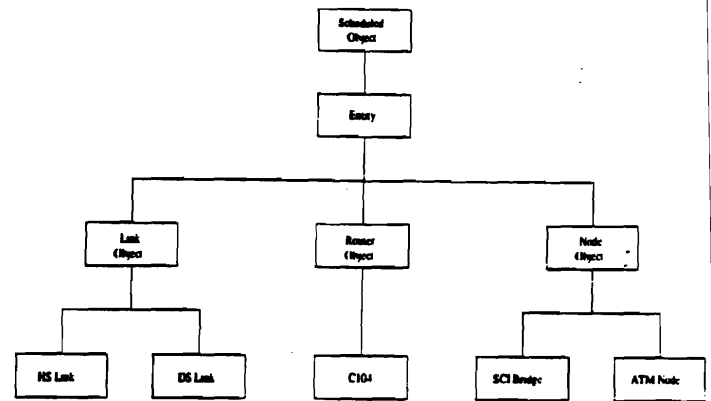


Figure 3.4. Class hierarchy developed for the simulator.

general class that defines things common to a set of related entities. The derived classes may then add only those traits unique to them.

One fundamental base class in our simulator is the *ScheduledObject* class. Any entity in the simulator having an activation event and capabilities of rescheduling must inherit this class. The most important class derived from the *ScheduledObject* is the *Entity* class defining the common functionality of both nodes, links and routers to be simulated. Any object that should be connected into a topology and be able to handle packets must inherit the *Entity* class. Derived from the entity we have generic link, node and router objects. Each defining what every link, node and router have in common, respectively. The specialties of each of the HIC components are defined in sub-classes to these generic ones. The total class hierarchy is shown in figure 3.4.

The reason for employing this structure is when new components become available, it is easy to subclass the corresponding generic class and the rest of the simulator will continue to operate as usual, smoothly interfacing the new object. Hence, the programming effort needed to support new components will be modest. Variants of the current objects may also easily be modelled by deriving new classes from the ones at the bottom of the hierarchy in figure 3.4. In this way, the simulator becomes adaptable and will probably be useful in the future when one starts exploiting the HIC technology for building large networks.

### C. Packet transport

In our present simulator a packet is modelled as a linked list of two-byte (16 bit in parallel) blocks<sup>4</sup> passing through the network. This implies that each two-byte element in the list needs a pointer to the next byte. A pointer is in most programming languages stored as a 32 bit quantity, hence each block takes up  $32 + 2 \cdot 8 = 48$  bits. For a minimum SCI packet of only a 14 byte header and a two byte CRC<sup>5</sup> we need to store 8 blocks which is  $8 \cdot (32 + 2 \cdot 8) = 384$  bits or 48 bytes. This is 200% more memory than

<sup>4</sup>These are called *symbols* in SCI

<sup>5</sup>Cyclic Redundancy Check

strictly needed for storage of the packet. As memory is a limited resource on most computers, this approach is impractical when a lot of fairly long packages are to be simulated.

In the simulator under development the SCI or ATM<sup>6</sup> packet classes must be derived from a common C++ packet class. This generalized packet will be modelled as a collection of bytes characterized by the two integers giving total packet length and the length of the header. The packet length is the sum of the header length and the length of the payload of the packet, not including any end-of-packet control character. The only bytes actually stored in the packet are the header bytes as these will be used for routing the packet through the network. An entity may be allowed to add bytes to this header or take bytes from it. This to support the cases where an SCI packet gets some address bytes added to its header for routing the packet through the HIC network. At the destination node these bytes are stripped off, and the original SCI packet is recovered.

#### D. Validation

A first test on the accuracy of the developed simulator will be to validate the simulated results against predictions done by queueing theory for small and regular networks. Thereafter the results should be compared against measurements from third party real networks. Which networks this should be remains an open question.

### IV. CONCLUSION

We have shown that our current simulator may be used to evaluate a data acquisition network design prior to realizing it in hardware. To explore the possibilities provided by the HIC technology, we are currently developing a simulator tailored for simulation of HIC networks. However, as the simulator exploits modular modelling, it can easily be extended to also simulate SCI or ATM networks. The accuracy of the simulator will be validated against appropriate real networks.

### REFERENCES

- [1] IEEE, *The Scalable Coherent Interface*, 1992, Standard 1596.
- [2] Ernst H. Kristiansen, Geir Horn, and Svein Linge, "Switches for point-to-point links using OMI/HIC technology", in *International Data Acquisition Conference on Event Building and Data Readout*, 1994, This proceedings record.
- [3] Randolph D. Nelson, "The mathematics of product form queueing networks", *ACM Computing Surveys*, vol. 25, no. 3, pp. 339-369, Sep 1993.
- [4] John Weding Bothner and Trond Ivar Hulaas, *Topologies for SCI-based systems with up to a few hundred nodes*, Master thesis, University of Oslo, Department of Informatics, P.O. Box 1080, Blindern, N-0316 Oslo, Norway, Feb 1993. Electronic copy available from anonymous ftp to ifi.uio.no, or WWW, on the file pub/sci/Topology.Thesis.ps.
- [5] E. H. Kristiansen, J. W. Bothner, T. I. Hulaas, E. Rongved, and T.B Skaali, "Simulations with SCI as a data carrier in data acquisition systems", *IEEE Transactions on Nuclear Science*, vol. 41, no. 1, pp. 125-130, Feb 1994.
- [6] John Bothner and Ernst H. Kristiansen, "Simulator report for SCI systems with HIC-router", Tech. Rep., SINTEF Instrumentation, P.O. Box 124, Blindern, N-0314 Oslo, Norway, Sep 1994, ISBN 82-595-8732-7.
- [7] Hans Müller and Andre Bogaerts, "Cern SCI project for data acquisition (RD24 research program)", in *European SCI Workshop*, Stein Gjessing and Ernst H. Kristiansen, Eds., University of Oslo, Department of Informatics, Att: Stein Gjessing, P.O. Box 1080, Blindern, N-0316 Oslo, Norway, Sep 1994, Available directly from the authors, Hans@sunshine.cern.ch or Bogaerts@dxcern.cern.ch.
- [8] M. D. May, P. W. Thompson, and P. H. Welch, *Networks, Routers and Transputers: Function, Performance, and Applications*, INMOS Limited, 1000 Aztec West, Almondsbury, Bristol, BS12 4SQ, United Kingdom, 1993, ISBN 90-5199-129-0. Electronically available from anonymous ftp to inmos.co.uk on the directory inmos/info/comms/book.
- [9] Geoffrey Gordon, *System Simulation*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA, 1969.
- [10] K. Chung, J. Sang, and V. Rego, "A performance comparison of event calendar algorithms: an empirical approach", *Software-Practice and Experience*, vol. 23, no. 10, pp. 1107-1138, Oct 1993.
- [11] Douglas W. Jones, "An empirical comparison of priority-queue and event-set implementations", *Communications of the ACM*, vol. 29, no. 4, pp. 300-311, Apr 1986.
- [12] William M. McCormack and Robert G. Sargent, "Analysis of future event set algorithms for discrete event simulation", *Communications of the ACM*, vol. 24, no. 12, pp. 801-812, Dec 1981.
- [13] Stavros D. Nikolopoulos and Roderick MacLeod, "An experimental analysis of event set algorithms for discrete event simulation", *Microprocessing and Microprogramming*, vol. 36, pp. 71-81, Mar 1993.
- [14] Douglas W. Jones, "Algorithms for event set manipulation", Jul 1994, Personal electronic mail.
- [15] Jean Vaucher, "Algorithms for huge pending event sets", Jul 1994, Personal electronic mail.
- [16] Jeffrey Kingston, "Algorithms for huge event set manipulation", Jul 1994, Personal electronic mail.
- [17] Daniel Dominic Sleator and Robert Endre Tarjan, "Self-adjusting binary trees", in *Proceedings of the ACM SIGACT Symposium on Theory of Computing*, 1983, pp. 235-245, Association for Computing Machinery (ACM).
- [18] Daniel Dominic Sleator and Robert Endre Tarjan, "Self-adjusting binary search trees", *Journal of the Association for Computing Machinery*, vol. 32, no. 3, pp. 652-686, Jul 1985.
- [19] Randy Brown, "Calendar queues: A fast O(1) priority queue implementation for the simulation event set problem.", *Communications of the ACM*, vol. 31, no. 10, pp. 1220-1227, Oct 1988.
- [20] Jeffrey H. Kingston, "Analysis of Henriksen's algorithm for the simulation event set", *SIAM J. Comput.*, vol. 15, no. 3, pp. 887-902, Aug 1986.
- [21] James O. Henriksen, "Event list management - a tutorial", in *Proceedings of the Winter Simulation Conference*, 1983, pp. 543-551. IEEE, Conference held in Piscataway, New Jersey.
- [22] Herbert Schildt, *C++: The Complete Reference*, Osborne McGraw-Hill, Berkeley, CA, USA, 1991, ISBN 0-07-881654-8.

<sup>6</sup>Asynchronous Transfer Mode; Packet format chosen by the CCITT as basis for the Broadband Integrated Services Digital Network (B-ISDN)



## **Application of SCI in the STAR data acquisition system**

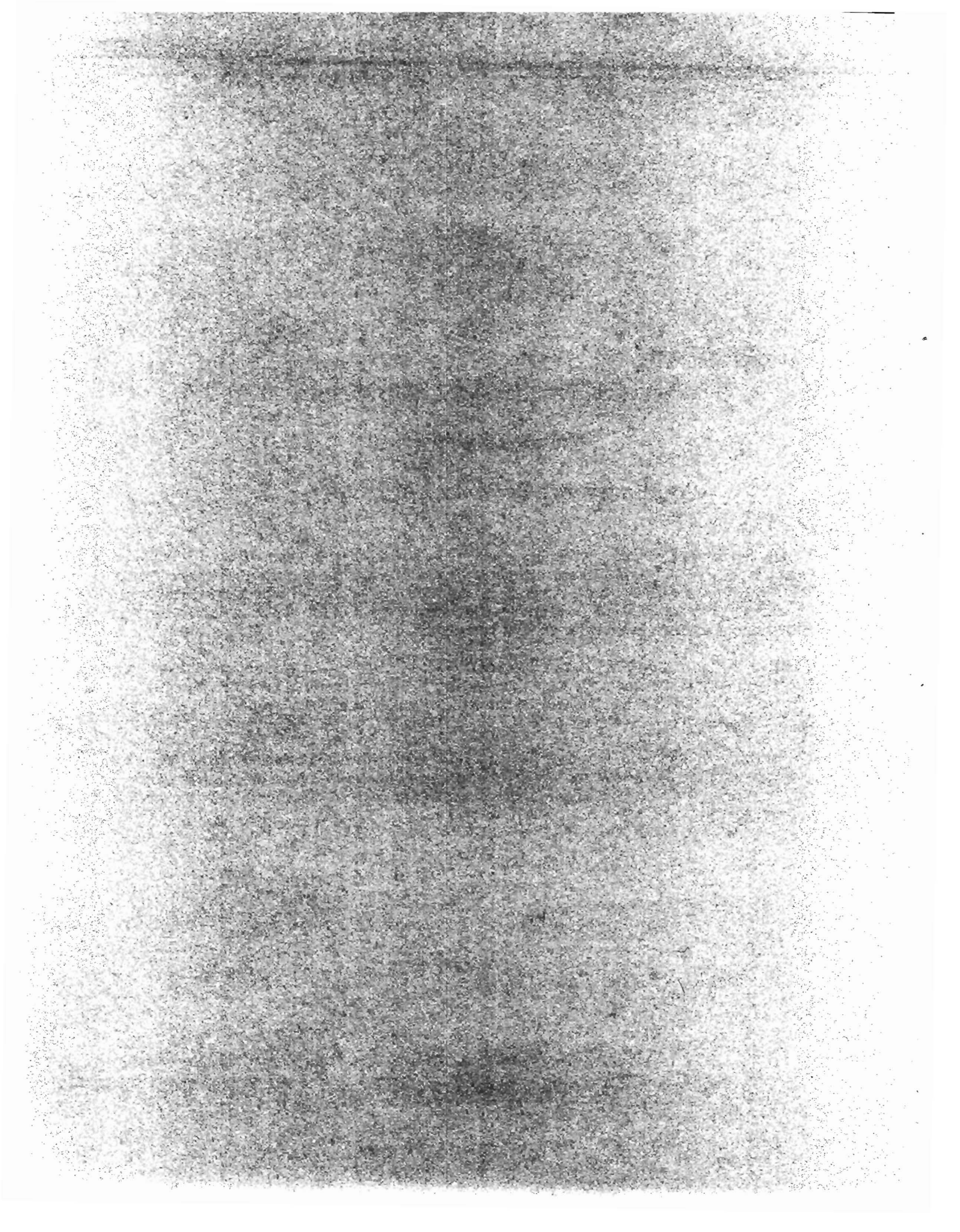
**Volker Lindenstruth**

**Lawrence Berkeley Laboratory**

STAR (Solenoidal Tracker At RHIC) is a large scale high-energy detector. It has 140000 TPC channels and 103000 channels of a Silicon Vertex Tracker. Although the trigger rate of 100/sec is low, the data volume is very high due to the large channel count and high particle multiplicity of about 2000 charged particles per central Au+Au event. Current tape technology allows only about one event per second to be stored.

STAR will have a third level trigger processor farm to perform the required selection of the interesting events. The requirements of the third level trigger algorithms towards data rate and connectivity will be outlined. It will be shown how this is planned to be implemented within the SCI framework.

The most important features of SCI for the STAR setup will be emphasized. One important component for the STAR system is a bridge that interconnects PCI with the SCI network. It will be shown what concrete requirements are existing for that device and how we plan to implement it.



# STAR SCI Backbone

Volker Lindenstruth  
Lawrence Berkeley Laboratory  
1 Cyclotron Rd, M/S 50D Berkeley, CA 94720  
lindenstruth@lbl.gov



## Abstract

*This paper describes the STAR SCI network topology. The data flow requirements and data rate requirements will be discussed.*

*For STAR an SCI-PCI bridge is very important. The requirements for the SCI-PCI bridge are outlined.*

## 1 Introduction

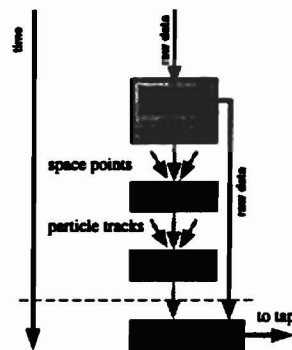
STAR (Solenoidal Tracker At Rhic) is a large scale high-energy detector. It has 140000 TPC channels (1024 time buckets per channel) and 103000 channels (256 time buckets per channel) of a Silicon Vertex Tracker. The data of all other detectors is comparably small. Although the accepted trigger rate of 100/sec for TPC and SVT is low, the data volume is very high due to the large channel count and high particle multiplicity of about 2000 charged particles per central Au+Au event. Current tape technology allows only about one event per second to be stored. Consequently STAR will have a third level trigger processor farm to perform the required selection of the interesting data.

SCI is an approved IEEE standard (1596) defining a high speed (1 GByte/sec) split transaction network. It merges the shared memory concept with a bus-like architecture. SCI is an ideal network for closely coupled multiprocessor systems like data acquisition systems. The remainder of this paper will describe what SCI topology is planned for the STAR setup.

## 2 Structural Architecture

The most demanding part of the system is the interface between the third level trigger processor farm and the TPC and SVT receiver boards. These detectors produce the largest amount of data (refer to table

1). Since the third level trigger processor farm will be designed to select only  $10^{-2}$  of the events that were not vetoed by trigger level one and two, the required bandwidth to communicate between the TPC and SVT receiver boards and the third level trigger processors is about two orders of magnitude higher than the taping requirements. Since the existing requirement estimates are based on simulations and are therefore uncertain it is important that the proposed interface is flexible enough to accommodate later upgrades and changes.



*Figure 1: The data flow hierarchy in STAR. The uncompressed raw data is sent to receiver boards in the counting house. The next steps of analysis are grouped in layers with a given requirement of connectivity.*

Figure 1 shows a sketch of one way to implement the logical data flow within STAR. The uncompressed raw data is shipped off the detector using the HP GigaLink chips. It is stored in event buffers on the DAQ front-end the receiver boards. The front-end processors on the receiver boards will derive space points that are shipped to the next processing layer the third level trigger local sector or segment track finder. The next processing layer the global level three layer is the first layer combining information of several detectors. A fourth layer of processing is currently discussed. This processing layer may be required if the time required to perform the L3 local and global analysis exceeds the available pipeline buffer space on the receiver boards. In this case the compressed raw data of a not yet accepted or vetoed event would be read out and sent to the L4 processors.

Another trigger analysis scenario does not imple-

ment a global selection/rejection algorithm like the one previously sketched but an intelligent data reduction algorithm. For example the first levels of analysis could be performed in the third level trigger farm and only the results sent to tape. The advantage of this implementation is the ability to record higher event rates than one event per second.

There is a large variety of structural scenarios of how to implement the third level trigger processor farm. However one particular architecture appears desirable taking into account that the bulk of the TPC and SVT trigger analysis (hit and track segment level - L3 local in figure 1) are completely independent. Consequently the first level of analysis can be done on a sector by sector basis. Figure 2 shows a third level trigger data flow diagram resulting from these assumptions. Each TPC L3 sector network would be comprised of 6 receiver boards serving one TPC sector. Each SVT segment network would be comprised of three receiver boards. The number of L3 local processors is not defined yet. The TPC and SVT tracks are sent from each L3 local network to the L3 global processor farm. At this processing level the summary data of all STAR detectors will be merged and a trigger decision derived.

Table 1 shows a breakdown of the data sizes of the objects that have to be moved at the various layers of the systems. The minimal data rate requirements neglecting processor synchronization and event monitoring traffic can be derived by taking into account that that system has to sustain 100 events

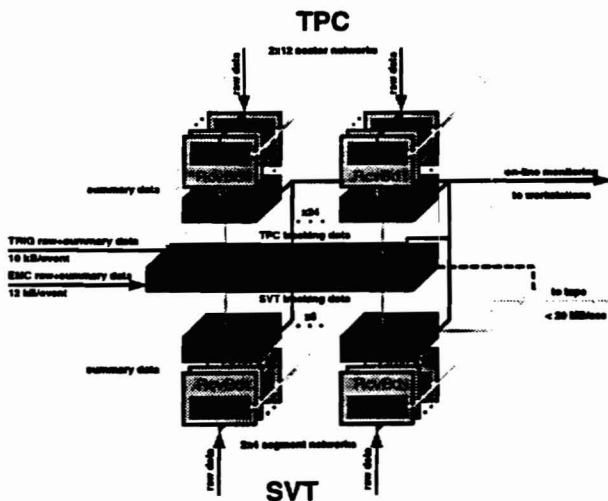


Figure 2: The STAR third level trigger data flow diagram.

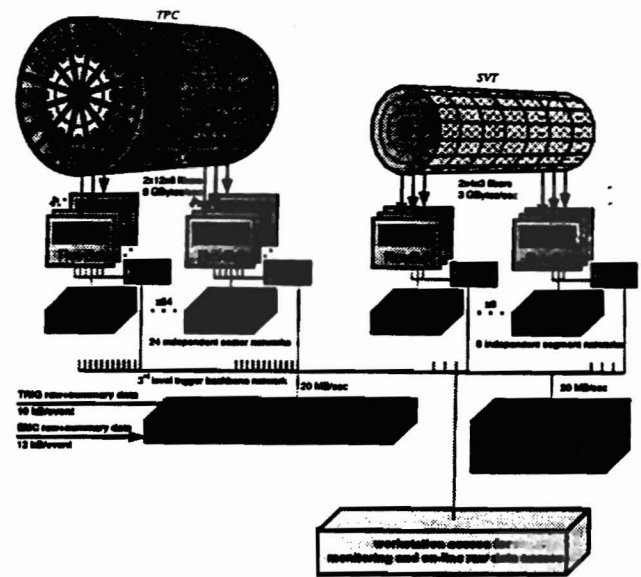


Figure 3: The STAR SCI topology.

per second. For example, the minimal data rate requirement of the TPC sector network is 5.4 MB/sec if the system runs in the event selection mode and 70MB/sec if it runs in event analysis and compression mode as outlined before. In this scenario, the zero suppressed raw data would have to be sent to the L3 local processors since the front-end processors on the receiver boards do not have floating point capabilities.

Figure 3 sketches the STAR SCI network architecture. All local L3 networks are connected through bridges to a global network (third-level trigger backbone). This network serves several functions: data transport mechanism for the L3 summary data to the L3 global processor farm, event build and read-out, and on-line monitoring access and transport mechanism for local network cross communication.

## 1 Implementation

In order to build the STAR data acquisition and third-level trigger system, using SCI as the underlying network, SCI interfaces to all components involved in the system are required. The receiver boards will have multiple front-end processors that are all connected to a global bus. This bus was chosen to be PCI since there are appropriate interface chips available. Consequently, for the receiver board interface an SCI-PCI interface would be required. There is no decision yet which processor architecture will be

Table 1: STAR third level trigger data volumes

	TPC sector	SVT segment
<b>raw data</b>	5.8 MB	3.3 MB
<b>occupancy</b>	10 %	3.9 %
<b>zero-suppressed data</b> (including 20% overhead)	700kB	160 kB
<b>space point data</b> (6 real numbers per space point, outer 16 pad rows only)	54 kB	27 kB
<b>tracks</b> (10 real numbers per track)	80 kB	80 kB

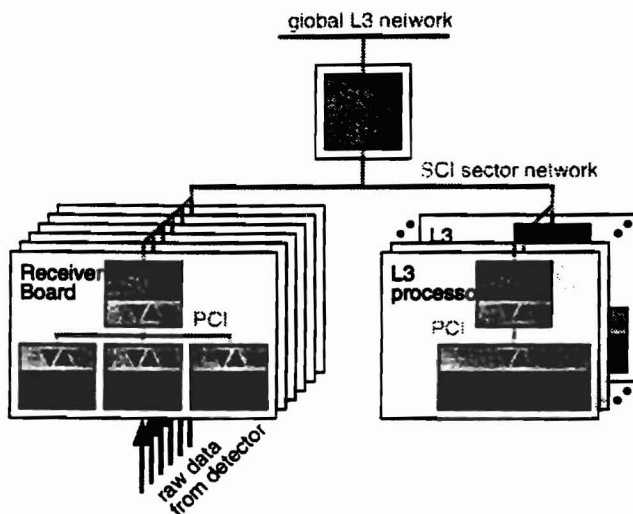


Figure 4: The TPC sector network implementing the SCI-PCI bridge.

used for the STAR third-level trigger farm. However, the DEC alpha and the PowerPC architecture are strong candidates. Most implementations of these processors use PCI as the I/O bus. Consequently an SCI-PCI bridge appears very useful for the third-level trigger framework, too. Figure 4 sketches a TPC sector network using an SCI-PCI bridge to connect all components to the SCI sector network. The first prototypes of the TPC and SVT receiver boards will, however, be VME based. They will have a PCI slot available for the SCI-PCI bridge as a migration path to the final design.

Certainly there will be VME systems used within STAR. In order to integrate these systems into the SCI network an SCI-VME bridge is required. The STAR architecture as outlined does not require complex switches. It is intended to build the required bridges using SCI NodeChips in a back-to-back configuration.

## 1 SCI-PCI Bridge

A prototype of an SCI-PCI bridge is currently being designed as a joint effort between STAR and RD24 (CERN). The essential requirements of that device are:

- Address translation from PCI 32 to 64-bit SCI address.
- Memory protection against incoming SCI write requests.
- Transparent read/write functionality for both PCI→SCI and SCI→PCI transactions.
- SCI lock transaction support – most important are Fetch&Add and Compare&Swap.
- Chain-mode DMA support on bridge.
- It is required to be able to initialize the bridge and correspondingly the far-end bus or network from both SCI and PCI.
- Data transfer rate SCI→PCI write 70MB/sec, PCI→SCI (DMA) write 30MB/sec. Read transfer rates depend on latencies on the responder side. Read-ahead scenarios may be implemented to amortize the latency over several read requests.
- Low cost

## Acknowledgements

I want to thank Mike LeVine and Hans Mueller for the excellent support and many fruitful discussions.

My work is partly supported by the German Humboldt program.





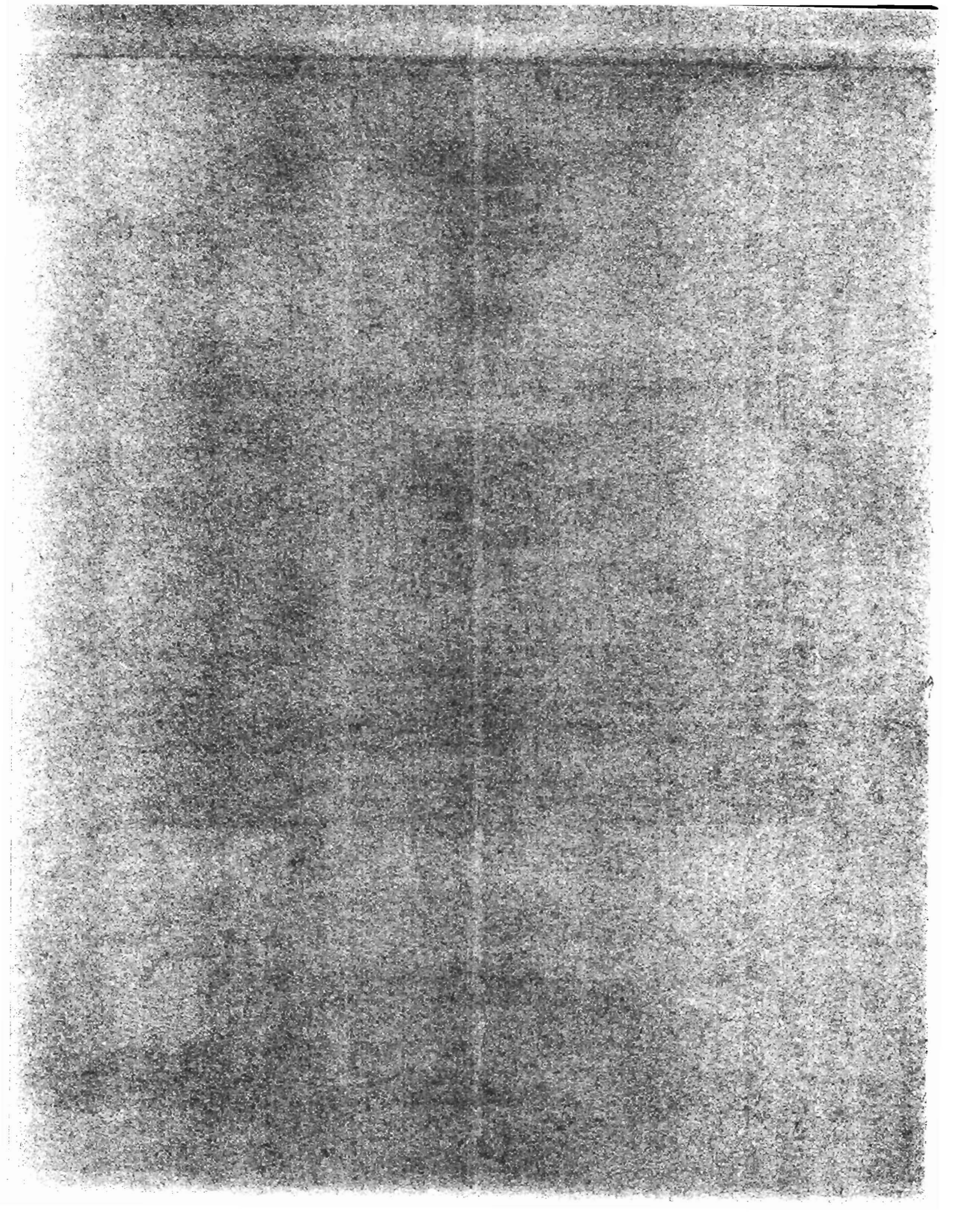
## **Event Building Using an ATM Switching Network in the CLAS Detector at CEBAF**

**David C. Doughty Jr.**

**CEBAF**

In the CLAS detector at CEBAF an ATM switching network will be used to move data from front-end readout controllers (ROC's) to the on-line processor farm. To avoid congestion problems and cell-loss, a linked token-passing system has been devised, with two different types of tokens being passed through the switch. The processor token controls which processor (or processor cluster) will receive a specific event block, while the ROC token controls which readout controller will pass fragments from that event block to that processor.

Multiple processor tokens may be active simultaneously, each with its own associated ROC token, leading to a 'barrel shifter' type of parallel data transfer. We describe the hardware planned, the token passing and allocation scheme, and some preliminary simulation results.



# Event Building Using an ATM Switching Network in the CLAS Detector at CEBAF

David C. Doughty Jr., David Game, Lisa Mitchell  
Christopher Newport University, Newport News, VA, USA

Graham Heyes, W. A. Watson III  
Continuous Electron Beam Accelerator Facility, Newport News, VA, USA

## Abstract

In the CLAS detector at CEBAF an ATM switching network will be used to move data from front end readout controllers to the on-line processor farm and to tape. To avoid contention problems and cell-loss, a linked dual token passing algorithm has been devised, with two different types of tokens being passed through the switch. The event request token controls which processor (or processor cluster) receives a specific event block, while a data request token controls which group of readout controllers will pass fragments from that event block to that processor at a given time. Multiple data request tokens may be active simultaneously each with its own associated processor which is to receive the event. This leads to a 'barrel shifter' type of parallel data transfer. We describe the hardware planned, the token passing and event allocation algorithm, and some preliminary simulation results.

## I. INTRODUCTION

The CEBAF Large Acceptance Spectrometer (CLAS) is designed for kinematic analysis of several particles in the final state of nuclear interactions. A toroidal magnetic field generated by six coils is used for momentum analysis; for this reason the detector package has been partitioned into six wedges or sectors. Each sector fits between two adjacent coils and consists of four types of detectors: six superlayers of drift chambers for tracking, and Cherenkov detectors, scintillation counters, and an electromagnetic calorimeter for particle identification. Details of the detector design are given in reference [1].

The CLAS detector is designed to run at luminosities exceeding  $10^{34} \text{ cm}^{-2}\text{s}^{-1}$  producing a hadronic interaction rate of several Megahertz. The data acquisition system is being designed to handle event sizes of 20-40 Kbytes and an expected event rate in excess of 2 KHz. A two level hierarchical trigger system [2] selects events of interest, communicating with a trigger supervisor module [3] which controls the conversion and readout sequence. After the front end electronics modules have finished conversion and local buffering, the trigger supervisor places the event in a first-in-first-out (FIFO) queue to communicate with the readout controllers (ROCs). Readout then proceeds asynchronously with the acquisition and conversion of future events. There are approximately twenty ROCs in the CLAS detector, each reading out a portion or fragment of the data from one event. The accumulation of

fragments of data from all of the readout controllers requires some method of 'event building' or consolidating all fragments from one event so they may be analyzed (for possible trigger cuts) and written to tape. The architecture of the event builder planned for the CLAS detector is the subject of this paper.

## II. COMPARATIVE EVENT BUILDING ARCHITECTURES

### A. Overview

A block diagram of a typical event building architecture is shown in Figure 1. The event builder has input connections from many ROCs, each passing a fragment of an event. The output connections go to a series of on-line farm processors (OLFPs), which may reformat the data, partially analyze it for monitoring or triggering purposes, and write it to tape. The function of the event builder is to route all the data from a specific event (or block of events) to one of the OLFPs, the data from the next event or block to a different processor, and so on. In this way the computational load is distributed among many processors.

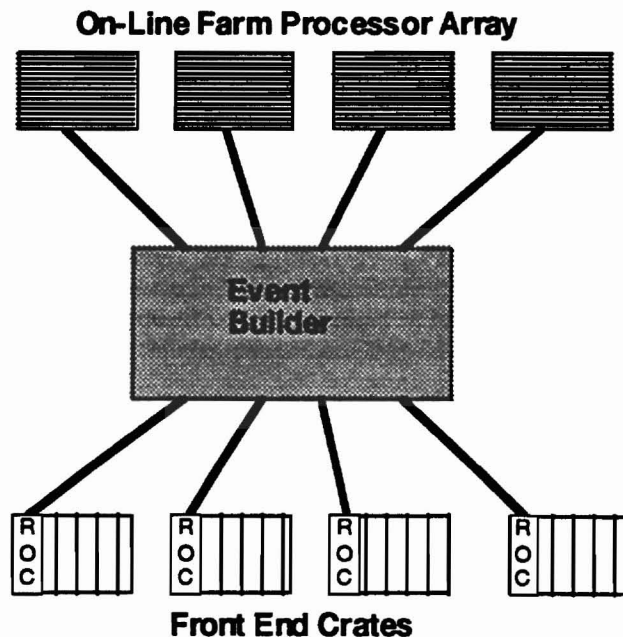


Figure 1. A block diagram of a typical event building architecture concentrating on the input and output data links. Details inside the event builder are not shown.

All types of event builders must not only have a mechanism for delivery of data from the ROCs to the OLFPs, but must also have some mechanism for deciding which OLFP gets which event. This information must then be used to control the data flow from the ROCs so that it reaches the correct destination. Many approaches to event building have been used or proposed, and we first discuss non-switched architectures, then switched architectures and ATM.

### *B. Non-Switched Architectures*

Older event building architectures typically involved gathering data from several ROCs into one node, then gathering the data from several of those nodes into a secondary node and so forth until all the data had arrived at the topmost node. This pyramidal scheme has a basic weakness in that the amount of parallelism is reduced as the data flows up, creating bottlenecks which can severely impact the data rate. The original CDF event builder operated this way, and the event rate (with a single event builder) was limited to 10-15 Hz [4].

A more modern non-switched approach involves using a memory array in which each input link connects to all memories in one column of the array, while each output link connects to all memories in one row of the array. All ROCs write the data from a specific event into the same memory in the column they are connected to, so that the entire event ends up in memories in the same row. The next event is written to the next row, and so on. Each OLFP obtains an event by reading the fragments from all memories in its row. This approach has been used by the D0 experiment at Fermilab (among others) and is more fully described in reference [5].

Another approach is to use a high speed network to pass the data from ROCs to OLFPs. If the bandwidth is sufficiently high each ROC can pass the data to a specific OLFP over the network. This design has been used in the CDF upgrade [6], where the Ultrane network was chosen. FDDI networks could also be used in this fashion.

Each alternative design has weaknesses. In the memory array architecture, the number of memories grows as the product of the number of input and output links. Also, some mechanism for controlling the memory access must exist, whether by directly controlling the memory array, or broadcasting to the ROCs. Memory based designs typically involve some custom hardware and software to make them function. Use of a single high speed network will ultimately limit the readout speed to the capacity of the network. Pushing the bandwidth to the limit may necessitate another network for control, as in the use of the SCRAMNET [7] network at CDF. Switched network architectures and the large aggregate bandwidths they currently provide can solve many of these problems.

### *C. Switched Architectures and ATM*

Switching architectures offer many attractive features for use in event building. In a switched architecture, each of the input links from the ROCs is connected to one port of the switch. Each of the output links to the OLFPs is also

connected to one port of the switch. In this architecture all of the ROCs send data from a given event or event block to the same OLFP. The switch controls the routing of the data, so all fragments of one event reach the proper destination.

There are a few complications in using the switch which need to be addressed. As in all event builder models there needs to be a way to determine which of the OLFPs is to receive the next event. This information must then be propagated to the ROCs so they know the destination. An additional problem for a switched architecture is that because access is typically unconstrained (unlike a token ring network), there must be some way of controlling the access and routing, for if all ROCs attempt to send data to the same processor at one time, there will be contention and possibly buffer overflow at the output port.

Asynchronous transfer mode or ATM is one of the newest switching architectures and is rapidly gaining wide acceptance. In ATM all data is transferred in 53 byte cells, comprised of a 5 byte header and 48 bytes of data. The header controls the routing of the cell through the network switches. This small and fixed cell size is designed to provide the low latency switching required by integrated video and data networks. ATM does not provide guaranteed delivery of cells, and ATM switches typically have limited buffering, so contention for output bandwidth may cause cell loss. This is not a serious problem in video applications, but for data transfer a higher level protocol must retransmit lost cells. In event builder applications this type of contention and retransmission may drastically reduce the throughput, so a way to eliminate it must be found in order to use ATM.

There are two other switched architectures which have been proposed for use in DAQ event building; switched FDDI and Fibre Channel. FDDI is somewhat limited in that its transmission speed is currently only 100 Megabits per second (Mbit/s) while both Fibre Channel and ATM offer several higher transmission speeds. A relative weakness of Fibre Channel is that there are only one or two switch vendors, and the supply of interface cards is also limited. By contrast ATM has been embraced by many manufacturers, and switching and interface cards are currently available from a number of sources. The problem of cell loss and retransmission may be somewhat worse in ATM than in some of the other switching architectures, but the increasingly wide acceptance and availability of ATM technology makes its use very attractive.

In the next section we discuss the use of an ATM switch as the event builder in the CLAS detector, and how the switch in combination with the dual token passing algorithm is used to solve the communication and contention problems mentioned above.

## III. EVENT BUILDING USING ATM

### *A. Overview*

Figure 2 shows a block diagram of the event building architecture to be used in CLAS. In this design the data from all ROCs is collected by six data concentrators (DCs) which

are connected to the ATM switch. The ATM switch has 16 fiber optic OC-3 (155.52 Mbit/s) ports, and serves as the vehicle to route both data and control information between OLFPs, DCs, and the tape processor. Communication is done using the TCP/IP sockets protocol, which guarantees reliability without having to rewrite applications to use a native ATM application programming interface (API). Six of the ports are used for the data concentrators leaving 10 for the OLFP network and the tape processor. The OLFPs merge the event fragments into complete events, perform partial analysis, and send the events back through the switch to the tape processor which controls the writing to tape.

### B. Input Data and Rates

The DCs are actually VMEBUS processors which collect the data from several FASTBUS and VME ROCs, merge the data into one stream of event fragments, and manage communication with and data transfer to the OLFPs. FASTBUS crates are read out using the FRC board developed at Fermilab [6] and data is transferred to the DCs over the scanner bus interface. VME crates transfer their data to the DC using a VME-to-VME interconnection.

Introducing the DCs solves two problems. There are more than twenty ROCs in CLAS, too many to connect each one to a switch port unless the switch is very large and expensive. Since the data rate from a single ROC is not very high (between 1-3 Mbyte/s depending on the electronics in the crate), it is a waste of bandwidth to dedicate a 155 Mbit/s link to each one. Reading the data from several ROCs into one DC greatly reduces the number of input data links needed for the switch. The second problem is that there are no FASTBUS interfaces to ATM, while several are available for the VME bus, with more being announced all the time.

Three of the DCs handle PMT based electronics channels and trigger information, and are each expected to have about 1 Kbyte of data per event. Even though the data in each of these DCs is not very large, the underlying electronics are quite separated geographically, making it difficult to combine them. The other three DCs handle the FASTBUS crates reading out the drift chamber data. In addition to handling network traffic these DCs perform valuable data compression. In the CLAS detector pulse width encoding has been used to multiplex two drift chamber wires onto one TDC channel, which reduces the channel count but increases the amount of data read out [8]. The digitized data being read out is compressed by a factor of

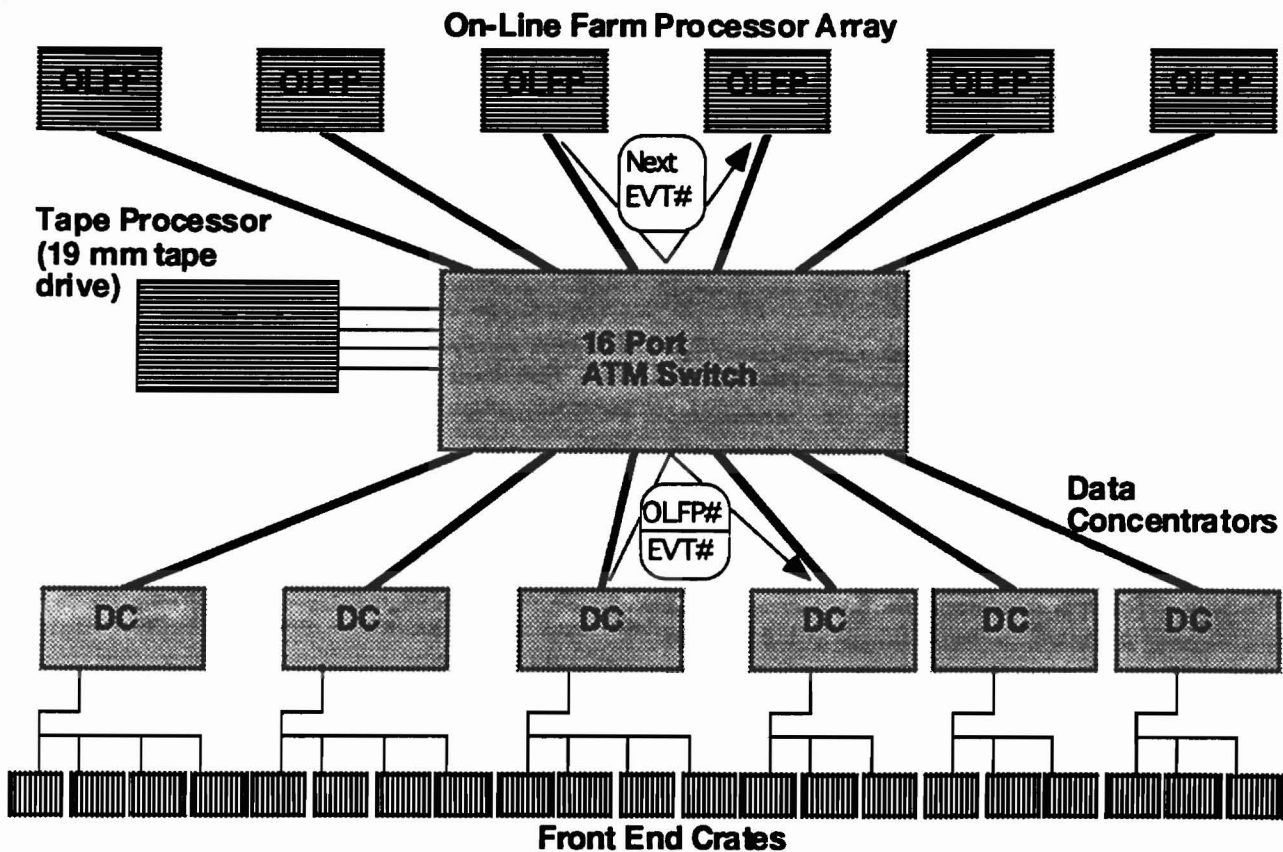


Figure 2. The event building architecture used in the CLAS detector. An ATM switch serves all communication needs, including token passing for control as well as data movement. The event request token can be seen circulating among the OLFPs at the top, while a data request token is circulating among the DCs at the bottom.

2.7 in the process of separating the two channels and recovering the original time and charge. This data extraction and compression can be done on a sizable fraction of the events by the FRCs themselves, but the DCs must finish this compression before sending the events out. This can be done by an additional processor in the crate. After compression the data from one event will be about 2.5 Kbytes at each drift chamber DC.

The total event size at the DC stage is therefore about 10.5 Kbytes. At an event rate of 2 KHz each of the drift chamber DCs needs to move data into the OLFPs at 5 Mbytes/s, with the other DCs needing to transfer at 2 Mbytes/s. This yields an aggregate rate of 21 Mbytes/s, significantly below the rates at which an ATM switch can perform. The difficulty thus lies not in the amount of data being transferred, but rather in smoothing the data flow to minimize contention. We now discuss the algorithm which handles this.

### *C. Tokens and Transfer Control*

As mentioned previously, the major problem in using an ATM switching network for event building is contention on the outbound links to the OLFPs, causing cell loss and retransmission, reducing the throughput. We have developed an algorithm using two types of tokens which not only solves this output link contention problem, but also handles the arbitration among OLFPs for events and communicates this information to the DCs. In this algorithm one token circulates among the OLFPs, which they use to arbitrate for events, while a second type of token (of which many may be outstanding) circulates from the OLFPs down to the DCs and back. Both tokens are passed from source to destination through the switch, so that the switch is used for control information as well as data movement.

To reduce the amount of control information needed, all arbitration and data transfer is done using blocks of 64 events. This greatly reduces the number of messages, acknowledgments, and interrupts involved. At an event rate of 2000 Hz the OLFPs need to arbitrate for only 33 blocks each second.

The OLFPs pass an event request token among themselves, in a round-robin fashion, using the switch. This token is shown in Figure 2 in transit between two OLFPs. As each processor receives the token, it checks to see if it is able to accept more events. If so it increments the 'next available event number' field by 64 and passes the token to the next OLFP in the chain, otherwise it passes the token unmodified. Because a processor may occasionally not take an event block, the token circulates slightly faster than the 33 event blocks per second which need to be accepted. This algorithm expects the OLFPs to take the events in a round robin fashion most of the time, and data transfer may slow down if much processor skipping occurs. We discuss this in more detail below.

Once a processor has taken a specific block of 64 events (beginning with event number K for instance) for itself it passes a data request token through the switch to the first DC,

identifying itself as the recipient of the block of events beginning with number K. Each processor may have more than one outstanding data request token, to keep the arbitration ahead of the data flow.

The first DC receives data request tokens from all of the farm processors. It places them in a queue in order of requested event numbers (not in order of received requests, which may be incorrect). When the event block requested by the token at the head of the data request queue is available (for the OLFPs should be arbitrating ahead of the data's arrival), this DC sends the event fragments to the destination processor, and then passes the data request token to the next DC.

The second DC receives the data request token from the first and places it in its queue. As soon as all previously requested event fragments have been sent, and the data requested by this token is available, it will be transmitted to the same processor. The data request token will then be passed to the DC next in line. This process is repeated from one DC to the next, with the order being fixed, until the last DC has sent its data. The data request token is then passed back to the first DC which returns it to the destination processor. Figure 2 shows a data request token in transit between two of the DCs.

The net effect of this algorithm is a pseudo-barrel-shifter approach in which ideally, at a given instant in time, each DC is holding a data request token from a different OLFP. There will be no contention among data transfers because the DCs will all be sending data to a different destination. The only contention that could arise in this situation would be between token messages and data. Because the token messages are short (about two cells since most of one cell is taken up by the TCP and IP headers) and not very numerous this should not be a problem.

This highly efficient throughput will only occur if two DCs don't attempt to send data to the same OLFP at the same time. This could happen if one OLFP takes two event blocks reasonably close together because several of the processors passed the event request token without taking an event block. Consider the scenario in which a certain OLFP passes down two data request tokens which are separated by only one other token from a different OLFP. The first DC will then send data from the first block to the 'over requesting' OLFP. The token then passes along the chain of DCs. But as soon as the first DC sees the second token from this OLFP, it will want to send a different set of event fragments to the same destination. This will almost certainly cause a conflict with a later DC which is trying to transmit its fragments of the earlier event to that same OLFP.

There are two conditions which, if met, minimize the impact of this problem. First, the number of farm processors must be greater than or equal to the number of DCs, or else this contention problem may arise even if no processors are skipped. Second, the OLFPs must take an event block when the token comes to them most of the time, and only occasionally let it pass unmodified.

If this problem does occur there are two ways to handle it. One way is to just let the switch and the TCP protocol handle

the contention, assuming it happens infrequently. When it does occur it will cause retransmission if cells are lost, reducing the throughput. The other approach is to allow the first DC to stall the second 'conflicting' data request token until the prior token from the same OLFP has been returned to it, indicating that all the DCs have finished sending their event blocks to that processor. This will also reduce the throughput of the switch. We are actively preparing simulations of the operation of this switching architecture, but because different switches have different buffer sizes and cell lost rates the final choice between these two methods will probably be made using actual tests of the system at our data rates.

An alternative approach would be to have one master OLFP which is responsible for arbitrating the event flow. Each OLFP would request an event block, and the master OLFP would return a token to the processor who was granted a token least recently. This has the disadvantage of requiring two transmissions for each event block while the round robin method requires slightly more than one. Other disadvantages include different and additional software running on the master, and the loss of processing power.

#### D. From the Farm Processors to Tape

Once all the fragments for a certain event block have been received by one of the OLFPs, it must reformat the data to put it together into complete events before writing it to tape. Some amount of analysis may also be done for monitoring or triggering purposes, but it has lower priority than the data reception and reformatting tasks.

The data rate into each OLFP will be the aggregate data rate of the DCs divided by the number of OLFPs. The total rate from the DCs is 21 Mbyte/s, so with six OLFPs the data rate into each will be about 3.5 Mbyte/s. If we assume that the analysis adds a little over 10% to the data and that no events are rejected by a triggering algorithm, the output from each OLFP should be about 4 Mbyte/s.

In this architecture all OLFPs send events which are to be written to tape to the tape processor. This will probably be a multiprocessor machine, to handle the demands placed upon it. It will have at least two high speed disk arrays for buffering data, and at least one high speed (16 Mbyte/s) tape drive. Because the input data rate can be as high as 24 Mbyte/s (6 OLFPs each with 4 Mbyte/s to tape) it has four ATM links to the switch with each one having a load of 6 Mbyte/s. Because data blocks received from the OLFPs may be out of order due to different processing times for the event blocks, they are internally reordered by event number. Events are taken off the top of the queue and written to one of the two disk arrays. When one disk array is nearly full, events are written to the other disk, while the first disk is emptied onto the tape.

### IV. HARDWARE TESTS AND SIMULATION

#### A. Hardware Tests

The rate at which data can be sent over the ATM links and the computational load required to support the TCP protocol

are critical factors in determining the success of this architecture. As mentioned above each DC will transmit data at rates up to 5 Mbyte/s, while the OLFPs will receive up to 3.5 Mbyte/s and transmit up to 4 Mbyte/s. The tape processor must be able to receive 6 Mbyte/s on each of four ATM links, and buffer them to disk, then transfer them onto tape.

To test the performance of ATM interfaces and the computational load required to run TCP over ATM, two EISAbus fiber optic ATM (OC-3) cards for Hewlett Packard computers were obtained from FORE Systems[9]. One was placed in an HP 735/125, the other in an HP 715/75, and the machines were connected by 1150 m of fiber. The driver software for these boards included a standard Berkeley sockets interface, which simplified testing. Two tests were conducted, one using the sockets interface directly, and one running CEBAF's DAQ software CODA [10], which uses sockets as its transport mechanism over any link.

For the sockets test 8 Kbyte packets were sent from one machine to the other as quickly as possible, and the data rates and computational loads were measured. Header checksums were computed but data checksums were not, although this had little effect on the performance. The TCP sliding window size was set to 56 Kbytes, which gave the best performance. The results of this test are shown in Table 1.

The CODA test simulated the conditions which would exist in sending data from a DC to an OLFP. Events of size 2.5 Kbytes were gathered by CODA into blocks of 64 (160 Kbytes), which were then sent over the TCP sockets connection with the same settings as before. The results of this test are also shown in Table 1.

Parameter	Sockets	CODA
Data Rate 715->735	6.1 Mb/s	6.2 Mb/s
Data Rate (events/s)		2300
715 Utilization	70%	75%
735 Utilization	30%	48%
Data Rate 735->715	8.3 Mb/s	4.2 Mb/s
Data Rate (events/s)		1600
715 Utilization	95%	26%
735 Utilization	30%	31%

Table 1. Test results using point-to-point ATM links between two HP computers. The sockets test only tests communication speed using sockets, while the CODA test includes some DAQ functions as well.

The implications of these tests are as follows. This combination of the TCP/IP protocol over ATM links appears to require approximately 8 MIPS per Mbyte of data transferred (this may be reduced in future network adapters). If the DCs use a VME processor of 125 MIPS we should be able to transfer about 2000 event fragments/s (5 Mbytes/s for the heaviest loaded DCs) using 40 MIPS or 32% of that processor. This would allow 85 MIPS to be used for collecting and



reordering the data from the ROCs. If the OLFPs are 250 MIP machines, reception of 333 events/s (3.6 Mbyte/s) would use about 29 MIPS or 11%. It would require another 13% to send it out, leaving 76% or 189 MIPS for analysis, monitoring, and triggering. This means that the processor could spend approximately 567,000 instructions on each event in merging the fragments and analyzing it.

### *B. Simulation*

Although aggregate data rates indicate that the system components are capable of both processing and transporting the data, the effects of timing have also been investigated through two simulations. The first approach used COMNET3 [11], a graphical based simulation product developed for the purpose of modeling local and wide area networks encompassing numerous media access protocols. COMNET3 contains specific support for ATM switches and TCP/IP, and allows customization of processor characteristics and process loading at each OLFP and ROC. The token mechanisms proposed were not modeled using standard token rings, instead the implementation used a series of traffic sources which were triggered by the receipt of specific message types (tokens). Verification of the correctness of the barrel shifting operation was possible by a visible inspection of the animation capabilities of COMNET3 and a detailed examination of the tracing features which provides text output of the movement of individual packets. This high level simulation has produced results indicating that the overall design is viable with minimal contention in the communication links at the data rates expected.

A second, more detailed, approach is underway using MODSIM2 [11], an object-oriented simulation language. This model allows for customization of different token strategies and a more detailed examination of the effects of various protocols such as TCP/IP versus the ATM API. The first simulation suggests that timing will not be an issue; this model is being used not only to answer the academic questions of how the strategies compare, but also to study the effects of data rates higher than those planned, predict buffer sizes and determine time constraints for various types of event processing.

## V. CONCLUSIONS

The event builder described here should be able to handle the event rates from the CLAS detector. The ATM links are fast enough to handle the data rates expected, and the TCP protocol handling leaves enough processing power so the OLFPs can perform some analysis. The use of the 'dual token barrel shifting' algorithm almost completely eliminates contention and cell loss in the ATM environment. Using the switch to pass the tokens simplifies the design by eliminating a separate control link or network.

This design is also very scalable. A larger switch with more ports would let additional processors be added, allowing more analysis to take place in each one. It would also reduce

the impact when one processor does not take the event request token. The processors themselves could be expanded into processor clusters allowing more analysis to occur. A second switch could be used to send the events to the tape processors. Finally, higher speed ATM links could be used when switches and interfaces at those speeds become available.

## VI. ACKNOWLEDGMENTS

The authors wish to acknowledge the CPSC 611 graduate class at Christopher Newport University for their work on the simulation. This work was supported in part by Department of Energy Contract DE-AC05-84ER40150.

## VII. REFERENCES

- [1] Conceptual Design Report, CEBAF Basic Experimental Equipment, CEBAF, April 13, 1990.
- [2] D. Doughty et al "A VXIbus Based Trigger for the CLAS Detector at CEBAF," IEEE Transactions on Nuclear Science, NS 39, 1992 pp 241-247.
- [3] E. Jastrzembki, D. R. Quarrie, W. A. Watson III. "The CEBAF Trigger Supervisor," Conference Record of the 1991 IEEE Nuclear Science Symposium, vol. 1, pp 569-573.
- [4] J. Pangburn, private communication.
- [5] D. Cullen-Vidal et al "D0 Level-2/Data Acquisition; the New Generation," Proceedings of the International Conference on Computing in High Energy Physics 1991, pp 659-652.
- [6] J. Patrick et al "The CDF UltraneT Based Data Acquisition System," Proceedings of the International Conference on Computing in High Energy Physics 1994, to be published.
- [7] Systran Copra, 4126 Linden Ave. Dayton, Ohio 45432 USA.
- [8] D. Doughty et al "Data Compression and Readout of Multiplexed Drift Chamber Data in the CLAS Detector at CEBAF," Proceedings of the International Conference on Computing in High Energy Physics 1994, to be published.
- [9] FORE Systems, Inc., 174 Thorn Hill Road, Warrendale, PA 15086 USA.
- [10] W. A. Watson III et al "CODA: A Scalable, Distributed Data Acquisition System" Conference Record of the 1993 IEEE Nuclear Science Symposium, vol. 1, pp 296-303.
- [11] CACI Products Company, 3333 North Torrey Pines Ct. La Jolla, California 92037 USA

## The Event Builder of the ZEUS Detector

Ulrich Heintz

DESY/ZEUS

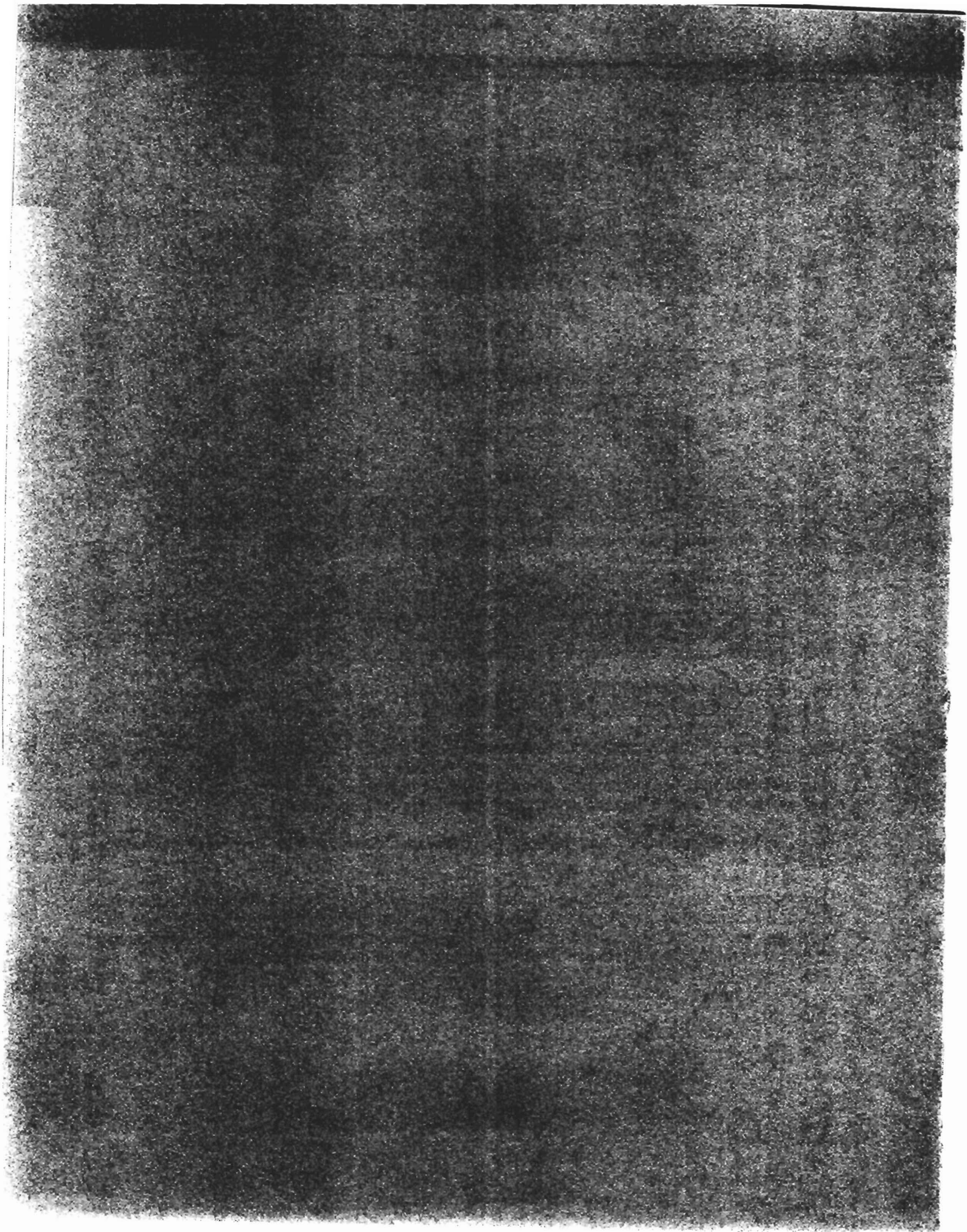
The task of the ZEUS Event Builder (EVB) is to collect the event data of the 14 subdetectors of the ZEUS detector and to reconstruct the event data into a format which can be used by the ZEUS event reconstruction program.

The main requirements for the EVB are given by the event data of the ZEUS detector. The event data are collected by the ZEUS subdetectors and are transferred to the ZEUS Event Builder. The event data are reconstructed into a format which can be used by the ZEUS event reconstruction program.

The implementation of the ZEUS EVB consists of a number of modules which are developed by the ZEUS detector groups. The modules are developed in C++ and are executed on the ZEUS detector groups. The modules are developed in C++ and are executed on the ZEUS detector groups.

The EVB is running online and offline for more than 3 years. The software is running on the ZEUS detector groups.

The system has been developed using structured system development tools. The system is developed in C++ and is executed on the ZEUS detector groups. The system is developed in C++ and is executed on the ZEUS detector groups.



# The Eventbuilder of the ZEUS Experiment

Ulf Behrens, Lars Hagge, Wolfgang O. Vogel\*  
Deutsches Elektronen-Synchrotron, Hamburg

February 9, 1993

## Abstract

The Eventbuilder of the ZEUS experiment is a real-time parallel data formatting and transport system. It combines data flows originating from various detector components and transfers them to the third level trigger processor farm. The Eventbuilder is based on an asynchronous packet-switching transputer network and uses transputer links for bulk data transfer. A high-speed  $64 \times 64$  custom made crossbar switch allows dynamic linking of any detector component to any branch of third level processor nodes, offering a bandwidth of more than  $24 \text{ MB/s}$  over a distance of  $100 \text{ m}$ . The use of structured system development techniques (SA/SD) resulted in a flexible and well-partitioned system structure and guaranteed that all requirements were met.

## 1 Introduction

HERA, the new electron-proton colliding facility at DESY, started operation for physics measurements in spring 1992. HERA provides electron-proton collisions at a CM energy of  $310 \text{ GeV}$ . The ZEUS collaboration constructed a detector for one of HERA's interaction regions.

Challenges for the ZEUS experiment are the short interval between beam crossings of only  $96 \text{ ns}$ , more than 250,000 readout channels and an initial raw data rate exceeding  $10 \text{ GB/s}$ . The data rate has to be reduced by a factor of at least  $10^4$  before data recording. This imposes strong requirements on the trigger and data acquisition system.

The trigger and data acquisition system of the ZEUS experiment is a highly-parallel distributed real-time system. It consists of several independent readout systems and three trigger levels for data filtering. Its central part, the ZEUS Eventbuilder, combines and formats the data flows originating from the various readout systems.

The Eventbuilder is subject to the highest data rate within the ZEUS data acquisition system. Due to its connections to almost all parts of the data acquisition system, the Eventbuilder is also an important tool for system analysis and diagnosis. This article describes the development of the ZEUS Eventbuilder, gives a brief overview of its hardware and software architecture and reports first results on the performance of the Eventbuilder and the data acquisition system.

---

\*now at Blohm & Voss, Hamburg

Table 1: Specification of component subsystems in the trigger and data acquisition system [Col89]

Detector Component		No. Readout Channels	maximum Event Length	Readout Processor
Central Tracking Detector	CTD	4,608	30 kB	Transputer
Forward/Rear Track. Det.	FRTD	5,778	15 kB	Transputer
Barrel Calorimeter	BCAL	5,184	20 kB	Transputer
Forward Calorimeter	FCAL	4,344	20 kB	Transputer
Rear Calorimeter	RCAL	2,336	10 kB	Transputer
Transition Radiation Det.	TRD	2,472	10 kB	Transputer
Hadron Electron Separator	HES	37,304	10 kB	Transputer
Backing Calorimeter	BAC	40,000	2 kB	Transputer
Vertex Detector	VXD	832	2 kB	68k-family
Beamline	BEAM		1 kB	Transputer
Barrel Muon Chambers	BMUO	62,256	0.6 kB	Transputer
Forward Muon Spectrometer	FMUO	18,948	0.5 kB	68k-family
Leading Proton Spectrometer	LPS	52,000	0.2 kB	68k-family
Luminosity Monitor	LUMI		0.2 kB	68k-family
Vetowall	VETO		0.01 kB	Transputer
Global Second Level Trigger	GSLT		20 kB	Transputer
Fast Clear	FCLR		2 kB	68k-family
	$\Sigma$ :	258,142	142 kB	

## 2 Overview of the ZEUS Trigger and Data Acquisition System

The ZEUS detector comprises several independently operating detector components, each of them equipped with their own so-called component subsystem (CSS). Component subsystems contain the "frontend" electronics required for the component control and readout. They interface to two levels of global trigger processors and the Eventbuilder. The layout of the ZEUS trigger and data acquisition system and the data throughput at its components are shown in figure 1. The component subsystems of the ZEUS experiment are listed in table 1.

Once a detector component has been read out, the data are stored in a  $5.5 \mu\text{s}$  first level trigger pipeline and analyzed by a local first level trigger processor. The results of the different component subsystems referring to the same beam crossing are input to the global first level trigger (GFLT), which computes an overall first level trigger decision. The maximum rate of GFLT accept decisions is designed to be  $1 \text{ kHz}$ . Up to the GFLT both the trigger and readout are deadtime free.

On GFLT accept, data accepted for further analysis are copied to a second level trigger pipeline. A GFLT accept rate of  $1 \text{ kHz}$  and a "copy" time of  $30 \mu\text{s}$  result in 3% deadtime. This is the only source of deadtime provided no buffer full states occur.

A second level trigger processor local to the component subsystem computes a trigger sub-decision, which is forwarded to the global second level trigger (GSLT) and used to compute an overall second level trigger decision. The GSLT is designed to accept ca. 10% of all GSLT

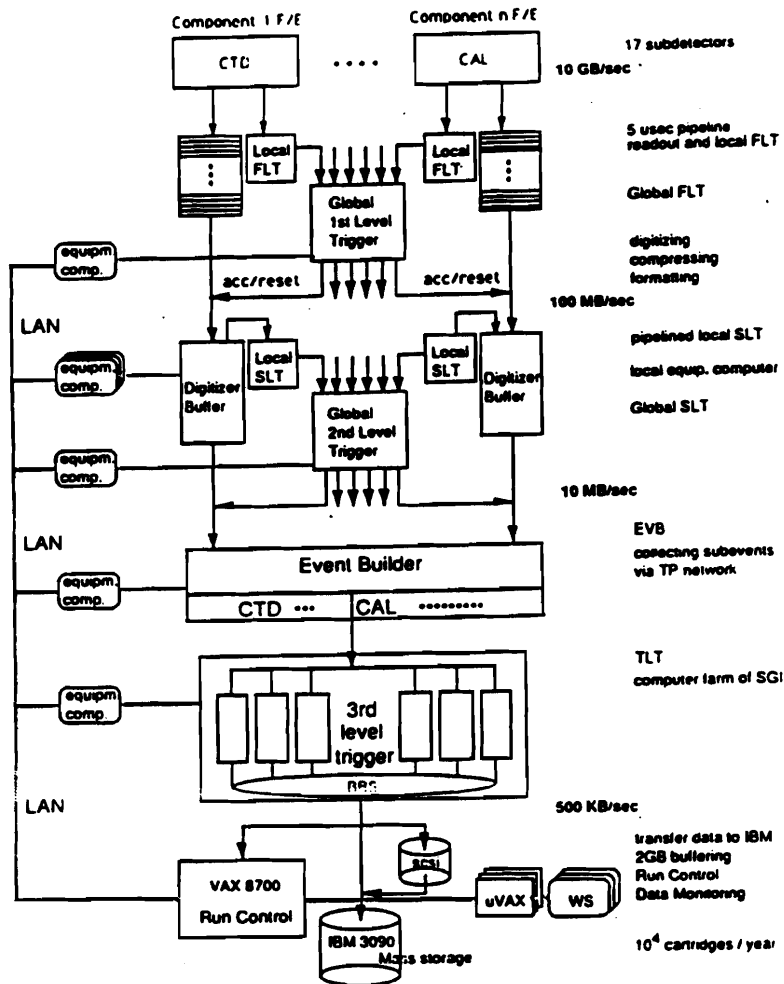


Figure 1: Layout of the Trigger and Data Acquisition System of the ZEUS experiment; on the right side, the data throughput at the different components of the system is shown.

accepted triggers.

In case a component subsystem receives a positive GSLT decision, the corresponding data are assigned a "GSLT decision number" and transferred to the Eventbuilder. The Eventbuilder combines and formats all the component data carrying the same GSLT decision number into one data set. This data set is called an "event", and its GSLT decision number is also referred to as the "event number".

Once an event is complete, it is input to the third level trigger (TLT). The TLT is a processor farm consisting of six branches of a total of 36 processor nodes. It performs the global event reconstruction and a final filtering and is designed to accept up to 5 events/s.

### 3 Developing the ZEUS Eventbuilder

The performance of the Eventbuilder has strong influence on the output of the entire experiment, and a careful design of both the systems hardware and software is mandatory for optimum operation. The use of structured development techniques (SA/SD) yielded a well-partitioned and flexible system structure and ensured all requirements being met.

The following sections list the requirements on the Eventbuilder and illustrate the analysis and design process. Additionally, the implementation of the Eventbuilder and its operation are also described. Finally, experience gained from system development and integration is summarized<sup>1</sup>.

### 3.1 Requirements

The requirements on the ZEUS Eventbuilder are defined by its position in the trigger and data acquisition system, the rate of positive GSLT decisions and the amount of data acquired from the component subsystems. The main issues are to:

- combine and format data from different components carrying the same event number into a single data record (event). Sufficient buffer space has to be provided to account for the asynchronously arriving component data. Complete events have to be transferred to the TLT, which involves a data transport over a distance of around 100 m.
- sustain a GSLT accept rate of at least 100 events/s. Taking into account the event sizes defined in table 1, this requires a total bandwidth of more than 15 MB/s and up to 3 MB/s at the interfaces to component subsystems.
- provide fault tolerance against failure of transmission lines to the TLT. The data transport to the TLT necessitates the use of serial transmission lines and a redundant hardware architecture.
- distribute the events over the TLT branches. By surveying the data throughput at the interfaces to the TLT, the load of its different branches of processor nodes can be estimated and used for load-balancing.

Further requirements include format checks of component data and generation of an index to the data objects inside an event record<sup>2</sup>, careful on-line monitoring for debugging and system analysis purposes, conceptual simplicity regarding maintenance and future upgrades, and low cost.

### 3.2 Essential Model

The Eventbuilder has to support interfaces to the various detector components, the six branches of third level trigger processor nodes (TLT), the Global Second Level Trigger (GSLT) and the Run Control console (RC). The Context Diagram (CD, fig. 2) shows, how the Eventbuilder is embedded in the trigger and data acquisition system.

Entity Relationship Diagrams (ERDs) show the data elements occurring in a system and highlight the relationships between them. In case of the Eventbuilder, an ERD can easily be derived from a description of the system's behaviour, where nouns refer to objects and verbs indicate relationships (fig. 3). Every detector component has to *respond* to a *GSLT-decision* by providing its *component data*. *Component data consists of several component-data banks*. Scanning the *component data* reveals the *component data composition*. *Matching* this to the *readout configuration* ensures only valid banks being built into the event. When

---

<sup>1</sup>This article introduces part of the notation of SA/SD. However, for the modelling technique we refer to [HP87, You89, PJ80].

<sup>2</sup>The ZEUS collaboration stores their data in the ADAMO format [FP90].

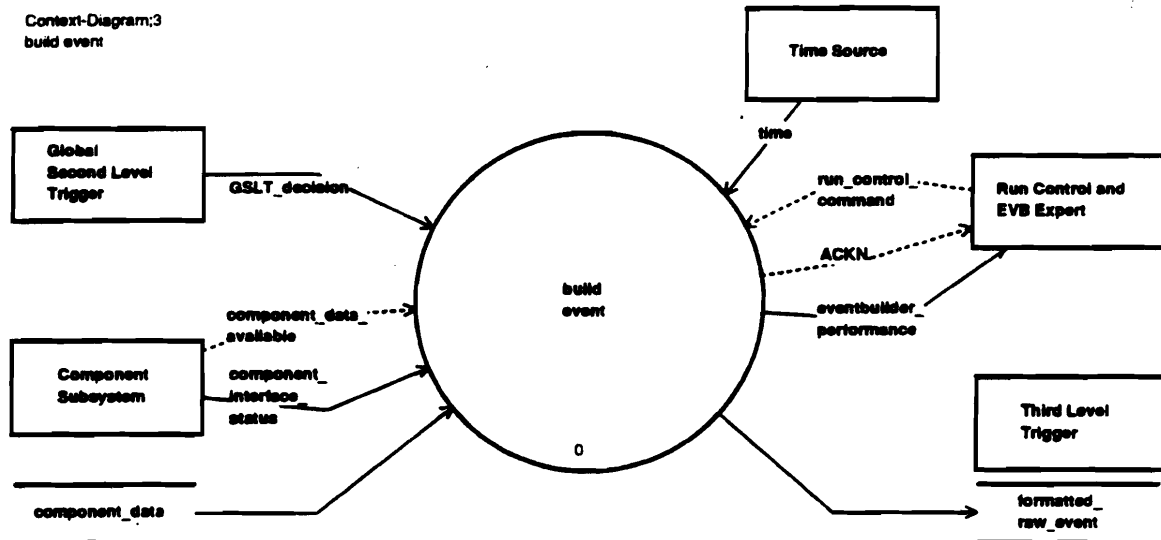


Figure 2: The Context Diagram defines the interfaces between the Eventbuilder and other components of the trigger and data acquisition system. Boxes represent external systems, bars common memory areas and arrows the flow of data (solid) or control information (dashed). The Eventbuilder is shown as a bubble, representing a process.

all *participating\_components* of a run have responded to the *GSLT\_decision*, the set of valid *component\_data\_banks* and the *event\_composition* can be combined into the *formatted\_raw\_event*.

Tasks operating on the data elements and establishing the relationships are defined in a Control and Data Flow Diagram (CDFD, fig.4). The diagram is an extension of the "build event" process in the Context Diagram. It has the same input and output flows, but gives a more detailed definition of how to build events. The processes of the CDFD are synchronized by a control unit (finite state machine, fig.5) which analyzes the process states and reacts on external signals.

### 3.3 Hardware Architecture

Transputers<sup>3</sup> proved to be well suited processors for the ZEUS Eventbuilder. Standard VME transputer modules offering two T800 transputers with 4 MB of private memory each and a triple-ported memory (TPM) of 128 kB or 512 kB on a double-height VME-module [NIK90] have been developed within the ZEUS collaboration. It was decided to use those modules wherever possible. Fig. 6 shows the layout of the Eventbuilder hardware.

Interfaces to component subsystems and to branches of TLT processor nodes connect the Eventbuilder with the trigger and data acquisition system of the ZEUS experiment. To keep the interfaces independent of the implementation of the external systems, common memory areas have been chosen for data input to or output from the Eventbuilder. The interfaces are implemented using the ZEUS standard transputer modules with the common memory areas

<sup>3</sup>Transputers are single VLSI devices with processor, memory and communication links to other transputers [inn89]. Transputers are building blocks for real-time parallel systems as described in [Hoa78]. Their links are designed for synchronisation purposes inside distributed systems, but may also be used for data transport.



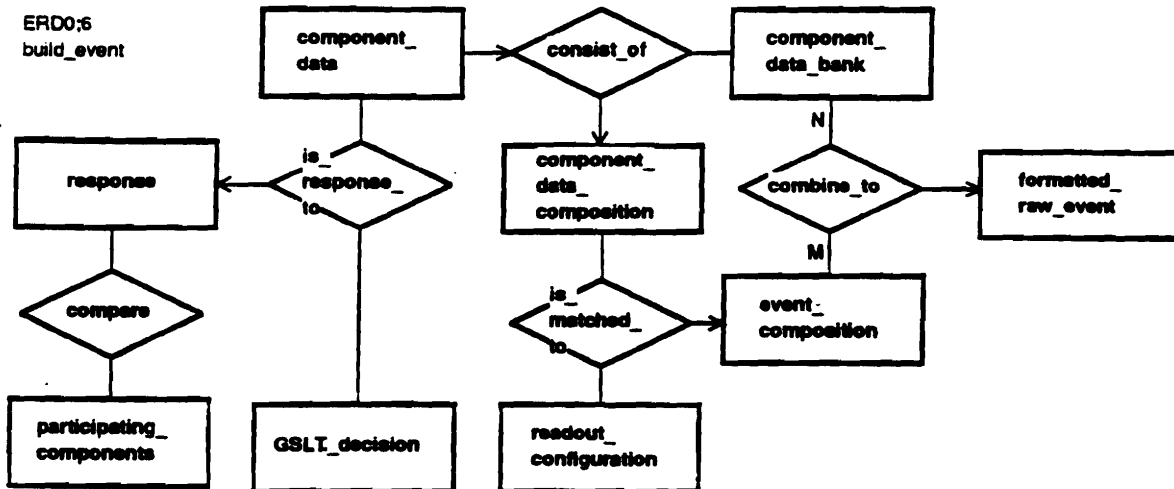


Figure 3: The data objects occurring within the boundaries of the Eventbuilder are defined in an Entity-Relationship-Diagram. Boxes indicate data objects, diamonds represent relationships between objects. The numbers classify the type of a relationship (one-to-one, one-to-many, ...).

being located in the TPMs. At the input side, one of the board's transputers is made available to the component subsystem. This way, components can access the memory via VME or transputer. The third level trigger obtains its data by VME accesses to the TPMs.

A network of data paths has to be foreseen in the Eventbuilder to transport data from every component subsystem to any branch of TLT nodes. A freely configurable network (crossbar switch) has proven to be the best solution [Hag90]. Crossbar switches can connect any of their inputs to any of their outputs. In case of an  $N \times N$  crossbar switch,  $N$  such connections can be established simultaneously. The Eventbuilder's custom made crossbar switch for transputer links is based on Inmos C004 chips [Loh].

For maximum performance, fibre-optical link connections [IfH] have been developed for the long distance data transfer to the third level filter farm. The data transfer is limited by the handshake protocol on transputer links. Currently, a peak data throughput of 600 kB/s/link is achieved, limiting the total sustained bandwidth to 24 MB/s.

A control unit (Supervisor) provides the interface to Run Control and configures the crossbar switch according to the data arriving at the component interfaces.

### 3.4 System Implementation and Operation

To implement the Eventbuilder, the processes of the Essential Model were allocated to the different processor groups. Then the code for each transputer and the protocols between them were designed. The code is written in parallel C. An SGI4D/25S unix workstation with a purpose built transputer interface served as host and development platform.

The Eventbuilder operation principle can be summarized as follows:

- Component subsystems provide their data in a common memory area and signal its availability to the Eventbuilder. The Eventbuilder then checks the component data for the correct format.
- As soon as the GSLT decision is available for an event, the Eventbuilder tries to transfer

100.1  
build event

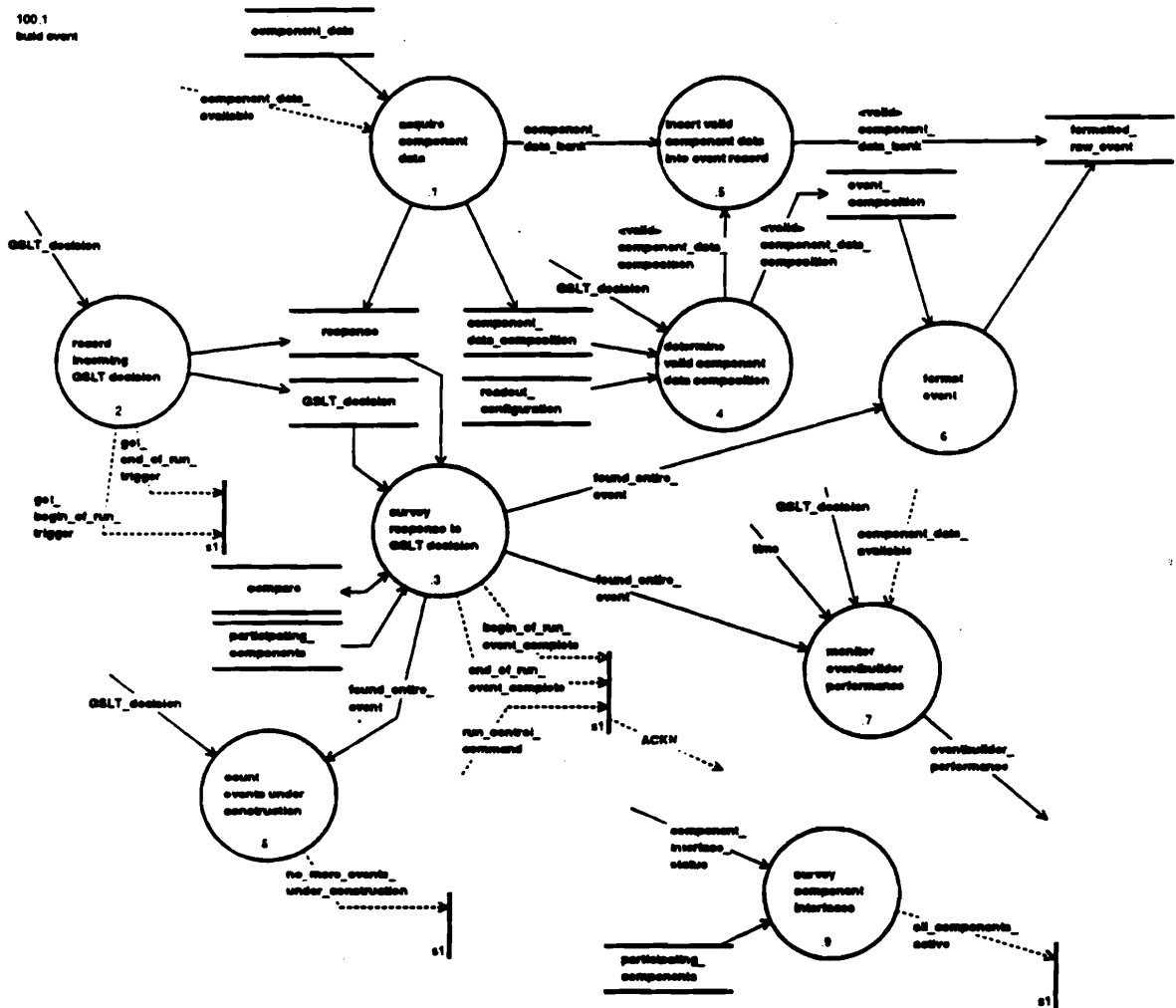


Figure 4: The Control and Data Flow Diagram defines how the Eventbuilder processes objects and establishes relationships amongst them. The notation is similar to the Context Diagram. The vertical bar denotes the interface to the finite state machine which is synchronizing the processes.

100-01.2  
control event building

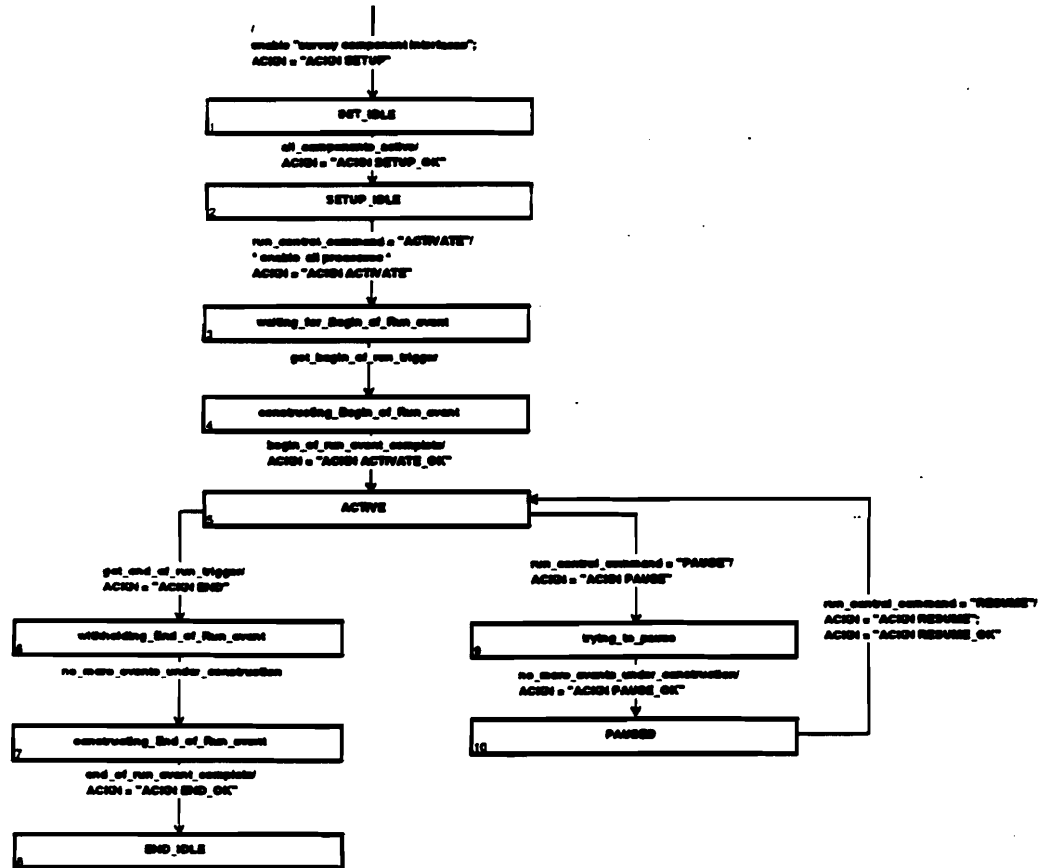


Figure 5: The State Transition Diagram (STD) defines a finite state machine. Boxes denote system states, arrows show transitions between them. Labels on the arrow define the condition under which a transition occurs and list the actions to be taken.

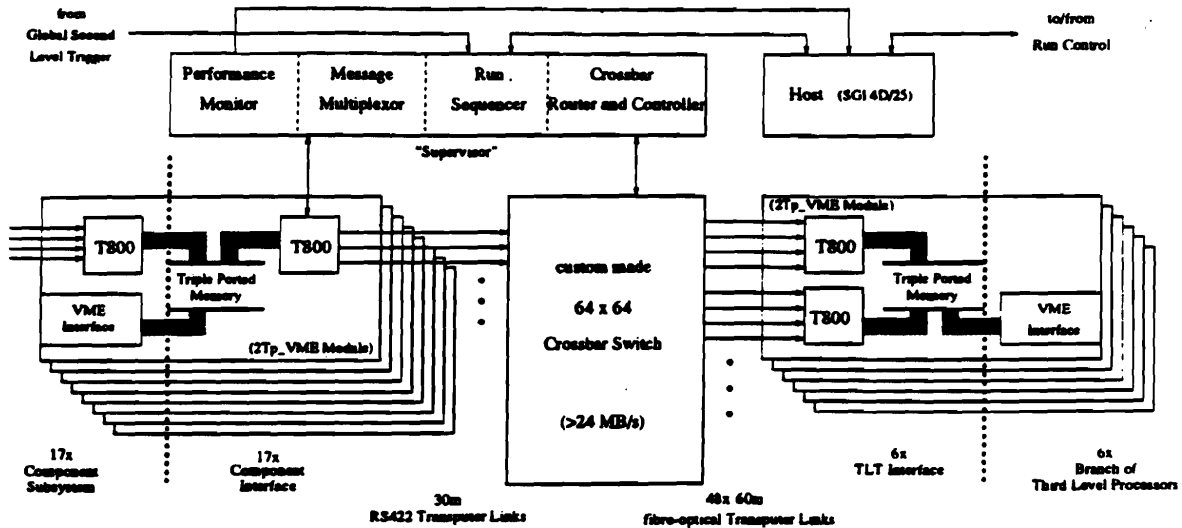


Figure 6: Layout of the Eventbuilder hardware. Interfaces to Component Subsystems and branches of TLT processor nodes use ZEUS standard transputer modules, while Supervisor and Crossbar Switch are custom made. For maximum performance fibre-optical transputer link connections have been developed for the data transfer between crossbar switch and interfaces to the third level filter farm.

all the component data to one of its TLT interfaces. For this purpose, the component interfaces issue a "link request" to the crossbar router whenever they have data ready for transfer. Once they receive a "link ready"-message, the data transfer to the TLT interface is immediately started on the specified link. The availability of this link is again signalled by a "link release"-message.

- The decision, which event should be transferred to which TLT branch, is computed by the crossbar control task. It traces the buffer and I/O loads on each TLT branch to avoid new events being directed to busy branches. The connections between component and TLT interfaces are installed by a router which is tuned to minimize deadtime on the transmission lines.
- When all component data of an event have been transferred to a TLT interface board, the formatting of the event is triggered by the control unit. Formatted events are copied to the common memory area with the TLT.

### 3.5 Experience

The Eventbuilder of the ZEUS experiment has been developed, implemented and tested between 1988 and 1991, consuming about 11 man years. Most of the effort has been spent on software development (7 man years). Because of the extent of the Eventbuilder system (more than 50 transputers distributed over 24 VME crates) and its numerous interfaces, about half of this time went into system integration and verification.

The use of SA/SD techniques proved to be helpful in many situations. The software model is well partitioned and of a flexible structure, so that modifications of requirements usually affect only a single process. The encapsulation of processes enabled prototyping and partial

implementation and supported system integration at an early stage. As all process interfaces were well defined, simulators of the different processor groups and external systems have been developed. Thus, very reliable performance estimates have been available at any stage of system development.

Transputers have shown to be easy-to-handle multi-purpose processors for real-time parallel systems. However, testing and debugging software distributed over several transputers turned out to be a difficult and very time consuming task as no tools were available which allow to analyze a transputer network without changing its real-time behaviour. For the tracing of synchronization problems, again the diagrams of the Essential Model were indispensable.

## 4 Eventbuilder Performance

The Eventbuilder of the ZEUS experiment has seen successful operation for more than one year. During this time, the Eventbuilder performance has been carefully monitored and evaluated. This fact and its central position in the data acquisition chain have enabled the Eventbuilder to become an important diagnostic and analytic tool for the entire trigger and data acquisition system.

### 4.1 Monitoring Concept

Eventbuilder operation involves several hundred processes which are distributed over more than fifty transputers and have to share limited system resources like buffer space or transfer lines. A set of characteristic quantities is monitored during Eventbuilder operation to trace the system performance.

Monitoring data are collected in different parts of the Eventbuilder, but have to be analyzed on a dedicated processor. By passing the data along with the synchronization messages, no extra traffic is introduced on the Eventbuilder network. To keep the extra load which monitoring imposes on the Eventbuilder processors as low as possible, monitoring data are collected while the events are transferred instead of being taken at fixed intervals. Time stamps allow tracing of the Eventbuilder performance. To allow for correlations of monitoring data acquired in different parts of the system (i. e. on different transputers), a "real time" is defined throughout the whole system [Sch92].

### 4.2 Performance

Requirements on the bandwidth of the Eventbuilder arise from the GSLT frequency,  $f_{\text{GSLT}}$ , and the amount of data acquired from each component subsystem,  $L_{\text{Comp}}$ . Their nominal values are listed in section 3.1. The response time of a component subsystem to a GSLT decision,  $d_{\text{Comp}}$ , defines the minimum buffer capacities required at the component interfaces.

During the first year of operation, the mean GSLT decision rate,  $f_{\text{GSLT}}$ , was kept below twenty events/s. Therefore, the limit of the Eventbuilder has not been reached. Measurements have shown the total bandwidth to be at least 24 MB/s. The Eventbuilder can construct up to 72 events in parallel. Its buffers can accommodate at least 75 more events, depending on the event size. Fig. 7 shows data sizes and response times for components as observed during the pilot run and compares them with the specification.

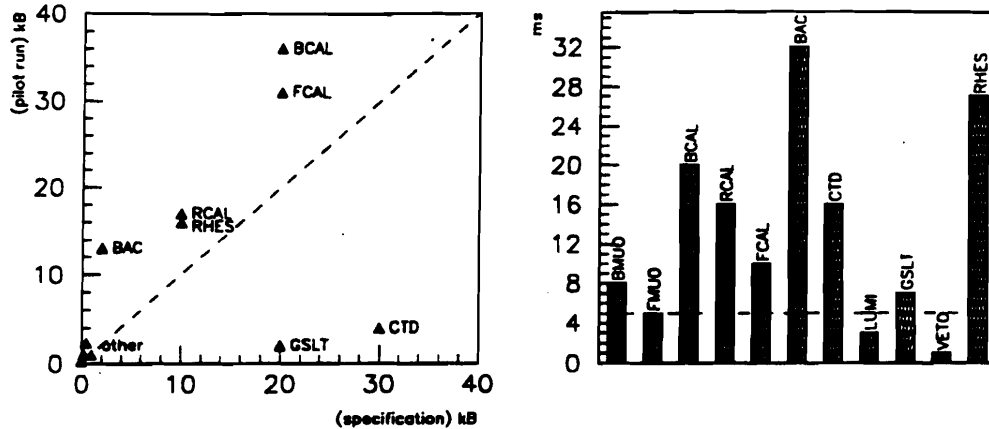


Figure 7: Behaviour of component subsystems as observed during pilot run. Left: Average amount of data acquired per event, compared to specification values. Right: Average response time on trigger decision (specification: 5 ms).

### 4.3 Online Monitoring and System Analysis

For on-line monitoring purposes it is sufficient to simply survey the load of the buffers inside the Eventbuilder. Any unusual system behaviour can be detected, sometimes even predicted from heavy buffer load. As an example, fig. 8 shows how the Eventbuilder's buffers fill when the accept rate of the second level trigger (GSLT) exceeds the speed of the third level trigger (TLT). As the TLT is located downstream from the Eventbuilder, buffers are expected to start filling at the backend. Indeed, the common memory areas with the TLT fill up first (P.TLT $n$ ), followed by the internal buffers of the interfaces to the TLT (I.TLT $n$ ). Finally, the buffers inside the private memory of the interfaces to the component subsystems (I.Comp) fill. The figure shows, that when all buffers in the Eventbuilder were filled, the data acquisition system stabilized at a GSLT accept rate of 44 events/s.

Monitoring the GSLT accept rate and the data flow into the Eventbuilder allows to determine the maximum event rates which can be handled by the different component subsystems. Even at low GSLT accept rates, consecutive positive GSLT decisions may be separated only by a few milliseconds. Fig. 9 shows for a run with an average GSLT rate of 18 Hz the interval between two consecutive GSLT decisions, DTTRIG, going down to 2 ms (upper left and right). Component subsystems should have a constant response time on GSLT decisions, therefore the interval between two consecutive component data sets, DTComp, is expected to equal the corresponding DTTRIG. However, plotting DTComp against DTTRIG shows DTComp to saturate (lower right). Obviously, the component subsystem cannot keep pace if the trigger decisions are coming in too fast, and the corresponding data start piling up in the system's buffer. Only when DTTRIG increases beyond the minimum of DTComp the component subsystem can start emptying its buffers, and DTComp < DTTRIG. Determining the minimum of DTComp allows to derive the maximum event rate which can be handled by a component subsystem.

The last issue shows that monitoring Eventbuilder operation may also be used to survey the performance of those components interfacing the Eventbuilder. This way, the Eventbuilder has become an important diagnostic and analytic tool for the entire data acquisition system. Currently, the Eventbuilder environment is used for the construction of a prototype expert system [BFHO, BFH92] which can survey and analyze the monitoring data. Its goal is to provide on-line diagnostics and guidance for the shift crew running the experiment.

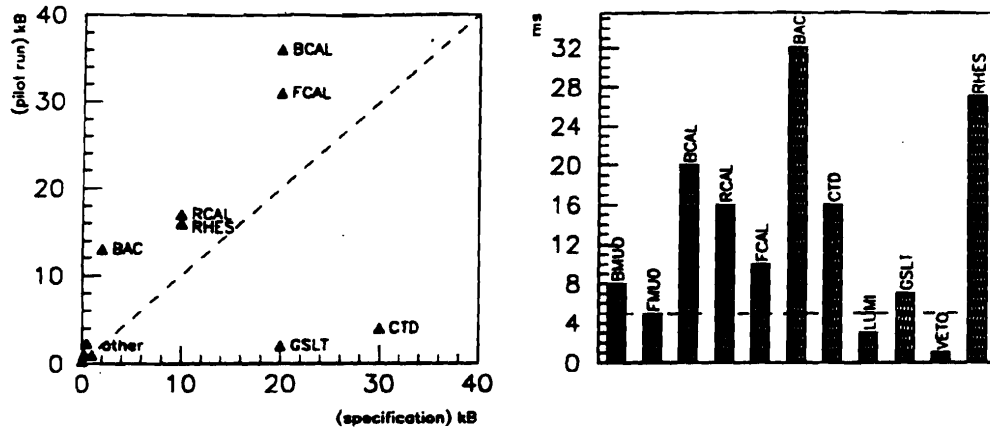


Figure 7: Behaviour of component subsystems as observed during pilot run. Left: Average amount of data acquired per event, compared to specification values. Right: Average response time on trigger decision (specification: 5 ms).

### 4.3 Online Monitoring and System Analysis

For on-line monitoring purposes it is sufficient to simply survey the load of the buffers inside the Eventbuilder. Any unusual system behaviour can be detected, sometimes even predicted from heavy buffer load. As an example, fig.8 shows how the Eventbuilder's buffers fill when the accept rate of the second level trigger (GSLT) exceeds the speed of the third level trigger (TLT). As the TLT is located downstream from the Eventbuilder, buffers are expected to start filling at the backend. Indeed, the common memory areas with the TLT fill up first (P.TLT $n$ ), followed by the internal buffers of the interfaces to the TLT (I.TLT $n$ ). Finally, the buffers inside the private memory of the interfaces to the component subsystems (I.Comp) fill. The figure shows, that when all buffers in the Eventbuilder were filled, the data acquisition system stabilized at a GSLT accept rate of 44 events/s.

Monitoring the GSLT accept rate and the data flow into the Eventbuilder allows to determine the maximum event rates which can be handled by the different component subsystems. Even at low GSLT accept rates, consecutive positive GSLT decisions may be separated only by a few milliseconds. Fig.9 shows for a run with an average GSLT rate of 18 Hz the interval between two consecutive GSLT decisions, DTTRIG, going down to 2 ms (upper left and right). Component subsystems should have a constant response time on GSLT decisions, therefore the interval between two consecutive component data sets, DTComp, is expected to equal the corresponding DTTRIG. However, plotting DTComp against DTTRIG shows DTComp to saturate (lower right). Obviously, the component subsystem cannot keep pace if the trigger decisions are coming in too fast, and the corresponding data start piling up in the system's buffer. Only when DTTRIG increases beyond the minimum of DTComp the component subsystem can start emptying its buffers, and DTComp < DTTRIG. Determining the minimum of DTComp allows to derive the maximum event rate which can be handled by a component subsystem.

The last issue shows that monitoring Eventbuilder operation may also be used to survey the performance of those components interfacing the Eventbuilder. This way, the Eventbuilder has become an important diagnostic and analytic tool for the entire data acquisition system. Currently, the Eventbuilder environment is used for the construction of a prototype expert system [BFHO, BFH92] which can survey and analyze the monitoring data. Its goal is to provide on-line diagnostics and guidance for the shift crew running the experiment.

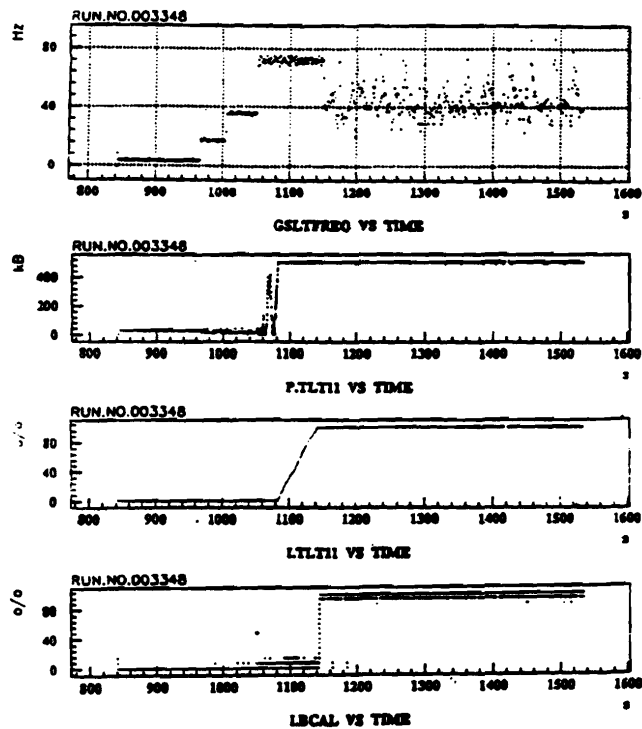


Figure 8: Buffer loads reveal unusual system behaviour (explanation in the text).

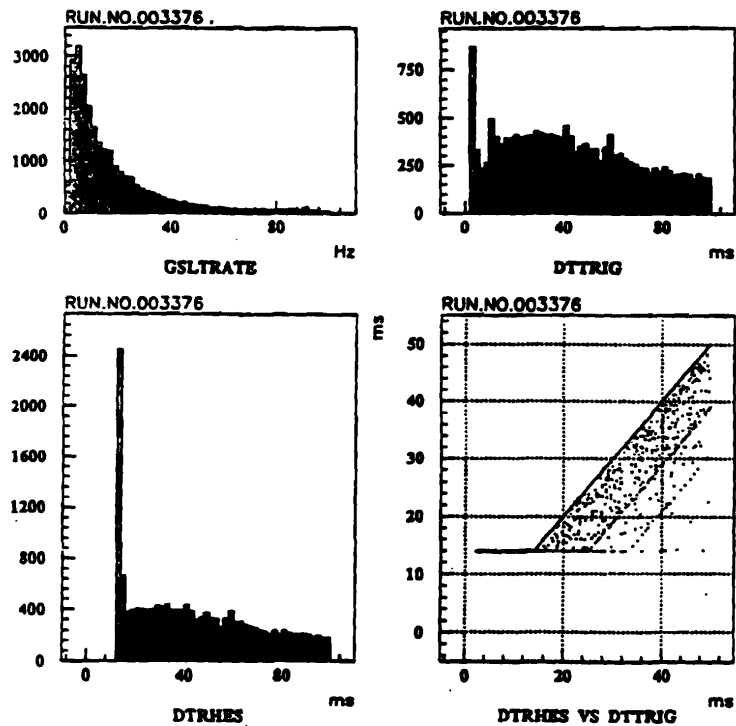


Figure 9: The maximum event rate which can be handled by a component subsystems can be determined from monitoring the GSLT accept rate and the data flow into the Eventbuilder (see text).



## 5 Conclusion

The Eventbuilder of the ZEUS experiment is a transputer-based real-time parallel data formatting and transport system with a total bandwidth of at least 24 MB/s. It has seen successful operation for more than one year now.

The Eventbuilder has been developed making extensive use of structured system development techniques. Application of Structured Analysis and Structured Design (SA/SD) yielded a well-partitioned and flexible system structure and ensured that all requirements were met.

Its central position has enabled the Eventbuilder to become an important diagnostic and analytic tool for the entire trigger and data acquisition system of the ZEUS experiment. The full potential of the Eventbuilder diagnostics will be achieved when the expert system [BFHO] becomes fully available.

## References

- [BFH92] Ulf Behrens, Mariusz Flasiński, and Lars Hagge. ZEXP - The ZEUS Expert System. DESY-Report 92-141, DESY, Notkestr. 85, D-2000 Hamburg 52, 1992.
- [BFHO] Ulf Behrens, Mariusz Flasiński, Lars Hagge, and Kars Ohrenberg. Prospects of an Expert System for ZEUS. ZEUS-Note, in preparation.
- [Col89] ZEUS Collaboration. The ZEUS Detector, Status Report 1989. Technical report, DESY, March 1989.
- [FP90] Steve M. Fisher and Paolo Palazzi. *The ADAMO Data System*, 3.1st edition, November 1990.
- [Hag90] Lars Hagge. Anwendungen von Transputern in der Hochenergiephysik am Beispiel der Entwicklung des ZEUS Eventbuilders. Diplomarbeit, University of Hamburg, October 1990.
- [Hoa78] C. A. R. Hoare. Communicating Sequential Processes. *Communications of the ACM*, 21(8):666-677, August 1978.
- [HP87] Derek J. Hatley and Imtiaz A. Pirbhai. *Strategies for Real-Time System Specification*. Dorset House Publishing Co., Inc., 353 West 12th Street, New York, NY 10014, 1987.
- [IfH] We would like to thank Holger Leich and associates of IfH Zeuthen for the development of the fibre-optical transputer links.
- [inm89] inmos Ltd., Bristol, UK. *The Transputer Databook*, 2nd edition, 1989.
- [Loh] We would like to thank Frank O. Lohmann for the construction of the crossbar switch.
- [NIK90] NIKHEF, Electronic Department, Amsterdam, The Netherlands. *Short Hardware Description of the 2TP-VME Module*, September 1990.
- [PJ80] Meilir Page-Jones. *The Practical Guide to Structured Systems Design*. Yourdon Press, New York, N.Y., 1980.

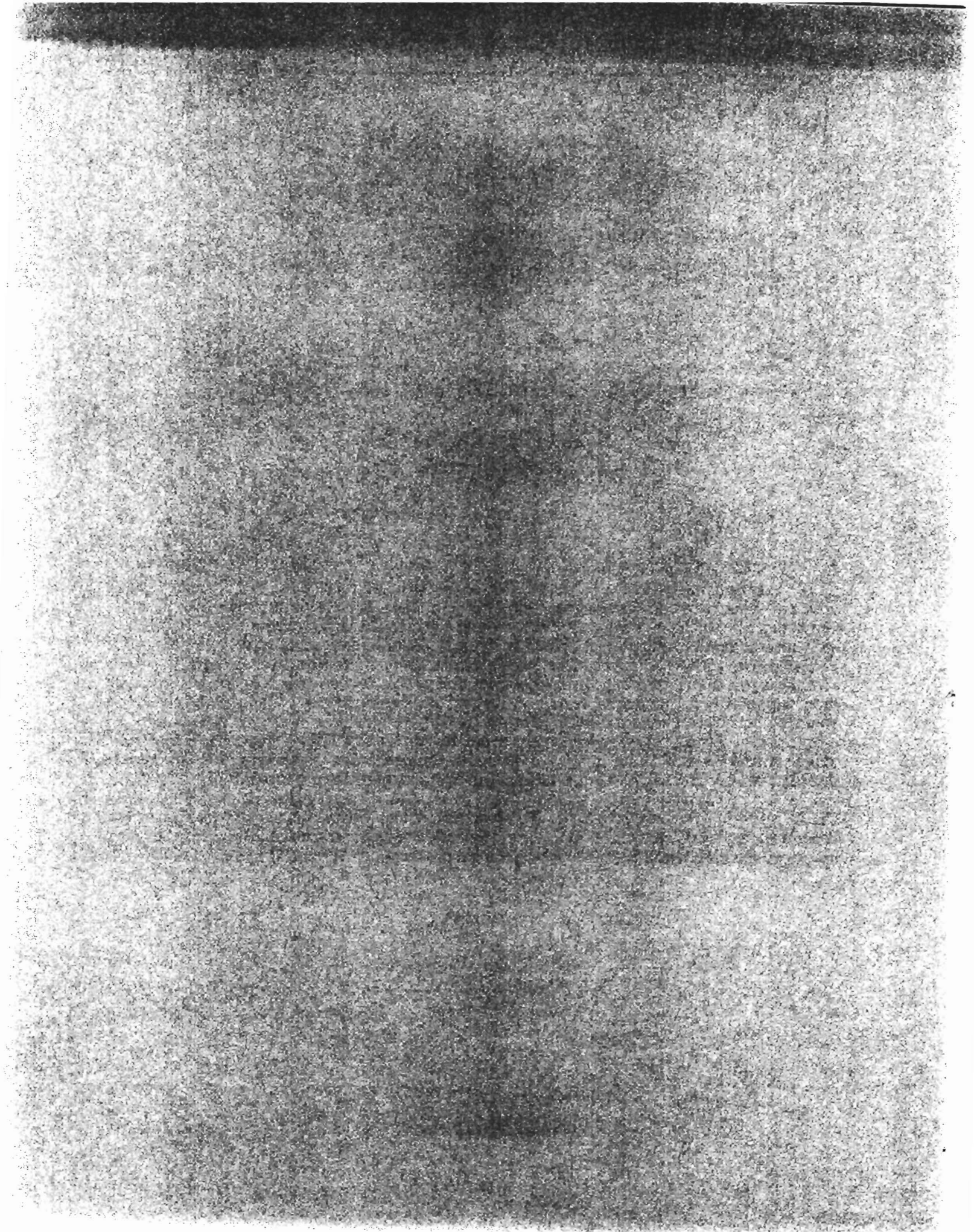
- [Sch92] Thorsten Schlichting. *Überwachung und Auswertung des Datenflusses am ZEUS Eventbuilder*. Diplomarbeit, University of Hamburg, April 1992.
- [You89] Edward Yourdon. *Modern Structured Analysis*. Yourdon Press Computing Series. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1989.

## **The CLEO III Data Acquisition System**

**Klaus Honscheid, et. al.  
Ohio State University**

Major upgrade programs have been approved for the CESR e+e- storage ring as well as for the CLEO experiment. In a few years the CESR luminosity will reach  $1-2 (10^{33}) \text{ cm}^{-2} \text{ s}^{-1}$  which will result in a trigger rate of up to 1000 Hz. New detector components like a silicon vertex detector and a particle identification system will significantly increase the event size and we expect a data volume as high as 50 MBytes/s. A new data acquisition system is being designed that allows to handle. The front-end electronic is housed in either VMEbus and Fastbus crate. A first set of buffers is integrated directly on the databoards keeping the readout time below 20 $\mu$ s. This corresponds to a readout induced deadline of 2%. Dedicated EMA engines in each crate collect the event fragments and send the data over an optical link to an eventbuilder. Several eventbuilder options are under discussion: a fast workstation (1000 MIPS) with several PCI slots to receive the optical link adapters, a reflective memory system by DEC, or a set of VME CPU modules with an additional fast interconnection.

CLEO III will have a distributed slow control system based on commercial products. Programs like LabView will be used on the detector component level and the central control and logging facilities will be implemented using a product like Vaccess by Vista.



## The CLEO III Data Acquisition System <sup>1</sup>

### Abstract

For the planned upgrade of the CLEO experiment and the CESR  $e^+e^-$  storage ring to operate at luminosities of  $2 \times 10^{33} \text{ cm}^{-2}\text{s}^{-1}$  new front-end electronics and a new data acquisition system are required. Extrapolating from the experience obtained with the current CLEO II detector, a read-out rate of up to 1 KHz and event sizes around 25 KBytes are expected. In this paper we discuss the components of the proposed data collection system as well as the structure of a distributed control system to monitor detector performance.

## 1 CLEO III and CESR

The CLEO experiment is at the forefront in the world studying the properties of  $b$  and  $c$  quarks, two photon interactions, and  $\tau$  leptons. CLEO's discovery of electromagnetic penguin decays at the  $10^{-5}$  level was the major high energy physics result of 1993. The CESR  $e^+e^-$  storage ring has achieved record luminosities of  $2.9 \times 10^{32} \text{ cm}^{-2}\text{s}^{-1}$  and an integrated luminosity of  $284 \text{ pb}^{-1}$  in a single month. An upgrade program for CESR to increase the luminosity by an order of magnitude has been approved.

With a tenfold increase in statistics we will make precision measurements that severely challenge the Standard Model and help us to understand the next level of  $B$  physics including the reconstruction of exclusive  $b \rightarrow u$  final states and a full analysis of  $b \rightarrow s$  and  $b \rightarrow d$  penguin decays. In order to investigate this important physics, detector upgrades are necessary both to accommodate the requirements of the accelerator in the interaction region and to provide the detection resolution and particle identification needed to extract the physics. The interference between the particle identification system with the present CLEO tracking chambers dictates that major components of CLEO II must be replaced. This includes the beampipe, silicon detector, drift chambers, and time-of-flight systems. The superconducting magnet, the muon system and most important, the excellent CsI electromagnetic calorimeter can be retained. A crucial new component of the detector is the addition of a charged particle identification system to provide  $4\sigma \pi/K$  separation up to particle momenta up to 2.8 GeV/c. Data-taking with the new CLEO III detector is scheduled to start early 1998.

---

<sup>1</sup>Contact: Klaus Honscheid, Ohio State University (kh@mps.ohio-state.edu)

## 2 Data Acquisition

From our experience with the CLEO II experiment we extrapolate that even at instantaneous luminosities around  $2 \times 10^{33} \text{ cm}^{-2}\text{s}^{-1}$  the CLEO III trigger rate will be not larger than 1000 Hz. This combined with an estimated average event size of 25 KBytes defines the performance requirements for the data collection system. We discarded a dead-time free solution; allowing small amounts of dead-time helps to reduce costs as well as manpower requirements without sacrificing performance. The new CLEO III front-end databoards will have a read-out and conversion time of less than  $20 \mu\text{s}/\text{event}$  so that even at maximum trigger rates the read-out induced dead-time will not exceed 2%. Other design criteria include modularity as well as the usage of standards and commercial components where ever possible.

A schematic view of the elements of the CLEO III data acquisition system is shown in Figure 1. For each event trigger, approximately 600,000 detector channels have to be read. Front-end data are digitized in parallel and buffered locally on each data-board for later asynchronous readout by the data acquisition system. Data sparsification is performed directly on the data-boards. The Data Mover, a dedicated module in each front-end crate, assures transfer times below  $500 \mu\text{s}$  and provides a second buffer level. Approximately 25 front-end crates are needed for the CLEO III detector. Both, Fastbus and VME are supported. Using optical data links the data will be transmitted from the front-end crates to the eventbuilder unit where complete events are assembled. The eventbuilder is followed by a final trigger stage (Level 3) implemented in software on a fast workstation. Independent from the main data path, a slow control system will monitor the individual detector components. Run control as well as the initialization of the complete detector system will also be part of slow control.

Data collection and slow control, the two main components of the CLEO III DAQ system will be discussed in greater detail in the following sections.

## 3 Data Collection

### 3.1 Front-end Crates

Besides the detector component specific data-boards, a CLEO III front-end crate contains a crate controller CPU and a data mover module. We will use Fastbus for commercial systems and 9u VME for custom designs.

The selection of the crate controller module is not critical. Only requirement is a network interface with TCP/IP support. A server program installed on each crate controller allows remote access to the front-end crate.

Event fragments are collected from the data-boards Via the backplane bus and are buffered again on the Data Mover module. Commercial modules, eg. the RIO II by CES, provide fast VME DMA engines and Megabytes of buffer space. CLEO specific extensions, such as a high speed serial data link or an interface to the data flow control system can be added in form of PCI mezzanine boards. In Fastbus systems we will use the FRC developed by FNAL.

The Data Mover tags the event fragments with an event number and transfers the data

to the eventbuilder via a high speed serial link. This part of the data-collection sequence will be data driven. We are currently developing an optical data link with a PCI interface. The AMD Taxi chipset provides sufficient performance for our data rates (10 MBytes/s).

### 3.2 Eventbuilder

The design of the CLEO III eventbuilder is still under discussion. In the current model, data are received via optical links and FiFo memories provide temporary storage (Fig. 2). These receiver cards are designed as PCI modules. They also contain some logic relevant to the control of the data flow and can be plugged directly into PCI slots available in the newest generation of high performance workstations (e.g. Digital Equipment 2100 multiprocessor server). No other hardware<sup>2</sup> is needed and eventbuilding is reduced to a software process. An alternate, more conservative approach with a stand-alone eventbuilder is shown in Figure 3. VME-Host interfaces with sufficient performance are commercially available. However, the additional VME module with 4 PCI mezzanine slots has to be custom designed. Each of these boards is connected via PCI - PCI bridge modules to a fast VME CPU module where the events are finally assembled.

In both scenarios, a dedicated, PCI based interrupt module will be used to reduce the number of interrupts to be served by the eventbuilder. An interrupt is issued when all fragments belonging to the next events have been received. Synchronizing the data flow at this point significantly reduces the complexity of the eventbuilding process without sacrificing performance since sufficient buffer space is already provided in the Data Mover module. The eventbuilder will be located in an area that is accessible during data taking.

### 3.3 Level 3 and Data Flow Control

A third trigger level will be implemented in software. A computer delivering at least 1000 Specint92 is needed to process the event stream in real time. A special process, the Event-Broker, reads the eventbuilder output buffer and distributes the events to the different event consumers. Events passing this trigger level are stored on magnetic tape.

A schematic drawing of the data flow is shown in Figure 4. The data transfer between the different buffer stages is asynchronous. A combination of hardware and software signals is used to prevent each stage from overflowing. Data are transferred as long as the backpressure bit indicates that at least one more event buffer is available. The communication between the Data Mover and the eventbuilder is controlled by a counter on the receiver board and a backpressure line going back to the corresponding front-end crate. Event fragments sent by the crates are limited to a maximum size. A special symbol indicates the completion of the transmission. This symbol is caught by the receiving logic to update a slot counter. The backpressure signal is activate when this counter reaches a predefined threshold. The counter is automatically decremented when an event fragment is transferred to the eventbuilder. Data-taking will be disabled should all buffers including the data-boards themselves fill up.

---

<sup>2</sup>With the exception of some fan-out system to increase the number of available PCI slots to 25.

## 4 Slow Control

The slow control system is responsible for controlling and monitoring the detector. It has to guarantee that all sub-components work in established limits. We have chosen to implement the CLEO III control systems as distributed system. Spreading the functionality over several computers allows the more complex detector components to have their own, dedicated control systems while other tasks can be combined on a central machine. The initial view of this system is a set of component or equipment computers interconnected by a network and running programs based on commercial products such as National Instrument's LabView. The central machine will coordinate the sub-systems and provide the user interface for the shift personnel. The design is based on these guidelines

- **Modularity.** A modular design is essential for easy maintenance and to guarantee standard solutions for similar problems.
- **Commercial Solutions.** To limit the manpower requirements we employ commercially available products where possible.
- **Platform Independence.** By choosing standard network protocols such as TCP/IP we can design platform independent communications packages. This allows the detector components to select the best platform for their specific requirements.

A block diagram of the system is shown in Figure 5.

### Central Slow Control

Central elements of the slow control system such as alarm handling, run control and user interface are implemented in the master slow control process. A database will be used to keep a record of the detector configuration and to store calibration constants. Information is gathered from the sub-detector slow control systems using a client-server approach over a local area network. This path is completely independent form the data read-out path.

### Local Slow Control

The component specific slow control systems lie in the responsibility of the individual sub-detector groups. A typical local slow control system will consist of a personal computer running a program like LabView and some data acquisition hardware such as temperature and position sensors as well as ADC's etc. A connection to the front-end crates can be established via the slow control local area network and the crate controller CPU. The crate controller also monitors the operation of the data-boards during data taking. The local slow control systems in combination with the crate controller CPUs are also used to configure the data acquisition system at the begin of a data taking run and to download the calibration constants to the front-end data-boards.



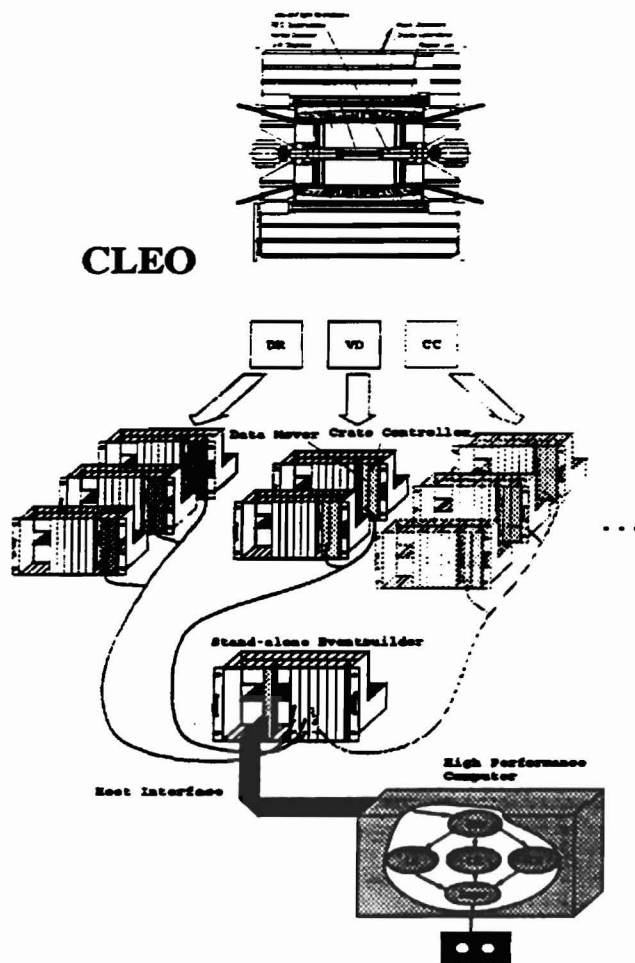


Figure 1: Schematic view of the CLEO III data acquisition system. The slow control network is not shown in the diagram.

## Optical Data Link (Receiver)

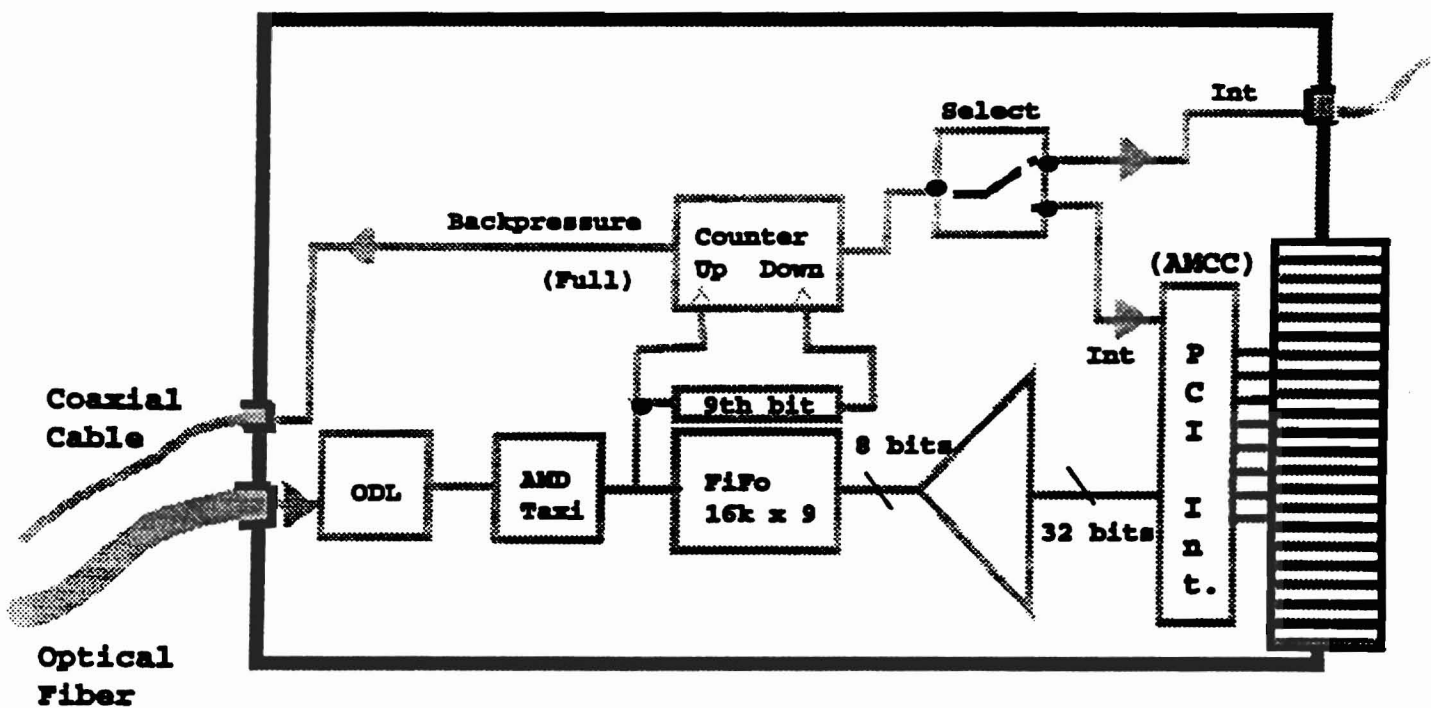


Figure 2: Schematic view of the data link receiver module

# CLEO III Eventbuilder (Stand Alone Version)

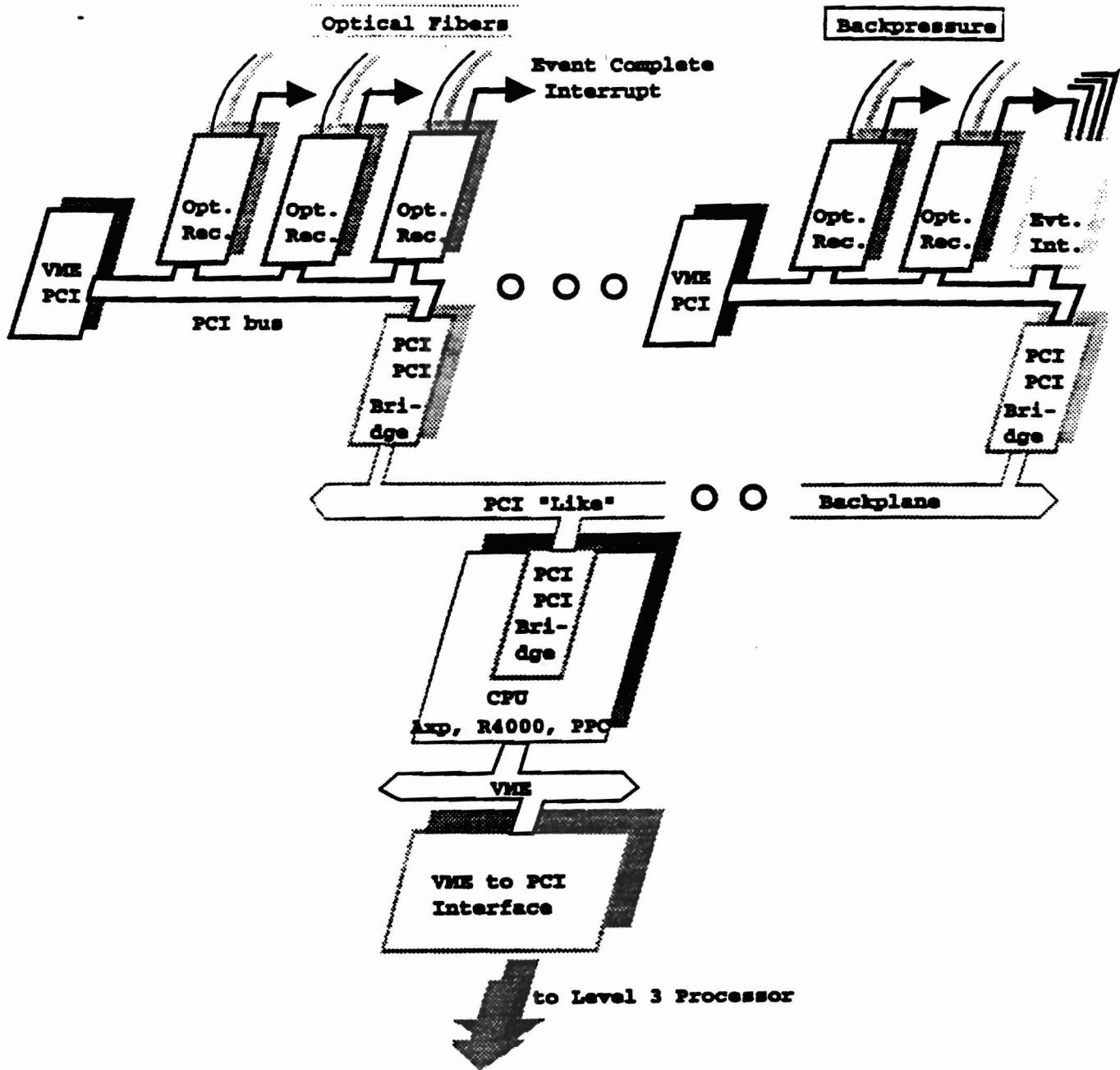


Figure 3: Schematic view of the CLEO III eventbuilder (Stand-alone version).

CLEO III Data Flow

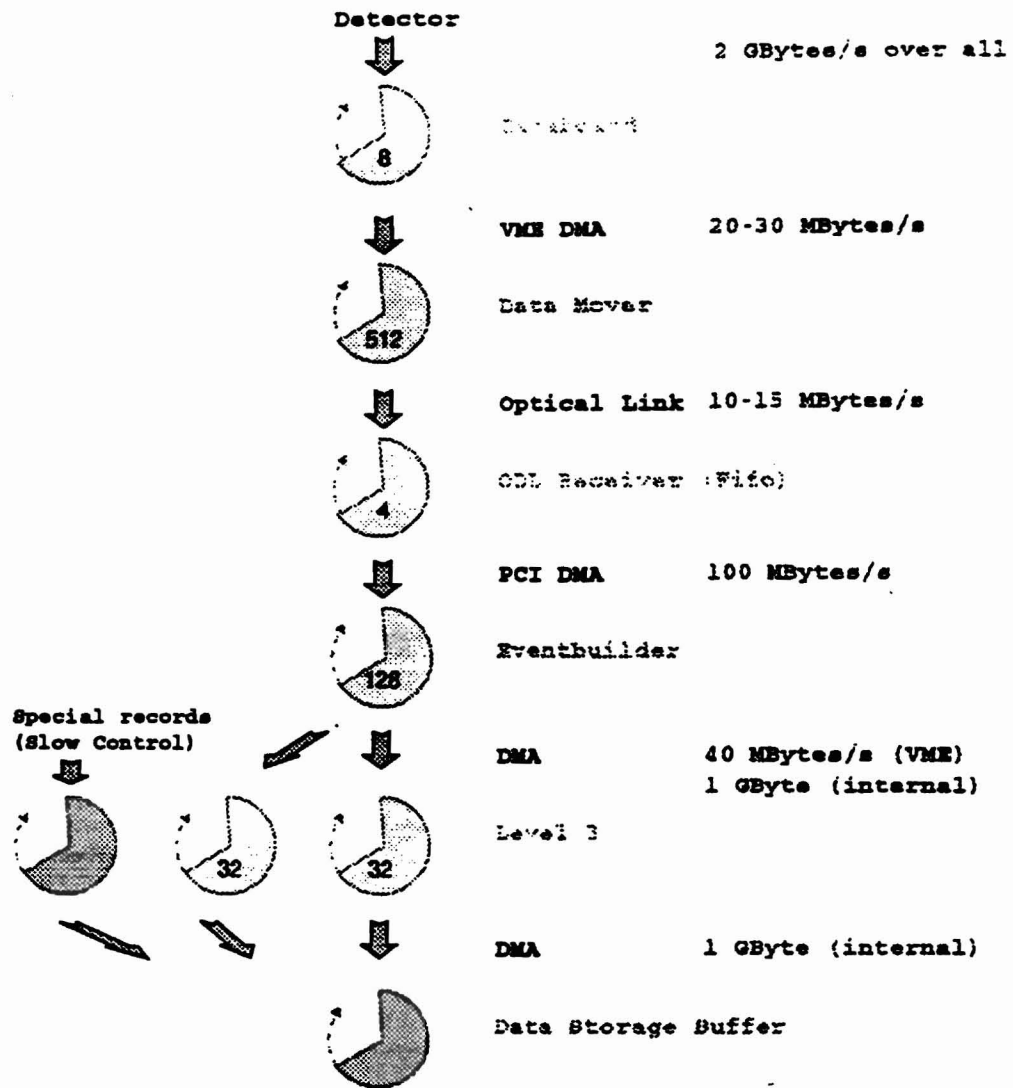


Figure 4: The CLEO III Dataflow. The circle indicate the different buffer stages. The number of slots as well as the required data transfer bandwidths is also given.

# Master CLEO Slow Control

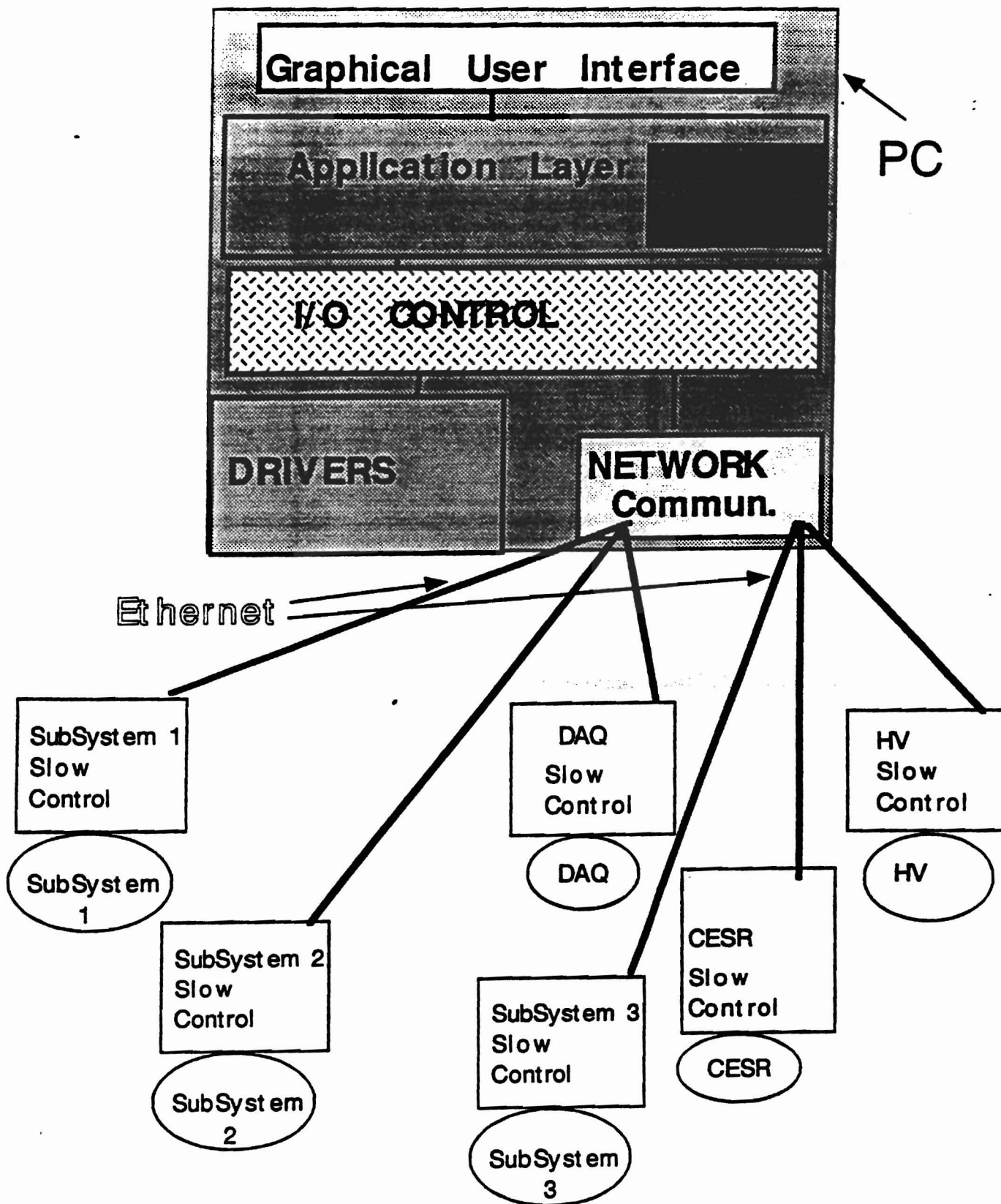


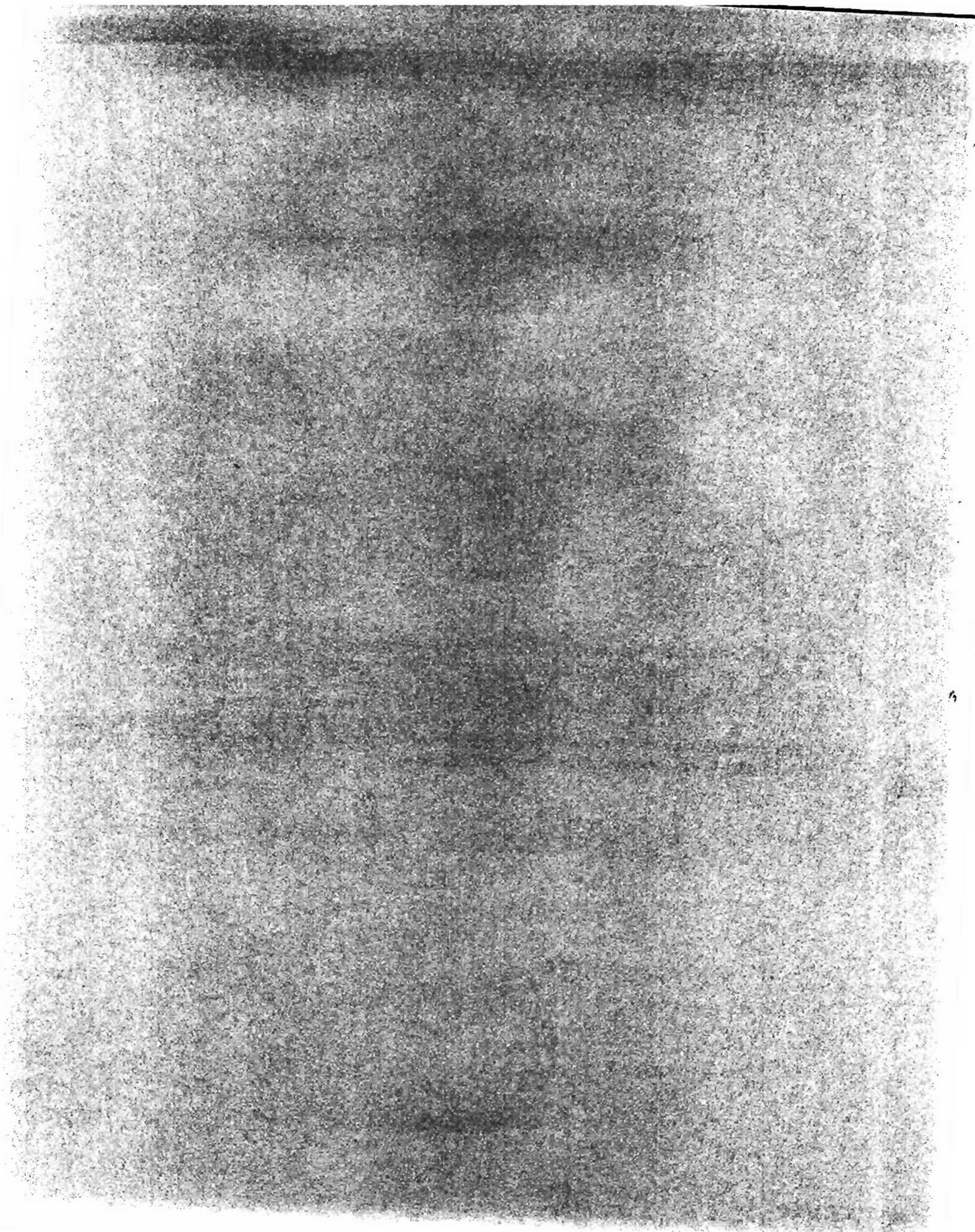
Figure 5: Schematic view of the CLEO III slow control system.

## **DART Data Acquisition System**

**Gene Oleynik**

**Fermilab**

**DART is the data acquisition collaboration between the Fermilab Computing Division and a number of Fermilab experiments. The collaboration's goal is to meet experiment needs for data taking in the 94-96 time frame and beyond. These needs include data rates into level 3 of up to 20KHz and 70 Mbytes/sec, online event filtering CPU power for acceptance ratios of up to 1 in 40 events, data logging rates from 1-20 Mbytes/sec, and incrementally functional systems for detector commissioning. DART provides a common integrated set of hardware and software to this end using well-established technologies and techniques.**



# DART Data Acquisition System Architecture\*

G. Oleynik, L. Appleton, J. Anderson, D. Berg, D. Black, R. Forster, J. Franzen,  
S. Kent, R. Kwarciany, J. Meadows, C. Moore, V. O'Dell, R. Pordes, D. Slimmer,  
J. Streets, O. Trevizo, L. Udumula, M. Vittone, M. Votava, N. Wilcer

Online Systems Department, Fermilab

V. White, Computing Division, Fermilab

Jürgen Engelfried, E781, Physics Section, Fermilab

Taku Yamanaka, E832, Osaka University

Cedric M. Guss, E811, Cornell University

Eric Stern, E815, Columbia University

George Zioulas, Eric Van Drunen, E835, University of California at Irvine

Francesco Prelz, E831, Physics Section, Fermilab

## Abstract

DART is the data acquisition (DA) collaboration between the Fermilab Computing Division and a number of Fermilab experiments [1]. The collaboration's goal is to meet experiment needs for data taking in the 1994-96 time frame and beyond. These needs include data rates into level 3 of up to 20KHz and 70 Mbytes/sec, online event filtering CPU power for acceptance ratios of up to 1 in 40 events, data logging rates from 1-20 Mbytes/sec, and incrementally functional systems for detector commissioning. DART provides a common integrated set of hardware and software to this end using well established technologies and techniques.

We describe the DART System Architecture in this paper.

## Introduction

DART has been established as a collaborative project between a number of Fermilab experiments and the Fermilab Computing Division. The system hardware and software architecture must be simple enough for the small experiments, yet extensible and fast enough for the larger ones.

The DART system architecture is *parallelized, extensible, networked and distributed*. In terms of hardware

TABLE 1. DART DA Parameters

	Small Expts	Large Expts
Trigger rate (KHz)	<.1	10-20
Event size (KByte)	1-12 (up to 200)	5-8 (up to 200)
Rate to event builder (MByte/sec)	1-3	30-160
Event building (MByte/sec)		50-160
# parallel streams	4-6	4-12
# parallel event building VME crates	1	1-4
Max. rate per stream (MByte/sec)		20-40
CPU power for event filter (Mips)	None	1000-3000
Logging (MByte/sec)	1	6-20

components, this means that sub-systems and readout are independent and in parallel, the event building architecture is modular and extensible, and Ethernet is used for control.

For the software architecture, support is given for stand-alone use of sub-systems and embedded processors for commissioning, and for integration of multiple copies of DA components as a tightly coupled system

\* This work is sponsored by DOE contract No. DE-AC02-76CH03000



during data taking.

## 1 DART Hardware Architecture

The major considerations of the DART hardware architecture were that it scale to all proposed and upcoming experiments (see Table 1), that it support the large variety of front end modules and readout controllers in use by the experiments, that all technology be well established, and that all hardware modules be commercially available as much as possible.

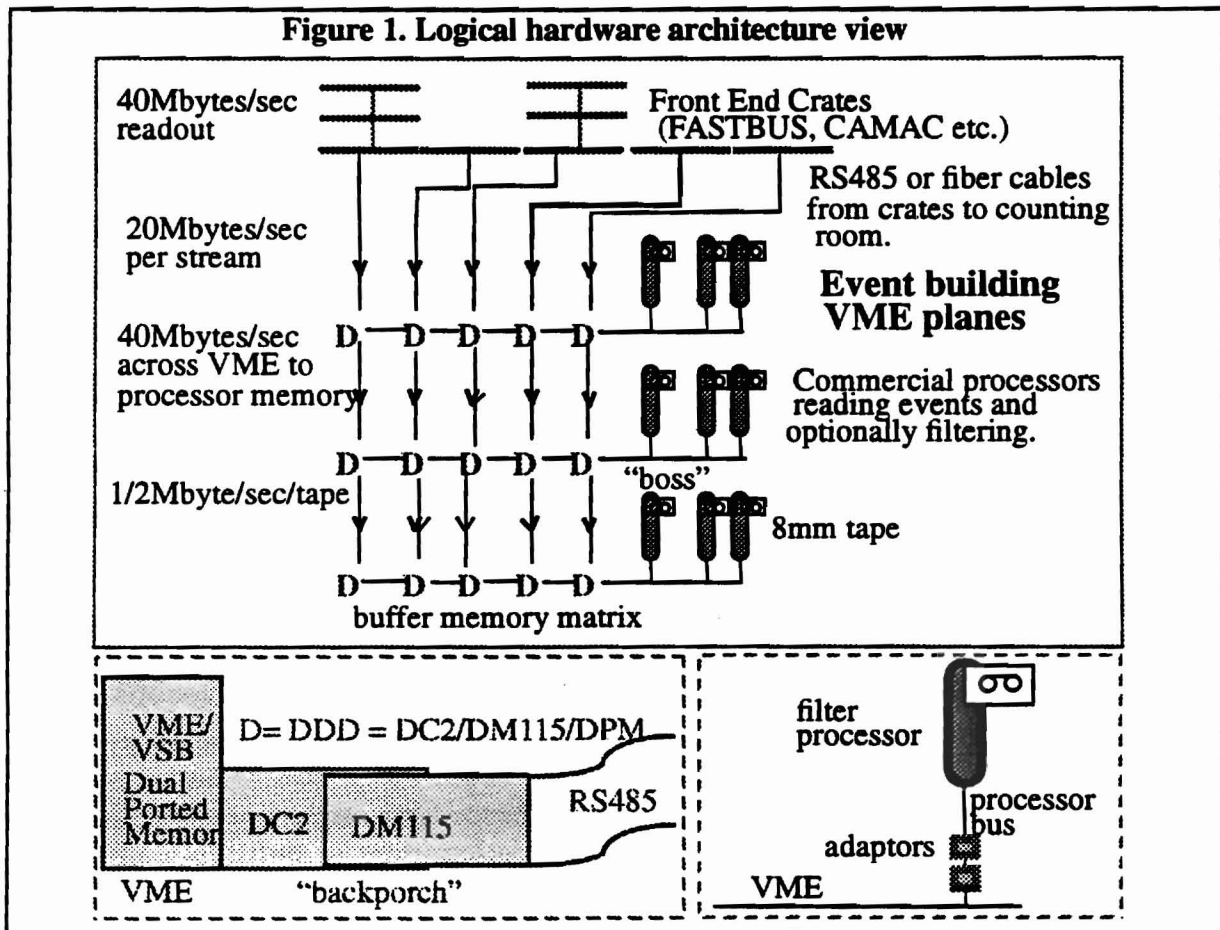
A logical view of the hardware architecture is shown below. Event fragments are read out from front end digitization crates in parallel "streams" in order to maximize experiment live-time. These fragments are buffered in intelligent dual ported memory nodes (D in the figure) residing in one of a number of VME crates for later readout. Event fragments from the memory nodes are read into filter processors where events are built,

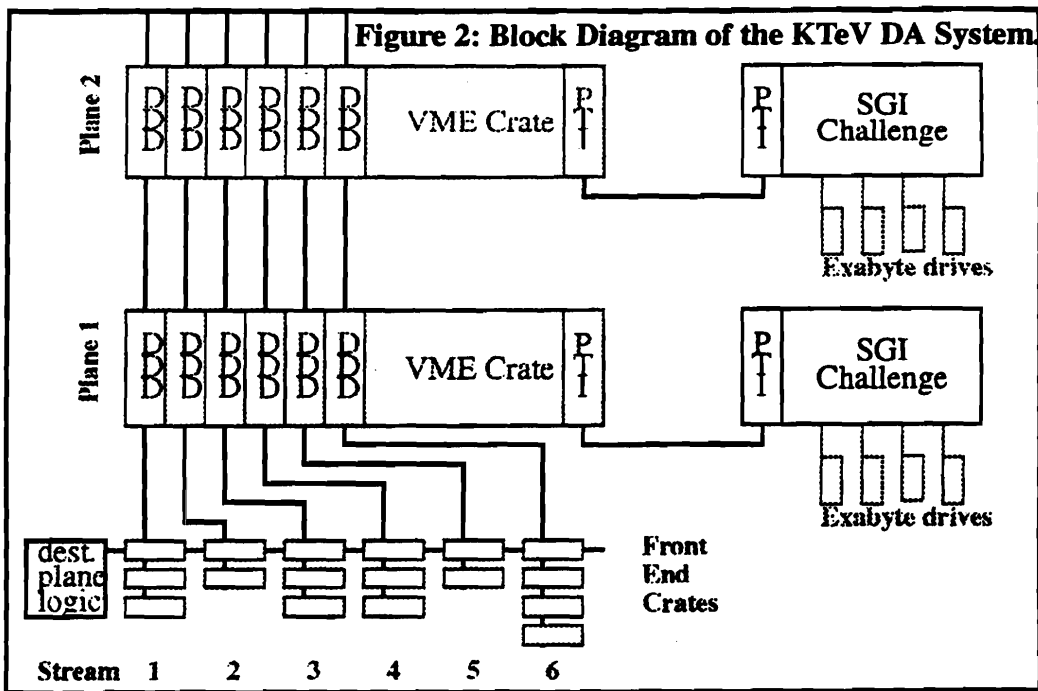
analyzed for acceptance, and logged to tape.

Experiments with bandwidths higher than that of a single VME bus use more than one VME plane to absorb the extra bandwidth. In this case, each plane, in conjunction with its filter processors, acts as an independent event builder/filter.

The KTeV experiment uses multiple planes, and its DA system [2] is shown in Figure 2. KTeV filters on Silicon Graphics Challenge L processors. As shown in the inset below, the filter processors are not required to reside in the VME crates, but can be linked to these crates via VME-to-processor adaptors, as with the Performance Technology (PTI) link in the KTeV DA.

The key elements of the architecture are the intelligent dual ported DDD memory nodes, which consist of a triumvirate of modules - the DM115, DC2 and Dual-ported VSB/ VME memory - the protocol specification used over the RS-485 input, and firmware to





manage event data in the memories.

The DM115 [3] provides for input from RS485 at up to 40 MByte/sec to a 4 KByte data FIFO, and for receipt of data in different VME crates based on the value of an address word in the data stream; the DC2 [4] controls data flow at up to 22 MByte/sec from the FIFO over VSB to commercial dual-ported VSB/VME memories (DPMs) and handles their memory management and flow control through custom firmware. The DC2's embedded 68340 processor gives it flexibility at the expense of simplicity, but this was a trade-off we accepted in order to use an already commercially available design.

The DPMs can be configured in size and number to meet the individual experiment's needs. Each memory is split into a control and event data area. The DC2 communicates the location of event data in the DPM to VME by building a table of pointers in the control area. Software in the filter processors uses this table to locate and DMA event data into its memory across the link.

The RS-485 protocol specified by DART has been implemented on a number of front end read-out controllers: the Fastbus Smart Crate Controller, the DYC+ (FERALine), the CAMAC Smart Crate Controller, and is

being implemented for custom experiment readouts such as the CROS and RMH systems in use by E781 [5].

## 2 DART Software Architecture

The ranges in size, complexity, and requirements of the DART experiments, and the requirement to have the components of the DA available incrementally for commissioning, led to several key architectural characteristics: Modularity; Fully distributed in the network sense; Easily Configurable for multiple configurations (e.g. normal vs. calibration running); Easily Extensible, tailorable, and customizable, with a base system that covers most experiments needs with little extension; Simple to use.

By fully distributed, we mean that the software supports a multi-node DA, including a controlling host node, filter processors and embedded front end processors such as read-out controllers in Fastbus and VME. For embedded processors, the VxWorks [8] operating system provides standard BSD networking. The software architecture allows an application on any node to participate in the DA at a high level without having to have knowledge of the network topology or other applications on the network. This is accomplished through the

client server model with a number of servers providing the various distributed functionality. DA applications are addressed logically with named groups rather than physically by node address and process ID.

Each application in the DA defines a set of configuration parameters which are kept in a "database" which is accessible over the network. This database can be loaded with values from an ASCII file, and each DA application provides such a file as a template which is then customized for individual experiment configurations.

The whole of the DA is easily extensible, from modifying the operator control panel and its commands to adding new applications that can respond to existing or new run control commands. This is partly supported by the use of the tcl [6] command line interpreter, which allows a large portion of the DA control to be written in scripts that do not require re-compilation when modified, and the companion tk and wish graphical interface toolkit. The control architecture uses techniques based on a high level of abstraction, such as implementing run control on top of a group-multicasting framework, in which run control commands are multicast to named groups to which DA applications register.

The major DART components [1,7,9] that support these architectural goals are:

- A graphical and line mode control program, from which the DA operator "multicasts" commands to a distributed set of DA applications, and a function library which DA applications use to register for, receive, and process these commands.
- A program which allows all DA applications to be started from single script from a single host node, and through which the terminal output from all DA applications can be logged and displayed.
- Network accessible "databases" for obtaining application specific configuration parameters, recording a run by run history, and distributing DA statistics for mon-

itoring the DA system - all supported by a single framework.

- A distributed error reporting system that decodes, displays and logs messages, and provides Unix applications and libraries with a VMS MESSAGE like way to return status up from function calls.
- Local buffer manager and (buffer) service provider software. Conceptually, this software provides extensions to the operating system in areas of memory management and process queueing specifically for data acquisition needs.
- Gateways to DMA event data from the Event buffering VME crates into filter processors.
- Ancillary event distribution software that allows back-end analysis programs to connect to event servers on front end or filter nodes to sample events without interfering with data acquisition.
- Logging software that supports logging to disk files, streaming to multiple tape drives and automatically switching to free drives while the previous rewind.

DART software packages are designed either as libraries to be embedded in experiment applications, or applications that support hooks for inclusion of experiment specific code, and are all configurable through the distributed configuration parameter system.

In addition to this software, DART specifies a number of standards which make the use of these software tools cohesive, some of which are:

- Standard run control command names, e.g. da\_start, da\_stop, da\_snapshot, etc., and arguments. These are implemented in the operator control program as customizable tcl procedures.
- Standard group names to which these commands are "multicast", such as "logger", "trigger-manager", "filter". These are chosen so that commands can be multicast to the groups in the correct time sequence. They are used by the operator program's command procedures.

- DA parameter, statistic, and run-history information name spaces.

These software products and templates are organized into a standard DA account product that is delivered to experiments as a base system.

Comprehensive information and documentation about the DART data acquisition system is available through a DART World Wide Web server a URL of: <http://fndaub.fnal.gov:8000/>.

## References

- [1] DART - Data acquisition for the next Generation Fermilab Fixed Target Experiments. G. Oleynik, J. Anderson, L. Appleton, D. Berg, D. Black, J. Engelfried, B. Forster, J. Franzen, S. Kent, R. Kwarciany, J. Meadows, C. Moore, R. Pordes, D. Slimmer, J. Streets, O. Trevizo, L. Udumula, M. Vittone, M. Votava, V. White. G. Oleynik et al., IEEE Transactions on Nuclear Science, Vol 41, No 1.
- [2] KTeV Data Acquisition System, Internal memo, V. O'Dell Fermilab, T. Yamanaoka, Osaka University.
- [3] DART Specification Document for DM115 Receiving Board, Oscar Trevizo, Fermilab.
- [4] DC2 VSB Input Controller User Manual, Access Dynamics Inc.
- [5] The E781 Trigger and DAQ System, Internal memo H-643, J. Engelfried Fermilab, et al
- [6] Tcl and the Tk Toolkit, J. Ousterhout, Addison Wesley Computing Series.
- [7] MURMUR - A message Generator and Reporter for UNIX, VMS and VxWorks, G. Oleynik, L. Appleton, B. Mackinnon, C. Moore, G. Sergey, L. Udumula, FERMILAB-PUB-93, Jun 1993, Submitted to IEEE Trans. Nucl. Sci.
- [8] VxWorks is a registered trademark of Wind River Systems, Inc.

- [9] Data Flow Manager for DART, D. Berg et al.; DBS, an rlogin Multiplexor and Output Logger for DA Systems, G. Oleynik et al., Fermilab, presented at Chep '94.



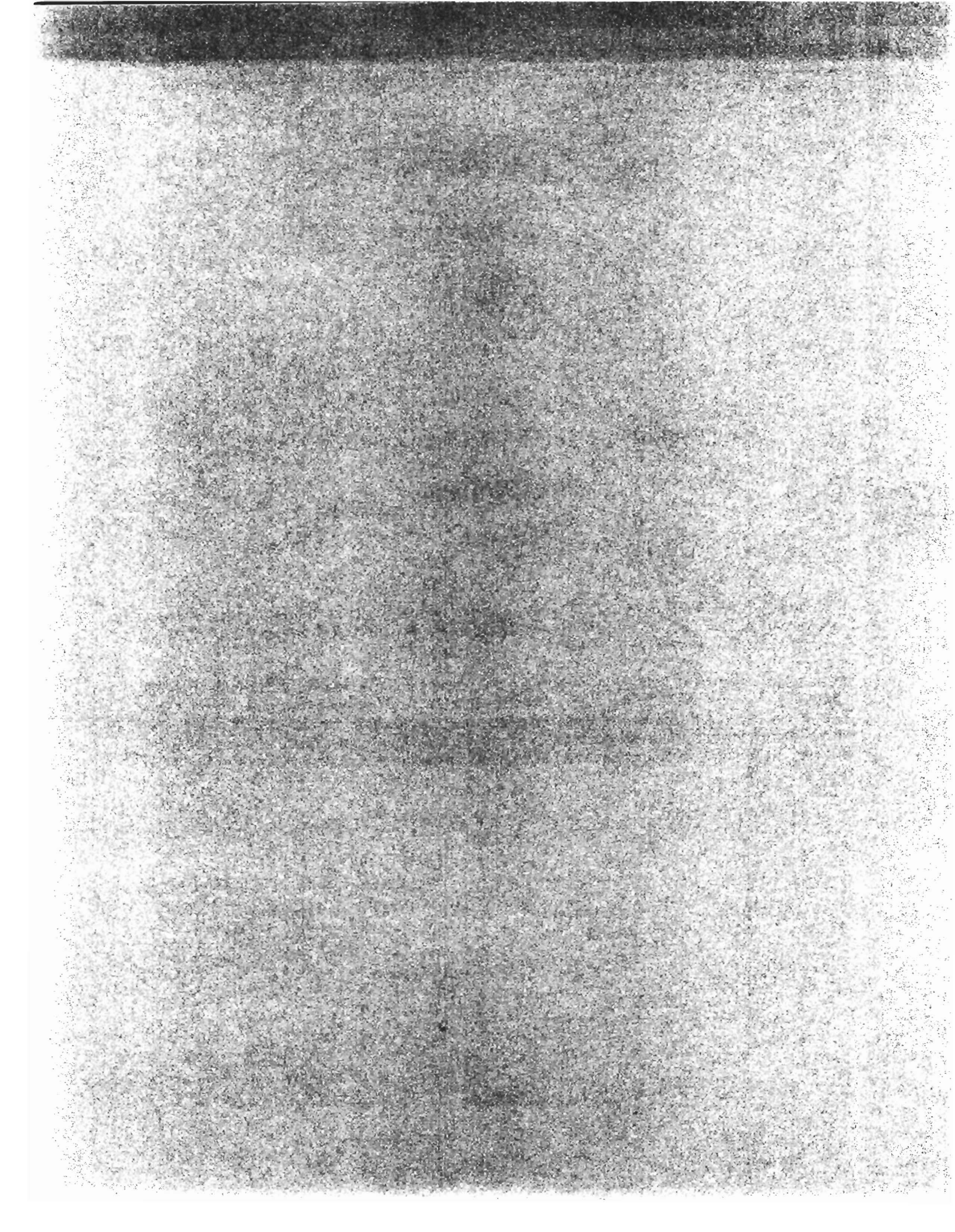
## **Sloan Digital Sky Survey Data Acquisition System**

**Don Petravick**

**Fermilab**

**The Sloan Digital Sky Survey (SDSS) will image  $\pi$  steradians about the northern galactic cap in five filters, and acquire one million spectra using a dedicated 2.5 meter telescope at the Apache Point Observatory in New Mexico.**

**We describe the data acquisition system for the survey's three main detectors: an imaging camera utilizing 54 SITE charge-coupled devices (CCD); a pair of spectrographs, each using a pair of CCDs, and a smaller monitor telescope camera. We describe the system's hardware and software architecture, and relate it to the survey's special requirements of high reliability and well understood instrumental systematics so it can produce a consistent survey over a five year period.**



## Sloan Digital Sky Survey Data Acquisition Systems

D. Petravick, E. Berman, B. MacKinnon, T. Nicinski,  
R. Pordes, G. Sergey, R. Rechenmacher, J. Bakken  
Online System Department

J. Annis, S. Kent, T. McKay, C. Stoughton  
Experimental Astrophysics Group

D. Husby  
Electronic Systems Engineering Department

Computing Division  
Fermi National Accelerator Laboratory  
Batavia IL, 60510

### ABSTRACT

The Sloan Digital Sky Survey (SDSS) will image  $\pi$  steradians about the northern galactic cap in five filters and acquire one million spectra using a dedicated 2.5 meter telescope at the Apache Point Observatory in New Mexico.

We describe the data acquisition system for the survey's three main detectors: an imaging camera utilizing 54 SITE charge-coupled devices (CCD), a pair of spectrographs each using a pair of CCDs, and a smaller monitor telescope camera. We describe the system's hardware and software architecture, and relate it to the survey's special requirements of high reliability and well understood instrumental systematics necessary to produce a consistent survey over a five year period.

### 1.0 BACKGROUND

The SDSS is a collaborative effort between Fermi National Accelerator Laboratory (Fermilab), the University of Chicago, Princeton University, the Institute for Advanced Study, Johns Hopkins University, and the Japan Promotion Group. The survey will be conducted in the period 1995-2000. Its main results will be a photometric imaging survey and a redshift spectroscopic survey of galaxies and color selected quasars across a quarter of the sky about the North Galactic Cap. The imaging survey will consist of  $10^{12}$  bytes of data, from which we will extract the images of some  $10^8$  galaxies and  $10^6$  quasars. A million of the objects will be observed in the spectroscopic survey. Collectively, these data will allow the construction of a three dimensional map of the universe, whose volume is many times larger than the structures predicted by current theories of structure formation or observed in existing redshift surveys [1].

### 2.0 INSTRUMENTS

#### 2.1 Cameras

Data Acquisition for the SDSS serves three types of cameras. The Spectrograph and Monitor telescope have cameras mounting four and one CCDs, respectively, and do not impose any extraordinary system requirements[2]. However the SDSS CCD Camera, or imaging camera, depicted in Figure 1, is an extraordinary instrument which integrates 54 CCDs, produces data at 9 Mb/S, and dictates the overall requirements for the SDSS Data Acquisition systems.

The photometric array makes up the central part of the imaging camera. It consists of 30 2048x2048 CCDs, arranged in 6 scan lines of 5 chips. A different filter is mounted in front of each CCD of a scan line, allowing simultaneous imaging in five colors. To image a three degree width of sky in five filters, it is necessary to make two Time Delayed Integration (TDI) scans of twelve scan lines. The A/D conversion in the camera electronics has been carefully engineered to provide only the appropriate number of noise bits during conversion, enhancing the compressibility of the data.



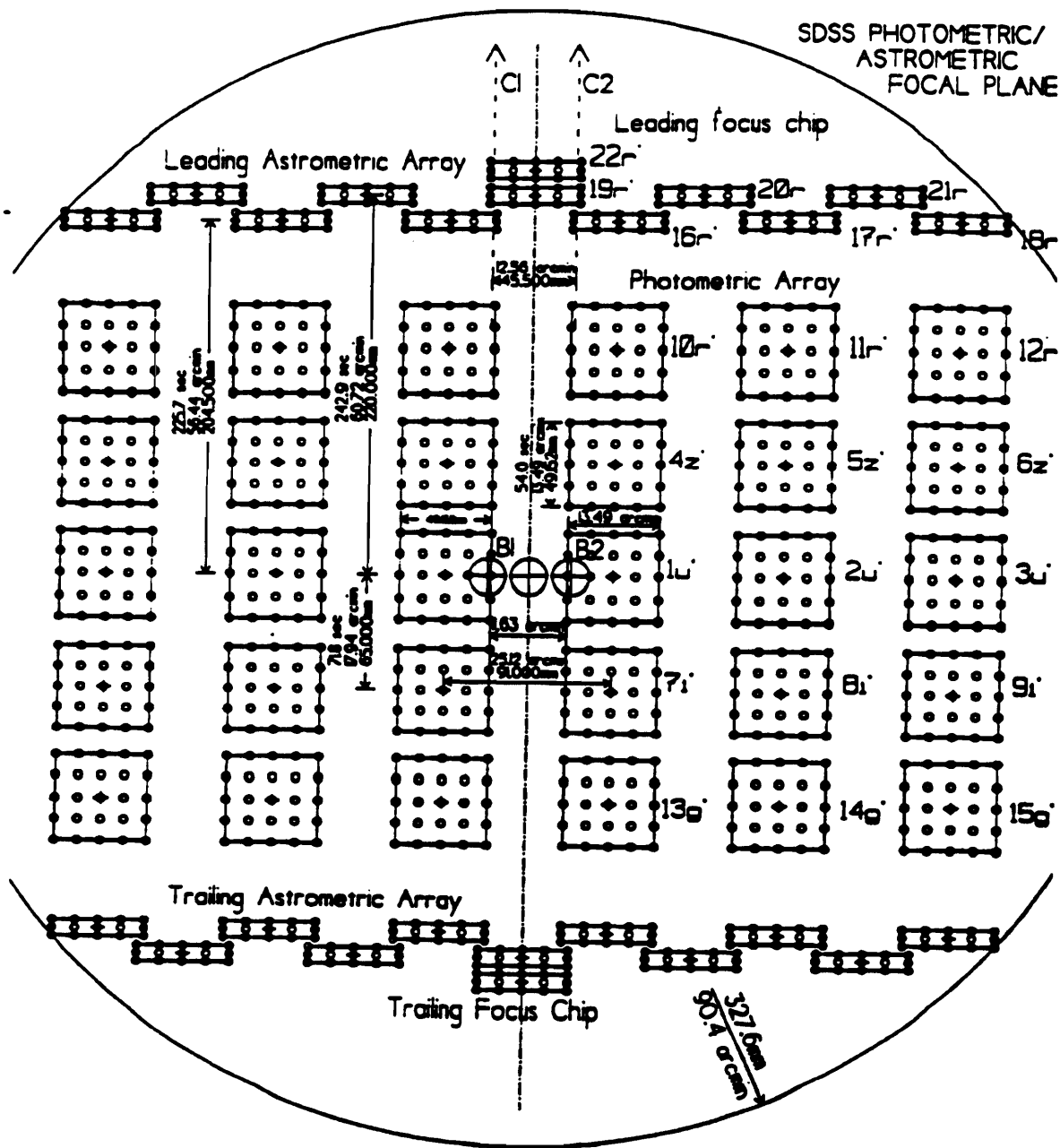


Figure 1 SDSS Imaging Camera

The remaining CCDs have 2048x400 pixels. The leading astrometric array and trailing astrometric arrays are made up of two ranks of 2048x400 chips. They are covered with a neutral density filter and will not saturate when imaging bright stars.

Images from the two focus chips are mounted about 200 microns behind the focus. Half of each device will be covered by a window altering the focus. A comparison of images between the two halves will yield a differential measurement of the focus.

The whole imaging camera is controlled over a serial line, and generates its own internal timing. Pixels are transmitted to the data acquisition system after conversion over 10 fibers – one fiber per scan line, and one fiber per rank of astrometric CCDs.

## 2.2 Telescopes

The 2.5 meter telescope imposes few requirements on the data acquisition system. Because of its accurate pointing and tracking, the only feedback required is focus. The telescope is based on the successful Apache Point 3.5 meter telescope[3].

## 3.0 SYSTEM REQUIREMENTS

### 3.1 High Level Partitioning

In its original concept, the data from all survey instruments were to be reduced as part of the data acquisition process. This caused difficulties because the detailed requirements for data acquisition and data reduction conflict. The data acquisition system must be available in a robust form early in the commissioning of the survey, while the data processing system is allowed to evolve considerably during the system's test year. Further, equipment for the CPU intensive data reduction must be specified at a relatively late date to secure cost advantage, while adequate equipment for the data acquisition system was available in 4Q92. To simplify the specification, it was decided to divide the system into two parts, a Survey Operations System (which includes the DA system) at Apache Point, and a Data Processing System at Fermilab.

### 3.2 Data Acquisition Requirements

#### 3.2.1 Imaging

The CCDs on the imaging camera fall into three classes: photometric, astrometric, and focus. There are different handling requirements for data from each class of device.

The photometric data is to be recorded in its totality, blocked into frames of 1354 rows from the CCD. This is half the distance, in rows between CCDs in a scan line, allowing the first frame from the second chip to contain an image of the same part of the sky as the third frame from the first chip, and so on. The frames are to be written to tape such that:

- All data from a single scan line are on the same tape,
- Frames from corresponding parts of the sky are written together, and
- Frames are to be written using the FITS standard.

It is desirable to build a model of the flat field and point spread function (PSF) across the whole scan, and to have this model available for the reduction of the very first frame. To this end, the DA system builds two ancillary data sets from the CCD data. The first is quartiles of the distribution of the pixels in each column of each CCD for each frame. The second is postage stamps, a set of rectangular regions of pixels centered about a pixel which passes a simple thresholding test.

Quality Assurance (QA) requirements for the photometric system dictate that the images acquired over the last 45 minutes be maintained for inspection and that access times for a given frame be a few seconds. The quartiles and postage stamps must be maintained for quality analysis inspection over a night's observing. Additionally, the system must support the simultaneous display of images from at least one selected chip from each scan line.

All that is required of the astrometric data is that postage stamps be saved about pixels which exceed a threshold. QA for the astrometric system dictates that 45 minutes of images be available. Additionally, the system must support the simultaneous display of images from at least one selected chip from each rank of astrometric chips.

Data from the focus chips needs to be collected, and a focus adjustment computed from the PSF of the detected images. Its QA requirements are as the astrometric system.

#### 3.1.2 Spectrograph and Monitor Telescope Cameras

The DA for these systems simply needs to keep up with the ADCs in the camera electronics, a few microseconds/pixel. The Monitor Telescope System must be a self-contained sub-system, for its deployment date (3Q94) is considerably ahead of the other two instruments (1Q95).

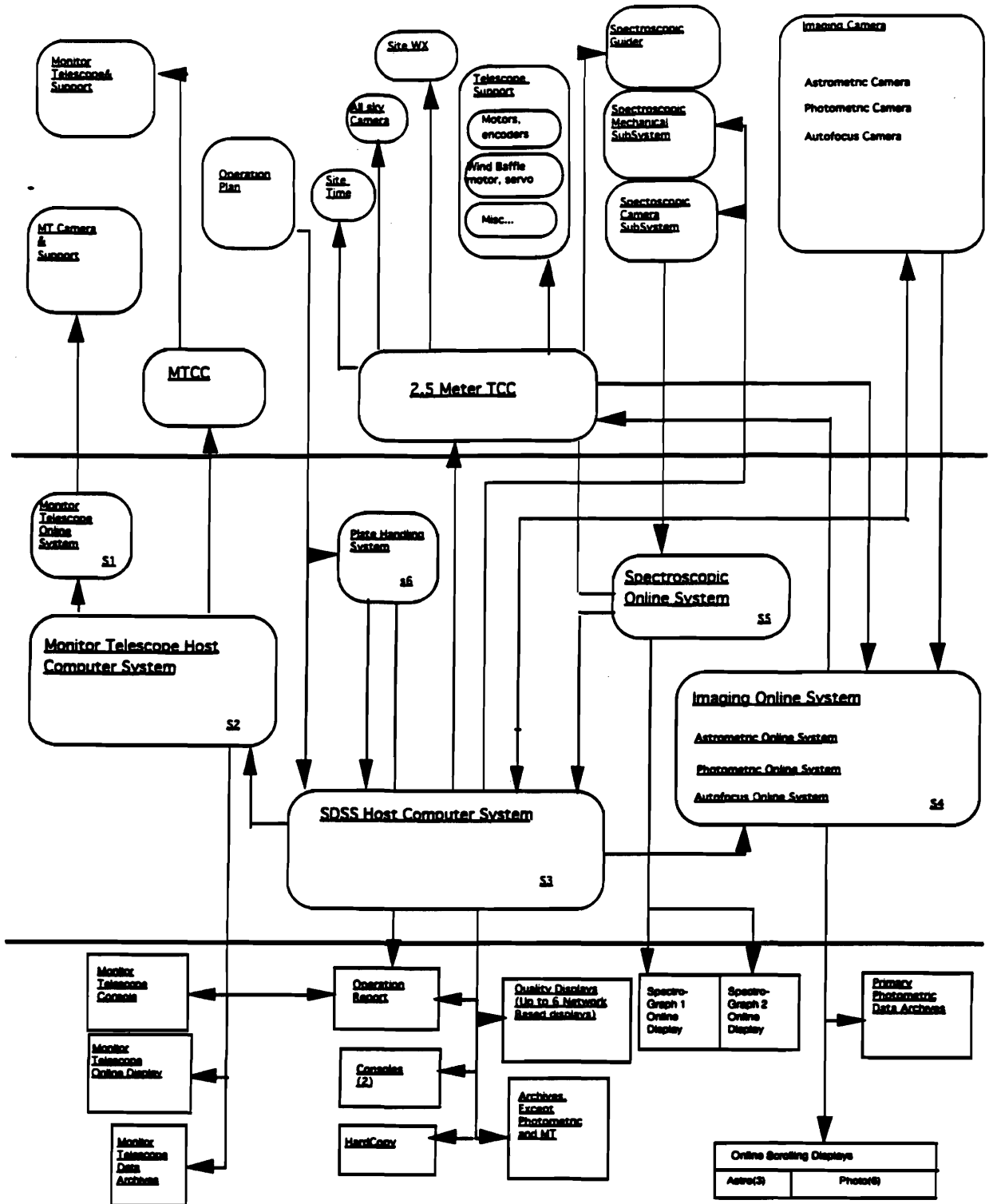


Figure 2 - Overview of Survey Operations

### 3.2 Control

The control features of all subsystems, along with all QA data are to be made available to a central control program, called Control Window (CW). A version of the program must run under a simple terminal interface to allow diagnostics to be run by experts who may not be on-site. Several copies of CW may be run simultaneously and these

copies must be made to cooperate in some fashion. The system needs to provide a number of programs indicating the status of operations.

In addition to the data products mentioned above, a large set of instrumental parameters are to be monitored, recorded, and maintained for the duration of the survey. To ensure consistency of the survey, certain parameters are to be changed only under the supervision of a software system outside the DAQ system known as Survey Strategy.

#### 4.0 HIGH LEVEL HARDWARE ARCHITECTURE

It is our experience that it is best to partition a large data acquisition problem into a number of host systems and on-line systems. Host systems are a root of system control and user interface. Online systems handle high-rate data flow.

Host systems inspect summary data and a subset of the actual data originating on the camera, serve as the root of system control, and handle ancillary data streams. These systems are the locus of the ad-hoc programming to diagnose failures and gain an understanding of the detector. Host systems run a UNIX operating system.

Online systems handle the complete data stream from the detector, compute summary data, and serve subsets of the data to the host computer. Their computing resources are carefully matched to the problem at hand. Their software is written by experts and is not subject to short term change. These systems are supervised by a host but do not depend on it for detailed intervention.

#### 4.2 System Diagram

Figure 2 illustrates how these considerations led to the high level architecture of the SDSS DA systems. The figure is divided into three sections. The uppermost section illustrates other survey components with which the system needs to interface. The middle section, excluding S6 (Plate Handling) represent the DA system. The lower section represents external interfaces.

Since the Monitor Telescope is to be delivered early a separate host and online system (S1,S2) have been provided for it. These systems are slaved to the SDSS host computer system in final operation.

Distinct online systems are supplied for the spectrograph (S5) and imaging camera (S4), making the architecture somewhat more robust should the delivery of the instruments slip relative to one another. A single host computer (S3) serves for imaging and photometry.

#### 4.3 High Level Features of Online Systems

The systems are built around VME backplanes connected by a VME interconnect and disk/tape systems are integrated around SCSI bus. The software uses VxWorks as its real time operating system.

#### 4.4 High Level Features of Host Systems

The host systems are off-the-shelf computers; a SGI 4D/35 with 112 Mb of memory is the Monitor Telescope Host System, and a SGI Crimson with 256 Mb of memory is the SDSS Host System. Each system has VME interfaces and are configured with several Gigabytes of disk.

#### 5.0 THE ONLINE SYSTEMS

All of the online systems are built around the eight modules and six interconnects. Figure 3 shows a configuration of these components in the Imaging Online System which services three scan lines of the photometric camera. The Imaging Online System is built around three of these VMEbus backplanes. The Spectroscopic Online System and Monitor Telescope Online System each have one similar backplane. The configuration of these VMEbus systems differs in detail.

# Photometric Online System for 15 CCDs

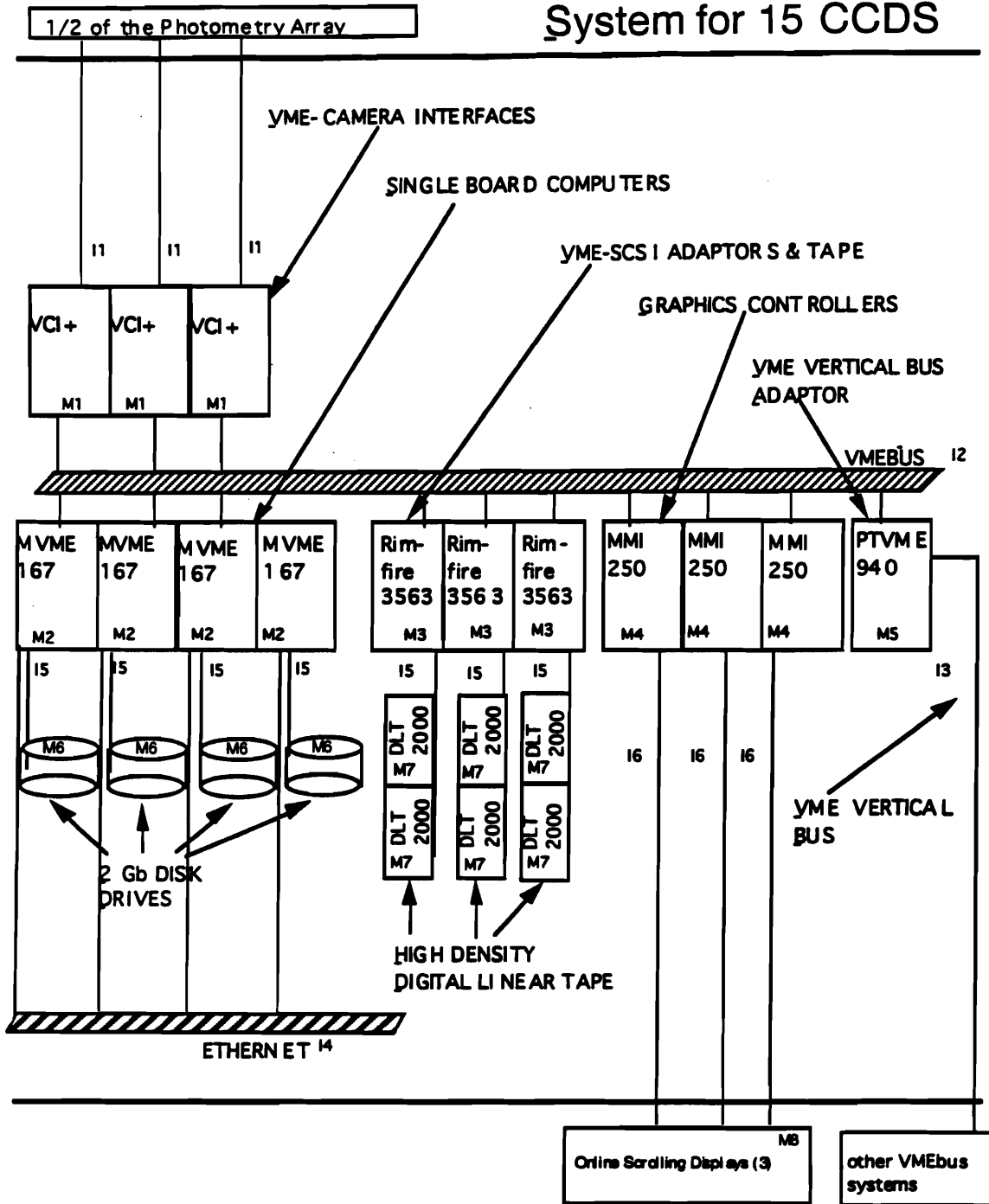


Figure 3- Online system for 15 CCDs

## 5.1 Interconnects

5.1.1 The optical link from the camera (11) All cameras transmit their pixels using the TAXI/FOXI system with a common data transfer protocol. We have specified a data transfer protocol where:

- The FOXI system is configured to transmit in 10 bit bytes.
- Each FOXI fiber handles data from up to 12 amplifiers.
- Each pixel is encoded in three such bytes, an amplifier number, most significant byte, least significant byte.
- An end of line byte signifies that all amplifiers have sent their data from the current line.

5.1.2 VMEbus, IEEE STD 1014 Rev D. (I2) This is a bus system which is well supported by industry.

5.1.3 VMEbus High Performance Data Network (I3) This is a cable linking the online systems and host computers together. It supports data transfers rates in excess of 30 Mbyte/sec, more than enough to allow images to flow from the online to host systems.

5.1.4 Ethernet (I4) Ethernet is used only to download programs across the systems, to pass error messages, and to receive telescope pointing information so the FITS headers can be filled out properly.

5.1.5 SCSIbus (I5) Three meter cable limitations still allow instantaneous transfer rates of 5 MB/sec, sufficient for this system.

5.1.5 RGB Video Cable (I6) These cables are long enough to span the distance between the computer room and the operation room.

## 5.2 Modules

5.2.1 VCI+ (M1) modules connect the Optical Data link from the camera and VMEbus. The VCI+ was designed and built at Fermilab around a FOXI receiver, SRAM buffers, and a Xilinx field programmable gate array. The VCI+ maintains VME readable buffer areas for each corner of the CCDs it services.

5.2.2 MVME167 (M2) is a VME single board computer. The boards have a 33 MHz MC68040 and 32 Mb of memory. An on-board DMA engine allows the computer to simultaneously acquire data and perform computations.

5.2.3 Rimfire 3563 (M3) is a general purpose VMEbus to SCSIbus adapter. It accepts lists of buffers to write to tape, and returns an interrupt when finished.

5.2.4 Vigra MMI 250 (M4) is a VMEbus graphics controller, capable of driving a 1024x128 color monitor. These boards are quite capable of presenting a smoothly scrolling display of the sky while pixels are being acquired from the camera.

5.2.5 Performance Computer Data Network Adapter Model 940 (M5) is a VME master interface to the VMEbus High Performance Data Network (I3). The card performs 32 and 64 bit block transfers while moving data from VMEbus to VMEbus, and may generate interrupts in remote crates.

5.2.6 Micropolis 1921 Disk Drives (M6) are 2 Gb 5400 RPM disk drives, on which is realized a pool for temporary storage of the images, quartiles, and postage stamps.

5.2.7 DEC DLT 2000s (M7) are high capacity single ended SCSI cartridge tape drives. We have measured uncompressed transfer rates of 1.2 Mb/S, and expect compressed data rates as high as 2.5 Mb/S.

5.2.8 Nanao Flexscan Monitors (M8) are capable of displaying a 1024x1280 eight bit color image.

## 5.3 Online System Software

The Imaging Online System exemplifies how these system components work together. Figure 3 gives the configuration for half the data acquisition system for the photometry array. Three fibers feed into the system – each one contains data from a scan line on the camera which consists of ten corners of five amplifiers. Four MVME167s computers service the VCI+ modules, orchestrating the data acquisition.

At the end of each fiber is a VCI+ with its buffer memory configured into ten buffers. Pixels from the right half of the CCDs are loaded into the buffer memories in ascending order, pixels from the left half in descending order. When the end-of-line byte is received from the camera each VCI+ board generates two VMEbus interrupts since each MVME167 board has enough CPU power to service four CCD's worth of data.

On receipt of the interrupt, the MVME167 boards initiate DMA transfer of the pixels from the VCI+ into their memories. When the transfer is complete, they signal the VCI+ board to free the buffer area and refill it with pixels from another line. This signaling is done by writing one of two distinct VMEbus addresses. The buffer area is flushed when both locations are written to. In this way, the VCI+ board synchronizes two MVME167 computers.

After each such interrupt the MVME167 carries out four operations:

- The pixel histogram for each column is updated.
- The data is searched using a simple thresholding algorithm, and postage stamps are cut out if appropriate.
- If the pixels are to be displayed on a scrolling display, they are transferred to the VIGRA board using the MVME167 DMA feature.
- The lines are compressed and moved into a buffer.

When full, the disk buffer is marked for write and replaced with a fresh buffer. Compressed pixels are stored on disk as a FITS binary tables. When a frame's worth of data has been acquired, the histogram and disk buffers are replaced with fresh buffers. In a separate task, and asynchronously to the line-by-line read out of the VCI+, quartiles are computed from the histograms.

Several other activities occur asynchronously on the board:

- An archiver task reads the image data and programs the RIMFIRE 3563 to spool them to tape. The data are reorganized, so that corresponding parts of the sky are logged to adjacent bits of the tape. The data are archived redundantly.
- The MVME167 boards serve the quartiles, postage stamps and images to the SDSS host computer.
- A command server listens for general control messages.
- A scrolling display task is interrupted by the VIGRA board 72 times a second and advances the display by the correct number of lines.

Each MVME167 maintains a status database in its local memory. A status entry is identified by an alphanumeric name, a type (integer, floating point, *etc.*), an actual value, minimum and maximum values, a description, protection and current validity. Each node maintains about 150 parameters. These locations may be read by the host computer.

The online systems can report error and status to the host computer over the ethernet using the Fermilab MURMUR package, which is best characterized by mentioning that it both displays urgent messages to observers and records significant events into a log file.

## 6.0 THE HOST SYSTEMS

The host system serve as the root of system control. As such, their software environment is their most interesting feature. It is best to begin by describing the survey standard software tools kits which have been incorporated into its construction. Many of the common tools are described in [5].

### 6.1 Baseline tools

SHIVA (survey Human Interface and Visualization Environment) [6] is the tool kit used for supporting the real time analysis of acquired data. Shiva provides C and TCL framework to access frame regions. Shiva was developed, in part, by integrating:

PGPLOT: a package for drawing simple scientific graphs on various displays, developed at Caltech[7].

FSAOIMAGE: an X11 window based, interactive, color or halftone image display program for astronomical images adapted from the venerable SAOImage package[8], developed at the Smithsonian Astrophysical Observatory.

FTCL: A Fermilab packaging of Tcl/Tk[9]. We have added command line help, command line editing integrated with the Tk event loop, other added value features, and packaged Neosoft's extended TCL package[10]. We run the TCL system under VxWorks in the online systems as well using a port from NOAO[11][12].

LIBFITS: A procedure call package, developed by Alan Uomoto of Johns Hopkins University[13].

Help and documentation is built upon the WWW wide-area hypermedia information retrieval system developed at CERN [14]. Information browsing is supplemented by the Mosaic browser developed at the National Center of Supercomputer Applications (NCSA) [15].

Data base management is based on a commercial object oriented data base, VERSANT[16]. This system is used to keep track of a number of operational data.

Error messages and log files are kept using the Fermilab MURMUR software tool[17].

## 6.2 Applications

Nearly all of the survey's software is built around the Ousterhaut's Tool Command Language (TCL). Tcl is a C and Lisp-like user extensible command interpreter. One writes command primitives in C, and declares them to a TCL interpreter. Observing programs can be constructed in TCL from these primitives. Because other survey software has been written for TCL, it is possible to re-use other primitives related to image display, databases and so forth.

## 7.0 PROGRESS TO DATE

The DAQ systems have been purchased and are physically installed at Fermilab. The core software system is complete. The Monitor Telescope systems are ready for deployment at Apache Point, NM and awaits the delivery of the telescope. A prototype system, the Fermilab Drift Scan Camera is deployed at Yerkes Observatory in Wisconsin.

## REFERENCES

1. "A Digital Sky Survey of the Northern Galactic Cap", *Proposal*, November 12, 1993.
2. Annis, J. *et al.*, "the Sloan Digital Sky Survey Monitor Telescope", *This Conference Proceedings*.
3. Owen, R., Siegmund, W. and Hull, C., "The Control System for the Apache Point 3.5m Telescope", *Instrumentation for Ground-Based Optical Astronomy*, ed. L. B. Robinson, pp. 686-690, Springer-Verlag, New York, 1988.
4. MacKinnon, B. *et al.*, "Development of the Sloan Digital Sky Survey Online Systems", *IEEE Transactions on Nuclear Science*, Feb., 1994, in Press.
5. S.Kent *et al.*, "Sloan Digital Sky Survey", *Proceedings of the Third Annual Conference on Astronomical Data Analysis Software and Systems*.
6. Berman, E. *et al.*, "The Shiva Book" *SDSS internal document*
7. "PGPLOT User's Manual", *California Institute of Technology*
8. VanHilst, H., "User Manual for SAOImage", *Smithsonian Astrophysical Observatory*.
9. Ousterhaut, J., "Tcl: An Embeddable Command Language", *Proceedings of the Winter 1990 USENIX Conference*.
10. Lehenbauer, K. and Diekhans, M., "Extended TCL Command Set", *unpublished manual page, NeoSoft, Inc., January, 1992*.
11. D'Anne Thompson, *private communication*.
12. Vittoni, M. *et al.*, "FTCL - Tcl at Fermilab", *Fermilab Computing Division Document PN 464, 1993*.
13. Uomoto, A., "LIBFITS Reference Manual", *Johns Hopkins University*.
14. Berners-Lee, T.J. Cailliau, R. Groff, J., Pollerman, B., CERN, "World-Wide Web: The Information Universe", *Electronic Networking: Research, Applications and Policy*, Vol. 2 No 1, pp. 52-58 Spring 1992, Meckler Publishing, Westport, CT, USA.
15. "NCSA Mosaic Documentation", *National Center for Supercomputer Applications*.
16. Versant Object Technology, Menlo Park, CA.
17. Oleynik, G., *et al.*, "Murmur - A Message Generator and Reporter for UNIX, VMS, and VxWorks", *IEEE Real Time '93 Conference Record*.





## **The KLOE DAQ System**

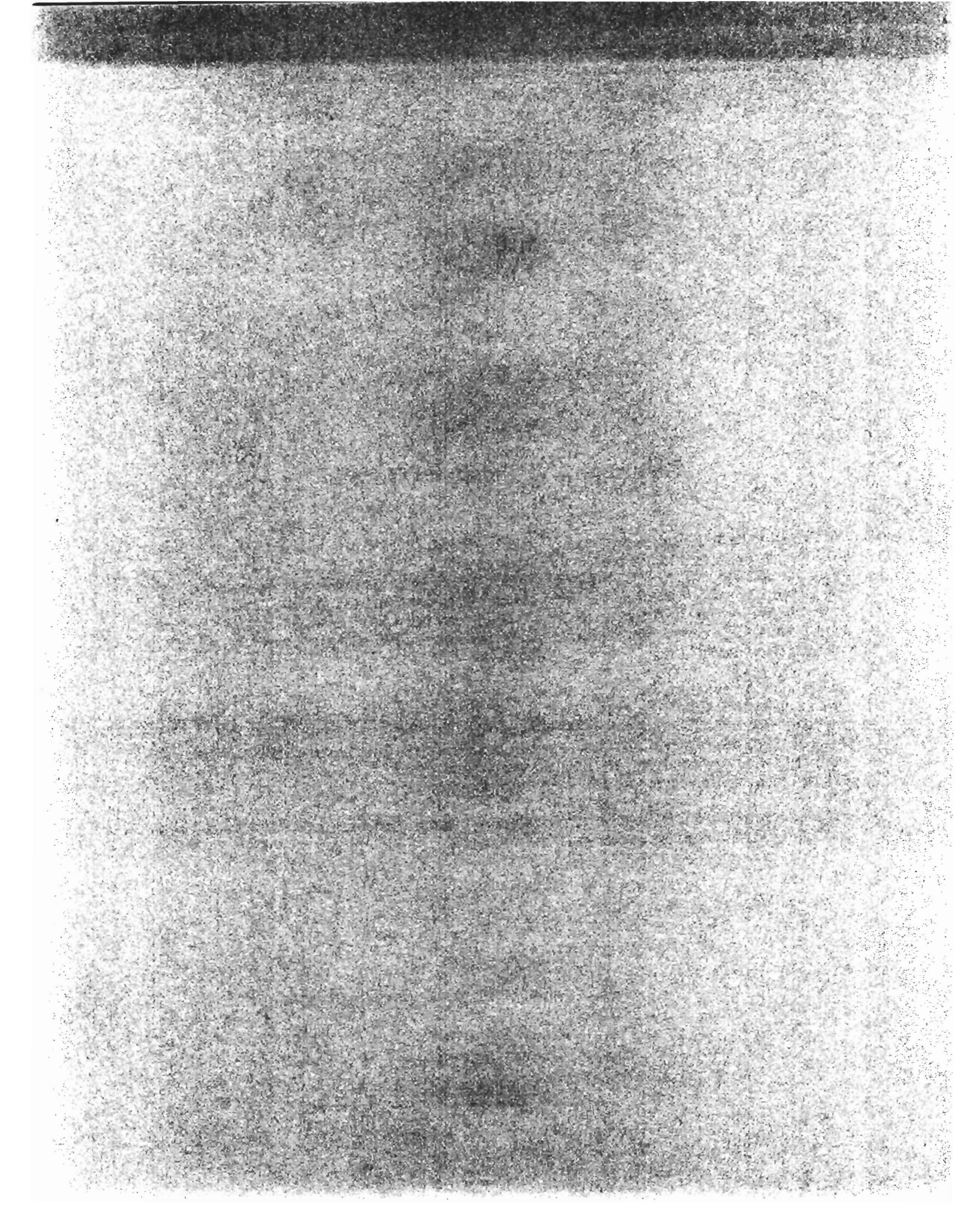
**Elizabeth Pace**

**Laboratori Nazionali di Frascati**

We describe the KLOE data acquisition system, DAQ. KLOE is a new experiment that will begin running at DAFNE, in Frascati, in 1996. The KLOE DAQ has to sustain a maximum throughput of 50 Mbytes/sec. The front end electronics is constituted of some hundreds of boards, housed in 40 VME crates. Data are collected at crate level, through a custom bus in the backplane, by a hardwired read out controller (ROCK), one for each crate. Groups of crates are connected, via a second custom bus, Cbus, to a ROCK manager (ROCKM).

Each ROCKM contains a piece of an event coming from a certain part of the apparatus, called sub-event. Each ROCKM is read by a VME processor that sends groups of sub-events to a farm of "single board computers", SBCs, via an FDDI switch, using the TCP/IP protocol. All the sub-events of the same event are received by the same SBC, where the event is tested, formatted, and sent to the mass storage system. The address of such SBC is assigned by a VME processor, the Data Flow Control, DFC.

Performance obtained using both commercial and custom hardware and software solutions are shown. Simulation results are presented.



# THE KLOE DATA ACQUISITION SYSTEM

A. Aloisio,<sup>e</sup> A. Andryakov,<sup>b</sup> A. Antonelli,<sup>b</sup> M. Antonelli,<sup>b</sup> F. Anulli,<sup>b</sup> C. Avanzini,<sup>h</sup>  
D. Babusci,<sup>b</sup> C. Bacci,<sup>j</sup> R. Baldini-Ferrolì,<sup>b</sup> G. Barbiellini,<sup>m</sup> M. Barone,<sup>h</sup> K. Barth,<sup>c</sup>  
V. Baturin,<sup>e</sup> H. Beker,<sup>h</sup> G. Bellettini,<sup>g</sup> G. Bencivenni,<sup>b</sup> S. Bertolucci,<sup>b</sup> C. Bini,<sup>h</sup>  
C. Bloise,<sup>b</sup> V. Bocci,<sup>i</sup> V. Bolognesi,<sup>g</sup> F. Bossi,<sup>b</sup> P. Branchini,<sup>k</sup> L. Bucci,<sup>h</sup> A. Calcaterra,<sup>b</sup>  
R. Caloi,<sup>h</sup> P. Campana,<sup>b</sup> G. Capon,<sup>b</sup> M. Carboni,<sup>b</sup> G. Cataldi,<sup>d</sup> S. Cavaliere,<sup>e</sup>  
F. Ceradini,<sup>j</sup> L. Cerrito,<sup>i</sup> M. Cerù,<sup>h</sup> F. Cervelli,<sup>g</sup> F. Cevenini,<sup>e</sup> G. Chiefari,<sup>e</sup>  
G. Ciapetti,<sup>h</sup> M. Cordelli,<sup>b</sup> P. Creti,<sup>d</sup> A. Doria,<sup>e</sup> F. Donno,<sup>b</sup> R. De Sangro,<sup>b</sup>  
P. De Simone,<sup>b</sup> G. De Zorzi,<sup>h</sup> D. Della Volpe,<sup>e</sup> A. Denig,<sup>c</sup> G. Di Cosimo,<sup>h</sup>  
A. Di Domenico,<sup>h</sup> E. Drago,<sup>e</sup> V. Elia,<sup>d</sup> O. Erriquez,<sup>a</sup> A. Farilla,<sup>a</sup> G. Felici,<sup>b</sup> A. Ferrari,<sup>g</sup>  
M. L. Ferrer,<sup>b</sup> G. Finocchiaro,<sup>b</sup> D. Fiore,<sup>e</sup> P. Franzini,<sup>h,j</sup> A. Gaddi,<sup>b</sup> C. Gatto,<sup>e</sup>  
P. Gauzzi,<sup>h</sup> E. Gero,<sup>b</sup> S. Giovanella,<sup>h</sup> V. Golovyatuk,<sup>d</sup> E. Gorini,<sup>d</sup> F. Grancagnolo,<sup>d</sup>  
W. Grandegger,<sup>b</sup> E. Graziani,<sup>k</sup> U. v. Hagel,<sup>c</sup> R. Haydar,<sup>b</sup> M. Imhof,<sup>c</sup> M. Incagli,<sup>g</sup>  
C. Joram,<sup>c</sup> L. Keeble,<sup>b</sup> W. Kim,<sup>l</sup> W. Kluge,<sup>e</sup> F. Lacava,<sup>h</sup> G. Lanfranchi,<sup>h</sup> P. Laurelli,<sup>b</sup>  
J. Lee-Franzini,<sup>b,i</sup> A. Martini,<sup>b</sup> A. Martinis,<sup>m</sup> M. M. Massai,<sup>g</sup> R. Messi,<sup>i</sup> L. Merola,<sup>e</sup>  
A. Michetti,<sup>h</sup> S. Miscetti,<sup>b</sup> S. Moccia,<sup>b</sup> F. Murtas,<sup>b</sup> M. Napolitano,<sup>e</sup> A. Nisati,<sup>h</sup>  
E. Pace,<sup>b</sup> G. F. Palamà,<sup>d</sup> M. Panareo,<sup>d</sup> L. Paoluzi,<sup>i</sup> A. Parri,<sup>b</sup> E. Pasqualucci,<sup>i</sup>  
M. Passaseo,<sup>h</sup> A. Passeri,<sup>k</sup> V. Patera,<sup>b</sup> F. Pelucchi,<sup>b</sup> E. Petrolo,<sup>h</sup> M. C. Petrucci,<sup>h</sup>  
M. Piccolo,<sup>b</sup> M. Pollack,<sup>l</sup> L. Pontecorvo,<sup>h</sup> M. Primavera,<sup>d</sup> F. Ruggieri,<sup>a</sup> P. Santantonio,<sup>b</sup>  
R. D. Schamberger,<sup>l</sup> A. Sciubba,<sup>h</sup> F. Scuri,<sup>m</sup> A. Smilzo,<sup>e</sup> S. Spagnolo,<sup>d</sup> E. Spiriti,<sup>k</sup>  
C. Stanescu,<sup>k</sup> L. Tortora,<sup>k</sup> P. M. Tuts,<sup>f</sup> E. Valente,<sup>h</sup> P. Valente,<sup>b</sup> G. Venanzoni,<sup>g</sup>  
S. Veneziano,<sup>h</sup> X. L. Wang,<sup>b</sup> S. Weseler,<sup>c</sup> R. Wieser,<sup>c</sup> S. Wölfle,<sup>b</sup> A. Zallo,<sup>b</sup>

(KLOE Collaboration)

<sup>a</sup> Dipartimento di Fisica dell'Università e Sezione INFN, Bari

<sup>b</sup> Laboratori Nazionali di Frascati dell'INFN, Frascati

<sup>c</sup> Institut für Experimentelle Kernphysik, Universität Karlsruhe

<sup>d</sup> Dipartimento di Fisica dell'Università e Sezione INFN, Lecce

<sup>e</sup> Dipartimento di Scienze Fisiche dell'Università e Sezione INFN, Napoli

<sup>f</sup> Physics Department, Columbia University, New York

<sup>g</sup> Dipartimento di Fisica dell'Università e Sezione INFN, Pisa

<sup>h</sup> Dipartimento di Fisica dell'Università e Sezione INFN, Roma I

<sup>i</sup> Dipartimento di Fisica dell'Università e Sezione INFN, Roma II

<sup>j</sup> Dipartimento di Fisica dell'Università di Roma III e Sezione INFN, Roma I

<sup>k</sup> Istituto Superiore di Sanità and Sezione INFN, ISS, Roma.

<sup>l</sup> Physics Department, State University of New York at Stony Brook.

<sup>m</sup> Dipartimento di Fisica dell'Università e Sezione INFN, Trieste/Udine

## ABSTRACT

The KLOE DAQ system manages a data throughput of 50 Mbytes/s. Its architecture is described in the article and new results of tests of some components are presented.

## 1 — INTRODUCTION

The major aim of the KLOE experiment at DAΦNE (Frascati) is to perform CP violation studies at sensitivities of  $\mathcal{O}(10^{-4})$ .<sup>[1-3]</sup> The KLOE data output of  $\mathcal{O}(10^{11}$  events/year) must be handled by its data acquisition system, DAQ, maintaining biases to values smaller than the experimental sensitivity. The maximum expected data rate from the KLOE detector, at full DAΦNE luminosity, has been estimated as  $10^4$  events per second of size of 5 kbytes each in average, corresponding to a total bandwidth of 50 Mbytes/s. The major components of the KLOE DAQ system are briefly presented in the following.<sup>[4]</sup> An overall view is given in fig. 1.

## 2 — ARCHITECTURE

Data comes from  $\sim 25,000$  Front End Electronics, FEE, channels housed in some 40 9U-VME crates. Signal conditioning and digitization is performed in a fixed time of  $\mathcal{O}(2 \mu\text{s})$ , to avoid biases depending on event configuration. Every FEE channel contains buffers of appropriate depth, in order to eliminate data overflows and to allow asynchronous read-out.

### 2.1 Fast Data Read Out

Data from the FEE are transferred to an on-line farm of Single Board Computers, SBC, using a two level concentration scheme. The first one is performed at crate level via a custom bus in the backplane, the AUXbus, and a hardwired read-out controller, ROCK, located in the crate itself. The ROCK implements the function of a sparse readout scanner collecting data related to each single trigger. The second level of concentration is performed by a ROCK manager, ROCKM, connected to chains of crates of suitable length with a cable bus, Cbus. Each ROCKM resides in a 6U-VME crate together with a VME processor which prepares sub-events for transmission to a given farm element. A commercial bus, VIC, connects all the crates in a chain allowing the VME processor to program, check and debug the FEE electronics.

The components of the DAQ system are interconnected via Ethernet for low bandwidth operations (controls, downloading, monitoring) and via FDDI for data transmission. A DEC FDDI GIGAswitch, with bridge functionality, is used to provide parallel paths between the VME processors and the farm in a scalable way. The number of switch ports dedicated to chains is chosen taking into account two factors: the maximum acceptable read-out ROCKM time and the throughput of the communication protocol achievable at VME and farm level. In order to improve the performance of the communication protocol, the sub-events related to the same group of consecutive trigger numbers are packed in sub-event-strings that must be gathered by a single SBC.

### 2.2 Data Flow Control and Event Building

The farm SBC's build and test the integrity of each event, implement the final events formatting, and perform quality control on samples of the data. The address of each farm element is assigned by an additional VME processor, the data flow controller, DFC, connected also via VIC channels to the ROCKM crates. The DFC manages the load of all the VME processors, maintaining a table which maps groups of trigger numbers and SBC addresses. DAQ resets and buffer flush-out commands are generated when misalignment is detected at farm level. Other error conditions will be similarly handled.

The farm is based on SBC's organized in crates. Each crate has a dedicated output SBC which manages the crate I/O to the storage devices. The total CPU power required for the whole

farm is estimated to be about 16,000 Specint'92. CPU boards adequate for this are beginning to appear in the market. We wait for a final decision upon the outcome of a joint project between INFN and DEC designing a custom SBC using the DEC Alpha chip.

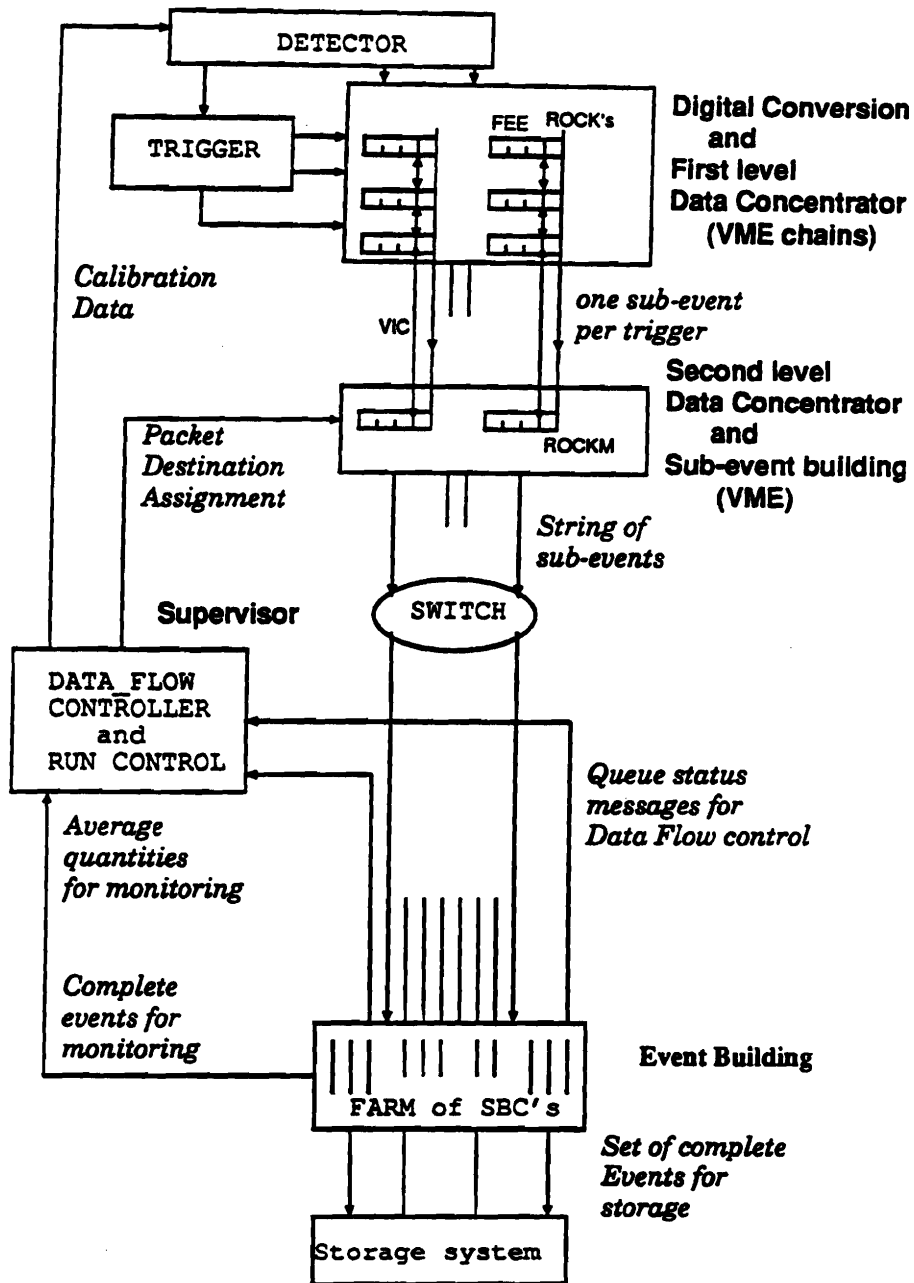


Fig. 1. System Architecture and Information Flow.

### 3 — SOFTWARE AND STORAGE

The on-line software represents a challenge because beyond the first level of data concentrators, the software must maintain the event synchronization. In a 10 kHz rate DAQ system, the latency time for message transmission is most important. In 1 ms there are 10 events accumulating in the buffer queues. Message losses and consequent retransmissions have to be considered as failures. The general software architecture is based on UNIX concepts implement-

ed on non-homogeneous hardware platforms. Real-time operating system will run in each board where diskless operation is needed. TCP/IP is used as underlying transport protocol for data transmission and message passing, while SNMP is used for network monitoring. Furthermore, DFC uses SNMP to build its tables.

The amount of data collected in a year of running is of the order of 500 Tbytes. While the on-line system requires only tape loaders, the off-line analyses require a compatible robotic system able to manage this huge quantity of data. A database is needed, which will contain all relevant information regarding the runs and the events, such as calibration constants, trigger condition, detector configuration, collider status and performance, fill numbers, run numbers, etc. Part of this information comes also from the on-line farm in dedicated banks such as the Run Header or the Event Header, to simplify trace back and raw data tape searches. No "event directory" is used for the raw KLOE data, but the chosen database will assure the compatibility with the file system used in the tape robots. The most promising tape system, in terms of performance/cost, appears to be the DEC DLTs.

#### 4 — CURRENT IMPLEMENTATION

##### 4.1 *ROCK and ROCKM*

The ROCKs and ROCKMs implementation is underway. The first ROCK prototype is under test. Results of a complete simulation were presented at CHEP94,<sup>[5]</sup> showing that the KLOE DAQ system, configured with chains of 8 crates, the maximum length of the Cbus chain, for the electromagnetic calorimeter (480 ADC or TDC per crate) and with chains of 4 crates for the tracking chamber (1536 TDC per crate), is able to sustain up to a 15 Mbytes/sec data transmission rate per chain, at an event rate of  $10^4/s$ .

##### 4.2 *TCP/IP protocol on FDDI*

The TCP/IP protocol performance has been studied on FDDI using different hardware and software platforms connected to the FDDI GIGAswitch. Some results related to VME CPU boards (CES/FIC8234 and HP742rt) and workstations are presented in the following table. Above a certain level of CPU power, see entries 1 and 2 in the table, optimization of the code implementing the TCP/IP protocol,<sup>[6]</sup> entry 4-5 vs 3, is very important.

	Hardware	Operating System	TCP/IP Throughput Mbytes/s	CPU Power Dhrystones
1	CES/FIC8234 Rockwell FDDI	LynxOS	1.2	24 k
2	HP 742rt Rockwell FDDI	HP-RT (LynxOS)	4.5	90 k
3	HP9000/735 EISA FDDI	HP-UX	5.0	280 k
4	DEC 4000/610 FBUS FDDI	OSF/1	11.0	250 k
5	DEC3000/800 TC FDDI	OpenVMS+UCX	11.0	270 k

##### 4.3 *Managing Multiple TCP/IP Connections*

We have studied different mechanisms for maintaining multiple concurrent TCP/IP connec-

tions between DAQ components. This is relevant when sending sub-events from the ROCKM's to the SBC's. Both the standard UNIX I/O multiplexing method of the "select" call, and the POSIX multithreading mechanism allow to maintain up to 100 different connections on the same processor, without lowering throughput even when small TCP/IP buffers, 8192 bytes, are used. We have tested the performance of multiple connections using ten senders on five different computers, DEC 3000/600 and HP9000/735, simulating the VME-CPU's. Strings of sub-events, of ~50 kbytes each, are sent, to one DEC 4000/610 (95 Specint'92), which orders single events and byteswaps. The throughput measured together with the real and CPU time required to handle 10,000 events for different actions in the receiver are given in the table below.

Action	Throughput (MB/s)	Real time (10000 events)	CPU time
receiving	11	4.3s	2.6s
receiving + ordering	10	4.9s	3.3s
receiving + ordering + byte/word swapping	4.5	10.5s	9.0s

## 5 — OUTLOOK

The new generation of FDDI interfaces implement the TCP/IP protocol on-board. Also VME CPUs with FDDI interface on board or on a PCI mezzanine card are becoming available. We plan to test soon the following boards: HP 743rt, AXPvme 160, CETIA Power PC, Motorola PowerPC.

We are confident that VME processors will communicate through FDDI channels at a speed greater than 5 Mbytes/s. To achieve the required throughput of 50 Mbytes/s, the KLOE DAQ therefore needs at most 10 VME chains connected to the switch.

## REFERENCES

1. *A GENERAL PURPOSE DETECTOR for DAΦNE*, The KLOE Collaboration, LNF-92/019 (1992).
2. *THE KLOE DETECTOR, Technical Proposal*, The KLOE Collaboration, LNF-93/002 (1993).
3. *THE KLOE CENTRAL DRIFT CHAMBER, Addendum to the Technical Proposal*, The KLOE Collaboration, LNF-94/028 (1994).
4. *THE KLOE DATA ACQUISITION SYSTEM, Addendum to the Technical Proposal*, The KLOE Collaboration, LNF-94 (1994).
5. CUSTOM SOLUTION FOR A DATA READOUT ARCHITECTURE: A SYSTEM LEVEL SIMULATION, A. Aloisio et al., Proceedings of CHEP94.
6. HIGH PERFORMANCE TCP/IP FOR OSF/1 ALPHA AXP WORKSTATIONS (White Paper), Digital Equipment Corporation Networks Engineering TCP/IP Program Office, Littleton, MA, March 1993





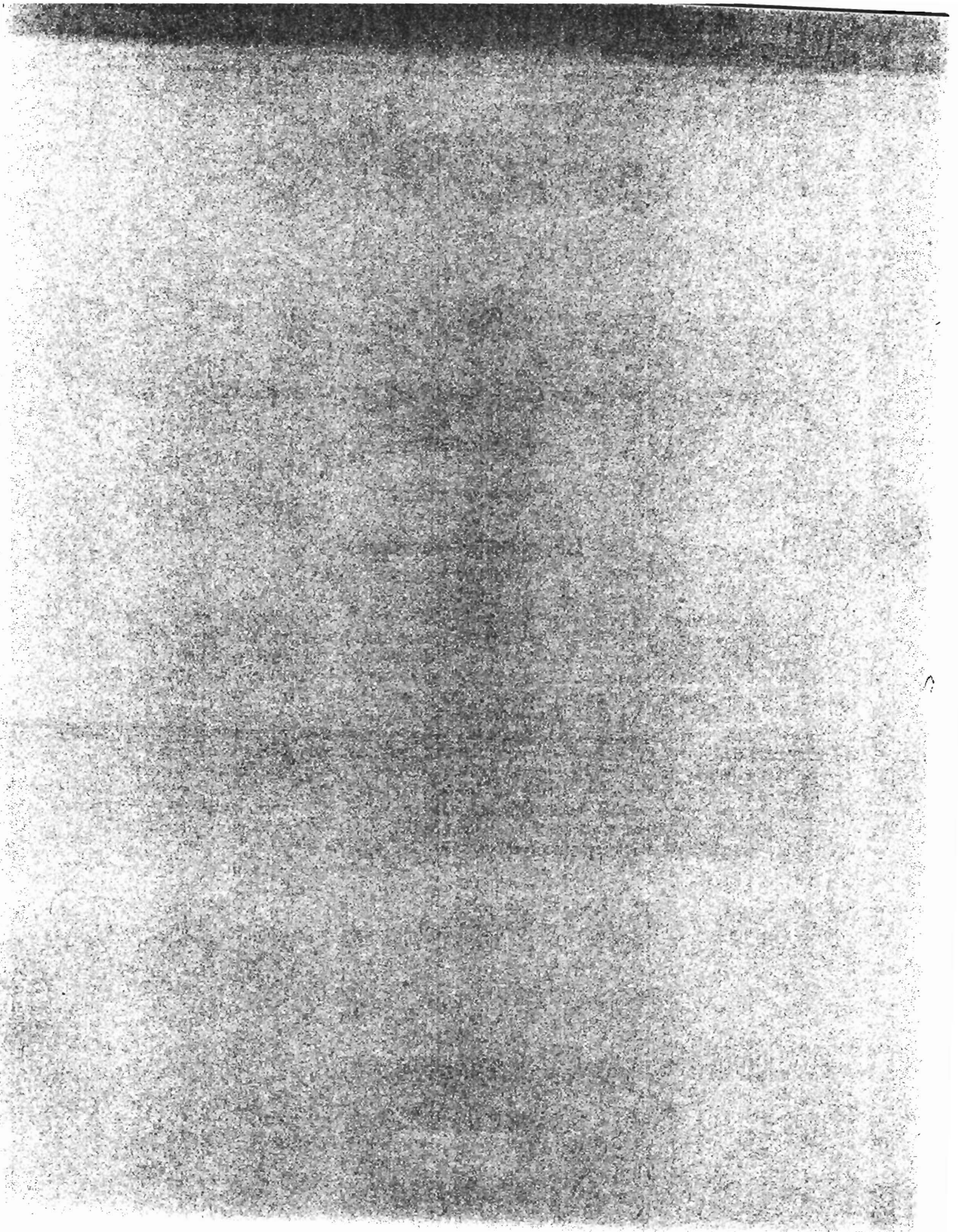
## **A Continuous Time Stamping Time Digitizer Architecture for HEP**

### **Applications**

**Mark Gorbics**

**LeCroy Corporation**

This new integrated circuit design combines the excellent time resolution of the LeCroy MTD133 monolithic TDC with a very flexible readout architecture. The design incorporates three distinctly different readout modes. In the continuous mode the measured times are readout as they occur, with a maximum average hit rate of 20 MHz for the entire chip. The data read out is the absolute time, beginning when the system was last reset. In the triggered mode, the data is stored internally until it either becomes too old, or a trigger is received. At that time a block of data corresponding to a particular time interval relative to the trigger is readout. The data read out is the relative time between the event and the trigger. The maximum time (after which the data is declared to be too old to keep) can be set to correspond to the maximum trigger delay time. Both the continuous and triggered modes allow deadtimeless operation, the inputs are always live and recording hits as they arrive. The third mode of operation corresponds to the start-stop mode of the current MTD133 TDC. The design of this integrated circuit is in progress, with first silicon expected in 1995.



## **SCI in Data Acquisition Systems**

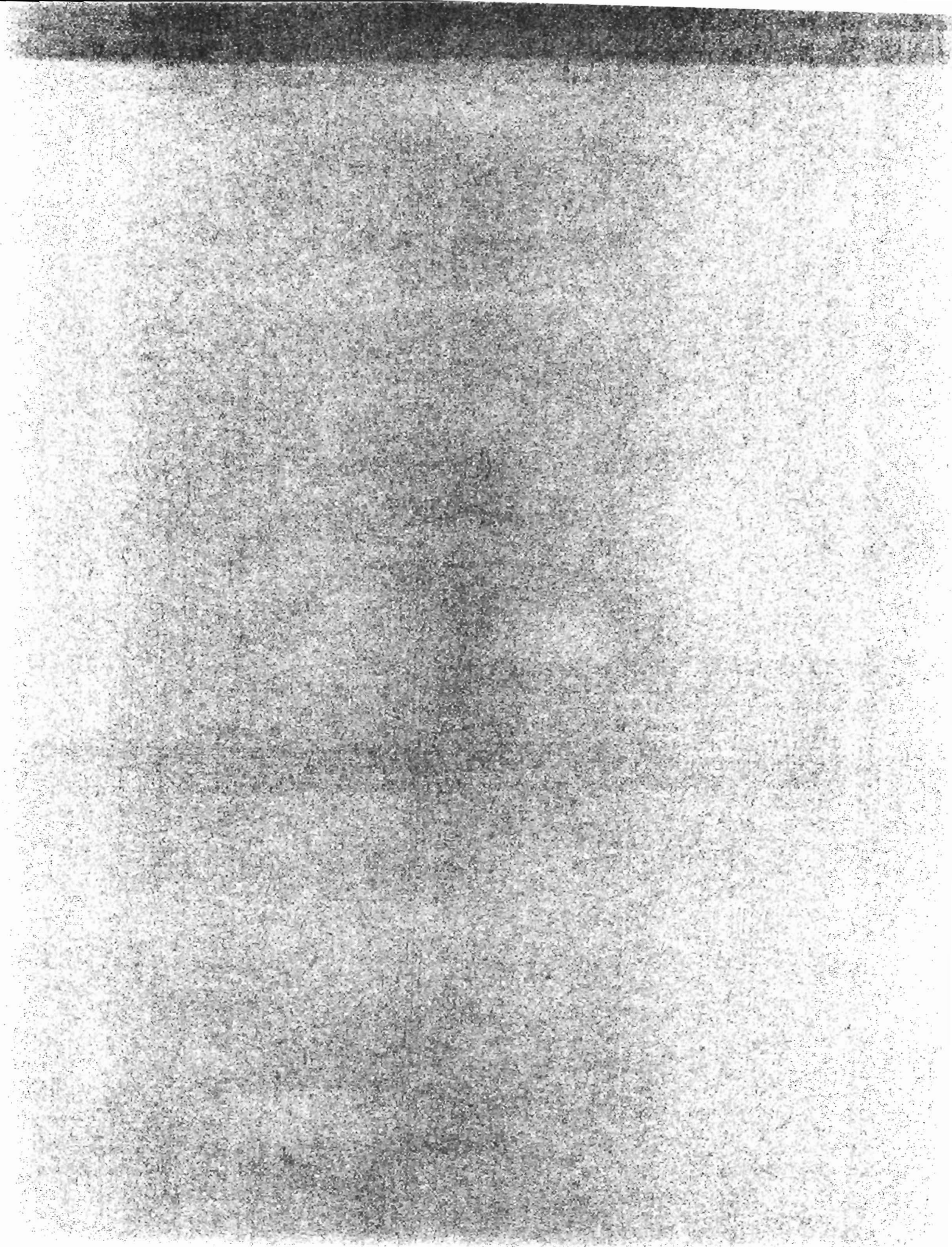
**Bernard Skaali**

**University of Oslo**

**The SCI activity at the Department of Physics covers:**

- I) Design and construction of SCI based instrumentation for use in data acquisition systems, and**
- II) Modeling and simulation of large SCI DAQ systems and SCI interface boards.**

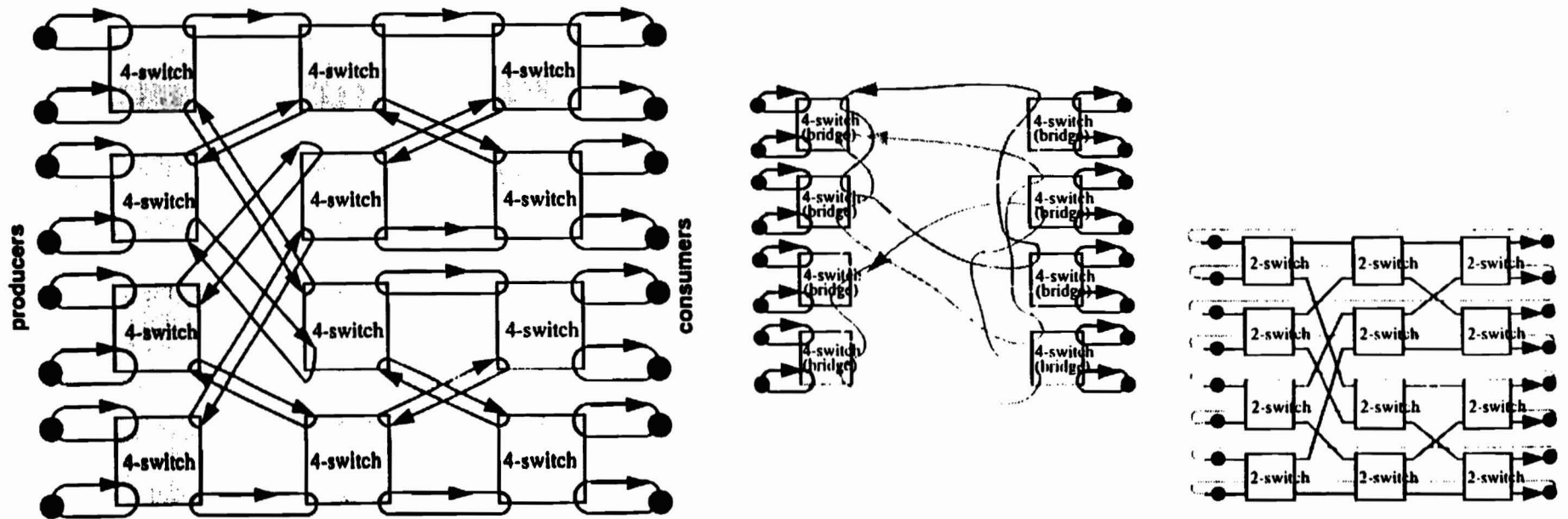
**Much of the activity is part of the CERN RD24 project, which investigates applications of SCI in DAQ systems for the new LHC accelerator. At the Department of Physics we have an SCI ring with Sun stations and locally developed SCI modules. A specially developed SCI Tracer ("LinkScope" from Dolphin Interconnect Solutions) provides high level debugging facilities for SCI link traffic.**



# SCI in DAQ Systems

B. Skaali and collaborators at the  
Department of Physics, University of Oslo, Norway

Email: [t.b.skaali@fys.uio.no](mailto:t.b.skaali@fys.uio.no)



$8_R \times 8_R$  DAQ multistage systems, various topologies

## **SCI activities at the Department of Physics, University of Oslo, Norway:**

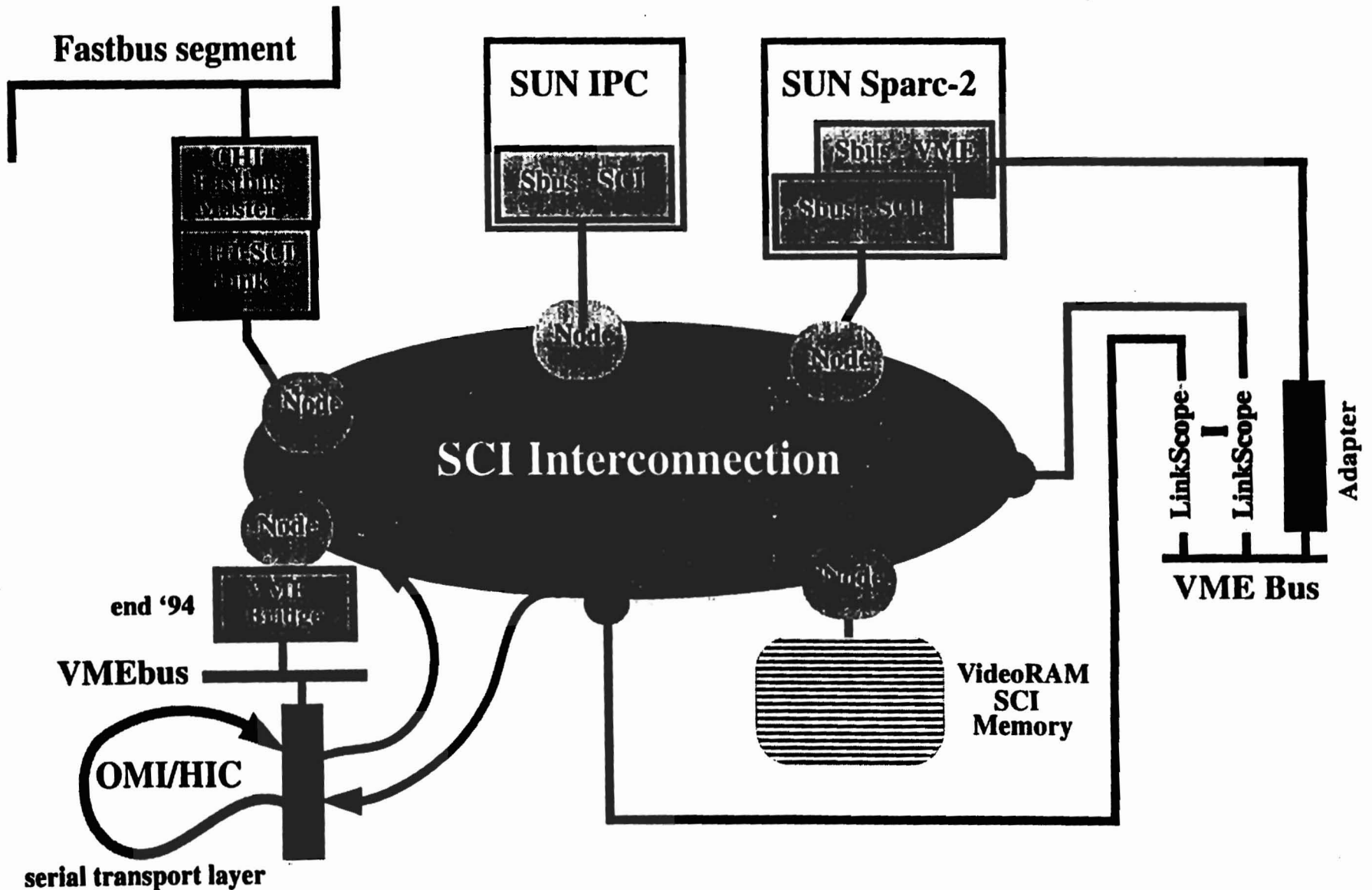
- Directed towards data acquisition (DAQ), in particular (very) large scale data acquisition systems at the planned LHC accelerator at CERN
- Participates in the CERN RD24 Research & Development project.
- Development of instrumentation and hardware modules for SCI in DAQ.
- Development of diagnostics tools for SCI.

## **Simulation programme for SCI**

- Modelling and simulation of various topologies for large SCI-based DAQ systems.
- Simulation of data flow in proposed DAQ-systems for LHC experiments (ATLAS, ALICE).
- Simulation studies of SCI switch concepts.
- Tools: MODSIM II programming language, CERN's SCILab package.
- Contact: Bin Wu, Dep. of Physics, Univ. of Oslo.  
Email: [bin.wu@fys.uio.no](mailto:bin.wu@fys.uio.no)



# SCI configuration - Dep. of Physics, Univ. Oslo



# SCI diagnostics tool - the LinkScope

- The LinkScope™ SCI Tracer project:  
collaboration Dolphin ICS - Univ. of Oslo
- LinkScope H/W: a single width VME module, for the CMOS SCI NodeChip, 200 MB/s.
- Snoops on an SCI link and captures and stores sequences of SCI packets according to a pre-defined set of trigger and acquisition criteria.
- Trigger/acquisition program is written in a high-level Tracer Control Language.

# LinkScope™ Supervisor Program

LinkScope

**Dolphin**  
INTERCONNECT SOLUTIONS

Welcome to LinkScope Supervisor

PROGRAM MONITOR ANALYZE CONFIG HELP

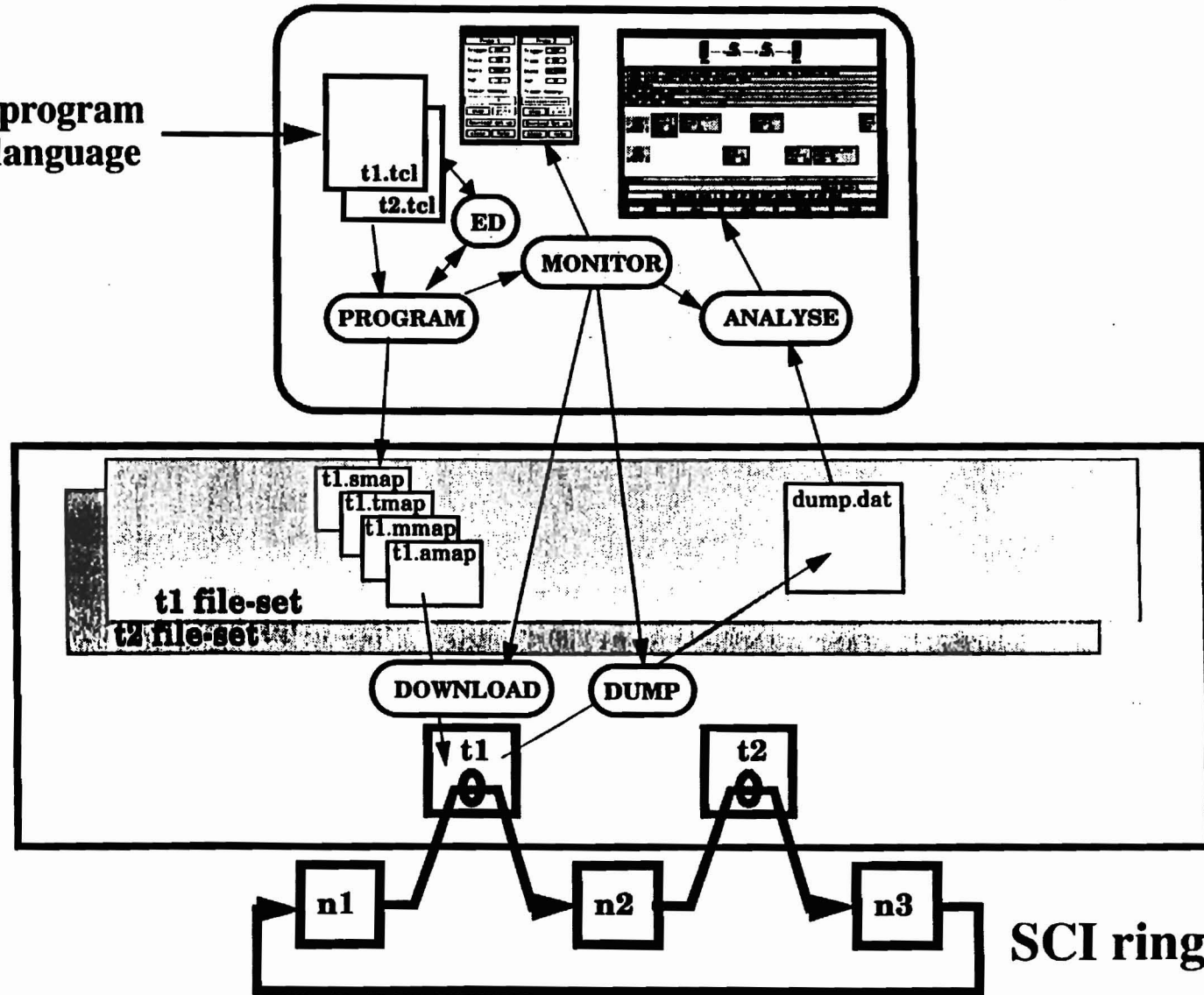
```

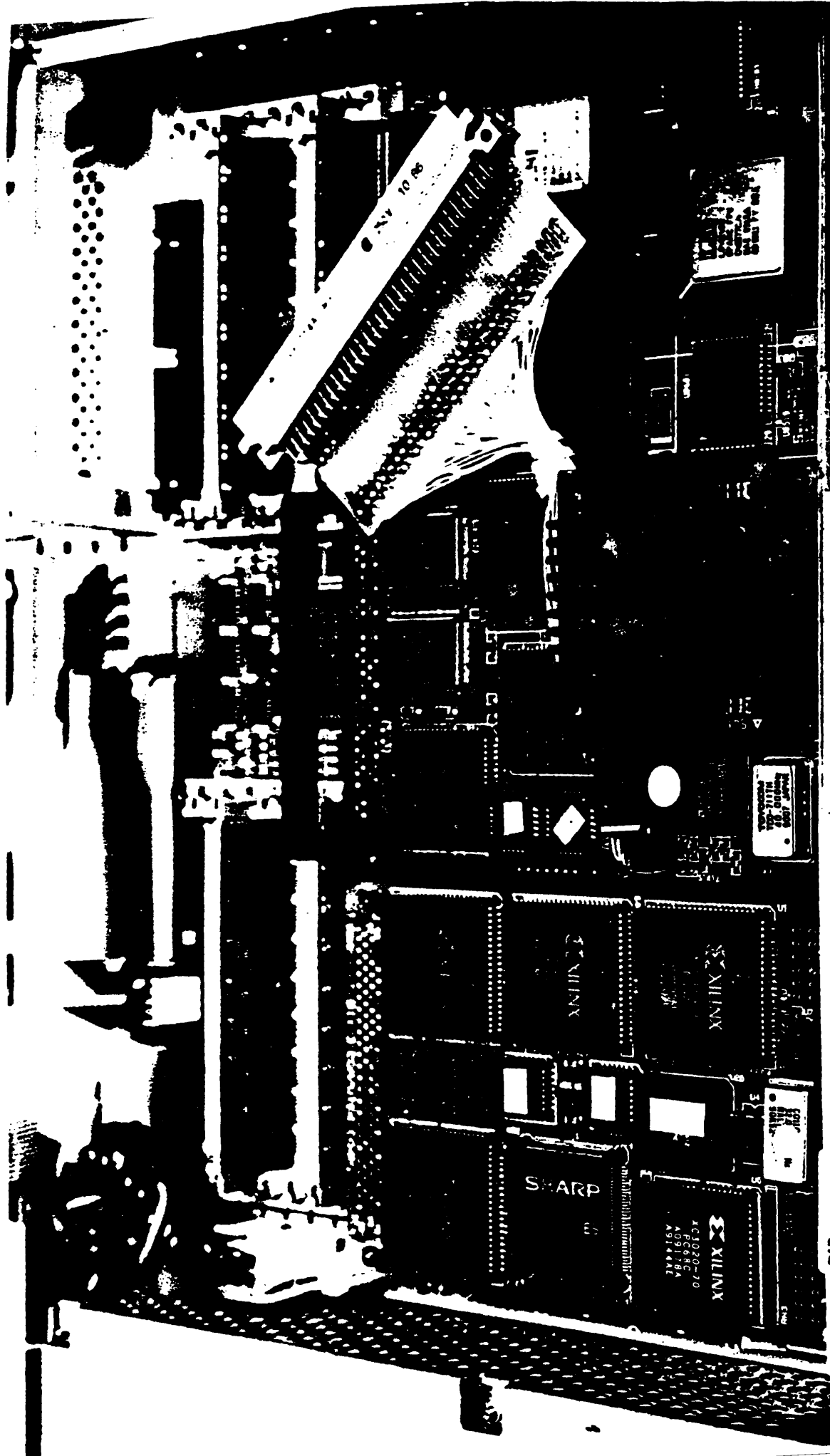
    graph LR
      N1[255] -- tracer 1 --> N2[255]
      N2 -- tracer 2 --> N3[255]
  
```

ECHO	RESPONSE	CREADOO
PKT#1	PKT#1	PKT#1
PKT#2	PKT#2	PKT#2
PKT#3	PKT#3	PKT#3
PKT#4	PKT#4	PKT#4
PKT#5	PKT#5	PKT#5
PKT#6	PKT#6	PKT#6
PKT#7	PKT#7	PKT#7
PKT#8	PKT#8	PKT#8
PKT#9	PKT#9	PKT#9
PKT#10	PKT#10	PKT#10
PKT#11	PKT#11	PKT#11
PKT#12	PKT#12	PKT#12
PKT#13	PKT#13	PKT#13
PKT#14	PKT#14	PKT#14
PKT#15	PKT#15	PKT#15
PKT#16	PKT#16	PKT#16
PKT#17	PKT#17	PKT#17
PKT#18	PKT#18	PKT#18
PKT#19	PKT#19	PKT#19
PKT#20	PKT#20	PKT#20
PKT#21	PKT#21	PKT#21
PKT#22	PKT#22	PKT#22
PKT#23	PKT#23	PKT#23
PKT#24	PKT#24	PKT#24
PKT#25	PKT#25	PKT#25
PKT#26	PKT#26	PKT#26
PKT#27	PKT#27	PKT#27
PKT#28	PKT#28	PKT#28
PKT#29	PKT#29	PKT#29
PKT#30	PKT#30	PKT#30
PKT#31	PKT#31	PKT#31
PKT#32	PKT#32	PKT#32
PKT#33	PKT#33	PKT#33
PKT#34	PKT#34	PKT#34
PKT#35	PKT#35	PKT#35
PKT#36	PKT#36	PKT#36
PKT#37	PKT#37	PKT#37
PKT#38	PKT#38	PKT#38
PKT#39	PKT#39	PKT#39
PKT#40	PKT#40	PKT#40
PKT#41	PKT#41	PKT#41
PKT#42	PKT#42	PKT#42
PKT#43	PKT#43	PKT#43
PKT#44	PKT#44	PKT#44
PKT#45	PKT#45	PKT#45
PKT#46	PKT#46	PKT#46
PKT#47	PKT#47	PKT#47
PKT#48	PKT#48	PKT#48
PKT#49	PKT#49	PKT#49
PKT#50	PKT#50	PKT#50
PKT#51	PKT#51	PKT#51
PKT#52	PKT#52	PKT#52
PKT#53	PKT#53	PKT#53
PKT#54	PKT#54	PKT#54
PKT#55	PKT#55	PKT#55
PKT#56	PKT#56	PKT#56
PKT#57	PKT#57	PKT#57
PKT#58	PKT#58	PKT#58
PKT#59	PKT#59	PKT#59
PKT#60	PKT#60	PKT#60
PKT#61	PKT#61	PKT#61
PKT#62	PKT#62	PKT#62
PKT#63	PKT#63	PKT#63
PKT#64	PKT#64	PKT#64
PKT#65	PKT#65	PKT#65
PKT#66	PKT#66	PKT#66
PKT#67	PKT#67	PKT#67
PKT#68	PKT#68	PKT#68
PKT#69	PKT#69	PKT#69
PKT#70	PKT#70	PKT#70
PKT#71	PKT#71	PKT#71
PKT#72	PKT#72	PKT#72
PKT#73	PKT#73	PKT#73
PKT#74	PKT#74	PKT#74
PKT#75	PKT#75	PKT#75
PKT#76	PKT#76	PKT#76
PKT#77	PKT#77	PKT#77
PKT#78	PKT#78	PKT#78
PKT#79	PKT#79	PKT#79
PKT#80	PKT#80	PKT#80
PKT#81	PKT#81	PKT#81
PKT#82	PKT#82	PKT#82
PKT#83	PKT#83	PKT#83
PKT#84	PKT#84	PKT#84
PKT#85	PKT#85	PKT#85
PKT#86	PKT#86	PKT#86
PKT#87	PKT#87	PKT#87
PKT#88	PKT#88	PKT#88
PKT#89	PKT#89	PKT#89
PKT#90	PKT#90	PKT#90
PKT#91	PKT#91	PKT#91
PKT#92	PKT#92	PKT#92
PKT#93	PKT#93	PKT#93
PKT#94	PKT#94	PKT#94
PKT#95	PKT#95	PKT#95
PKT#96	PKT#96	PKT#96
PKT#97	PKT#97	PKT#97
PKT#98	PKT#98	PKT#98
PKT#99	PKT#99	PKT#99
PKT#100	PKT#100	PKT#100

# LinkScope™ tracer system

Tracing program  
in TCL language





NY 10 78

Sbus - SCI

SHARP

XILINX

XILINX

XILINX  
XC020-70  
A09178A  
A91444E

# LinkScope™ modules

STATUS  
RUN  
SCON

RMT RST  
ABORT  
RESET

IBT 31  
DIGITAL CORPORATION  
REMOTE  
LOCAL  
READY

RESET MR  
TRON OFF  
TRACE DISPL  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0

RESET MR  
TRON OFF  
TRACE DISPL  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0

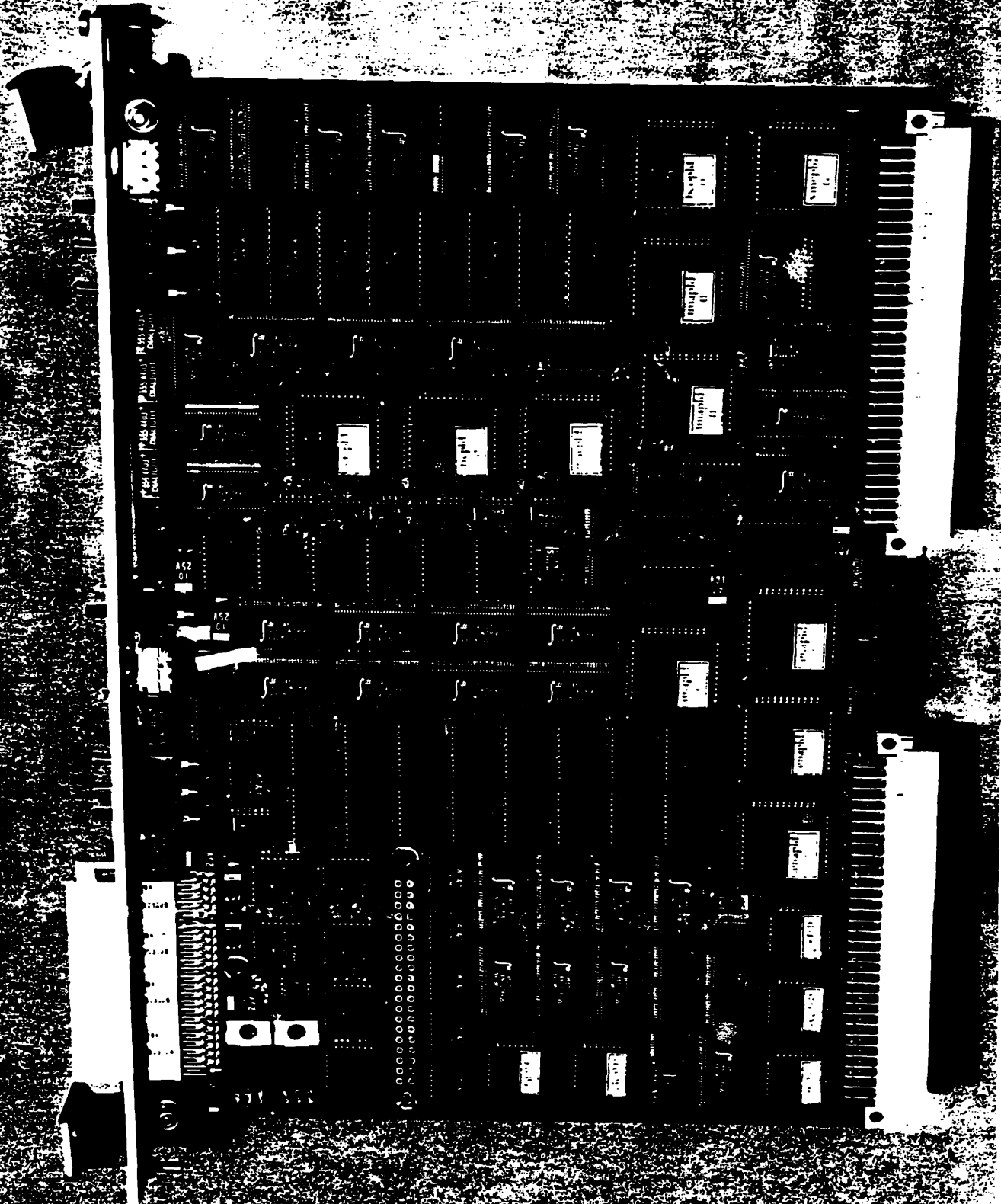
T.IN TRIGG  
T.IN  
OUT SEL  
TOUT  
STATE  
CHD CHD  
CHI CHI  
RES ERR  
DERR FERR  
CERR ACT IN

T.IN TRIGG  
T.IN  
OUT SEL  
TOUT  
STATE  
CHD CHD  
CHI CHI  
RES ERR  
DERR FERR  
CERR ACT IN

AMEBUS SIGNALS  
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31  
A16 016  
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31  
20 21 22 23 24 25 26 27 28 29 30 31  
424  
25 26 27 28 29 30 31  
AMO 1 2 3 4 5  
READ LWD OSO DACK BERR BBSY ACFL SCLK STBY STOP MON DIR LAT  
BRO 1 2 3  
DGO 1 2 3  
AS BCLR SFAIL SRES TROT 2 3 4 5 6 7  
DISPLAY MODE  
AND 1 2 3 4 5  
ENAM  
DGO 1 2 3  
ENGO  
STOP DEBR  
SEL DISPLAY MODE INT. RESET  
EXT TRIG.

AA

C



**LinkScope™ card**

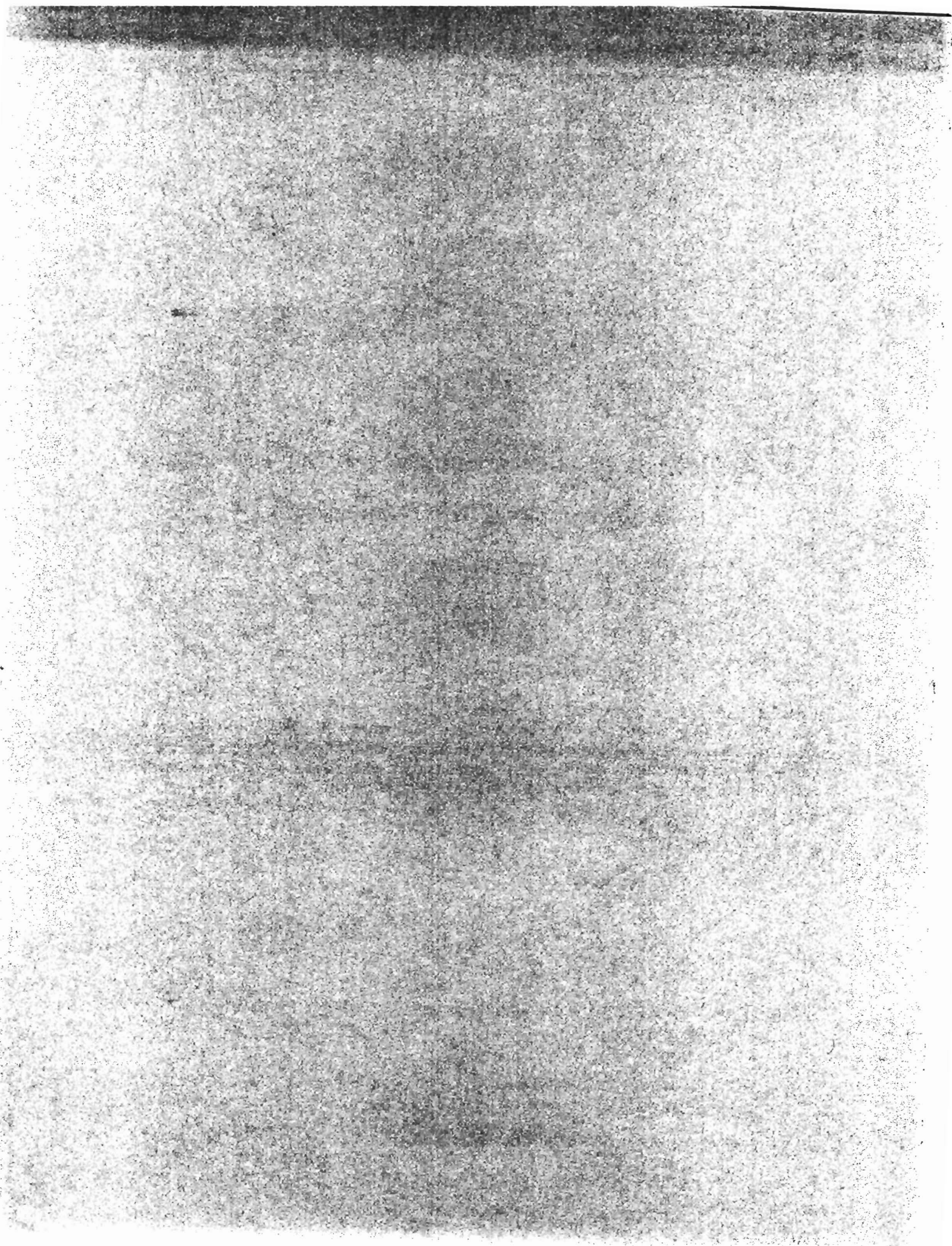
**The 3D-Flow System as Programmable Switch for Moving and  
Reducing Data in DAQ Applications**

**Dario Crosetto**

**SSCL**

Better described as an architecture rather than merely an ASIC, the 3D-Flow allows to move data from multiple sources to one or to multiple destinations in a programmable form. The system allows quick and flexible acquisition, exchange and data reduction in a balanced manner using regular connections and repeated components. The 3D-Flow system is scalable to permit solutions to many different types and sizes of applications.





# 3D-Flow as a programmable system for moving and reducing data in DAQ applications

Dario Crosetto

## Abstract

Really a system architecture and not just an ASIC, the 3D-Flow design facilitates acceptance of data from multiple sources, reducing it and sending it to one or more destinations, all in a programmable sequence. These operations are carried out in a balanced manner using regular connections and exceptionally few replicated components. The 3D-Flow system is scalable to permit adaptation to many different types and sizes of applications.

## I. INTRODUCTION

Given the well known problems to be solved by a trigger and data acquisition system for a large (or small) experiment, this report will describe how these problems can be solved with the 3D-Flow architecture [1] approach (system, logic, mechanics, cabling, etc.).

The 3D-Flow is an architecture built around a 16-bit ASIC processor that combines multiple execution units (MAC/DIV, two ALU's, two comparators, and event counter, an encoder and three shifter), four internal buses, six communication channels (North, East, West, South, Top and Bottom), and three memory banks (Data Memory 1, Data Memory 2, and Program Memory), designed to meet high performance real-time objective at a reasonable cost. Operation modes of the processor are determined by two external input mode pins (MIMD/SIMD and SYNC/Data Driven). The operation mode SIMD causes the processor to accept as its next instruction two 48-bit words through a single 48-bit input port valid for all four processors on the chip. In MIMD mode, each processor executes the instruction sequence in its own 64-words 96-bit wide program memory. SYNC mode implies that instruction execution proceeds with each clock pulse while Data-Driven implies that an instruction is executed only when all its inputs are satisfied. The combination of SIMD and Data Driven is meaningless.

The results reported here are derived from the design and simulation of an actual system (crates, boards, cabinets, cooling system, etc.) as well as from a chip design at the gate level using a CMOS gate array (0.5 micron technology, 3.3 volt). The gate array approach is among the most conservative, cost-effective, and reliable approaches, but it does not give the best technological performance. The actual

netlist of the design of the 3D-Flow at 40 MHz is suitable for today's experiments, but the reader should know that without necessarily using GaAs technology 10 years from now, when the LHC will be operating, it will be possible to have the same 3D-Flow chip in CMOS at 200 MHz.

In its original conception, the 3D-Flow system was designed to fulfill both triggering and data acquisition requirements. In the present article only the DAQ applications will be described, while the triggering capabilities are described elsewhere. [1].

## II. DATA ACQUISITION SYSTEM REQUIREMENTS

Requirements for a data acquisition system are typically the following [ref. 2-9]:

A data acquisition system (DAQ) collects the data from the detector front-end electronics when an event is selected by the trigger system, and sends these data to an on-line farm of computers.

Signals from the detector follow two different paths:

- Some of them from a few subdetectors, usually digitized at lower resolution, are sent to the fast trigger electronics, which takes the first decision to select events.
- All valid non-zero signals from all subdetectors are instead converted into digital form at higher resolution, and are stored on mass storage devices, after full event reconstruction and acceptance.
- For the duration of the decision time of the level-1 trigger (from 2 to 4 microseconds), all data (for low- and medium-occupancy subdetectors and only the valid data with their respective addresses for the very-low-occupancy subdetectors) must be stored in memory
- After level-1 trigger decisions, only the data of the accepted events must be sent to the farm of computers on which level-2 and level-3 trigger and selection will be executed.
- The fragmented data from different subdetectors must be synchronized, collected into coherent events, compressed and sent to the selection stage (typically level 3), where

the whole event data is analyzed to perform the final selection processes.

While the tasks performed on the data in a period longer than 2 to 4 microseconds can make use of standard processors, the front-end electronics and level-1 trigger, storing data for the first few microseconds and collecting the fragmented data to build an event will require a specialized design (using gate arrays and VLSI design, for example), and a particular architecture adapted to this application. This solution should also provide scalability, modularity, low cost, and high-speed performance.

The use of the basic features of the 3D-Flow processor in the 3D-Flow architecture is now described in relation to satisfying the above requirements in the different tasks. The starting point was the feasibility and simplicity of building the hardware at low cost, with the lowest number of required components, while providing a programmable solution.

In particular, great attention was paid to the high connectivity and high speed required by the application that should have a modular and scaleable assembly.

Figure 1 shows the main components of a typical trigger and data acquisition system. The approach of not considering the level-2 trigger as a separate component is not new to this scheme. In fact, the GEM experiment at SSCL had already integrated the level-2 trigger scheme into the hardware of the level-3 trigger, as shown in the technical design report [3]. Since computer technology is advancing rapidly, one is able to minimize the number of different hardware systems and to exploit hardware performance and low cost to distribute simpler tasks for fast decisions and more complex tasks for more sophisticated decisions.

### III. PROPOSED USE OF THE 3D-FLOW SYSTEM FOR DAQ

The right side of the figure 1 shows the path of partial data (typically from a calorimeter and/or muon subdetector) digitized at lower resolution and sent to the trigger system. The handling of the event data is also represented

schematically, and two possible ways of handling the inputs from the detector are also indicated. For high-to-medium occupancy detectors, the first buffer operates in a synchronous mode, and it records, for each event, the whole data information from a fixed number of input channels. When dealing with very-low-occupancy detectors instead, it is possible in principle to perform zero-suppression and address encoding "on the fly," as accomplished by the first buffer operating in asynchronous mode. These two mode of operation are described in more detail in the following. It is important to realize nevertheless how intrinsic flexibility and programmability of the 3D-Flow system allow to choose the appropriate mode of data handling according to the requirements of any specific experiment and/or detector.

#### A. Implementation of the synchronous first-stage buffer with 3D-Flow

The synchronous (to the bunch crossing) first-stage buffer can be implemented with the 3D-Flow processor by using its internal "data memory" and by writing a short, four-line program loop, as described in Table 1, to handle the "read and write pointers."

At each bunch crossing, new data from the detector is written to the "Top" port of the 3D-Flow processor (The fixed number of data, in a fixed sequence, that are transmitted synchronously with the bunch crossing, allow to identify each channel without the need of transmitting its address). The accept/reject information arriving from the trigger system is sent to the 3D-Flow "North" port. Line 2 of the program in Table 1 shows that data from the "Top" port is stored into data memory (DM), and data from the "North" port is stored in accumulator A1 while pointers are also incremented. On the next cycle the zero flag of accumulator A1 is tested. If the data from the "North" port (trigger "Accept") was not zero, then the data value that was recorded "x" cycles before will be sent out (the offset from write-to-read data is programmable by the user); if the data from the "North" port was zero, then the next data will be fetched without reading.

Table 1. 3D-Flow assembler program for the synchronous first-stage buffer.

Line 0	START: r1=const1, CLR_A2	Load read pointer offset to write pointer (L-1=trigger latency)
Line 1	r2=const2, ST_A3_r1	Initialize read pointer and load pointer increments
Line 2 Step 1	LOOP: DM=T, ST_A1_N, ADDU_A2+r2, ADDU_A3+r2	Get DAQ value from Top, get trigger Y/N from North port, increment read & write pointers
Line 3 Step 2	BRccCLR #1 LOOP, DMP=A2lo r13=A3lo	If L-1 Trigger "ACCEPT", go to next line, ELSE fetch next DAQ & TRIG. values.
Line 4 Step 3	DMP=r13	Initialize read pointer
Line 5 Step 4	B= DM, BRA LOOP	Send DAQ value to Bottom port and fetch next DAQ & Trig

It should be obvious how such a straightforward and generic procedure could be tailored to optimize data throughput performance for applications ranging, e.g., from 1 MHz to 40 MHz bunch-crossing and with 100 Kbyte or 5 Mbyte event sizes. It should be also stressed again how the netlist available today for a 3D-Flow at 40 MHz, can easily be scaled to 200 MHz a few years from now, improving the performance without necessarily changing the architecture.

As an example, let us see how the different use (programming, partitioning of the 3D-Flow internal data memory, size of the 3D-Flow synchronous first-stage buffer as described above, etc.) of the 3D-Flow chip and system architecture could give the greatest benefits to the user in price/performance in implementing the synchronous first-stage buffer.

In an application for an experiment with 40 MHz bunch crossing, about 1Mbyte data/event, the size of the required configuration will be determined by the speed of writing data into the data memory of the 3D-Flow. To make the overall system as economical as possible, one would like to write as many event data as possible per 3D-Flow processor into its data memory synchronous buffer. Realistically speaking, one cannot go behind writing 8 x 16-bit values in 25 ns (even assuming future technological improvements expected by the time the LHC should be operating). Thus, the partitioning of the 3D-Flow data memory will have the two data memory banks working in parallel with not more than 16 bytes for each event, and the data memory size is not required to be large.

In applications for experiments with 7-MHz bunch crossing, with the same event size, more data of the same event can be written at each bunch crossing, thus reducing the number of overall channels (or 3D-Flow processors) for the entire system. But even for this application the size of the 3D-Flow data memory required is not too large, i.e. a few Kbytes. In this application, instead of having partitioned the 3D-Flow data memory in two banks as before, one can concatenate the two memory banks to have a larger buffer.

The flexibility of the 3D-Flow architecture in the described first layer of processors, propagates directly into the second, asynchronous, layer, where a large number of input channels is funneled into a single 3D-Flow output chip (see Figure 2).

The overall consideration is that by using the 3D-Flow chip in the appropriate way to fit each application, one has the same advantages of programmability, flexibility, modularity, and short cable connection, thereby providing high-speed communication, throughout the entire DAQ system. Such advantages include all benefits of easier maintainability of a single component, board development system, etc., with the possibility of optimizing the cost for each application.

### *B. Implementation of the asynchronous first-stage buffer with 3D-Flow*

The asynchronous buffering mode at the first-stage, is exploited to store data coming from the very-low-occupancy

detectors, where for each datum it is also possible to encode the address.

To implement this buffer, more functionality of the 3D-Flow processor will be used. As described in [1] the 3D-Flow processor chip has two mode select pins: the first one sets operation as Single Instruction and Multiple Data; while the second selects operation in the data-driven or the synchronous mode. For the implementation of the asynchronous buffer, the 3D-Flow chip will operate in synchronous mode, and the program residing on each processor will do the polling among the input ports.

Each 3D-Flow processor is connected through the "West" and "East" ports to the neighboring processors to form a linear array. The 3D-Flow data memory will be organized in "banks." Data received from the "Top," "West," and "East" port with their respective address will be stored in the corresponding "bank." The "North" port of each processor is connected to the trigger accept/reject. In the case of a lot of interaction on a very-low-occupancy detector in a specific region, causing the generation of many hits in a small area, one 3D-Flow processor may run out of available "banks." In this case the program in each processor will forward the data to a neighboring processor with lower occupancy and with some free "banks."

When a specific trigger is received from the "North" port, the 3D-Flow processor will output data of the corresponding "bank." (See Figure 1.)

### *C. Second-stage DAQ buffer (asynchronous with channel reduction)*

The second buffer is also implemented with 3D-Flow processors. This makes better use of the high communication speed of the 3D-Flow. Data from the previous two first-stage buffers are received as input to this asynchronous second-stage buffer. In this stage, besides reducing the number of channels, the 3D-Flow functionality provides the physicist a tool to apply filters on the data, such as zero suppressing. As an example of the performance of the 3D-Flow architecture, the simulation of 4096 channels with fragmented event data for a partial event builder scheme is described in the next Section.

### *D. Simulation of a 4096-channel event builder scheme with 3D-Flow*

The evolution of event builders in recent years has been from a simple single-channel funneling to a computer, to a group of parallel channels (each with their own funneling and output speed limitation) sending data to a farm of computers. This change of scheme is due to the increase in the rate and size of accepted events, which has gone beyond what technology can offer in single-line speed transmission.

A 3D-Flow pyramid array was conceived to test the funneling of a large number of input channels to one 3D-Flow output chip. This scheme was then simulated for 4096 input channels or 3D-Flow input processors. A 3D-Flow system reflecting the real communication connections and assembly

requires one to consider that each 3D-Flow chip has four 3D-Flow processors and that the suggested assembly for the most efficient interconnectivity is a stack of matrices with a diminishing number of processors and boards in each successive layer.

The layers were defined as follow:

- Layer 0 = 4096 3D-Flow processors on 1024 3D-Flow chips assembled on 256 daughterboards.
- Layer 1 = 1024 3D-Flow processors on 256 chips assembled on 256 daughterboards (one chip per board in order to keep vertical connection simple in stacking the boards).
- Layer 2 = 256 3D-Flow processors on 64 chips assembled on 64 boards (longer cables between boards).
- Layer 3 = 64 3D-Flow processors on 16 chips assembled on 16 boards (longer cables between boards).
- Layer 4 = 16 3D-Flow processors on 4 chips assembled on 4 boards (longer cables between boards).

The routing table has been generated to interconnect the 3D-Flow pyramid with the nearest neighbor in all six directions. Each simulation program was executed in each 3D-Flow processor mode (MIMD and Data-Driven) of the simulator according to the functionality of the netlist ASIC of the 3D-Flow chip. The 96-bit instruction words of the programs written can be added as test vectors at the production time of the chip.

In summary, the 3D-Flow system for this DAQ application with 4096 channels (the array may be bigger) that can be connected to one or several subdetectors has the following characteristics:

- The first buffer (circular synchronous type that retains the history of the events) has a capacity of 4 MByte distributed on 4096 processors
- The second buffer used to derandomize the data has a capacity of 5.5 Mbyte of memory to handle a high event rate at the input. This second buffer is asynchronous.
- The flow of the data is regulated by the data-driven principle, and the data-dependency on input and on output has shown in this simulation that no data was lost and that it took 3079 3D-Flow cycles to transfer 4096 parallel input 16-bit data in serial into one 3D-Flow chip with 4 processors.
- The maximum throughput of a single 3D-Flow chip at the output "Bottom" port is 1.6 Gbyte/s for a 200-MHz 3D-Flow chip and 320 MByte/s for a 40-MHz 3D-Flow chip, but the effective throughput considering the delay of two cycles between boards and the program execution

of the data routing in the pyramid is one third, and requires three cycles for each input data.

To simulate this DAQ scheme, one day was required to write all programs and to load all 5500 processors, a half-day to debug it, and 5 hours to simulate it on a workstation and obtain the log file with the results.

#### IV. TIMING CONSIDERATIONS, EVENT IDENTIFICATION, AND TAGGING

The coherency of the timing is kept very simple in this scheme. The 3D-Flow system within the Level-1 trigger provides the event number (bunch-crossing) to the three buffers. For the asynchronous first-stage buffer, the "ACCEPTED" trigger event information must be sent a few cycles before it is sent to the synchronous first-stage buffer and to the asynchronous second-stage buffer, because a short 3D-Flow program needs to be executed to initialize the bank that has to be sent to the output.

Since the routing of data in the pyramid is well known, and is derived by the data-driven principle from the programs loaded into the pyramid the user will know which will be the first data of an event that will exit from the vertex of the pyramid. The user can thus tag events by providing at the input channel of the pyramid (that is known to be the first to reach the output according to the routing and 3D-Flow programs in the array), a header and the event bunch-crossing ID.

#### V. PERFORMANCE CONSIDERATIONS FOR LARGE AND SMALL SYSTEMS

The simulated module described above gives the results in number of 3D-Flow cycles. In order to evaluate the system performance of the 3D-Flow system for a particular application, the reader has to:

1. make the best use of the 3D-Flow chip in a particular application as reported in the example of Section III.
2. take the results of the simulation reported in Section III. D, reflecting the behavior of the 3D-Flow chip.
3. apply the simulation cycle time of the 3D-Flow chip available from industry for the year the system has to be implemented (at present 40 MHz).

It is acknowledged by many expert electronic engineers that, for what concerns the interconnection of chips (see figure 3), the layout of the entire 3D-Flow system as proposed in the report SSCL-445 and built for 1280 channels, can easily sustain any version of the 3D-Flow chip up to 500 MHz without incurring in major problems.

In order to give an idea of the performance of the system at different clock frequencies, results of the simulation are provided in Table 2.

Table 2. Simulation Results

3D-Flow rate clock speed chip	Number of input channels/modules (channel = 16-bit, module = 4K or 16K)	Input data rate of the module	Output data of last 3D-Flow chip in the pyramid
40 MHz	16K channels	3.2 KHz	106 MByte/s
40 MHz	4K channels	12.9 KHz	106 MByte/s
200 MHz	16K channels	16 K Hz	533 MByte/s
200 MHz	4K channels	64 KHz	533 Mbyte/s

The interpretation of these results tells us that the 3D-Flow architecture may be applied to small experiments as well as to large experiments. In Table 2 one can see that for most of the experiments (from present to LHC-type), the output rate of the Level-1 trigger is in the range of 3 to 64 KHz (used as the input data rate to the funneling of a large number of parallel input channels to one 3D-Flow chip). The best use of the 3D-Flow chip in order to find the best ratio price/performance is to find the best compromise for each application between the module input data rate desired and the use of the internal memory of the 3D-Flow chip as buffer.

## VI. 3D-FLOW ASSEMBLY

The basic elements for the construction of a 3D-Flow parallel-processing system are the daughterboard printed circuits. Each accommodates four 3D-Flow chips (each chip has four 3D-Flow processors) used to build the stack of the parallel-processing system. Another daughterboard, with the same dimensions and connectors as the previous and accommodating only one 3D-Flow chip, is used to build the pyramid on input and the pyramid on output of the system to distribute the data from a single source and to funnel data to a single output channel respectively. In most high energy physics applications one uses only the stack of boards for the parallel-processing system and the output pyramid to funnel parallel input signals to one output signal. At 90 degrees with respect to the stack of boards, a Data Acquisition system made of standard VME 3U-size board interfaces data from the detector front-end electronics to the 3D-Flow system.

Figure 3 shows the assembly of the daughterboards with their interconnections in a parallel-processing system with an output pyramid. This pyramid has been defined and simulated entirely with two types of printed circuit boards and short connecting cables of only slightly different length. Short in this context means that no other geometrical configuration can obtain shorter length in a scaleable manner. The boards are stacked together to form the 3D-Flow system and are joined at 90 degrees to a 3U Mini-Rack. Figure 4 shows the construction of a system for a 1280 channels data acquisition and figure 5 show the construction of a 3D-Flow trigger system suitable for calorimeters for an equivalent size of channels. Figure 6 shows the details of the construction of a

Mini-Rack with the connections among the 3D-Flow parallel processing system stack.

## VII. CONCLUSIONS

The present feasibility study and simulation of the 3D-Flow processor and system architecture aims to demonstrate that the 3D-Flow is suitable to solve the different functions typical of a data acquisition system (synchronous buffering, asynchronous buffering, funneling, etc.). A simulation of a 3D-Flow processor and system architecture for the funneling of 4096 fragmented 16-bit event data for a partial event builder has been made. This simulation, even if it demonstrates the suitability of the chip for this application, does not exploit all the intrinsic possibilities of the chip to execute much more complex algorithms (e. g., filtering, zero suppression, buffering, etc.) that may be taken advantage of by the inventive physicist. In the simulation, the 3D-Flow parallel-processing system was instead programmed only for simple operation of data movement in order to verify its functionality and to determine how many steps it would take to move all data from all parallel input channels of a module to one 3D-Flow output chip.

## VIII. ACKNOWLEDGMENTS

All simulations for this report were done at the author's home, following termination of the SSCL. The 3D-Flow was adopted as the digital trigger of the GEM experiment in 1993, and part of its development was funded during the close-out phase of SSCL. I am grateful to the design drafter Heidi Hazlett, for the drawings of some mechanical parts, and to machine technicians Mike Thomas and Shelly McMillion for assembling 80 Mini-Racks and 10 cabinets for a 1280-channel system. Paola Mastromarino from CRS4 and Abdul Akbari, a student, contributed to the project during the last phase, from June to September 1994. Thanks to the Centro di Ricerca, Sviluppo e Studi Superiori in Sardegna, Italy, for having given a three month grant to Paola Mastromarino to work on the project. Special thanks to A. E. Werbrouck from Dipartimento di Informatica of the University of Torino, Italy, and Sergio Conetti from the University of Virginia for the very useful interaction helping me to make these results more understandable to the reader. Also I would like to thank J. Naples for his invaluable help in editing this document.

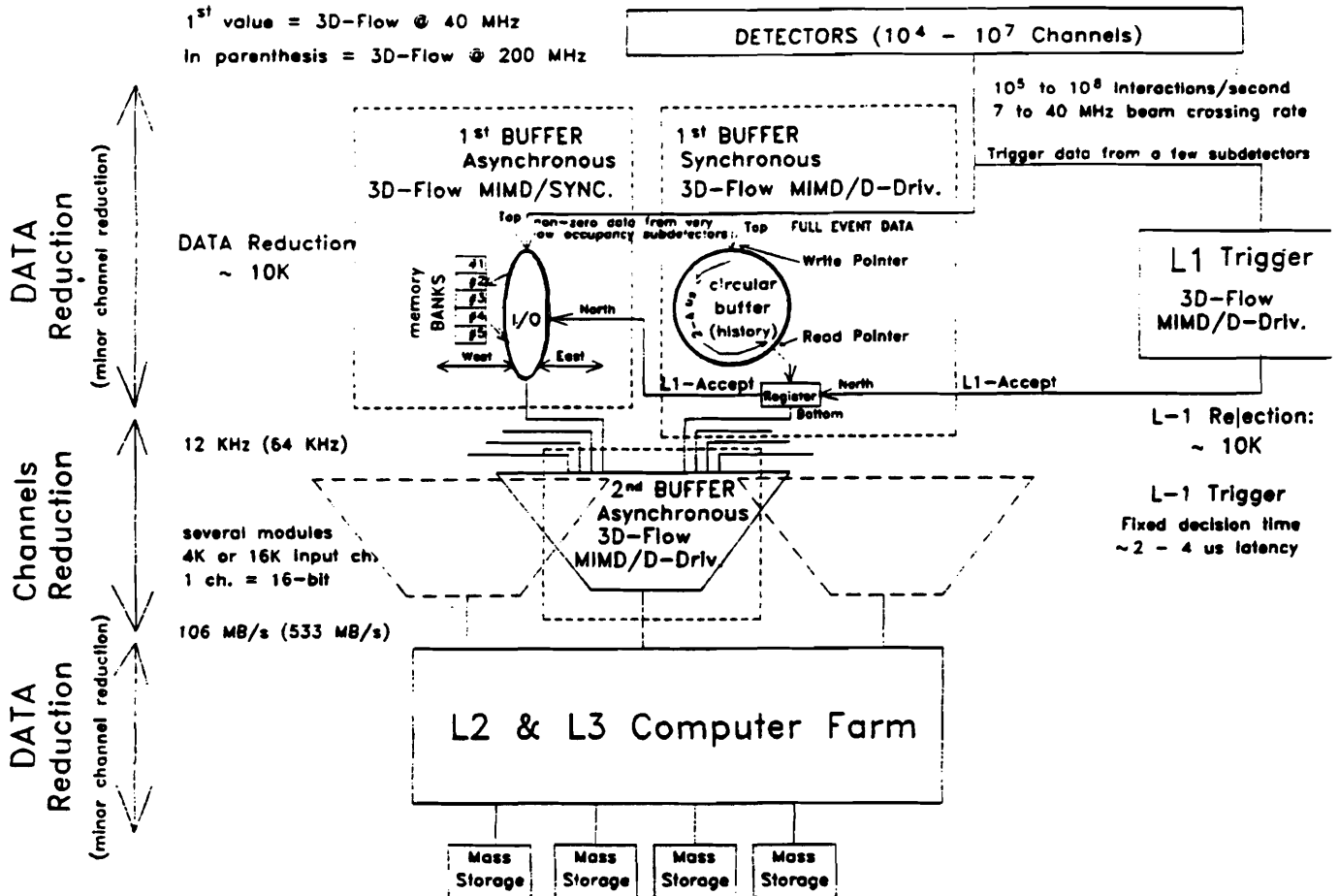


Figure 1. Main components of a typical trigger and data acquisition system.

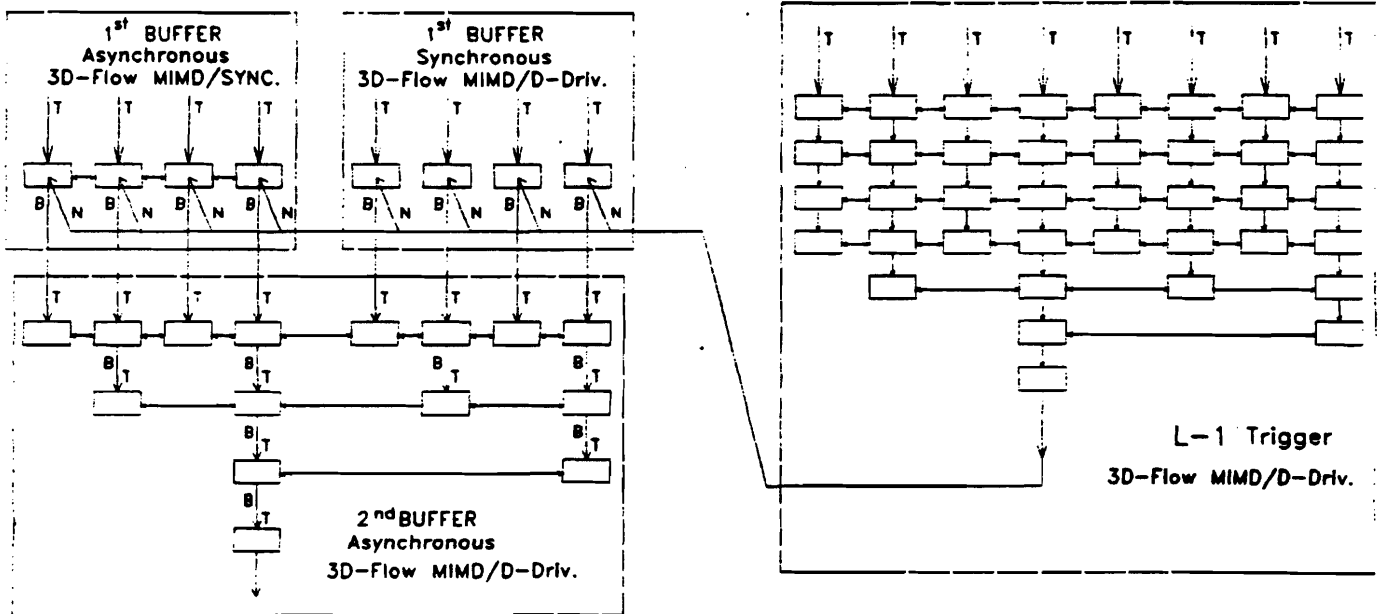


Figure 2. Event flow diagram in a 3D-Flow system.

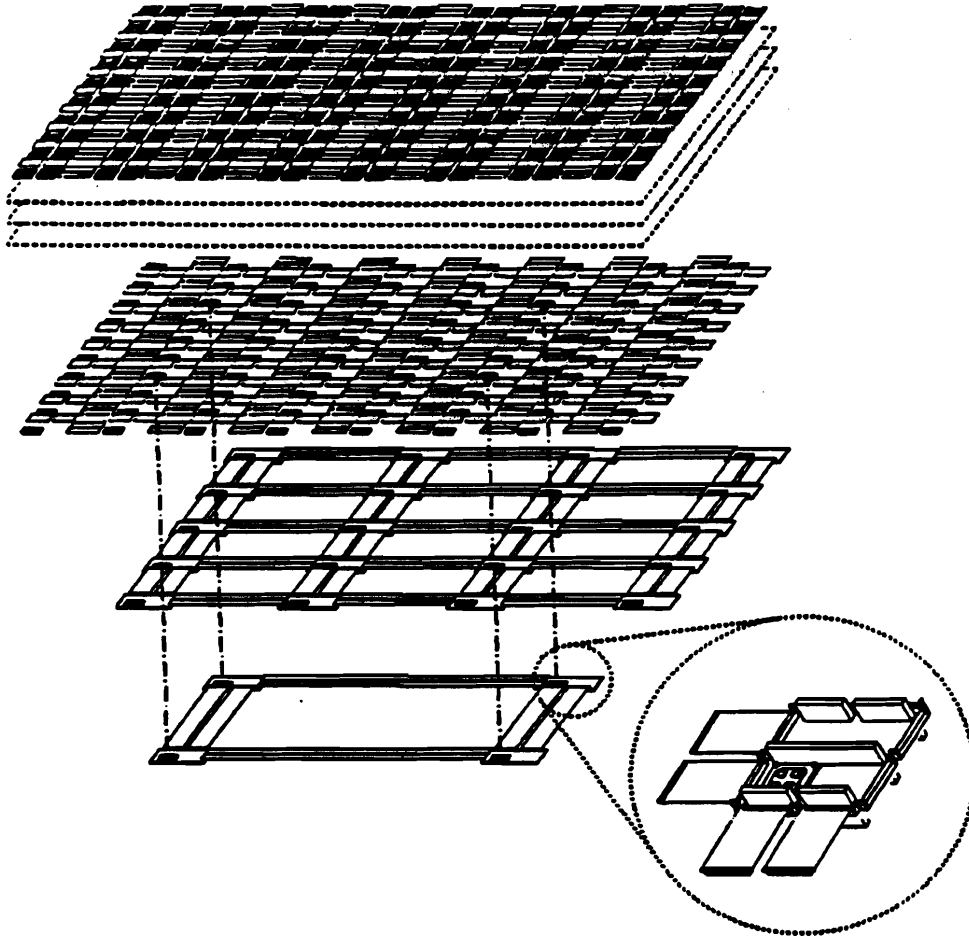


Figure 3. Pyramidal 3D-Flow daughterboards interconnection scheme for DAQ and Trigger channels reduction.

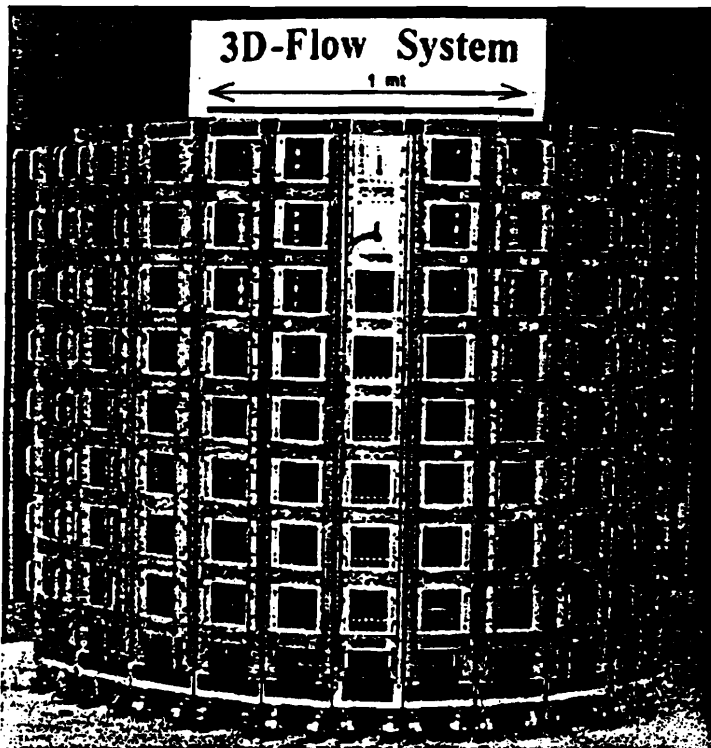


Figure 4. 3D-Flow system in a planar assembly for DAQ applications.

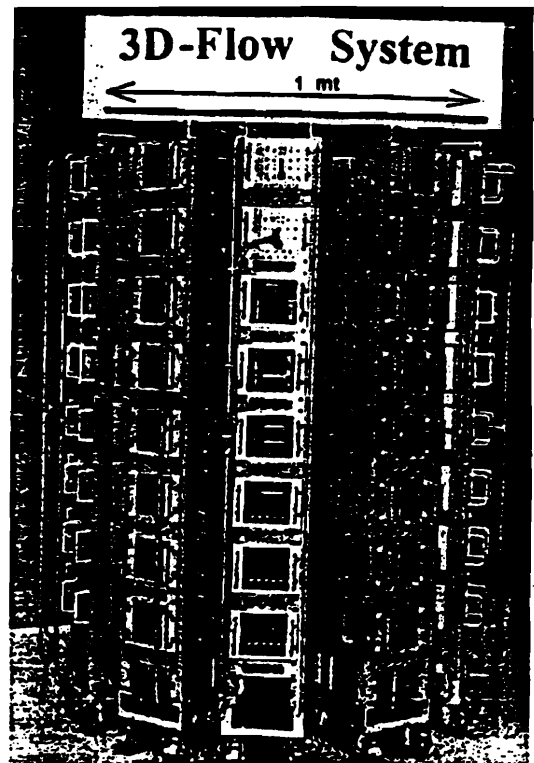


Figure 5. 3D-Flow system in a cylindrical assembly for trigger applications.



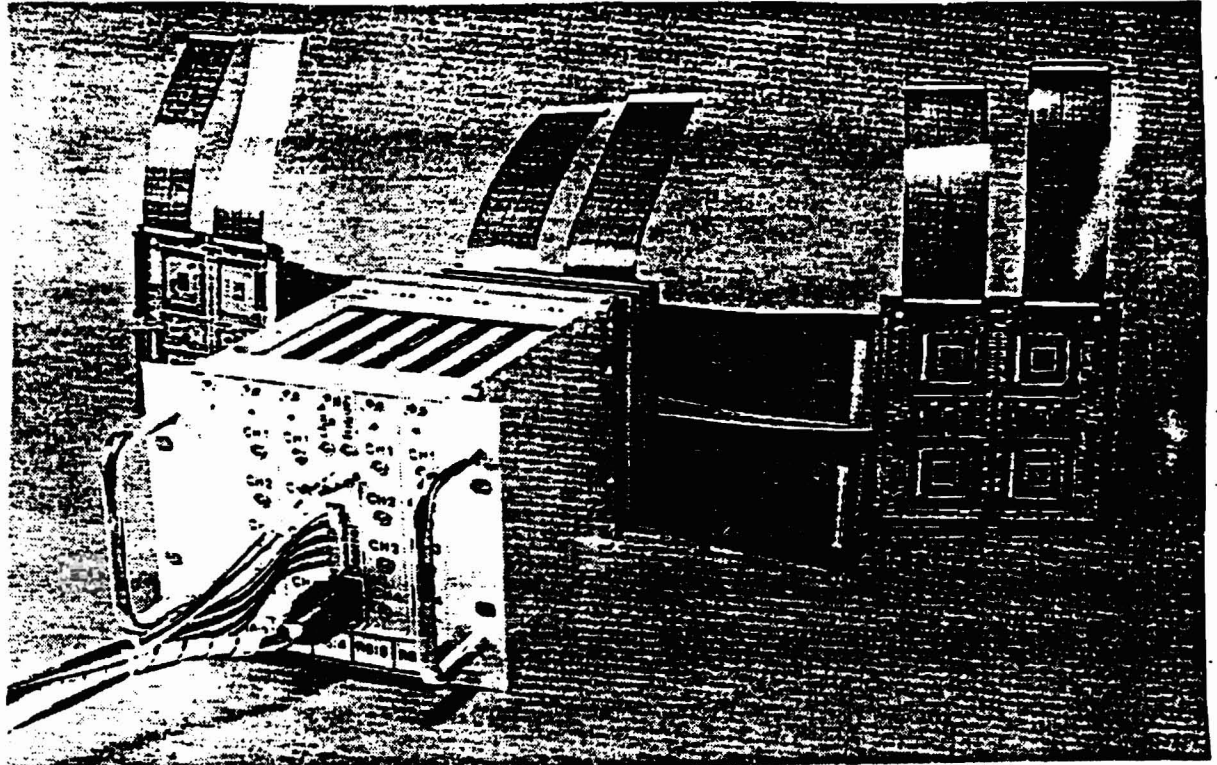


Figure 6. 3D-Flow Mini-Rack with standard 3U x 160 mm DAQ boards

## REFERENCES

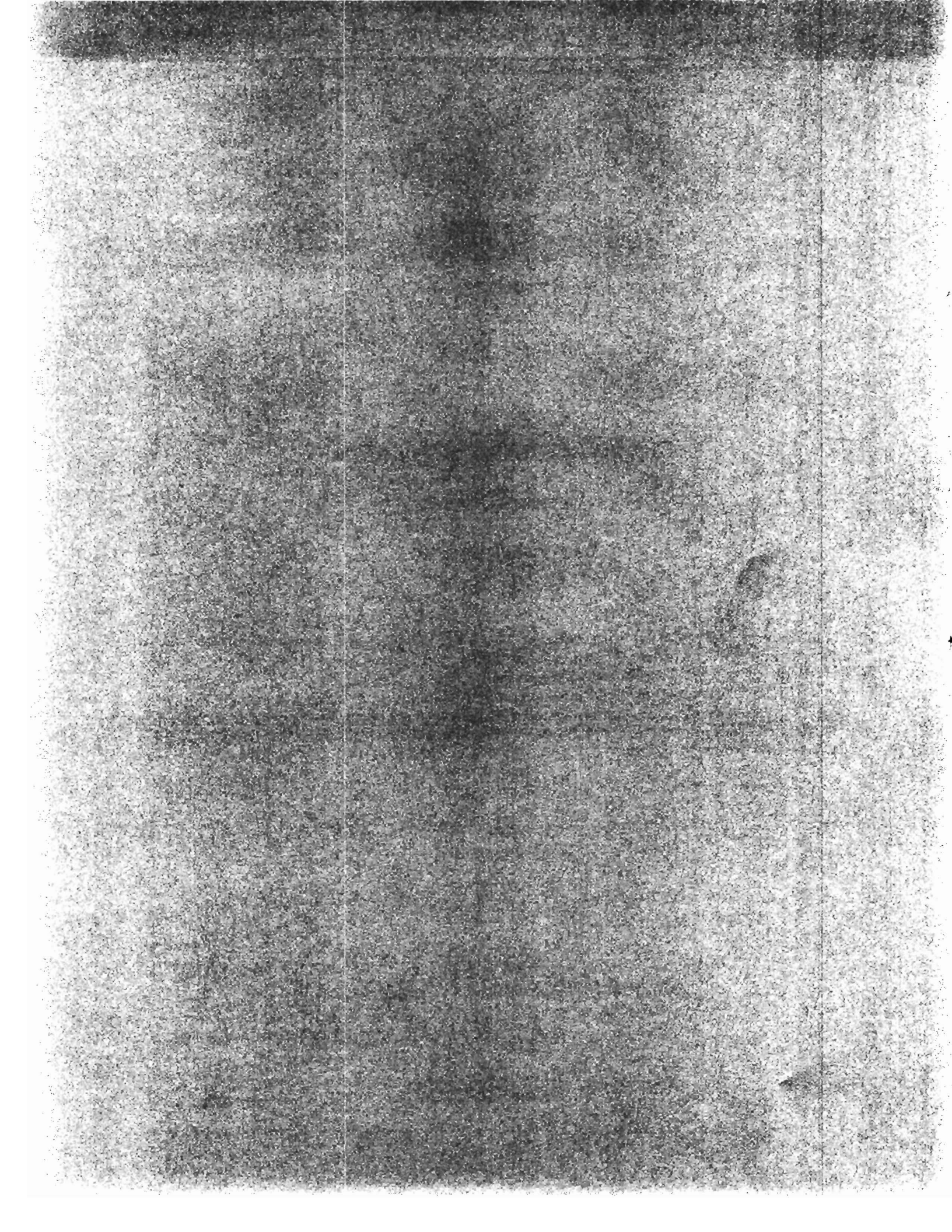
- [1] D. Crosetto "3D\_Flow with less than 100K gates versus processors with million of transistors for DAQ and Level-1 Trigger". Nuclear Science Symposium, 1994. October 30 - November 5, 1994. Norfolk, Virginia.
- [2] SDC Technical report (TDR)
- [3] GEM Technical report (TDR)
- [4] ATLAS Letter of Intent
- [5] CMS Letter of Intent
- [6] Proceedings of the Workshop on B Physics and Hadron Accelerators. Snowmass, Colorado. June 21 - July, 1993
- [7] M. Rijssenbeek, "The D0 Upgrade Program and its Physics Potential." Proceedings of the International Conference on High Energy Physics, August 6-12, 1992, Dallas, TX, p. 1902-1907. Ed., J. R. Sanford.
- [8] Proposal for an Upgraded CDF detector. CDF/DOC/PUBLIC/1172
- [9] STAR conceptual Design Report. PUB-5347

## **An SCI Video DRAM Memory Module**

**Bernard Skaali**

**University of Oslo**

A high speed SCI memory node utilizing Video DRAM has been designed. This type of memory is a good candidate for memory modules in SCI environments because of its high speed and simple R/W operations on cache lines. The memory node contains two blocks of memory, the main memory in VDRAM and an SRAM memory for storing the tags of the cache lines. Input and output FIFOs provides the data path to the external bus of the SCI nodechip. The state machine controller supports the whole set of commands of the "NodeChip" from Dolphin Interconnect Solutions.



# An SCI VideoDRAM Memory Module

**H. Golparian<sup>1</sup>, B. Skaali**  
**Department of Physics**  
**Ø. G. Larsen**  
**Institute of Informatics**  
**University of Oslo, Norway**

<sup>1</sup>Email: hamidgo@fys.uio.no

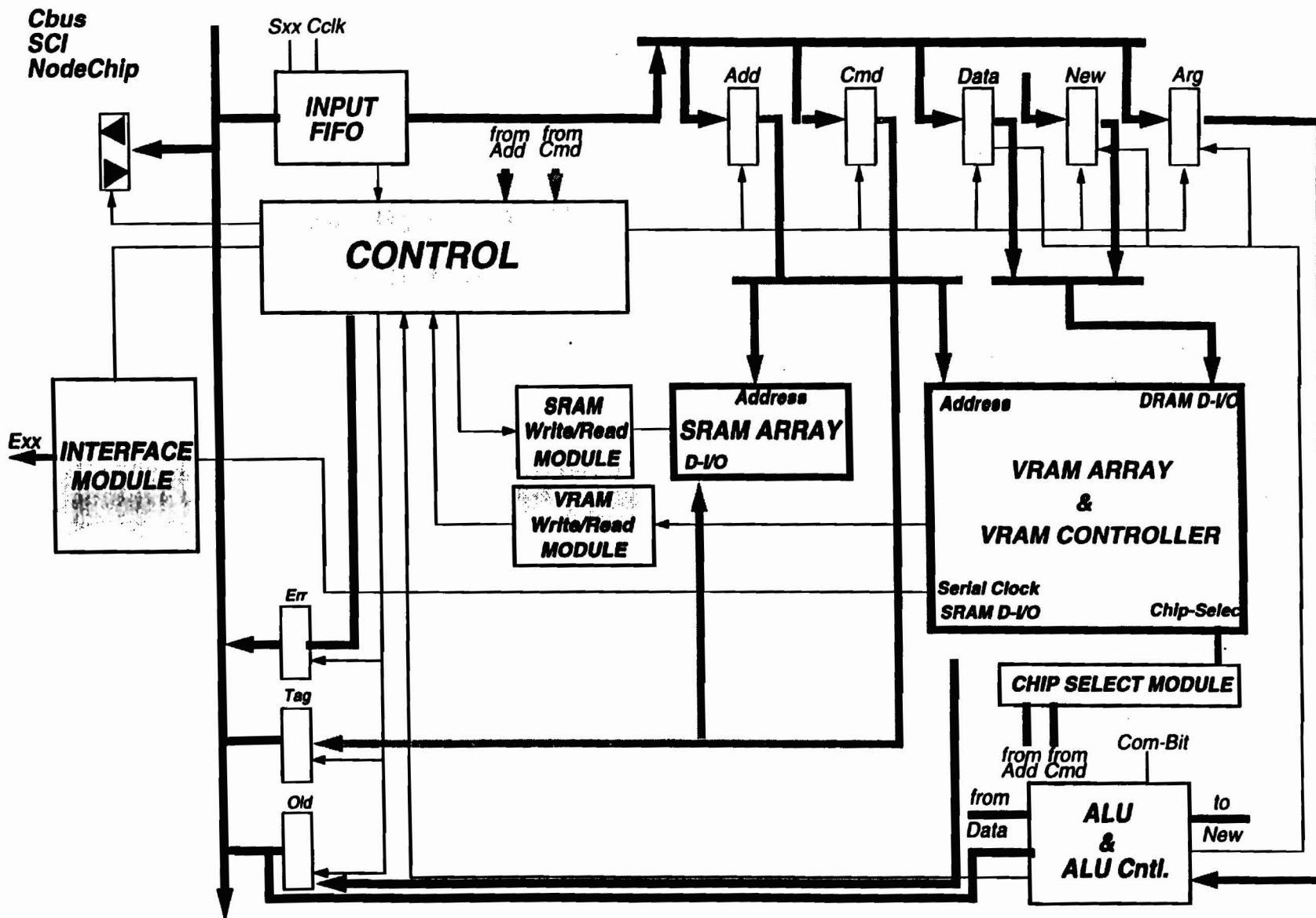


## The SCI VideoDRAM Memory Project

- VideoDRAM is DRAM with a 256-bit high speed register on board (SAM port). It offers an interesting choice for memory storage in an SCI environment:
  - ✓ will speed up and simplify read/write of cache lines, each line can be placed in consecutive locations of the same row.
  - ✓ the architecture permits concurrent SCI operations; high speed R/W through the SAM port independent of DRAM operations, i.e. a 64 byte SCI read and a 64 byte SCI write can overlap.

- The main components of the SCI VideoDRAM-memory module are:
  - 1) main memory in VideoDRAM
  - 2) cache tag directory memory in SRAM
  - 3) controller implemented as interacting state machines in EPLDs plus FIFO buffers.
- The module is built on a double Eurocard (power only from VME crate), 10-layer PCB, first version based on the Dolphin CMOS NodeChip mezzanine card, contains 2 Mbytes of VideoDRAM plus 96 kbytes of SRAM cache tag memory.

- Some of the features:
  - ✓ Implements the full set of Cbus request command from the SCI NodeChip, including R/W line coherent.
  - ✓ Employs fast EPLDs for implementation of state machines.
  - ✓ Uses a programmable Video DRAM controller in order to guarantee critical access timing parameters and automatic refresh.
  - ✓ With NodeChip Cclk at 25 MHz, simulations give a peak bandwidth of 200 Mbyte/s.
- Status: Module expected operational before end of '94



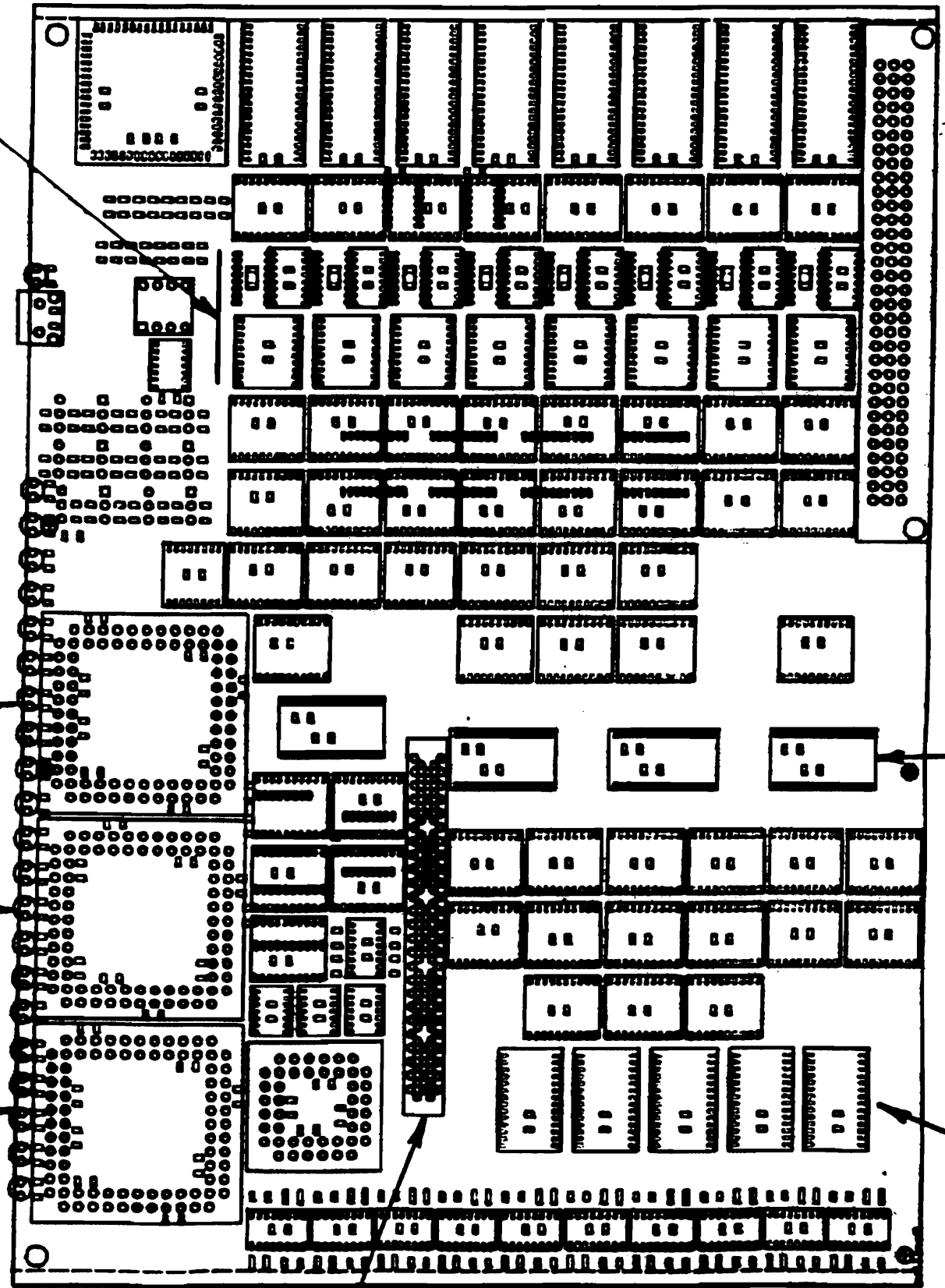
**Simplified Block Diagram of the SCI VideoMemory Controller**



ALU

CONTROL (EPLDs)

VRAM  
ARRA



IN/OUT

SRAM  
ARRAY

BUS  
CONN.

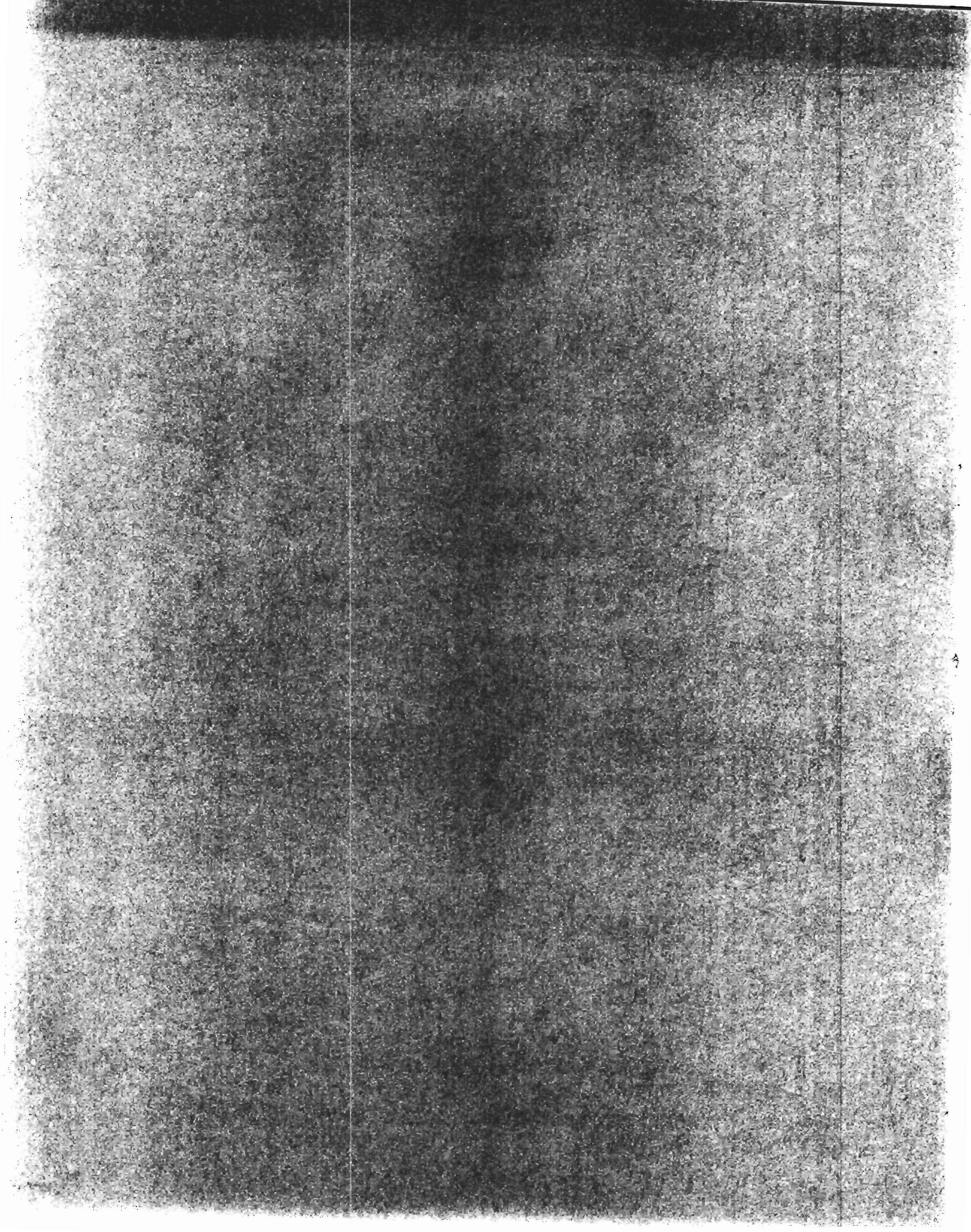
## **FASTBUS CHI-SCI Link**

**Bernard Sknali**

**University of Oslo**

The FASTBUS CHI-SCI link has been designed to provide a simple bridge between a CERN Host Interface (CHI) FASTBUS Master and SCI. The CHI contains an MC68030 processor, local memory and a triple port data memory accessible from the processor, the FASTBUS ports and an I/O host port.

The SCI link is implemented as a daughter board which is connected to the I/O host port. The link is a firmware driven FIFO interface, using a AMD29200 RISC processor. The first version uses a CMOS NodeChip on a mezzanine card mounted on the daughter board.

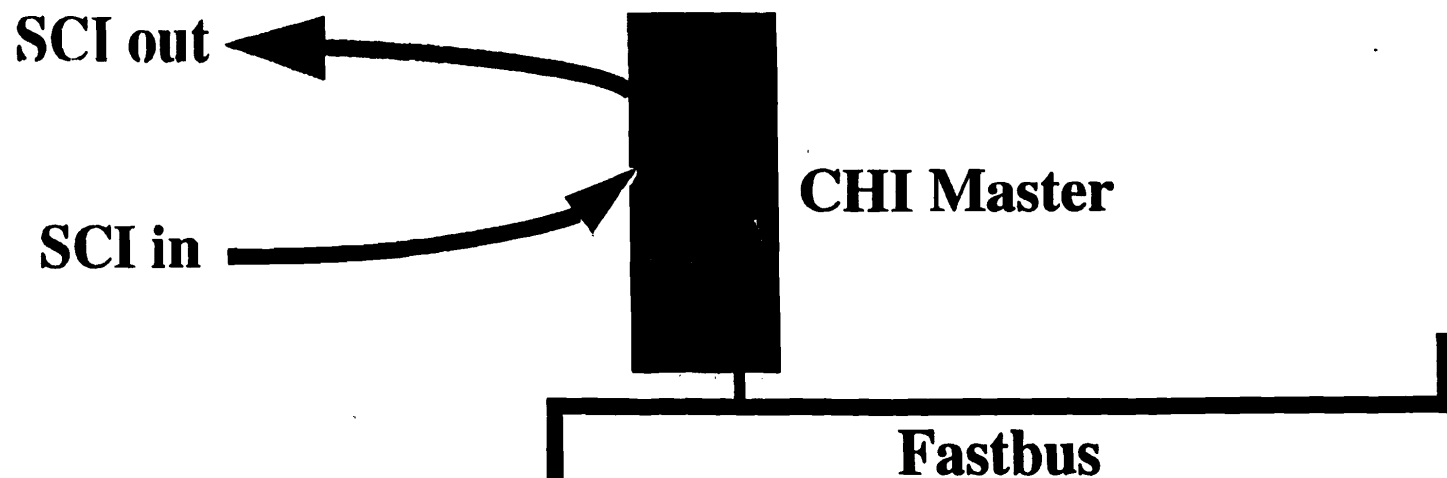


# Fastbus CHI-SCI Link

**J. Wikne, H.L. Opheim, B. Skaali**

**Department of Physics, University of Oslo, Norway**

**Email: [jon.wikne@fys.uio.no](mailto:jon.wikne@fys.uio.no)**



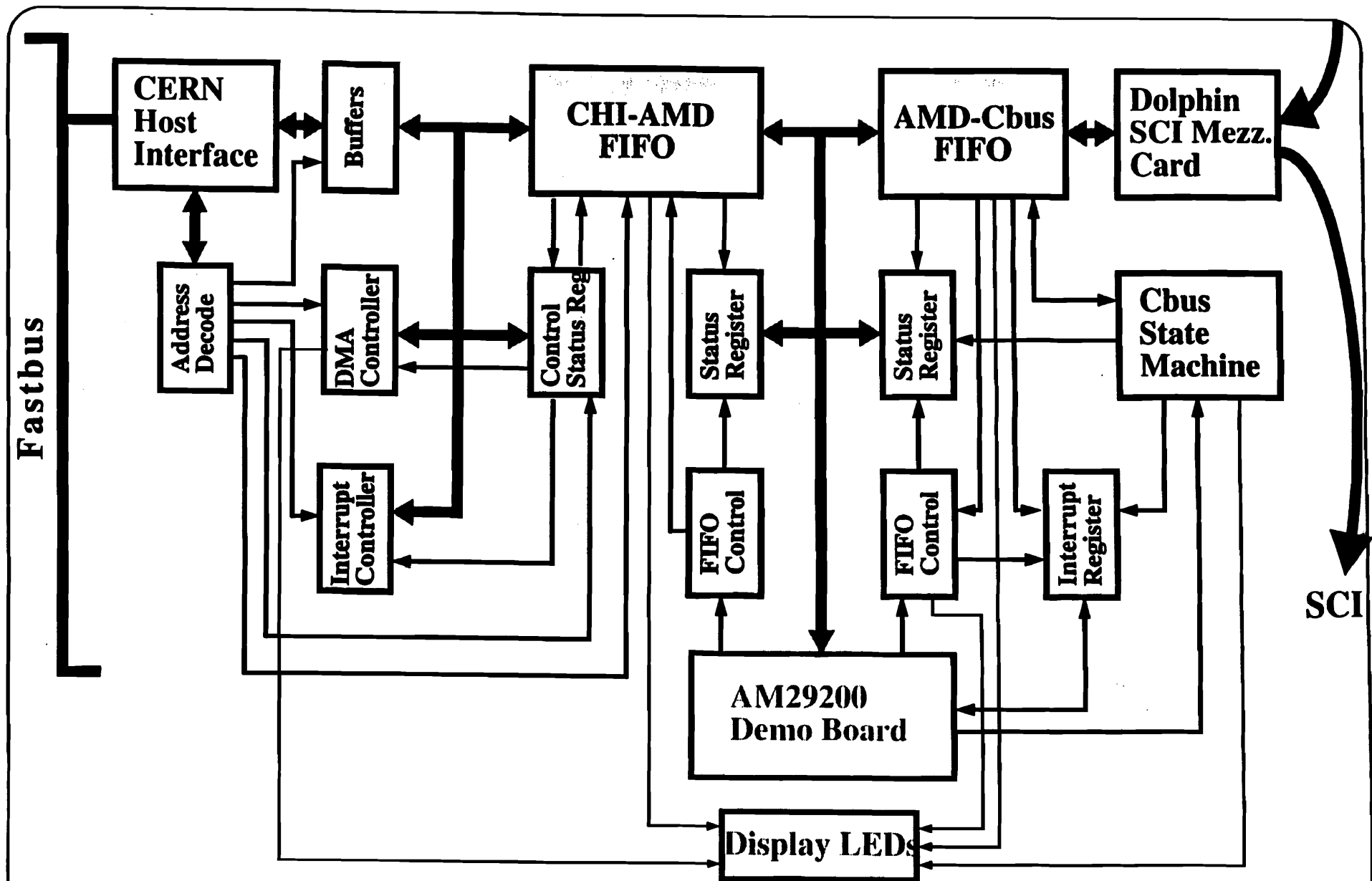
## **CERN Host Interface**

- **The CERN Host Interface (CHI) is a family of interfaces to interconnect Fastbus, VMEbus, and external host computers. The Fastbus interface consists of a processor board (CHI-P) with an MC68030 with FP coprocessor, and an I/O port to host daughter board. The CHI-P contains a 1MB triple-port data memory which allows concurrent access by Fastbus (as master or slave), the host link, and the on board processor. The CHI is manufactured by Struck, Germany.**

## **CHI-SCI Link**

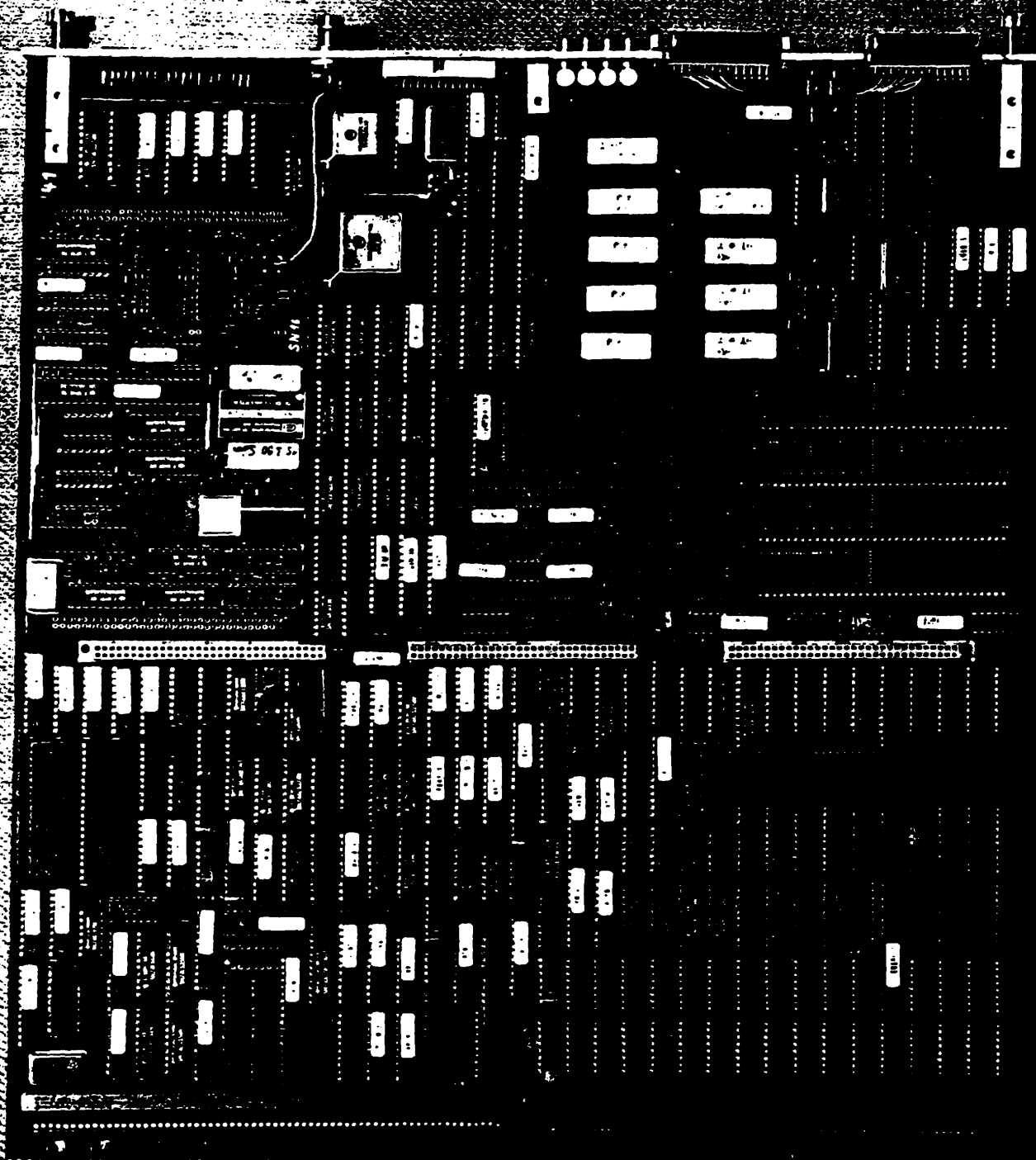
- The CHI-SCI link provides a simple bridge between Fastbus via the CHI and SCI. The CHI-SCI is implemented as a daughter board connected to the data memory via the I/O port of the module. The design has been done in collaboration with Struck.
- The link is a firmware driven FIFO based interface, using a AMD29200 RISC processor card. The FIFOs are used to implement a 64 bit wide data path + mailboxes.

- AMD29200: 16 MHz, DMA controller, interrupt controller, 16 programmable I/O lines.  
FIFO: Mosel MS76542, 36 bits, 256 words deep.  
Cbus state machine: 22V10 PALs 7.5 nsec.
- First version uses a CMOS NodeChip on a mezzanine card from Dolphin ICS. The Cbus controller state machine recognizes 14 SCI packet types, among those R/W Selected Byte, R/W 64 bytes non coherent and Move 64 bytes.
- Status: debugging mainly done



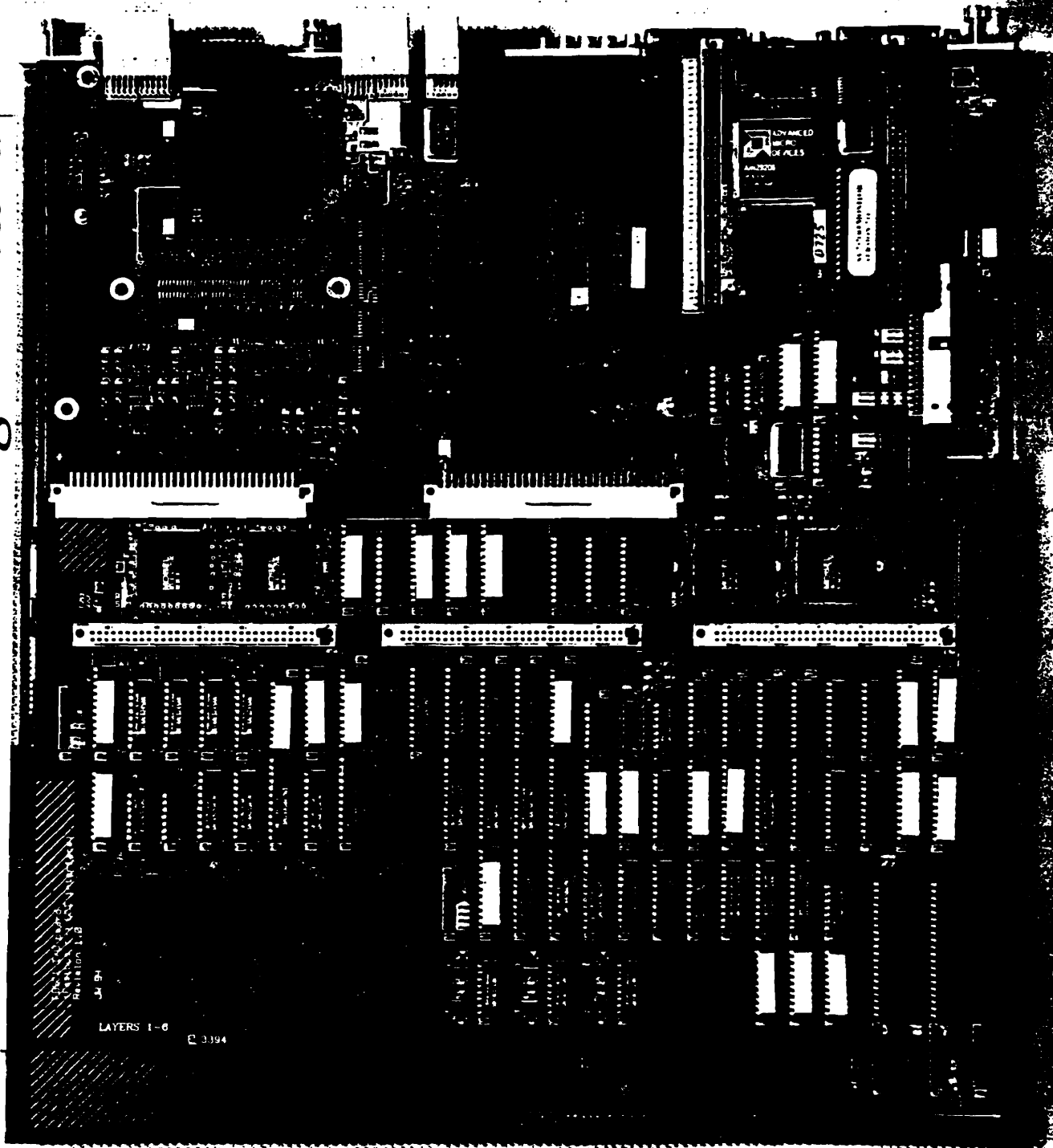
**Block diagram of the CHI-SCI Link**





# CERN Host Interface (CHI)

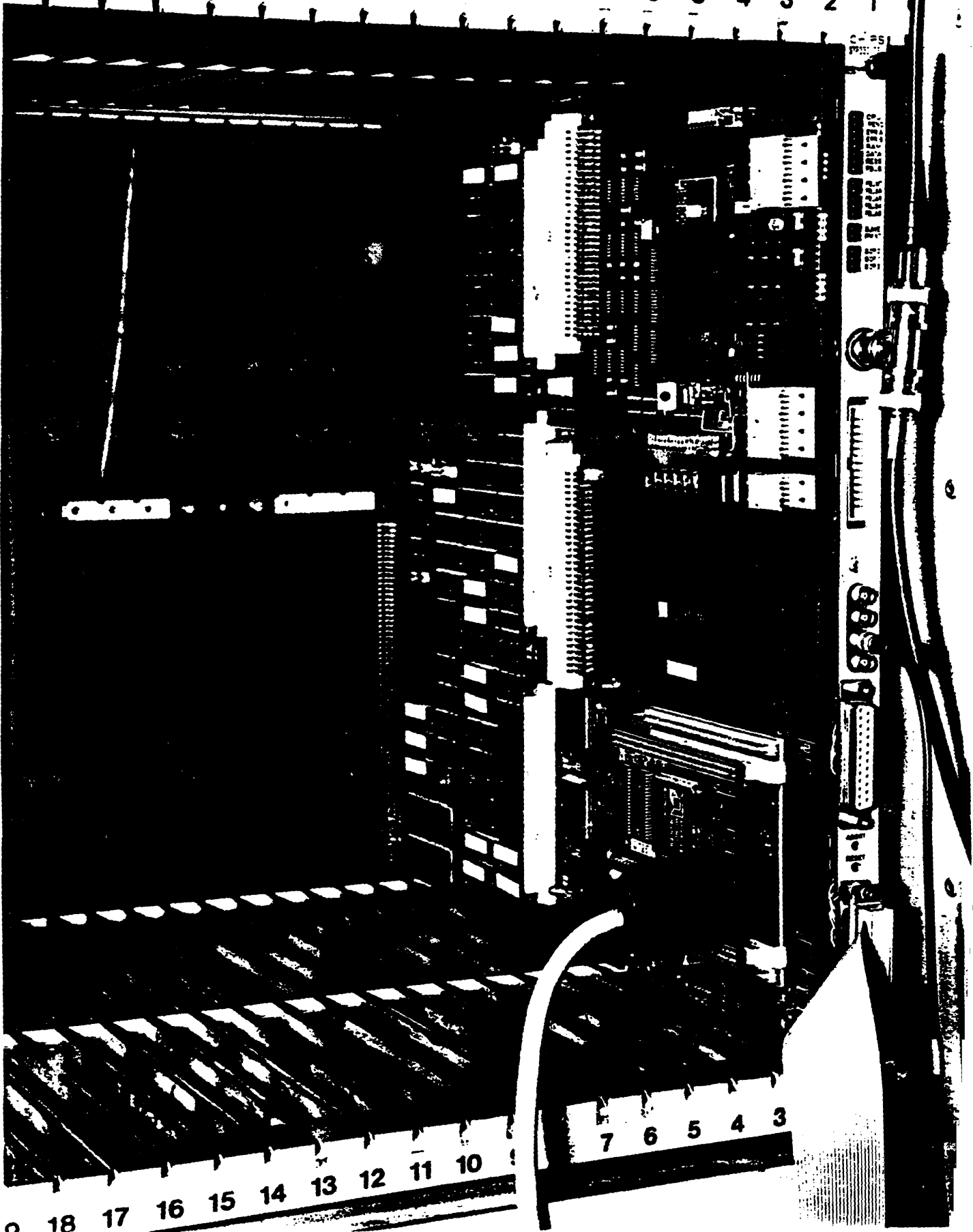
# CHI-SCI Link daughter card



2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

DR. B. STRU

18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1



18 17 16 15 14 13 12 11 10

7 6 5 4 3

# **SWIPP - Switched Interconnection of Parallel Processors - A General Purpose Heterogeneous Multicomputer Optimized For Data Acquisition**

**Yngvar Lundh      Oddvar Sorasen**

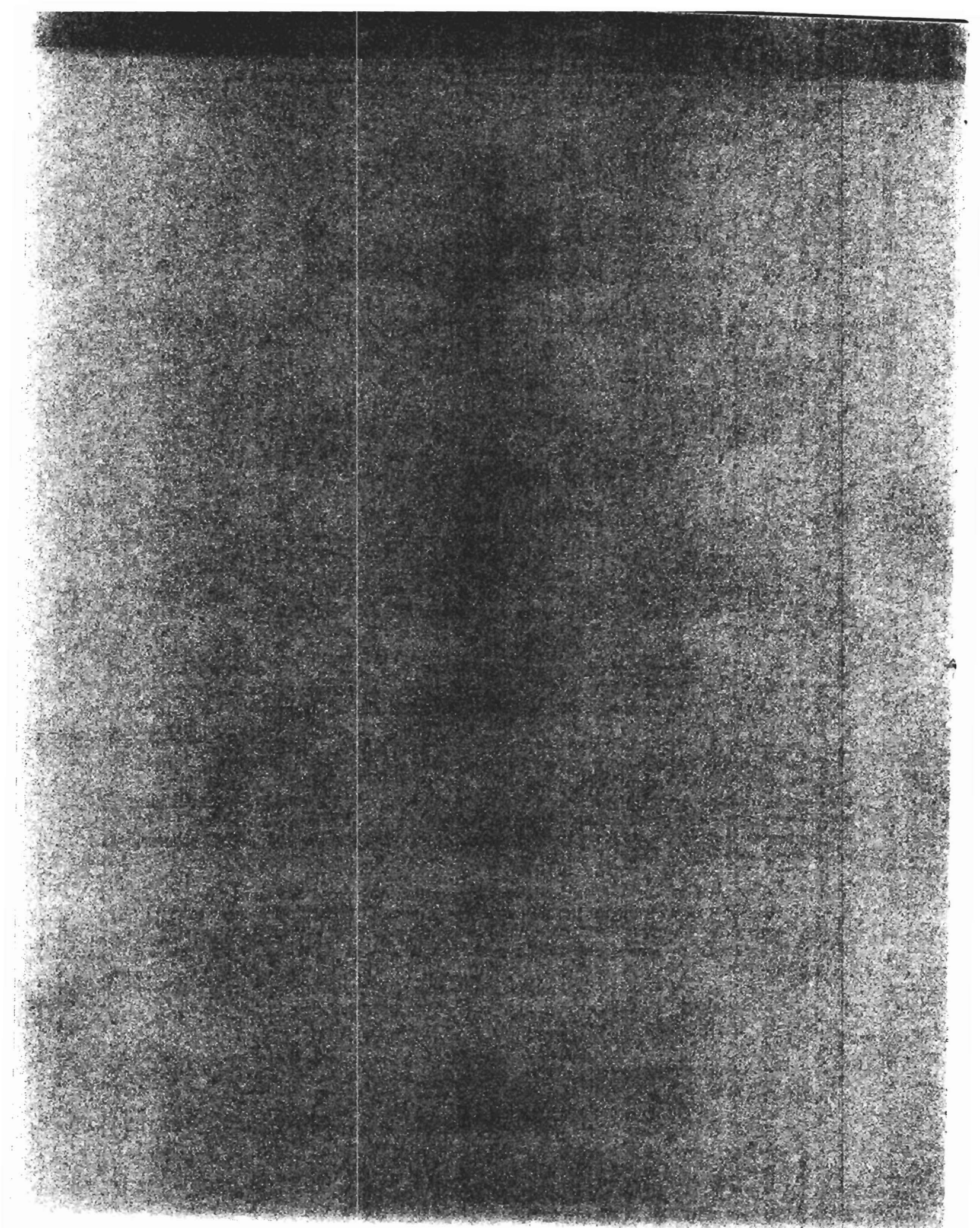
**University of Oslo**

Swipp is a method for interconnection of multiple computers. Various machine types, "Computs Engines" - CE - can be combined for cost-effective implementation of demanding information processing systems. Among specialized computer types, which can be building blocks of such multicomputers, are data-capturing detector modules and very high speed storage machines. "Detector Data Readout and Event Building in an LHC Inner Detector Experiment" has been described as an example (IEEE Trans. on Nuclear Science, Feb. 1994 pp.'s 246-251).

In this poster special consideration is given to Swipp regarded as a processor which is programmable by conventional methods in spite of being a powerful multicomputer. Load sharing between the various specialized constituent machines can be facilitated by parallelizing program compilation.

A key role in achieving scalability is played by an embedded "Control Computer" - CC - which is part of the "Protocol Engine" - PE. One PE is associated with each CE. Each CC can run part of a distributed operating system. This allows symbolic object names to be employed to avoid some of the address space limitations of shared memory systems. It also levitates the requirements for latency, which are typical of some interconnection methods, while retaining efficiency.

Swipp aims to exploit state of the art circuit integration and fibre transmission. I will complement some existing interconnection schemes. Each CE can be a standard or non-standard computer. Or it can itself be a multicomputer, multiprocessor or even a computer network. Hence Swipp lends itself well to be used as basis for a higher level programming paradigm.



# Swipp - Switched Interconnection of Parallel Processors - a General Purpose Heterogeneous Multicomputer Optimized for Data Acquisition

by Yngvar Lundh and Oddvar Søråsen<sup>1</sup>

Department of Informatics, University of Oslo, P.O.Box 1080 Blindern, N-0316 Oslo, Norway.

## Abstract

To take advantage of the cost effectiveness of specialized processors is the objective of the Swipp concept. It is an intelligent and efficient interconnect system to function as a distributed operating system. Together with parallelizing program compilation this allows standard application programming methods. Interconnect and operational control are performed by "protocol engines", with embedded control computers. They communicate through a switched network.

## I. Introduction

Data acquisition in particle physics experiments may be highly demanding in terms of processing capacity. In fact the requirements are such that special efforts are being made to find solutions within acceptable economic constraints. The "Large Hadron Collider" - LHC - experiments being planned at CERN represent extremely demanding data acquisition processes [1]. The data flow is unusually large and requires sustained processing. At the same time the processing needs to be high-level programmable due to the complexity of the experiments and project as a whole, with needs for alterations.

A heterogeneous multicomputer principle - "Swipp" - (SWitched Interconnection of Parallel Processors) is described. It allows demanding information processing loads to be shared by a number of "Compute Engines" - CEs - of various types. Highly special CEs can be part of the multicomputer, as necessary to perform special tasks not feasible for "standard" computers. A data acquisition and -processing multicomputer can be configured as a set of CEs which combine into a capability profile to match that of the processing demand.

The front end stages of the LHC data acquisition are used as examples of such a demanding task. The programmer need not necessarily know about computer types, address space etc.

## II. General Principles

Swipp is an intelligent interconnect system with certain management capabilities [2]. Information is passed between a number of compute engines - CEs, figure 1. Associated with each CE is a "protocol engine" - PE. Information transfer between a CE and its associated PE is handled by the following functional channels:

- CE has an internal memory M.
- PE can read and write in M, one word at a time in parallel, usually on a cycle stealing basis in a direct memory access - DMA - mode of operation.
- Each CE can generate a call signal to PE.
- PE can send an interrupt signal to CE.

All information transfer essentially takes place on the initiative of the PEs (masters). Also PEs can control the operation of CEs (slaves) by loading programs into them and starting execution. CEs report state changes to their respective PEs.

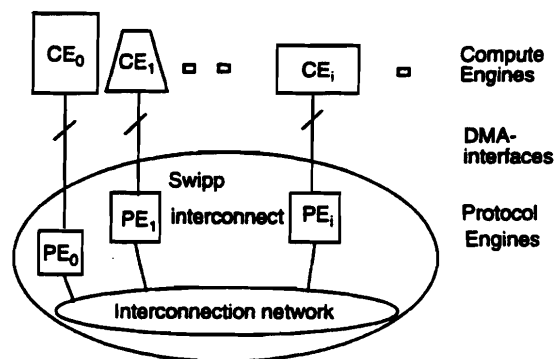


Figure 1. Swipp Multicomputer, principal configuration.

## III. Interconnection Network

PEs send information to each other, on behalf of their respective CEs, through a switched network, figure 2. PEs format the data into packets for transportation in the network. The sending PE retrieves information directly from the memory of its CE. PE itself controls the direct addressing. Similarly the receiving PE writes directly into memory of its CE.

The source PE applies a transmission route at the head of each packet. The route is a sequence of switch output port numbers. This sequence is rotated one step upon each passage of the

1. email: yngvar@ifi.uio.no, oddvar@ifi.uio.no

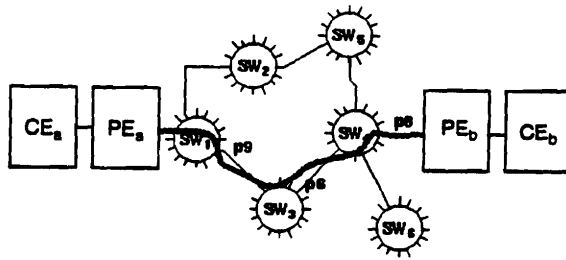


Figure 2. Switched interconnection network (example configuration). A route between protocol engines is indicated.

packet through a switch. Hence each packet finds its predetermined route through the switch network from source to destination  $PE_a$  to  $PE_b$  (source routing). Transmission, between SWs and between PE and SW, is carried by a high speed serial link such as an optical fibre pair [6]. Such a link is designed to have sufficient speed to match or surpass those of the source and destination CE-memories. Hence transmission speed is determined by the slowest of the two memories, in the sending or the receiving CE. No additional load is put on the CEs.

Each switch - SW - is a 16-port cross bar matrix switch plus routing and management circuits, figure 3. Both PE and SW-matrix circuits are also designed for compatible speed [3],[14]. Packets are commutated and decommutated to be handled by byte-parallel switch matrix circuits. Hence sufficient switch speed is achieved by matrix circuit complexity. For switch throughput speed, a small delay  $D$  is inserted in each input port, figure 3. This delay allows the matrix channel to be opened to let the packet directly through (wormhole routing). Delay time in  $D$  is the time required for decode and control circuits to extract the first output port number from the packet head, then to decode it and open the channel through the matrix. Hence packet transmission from  $PE_a$

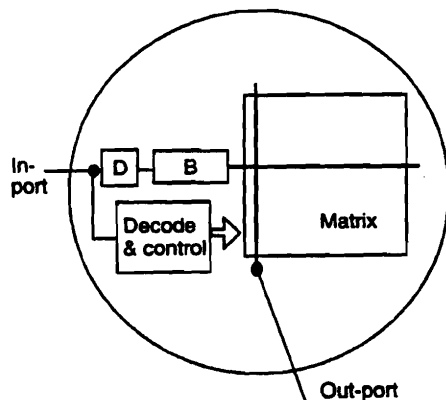


Figure 3. Switch principles. Each of the 16 switch ports has an in- and an out-port.

to  $PE_b$  goes directly through the network with essentially no delay other than hardware address decoding at each switch

Each switch has a FIFO-buffer  $B$ . If the onwards channel is clear, the stream goes straight through. If not, the data stream is buffered in  $B$ . "Almost full" - signals are returned upstream (in the return channel not shown in figure 3) as appropriate to halt transmission, preventing overflow in  $B$  and loss of information. In extreme cases, such buffering may fill up along the entire route. Switches have no further intelligence. Higher level flow control is handled by programs in the PEs.

The high speed packet handling circuits in SW and PE permit variable packet length. Flow control programs in the PEs may set packet length dynamically to optimize network throughput.

Typically, a large data object is transferred from  $CE_a$  to  $CE_b$  in a stream of packets. More precisely then: Besides possible queuing delay, transfer time is due to DMA read and write at the source and destination memories of the CEs, packetizing in the PEs, line propagation, and the sum of delays  $D$  in SWs along the route.

#### IV. Protocol Engines

The PE consists of Control Computer - CC - and Network Interface - NI, figure 4. CC is a programmable computer (embedded microcomputer). NI is a special hardware unit [4],[5]. For circuit speed PE is located physically closely to CE's memory.

The main functions of the NI are:

- To read and write data objects word for word in CE's memory.
- To packetize (outgoing) and unpack (incoming) data objects, including routing information, and to send and receive packets through the interconnection network as packet streams.
- To handle single packets as required for management. Such packets are identified by NI and are sent to CC, CE or other PEs, whatever the case may be.

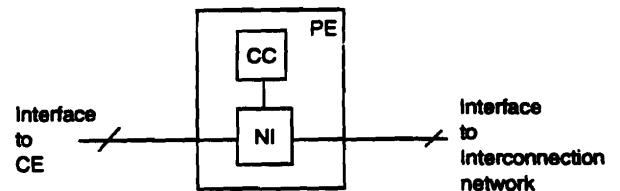


Figure 4. Protocol Engine principles.

NI consists of special logic to ensure fast performance of these functions. It comprises various buffering and formatting registers and control registers. CC treats NI as a set of IO-devices. The typical operation for transfer of a data object from  $CE_a$  to  $CE_b$  is:

- A short negotiation between  $PE_a$  and  $PE_b$ .  $CC_a$  and  $CC_b$  essentially do this negotiation by exchanging a few single packets.
- $CC_a$  and  $CC_b$  load appropriate information into the control registers of  $NI_a$  and  $NI_b$  respectively. Then  $CC_a$  starts  $NI_a$ 's operation.
- $NI_a$  and  $NI_b$  carry out transfer of the data object from  $CE_a$  to  $CE_b$  in a stream of packets. Upon completion, or in unexpected situations such as timeout or CRC error,  $NI$  notifies  $CC$ .

This operation is carried out according to a defined data object transfer protocol. It consists of actions by the  $NI$  and by a "bit-level" driver program in  $CC$ .

The special hardware in  $NI$  is designed to do all packet sorting and all handling of the contents of packet streams. Single packets used in negotiation and for various signalling purposes are identified only, then transferred directly to (or from)  $CC$ . Their contents are analyzed (or composed) by programs in  $CC$ .

$CC$ , who operates in multi-tasking mode, simultaneously executes higher level operational program. Essentially, the set of  $PE$ s in Swipp are managers of the  $CE$ s' operations and of data. When a need arises for transfer of a data or program object from  $CE_a$  to  $CE_b$  the appropriate data object transfer protocol is invoked.

A primary objective of this system design is to permit fast and efficient transfer of large data objects between  $CE$ s while retaining full programming flexibility for operating system design. "Fast" means at a speed limited by the fastest  $CE$  memory. "Efficient" means minimizing the load on the  $CE$ , restricted to memory cycle stealing.

## V. Heterogeneity

The  $CE$ s need not be of the same, nor even similar types. They only need to have a memory accessible by the functional channels listed in section II above. This means that each  $CE$  can be small or large. It can even be a multi-computer itself. This situation can be exploited as follows:

An information processor to cope with unusually heavy demands is built as a Swipp multicomputer. Its constituent computers are of types especially powerful and efficient for the most demanding parts of the information process - "tasks".

Two extreme cases may be considered for such an information processor. At one end of the spectrum a data flow processor can be designed where input enters at one or more compute engines doing the front end processing tasks. Following is a sequence of one or more tasks up until the output step where results are delivered. Such a

"pipeline" may be applicable where all or most of the processing steps are known in advance, at least as far as processing task types and capacity requirements are concerned. At the other end of the spectrum a completely unpredictable set of tasks can be expected. Even then advantage may be taken of processors with special capabilities for vector processing, storage and retrieval, database operations etc, in addition to general scalar operations, for cost effectiveness and high performance. Other task types which can be met by specialized computers are list processing, input data conditioning, presentation tasks e.g. by sound or video. Even analog processing devices may be employed. E.g. neural networks may be used for pattern recognition tasks. Again, the only requirement is that processing devices have digital memories etc. as stated in section II above for communication with its associated  $PE$ .

Demanding scalar tasks may be handled by super scalar computing systems. One interesting example is the IEEE standard for Scalable Coherent Interfacing - SCI [8],[9]. A variation protocol engine can be used to interconnect one or more SCI-rings to networks of one or more compute engines.

The Swipp principles are applicable to an entire such spectrum. Our first development goal is to meet the requirements of specific, demanding processes.

## VI. Distributed Operating Systems - Programming.

To program and operate a Swipp system, one of the Compute Engines can be assigned a special role as "Chief Executive Engine" - CEE. Figure 5 indicates how processing and operating system tasks are shared.  $PE$ s transfer data and programs between  $CE$ s and supervise the execution of information processing tasks in the  $CE$ s. This includes all details required for efficient management.  $PE$ s communicate with the top level operating system in CEE concerning the state of these transfers

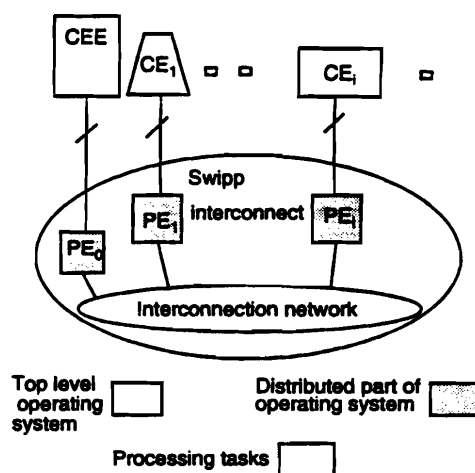


Figure 5. The Chief Executive Engine and the Protocol Engines perform the operational tasks together.



and tasks. CEE also comprises user access and programming tools for system- and application programming

Note that this structure allows several memory address spaces to be in cooperation by symbolic references through the distributed operating system. As an example: When operation has come to a point where matrix  $X$  needs to be transferred from  $CE_a$  to  $CE_b$ , the top level operating system in CEE knows exactly that. The same is known by the respective PEs. In addition the latter know further details of data types, local addressing etc. as needed for the detailed process management, including actually performing the transfer. This use of knowledge about process and data limits the importance of low latency in data transfer [7].

In designing an information processing system for demanding tasks, the multicomputer can be configured to make use of CEs with special capabilities. Use can be made of the designer's knowledge of the types of tasks and the availability of special CEs. These may be highly efficient processors or processing devices such as mentioned in section V above. Substantially improved cost/performance ratio is achievable by specialization. It is the main objective of the Swipp concept to take advantage of specialization while retaining standard application programming methods.

It is a long term goal to exploit this potential for distributed Unix and parallelizing program compilers. The Swipp platform as described lends itself to parallelizing compilation. When viewed from CEE the other CEs are seen as specialized engines. At program compilation special tasks are identified as matching special capabilities of selected CEs. Such tasks are scheduled for execution accordingly. Hence optimum utilization is made of CEs with special attributes.

## VII. Optimization for data acquisition

A possible particle detection system is shown in figure 6. Each CE consists of a number of detector modules - DM - and a Partial Event Buffer - PEB [10],[11]. DMs comprise particle detector devices and first level data conditioning and storage. Upon a real time first level event trigger  $T_1$ , generated elsewhere and supplied simultaneously to all DMs, data are transferred to PEB. A selected fraction of the stored objects (pertaining to an "event") are retrieved from PEB [12],[13].

In Swipp terms PEB is a CE ( $CE^f$  for "front end"), a source from which input data to the next step in the information process are retrieved. The front end units are thus interconnected for execution and information transfer according to Swipp protocols. To meet extreme requirements for speed, capacity, environment etc. special circuits are employed. These are designed to perform a subset of the functions of the ordinary Swipp protocols

only. However, they will never be asked to perform other than those subset functions. Similar functional subset types of PEs and other units can be built into both the DMs and the PEB front end.

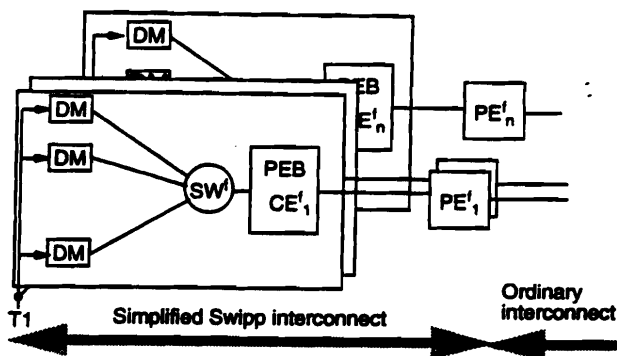


Figure 6. Front end particle data acquisition.

## VIII. Conclusion

A multicomputer concept has been described at the overall system level with emphasis on the aspects of its potential for technical economical optimization of highly demanding information processors. System aspects have been described which point out the potential for general programming methods, distributed operating system and the use of symbolic references to separate memory space. For many applications its intelligent master-slave communication and management system can make more efficient use of processing devices and alleviate the requirement for extreme latency. The system can be viewed as a generalized form of heterogeneous information processing systems which will complement more specific scalar techniques such as SCI.

## IX. References

- [1] CERN, "LHC - Large Hadron Collider", CERN Publications, European Laboratory for Particle Physics, Switzerland, June, 1990.
- [2] Lundh, Y., "Skisse av multiprocessorstruktur", Internal note (in Norwegian), Dept of Informatics, Univ of Oslo, Oct 1987.
- [3] Østby, J.M., Dr Sci thesis in preparation, Dept of Informatics, Univ of Oslo, 1995.
- [4] Blekastad, I., Hagen, M., "Protokollmaskin, en kommunikasjonsenhet for høyhastighets multiprocessor," Cand Sci thesis (in Norwegian), Dept of Informatics, Univ of Oslo, Jan 1989.
- [5] Esvall, R., Bonde, A.S., "Protokollmaskinen PE. En nettuavhengig kommunikasjonsenhet tilpasset SWIPP-konseptet", Cand Sci thesis (in Norwegian), Dept of Informatics, Univ of Oslo, Aug 1992.

- [6] Liao, G., Østby, J.M., Søråsen, O., Lundh, Y., "Self Synchronizing Data Transfer Scheme for Multicomputers", Poster at the 1991 International Conference on Parallel Processing, St. Charles, Illinois, Aug 1991.
- [7] Larsen, Ø.G., "Development and emulation of interaction mechanisms for a heterogeneous multi-computer, Dr Sci thesis, Dept of Informatics, Univ of Oslo, Oct 1991.
- [8] James, D., Laundrie, A., Gjessing, S. and Sohi, G., "Scalable Coherent Interface" IEEE Computer, June 1990.
- [9] IEEE, "The Scalable Coherent Interface", IEEE 1596-1992.
- [10] Nilsen, F.B., "Application of a switched interconnection concept for instrumentation at a high-energy physics experiment", Cand Sci thesis, Dept of Informatics, Univ of Oslo, May 1993.
- [11] Søråsen, O., Nilsen, F.B., Nygård, E. and Østby, J.M., "A Switched Interconnection Network for Large Scale Instrumentations", Atlas Internal DAQ Note 06, October 1992. RD-20 Note 10, October 1992
- [12] Søråsen, O., Lundh, Y., Nilsen, F.B., Nygård, E., Østby, J.M., "A High Performance Data Driven Packet-Switching Network for Detector Readout and Event Building in an LHC Inner Detector Experiment", IEEE Trans. on Nuclear Science, Vol. 41, No. 1, February 1994.
- [13] Lømo, M., Søråsen, O., "Switched interconnection of parallel processes (SWIPP) used for detector data readout and partial eventbuilding for LHC.", Poster at Computing in High-Energy Physics CHEP'94", San Francisco, CA, 21 - 27 April 1994.
- [14] Østby, J.M., Søråsen, O., "A packet-switched network for data readout from the LHC inner detector", Paper at Computing in High-Energy Physics CHEP'94", San Francisco, CA, 21 - 27 April 1994.



## **Dual Port Memory**

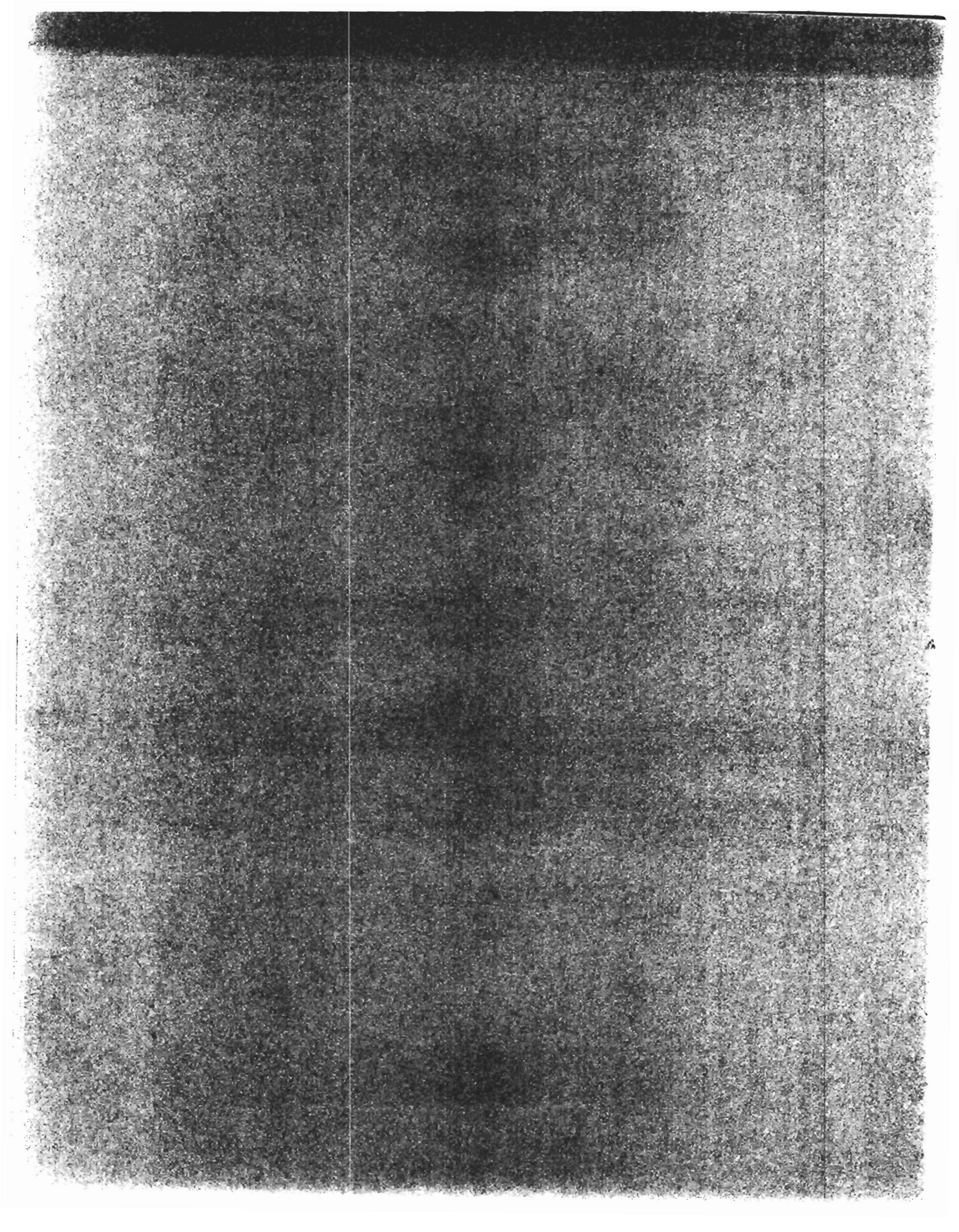
**Adolfo Fucci**

**CERN**

High-speed data buffering is required at different levels in the readout chain of LHC experiments. The basic unit is expected to be a programmable message-driven multiport memory (DPM) capable of handling high-speed rates of more than 200 MB/s between Input and Output combined.

At the present, prototype versions of the DPM could be used to evaluate readout components and event builder switches. RD12 (CERN/MIT) has a long experience of memory architectures and control. A simplified version of the DPM was already built with 1/2 MB of buffer memory and 400 MB/s of max rate for Input or Output (FDPM). It was used by several HEP institutions to test high-speed VLSI chips.

The current development effort is a prototype version of the DPM with 2 MB of buffer space and a very sophisticated memory event management able to perform the basic functions of the standard DAQ inner functions.



## **Performance Evaluation Tool for DAQ Computers (DAQBENCH)**

**Yoshiji Yasu (KEK)**

**National Laboratory for High Energy Physics (KEK)**

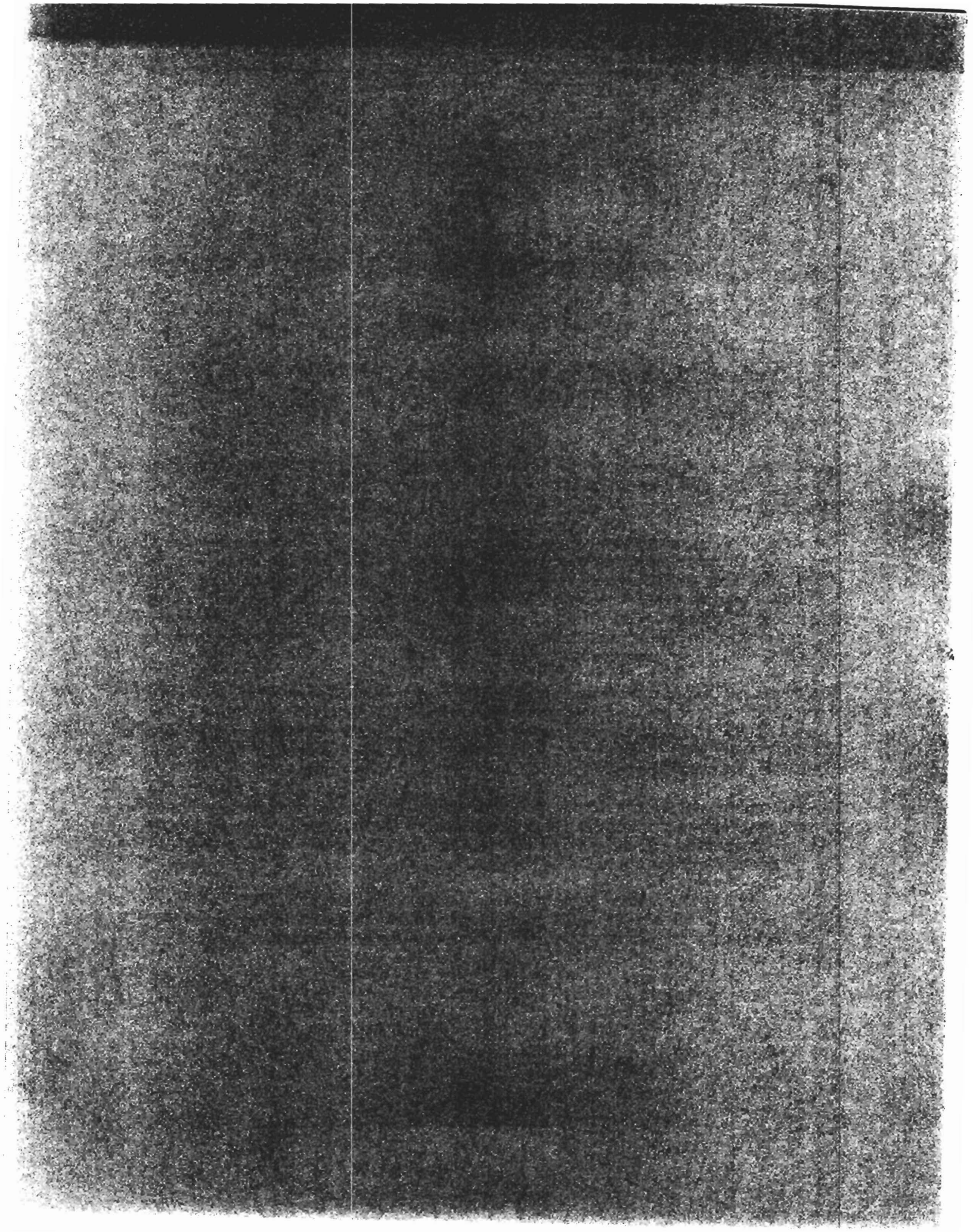
The DAQBENCH has been developed to evaluate computer performance for data acquisition. The benchmark results are useful to design data acquisition system and become the index to select DAQ computer.

The DAQBENCH has several kinds of benchmark programs to evaluate several DAQ parameters. There are programs to evaluate performance of Inter-Process Communications by using many types of IPC system calls which also include network functions and POSEK4 functions. From the benchmark results, overhead of the system calls, the context switching time and so on are evaluated. There is also a benchmark program to evaluate performance of copy functions from memory to memory on computer. The result depends on performance of not only CPU but also memory architecture.

To evaluate performance of VME & CAMAC accesses, there are three kinds of benchmark programs. One is for the single action. Another is for the block action. The other is for the interrupt handling. The benchmark program for the single action shows performance of the Programmed I/O. Performance of the Direct Memory Access is evaluated by that of the block action. Interrupt latency is measured by the interrupt handling program.

There are real programs for the data acquisition system to evaluate the performance. The kernel of the data acquisition system is the buffer manager. Performance in not only VME & CAMAC accesses but also the manager play an important role of the data acquisition system. The buffer manager NOVA used by UNIDAQ is evaluated.

Many computer systems have been evaluated. Namely, DECstation/ULTRIX, Alpha/OSFI, SUNSPARC/SUNOS, SUNSPARC/Solaris and HP742/HP-RT. Performances of HP735/HP-UX, i486/Linux and so on have been also measured except that of VME & CAMAC accesses.



Performance Evaluation Tool of DAQ Computer  
DAQBENCH

Yoshiji Yasu  
KEK, Oho 1-1, Tsukuba 305, Japan

Yasuhisa Tajima,  
Tokyo Institute of Technology, Meguro, Tokyo 152, Japan

ABSTRACT

This paper describes a DAQ performance evaluation tool called DAQBENCH. The tool evaluates the DAQ parameters which will be useful to design DAQ system and select DAQ computer. A famous benchmark suite, SPECmark is useful for high energy physics applications, but the benchmark suite is not convenient for evaluating the DAQ parameters. Recent real-time benchmark suite like Rhealstone can evaluate real-time parameters, but the suites does not include all of DAQ parameters. Those are the reason why DAQBENCH has been developed. DAQBENCH evaluates the DAQ parameters by measuring the performance of Inter-process communication (IPC), POSIX.4 IPC, Data Copy functions, Buffer Manager (NOVA), VME access and CAMAC access. Those parameters have been evaluated on the following computers, HP742rt/HP-RT, DEC3400/OSF1, DEC5125/ULTRIX, SPARC2/SunOS and SPARC2/Solaris. The part of the parameters has also evaluated on HP735 and i486DX2-66/LynxOS for the comparison.

1. INTRODUCTION and MOTIVATION

Why do we develop DAQBENCH? SPECmark is useful for high energy physics applications. Particularly, SPECint value is the most relevant performance indicator for standard HEP jobs[1] while CERN unit from CERN benchmark suite[2] corresponds about 4 times the SPECint value. A Monte Carlo simulation program, EGS4 code system[3] had also been evaluated in comparison with the CERN unit and the SPECint92[4]. However, the SPECint92 is not enough for evaluating the DAQ parameters because the context switch time which is one of the DAQ parameters, evaluated by DAQBENCH is not consistent to SPECint92 value. Fig. 1 plots the relation between SPECint92 value and the context switch cycles per second.

On the other hand, real-time benchmark program are discussed in real-time field. For example, Rhealstone benchmark suite[5] defines real-time parameters and can evaluate the parameters, but the suite does not include all of the DAQ parameters.

User and designer of DAQ system require;

1) Well defined DAQ parameters

2) Platform independence

3) Available distribution kit

DAQBENCH has been developed for those requirements.

DAQBENCH assumes that the DAQ computers should be based on VMEbus system with UNIX operating system including real-time UNIX[6].

2. Contents of DAQBENCH

DAQBENCH has several kinds of benchmark programs to evaluate DAQ parameters. There are programs to evaluate performance of IPC by using many types of IPC system calls which also include network functions and POSIX.4 functions. Those system calls are very important for synchronization of the process to correspond with each other. From the benchmark results, overhead of the system calls, the context switch time and so on have been evaluated. Fig. 2 shows the execution-time of semaphore and signal system calls on 7 types of computers and fig. 3 shows the time of FIFO, pipe and message queue system calls on the computers. Those results shows that the DEC5125 has good performance of semaphore with context switch, but the time



of FIFO, pipe and message queue without context switch on that is not so good.

There is also benchmark programs to evaluate performance of copy functions from memory to memory on computer. The copy function is used for event building to gather the pieces of events in scattered memory. Fig 4 shows Data Copy Performance on the computers. Memcpy system call has better performance than do-loop method and strncpy system call is not convenient for long message. The copy functions do not depend on only performance of CPU.

There are also real program for the data acquisition system to evaluate the performance. Core of the data acquisition software is buffer manager. Performance of the manager play an important role of the data acquisition system. The buffer manager called NOVA[7] used by UNIDAQ[7] is used. The round-trip time of data buffer on the processes handled by NOVA is shown in fig. 5. NOVA uses message queue system call for IPC.

To evaluate performance of VME and CAMAC accesses, there are three kinds of benchmark programs. One is for the single action. Second one is for the block action. The other is for the interrupt handling. The benchmark program for the single action shows performance of the Programmed I/O. Performance of the Direct Memory Access is evaluated by that for the block action. The interrupt task response time is measured by that for the interrupt handling. Performances of VME and CAMAC accesses are shown in table 1 and 2, respectively. Table 3 shows available computers for VME and CAMAC accesses used by DAQBENCH.

### 3. CONCLUSION

DAQBENCH has been developed. The tool evaluates the DAQ parameters which will be useful to design DAQ system and select DAQ computer.

The DAQ parameters have been evaluated by DAQBENCH on many computers including recent products with UNIX and real-time UNIX operating systems.

The distribution kit of DAQBENCH will be delivered from KEK in near future.

### REFERENCE

- [1]Sverre Jarp : RISC without RISK? an overview of current RISC systems used in HEP Batch, Proceedings of the International Conference on Computing in High Energy Physics '92, pp. 499-504 (1992)
- [2]Eric McIntosh : Benchmarking Computers for HEP, CERN CN/92/13
- [3]Walter R. Nelson, Hideo Hirayama and David W. O. Rogers : THE EGS4 CODE SYSTEM, SLAC-Report-265, (1985)
- [4]Y. Yasu et al., HEP Benchmark Program : Proceedings of the Third EGS4 Users' Meeting in Japan, KEK Proceedings 93-15 p83-104, December, 1993
- [5]Kar,R.P. and Porter,K., Rhealstone -- A Real-time Benchmarking Proposal, Dr. Dobbs, Journal 14(2);14-24, February, 1989
- [6]Y. Yasu et al., VMEbus based computer and Real-time UNIX as infrastructure of DAQ, Proceedings of the International Conference on Computing in High Energy Physics '94, San Francisco, U.S.A., April 21-27, 1994
- [7] M.Nomachi et al., UNIDAQ, Proceedings of the International Conference on Computing in High Energy Physics '94, San Francisco, U.S.A., April 21-27, 1994

Table 1. Performance of VME access

		HP742rt	DECS
WRITE access Speed from HOST to VME (MB/sec)	memcpy	7.1	3.6
	strncpy	7.1	1.1
	do-loop	7.1	3.8
READ access Speed from VME to HOST (MB/sec)	memcpy	4.5	1.2
	strncpy	3.8	0.3
	do-loop	4.3	1.1
Interrupt Task Response Time( $\mu$ sec)		72	450

HP742rt : HP742rt/HP-RT V1.1

DECS : DECStation 5000/125 ULTRIX V4.2A, DEC VMEbus adaptor

Table 2. Performance of CAMAC access

		HP742rt	Alpha	DECS	Sun-1	Sun-2	
Single Action( $\mu$ sec)	NDT	15	70	120	120	110	
	READ	22	96	160	125	130	
	WRITE	20	90	150	130	130	
Block Action	read	overhead( $\mu$ sec)	90	370	900	720	*
		speed(KB/sec)	980	1000	980	1020	*
	write	overhead( $\mu$ sec)	98	380	780	720	*
		speed(KB/sec)	940	530	430	810	*
Interrupt Handling( $\mu$ sec)		70	200	480	*	700	

HP742rt : HP742rt/HP-RT V1.1

Alpha : DEC3000/400 /OSF1 V1.3, DEC VMEbus adaptor

DECS : DECStation 5000/125 ULTRIX V4.2A, DEC VMEbus adaptor

Sun-1 : Sparc2/SunOS4.1.2 ; Sun-2 : Sparc2/Solaris2.3

\* means "not measured"

Table 3. Available Computer for VME&CAMAC

	HP742rt HP-RT	Alpha OSF1	DECStation ULTRIX	Sparc SUNOS	Sparc Solaris
CAMAC					
Single	O	O	O	O	O
Block	O	O	O	O	*
Interrupt	O	O	O	O	O
KEK list	O	O	O	O	O
Kinetic list	*	O	O	O	O
VME					
Map I/O	O	X	O	--	--
Interrupt	O	*	O	--	--

HP : HP742rt(VME board computer)

DEC : DECStation3000 with DEC's VMEbus Adaptor

DEC3000/400(Alpha AXP) with DEC's VMEbus Adaptor

SUN : Sparc IPC, Sparc IPX, Sparc 2, Sparc10, Sparc classic with SFVME(VMEbus Adaptor)

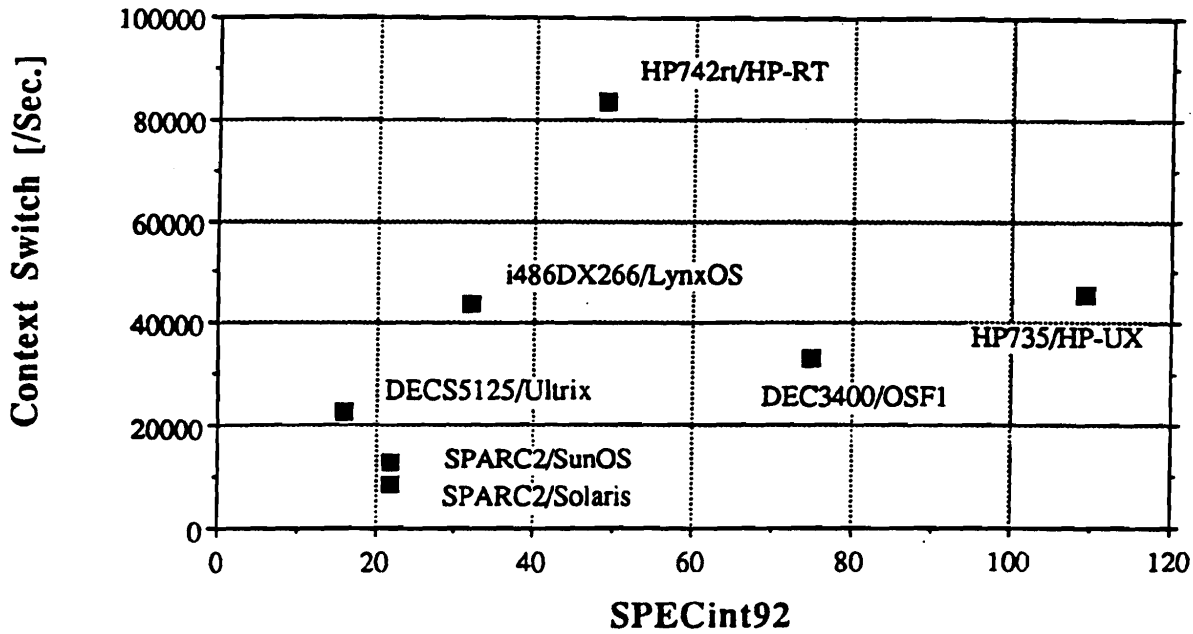
Sparc2E(VME board computer)

VME-CAMAC interface : Kinetic 2917

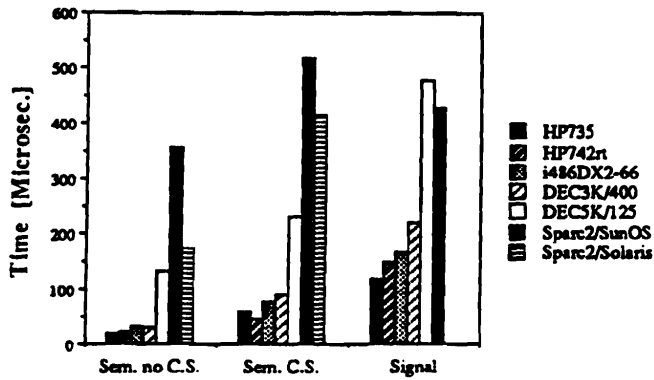
"O" means "supported"; "\*" means "to be supported";

"X" means "no plan"; "--" means "not scheduled";

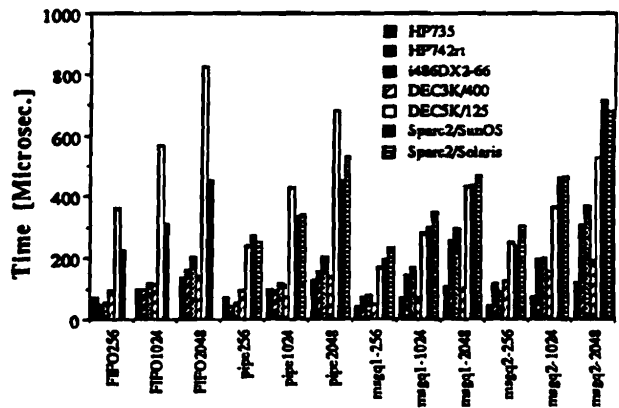
# Figure 1. Context Switch v.s. SPECint92



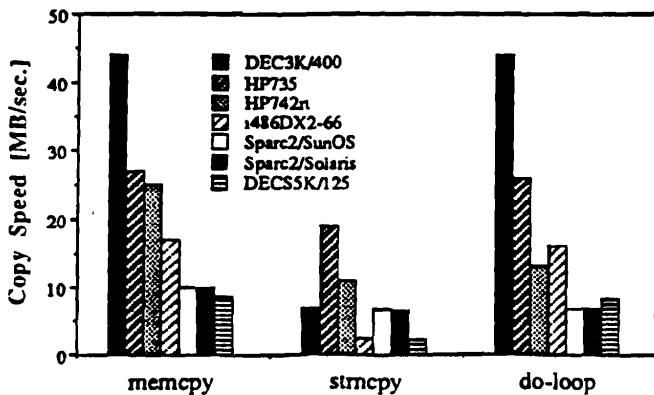
## Figure 2. Time of Semaphore and Signal



## Figure 3. Time of FIFO, pipe and message queue



## Figure 4. Data Copy Performance



## Figure 5. Round-trip Time of Buffer Manager (NOVA)

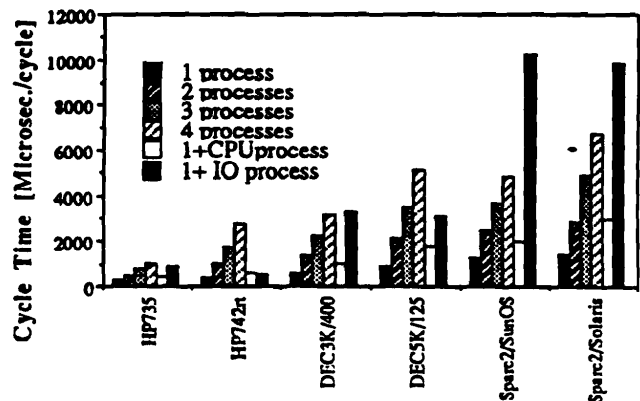


Figure 1. Context Switch v.s. SPECint92

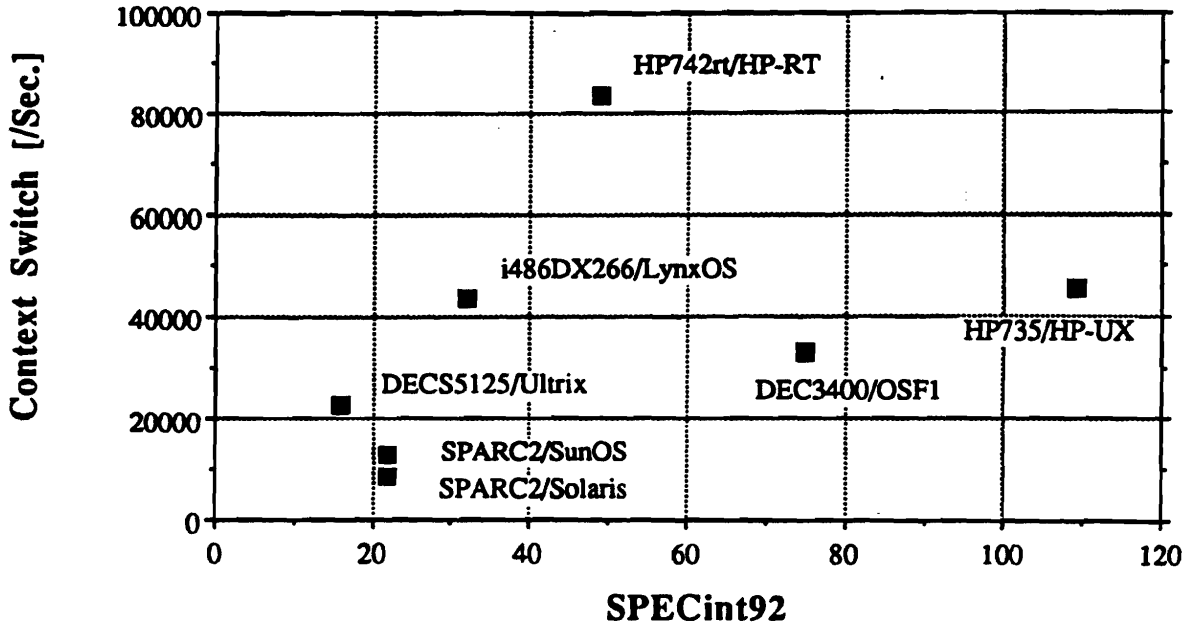


Figure 2. Time of Semaphore and Signal

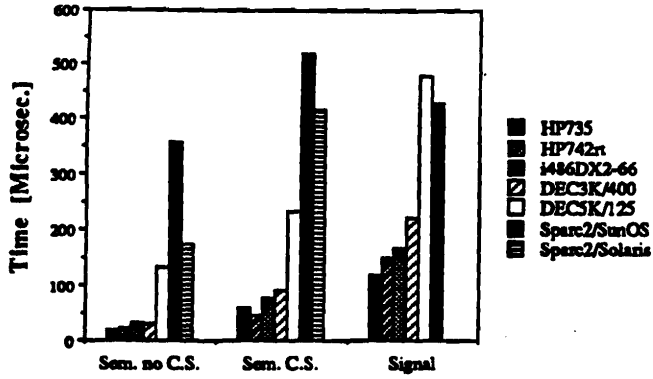


Figure 3. Time of FIFO, pipe and message queue

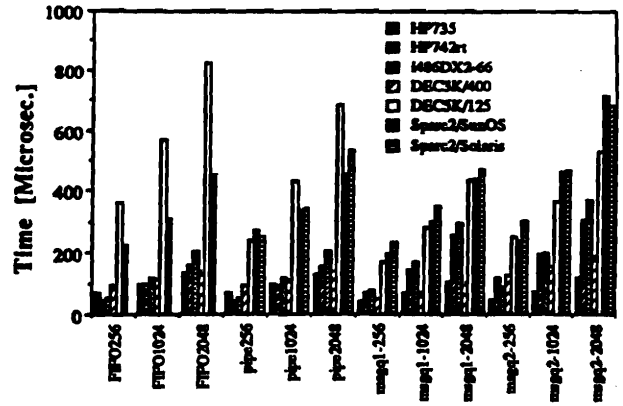


Figure 4. Data Copy Performance

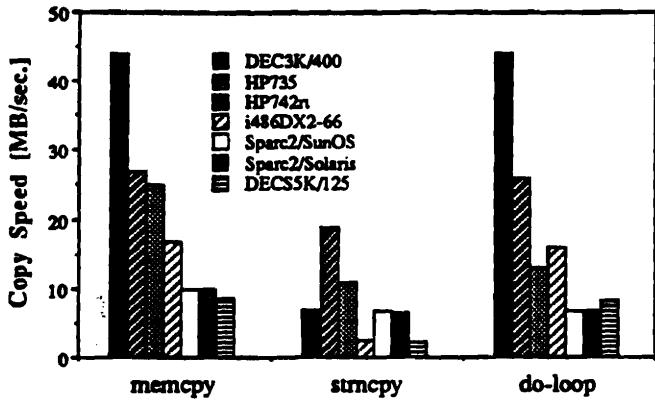
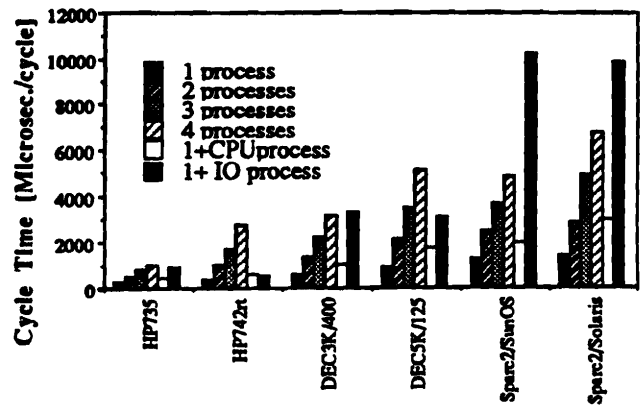


Figure 5. Round-trip Time of Buffer Manager (NOVA)

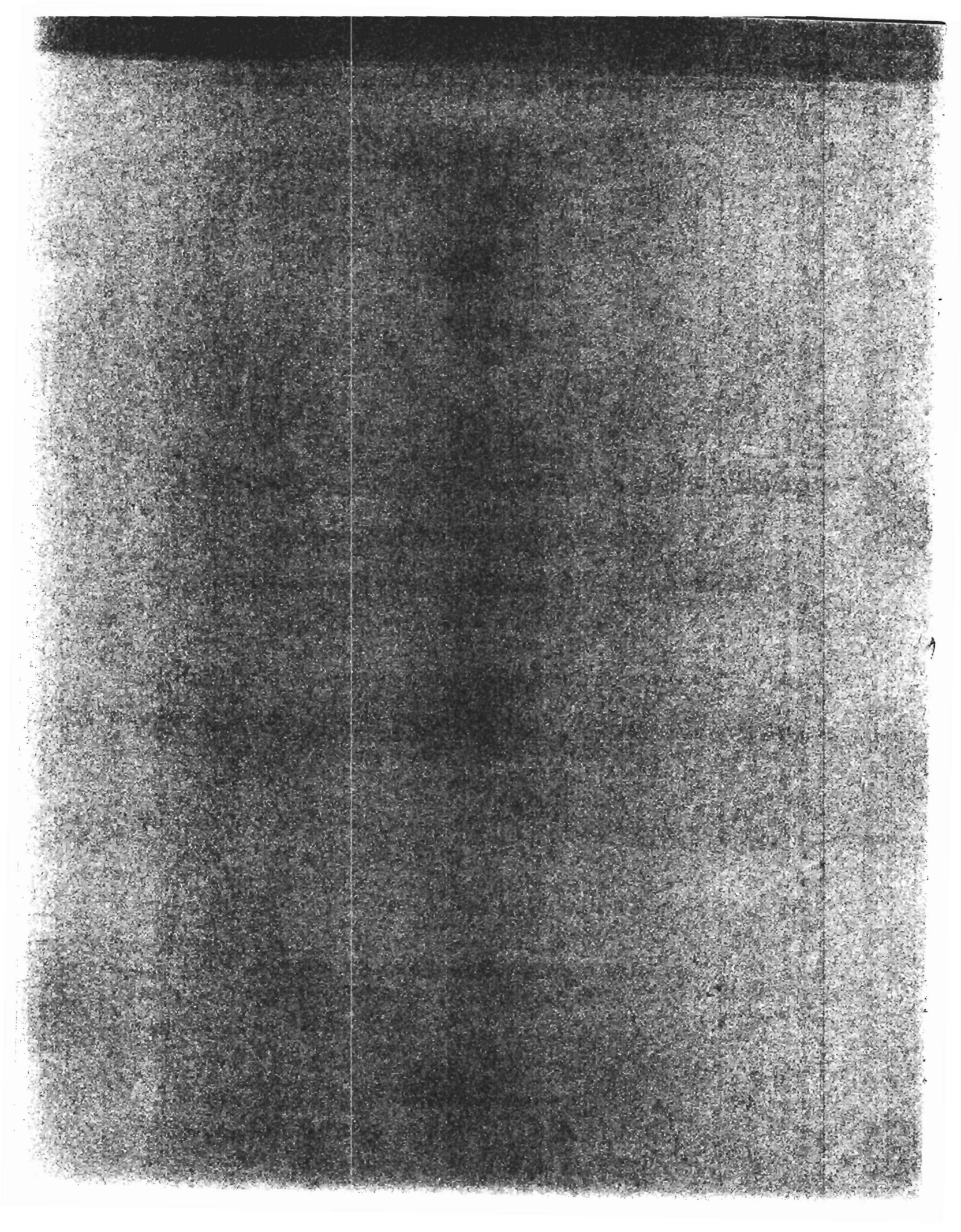


## **Global Traffic Control System on High Speed Event Builder**

**Y. Nagasaka, O. Sasaki, and M. Nomachi**

**National Laboratory for High Energy Physics (KEK)**

A Unique traffic shaping scheme has been proposed for a event builder. It is called "Global Traffic Control (GTC)" system. It was found that large scale event builders such 1,000 x 1,000 system were designed with just the same scheme as small ones using transparent switches. The prototype event builder based on GTC is being developed.



# Global Traffic Control System on High Speed Event Builder using Transparent Switches

M.Tairadate

*Grad. Univ. for Advanced Studies, Japan*

Y.Nagasaka, O.Sasaki and M.Nomachi

*KEK, Japan*

## Abstract

An unique traffic shaping scheme has been proposed for event builder. It is called "Global Traffic Control(GTC)" system. It was found that large scale event builders such as  $1,000 \times 1,000$  system were designed with just the same scheme as small ones using transparent switches. The prototype event builder based on GTC is being developed.

## 1 Introduction

There has been remarkable progress on the developments of data processing and data transfer hardware. The performance of DAQ systems have got a lot of improvements. However, the increase of the requirements on large experiments exceeds that progress. For example, the data rate of LHC experiments [1] [2] exceeds a few GByte/sec. Therefore, distributed processing is indispensable in the future large experiments.

The bottleneck on distributed systems is the network to distribute and to collect the data. We have studied the parallel event builder using a switching network. An event builder must handle heavy data traffic since the data from all detector subsystems has to be collected in one place. If each processor connects directly to all detector subsystems, many links are required. To replace these connections by switching network makes possible to large scale event building[3]. It can cope with the scalability increasing of the number of input/output ports.



## 2 Switching Network for Event Builder

There are roughly two types of switching networks. One type of switches is that data has an information of its destination. Since data appears to select its route by itself, this type of switches is called "Self Routing(SR)" switch. ATM, Fibre Channel and SCI switches are included in this type. The other type of switch is that data does not have an information of its destination. Since data go through the route selected by external controller, this type of switch is called "Global Traffic Control(GTC)" switch.

On a telephone line, there is little concentration of data on specific node. Then there is nothing congestion at the end of switching network. SR system is efficient in light traffic such as telephone line. On an event builder, there is concentration of data from most detector subsystems to each destination node. If it was not for anything care, the congestion at the end of switching network must have happened. The larger switch size is, the more serious this problem is. GTC system makes the data flow without congestion possible even in such a heavy traffic.

## 3 Global Traffic Control and Transparent Switch

The data flow of event builder is shown in Figure 1. The input queues are separated according to destinations. Each source node has  $M$  queues.  $M$  is the number of destination nodes. Each destination node has  $N$  queues.  $N$  is the number of source nodes. Event fragments coming from a detector subsystem are divided into input queues in order. Then they are transferred to decided destinations respectively. Each event fragment doesn't have to have an information of its destination.

In figure 1 if each input queue is linked physically to output queue,  $N \times M$  of transmitters, links and receivers are needed. In case of  $1,000 \times 1,000$  system, a million of them are required. Each input queue is linked virtually to output queue by time sharing of physical links. The number of physical links is larger one of  $M$  and  $N$ . From the point of view of data, the switch is seemed to be transparent. Such switching of links is handled by the external controller.

All event fragments are transferred to output queues without congestion. Each link is *independent* because of no affection from others. Analysis of one virtual link would suffice in a large scale switch. GTC system makes event builder scalable.

## 4 Occupancy of Input Queue

Occupancy of an input queue is analyzed by queuing theory. Suppose an event interval on a source node has exponential distribution, an event interval on an input queue has  $k$ -Erlang distribution where phase  $k$  is the number of input queues on an source node. Its probability function is given in the following:

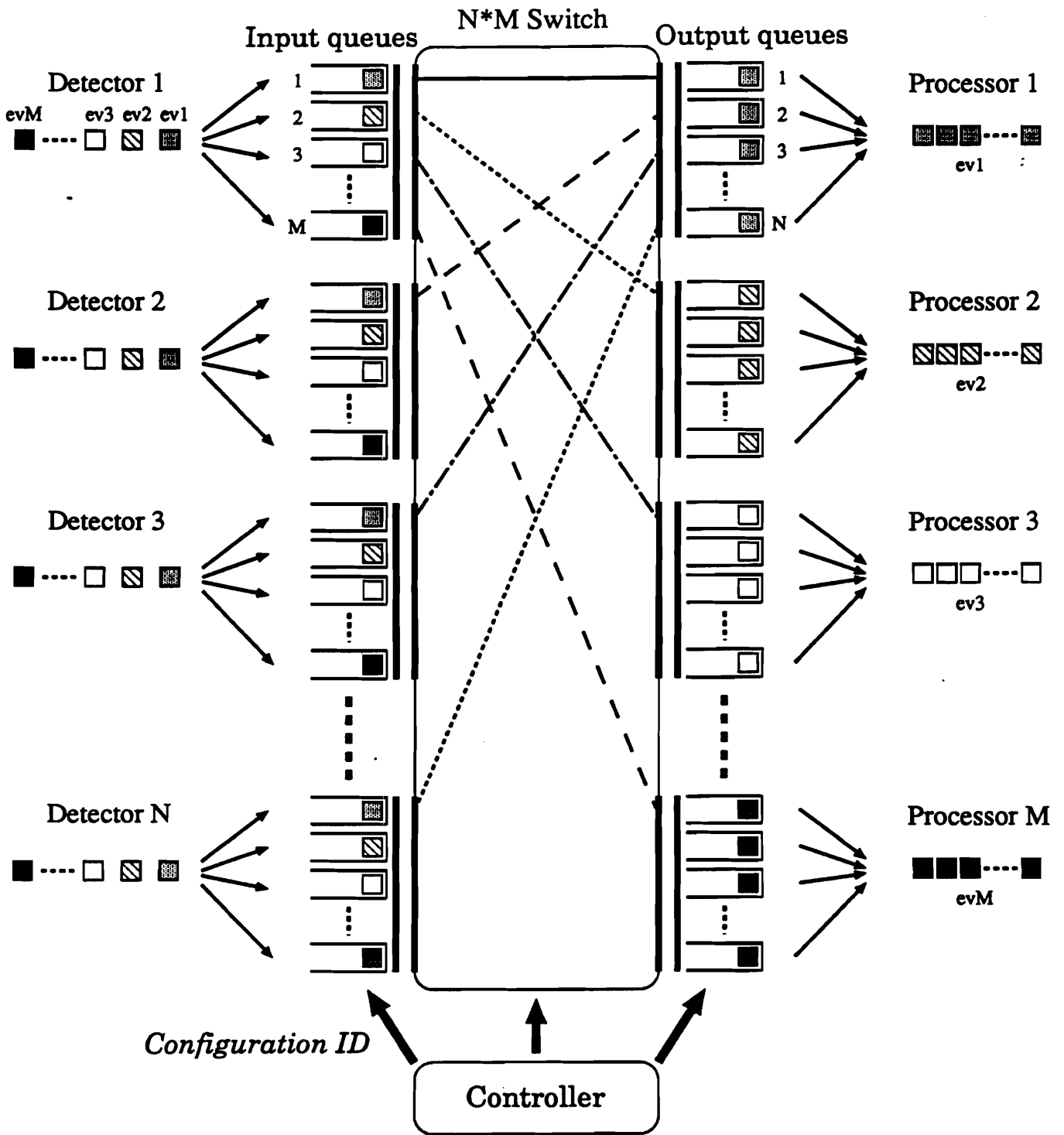


Figure 1: Data flow of event builder

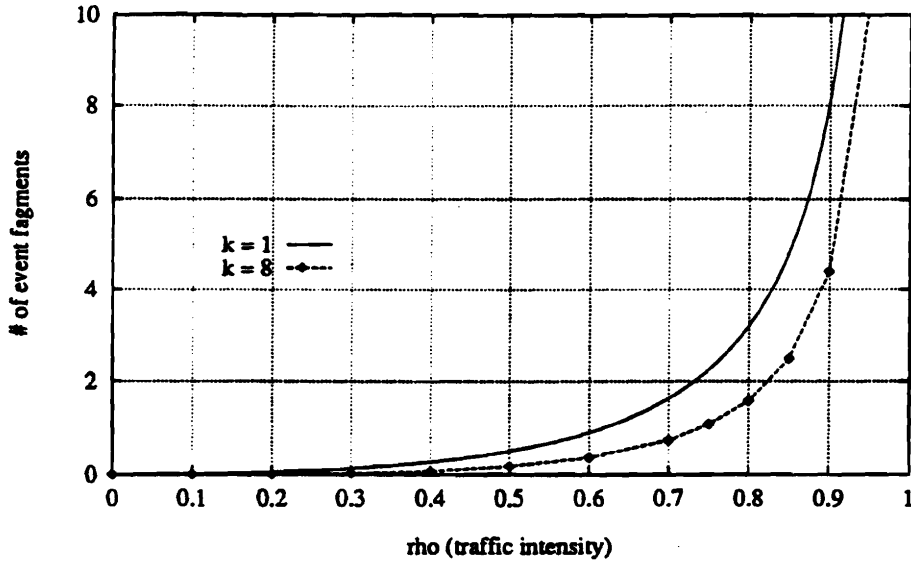


Figure 2: Average number of event fragments in an input queue

$$f(x) = \frac{(\lambda k)^k}{(k-1)!} x^{k-1} e^{-\lambda k x} \quad x \geq 0, k \geq 1, \lambda > 0 \quad (1)$$

where  $\lambda$  is the trigger rate. Each event fragment size is assumed to have exponential distribution. Then data transfer time has exponential distribution. Average number of event fragments in an input queue ( $L_q$ ) is represented as a function of traffic intensity ( $\rho$ ).

$$L_q(\rho) = \frac{\rho u_0^k}{1 - u_0^k} \quad (2)$$

where  $u_0 (0 < u_0 < 1)$  is solution of the following equation:

$$u^{k+1} - (k\rho + 1)u + k\rho = 0. \quad (3)$$

Traffic intensity is defined as follows:

$$\rho = \frac{\nu \lambda}{b} \quad (4)$$

where  $\nu$  is the average event fragment size and  $b$  is the bandwidth. Figure 2 shows  $L_q$  as a function of  $\rho$  when  $k$  is 1 and 8, respectively. If the traffic intensity exceeds 90%, average number of event fragments in an input queue increase dramatically. In addition, from this figure, required buffer size can be determined from trigger rate and event fragment size. The latency of this system is discussed in [4]. Even in case of large scale switch (= larger  $k$ ), suitable buffer size is predictable.

## 5 Summary

In this article, basic concept of GTC system was showed. It was found that large scale event builders such as  $1,000 \times 1,000$  system were designed with just the same scheme of ours as small ones. In such a large system, it is difficult to build the event with no congestion if a SR switch such as ATM is used. On the other hand, GTC switch would not be a problem because of its scalability. GTC is useful for large scale event builders.

We designed a prototype event builder of 8 by 8 switch. The switch modules have been developed already[5]. Each data link has 1 Gbps data transfer speed. This specification is sufficient for KEK B-factory. More detail simulation results about queue occupancy will be presented in the future.

## References

- [1] ATLAS collaboration, "Letter of Intents for a General Purpose pp Experiment at the Large Hadron Collider at CERN", CERN/LHCC/92-4, October 1992
- [2] CMS collaboration, "Letter of Intents for a General Purpose Detector at the LHC", CERN/LHCC/92-3, October 1992
- [3] D.Black *et al.*, "Results from a Data Acquisition System Prototype Project Using a Switch-Based Event Builder", 1991 IEEE NSS, Santa Fe.
- [4] M. Nomachi, "Event Builder Queue Occupancy", SDC-93-566 August 1993
- [5] O. Sasaki *et al.*, "A VME Barrel Shifter System for Event Reconstruction for up to 3 Gbps Signal Trains", IEEE Trans. Nucl. Science Vol.40, No.4, August 1993

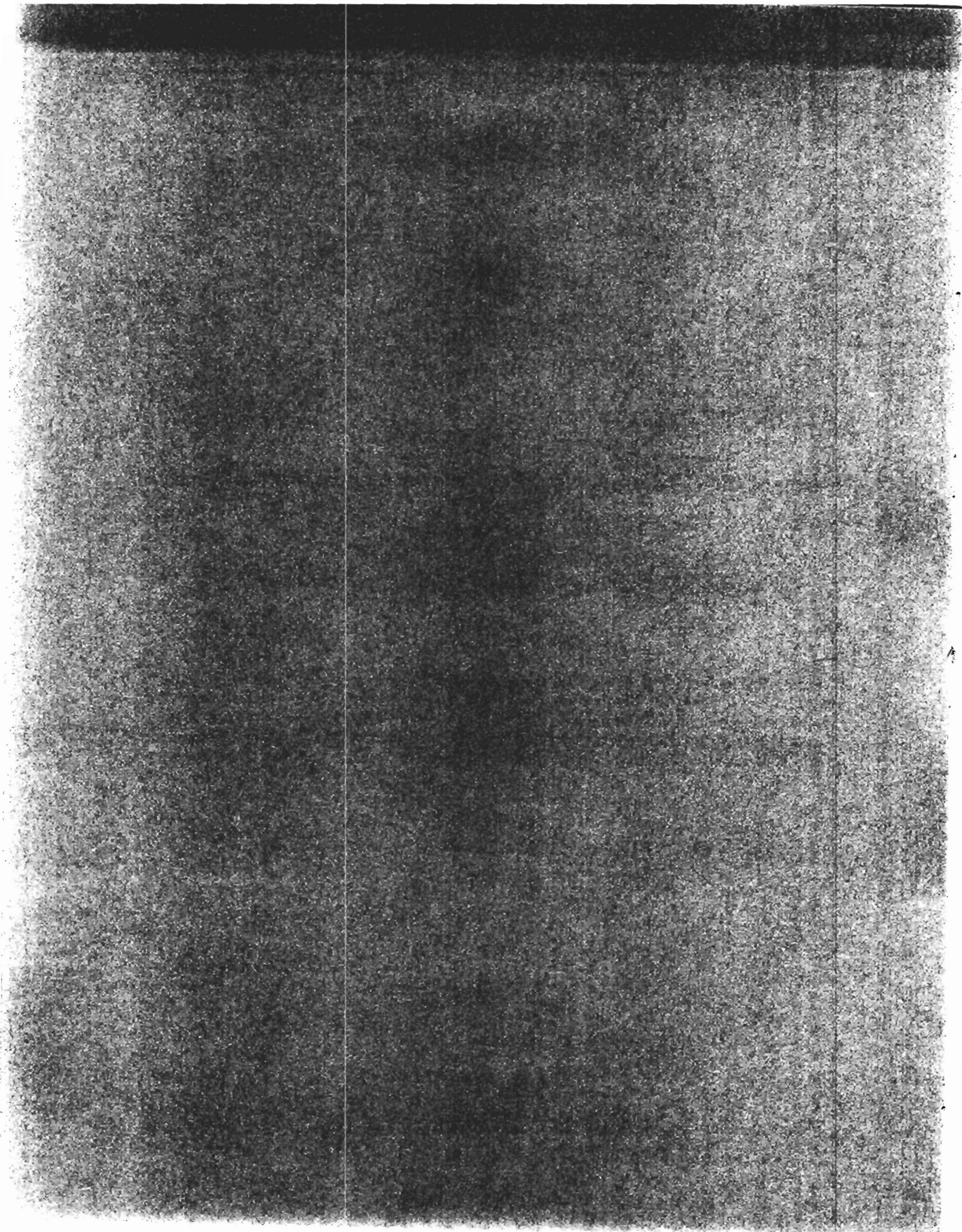


## **Testing of the HP G-link Chip Set for an Event Builder Application**

**O. Sasaki**

**National Laboratory for High Energy Physics (KEK)**

The Hewlett Packard HDMP-1000 G-Link transmitter and receiver chip set was tested for an event builder application. The re-lock time of the serial link when a data path is changed is less than 30 usec provided fill frames are transmitted. With a 1kHz data path switching rate, this results in less than a 3 % data rate inefficiency due to re-synchronization which makes the HP G-link chip set effective for use in a large event builder for a hadron collider experiment.



# Testing of the HP G-Link Chip Set for an Event Builder Application

Osamu Sasaki<sup>1</sup>, Jeffry Andresen, Hector Gonzalez, Masaharu Nomachi<sup>1</sup>, and Ed Barsotti  
Fermi National Accelerator Laboratory  
Batavia, IL. 60510

## Abstract

The Hewlett Packard HDMP-1000 G-Link transmitter and receiver chip set was tested for an event builder application. The re-lock time of the serial link when a data path is changed is less than 30  $\mu$ s provided fill frames are transmitted. With a 1 kHz data path switching rate, this results in less than a 3% data rate inefficiency due to re-synchronization which makes the HP G-Link chip set effective for use in a large event builder for a hadron collider experiment.

## I. INTRODUCTION

A large high energy physics detector requires a large scale data acquisition system which could require several thousand data links at a speed of several tens of MBytes per second per link. A transparent switch network with global traffic control being used as an event builder for such a large experiment has already been proposed [1]. The event builder receives event data fragments from many sources via serial links. The transparent switch performs the event reconstruction by the switching of the serial data links through the use of global traffic control. One of the most important hardware parameters of an event builder switch is the time that is taken to re-configure the switch connections and to recover a synchronized clock to a new data stream. This re-synchronization time needs to be a small percentage of time in comparison to the switching interval to achieve sufficient data throughput.

The Hewlett Packard HDMP-1000 G-Link chip set was selected to be tested as it is one of the candidates for a serial data link protocol chip set [2]. The purpose of this test was to understand what other design parameters have to be satisfied to build a switch based on using G-Links. The HDMP-1000 consists of a HDMP-1002 (transmitter) and a HDMP-1004 (receiver). The transmitter serializes either 16 bits or 20 bits of parallel data, adds 4 coding bits, and transmits the data at a serial speed as high as 1.4 Gbps. The G-Link receiver converts the serial data to its original parallel form. The G-Link chip set has three kinds of frames (data sets) which are a data frame, a control frame, and a fill frame. The transmitter sends fill frames if Data Available (DAV\*) and Control Word Available (CAV\*) are false or if Enable Data (ED) is false. Data frames and control frames are treated in the same way by the G-Link chip set with the control frames providing a way for the user to differentiate between control bits and data bits. Fill frames are sent by the transmitter at start up, whenever data frames or control frames are not being sent by the user, and whenever there is an error condition. A fill frame has the same duration as a data frame or a control frame with one master transition to allow the receiver to acquire frequency lock (both frame

synchronization and bit synchronization). Once the receiver has frequency lock, data transmission can begin. The 4 coding bit field that is sent with the data has one master rising/falling edge that the receiver's Phase Lock Loop (PLL) circuitry uses to maintain bit synchronization as it recovers the clock from the serial data stream. This phase lock has a narrow frequency detection range. If phase lock is lost, fill frames are transmitted until re-lock can occur through the use of frequency lock. It should be noted that the receiver can never be re-locked when data or control frames are being received. The G-Link system maintains DC balance on the serial data line by inverting the transmitted data or control frame whenever necessary. Figure 1 is HP's example of a simplex G-Link configuration.

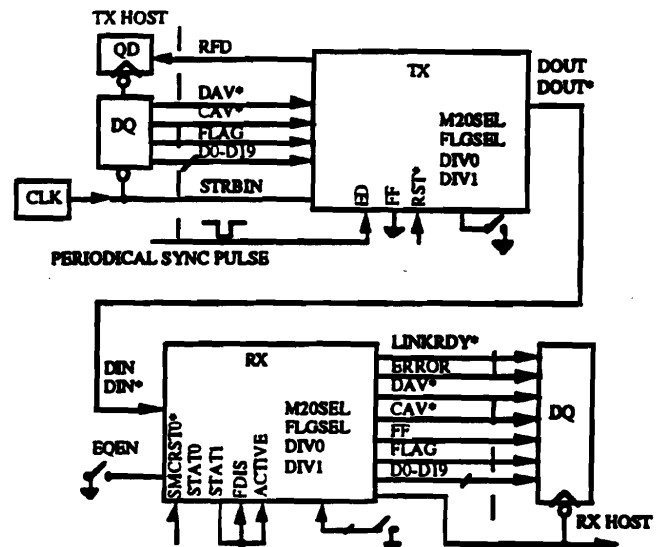


Figure 1: HP's Simplex Configuration Example

## II. TESTS

### A. Testing Hardware

This testing utilized a HDMP-1000 G-Link evaluation board simulating a simplex configuration [3]. In the following tests, all serial signals were transmitted over 50 $\Omega$  SMA coaxial cable. The G-Link TX device on the evaluation board was clocked from the STRBIN input by an external pulse generator at 42 MHz. This frequency corresponds to a 1 Gbps bit stream when the 20 bit data transfer mode is selected. The KEK PECL 4X4 switch [1] was used to receive a 1 Gbps bit stream from the HP HDMP-1002 TX device. PECL fanout/level adapter boards were used to adapt the G-Link logic levels to PECL levels since the serial outputs from the TX are not ECL but are buffer line logic (BLL) [2]. A clock divider board and a NIM signal discriminator provided a 1 kHz NIM level pulse that was used to change the KEK switch

<sup>1</sup> National Laboratory for High Energy Physics (KEK)  
Tsukuba, Ibaraki-Ken, 305 Japan



configuration. The bit stream exited the switch and was sent to the HP HDMP-1004 RX device. A 20 bit random data pattern when data was being transmitted. The eye patterns of the serial bit stream and the receiver strobe out signal were observed on a sequential sampling oscilloscope. The re-lock time was measured by observing the Error and Link Ready signals from the RX on a digital oscilloscope.

### B. Random Data Pattern Test

A 20 bit random data pattern, along with the 4 coding bits that the G-Link generates, was transmitted to the switch at 1 Gbps. As can be seen in figure 2, the oscilloscope eye pattern clearly shows the 4 coding bits along with the data bits.

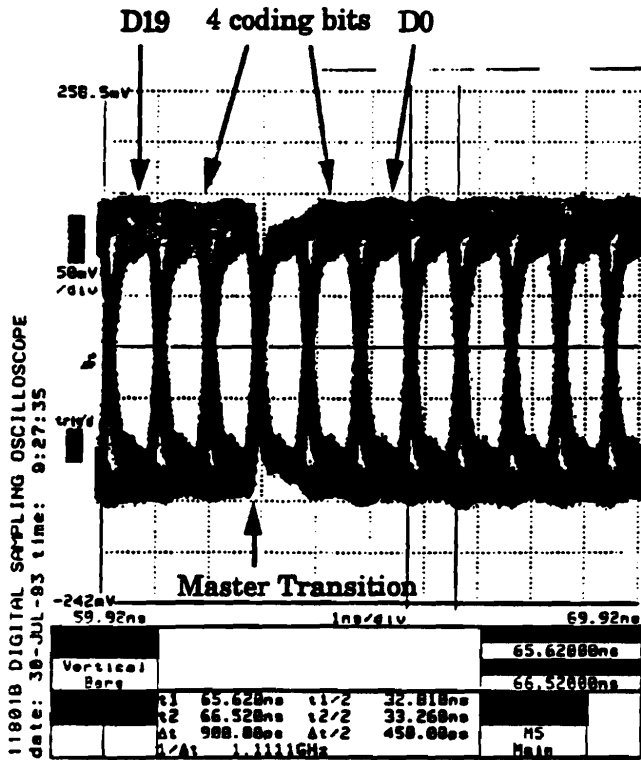


Figure 2: Coding and Data Bit Eye Pattern

The coding bit field has a master transition of a single rising/falling edge which serves as a fixed timing reference for the G-Link receiver's clock recover circuit. For this test in order to send a continuous random data pattern regardless of the lock condition, DAV\* and ED were always enabled through the use of jumpers. In a non-switching mode, the G-Link chip set transmitted and received the 20 data bit pattern without losing lock. As a result, there was no re-lock time to measure. When the switch was in its 1 kHz switching mode, the G-Link receiver would lose lock and never re-lock. For the RX G-Link to synchronize with the TX G-Link when only one clock source is used in the simplex configuration as shown in Figure 1, the G-Link system requires the TX to send fill frames to the RX G-Link.

### C. Fill Frame Phase Shift Test

In this test, the G-Link evaluation board transmitted and received a constant fill frame. The phase of the serial signal was changed along with the timing of the changing of the switch in relation to the fill frame. The top scope trace in figure 3 shows the G-Link fill frame.

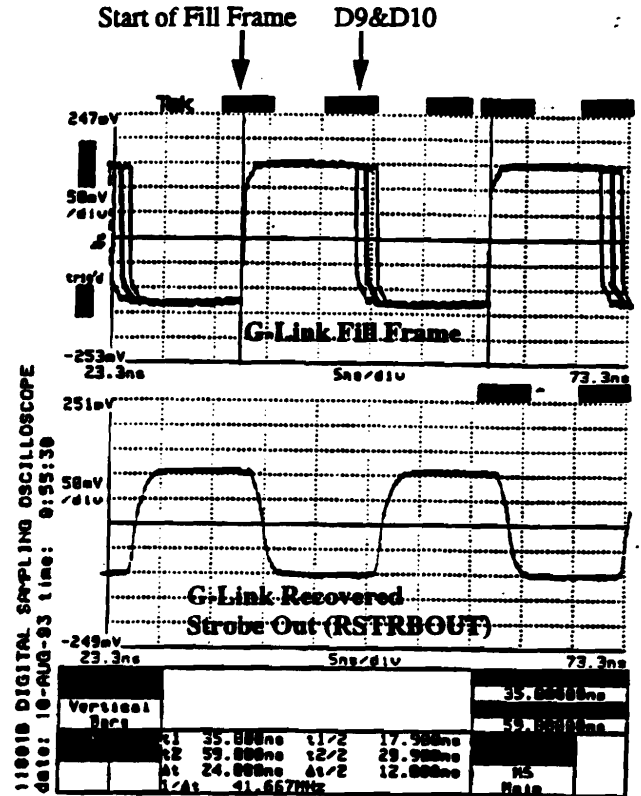


Figure 3: G-Link Fill Frame

There are two types of fill frames [4]. At startup, fill frame FF0 is transmitted which has a single falling edge in the data field going from a high D9 to a low D10. Once frequency lock occurs, fill frames FF1L and FF1H are transmitted. With the FF1 fill frames, the position of the falling edge in the data field is shifted forward or backward by one bit. This is accomplished by toggling data bits D9 and D10. FF1L transmits zeros for D9 and D10, and FF1H transmits ones for D9 and D10. The transmitter sends either FF1L or FF1H to reduce the cumulative serial DC offset. FF0 maintains DC balance as it is a square wave with a 50% duty cycle. The top scope trace is the fill frame. The rising edge of the fill frame is used by the receiver to achieve frequency lock. The lower scope trace in figure 3 is the receiver's strobe output (RSTRBOUT). This is the clock that has been recovered from the receiver's serial input. The TX serial link was connected to the input of a PECL fanout board. Figure 4 is a block diagram of the test setup.

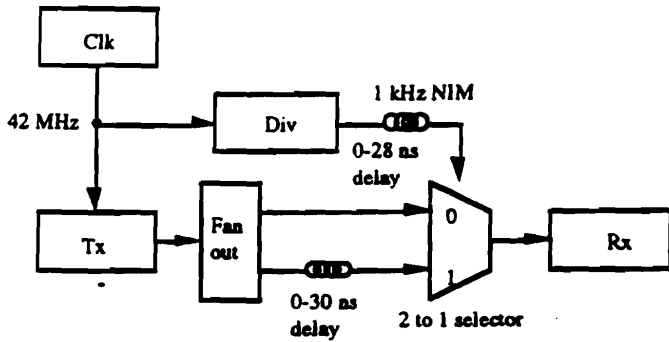


Figure 4: Cable Delay Test Configuration

The PECL KEK 4X4 switch was used as a "2 to 1 SELECTOR" which changed the signal path every millisecond. The serial data cable lengths from the fanout to the switch were varied to create signal delays. The delay was incremented in steps of 5 ns for the serial signal from the fanout. Delay time is 5 ns per meter of cable. Delays in 4 ns increments were added to the 1 kHz signal for the selector switching to change where in the fill frame the changing of the switch occurs. There are two parameters affecting the re-lock time in the configuration of figure 4. The first is the switch changing the signal path which shifts the phase of the serial signal. The second parameter is where within the fill frame the changing of the switch occurs. The re-lock times were measured as a function of these two parameters. Figure 5 is an example of a re-lock time reading.

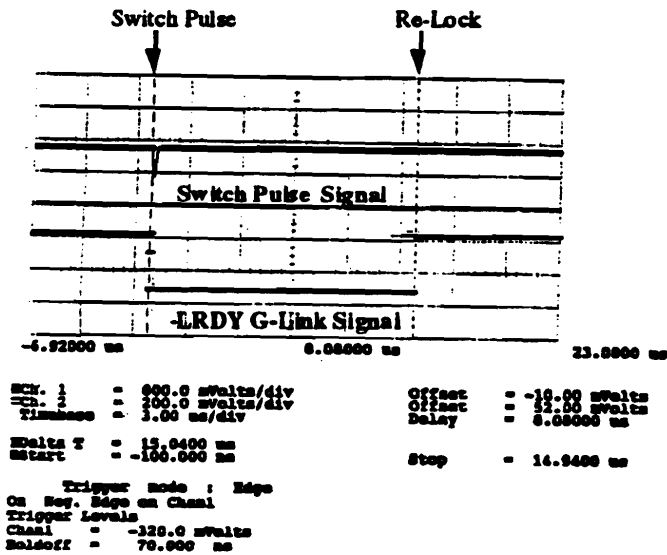


Figure 5: G-Link Re-Lock Time

To begin the test, all serial data cables were the same 1 meter length. The G-Link evaluation board operated without losing lock which indicated that the switching did not create errors when the phase difference between the two serial signals is zero. The switching from a 0 ns delay to a 24 ns serial data delay results in a zero phase shift as the frame has a 24 ns period. The system did not lose lock when this 24 ns delay was tested. The system did lose lock as the 5 ns delays steps

where introduced. The re-lock time was as short as 8.1 us and as long as 27.4 us. Adding delay to the 1 kHz NIM signal did not affect the maximum re-lock time. Figure 6 is a graph of the re-lock time in relationship to the delay time.

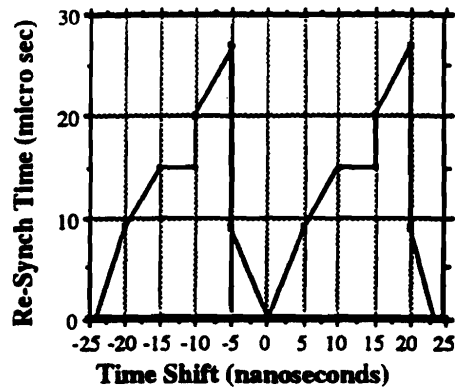


Figure 6: Re-Lock Time vs. Phase Change

Figure 6 shows that if the clock is in the correct phase, the G-Link operates without losing lock. The negative time readings were when the switch went to a shorter cable length. The maximum re-lock time was 27.4 us. This maximum re-lock time did not come when the phase difference was the greatest. The -5 ns and 20 ns points are sensitive points in that either a small re-lock time or the maximum time occurs at these two points. The -10 ns and 15 ns delays also produced two different re-lock times, but the difference between the times were smaller. The re-lock times in figure 6 are in a pattern that repeated in succeeding frames.

#### D. Fill Frame Frequency Change Test

Figure 7 is a diagram of the frequency change test.

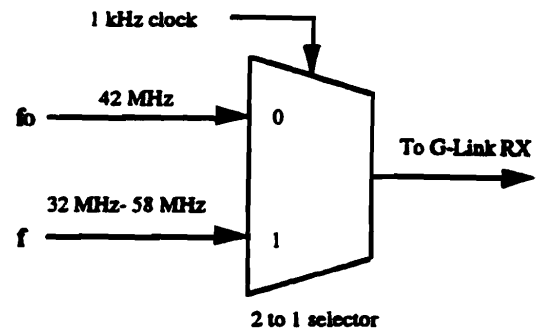


Figure 7: Frequency Change Test

This test did not use the G-Link TX device. The first clock generator was connected to one input of the switch with a fixed frequency of 42 MHz. A second clock generator was connected to another input of the switch with a frequency that was changed in incremental steps on both sides of the 42 MHz frequency of f<sub>0</sub>. Since the G-Link RX device recovers its clock from the serial stream, the 32 MHz to 58 MHz frequencies are in the same frequency range as the fill frame

rate that the RX can receive. In other words, the two clock generators transmit similar serial signals as that of the fill frame signals of the TX. The G-Link RX device did not lose lock when the two switch inputs were at the same frequency of 42 MHz without a phase difference. Figure 8 shows that as the difference in frequency increased, the re-lock time increased.

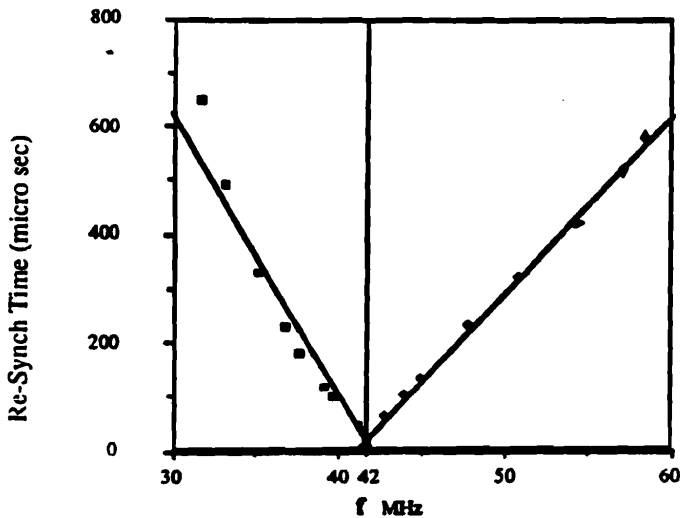


Figure 8: Frequency Change Re-Lock Time

In Figure 8,  $f_0$  is the fixed frequency of 42 MHz while  $f$  is the frequency being varied from 32 MHz to 58 MHz. As the difference in frequencies becomes greater, the re-lock time becomes greater. The sequence of switching from  $f_0$  to  $f$  or from  $f$  to  $f_0$  resulted in the same re-lock times. Figure 9 is a diagram of the clock on/off test.

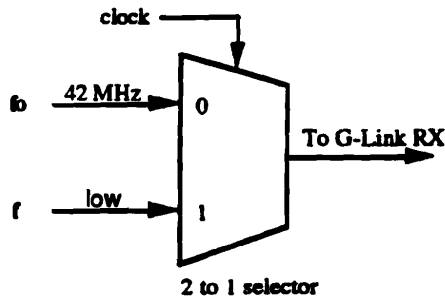


Figure 9: Clock On/Off Test

When the switch input connected to the low level was sent to the RX, the PLL circuit was disabled. When the signal was then switched back to the 42 MHz frequency, the resulting re-lock time was 1.8 ms.

### E. Simplex Dual Clock Data Pattern Test

In the simplex configuration with one clock generator as shown in Figure 1, when RX unexpectedly loses lock, RX can not re-lock without receiving fill frame signals from the TX. This configuration does not provide the TX a way of knowing that RX lock has been lost and that fill frames need

to be transmitted. Figure 10 is a diagram of a simplex data pattern test with dual clocks that was done. The RX in this dual clock simplex configuration has the ability to recover the lock by itself.

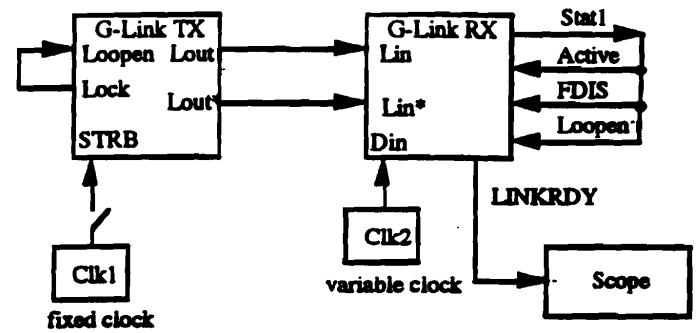


Figure 10: Simplex Dual Clock Data Pattern Test

In this test, a 1 Gbps data pattern (a data frame) was sent from the TX G-Link to the RX G-Link. The TX and the RX have dual ports for the serial signal. The Loopen signal controls whether the Dout or Lout output and the Din or Lin input are currently enabled. When Stat1 is low, Dout and Din are active. Clk2 provides the frequency pattern for the receiver to lock upon. When lock occurs, Stat1 goes high which activates Lin instead of Din. The TX Lock output was connected directly to the TX Loopen input. A fixed precision crystal clock was used for the transmitter clock. A HP 8110A pulse generator was used as the variable receiver clock as it has better than 0.1% stability, period steps of 10 ps, and duty cycle steps of 0.1%. The fixed clock was disabled and then enabled with a second pulse generator. When the TX clock is disabled, TX Lock and TX Loopen go low which disables the Lout and Lout\* outputs. This results in the RX losing lock and switching to the Din input. The RX clock then supplies a fill frame like clock for the receiver to re-lock upon. Fixed clock frequencies of 40.0000 MHz and 25.002 MHz were tested with Figure 11 showing the 40.0000 MHz re-lock times. The 25.002 MHz test produced similar results.

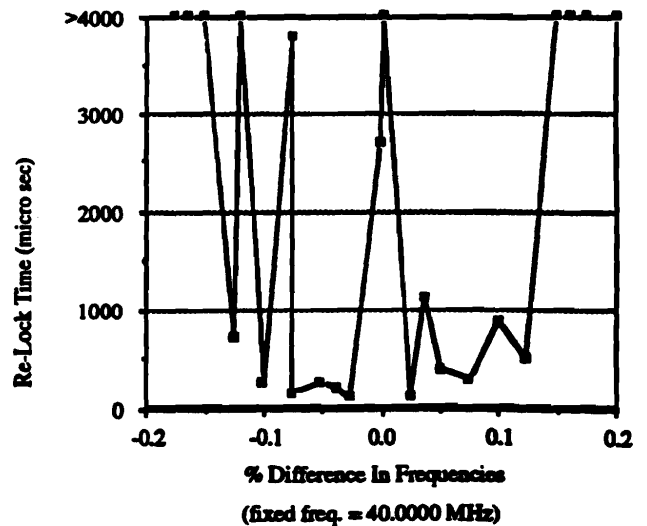


Figure 11: Simplex Test Re-Lock Times

Figure 11 indicates that the two clocks need to be within the 0.1% of having the same clock frequencies that HP specifies [6] and agrees with the test that was done at LBL [5]. The points at the top of the graph are actually points going beyond the graph indicating that re-lock did not occur at those test points. When the same clock generator was used for the transmitter and receiver in the figure 10 setup, there was no difference in frequencies with re-lock never occurring. When the two different clocks were within a 0.1% difference in clock frequency, the G-Link regained lock provided the duty cycle of the variable RX clock was 50%. There were duty cycles that would cause the RX G-Link to never re-lock. Figure 12 is a diagram of the test setup that was used to test the duty cycle of the receiver's clock.

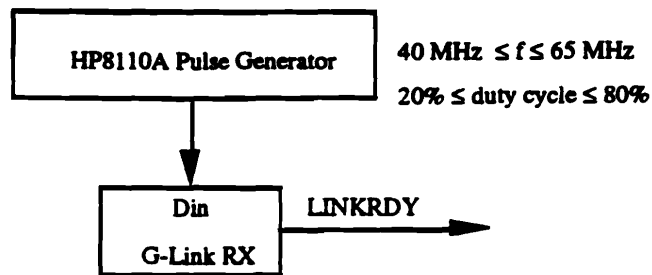


Figure 12: Simplex Receiver's Clock Duty Cycle Test

The HP 8110A pulse generator provided a stable clock at the selected test frequencies. The duty cycle of the clock was changed in 0.1% steps from a 20% duty cycle to a 80% duty cycle. Figure 13 shows the frequencies that were tested and the duty cycle ranges in which the RX G-Link would not re-lock.

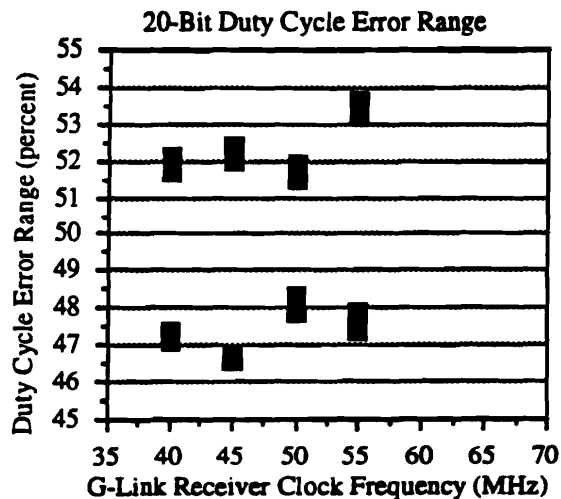


Figure 13: Duty Cycle Error Ranges of G-Link RX Clock

The test results in figure 13 show that there is a narrow duty cycle range on each side of 50% for each frequency in which the G-Link receiver will not re-lock. When this same test was repeated except that the duty cycle of the G-Link transmitter clock was varied, the G-Link never lost lock. The G-Link transmitter compensates for its clock not having a 50% duty cycle. The inability to re-lock occurred in 16-bit mode and 20-bit mode and is when the G-Link is used in the

two clock simplex mode. This is important as the duty cycle of a clock used in an actual G-Link data system may not be exactly 50%. Further testing needs to be done on newer versions of the G-Link chip set including the HDMP-1012 transmitter and HDMP-1014 receiver to verify if this RX clock duty cycle still gives the same results.

### III. Conclusion

In an event builder switch setup, it is required that a serial data link system have the ability to regain lock within a reasonable time on the change of the switch configuration. The HP G-Link chip set which is a candidate for a high speed serial link was tested. The tests demonstrate that the G-Link can re-lock in less than 30  $\mu$ s provided fill frames are sent. In the application of the event builder, the TXs can be forced to send fill frame signals on each change of the switch configuration. With an expected minimum switching interval of 1  $ms$ , devoting a period of 30  $\mu$ s to send fill frames to assure re-lock should be a reasonable data throughput delay. In other words, 3% data throughput inefficiency due to the re-synchronization of the receiver is reasonably small. Further testing needs to be done in order to measure bit error rate along with the remaining measurements in the simplex dual clock configuration even though this configuration would not be used for the proposed application for the event builder.

### IV. Acknowledgments

We would like to thank J. Butler, Prof. S. Iwata, T.K. Ohska, and Y. Watase for their support.

### V. References

- [1] O. Sasaki, M. Nomachi, T.K. Ohska and H. Fujii, "A VME Barrel Shifter System for Event Reconstruction for up to 3 Gbps Signal Trains", IEEE trans. Nucl. Sci., NS-40 (1993) 603.
- [2] "Gigabit Rate Transmit Receive Chip Set Technical Data", Hewlett Packard.
- [3] "Reference Guide for the Gigabit Rate Transmit/Receive Chip Set (HDMP-1000) Evaluation Board", Hewlett Packard, October 1992.
- [4] "G-Link: A Chipset for Gigabit-Rate Data Communication", Hewlett-Packard Journal, October 1992.
- [5] B. Turko, M. Wong, "Measurements With External Clock At Receiver To Re-Establish Lock in Simplex Mode", LBL, March 1993.
- [6] "Low Cost Gigabit Rate Transmit/Receive Chip Set", Hewlett Packard, November 1993.



## **SCI with DSPs & RISC Processors for LHC 2nd Level Triggering**

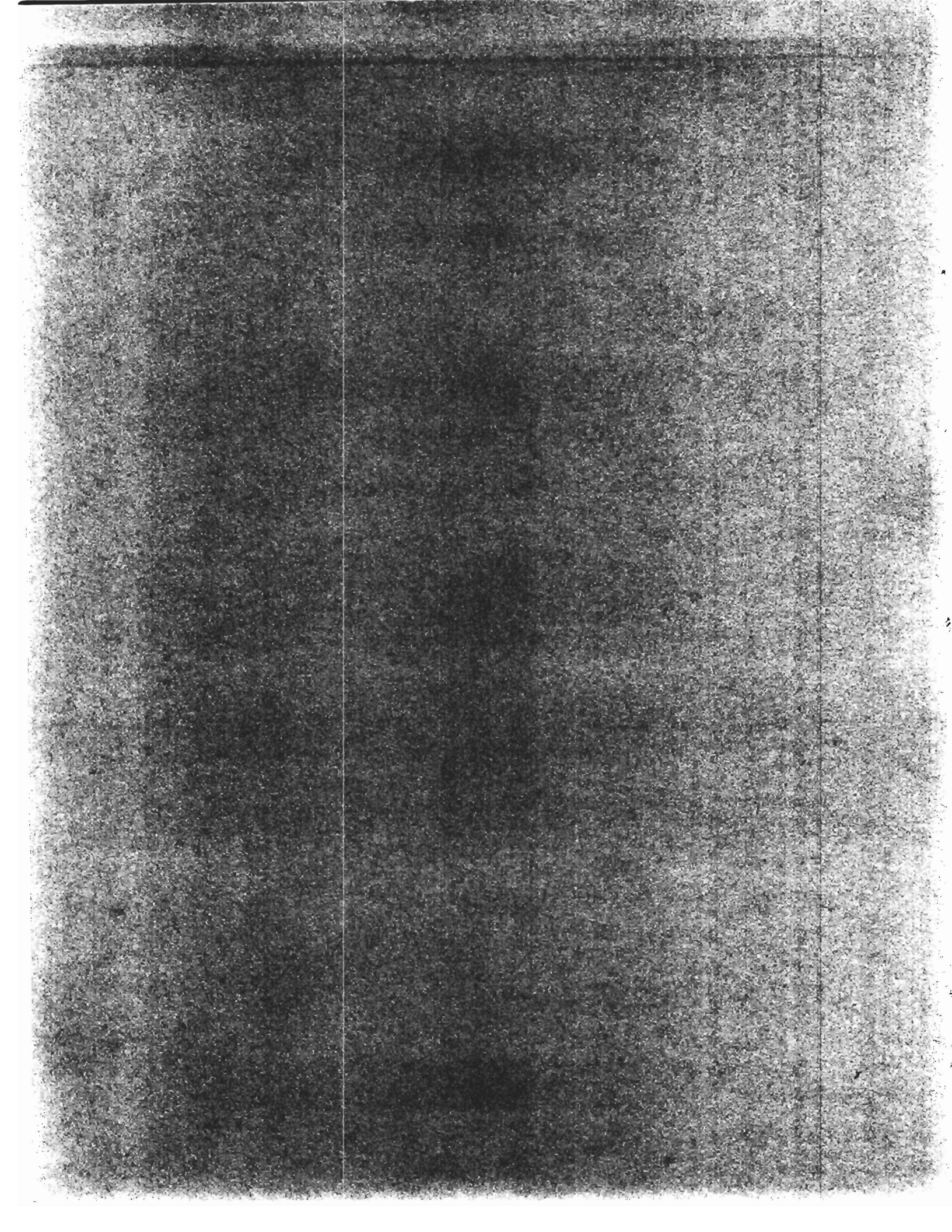
**Robin P. Middleton**

**Rutherford Appleton Laboratory**

A system for real-time selection of data from a high energy physics detector is being studied for use at a future LHC experiment. The design goals are for a system capable of 100,000 selections per second, a processing latency of around 1 millisecond and an overall data input capability of many Gbytes/s. High bandwidth interconnects and neural partitioning are exploited.

A two-stage, small scale prototype has been designed and built for assessment of architectural and technological choices. The first stage, using TMS320C40 DSPs, controls data input, buffer management and initial data reduction and routing. A second stage, based on RISC processors, merges data from multiple DSP sources on an SCI ring before making data selections.

Experiences in operation of the system in various test environments is reported. A comparison of measured performance with modelling predictions is made and plans for the future given.



# SCI with DSPs and RISC Processors for LHC 2nd Level Triggering

P.E.L.Clarke, R.Cranfield, G.J.Crone  
University College London, Gower Street, London, WC1E 6BT, UK.

B.J.Green, J.A.Strong  
Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK.

R.E.Hughes-Jones, S.Kolya, R.Marshall, D.Mercer  
University of Manchester, Manchester, M13 9PL, UK.

K.Korcyl  
Institute of Nuclear Physics, Cracow, Poland.

R.Hatley, R.P.Middleton, F.J.Wickens  
Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, OX11 0QX, UK.

A.Guglielmi  
Digital Joint Project - CERN, c/o CERN, Bldg 513 1-026, CH-1211 Geneve 23, Switzerland.

## INTRODUCTION

Detectors at the Large Hadron Collider (LHC) at CERN will have to handle raw data rates of order  $10^{15}$  bytes per second. To this end, a 3-level trigger system [1] is under study to reduce this rate by  $10^7$ , so that only interesting event data from proton - proton collisions is recorded for subsequent physics analysis. The first level, consisting of custom designed hardware is expected to reduce the rate by a factor of  $10^4$ , the second level by a factor of  $10^2$  and the third level by a factor of 10.

This paper is concerned with studies of a candidate level-2 system based on particular choices of technology.

## ARCHITECTURE

The level-2 system (see figure 1) is split into local and global parts. A processor in the local system are used to process data from a specific, small region of a detector. whilst a global processor is used to process data derived from all regions of all detectors

In the local part, guidance is taken from the level-1 system to extract fine grain raw data from the level-2 buffer to produce specific regions of interest (RoI) of the detector. The data are processed in feature extractors (FEX) to produce a feature (e.g. for a calorimeter this would be a cluster energy and position with some associated 'particle' classification).

The resulting features from all detectors participating in the level-2 system are then gathered together in a data concentration phase prior to passing through a network to the global sub-system. Here, features from different detectors which correspond to the passage of a particular particle or jet through the detector are combined and a probable particle identification assigned. Certain physical quantities are then

evaluated for use in classification of events according to topology and likely under-lying physics processes. The end result is a decision as to whether to accept or reject the data.

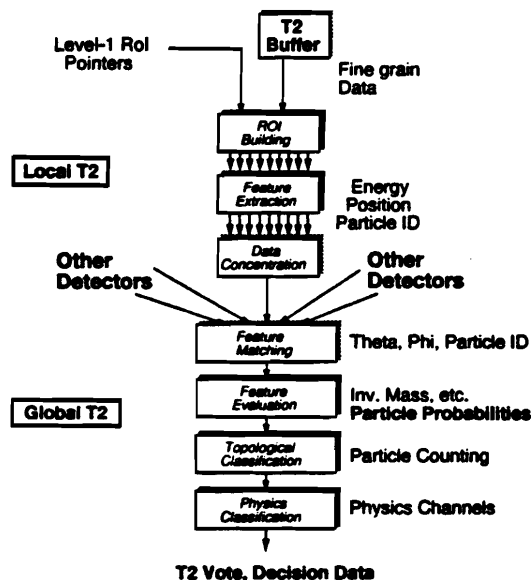


FIG. 1 Functional Architecture of a Level-2 System

The essential features to note are :-

- Localised processing of fine grain raw data
- Reduction of required level-2 system input bandwidth by limiting processing to specified Regions of Interest.
- Parallel processing in both local and global sub-systems to achieve a design decision frequency of



10<sup>5</sup> Hz, but with a processing time of a few milliseconds for each event.

- System scalability to track the evolution of algorithms and corresponding physics goals

## INITIAL STUDIES

### Selection of Technologies

The Texas Instruments TMS320C40 digital signal processor [2] is a leading floating point processor designed primarily for image processing applications. It has a simple (RISC-like) instruction set, parallel operations and six 20 Mbyte/s communications links (each supported by a separate DMA channel). The combination of a high performance processor with integral communication capabilities makes it an excellent choice for both FEX processor, data routing

function and buffer management, since it is well suited to handling data from neighbouring regions during feature extraction.

Scalable Coherent Interface (SCI) [3] provides a very high performance interconnect between processors and memory through a network of point-to-point links combining the advantages of backplane buses and traditional networking. SCI nodes are usually organised in a ring structure, with each SCI transaction consisting of request and response sub-actions. Since all SCI links can transfer data concurrently, there is no arbitration bottleneck. 16 bits of the 64 bit SCI address space are used for node addressing thus permitting extremely large rings to be constructed. However, it is more usual to configure modest rings (e.g. just 10 nodes) interconnected by SCI bridges or switches to limit transaction latencies to a manageable level.

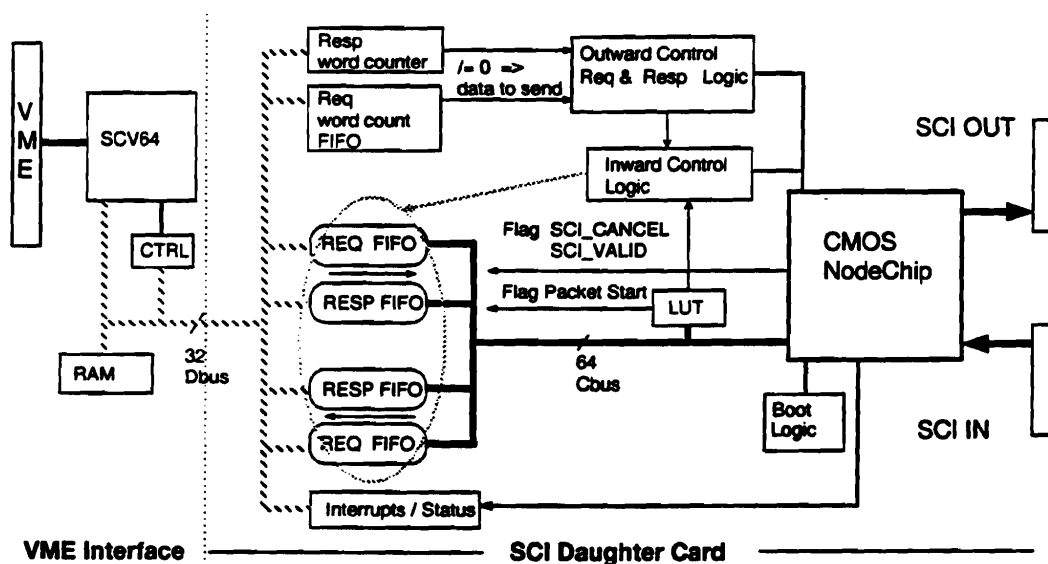


FIG. 2 Block Diagram of an SCI Node

### Building Blocks

Commercial units, namely DBV42 & DBV44 modules [4], provided a maximum of either two or four C40s per card, and were used for all C40 processors. The level-2 buffer [5] was implemented as a sister module to a C40 board and was designed to operate at up to 100 kHz.

All SCI nodes were custom built, using SCI NodeChips<sup>(TM)</sup> [6] and high performance fifos to decouple the processor bus from the NodeChip. Figure 2 shows a block diagram of one of two types of interface used. To ensure deadlock free operation of both request and response sub-

actions, four fifos were used. The SCI standard defines specific packet formats for each transaction type.

A processor wishing to initiate a transaction constructs a packet in the Request Output fifo and then initiates packet transmission through the NodeChip. Responses from remote nodes return to the originating node and pass through the Response Input fifo to be handled by the processor.

A node also responds to an external request received through the Request Input fifo and returns any data through the Response Output fifo.

The SCI interface logic described was implemented on a 6U Eurocard and was combined with additional logic to interface either to VME (as in figure 2) or to the C40 global bus. The SCI to VME interface, thus formed, was used both with an embedded VME controller and through a memory mapped interface into a DEC Alpha system.

<sup>1</sup> NodeChip is a trademark of Dolphin Interconnect Solutions

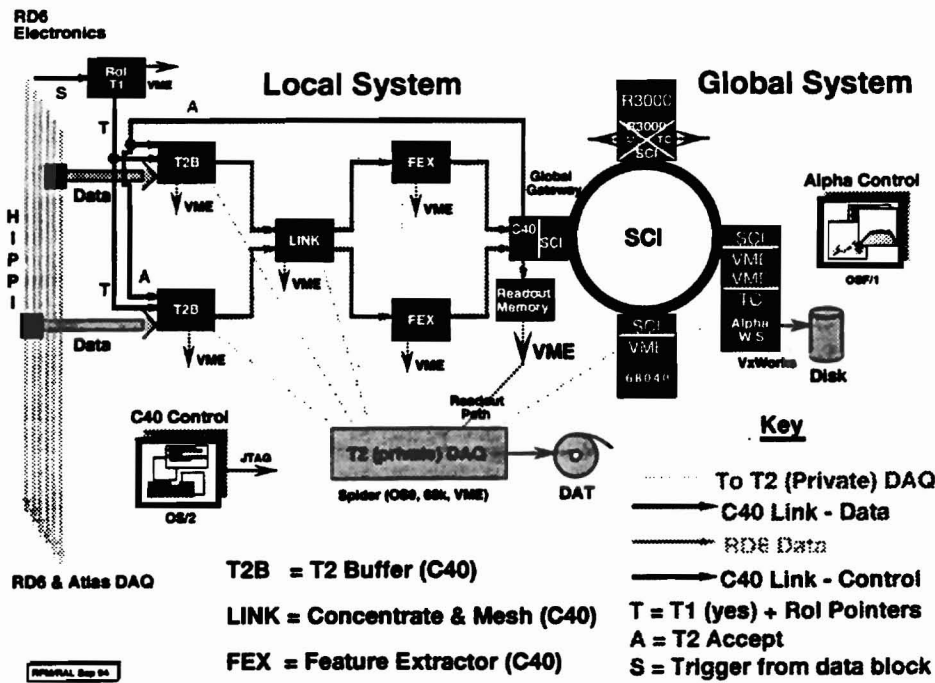


FIG. 3 Beam Test Set-up

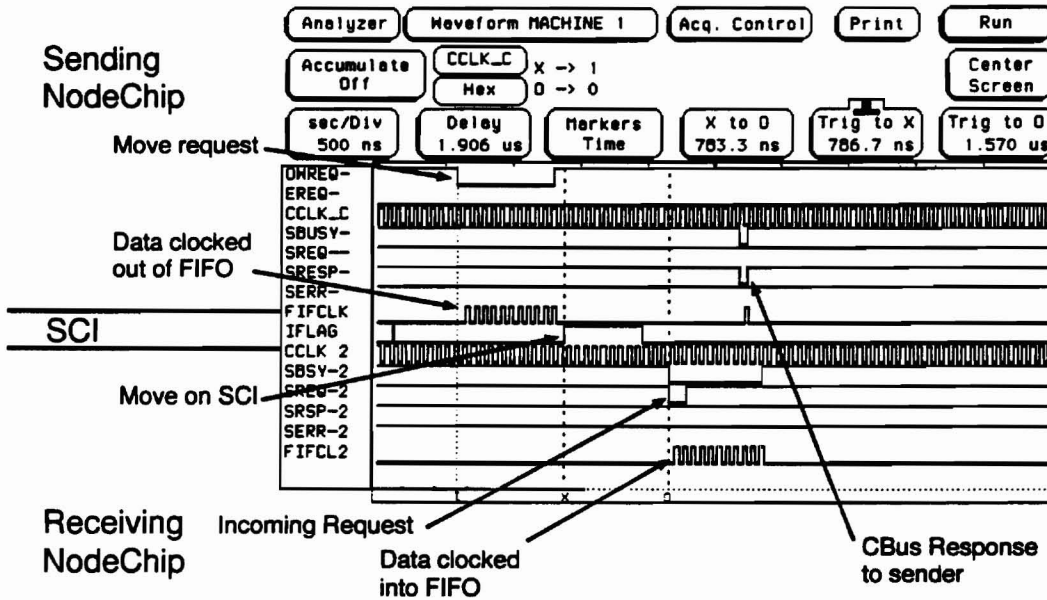


FIG. 4 SCI Move64 Timing Trace

*Test Set-up*

Figure 3 shows the interconnection of modules used to build the first test system. Data was derived from a HIPPI spy unit which formed part of a router system [7] placed in the data readout path of a Transition Radiation Detector (TRD)

[8]. Two channels were equipped to feed data into two level-2 buffers under the supervision of a C40 based buffer manager.

Data was fed through a link unit to enable routing of data to the appropriate FEX (again both C40 based). After feature extraction (a null algorithm in initial tests) the data was concentrated in the global gateway (another C40) where it was sent on to a SCI node for final processing by a global processor. Events were recorded at the global gateway by an

OS9 system and at the target SCI node consisting of a DEC Alpha processor [9] running the VxWorks<sup>2</sup> real-time kernel.

The whole system was self-triggered by the passage of an event on the RD-6 HIPPI lines.

The C40 sub-system was controlled from a PC through a daisy-chained JTAG interface. Nodes in the SCI sub-system were each self-configuring.

### Tests

The complete system, consisted of two level-2 buffers, one link unit, two FEXs, global gateway and a four node SCI ring. It was first successfully operated in the ATLAS test beam line at CERN during September/October 1994 using data from the TRD of the RD-6 collaboration. Parasitic operation ensured minimal disruption to other parts of the ATLAS tests and enabled the work programme to remain independent of other activities. Invaluable experience was gained in integrating with real detectors providing real data.

It is believed that this is the first time that 'live' detector data has been passed round an SCI ring at a beam line.

Figure 4 shows a logic analyser trace of a move-64 byte transaction between two SCI nodes (64 bytes of user data with a 16 byte header). The time taken for the transmitting node to send the move request on to the SCI ring and receive the response from the remote node was 2.1 $\mu$ s. The time from when the data is placed on to the SCI to when the data is clocked into the receiver's fifo is 1.48 $\mu$ s. The SCI ringlet was occupied for 850ns in transmitting the request.

### FUTURE PLANS

It is planned to expand the scope of the project in the near future by enlarging the system to accept data from more than one detector and to field an adequate set of RISC processors to handle the data. Proper feature extraction algorithms will be implemented, tailored to individual detectors. Features from different detectors for the same RoI must be matched and subsequently processed by a global algorithm to yield an overall Level-2 decision.

### ACKNOWLEDGEMENTS

The authors would like to extend thanks to the RD-6 collaboration for permitting data access through a spy mechanism, to colleagues in the University of Jena and JINR Dubna for use of the router system and to the ATLAS test beam organisers for the tests. The authors would also like to thank colleagues in the RD-24 and EAST collaborations at CERN for their help and support and Digital Equipment Corporation (through the CERN-DEC Joint Project Office at CERN) for their collaboration in the development of the SCI sub-system and Alpha processing node. Financial support from the UK Particle Physics and Astronomy Research Council for this project and partial support for one of the authors (KK) from the Polish State Committee for Scientific

Research (grant No. 2 P302 047 06) is gratefully acknowledged.

### REFERENCES

1. ATLAS Collaboration Letter of Intent, CERN/LHCC/92-4, Chapter 5, p67.
2. TMS320C40 Users Guide, Texas Instruments.
3. "SCI, Scalable Coherent Interface", IEEE standard 1596-1992.
4. DBV42 & DBV44 Technical Reference Manuals, Loughborough Sound Images Ltd, Loughborough, UK.
5. A Second Level Data Buffer with LHC Performance; B.J.Green et al; 6th Pisa meeting on Advanced Detectors
6. L64601 SCI NodeChip Technical Manual, LSI Logic Corporation.
7. CERN/EAST note 92-09.
8. RD-6 Collaboration, CERN/DRDC/P8.
9. Alpha Architecture Reference Manual, ISBN 1-55558-098-X, Digital Equipment Corporation.

---

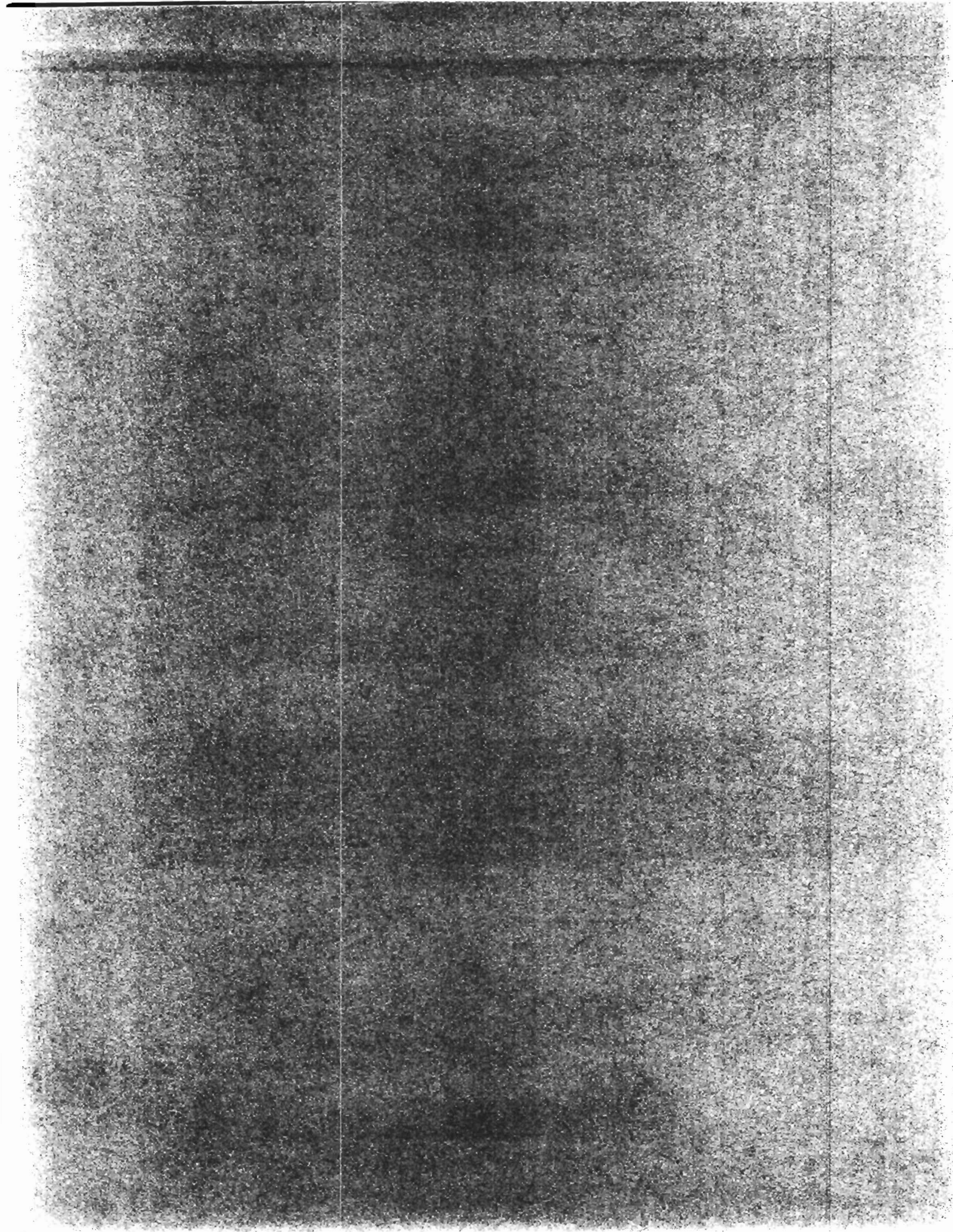
<sup>2</sup>VxWorks is a trademark of Wind River Systems Inc.

**Vortex: A High Performance Parallel Processing Event Server with  
an ATM Interface**

**Michael Mojaver**

**University of California San Diego**

Vortex is a data processing module being developed at UC San Diego, which addresses the problem of buffering event data from front-end sub-systems and serving those events to a processor farm via an ATM switching network in an optimal fashion. The Vortex design takes advantage of commercially available, fully programmable, parallel processing blocks which combine very high data bandwidth, massive processing power and low cost. The advantages of a fully programmable solution include: minimal system constraints, system behavioral modification without hardware redesign, and optimal utilization of event-building resources.



# **A High Performance, Parallel Processing Event Server with an Asynchronous Transfer Mode Interface.**

Michael Mojaver, Jim Branson,

*Physics Department, University of California San Diego, La Jolla, California 92093*

October 24, 1994

## **Abstract**

A proposed architecture for an ATM based, scaleable event server architecture and the status of work to implement this architecture is presented. An event server is a critical component of interfacing a commercially available ATM based event builder to detector systems. This interface establishes physical links, buffers events, tracks event data, handles protocol conversion, and performs other data processing to insure data integrity and optimal data flow. To address the many complex issues involved in the design of an event server, a fully programmable parallel-processing architecture with standard peripheral extensions is being investigated. This approach has been made practical with the recent introduction of a commercially available parallel processor which combines high processing power, large IO bandwidth, and low cost.

## **Introduction**

Data rates for some new proposed detectors are tremendous. The CMS experiment, which is the focus of our event server architecture, specifies an event builder with an aggregate data rate in the order of 50-100 Gbyte/sec. A cost-effective event builder based on ATM technology has been studied in some detail [1] and appears feasible. An event builder - detector interface, sometime referred to as the "buffer memory", "dual port memory", "triple port memory", "multiport memory", to emphasize its buffering aspects, has to perform the following functions to make a seamless interface:

- Buffer event data sent by front-ends until requested.
- Perform protocol conversion between the front-ends and the switching network.
- Perform some data processing, to insure data integrity and optimal data flow.

A number of other related functions at this interface are applicable and need to be considered. A comprehensive list of the functions is difficult to assemble with incomplete data acquisition parameters. The following is a partial list of possibilities:

- Compression of event data.
- Suppression of noise hits in some detectors.
- Error detection and recovery.
- Gathering or monitoring statistical data.
- Transmission of back pressure and other control signals to the front ends.
- Hardware trigger interfaces.

It can be argued that some of these functions will enhance the operation of the data acquisition system and others have the potential to reduce the cost of the event builder or improve its throughput. Each of the functions is discussed in a separate section below. Most of these functions can be implemented at no additional cost using the programmable event server architecture discussed in this document.

## System Architecture

The basic philosophy in the event server design is to perform data processing functions using high performance processors, rather than dedicated hardware, use the processor memory for all data transactions and rely on extensive peripheral pipelining. This approach has been made practical with the advent of a new generation of fully parallel single chip multiprocessors developed for multimedia/video applications (the same technology that drives ATM networking advances.) Multimedia applications, like the event server, require large high performance memories.

The design as described here is expected to deliver all the functionality mentioned in the previous section. The assumption made is that the event builder is ATM based. The simplified system block diagram of the event server module is shown in Figure 1. The system is composed of a C80 processor, the main memory system, local bus ATM link, PCI local bus bridge, PCI-VME64 bridge, PCI-Ethernet interface, and a PCI Mezzanine Card (PMC) extension. The advantages of this architecture include:

- Up to 4 Gbytes of buffer address space.
- Large Event buffer memory, with virtual multi-porting.
- Intelligent buffer allocation and memory management.
- Dynamic optimization of ATM event builder link.
- High IO bandwidth (400Mbyte/s local bus with 50 MHz clock.)

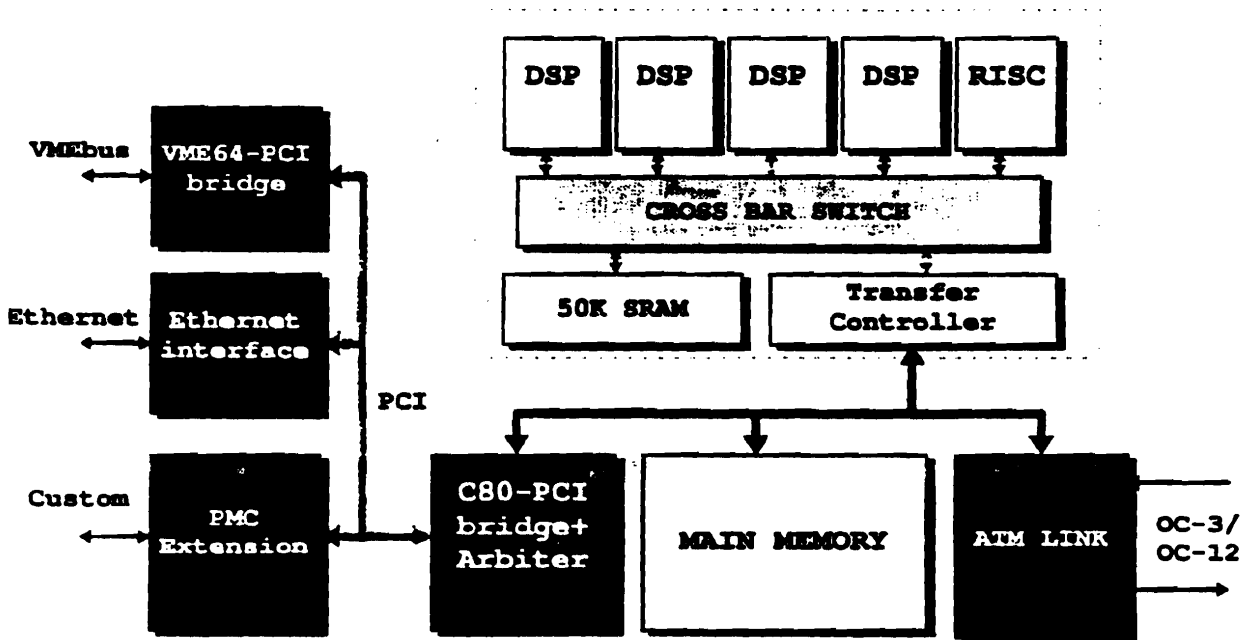


Figure 1. System Block Diagram.

- Surplus CPU power for pre-processing , e.g. thresholding, data compression, error checking and correction.
- Standard 6U x2 packaging and VME64 interface.
- Standard 64-bit PCI local bus interface, with PMC expansion.

The multiprocessor subsystem is a single commercially available integrated circuit available from Texas Instruments (the TMS320C80) which includes four fully programmable Parallel Processors, a RISC master processor, an intelligent memory transfer controller, external memory controller circuitry, and on-chip SRAM.

Each of the parallel processors is capable of performing many RISC-equivalent operations in a single cycle. The fifth processor, the master processor, is a 32-bit RISC CPU and includes a high-performance IEEE-754-compatible floating-point unit. All five processors can be programmed in both C and assembly language. The processors are capable of performing the equivalent of over two billion RISC-like operations per second. The processor bandwidth is 2.4 Gbyte/sec for internal data, 1.8 Gbyte/sec for internal instruction fetches and 400 Mbytes/sec externally.

The extremely high on-chip bandwidth is made possible by a crossbar switch on the C-80 which supports a shared memory model. Crossbar-shared memory is the most flexible multiprocessor memory architecture because it places the fewest restrictions on where data must be loaded. Although the crossbar maintains nearly 1000 data and address lines that are connected among the processors and memory, it is practical to use because all memory connections are integrated on one chip. The crossbar's flexibility translates into better efficiency in terms of execution speed and ease of programming. In addition to managing memory accesses, the crossbar is also used to send command words (or interprocessor commands) between processors.

There are sixteen independent on-chip RAM blocks on the crossbar that can be accessed by any of the processors in a given cycle. These RAMs are referred to as shared RAMs. Single-cycle access to the on-chip shared RAMs by any of the processors reduces the traditional bottleneck for multi-chip parallel processing, such as delays associated with passing data between different memory spaces and wait-stated access to off-chip devices.

Block transfers between the processors and memory/peripherals is handled using the on-chip transfer controller (an intelligent DMA controller) which manages all memory traffic. The transfer controller performs packet transfers that move data between on- and off-chip memory, and peripherals. These packet transfers include instruction and data-cache servicing, as well as complex programmable byte-aligned array transfers.

The main memory system plays a key role in the event server functionality. In addition to storing processor instruction and data structures, all event data will be stored in the memory while transfer decisions are being made. Memory organization to a major extent is determined by the extent of buffering required. Buffering requirements are a function of data rate, average event size and processor latencies. Multi-porting is not required since it can be virtual. This is possible because of the massive system IO bandwidth available and with pipelined peripheral interfacing. The advantage of virtual multi-porting over hardware based techniques is that optimum memory allocation schemes can be implemented using software techniques. External memory system granularity, the smallest amount of memory that can be added at a time, is technology dependent . We are currently considering 4-16 Mbit Synchronous DRAM devices.

The event server design assumes an ATM event builder interface, so a high performance ATM link on the C80 bus is included in the design. The performance goal is 1 to 2Gbit/sec links, but currently 622Mbit/s is readily available. The ATM physical layer interface (and may be higher level layers) will be implemented using a module from Triquint Semiconductor inc. Specifications of the module are currently unavailable, but some pipelining, byte alignment and glue logic may be required to interface the card to the C80 bus. The module supports SONET OC-3 (155Mbit/s) or SONET OC-12 (622Mbit/s) interfaces.

SONET (Synchronous Optical Network) is a fiber-optic-standard for ATM communication networks. SONET converts the digital content of cells into robust analog data streams and recovers data at the receiving end,



using the HEC byte in the header as a synchronization mark to reassemble data cells. The SONET mapping process assembles ATM cells into fixed-size frames along with additional embedded control and error-detection bytes. Sonet protocol is quite complex but in exchange offers high noise immunity, and a wide variety of data rates.

A 64-bit Peripheral Component Interconnect (PCI) bus will be used to interface all other peripheral devices in the event server. A bridge between the PCI bus and the C80 bus, should offer a high bandwidth low-latency path to the main memory system for the purpose of block transfers. Each device connected to the PCI bus can function as a master or a slave, and can initiate data transfers. Pipelined stages in the PCI-C80 bridge decouple transaction on each bus, facilitating parallel transfers. Maximum data transfer rate on PCIbus is 132 Mbyte/sec or 264 Mbyte/sec for 32-bit and 64-bit transactions respectively.

Taking advantage of hierarchical bus architecture available with PCI, VMEbus will be directly supported as a secondary bus in the event server. VME secondary bus support is integral part of the 6U VME packaging format. The event server can connect with devices on the VMEbus using VME64 format (60 Mbyte/sec. transfers.) The VMEbus interface is a single-chip device (on PCI local bus) manufactured by Newbridge Microsystems, Canada.

The PCI local bus simplifies hardware aspects of adding peripherals to the event server. The Ethernet interface for example is also a single-chip solution available from Advanced Micro Devices. Alternative or custom interfaces can be supported using the PCI mezzanine Card (PMC) connection which is also known as the IEEE P1386 standard.

### **Software Constructs**

The functional behavior of the event server is "soft" and determined by a collection of internal algorithms that establish the flow of data and control. The number and type of functions that can be supported by the system is dependent on the availability of system resources. To create a function, an algorithm must negotiate a set of resources, which when allocated constitutes a process. With multiple concurrent processes a number of complex issues need to be addressed. While some processes may allocate unique resources, other hardware resources such as the main memory and the system bus are designed to be shared so arbitration is necessitated. In addition, each process can initiate one or more tasks (threads of execution) to perform the expected function and inter-task communication may be needed. Multi-tasking issues are handled by a small operating system running on the master processor, that provides local control of on-chip parallel processing tasks and presents a uni-processor-like interface to the C80.

As an example a process to locate requested events and to keep track of free pages in memory when events are transmitted or expire is crucial to the event server operation. The list of free pages is used to route incoming packets to valid destinations in the buffer memory space. Because memory management is continuous and inherently a high rate process, efficient search algorithms for locating requested events and providing free pages to the input process are needed, and currently being developed.

### **Current Status and Plan of Work**

We have currently a working knowledge of the C80 processor, its software development environment, and have the capability to benchmark algorithms using the C80 simulator or emulator. We have begun to implement frozen portions of the design, including printed circuit layout using Cadence design tools. We have prototyped and developed software for the first generation VME64 interface and will be a Beta site for the VME-PCI chip which will be sampling in February 1995.

C80 Silicon currently is available as engineering samples. The device, with the exception of a few minor bugs appears fully functional. The current revision of the Silicon has an instruction cycle of 33ns (66Mhz

clock) and lacks SDRAM support. The 20ns device (100Mhz clock) with SDRAM support will be available first quarter in 1995, and currently planned as 3.3V only, which complicates some design issues.

Although our original goal was to have a prototype at the end of 1994, most likely this will be postponed to early 1995 due to insufficient design data and the lack of a commercial C80-PCI bridge. The prototype may not have all the hardware characteristics listed in this document, but will be invaluable as a S/W development platform and as an architectural evaluation tool.

#### References

- [1] RD-31 (Nebulas) Status report, CERN (1993.)
- [2] T. Shanley, D. Anderson., PCI System Architecture, (Mindshare Inc., TX, 1994)
- [3] Asynchronous Transfer Mode Networks, performance issues, R.Onvural (Artech House, MA, 1994.)
- [4] TMS320C8x (MVP) Reference, Texas Instruments Inc. (1994)
- [5] Common Mezzanine Card Family, IEEE Standards Department, (IEEE, New York, 1994)

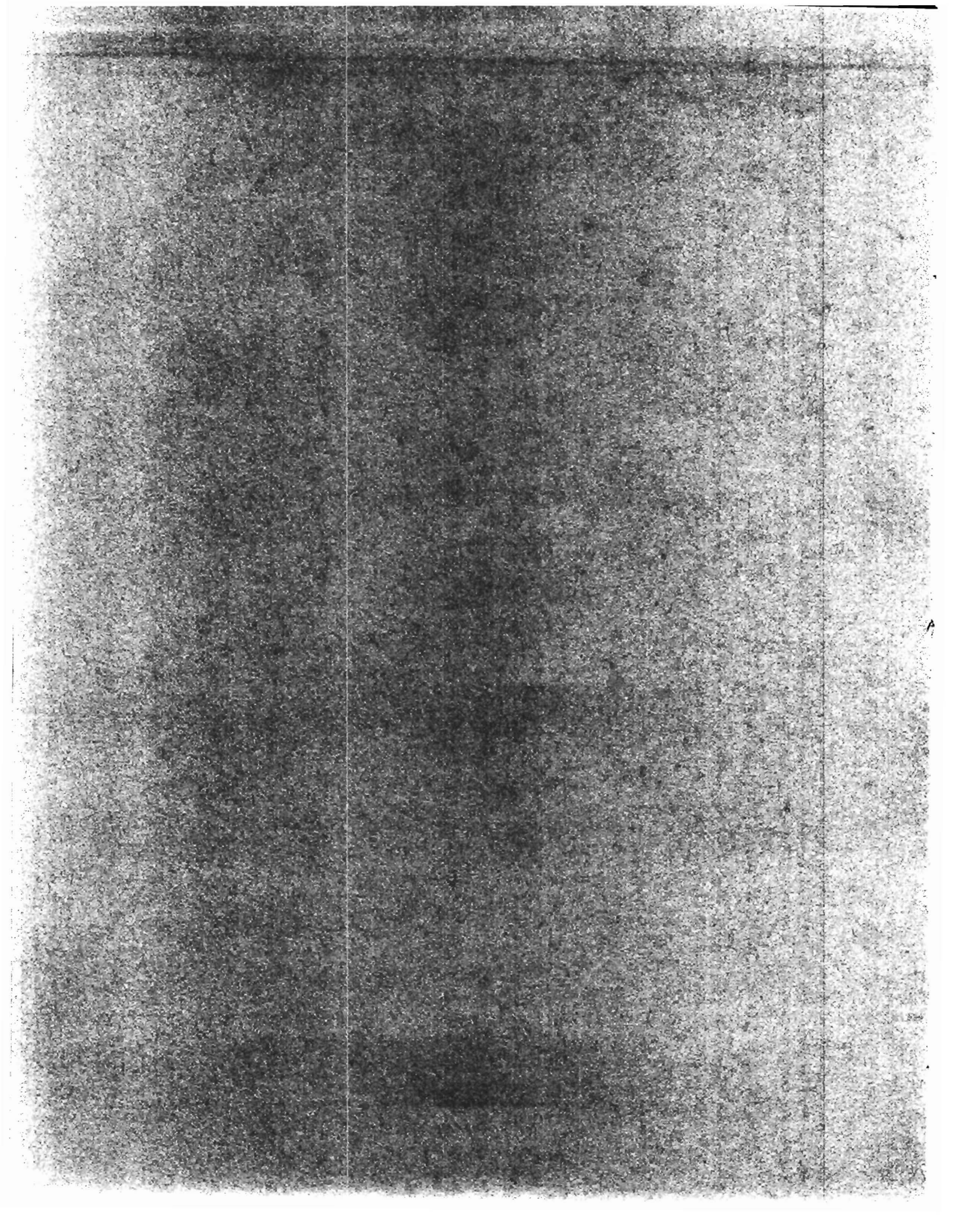


**Prototype of an Event Building System Based on HIPPI**

**Ralf Spiwoks**

**CERN**

The poster will present a running prototype of an event building system, as it is implemented using a HIPPI switch, running the acquisition software on a VME based RISC processor.



# Prototype of an Event Building System based on HiPPI

S. Buono, I. Gaponenko<sup>1</sup>, R. Jones, V. Kozlov<sup>1</sup>, L. Mapelli<sup>2</sup>,  
G. Mornacchi, D. Prigent, I. Soloviev<sup>3</sup>, R. Spiwoks<sup>4</sup>  
*CERN, Geneva, Switzerland*

G. Ambrosini, G. Fumagalli, G. Polesello  
*Dipartimento di Fisica dell'Universita e Sezione INFN di Pavia, Italy*

P.Y. Duval, A. Le Van Suu  
*Centre de Physique des Particules de Marseille, IN2P3, France*

K. Djidi, M. Huet  
*Departement de Physique Nucleaire - STEN, C.E. Saclay, France*

## ABSTRACT

One of the goals of the RD13 project at CERN [1] is to investigate the feasibility of different event building techniques for LHC detectors. These studies have been started using the HiPPI standard [2] and a commercial HiPPI switch [3]. A first prototype has been built and successfully tested with two sources and one destination and a total data throughput of 5 MB/s limited only by the slow DMA device of the VME processor chosen. The pure I/O data throughput is 40 MB/s. The system has been made in a modular way and will be extended to have more source and destination modules and to use different hardware standards.

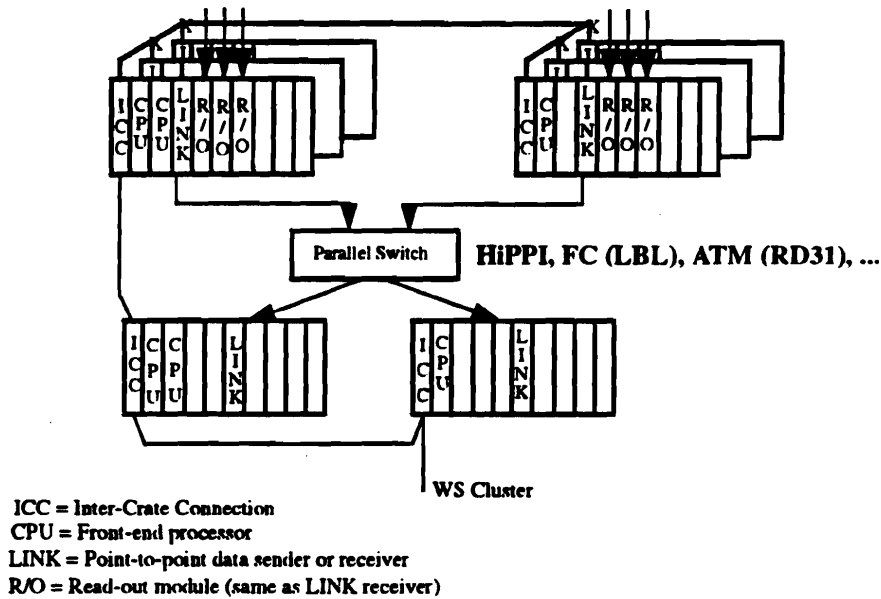
## I. Introduction

The event building system of a future detector for the LHC will have to cope with unprecedented high data rates (of about 10 GB/s with event rates of about 1 kHz). Since "classical" solutions cannot cope with this data rate, new techniques have to be considered. Parallel event building using fast switching networks seems a possibility which the RD13 project at CERN is investigating. The goal is to build a testbed where different hardware components and control schemes can be combined in order to understand the feasibility and the requirements. A

- 
1. On leave from the Budker Institute of Nuclear Science, Novosibirsk, Russia.
  2. Spokesperson.
  3. On leave from the Petersburg Nuclear Physics Institute, St. Petersburg, Russia.
  4. Also at University of Dortmund, Dortmund, Germany.

first set of requirements of a modular and scalable event building system has been defined [4] and a first prototype using a commercial HiPPI switch and VME-HiPPI interfaces has been implemented and successfully tested. This system should be regarded as a starting point for the research in this field. The full layout of the system planned is shown in FIGURE 1.

FIGURE 1. : Functional Model of the RD13 DAQ



## II. The Hardware

The whole prototype system is housed in one VME crate except for the HiPPI switch itself. Apart from the switch the system includes a VIC board, a RAID board, two HiPPI/S and one HiPPI/D interface. They are described in the following:

The IOSC HiPPI switch [3] is fully compliant with the HiPPI standard [2], has 8 input and 8 output ports and a full bandwidth of 800 Mbits/s. The arbitration of the source requests is done in a first-in-first-out way making the request “camp on” as long as the destination is busy. The actual switching delay is less than 1 us and the addressing can be done directly or using tables for each port independently. The switch can be programmed and monitored using a RS232 interface.

The RIO 8252 HiPPI/S or HiPPI/D [5] is acting as a VME-HiPPI interface. The board consists of a R3051 Risc controller, 4 MByte DRAM, VME master and slave interfaces and a HiPPI interface implementing the HiPPI protocol in hardware.

The RAID 8235 [6] is a R3000 processor based board with 32 Mbyte DRAM, a DMA device and VME master and slave interfaces. It runs a real-time UNIX called EP/LX and is used for the control part of the EB system. A VIC 8251 is used for the arbitration in the VME crate.

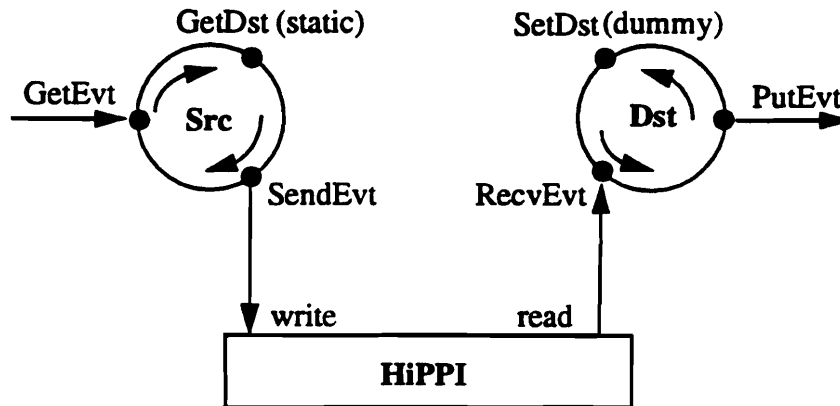
### III. The Software

The software is made in two different layers: a lower layer for the HiPPI dependent code and a higher level for the generic event building features.

On the lower level the firmware for the HiPPI/S [7] takes care of the packeting, sending the data, accumulates statistics and monitors the status of the I/O. For the HiPPI/D module a new firmware was written [8] to fulfill the task of receiving the data, do the merging, accumulate statistics and monitor the status of the I/O. The firmware for the HiPPI/S and HiPPI/D modules was optimized in transfer speed and saturates at 40 MByte/s for packets of 1MByte.

The higher level consists of one process for each HiPPI/S and HiPPI/D module: these processes (called *Src* and *Dst*) run in a loop and will be used to interface them with a full DAQ system. At the moment the *Src* process has a dummy input and sends pre-loaded data to the HiPPI switch, the *Dst* process receives the full event from the HiPPI/D module and writes it to a dummy output. Both processes are to be seen as an application of the HiPPI dependent code and can easily be extended to different hardware components and to more processes. The protocol for the EB is very simple: a basic "PUSH" scheme with time-outs and retries. The destination assignment is done in a static way using no feedback from the *Dst* side.

FIGURE 2. : The software layout of the EB prototype



### IV. Data Merging

The data merging is done in the HiPPI/D module (using its local processing power) but could also be pushed into the higher level because the code is completely independent. It is based on three buffers with fixed sized data slots: one buffer is for the free events coming in (*EvtBuf*), one for the events which are being built (*BvtBuf*) and one for the full events (*FvtBuf*).

The HiPPI/D module receives new event fragments as long as there is still buffer space in the *EvtBuf*. Then it looks up in the *BvtBuf* if there is already an event with the same identifier. If

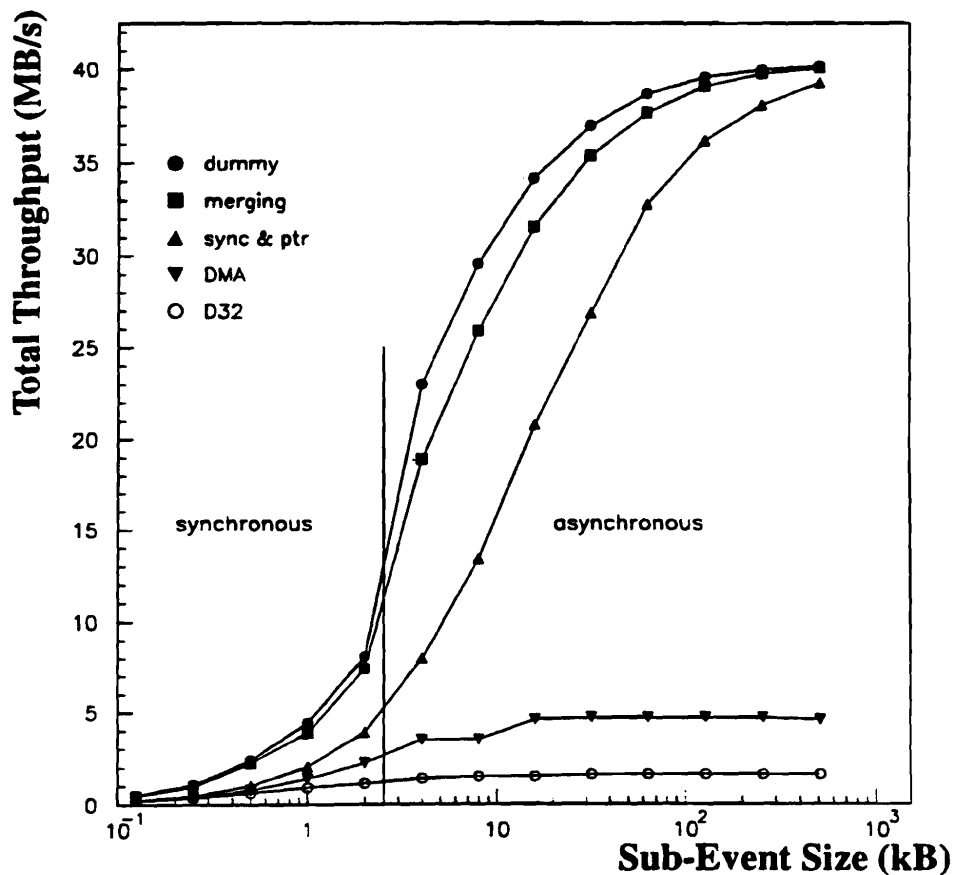


not, the event fragment is put in the *BvtBuf*. If there is an event found the new fragment is merged with it by chaining the pointers. Then the number of event fragments in the chain is counted and compared to the number of fragments expected. If the event is complete it is put in the *FvtBuf* and a signal is sent to the user. The application program can then pick up the chain of pointers and gather the scattered data. At the end the application program has to release the events by putting them back into the *EvtBuf*.

## V. The Performance

The pure I/O rate saturates at 40 MB/s. When adding the merging in the HiPPI/D module and the synchronization between the HiPPI/D module and the *Dst* process there is a drop of a few MB/s. The read-out of the data from the HiPPI/D module to the RAID processor drops the performance to 4.7 MB/s using the DMA device on the RAID, or to 1.7 MB/s using single word transfer. This is shown in FIGURE 3.

FIGURE 3. The performance of the RD13 EB prototype (2Src -> 1Dst)



For small event fragment sizes an additional synchronization mechanism is needed because the *Src* processes are completely software-driven, and since they are running on the same operating system they are not independent.

## VI. Conclusions

The event building prototype based on a HiPPI switch could be tested successfully. It is running in principle and the low performance is only limited by the slow DMA device on the RAID board.

Future extensions of the system are planned and should be implemented easily. Among these extensions is an increase in the number of HiPPI/S and HiPPI/D modules to make the system run as a parallel event building system. Another extension is to increase the number of processors, leading to a farm of processors.

The prototype will be implemented for a testbeam DAQ system to be used under real conditions, and will be modelled using DSL [9] to understand its scalability. The RD13 project is in collaboration with the RD31 [10] project for using an ATM switching network and in contact with LBL for Fibre Channel [11].

## VII. References

- [1] L. Mapelli et al., A Scalable Data Taking System at a Testbeam for LHC, CERN/DRDC 90-64, CERN-DRDC 94-24.
- [2] ANSI X3.183-1991 HiPPI-PH High Performance Parallel Interface Physical Standard.
- [3] Input Output Systems Corporation, HiPPI Switch, 1994.
- [4] R. Spiwoks, RD13 TN 111, Requirements of an Event Building System, April 1994.
- [5] Creative Electronics Systems, RIO 8262 HiPPI/S or HiPPI/D HiPPI to VME Interface, 1992.
- [6] Creative Electronics Systems, RAID 8235 VME Risc Processor Board, 1992.
- [7] E. v.d. Bij, RIO 8262 HiPPI/S firmware manual, 1993.
- [8] R. Spiwoks, RD13 TN 129 & 130, HiPPI/D firmware & User Library, October 1994.
- [9] R. Spiwoks, DAQ Simulation Library, these proceedings.
- [10] J. Christiansen et al., RD31 Status Report, CERN/DRDC 93-55.
- [11] W. Greiman, Switch Operation, Buffering and Queuing, these proceedings.

All documentation of the RD13 project can be found on WWW under <http://rd13doc/welcome.html>.

---



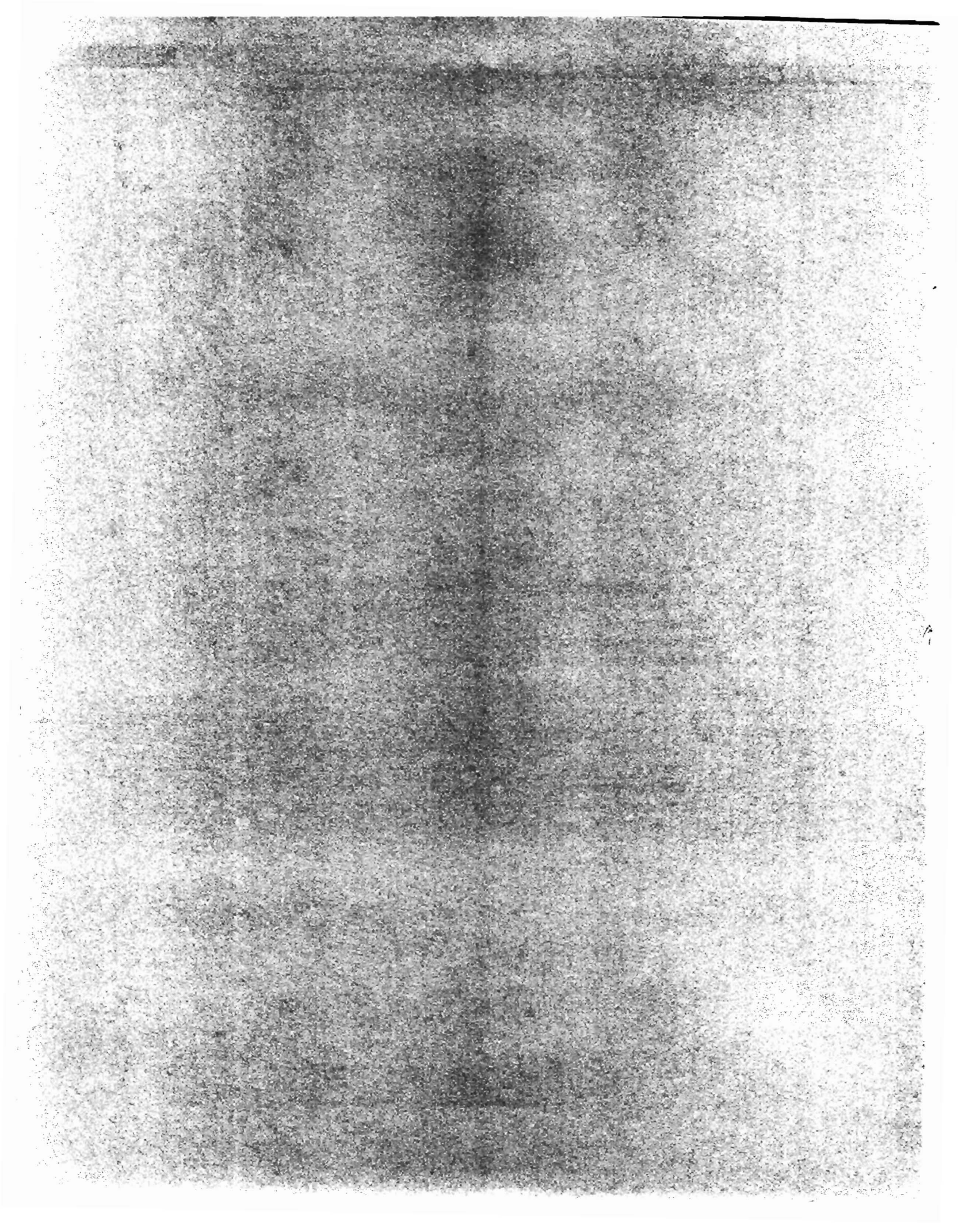
## **SyncC++, a Concurrent Language Based on C++**

**Andre Wiesel**

**EPFL**

The object oriented programming has eased the production of reliable and reusable programs and has been at the basis of many development methods. These methods are efficient for certain applications (graphical applications, databases, . . . ), but they fail when they are applied to real time systems and to applications that have to handle concurrently many kinds of events, such as process control, protocols, interactive man-machine interfaces, etc.

We have defined a language, based upon C++, that defines active objects, which are created, called, inherited and deleted exactly like the usual C++ objects and need only a very few extra keywords. We have written many program examples and several complete applications, which has shown that this approach is quite compatible with object orientation but that the event handling can be very easily included in the development. For example the integration of coordinated finite state machines, an essential concept for communications, protocols, process control, etc., is straightforward. Our environment runs already on several environments (UNIX, PC) and on a bare RISC processor which will be used for DAQ.





Abel Divin, Claude Petitpierre  
IN (Ecublens) CH - 1015 LAUSANNE (Suisse)  
TÉLÉPHONE: +41 21 - 693 47 97/26 50 TÉLÉFAX: +41 21 - 693 66 00 E-mail: divin/petitpierre@di.epfl.ch

## The language SyncC++

### Introduction

The language SyncC++ is a concurrent version of C++. It represents the smallest of all concurrent extensions we know of. It follows the object-orientation very closely. SyncC++ defines active objects, which contain their own thread of execution and play the role of tasks. The syntaxes of the creation, destruction, call, inheritance of active objects are identical to the syntaxes defined for the usual (passive) objects. C++ programmers can thus learn it very quickly. Actually this leaflet presents all new keywords.

It uses a preprocessor that produces normal C++ code that can be link-edited with any UNIX library (DEC, SUN, SGI). A version for PC under Windows is being developed.

Applications such as client/server systems, video conference, data base of CD (music), sliding window protocols, simulators (the kernel can be set in accelerated time mode) have been developed.

### Use as a concurrent language

This language can be used to program real time applications. It can be described with theories like CSP or CCS, which provide useful tools to validate event driven applications (communication protocols, industrial processes, etc). Unlike FDTs (formal description techniques) like LOTOS or SDL, it is object-oriented and the production of code from its source is straight forward. The few new keywords can replace all functions (semaphores, send-receive-reply...) found in other real-time systems.

A task is represented by an active object. An active object can call the method of another object. However within an active object only one thread can be active at any one time. If an active object is busy (its thread is running), the calls to its methods, originating from the other objects, are blocked. In the same way, if a method is already being executed when any other method is called within the same active object, it is blocked until the previous call is finished. The active objects are thus protected in order to avoid the uncontrolled quasi-simultaneous handling of shared variables.

### Applications using man-machine interfaces

SyncC++ is a very efficient tool for creating applications that request man-machine interfaces. It can replace the "interface builder" or "call-back" approach. With an interface builder, a programmer must start his/her project with the definition of the display elements (button, scales, text fields, menus...) and insert the functions of the applications afterwards, "behind the screen". With SyncC++, it is possible to structure the application first and only then add the display elements.

We have encapsulated the sockets and the Motif primitives in active objects. Unlike what happens with the call-back mechanisms, which request that the display elements are told what application function they must call when they are activated, our programs can read the display elements like a keyboard. Applications written with SyncC++ are much easier to understand, because the structures of the programs are visible and not scattered among all call-backs.

### Example

The following example shows how to create a window with a button, open a socket and then await the first event, either the activation of the button or the arrival of a message in the socket, whichever happens first.

```
BulletinBoard X(x,y,"name");           // creation of a window
PushButton Pb(&X,x,y,"STOP");         // creation of a button

TCPsocket sd = new TCPsocket ("host",portno); // creation of a pointer to a socket

select {
  Pb.Pressed();
  ..... actions 1.....
||
  sd->Read(&message, sizeof(message));
  ..... actions 2.....
}
.... actions 3.... // continue here if either Pb.Pressed or sd->read has been activated and
// corresponding actions (1 or 2) have been executed
```

In a system with a call-back mechanism, the actions (1 and 2) would be stored in two different procedures. The actions 3 should be stored in a procedure shared by the two previous procedure. In the above example, it is easier to determine

which actions are ready to be executed at any time and what are their dependencies. The *select* statement has exactly the same role as the *select* function in UNIX, but as it is integrated in the language, it can choose between different kinds of events, sockets, event flags or other object communications.

### Intertask rendezvous

The internal thread can suspend its execution to await a call to one of its method, with the instruction described below on the left. The instruction on the right is executed by another object.

```
Object XXX:                                Object YYY:
accept MyMethod;                            XXX->MyMethod(x,y,z);
                                     <=>
```

The two instructions above represent a rendezvous (the sign  $\Leftrightarrow$  is not part of the syntax). If object YYY arrives at the call before XXX has executed the *accept*, its execution is suspended until XXX executes the *accept*. Conversely, XXX is suspended if it arrives at the *accept* before YYY. During the rendezvous, when both objects are suspended, the method *MyMethod* is executed. This is exactly what happens with Ada. (However Ada's selection cannot contain calls like SyncC++ - see below - and if Ada'94 defines the concept of object, it does not integrate the task and the object).

### Intertask synchronisation and selection

In the example below, a rendezvous is defined within a selection. Its functioning is easily understood from the combination of the two examples above.

```
Object XXX:                                Object YYY:
select {                                     select {
  accept MyMethod;                            XXX->MyMethod(x,y,z);
  .... actions ....                            .... actions ....
}                                               |
|                                               Obj2->OneMethod;
|                                               .... actions ....
|                                               }
Obj1->HisMethod;                               <=>
```

### Availability

Further explanations can be obtained on the WWW server: <http://diwww.epfl.ch/w3lti>

The package is freely available on the anonymous ftp server: [ltisun.epfl.ch](ftp://ltisun.epfl.ch)

### References

#### SyncC++

A. Divin, C. Petitpierre, "An Object Oriented Method for Implementing Layered Protocols", Formal Description Techniques FORTE, Boston 1993.

G. Caal, A. Divin, "Implementing Real-Time Applications with Concurrent Objects", Euromicro Workshop on Real-Time Systems, Sweden 1994.

G. Caal, A. Divin, C. Petitpierre, "Active Objects: a Paradigm for Communications and Event Driven Systems, Globecom conference, San Francisco 1994.

#### General

M. Riese and G. Conti, "Drawbacks of ADA's synchronization mechanism and a simple solution", 3. international workshop on real-time ADA issues, 1989.

R.J.A. Buhr, "System Design with ADA", Prentice Hall, 1984.

B. Stroustrup and M.A. Ellis, "The annotated C++ reference manual", Addison-Wesley, 1991.

R. Milner, "Communication and concurrency", International series on computer science. Prentice Hall, 1989.

P. America, F. van der Linden, "A parallel Object-Oriented Language with Inheritance and Subtyping, ECOOP/OOPSLA '90.

G. Agha, P. Wegner and A. Yonezawa, "Research Directions in Concurrent Object-Oriented Programming, MIT Press 93.

## **DSP based Data Acquisition Systems**

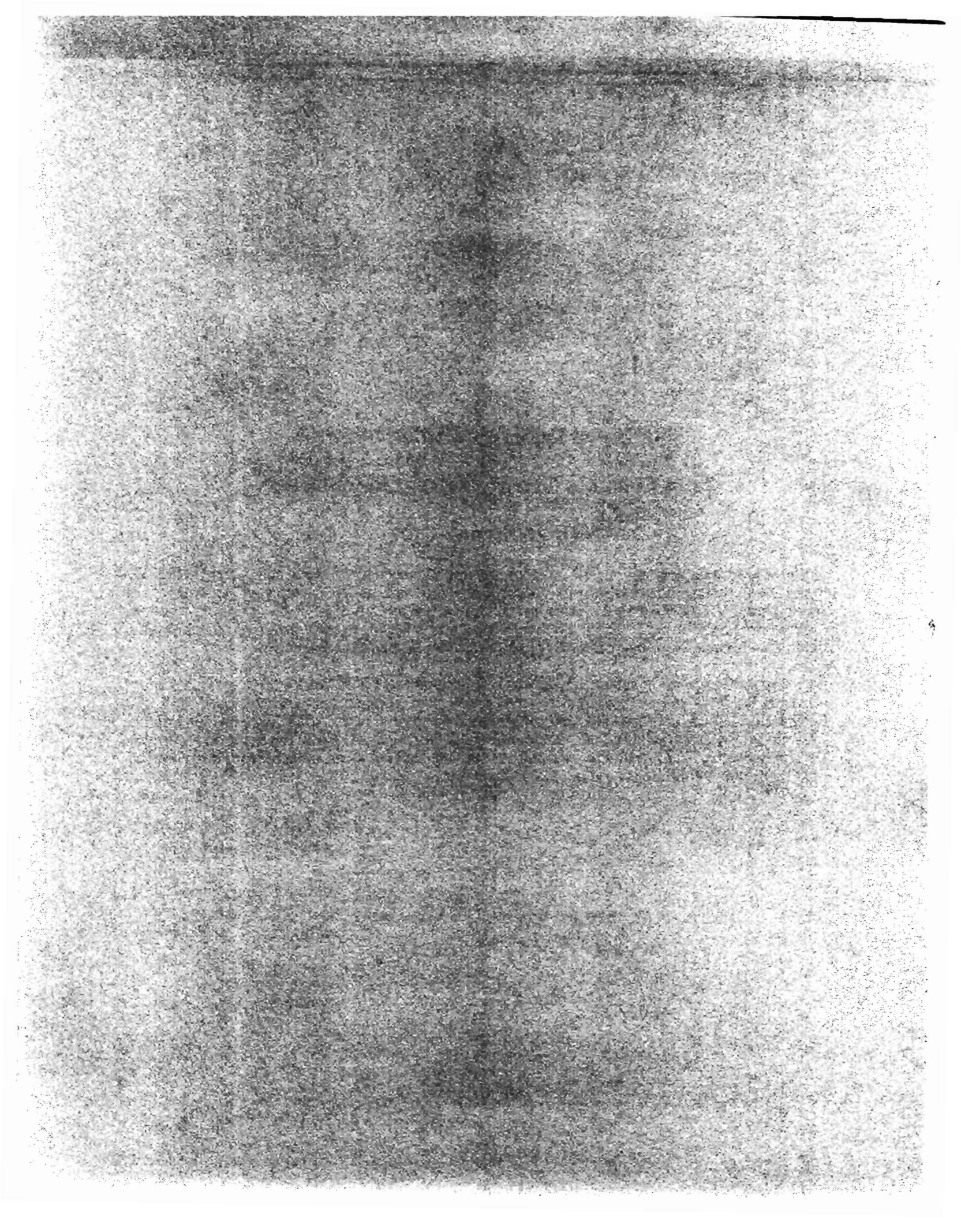
**James Francis Orner, Halger Oelschlaeger**

**Dr B. Struck Co.**

A high speed data acquisition and readout system has been developed using the DSP96002 based Data Stream Processor. In cooperation with and under contract of University of Heidelberg a VXI/VME based system has been specified and implemented. Through a special 64bit Source Synchronous Block Transfer Mode it is possible to read data from the front-end modules with a peak transfer rate of 160Mbytes/s using standard backplanes.

By using a token passing scheme an entire crate with up to 11 VXI- or 19 VME-modules could be read out with minimal addressing overhead. In cooperation with CERN/WA89 and Desy/HERMES two of these DSP building blocks are used for a FASTBUS Readout Engine. This module is designed as an add-on board for the STR330 CH1 FASTBUS master. Through the sophisticated FIFO structure very high sustained data rates can be achieved, e.g., a 50ns FASTBUS slave module is read out at 45Mbyte/s.





# DSP based Data Acquisition Systems

Holger K. Ölschläger, James F. Orner

STRUCK, D-22889 Tangstedt, Germany  
support@struck.de

## Abstract

A high speed data acquisition and readout system has been developed using the DSP96002 based Data Stream Processor.

In cooperation with and under contract of University of Heidelberg a VXI/VME based system has been specified and implemented. Through a special 64bit Source Synchronous Block Transfer Mode it is possible to read data from the front-end modules with a peak transfer rate of 160Mbytes/s using standard backplanes. By using a token passing scheme an entire crate with up to 11 VXI- or 19 VME-modules could be read out with minimal addressing overhead.

In cooperation with CERN/WA89 and Desy/HERMES two of these DSP building blocks are used for a FASTBUS Readout Engine. This module is designed as an add-on board for the STR330 CHI FASTBUS master. Through the sophisticated FIFO structure very high sustained data rates can be achieved, e.g. a 50ns FASTBUS slave module is read out with 45Mbyte/s.

## I. INTRODUCTION

Modern detectors require systems that are able to handle very high data rates and intelligent readout schemes. On the other hand commercial available bus systems should be used to have easy (and less expensive) access to more standard hardware like processors, memory etc. At Crystal Ball event rates up to 10000/s are expected and high resolution spectroscopy must be possible. The chosen VXIbus standard has the advantage of well defined EMR considerations and the VMEbus as data transfer bus, so this system fulfills the requirements for low noise and high speed.

The VXI system consists of two types of modules in one crate, both equipped with the same DSP96002 submodule: The front end interface processors STR8090 and the readout engine STR8080. Up to 11 STR8090 are housed in one VXI crate (up to 19 in a VME crate) and are read by one STR8080. To achieve the desired data rate of 160Mbytes/s, the modules are capable of executing a special 64bit Source Synchronous Block Transfer (SSBLT) mode, nevertheless the modules are in compliance with the VMEbus specification. All initialization tasks, software downloading etc. are done by a standard resource manager like the STR8032.

To use the advantages of FASTBUS in front-end systems, a similar readout scheme is introduced by the CHI-add-on STR330/FRE. This module uses two DSP's to get data from ADC's, TDC's etc. in block transfer mode, do an event formatting and push the data to further event builders.

## II. HIGH-SPEED ENHANCED LINKING PROCESSOR (HELP) - STR8090

The VME/VXI High-speed Enhanced Linking Processor is designed to act as a multi-purpose interface, used to retrieve as master or accept as slave event data from an external source (such as ADC's and TDC's), and process this data (through formatting, zero suppression, data reduction algorithms, etc.) at very fast rates before readout is performed. Using Digital Signal Processing (DSP96002) technology and a sophisticated FIFO architecture, the STR8090 achieves very high sustained data rates. The module is implemented as a register based VXI slave or a VME slave, for communication DSP-VMEbus a interrupt driven mailbox is provided.

### A. Multi-purpose Interfacing

An extra wide 32bit interface on the front panel enables the experimenter to establish a data path from the event source electronics to the board's input FIFO. This permits the module not only to extract data from the source, but also accepts data directly from the source (Push Mode) for data cycles down to 75ns.

All interfacing is driven by a user programmable sequencer, providing 51 additional user defined input/output and 14 VME address lines.

The DSP controls this sequencer by eight command registers, so even very complex timings can be implemented with minimal processor interaction.

### B. DSP Integration

Through the use of the piggyback STR371, the DSP96002 based Data Stream Processor (DDSP), event data can be retrieved or accepted, processed, formatted and suppressed directly on the motherboard. Using a powerful FIFO architecture, the DDSP is able to accept data from the front-end modules via the Input FIFO, process the data, write the completed information into the Output FIFO, and stand ready for the next event. Readout of the DDSP can be performed in both 32bit and 64bit modes, with a peak rate of 160Mbytes/s.

### C. Token Passing Scheme

Multiple STR8090's can be read out via the VMEbus using a fast readout mode employing a token-passing concept. This technique allows readout of all slaves in the crate with one access in block transfer mode. The Output FIFO supports

64bit VMEbus data transfers which ensures maximum data transfer speed as well as normal VME 32bit transfers. In either case, the data transfers are performed autonomously through the use of a DMA controller on the STR8080 module. The readout is performed in parallel to the DSP data treatment.

### III. MULTIPLE OUTPUT READOUT ENGINE (MORE) - STR8080

The VME/VXI readout engine is designed to read out, process and accelerate data produced by front-end modules, such as the STR8090 Linking Processor to external storage devices or higher level of processing. In addition to the standard 32bit transfer mode, the STR8080 is equipped to use a special 64bit Source Synchronous Block Transfer (SSBLT) mode to read data from the front-end modules with peak data rates up to 160 Mbytes/s. Through the use of the token-passing protocol, it is possible to read an entire crate with minimal addressing overhead.

#### A. Data Output Interfacing

Flexibility is one of the most important design goals of the STR8080, so a wide range of output interfaces are available for the use with the readout engine, examples include the local VSBus, DT32 Differential ECL (either as source or controller) or VSB Differential Bus, VICbus, HIPPI and SCI are under preparation. If the data must be pushed over long distances to its destination, the STR8080 can be equipped with an optical point-to-point link.

#### B. DSP Integration

The STR8080 uses the same DSP piggyback board as the STR8090, so the same features like high processing power and the FIFO structure is available on the STR8080.

### IV. FASTBUS IMPLEMENTATION

The system structure is similar to the VME/VXI architecture described above. Again a general purpose master like a STR330/CHI does all the initialization etc., a number of front-end modules take data from the experiment and a readout-engine collects and pre-processes these data and pushes it to a higher level event builder.

#### A. STR330/CPU CHIPS Processor

This single slot module acts as a FASTBUS „resource manager“ initializing the crate, communicating via LAN with high level control and controlling the readout-engine.

#### B. STR330/FRE FASTBUS Readout Engine

The STR330/FRE is designed as an add-on module of the STR330/CPU, speed optimized for readout tasks in 32-bit

block transfer mode, combined with two DSP96002 submodules. FASTBUS readout speed is about 35ns plus the Slave DS/DK delay time, so the readout of a 50ns FASTBUS slave results in a peak transfer rate of about 45Mbytes/s. The flexible design of data paths allows to push data either to point-to-point links like HIPPI or DT32, into the data memory of the STR330/CPU or to the integrated FASTBUS cable interface.

#### C. STR330/FOL Fibre Optic Link

Based upon the CERN design of optical source and destination piggy-backs, this module transfers data to and from the STR330/CPU data memory with a speed of up to 10Mbytes/s via optical fibres. Each optical channel has a command and a data path, so the full bandwidth could be used for data. The STR330/FOL reads and writes data in DMA mode.

### V. CONCLUSION

The DSP based readout concept uses standard bus systems, high data rates are possible by using sophisticated FIFO structures. Through the use of a source driven block transfer mode the bandwidth of the VMEbus is doubled and a complete crate readout could be done with minimal addressing overhead. All described system modules are under production and commercially available.

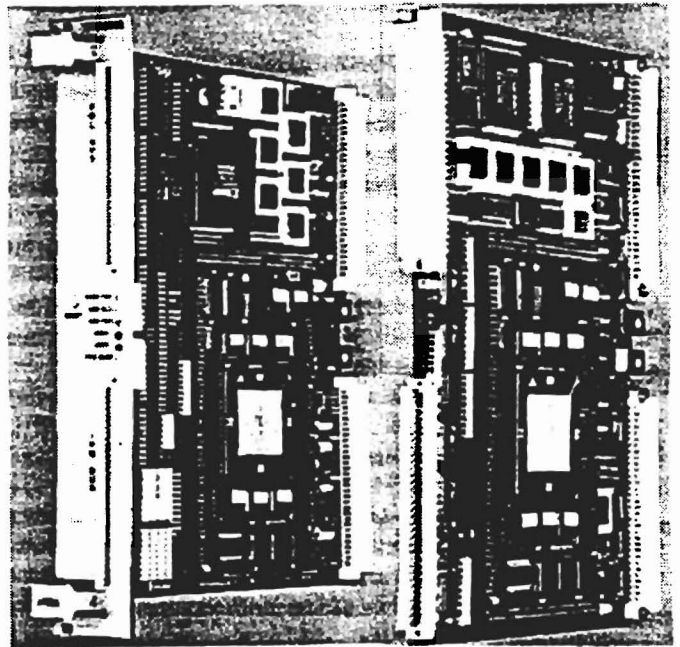
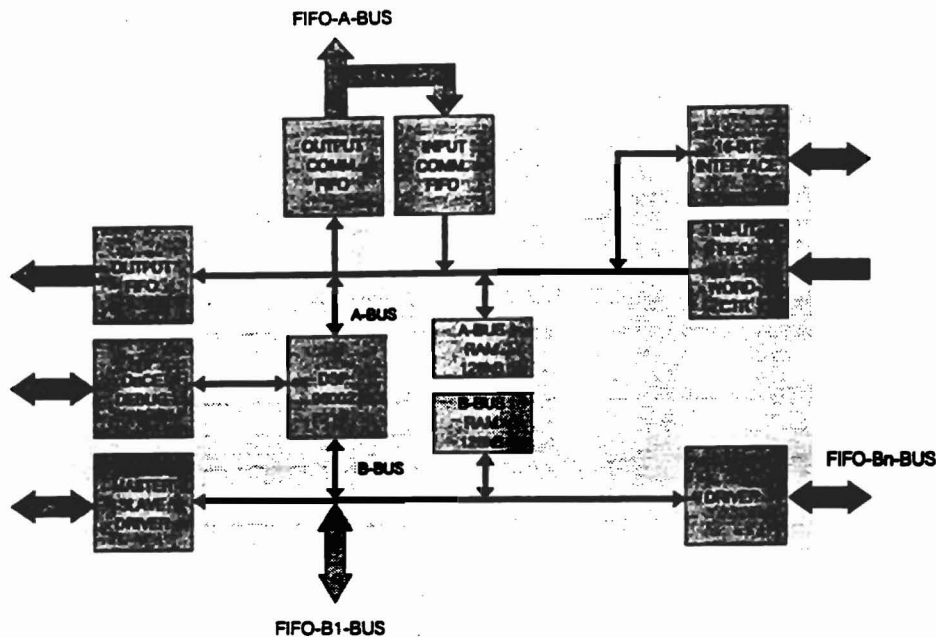


Fig.1: STR8090 (left), STR8080 (right)

### VI. REFERENCES

- [1] Dr. Christoph Ender, „VXI Electronics for EUROBALL, full Specification“, February 1994
- [2] J. R. Alexander, „EUROGAM Project: Specification of the VXI Readout Mechanism“, Version 1.0, April 1991

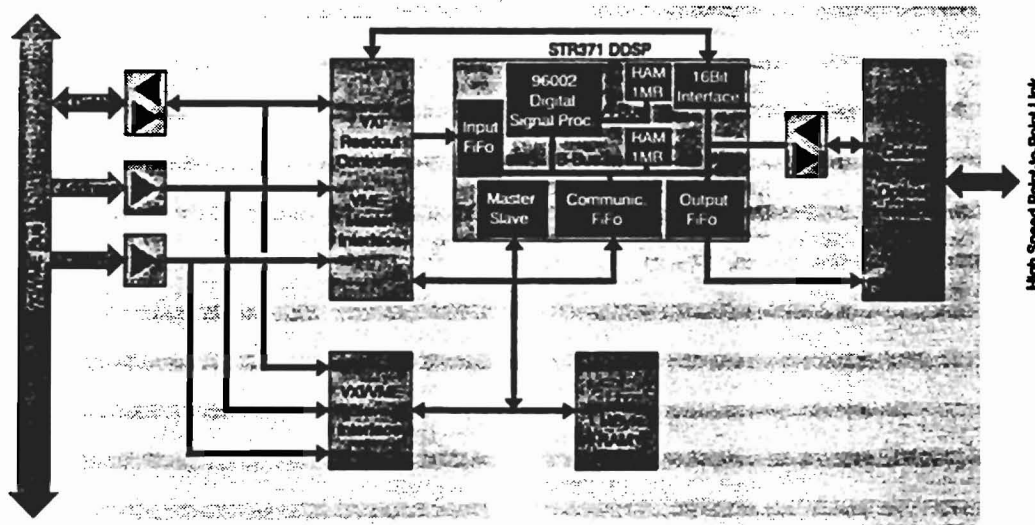
# STR371 DSP Piggy Back



- Motorola 96002/40 MHz DSP
- 128kB A-Bus RAM, Zero Wait States
- Up to 1MB B-Bus RAM, Zero Wait States
- 32Bk\*1k Input Fifo, Peak Transfer Rate 133MByte/s
- 32Bk\*1k Output Fifo, Peak Transfer Rate 133MByte/s
- 32Bk\*1k Communication Fifo for adjacent STR371's
- B-Bus Master/Slave Interface for external Access to all onboard/onchip Resources



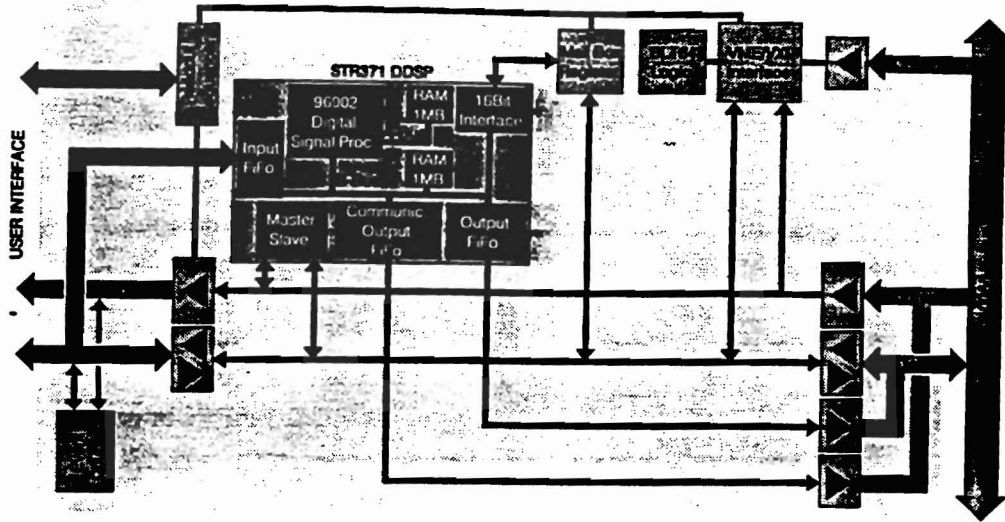
# STR8080 VXI/VME Readout Engine



- Powerful DSP96002 Submodule for Data Preprocessing
- Pipeline Architecture for fast Data Readout
- 64Bit VXI Readout, 160Mbyte/s (SSBLT like)
- Token Passing Scheme for Event Readout
- Register based VXI Slave, VSB Slave, VME Master and Slave
- Crate Readout interface (DT32, Optical Link, SCI Node)
- 4 MB local dual ported RAM



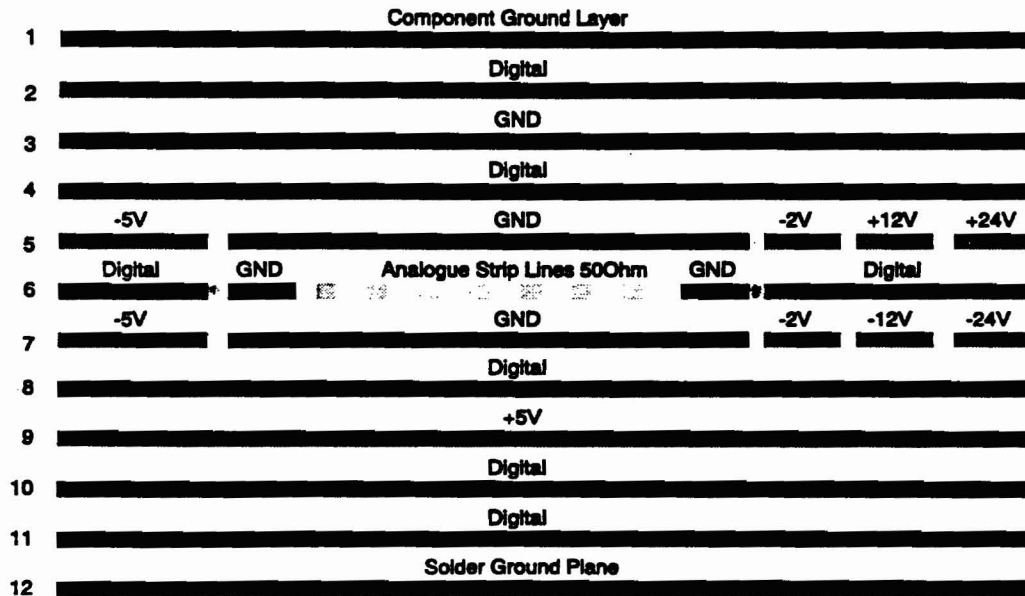
# STR8090 VXI/VME Data Compressor



- Powerful DSP96002 Submodule for Data Preprocessing
- Pipeline Architecture for fast Data Readout
- 32Bit Input Data Path
- 64Bit VXI Readout, 160Mbyte/s (SSBLT like)
- Token Passing Scheme for Event Readout
- Interrupt driven Mailbox Communication VXIbus - DSP
- Register based VXI Slave

## Layer Topology STR8090

Component Side

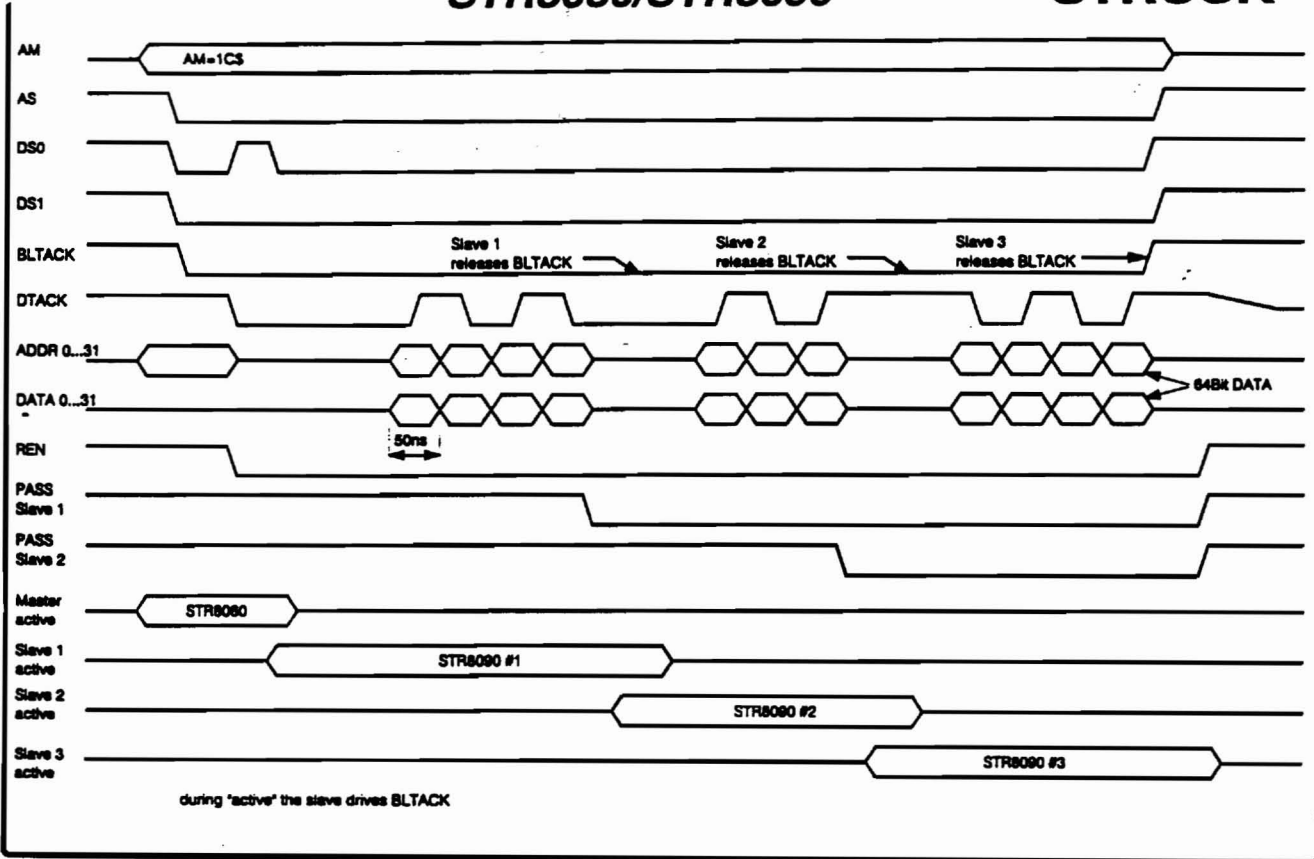


Solder Side

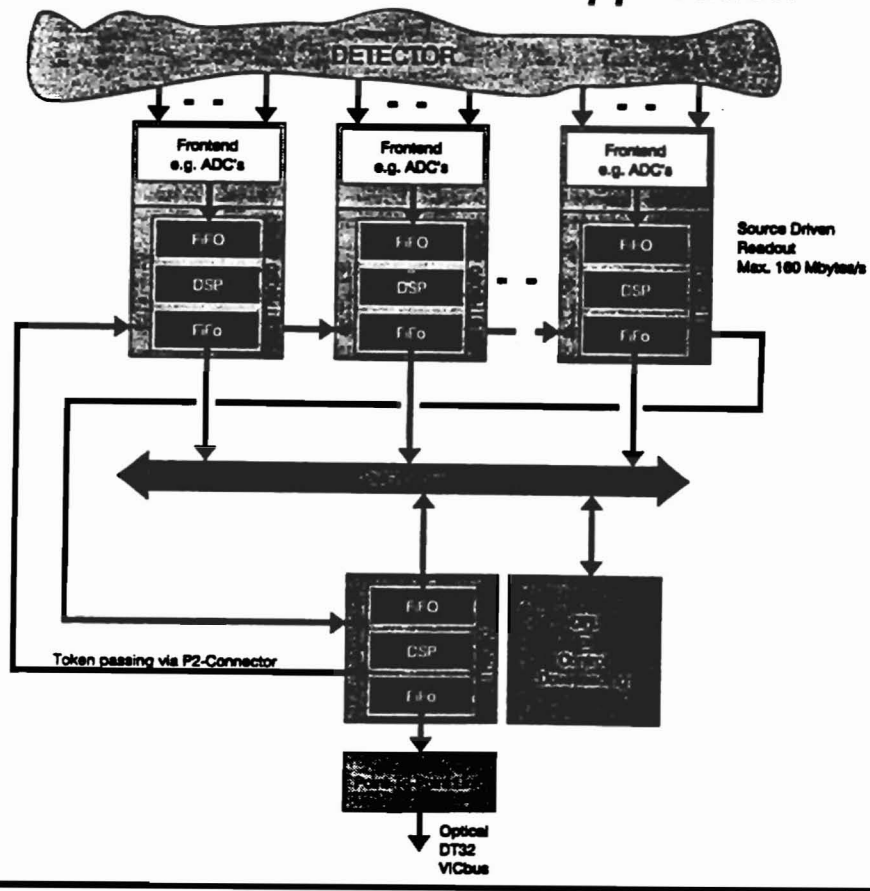
Note: To minimize noise all clock lines have parallel ground lines



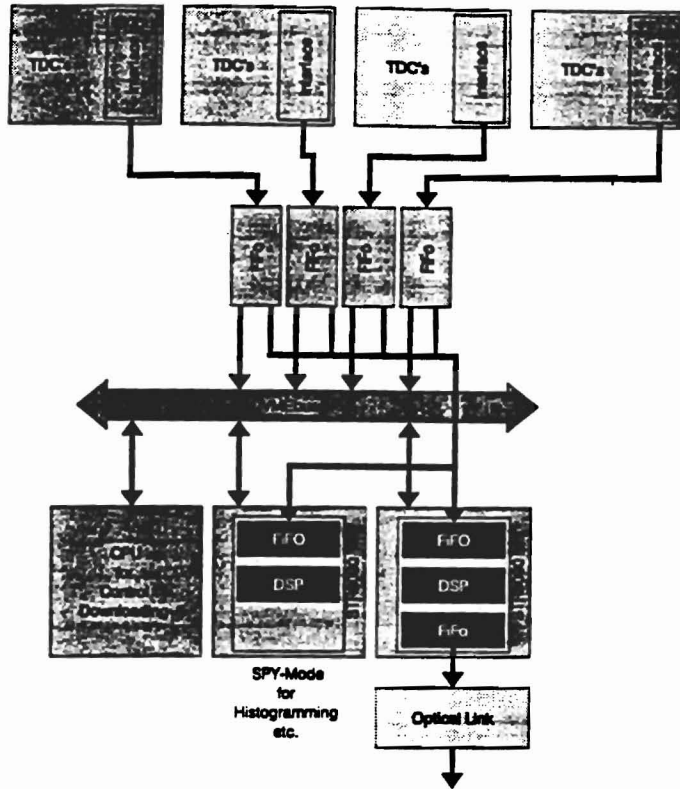
# STR8080/STR8090



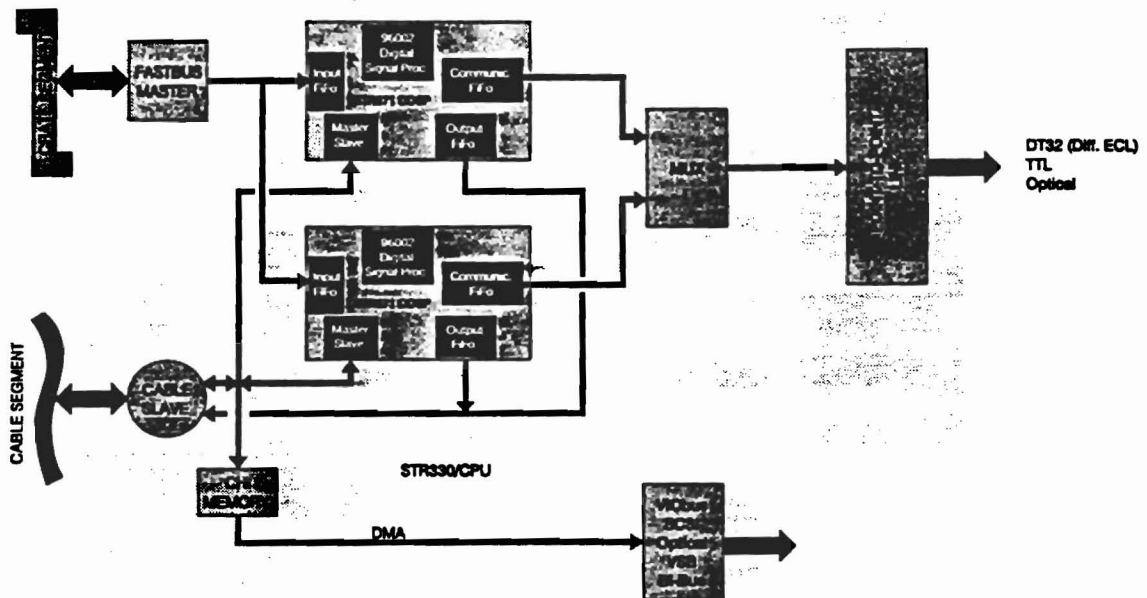
## STR8080/STR8090 Application



# STR8090 Application NA48 (Univers. of SIEGEN)

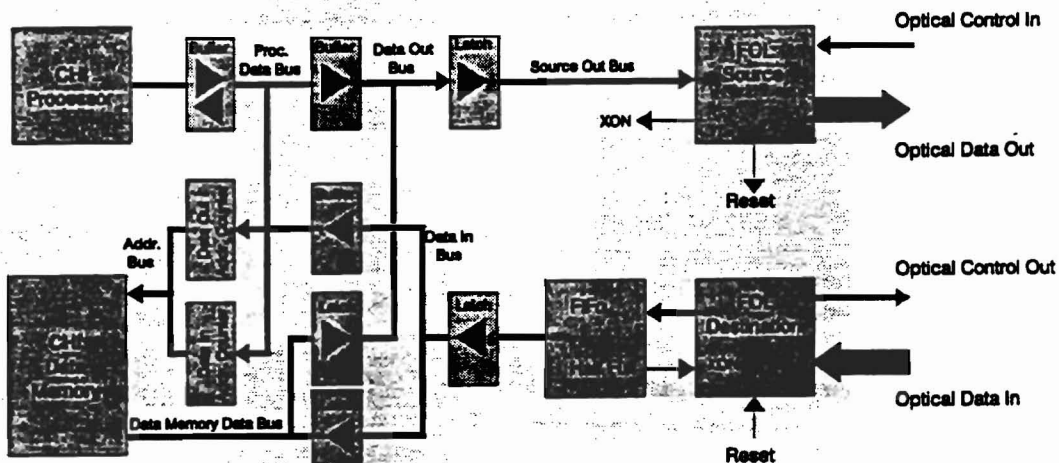


# STR330/FRE FASTBUS Readout Engine



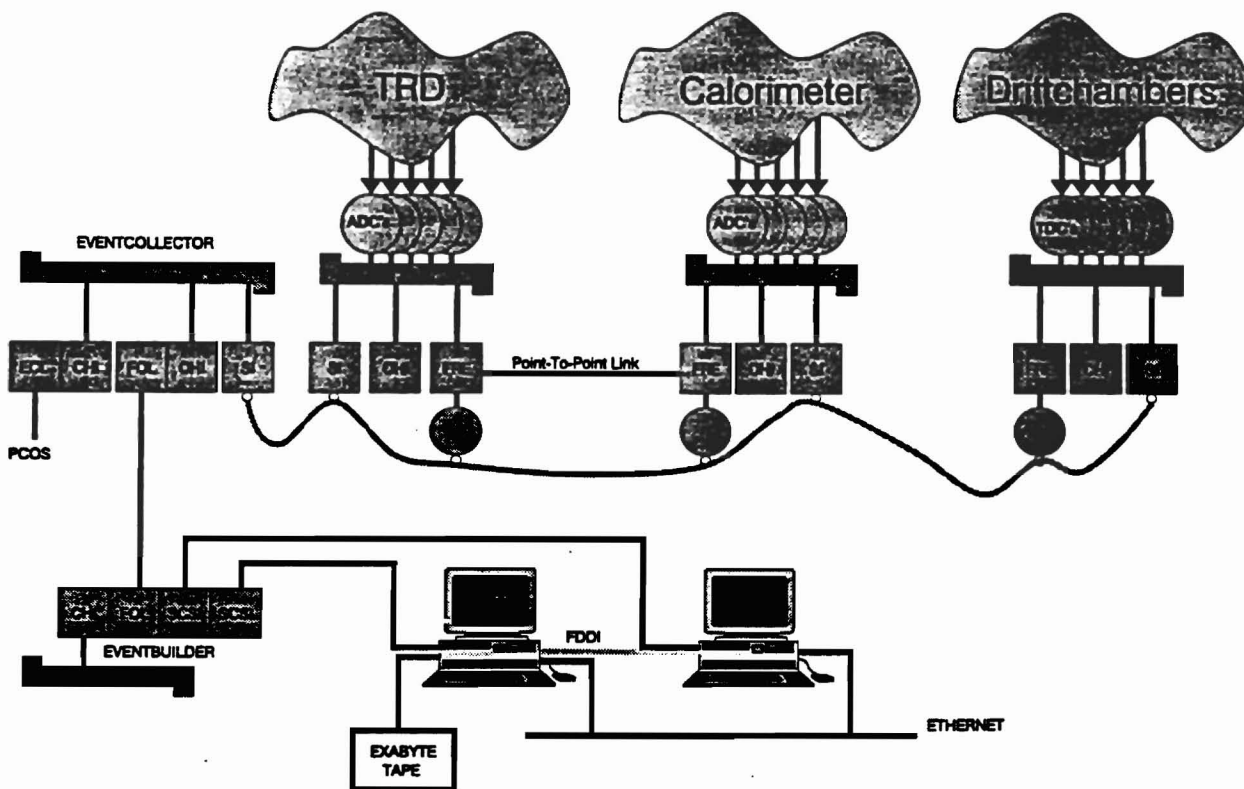
- Single Slot Add-on for STR330 CHI
- High Speed FASTBUS Readout Controller
- Two Powerful DSP96002 Submodules for Data Preprocessing
- Readout over High-Speed Point-to-Point Link
- Transfer Peak Rate: 37.5ns + Slave DS/DK Delay Time
- Cable Segment Slave
- DSP Software Support

# STR330/FOL 10 MByte/s Fibre Optic Link



- 10 MByte/s DMA Data Transfer Speed
- Fibre Length up to 2km
- Separate Fibres for Data and Control
- Error Detection Scheme on Data Channels
- Full Software transparent Protocol
- Command / Data Word Distinction

# FASTBUS Readout Application HERMES (Desy)







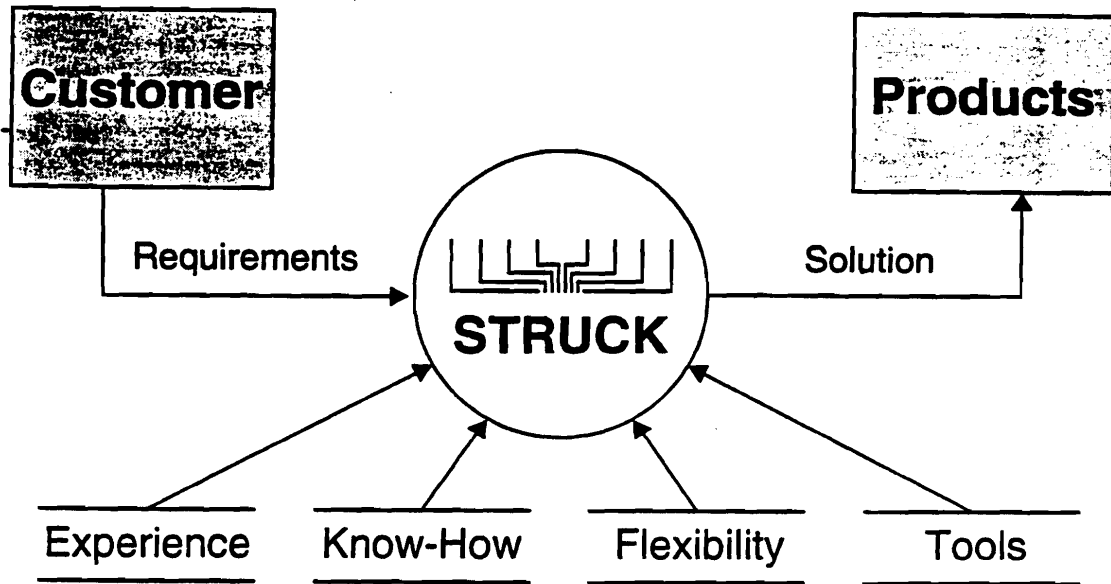
More than a good link

**STRUCK**

Electronics for High Energy Physics  
and Industry

**Hardware / Software**

**Development - Manufacturing - Sales**



**Founded:**  
**Employees:**

1974 by Dr. Bernd Struck  
a total of 33  
including 13 Engineers / Physicists

**Dr. Bernd Struck:**  
**Contacts:**  
Volker Wöbber:  
Holger Ölschläger:  
Ronald Ölschläger:  
Karl Heinz Friedrichs:

Head and Owner of the Company  
Production Department  
Development Department  
Marketing / Sales Department  
Purchasing / Finance Department

**Mailing Address:**

Dr. Bernd Struck  
Postfach 1141  
D-22886 Tangstedt  
Bäckerberg 6  
D-22889 Tangstedt

**Delivery Address:**

**Communication:**

Telefon: (04109) 55-0  
Telefax: (04109) 55 33  
Telex: 2 180 715 tegs d  
E-Mail: sales@struck.de;  
support@struck.de

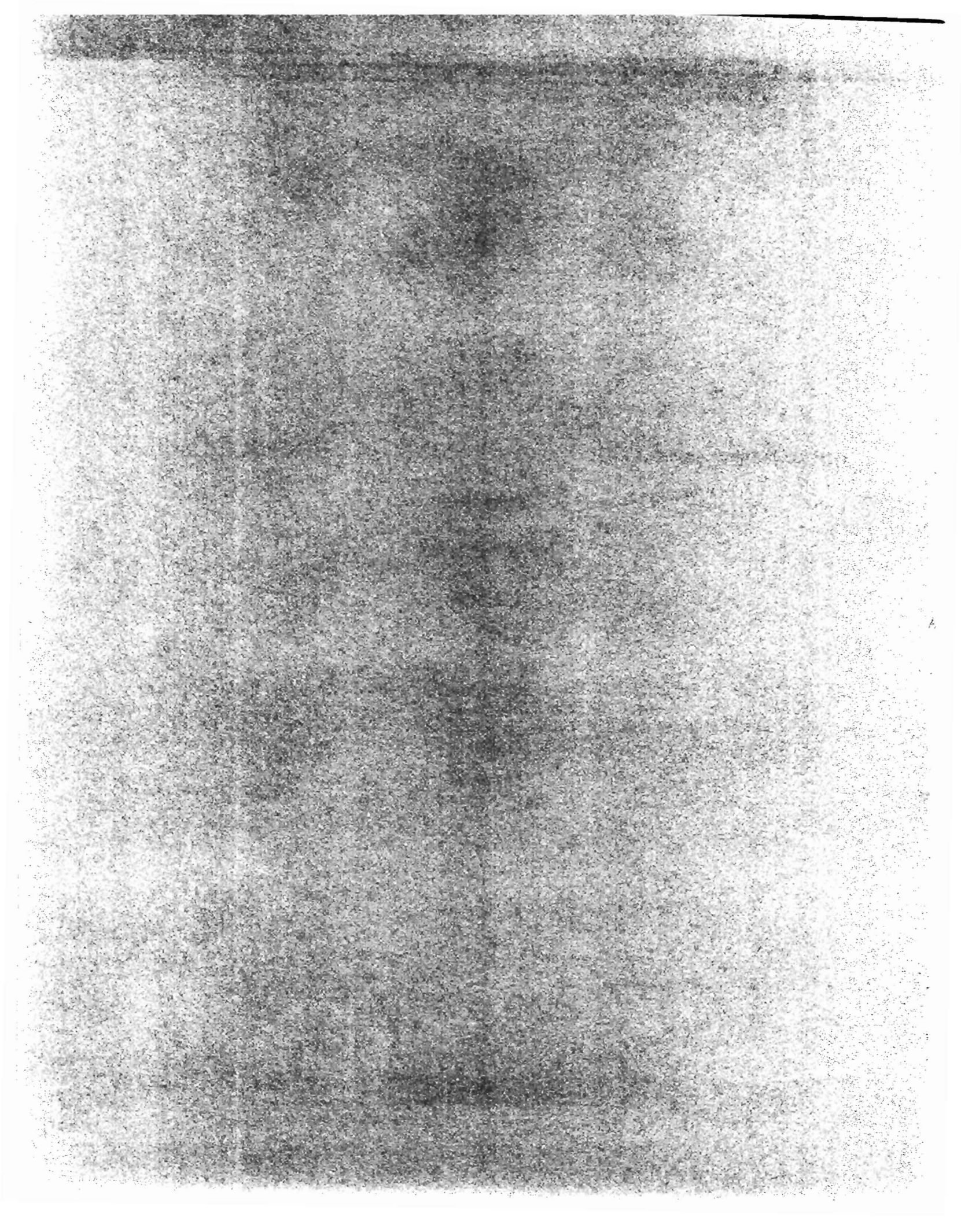
Since 1974, the firm **Dr. Bernd Struck** has been an instrumental partner in the development, production and world-wide sales of hardware and software components for basic science physics research and industrial applications. We specialize in ultra-fast data acquisition systems (Flash Analog-to-Digital Converters up to 1 GHz), digital interfaces and signal processing technologies in standardized bus systems (FASTBUS, VME-bus, VXIbus, CAMAC, etc.) and optical inspection systems.

**Initial Experiences With a Network of INMOS C104 Packet Routing  
Switches**

**Roger Heeley**

**CERN**

The C104 is an asynchronous 32-way packet routing switch developed by INMOS, giving 100MBit/s over all of the 32 Data Strobed links. It also has the added functionality of universal and grouped adaptive routing. The performance of this switch will be discussed within the framework of the GPMIND project at CERN and the future prospects of this switch will also be presented.



---

# Initial Experiences With The IMS C104 Packet Routing Switch

---

R.W.Dobinson, D.Francis, R.Heeley, W.Lu,  
M.P.Ward

---

## Abstract

The C104 is an asynchronous 32-way packet routing switch with data strobed (DS) links operating at 100 Mbits/s developed by Inmos. It supports Universal and Grouped adaptive routing to avoid network congestion. The performance of this switch is presented within the framework of the GPMIMD project at CERN and the future application of this switch will also be discussed.

## 1.0 Introduction

---

The IMS C104 is a programmable single-chip VLSI device which can interconnect up to 32 devices, including other instances of itself. Its features are:

- 32 Way Asynchronous Packet Switch,
- 32 x 100 Mbits/s Serial Bi-directional Links
- 300 Mbytes/s Bandwidth
- Supports Variable Packet Length
- Less Than 1  $\mu$ sec Packet Latency
- Wormhole Interval Routing Algorithm
- Implements Universal & Grouped Adaptive Routing
- Non-blocking Crossbar
- Concurrent Processing of Packets
- Separate Control System
- Bit and Packet Level Error Handling
- Highly Configurable: 28Kbits User-programmed Data

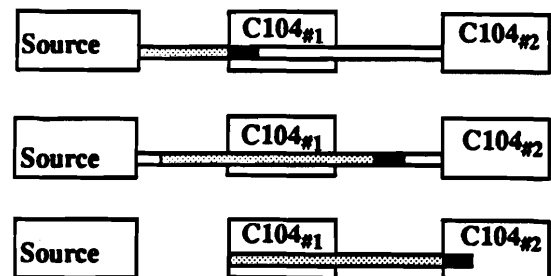
### 1.1 Interval Labelling

Interval labelling is the technique used to route packets through a C104 i.e. to choose the required output link. Each output link of a C104 is assigned a range of device labels (an interval). A device label identifies one particular device accessible via that physical link. When a packet enters a C104, the header is compared with these intervals and the output link with the interval in which the header of the packet lies is selected

### 1.2 Wormhole Routing

In this method a routing decision is taken as soon as a packet enters a C104 (see fig.1). A temporary circuit is created through the C104, as the end of the packet is pulled through, the circuit vanishes. In addition, a single packet may be passing through multiple C104's at any one time. The header of a packet may also be received by the destination before the whole packet is transmitted, hence minimizing the latency.

FIGURE 1. Wormhole Routing



### 1.3 Universal Routing

Universal (or two phase) routing is implemented to avoid communications bottlenecks or hot spots in large networks. It does this by spreading the traffic entering a network at a particular point.

A packet entering a two phase network has a random header added upon entering the first phase. The packet is

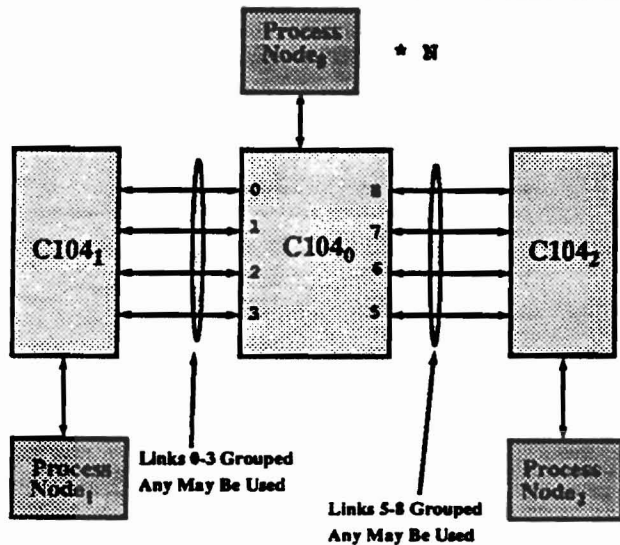
then automatically routed to a randomly chosen intermediate destination. In this way the load is balanced across the network, giving bandwidth and latency improvements under high load conditions. Conversely peak bandwidth and latency performance are reduced under low load conditions.

The additional header generation required at the first phase and the header deletion required at the second phase are performed by hardware in the C104's.

### 1.4 Grouped Adaptive Routing

In switching networks there will often be many possible routes that a packet may take to reach a certain destination. It is desirable that should one of these links be in use or in error then an alternative link is chosen. To fulfill this requirement the C104 supports grouped adaptive routing. Output links can be grouped so that the packets routed to the first link of the group can be routed to other links of that group in the case that the first link is not available (see fig. 2). Grouped adaptive routing, thus provides a level of automatic fault tolerance on the links and improved network performance in terms of latency and throughput.

FIGURE 2. Grouped Adaptive routing



## 2.0 Initial Implementation

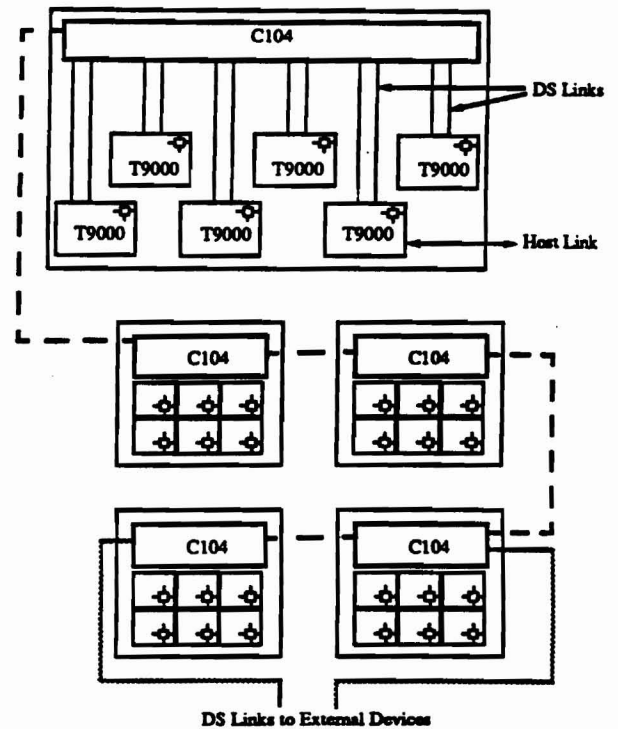
Initial experience with the C104 has been gained via the implementation of a processor farm in the CPLEAR experiment at CERN.

Thirty T9000 processors, the latest generation of the trans-

puter from Inmos, and 5 C104's for networking, were used as a real-time processing farm performing standard CPLEAR data production and filtering. The C104 provided the exclusive method of communication and control between the T9000's. The inter-connectivity offered by the C104 removed any requirement for through-routing software, as has been necessary with previous generations of the transputer. The 30 T9000 were housed in five modules (see fig.3), each module containing 1 C104 and 6 T9000's.

During a three week run in September/October 1994 the system processed twenty million events in real-time. In this period of running a stable platform was achieved, after which the system ran for 128 hours with no failures.

FIGURE 3. Initial CPLEAR Implementation



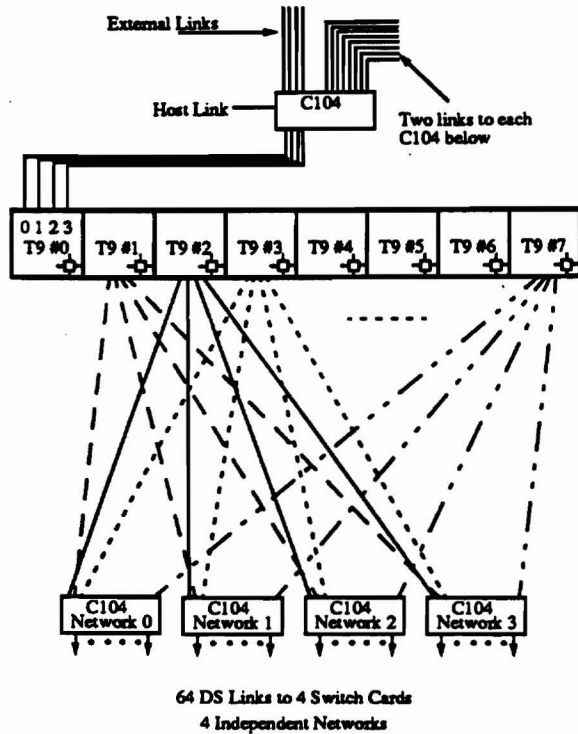
## 3.0 The GPMIMD Machine

The GPMIMD machine is being developed as part of the ESPRIT program. It will consist of 64 T9000 processors, with 56 C104's providing full inter-connectivity and is currently being assembled at CERN. Eight motherboards (see fig. 4) will each carry 8 T9000's and 5 C104's. Four switch cards each carrying 4 C104's provide connectivity between the mother cards.

This architecture gives four independent networks allowing; fault tolerance, communication priorities and global shared memory.

During three weeks in September/October 1994, a partial machine processed events in real-time from the CPLEAR experiment. This machine had 24 processing nodes on 3 motherboards. It ran for 128 hours without any failures.

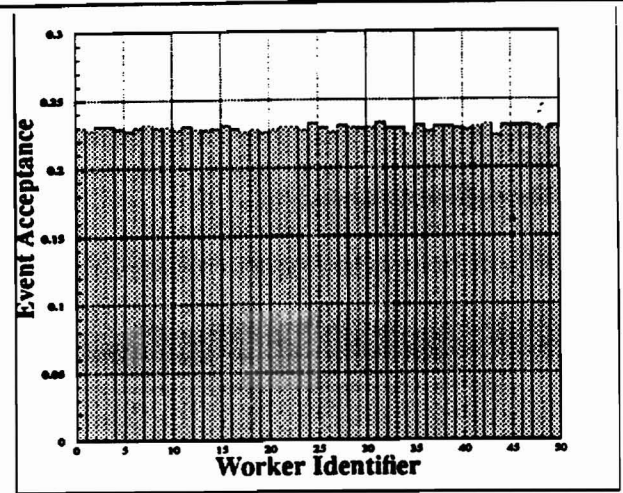
FIGURE 4. A GPMIMD Motherboard



This system was combined with the system described in section 2, thus a processor farm comprising of 54 processing nodes and 20 C104's was implemented.

As a method of monitoring the performance of the processor farm, the acceptance of the CPLEAR production code as seen by each processing node was monitored. This was achieved by a process running on a T9000, which via the C104 network, summed the monitoring histograms from the processing nodes. The resulting acceptance as a function of Worker Identifier is shown in figure 5.

FIGURE 5. Worker Acceptance versus Worker ID

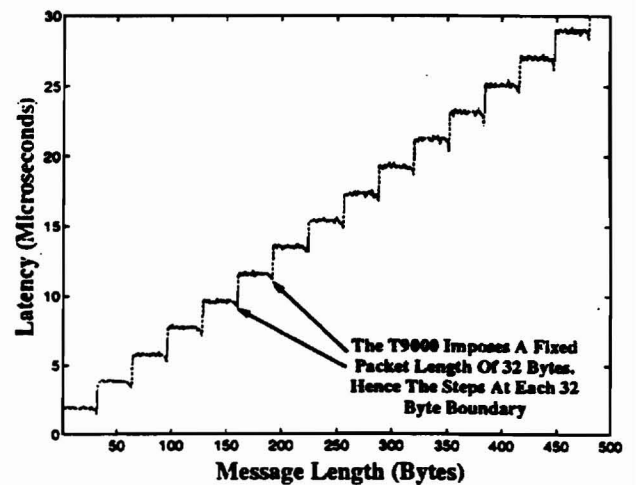


## 4.0 C104 Performance

### 4.1 Latency of the C104.

The latency of the C104 was measured by comparing the time for a single packet to pass between two processes each on a separate T9000. These measurements were made for T9000's connected directly and T9000's that were connected using C104's. The minimum latency for a single packet was measured to be 1µsec. This result is shown in figure 6. In understanding figure 6, one must bear in mind that an acknowledge packet is transmitted for every data packet transmitted.

FIGURE 6. C104 Latency versus Message Length

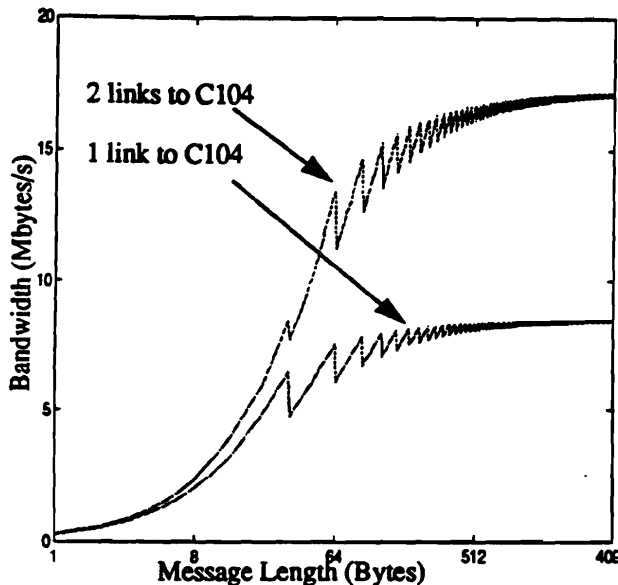


## 4.2 Bandwidth measurement

Figure 7 shows the bandwidth available to T9000 processors over one and two links via a C104. This is a measure of the usable bandwidth and not just the raw rate at which the links operate. The measurements are uni-directional and for 5 virtual channels per physical link. A virtual channel is single logical communication channel mapped onto a physical link.

It is clear that the performance of a C104 link is not affected by using another link on that C104.

**FIGURE 7.** Bandwidth Measurement



## 5.0 Summary & Outlook

We have implemented a network of 20 C104's connecting 54 T9000 transputers. This platform was stable in a real experimental environment.

The functionality of Universal and Grouped adaptive routing are expected to minimize the problems of congestion in large networks. The MACRAME project will investigate event building studies and second level triggering for LHC experiments using large networks of C104's.

The Data Strobed (DS) link technology is becoming an IEEE standard (P1355). The performance of the DS link will improve from its present 100 Mbits/s to 200 Mbits/s (uni-directional) by the middle of 1995.



# **The C104 Packet Routing Switch**

## **and**

# **Initial Applications**

**R. Heeley and D.Francis**





## Participants



<b>R. Dobinson</b>	<b>CERN/Univ. of Liverpool</b>
<b>S. Fisher</b>	<b>Univ. of Liverpool</b>
<b>D. Francis</b>	<b>CERN</b>
<b>R. Heeley</b>	<b>Univ. of Liverpool/CERN</b>
<b>W. Lu</b>	<b>CERN</b>
<b>B. Martin</b>	<b>CERN</b>
<b>M. Ward</b>	<b>Univ. of Liverpool/CERN</b>

<b>Inmos</b>	<b>U.K.</b>
<b>Parsys</b>	<b>U.K.</b>
<b>Telmat</b>	<b>France</b>

<b>Contact Person</b>	<b>R.Dobinson</b>
<b>email</b>	<b>bobdob@cernvm.cern.ch</b>
<b>phone</b>	<b>+(41) 22 767 3066</b>



University of Liverpool

## The C104 Specification



- ☛ **32 Way Asynchronous Packet Switch, With 100 Mbits/s Serial Bi-directional Links**
- ☛ **300 Mbytes/s Bandwidth**
- ☛ **Variable Packet Length, Less Than 1  $\mu$ sec Packet Latency**
- ☛ ***Interval Labelling***
- ☛ ***Wormhole Routing Algorithm***
- ☛ ***Implements Universal Routing & Grouped Adaptive Routing***
- ☛ **Non-blocking Crossbar**
- ☛ **Concurrent Processing of Packets**
- ☛ **Separate Control System**
- ☛ **Bit and Packet Level Error Handling**
- ☛ **Highly Configurable: 28Kbits User-programmed Data**

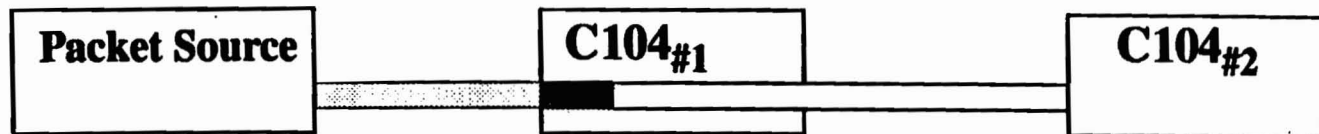


# Wormhole Routing

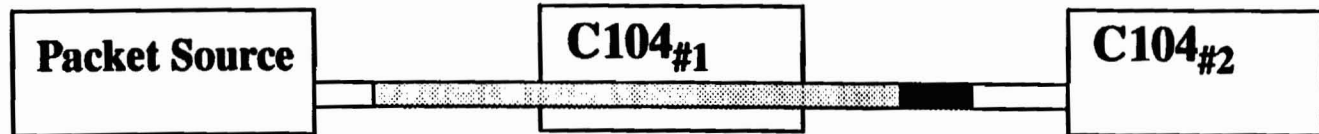


- ☛ **Routing Decision is Taken as Soon as The Header of the Packet is Input**
- ☛ **The Packet Header Creates a Temporary Circuit Through the C104 Network. As the End of The Packet is Pulled Through the Circuit Vanishes**
- ☛ **A Single Packet May be Passing Through Multiple C104's at Any Time. The Head of a Packet May be Received by the Destination Before the Whole Packet is Transmitted, Hence Latency is Minimised.**

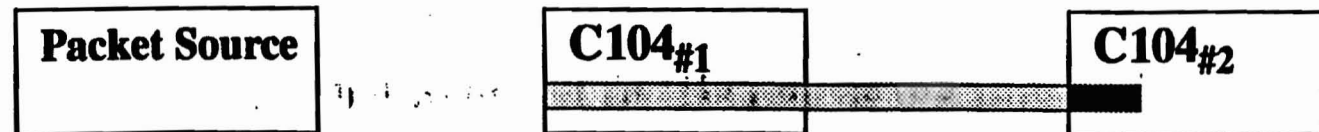
1) Packet Header Arrival Causes Routing Decision to be Taken, Creating a Temporary Circuit Through the C104



2) Packet Sent Directly to Output



3) Tail is Pulled Through and Circuit Vanishes. Header May Enter Next Switch Before Tail Leaves Previous Switch





## Universal Routing



- ☛ **Universal (Two Phase) Routing is Also Implemented to Avoid Communication Hot Spots or Bottlenecks in Large Networks**
- ☛ **A Packet Entering a Two Phase Network Will Have a Random Header Added Upon Entering the First Phase**
- ☛ **The Packet is Then Automatically Routed to a Randomly Chosen Intermediate Destination**
- ☛ **Thus the Load is Balanced Across the Network, Giving Bandwidth and Latency Improvements Under High Load Conditions [Peak Bandwidth and Latency Performance Will be Reduced Under Low Load Conditions]**
- ☛ **The Additional Header Generation Required at the first Phase and the Header Deletion Required at the Second Phase are Performed by the C104's**



# Applications

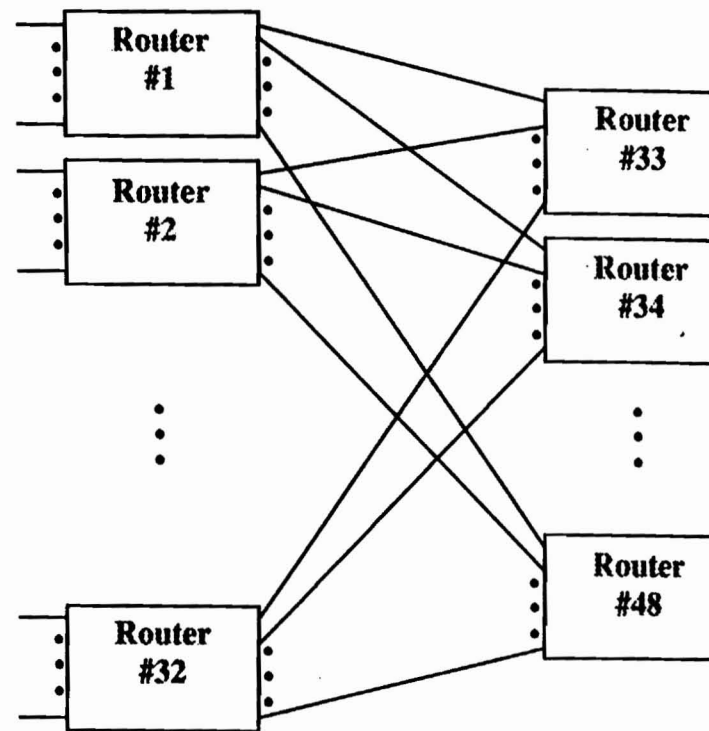


From These Switching Nodes Large Networks of Varying Topologies may be Built

Supports Scalable Architectures in Which Communication Throughput Must be Balanced With Processing Throughput.

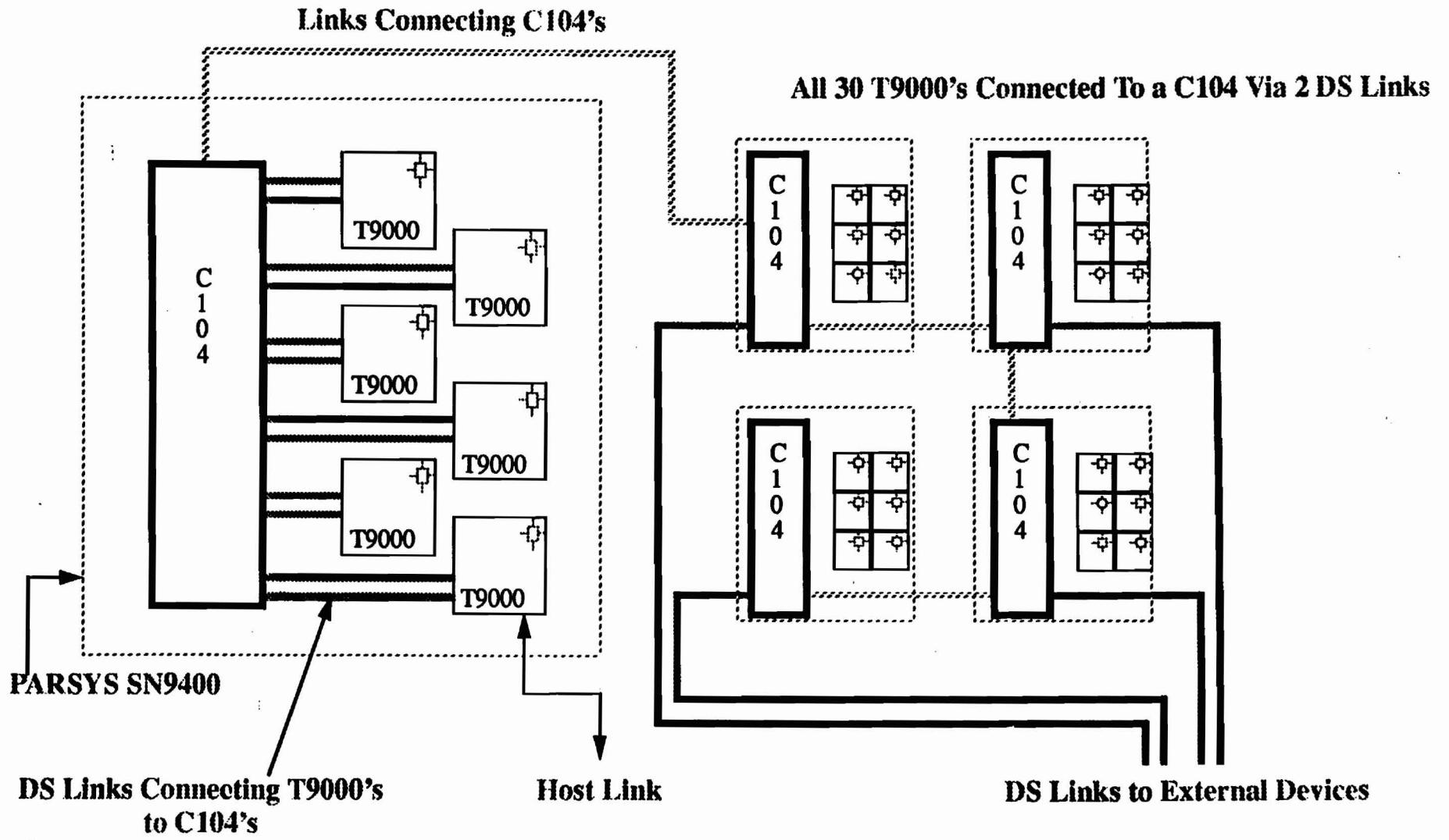
In Such Architectures, it is Known That Overall Communication Capacity Must Grow Faster than the total Number of processors - a Larger Machine Must Have Proportionally More Routers

**Switching Network Technology  
Could be Applied to Data Acquisition  
Techniques in the Next Generation of  
HEP Experiments.**



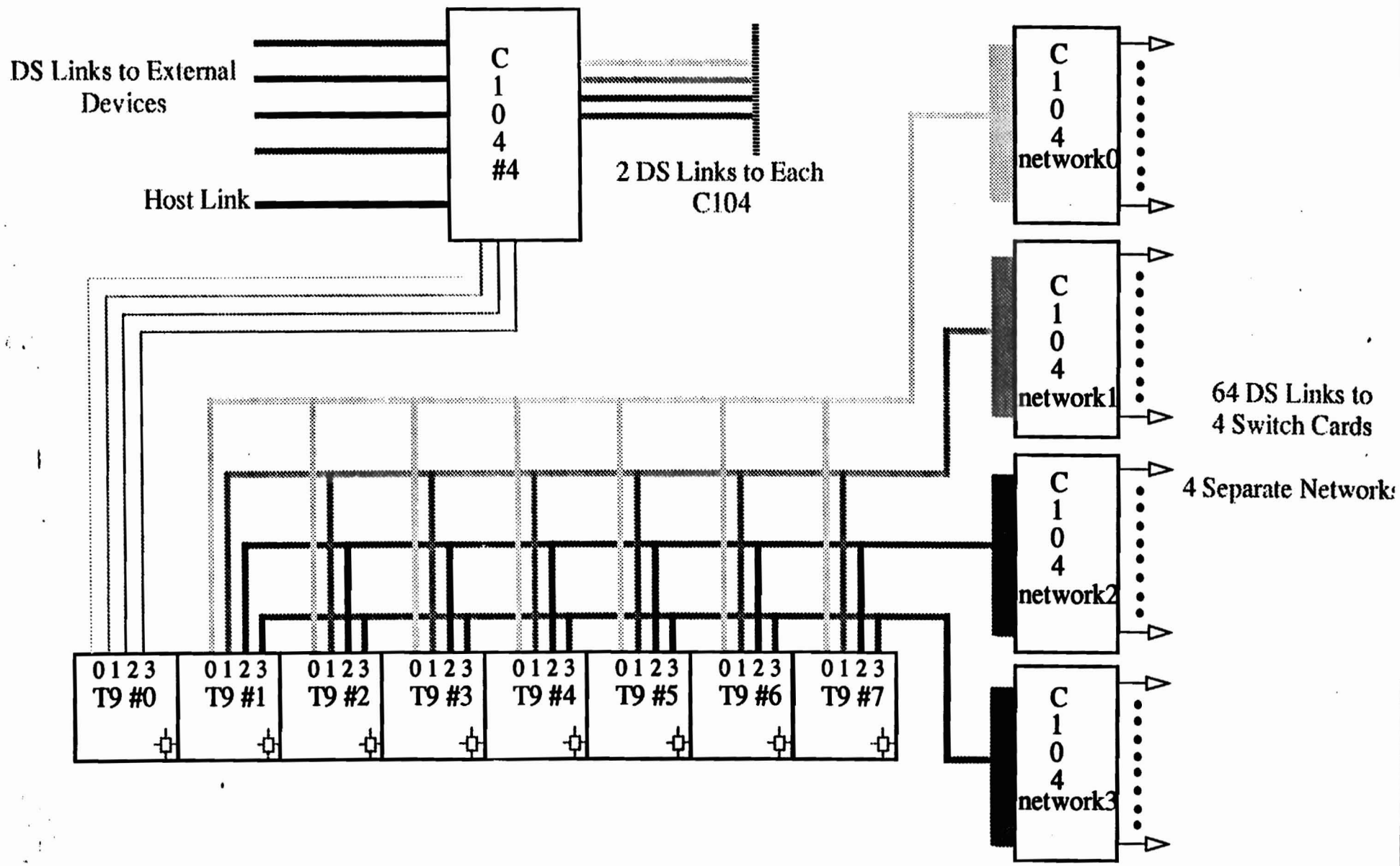
Example: 48 C104's maybe connect 512 sources with only 3 routing delays

# Initial CPLEAR Layout





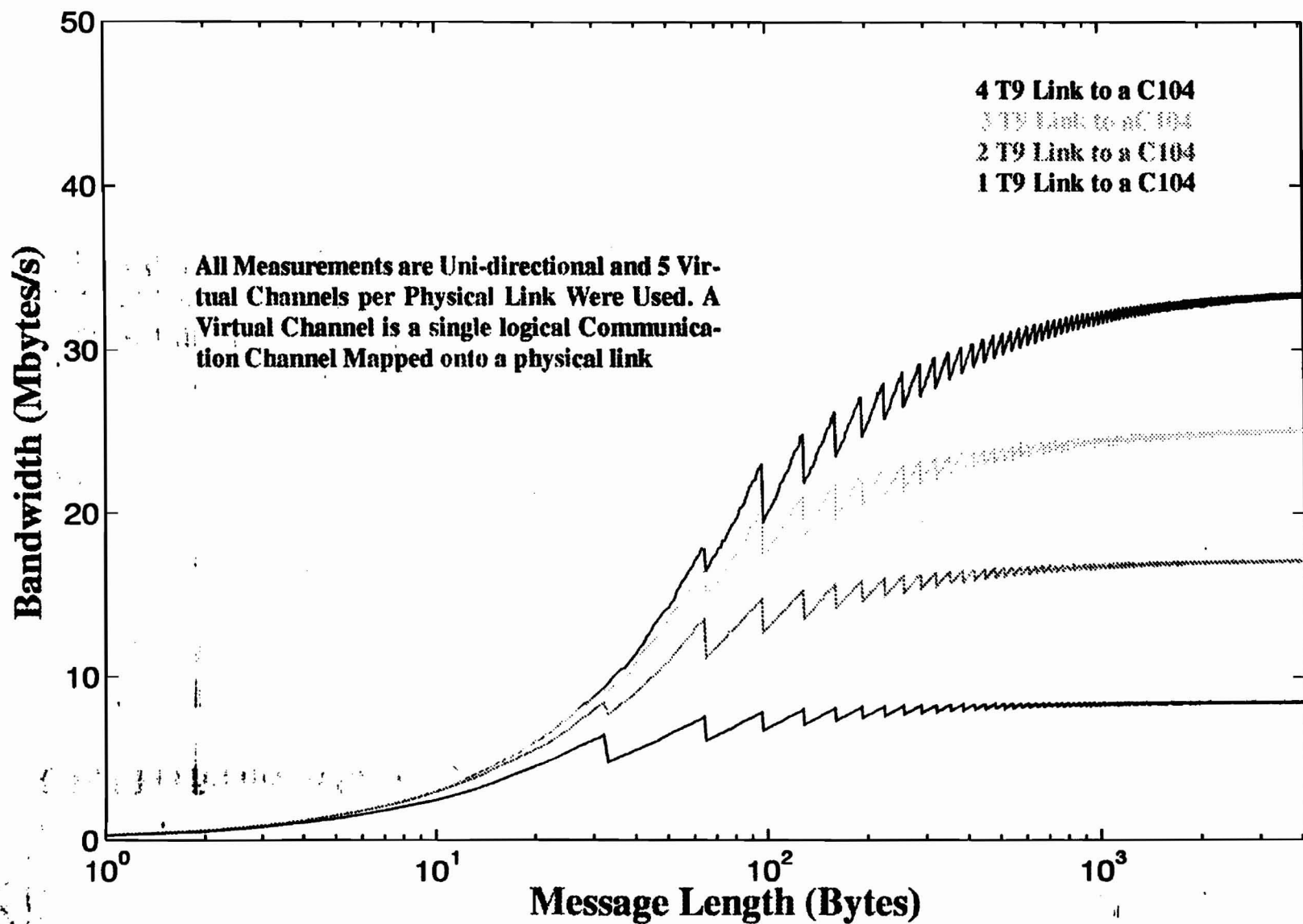
# The GPMIMD Mother Board





University of Liverpool

# Link Performance





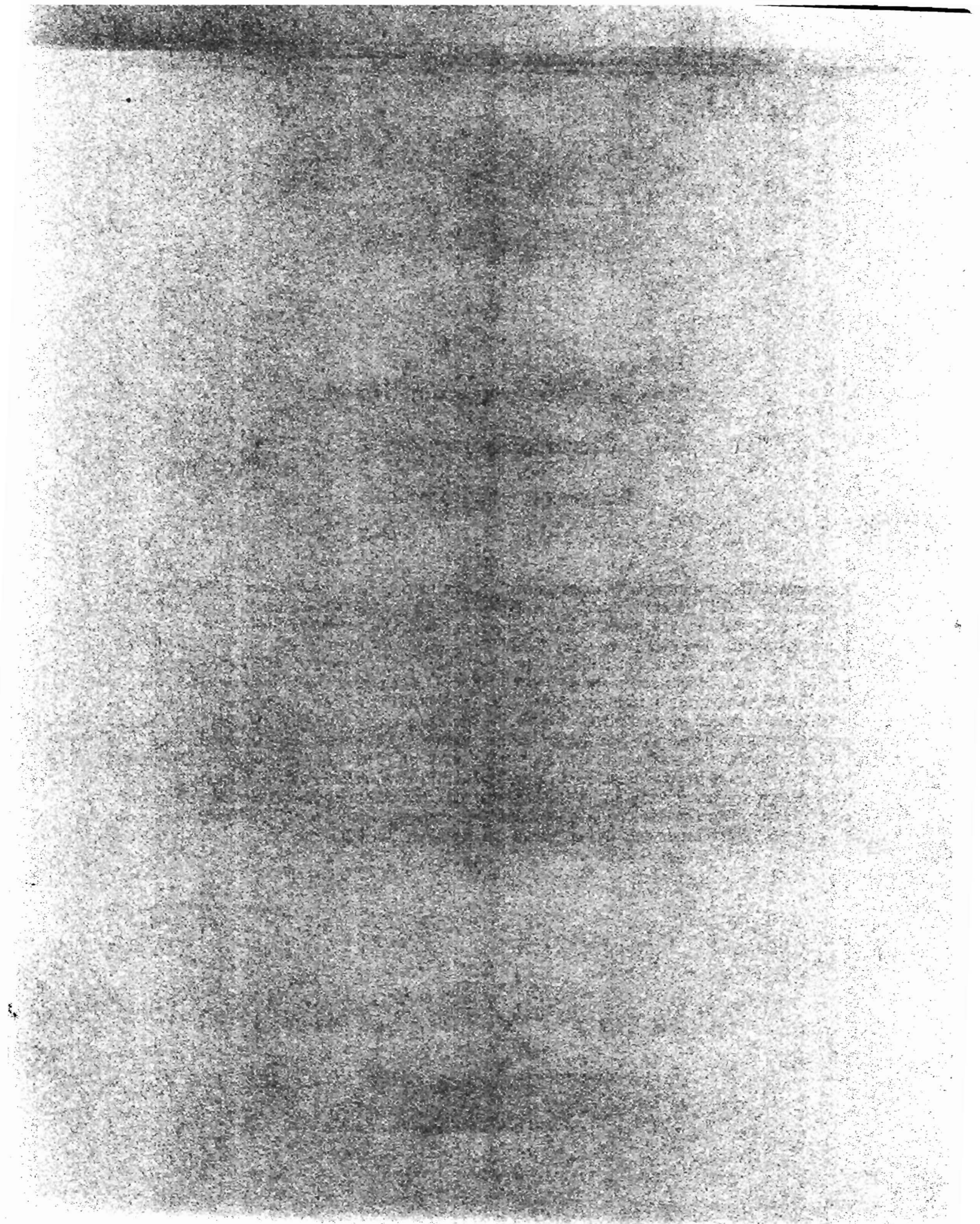


## **The G-2 Data Acquisition System**

**Charles Timmermans**

**University of Minnesota**

Experiment E821 (muon G-2 measurement) starts taking data in January 1996. In this experiment we will detect muon decays in 24 calorimeter stations. The total data rate could be as high as 5 MB/sec. The front end electronics is housed in 6 VME crates, each serving 4 detector stations. The data flows from these front end crates to an event builder VME crate across MXI links. The event builder crate contains a 68040 based single board computer running VxWorks which reads the data from the front end electronics and writes it out to tape. The DAQ is controlled from a UNIX host (HP 9000/715) through ethernet. Our data acquisition is staged in such a way that for the high rate runs there will be front end single board computers. These will be used for buffering and data compression and thus reducing the total throughput. Our data acquisition software is based on UNIDAQ. UNIDAQ was designed by the University of Michigan, KEK, and SSC-lab to be a scalable data acquisition system. This design makes it easy to add processes to UNIDAQ and make it work for our configuration. Our first indications show that the overhead introduced by UNIDAQ does not slow down the maximum data throughput of our configuration significantly.



# The G-2 Data Acquisition System

C. Timmermans, P. Cushman, S. Lopatin  
University of Minnesota  
148 Tate Lab. of physics  
116 Church St. S.E.  
Minneapolis, MN, 55455

November 4, 1994

## Abstract

Experiment E821 (muon G-2 measurement) starts taking data in January 1996. The total data rate could be as high as 5 MB/sec. The front end electronics is housed in 6 VME crates, each serving 4 detector stations. The data flows from these front end crates to an event builder VME crate across MXI links. The event builder crate contains a 68040 based single board computer running VxWorks which reads the data from the front end electronics and writes it out to tape. The DAQ is controlled from a UNIX host through ethernet. Our data acquisition software is based on UNIDAQ. The first indications show that the overhead introduced by UNIDAQ does not slow down the maximum data throughput of our configuration significantly.

## 1 Introduction

The G-2 experiment will measure the muon anomalous magnetic moment with high precision (.35 ppm), a factor 20 better than the previous measurement [1, 2]. This means we can be sensitive to weak interaction contributions to  $a_\mu$  (to 20 %), allowing a sensitive test of the renormalizability of the electro-weak theory. Other tests include CPT tests ( $\tau_{\mu^+}$  vs  $\tau_{\mu^-}$  and  $a_{\mu^+}$  vs  $a_{\mu^-}$ ), an improved muon lifetime measurement, and measuring a new limit on the electric dipole moment of the muon. Muons with a momentum of 3.094 GeV/c will be stored in the G-2 storage ring. Electric fields in the rest frame of the muon due to focussing quadropoles do not influence the spin precession frequency to first order at this momentum. We will count the number of muon decay electrons in 24 calorimeter stations. The number of electrons detected at each station depends on the direction of the muon spin. By monitoring the electron count over time, the muon spin precession frequency can be measured.

# The G-2 Data Acquisition System

C. Timmermans, P. Cushman, S. Lopatin  
University of Minnesota  
148 Tate Lab. of physics  
116 Church St. S.E.  
Minneapolis, MN, 55455

November 4, 1994

## Abstract

Experiment E821 (muon G-2 measurement) starts taking data in January 1996. The total data rate could be as high as 5 MB/sec. The front end electronics is housed in 6 VME crates, each serving 4 detector stations. The data flows from these front end crates to an event builder VME crate across MXI links. The event builder crate contains a 68040 based single board computer running VxWorks which reads the data from the front end electronics and writes it out to tape. The DAQ is controlled from a UNIX host through ethernet. Our data acquisition software is based on UNIDAQ. The first indications show that the overhead introduced by UNIDAQ does not slow down the maximum data throughput of our configuration significantly.

## 1 Introduction

The G-2 experiment will measure the muon anomalous magnetic moment with high precision (.35 ppm), a factor 20 better than the previous measurement [1, 2]. This means we can be sensitive to weak interaction contributions to  $a_\mu$  (to 20 %), allowing a sensitive test of the renormalizability of the electro-weak theory. Other tests include CPT tests ( $\tau_{\mu^+}$  vs  $\tau_{\mu^-}$  and  $a_{\mu^+}$  vs  $a_{\mu^-}$ ), an improved muon lifetime measurement, and measuring a new limit on the electric dipole moment of the muon. Muons with a momentum of 3.094 GeV/c will be stored in the G-2 storage ring. Electric fields in the rest frame of the muon due to focussing quadrupoles do not influence the spin precession frequency to first order at this momentum. We will count the number of muon decay electrons in 24 calorimeter stations. The number of electrons detected at each station depends on the direction of the muon spin. By monitoring the electron count over time, the muon spin precession frequency can be measured.

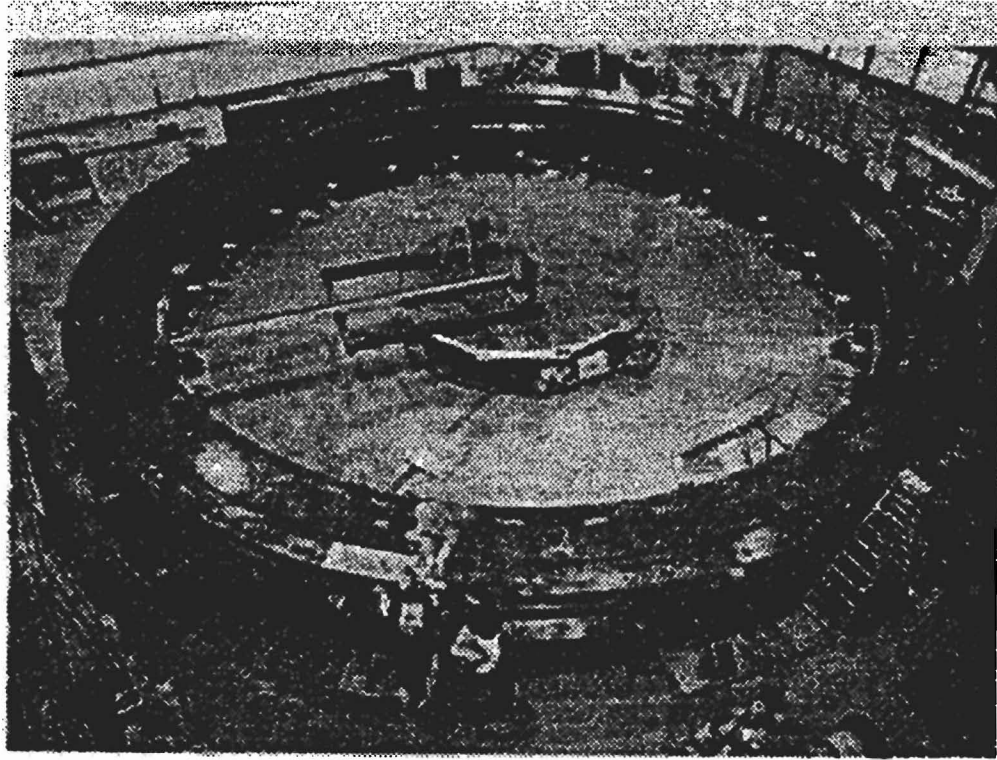


Figure 1: Picture of the G-2 storage ring.

## 2 Event Rates

The event rate depends on the number of muons stored in the storage ring. We expect 1630 muons per fill in 1996 ( $\pi$  injection), and 17,360 per fill starting in 1997 ( $\mu$  injection). The rates calculated below are assuming 17,360 muons per fill. The measured event rate per detector is:

$$\text{Rate} = N_{\mu} \times \exp\left[-\frac{\tau_w}{\tau_{life}}\right] \times \left(\frac{1}{24\text{Detectors}}\right) \times \left(\frac{1}{\tau_{ife}}\right) \times \epsilon_d$$

Under the assumptions that 54 % of the electrons enter the calorimeter stations and that the dead time at the beginning of a fill is 5  $\mu\text{s}$  we can calculate the event rate per station (the muon lifetime at this energy is 64.4  $\mu\text{s}$ ). The initial event rate per station is 5.6 Mhz. This rate sets the requirements for the front end electronics. During an AGS cycle we will get 12 fills 50 msec apart. The average rate in this 50 msec per station is 7.2 kHz. The AGS cycles are 2 seconds apart therefore, the average rate per station is 2.2 kHz.

### 3 Data Acquisition Setup

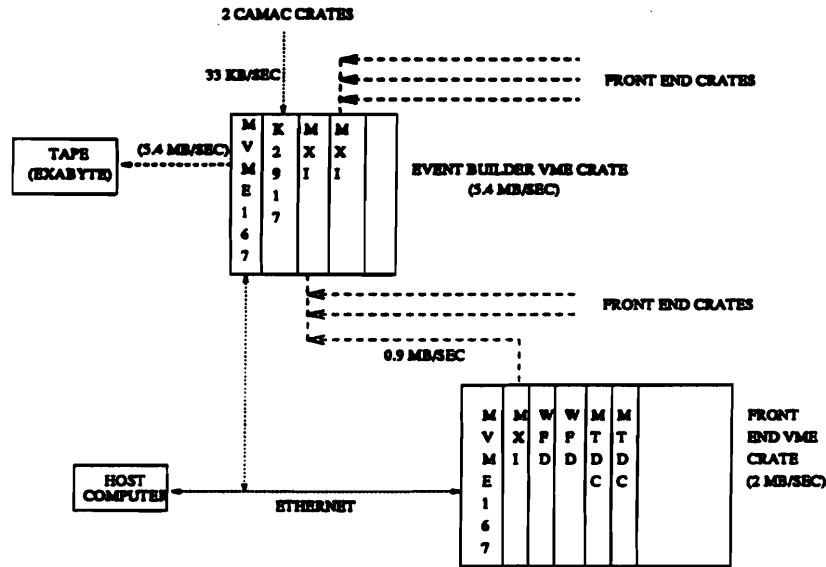


Figure 2: The G-2 Data Acquisition Setup.

The energy and time of the decay electrons are measured using a custom Waveform Digitizer and a Multi hit TDC [3]. Both VME modules have on board memory in which all data from a fill can be stored. The electronics of 4 detector stations (2 MTDCs and 2 WFDs) are combined into one VME crate. We will have a total of six front end crates. Since a detected electron generates 72 bytes of data [2] the initial rate is 403 MB/s/station (= 1.6 GB/s/VME crate). These rates are buffered by the front end modules. Between fills the data need to be transferred to a local CPU since we do not have enough memory in the front end modules to store all data from an AGS cycle and the bandwidth of our VME-VME interconnects is not high enough to send the data to the event builder. In the local CPU lossless compaction can be performed if so desired. This imposes a bandwidth requirement of 2.1 MB/s/crate, easily handled by the VME backplane. Between AGS cycles the data will be sent up to the event builder VME crate. Processes running in the event builder CPU receive all data, store it onto tape and send a sample to the host computer. Sending all data up to the event builder CPU between cycles (1.4 sec) requires a bandwidth of 5.4 MB/sec. The MXI bus [4] is used for VME-VME data links. The MXI bus forms a double star formation. The throughput of a single link has been measured at 3 MB/sec [5]. A small amount of (calibration) data (33 kB/s) will be read from CAMAC in the event builder, using a kinetics interface (K2917).

## 4 Online Software requirements

The G-2 online software tasks are divided between UNIX and VXWORKS operating systems. The host computer (UNIX) provides the user interface, an event display, online histogramming and analysis, a run control, and the interface with slow control. The MVME167 (VXWORKS) provides the online software to read data from different sources, build events, write all data to tape, and transfer a fraction of the data to the UNIX host for monitoring. This diversity in tasks requires a modular data acquisition system which is supported on both UNIX and VXWORKS. The high data rate expected require that the system impose a low software overhead. UNIDAQ [6] meets these demands.

## 5 UNIDAQ

UNIDAQ started as a portable Data Acquisition System for the SDC collaboration, although it has been maintained after the SSC termination. The participating institutes are U. Michigan, K.E.K., T.I.T. and U. Minnesota. It is designed as a modular, extendable and scalable system. No additional packages are needed to run UNIDAQ. The inter process communication uses shared memory and message queues within a single machine. This is expanded to a multiple machine environment using RPC's. UNIDAQ can easily be interfaced with other packages. Interfaces to some packages (murmur, Tcl/Tk) are available. UNIDAQ provides support for VME and CAMAC (through K2917) commands.

The UNIDAQ functionality can be divided into:

- **Data Handling:** Data is read in by the Collector process, and passed on to the NOVA buffer manager. NOVA passes the buffers from collector to recorder, analyzer, etc. NOVA has the capability of transporting the data between different machines.
- **User Control:** Both run control and operator (Tcl/Tk [7] or only X windows) are configurable from ASCII scripts. The output of the logbook process is stored in ASCII files, but can also be displayed (using MURMUR).
- **DAQ Control Processes:** Several control processes and tools exist to guarantee UNIDAQ to be up and running ( e.g. the XPC process which checks if all listed UNIDAQ processes are present, the status tool which gives the current status of all the unidaq processes, the repair tool which cleans up shared memories and message queues used by UNIDAQ, etc.). Auto starting of processes is possible, saving important information (e.g. data source, output file) from the previous invocation.



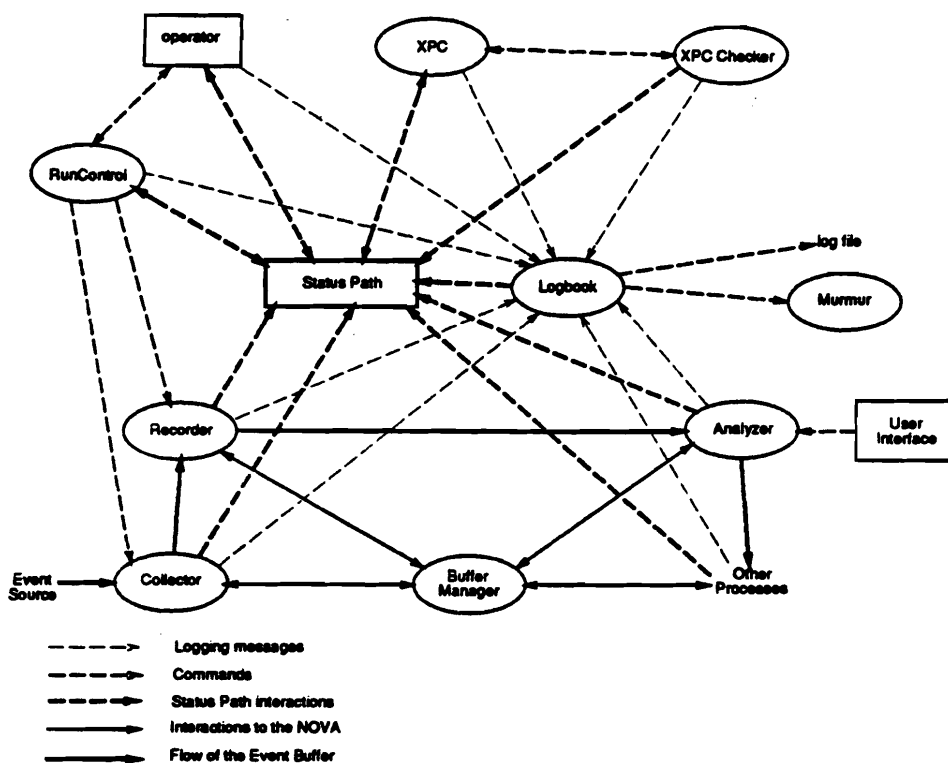


Figure 3: The UNIDAQ processes.

By choosing only the modules needed, the software overhead is minimized. This scalability makes UNIDAQ ideal for small (test beam) and medium size DAQ projects. All UNIDAQ processes are event driven. Examples of events are messages between Processes, interrupts or signals to start data taking (collector), getting a NOVA buffer (recorder, analyzer, ...), clicking a mouse button (operator), etc.

## 6 UNIDAQ for G-2

Some additional UNIDAQ processes are needed in order to use it as the final G-2 online system. We are gradually building our system, using the most recent version in test beam experiments. We plan to run a collector process in the MVME167 located in the front-end VME where data compression may take place. Afterwards data will be sent across MXI to the event builder CPU. Decoupling the event builder from the readout processes is not standard in

UNIDAQ. The NOVA buffer manager is not capable of sending a fixed fraction of the data across the network. It can be set up to either send all data, or to only send data if that does not slow down data taking. Sending a fixed sample of the data to UNIX requires the addition of a filter process. We also modified the recorder process to allow it to write directly to tape from VXWORKS. All required UNIX processes are present in UNIDAQ, but some user-defined modifications are easily made in the user interface by reprogramming a Tcl/Tk script. The online analyzer will be modified to display the important parameters of our experiment. This requires programming in fortran. Standard CERN histogramming routines can be used since the analyzer interfaces to PAW. The interface to our slow control system (FactoryLink) is currently being designed which will respond to UNIDAQ messages.

## 7 Throughput Tests

We have done tests on the throughput of a single MXI link, using VME memory and the MVME167. We read out the memory with both a dedicated program and with UNIDAQ. The results (table 1) indicate that UNIDAQ does not slow down the data taking significantly [5], provided the event length exceeds 1 kB.

Transfer across			DMA	
	Read	Write	Read	Write
VME backplane	4.2	6.5	11.2	18.8
(with UNIDAQ)	4.2		10.8	
MXI	2.3	2.6	3.1	3.4
(with UNIDAQ)	2.2		3.0	

Table 1: Data Transfer Rates (MB/s) across VME backplane and across MXI.

## References

- [1] J. Baily et al., Nucl. Phys. B150, 1 (1979)
- [2] Design Report BNL AGS E821, april 1994.
- [3] E. Hazen, G. Varner, Guide to the Multi-hit TDC, g-2 note 218, 1994.  
R. Carey et al., A 400 MPS Waveform Digitizer for E-821, g-2 note 208,1994.
- [4] VME-MXI user manual, october 1992.

- [5] C. Timmermans, P. Cushman, G-2 DAQ throughput tests, g-2 note 213, 1994.
- [6] UNIDAQ documentation set, SDC-93-573, september 1993.
- [7] J. K. Ousterhout, Tcl and the TK Toolkit, Addison Wesley, 1994.

# The G-2 Experiment

## Motivation

This experiment will measure the muon anomalous magnetic moment with high precision (.35 ppm), a factor 20 better than the previous measurement. Reasons to do the experiment are:

- high precision test of the theory
- measurement of weak interaction contributions to  $a_\mu$  (to 20 %)
- CPT tests ( $\tau_{\mu^+}$  vs  $\tau_{\mu^-}$  and  $a_{\mu^+}$  vs  $a_{\mu^-}$ )

## The setup

Muons with a momentum of 3.094 GeV/c will be stored in the G-2 storage ring. Electric fields do not influence the spin precession frequency at this momentum.

We will measure muon decay electrons in 24 calorimeter stations. The number of electrons detected in

the stations depend on the direction of the muon spin. By monitoring the electron count over time one gets the muon spin precession frequency, which is related to G-2.

## Event Rates

Event Rates depend on the efficiency of storing muons in the storage ring. We expect 1630 muons per fill in 1996, and 17,360 per fill starting in 1997. The rates calculated below are assuming 17,360 muons per fill. The measured event rate is:

$$\text{Rate} = N_{\mu} \times \exp\left[-\frac{\tau_w}{\tau_{life}}\right] \times \left(\frac{1}{24\text{Detectors}}\right) \times \left(\frac{1}{\tau_{life}}\right) \times \epsilon_d$$

Under the assumptions that 54 % of the electrons enter the calorimeter stations and a dead time at the beginning of a fill of 5  $\mu\text{s}$  we can calculate the event rate per station (the muon lifetime at this energy is 64.4  $\mu\text{s}$ ). The initial event rate per station is 5.6 Mhz. This rate sets the requirements for the front end electronics.

the stations depend on the direction of the muon spin. By monitoring the electron count over time one gets the muon spin precession frequency, which is related to G-2.

## Event Rates

Event Rates depend on the efficiency of storing muons in the storage ring. We expect 1630 muons per fill in 1996, and 17,360 per fill starting in 1997. The rates calculated below are assuming 17,360 muons per fill. The measured event rate is:

$$\text{Rate} = N_{\mu} \times \exp\left[-\frac{\tau_w}{\tau_{life}}\right] \times \left(\frac{1}{24\text{Detectors}}\right) \times \left(\frac{1}{\tau_{life}}\right) \times \epsilon_d$$

Under the assumptions that 54 % of the electrons enter the calorimeter stations and a dead time at the beginning of a fill of 5  $\mu\text{s}$  we can calculate the event rate per station (the muon lifetime at this energy is 64.4  $\mu\text{s}$ ). The initial event rate per station is 5.6 Mhz. This rate sets the requirements for the front end electronics.

During an AGS cycle we will get 12 fills 50 msec apart. The average rate in this 50 msec per station is 7.2 kHz.

The AGS cycles are 2 seconds apart, reducing the average rate per station to 2.2 kHz.

---

## Readout Electronics

The most important measurements of the decay electron are:

- Energy (waveform of the calorimeter signal)
- Time (scintillator signal)

The electronics used are a Waveform Digitizer and a Multi hit TDC. Both VME modules, designed at Boston University, have on board memory in which all data from a fill can be stored.

The electronics of 4 detector stations (2 MTDCs and 4 WFDs) are combined into one VME crate. We will have a total of six front end crates.

## Data Acquisition Setup

A detected electron generates 72 bytes of data. Therefore the initial rate is 403 MB/s/station (= 1.6 GB/s/VME crate). These rates are handled by the front end modules.

Between fills the data can be transferred to a local CPU. This imposes a bandwidth requirement of 2.1 MB/s/crate, easily handled by the VME backplane.

---



Between AGS cycles the data will be sent up to the event builder VME crate. Processes running in the event builder CPU receive all the data, store it on tape and send a sample to the host computer. Sending all data up to the event builder CPU between cycles (1.4 sec) requires a bandwidth of 5.4 MB/sec.

The MXI bus is used for VME-VME data links. The MXI buses form a double star formation. The throughput of a single link is tested to be 3 MB/sec. A small amount of (calibration) data (33 kB/s) will be read from CAMAC in the event builder, using a kinetics interface.

# Online Software Requirements

The G-2 online software tasks are divided between UNIX and VXWORKS systems. On the UNIX host computer it has the following tasks:

- online user interface
- event display
- online analysis
- run control
- interfacing with slow control

On the MVME167 running VXWORKS the online software needs to:

- Read data from different sources
  - Build events
  - Write all data to tape
  - Send data sample to UNIX
-

This diversity in tasks requires a modular system, supported on both UNIX and VXWORKS. The total data rate expected requires a low software overhead. UNIDAQ meets these demands.

## UNIDAQ

UNIDAQ started as a portable Data Acquisition System for SDC, though has been maintained after the SSC termination. The participating institutes are U. Michigan, K.E.K., T.I.T. and U. Minnesota. It is designed as a modular, extendable and scalable system. No additional packages are needed to run UNIDAQ. The inter process communication uses shared memory and message queues within a single machine. This is expanded to a multiple machine environment using RPC's.

UNIDAQ provides support for CAMAC (through K2917) and VME commands.

UNIDAQ can easily be interfaced with other packages. Interfaces to some packages (murmur, Tcl/Tk) are available.

The UNIDAQ functionality can be divided into:

- Data Handling

Data is read in by the Collector process, and passed to the NOVA buffer manager. NOVA passes the buffers from collector to recorder, analyzer, etc. NOVA has the capability of transporting the data between different machines.

- User Control

Both run control and operator (Tcl/Tk or only X windows) are configurable from ASCII scripts. The output of the logbook process is stored in ASCII files, but can also be displayed (using MURMUR).

- DAQ Control Processes

Several control processes and tools exist to guarantee UNIDAQ to be up and running. Auto starting of processes is possible, saving important information (e.g. data source, output file) from the previous invocation.

By choosing only the modules needed, the software overhead is minimized. The scalability makes

---

UNIDAQ ideal for small (test beam) and medium size DAQ projects.

All UNIDAQ processes are event driven. Examples of events are:

- Messages between Processes
- Interrupts or signals to start data taking (collector)
- Getting a NOVA buffer (recorder, analyzer, ...)
- Clicking a mouse button (operator)
- etc.

## UNIDAQ for G-2

Some additional UNIDAQ processes are needed in order to use it as the final G-2 online system. We are gradually building our system, using the most recent version in test beam experiments.

### Real Time Processes

We plan to run a collector process in the front-end VME. Here data compression may take place. Afterwards data will be sent across MXI to the event builder. The MXI links do not require a modification of the software, since they act as VME bus extenders. Decoupling the event builder from the readout processes is not standard in UNIDAQ.

The NOVA buffer manager is not capable of sending a fixed fraction of the data across the network. It can be set up to either send all data, or to only send data if that does not slow down data taking. Sending a fixed sample of the data to UNIX requires the addition of a filter process, removing data from the NOVA buffers.

---

## UNIX Processes

On UNIX all online processes needed are present in UNIDAQ. Some modifications are needed in the user interface, which requires reprogramming a Tcl/Tk script.

The online analyzer needs to be modified to display the important parameters of our experiment. This requires programming in fortran. Standard CERN histogramming routines can be used since the analyzer interfaces to PAW.

The interface to our slow control system (FactoryLink) is currently being designed. A simple interface is to have our slow control respond to UNIDAQ messages.

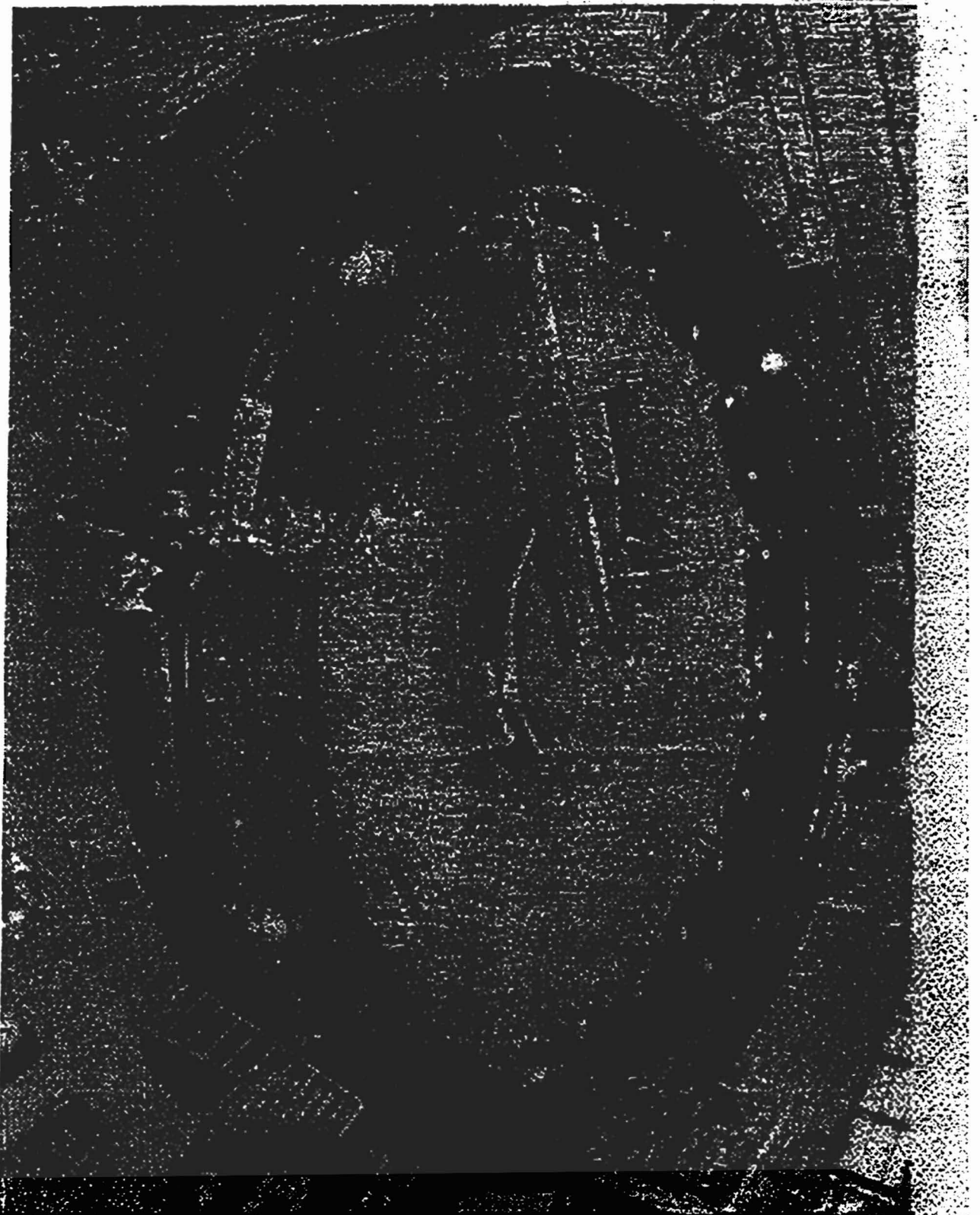
## Throughput Tests

We have done tests on the throughput of a single MXI link, using VME memory and a 167 single board computer. We read out the memory with a dedicated program and with UNIDAQ. The results indicate that UNIDAQ does not slow down the data taking significantly.

			DMA	
	Read	Write	Read	Write
VME backplane	4.2	6.5	11.2	18.8
UNIDAQ	4.2		10.8	
MXI	2.3	2.6	3.1	3.4
UNIDAQ	2.2		3.0	

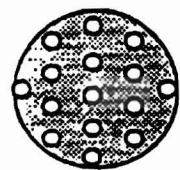
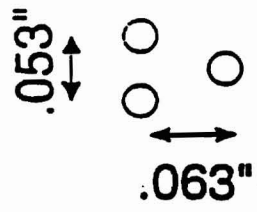
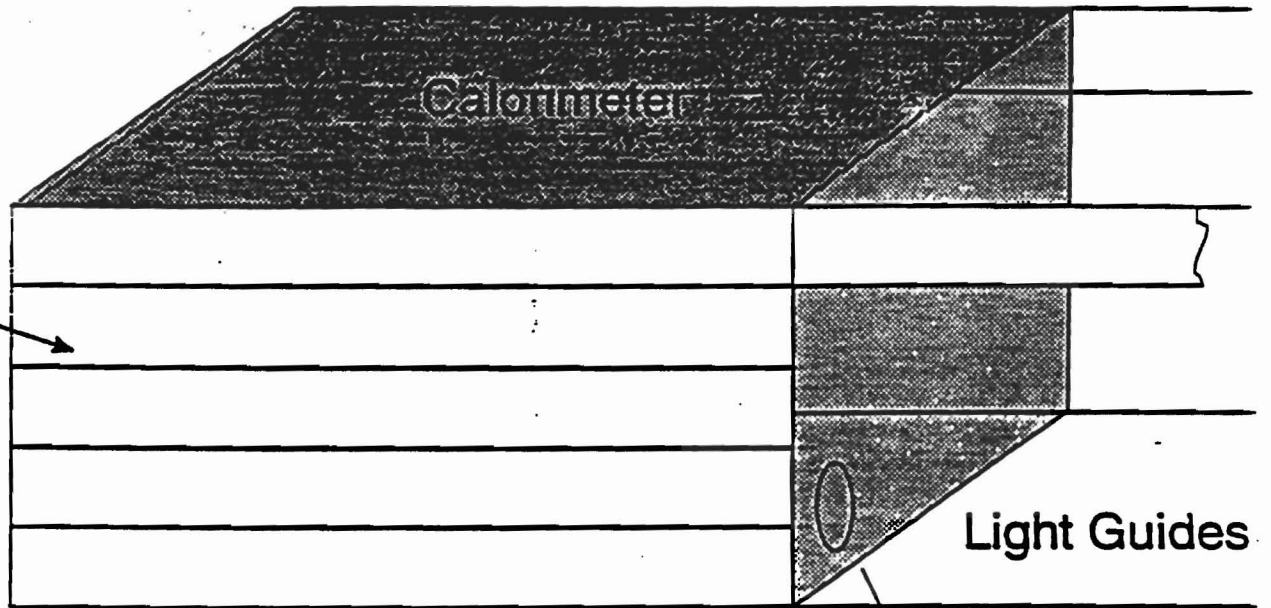
**Data Transfer Rates (MB/s) across VME backplane and across MXI.**



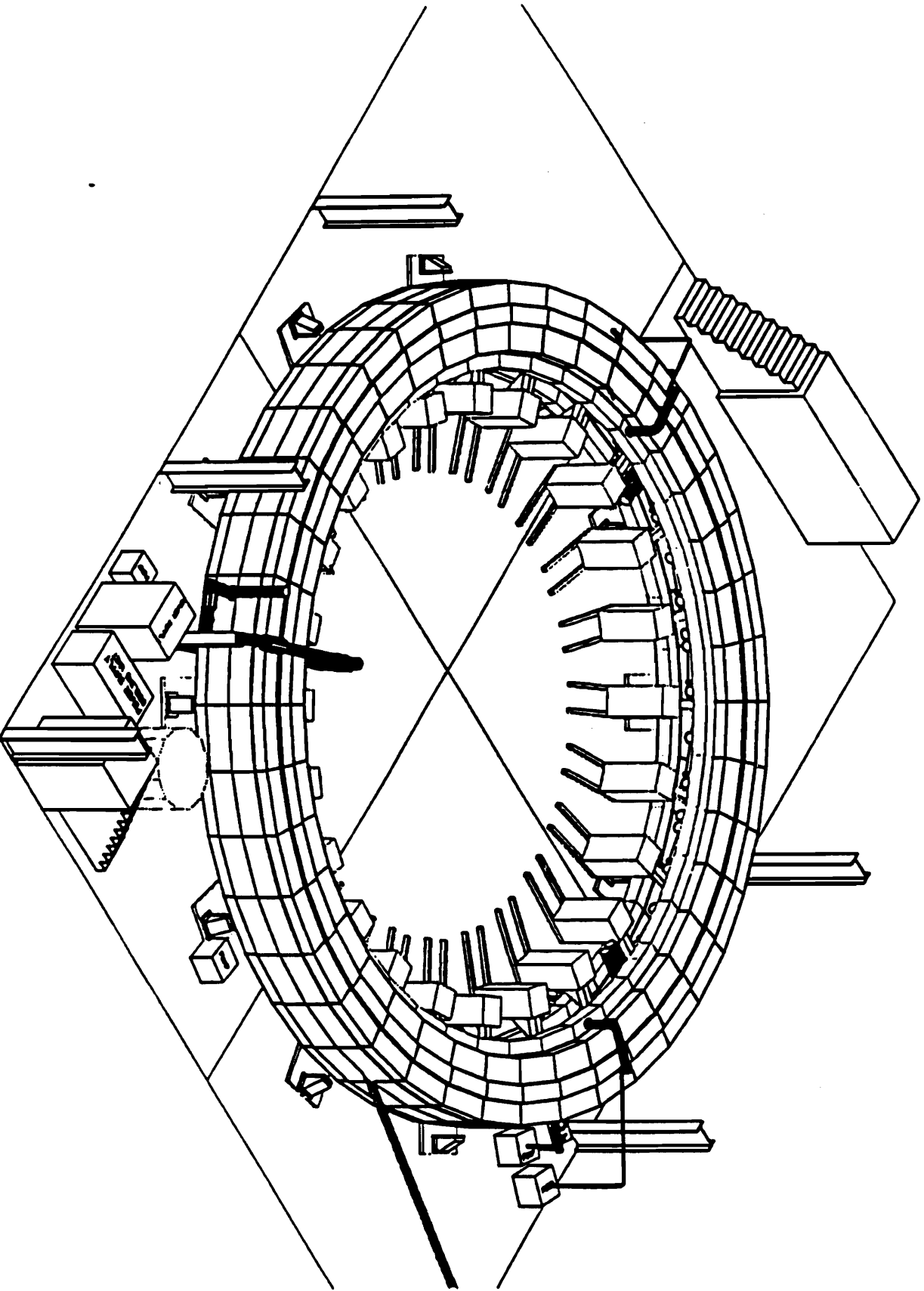


Picture of the G-2 storage ring.

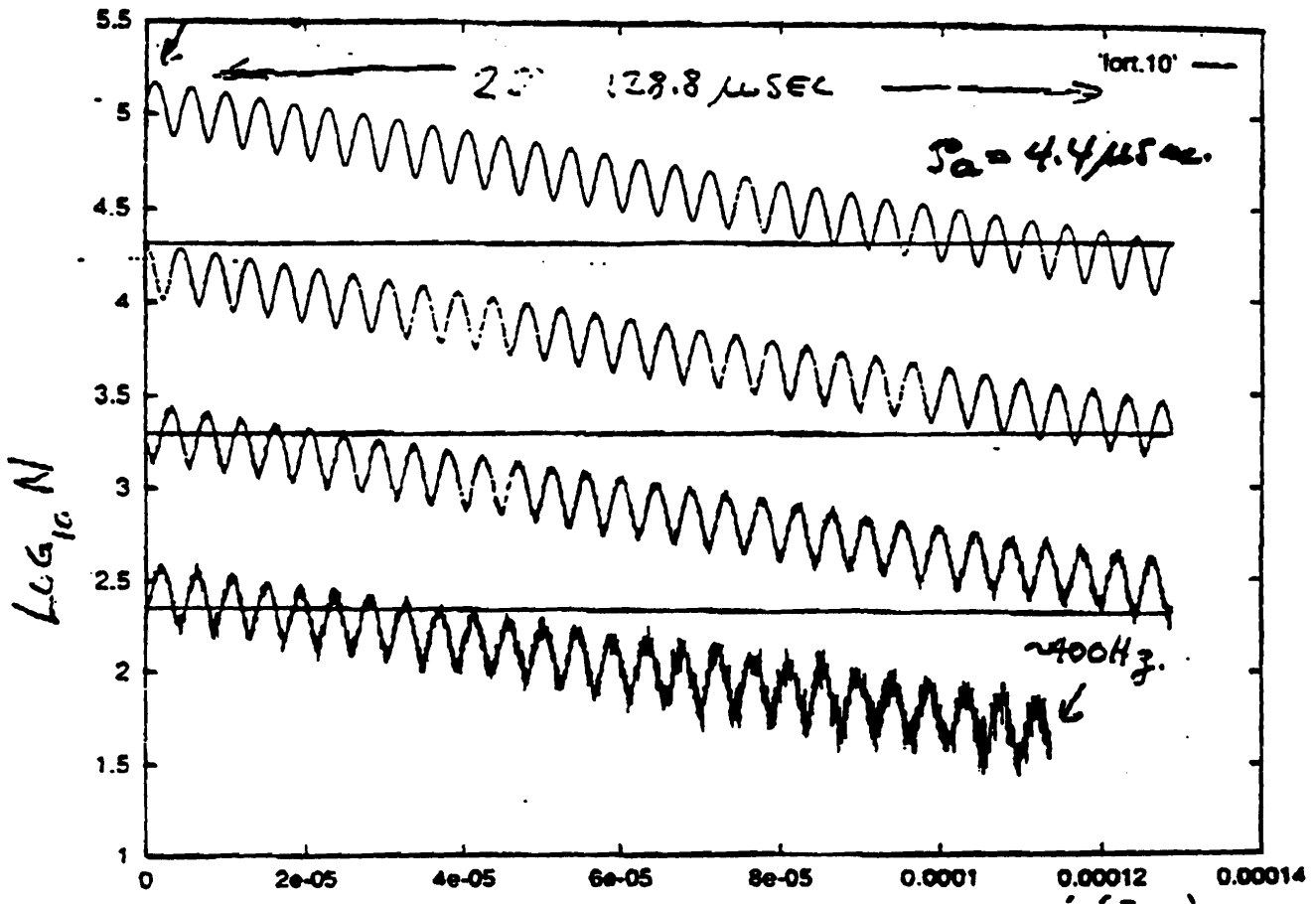
Front  
Scintillator  
Detectors



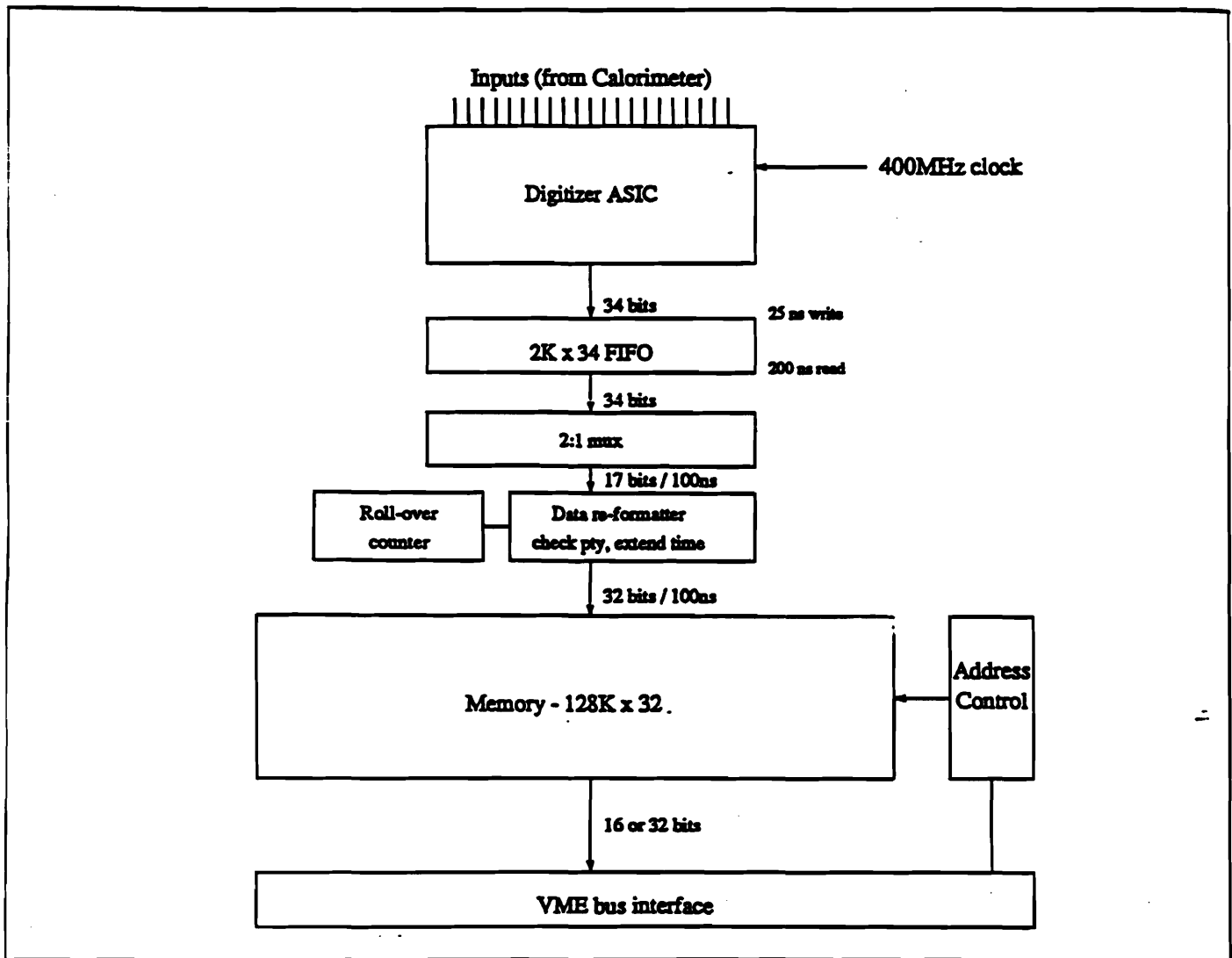
Calorimeter design.



Storage ring, showing 24 calorimeter stations.

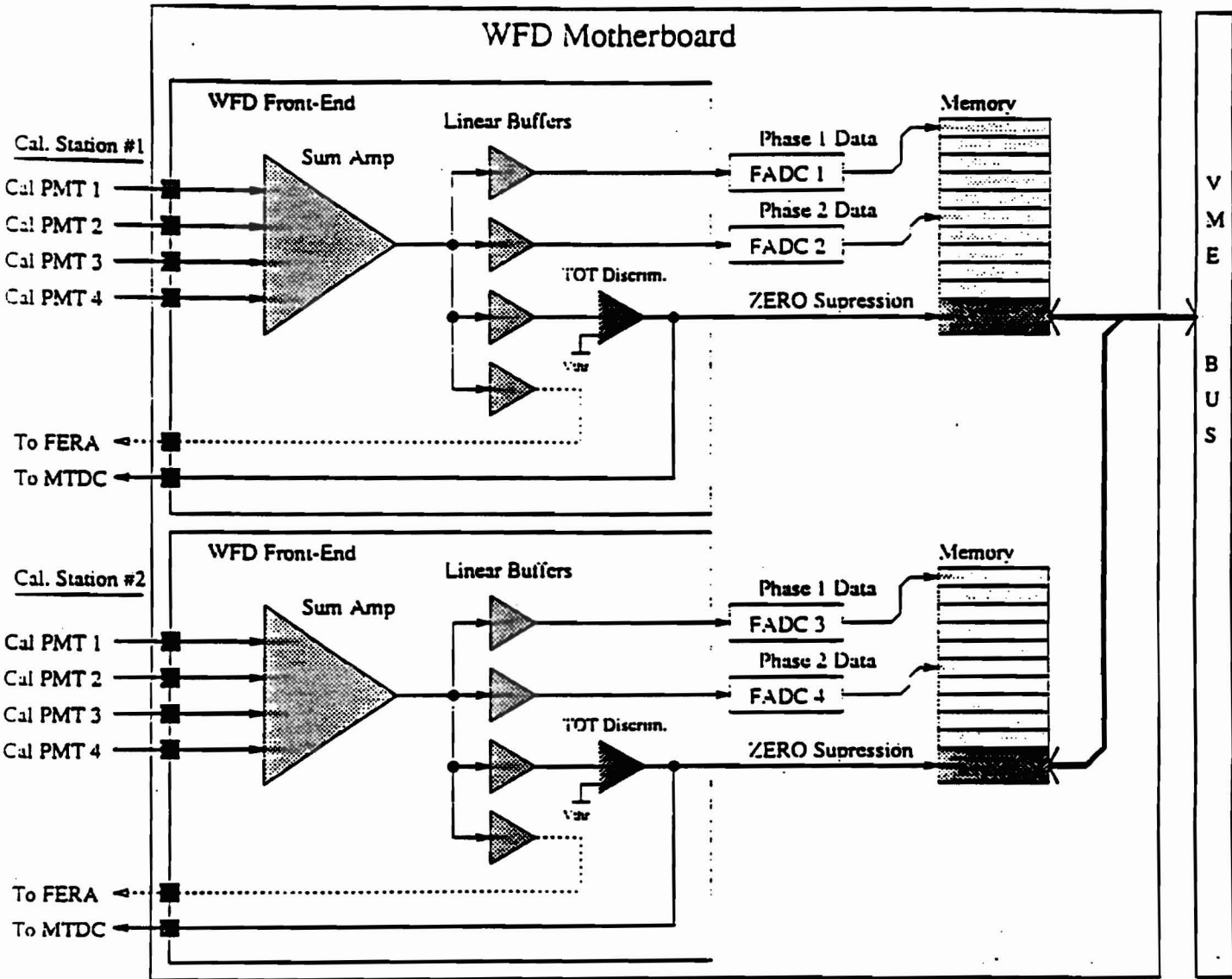


Expected event rate.

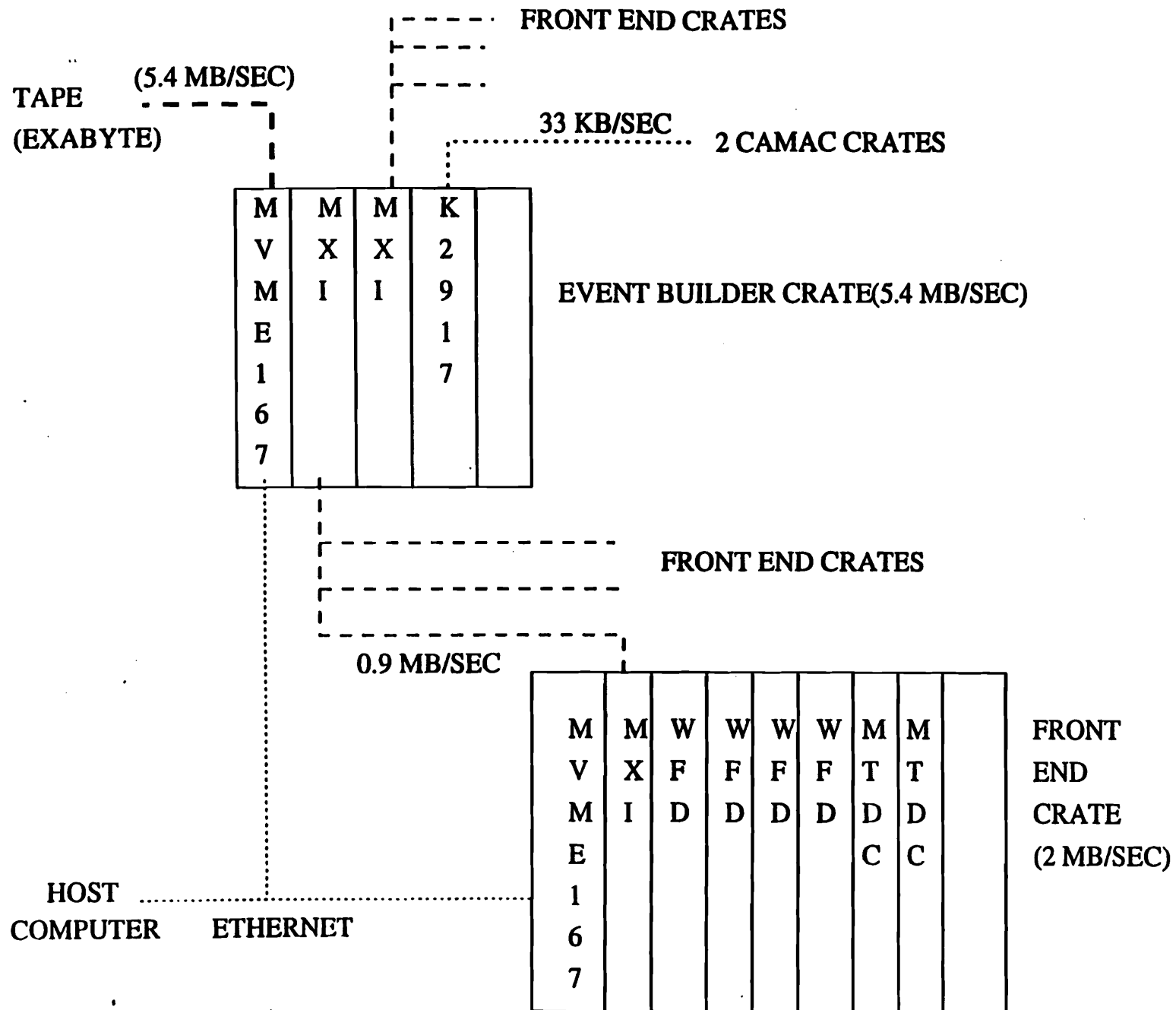


**MTDC block diagram.**

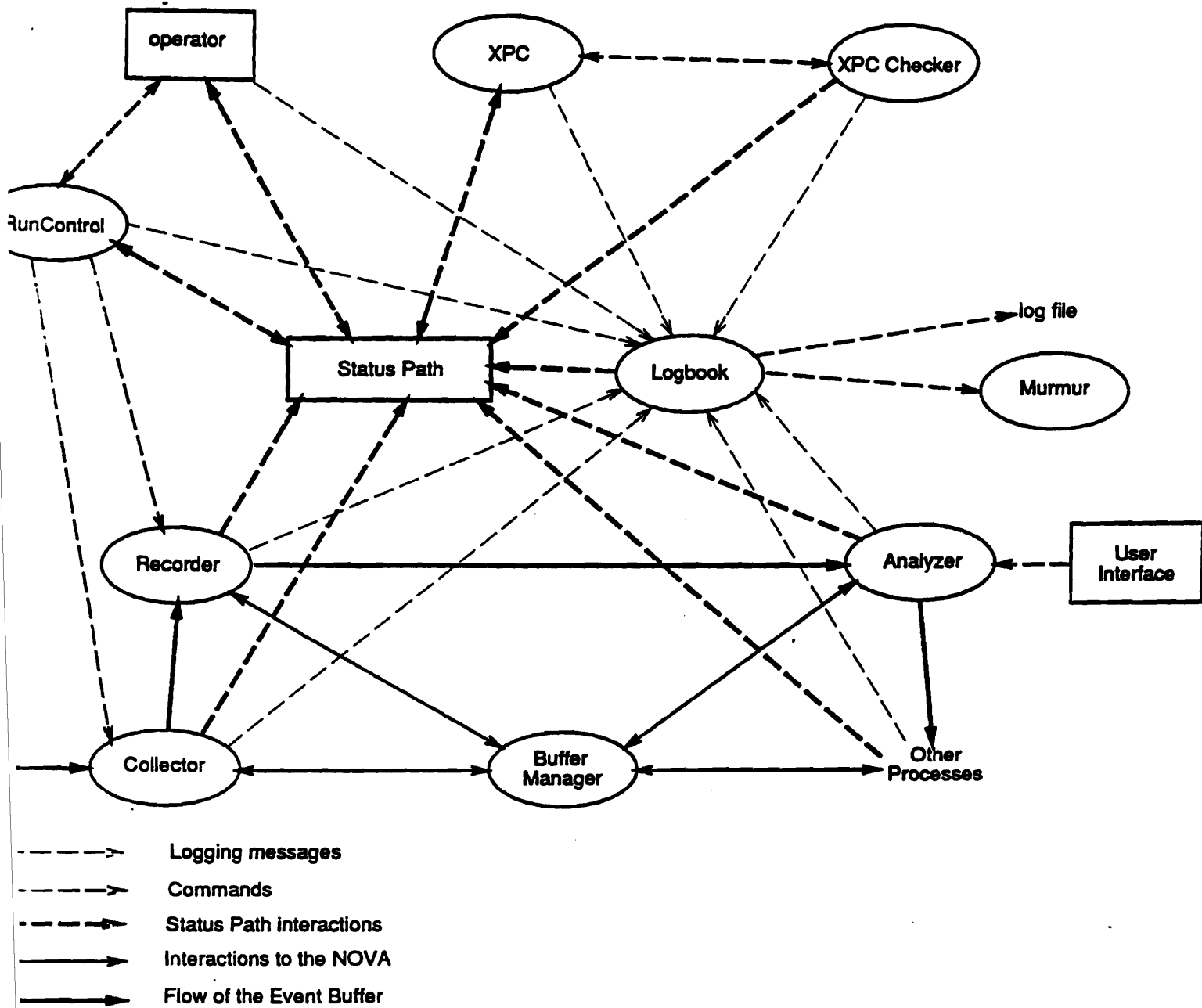
# WFD Summing and Fanout Details



Schematic structure of WFD operation.



**G-2 DAQ setup.**



**Standard UNIDAQ processes.**



# UNIDAQ PORTABLE DATA ACQUISITION SYSTEM

## General User Interface

User name: gm2      Host name: g2sun      Host type: SunOS      11:09

### Run and Storage parameters

Storage: dummy  
Runnr: 0      Max. Events: 0  
Event source: PSEUDO      Event Number: 699371

Status: Run

Start time: Sun Oct 23 10:58:11 1994

End Time:

### Run control commands

Begin Cold      <sup>44</sup>Begin Warm  
Continue      Suspend  
End      Initialize

### Processes in Use

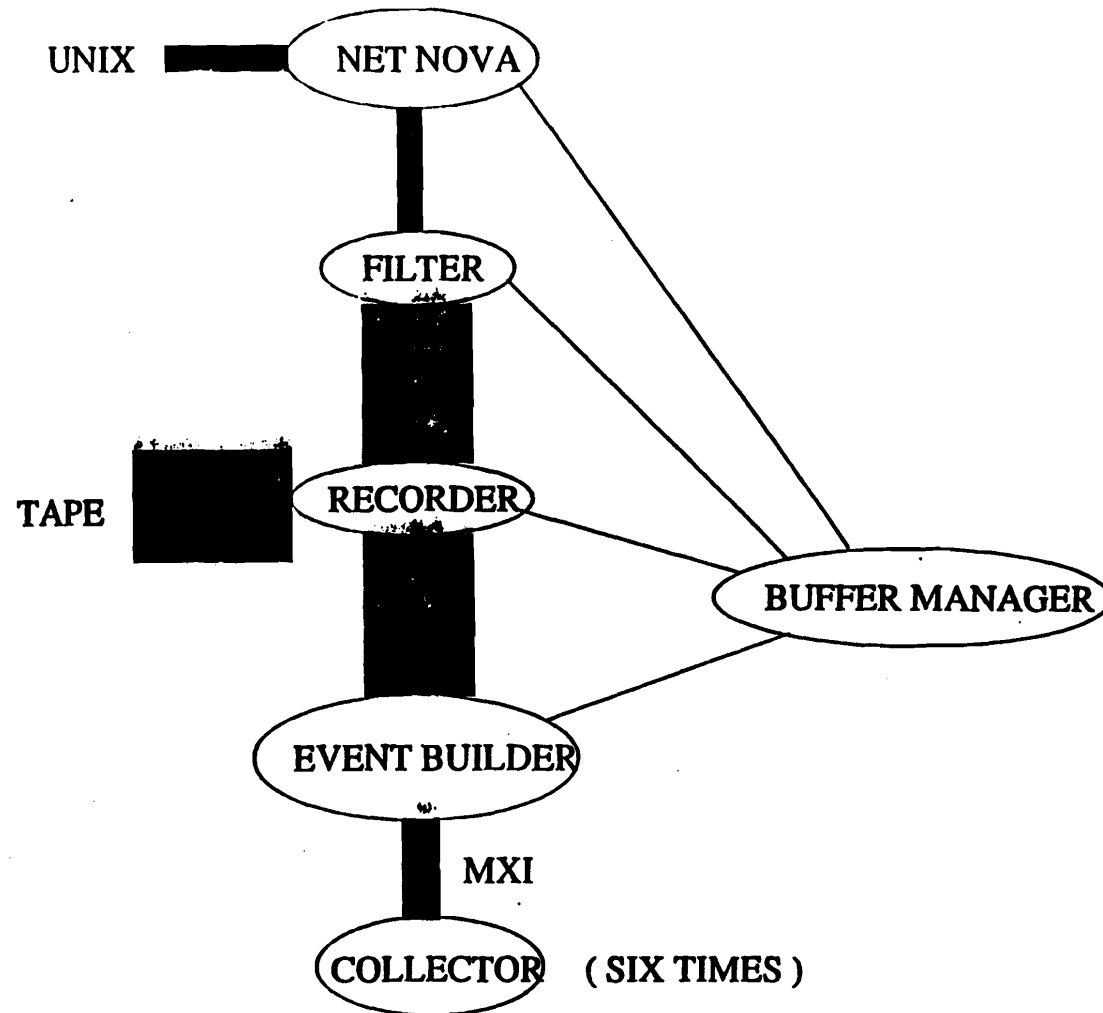
runcontrol: runco  
collector: collector  
recorder: recorder

PRIVILEGED

### Run control commands

command > >

UNIDAQ user interface.



**UNIDAQ processes in VXWORKS**



## **Fast Data Link & Modern RISC Processors for HEP Projects**

**Abdenour Lounis**

**Creative Electronic Systems**

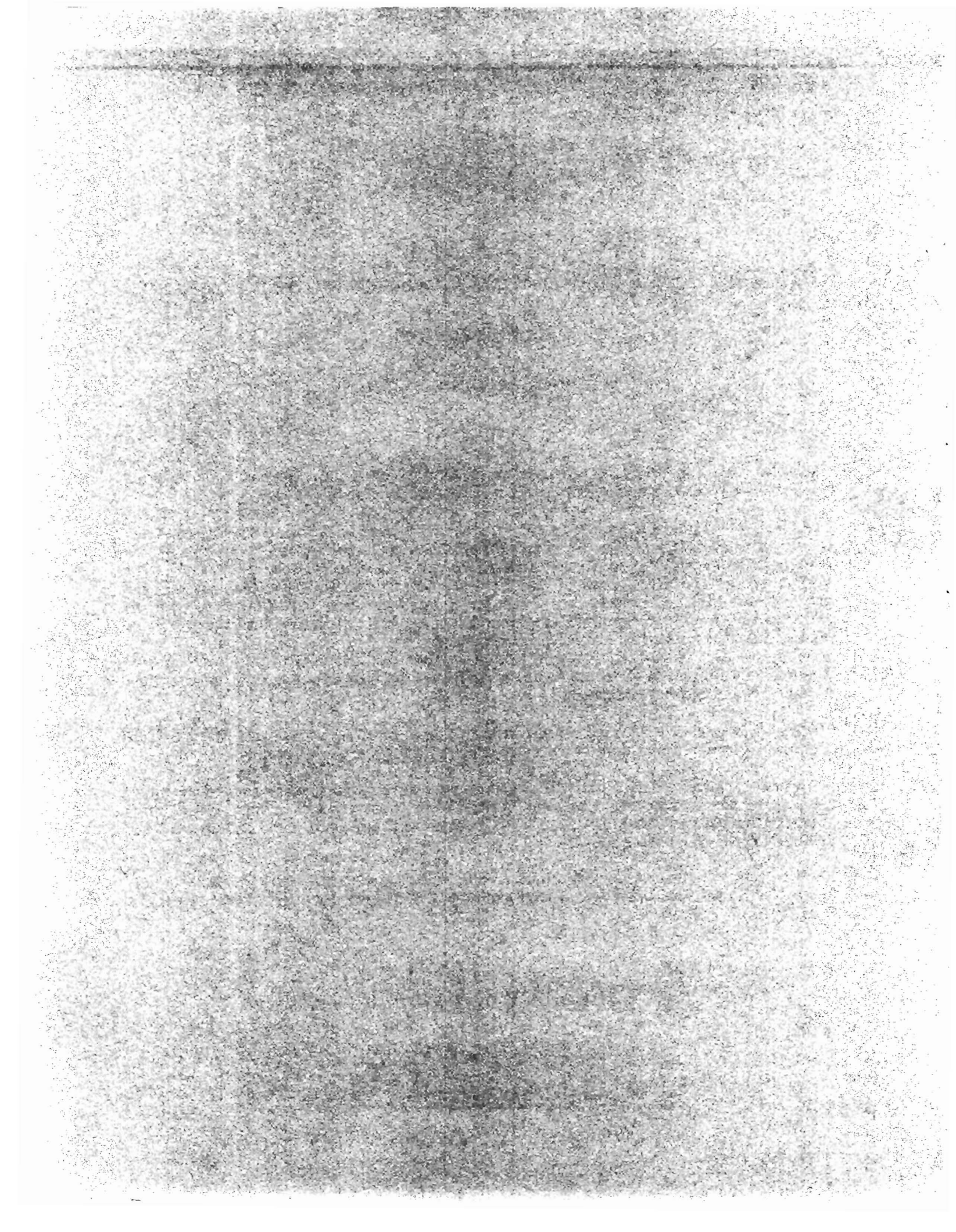
The Fast Data Link addresses the problem of data collection from digitizer units and data scattering to processor's memories at speed matching those of the processors.

The FDL8050 board could be seen as a set of VME hardware and software tools designed to transfer data "intelligently" between multiple sources and destinations. Such device is a multimaster/multidrop (up to 15 connections) cooper link synchronized at 50 MHz. The support medium is a cable composed of 25 twisted pairs extending over a maximum distance of 30 meters. The DL works in a client-server scheme. Upon request from a client or occurrence of an external event, the interface gathers data from digitizers, buffer memories or registers and stores it locally in an intermediate buffer.

Routing information are added to the data packet so that destination nodes can route the information to their final destination.

The Fast Data Link concept is already a significant step in actual implementation of high speed data links. Moreover, new developments are underway in two major directions:

- Design of high performance (>100 MIPS) general purpose real-time processor suited for online data acquisition scalable event builders. This VME board features a high speed VME Master/Slave interface (80 MByte/s), a R4400 or Power-PC 603 RISC CPU and allows for a PCI bus mezzanine interface connection.
- Implementation of gigabit type connections (ATM, Fiber channel, SCI) on a general purpose low cost VME platform using two IEEE standard PCI bus interfaces to Peripheral Mezzanine Card (PMC).



# MODERN RISC PROCESSORS FOR Events BUILDERS AND HIGH BANDWIDTH INTELLIGENT I/O INTERFACES

*by Abdenour Lounis, Michel Chorowicz*

Creative Electronic Systems  
70, Route du Pont-Butin - P.O. Box 107  
1213 Petit-Lancy 1 - Geneva - Switzerland  
Tel: +41.22.792.57.45 - Fax: +41.22.792.57.48  
Email: ces@lancy.ces.ch

## INTRODUCTION

We will describe the use of RISC processor architecture in Real-Time data acquisition applications. RISC processors can be used in Real-Time at two levels:

The first level is the most commonly used: it is the general purpose processor level (Single Board Computers in VME environment).

The second level is an application level spreading rapidly because of the inherent complexity of data communication functions: it is the embedded I/O processor.

We will describe two applications where RISC structure has been used at the benefit of the overall operations of the application.

- a high speed Real-Time Network FASTLINK
- a general purpose VME Real-Time Processor

## 1. THE FAST DATA LINK SYSTEM

### *1.1 Target Specifications*

In a nutshell, the target was to design a Real-Time network system which had to be:

- **Fast**                    50 to 100 Mbytes/s bus throughput  
                              Local 50 Mbytes/s VME transfers
- **Intelligent**        Crate Scan, List Processor, Read-Out Lists, twin 25 MHz R3000 architecture
- **Multidrop**         15 nodes copper, 225 nodes fibre, full duplex bus
- **Deterministic**    Response time guaranteed within micro seconds
- **User friendly**     Easy to program

### 1.3 Logical Building Blocks

A simplified logical block diagram is shown below:

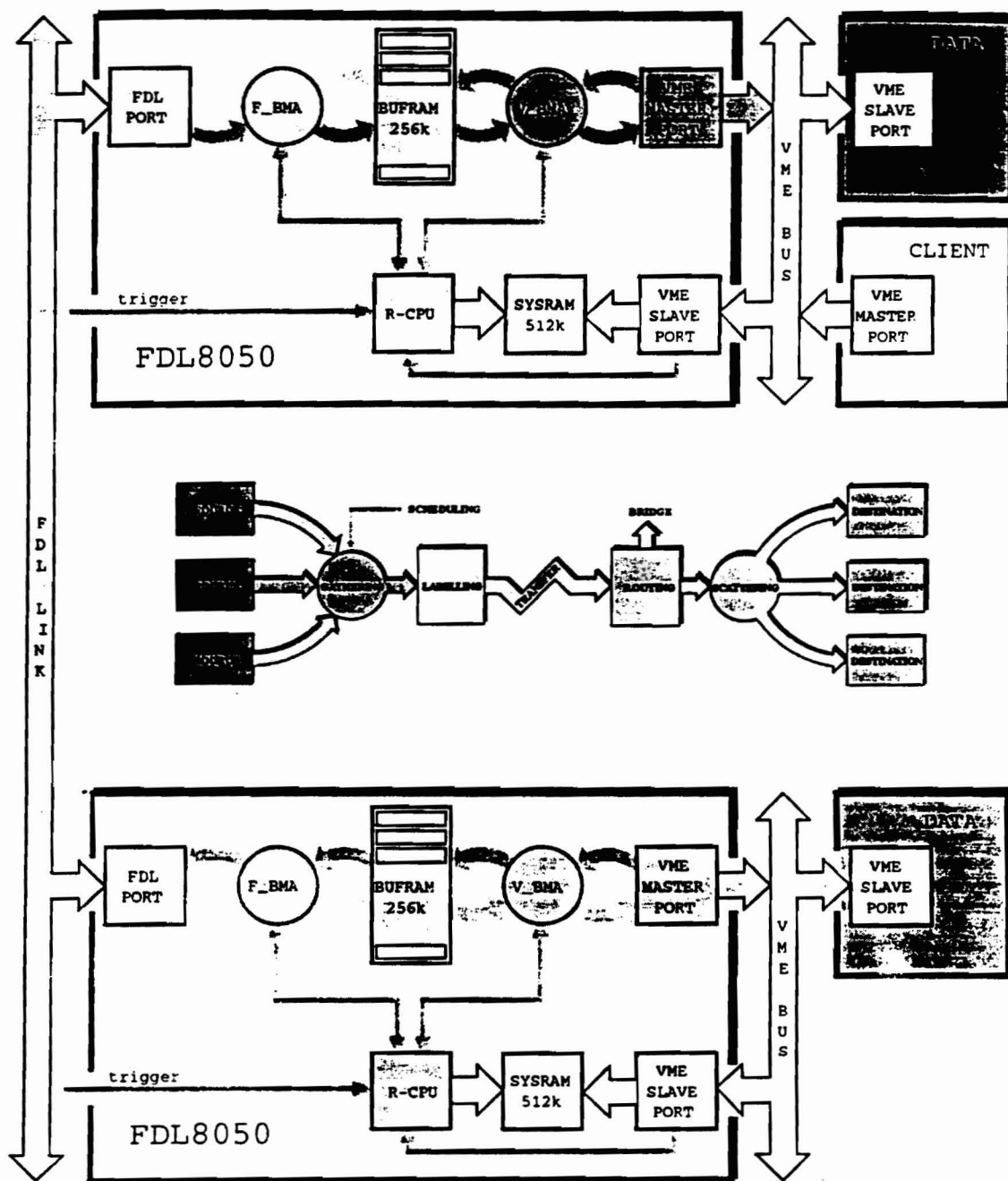


Fig. 2. Fast Link logical block diagram

The FDL firmware consists of two RISC processors; one handling the local data acquisition and the other one handling the transmission on the Real-Time network.

## *1.6 Data Transmission Section*

Control of the connection is realised with another RISC processor. Because of the speed of the transmission (50 to 100 Mbytes/s) and of the on-line supervision required for the different transmission modes only a RISC processor could handle the task. The choice for best combination of processing power, space requirement and power consumption has lead CES to use the R3052 from IDT.

## *1.7 Transmission modes*

The Fast Link protocol supports two transmission modes:

- Asynchronous
- Isochronous (or global synchronous)

The asynchronous mode is the standard transmission mode used for the transmission of large blocks of data. The isochronous mode organises time slots and is used for data transfers which require a deterministic response time, either register oriented or block transfer oriented. The isochronous transfers are activated periodically, with programmable period and time slots. In both modes, the global time information is maintained automatically through the complete system and is made available to the outside world. The two transmission modes can be used simultaneously which makes the Fast Link as powerful for large data blocks, as for single shot operations (such as VME registers examination, time stamping, event tagging, ....).

## *1.8 Data Structure*

The data blocks are organised by the on-board firmware in 1 Kbyte packets, which are divided into 32-byte elementary cells.

The Fast Link protocol supports three types of packets:

- **Standard**      Used for normal data transportation.
  - **High priority**      Used to transport control and status information, link requests and acknowledgements, as well as to transmit any emergency information.
  - **Isochronous**      Used when deterministic operations are required, have the same structure as standard packets, but are inserted at regular time slots.
-



## 2. MODERN RISC PROCESSORS

### 2.1 Introduction

The processors presented are provided to perform mainly two functions in a Real-Time system and thus classified in two categories.

The first type enters in the family of high computing power processors suited for large event builder farms in High Energy Physics experiments, medical imaging or flight simulation in aerospace applications while the second type is an I/O intelligent platform to receive and send large data flow after decoding and encoding the packets using the on-board communication channels which are available.

For the first category mentioned, we will introduce both the Power-PC and the R4600 CPU based processors. Then, we will describe in the following, a new RISC Input Output VME I/O board equipped with a Power-PC 603 processor and with two high speed data communication channel.

### 2.2 The VME Real-Time Processors with VME / VSB / PCI Interfaces and Intelligent List Processor: RTPC 8067 and RAID 8240

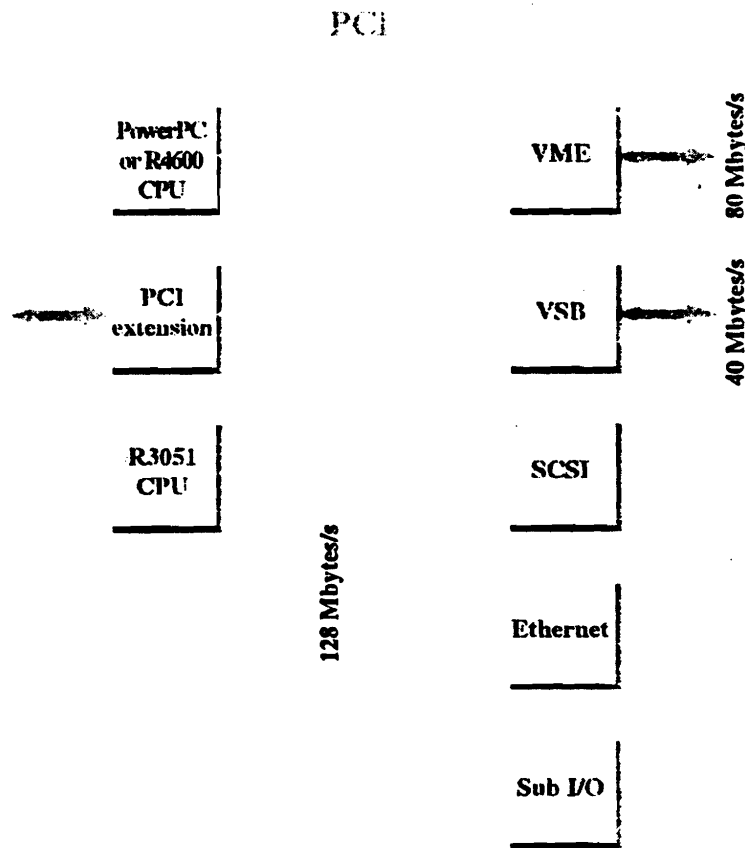


Fig. 4. Overview of the Block Diagram of the Processors

### 2.2.2 The CPU daughter board

The CPU subsystem is contained on a daughter card. The RTPC 8067 main CPU is the Power-PC 603 from IBM clocked at 66 MHz and has an on-chip 2 x 8 Kbytes first level cache. It is interfaced to a 256 Kbytes second level cache and is bridged to the PCI bus by means a Bridge chipset.

The RAID 8240 is based on the R4600 ORION MIPS processor, 133 MHz, 133 MIPS, 44 MFLOP/sec, 90 SPECint92, 80 SPECfp92. It has an on-chip 2 x 16 Kbytes first level cache and is interfaced 512 Kbyte second level cache.

The DRAM size ranges from 8 Mbytes to 128 Mbytes for both processors.

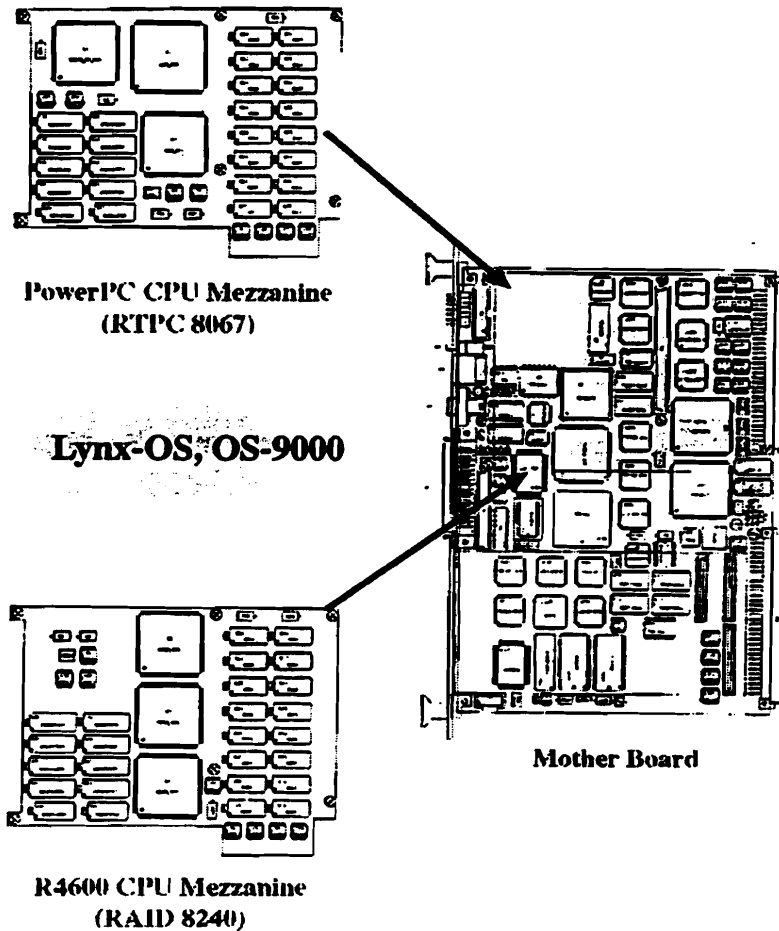


Fig. 5. RTPC and RAID Physical Implantation

## List Of Vendors

Vu Anh Lai

W.L. Gore & Associates  
7811 Burleson - Manor Rd.  
Manor TX 78653

Mike Schwieterman  
mikes@efficient.com  
Efficient Networks  
4201 Spring Valley Road Suite 120  
Dallas TX 75244-3666

Don Pederson  
donavanp@aol.com  
AnCor Communications  
% ComNet Research Corp.  
2817 Anthony Lane South  
Minneapolis MN 55418

Richard Sumner  
sumner@fnalv.fnal.gov  
LeCroy Corporation  
700 Chestnut Ridge Rd.  
Chestnut Ridge NY 10977

Knut Alnes  
knal@netcom.com  
Dolphin Interconnect Solutions USA  
5201 Great America Parkway Suite 3  
Santa Clara CA 95054

Jerry Rawls  
jsrawls@aol.com  
Finisar Corporation  
3515 Edison Way  
Menlo Park CA 94025

David Brewster  
75054.754@compuserve.com  
Siemens Industrial Automation Inc.  
100 Technology Drive  
Alpharetta GA 30202

John McDonough  
EG&G Nuclear Instruments  
100 Midland Road  
Oak Ridge TN 37831

Bob Rauworth  
KineticSystems Corporation  
11 Maryknoll Dr.  
Lockport IL 60441

Keith Mansk  
none  
W.L. Gore & Associates  
7811 Burleson - Manor Rd.  
Manor TX 78653

Ian Mahood  
Alcatel Network Systems  
2912 Wake Forest Rd  
Raleigh NC 27609

## List Of Attendees

Farhad Abar  
fnal::abar  
Fermilab  
Fermilab MS-368  
P.O. Box 500  
Batavia IL 60510

David J. Abbott  
abbottd@cebaf.gov  
SURA/CEBAF  
12000 Jefferson Ave. (MS-12H)  
Newport News VA 23602

Knut Alnes  
knal@netcom.com  
Dolphin Interconnect Solutions USA  
5201 Great America Parkway Suite 3  
Santa Clara CA 95054

Pierre Amaudruz  
  
TRIUMF  
4004 Wesbrook Mall  
Vancouver CANADA V6T 2A3

John Anderson  
janderson@fnal.fnal.gov  
Fermilab  
FERMILAB MS - 234  
P.O. Box 500  
Batavia IL 60510

Vu Anh Lai  
  
W.L. Gore & Associates  
7811 Burleson - Manor Rd.  
Manor TX 78653

Perry Anthony  
anthony@slac.stanford.edu  
Stanford Linear Accelerator Center  
P.O. Box 4349  
MS42  
Stanford CA 94309

Karim Ashktorab  
karim@bnlhi2  
Brookhaven National Laboratory  
Bldg. 510D Physics Department  
Upton NY 11973-5000

Michael Athanas  
anhanas@lns62.lns.cornell.edu  
University of California  
Wilson Lab , Cornell University  
Ithaca NY 1485

Robert Ball  
mich::ball  
University of Michigan  
Physics Dept., University of Michigan  
500 E. University  
Ann Arbor MI 48109-1120

Ulf Behrens  
behrens@x4u2.desy.de  
DESY  
Notkestrasse 85 D-W-2000  
Hamburg 52 GERMANY

Dave Berg  
BERG@FNAL  
Fermilab  
FERMILAB MS-120  
P.O. Box 500  
Batavia IL 60510

Kurt Biery  
biery@fnald.fnal.gov  
McGill University  
CDF/ Fermilab MS-318  
Batavia IL 60510

Dennis Black  
fnal::black  
Fermilab  
FERMILAB MS-120  
P.O. Box 500  
Batavia IL 60510

Robert E. Blair  
reb@anlhp  
Argonne National Laboratory  
9700 South Cass Avenue  
Argonne IL 60439

Andre Bogaerts  
bogaerts@dxcern.cern.ch  
CERN  
CH - 1211  
Geneva 23 SWITZERLAND

Joanne Bogart  
jrb@sld.slac.stanford.edu  
SLAC  
SLAC MS/71  
Stanford CA 94305

Fred Borcharding  
fredob@fnal  
Fermilab  
Fermilab MS-318  
P.O. Box 500  
Batavia IL 60510

## List Of Attendees

Francois Bourgeois  
 bourfran@cernvm.cern.ch  
 CERN  
 ECP Division  
 Geneva 23 SWITZERLAND

James G. Branson  
 branson@ucsd.edu  
 UCSD

David Brewster  
 75054.754@compuserve.com  
 Siemens Industrial Automation Inc.  
 100 Technology Drive  
 Alpharetta GA 30202

Don Briggs  
 briggs@slac.stanford.edu  
 SLAC  
 MS 95  
 2575 Sand Hill Rd  
 Menlo Park CA 94025

Gean C. Brisson  
 brissons@frcpn11.in2pe.fr  
 Saclay

Joel Butler  
 butler@fnal  
 Fermilab  
 Fermilab MS-120  
 P.O. Box 500  
 Batavia IL 60510

Tom Carey  
 tcarey@lanl.gov  
 Los Alamos National Laboratory  
 Group P-25, MS-D456  
 Los Alamos NM 87545

Tom Carter  
  
 Fermilab  
 Fermilab MS - 122  
 P.O. Box 500  
 Batavia IL 60510

Fabrizio Catarsi  
 cean@vm.cnuce.cnr.it  
 CAEN  
 Via Vetraria 11 55049 Viareggio (LU)  
 ITALY

Cheng-Yi Chi  
 chi@nevis.nevis.columbia.edu  
 Nevis Lab  
 Columbia University/Nevis lab  
 136 South Broadway  
 Irvington NY 10533

Michael Chorowicz  
 ces@leacy.ces.ch  
 Creative Electronic Systems S.A.  
 70, route Du Port Butin  
 1213 Petit Lancy  
 Geneva 23 SWITZERLAND

Richard Claus  
 claus@slac.stanford.edu  
 SLAC  
 P.O. Box 4349 MS-17  
 Stanford CA 94309

Dieter Cords  
 CEBAF MS-12H  
 CEBAF  
 12000 Jefferson Ave  
 Newport News VA 23606

Dario S. Crosetto  
 CROSETTO@SSCVX1.SSC.GO'  
 SSCL  
 SSCL  
 2550 Beckleymeade Ave.  
 Dallas TX 75237

Roger Cummings  
 Roger\_Cummings@Stortek.com  
 StorageTek  
 MS 0268  
 2270 South 88th St.  
 Louisville CO 80028-0268

David Cutts

Brown University  
 MS-357

John W. Dawson  
 jwd@hep.anl.gov  
 Argonne National Laboratory  
 9700 S. Cass Avenue Bldg. 362  
 Argonne IL 60439

Chris Day  
 Ctday@lbl.gov  
 Lawrence Berkeley Laboratory  
 1 Cyclotron Rd.  
 MS 50B-3238  
 Berkeley CA 94720

Julius Dejongh  
 juls@fawn.nl.nuwc.navy.mil  
 Naval Uncersear Farfare Center  
 Code 3311 Building 80  
 New London CN 06320

## List Of Attendees

Manuel Delfino  
delfino@scri.fsu.edu  
Supercomputer Computations Research  
Florida State University B-186  
Tallahassee FL 32306-4052

Phil Demar

Fermilab  
Fermilab MS-368  
P.O. Box 500  
Batavia IL 60510

K. Djidi  
dijidi@hpe.saclay cea.fr  
C.E.A. Saclay  
Centre d'Etudes de Saclay  
91190 GIF/YVETTE  
FRANCE

David C. Doughty  
doughty@clas01.ceba.gov  
CEBEF  
12000 Jefferson Av, MS12H  
Newport News VA 23606

Pierre-Yves Duval  
duval@marvax.in2p3.fr  
Centre de Physique des Particules de N  
163 avenue de Lumini

John E. Elias  
Elias@fnal  
Fermilab  
Fermilab MS-318  
P.O. Box 500  
Batavia IL 60510

Ton Engbersen  
apj@zurich.ibm.com  
IBM Research Division  
Zurich Research Laboratory  
Saemmerstrasse 4, CH-8803 Rueschlikon  
Geneva SWITZERLAND

Juergen Fent  
juf%dmumpiwh.bitnet@vm.gmd.de  
Max Planck Institut fuer Physik  
Fohringer Ring 6  
GERMANY

Maria Lorenza Ferrer  
ferrer@ln.infn.it  
INFN - LNF  
Via Enrico Fermi 40 00044 Frascati  
Rome ITALY

Mark Fischler  
mf@fnal.gov  
Fermilab  
Fermilab MS 368  
P.O. Box 500  
Batavia IL 60510

Bob Forster  
forster@fnal.gov  
Fermilab  
Fermilab MS-234  
P.O. Box 500  
Batavia IL 60510

David Francis  
djf@cernvm  
CERN  
Division PPE, CERN, CH-1211 Suisse  
Geneva 23 SWITZERLAND

Adolfo Fucci  
CERN  
CERN/ECP  
Geneva 23 SWITZERLAND

Stuart Fuess  
fuess@fnal.gov  
Fermilab  
Fermilab MS-357  
P.O. Box 500  
Batavia IL 60510

Irwin Gaines  
gaines@fnacp.fnal.gov  
Fermilab  
Fermilab MS 127  
P.O. Box 500  
Batavia IL 60510

Rob Gardner  
rwg@uihepa.hep.uiuc.edu  
University of Illinois  
412 Loomis Laboratory of Physics  
1110 West Green Street  
Urban IL 61801

Lee Goldberg  
no email  
Electronic Design Magazine  
611 Route #46 West  
Hasbrouck NJ 07604

Paulo Gomes  
gomes@lipulsi.lip.pt  
LIP  
Av. Elias Garica, 14-1 P - 1000  
Libson PORTUGAL

## List Of Attendees

Hector Gonzalez  
gonzalez@fnal  
Fermilab  
Fermilab MS-222  
P.O. Box 500  
Batavia IL 60510

William Greiman

Lawrence Berkeley  
1 Cyclotron Road Bldg 50B/3238  
Berkeley CA 94720

Alberto Guglielmi  
guglielmi@cern.gvc.dec.com  
Digital  
Digital Equipment Corporation  
Geneva 23 SWITZERLAND

Vijay Gurbani  
vijay@fndaut.fnal.gov  
Fermilab  
Fermilab MS-234  
P.O. Box 500  
Batavia IL 60510

Leif Gustafsson  
lrg@tsl.uu.se  
Uppsala University, ISV  
THUNBERG SVAGEN 5, BOX 535  
Uppsala SWEDEN 751 21

John Haggerty  
haggerty@bnlku2.phy.bnl.gov  
Brookhaven National Laboratory  
Physics Dept.  
Upton NY 11973-5000

Richard D. Hance  
hance@fnal.gov  
Fermilab  
Fermilab MS-357  
P.O. Box 500  
Batavia IL 60510

Mike Haney  
m-haney@uiuc.edu  
University of Illinois  
457 Loomis Lab  
1110 W. Green St.  
Urbana IL 61801

John R. Hansen  
renner@nbivax.nbi.dk  
Niels Bohr Institute  
Blegdamsvej 17 DK-2100 Copenhagen  
DENMARK

Jorn D. Hansen  
dines@nbivax.nbi.dk  
Niels Bohr Institute  
Blegdamsvej 17  
KD-2100 DENMARK

John Harvey  
harvey@aloni.cern.ch  
CERN  
ECP Division Cern, CH 1211  
Geneva 23 SWITZERLAND

Bill Haynes  
bill.haynes@rl.ac.uk  
Rutherford Appleton Laboratory  
Chilton, DIDCOT,  
Oxfordshire, GREAT BRITAIN

Roger Heeley  
rhelley@ecpdaharmony.cern.ch  
CERN  
Meyrin, 1211  
Geneva 23 SWITZERLAND

Jack Hoeflich  
jjh@slac.stanford.edu  
SLAC  
SLAC MS/71  
Stanford CA 94305

Jan S. Hoftun  
hoftun@brhep1.physics.brown.edu  
Brown University  
Dept. of Physics  
Box 1843, Brown University  
Providence RI 02912

Michael Huffer  
mehsys@slo.slac.stanford.edu  
SLAC  
PO Box 4349 MS/71  
Palo Alto CA 94309-4349

Kudla M. Ignacy  
kudla@cernvm  
CERN  
CERN ECP Division  
Geneva 23 SWITZERLAND

Sami Inkinen  
sinkin@cernvm.cern.ch  
CERN  
CERN / ECP  
Geneva 23 SWITZERLAND

## List Of Attendees

Walter Innes  
walt@slac.stanford.edu  
SLAC  
PO Box 4349  
Stanford CA94309

Werner Jank  
werner\_jank@cern.ch  
CERN  
CERN/ECP  
Geneva 23 SWITZERLAND

Rolland Johnson  
rolland.johnson@mailgw.er.doe.gov  
DOE  
Division of High Energy Physics, EF  
U.S. Department of Energy, GTN  
Washington D.C.20545

Clint Jurgens  
ClintJ@tricord.ancor.com  
AnCor Communications  
6130 Blue Circle Drive  
Minnetonka MN 55343

Alexander Kluge  
alex@sunlinz.chern.ch  
CERN  
CERN CH-1211  
Geneva 23 SWITZERLAND

Joseph G. Kneuer  
  
AT&T Bell Laboratories  
101 Crawford Corners Rd  
Room 4B-505  
Halmders NJ07733

Thomas Kozlowski  
kozlowski\_thomas@lanl.gov  
Los Alamos National Laboratory  
Los Alamos  
Los Alamos NM87545

Ernst Kristiansen  
Ernst.kristiansen@fys.uio.no  
SINTEF  
P.O.Box 124, Blindern  
N-0314, OSLO NORWAY

Paul N. Krystosek  
krystosek@anl.gov  
Argonne National Laboratory  
9700 S. Cass Avenue ECT 221  
Argonne IL60439

Timothy C. Kuhfuss  
kuhfuss@anl.gov  
Argonne National  
9700 S. Cass Ave. ECT 221  
Argonne IL60439

Larry Larsen  
larry@bit3.com  
Bit 3 Computer  
8120 Penn Ave. South  
Minneapolis MN55431

Marcus H. Larwill  
larwill@fnal.fnal.gov  
Fermilab  
Fermilab MS-222  
P.O. Box 500  
Batavia IL60510

Jean-Yves LeBoudec  
leboudec@di.epfl.ch  
EPFL-DI  
Labo Reseaux de Comm 6610  
SWITZERLAND

Michael F. Letheren  
letheren@sunvlsi.cern.ch  
CERN  
ECP Division, CERN  
Geneva 23 SWITZERLAND

Michael Levi  
levi@lbl.gov  
Lawrence Berkeley  
MS 50A-2160, Physics Division  
Berkeley CA94720

Michael LeVine  
levine@bnl.gov  
Brookhaven National Laboratory  
20 Pennsylvania Avenue 510D  
Upton NY 11973-5000

Lorne Levinson  
fhlevins@weizmann.weizmann.ac.i  
Weizmann Institute of Science  
Rehovot 76100  
ISRAEL

Volker Lindenstruth  
lindenstruth@csa5.lbl.gov  
Lawrence Berkeley Laboratory  
Bldg 50D Room 111  
1 Cyclotron Rd, M/S 50D  
Berkeley CA94720



## List Of Attendees

Bo Lofstedt  
LOFSTEDT@VXCERN.CERN.CH  
CERN  
CH-1211  
Geneva 23 SWITZERLAND

Abdenour Lounis  
lounis@lancy.ces.ch  
Systems & Sales Department  
System & Sales Department  
70 Route Du Port Butin  
Geneva 23 SWITZERLAND

Yngvar Lundh  
yngvar@ifi.uio.no  
University of Oslo  
Institute for Informatics  
P.O. Box 1080 Blindern  
Oslo NORWAY

Bob Maher  
Maher@sqg89.bwi.wec.com  
Alliant Tech Systems.  
Alliant Tech Systems /WEC  
Mukilteo WASHINGTON 98275

Ian Mahood

Alcatel Network Systems  
2912 Wake Forest Rd  
Raleigh NC27609

Irakli Mandejavidze  
mandjavi@sunvlsi.cern.ch  
CERN  
CH 1211  
Geneva 23 SWITZERLAND

Keith Mansk  
none  
W.L. Gore & Associates  
7811 Burleson - Manor Rd.  
Manor TX 78653

Alexandro Marchioro  
marchior@boing.cern.ch  
CERN  
Route de Meyrin  
Geneva 23 SWITZERLAND

Robert Martin  
rmartin@bnl.gov  
College of William  
Department of Physics  
College of William and Mary  
Williamsburg VA 23187

J.P. Martin  
  
University of Montreal

Pietro Matteuzzi  
matteuzzi@inf.n.it  
INFN-CNF  
Viale Ercolani, 8 I-40138  
Bologna ITALY

Larry McAdams  
mcadams@optivision.com  
Optivision  
4009 Miranda Av.  
Palo Alto CA 94305

Ian McArthur  
i.mcarthur@physics.oxford.ac.uk  
Oxford University  
Keble Rd  
Oxford UK

John McDonough  
  
EG&G Nuclear Instruments  
100 Midland Road  
Oak Ridge TN 37831

Daniel Mendoza  
fnal::mendoza  
Fermilab  
Fermilab MS - 357  
P.O. Box 500  
Batavia IL 60510

Martin Moebus  
moebus@vsdec.nl.nuwc.navy.mil  
NUWC  
Code 3331  
New London CN 06320

Michael Mojaver  
mojaver@sdphv1.ucsd.edu  
University of California

Thomas Moore  
Moore25@lnl.gov  
Lawrence Livermore Laboratory  
7000 East Avenue.  
Livermore CA 94550

Carmenita Moore  
fndaq:moore  
Fermilab  
Fermilab MS-120  
P.O. Box 500  
Batavia IL 60510

Manuel S. Mota  
mota@lipvlsi.lip.pt  
LIP  
AV. Elias Garcia, 14-11000  
Lisboa PORTUGAL

## List Of Attendees

Klaus Mueller  
k.d.mueller@kfa-juelich.de  
KFA Juelich  
GERMANY

Nasushi Nagasaka  
nagasaka@kekvox.kek.jp  
KEK  
National Laboratory for High energy  
Online Group, Dept. of Physics  
Ibaraki-ken 305 JAPAN

Wayne Nation  
nation@vnet.ibm.com  
IBM Corporation  
3605 US 52 North  
Rochester NY 55901

Ron Nelson  
ron@lanl.gov  
Los Alamos National Laboratory  
MS H805  
Los Alamos NM 87545

Norbert Neumeister  
neumeist@cernvm.cern.ch  
CERN  
Institut For High Energy Physics  
PPPE CH-1211  
Geneva 23 SWITZERLAND

Gareth Noyes  
gareth@gareth.desy.de  
Rutherford Appleton Laboratory  
H1/F22, DESY Notkestrasse 85  
Hamburg GERMANY 22603

Vivian O'Dell  
fnal::odell  
Fermilab  
FERMILAB MS-234  
P.O. Box 500  
Batavia IL 60510

Holger Oelschlaeger  
sales@struck.de  
Dr. B. Struck Co.

Karen Ohl

Fermilab MS-318  
Fermilab MS-318  
P.O. Box 500  
Batavia IL 60510

Gene Oleynik  
fnal::oleynik  
Fermilab  
FERMILAB MS-120  
P.O. Box 500  
Batavia IL 60510

Jim Omer  
sales@struck.de  
Dr. B. Struck Co.

Gerard Oxoby  
gjeb@slac.stanford.edu  
Stanford Linear  
SLAC MS62, P.O. Box 4349  
Stanford CA 94309

James Pangburn  
pangburn@D0.fnal.gov  
Fermilab  
Fermilab- MS-318  
P.O. Box 500  
Batavia IL 60510

James Patrick  
patrick@fnal.fnal.gov  
Fermilab  
Fermilab MS-318  
P.O. Box 500  
Batavia IL 60510

Don Pederson  
donavanp@aol.com  
AnCor Communications  
% ComNet Research Corp.  
2817 Anthony Lane South  
Minneapolis MN 55418

Don Petravick  
fnal::petravick  
Fermilab  
Fermilab MS-234  
P.O. Box 500  
Batavia IL 60510

Barry Phillips

Adger Smythe Corp.

Renee Poutissou  
renee@triumf.ca  
TRIUMF  
4004 Wesbrook Mall  
Vancouver, BC CANADA V6T 2A3

Attila Racz  
racz@cernvm.cern.ch  
CERN  
CERN  
Geneva 23 SWITZERLAND

## List Of Attendees

Bob Rauworth

KineticSystems Corporation  
11 Maryknoll Dr.  
Lockport IL60441

Jerry Rawls

jsrawls@aol.com  
Finisar Corporation  
3515 Edison Way  
Menlo Park CA94025

John D. Roof

fnalv::roof  
Fermilab  
FERMIALB MS-318  
P.O. Box 500  
Batavia IL60510

Carmen Rotolo

larwill@fnal.fnal.gov  
Fermilab  
Fermilab MS - 222  
P.O. Box 500  
Batavia IL60510

Robert Russell

rdr@cs.unh.edu  
University of New Hampshire  
Computer Science Department  
Durham NH 03824

James J. Russell

SLAC  
PO Box 4349 MS/71  
Palo Alto CA94309-4349

Vladimir Rybnikov

rybnikov@vxdesy.desy.de  
DESY  
Notkestr. 85 D-22603  
Hamburg GERMANY

Earl E. Rydell

eerydell@cacd.rockwell.com  
Rockwell - Collins CACD  
400 Collins Rd N.E. MS 107-140  
Cedar Rapids IOWA52498

Valeri Rytchenkov

none  
Fermilab  
Fermilab MS-307  
P.O. Box 500  
Batavia IL60510

Davide Salomoni

salomoni@infn.it  
INFN-CNAF  
Viale Ercolani, 8 I-40138  
Bologna ITALY

Osamu Sasaki

sosamu@kekvox.kek.jp  
KEK  
Physics Division, 1-1 Oho  
Tsukuba Ibaraki 305 JAPAN

Joachim Schambach

jschamba@utpapa.ph.utexas.edu  
University of Texas  
Physics Department  
RLM 5.208  
Austin TX78712

Peter Schulz

schuetz@cernvm.cern.ch  
CERN  
CERN/PPE, CH-1211  
Geneva 23 SWITZERLAND

Ulrich Schwendicke

ulrich@ifh.de  
DESY  
Plataneallee 6  
Zeuthen GERMANY

Mike Schwieterman

mikes@efficient.com  
Efficient Networks  
4201 Spring Valley Road Suite 1200  
Dallas TX75244-3666

Peter Sharp

psh@uk.ac.rc.rl.ib  
Rutherford Appleton Laboratory  
Chilton, Didcot Oxfordshire

Theresa M. Shaw

Tshaw@fnal  
Fermilab  
Fermilab MS-331  
P.O. Box 500  
Batavia IL60510

Bernard Skaali

t.b.skaali@fys.uio.no  
University of Oslo  
Department of Physics  
Oslo NORWAY

## List Of Attendees

Jean Slaughter  
Slaughter@fnal  
Fermilab  
MS-221 E791  
P.O. Box 500  
Batavia IL 60510

Dave Slimmer  
fnal::slimmer  
Fermilab  
FERMILAB MS-120  
P.O. Box 500  
Batavia IL 60510

J.B. Spelt  
jansp@paramount.nkkalk.nikhef.nl  
NIKHEF  
P.O. Box 41889100g DB  
Amsterdam THE

Ralf Spiwoks  
spiwoks@lhctb01.cern.ch  
CERN  
CERN/ECP  
CH-1211 Geneva 23, Switzerland  
Geneva 23 SWITZERLAND

Jonathan Streets  
STREETS@FNAL  
Fermilab  
FERMILAB MS-120  
P.O. Box 500  
Batavia IL 60510

Walter Stuermer  
Stuermer@fnal  
Fermilab  
Fermilab -MS - 331  
P.O. Box 500  
Batavia IL 60510

Richard Sumner  
sumner@fnalv.fnal.gov  
LeCroy Corporation  
700 Chestnut Ridge Rd.  
Chestnut Ridge NY 10977

Sham Sumorok  
sumorok@mitlns.mit.edu  
MIT  
24-033 MIT/LNS  
77 MASS Ave  
Cambridge MA 02139

Zenon M. Szalata  
zmsanu@slac.stanford.edu  
SLAC  
Stanford Linear Accelerator Ctr.  
Stanford University, Po Box 4349  
Stanford CA 94305

Masaru Tairadate  
taira@kek.jp  
The Graduate Univ. for Advanced Stud  
Computing Center  
National Lab. for High Energy Physic  
(KEK)1-1 OHO,

Steve Tether  
tether@cdfmt1.fnal.gov  
MIT  
MIT MS 318 Fermilab  
P.O. Box 500  
Batavia IL 60510

Murray Thompson  
thompson@wishpa.physics.wisc.edu  
University of Wisconsin  
1150 University Ave  
Madison WI 53706

Roberts Timellini  
time@vxcmr  
INFN Bologna  
RET Des Tetras IT  
01710 Thoiry FRANCE

Charles Timmermans  
Dept. of Physics 0 .  
Univeristy of Minnesota  
116 Church St.  
Minneapolis MINNISOTA 55455

Robert Trendler  
trendler@fnal  
Fermilab  
Fermilab MS-208  
P.O. Box 500  
Batavia IL 60510

Oscar Trevizo  
TREVIZO@FNAL  
Fermilab  
FERMILAB MS-234  
P.O. Box 500  
Batavia IL 60510

Joerg Tutas  
tutas@dice2.desy.de  
University of Heidelberg  
c/o DESY FH1K  
Hamburg GERMANY

Lourdu R. Udumula  
udumula@fndaut.fnal.gov  
Fermilab  
Fermilab MS-120  
P.O. Box 500  
Batavia IL 60510

## List Of Attendees

Michael J. Utes  
fnal::utes  
Fermilab  
Fermilab MS - 357  
P.O. Box 500  
Batavia IL60510

Hendrik Van Der Bij  
vanderby@vxcern.cern.ch  
CERN  
ECP Division, CERN  
Geneva 23 SWITZERLAND

Pierre Vande Vyvre  
pvv@vxcern.cern.ch  
CERN  
ECP Division  
Geneva 23 SWITZERLAND

Andrew VanderMolen  
vdmolen@nscl.nslc.msu.edu  
National Superconducting Cyclotron  
Michigan State University  
E. Lansing MI48824-1321

Joao Varela  
varela@axlipo.cern.ch  
CERN  
CERN  
Geneva 23 SWITZERLAND

Alessandro Vascotto  
alessandro.vascotto@cern.ch  
CERN  
CH 1211  
Geneva 23 SWITZERLAND

Sandro Ventura  
INFN  
Padova Via Marzolo 8  
Padova ITALY

Richard Vidal  
FNALD::Rivdal  
Fermilab  
Fermilab MS-318  
P.O. Box 500  
Batavia IL60510

Margaret Votava  
votava@fnal.gov  
Fermilab  
Fermilab MS-120  
P.O. Box 500  
Batavia IL60510

Duane Voy  
Voy@crusher.fnal.gov  
Fermilab  
Fermilab MS-341  
P.O. Box 500  
Batavia IL60510

Donald Walsh  
dwalsh@daesn4-fnal-gov  
Fermilab  
Fermilab MS-222  
P.O. Box 500  
Batavia IL60510

Peter Wegner  
wegnerp@ifh.de  
DESY  
DESY - IFH Zeuthen  
Zeuthen GERMANY

Vicky White  
white@fnal  
Fermilab  
Fermilab MS-120  
P.O. Box 500  
Batavia IL60510

Fred Wickens  
f.wickens@rl.ac.uk  
Rutherford Appleton Laboratory  
Chilton,Nr Didcot,  
ENGLAND

Andre Wiesel  
wiesel@ltisun.epfl.ch  
Laboratoire de TeleInformatique  
Departement d'Informatique  
INN - ECUBLENS  
Geneva 23 SWITZERLAND

Torsten Wildschek  
wildsche@cernvm.cern.ch  
Instiute fuer Hochenergiephysik  
Nikolsdorfergasse 18  
AUSTRIA

Holger Witsch  
witsch@esrf.fr  
European Synchrotron  
Av. des Martyrs  
Cedex FRANCE

Andreas Wolf  
aw@lms62.lms.cornell.edu  
Ohio State Univ.  
174 W. 18th Avenue  
Columbus OHIO43210

## List Of Attendees

Elliott Wolin  
wol@wmheg.physics.wa.edu  
College of  
P.O. Box 8795  
Williamsburg VA 23187-8795

Bin Wu  
bin.wu@fys.uio.no  
CERN  
Department of Physics  
CH 1211  
Geneva 23 SWITZERLAND

Claudia E. Wulz  
wulz@cernvm.cern.ch  
CERN  
Institute for H.E. Physics, Vienna & C  
CH-1211  
Geneva 23 SWITZERLAND

Sergio Zimmermann  
fnal::zimmer  
Fermilab  
Fermilab MS-222  
P.O. Box 500  
Batavia IL 60510

Klaus Zwoll  
k.d.mueller@kfa-juelich.de  
KFA Juelich  
52425 Juelich  
GERMANY

## **The Conference Organizing Committee:**

Ed Barsotti	Fermilab	barsotti@fnalv.fnal.gov
Mark Bowden	Fermilab	mark_bowden@qmgate.fnal.gov
Sergio Cittolin	CERN	sergio_cittolin@macmail.cern.ch
Robert Downing	Illinois	rwd@uihepa.hep.uiuc.edu
Jean-Pierre Dufey	CERN	jpd@vxomeg.cern.ch
Bill Haynes	Fermilab	haynes@fnalv.fnal.gov
Maribel Herrera	Fermilab	marih@fncd00.fnal.gov
Marvin Johnson	Fermilab	mjohnson@fnalv.fnal.gov
Walter Knopf	Fermilab	knopf@fnal.fnal.gov
Patrick LeDu	SACLAY	ledu@dphvx2.saclay cea.fr
Livio Mapelli	CERN/LBL	mapelli@lbl.gov
Robert McLaren	CERN	mclaren@vxcern.cern.ch
Hans Muller	CERN	hans@sunshine.cern.ch
Masa Nomachi	KEK	nomachi@kekvox.kek.jp
Ruth Pordes	Fermilab	ruth@fnalv.fnal.gov
Paris Sphicas	MIT	paris@fnald.fnal.gov
Sonya Wright	Fermilab	sonya@fnalv.fnal.gov

## **The Local Organizing Committee:**

Elizabeth Brown	Fermilab
Denise Bumbar	Fermilab
John Elias	Fermilab
James Franzen	Fermilab
Cynthia Sazama	Fermilab
Colleen Yashikawa	Fermilab

Program Design by James Franzen  
Program Front Cover Logo by Sergio Cittolin

## **VME Standards Organizations & Our Physics Community Working Together On Extending VME Standards For Physics Applications**

VME has become popular in data acquisition systems over the last few years because of industry's huge product support of the standard and because of the large number of board level processors, associated compilers, operating systems, debuggers, etc., available for these modules. These VME systems have also found their way into near front end use with mixed results. Additionally many physics laboratories and universities have their own version of the VME standard. In fact, there are several versions of VME at single laboratories. One such laboratory has implemented fourteen different versions of VME!

Recently, the VME Standards Organization (VSO) and the VME International Trade Association (VITA) have invited areas of industry and science to form special VME interest groups so their specific requirements can be met through officially sanctioned extensions to the VME standard. For the last few months, members of the physics community worldwide have been involved with the formation of a VME interest group for physics (VME-P). ESONE, the European standards organization and NIM, its North American counterpart, with substantial participation by CERN, Fermilab and increasingly more labs and universities, have been working independently and with VSO and VITA to further define our needs to VSO and VITA. For example VITA, the main body responsible for changes in the extended base VME specification, has been working with connector manufacturers on a keying mechanism for the VME backplane. If this mechanism was to become part of the extended VME specifications, special interest groups can assign functionality to user pins and still maintain compatibility with pure VME modules and VME modules from other special interest groups. Thus, modules designed for physics applications can only plug into VME crate slots with specific keying for these modules. Likewise, these modules would not be able to plug into other VME slots and other VME modules would not be able to plug into VME crate slots keyed for physics applications. Features such as special voltages, higher power, geographical addressing, etc., can be added for our physics applications while maintaining compatibility and interchangeability with the base VME standards. Keying alone, should all but eliminate the very costly VME variations within labs and universities in future system implementations.

A conference presentation first thing Thursday morning, S3-6 "New VME Standards For Physics Applications", gives further details of our initial work in this area. We also have sign up sheets at the registration desk for people interested in participating in and/or reviewing our VME for physics interest group's standards work. Thus far there is both a North American and European working group for this work. We hope to add a Japanese working group or minimally Japanese participants to this effort. Hopefully, this effort will significantly reduce the need for implementers to design their own in-house packaging and bus systems.



