Fermi National Accelerator Laboratory

FERMILAB-Conf-94/103

Fermilab's DART DA System

R. Pordes et al

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

April 1994

Presented at the *Conference on Computing in High Energy Physics 94*,
San Francisco, California, April 21-27, 1994

Operated by Universities Research Association Inc. under Contract No. DE-AC02-76CHO3000 with the United States Department of Energy

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, express 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.

Fermilab's DART DA System[*]

R. Pordes, J. Anderson, D. Berg, D. Black, R. Forster, J. Franzen, S. Kent,
R. Kwarciany, J. Meadows, C. Moore, G. Oleynik, V. O'Dell, D. Slimmer,
J. Streets, O. Trevizo, L. Udumula, M. Vittone, M. Votava, N. Wilcer
Online Systems Department, Fermilab
V. White, Computing Division, Fermilab
Jürgen Engelfried, E781, Physics Section, Fermilab
Taku Yamanaka, E832, Osaka University
Cedric M. Guss, E811, Cornell University
Eric Stern, John Kim, E815, Columbia University
George Zioulas, E835, University of California at Irvine
Anna Majewska, E835, Pennsylvania State University
Art Kreymer, E831, Physics Analysis Tools, Fermilab

Abstract

DART is the new data acquisition system designed and implemented for six Fermilab experiments by the Fermilab Computing Division and the experiments themselves.

The complexity of the experiments varies greatly. Their data taking throughput and event filtering requirements range from a few (2-5) to tens (80) of CAMAC, FASTBUS and home built front end crates; from a few 100 KByte/sec to 160 MByte/sec front end data collection rates; and from 0-3000 Mips of level 3 processing.

We report on the architecture and implementation of DART to this date, and the hardware and software components that are being developed and supported.

1 Introduction

DART has been established as a collaborative project between six Fermilab experiments and the Fermilab Computing Division to develop and deploy the experiments' data acquisition systems [1]. The system hardware and software architecture must be simple enough for the small experiments, yet extensible and fast enough for the large.

This paper presents a summary of the hardware and software components being supported, the current status of the project, together with details of some of the project strategies and issues.

2 DART Architecture

DART DA Parameters

	Small Expts	Large Expts
Trigger rate (KHz)	<.1	10-20
Event size (KByte)	1-12 (up to 200)	5-8 (up to 200)
Rate to event builder (MByte/sec)	1-3	30-160
Event building (MByte/sec)		50-160
# parallel streams	4-6	4-12
# parallel event building VME crates	1	1-4
Max. rate per stream (MByte/sec)		20-40
CPU power for event filter (Mips)	None	1000-3000
Logging (MByte/sec)	1	6-8

The DART system architecture is "parallelized" "extensible" "networked" and "distributed". In terms of hardware components, this means that sub-systems and readout are independent and in parallel, the event building architecture is modular and extensible, and

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

Ethernet is used for control.

For the software architecture, support is given for stand-alone use of sub-systems and embedded processors for commissioning, and for integration of multiple copies of DA components as a tightly coupled system during data taking.

3 DART Hardware Summary

As far as possible all DART hardware modules are commercially available. KTeV has the most challenging data acquisition requirements; its data flow architecture is shown below [2]. DDDs consist of a triumvirate of modules to support the needed flexibility -- the DM115, DC2 and Dual-ported VSB/VME memory. The DM115 [3] provides for input from RS485 at up to 40 MByte/sec to a 4 KByte data FIFO, and for receipt of data in different VME crates based on the value of an address word in the data stream; the DC2 [4] controls data flow at up to 22 MByte/sec from the FIFO over VSB to commercial dual-ported VSB/VME memories (DPMs) and handles their memory management and flow control. The DC2's embedded 68340 processor gives it flexibility at the expense of simplicity, but this was a trade-off we accepted in order to use an already commercially available design.

The dual ported memories can be configured in size and number to meet the individual experiment's needs. Data is also delivered to the VME backplane through other supported interfaces such as the CAMAC drivers listed in the table below.

Event data is read directly by a commercial 68040 processor board (Motorola MVME167) and written to tape, or written through high speed interconnects into commercial UNIX™ workstations for physics analysis before a selected sample of events are written to tape.

4 DART Software Summary

DART software products include all the traditional components of data acquisition systems. Two of these products — the data flow manager (dfm) and distributed system bootstrap and monitoring software (dbs) — are

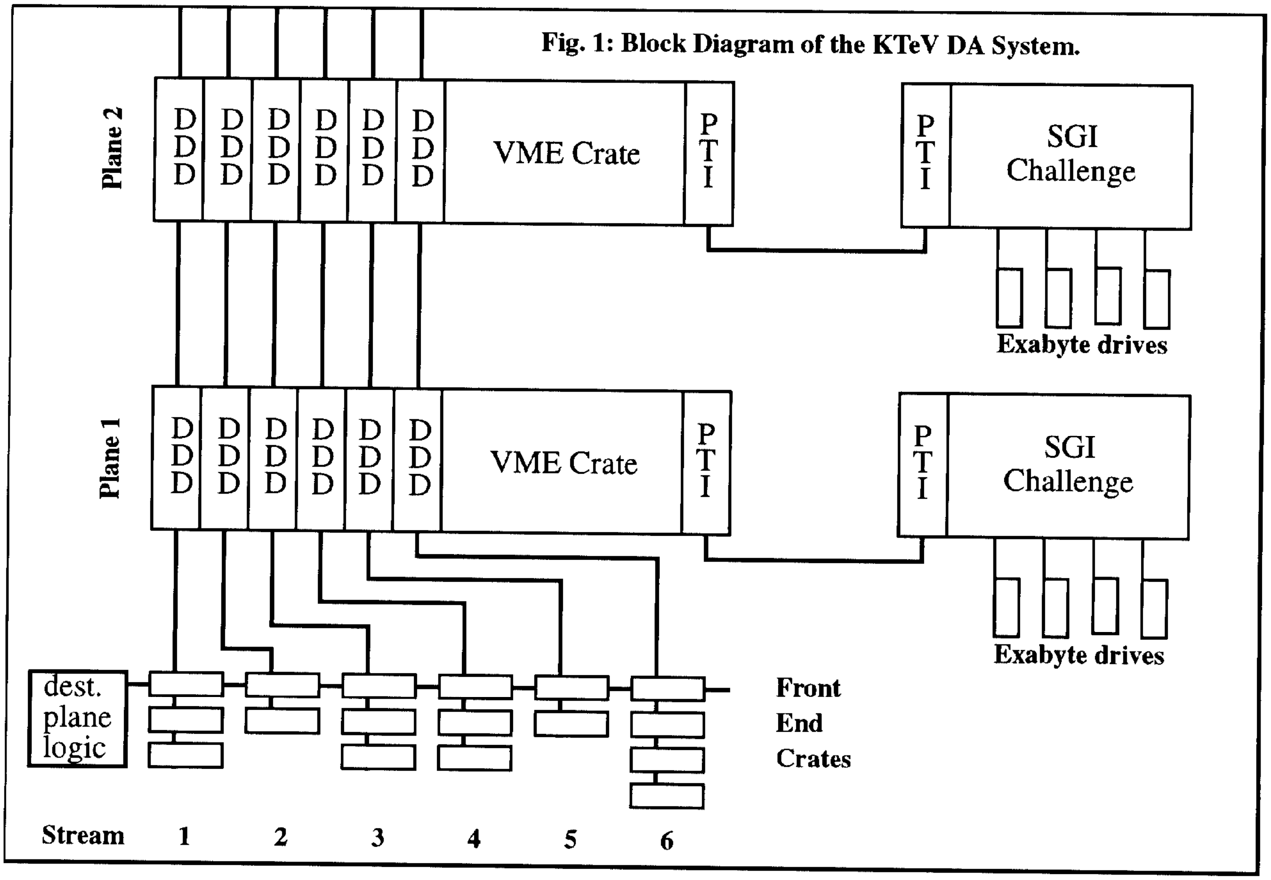

Fig. 1: Block Diagram of the KTeV DA System.

Supported Hardware Modules [12]

Description	Module	Software
68040 VME processor	MVME167	VxWorks, user libraries
FASTBUS Readout Controller	FSCC	VxWorks, FB IEEE routines
FERA Readout controller	DYC+	—
RS485/Fiber adaptor	FOXR/FOXT	—
VME/CAMAC Parallel BHD	CES 8210	VxWorks CAMAC IEEE routines
VME/CAMAC Serial BHD	HYTEC 2992	VxWorks CAMAC IEEE routines
VME/ SCSI adaptor	RIMFIRE 3513	VxWorks 8mm driver
RS485->VSB adaptor	DC2/DM115	Embedded s/w
VSB/VME memories	MMI 6390D	—
VME/VME or VME/processor bus	BIT3	IRIX device driver
VME-VME adaptor	PTI 940	IRIX/VxWorks I/O drivers
UNIX Workstation for event filtering	SGI	Drivers for VME/tape
Host/DA Monitoring Computers	IRIX, SunOS, HPUX?, AIX, VMS	I/O drivers to DA buses, VME

described in other papers at this conference [10]. Conceptually, dfm provides extensions to the operating system in areas of memory management and process queueing specifically for data acquisition needs. DART software packages are designed either as libraries to be embedded in experiment applications, or support user hooks for inclusion of experiment specific code.

DART uses tcl [5] as a common user and program interface. Run control commands are distributed as tcl command strings which are interpreted and dispatched by the receiving program. Graphical interfaces are layered on top of the tcl command line interface with Tk and wish. DART run control and configuration management software addresses DA components logically with named groups rather than physically by node address and process ID, and sends tcl command strings input from scripts or generated internally and then multicast.

Three complementary diagnostic and monitoring tools are used: a message reporter and display program [6]; trace — a continuous recording in time order of diagnostic information; and snapshot — a one time dump of information.

Some Strategies

5 VME as Data Acquisition "Hub"

VME backplanes form the data acquisition hub of the DART data flow architecture. As shown in Fig. 1, data is sent to the VME backplane from many different sources and can then be delivered to any processor connected to VME. Multiple VME planes act as parallel event builders to deliver the maximum required data throughput, 160 MByte/sec.

6 Use of C++ Wherever Suitable

The real time components of DART, and those user libraries that must be easily portable are being coded in C. However, with its advantages in strong typing, and emphasis on proper modulization and encapsulation of code, we are writing DART in C++ [7] wherever it can be justified. It remains to be seen whether this will cause problems under fire: but the experience so far indicates the code is written is more quickly and correctly, and is robust and maintainable. So far, our choices have led to no compromise in data throughput.

Interfaces to FORTRAN are included, to allow the experimenters to integrate in their physics analysis and monitoring code; the strategy, however, is to gently lead them to C and C++ for their time critical applications.

7 Portability across Operating Systems

DART sees the first serious use of UNIX for the main DA by Fermilab Fixed Target experiments. While VxWorks™ [8] is supported on the embedded processor boards, a single flavor of UNIX is supported for the critical level-3 filter processors (currently SGI/IRIX), and several UNIX platforms and VMS for the back end monitoring and host computers. To meet the requirements of both the small and the large experiments, most applications are supported for both VxWorks and IRIX. Coding standards are defined with an eye to portability and POSIX as a reality in the not so distant future.

8 Tcl/Tk as the User Interface and Command Structure

The user interface is based on public domain command line and GUI software, tcl and Tk, with extensions made by us for our environment. The windowing environment, Tk, is layered above this and can be tailored to the experimenters' individual taste.

9 Leveraging

There is lively cross-fertilization of ideas and implementation of tool-kits across DART and the project for delivery of the Sloan Digital Sky Survey data systems [11]. The overlap in general requirements is leading to benefits in terms of new ideas and designs in both.

CDF and D0 are planning major upgrades in their DA systems over the next few years. Both experiments are actively looking at the underpinnings of DART (e.g., VxWorks and TCL). It will be an interesting challenge to see if three such massive bodies as CDF, D0 and the Fermilab Computing Division can leverage their efforts efficiently and effectively.

Some Remaining Issues

10 Level 3 Processor Integration (Data Input and Logging)

KTeV requirements dictate writing to 8mm tape at 3 MByte/sec from a UNIX workstation for data taking. Our benchmarks show that an SGI Challenge can log to 3 Exabytes in parallel on a single SCSI bus at an integrated rate of 1.2 MByte/sec only if very large tape records are written. (The CPU usage is then a few percent.) We need to support input data rates of 40 MByte/sec and logging of 3 MByte/sec simultaneous with data taking.

11 Run Control and DA Management

Fast, reliable, "push button" run start, stop and restart must be provided for systems with 4-6 UNIX stations, and 20-30 front end processors. Our experiences tell us that we must be very careful to understand the interdependencies of the components of the DA, and provide for accurate, simple reporting of where problems occur and what actions should be taken.

12 Robustness of UNIX for Real Time DA

We will be supporting UNIX workstations as an integral part of real time high throughput DA systems. We have developed significant expertise in analyzing VMS system issues, and must gain this experience with UNIX. Our programmers are still legitimately putting "a bug here, a bug there, and pretty soon you have UNIX" on their mail headers.

13 Integration, Monitoring and Analysis

Successful experiments require well integrated "on-spill" and "off-spill" data taking and monitoring systems. The delay in the Fermilab run schedule should stimulate us to expend effort towards making the overall acquisition of data as seamless and robust as possible, with a goal of reducing the time between data taking and writing of physics papers.

14 Status

DART V1.0 [13] was released in September 1993 and was integrated to allow E811 to take data this collider run.

Other experiments are using parts of the system for detector testing. DART uses ups [9] and sccs for product management, and all software is available through the main Fermilab distribution mechanisms.

Most components of DART V2.0 are under test and an integrated system will be released early this summer. This will allow the experiments to take, log and analyze data, but lacks full multi-stream throughput, fully functional run control, level 3 filtering, configuration management, and system monitoring capabilities.

Once we have fully integrated small DART DA systems we will be addressing the robustness, and diagnostic features of the system in earnest.

References

[1] DART - Data acquisition for the next Generation Fermilab Fixed Target Experiments, G. Oleynik, J. Anderson, L. Appleton, D. Berg, D. Black, J. Engelfried, B. Forster, J. Franzen, S. Kent, R. Kwarciany, J. Meadows, C. Moore, R. Pordes, D. Slimmer, J. Streets, O. Trevizo, L. Udumula, M. Vittone, M. Votava, V. White.G. Oleynik et al., IEEE Transactions on Nuclear Science, Vol 41, No 1.

[2] KTeV Data Acquisition System, Internal memo, V. O'Dell Fermilab, T. Yamanaka, Osaka University.

[3] DART Specification Document for DM115 Receiving Board, Oscar Trevizo, Fermilab.

[4] DC2 VSB Input Controller User Manual, Access Dynamics Inc.

[5] Tcl and the Tk Toolkit, J. Ousterhout, Addison Wesley Computing Series.

[6] MURMUR - A message Generator and Reporter for UNIX, VMS and VxWorks, G. Oleynik, L. Appleton, B. Mackinnon, C. Moore, G. Sergey, L. Udumula, FERMILAB-PUB-93, Jun 1993, Submitted to IEEE Trans. Nucl. Sci.

[7] Programming in C++, Rules and Recommendations, Ellemtel Telecommunication Systems Laboratory.

[8] VxWorks is a registered trademark of Wind River Systems, Inc.

[9] ups - UNIX Product Support, M. Votava et al. Conference Proceedings, Real Time '91.

[10] Data Flow Manager for DART, D. Berg et al.; DBS, an rlogin Multiplexor and Output Logger for DA Systems, G. Oleynik et al., Fermilab, presented at Chep '94.

[11] The Distributed Development Environment for SDSS Software, E. Berman et al., presented at Chep '94.

[12] MVME167, CES, Hytec, MMI, SGI, IRIX, SunOS, HPUX, AIX, VMS, Bit3, PTI are all registered trademarks.

[13] DART documents are available at url http://fndaub.fnal.gov:8000.