SSCL-PP-445

Digital Programmable Level-1 Trigger with 3D-Flow Assembly

D. Crosetto

August 1993

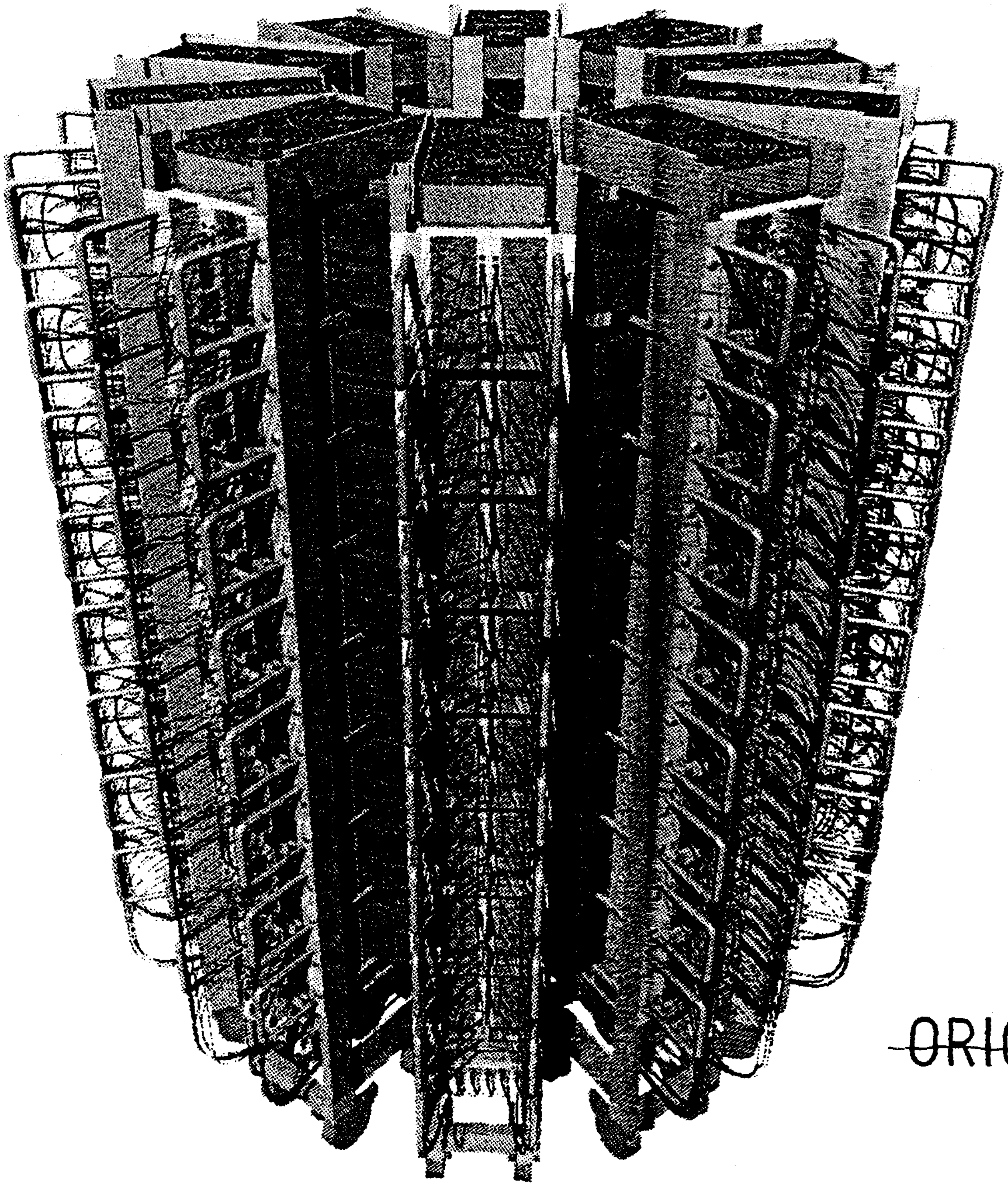


Figure 1. 3D-Flow system in a cylindrical assembly (camera view).

SSCL-Preprint-445

Digital Programmable Level-1 Trigger with 3D-Flow Assembly

D. Crosetto

Superconducting Super Collider Laboratory*
2550 Beckleymeade Ave.
Dallas, TX 75237

August 1993

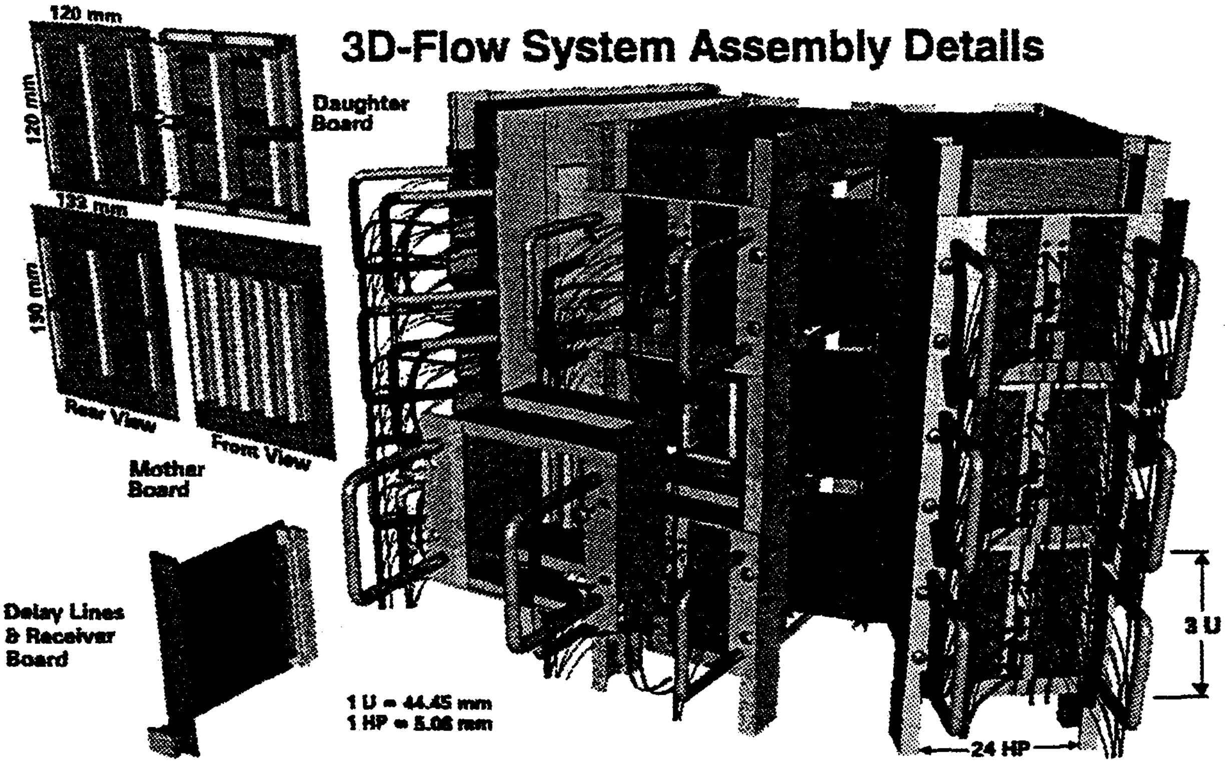


Figure 2 3D-Flow system assembly details (camera view).

* Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

Digital Programmable Level-1 Trigger with 3D-Flow Assembly

D. Crosetto

Abstract

The 3D-Flow parallel processing system (See Figures 1 and 2) is a new concept in processor architecture, system architecture, and assembly architecture. Compared to the electronics used in present systems, this approach reduces the cost and complexity of the hardware and allows easy assembly, disassembly, incremental upgrading, and maintenance of different interconnection topologies. At present it is not possible to use the electronics for the Level-1 trigger of one experiment, and use it in another experiment, because most of them implement only one algorithm that may be different from one experiment to the next. The programmability of the 3D-Flow system offers the possibility of using the same electronics for different algorithms in different experiments. The size of overall racks is approximately 10% to 20% the size of racks used in other designs for similar systems. Using low-cost, standard components, this new design improves the capability of the system while allowing programmability not previously envisaged. The goal of this parallel-processing architecture is to acquire multiple data in parallel (up to 100 million frames per second) and to process them at high speed, accomplishing digital filtering on the input data, pattern recognition, data moving, and data formatting. The system is suitable for "particle identification" applications in high-energy physics (calorimeter data filtering, processing and data reduction, track finding and rejection), pattern recognition in radar systems, biological molecular studies, graphics processing, and other uses. The main features of the system are its programmability, scalability, high-speed communication, and low cost. The assembly uses standard, commercially available components (except for the 3D-Flow chip), thus minimizing cost. The 3D-Flow architecture makes possible the construction of a parallel-processing system with six-directional communication links between neighboring processors. It is suitable for the mapping of detector elements to processing elements, a solution that guaranties fast timing. Different detector element interconnection schemes can be efficiently implemented with the 3D-Flow parallel processing system in one-dimensional, two-dimensional, and three-dimensional interconnection topologies by arranging the system in a planar, cylindrical, or spherical assembly, respectively. The interconnection length is kept to a minimum, and the interconnection topology ensures short cable length and, therefore, fast data moving (in less then 2.5 ns using BiCMOS drivers), compared to the greater delay variations that can exist in conventional systems. The result is high speed with low power consumption. The greater cable length in conventional parallel-processing systems, besides requiring higher-power cable drivers, results in a delay in the arrival of data from the longest cable before the next operation is executed. Thus the total algorithm execution time increases. To sustain the same input data rate, both the algorithm execution time must increase along with the number of pipelined stages, thereby increasing the complexity and the cost. The use of a standard rack assembly allows a combination of existing data acquisition systems in 3U (Euroboard), 6U (VME), and 9U boards tightly and efficiently interconnected to a stack of boards in the parallel-processing system. The data acquisition and parallel-processing systems are joined at a 90° angle. The number of boards in the stack varies according to the required performance of the system, and incremental upgrading of the system at a later time is possible. Thus implementation of revised and optimized algorithms can be achieved by incorporating hardware advances with little effect on the installed system. A total of 6000 lines of VHDL code, describing the behavior of a single 3D-Flow processor and its interconnection with neighboring processors, has been completed. This allows the user to simulate algorithms and to check the timing of all signals in the circuit. A prototype of the Mini-Rack of the 3D-Flow system with daughterboards, motherboard, receiverboard and contollerboard has been built, and tests on the transmission of the signals have been performed with BiCMOS drivers up to 140 MHz.

FIGURES

TABLES

1.0 INTRODUCTION

Currently the Level-1 trigger on CDF and on D0 experiments [ref. 1, 2], and those in the baselines for Solenoidal Detector Collaboration (SDC) [ref. 3], and Gammas Electrons and Muons (GEM) [ref. 4], have a very limited degree of programmability (mainly programmable thresholds or at most one type of fixed algorithm implementation). Recent work, described in this report, shows that a programmable Level-1 trigger is both feasible and cost-effective. A 3D-Flow architecture [ref. 5] has been conceived that takes advantage of the new proposed programmable Level-1 trigger approach.

This work originated by understanding the requirements of Level-1 triggers for different experiments, past and present, as well as one designed for the future. Each one has been studied in some detail, with visits to the site of the experiment when possible and attempts to define a system architecture, processor architecture, and assembly architecture that had the commonality features to implement all of them.

The issues associated with pursuing this approach for use in current and future experiments are described in Sections 4 and 5. The cost of the entire Level-1 system is compared with the alternative non-programmable analog solution in the two cost-estimate reports (digital and analog) for the GEM experiment. The feasibility study of the entire system (racks, connectors, cable, etc.) assumes use of standard components and parts except, of course, for the 3D-Flow chip, for which a feasibility study has been carried out and cost estimates have been obtained from several companies. Work on the drawings and details of the system are described. The complete architecture of the 3D-Flow processor has been designed. The feasibility of the approach has been confirmed by industry experts and consultants. The behavioral functionality has been described in more than 6000 lines of code of VHDL (Very High Level Description Language) [ref. 6] and processes and procedures and the instruction set have been defined. Simulation of the most common Level-1 trigger algorithms shows that they may be executed efficiently in a very limited number of instructions on the 3D-Flow processor. (See Section 9.1.8 for a detailed example.)

These features make possible the design of a trigger that can satisfy current requirements and permit future growth opportunities by accepting new threshold sets, implementing revised and optimized algorithms, and incorporating hardware advances with little effect on the installed system.

The reader is referred to Figure 8, "3D-Flow system assembly details," in Section 5.4 as an aid in understanding this document. The figure provides an overview of the principal parts and how they are connected to form a physical system.

The description of the 3D-Flow processor--how its architecture is suitable in High Energy Physics experiments involving very high data rates (Gigabit/s), and how it is suitable to the Level-1 trigger algorithms--is found in Sections 3 and 9.1.8. Simulations and internal timing on the 3D-Flow processor are also provided in Section 9.1.8.2. Tesr results on signals transmission on a prototype are described in Sections 6.4.2 and 7.1.2.

2.0 GOAL

The goal is to implement a new, programmable Level-1 trigger by using a "3D-Flow" processor system. This will simplify the hardware and reduce the cost of Level-1 trigger systems. It can be used in current experiments and is intended to open doors to new ways of doing triggering in experimental high energy physics. This new, more powerful tool will allow implementation of different first level trigger algorithms, enabling researchers to find interesting events with much greater flexibility than existing approaches offer.

The concept is rather simple. The user translates any digital filter and/or pattern recognition, and/or data moving algorithm (from Monte Carlo simulation) into a real-time program of the type described in Table 2 of Report SSCL-607 and in [ref. 7]. The user's effort is minimal and typically requires writing only one and one-half pages of code.

Currently, different experiments use different electronics hardware that is not applicable to other experiments. This architecture is very flexible and uses only one small electronic board (12 cm × 12 cm) that includes 4 × 3D-Flow processor chips (already designed but yet to be synthesized into silicon from VHDL).

The way in which the 3D-Flow parallel-processing system maps the processing elements to the detector elements guarantees fast timing. An important parameter in the performance of a Level-1 trigger system is not only the processing capability, but also fast data communication between elements. The 3D-Flow system allows arranging the processing elements in the same relative positions as the detector elements, allowing implementation of different topologies. In a parallel-processing system, where results of a calculation of pattern recognition may be dependent on the data coming from the neighboring elements, the overall communication speed will obviously be determined by the longest cable. Thus it is important to keep cables short and approximately within the same length. (The FIFOs compensate for the small differences on cable length.) The 3D configuration permits this.

3.0 ARCHITECTURE OF THE 3D-FLOW PROCESSOR

3.1 3D-Flow Processor Cell Description.

The 3D-Flow processor (Figures 3(a) and 3(b) is a programmable, data stream pipelined device that allows fast data movements in six directions with digital signal-processing capability. The design of the processor has been completed, and 225 hours of consultancy from industry have been spent to check the feasibility of the 3D-Flow idea. A total of 6000 lines of VHDL code, describing the behavior of the single units and their interconnection, allows the user to simulate algorithms and to check the timing of all signals in the circuit (See Section 9.1.8.2). Use of table format in Microsoft Excel enables the user to set the input/output conditions at the external pins of the processor at each state. Other formats are used to download into the processor data-memory values, program memory, thresholds, and counter values.

The 3D-Flow operation is based on a data-driven principle. Program execution is controlled by the presence of the data at five ports (North, East, West, South, and Top) according to the instructions being executed. When an input (or output) instruction is issued and data are not present (or external FIFOs are full), then the 3D-Flow processor holds execution until data becomes available (or external FIFOs are not full.) (See Section 9.1.8.2, Figure 43 and note 2 for details.) A clock synchronizes the operation of the cells. (A prototype will be made at 60 MHz.) With the same hardware one can build low-cost, programmable, Level-1 triggers for a small and low-event-rate calorimeter, or high-performance, programmable Level-1 triggers for a large calorimeter capable of sustaining up to one event per clock.

At each input port of the 3D-Flow processor there is a FIFO that derandomizes the data from the calorimeter to the processor array. North, East, West, and South ports are 12-bit parallel bi-directional on separate lines for input and output, while the Top port is 12-bit parallel input only, and the Bottom port is 12-bit parallel output only. North, East, West, and South ports are used to exchange data between adjacent processors belonging to the same 3D-Flow array (stage).

Top and bottom ports are used, under program control, to route input data and output results between stages. Each 3D-Flow cell consists of a Multiply Accumulate unit (MAC); arithmetic logic units (ALUs); comparator units; encoder units; a register file; an interface to the Universal Asynchronous Receiver and Transmitter (UART), used to preload programs and to debug and monitor during their execution; data memories to be used also as a look-up table to linearize the compressed signal, to remove pedestals, and to apply calibration constants; and a program storage surrounded by a system of three-ring buses. At each clock, a three-ring bus system allows input data from a maximum of two ports and output to a maximum of five ports. During the same cycle, results from the internal units (ALUs, etc.) may be sent through the internal ring bus to a maximum of five ports. The architecture of the 3D-Flow Processor cell is shown in Figure 3(a), the input/output in Figure 3(b).

Several 3D-Flow processing elements, shown in Figure 3, can be assembled to build a parallel-processing system, as shown in Figure 4.

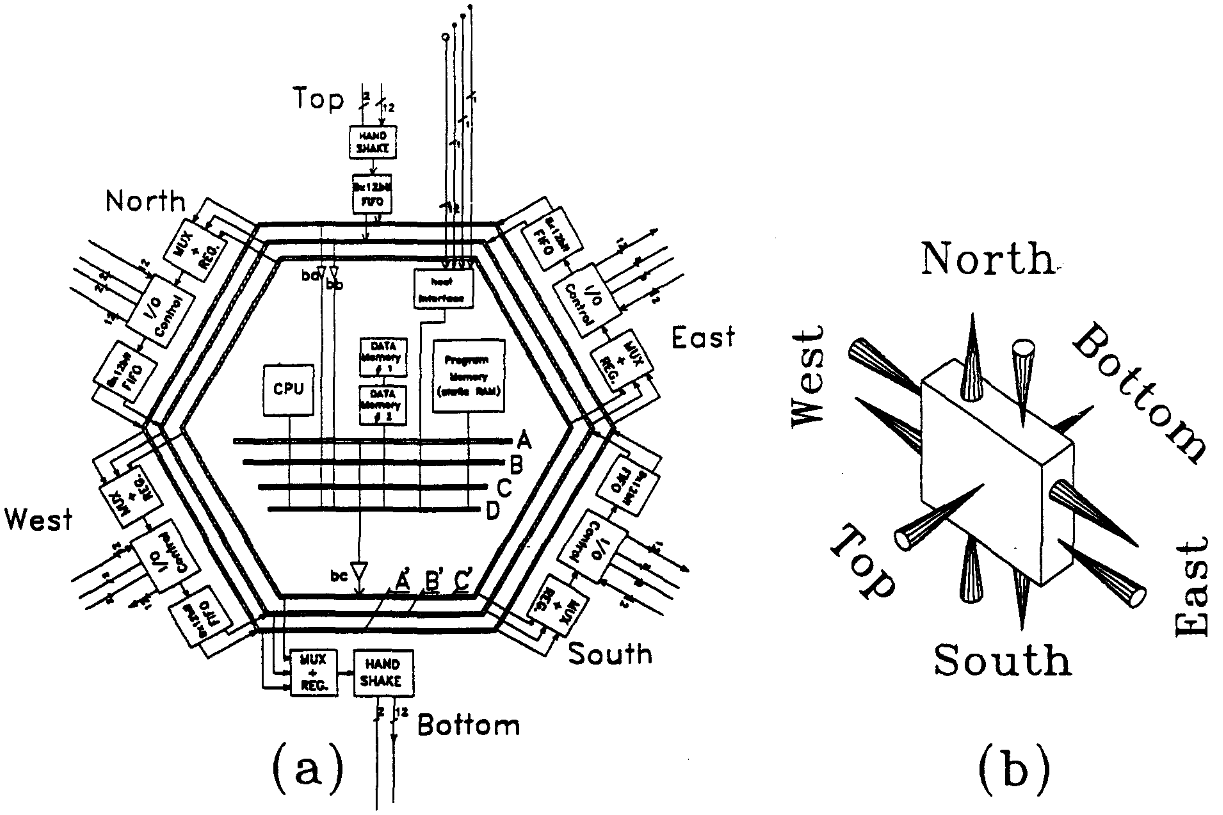


Figure 3. (a) 3D-Flow processor cell architecture. (b) 3D-Flow Input/output

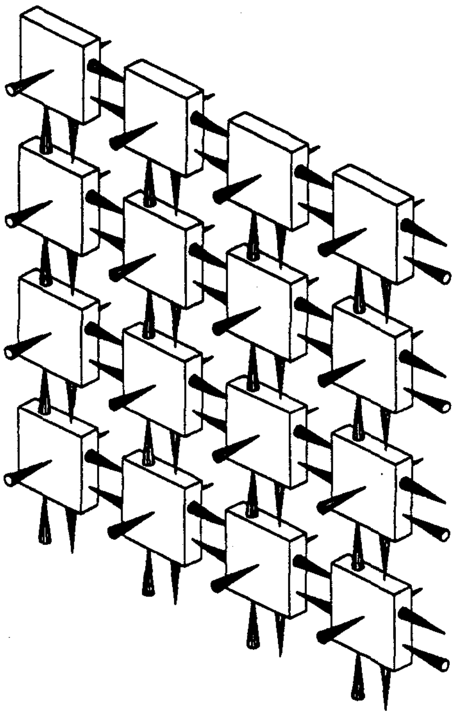


Figure 4. One stage (or layer) of 3D-Flow parallel processing system.

3.2 Short-Form User Manual of the 3D-Flow Processor.

From the 3D-Flow design schematic as simulated in VHDL. the following short-form user manual can be extracted. Examples of the use of the 3D-Flow instruction are described in detail in Section 9.1.8.2. A typical Level-1 trigger algorithm has been coded in the 3D-Flow instructions and is also simulated.

1. Registers seen by the User
   - 32 × 12-bit general registers (Rx)
   - 2 memory address Registers (Rx)
   - 3 arithmetic result registers (ACC1. ACC2. ACC3)
2. Buses
   - 4 internal buses (A. B. C. and D)
   - 3 ring buses (Ring A, Ring B, and Ring C)
   - 2 register file buses (AR and BR)
3. Communication links buffered with input FIFOs
   - one input link (Top)
   - one output link (Bottom)
   - 4 bidirectional links (North, East. West. and South)
4. functional units (operating in parallel)
   - one multiplier-accumulator (MAC)
   - two ALUs (ALU1 and ALU2)
   - one multihit encoder
   - one parallel comparator
   - two data memory spaces
5. instruction format (very long word)
   - operations of calculation and data movement
   - immediate fields can contain either Constants, Branch Addresses. Memory Addresses
   - all programming possible in symbolic expression

4.0 NEED FOR IMPROVED LEVEL-1 TRIGGER

Based on discussions with physicists involved in SDC, GEM, D0, CDF. and CERN detectors. the Level-1 trigger should be simple and should reduce the event rate by a factor of 100 or 1000 with simple logic (mainly discriminators). However, better efficiency in event rejection is desired. Physicists from a variety of experiments (SDC, GEM, CDF, D0, etc.) have demonstrated that by running different Monte Carlo simulations, generating plots by applying different thresholds. vetoing on the basis of hadronic energy content. checking for isolation, finding clusters, calculating cluster energy. counting particles. combining with muon and/or tracking information, etc., a substantial increase in efficiency is possible [ref. 8-22]. Until this new approach was conceived, hardware that implements different algorithms tested with Monte Carlo (at the most one algorithm has been implemented in current and past experiments) was not considered for the Level-1 trigger, because it was not available at an acceptable cost.

The flexibility of having a programmable Level-1 trigger offers the advantage of being able to experiment with different algorithms in the year 2003 that one may not even think of today. It can also check the efficiency, in a real-time environment, of the different algorithms tested with Monte Carlo simulation. By allowing selection of the best algorithm at a later time, it saves cost in the development of many different large boards [ref. 11] for different experiments through the alternative implementation of a single 12 cm × 12 cm board for the core of the parallel-processing system. Only the interface boards may change to connect (input/output) signals from different experiments.

5.0 ARCHITECTURAL DESCRIPTION OF THE 3D-FLOW SYSTEM

The 3D-Flow architecture is suitable for several applications, and it can be upgraded with advancements in technology. The main features of the system are its programmability, scalability, high-speed communication, and low cost. The 3D-Flow architecture makes possible the construction of a parallel-processing system with six-directional communication links between neighboring processors.

The assembly uses standard, commercially available components (except for the 3D-Flow chip), thus minimizing cost. It is suitable for the mapping of detector elements to processing elements, a solution that guarantees fast timing. Different detector element interconnection schemes can be efficiently implemented with the 3D-Flow parallel-processing system in one-dimensional, two-dimensional, and three-dimensional interconnection topologies by arranging the system in a planar, cylindrical, or spherical assembly, respectively. The interconnection length is kept to a minimum, and the interconnection topology ensures short cable length and, therefore, fast data movement (from 1 to 2.5 ns using BiCMOS drivers), compared to the greater delay variations that can exist in conventional systems. High speed and low power consumption are, therefore, achieved.

One of the most challenging problems that the high energy physics community has proposed for itself and its outside-technology supporters is that of useful data acquisition (DAQ) from beams crossing every 16 ns, as foreseen in the Superconducting Super Collider and the Large Hadron Collider. Thanks to advances in technology, the world of signal processing has been migrating from analog to digital methods, yielding improvements in programmability, stability, and uniformity, and raising the possibility of exploiting certain functions not possible in analog, such as adaptive filters used in the spread-spectrum techniques at the base of tomorrow's secure digital mobile communication systems.

A priori one would think that the useful DAQ problem cited above could not be solved by digital means since 16 ns is about the time taken for one instruction in today's leading workstations. These difficulties, known for many years, have stimulated extensive research and experimentation in parallel processing. There are even parallel processors available commercially, although programming them is much more difficult than programming a conventional sequential processor, and the success of a given programming effort is often strongly dependent on the parallel architecture employed. In fact the original advice to choose first the algorithm (or class of algorithms) before fixing the architecture is still at the basis of today's most successful parallel solutions.

5.1 Introducing the Third Dimension in the 3D-Flow Parallel-Processing System.

In the applications where the processor algorithm execution time is greater than the time interval between two consecutive data inputs, one stage (or layer) of 3D-Flow processor is not sufficent.The problem can be solved by introducing the third dimension in the 3D-Flow parallel-processing system, as shown in Figure 5.

In the pipelined 3D-Flow parallel-processing architecture, each processor executes an algorithm on a set of data from beginning to end (e.g., the event in high energy physics experiments, or the picture in graphic applications). Data distribution of the information sent by the calorimeter as well as the flow of results to the output are controlled by a sequence of instructions residing in the program memory of each processor.

Each 3D-Flow processor in the parallel-processing system can analyze its own set of data (a portion of an event or a portion of a picture), or it can forward its input to the next layer of processors without disturbing the internal execution of the algorithm on its set of data (and on its neighboring data set at North, East, West, and South that belongs to the same event or picture).

The programming of each 3D-Flow processor determines how processor resources (data moving and computing) are divided between the two tasks or how they are executed concurrently.

See Section 9.1.8 for a detailed example of programming the 3D-Flow (with simulation and timing analysis of the signals) and for executing the different tasks of computing and data moving.

A schematic view of the system is presented in Figure 5, where the input data from the external sensoring device are connected to the first stage of the 3D-Flow processor array. The program execution at stage 1 must not only route the new incoming data from the sensor to the next stage in the pipeline (stage 2), but must also execute its own algorithm. It then sends its results to the stage 2 processor array, which passes them on. At this point the stage 1 processor begins to re-execute its algorithm, receiving the new data from the sensor device and processing those values. The output results from all processors flow (like the input data) through the different processor stages. The last processor outputs the results from all processors at a rate up to 100 MHz. Several operations can be executed in one 3D-Flow instruction cycle. The main functions that can be accomplished by the 3D-Flow parallel-processing system are:

1. Operation of digital filtering on the incoming data related to a single channel;
2. Operation of pattern recognition to identify particles; and
3. Operations of data tagging, counting, adding, and moving data between processor cells to gather information from an area of processors into a single cell, thereby reducing the number of output lines to the next electronic stage. (See the simulation of algorithms that are moving data to an exit point in ref. 7.)

In calorimeter trigger applications, the 3D-Flow parallel-processing system can identify particles on the basis of a more or less complex pattern recognition algorithm and can reduce the input data rate and the number of input data channels.

In the real-time tracking applications, the system calculates tracks slopes, momentum, p_t, and the extrapolated coordinate of a hit in the next plane.

Figure 6 shows the timing (at the bunch crossing rate) of the input data to each stage (or layer) and the algorithm execution time (latency) in the 3D-Flow pipelined architecture.

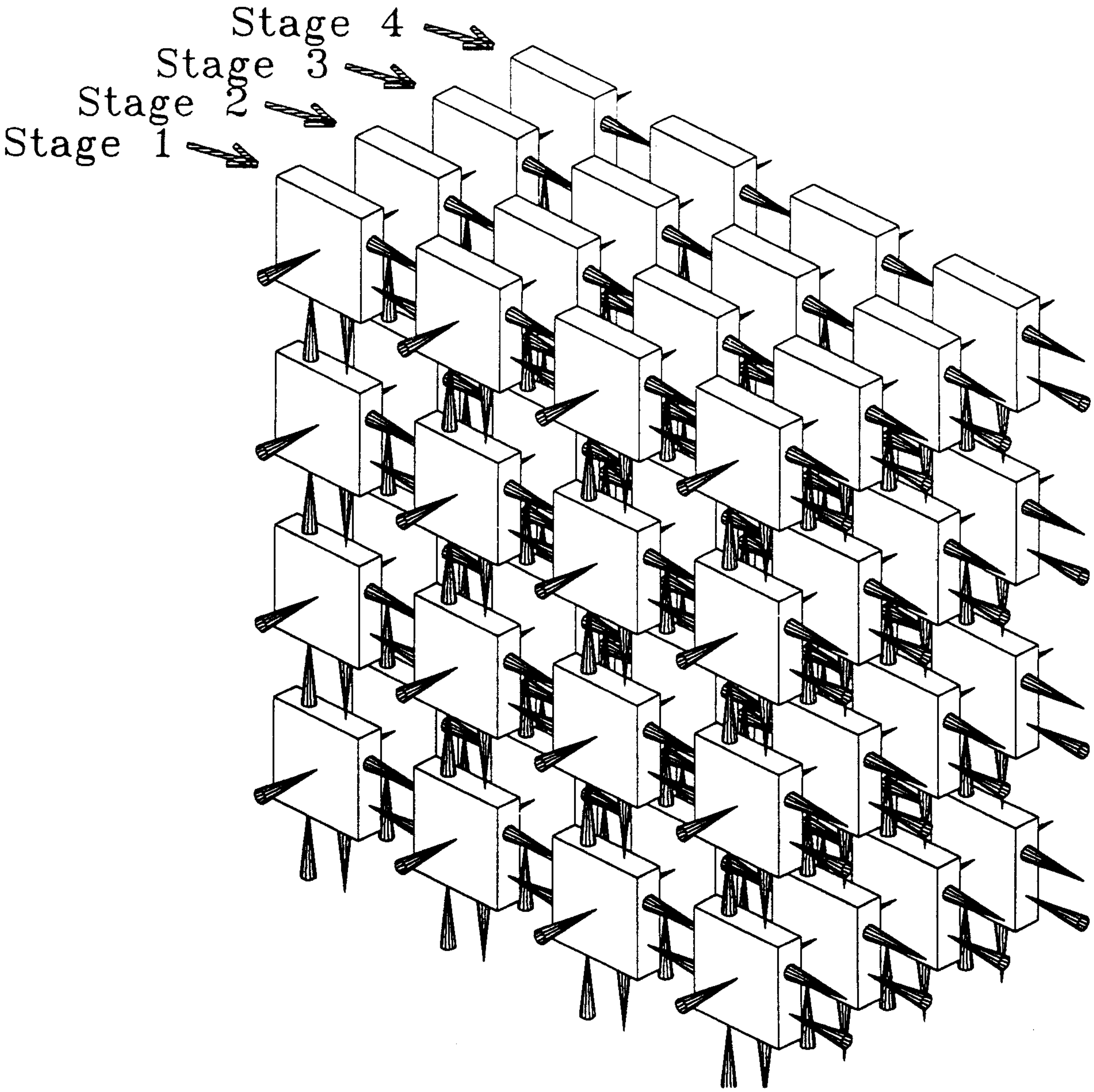


Figure 5. General scheme of the 3D-Flow pipeline parallel-processing architecture.

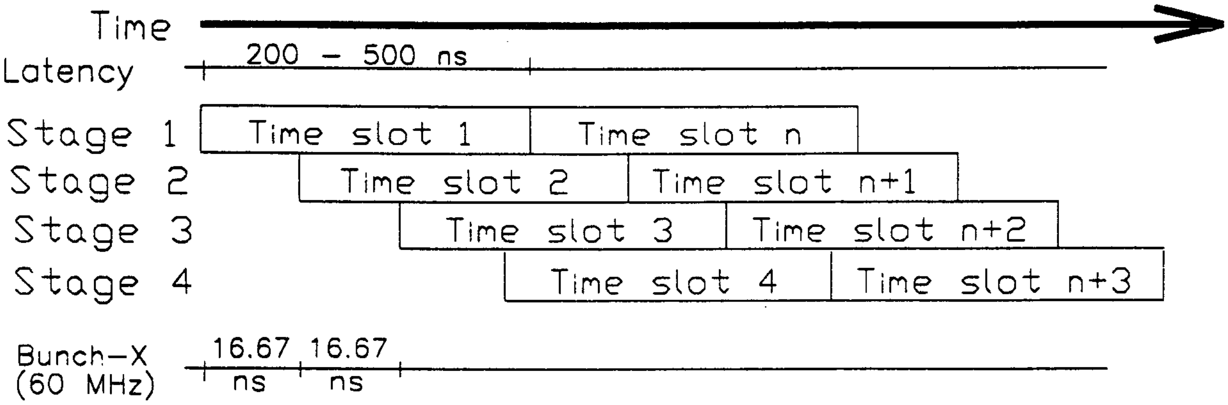


Figure 6. Timing diagram of four 3D-Flow pipelined stages.

5.2 Modularity and Scalability Options Using Standard Dimensions or Parts

The architecture can be built with racks of different sizes. Table 1 shows the dimensions of three systems with different sizes using standard. commercially available material: mini-size. VME-size. and large size. For convenience and because it is more applicable to the described applications. the drawings and descriptions in this report will refer to a system made of Mini-Racks. Analogous systems can be built in VME and large sizes. (1 U = 44.45 mm. 1 HP = 5.08 mm.)

Table 1. 3D-Flow assembly options using standard parts.

Rack name	height	width	depth (mm)	slot/ rack	Receiver board H × W × board thickness (mm)	Motherboard H × W × board thickness (mm)	Daughterboard H × W × board thickness (mm)
Mini-Rack (a)	3 U	24 HP	112.24	6	100 × 100 × 1.6	130 × 133 × 3.2	120 × 120 × 1.6
Mini-Rack (b)	3 U	24 HP	172.24	6	100 × 160 × 1.6	130 × 133 × 3.2	120 × 120 × 1.6
Medium-Rack (a)	6 U	42 HP	172.24	10	233.4 × 160 × 1.6	263.3 × 214 × 3.2	220 × 220 × 1.6
Medium-Rack (b)	6 U	42 HP	232.24	10	233.4 × 220 × 1.6	263.3 × 214× 3.2	220 × 220 × 1.6
Large-Rack (a)	9 U	63 HP	292.24	15	366.7 × 280 × 1.6	396.6 × 316 × 3.2	320 × 320 × 1.6
Large-Rack (b)	9 U	84 HP	412.24	21	366.7 × 400 × 1.6	396.6 × 423 × 3.2	380 × 380 × 1.6

Consider. for example. the overall requirements for the implementation of a typical Level-1 trigger algorithm obtained from Monte Carlo simulation. as described in detail in Section 9.1.8 (receive data from the calorimeter. convert compressed 8-bit data into linearized 12-bit value. calculate E_t. E_x. E_y. calculate front-to-back [Had/EM]. compare each of these calculated values with eight different thresholds) for a detector with 1280 trigger towers. running in an experiment with a 10-MHz bunch crossing rate.

As described in Section 9.1.8, the trigger algorithm will foresee the input of two compressed 8-bit data for each event (one from the hadronic compartment and one from the electromagnetic), and the total program execution length will be of 12 steps. Considering implementation of the first version of the 3D-Flow processor at 60 MHz, the algorithm execution time will require two layers of 3D-Flow processors. The technology has already surpassed 200 MHz (*e.g.*, the ALPHA processor from DEC, delivered at 100 MHz, but available soon at 250 MHz, and DSP from NEC), but a better price for this high-speed CMOS technology for the lower volumes needed in the case of the 3D-Flow processor will be available two or three years from now.

The overall system will then require 80 × Mini-Rack (a) as described in the first row of Table 1. with two 3D-Flow daughterboards in the back. as described in Section 5.7.

This configuration can easily grow in order to be able to implement future physics by accepting new threshold sets. implementing revised and optimized algorithms (*e.g.*. adding isolation. correlating calorimeter data with other detector information. *etc.*). and incorporating hardware advances with little effect on the installed system.

The high communication speed allows fast data exchange between neighboring elements. With the described system. one can easily reproduce the detector elements topology onto the processing elements interconnection topology. For the parallel-processing system described above. it is possible to keep very short the length of lines driven by the high-speed components, thus minimizing power consumption without sacrificing high-speed communications. In an overall parallel-processing system, both processor speed and communication speed must be considered for a fast algorithm execution that requires data interchange between processors. If data are exchanged between processors at the same time (but not necessarily synchronously because they are derandomized by the presence of the FIFOs at each input port), and if the condition for each processor to continue its algorithm is that it receive the expected data from the neighboring processor, then the time constraint for all algorithms to advance in the process will be determined by the longest connection (or longest cable).

In a conventional assembly, where racks are housed in conventional cabinets, one cannot avoid having long and short cables if implementation of different topologies is desired. As a consequence, in order to obtain the same performance as the present system, use of high-current drivers capable of driving longer distances is required: but longer cables are equivalent to longer delays, no matter how fast the driving circuit. The result is that more processor pipeline stages are required (at a higher hardware cost) to execute the same algorithm.

5.3 Standard Electronic Enclosures

The 3D-Flow system assembly can be built using standard electronic enclosures for microprocessor packaging systems that meet the following standards: CERN-Spec. No. 385, IEC 297-1, IEC 297-3, IEC 97.2, IEC 97.3, DIN 41494, and IEEE 1101, compatible with VME enclosures. Figure 7 shows the dimensions conforming to the preceding standards.

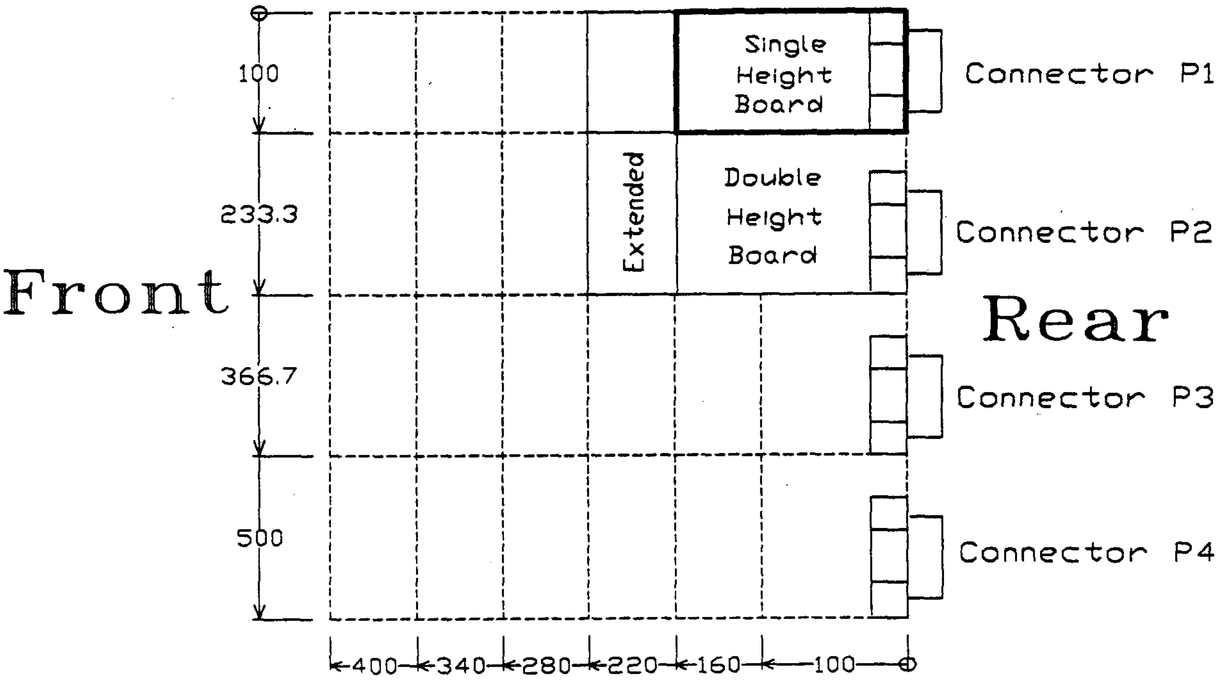


Figure 7. Typical Euroboard dimensions. Conformance to IEEE 1101, IEC 297-3, DIN 41494.

5.4 3D-Flow Assembly Details

All essential items are identified in Figure 8 by a circled number. Detailed descriptions are contained in various sections of this report according to the following scheme:

- Section 5.5, item number 2
- Section 5.6, item number 3
- Section 5.7, item number 1
- Section 5.8, item number 4
- Section 6.1, item numbers 5, 6
- Section 6.2, item numbers 7, 8
- Section 6.3, item numbers 9--18
- Section 6.4, item numbers 19--39
- Section 7.1, item number 40

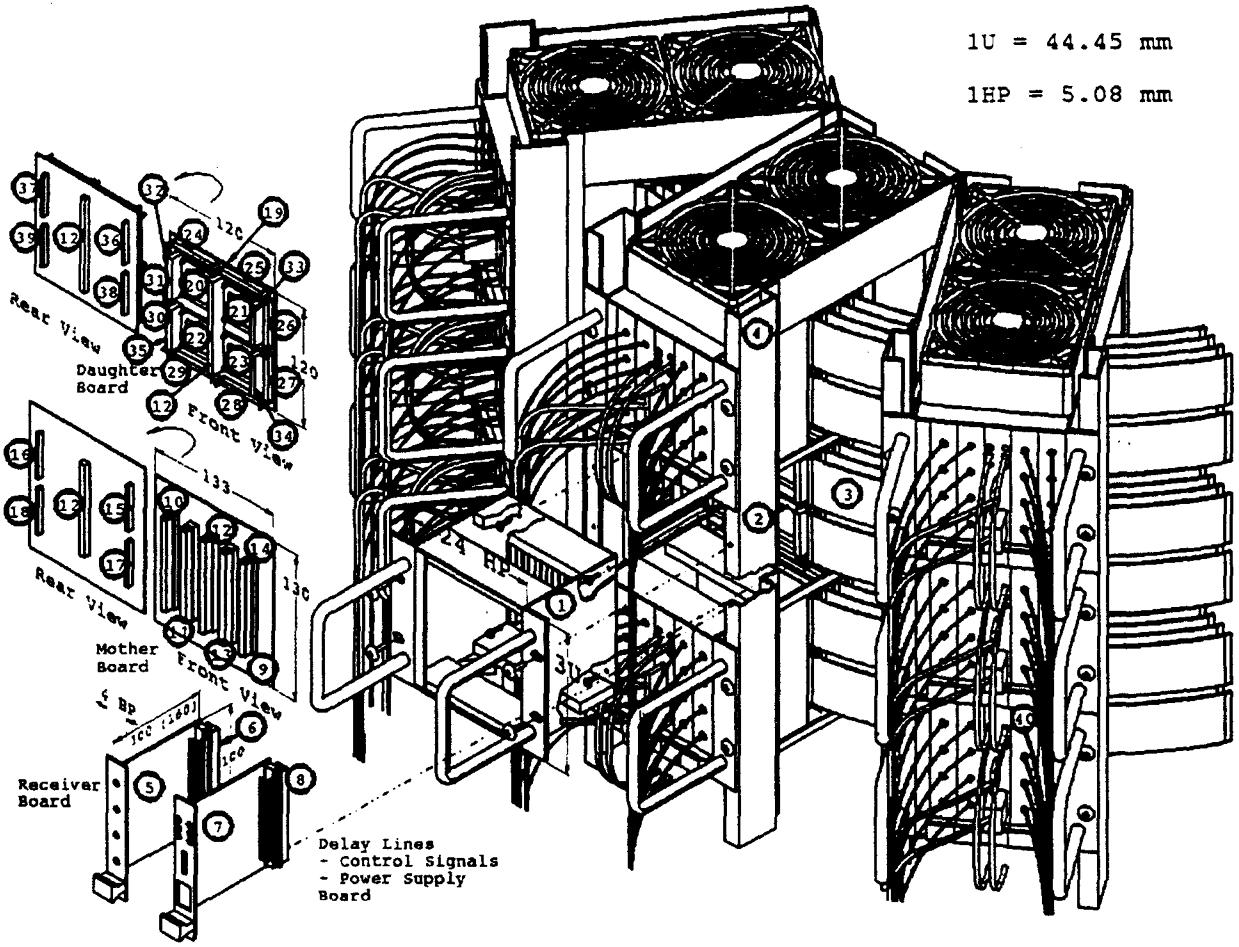


Figure 8. 3D-Flow system assembly details.

5.5 3D-Flow Daughterboards Stack Assembly

The basic element for the construction of a 3D-Flow parallel-processing system is a 3D-Flow daughterboard printed circuit (PCB) with 16 × 3D-Flow processors (4 chips in a 12 cm × 12 cm dimension). Figure 9 shows the detailed assembly of a stack of 3D-Flow daughterboards.

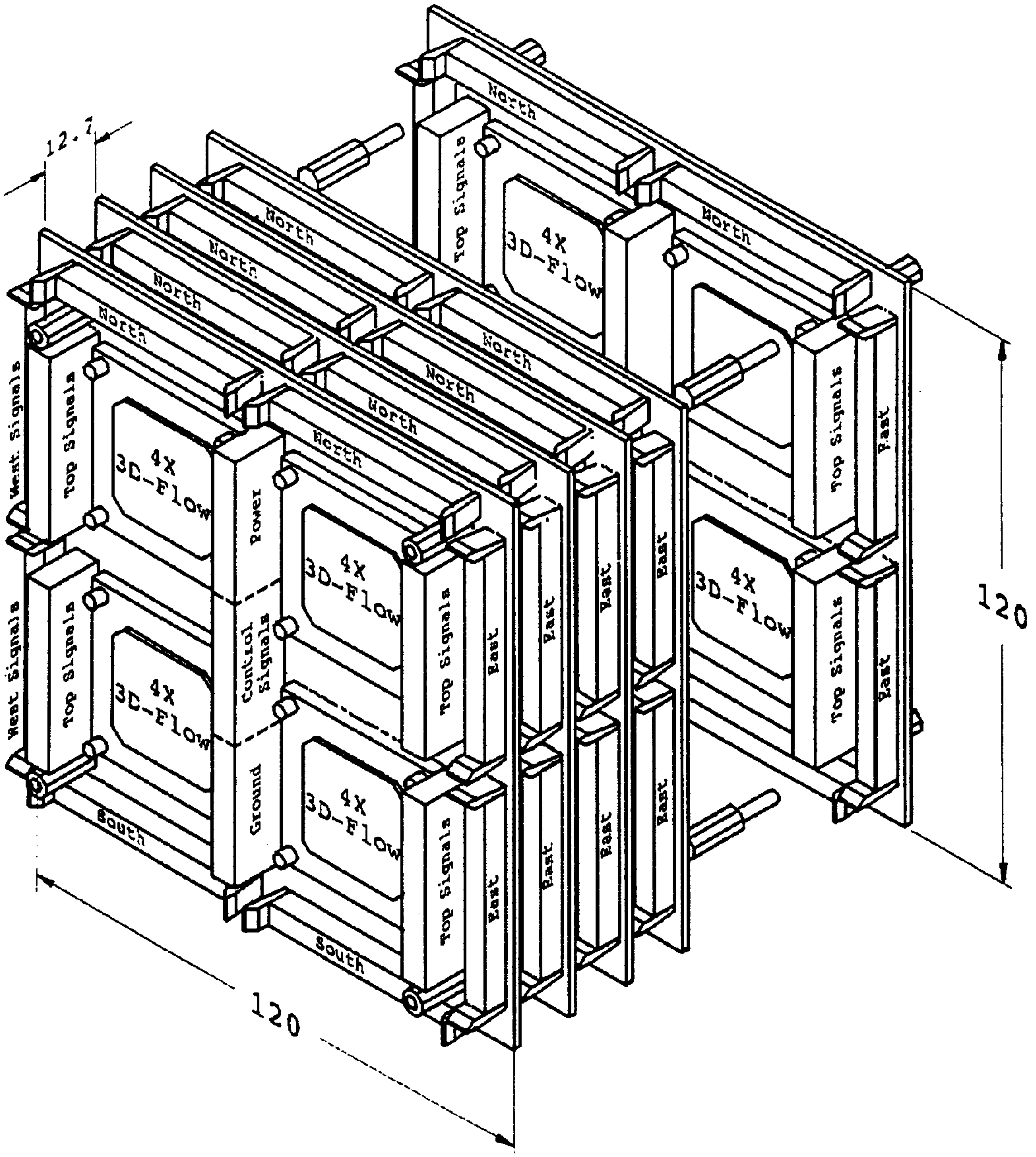


Figure 9. 3D-Flow daughterboard stack assembly.

5.6 Interconnection of the 3D-Flow Daughterboards

The connection between the daughterboards in the North. East. West. and South directions is made through a flexible printed circuit and 60-pin female connectors. The length of the connection is less than 12 cm. Figure 10(a) shows the layout of the connection between two connectors. The use of flexible printed circuit connections enhances the robustness of the part and allows for matching of different impedance.

The female connector at the edge of the flexible printed circuit (see Figure 10(b)) has a rigid stiffener 0.8 mm thick attached to the flexible printed circuit to increase robustness. The male connector on the printed circuit has a latch extractor that allows extraction of the female from the male connector without damaging the flexible printed circuit.

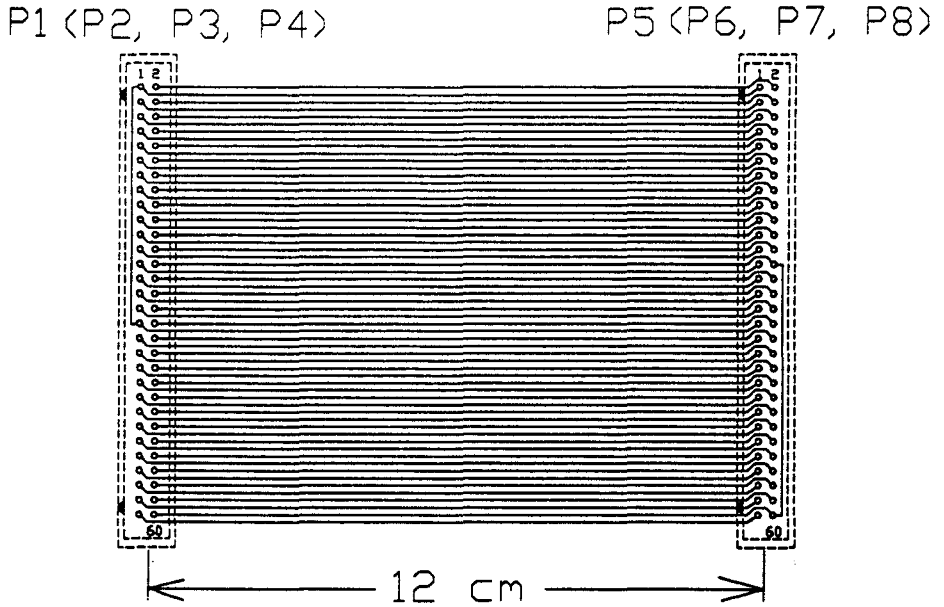


Figure 10(a). The 60-wire flexible printed circuit with female connectors used to connect the 3D-Flow daughterboard of one Mini-Rack to the daughterboard of the adjacent Mini-Rack at North, East, West, and South.

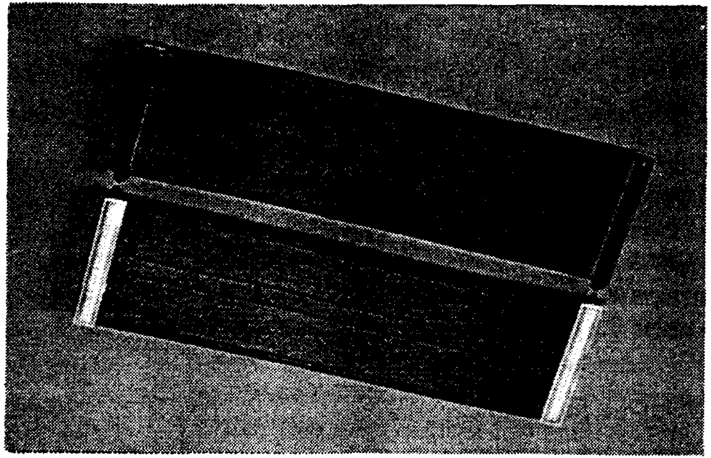


Figure 10(b). Prototype of the 60-wire (with 100 ohm impedence) flexible printed circuit.

5.7 3D-Flow Mini-Rack Assembly

The Mini-Rack assembly consists of two parts: the front end for data acquisition through external devices (through optical fiber or copper wire in serial or parallel form), and a second stack of boards, tightly joined to the defined Mini-Rack (see Table 1) assembly at a 90° angle. This is the parallel-processing system that, with its high degree of interconnection in six directions, can achieve high-speed communication.

The Mini-Rack (Figures 11(a) and 11(b)) is the basic modular unit. It receives input signals from the front panel of the modules as well as from the power supply and control signals cables. The data path is then moved from the front to the back of the Mini-Rack through the processing and high-speed interconnection capability of the 3D-Flow system. The result of the parallel-processing system is a reduced data set with respect to the input data; the data set is then routed to the exit point connectors.

Figure 11(a) shows a Mini-Rack of type (b) with 3U receiver boards of 100 x 160 mm and a stack of several daughterboards connected at the rear through a motherboard. It also shows the connection to adjacent daughterboards belonging to neighboring Mini-Racks. Connections between daughterboards are made through two flexible printed circuit cables (a total of 120 connections for each side).

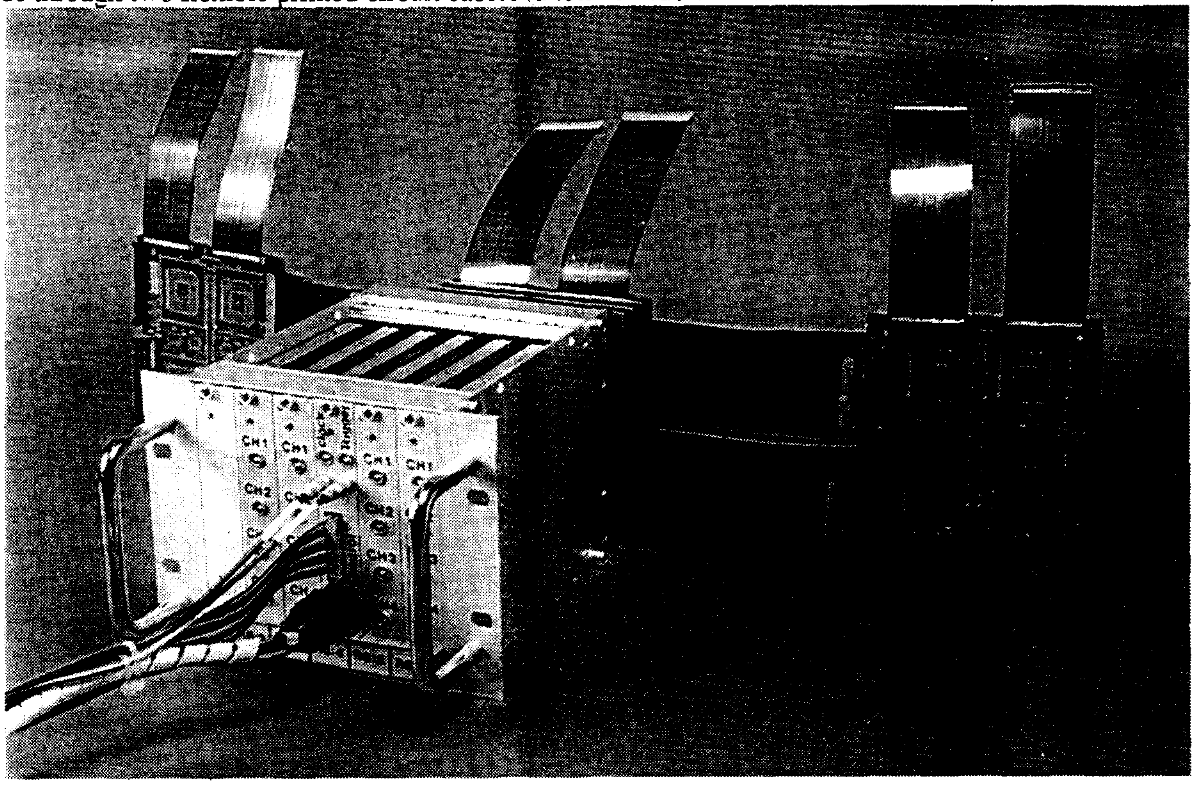


Figure 11(a). 3D-Flow Mini-Rack prototype with standard 3U × 160 mm DAQ boards

Figure 11(b) shows a Mini-Rack of the type (a) with 3U receiver boards of 100 x 100 mm and a stack of six daughter boards. Signals from the Bottom port of one daughterboard are transferred to the Top input port of the next daughterboard throught four 60-pin surface-mounted connectors. These connectors carry the 16 channels of input to the Mini-Rack (each with a 12-bit path), from the Top to the Bottom port of the 3D-Flow Processor stages. Thus on each daughterboard printed circuit, the signal is received from the Top input port, sent to the 3D-Flow processor, and routed through the Bottom output port to the surface-mounted female connector situated on the opposite side of the printed circuit, in the same position as the Top port male connector. In the central part of the daughterboard there is instead a through hole connector with 100-MIL spacing and with larger pins in order to carry high-current power supply and the control signals. (In this case the signals are the same for all daughterboards.)

A motherboard (see detailed description in Section 6.3), placed at the rear part of the Mini-Rack, is the interface between the data acquisition system (made of up to six 3U receivers and/or control modules) and the stack of the 3D-Flow parallel-processing system (see Figure 9). The relatively long handles in the front of the Mini-Rack allow one to support the optical-fiber links connected to the front panel of each module. Allowing the optical fiber to have a large curvature and to be mechanically attached to the handle helps to avoid mechanical stress to the connector joint point of the fiber. The copper wires of the control lines and the power supply cable are not very critical to the curvature radius.

A 3D-Flow system can be made of several types of rack, with the board dimensions shown in Table 1 and Figure 7. As an example, this report will represent only the detailed assembly and disassembly of a Mini-Rack (a) of 3U with a receiver board of 100 mm × 100 mm.

The assembly and disassembly of the Mini-Rack (a) are shown in Figures 12 and 13.

Access to any parts of the parallel-processing system is provided from the front and from the rear of the system. This makes it suitable to use different types of assembly for different interconnection topologies (e.g., planar, cylindrical, spherical, cubical, etc.), preserving easy access to all parts for maintenance. The communication paths between processors are kept to a minimum, and incremental upgrading with little effect on the installed system is easy to implement.

The two parts of the data acquisition and stack of the 3D-Flow parallel-processing system can be easily separated by removing four screws, as indicated in Figure 12.

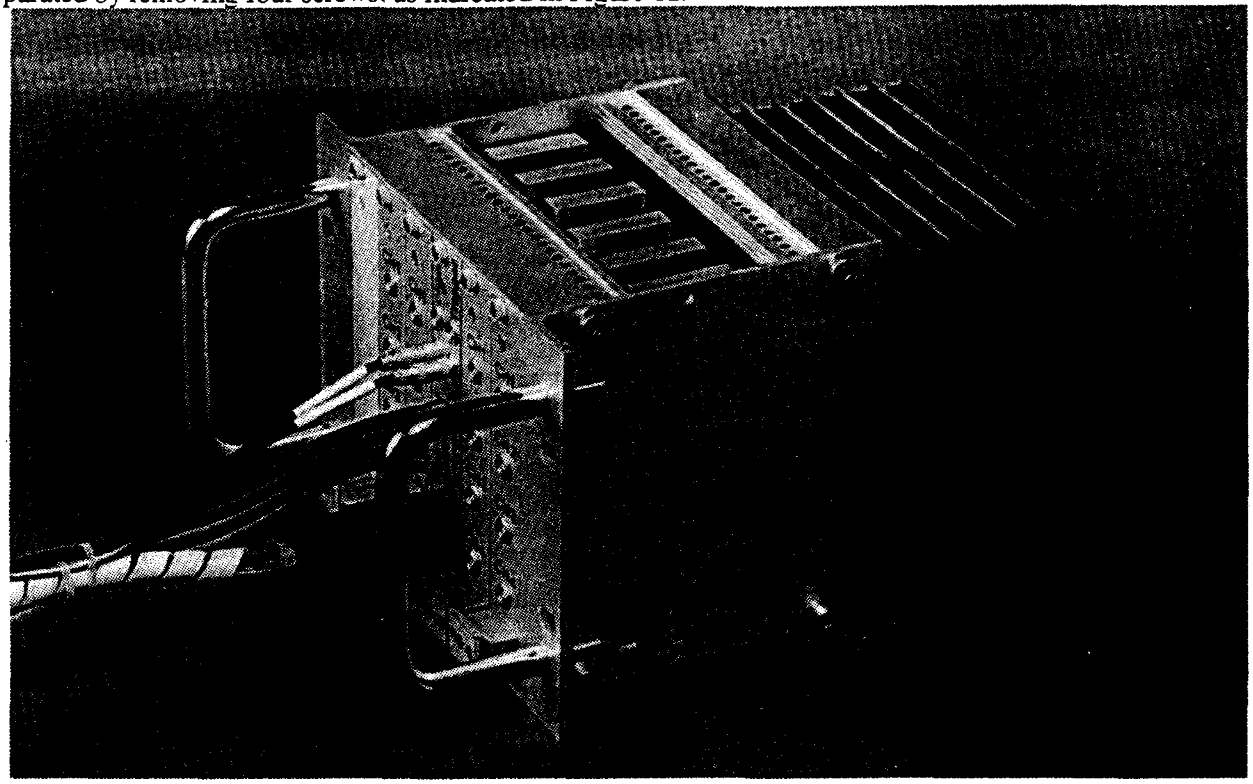


Figure 11(b). 3D-Flow Mini-Rack prototype with standard 3U × 100 mm DAQ boards.

5.8 3D-Flow Mini-Rack Tower Assembly

The 3D-Flow Mini-Rack tower (shown in Figure 14) is the cabinet support for several Mini-Racks assembled one on top of the other.

If a one-dimensional interconnection topology of the 3D-Flow parallel-processing system has to be built, then each connector on the four sides of the adjacent 3D-Flow has to be connected to the 3D-Flow daughterboard on the same Mini-Rack stack of boards. All these interconnections in one dimension provide a linear array of a group of 16 processors with very high bandwidth communication speed. The Mini-Rack towers can be placed one adjacent to the other, without interconnections and with side covers for air flow.

In the event that a two-dimensional or three-dimensional interconnection topology of 3D-Flow has to be built, then one Mini-Rack tower has to be placed on the side of others, and all Mini-Racks have to be connected horizontally and vertically.

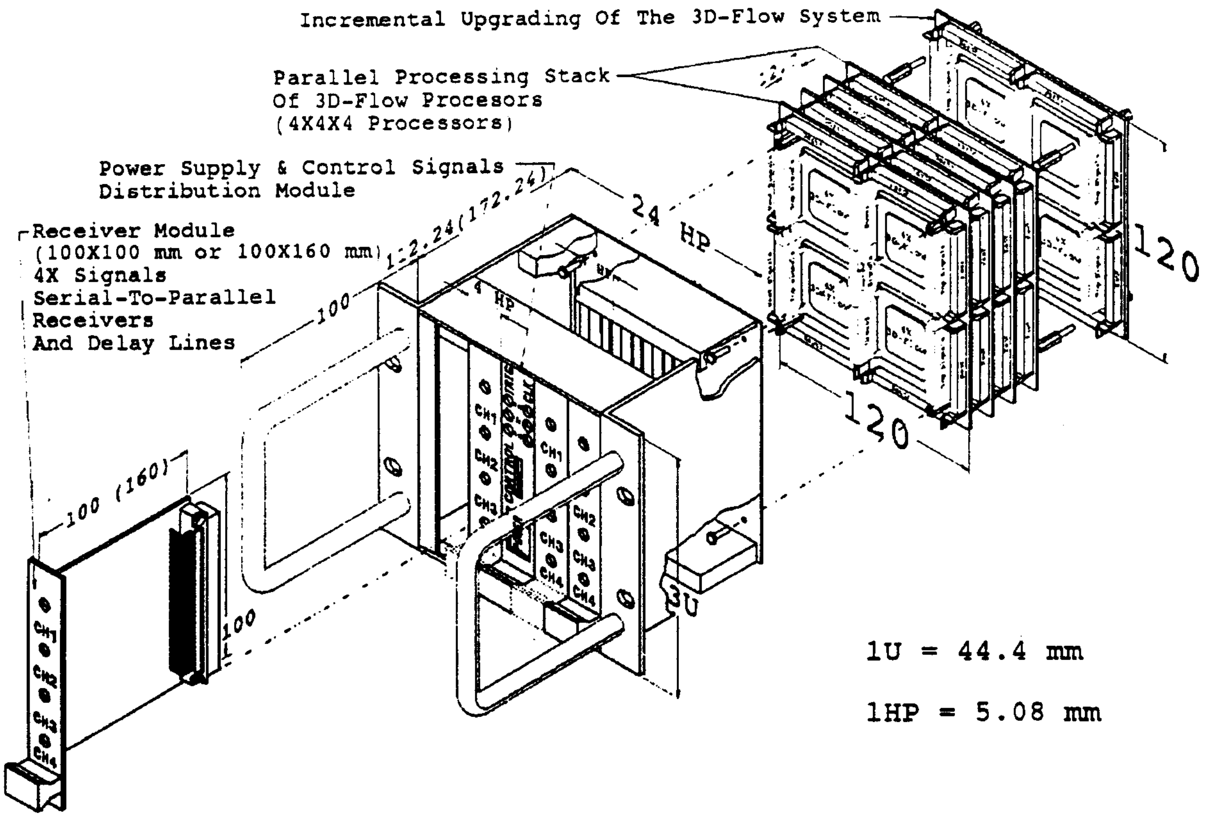

Incremental Upgrading Of The 3D-Flow System
Parallel Processing Stack Of 3D-Flow Procesors (4X4X4 Processors)
Power Supply & Control Signals Distribution Module
Receiver Module (100X100 mm or 100X160 mm) 4X Signals Serial-To-Parallel Receivers And Delay Lines
1:2.24(172.24)
24 HP
4 HP
100
120
120
3U
100 (160)
100
CH1
CH2
CH3
CH4
1U = 44.4 mm
1HP = 5.08 mm

Figure 12. How to assemble, disassemble, and add 3D-Flow processor boards to a Mini-Rack (with dimensions).

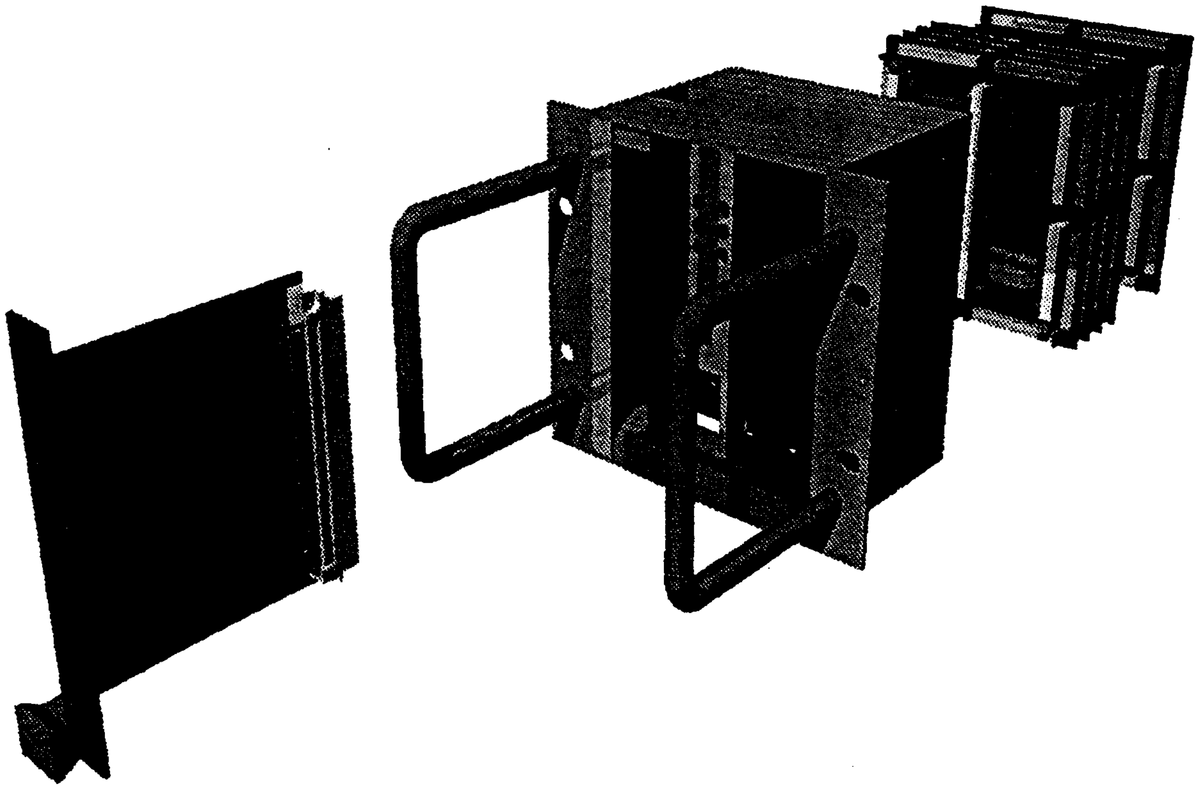


Figure 13. How to assemble, disassemble, and add 3D-Flow processor boards to a Mini-Rack (camera view).

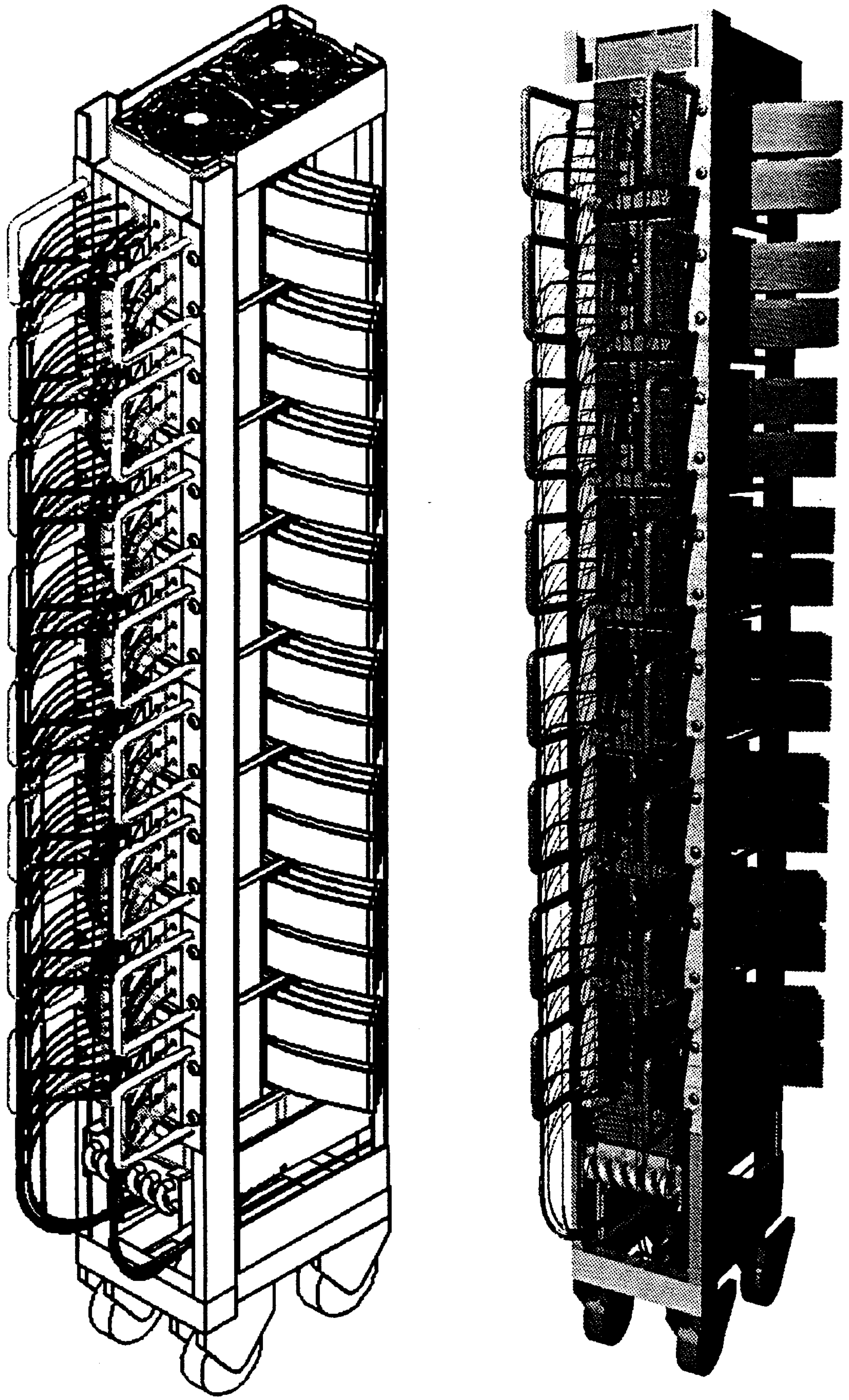


Figure 14. 3D-Flow Mini-Rack Tower.

5.9 3D-Flow System in a Planar Assembly

Several Mini-Rack tower assemblies joined together side-by-side form a planar 3D-Flow parallel-processing system assembly (Figure 15) This allows one to efficiently build two-dimensional and three-dimensional interconnection topologies with four boundaries--one at North, one at South, one at West, and one at East. The processors at the above-mentioned border should not be expected to receive or to send data.

This assembly arrangement allows the user to keep the cable length to a minimum and to maintain approximately equal cable length (and, therefore, equal timing). The planar assembly is particularly suitable where it is necessary to have a continuity in data analysis and representation from input devices or output display in the form of a plane (e.g., in a series of planes in a tracking detector, in an end cup plane of a calorimeter or a Time Projection Chamber detector, or in an image plane of a graphic image).

The overall structure in a planar assembly has the shape of an arc with a large radius because the front panel of the Mini-Rack (data acquisition part) has a wider dimension with respect to the rear part (3D-Flow parallel-processing stack). Nevertheless, it allows easy access from both parts, front for access to the data acquisition system and rear for 3D-Flow stack maintenance and incremental upgrading.

In the planar assembly, at each Mini-Rack eight optical fibers are firmly attached to the left handle and eight to the right handle. The power supply and control line cables (in copper) also are attached to the handles and go down along the Mini-Racks.

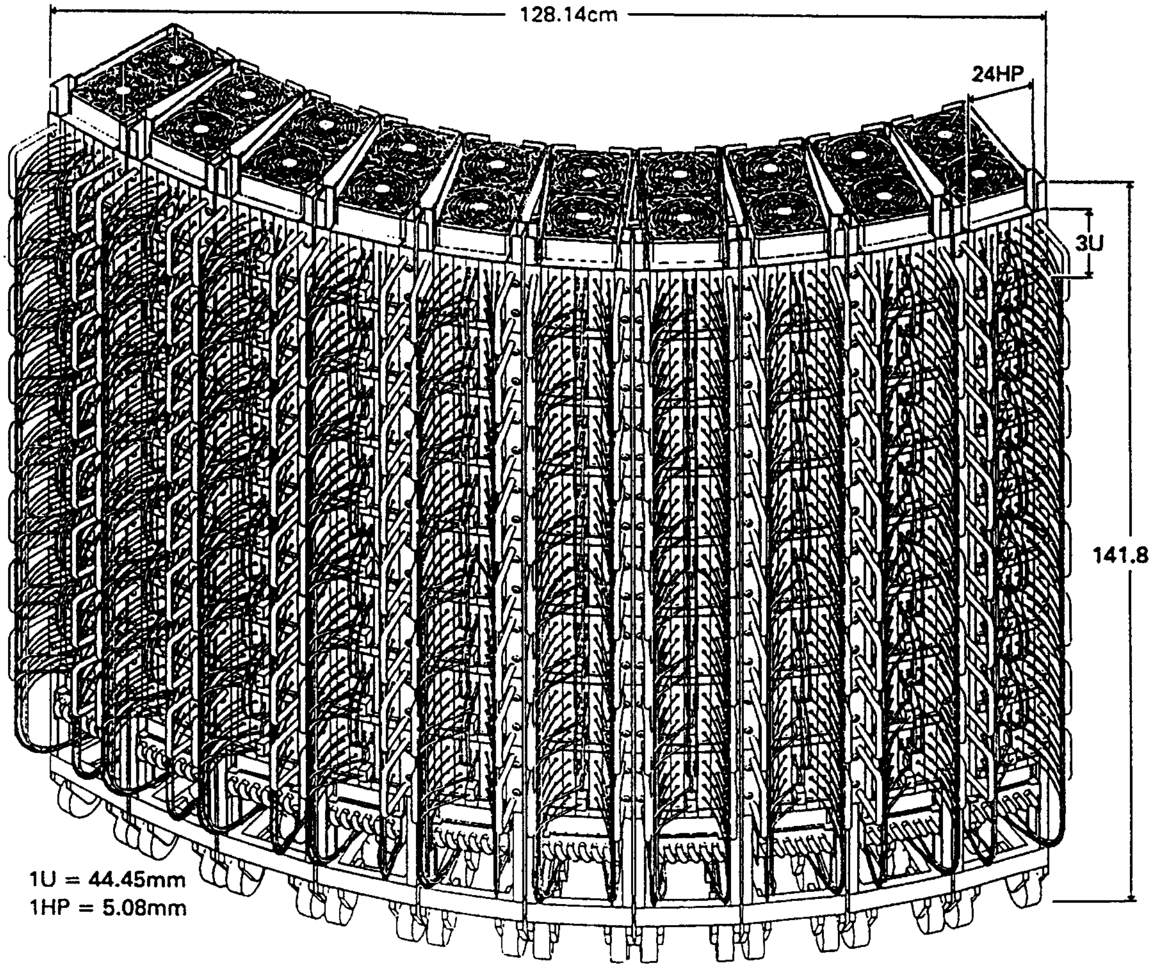


Figure 15. 3D-Flow system in a planar assembly (with dimensions).

5.10 3D-Flow System in a Cylindrical Assembly

The 3D-Flow system shown in Figure 16 represents the actual size of a complete Level-1 calorimeter trigger for 1280 trigger towers. the number for trigger towers of experiments such as D0 or GEM. There is a total of 80 × Mini-Racks arranged in a cylindrical form in order to map one detector element (trigger tower) to one processor element (3D-Flow cell) with the guarantee of having fast timing between neighboring elements. The difference among the two systems (D0 and GEM) will only be the number of 3D-Flow daughterboards stacked one on top of the other in order to implement algorithms with different complexity at a high or low data rate. In the case of the D0 experiment. only two 3D-Flow daughterboards will be sufficient to implement the trigger algorithm described in Section 9.1.8. For the GEM experiment. which will run on an accelerator with a bunch-crossing rate about 8 times higher then the one for D0. the number of 3D-Flow daughterboards will depend upon the speed of the 3D-Flow processor at that time (The ALPHA CMOS processor from DEC. for example was delivered at 100 MHz. but will soon be delivered at 250 MHz) and will depend upon the increased complexity of the algorithm for GEM (with or without digital filter).

The advantage in implementing a 3D-flow parallel processing system in a cylindrical assembly (Figure 16) is that it allows the user to efficiently build two-dimensional and three-dimensional interconnection topologies with only two boundaries. one at the top and one at the bottom. The processors at the top border should not expect to receive or to send data from the North port. while the processors at the bottom border of the cylinder should not expect to send or receive data from the South port. This assembly arrangement allows the user to keep the cable length to a minimum and to keep approximately equal cable length and. therefore. equal timing. The cylindrical assembly is particularly suitable where it is necessary to have a continuity in data analysis and representation from input devices or output display of the form of a barrel (e.g.. in the barrel of a calorimeter detector. axial tomography. *etc.*).

In the cylindrical assembly. at each Mini-Rack 8 optical fibers are firmly attached to the left handle and 8 optical fibers are attached to the right handle. Down along all the Mini-Racks the fibers pass in between the lowest Mini-Rack and the bottom support with wheels. The power supply and control lines cables (in copper) are also attached to the handles and go down with the optical fibers to the center of the cylinder and the double floor (one lower floor for the cables and an upper floor for the 3D-Flow system).

Maintenance of a single 3D-Flow board is made by disconnecting the cables and the optical fiber of the Mini-Rack unit under investigation. unplugging the four receiver modules. and removing the four screws from the front part of the Mini-Rack and the four screws from the bottom part of the Mini-Rack (at the motherboard). Thus the complete Mini-Rack data acquisition part can be extracted and the daughterboards with the 3D-Flow processors can be taken out one at a time.

In order to increment the parallel-processing system performance with more daughterboards. a complete column of flexible printed circuit cables from West to East must be removed. At that point the cylinder assembly can be opened to become a planar assembly. and access to the back for incrementing the system can be gained.

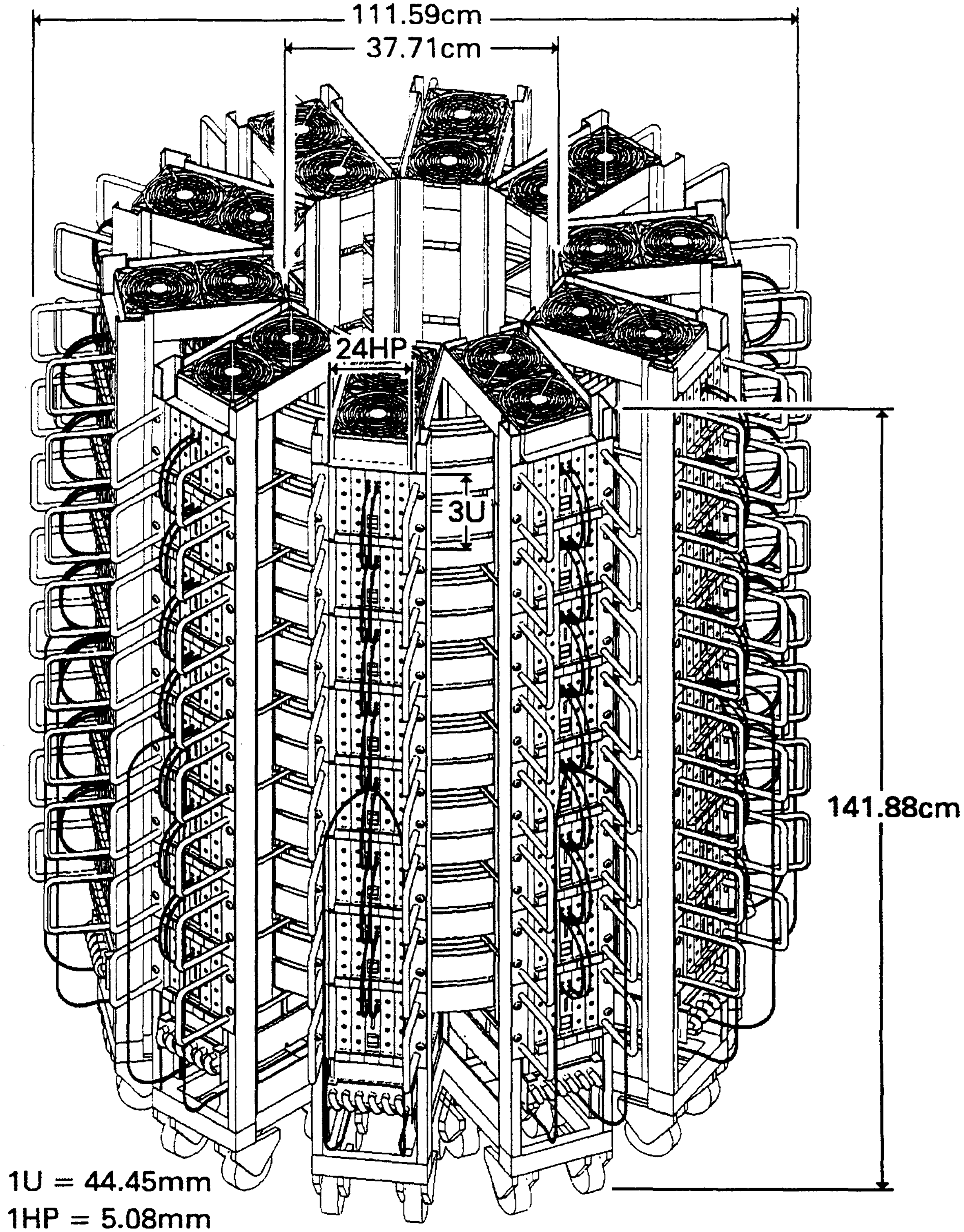


Figure 16. 3D-Flow system in a cylindrical assembly (with dimensions).

5.11 3D-Flow System in a Spherical Assembly

Figures 17--19 show an interconnection scheme of 3D-Flow processors, each with a neighbor at either side. In this case, 96 Mini-Racks, for a total of 1536 3D-Flow processors per layer (or stage), is contemplated, but other sizes of processors can be implemented.

The spherical arrangement of the system offers the advantage of building a 3D-Flow parallel-processing system without any boundary. Each processor of the system will always find a neighboring processor attached to any of the four ports (North, East, West, and South). One can thus load the same (or different) programs in all processors, all with the ability to receive or send data from or to all neighbors.

This assembly arrangement offers the minimum cable length between processors, thus allowing one to achieve very high performances. The spherical assembly is particularly suitable in applications where it is necessary to have a continuity among adjacent elements in the four directions (North, East, West, and South).

The input cables (or optical fibers), as well as the power supply and control signals to the data acquisition system, are distributed along the handles. The orientation of the Mini-Rack is dictated by the most convenient direction for reading the labels on the front panel of the modules in the Mini-Rack data acquisition system and for easy assembly of the cables along the handles from the top to the bottom of the system. There is no constraint in orientation of the 3D-Flow processor board, because assignment of the name to the 3D-Flow processors may follow any convention. Thus during the initial loading phase of the programs into the individual 3D-Flow processors, a conversion table can be applied to assign a logical name to each 3D-Flow processor that may differ from the physical port-name in the data-sheet of the chip.

The output cables are attached to a patch panel that is receiving the signals from the last 3D-Flow daughterboard (last with respect to the data acquisition receiver modules and the 3D-Flow motherboard) and are distributed inside the sphere, down to the bottom.

Maintenance of the 3D-Flow parallel-processing stack is similar to that described in Section 5.10 for cylindrical assembly. Maintenance and assembly of the entire system are carried out by unplugging a series of connectors between Mini-Racks along a circumference, as shown in Figure 19.

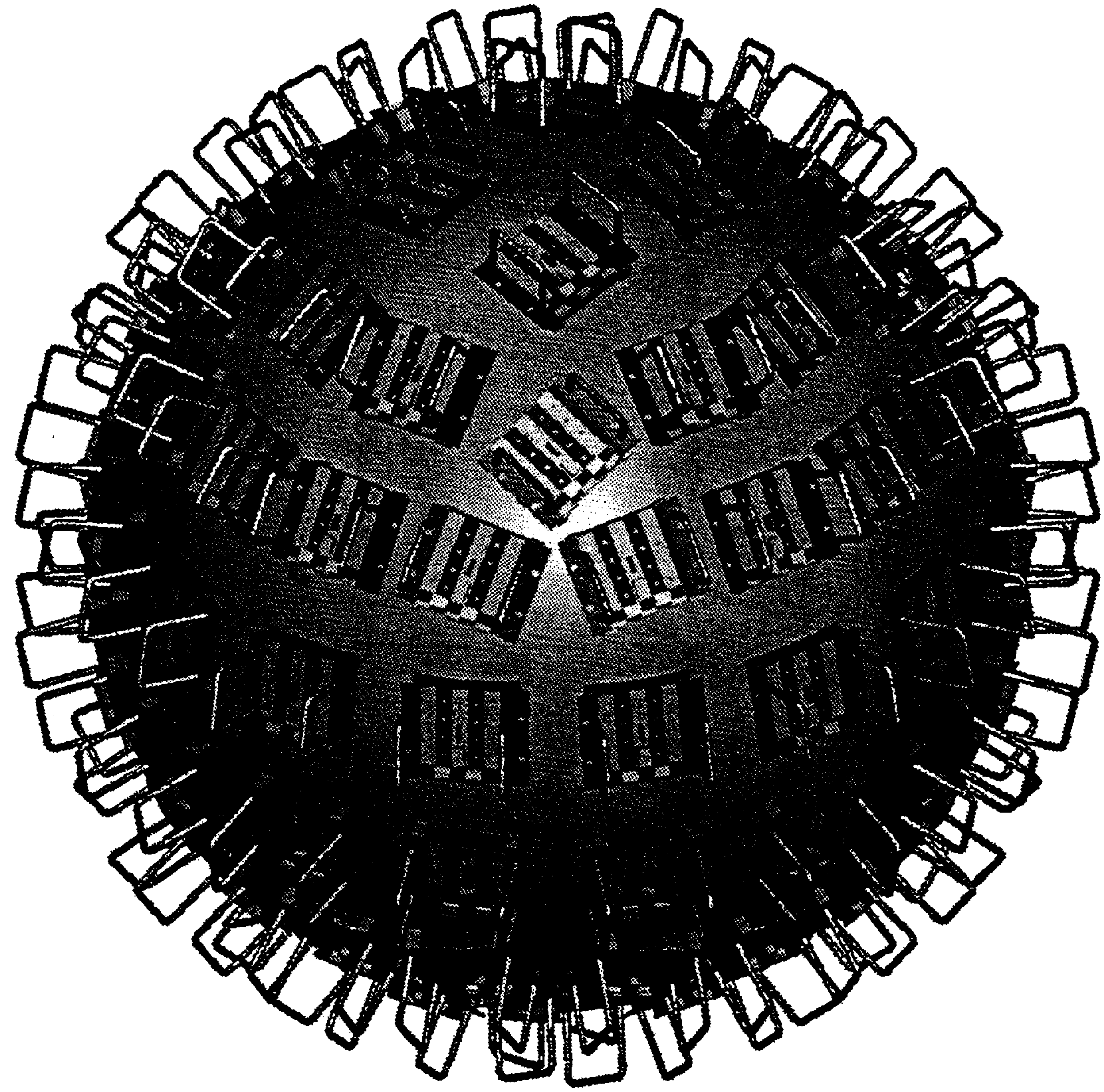


Figure 17. 3D-Flow system in a spherical assembly (detail without support).

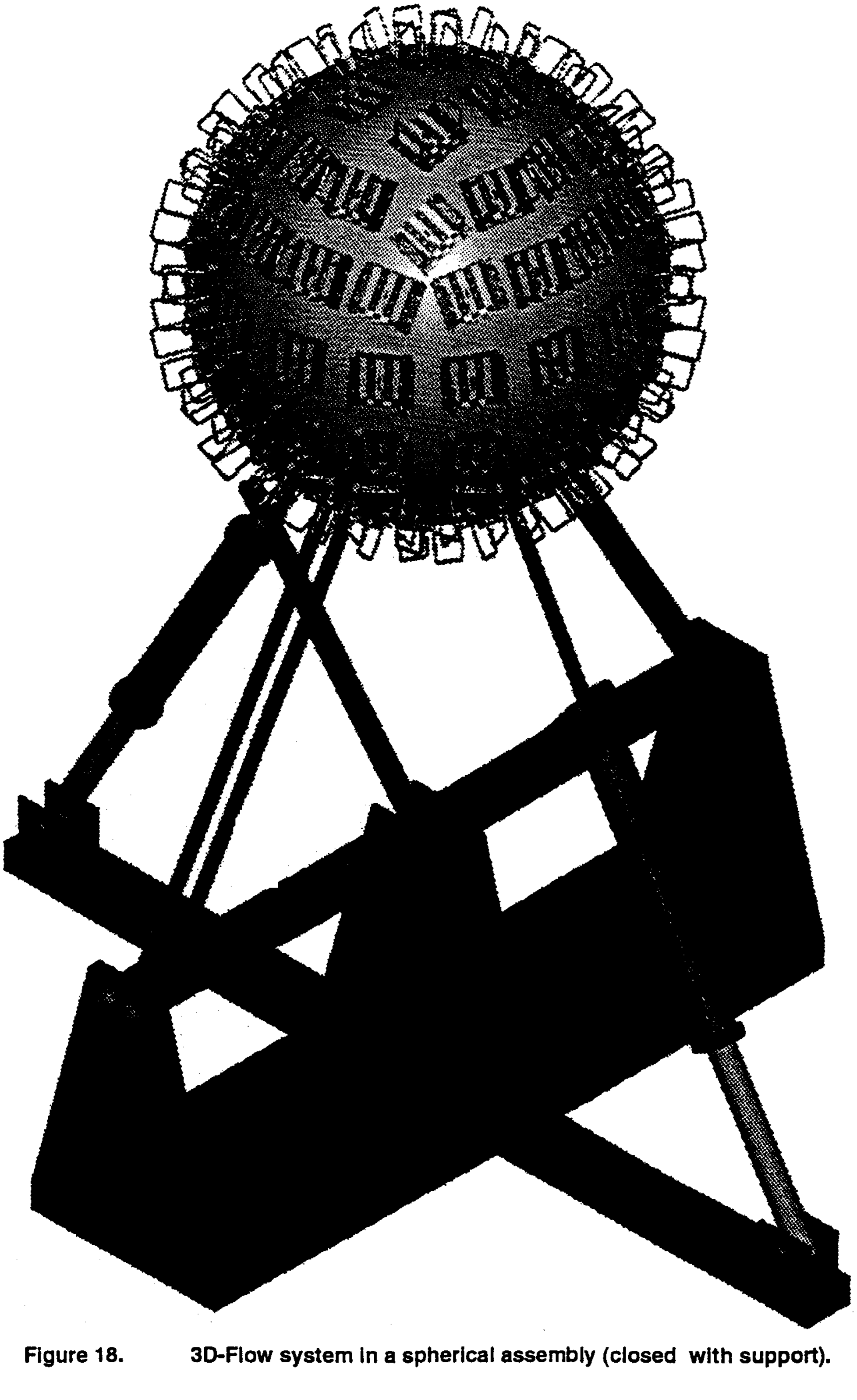


Figure 18. 3D-Flow system in a spherical assembly (closed with support).

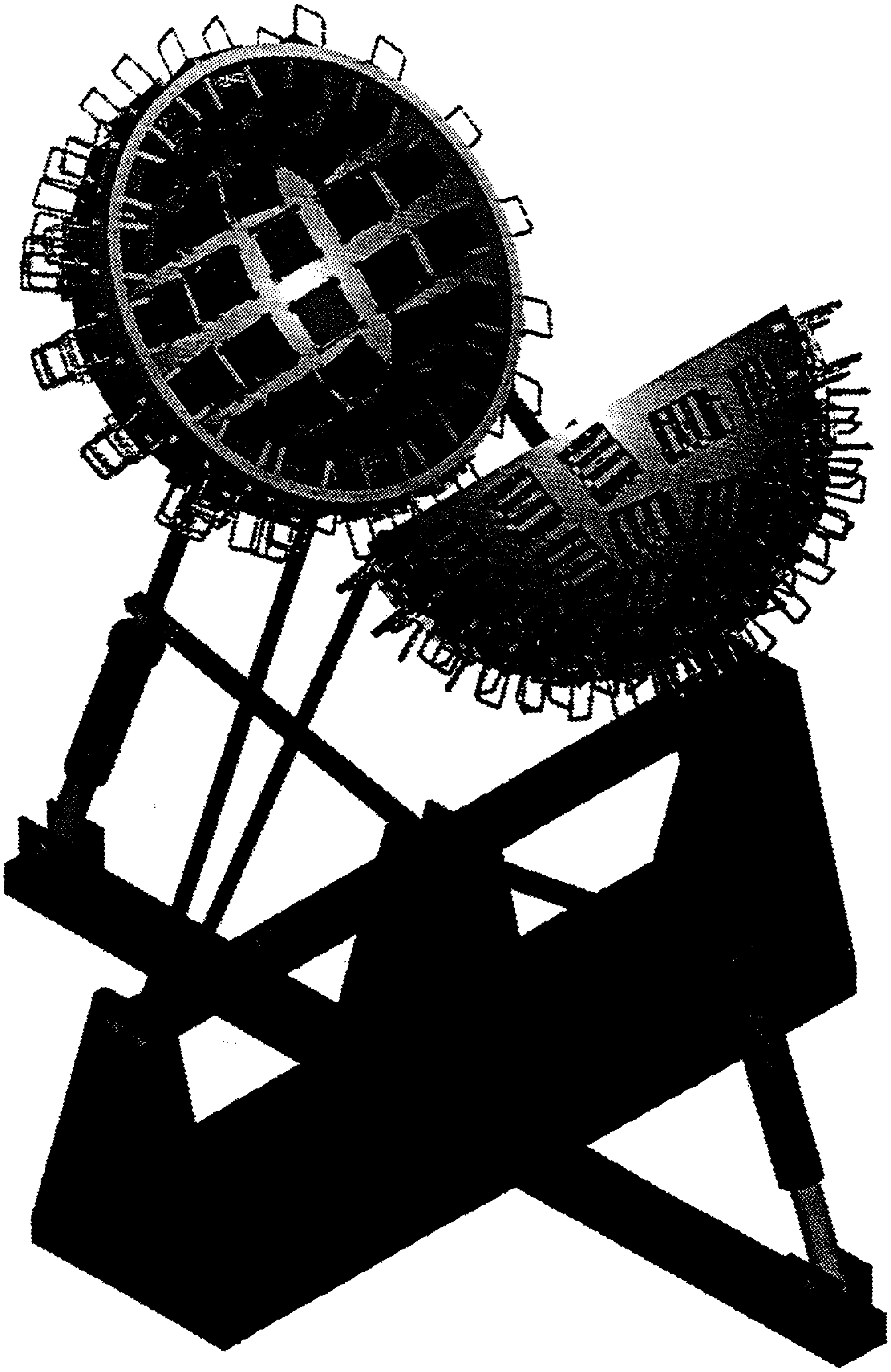


Figure 19. 3D-Flow system in a spherical assembly (open during maintenance).

6.0 DESCRIPTION OF THE BOARDS AND THE 3D-FLOW CHIP

6.1 Receiver Board

The receiver board (Figure 20) is a standard board 3U in height and 100 mm (or 160 mm) in depth. This board can accommodate the electronics that interface the device (sensors) of a particular application to the 3D-Flow parallel-processing system. The input to this board can be analog, serial, or parallel in digital form. The rear connector provides the signal adapted to the input to the first layer of the 3D-Flow parallel-processing unit.

6.2 Control and Power Supply Signals Board

The control and power supply module (Figure 21) consist of a 3 U board 100 mm × 100 mm (or 100 mm × 160 mm) with front panel connectors for power supply, control lines, trigger, and clock and a 64-position rear connector that distributes power supply and control signals. These signals of the 64-pin rear connector are carried from board to board (same signal) through the stack of the 3D-Flow daughterboards.

In order to distribute the clock and trigger signals with equal timing to the different 3D-Flow processors located in different geographical locations, two programmable delay lines are applied. The delayed output is then sent to the stack of 3D-Flow daughterboards of that Mini-Rack. Section 7.1 describes how these signals are distributed over the entire 3D-Flow parallel-processing system and how they are programmed at each Mini-Rack control module. The power supply is received at the front panel with 6 pins (3 for +5V and 3 for ground), each carrying up to 13 A.

The control signals connector carries:

1. Signals to the Universal Asynchronous Receiver and Transmitters (UARTs) of each 3D-Flow processor chip (four transmit Tx, four receive Rx, respectively, to each 3D-Flow stack of processors U1, U2, U3, U4, as shown in Figure 9, and the UART clock).

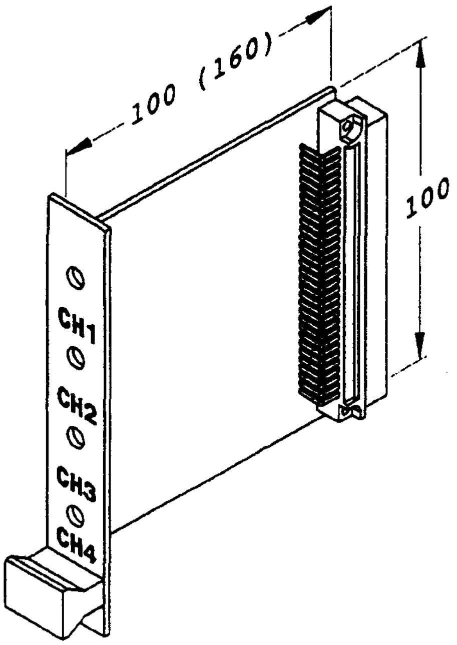


Figure 20. Receiver board.

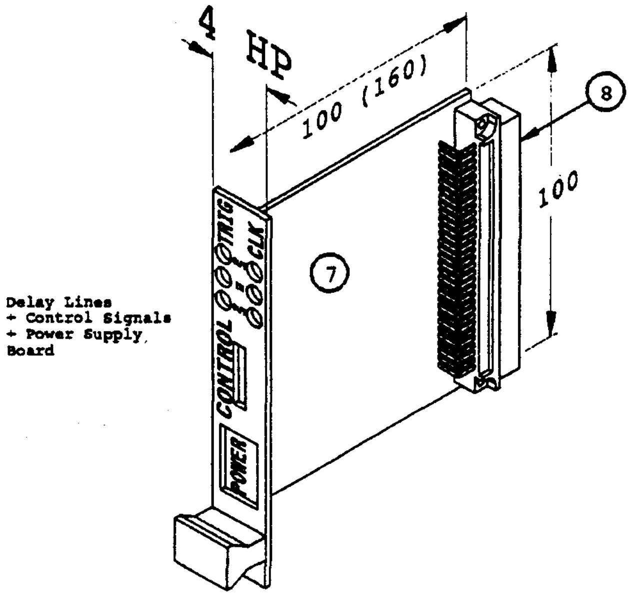


Figure 21 Control lines and power supply board.

2. Reset.
3. On two separate input 50-ohm Lemo connectors the signals, 3D-Flow clock, and trigger (or watchdog, or external device data-strobe) at NIM logic level are received.
4. Four JTAG signals for troubleshooting the system.
5. Control signals for the programmable delay-lines: CLKDLY = clock signal to initialize the delay lines; TINDLY and CINDLY = serial input of delay for the trigger and clock signals respectively; LOADLY = load of programmable delay lines. Figure 22 show the circuit for the programmable delay lines implemented in the control lines and power supply board. In the prototype used for the test a programmable delay line DS1020 from Dallas Semiconductor was used. This device operates at frequencies of 10 MHz (tests have been performed up to 60 MHz with a limited range of programmable delays). For higher frequencies of operation the ECL device MC10E195 from Motorola will provide a 2 ns delay range with 20 ps/delay step resolution at frequencies of > 1 GHz. This device allow cascading multiple components for increased programmable range. Figure 23 shows the timing of the circuit of Figure 22 for the DS1020 device.

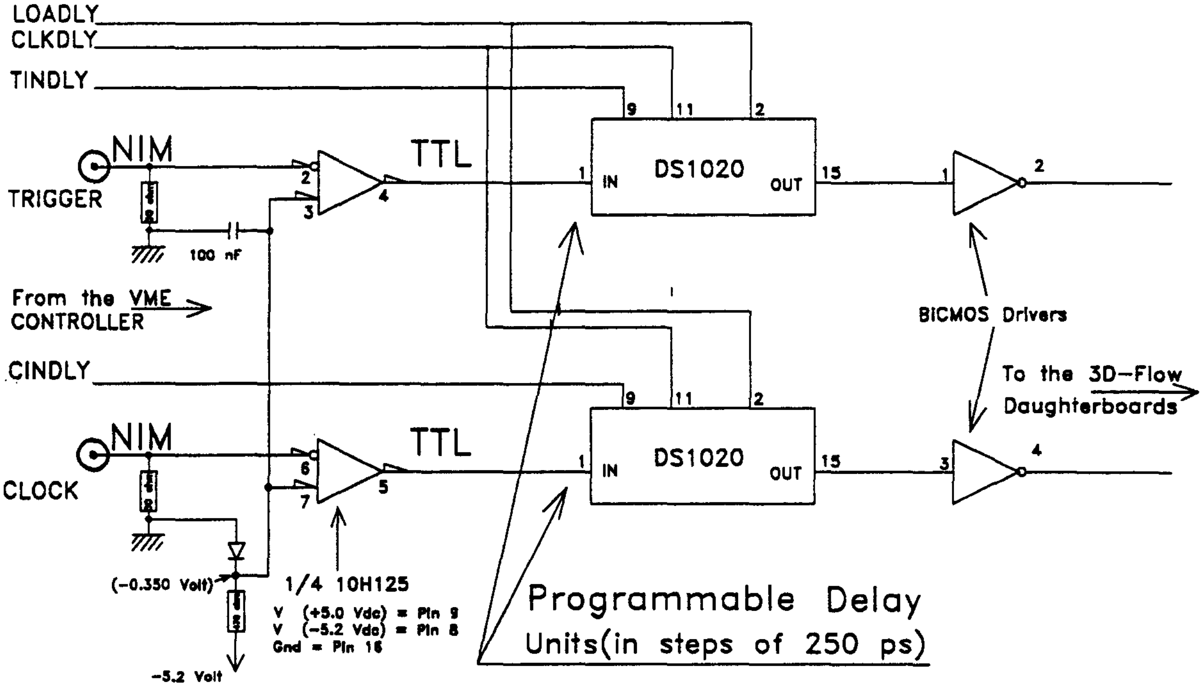
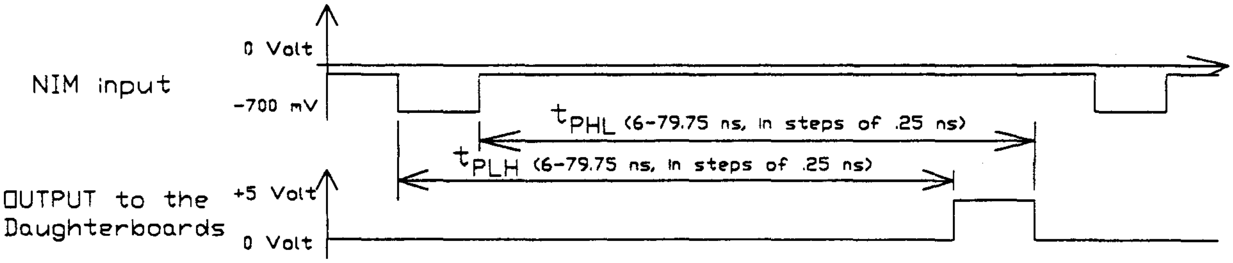


Figure 22 **Programmable delay circuit for trigger and clock signals implemented on the control board.**



NIM input
0 Volt
-700 mV
t_{PHL} (6-79.75 ns, in steps of .25 ns)
t_{PLH} (6-79.75 ns, in steps of .25 ns)
OUTPUT to the Daughterboards
+5 Volt
0 Volt

Figure 23 **Timing of the circuit of Figure 22 (The delay of the BICMOS buffer changes with the change of the load of the numbers of layers of 3D-Flow processors and is compensated by the programmable delay line).**

6.3 3D-Flow Motherboard

The 3D-Flow motherboard (Figures 24-25) is a six-layer printed circuit board that interfaces the data-acquisition system to the stack of the 3D-Flow parallel-processing system. On one side are the connectors to receive data from the receiver module. and on the other side are the connectors to match the top connectors of the 3D-Flow daughterboard.

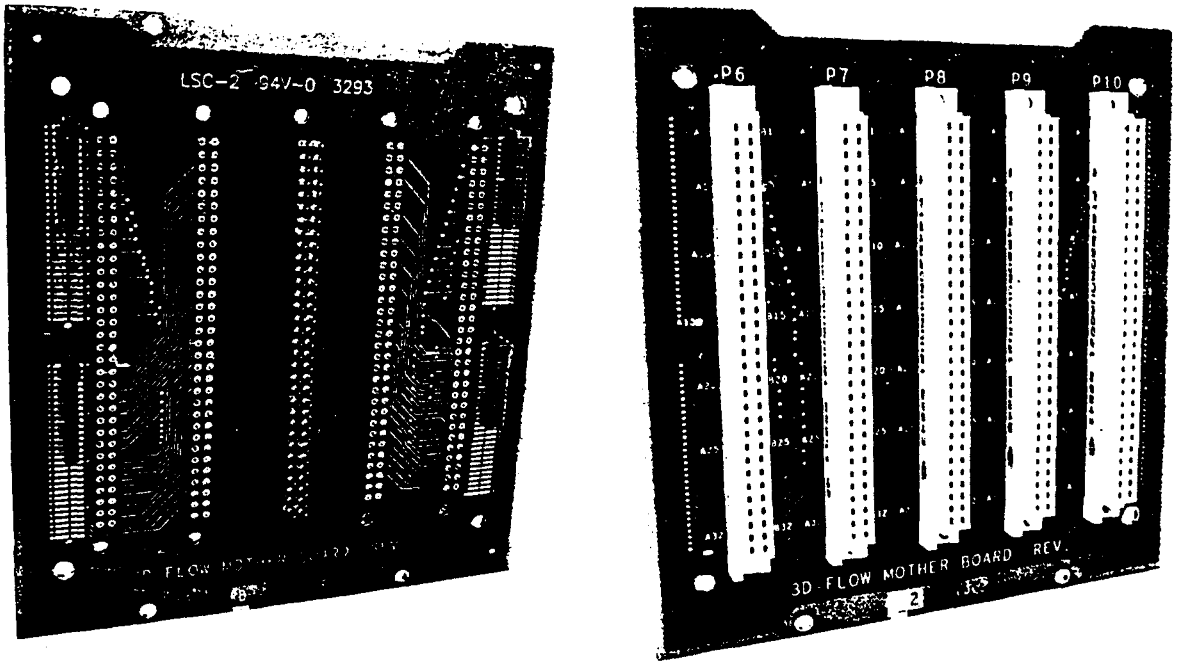


Figure 24. 3D-Flow six-layers PCB motherboard prototype.

The motherboard PCB is a six-layer printed circuit board. Figures 25 show the front and rear view of the board with connectors.

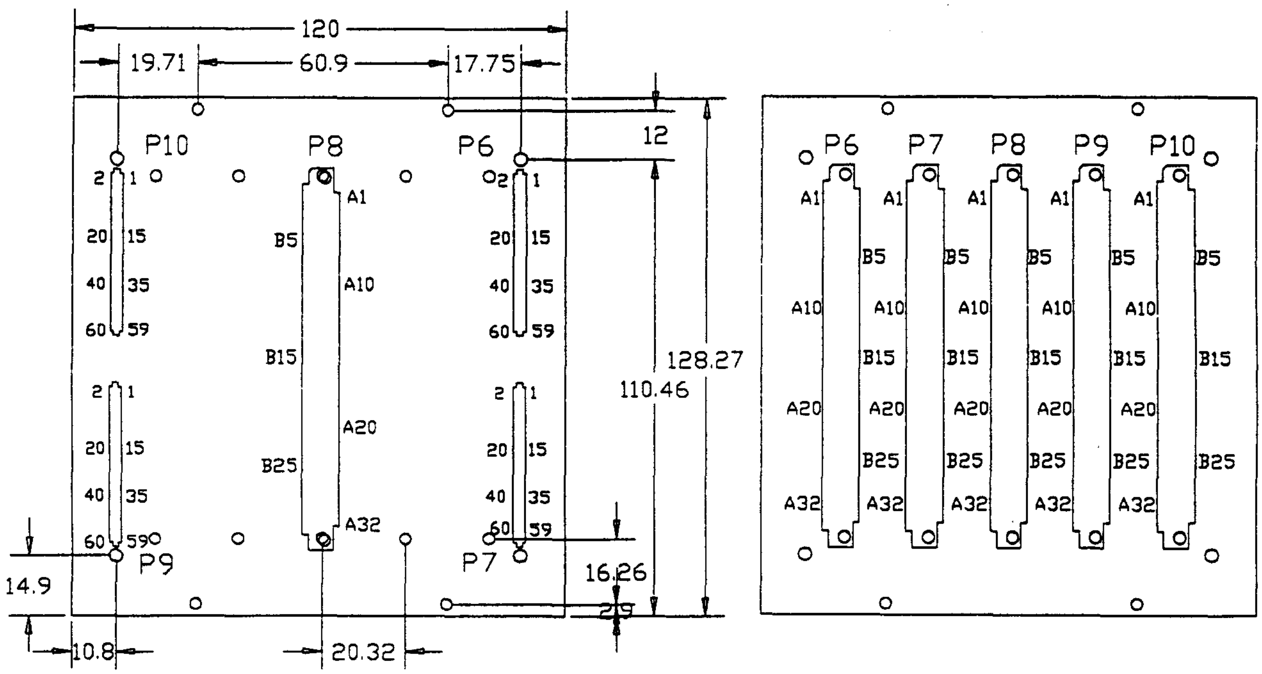


Figure 25. 3D-Flow motherboard PCB (Front view to the right, rear view to the left).

6.4 3D-Flow Daughterboard

6.4.1 Printed Circuit Board Prototype layout.

The design of the 3D-Flow daughterboard is an eight-layer printed circuit board. Figure 26 shows the layout of the components and connectors.

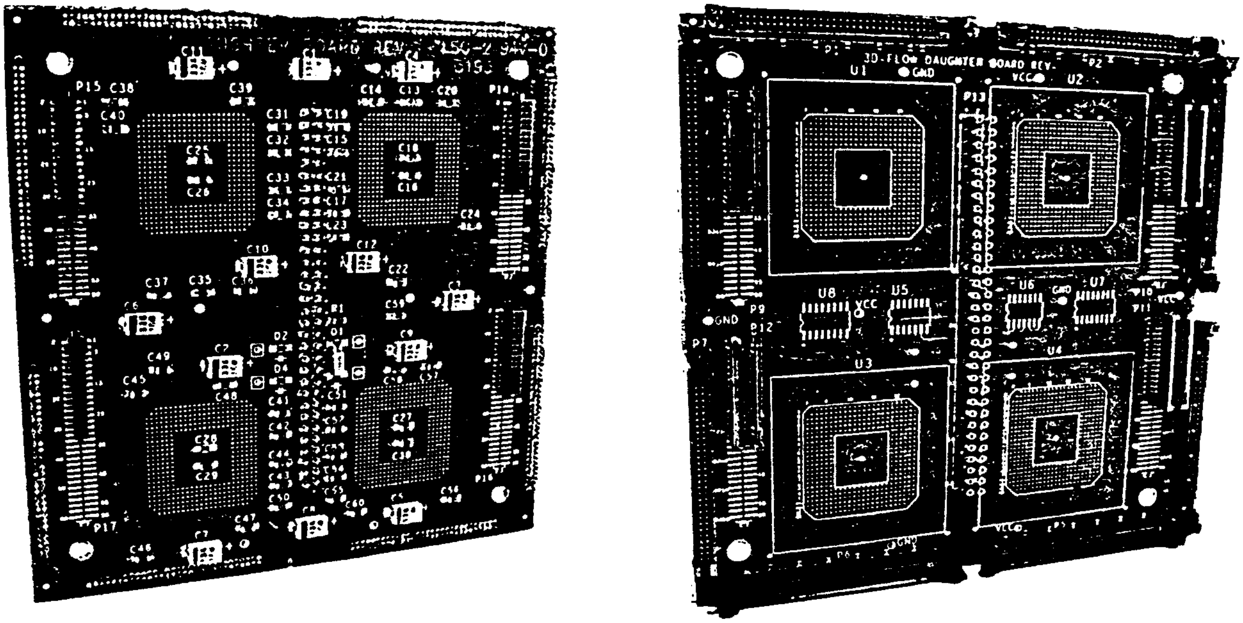


Figure 26. 3D-Flow eight-layers PCB daughterboard prototype.

The 3D-Flow daughterboard PCB (Figures 27) has 32 × 12-bit communication paths that are separate for input and for output in the North, East, West, and South directions. Each side (North, East, West, and South) has four 12-bit paths in input and in output that are carried with the control lines through two flexible printed circuits with 60 connections each (as described in Section 5.6). The Top to Bottom 16 × 12-bit unidirectional communication path from board to board in the same stack is carried with the control lines through the connectors P9, P10, P11, P12, P14, P15, P16, and P17 (60-pin surface mounted connectors with 50-MIL spacing).

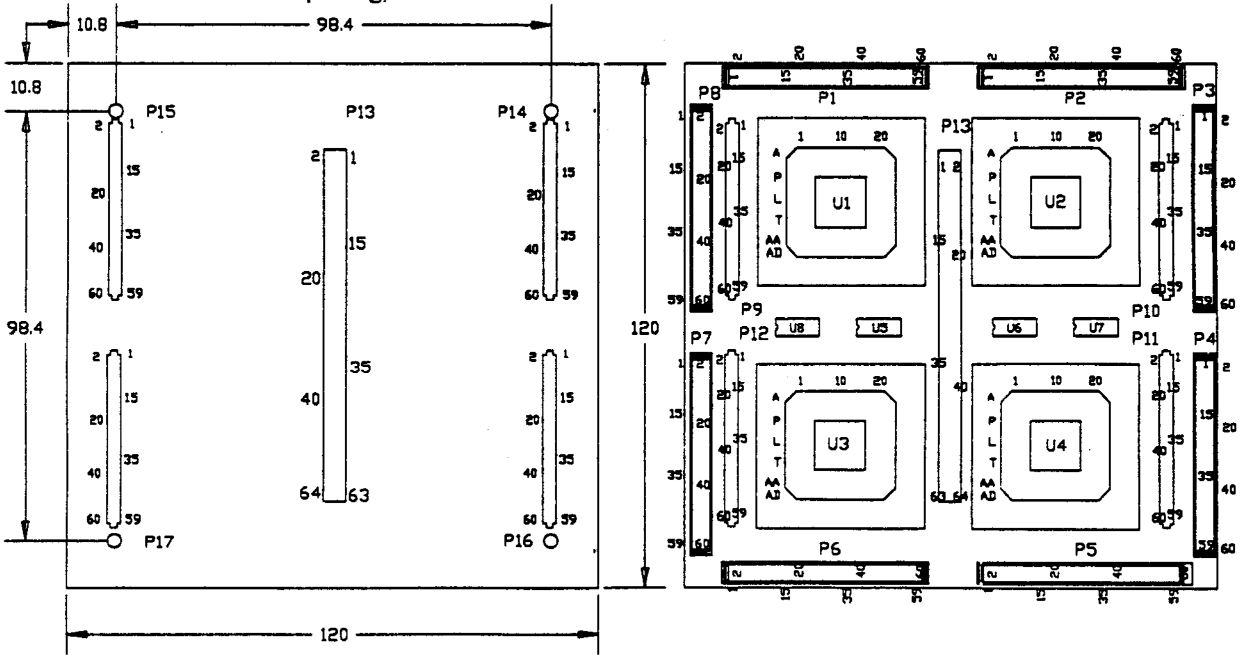


Figure 27. 3D-Flow daughterboard PCB (Front view to the left, rear view to the right).

6.4.2 Test Results of the Signals Transmission Between Adjacent Daughterboards on the 3D-Flow System Prototype

Tests performed on a prototype with a Digitalizing Oscilloscope HP 54111D 2GSa/s are shown in the following figures. Figures 28(a) and 28(b) show the hardware on which the tests have been performed. Figure 29(a) shows the test results of transmission between daughterboards at 60 MHz with BiCMOS drivers. and Figure 29(b) shows the test results at 140 MHz. The logical circuit is described in Figure 30.

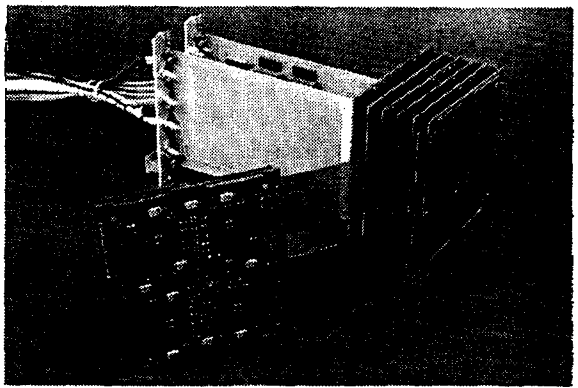


Figure 28(a) 3D-Flow Mini-Rack prototype assembly for the test of signal Transmission between adjacent daughterboards.

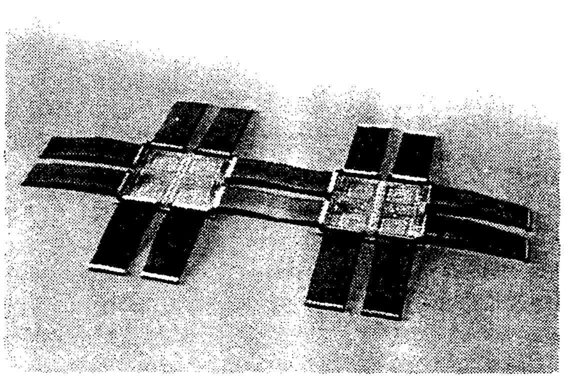
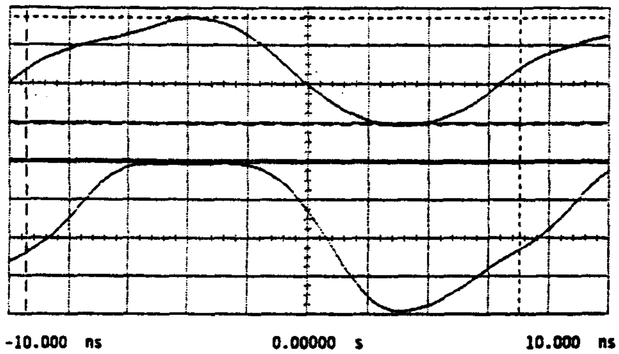


Figure 28(b) Assembly of two adjacent prototype daughterboards.

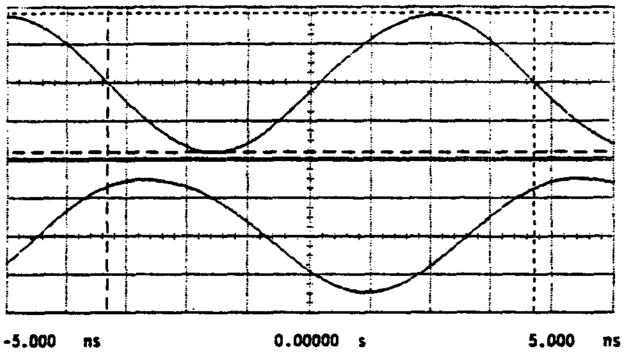


```
Ch. 1      =  2.000  Volts/div          Offset     =  1.860  Volts
Ch. 2      =  2.000  Volts/div          Offset     =  1.740  Volts
Timebase   =  2.00 ns/div               Delay      =  0.00000  s

Channel 1 Parameters                    P-P Volts =  5.460  Volts
Fall Time =  3.840    ns                Freq.      =  60.6997 MHz
Period     =  16.470  ns                +Width     =  8.960    ns
-Width     =  7.510   ns                Overshoot =  0.000  Volts
Preshoot   =  0.000  Volts              Avg Volts =  2.654  Volts
RMS Volts =  3.283  Volts               Dutycycle =  54.40 %
Delta V    =  5.468  Volts
Vmarker1   = -171.2 mVolts              Vmarker2   =  5.297  Volts
Delta T    =  16.470  ns
Start      = -9.450   ns                Stop       =  7.020    ns

Trigger  mode  :  Edge
On  Neg. Edge on Chan1
Trigger Levels
Chan1      =    1.860  Volts
Holdoff    =    70.000  ns
```

Figure 29(a) Test results for the signal transmission between adjacent daughterboards at 60 MHz. (Waveforms show the delay between transmitting and receiving; measured values refer to transmitting.)



```
Ch. 1      =  1.000  Volts/div          Offset     =  1.890  Volts
Ch. 2      =  1.000  Volts/div          Offset     =  1.820  Volts
Timebase   =  1.00 ns/div               Delay      =  0.00000  s

Channel 1 Parameters                    P-P Volts =  3.610  Volts
Rise Time =  2.250    ns                Fall Time =  2.080    ns
Freq.      =  142.358 MHz               Period     =  7.020    ns
+Width     =  3.540   ns                -Width     =  3.490    ns
Overshoot =  0.000  Volts               Preshoot   =  0.000  Volts
Avg Volts =  1.891  Volts               RMS Volts =  2.259  Volts
Dutycycle =  50.30 %
Delta V    =  3.610  Volts
Vmarker1   =  90.00 mVolts              Vmarker2   =  3.700  Volts
Delta T    =  7.020   ns
Start      = -3.310   ns                Stop       =  3.710    ns

Trigger  mode  :  Edge
On  Pos. Edge on Chan1
Trigger Levels
Chan1      =    1.890  Volts
Holdoff    =    70.000  ns
```

Figure 29(b) Test results for the signal transmission between adjacent daughterboards at 140 MHz. (Waveforms show the delay between transmitting and receiving; measured values refer to trasmitting.)

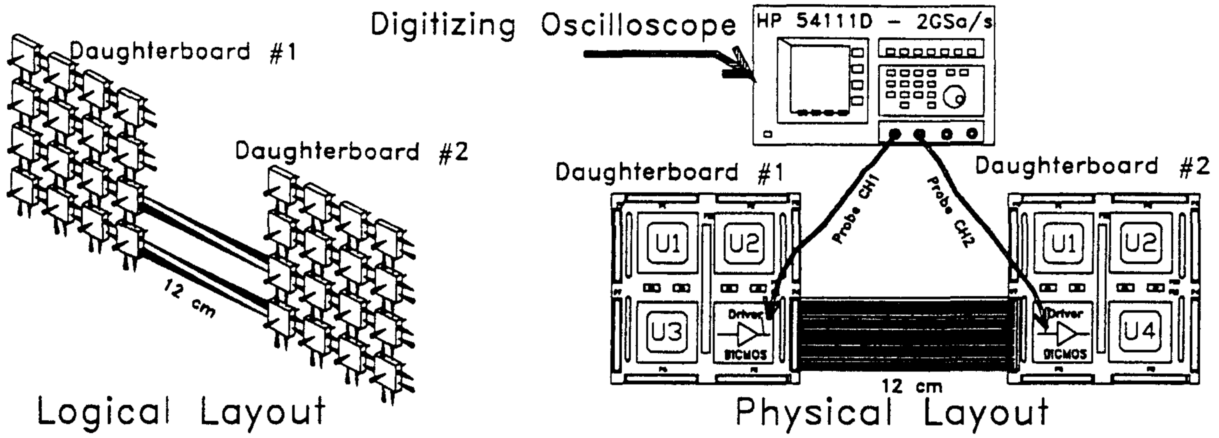


Figure 30 **Circuit layout for the test of the signal transmission between adjacent daughterboards.**

6.5 3D-Flow Chip

A careful study [ref. 23-25] (approaching at first the goal of accommodating 32 × 3D-Flow processors in one chip), including several requests for information and quotes from industry (IBM, AT&T, Motorola, LSI, etc.), has demonstrated that the most convenient 3D-Flow chip in terms of cost/performance at the present state of art is one that accommodates four 3D-Flow processors at 60 MHz as shown in Figure 31 (even if technology at present can deliver chips at 250 MHz).

The present packaging has led to having four processors with 12-bit I/O bus (see Figure 28), with a pin grid array PGA or with a lower-cost 420-pin package Lan Grid Array (LGA), or Lan Bulk Array.

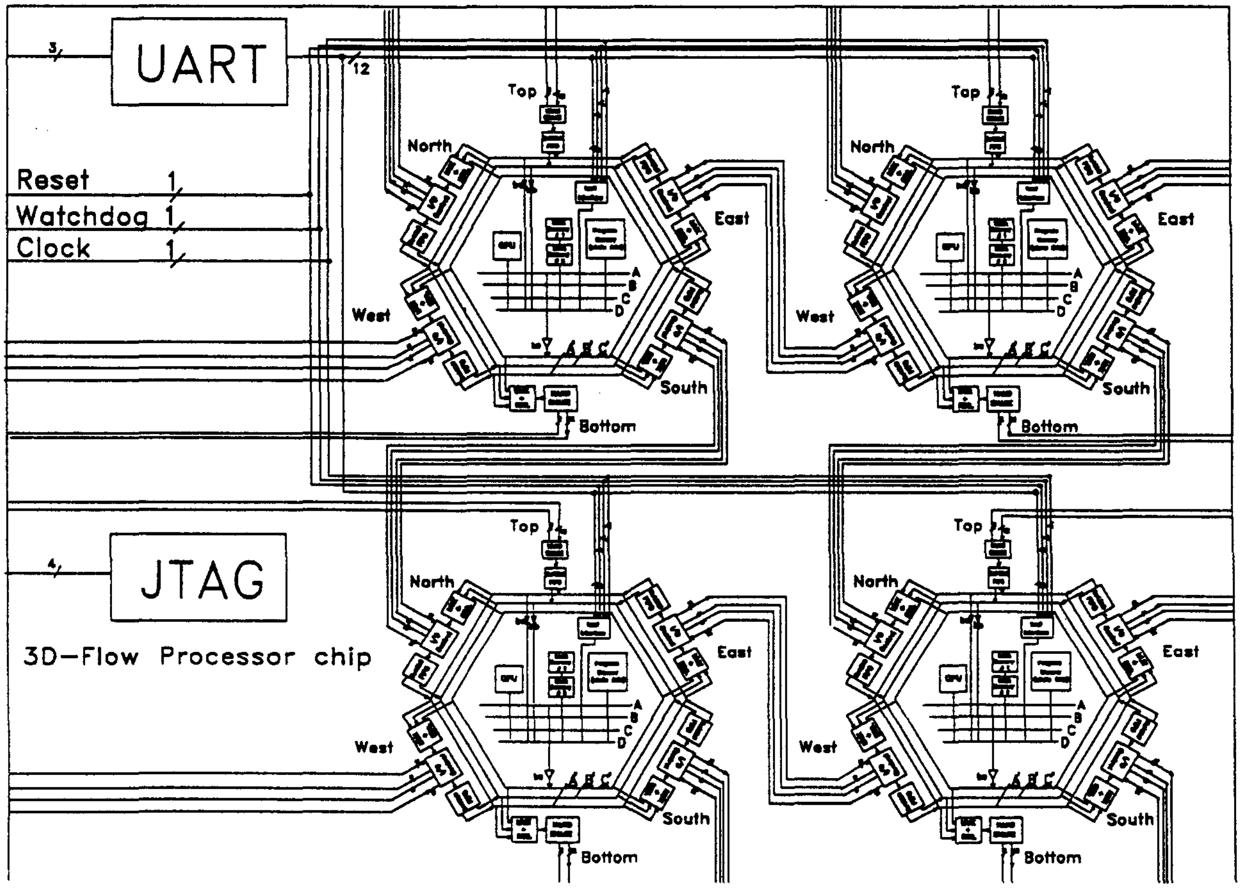


Figure 31. 3D-Flow chip.

7.0 SIGNALS DISTRIBUTION AND SYNCHRONIZATION

7.1 Clock and Trigger Signals Distribution

The hardware assembly of the 3D-Flow system allows good distribution of the clock signals with a short time difference between the same signals in different geographical locations. This feature is achieved by providing at each Mini-Rack (located in the power and control receiver board) programmable delay lines. The fine adjustment of the control signals can be achieved by tuning the programmable delay lines at each individual Mini-Rack (see Figure 32 and the details of the delay lines circuit at Section 6.2)). The crate controller (*e.g.*, a VME that may consist of a standard CPU such as the Motorola 68K and several boards with serial RS232C communication links and a few boards generating fan-out for the control signals) has fan-out that generates as many trigger and clock signals as the number of 3D-Flow Mini-Rack towers. The signals are sent to the tower as shown in Figure 32 and 33. The module in the Mini-Rack receives NIM logic level signals, propagates them to the next Mini-Racks through an external output Lemo connector, converts them on the controller module to TTL logic level, and sends them through a clock driver circuit (applying a delay according to the geographical position of that particular Mini-Rack in the overall system) to its 3D-Flow stack of daughterboards housed at a 90^{o} angle in the back of the Mini-Rack.

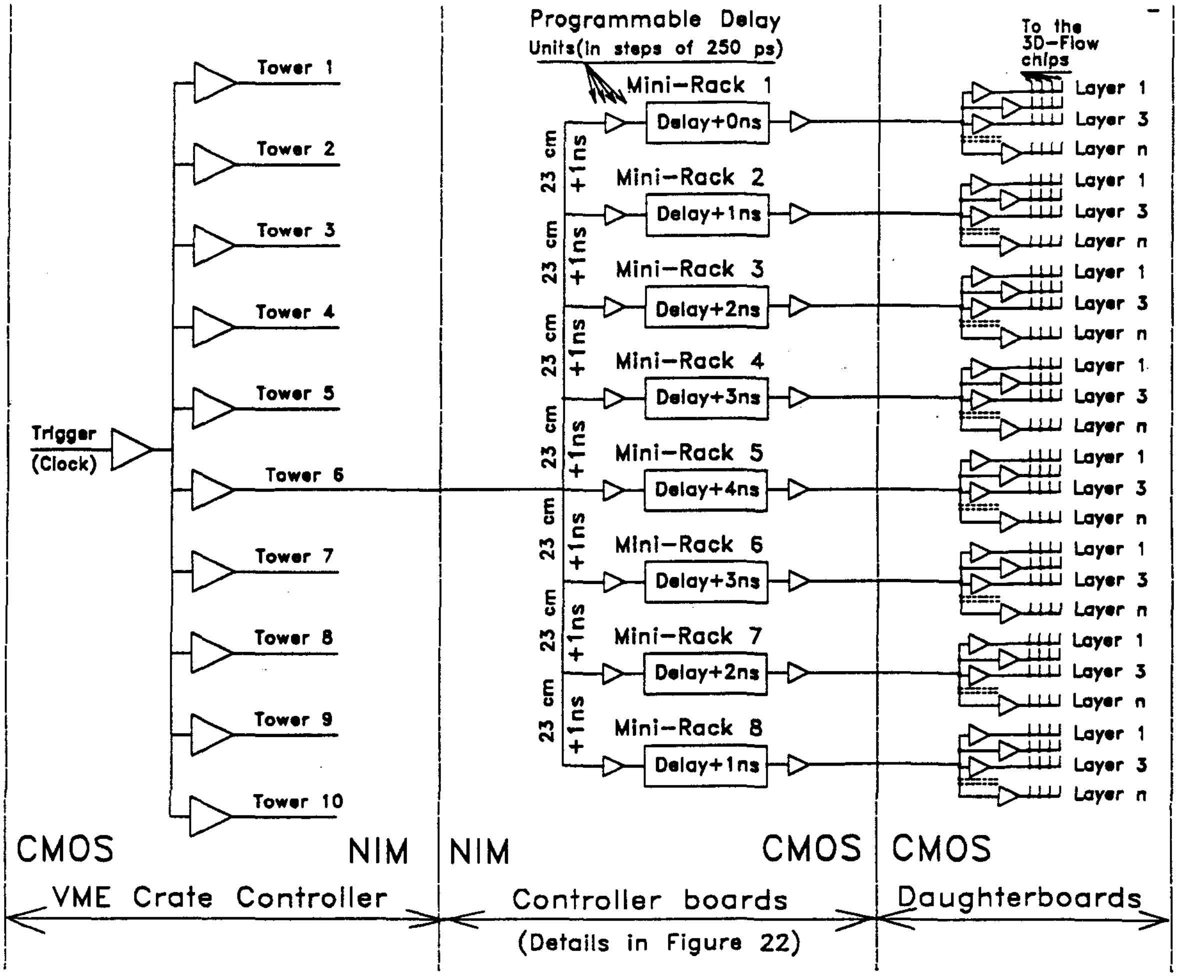


Figure 32. Clock and trigger signals distribution and synchronization. (logical layout).

Since the path of each control line at this stage has only one buffer, and the length of the connection to the furthest 3D-Flow processor is estimated, for the mentioned applications, to not exceed 23 cm, then after calibration, the maximum difference in timing among all processors of the 3D-Flow parallel-processing system would be equal the delay time provided by the type of driver used for 23 cm. If greater precision is desired, then programmable delay lines should be accommodated on each 3D-Flow daughterboard.

Calibration is done in the following manner:

- Each programmable "delay line" at each Mini-Rack signal control receiver board is programmed with a delay timing according to its distance from the receiving signal. Figure 32 shows the different delays applied to different Mini-Rack signals of a "Tower." Figure 33 shows the physical layout of the distribution of the trigger and clock signals. From one Mini-Rack to the adjacent one above or below there is a cable length of 23 cm, equivalent to 1 ns in NIM logic.
- A known data stream is then sent from the source (simulated at the detector site). Each 3D-Flow processor is running a diagnostic program that fetches input data from the "Top" port at different speeds asynchronously with respect to the clock of the 3D-Flow processor. Each datum fetched by each 3D-Flow processor has a time stamp associated with it (by the program) at the output. The 3D-Flow processors with a time stamp different than expected will need a calibration in timing by means of the programmable "delay line" at its Mini-Rack.
- An automatic calibration of each delay line can be accomplished by sending two different consecutive data to the "Top" input port of the 3D-Flow processor, and by changing under program control the delay line, and constantly reading the data value from the "Top" port of the 3D-Flow processor until its value changes.

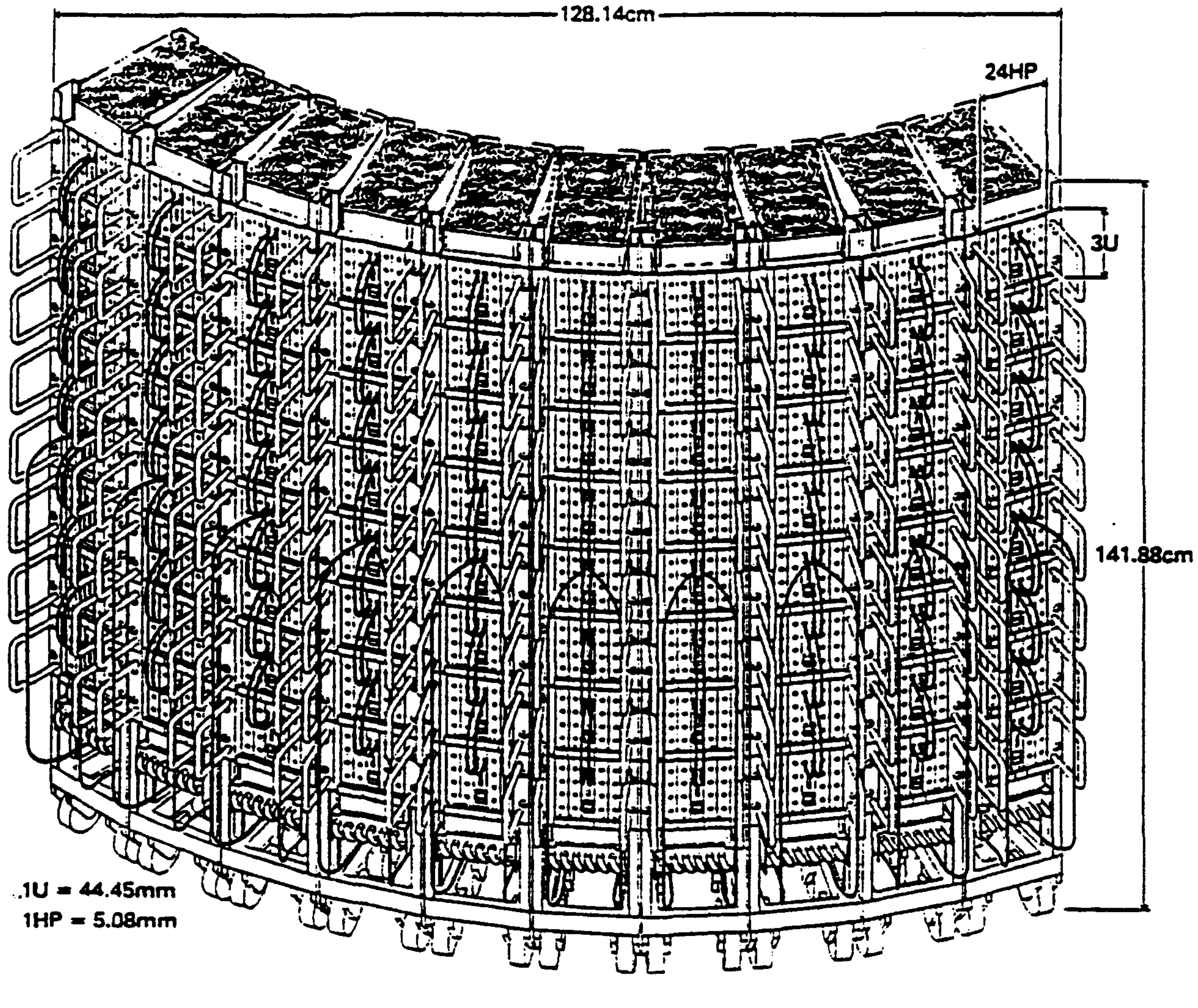


Figure 33. Clock and trigger signals distribution and synchronization. (Physical layout on 80 x Mini-Racks.)

7.1.1 Clock and Trigger Signal Distribution on the 3D-Flow System Prototype.

Figures 34(a) and 34(b) show the prototype 3D-Flow assembly used to perform the tests. A controller board receives signals from the trigger and the clock connectors. and converts these input signals from NIM logic to CMOS logic. as described in figure 22 and Section 6.2. The converted signals then go through the programmable delay lines and are sent through BiCMOS drivers to the daughterboards.

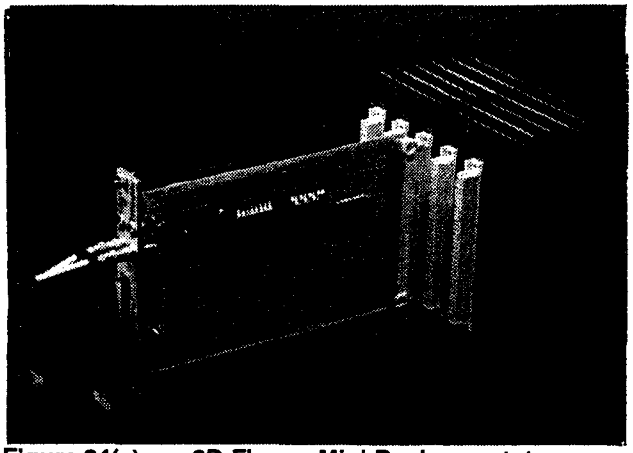


Figure 34(a). 3D-Flow Mini-Rack prototype assembly for the test of clock and trigger signal distribution. (front view).

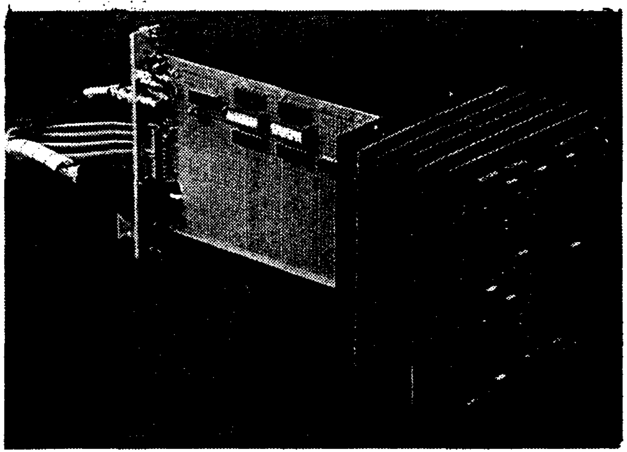
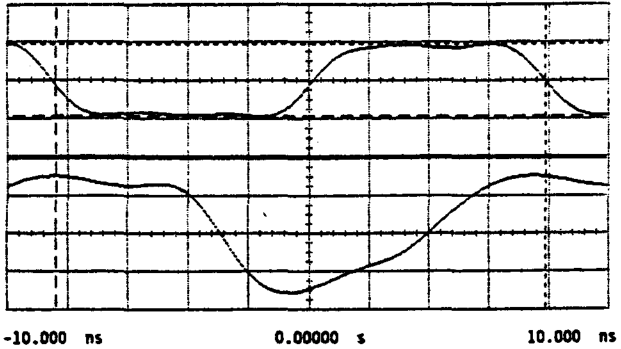


Figure 34(b). 3D-Flow Mini rack prototype assembly for the test of clock and trigger signal distribution. (rear view).

7.1.2 Test Results of the Trigger and Clock Signals Distribution.

Figure 35 show the test results at 60 MHz for zero value program delay of the signals (trigger and clock) from the input, through the programmable delay lines, to the motherboard center connector.

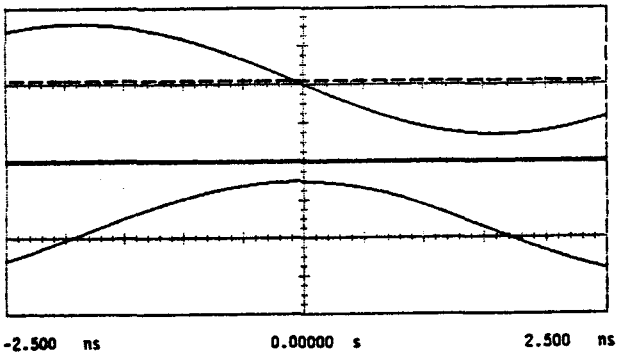
Figure 36 show the test result at 140 MHz of the delay of the clock (and trigger) signal between the first and the last layer of daughterboards



```
Ch. 1       =  400.0 mVolts/div          Offset     = -335.0 mVolts
Ch. 2       =  2.000  Volts/div          Offset     =  1.740  Volts
Timebase    =  2.00 ns/div               Delay      =  0.00000  s

Channel 1 Parameters                     P-P Volts  =  780.0 mVolts
Rise Time  =  2.140    ns                Fall Time  =  1.860    ns
Freq.      =  61.3961 MHz                Period     =  16.290   ns
+Width     =  7.760    ns                -Width     =  8.530    ns
Overshoot  =  25.00 mVolts               Preshoot   =  6.249 mVolts
Avg Volts  = -356.3 mVolts               RMS Volts  =  484.2 mVolts
Dutycycle  =  47.60 %
Delta V    =  748.0 mVolts
Vmarker1   = -708.0 mVolts               Vmarker2   =  40.00 mVolts
Delta T    =  16.290  ns
Start      = -8.390   ns                 Stop       =  7.900   ns
```

Figure 35 Test results of the programmable delayed clock and trigger signals between the Input NIM level and the CMOS level at the motherboard at 60 MHz.



```
Ch. 1       =  800.0 mVolts/div          Offset     =  1.860  Volts
Ch. 2       =  1.000  Volts/div          Offset     =  1.800  Volts
Timebase    =  500  ps/div               Delay      =  0.00000  s

Channel 1 Parameters                     P-P Volts  =  2.272  Volts
Fall Time  =  2.020    ns                Overshoot  =  0.000  Volts
Preshoot   =  0.000  Volts               Avg Volts  =  1.933  Volts
Delta V    =  0.000  Volts
Vmarker1   =  1.936  Volts               Vmarker2   =  1.936  Volts
Delta T    =  0.00000  s
Start      =  47.500  ns                 Stop       =  47.500  ns
```

Figure 36 Test results of the clock and trigger signals at 140 MHz between the first and the last layer of daughter boards.

Figure 37 show the logical layout for the test of the clock, trigger and data-flow signals distribution. Figure 38(a) and 38(b) show the maximum delay that occurs between Top and Bottom ports of two 3D-Flow processors placed at the first and the third and at the first and the sixth layer, respectively, of a stack of daughterboards. The tests were performed by inserting a jumper at the pinout of the LGA of the 3D-Flow processor between the Top and Bottom port on each daughterboard. Complete delay between the first and last layer of the 3D-Flow stack should also take into consideration the intrinsic delay of each 3D-Flow from the Top to the Bottom port. This will depend on the technology used in the fabrication of the chip.

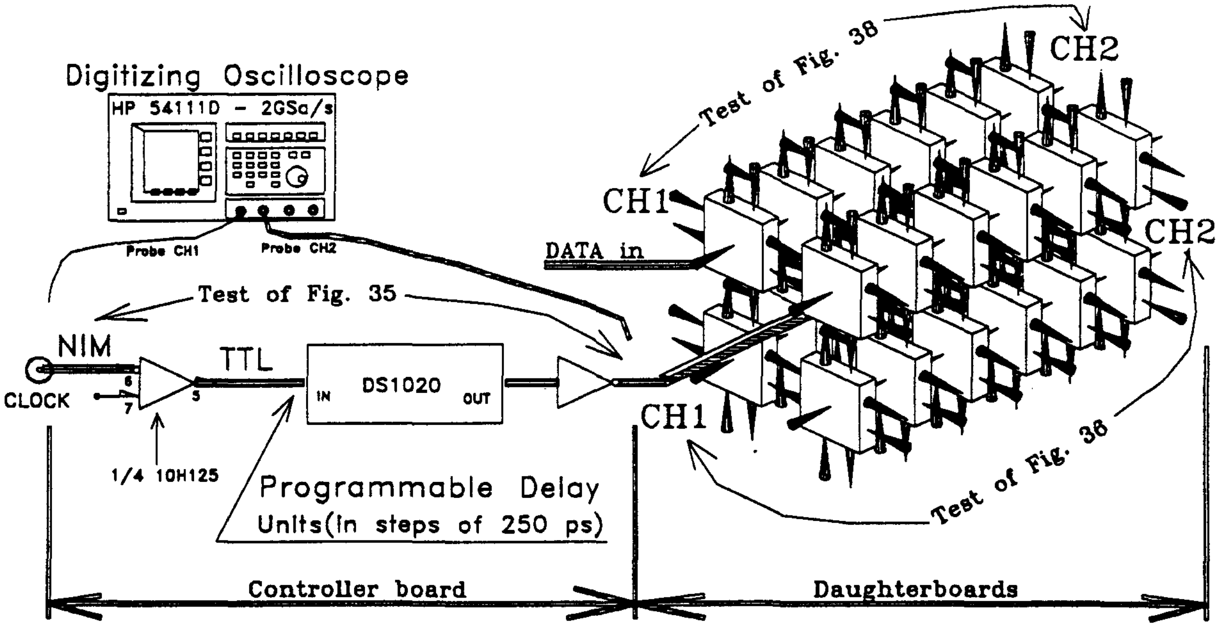
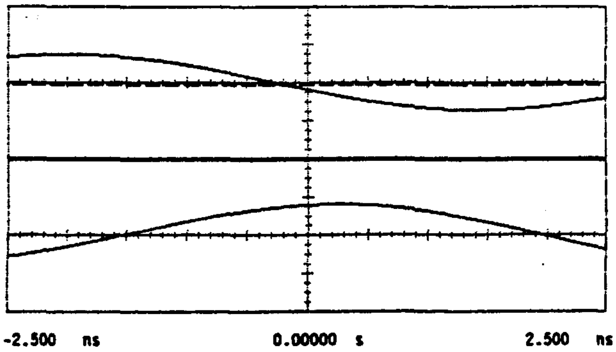


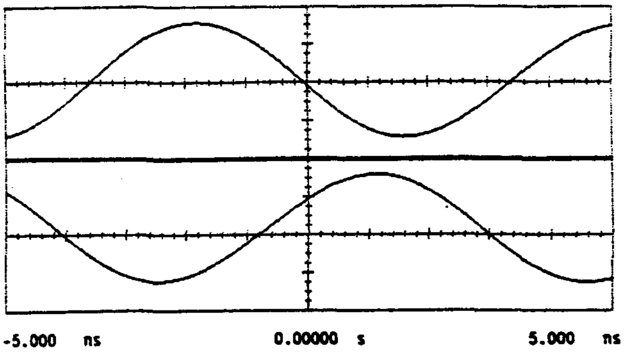
Figure 37 **Circuit layout for the test of the clock, trigger, and data-flow signals distribution.**



```
Ch. 1      =  2.000  Volts/div             Offset     =  1.840  Volts
Ch. 2      =  2.000  Volts/div             Offset     =  1.780  Volts
Timebase   =  500  ps/div                  Delay      =  0.00000  s

Channel 1 Parameters                       P-P Volts =  2.940  Volts
Fall Time =  1.950   ns                    Overshoot =  0.000  Volts
Preshoot   =  0.000  Volts                 Avg Volts =  1.760  Volts
Delta V    =  0.000  Volts
Vmarker1   =  1.760  Volts                 Vmarker2   =  1.760  Volts
Delta T    =  0.00000  s
Start      =  47.500  ns                   Stop       =  47.500  ns
```

Figure 38(a) **Test results of the Top to Bottom 3D-Flow signals at 140 MHz between the first and the third layer of daughterboards. (Waveforms show the delay between transmitting and receiving; measured values refer to transmitting.)**



```
Ch. 1      =  1.000  Volts/div             Offset     =  1.840  Volts
Ch. 2      =  800.0 mVolts/div             Offset     =  1.780  Volts
Timebase   =  1.00 ns/div                  Delay      =  0.00000  s


Trigger  mode  :  Edge
On  Neg. Edge on Chan1
Trigger Levels
Chan1      =    1.840  Volts
Holdoff    =    70.000  ns
```

Figure 38(b) **Test results of the Top to Bottom 3D-Flow signals at 140 MHz between the first and the sixth layer of daughterboards. (Waveforms show the delay between transmitting and receiving; measured values refer to transmitting.)**

The compactness. modularity. and regularity of path distances among electronic components in the Mini-Rack assembly (traditional assembly with vertical boards in slots will require some short and some long connections) will allow us to implement the same system with the higher-speed components that will be available in the future. A 240-MHz 3D-Flow will not be a problem in such architecture. as it still gives a good margin of tolerance in signal synchronization with normal low-power consumption components. If one would then like to shrink the system by using higher-density connectors. more precise (costly) mechanics. cooling system. etc.. then the architecture is still suitable for reaching very-high-speed communications and higher performance.

7.2 Other Control Signals Distribution

The reset signal is sent to all 3D-Flow processors. Since the signal is not time-critical. it is not controlled in timing as are the previous trigger and clock signals. It requires only to be buffered in order to provide enough current to all the inputs of that signal within a Mini-Rack. The "jtag" control signals are provided to check the 3D-Flow boards and for automatic debugging and troubleshooting. In order to program the delay line. the following signals must be provided: load delay (LOADLY): clock to program the delay line (CLKDLY): serial input to program the delay line of the trigger signal (TINDLY): and the serial input to program the delay line of the clock signal (CINDLY).

7.3 Serial I/O RS232C Signals Distribution

The serial I/O signals are generated by the integrated circuit Philips SCC2698B (Octal UART or equivalent) residing in the VME system controller. They are distributed to the 3D-Flow parallel-processing system through the control lines connector placed on the front panel of the central module of the Mini-Rack. For example, a trigger system such as the GEM experiment, with approximately 1280 trigger towers corresponding to 1280 × 3D-Flow processors, will require 40 integrated circuits from Philips SCC2698B (or equivalent) in the crate controller. This setup is assigning one serial I/O for each stack of 3D-Flow chips mapped into four trigger towers.

Depending on the number of 3D-Flow processor array layers (or stages), each UART controller in the "system crate controller" (*e.g.*, VME) will handle communication with one 3D-Flow processor and the ones associated to it in the other layers (or stages). This fine distribution of RS232C signals is very important and convenient for monitoring the entire 3D-Flow parallel-processing system during run-time and it will also provide the capability of parallel loading of all programs and constants during initialization phase of the system (power up).

Each Mini-Rack controller module will contain the following electronics:

- Four RS232C transmitters. each receiving information from four different UARTs in the "system crate controller" and each transmitting the information to up to n × 3D-Flow UARTs receivers. This implies that during a broadcast operation up to n × 3D-Flow UARTs will receive the information. (The number "n" of 3D-Flow processor stack will determine the type of driver to be used);
- Four RS232C receivers. each sending information to four different UARTs in the "system crate controller" and each receiving information from up to n × 3D-Flow UARTs transmitters. (An SN75174 quad line drivers with nand enabled three-state outputs has been used in the prototype. This driver meets EIA-485. EIA-422A Standard and CCITT recommendations V.11 and X.27. They are designed for multipoint transmission and long bus lines in noisy environments.

Depending on the dimension of the overall 3D-Flow parallel-processing system that has to be implemented. these control lines should have a fanout ranging from 1 to 48 loads. Communication to the 3D-Flow chips through the serial RS232C lines will be as follows:

The broadcasted message information can be a broadcast talk to all 3D-Flow chips. or a message to a specific 3D-Flow chip (among the set of 3D-Flow processors in a stack) in either listening or talking mode.

In the case of talking, the message contains the ID number of the 3D-Flow chip under control. (Note that each 3D-Flow chip has four 3D-Flow processors.) At each 3D-Flow chip the following operation takes place in order to understand whether the message was addressed to itself. At each 3D-Flow processor, the message is fetched and compared with its ID number (determined by comparing 6-bit to a switch set on its specific 3D-Flow daughterboard and depending on its physical position in the board itself). When a particular 3D-Flow chip recognizes the message for itself, it prepares itself to listen and to load the program or constants into its memories, or it prepares to talk and to send the requested information by enabling the signals on the common (to the other 16 × 3D-Flow chips) transmitting line. A clock is also distributed to all UARTs from the signals control module.

8.0 DIAGNOSTICS AND MAINTENANCE

8.1 Hardware Maintenance of the System.

The assembly of the parallel-processing system was designed to allow easy access for future ugrades and for maintenance of the system.

The system comprises several Mini-Racks, each made of a standard assembly (the front part used for data acquisition) and the 3D-Flow daughterboards joined at 90°. The Mini-Rack and the stack of 3D-Flow daughterboards are joined by the "motherboard." All these parts are shown assembled and disassembled in Figures 12 and 13.

The Mini-Racks are assembled in a rack cabinet, called a "Mini-Rack Tower," made of standard parts, as shown in Figure 14.

If a 3D-Flow daughterboard requires maintenance, the board to be repaired can be easily accessed from the back. In case a cylinder assembly is used, one can access any daughterboard from the front in the following manner:

- Disconnect the cables at the front panels of the Mini-Rack to be repaired;
- Extract the receiver boards from the Mini-Rack;
- Remove four screws at the back of the motherboard and four screws at the front panel, as shown in Figure 12;
- Extract the Mini-Rack from the Mini-Rack Tower enclosure;
- Disconnect the side (North, East, West, and South) connectors from the 3D-Flow daughterboard and extract for maintenance.

Any upgrade of the system can be done very easily from the back by adding daughterboards, as shown in Figure 9.

8.2 Diagnostic Features of the System

Diagnostics on the 3D-Flow parallel-processing system were given great consideration during the design phase. There are several levels of diagnostics, including troubleshooting and system-monitoring tools to help the user to develop and debug applications.

An electrical check of the 3D-Flow processors is made through the boundary- and full-scan jtag circuitry. Loading of the program and constants into the single 3D-Flow processors as well as monitoring of the entire system is done through serial I/O RS232. For that purpose the Philips integrated circuit SCC2698B (Octal UART, or Motorola equivalent) is used for 8 trigger tower signals (one serial I/O for each trigger tower-associated 3D-Flow processor). In the case of GEM or D0 application to the Level-1 Calorimeter Trigger, approximately 160 integrated circuits will be required. The UART interface inside the 3D-Flow chip is designed so that most of the internal "Units" are accessible to the UART, thus allowing monitoring of the entire system through the Serial I/O port. Other tools currently under development, such as assembler, simulator, and diagnostic programs, will allow the user to debug and troubleshoot system problems.

9.0 APPLICATIONS

9.1 Calorimeter Programmable Level-1 Trigger

The Superconducting Super Collider (SSC) and the Large Hadron Collider (LHC) are being built to study high-energy physics. Every 16 ns (40 ns), proton beams will collide, and the particles produced by the collision must be identified and studied.

The type of detectors and the physics involved in present experiments are reaching a level of cost and complexity so great that it is preferable to implement a programmable trigger solution at all levels rather than a system realized with cabled logic.

Many detectors will be used to detect and identify the particles. The calorimeter is one of the subdetectors to be used at the SSC and LHC. Two proton beams will collide in the center of the calorimeter sending particles to the calorimeter towers in the barrel and end caps. The amount of energy released in the collision is detected and then transferred through electronic channels to digital processors, where the identification of particles is begun in the Level-1 triggering.

9.1.1 Calorimeter Information at Level-1 Trigger

There are many conditions to test when making the Level-1 decision [ref. 26-29]. For distinguishing electrons and photons the energy of the calorimeter electromagnetic (em) trigger tower compartment must be greater than a threshold, the energy of the calorimeter hadronic (had) trigger tower compartment to "em" ratio must be very small. If isolation is to be achieved in Level-1, the surrounding towers must contain only small amounts of energy. For jet identification, the sum of a tower matrix of digitized energies in "em" and "had" in $\Delta\eta$ and $\Delta\Phi = 0.1 \times 0.1$ to 0.8×0.8 must be tested against several thresholds. To distinguish neutrinos, the E_t sum must be compared with a threshold.

There are several methods by which to verify the existence of such conditions. As an example of a programmable system, a few methods that will verify these conditions will be implemented using the 3D-Flow parallel-processing system array.

Another method recognizes clusters, and also tries to distinguish between an isolated electron and a jet. An isolated electron should be recognized by a large amount of energy deposited in a small area (about one tower wide), while a jet's energy should be spread out to cover a large matrix of calorimeter towers.

This method of electron finding considers the possibility of a "hit" occurring between two towers, with the result that the energy of the electron would be divided between the two towers. Therefore, an electron is distinguished from other particles by a 1 × 2 (sum of the reference trigger tower and the one above it) or 2 × 1 (sum of the reference trigger tower and the one on its right) region containing most of the energy, while the surrounding towers receive almost none [ref. 9]. Furthermore, an electron is considered to be isolated if the 2 × 2 "em" matrix contains most of the energy while the surrounding 12 "em" towers (in a 4 × 4 matrix) and the 16 "had" towers contain little energy.

There exist several jet-finding algorithms. A Monte Carlo simulation run at the SDC showed that for high-energy particles, the 8 × 8 trigger tower matrix was more efficient, while for lower-energy particles, the 4 × 4 trigger tower matrix was more reliable [ref. 9]. For this reason, both techniques are included in our algorithms.

Figure 39 shows a few examples of programmable trigger algorithms. The 3D-Flow system is not limited to the execution of these examples; any algorithms can be implemented by executing in the 3D-Flow processor a sequence of steps that are part of the 3D-Flow instruction set. Figure 40 shows a flow chart of an algorithm.

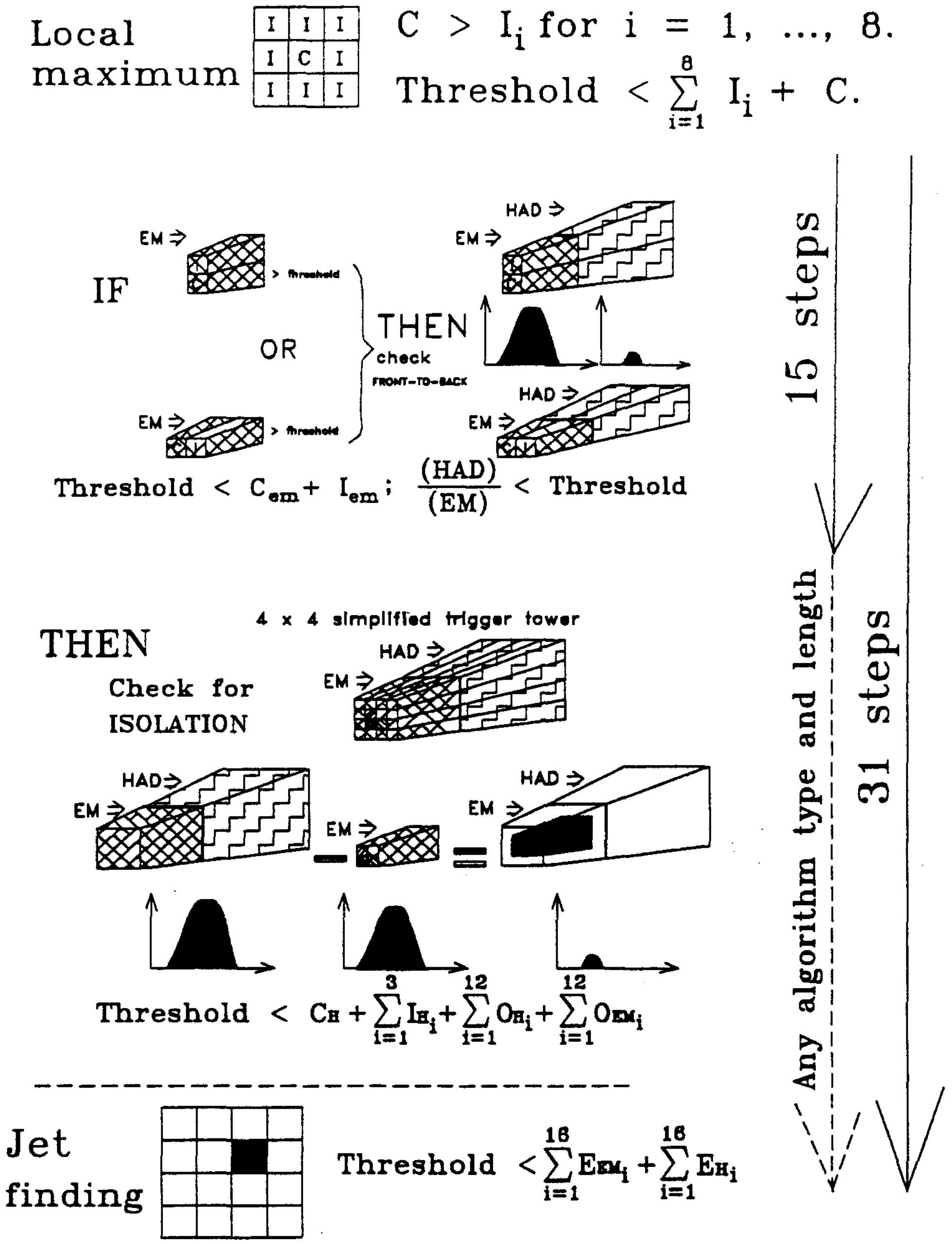
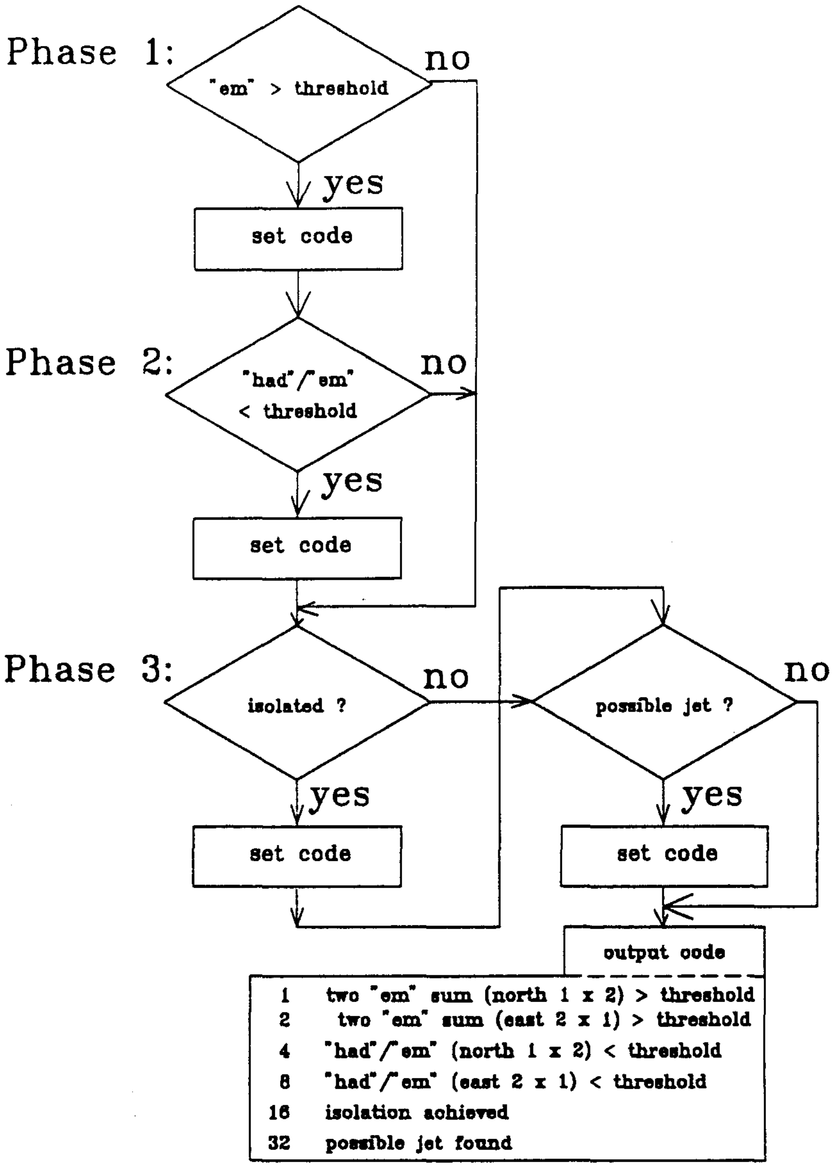

Local maximum
I I I
I C I
I I I
$C > I_i$ for $i = 1, \ldots, 8.$
$\text{Threshold} < \sum_{i=1}^{8} I_i + C.$
IF
EM ⇒
> threshold
OR
EM ⇒
> threshold
THEN
check
FRONT-TO-BACK
HAD ⇒
EM ⇒
HAD ⇒
EM ⇒
$\text{Threshold} < C_{em} + I_{em};\ \frac{(HAD)}{(EM)} < \text{Threshold}$
THEN
Check for ISOLATION
4 x 4 simplified trigger tower
HAD ⇒
EM ⇒
HAD ⇒
EM ⇒
EM ⇒
HAD ⇒
EM
$\text{Threshold} < C_{H} + \sum_{i=1}^{3} I_{H_i} + \sum_{i=1}^{12} O_{H_i} + \sum_{i=1}^{12} O_{EM_i}$
Jet finding
$\text{Threshold} < \sum_{i=1}^{16} E_{EM_i} + \sum_{i=1}^{16} E_{H_i}$
15 steps
Any algorithm type and length
31 steps

Figure 39. Trigger algorithm examples.



1	two "em" sum (north 1 x 2) > threshold
2	two "em" sum (east 2 x 1) > threshold
4	"had"/"em" (north 1 x 2) < threshold
8	"had"/"em" (east 2 x 1) < threshold
16	isolation achieved
32	possible jet found

Figure 40. Algorithm flowchart. For example, a 3D-Flow processor may return a code 37 (1+4+32), stating that a possible electron was found, but it was not isolated from the surrounding energy, and that cell may be part of a 4 × 4 jet.

9.1.2 Calorimeter Array Versus Processor Array

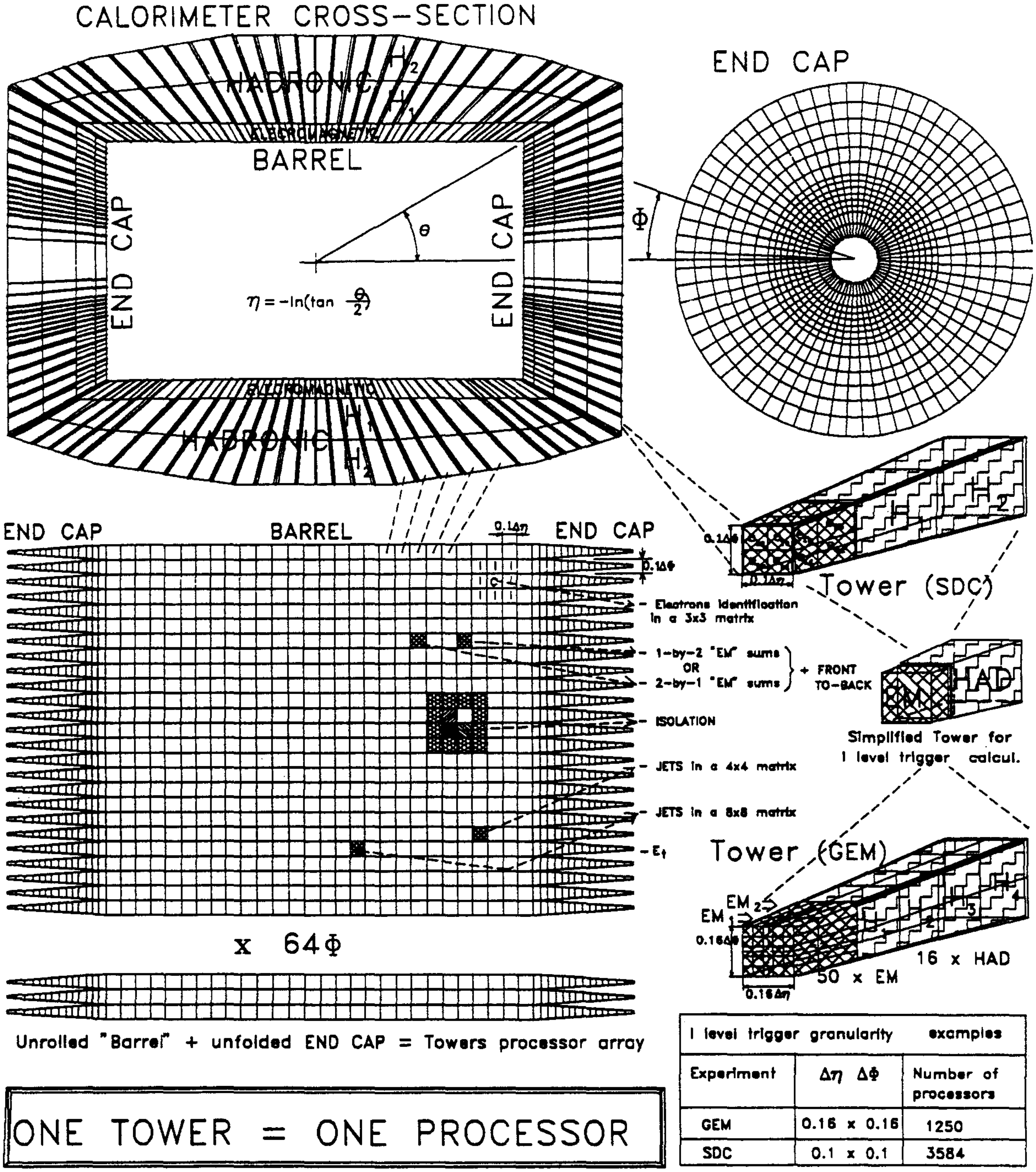
A lengthwise cross section and a side view of the end caps of the calorimeter are shown in Figure 41. In the experiments within GEM and SDC at the SSC Laboratory, there are varying calorimeter type, segmentation, and granularity of the digitized information for the Level-1 trigger. While GEM is experimenting with a 0.16 $\phi \times 0.16\ \eta$ calorimeter, SDC is developing a 0.1 $\phi \times 0.1\ \eta$ calorimeter trigger towers. Although, in SDC, each individual tower of the calorimeter is divided into four (barrel) to eight (end cap) "em" sections and two "had" sections (see center right of Figure 41), for the purpose of the Level-1 trigger, the "em" sections are combined into one value, as are the "had" sections.

The geographical representation of the calorimeter can be related to a processor array. Each calorimeter tower (consisting of an "em" part and a "had" part) has a one-to-one correspondence with an input processor cell in the first stage of the processor array (see bottom left of Figure 41). A description of both GEM and SDC towers as they relate to the simplified towers is shown on the right of Figure 41. The size of the processor array depends on the segmentation and granularity of the calorimeter (see Table 2).

The types of possible investigations that can be done on such a processor array in order to identify particles and to obtain relevant information are shown in boldface on the tower matrix array of Figure 41. A listing to the right of the matrix is provided.

Table 2 Examples of calorimeter segmentation.

Experiment	Subsystem	$\Delta\eta \times \Delta\Phi$	Total number of channels at full granularity	Macro-granularity for Level-1. Total number of towers=total number of input processors at the first stage.
SDC	EM HAD	0.05×0.05 0.1×0.1	21 504 7168	3584
GEM	EM HAD	0.032×0.032 0.8×0.8	30 000 × 2 5000 × 4	1250



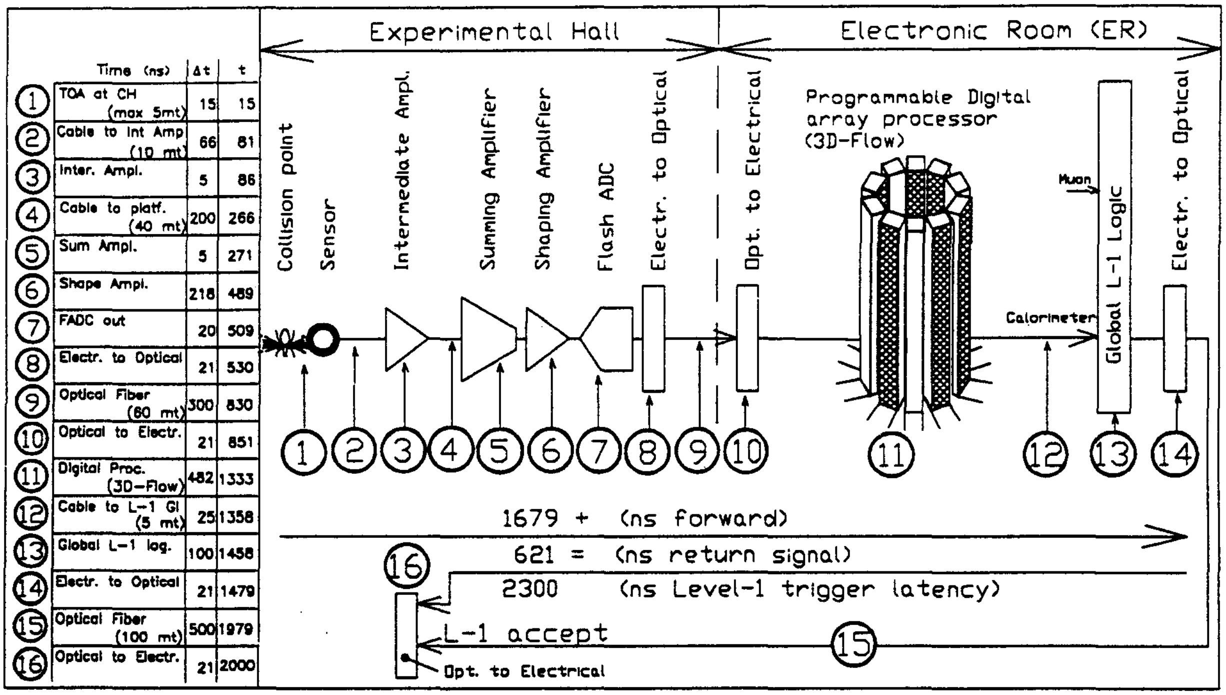
ONE TOWER = ONE PROCESSOR

I level trigger granularity		examples
Experiment	Δη ΔΦ	Number of processors
GEM	0.16 x 0.16	1250
SDC	0.1 x 0.1	3584

Figure 41. Calorimeter array versus processor array.

9.1.3 Logical Layout of GEM Calorimeter Level-1 Trigger

The logical layout of the Level-1 trigger is shown in Figure 42 for the GEM experiment calorimeter. The digital Level-1 trigger array is located in the electronic room (ER). Digitized trigger sum signals (4 × EM signals, HAD1, and HAD2 signals) arrive from the calorimeter readout boards through optical fibers approximately 850 ns after the beam crossing at a rate of 60 MHz. The signals arriving from the calorimeter are converted from optical to electrical, processed in the 3D-Flow programmable parallel-processing system, and then sent to the global Level-1 trigger. A schematic view of the system is presented in Figures 43-46.



	Time (ns)	Δt	t
1	TOA at CH (max 5mt)	15	15
2	Cable to Int Amp (10 mt)	66	81
3	Inter. Ampl.	5	86
4	Cable to platf. (40 mt)	200	266
5	Sum Ampl.	5	271
6	Shape Ampl.	218	489
7	FADC out	20	509
8	Electr. to Optical	21	530
9	Optical Fiber (60 mt)	300	830
10	Optical to Electr.	21	851
11	Digital Proc. (3D-Flow)	482	1333
12	Cable to L-1 Gl (5 mt)	25	1358
13	Global L-1 log.	100	1458
14	Electr. to Optical	21	1479
15	Optical Fiber (100 mt)	500	1979
16	Optical to Electr.	21	2000

Figure 42. Logical layout of the GEM programmable digital Level-1 trigger.

9.1.4 3D-Flow Architecture for Calorimeter Level-1 Trigger

The algorithm of pattern recognition (particle identification) is executed on each processor without boundary limitation. In order to calculate the total energy E_t; the missing E_t: the transverse energy E_x, E_y; the number of electromagnatic sums EMs (EM = 0.16 × 0.16); the number of single particles SPs = EM+HAD (SP = 0.32 × 0.32); and the number of JET, the 3D-Flow processor array is segmented in a superblock of processors (8 × 8), and results for that area are routed to a single processor, which will be the exit point of the results for that superblock. The timing information is shown in Figure 45. The first five stages of the 3D-Flow processor array do not have connections with neighboring processors, but only Top to Bottom connections. They apply a digital filter algorithm while digitizing signals coming from the calorimeter. Each parallel-processing array stage has a time slot for algorithm execution of the data from a single event. Data from different events are sent to different stages at the bunch-crossing rate. Figure 46 shows the flow of the data among 3D-Flow processors of different stages and among the same stages, the type of filtering operation executed in the first five stages, and the particle identification parameters calculated in the subsequent stages.

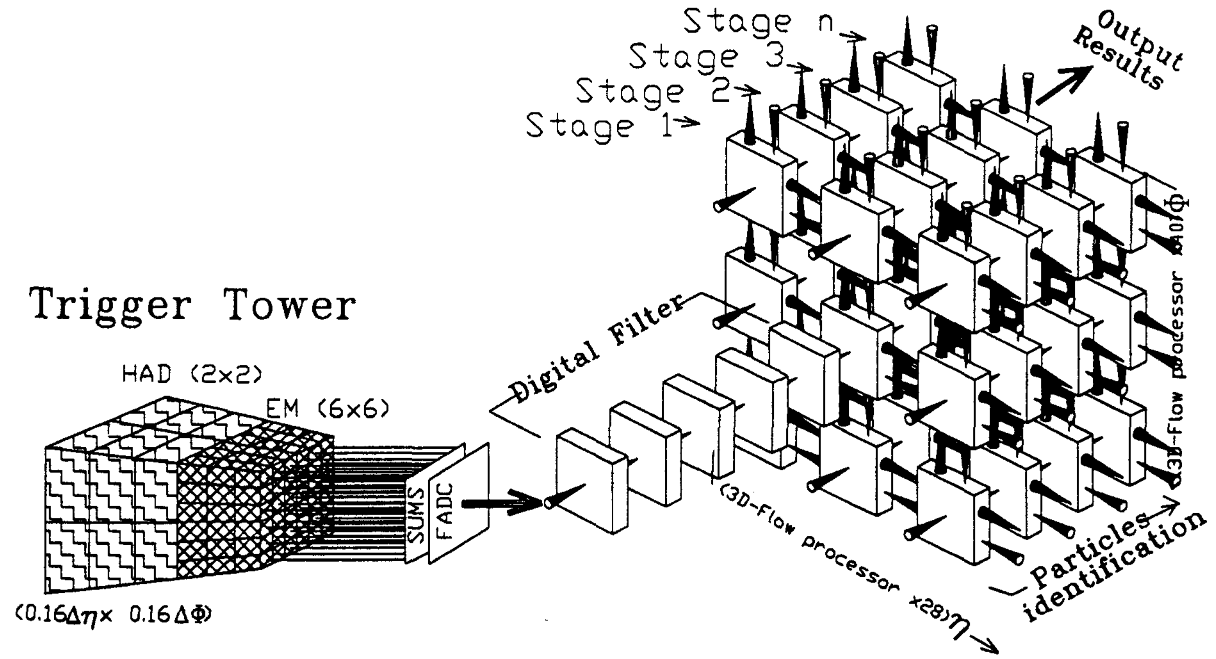


Figure 43. Connection between the calorimeter trigger towers and the 3D-Flow array after analog summing and digitization by FADCs.

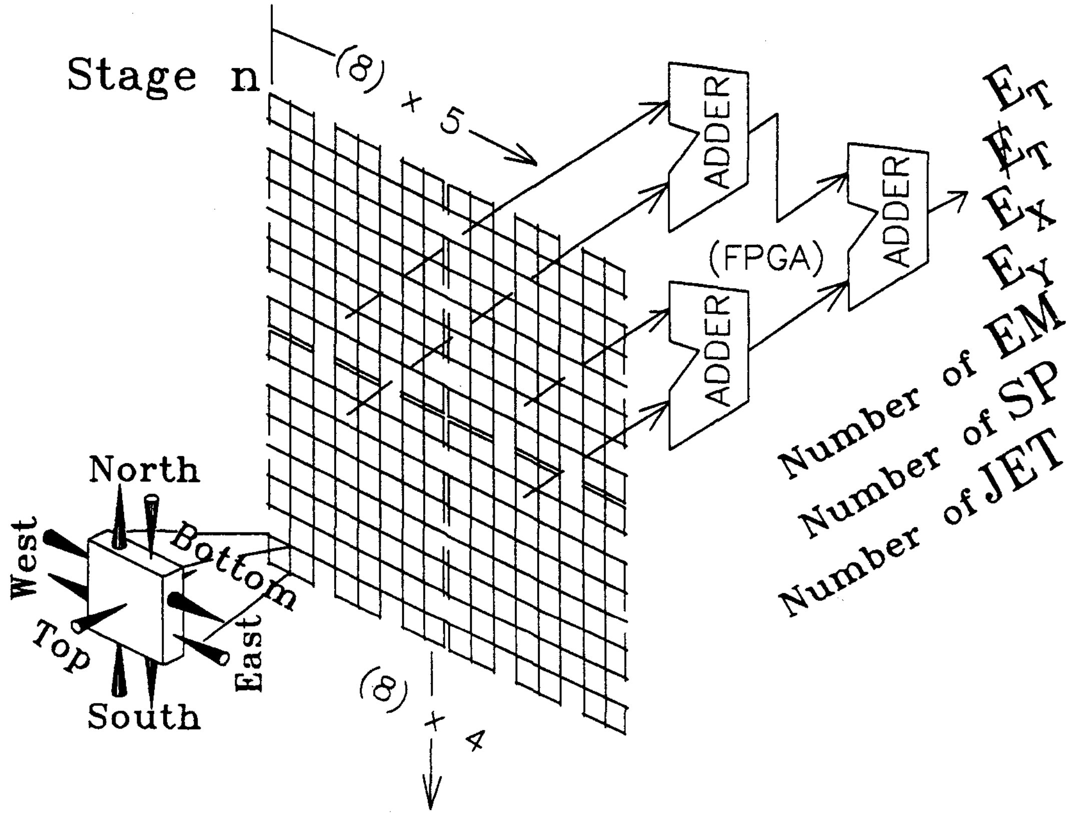


Figure 44. Calculation of the superblock and global information. Each processor is connected to its neighbors, as in the physical layout of the elements of the calorimeter. In order to calculate the global sums, the 3D-Flow processor array is segmented into 8 × 8 superblocks, and the results for a given block are routed to a single processor that becomes the exit point for that block to the gate array system.

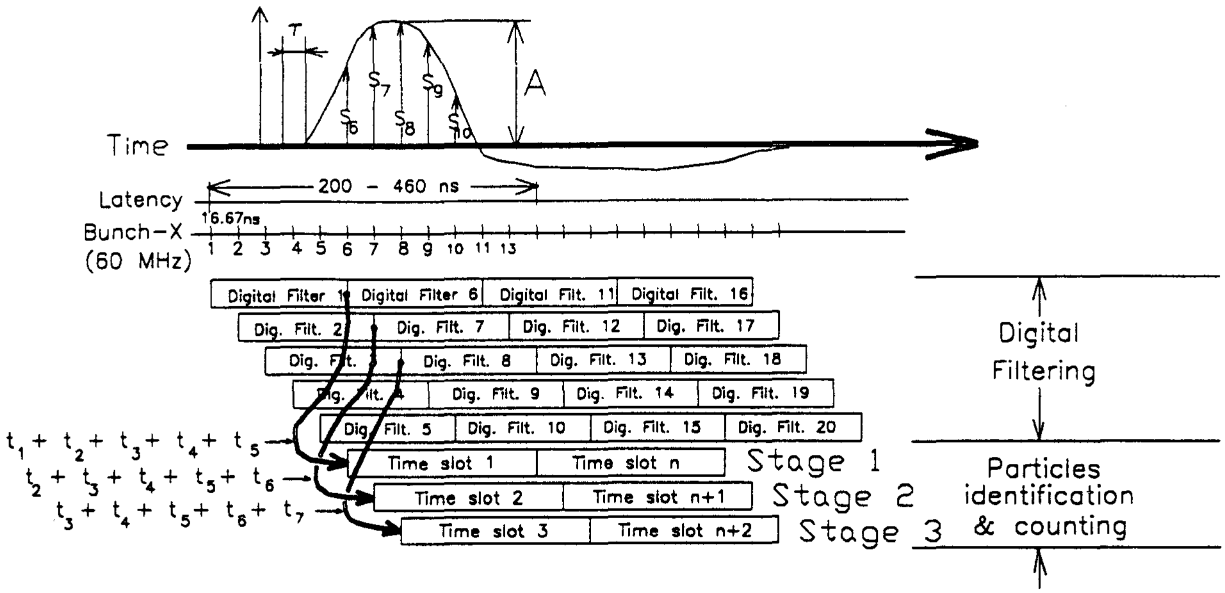


Figure 45. **Timing of the pulse samples and the processor calculations. The first five stages of the 3D-Flow processor array are used to apply the digital filter algorithm. Each subsequent array stage has a time slot for a pattern recognition algorithm execution of the data from a single event.**

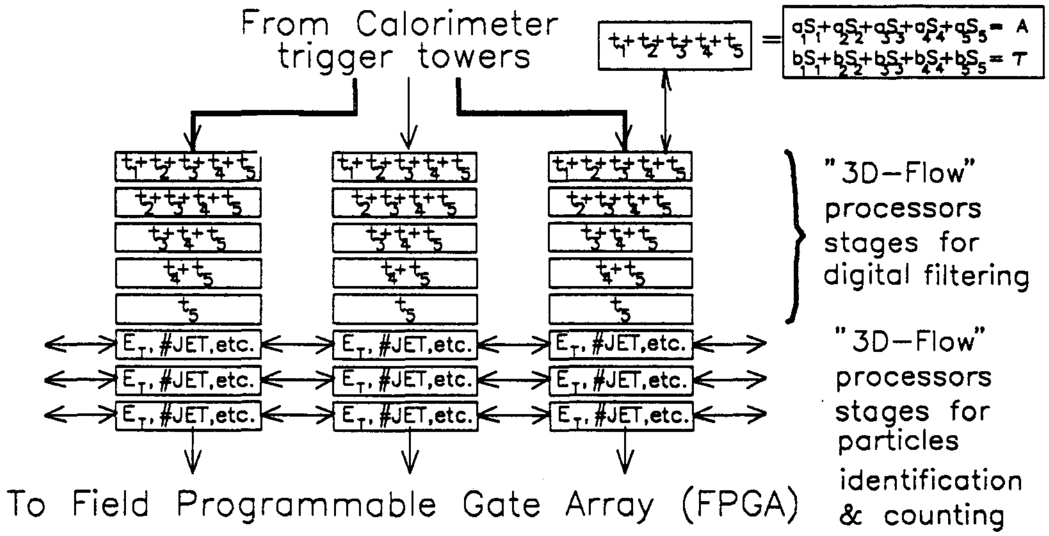

From Calorimeter trigger towers
$t_1+t_2+t_3+t_4+t_5$
$a_1S_1+a_2S_2+a_3S_3+a_4S_4+a_5S_5= A$
$b_1S_1+b_2S_2+b_3S_3+b_4S_4+b_5S_5= T$
$t_1+t_2+t_3+t_4+t_5$
$t_2+t_3+t_4+t_5$
$t_3+t_4+t_5$
t_4+t_5
t_5
E_T, #JET,etc.
"3D-Flow" processors stages for digital filtering
"3D-Flow" processors stages for particles identification & counting
To Field Programmable Gate Array (FPGA)

Figure 46. **Flow of the data among the 3D-Flow processors. As in Figure 45, the processing is broken into two parts, for application of the digital filtering and for particle identification and counting.**

The program execution at stage 1 (see Figure 43) must not only route the new incoming data from the calorimeter to the next stage in the pipeline (stage 2), but must also execute its trigger algorithm. It then sends its results to the stage 2 processor, which passes it on. At this point the stage 1 processor begins to re-execute its algorithm, receiving the electromagnetic and hadronic values from the calorimeter and processing those values. The output results from all processors flow (like the input data) through the different processor stages. The last processor outputs the results from all processors at a rate of 60 MHz.

9.1.5 Input Data Rate Reduction with the 3D-Flow System

The reduction of the input data rate is the result of the parallel-processing calculation (pattern recognition) on the input data. The interesting particles (or detected objects in a graphic application) are very few compared to the total number of input data.

Each processor is executing the pattern recognition algorithm among its neighboring elements without boundary limitation, but only a few will have a positive result. The expected number of interesting particles found may be well known for each experiment from Monte Carlo simulations, and this will determine the design of the output interface to the 3D-Flow parallel-processing system in order to sustain a high or low output data rate.

System layout and program examples for pattern recognition in high energy physics applications with their results are reported in SSCL-576 [ref. 7] and SSCL-607s.[ref. 5]. These examples illustrate how to implement a digital filter, and they present the program listing of local maximum, cluster-finding, and particle identification algorithms.

9.1.6 Input Channels Reduction with the 3D-Flow System

Since the output data rate expected is lower than the input data rate, a reduction of the number of channels can also be accomplished internally to the 3D-Flow parallel-processing system by routing data from one area of the 3D-Flow cell (called superblock) to an exit point. Simulations have been made [ref. 7] for a case of reduction of 64 channels to one exit point. As shown in Figure 47 and 48, only eight cycles of data movement in an area of 64 3D-Flow processor cells will be required to gather information from all of them and send it to an exit point.

The routing of the data between cells has been checked with a simulator to verify that there is no deadlock in communication. Examples of programs that gather information from an area of 8 × 8 processors (or trigger towers) are also given in ref. 7.

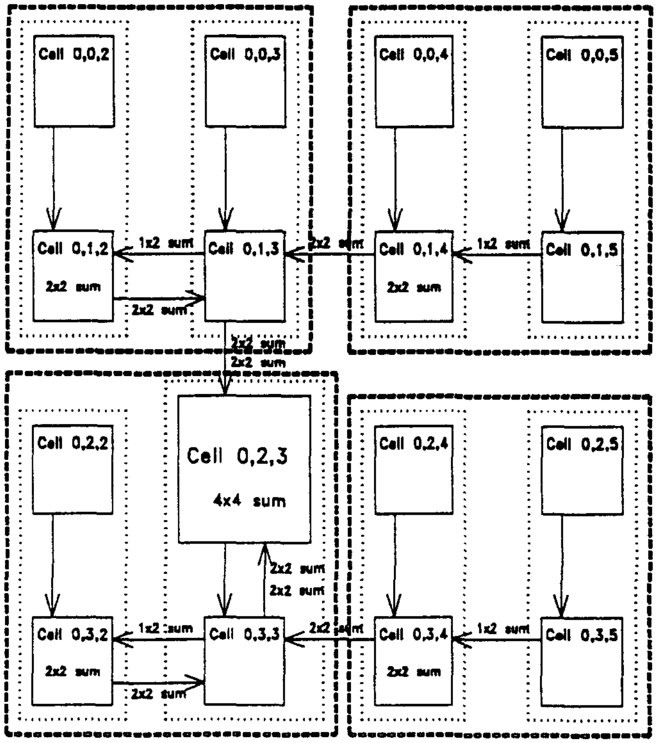


Figure 47. Routing 4 × 4 "data-cells." (Requires four moving cycles)

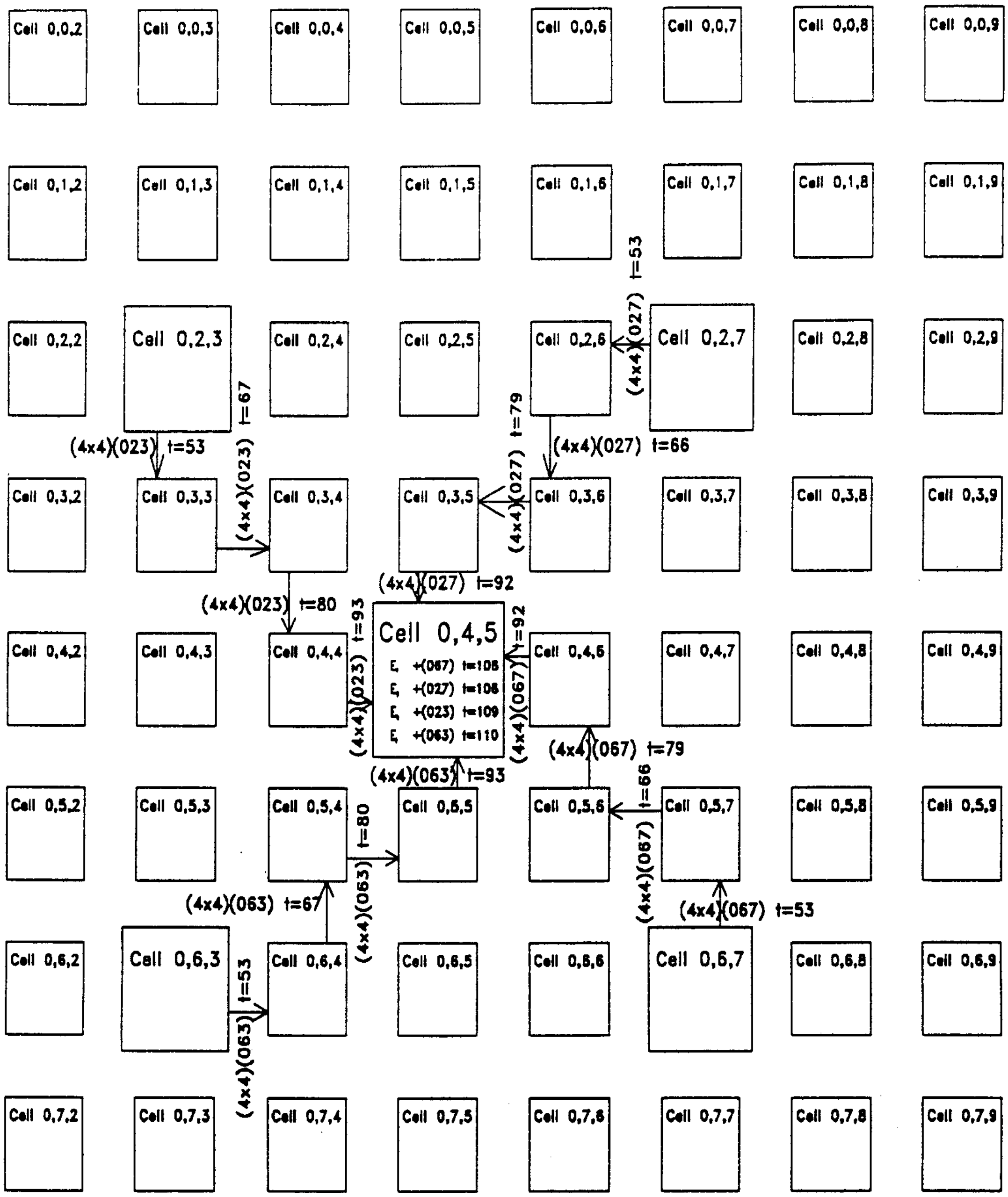
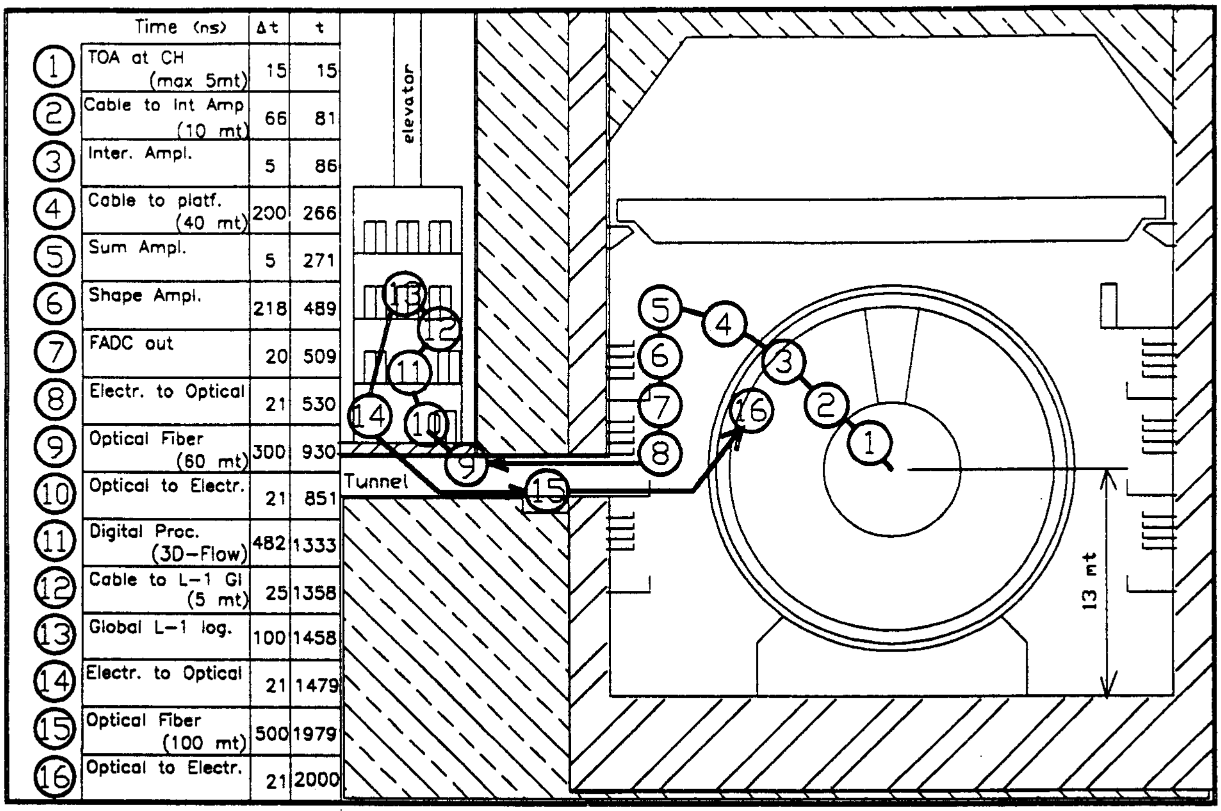


Figure 48. Routing 8 × 8 "data-cells." (Requires four additional moving cycles after operations shown in Fig. 47)

9.1.7 Physical Layout of the GEM Calorimeter Level-1 Trigger

The physical layout of the Level-1 trigger is shown in Figure 49.



	Time (ns)	Δt	t
1	TOA at CH (max 5mt)	15	15
2	Cable to Int Amp (10 mt)	66	81
3	Inter. Ampl.	5	86
4	Cable to platf. (40 mt)	200	266
5	Sum Ampl.	5	271
6	Shape Ampl.	218	489
7	FADC out	20	509
8	Electr. to Optical	21	530
9	Optical Fiber (60 mt)	300	930
10	Optical to Electr.	21	851
11	Digital Proc. (3D-Flow)	482	1333
12	Cable to L-1 Gl (5 mt)	25	1358
13	Global L-1 log.	100	1458
14	Electr. to Optical	21	1479
15	Optical Fiber (100 mt)	500	1979
16	Optical to Electr.	21	2000

Figure 49. Physical layout of the GEM calorimeter level-1 trigger.

9.1.8 Example of a Level-1 Trigger Algorithm Executed by the 3D-Flow Processor

What follows is an example of how to execute a Level-1 trigger algorithm with the 3D-Flow processor. The algorithm, executed in each 3D-Flow processor (corresponding to a trigger tower calorimeter element), compares the energy with several thresholds, calculates front-to-back and precalculates the data to be sent to the neighbor for the calculation of the total energy E_t and the transverse energy, E_x and E_y.

The program executing the algorithm can be modified according to one's needs: to compare the energy of one element to several thresholds, or the sum of 1×2 or 2×1 elements with different thresholds: to calculate front-to-back [Had/Had+EM] (instead of [Had/EM], as shown in this example): and to implement different isolation techniques: the one used at LHC (subtract a small area from a larger area of the calorimeter), or the one suggested by SDC, or the one suggested by the analog Level-1 trigger option for GEM. (Compare each element with its neighbor and consider at first a local maximum if all neighbors are below a given threshold except one, *etc.*).

The execution of this algorithm aims to demonstrate the compactness of the program (very few lines of code) running on the 3D-Flow, its multioperation feature and efficiency in performing the two functions of data moving and data processing necessary for the Level-1 trigger, and the possibility to analyze all signals of the internal operation of the 3D-Flow processor and the possibility provided by the coding in VHDL of relating the timing of hundreds of internal signals at high resolution. Industry has confirmed that the above 6000 lines of VHDL coding can be translated into silicon.

Let us consider the example of implementing an algorithm that can receive data from the calorimeter: convert compressed 8-bit data into linearized 12-bit values: calculate E_t, E_x, E_y: calculate front-to-back

[Had/EM]; and compare each of these calculated values with eight different thresholds. For a first version of the 3D-Flow operating at 60 MHz, we assume the following:

> The calorimeter would have 1280 trigger towers. This number applies for GEM and D0. Each trigger tower would be mapped to one 3D-Flow processor and would send two compressed 8-bit digital values (one representing the energy value of the hadronic compartment, and one representing the energy of the electromagnetic compartment) for each event. Most of the time the event trigger at level-1 would correspond to the bunch crossing of the collider.

To execute this algorithm for an experiment such as D0 or CDF, running at the proposed upgrade of 7.57 MHz (among the different Tevatron upgrade options, this would be the highest frequency of operation), 640 × 3D-Flow chips will be required. (The algorithm requires 12 steps × 16.6 ns per step because the speed of the 3D-Flow is 60 MHz, and because the total time of 192 ns exceeds the 132-ns bunch crossing time. Two layers of 3D-Flow boards are necessary for a zero dead-time system.) Later incremental upgrading of the system to implement more complex algorithms (such as isolation) will be possible by adding 3D-Flow daughterboards to the system.

9.1.8.1 Description of an Example of a Level-1 Trigger Algorithm

The following provides an example of a "typical" Level-1 trigger algorithm that may be implemented in the case of a calorimeter trigger. It should be made clear that this algorithm may be modified to fit other requirements.

1. Get the energy value of the hadronic compartment from the calorimeter.
2. Get the energy value of the electromagnetic compartment from the calorimeter.
3. In order to detect the hits at the border of a calorimeter element, add the electromagnetic energy value to the energy value of the North element and compare it with eight different thresholds. Encode the result of the comparison in 3-bit value. Perform the same operations with the element to the East.
4. Add the hadronic energy value with the energy value of the North element. Do likewise for the element to the East.
5. Check the ratio between the energy found in the hadronic compartment divided by the energy found in the electromagnetic compartment. (Perform two divisions, one relative to the element with its North and the other with the element and its East.) Compare the result of the two divisions with two sets of eight thresholds, encode the result of the comparison in two sets of 3-bit each. Among all these comparisons, one would like also to set a criterion. For example, if one of the four results found is greater than a threshold, then a flag is set to indicate that it is a possible electron candidate; it is passed on to the part of the algorithm that checks for isolation.
6. Add the energy value received from the electromagnetic and hadronic compartments to obtain the total energy calculation and for the successive operations on transverse energy. Multiply the previous result by a second constant in order to find the "x" component of the transverse energy; multiply it by a third constant to find the "y" component.
7. Send to the neighboring processors of the same array the values of the local E_t, E_x, and E_y calculated in the previous steps. Send to the output port (Bottom port of the 3D-Flow) the 4 × 3-bit encoded value of the comparisons.

9.1.8.2 Description of the 3D-Flow Operations Implementing the Above Algorithm

Figure 50. shows the 3D-Flow steps to implement the previous algorithm example

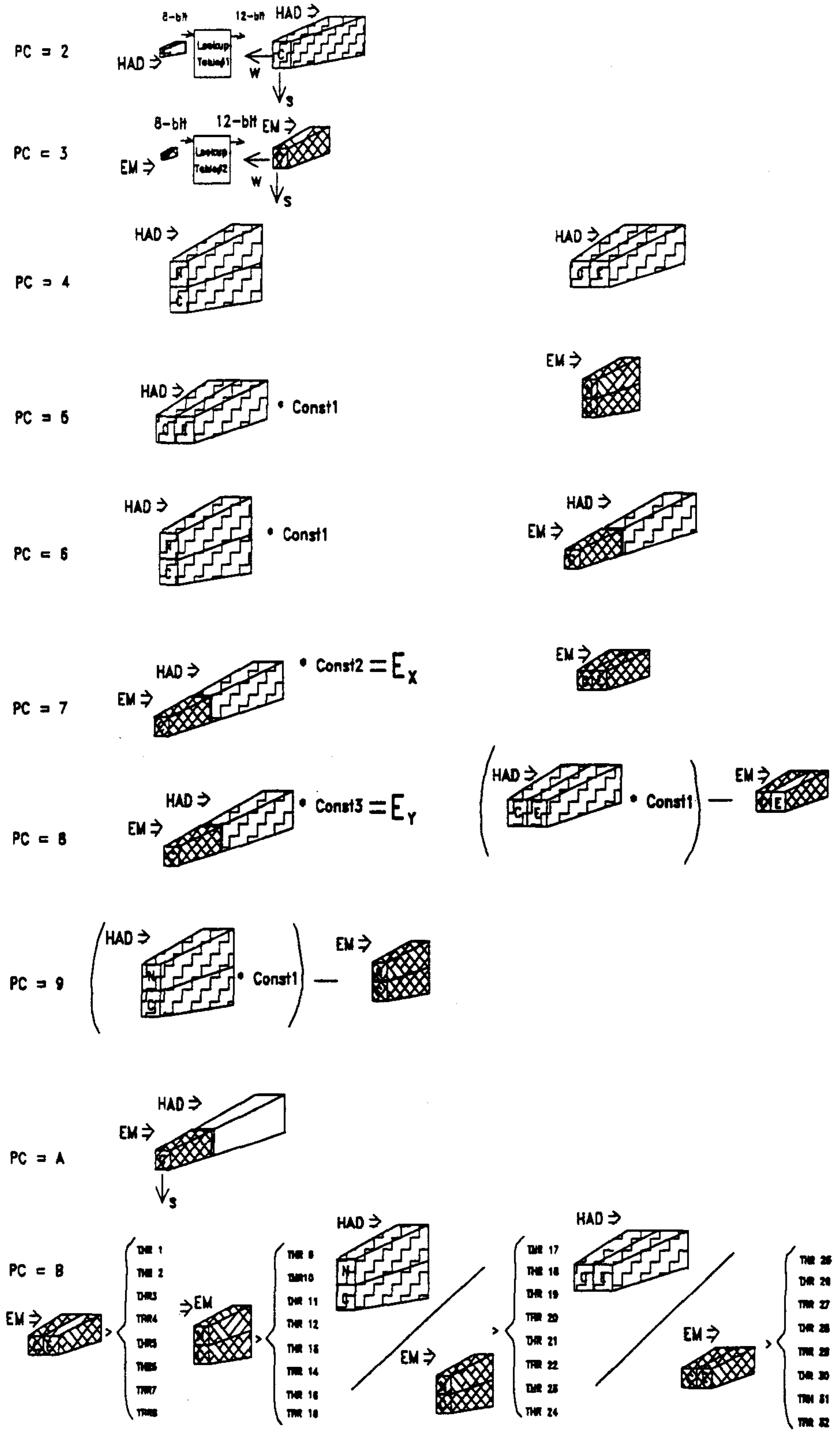


Figure 50. 3D-Flow steps to implement the Level-1 trigger algorithm example.

Table 3 shows the input condition of the 3D-Flow chip at each cycle provided to the 3D-Flow VHDL simulator. The following steps (corresponding to 3D-Flow cycles), are shown in Figure 50 and Table 4.

PC=2. Get the energy value of the hadronic compartment from the calorimeter (Top port), convert it via a look-up table from 8-bit compressed data to 12-bit linearized data, send this value to the 3D-Flow processors to the South and to the West, and store in the internal register "R23"

Instruction: R23=S=W=lookup1<-T.

PC=3. Get the energy value of the hadronic compartment from the calorimeter, convert it via a look-up table from 8-bit compressed data to 12-bit linearized data, send this value to the 3D-Flow processors to the South and to the West, and store in the internal register "R23".

Instruction: R7=S=W=lookup2<-T.

PC=4. Since all processors in the processor array have performed the previous operation, each of them has a data present in the FIFO at the East port and in the FIFO at the North port. Consequently, each processor gets the North-neighbor hadronic energy value (which is the first data that has been sent in the first step to the South). Add it to the original hadronic compartment energy value received from the Top port.

Instruction: R8=R23+N.

Perform the same operation for the East-neighbor hadronic energy value.

Instruction: R19=R23+E.

PC=5. Division is too time-consuming to be performed in these types of real-time calculations. Knowing that the ratio between the electromagnetic and hadronic energies should be about 10 in order to be a likely electron candidate, we can multiply the hadronic energy by a constant (=10 in this case). Perform the above operation with the result of the North-neighbor hadronic compartment energy sums.

Instruction: R30=R19*Const1.

During the same cycle, fetch the North-neighbor electromagnetic energy value (which is the second data that has been sent in the second step to the South). Add it to the original electromagnetic energy value received from the Top port.

Instruction: R4=R7+N.

PC=6. Repeat the same part of the first calculation of the previous line for east-neighbor-hadronic energy sum.

Instruction: R14=R8*Const1.

During the same cycle, add the energy values originally received from the calorimeter through the Top port in order to calculate E_t.

Instruction: R22=R7+R23.

PC=7. Multiply the previous result [EM+HAD] by the second constant in order to find the "x" component of the transverse energy.

Instruction: R15=R22*Const2.

Likewise in the same cycle, get the East-neighbor electromagnetic energy value (which is the second data that has been sent in the second step to the West). Add it to the original electromagnetic energy value received from the Top port.

Instruction: R12=R7+E.

PC=8. Multiply the result [EM+HAD] times the third constant in order to find the "y' component of the transverse energy.

Instruction: R16=R22*Const3.

Execute the difference for the East-neighbor sum [(Had*Const1)-EM], which is equivalent to a division in real-time processing calculation.

Instruction: R28=R30-R12.

PC=9. Repeat the above for the North neighbor sum.

Instruction: R20=R14-R4.

PC=A. Send the E_X values to the neighbors.

PC=B. Compare the result of line 5 (R4) to eight different thresholds and encode the result of the comparison in 3-bit. Perform the same operation on the result of line 7 (R12), line 8 (R28) and line 9 (R20). Use the result of the comparison with the value of threshold 6 in order to set the flag for the next branch operation.

Instruction: CMP R6=(R4,R12,R20,R28)-THR6.

PC=C. Depending on the positive or negative result of the comparison in the previous operation, proceed to the next line of the program, or else branch to other path. For simplicity, here we continue or we branch to the beginning of the program in order to test that the condition code set in the previous instruction is working properly. In case the isolation check has to be made, this branch will eventually execute the isolation algorithm.

Instruction: BPLCO Loop1.

During the same cycle, the result of the comparison (the 4 sets of 3-bit encoded values) is sent out through the Bottom output port.

Instruction B=R6.

Table 4 shows the actual operations executed at each cycle by the 3D-Flow processor. Figure 51 shows the timing of the most important internal signal of the 3D-Flow during program execution. (Note the value of the program counter at the row "U1/mapmctl" and the hexadecimal value of all the core buses A, B, C, D, and the ring buses Ring A, Ring B, Ring C.)

Table 3. Input data to the 3D-Flow processor.

CLOCK #	TOP_FIFO	NORTH_FIFO	EAST_FIFO	WEST_FIFO	SOUTH_FIFO	Ext_Bfull	Ext_Nfull	Ext_Efull	Ext_Wfull	Ext_Sfull
	bit-11	bit-11	bit-11	bit-11	bit-11					
0000	006			x	x	0	0	0	0	0
0001	055	004	001	x	x	0	0	0	0	0
0002	x	300		x	x	0	0	0	0	0
0008	x	x	060	x	x	0	0	0	0	0
0009	x	x	x	x	x	0	0	0	0	0
0010	x	x	x	x	x	0	0	0	0	0
0011	x	x	x	x	x	0	0	0	0	0
0012	x	x	x	x	x	0	1	0	0	0
0013	006	x	x	x	x	0	0	0	0	0
0014	055	x	x	x	x	0	0	0	0	0
0015	x	x	x	x	x	0	0	0	0	0

Table 4. Example of a simple Level-1 trigger algorithm for calorimeter detector with the 3D-Flow processor.

PC= map mctl	Label	Operation No. 1	Operation No. 2	Comment OP No.1	Comment OP No. 2
				receive 8-bit "had" from cal. a	Clear ACC2
02	Loop1:	R23=S=W=lookup1<-T,	ACC2=0. ACC1=0	and convert to lin. 12-bit	and ACC1
				receive 8-bit "em" from cal.	
03		R7=S=W=lookup2<-T		and convert to lin. 12-bit	
04	Loop2:	R8=R23+N.	R19=R23+E	North 1 x 2 "had" sum	East 2 x 1 "had" sum
05		R30=R19*Const1.	R4=R7+N	2 x 1 "had" * Const1	1 x 2 "em"
06		R14=R8*Const1.	R22=R7+R23		1 x 2 "had" * Const1 and Etot
07		R15=R22*Const2.	R12=R7+E		Ex and 2 x 1 "em"
08		R16=R22*Const3.	R28=R30-R12		Ey and front-to-back 2 x 1
09		R20=R14-R4			front-to-back 1 x 2
0A	Ex	S=R15		send Ex to neighbor	
				compare (1x2 "em"). (2x1 "em").	(f-to-b 1x2). (f-to-b 2x1) to 32 thresholds
0B		CMP R6=(R4.R12.R20.R28)-THR6		Set condition code flag if test passes threshold 6.	Store encoded result (3-bit for each comparison set) in register R6
0C		BPLCO Loop1	B=R6		Send result to Bottom
					Port
0D		BRA Loop2			

Note 1:At line of PC=B of the program example in Table 4. in a single operation 32 threshold values are compared in blocks of eight to four different precalculated values. Table 5 shows the values loaded before program execution into the look-up table data memories. Table 6 shows the values of the four sets of eight different thresholds loaded into the threshold registers.

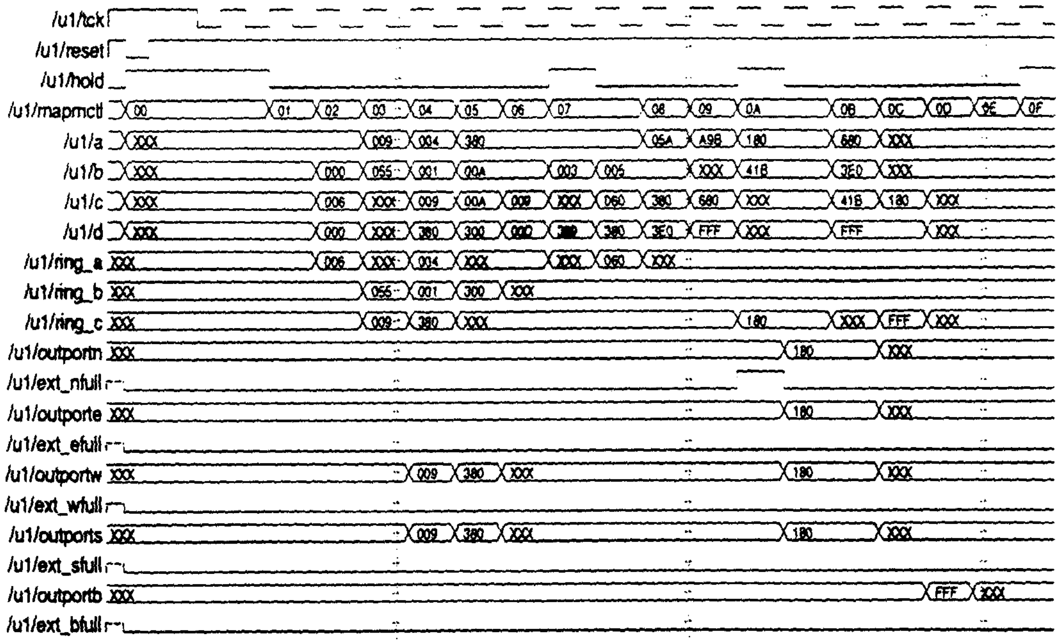


Figure 51. **Timing diagram of the main signals (and buses) of the 3D-Flow processor generated by the VHDL simulator.**

Note 2: At cycle "07" the signal "U1/hold" indicates that the program is in hold, because in Table 3 one could see that the data from the East port is available only at cycle 8 while it should be available at cycle 7. (This will cause the data-driven processor 3D-Flow to wait for the data to arrive at the East port.) Note in the timing of Figure 51 that the signal "U/1hold" is asserted also at cycle "0A." This is because the 3D-Flow processor would like to execute the instruction of writing to the West port, but at that port the FIFO is full.

Table 5. **Four sets of 8 × threshold values to be compared in one cycle of the 3D-Flow processor against four unknown values.**

	THx_MEM(0) bit-11	THx_MEM(1) bit-11	THx_MEM(2) bit-11	THx_MEM(3) bit-11	THx_MEM(4) bit-11	THx_MEM(5) bit-11	THx_MEM(6) bit-11	THx_MEM(7) bit-11
THA_MEM(x)	000	001	002	003	004	005	006	007
TH_MEM (0-7) THB_MEM(x)	000	001	002	003	004	005	006	007
TH_MEM (8-15) THC_MEM(x)	000	001	002	003	004	005	006	007
TH_MEM (16-23) THD_MEM(x)	000	001	002	003	004	005	006	007
TH_MEM (24-31)								
	Count_In	Count_bypass						
Ld_counter	002	006						

Table 6. **3D-Flow data memories content to be used in the calorimeter algorithm as look-up tables to convert compressed 8-bit input data into 12-bit linearized values. (The table shows only partial data.)**

	Data_ Memory_1					Data_ Memory_2		
	bit (11-8)	bit (7-4)	bit (3-0)			bit (11-8)	bit (7-4)	bit (3-0)
Addr.000	0000	0000	0000		Addr.000	0000	0000	0000
Addr.001	0000	0000	0001		Addr.001	0000	0000	0001
Addr.002	0000	0000	0010		Addr.002	0000	0000	0010

9.2 Real-Time Track Finding and Rejection

The problem of real-time track-finding has been performed to date with CAM (Content Addressable Memories) or with fast coincidence logic, because the processing scheme was thought to have much slower performance. Advances in technology together with a new architectural approach make it feasible to also explore the computing technique for real-time track finding thus giving the advantages of implementing algorithms that can find more parameters such as calculate the sagitta, curvature, pt, etc. with respect to the CAM approach.

The look-up table technique has a very fast response, but it requires a large amount of memory and is limited to recognizing only tracks that have been prerecorded into the memory.

The following approach of real-time tracking with the 3D-Flow parallel-processing system offers a fast and programmable response that may solve the problem in some real-time tracking applications.

Figure 52 depicts how information from different sub-detectors is sent into the 3D-Flow parallel-processing system.

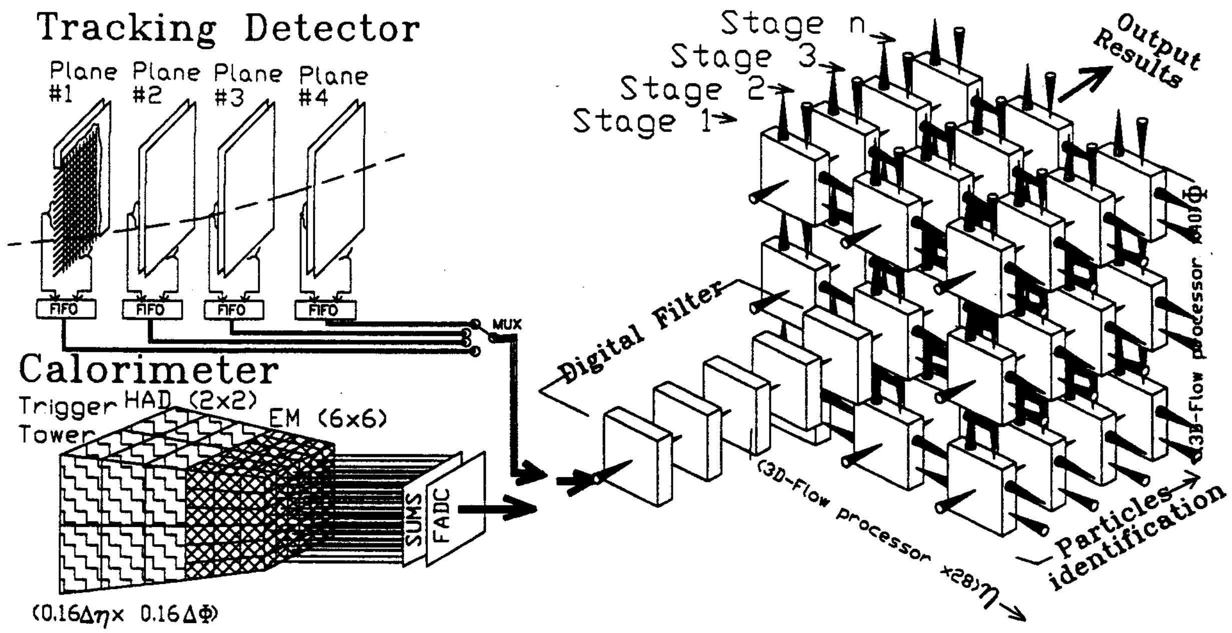


Figure 52. 3D-Flow system receiving data from different sub-detectors.

In the track-finding application, a number of 3D-Flow processors are used for each "plane." Depending on the complexity of the algorithm and the number of tracks expected in a given area, the user decides the most convenient price/performance segmentation of the "plane" in smaller areas, each of which sends the information to a 3D-Flow processor.

As an illustrative example, Figure 53 shows the mapping of the strip (wire) signals to a 3D-Flow processor array, while Figure 54 shows the mapping of the signals from a subset of a tracking detector into a 3D-flow processor. Thus if we have a "plane" (consisting of several subplanes "x", "y", "u", "v") of 512 wires or strips and we know from Monte Carlo simulation that the number of expected tracks is not greater than 10, then a convenient segmentation would be an 11 × 11 3D-Flow processor array for each "plane," such that each one receives as input a small fraction of information of the entire plane (e.g., 48 wires or strips of each subplane).

We assume that an approximate vertex point has been located in a first step of the Level-1 tracking program [ref. 30]. For each detector plane there is a 2-dimensional 3D-Flow processor array; for successive detector planes there are successive arrays (or stages).

Each 3D-Flow processor takes the x and y coordinates from a hit on the first plane. and computes the predicted coordinates on the next plane by a straight-line extrapolation. If curved tracks are expected in one or two dimensions. the processor in the next array should look for a hit in a wider region of interest. In the next plane. and in the corresponding small area. the 3D-Flow processor checks whether the predicted x and y coordinates lie in its region of operation. If so. the processor should find a hit. which may come close to a straight-line predicted value (or deviates by a relatively small amount if a curvature is expected). The processor calculates. for this track. the new slopes (in x and y). the sagitta. the momentum (P). and the transverse momentum (or its y-component). The results of the calculation are passed on to the 3D-Flow processor, which will operate on the corresponding area element in the next plane.

If the calculation to see whether the predicted x. y coordinate pair lies in the operating region of the individual 3D-Flow processor shows that it does not. the processor forwards the received quantities to the adjacent 3D-Flow processor in the same array (or stage). The processor that finds that the predicted coordinates match its operating area then checks for continuity of the track in that plane by searching for a hit in its region. If the hit is found. the processor calculates the momentum. etc., and the result is forwarded to the next processor array (or stage). and so on.

9.2.1 Tracking Detector Versus 3D-Flow Processor Array

The tracking detector versus the 3D-Flow processor array is shown in Figure 53.

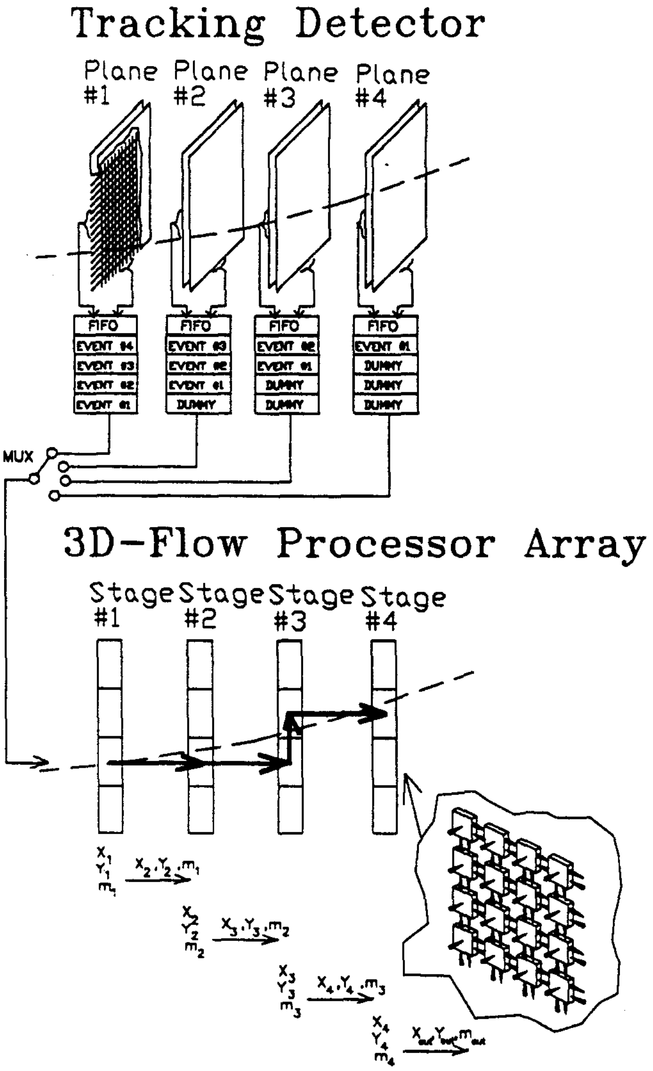


Figure 53. Tracking detector versus 3D-Flow processor array.

The signal from the wires of the central plane are sent to the input of all 3D-Flow processors of the second column, as the signals from the wires of the other planes are sent to the other processors as shown in Figure 54.

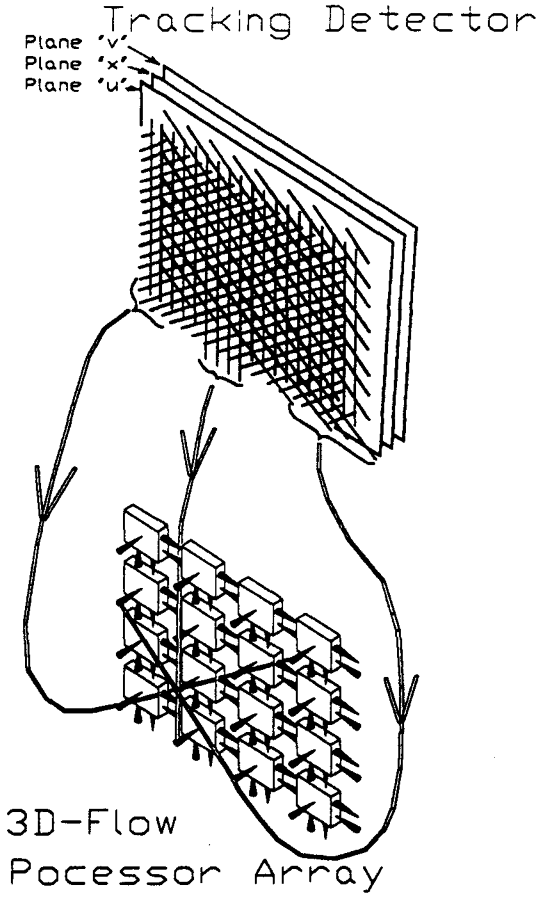


Figure 54. Mapping of signals from a subset of a tracking detector into a 3D-Flow processor.

9.2.2 Timing and Synchronization

Depending on the amount of computing required to calculate the unknown parameters, and the number of hits per plane, the user selects an appropriate segmentation of the plane and associates it to a 3D-Flow processor array. Note that the high communication speed of the 3D-Flow processor allows the exchange of data between adjacent areas, thus allowing a system with no boundary limitation.

In Table 7 the four columns represent the activity of the processors in the four arrays (or stages); the rows indicate (from top to bottom) the timing sequence; the activity at each timing sequence for the arrays is indicated in the corresponding row.

Since each processor has the capability to simultaneously move data and perform calculations, two columns have been reserved for each processor array in order to indicate these activities. For example, row "zero" indicates that data event #1, from detector plane #1, is moved to processor stage (or array) #1; and row "one" indicates that the received data of event #1, from plane #1, is processed in the processor stage (or array) #1, at the same time that the processor is receiving the data of event #2 from plane #1, and so on for row "three," *etc*.

Following this sequence, by row "eight" the results of event #1 are ready for output. At this time, the pipe is full, and all the processors are performing the two operations of moving and computing on data from different events.

Table 7. **Timing of "data moving" and "data processing" on each 3D-Flow stage. Results are moved simultaneously while computing.**

Time	Detector Plane #1 = 3D-Flow array # 1		Detector Plane # 2 = 3D-Flow array # 2		Detector Plane # 3 = 3D-Flow array # 3		Detector Plane # 4 = 3D-Flow array # 4	
	3D-Flow Processing	3D-Flow Data mover	3D-Flow Processing	3D-Flow Data mover	3D-Flow Processing	3D-Flow Data mover	3D-Flow Processing	3D-Flow Data mover
0		in EV1-PL1-ST1						
1	Computing EV1-PL1	in EV2-PL1-ST1						
2	Computing EV2-PL1	Res EV1 to ST2 in EV3-PL1-ST1		in EV1-PL2-ST2				
3	Computing EV3-PL1	Res EV2 to ST2 in EV4-PL1-ST1	Computing EV1-PL2	in EV2-PL2-ST2				
4	Computing EV4-PL1	Res EV3 to ST2 in EV5-PL1-ST1	Computing EV2-PL2	Res EV1 to ST3 in EV3-PL2-ST2		in EV1-PL3-ST3		
5	Computing EV5-PL1	Res EV4 to ST2 in EV6-PL1-ST1	Computing EV3-PL2	Res EV2 to ST3 in EV4-PL2-ST2	Computing EV1-PL3	in EV2-PL3-ST3		
6	Computing EV6-PL1	Res EV5 to ST2 in EV7-PL1-ST1	Computing EV4-PL2	Res EV3 to ST3 in EV5-PL2-ST2	Computing EV2-PL3	Res EV1 to ST4 in EV3-PL3-ST3		in EV1-PL4-ST4
7	Computing EV7-PL1	Res EV6 to ST2 in EV8-PL1-ST1	Computing EV5-PL2	Res EV4 to ST3 in EV6-PL2-ST2	Computing EV3-PL3	Res EV2 to ST4 in EV4-PL3-ST3	Computing EV1-PL4	in EV2-PL4-ST4
8	Computing EV8-PL1	Res EV7 to ST2 in EV9-PL1-ST1	Computing EV6-PL2	Res EV5 to ST3 in EV7-PL2-ST2	Computing EV4-PL3	Res EV3 to ST4 in EV5-PL3-ST3	Computing EV2-PL4	Res EV1 to Out in EV3-PL4-ST4
9	Computing EV9-PL1	Res EV8 to ST2 in EV10-PL1-ST1	Computing EV7-PL2	Res EV6 to ST3 in EV8-PL2-ST2	Computing EV5-PL3	Res EV4 to ST4 in EV6-PL3-ST4	Computing EV3-PL4	Res EV2 to Out in EV4-PL4-ST4
10	Computing EV10-PL1	Res EV9 to ST2 in EV11-PL1-ST1	Computing EV8-PL2	Res EV7 to ST3 in EV9-PL2-ST2	Computing EV6-PL3	Res EV5 to ST4 in EV7-PL3-ST4	Computing EV4-PL4	Res EV3 to Out in EV5-PL4-ST4
	Computing	Res EV10 to ST2	Computing	Res EV8 to ST3	Computing	Res EV6 to ST4	Computing	Res EV4 to Out

In the previous case the applications were optimized for maximum input data rate with higher latency time. This performance is obtained because normally it takes only 20% of the processing time to transfer the results from one 3D-Flow array to the subsequent array. Thus if the transfer of the results takes place during the next 3D-Flow processing time slot, for 80% of the time the 3D-Flow will only compute and will not move data between the ports. The optimal use of the 3D-Flow would be to maintain a 100% utilization of both "data moving" and "computing" simultaneously. But if short latency times are important with respect to the input data rate, then the two operations of "computing" and "moving" the results from one array to the subsequent one can be carried on in sequence, as shown in Table 8. In this case, results are moved from one stage (or array) to the next while the 3D-Flow processors are idle, waiting for data.

Table 8. Timing of "data moving" and "data processing" on each 3D-Flow stage. Results are moved in sequence after computing.

Time	Detector Plane #1 = 3D-Flow array # 1		Detector Plane # 2 = 3D-Flow array # 2		Detector Plane # 3 = 3D-Flow array # 3		Detector Plane # 4 = 3D-Flow array # 4	
	3D-Flow Processing	3D-Flow Data mover	3D-Flow Processing	3D-Flow Data mover	3D-Flow Processing	3D-Flow Data mover	3D-Flow Processing	3D-Flow Data mover
0		in EV1-PL1-ST1						
1	Computing EV1-PL1	in EV2-PL1-ST1		in EV1-PL2-ST2				
2		Res EV1 to ST2						
3	Computing EV2-PL1	in EV3-PL1-ST1	Computing EV1-PL2	in EV2-PL2-ST2		in EV1-PL3-ST3		
4		Res EV2 to ST2		Res EV1 to ST3				
5	Computing EV3-PL1	in EV4-PL1-ST1	Computing EV2-PL2	in EV3-PL2-ST2	Computing EV1-PL3	in EV2-PL3-ST3		in EV1-PL4-ST4
6		Res EV3 to ST2		Res EV2 to ST3		Res EV1 to ST4		
7	Computing EV4-PL1	in EV5-PL1-ST1	Computing EV3-PL2	in EV4-PL2-ST2	Computing EV2-PL3	in EV3-PL3-ST3	Computing EV1-PL4	in EV2-PL4-ST4
8		Res EV4 to ST2		Res EV3 to ST3		Res EV2 to ST4		Res EV1 to Out
9	Computing EV5-PL1	in EV6-PL1-ST1	Computing EV4-PL2	in EV5-PL2-ST2	Computing EV3-PL3	in EV4-PL3-ST3	Computing EV2-PL4	in EV3-PL4-ST4
10		Res EV5 to ST2		Res EV4 to ST3		Res EV3 to ST4		Res EV2 to Out
11	Computing EV6-PL1	in EV7-PL1-ST1	Computing EV5-PL2	in EV6-PL2-ST2	Computing EV4-PL3	in EV5-PL3-ST3	Computing EV3-PL4	in EV4-PL4-ST4
12		Res EV6 to ST2		Res EV5 to ST3		Res EV4 to ST4		Res EV3 to Out
13	Computing EV7-PL1	in EV8-PL1-ST1	Computing EV6-PL2	in EV7-PL2-ST2	Computing EV5-PL3	in EV6-PL4-ST4	Computing EV4-PL4	in EV5-PL4-ST4
14		Res EV7 to ST2		Res EV6 to ST3		Res EV5 to ST4		Res EV4 to Out

10.0 COST CONSIDERATIONS

The core of the 3D-Flow system is the 3D-Flow chip and the 3D-Flow daughterboard (12 cm × 12 cm). The daughterboard (as well as its motherboard) has already been developed. The 3D-Flow chip has been designed and its feasibility has been checked through industry consultants and use of a VHDL simulator.

The total cost of the development of an ASIC chip for other system designs (like the one estimated and under development by SDC and GEM) ranges from $300K per chip for SDC to $640K per chip for GEM. Other systems typically require the development of several boards many times the size of the 3D-Flow daughterboard, with hundred of components, including new ASICs.

In comparison, taking account of the work that has already been done (my personal time, 305 hours for part of the mechanical drawings and 225 hours for checking the 3D-Flow design in a feasibility study with industry consultants), the cost for going to silicon for the 3D-Flow chip will be less than $240K.

For the production cost of a complete Level-1 trigger system, refer to the example described in Section 5.7 of this report. From this example, one could extrapolate the cost for other experiments running under different conditions and with different needs.

The example shows the number of trigger towers and the number of 3D-Flow chips for a Level-1 trigger calorimeter application in D0. A similar application at the SSC, with the same number of trigger towers, can be applied to the Level-1 trigger of the GEM experiment, although in this case the bunch crossing frequency is about 8 times that of the Tevatron for D0. However, the speed of the 3D-Flow chip some years from now will be many times greater than the speed we assume in this example. (The ALPHA CMOS processor from DEC, for example, was delivered at 100 MHz, but it will soon be delivered at 250 MHz. NEC is also producing a DSP processor at 250 MHz.)

In addition to the core 3D-Flow trigger system, for different applications the user has only to design the receiver board and the output board that collects the results of the calculations from the 3D-Flow parallel-processing system.

The comparison is between 640 × 3D-Flow chips (ASICs made of 100K gate at 60 MHz) placed on a 160 × 3D-Flow daughterboard (3U board, 12 cm × 12 cm) for D0 and GEM experiments (1792 × 3D-Flow in the case of the SDC experiment) as against the development of several 9U boards, the development of several new ASICs, and the production of several hundred 9U boards. The latter are described in several reports for the implementation of the Level-1 trigger, one of which is the SDC Note-92-198 [ref. 11].

11.0 CONCLUSIONS

A new approach to a programmable Level-1 trigger has been described based on a study of the requirements of Level-1 triggers of various experiments---past, present, and future. Considering both the physics requirements and the possibility of finding a degree of commonality in the design of circuits for a variety of experiments, this approach had the following goals:

- Design of a circuit (3D-Flow processor) that satisfies physics requirements (implementing different Level-1 trigger algorithms);
- Design of the support electronics (a 3D-Flow daughterboard, motherboard, Mini-Rack, and an interconnection scheme that would satisfy different requirements) to allow flexibility;
- Design of an overall parallel-processing system (an acquisition system and parallel-processing system that would allow mapping of detector elements into parallel-processing elements that guarantee fast communication between the elements) that makes possible fast data communication, low power consumption, and low cost implementation: and

- Demonstration of how the parallel-processing system fits (interfaces) in the overall Level-1 trigger system of different experiments. The flexible assembly system allows for implementation of the above detectors element arrangement in the interconnection topologies of the different processors.

The result is a new approach that, when compared to current and past designs, has a number of important advantages:

1. It is more FLEXIBLE, because of its programmability;
2. It has LOWER COST, in both the development and production phases;
3. It has APPLICATIONS to other problems in addition to the calorimeter trigger problem (e.g., real-time trigger tracking);
4. It is EASILY UPGRADED, incrementally;
5. It is EASILY SCALED, both in size (for different detectors) and in speed (for different experiments running on different colliders at different bunch crossing rates); and
6. It offers a POTENTIAL COMMON SOLUTION to the requirements for several current and future experiments.

These goals have been accomplished at a relatively low cost (see Section 9.0). The Mini-Rack, the 3D-Flow motherboard, 3D-Flow daughterboards, and signal and power supply distribution boards have been built and are being tested. On the one hand, more than 6000 lines of VHDL code has been written to solve the Level-1 trigger algorithm, and on the other hand, proof that the 3D-Flow component is feasible has been verified through industrial collaboration.

The report describes the feasibility of implementing different Level-1 trigger algorithms under different sequences of operation of a 3D-Flow processor (see Section 5.7). The Level-1 trigger algorithm can perform the following operations: after receiving data from the calorimeter, convert compressed 8-bit data into linearized 12-bit values, calculate E_t, E_x, E_y, calculate front-to-back [Had/EM], and compare each of these calculated values with eight different thresholds; or it can be modified to compare the energy of one element to several thresholds, or the sum of 1×2 or 2×1 elements with different thresholds; or it can calculate front-to-back [Had/Had+EM]. It can also implement different isolation techniques: the one used at LHC (subtracting a small area from a larger area of the calorimeter), or the one suggested by SDC, or the one suggested by the analog Level-1 trigger option for GEM (e. g., compare each element with its neighbor and consider at first a local maximum if all neighbors except one are below a given threshold).

The programmability feature affords physicists the opportunity to include among the first-level trigger some algorithms otherwise left for the higher-level trigger; the result might be better rejection of background events. Today, as in the past three years, physicists continue to perform Monte Carlo simulations, and they are continually revising the trigger requirements to improve background rejection from the trigger algorithm. Experience in most cases teaches that the fine tuning of the trigger is best done after analyzing the data first acquired. When this is done, programmability is accepted by a large number of physicists.

Finally, the report proposes a programmable system with the potential to provide a common solution to the requirements of several current and future experiments. It describes how to build a complete Level-1 trigger system and makes comparisons with respect to conventional apparatus, while providing estimates of development and production costs.

ACKNOWLEDGMENTS

I would like to acknowledge A. Montgomery and L. Graw of SSC Technology Transfer for their constant support, and, in particular, A. Montgomery for the interesting and fruitful discussions I had with him. I thank L. Mapelli, M. Shaevitz, V. Radeka, M. Abolins, and D. Edmunds for discussions about calorimeter detectors, W. Selove for discussions of tracking detectors and track-finding algorithms, Prof. A. Werbrouck from Computer Science University of Torino (Italy) for dedicating his time to verifying the functionality of the trigger algorithm on the 3D-Flow, T. Pal and H. Lubatti for discussions about tracking and Time Projection Chamber detectors, and H. Crawford and P. Yepes for discussions about the TPC detectors. Thanks, too, to J. Krebs, S. Beibei, E. Chesi, F. Bal, and E. Zimmer-Nixdorf for their support, to R. Gates for the fast editing of some drawings done mostly during weekends, and to Chuck Hoekstra for his patience in manipulating drawing files of tens of Mbytes that were reaching the limits of the capabilities of good graphic packages. Chuck patiently submitted and monitored jobs, understood the limits of the system, and found ways to get around those limits. I also thank S. Oxley for drawing modeling; he knows everything about graphic formats and can convert any picture. J. Naples and R. Rooney provided invaluable editorial assistance.

REFERENCES

1. CDF- The CDF Collaboration, Proposal For an Upgraded CDF Detector, Oct. 9, 1990, CDF/DOC/PUBLIC/1172.

2. 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.

3. SDC Technical Design Report, SDC-92-201 (SSCL-SR-1215). Solenoidal Detector Collaboration, 1 April 1992.

4. GEM Technical Design Report, GEM-TN-93-262,. SSCL-SR-1219.

5. D. Crosetto, General Programmable Level-1 Trigger with 3D-Flow Assembly System for Calorimeters of Different Sizes and Event Rates. SSCL-607, Dec. 1992

6. IEEE Standard VHDL Language Reference Manual. IEEE Std 1076-1987. Published by The Institute of Electrical and Electronic Engineers, Inc., 345 East 47th Street, New York, New York 10017, USA,. 1987.

7. D. Crosetto and L. Love, "Fully Pipelined and Programmable Level-1 Trigger," SSCL-576,. July 1992.

8. N. Bains et al., "The UA1 upgrade calorimeter trigger processor," *Nuclear Instruments and Methods in Physics Research*, A292 (1990) 401-423.

9. N. Ellis at al., "First-level trigger system for LHC experiments," CERN/DRDC/92/17.

10. T. Gorski, J. Lackley, W. H. Smith, W. Temple, "Isolated Electron Pattern Logic Design and Performance at SSC," SDC Note SDC-91-087, 1991.

11. W. H. Smith, J. Lackey, and T. Gorski, "Revised Level-1 Trigger Decision for the SDC," SDC Note SDC-92-198.

12. The ZEUS Detector Status Report 1989, PRC 87-02.

13. L. Mapelli, "The Challange of Triggering and Data Acquisition at Supercollider Experiments," Nuclear Instruments and Methods in Physics Research, A315 (1992) 460-471.

14. S. Cittolin, "Readout System Test Benches" (CERN/DRDC/92-10, CERN/RD-12)

15. M. Abolins. D. Edmunds. P. Laurens. J. Linnemann. ana B. Pi. The Level-1 Calorimeter Trigger for D0. Oct. 25. 1988. D0-Note-706.

16. F. Dydak and P. Jenni. ATLAS Letter of Intent for a General Purpose pp Experiment at LHC.

17. CMS. The Compact Muon Solenoid. Letter of Intent. CERN/LHCC 92-3. LHCC/I 1. 1 October 1992.

18. M. Contreras et al. "Proposal for the SDC Level-1 Trigger Design or is a Simple L1 Adeguate for SDC ?." SDC note. SDC-91-100.

19. W. E. Cleland and E. G. Stern. "Signal Processing for Liquid Ionization Calorimeters," to be submitted to Nucl. Instr. and Meth.

20. L. Mapelli et al., A Scalable Data Taking System At A Test Beam for LHC. (CERN/DRDC/90-64. CERN/DRDC/92-13).

21 G. Jarlskog and D. Rein. Large Hadron Collider Workshop. CERN 90-10, ECFA 90-133, Vol. I, II. III. Aachen. 4-9 October 1990.

22. W.H. Smith et al., "Isolated electron pattern logic design and performance at SSC." Solenoidal Detector Notes SDC-91-00087. November 11. 1991.

23. D. A. Patterson and J. L. Hennessy. Computer Architecture a Quantitative Approach, Morgan Kaufmann Publisher Inc.

24. H. S. Stone,. High-Performance Computer Architecture,. Addison-Wesley Publishing Company.

25. C. Mead and L. Convav. Introduction to VLSI Systems, Addison-Wesley Publishing Company.

26. D. Crosetto, "A fast cluster finding system for future HEP experiments," *Nuclear Instruments and Methods in Physics Research*, A311, (1992), 49-56.

27. D. Crosetto, "3D-Flow Processor for a Programmable Level-1 Trigger," Computing in High Energy Physics. CHEP92, Annecy, France (1992), 803-806.

28. D. Crosetto. "Calorimeter Programmable Level-1 Trigger," III International Conference on Calorimetry in High Energy Physics. Corpus Christi, Sept. 29-Oct. 2, 1992.

29 D. Crosetto. "3D-Flow Processor for a Programmable Level-1 Trigger." SSCL-Preprint-164. Nuclear Science Simposium (NSS). Medical Imaging Conference (MIC). Orlando, Florida. October 25-31.

30. W. Selove. Impact Parameter Trigger and Vertex Detector for Forward Collider. Univ of Pa. Report UPR-222E, Aug. 31, 1993 Submitted to Proceedings of the Workshop on B Physics at Hadron Accelerators. Snowmass. Colorado. June 1993.