



Fermi National Accelerator Laboratory

FERMILAB-Conf-95/323

**Extending DART to Meet the Data Acquisition Needs
of Future Experiments at Fermilab**

G. Oleynik, R. Pordes and E. Barsotti
For the DART Collaboration

*Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510*

October 1995

Proceedings of the *Computing in High Energy Physics 1995 (CHEP 95)*,
Rio de Janeiro, Brazil, September 18-22, 1995

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Extending DART to meet the Data Acquisition Needs of Future Experiments at Fermilab *

Gene Oleynik, Ruth Pordes, Ed Barsotti

For the DART Collaboration

*Fermilab Computing Division, Fermilab, P.O.Box 500, Batavia,
Illinois 60510*

The DART project at Fermilab is a major collaboration to develop a data acquisition system for multiple experiments. The initial implementation of DART has concentrated on providing working data acquisition systems for the (now eight) collaborating experiments in the next Fixed Target Run. In this paper we discuss aspects of the architecture of DART and how these will allow it to be extended to meet the expected needs of future experiments at Fermilab. We also discuss some ongoing developments within the Fermilab Computing Division towards these new implementations.

1 Introduction

The DART project at Fermilab is a collaboration between the Computing Division and eight experiments to develop a data acquisition system to fit their needs [1,2]. Versions of DART have been used to take data in small experiments in this Collider Run. Some parts of DART are in use at CDF, and all collaborating experiments have working - albeit incomplete - data acquisition systems which they are using for sub-system commissioning and hardware testing.

The DART architecture design is complete and the system itself will be complete for the start of the Fixed Target Run in 1996. While the initial implementation of DART has concentrated on providing working data acquisition systems for the collaborating experiments, the architecture and design of the system are such that the system can be extended in various ways to meet the needs of new data acquisition applications.

2 DART Readout Architecture

The DART hardware architecture is based around a data pipeline over parallel, multi-drop, streams (implemented as a simple data transfer protocol on an RS485 cable) which feed multiple buffer memories in VME. Data is then collected into one or more processors which are directly connected to the VME for software analysis and/or logging.

The data can be moved into the buffer memories in a round robin fashion, or the value of first data word on a stream can be used to select the particular memory into which the data is placed. The latter is used, for example, to select the type or amount of events delivered to a particular VME crate, and allows an event filtering system connected to the crate to be tailored to a particular event type or selection - for example to allow a more thorough analysis to be applied to a smaller sample of events. This optional addressing scheme also permits multicasting data to several destinations.

* This work is sponsored by DOE contract No. DE-AC02-76CH03000

Using the arrangement of multiple streams and VME planes as shown in figure 1, the bandwidth of the system can be increased by adding streams and/or VME crates. Each stream can handle 20 MBytes/sec (40 burst), and each VME plane about 40MBytes/sec (with D64 memories). For example, a 100 Mbyte/sec system would require 3 VME planes and at least 5 readout streams (more streams may be required to reduce dead-time). One could also cascade levels of VME planes, partially building events at one level, passing them on to the next for further concatenation.

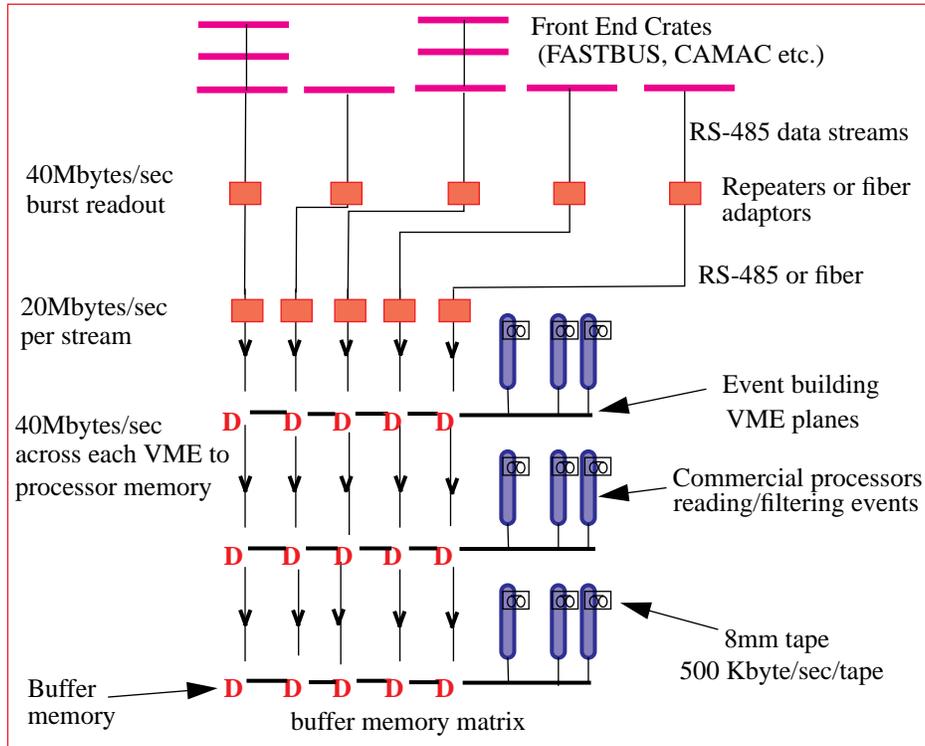


Figure 1: DART Readout Architecture

2.1 Parallel Readout and Transfer of Detector Data

Event data is delivered to buffer memories down multiple parallel cables with minimal control. Sources on a stream (DART data cable) are responsible for all data readout “above” them (see figure 1). Typically they readout one or several crates worth of data, and provide some buffering to smooth out the flow of data.

A data record on the DART Cable is defined by the transmission of an initial word-count or at termination by an End Of Record strobe (EOR). Data flow control on the cable is handled by a single WAIT return. Event data is readout and collected into partial or full events from these buffer memories, across the VME backplane, independently and without the need to know the source of the data.

2.2 *Extensions to the DART Cable*

Two devices have been developed to provide for arbitrary distances between the front end data readout source and the VME crate containing the event buffering. These are an RS-485 repeater - which provides for active regeneration of the signals on the DART cable - and a Fiber Adaptor module pair - which provides translation and later regeneration of the RS-485 signals and protocol into a serial protocol for transmission over fiber optics.

These devices fit naturally into the DART architecture to remove any restriction due to the physical layout of the experiments' detector.

2.3 *(Sub-)Event Buffering*

The use of commercial memories for sub-event buffering - which allows each experiment to tailor the amount of buffering to their particular needs - means that the DART architecture can be implemented efficiently for experiments with a wide range of data sizes and rates. It also provides for experiments where the triggering is essentially DC and those where events occur in 20 seconds out of 60 (the current Fermilab Fixed Target Beamline spill structure). Any change to the spill structure will cause a change in the amount of buffering needed. This can easily be accommodated, without changing any other part of the data acquisition system, by purchasing new memory boards.

2.4 *Building Events*

Since VME is used to readout sub-events from the memories in order to assemble them into full events, all sub-event data for an event must be available in buffer memories that reside in the same VME backplane. In DART, the concatenation of event data is regarded as an experiment specific function. The tools for data readout and buffer handling are provided by DART, but the application that performs this is left outside the DART system and is provided by each experiment.

3 Event Builder

The system architecture of DART allows each experiment to tailor where in the data flow pipeline the sub-events are concatenated and treated as a single entity. All experiments write the data from a single event as "a unit" to tape or disk. However, some experiments do not concatenate the data from a single event at all before logging, while others combine the readout and event building processes by reading the sub-event data across the VME backplane directly into a sequential buffer in a processor's memory.

Current DART experiments have no common definition of an "Event Builder". Each experiment has chosen the most efficient and effective point at which to finally tie together all the data from a single event. It turns out that this is different from experiment to experiment, and depends on such factors as whether beam is delivered DC or with a spill structure, and whether the experiment will reduce its event sample online.

For example, experiment E781 reads out sub-event data from several events into a single buffer, passes this buffer on to one of several filter processes, and only assembles data into an event if it accepted to be logged. For experiment E831, which performs no filtering, the events are assembled in the readout program and logged directly from there. Both experiments use the same hardware and software tools for readout, but their readout applications are different. A DC experiment, E835, has a different read out algorithm to account for the continuous arrival of data.

4 Data Acquisition control, configuration, and monitoring

A companion paper at this conference describes in detail some concepts and implementation of DART run control [3,4]. It will suffice to say here that the data acquisition and monitoring software supports replicated applications across networked processors, parallelizing of start-up and control across multiple nodes for minimal latency, and synchronization across a distributed system. DART run control can support a multi-level data acquisition system of arbitrary complexity; because it uses client-server models, components of the system are loosely coupled, which allows experiment applications to be incorporated independent from the rest of the system with minimal effort.

5 Logging Data to the Offline Analysis Systems

Fixed Target Experiments at Fermilab have traditionally logged their data to tape locally, in the experiment's counting room. In the next year we are installing fiber optic networking throughout the experiment areas. This will allow experiments to transfer all or part of their raw data directly to central offline and analysis systems.

The DART software architecture can support this extension in a natural way. The DART tape logger (dot) can write data to disk or tape, and a unspooling program (duf) can copy events spooled to disk to tape or other locations. In addition, DART buffer management software supports the delivering of buffers in parallel or in series to multiple applications, and DART run control and monitoring can support the execution of multiple loggers.

At least one experiment is planning to take advantage of this new mode of operation at the start of data taking. They will send 10% of their data both to tape and across the networks to the offline storage systems for fast turn around analysis and calibration.

6 Front End Calibration and Monitoring Support

The core DART hardware architecture implements a unidirectional data flow pipeline for the main data collection from the front ends. However, there is no restriction in DART to have a unique path to any front end or data readout module, and DART readout controllers in general support multiple local data readout programs or readout controllers resident in the same crate or on the same cable. This provides for high level access to front end module data for calibration and monitoring purposes.

7 New Data Acquisition Developments

The Fermilab Computing Division is collaborating with CDF and CMS groups in the investigation and testing of new event collection implementations. The first tests will be with an 8 x 8 ATM switch fed by embedded Power-PC VME processors feeding similar processors as event receivers [5].

The ability of these processors to support running of the same real time operating systems as DART - VxWorks - will allow us to port the DART applications to this new event building architecture with minimal changes. The algorithms used in DART by the filter processors to poll for events in the sub-event buffers and determine which events to read out are not applicable here, but the event filtering, data buffer, logging and control structure will all provide the needed functionality in this new architecture.

DART defines a loose coupling between the event filter processors, the front end data collection, and the run control systems. This allows easy replacement by other event building and delivery paradigms. A recent prototype integrating DART run control with the CDF Run 1B level 3 system was accomplished with very little effort.

8 Conclusion

The DART hardware and software architecture provides a common, yet cost effective, data acquisition system for experiments with data collection rates of 500 KBytes/sec to 200 MBytes/sec. This is a wide range which is provided for by the provision for replication of basic hardware components.

Low rate experiments use one VME crate and/or a few streams to collect their data; and high rate experiments use several VME crates and up to ten parallel sub-event data collection streams. Additionally, DART supports experiments which record all collected data as well as those who employ computing power of up to several thousand MIPs to select interesting events and reduce their data logging rate by up to factors of 40.

The DART architecture supports replication of any component or level of the data collection system, thus providing a mechanism for extension of the system to provide for any particular experiments needs.

9 References

1. DART documents are available at URL <http://www-dart.fnal.gov:8000>
2. R. Pordes et al, Fermilab's DART DA System, Proceedings of CHEP94
3. Fermilab DART Run Control, Proceedings Real Time '95
4. Run Control Techniques for the Fermilab DART Data Acquisition System, These proceedings
5. Plan to test ATM, Internal Note