LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Document Type	**LIGO-T960010-00 - Cxx**	7/12/96

CDS Data Acquisition System Conceptual Design

R. Bork, D. Barker

Distribution of this draft:

This is an internal working note
of the LIGO Project.

California Institute of Technology
LIGO Project - MS 51-33
Pasadena CA 91125
Phone (818) 395-2129
Fax (818) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

WWW: http://www.ligo.caltech.edu/

1 INTRODUCTION

The LIGO observatory systems will generate large amounts of continuous data (approximately 12MBytes/sec for the Hanford site). These data must be stored in such a way that they may be recovered at a later time for off-line data analysis and distributed to various on-line ananlysis and diagnostic systems. The process of collecting, storing and distributing LIGO data is to be performed by the Data Acquisition System (DAQS).

1.1. Purpose

This technical note presents the conceptual design for the LIGO Data Aquisition System (DAQS). This design has been produced in direct response to the LIGO DAQS Design Requirements Document (DRD), LIGO T960009-C. Since it is early in the design phases of LIGO, it is understood that more requirements will be placed on the system over time, particularly as data reduction methods are devised. Therefore, it is the intent of this document to present a baseline design, based on the present requirements, which is flexible enough to incorporate new features in the future.

All designs presented in this document will use technology available at the time of writing. However, the field of data acquisition is rapidly expanding as new technologies become available, and this may open the design to better, faster, cheaper options in the future. Therefore, where possible, some degree of flexibility has been included into the design in allow future developments to be incorporated into the design where ever it is applicable. Also, while specific manufacturers and model numbers of equipment are shown as part of the design in this document, they may not necessarily be the ones used, but rather are meant to be representative of the equipment to be used.

1.2. Scope

A LIGO DAQS is to be developed which meets the requirements set forth in the DRD. The DAQS shall provide the facilities to:

- Acquire LIGO data from various LIGO control and monitoring systems, either via direct analog connections or network connections.
- Format the acquired data into defined data blocks, known as frames.
- Store data frames to long and short term storage media.
- Provide data on request via computer networks from either its short term storage devices or “live” data immediately after it is framed.
- Provide operator views into the acquisition processes.
- Configure the DAQS and its acquisition, storage and distribution processes.

Specifically not considered to be within the scope of the DAQS are:

- Data networks for the distribution of DAQS data to other systems (networks are to be provided by CDS under control and monitoring and by LIGO site general computing).
- Mass storage units, processors and software necessary to read and distribute data from tapes written by the DAQS (considered to be the responsibility of data analysis systems and/or LIGO general computing).

- Tape duplication and distribution facilities.

1.3. Definitions

1.3.1. Front End Systems.

A Front End System is that part of a distributed system which interfaces with the signals to be measured. It is typically a real-time, crate based system, with direct electrical connections to the detector hardware.

1.3.2. Latency.

The time to deliver data over a digital network. Latency time is normally a combination of media access times plus transmission time. Latency times may vary on non-deterministic networks (e.g. ethernet).

1.3.3. VME.

Versa Module Eurocard, a bus based crate system allowing card based modules to communicate with each other via an arbitrated bus. Most LIGO front end systems are based on VME systems.

1.3.4. Real-Time Software.

Real-time software is that software which is deterministic in its task scheduling and duration. Throughout this document, this term refers to software which runs on a VME micro-processor under control of a real-time operating system (VxWorks).

1.3.5. Non-Real-Time Software.

Non-real-time software refers, in this document, typically to that software which runs under the UNIX operating system. This is due to the non-deterministic scheduling and task duration under this operating system.

1.4. Acronyms

API: Application Programmer’s Interface

CA: Channel Access (EPICS Control and Monitoring system network protocol).

CDS: Control and Data System

DAQS: Data Acquisition System

DRD: Design Requirements Document

EPICS: Experimental Physics and Industrial Control System.

FCR: Facility Control Room

GUI: Graphical User Interface

Hz: Hertz

IFO: Interferometer

IP: Internet Protocol.

LRU:

LVEA: Laser and Vacuum Equipment Area

MTBF: Mean Time Before Failure

MTTR: Mean Time To Repair

OSB: Operations Support Building

TBD: To Be Determined

TCP: Transport Control Protocol.

UDP: User Datagram Protocol.

1.5. Applicable Documents

1.5.1. LIGO Documents

- CDS Control and Monitoring Design Requirements Document LIGO-T950054-01-C
- CDS Control and Monitoring Conceptual Design LIGO T950120-C
- Interferometer Diagnostics Conceptual Design LIGO T960108-C

1.5.2. Non-LIGO Documents

2 GENERAL DESCRIPTION

2.1. Product Perspective

The LIGO CDS is divided into three functional components, which must be tightly integrated, as shown in Figure 1: CDS Components. These components are defined as:

- Control & Monitoring Systems (CMS): Provides for the control and monitoring of LIGO interferometers and other scientific instruments, along with the LIGO vacuum systems. It also provides the basic infrastructure for the CDS, which includes such functions as networks, timing, and operator stations.
- Data Acquisition System (DAQS): Provides for the acquisition of all LIGO data integral to gravitational wave analysis and data for use by the IFODS.
- IFO Diagnostics System (IFODS): Provides on-line processing and display of data from the control & monitoring and data acquisition systems for the purposes of diagnosing, characterizing and improving interferometer performance; provides various automated test routines and virtual test instruments.

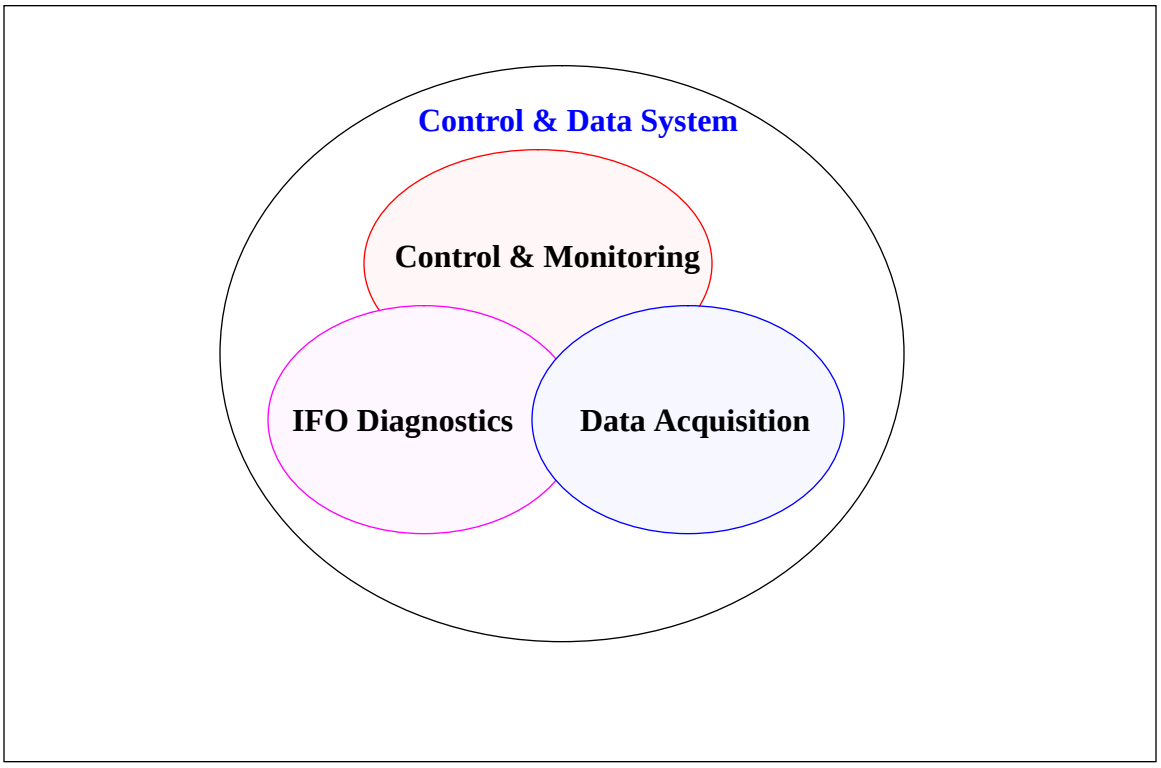


Figure 1: CDS Components

2.2. General Requirements

The specific requirments which this system must meet are given in the DRD, Ligo T960009-C. The primary requirement on the system, which heavily drives the design which follows, is to be able to:

- Acquire analog data at high rates (up to 16K samples/sec) directly from various points throughout the ligo facility (up to 4km from the central OSB)
- Acquire slow data (up to 10 samples/sec/channel) from CMS via CDS networks.
- Transport all of this data to a central location, format it, and store it to long term and short term storage devices for later analysis (~6Mbytes/sec/interferometer).

While the current idea is to store all acquired data channels continuously, the design of the DAQS must also provide flexibility, such that if various data reduction methods are considered and approved in the future, the DAQS is capable of providing those facilities. Examples of some data reduction methods which have been discussed are:

- Storing only a limited data set unless event triggers are generated by on-line analysis, at which time all data channels are recorded for a limited time frame around the event.
- Full data sets are only recorded when not vetoed by abnormal interferometer conditions.

2.3. Functional Overview

The LIGO DAQ will collect and store those LIGO data which are required for data analysis and diagnostics. These data will be sampled at various collection nodes (Data Collection Units) physically distributed over the entire LIGO observatory site. At periodic times, these data will be sent to one or more central collection points (Framebuilders) such that they can be organized into data sets. All data in one data set (frame) will relate to the same global time period. These frames will be stored on short term and long term media, and will be made available for on-line analysis and diagnostics and off-line analysis.

All DAQ functional components require global (site wide) configuration management such that they all interoperate in acquiring and storing the correct data at the correct data rates. DAQ components also need to link with a controls system such that the DAQ may be controlled and monitored from one or more centrally located operator stations.

The figure below shows the functional overview of DAQ. The figure shows functional modules, the main LIGO data flow (bold lines) and the configuration and control data (thin lines).

Note that this diagram is purely a functional view which was generated by the DRD, and does not necessarily translate into distinct and separate hardware systems. The DAQ is designed such that, as much as is possible, the coupling between these functional units is weak. Functional units will couple over single, well defined interfaces. This will provide a flexible system which may be reconfigured with few problems (e.g. porting a function onto new hardware, moving a functional block from one computer to another).

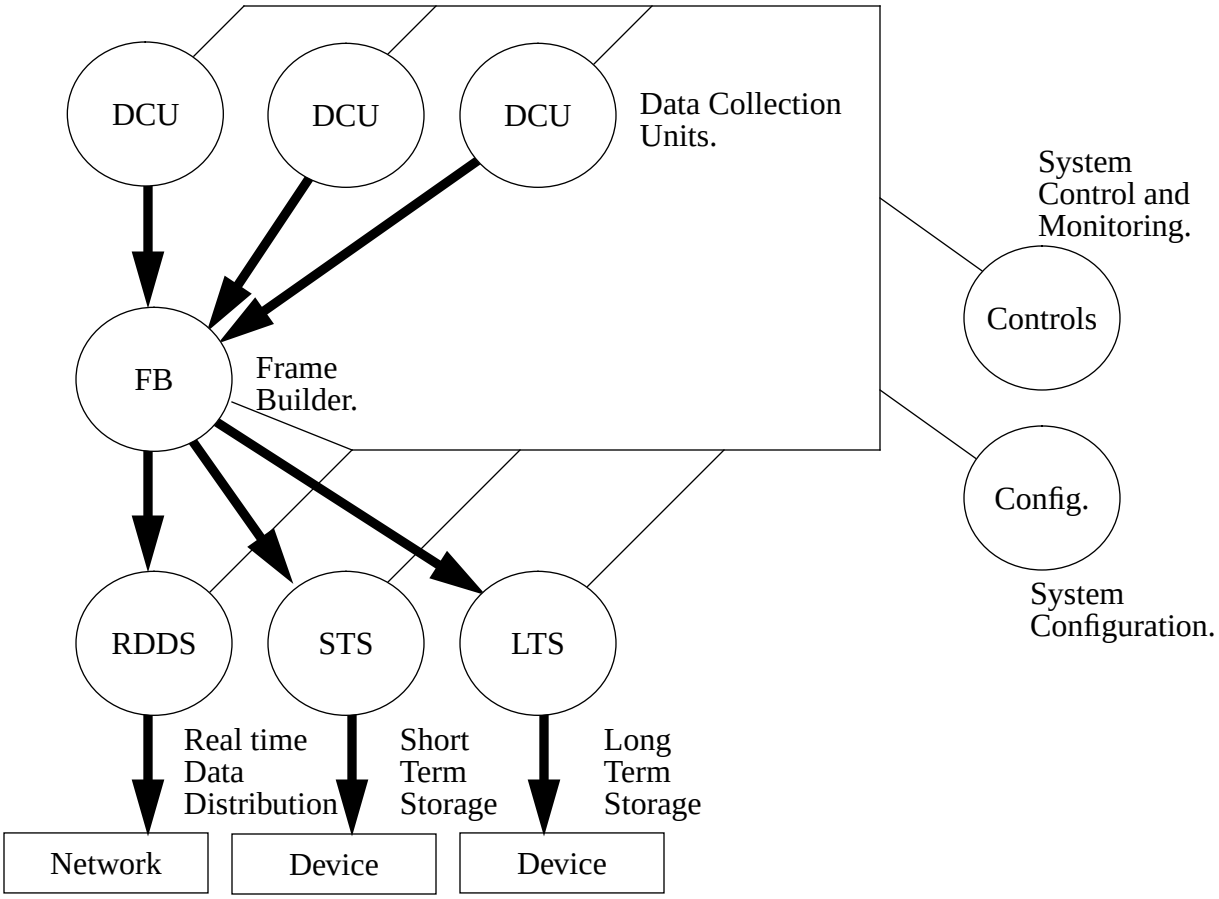


Figure 2: DAQ Functional Overview.

3 DESIGN.

3.1. Overview

3.1.1. System Architecture

Figure 3: Hanford Site DAQS Configuration depicts an overview of the DAQS layout. The Livingston site layout will be similar, but approximately half of the components shown here.

Fourteen VME based DCU will be deployed in the LIGO vacuum equipment areas. The purpose of the DCU is to directly acquire LIGO data at high rates (2K to 16K samples/sec). The 10 DCU in the LVEA equate to one for each set of CDS rack bay locations. Exact quantities and locations will be refined as interferometer designs and layouts progress. In any event, this should be the largest number needed to meet the channel count and rate requirements set forth in the DRD.

In addition to the DCU in the laser and vacuum equipment areas, a Network Data Collection Unit (NDCU) will be housed in the OSB. Its function is to acquire data via CDS networks from LIGO control and monitoring subsystems. This is primarily the slow data (<10Hz) which the various control and monitoring subsystems already acquire as part of their tasks which must be stored by the DAQS along with the fast (>10Hz) data directly acquired by DAQS DCU.

Also located in the OSB will be two Framebuilders. Their function is to collect all of the data acquired by the various DCU, group it into frame structures, and archive it to long and short term storage media. The framebuilders must also provide “live” data feeds, on request, to the Interferometer Diagnostic System (IFODS) and on-line data analysis systems.

Interconnection of the DCU and framebuilders is to be done using a reflective memory network. This will be a private DAQS network for moving data. For more general information on this type of network, refer to Appendix 1 Introduction To Reflective Memory.

Finally, operator workstations will provide the interface into the system for DAQS users. From here, GUI software will be provided to configure the system and to monitor its performance. Connection to the various DAQS processing subsystems will be via CDS standard networks.

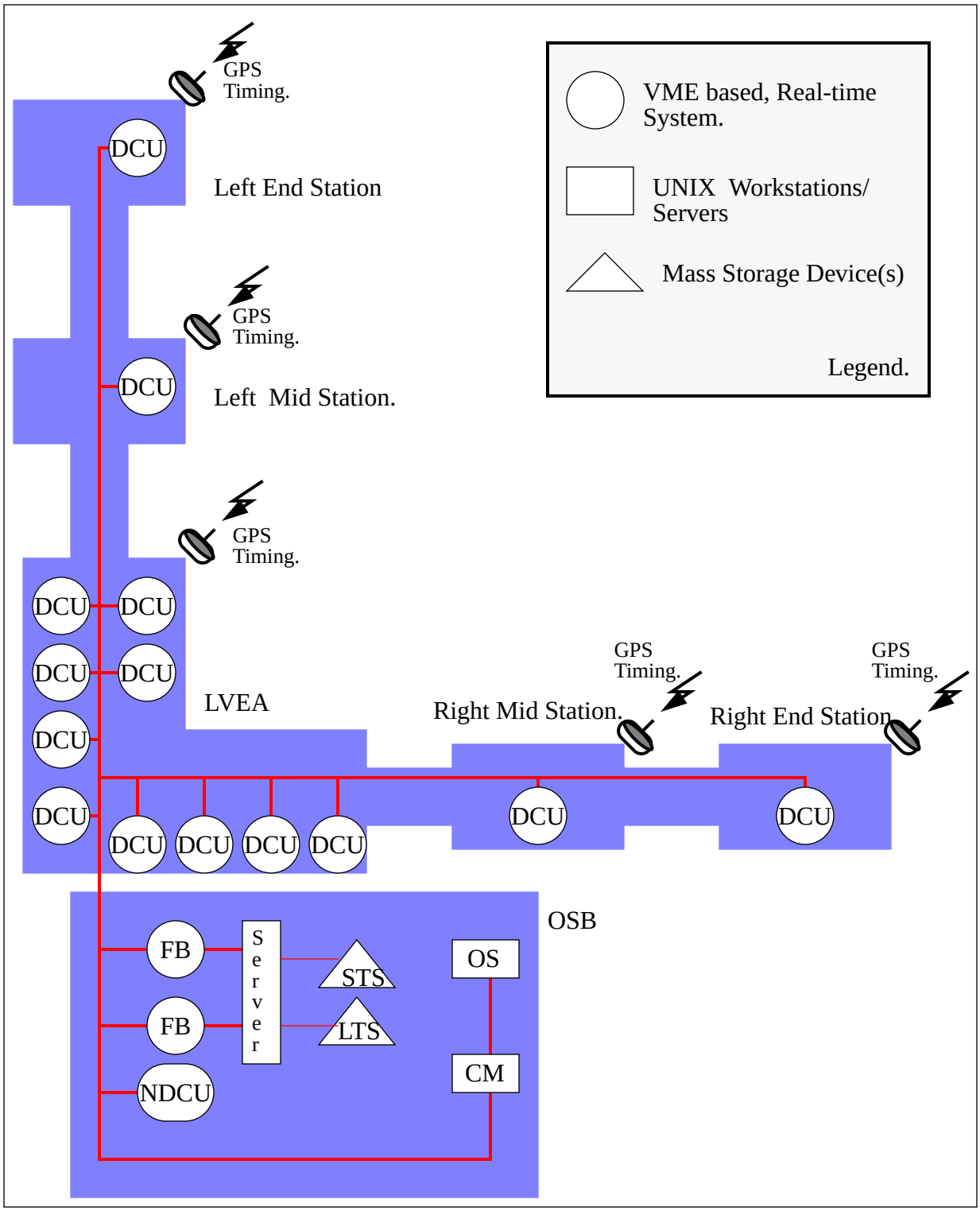


Figure 3: Hanford Site DAQS Configuration

3.1.2. Basic Data Flow

Figure 4: DAQS Dataflow shows a simple diagram of how data will flow through the DAQS. Signals are digitized at the ADC and stored within on-board memory for 1/32 of a sec (512 samples @ 16K samples/sec). This is then moved into the reflected memory, where it is automatically transferred to reflected memory at the framebuilders. At one second intervals, data is pulled from the reflected memory into framebuilders, which organize the data received from all of the DAQS ADC. Once framed, the data is sent via fibre channel to a DAQS server, which has short and long term storage devices mounted.

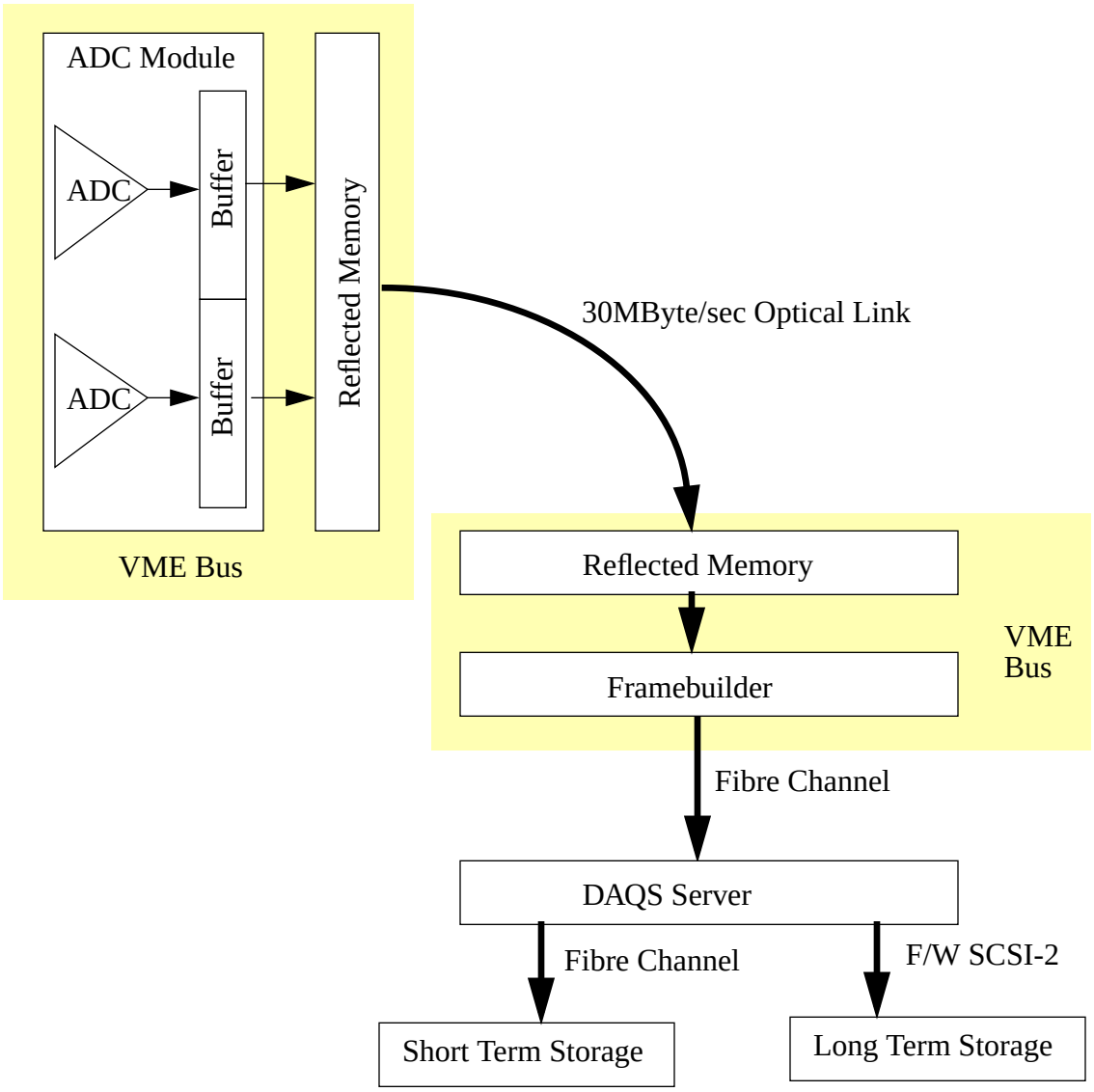


Figure 4: DAQS Dataflow

3.2. Data Collections Units

The DCU are to be the front end components of the DAQS. Figure 5: DCU Interfaces shows how the DCU are to be configured into the overall CDS. The DCU will share a VME crate with particular CMS subsystems. The backplane of this crate, however, is split, such that the two functional subsystems share the crate, but command and control their own backplane. This is done primarily to save on crate costs, as CMS and DCU subsystems normally require less than half of the 21 slots available in a crate at any given location within LIGO. The backplane is split such that each system is independent of the other as far as backplane data contention is concerned.

The interface between CMS and DCU (for analog signals) is at the CMS provided patch panel within the rack enclosure. Analog signals to be acquired typically will come from the CMS analog modules (housed in a 6U Eurocard Cage), and be connected to the patch panel. These signals are then patched into the Analog to Digital Convertor (ADC) modules in the DCU.

Other interfaces into the DCU are a connection to the CDS ethernet (provided by Global CDS) and a reflected memory interface. The ethernet is used to receive configuration information and obtain DCU status information. The reflected memory (discussed further in section 3.3. Data Transport) is employed to send acquired data back to the framebuilders.

.

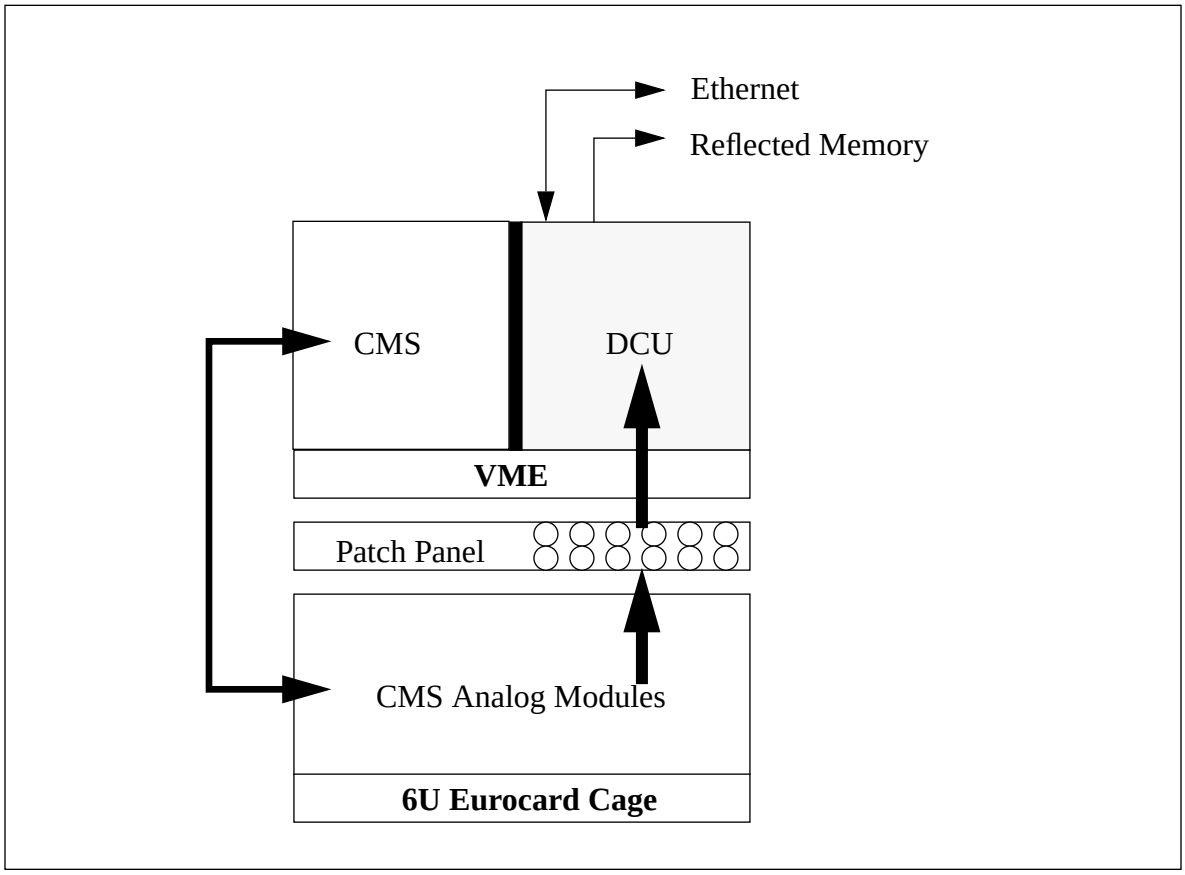


Figure 5: DCU Interfaces

3.2.1. DCU Hardware Components

Figure 6: DCU Components and Connections provides a more detailed view of the components which make up a DCU. The specifications and general functions of these components are discussed the following subsections. How they operate together to acquire data is described in the following section 3.2.2. Operational Overview.

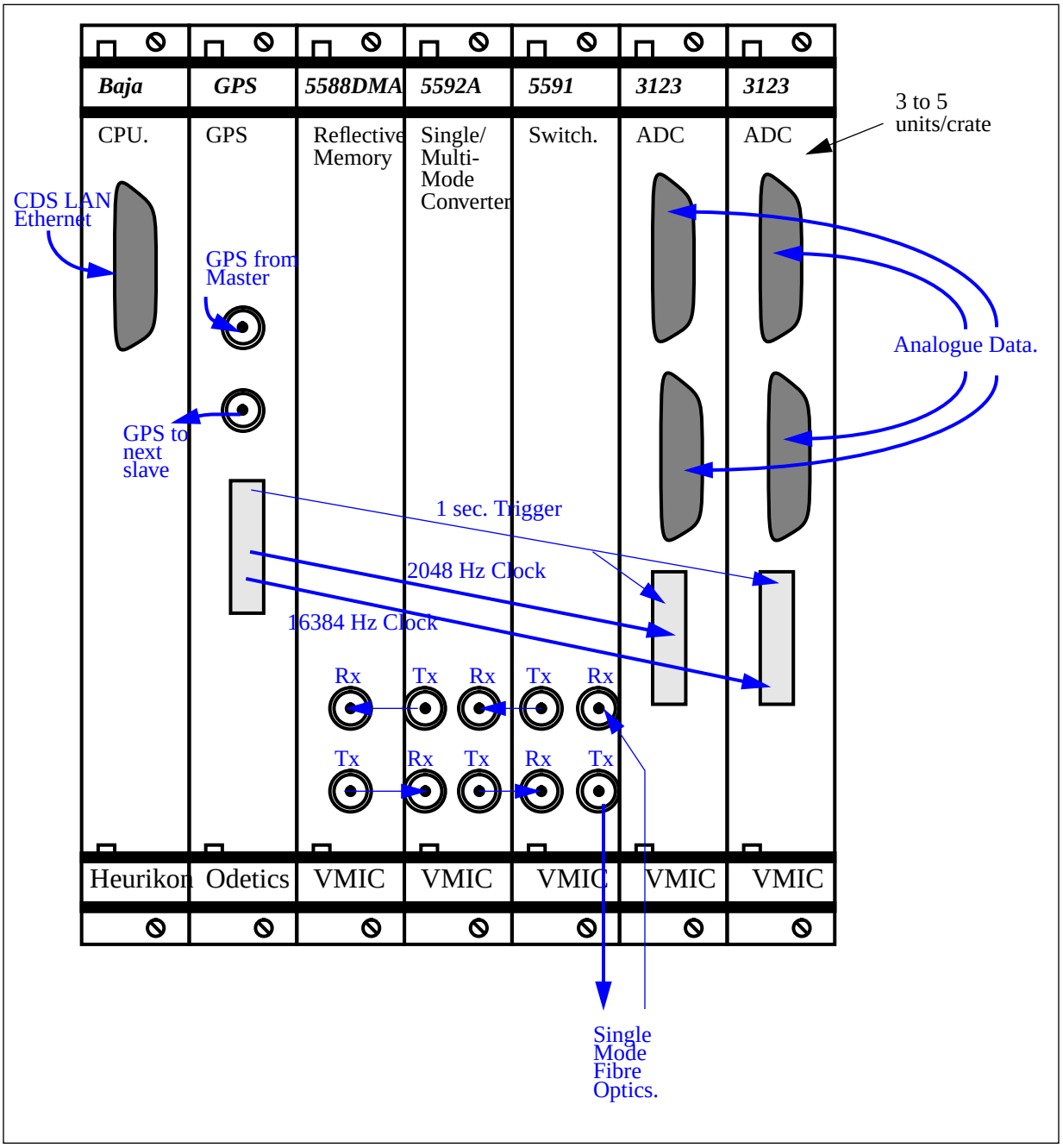


Figure 6: DCU Components and Connections

3.2.1.1 Baja CPU.

The DCU will be controlled and configured using a Heurikon Baja 4700. This is a MIPS 4700 CPU based single board computer running the VxWorks real time operating system.

The Baja will perform the following functions:

- Run the EPICS Control and Monitoring system.
- Run the DAQ software.
- VME Bus Controller.
- Provide sockets based interface to the ethernet.
- Perform DAQ Interrupt Handling Functions.
- Supervise 5588 DMA operations.
- Build DAQ packet header, raise remote interrupts on Frame Builders.

3.2.1.2 Global Positioning System

The GPS board receives data from the roof mounted GPS antenna, either directly or from another GPS board. The GPS board will provide the following:

- Global time service over the VME bus.
- External trigger for “slow” ADC.
- External trigger for “fast” ADC.
- 1Hz VME interrupt.

3.2.1.3 VMIC VMIVME 5588DMA Reflective Memory.

The 5588DMA is the reflective memory card. It will DMA read data from the ADCs when prompted to do so by the Baja. The 5588 provides:

- VME data transfer master using block transfer, D32 and DMA.
- Automatically updates its memory in remote RM units at 30MBps over fibre optics.
- Dual ported memory is mapped onto the VME bus.
- Available in memory sizes from 256KBytes to 16MBytes.

3.2.1.4 VMIC VMIVME 5591 Active Fibre Optics Bypass Switch.

In the event of a system failure on the DCU, the 5591 will allow the reflective memory loop to be bypassed at the DCU. If the failure is in software, the 5591 may be commanded to close the loop. In this case the 5591 is actively regenerating the signal.

If the system failure is hardware related, e.g. power loss, the 5591 will default to a fibre optics “short”.

NOTE: this unit may be replaced with an external 2x2 Optical Bypass switch (AMP Part. No. 9917-2 e.g.). External unit connected to 5588 via front panel connector.

3.2.1.5 VMIC VMIVME 5592A Multi-mode Single-mode Converter.

End stations DCUs will use the 5592A to convert the multi-mode reflective memory signal (max range 1000 ft.) to a single mode single (max range 10km) for the long haul to the LVEA. Typical DCU within the LVEA will not require these units.

3.2.1.6 VMIC 16 channel 16 bit ADC.

The VMIC3123 ADC module (or similar) will be used to digitize most signals acquired by the DAQS. However, since the primary data output, the asymetric port, of the LIGO detectors may have greater requirements, that particular channel will be treated separately during following design phases.

The VMIC3123 provides:

- 16 individual, differential input, 16-bit no missing codes ADC
- Input range: +/-5V
- Integral and Differential Nonlinearity: +/-0.0053%
- 390 Hz to 200 kHz sampling rates.
- 1M sample onboard storage.
- A32 D32 BLT Slave VME data readout (Data access @ 32 bits/210 nsec).
- VME interrupt generation when buffer is half-full or 100% full.
- Built-in test and digital calibration (No channel trimmers)

3.2.2. Operational Overview

This section is intended to give an overview of a DCU in “Normal” operation. Normal operation here is defined as the DAQS has been configured and armed to run and continuous data acquisition is taking place. Designs involving DCU startup, fault handling, etc. is left to the preliminary design phase.

Taking the number of data channels from the DRD, the average DCU loading would be 21 channels @ 16K samples/sec, 10 channels @ 2048 samples/sec and 40 channels @ 256 Hz. The following discussion and timing information assumes a loading of 32 channels @ 16KHz and ignores the lower rate ADC. However, the methods employed for the lower rates will be the same and would minimally affect the timing.

Data flow and timing are shown in Figure 7: DCU Data Flow and Figure 8: DCU Event Timing. At point (1), a 1Hz trigger signal is generated by the GPS module. This signal is wired to the ADC modules, which begin continuous digitization at the GPS supplied clock frequency of 16KHz. Each ADC sample is stored on the ADC module in a RAM buffer. This buffer is software configured to hold 1024 data words for each ADC channel.

At point (2), 512 samples of each ADC have been taken and a “buffer half full” interrupt is generated by the ADC module and detected by an Interrupt Service Routine (ISR) in the CPU. This ISR in turn notifies the Reflected Memory Module (RMM) that data is ready and what memory locations are to be read from the ADC into what locations in the RMM. The RMM then takes over as the VME bus master and moves the appropriate data. Since the ADC module has dual ported RAM, it continues to digitize and store new data into the second half of its data buffer while the

first half is being read out to the RMM. The total time to move 512 samples from each of 32 ADC channels into the RMM is ~2msec. Transmission time of all data from 14 DCU over four reflected memory nets to the framebuilders is an additional ~4msec.

This process continues throughout time period (3) every 62.5msec. At the next 1Hz trigger from the GPS (4), along with continuing the acquisition process, the CPU is interrupted by the GPS. At this point, once the normal data set is sent, the CPU also writes a header for the one second of data in memory. The exact format of this header is TBD, but, along with data set information, it will contain DCU diagnostic information for use by the framebuilders.

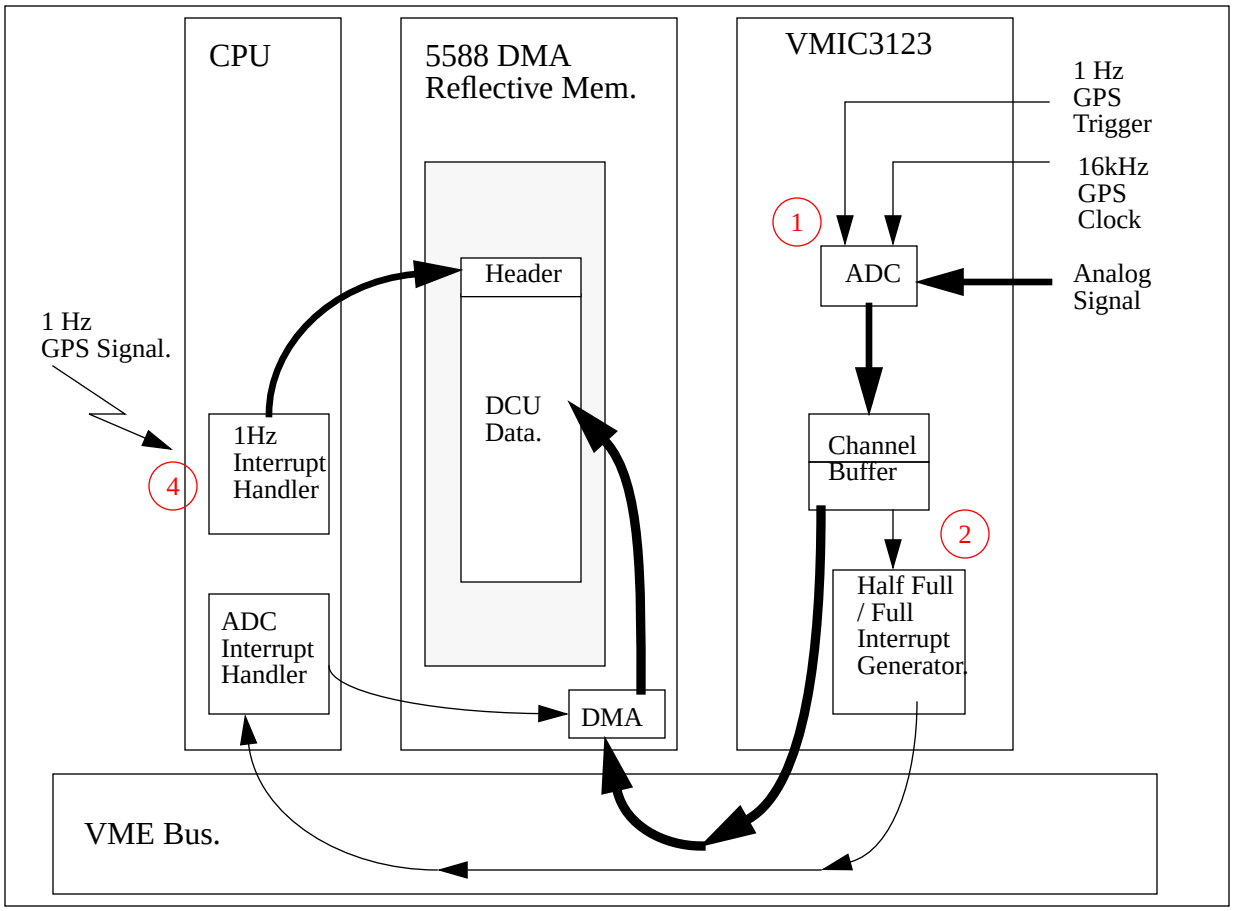


Figure 7: DCU Data Flow

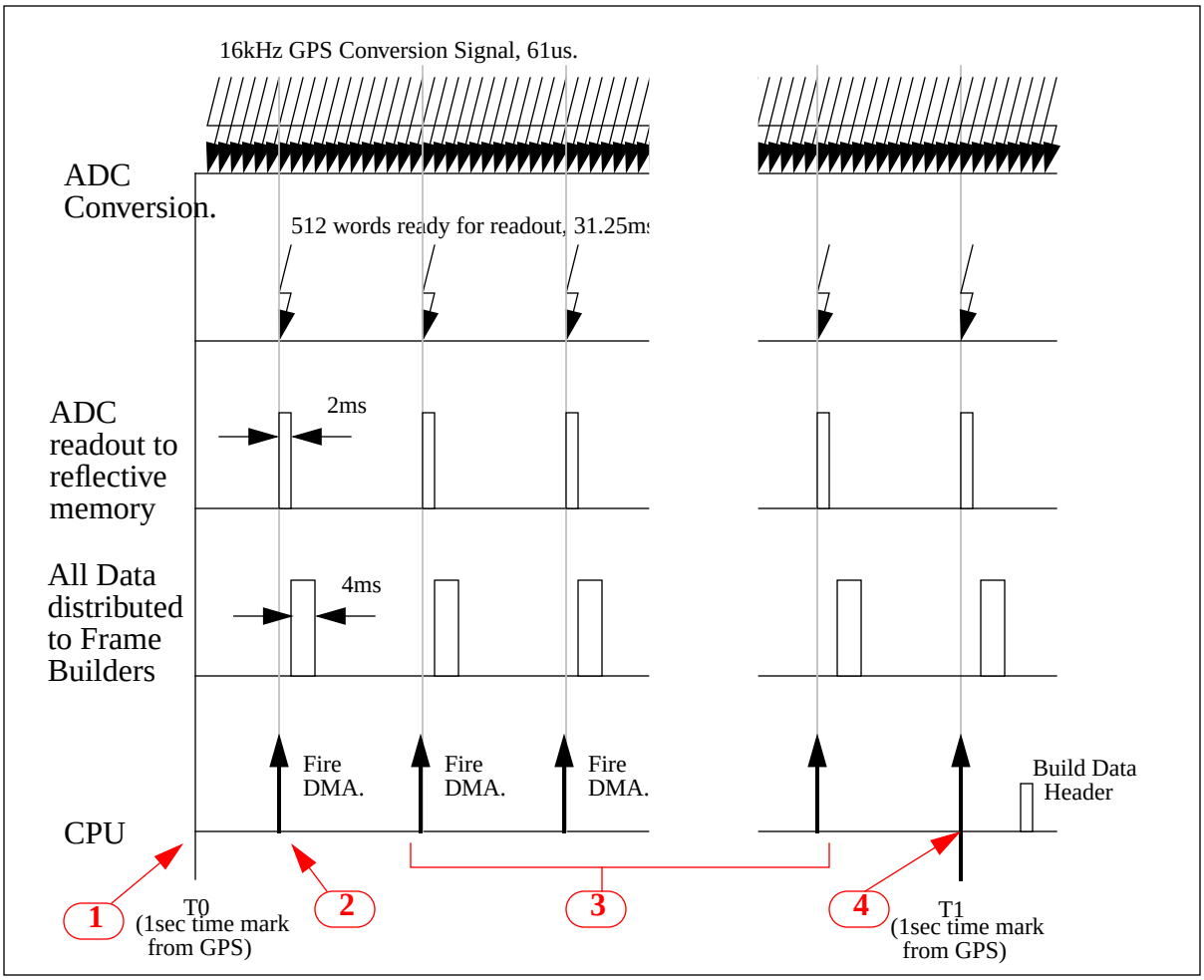


Figure 8: DCU Event Timing

3.2.3. DCU Configuration, Control and Monitoring

Software for control, monitoring and configuration of DCU, along with the remainder of the DAQS, will be developed using EPICS. The use of EPICS, with its network protocols and operator interfaces, will keep it compatible and consistent with the remainder of CDS systems.

Configuration of a DCU primarily pertains to the setup of what data collect, how fast, and where to put it. This is foreseen to be a set of database tables. There will be two tables, an ADC module table and a channel table related to each ADC module. Information included in the ADC table for each module are such items as:

- ADC model number
- ADC module serial number
- Data acquisition rate
- Memory block location (area its data is to be written in the reflected memory)

For each ADC module, it has related information in the channel table. Information included for each channel is:

- Channel number
- Signal Name
- Acquisition rate
- Decimation/Average: The rate at which data is acquired by an ADC is set for the entire module. If data from a particular channel is to be taken at less than the module rate, decimation or averaging may be chosen to reduce the channel sample rate.
- RM location: Area within the ADC module block where channel data is to reside in the reflected memory.
- Whitening information
- Calibration information

Configuration information can be sent to DCU from a central configuration manager at any time, including at startup and when the system is already acquiring data. On normal DAQS startup, configuration information is distributed to all DCU, which then wait for a run command to begin acquisition. To allow one or multiple DCU to be reconfigured when already in an acquisition mode, or to allow a new DCU (one which had been off or inoperative) to join the DAQS while it is running, provisions will be designed in such that it can be accomplished without having to halt the DAQS acquisition processes. Basically, the fields in the configuration tables can be changed at any time. When the new configuration is to be implemented, a new config command is sent to the DCU. DCU, inturn, respond back to the framebuilder controllers when they are ready to execute the the new configuration. Referring back to Figure 8: DCU Event Timing, at point (4), after a data set header has been sent for the last one second of data, if all DCU have responded correctly to the new configuration, the framebuilder controller will set a value into a defined reflected memory location. All DCU will be interrupted by this event, and start operating under the new configuration. All configuration parameters of the DCU may be changed in this fashion, with the exception of the ADC module clock rate. This is a hardwire connection between the GPS and ADC modules and requiresjumpers to be changed at the DCU.

Concurrent to execution of a new configuration, a DCU will upload “old” configuration information and the time it was in effect to a central database. This is to allow later lookup of DCU historical information as it might pertain to stored data. The “new” configuration is also uploaded to the database for viewing of present configuration information. Having the DCU upload this information also allows comparison by a central unit to ensure that the configuration was received and executed correctly.

3.2.4. Network Data Collection Unit (NDCU)

The NDCU is to be a VME based processor with reflected memory, located in the mass storage room of the OSB. It may in fact reside within the same VME crate as framebuilders.

The NDCU has no ADC modules, but rather collects data from control and monitoring systems at low rates (<10Hz). Since it is collecting data at lower rates than the DCU, it will collect one second blocks of data and only place it into the reflected memory at the 1Hz time markers from the GPS.

The configuration of the NDCU will be similar to the DCU, but only contain the channel table. Channel names in this table will be those of the EPICS channels in control systems from which data are to be acquired. The software will then take the channel list and make channel access connections to the various control processors to collect data, using standard CDS networks.

3.3. Data Transport

For moving data from the DCU back to framebuilders, relected memory is the present choice. ATM and Fibre Channel (FC) will also be considered as designs develop further. These are the three network types which were found to be capable of the bandwidths required for the DAQS. Reflected memory is considered at this time to be the most suitable candidate due to its following features:

- High guaranteed end-to-end bandwidth (30MBytes/sec).
- No software driver development necessary.
- Minimal programming necessary (interrupts, DMA).
- Uses dual ported VME memory.
- Supports a variety of hardware systems.
- DMA bus master, thereby not requiring VME CPU time to move data.

The key components for reflected memory were described back in sections 3.2.1.3 through 3.2.1.5. How the reflected memory components tie into a system is shown in Figure 9: Reflected Memory Topology. The network will be set up as four rings. Due to the distances involved, connecting to the mid and end stations will be via single mode fiber. Since reflected memory modules come standard with multi-mode connections, single mode to multi-mode convertor units are installed at the mid and end station units, as well as within two units at the OSB.

As shown in the diagram, along with the DAQS, the Interferometer Diagnostics System (IFODS) is on the reflected memory net. This is to be the interface to the IFODS, such that that system has immediate access to live DAQS data for analysis and presentation to operators. Note also that the IFODS unit and the framebuilders have four reflected memory modules and receive data from both loops.

The amount of memory on each reflected memory module will vary, as memory is a cost driver. Each DCU will typically contain 2 MByte, while the IFODS and framebuilders will contain 8 MByte modules. Therefore, the units receiving all of the data are just the IFODS and framebuilders, not the DCU.

Each DCU will be assigned its block as it relates to the full memory at the framebuilder. To the extent possible, each DCU will place data into its section of memory exactly as it will appear in the final frame produced by the framebuilder, such as the amount of data movement and manipulation required of the framebuilder is minimized. At any time, at least 2 seconds of data are maintained in the reflected memory system.

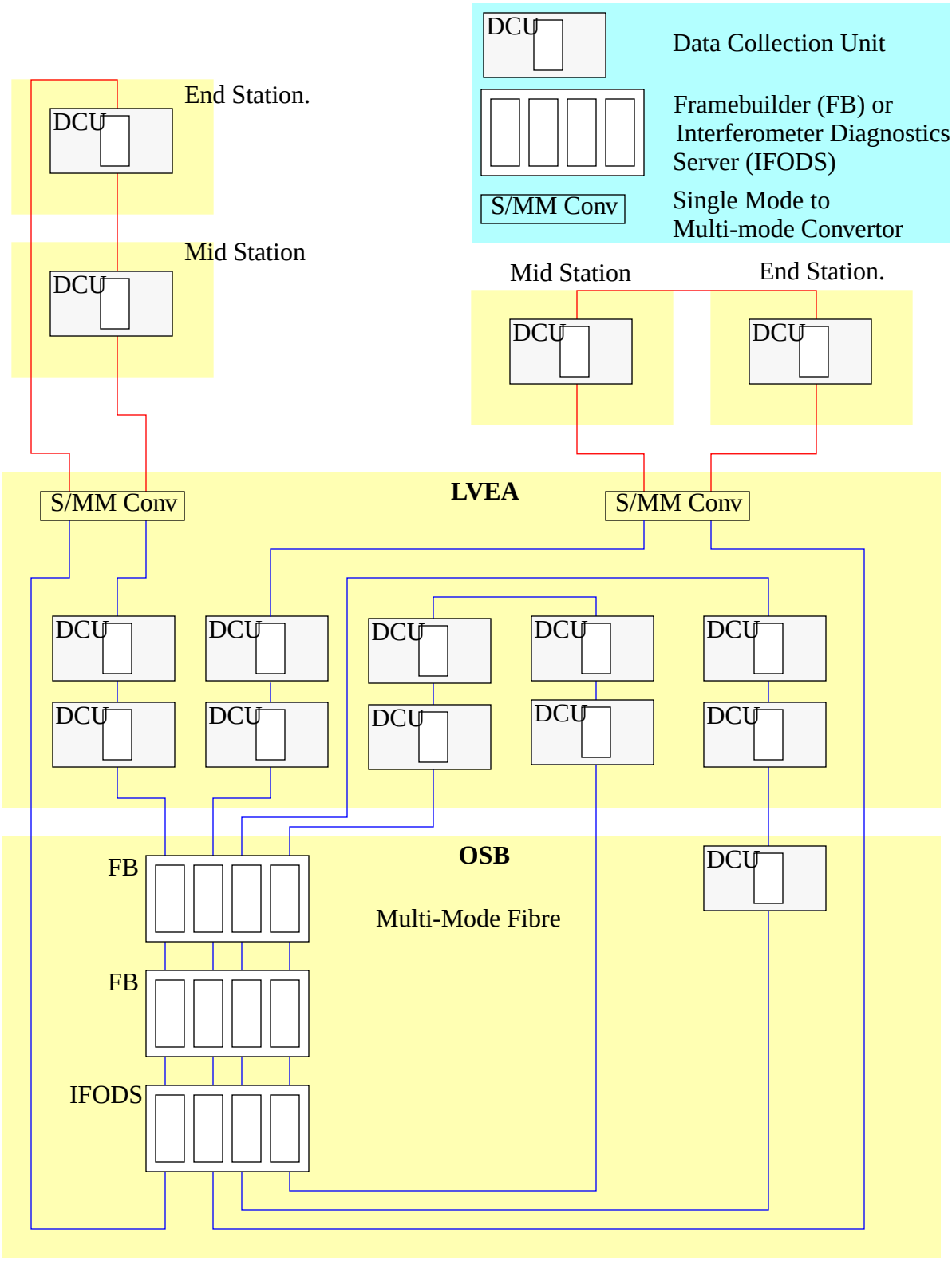


Figure 9: Reflected Memory Topology

3.4. Central Data Collection and Processing

Located within the mass storage area of the OSB is to be located the framebuilders and other equipment which gather of the data from the DCU and store and distribute the data. A concept of the equipment and layout is shown in Figure 10: Central Processing Equipment.

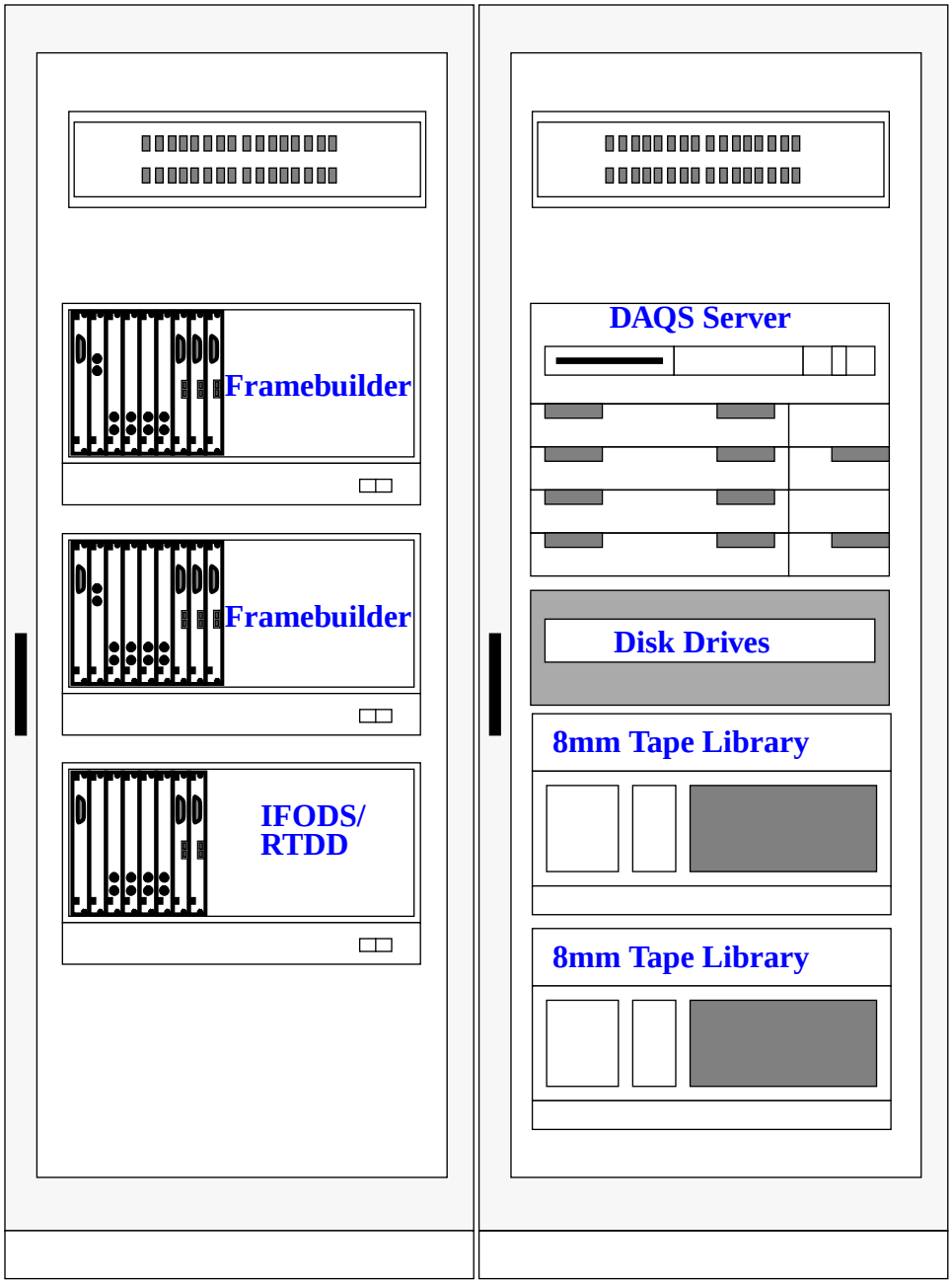


Figure 10: Central Processing Equipment

3.4.1. Framebuilder

There is to be one framebuilder for each interferometer, with key components are as shown in Figure 11: Framebuilder Layout. The modules shown provide the following functions:

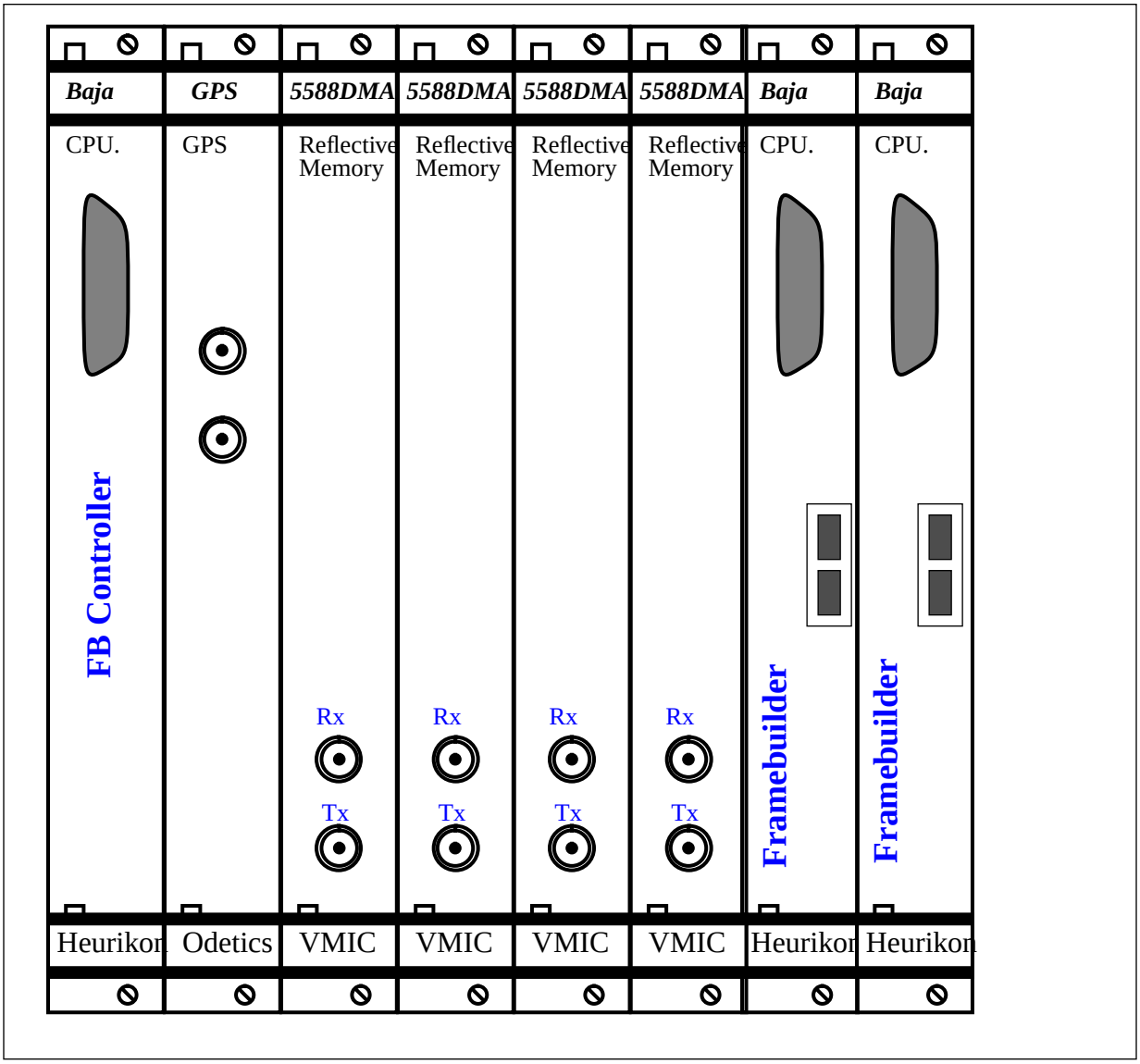


Figure 11: Framebuilder Layout

1. Framebuilder Controller (FBC): A real-time CPU, this unit controls and monitors the acquisition process, ensuring all DCU have delivered data at proper times and the system is functioning properly. This unit also coordinates configuration of the system.
2. GPS module for providing timing information to the framebuilder.
3. Four reflected memory modules, which, as previously described, continuously contain the last two seconds of data from all DCU.
4. Framebuilders: Real-time or Sparc CPU which perform the framing functions and send the data to be archived.

3.4.1.1 FrameBuilder Controller (FBC)

The FBC controller has three primary tasks:

- Write the frame header information which will be used by the other controllers in the crate for writing data.
- Coordinate system configuration.
- Perform system diagnostics

Following receipt of header information from all DCU, indicating a one second frame of data is complete, the FBC will verify the operation of the system (all DCU responding properly and all data received, timing system is operating properly as indicated by timestamps on DCU headers, etc.) and write a header for the complete one second data set. It will then signal the framebuilders that data is available for framing and storage.

The FBC would also be responsible for putting together the Special Purpose Frames (SPF), as outlined in the DAQS DRD. These would be prepared and then the appropriate framebuilder(s) signaled to store these frames, along with their normal data sets.

3.4.1.2 Framebuilders

The framebuilders perform data framing operations and work through the DAQS server to store data (Figure 12: DAQS Data Storage). On the event trigger from the FBC that a second of data is available, these units frame up data in accordance with their configuration tables, which includes the type of frame they are to store and what data is to be contained in that frame type. The data to fill these frames is pulled directly from reflected memory.

To store the data, the framebuilders will NFS mount disk partitions controlled by the DAQS server. Connections to the DAQS server and the server to disk is accomplished via fibre channel. In the case of the long term storage (tape), once it has filled a partition of disk, it:

- Switches further data recording to a new partition.
- Assembles a tape header, with TBD media preambles and data summary information.
- Signals a process in the server to begin backup of the disk partition to a designated tape drive.

The advantage seen to storing the data via the DAQS server instead to units directly connected to the framebuilders is that now processes to serve data to users can be developed and run on the server, which is to be a high performance, multi-processor platform. These processes would have access to the same data files (at a lower priority) and data requests would not need to be handled by or impede the operation of the framebuilder processors.

This initial design shown has two framebuilders, roughly assigning one to handle long term storage and the other short term storage. Since there is a large amount of data at high rates for these units to handle, and they may be required to do more than frame data in the future, the number of CPUs could grow, with processors working in tandem to keep up with the data rates.

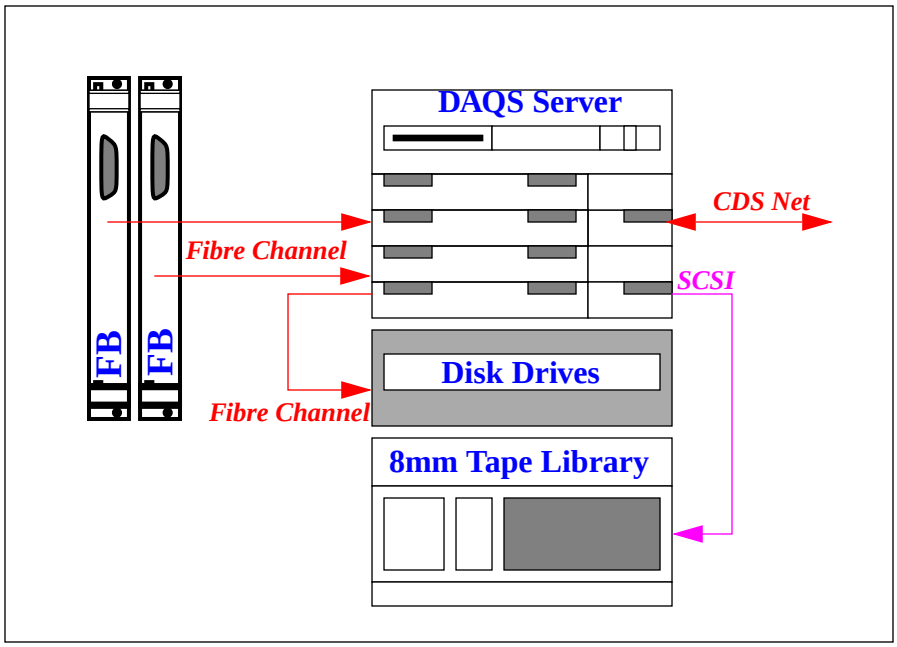


Figure 12: DAQS Data Storage

3.4.2. Real Time Data Distribution (RTDD)

DAQS data will be available to remote units via one of the CDS fibre channel networks. A processor, co-located in a crate with the IFODS, will provide a subscription service, allowing authorized hosts to request delivery of DAQS data. These frames will be sent to the host until the service request is terminated. Two implementation possibilities will be studied during the preliminary design phase:

- Client/Server: Data requestor makes a direct, socket type connection to the RTDD CPU. The RTDD CPU sends data to the client based on the client request.
- Broadcast: At the incoming rate of data to the FB (every 1/32 of a second), the RTDD unit broadcasts the data (perhaps on different “channels” similar to network broadcasts of audio and video) and any process(or) on the net which wants to listen in can.

3.4.3. DAQS Server

The DAQS server is to provide the central functions of:

- Providing and controlling all DAQS mass storage media
- Serving data and DAQS control and status information to other systems
- Managing the DAQS configuration and configuration databases
- DAQS error reporting and logging

3.4.3.1 Short Term Storage

The DAQS DRD calls for a large random access disk storage system (to 400GByte at Hanford). Since the first few years of LIGO operation will involve installation and commissioning of sys-

tems, the larger data storage capacity won't be necessary at that time. Therefore, the proposal here is to start with a smaller disk storage system, which can either be expanded during operations, or, at the rate technology is this area moves, be replaced or augmented by a larger, faster system once LIGO is in full observation mode. The initial drive being considered is a 63GByte storage array (30x1.2GByte 7200 RPM) system with a fibre channel (25MByte/sec) interface.

3.4.3.2 Long Term Storage

With the same reasoning as given for STS, the initial tape systems delivered by the DAQS will be of lower performance and capacity than that required in full observation mode. Moving from a unit which can handle 1MByte/sec storage rate to one which can handle >6MBytes/sec results in a 20 times cost increase (roughly $3000/drive to $60,000/drive). Therefore, it is proposed to initially deliver a 140GByte, 8mm robotic (10x14GBytes/tape) tape storage unit for each frame-builder and defer the higher performance and capacity units until LIGO operations. At that time, these smaller units would be used to store limited data sets.

3.4.3.3 DAQS Control and Monitoring.

DAQ will be centrally controlled and monitored from one or more CM workstations. CM workstations will be Sun Microsystems workstations, with one or more bit mapped graphical screens. The CM will run the X Windows X-Server software (release 5 or later).

All graphical displays will be run using the SAMMI GUI system. SAMMI allows the controls processing and windows display to be functionally separated. DAQ will provide a SAMMI API which will run on a Sun UNIX system. This system will have network access to all DAQ modules.

The CM system will:

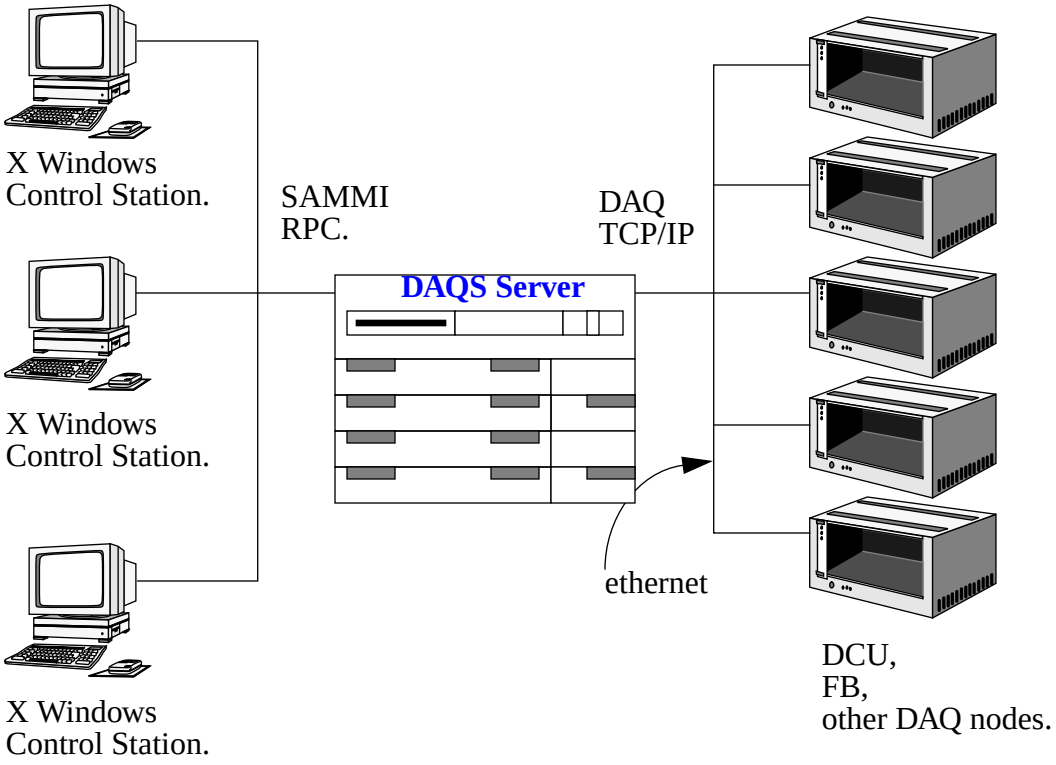
- Display DAQ system status.
- Display detailed status of each DAQ unit.
- Allow activation/deactivation of a DAQ configuration.
- Allow DAQ units to be taken off line and brought on line.
- Show all storage systems status.
- Show all network systems status.
- Report all alarms, warnings and message logs.

3.4.3.4 DAQ Configuration

Overall configuration of the DAQS will be accomplished using Sammi based GUI into database tables. The design and development of a database will occur during the

A DAQ configuration consists of:

- Full list of all signals to be acquired (pre-defined by Format, e.g. CDF, LDF).
- Data rates for all signals.
- Frame Builder ID to service data framing.
- Data storage configuration (High Capacity LTS, Low Capacity LTS, STS, RDDS).



DAQ configurations will be defined off line by the operator using a GUI based toolkit which will be provided by DAQ. This will allow the operator to select which channels to acquire, at which rates, and how to format these data.

The CM will allow the operator to switch configurations on and off in real time. Configuration switches will be implemented as a seamless operations, such that no data will be lost or discarded during this procedure.

It is anticipated that all DAQ signals will be acquired at maximum acquisition rates. The DAQ system has sufficient DCU and network resources to implement this option. Configuration Management in this scenario reduces to data processing at the Frame Builder. The FB will have the complete IFO data set available, and will either record the complete set (CDF frames) or a sub-set of the data (LDF) depending upon the running configuration.

Data may be recorded at rates lower than the ADC rates by implementing data decimation at the Frame Builder (data polling).

Data rates may also be reduced by implementing a data reduction algorithm at the Frame Builder (e.g. averaging).

3.4.3.5 DAQS Database

The DAQS DB will be the central repository for all CDS configuration data. This will include:

- Formats for Frame Builders.
- Reflective Memory memory maps.

- Identities and details of all signals connected to the DCUs.
- Defined configurations.
- Run time information for all storage and distribution systems (LTS, STS, RDDS).
- All run time data necessary for the operation of the DAQ.

These data will be stored and accessed using a commercial relational database product. All DAQ nodes will have access to the database via the CDS provided TCP/IP LAN.

3.4.3.6 DAQ Error and Message Logging.

The DAQ will have a station dedicated to the task of alarm, warning and message logging. This station will allow:

- Graphical display of DAQ Alarms on any X-Window Server.
- Graphical display of DAQ Warnings on any X-Window Server.
- Graphical display of DAQ messages (system or operator originated) on any X-Window Server.
- Archiving all alarms, warnings and messages to disk with time-date information and originating system/user identification.

APPENDIX 1 INTRODUCTION TO REFLECTIVE MEMORY

To pass data at high rates through the DAQS, the DAQS will be provided with its own network based on reflected memory. The specifics of how it is used in the DAQS is shown in other sections, but an overview of reflective memory is given here as background to understanding the detailed implementation.

Reflective memory allows two or more computers to exchange data through the use of duplicated memory. Each computer is assigned a reflective memory card with a unique identifier. All of the cards are then joined by multi-mode fibre optics cable into a ring topology.

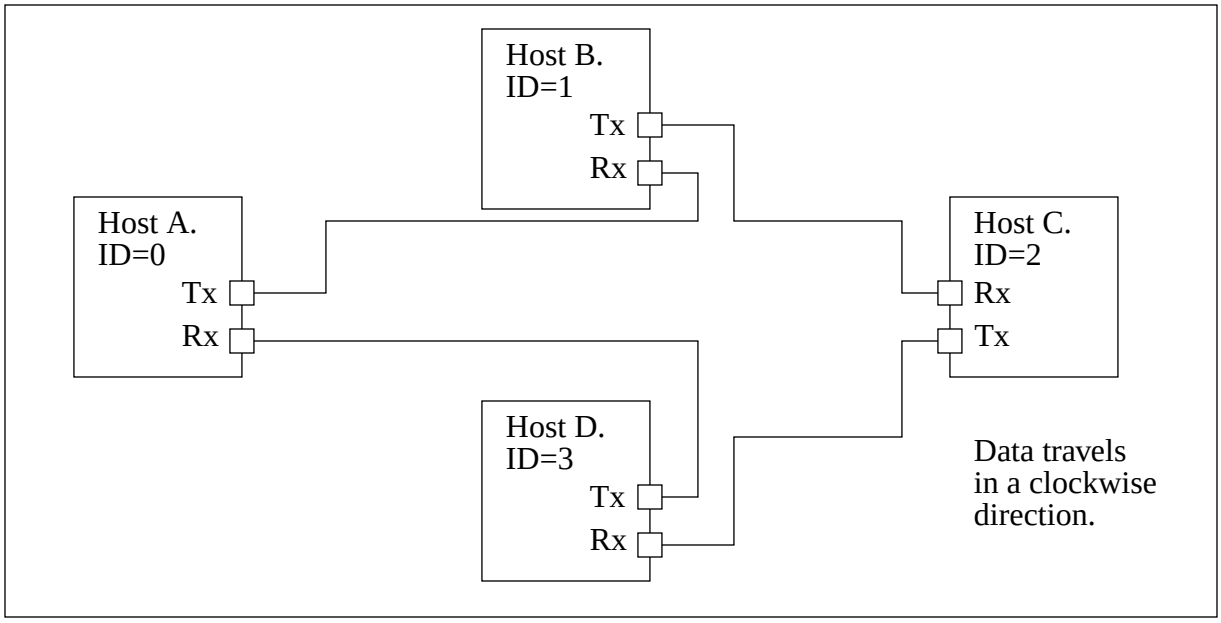


Figure 13: Reflective Memory Ring Topology

Any data written by any host processor into its own reflective memory will be automatically copied into the same memory location on all the other reflective memory cards. Therefore each host has local access to a complete, constantly updated copy of the same data.

If the reflective memory cards are directly joined using multi-mode fibre the maximum inter nodal distance is 1000 ft. If the multi-mode signal is converted to single-mode, then this distance increases to a maximum of 10km.

Nominally every data change will be distributed to each and every node in the fibre loop. Data however may be “targeted” to specific hosts by breaking a large memory space up into smaller blocks. There exist many memory size options for the Reflective Memory cards, and therefore a host will only be updated with that data which lies in its onboard memory range. As an example, from Figure 13: Reflective Memory Ring Topology above, Host A, B and C could each have a 1MByte memory and Host D 4 MBytes. The memory addresses of A, B and C could be set such that they map into 3 different 1 MByte areas of Host D. Therefore, Host D would see all of the

memory of A, B and C, while A, B and C only see their own. As will be shown for the DAQS, this configuration is useful in that only the central systems (framebuilders, etc.) need to see all of the data, but not the DCU. This allows less expensive units to be used where full data memory is not required.

Reflective Memory technology has several advantages of other data networks presently available for the DAQS application:

- High bandwidth transfer of data between nodes (30MBps).
- No system configuration necessary (except when using interrupts).
- Plug and play technology.
- No software overheads.
- DMA block data transfers (No host CPU overhead in transferring data).

Reflective memory is designed purely to provide a fast memory duplication service. It does not support any higher level types of services found in networking systems such as Ethernet, FDDI, ATM or Fibre Channel.