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

Document Type	**LIGO-T970136-00 - Cxx**	7/12/96

CDS Data Acquisition Preliminary 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 preliminary 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* and reflects an update to the conceptual design presented in *CDS Data Acquistion Conceptual Design, LIGO-T960010-00-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. Changes to Conceptual Design

No major changes are presented here to the DAQS conceptual design documented in *CDS Data Acquistion Conceptual Design, LIGO-T960010-00-C*. Primarily, this document provides additional design details and some minor changes based on additional experience gained from a prototype which was developed over the past six months.

1.4. Prototype Test Results

During the preliminary design phase, a prototype DAQS was built which approximates the design set forth later in this document. This system is described in *LIGO-T970126-00-C 40m Data Acquisition System Quick Reference*.

This system is installed on the Caltech 40m interferometer and has been fully operational for the last 3 months, acquiring data from 18 channels @16KHz and 128 channels @16Hz and writing them to disk and tape. The standard VIRGO/LIGO data format has been used for writing data "frames" and VIRGO developed frame libraries used to write and extract the data to/from frames.

1.4.1. Acquistion rates

The prototype system is limited to 1MByte/sec acquisition, far short of the 6MByte/sec/interferometer required by LIGO. The present limiting factor is a fast ethernet link between the processor which performs the data framing (framebuilder) and the server, which connects the tape and disk drives. In the prototype, the framebuilder formats a frame and writes frame files to disk by Network File System (NFS) mounting the server disk drives. This is not intended to be done in the final design, but the interfaces were readily available for both the real-time processor doing the framing and the Sun Ultrasparc server. Designs call for this to be a fiber channel, a network which has been tested on the real-time processors to provide >35MByte/sec throughput. However, the fibre channel transceiver for the UltraSparc was found not to operate at all (manufacturer had only tested on a Sun Sparc2). The manufacturer is now working the problem and it is hoped that these units will be available in the next six weeks.

1.4.2. Data Testing

With this system, data has been written by the DAQS and has been independently read out by LIGO scientists performing data analysis using VIRGO developed frame libraries. The only problems encountered were some early memory leaks in the frame libraries which would, over time, cause the system to fail. These problems have since been repaired. As of this writing, the prototype system has been operating continuously for the past 500 hours. The only errors noted have been the loss of 130 data frames (1 sec. each) out of the 1.8 million acquired during this time period (.007%). This frame loss has been found to be due to the fast ethernet link to the UltraSparc. At infrequent and random times, the UltraSparc server goes off to perform some other housekeeping task for a 3-4 second time period and ignores the incoming data (in fact, initiates a collision condition on the fast ethernet), causing the framebuilder to drop frames. Although it is

not intended to use the fast ethernet and NFS to write frames for LIGO, this type of occurrence has also been noted by others on the LIGO ATM network, thereby indicating a possible problem to test for when fibre channel is installed.

1.5. Definitions

1.5.1. 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.

1.5.2. 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).

1.5.3. 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.

1.5.4. 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).

1.5.5. 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.

1.6. 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.

1.7. Applicable Documents

1.7.1. LIGO Documents

- CDS Control and Monitoring Design Requirements Document LIGO-T950054-01-C
- CDS Control and Monitoring Preliminary Design LIGO T960142-C
- Interferometer Diagnostics Conceptual Design LIGO T960108-C
- Specification of a Common Data Frame Format for Interferometric Gravitational Wave Detectors (IGWD) LIGO-T971030-00-E
- 40m Data Acquisition System Quick Reference LIGO-T970126-00-C

1.7.2. 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. The current design for this system is documented in CDS Control and Monitoring Preliminary Design LIGO T960142-C.
- 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.

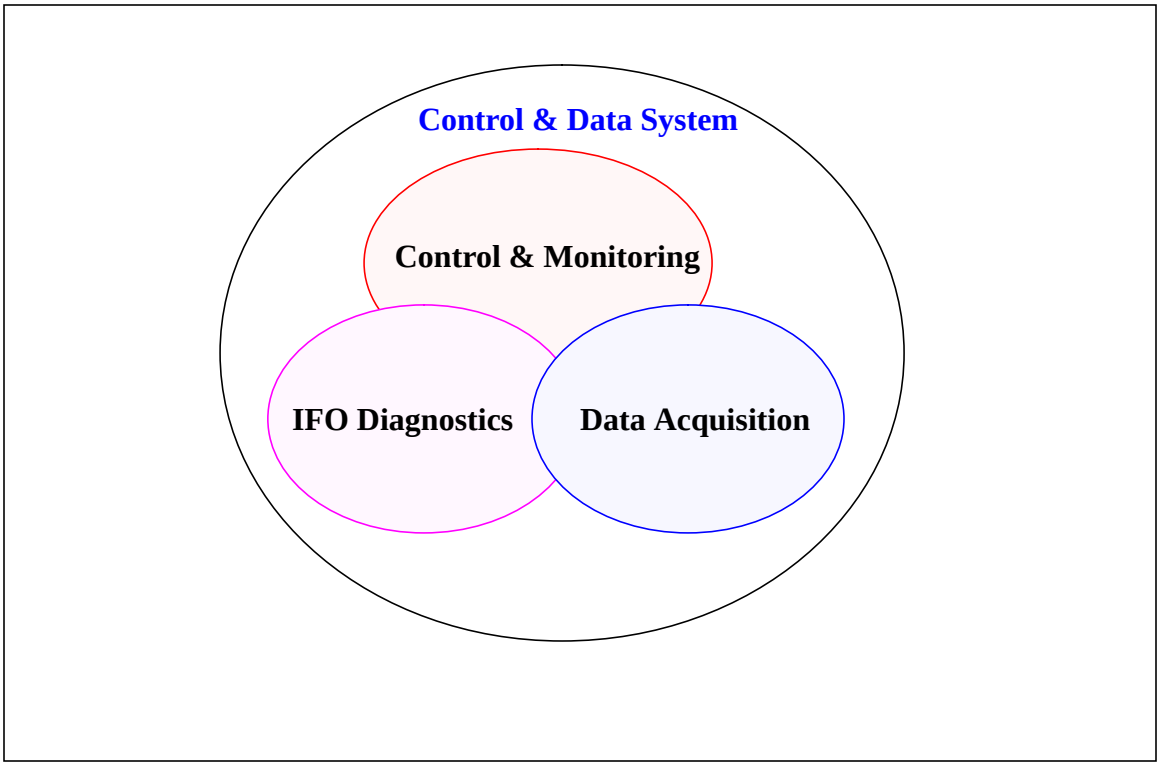


Figure 1: CDS Components

2.2. General Requirements

The specific requirments which this system must meet are given in the DRD, LIGO-T960009-C. The primary requirements 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 (1 sample/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).

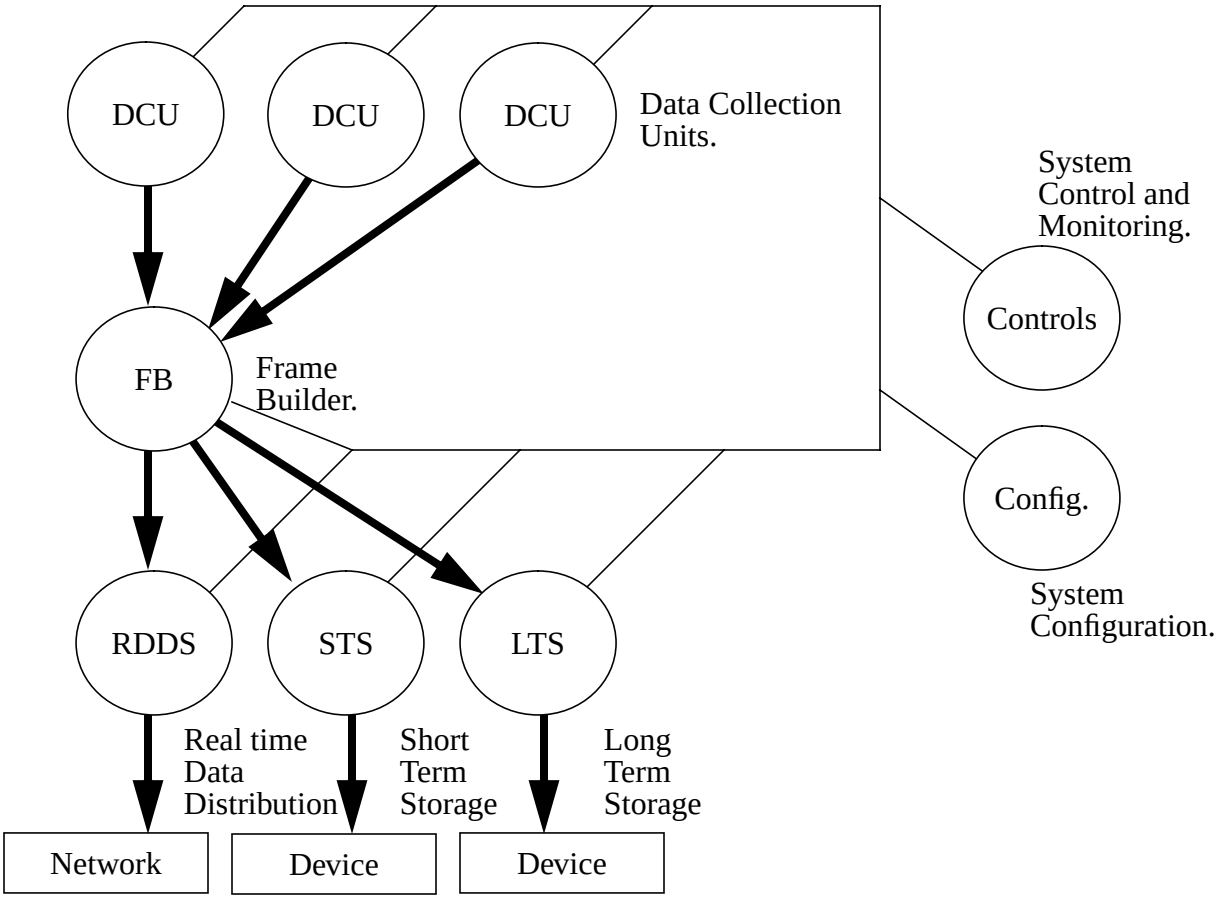


Figure 2: DAQ Functional Overview.

3 DESIGN OVERVIEW

3.1. 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.

Eleven 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 (512 to 16K samples/sec). The seven 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. A first cut at detailed layouts of these units is shown in Figure 1 of Appendix A.

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 (1Hz) 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 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. Data is also reflected to the Interferometer Diagnostic System (IFODS), which provides “live” data feeds and analysis for on-line systems.

Interconnection of the DCU and framebuilders is to be done using a reflective memory network. This is a high speed optical network (30MBytes/sec) which automatically copies data put into memory by one processor into identical memory locations in all units on the network.

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.

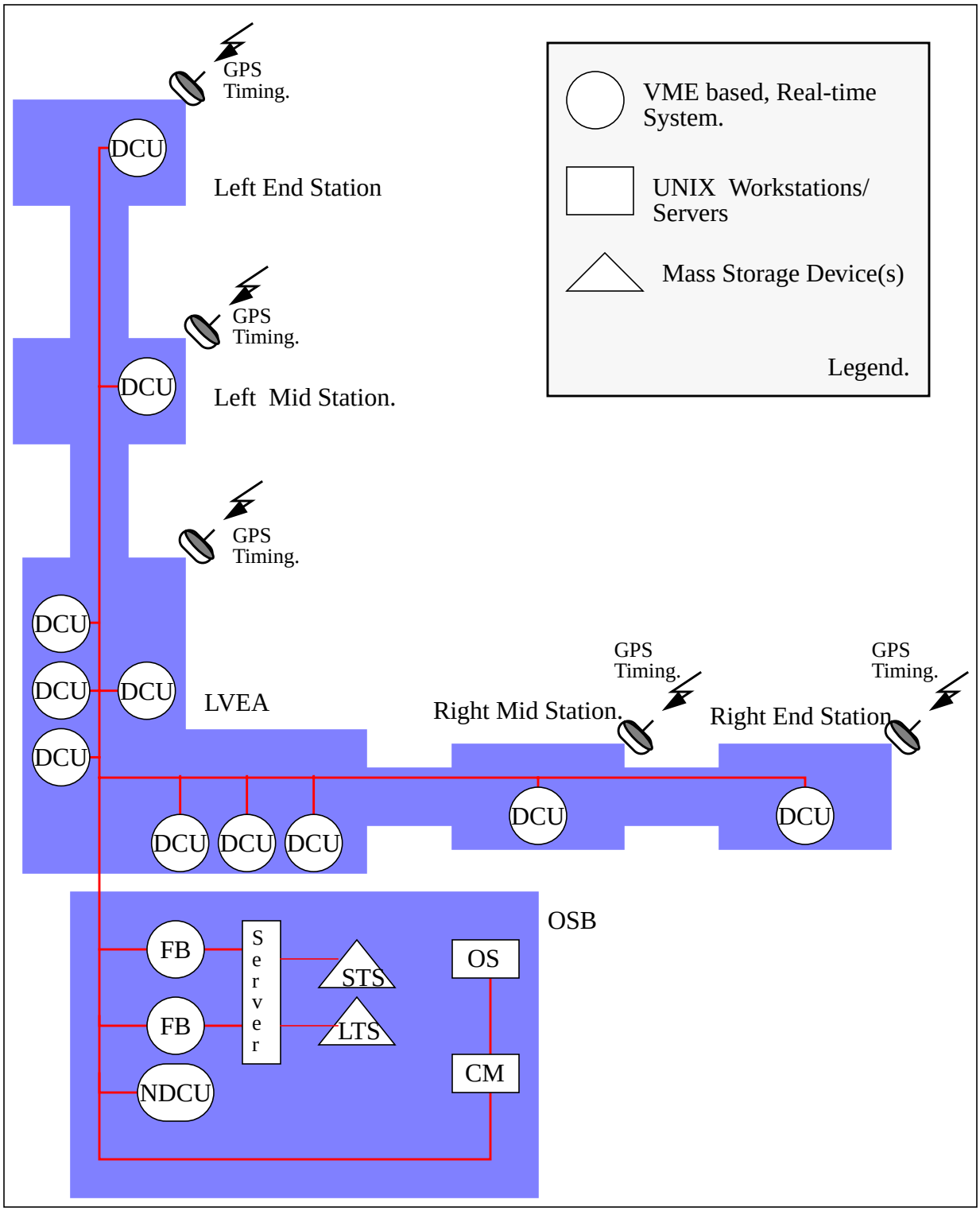


Figure 3: Hanford Site DAQS Configuration

3.2. Basic Data Flow

Figure 4: DAQS Dataflow shows a simplified diagram of how data will flow through the DAQS. Signals are digitized at the ADC and stored within on-board memory for 1/16 of a sec (1024 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.

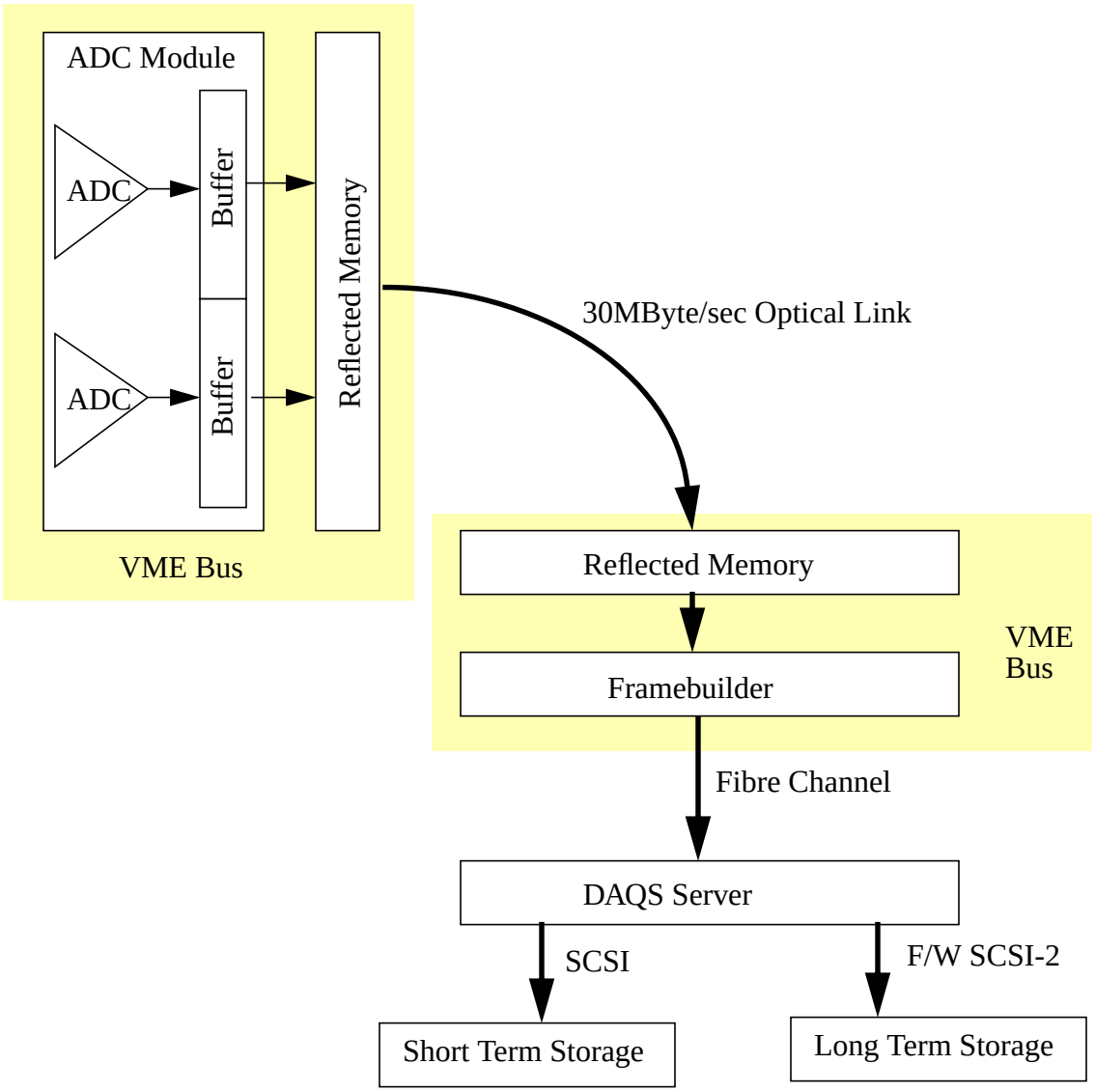


Figure 4: DAQS Dataflow

3.3. Interfaces

The interfaces to the DAQS are shown in Figure 5: DAQS Interfaces.

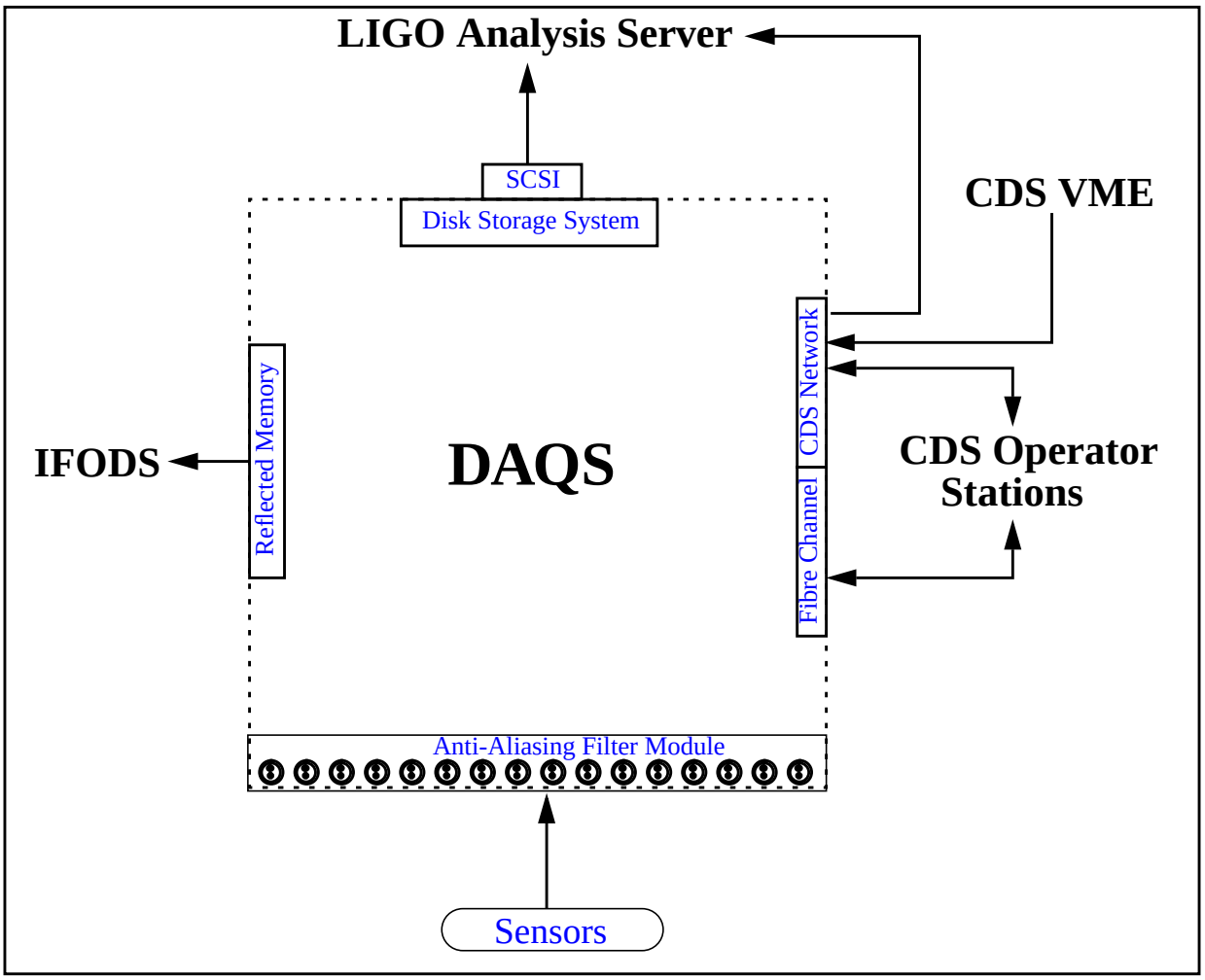


Figure 5: DAQS Interfaces

3.3.1. Sensors

The various sensors from which data is to be acquired are provided as part of other LIGO systems. The common interface is at differential LEMO connectors at the front panel of DAQS provided anti-aliasing filter modules. (Note: Since various LIGO systems are digital in implementation, direct digital interfaces may be provided along with the analog interfaces described in this document. This will be further investigated during the final design phase of the DAQS and as designs proceed on control and monitoring systems which could provide digital information directly.)

3.3.2. LIGO Data Analysis Systems

Computer and software systems for the analysis of LIGO data are to be provided by others. The primary interface to these systems will be at a second SCSI disk controller in the DAQS storage system. This will allow a data analysis server to directly connect to the DAQS file system. In addition, a network connection via the CDS network infrastructure will be provided to pass data file and configuration information.

3.3.3. CDS VME

Slow data (1Hz) is to be monitored from various CDS VME crates. This interface will be the CDS network backbone.

3.3.4. CDS Operator Stations

Operator consoles and addition compute servers are to be provided as part of the control and monitoring function of CDS. Connection to these components will be via the CDS network backbone and a private fibre channel network.

3.3.5. Interferometer Diagnostic System (IFODS)

The IFODS is to be connected directly to the DAQS reflected memory networks, thereby providing high speed, independent data access for interferometer diagnostic software.

4 DATA COLLECTIONS UNITS

The functions of the DCU are to:

- Synchonously digitize the input signals from various LIGO sensors.
- Read data from the ADC modules and send the data to the framebuilders over reflected memory.

The layout for a typical DCU VME crate is shown in Figure 6: DCU Components and Connections.

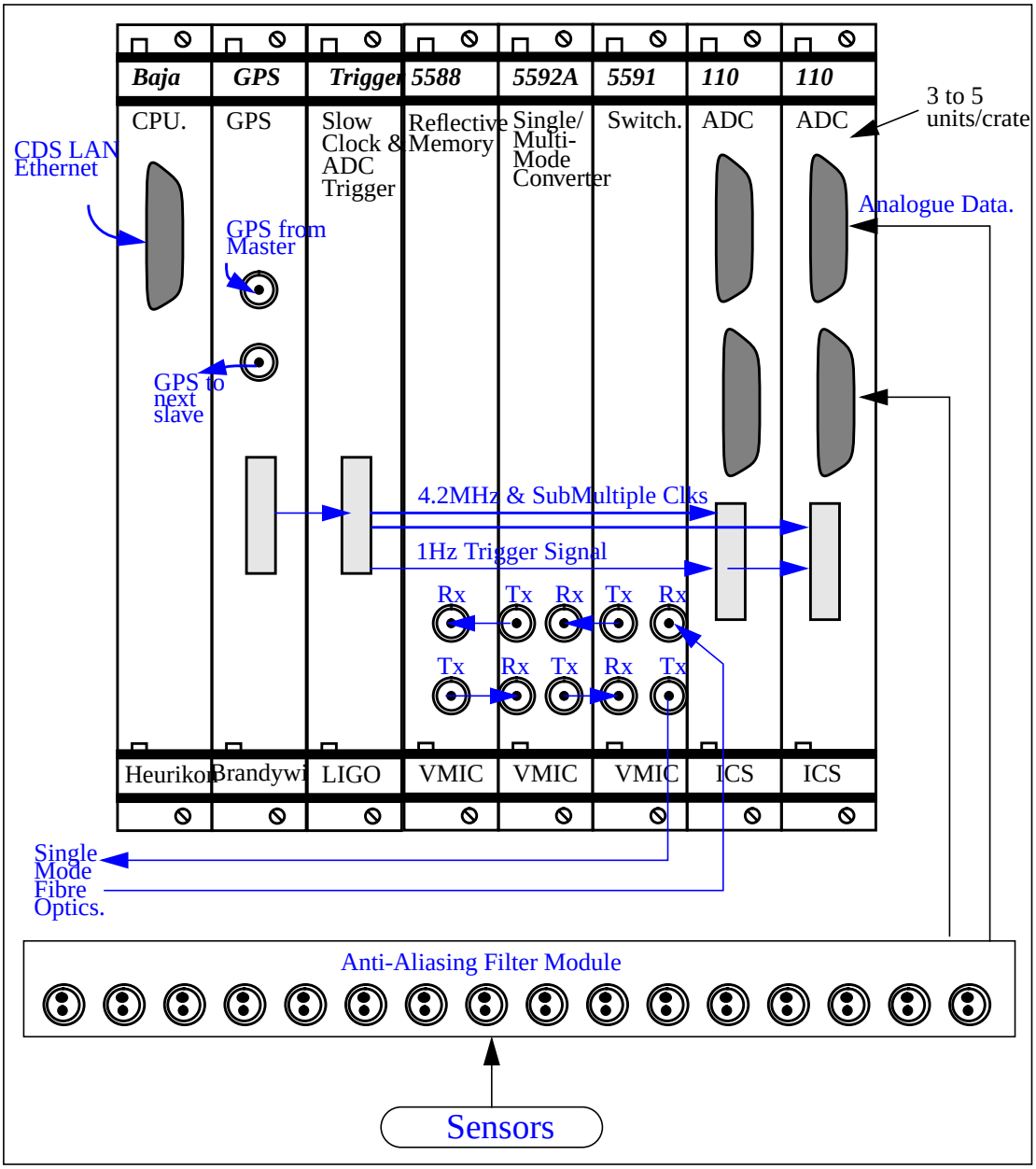


Figure 6: DCU Components and Connections

4.1. DCU Hardware

The following subparagraphs describe the various electronic components which comprise a DCU.

4.1.1. Anti-aliasing Filters

Data is to be acquired by the system at three different rates(512,2048,16384Hz), with anti-aliasing filters provided at 256,1024 and 8192Hz. All anti-aliasing filters are intended to be 4 pole butterworth low pass filters. In addition, each filter channel has a comparator which flags overvoltage at the input. The units designed and prototyped for the DAQS are rack mounted, with single printed circuit boards containing 16 channels each. Front panel connections (to sensors) are differential LEMO connectors. Connection to ADC modules are via ribbon connectors.

4.1.2. 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.

4.1.3. 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.
- Clock for ADC channels (16384Hz)
- 1Hz VME interrupt.

4.1.4. Reflective Memory

4.1.4.1 VMIC 5588 Reflective Memory Module

The 5588 is the reflective memory card. The 5588 provides:

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

4.1.4.2 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.

4.1.4.3 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.

4.1.5. ICS110 32 channel 16 bit ADC.

The ICS ADC module (or similar) will be used to digitize 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(s) may be treated separately during following design phases.

The ICS110 provides:

- 32 individual, differential input, 16-bit, no missing codes, sigma delta ADC
- Input range: +/-1V
- Integral and Differential Nonlinearity: +/-0.0053%
- 390 Hz to 200 kHz sampling rates.
- 64K sample onboard storage.
- A32 D32 BLT Slave VME data readout (Data access @ 32 bits/210 nsec).
- VME interrupt generation when FIFO is half-full.

4.2. Operational Overview

An overview of the DCU operation, module connections and reflected memory layout is shown in Figure 2 of Appendix B and described as follows:

- The ADC modules receive their clock from the GPS module at 4194304 Hz. This produces an output of the ADC module at 16384Hz.
- Once 1024 samples of each channel are written to the FIFO of the ADC module (16Hz), a bus interrupt is generated by the ADC module on the VME backplane and captured by an Interrupt Service Routine (ISR) in the DCU CPU.
- The DCU software retrieves the 1024 values for each channel from the ADC FIFO and places them into the reflected memory, automatically causing it to be transferred to the framebuilders. It then writes designated reflected memory locations (cycle counter) which generate an interrupt at the framebuilders to signal that new data is available.

A further description of the DCU software is shown in Figure 7: DCU Software Flow Diagram. Here, the sequence is basically as follows:

- On power up or reset, the DCU boots and receives initial software parameters via the CDS ethernet. It sets a status flag in the reflected memory to indicate to the framebuilder that it is initialized and waiting configuration. After the initial boot from the CDS ethernet, all other communications with the DCU are through a header area of the reflected memory.
- The framebuilder controller sends configuration information and a configuration command. This information includes: ADC module data rates, memory locators, individual channel data rates.
- The DCU, once configured and ready, will set a status flag in the reflected memory header indicating that it is ready to acquire data.
- When ready, the framebuilder will send a command to the DCU to begin acquisition. This command is sent ~50msec after a GPS 1Hz clock pulse. The DCU will then arm the ADC clock module, allowing the next 1 Hz clock pulse from the GPS module to trigger acquisition of data by the ADC modules. In this manner, all ADC modules in the system are synchronized.
- The DCU software now waits the 16Hz interrupt from the ADC modules. On receipt, the software will (1) Read the ADC FIFO, (2) write the data to reflected memory, and (3) set an interrupt to the framebuilder that data is ready.
- Once each second, the DCU software will check the overrange bits of the anti-aliasing filters and write out the summary statistics for a one second period.
- Reflected memory is checked to determine if new commands have been issued by the framebuilder, such as new configurations. If no new commands have been received, the DCU software continues in a loop acquiring data.

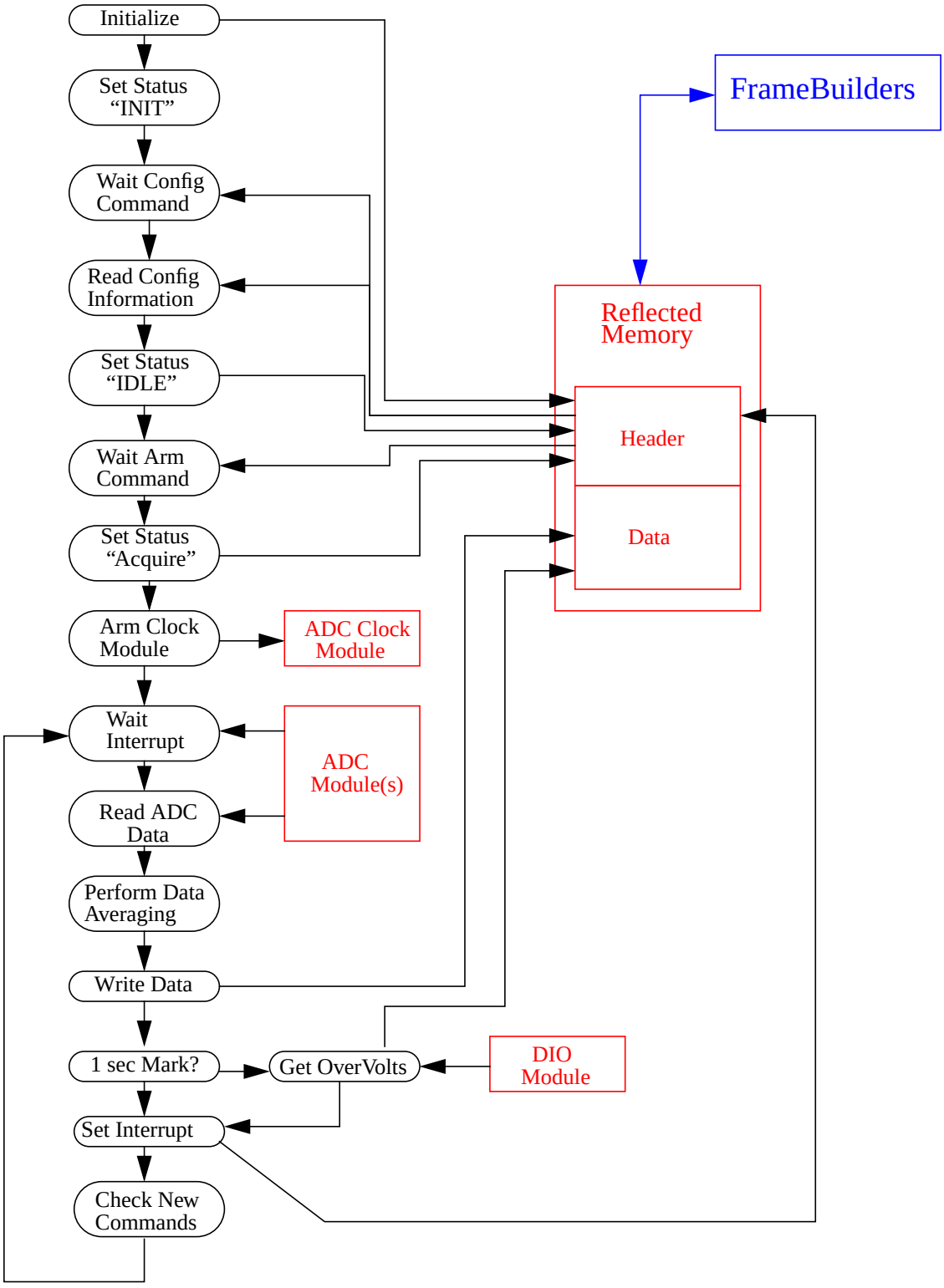


Figure 7: DCU Software Flow Diagram

4.3. Process Timing

The prototype DAQS performs in much the same manner as described above for a DCU.. The following timing numbers are indicative of those found in the prototype system.

Based on the data channel requirements of the DAQS DRD and the preliminary design layout, the most heavily loaded DCU is the PSL/Suspensions crate. This unit is designed to handle 64 channels at each of 256, 2048 and 16384Hz. This corresponds to ~1.2 M samples/sec or 2.4MBytes/sec. Using this example and test results from the DAQS prototype, the timing chart of Figure 8: DCU Event Timing is developed.

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 16K, 2048 and 256Hz. Each ADC sample is stored on the ADC module in a FIFO. This buffer is software configured to hold 2048 data words for each ADC channel.

At point (2), 1024 samples of each ADC (running at 16KHz) have been taken and a “FIFO half full” interrupt is generated by the ADC module and detected by an Interrupt Service Routine (ISR) in the CPU. This ISR in turn causes the readout across the VME bus of the ~75Kwords of data which is in the ADC FIFOs. The total time to move this data into the CPU is ~19msec. Transmission time of this data from the CPU on to the reflected memory and framebuilder is an additional 5msec. Therefore, total time to move data through a DCU configured for this many channels is ~24msec out of the 62.5msec between data interrupts, allowing ~37.5msec for any data processing, such as window averaging and data summaries described previously.

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 and summary information. 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.

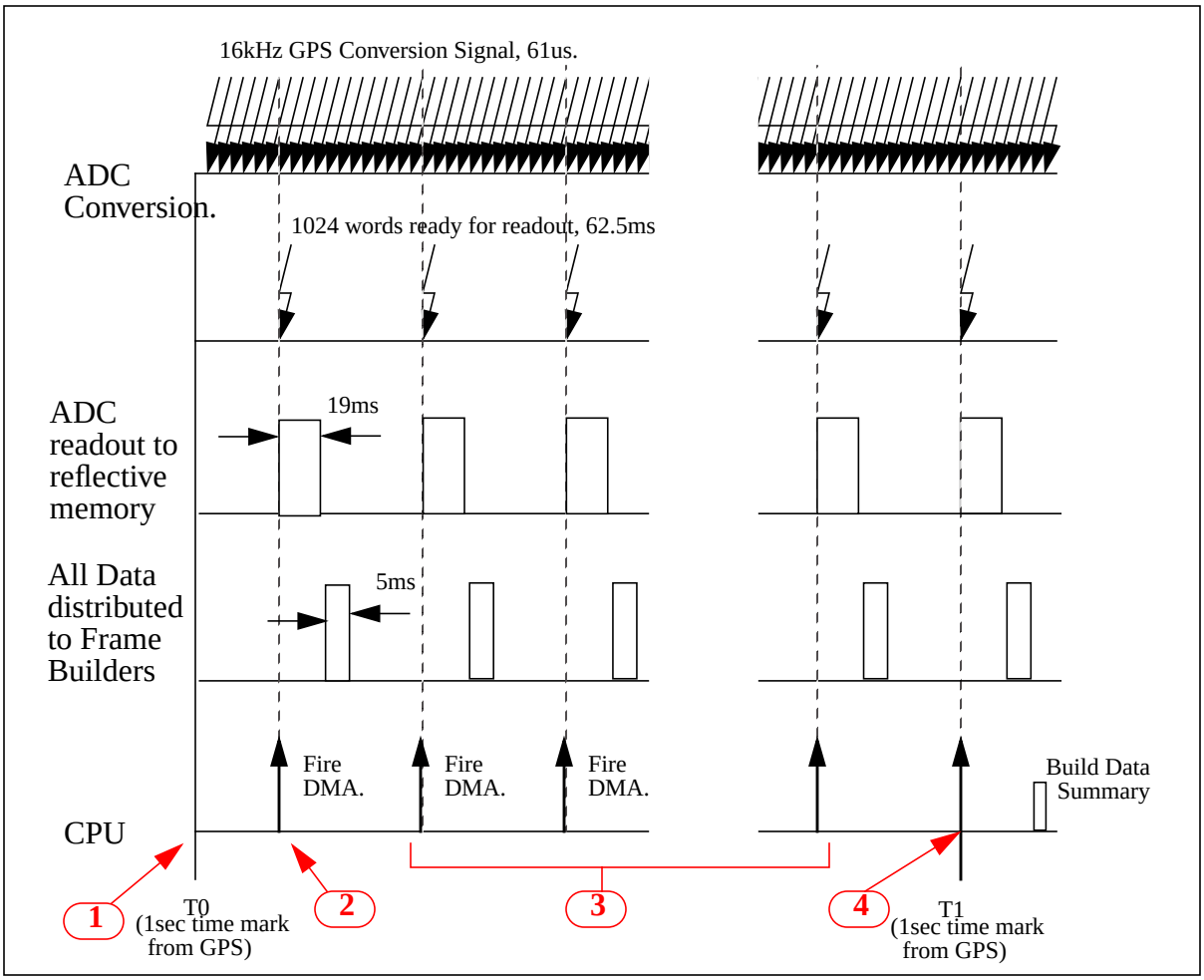


Figure 8: DCU Event Timing

4.4. DCU Configuration, Control and Monitoring

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: 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 (window 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.

4.5. 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 (1Hz). 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.

5 DATA TRANSPORT

For moving data from the DCU back to framebuilders, relected memory is the present choice. Fibre Channel (FC) is also be considered as a possible alternative (due to its lower cost), pending outcome of further prototype testing. 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 the DAQS layout of Figure 1 of Appendix A . The Hanford DAQS network will be set up as seven independent rings, three for each interferometer and one for PEM. 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 one of the framebuilders. Network rings which have multiple crates will also contain bypass switch units, which maintain network continuity even when one crate in the network is powered down.

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. Memory will be sized such that 1 second of data is always resident within each framebuilder. The layout of data in the reflected memory is shown in the diagrams of Figures 2 and 3 of Appendix A.

6 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 9: Central Processing Equipment.

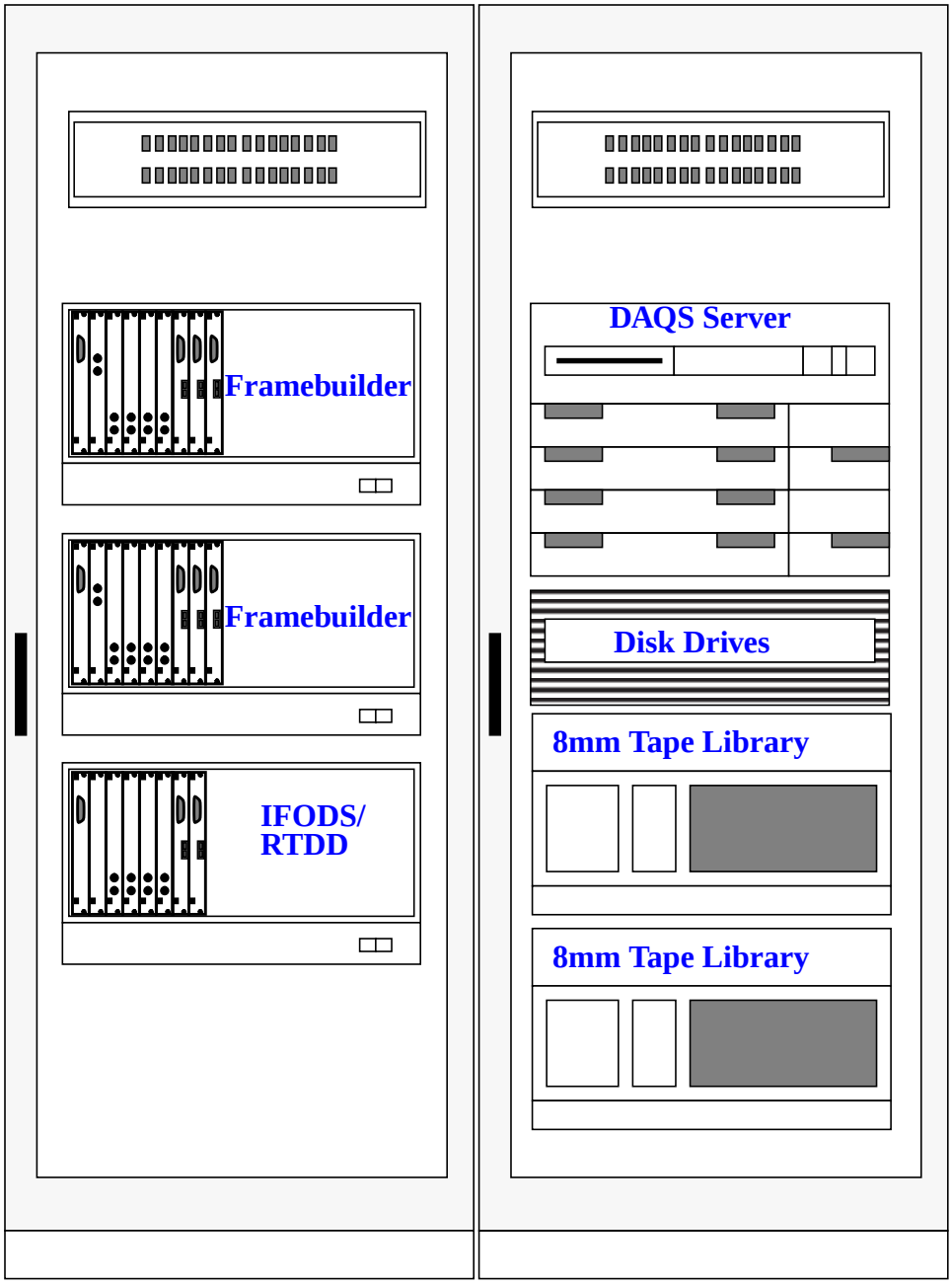


Figure 9: Central Processing Equipment

6.1. Framebuilder

The framebuilders format the data received from the DCUs into a standard frame format, as described in *LIGO-T971030-00-E Specification of a Common Data Frame Format for Interferometric Gravitational Wave Detectors (IGWD)*. Basic data flow and software components are shown in Figure 3 of Appendix A.

Three types of frames are produced by the framebuilders:

- Full Frame: Contains data from all channels being acquired by the DAQS.
- Analysis Frame: Contains a selected subset of data channels, typically those of primary interest in first glance gravity wave analysis.
- Trend Frame: Contains all slow data channels (1Hz) and summary information (max, min, average and standard deviation) of all other data channels for a 1 second period. This allows for many hours of trend information to be quickly retrieved and displayed without the need to sift through all of the larger full and analysis data frames.

All frames produced by the framebuilders are one second in duration. These are then passed on to the DAQS server via a fibre channel network.

6.1.1. Hardware

There is to be one framebuilder for each interferometer, with key components are as shown in Figure 10: Framebuilder Layout. The modules shown provide the following functions:

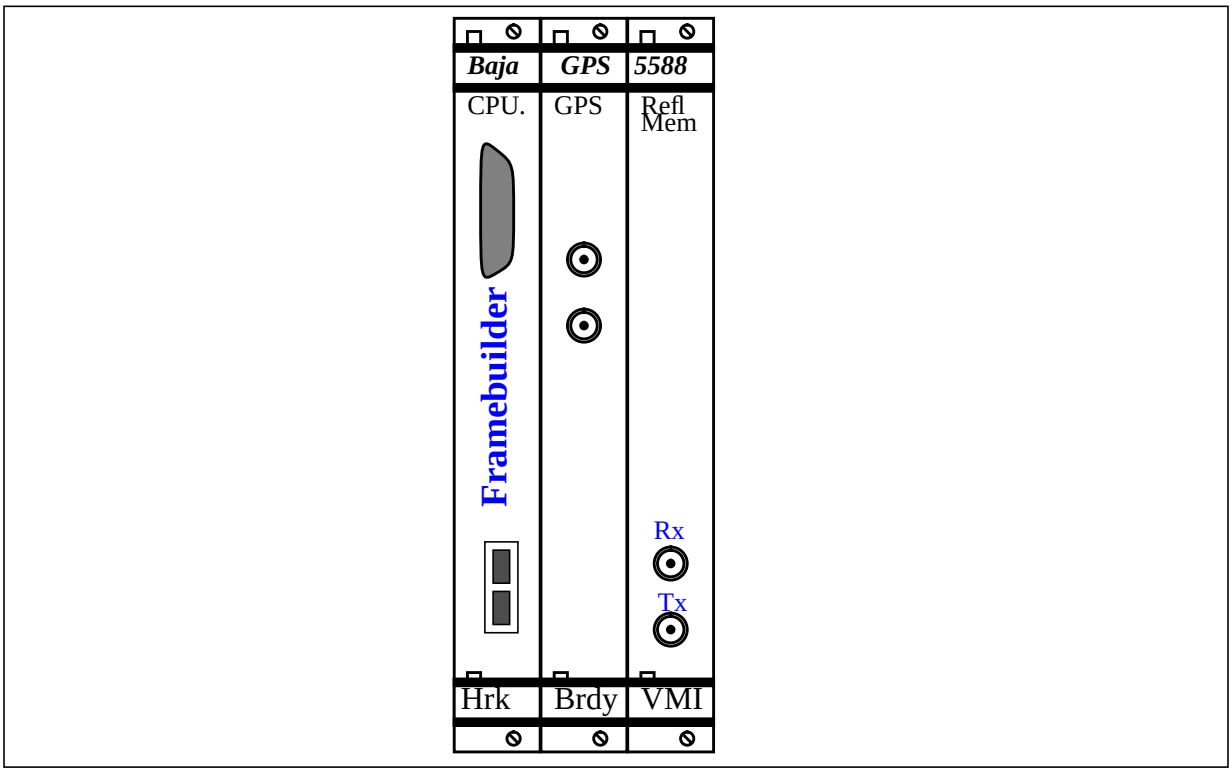


Figure 10: Framebuilder Layout

1. Framebuilder: A real-time CPU, this unit controls and monitors the acquisition process and formats all data into standard frame formats. This may be multiple CPU units, dependent on CPU loading.
2. GPS module for providing timing information to the framebuilder.
3. Seven reflected memory modules, which, as previously described, continuously contain the last second of data from all DCU.

6.1.2. Software

The primary software components to the framebuilders is shown in the diagram of Appendix A Figure 3 and described in the following subparagraphs.

6.1.2.1 FB Controller

The FB Controller software provides communication functions to other CPUs within the DAQS and to operator systems and provides software coordination within the framebuilder. This is to be an EPICS based software module, thereby allowing EPICS channel access to be used as the communication media.

6.1.2.2 Configuration Manager

To configure the DAQS, configuration files are produced using an off-line editor. On operator command, these configurations are sent to the framebuilders and received by the configuration manager. This software then parses the necessary information to the DCU and framebuilders. As shown in the software diagram, each framebuilder CPU contains two framebuilder software modules. One is active as any given time, with the second awaiting new configuration information from the configuration manager. The two framebuilder code modules allow for seamless transition to a new data acquisition configuration ie system does not need to stop to process a new setup.

On receipt of a new configuration command via the FB controller, the configuration manager will retrieve the configuration information from a named configuration file on the DAQS server. Settings are then sent to the DCU via the header areas of the reflected memory and to the framebuilder code module which is presently inactive. Once all DCU and the framebuilder signal the FB controller that they are ready to implement the new configuration, the FB controller will signal the DCU and framebuilder to synchronously switch. The framebuilder code which was active will now become inactive, waiting for the next new configuration and the newly configured framebuilder code module will begin actively formatting and transmitting framed data to the server.

6.1.2.3 Framebuilder Software

The basic sequence of the framebuilder code module is shown in Figure 11: Framebuilder Software Flow Diagram. The intialization and configuration blocks perform similarly to those described for the DCU. Once set to acquire, the framebuilder will wait for an interrupt from a DCU indicating that data is ready. Data is then read from the reflected memory and written into the frame structure. This will continue at 16Hz until one full second of data is framed. This frame is then shipped to the DAQS server via the 1Gbit/sec fibre channel network.

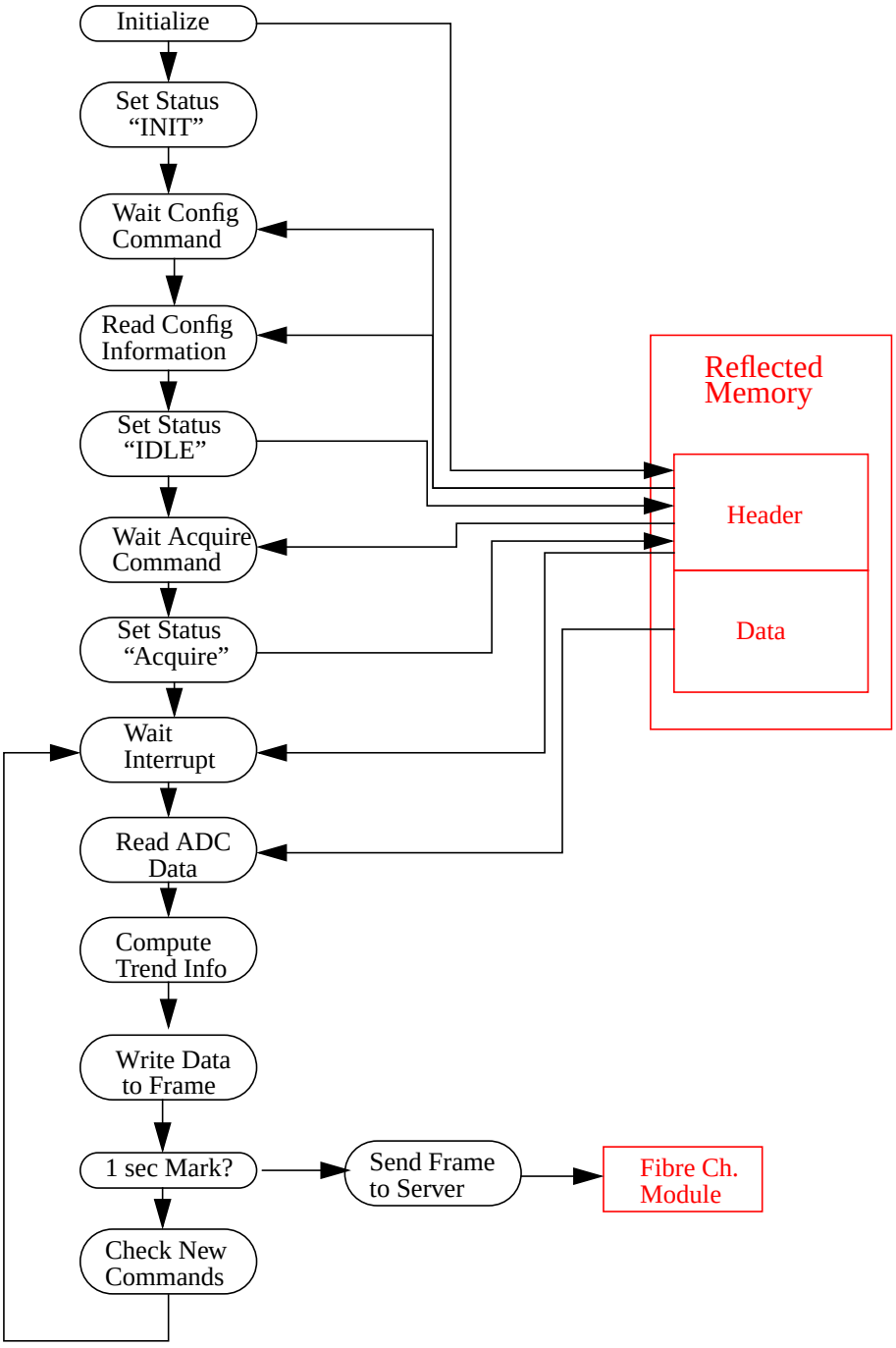


Figure 11: Framebuilder Software Flow Diagram

This model is for the full and analysis framebuilders. In the case of the trend framebuilder, it is interrupted only once per second. It then extracts all the 1Hz data channels from the reflected memory, along with all of the trend data. This data is NOT framed by the framebuilder, but rather passed along to the DAQS server, which will perform the framing function.

6.2. 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 (***Note: The DAQS will only provide data services to operator stations and other CDS subsystems. The role of providing data services to all systems outside of CDS will be the responsibility of the LIGO data analysis system***).
- Managing the DAQS configuration and configuration databases
- DAQS error reporting and logging
- Framing the trend data passed on by the framebuilder.
- Writing frames to mass storage devices.

6.2.1. Hardware

The architecture of the server side of the DAQS in shown in Figure 12: DAQS Server and Interfaces and described in the following subparagraphs.

6.2.1.1 Compute Server

The computer for the DAQS server is to be an UltraSparc 3000 (or equivalent), configured with up to six processors and 1GByte of RAM. As depicted in the figure, this unit contains a number of interfaces to the other DAQS components, including:

- SCSI to disk drives
- SCSI to tape drives
- Fibre Channel, run in arbitrated loop mode, to the framebuilders. This is the data flow path for DAQS data to the server. In initial prototype testing with the processors intended to be used by LIGO, throughput rates >35MBytes/sec have been measured on this network.
- Second fibre channel network connecting the IFODS processor(s), server and operator console computers. This network is intended for the transfer of interferometer diagnostic information extracted from the DAQS.
- Ethernet and/or ATM connection onto the CDS networks.

6.2.1.2 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 systems, 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 (~100MByte), 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.

As shown in the DAQS server diagram, this disk storage system would also be the primary data interface to the data analysis system. The unit will be configured with two host controllers, such that the DAQS would write frames to disk and the analysis system can independently extract data from the same disk units.

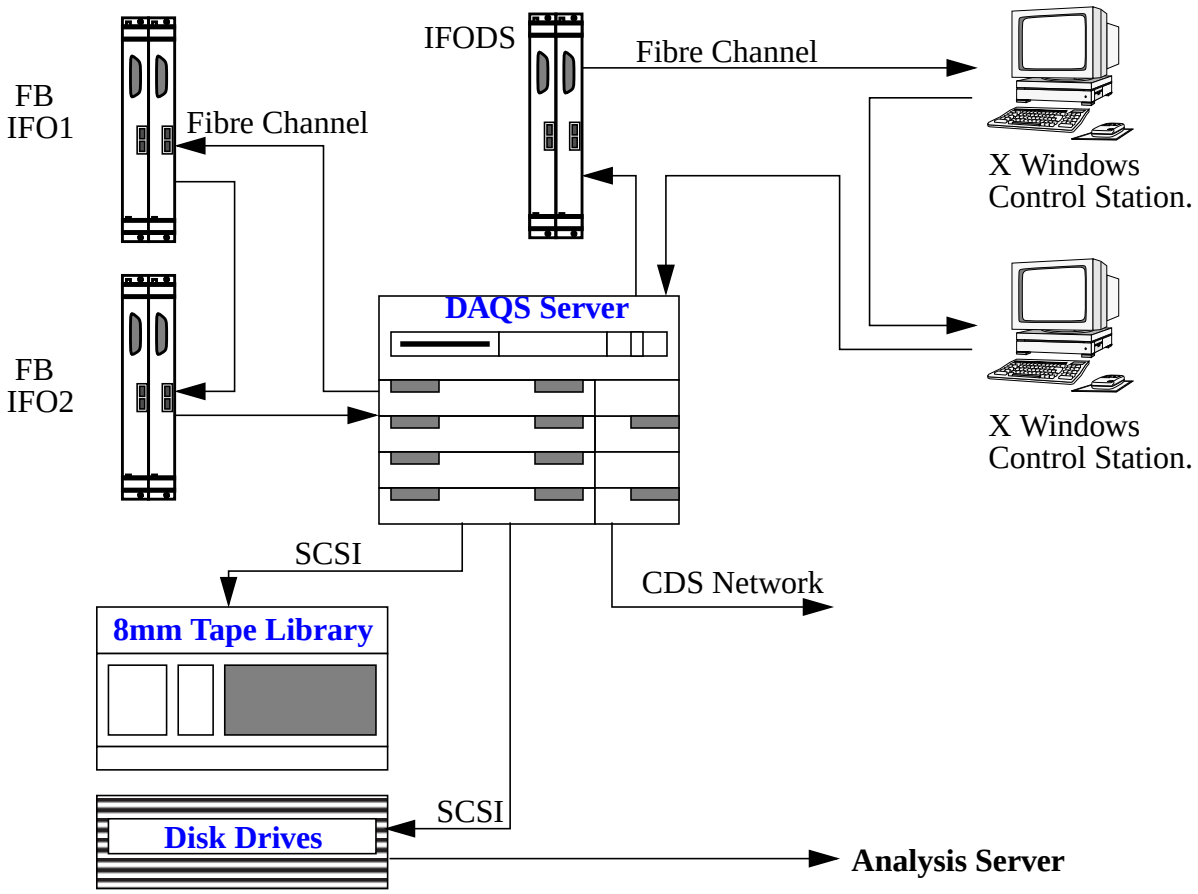


Figure 12: DAQS Server and Interfaces

6.2.1.3 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 240GByte, 8mm robotic (10x24GBytes/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.

6.2.2. Software

An overview of the software to run on the server is shown in Figure 13: Server Software Basic Block.

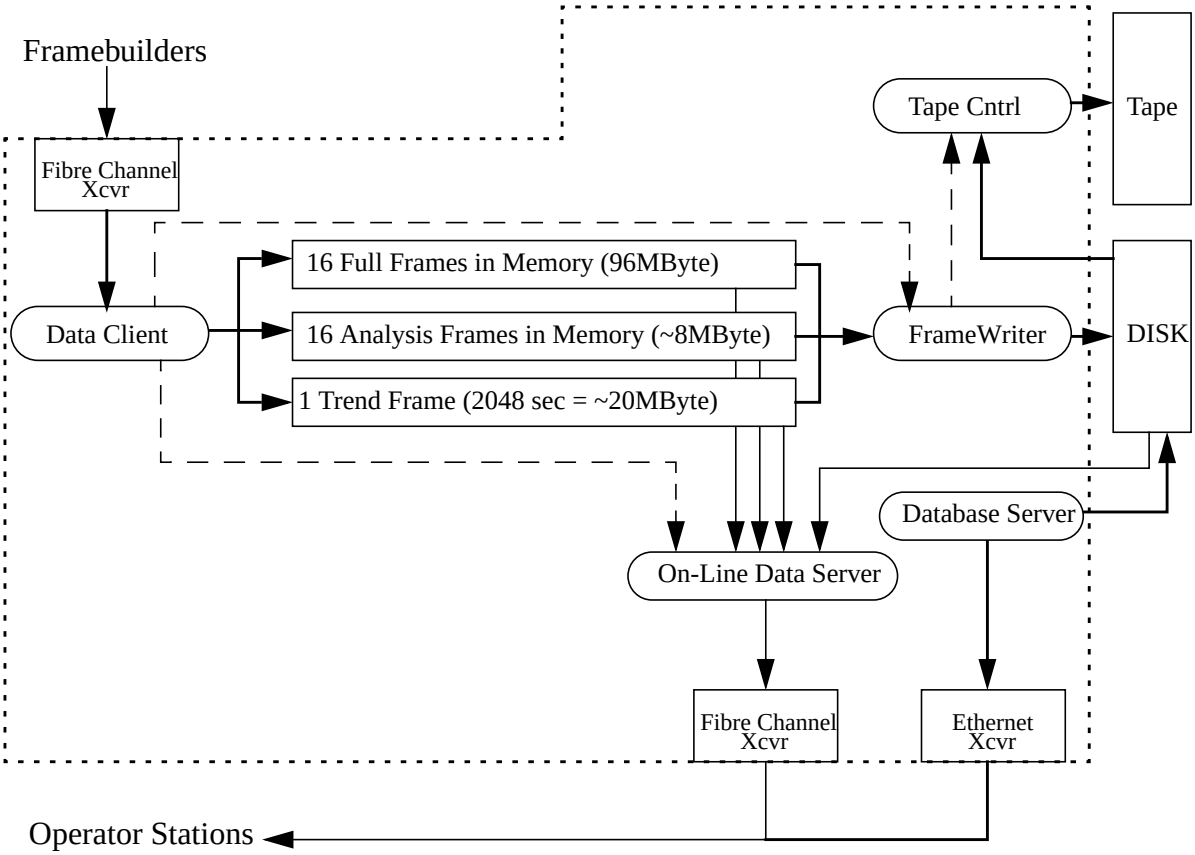


Figure 13: Server Software Basic Block

6.2.2.1 Data Flow

6.2.2.1.1 Data Client

Data is received by the server via data client software which receives full and analysis frames and trend data buffers from the framebuilders via fibre channel once per second. This data is then moved into shared memory buffers for access by the other server software components.

6.2.2.1.2 Frame Writer

On signal from the data client software, the frame writer process writes the frame to file. While the standard frame size for full and analysis frames will be one second, the number of frames per file is configurable from 1 to 32. This software will also generate and store trend frames, the standard trend frame being 64 seconds in duration, again configurable to store 1 to 32 frames per file.

Once the frame writer has written a file, it broadcasts the name of the new file over the CDS networks. This signals the LIGO analysis systems that new data is now available on the disk system and where it is located.

6.2.2.1.3 Tape Control

The DAQS disk drive unit is set up in a circular buffer arrangement, such that once full, it begins to remove old frame files to write the new frame files. As an example, the arrangement from the DAQS prototype is shown in Figure 14: Tape Storage Sequence and the LIGO DAQS will be arranged in a similar manner.

At present, directories are set up on the disk such that the frame writer produces frame files in a directory for one hour. After an hour, the frame writer starts writing frame files to the next directory and signals the tape control process. The tape control process now backs up the previous hour of data to tape and erases all data files from the directory one hour in advance of where the frame writer is presently active. In the prototype, this continues such that the previous 8 hours of data are always available on disk.

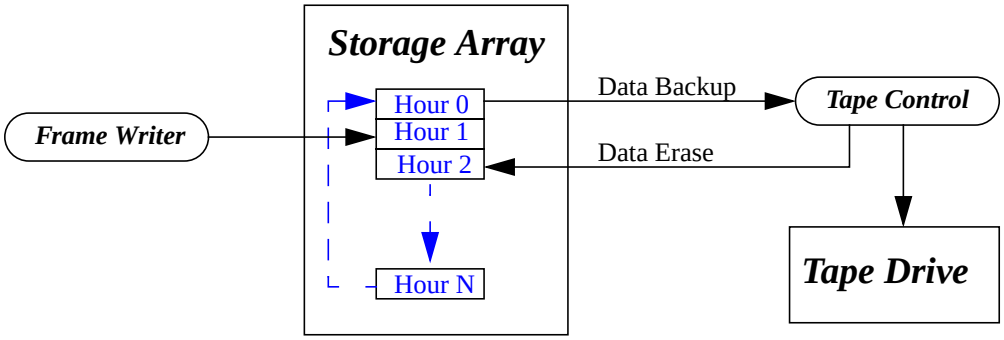


Figure 14: Tape Storage Sequence

The tape control process also controls the tape robot unit. The units proposed for initial LIGO commissioning contain two drives and 10 tapes in a robotic unit. The tape control process monitors and controls this system, automatically removing tapes as they become full and inserting new tapes.

6.2.2.1.4 Database Server

While the bulk of information necessary to perform gravitational wave analysis will be contained within each standard data frame, there is anticipated to be a not insignificant amount of information about the configuration of LIGO interferometers and its data channels which may be impractical to store in every data frame. This is data which does not change very often (days, months), but is necessary to note for later analysis. Some examples of this type of information are:

- Additional information describing the data channels, such as normal ranges, calibrations, conversions, etc.
- Electronic module serial numbers and calibrations and related component maintenance/ replacement at the time of data taking.
- Electronic module or other interferometer settings/parameters which are unavailable to the DAQS, either directly or through the CDS, and must be noted by operators into an electronic log.
- Operator logs and observations.

Further defining this information set and where/how to store it is still TBD during the final design phase. Because it may present a high overhead to store all of this information in every data frame, the initial working concept is to place this information into an on-line database. The standard data frames would then be tagged with pointers, which could be checked quickly by analysis software to determine if the database has been changed and effects the analysis and where to look for the new information in the database.

6.2.2.2 Data Storage

6.2.2.2.1 Disk

The DAQS disk drive system will be partitioned as follows:

- System Software: Directory structure will contain all software necessary to boot and operate the DAQS.
- Full Frame Partition: Will contain data directories to support a circular buffer arrangement for up to 8 hours of LIGO data in full data frame formats.
- Analysis Frame Partition: Identical to above, but set up for analysis frames.
- Trend Frame Partition: Again, setup as above, but will contain a minimum of 1 month of trend information.
- Database Partition: Supports the database server.

Frame files will be written with a standard naming convention, such as C1-97_05_12_23_14_01, with the following fields:

- First two characters (C1 in example) denote the interferometer, with:
    - C = Caltech
    - M = MIT
    - H = Hanford
    - L = Livingston
- Remaining fields denote the year, month, day, hour, minute and second (GMT) that the first frame within the file was acquired.

In addition, these file names contain a suffix to identify the type of frames stored in the file:

- No suffix indicates a full data frame file (C1-97_05_12_23_14_01)
- .A denotes an analysis frame file (C1-97_05_12_23_14_01.A)
- .T denotes a trend frame file (C1-97_05_12_23_14_01.T)

6.2.2.2.2 Tape

Storage to tape will be as standard Unix tar files or similar. The present prototype uses a proprietary storage format particular to the tape unit which provides for faster file access by maintaining a directory structure on the tape of all the files written to it. Such a method may or may not be used in the final DAQS.

6.2.2.3 Data Distribution

Data is distributed within the LIGO CDS system by the On-Line Data Server software running on the DAQS server and by the IFODS.

6.2.2.3.1 Framed Data

A separate fibre channel network is run between the DAQS server and the CDS operator stations and TBD compute servers. Data, in the form of frames, is available on this network on request from the On-Line Data Server software module running on the DAQS server. Real-time data is available once per second from the DAQS server shared memory and older data available from disk.

6.2.2.3.2 Unframed Data

Data for software running various diagnostics on CDS computers can be obtained from the IFODS. This data is not framed, but rather requested by signal name and rate. This data is available on the fibre channel network every 1/16 second, the interrupt rate of the DAQS. More information on this data distribution system will be provided in IFODS design documentation.

6.2.2.4 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 graphics 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.

6.2.2.4.1 DAQS Configuration

As shown in Figure 15: DAQS Configuration, the DAQS will be configured through a TBD database via a user interface. This database will, in turn, produce four ascii files which describe the required configurations. The first file, full system configuration, is set up for use by the system DCU and framebuilders. This file describes the actual configuration of the DAQS itself, including ADC modules, reflected memory addresses, etc. The other three files describe the frames to be built by the DAQS, one for each frame type. This information is for use by the framebuilders. The actual format of these files is still TBD.

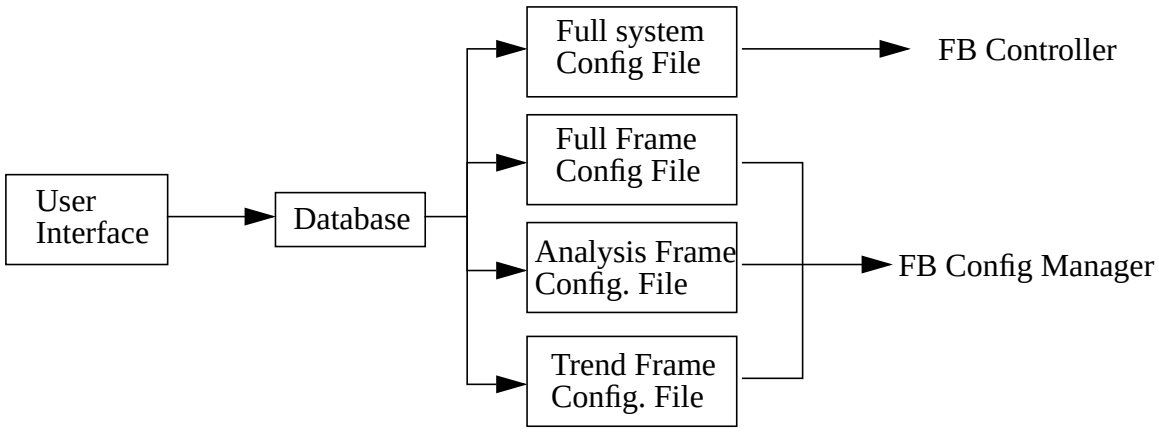


Figure 15: DAQS Configuration

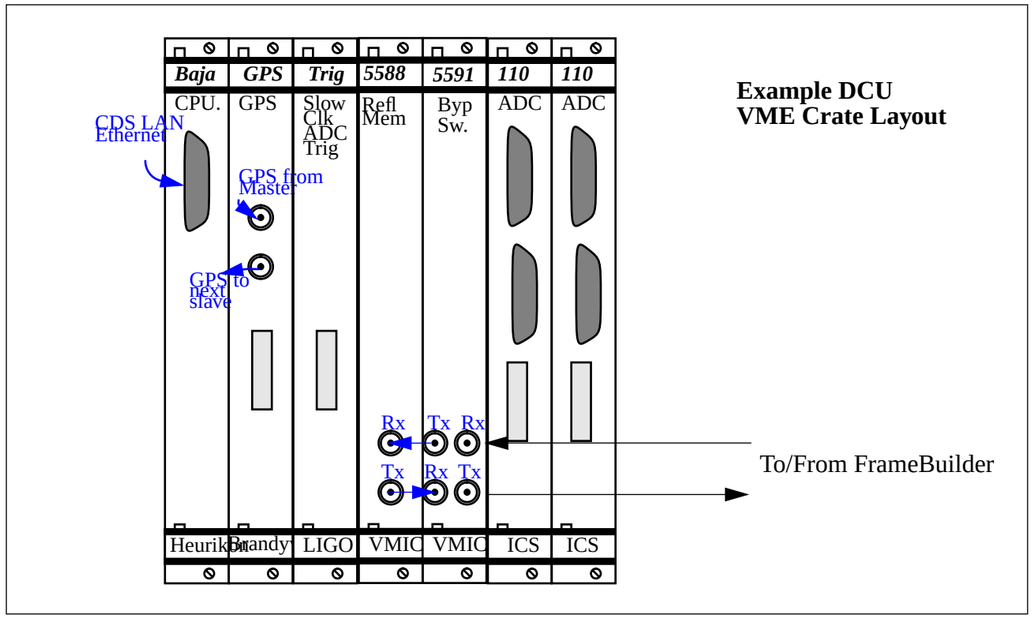
6.2.2.4.2 DAQ Error and Message Logging.

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 DAQS DIAGRAMS

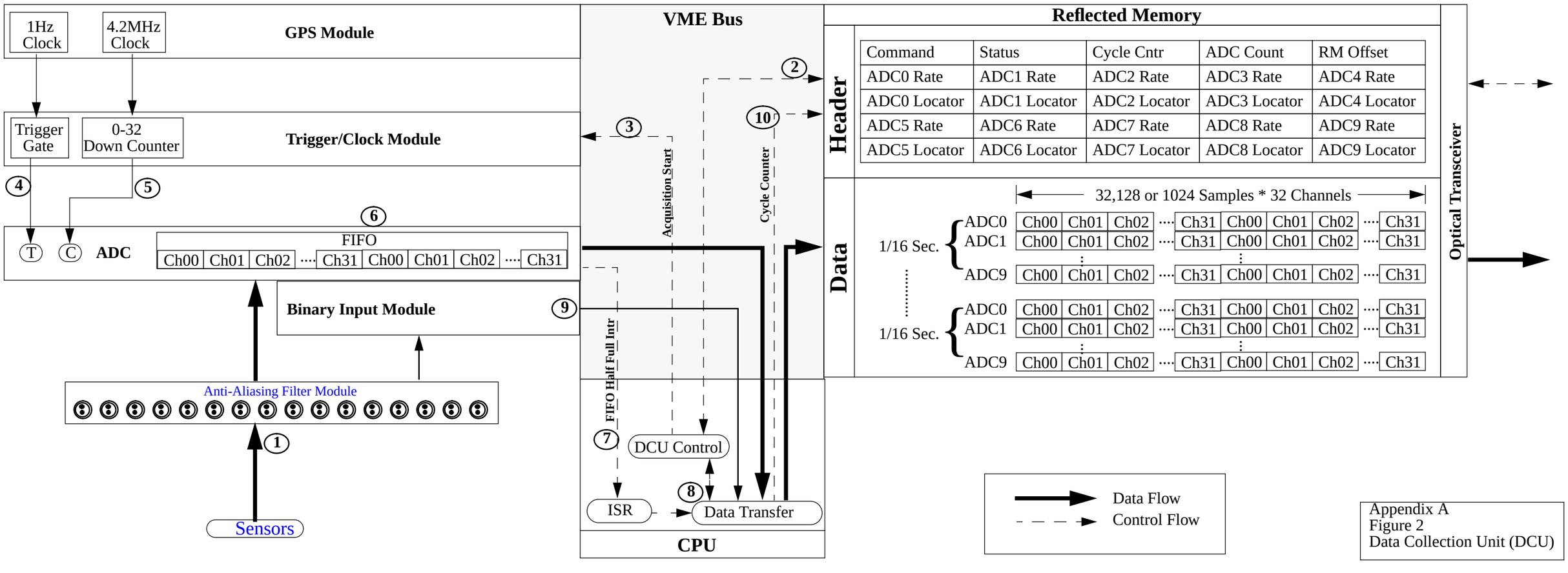
Appendix A
Figure 1
DAQS Overview



Example DCU VME Crate Layout

Operational Description

1) Sensors are connected to the DCU via differential LEMO connectors at the Anti-aliasing module.
2) After processor boot, the DCU CPU receives configuration information and startup commands from the framebuilder via a designated header area of reflected memory.
3) On receipt of a start acq. command from the framebuilder, the DCU control software writes a start command to the Trigger/Clock VME module.
4) On the next 1Hz clock output from the GPS module, the Trigger/Clock module sends a trigger signal to the ADC modules to start acquisition.
5) The 4.2MHz clock from the GPS module is downcounted to provide the appropriate clock to the various ADC modules, based on their desired acquisition rate (512, 2048 or 16384Hz).
6) The ADC modules digitize the sensor signals and accumulate data in an onboard FIFO.
7) When the FIFO is half full, an interrupt is generated by the ADC module (every 1/16th of a second when acquiring at 16384Hz) and acknowledged by an Interrupt Service Routine (ISR) in the CPU.
8) The ISR signals a data transfer thread, which copies the half FIFO of data (1024 points x 32 channels) from the ADC to the defined locations in reflected memory.
9) Once per second, following an interrupt from the ADC, the data transfer thread will also read the binary input module, which will contain any overvoltage bits set by the anti-aliasing filter for the previous one second.
10) The data transfer thread, when complete, will increment the cycle counter in the reflected memory header. This, along with the data, is automatically transferred to the reflected memory located in the framebuilders. The writing of a new cycle count will cause an interrupt at the framebuilders, indicating that new data is available.



Appendix A
Figure 2
Data Collection Unit (DCU)

Operational Description

1) The reflected memories of the various DCU are mapped into the larger reflected memory modules of the framebuilder.
2) On startup, the FB Controller software reads a DCU configuration file and passes the information to the DCUs via the header sections of the reflected memory. It then verifies that all DCU report ready for acquisition.
3) On operator command, the FB Controller receives a data configuration request and file pointer, which are passed to the Configuration Manager.
4) The Configuration Manager reads the configuration file(s) and sends the appropriate configuration information to the framebuilder code module which is presently in an idle state. (Note: On startup, both will be idle, but during normal operation, one will be running while the other is idle awaiting a new configuration). 5) Once the framebuilder software is ready, it will report a ready status to the FB Controller, which passes this information back to the operator.
6) On command to start acquisition (from the operator), the FB Controller will arm an Interrupt Servi Routine (ISR) to trap the next one second marker from the GPS module.
7) On receipt of this interrupt, the FB Controller will then send acquisition start commands to all DCU via the reflected memory header areas. The configured framebuilder code module will also arm its ISR to await data ready signals from the DCU.
8) At the next 1Hz GPS marker, the DCU will begin acquiring data. Every 1/16th of a second thereafter, data ready will be reported by the cycle counter in the DCU header, which generates an interrupt trapped by the framebuilder ISR.
9) The active framebuilder module reads the data from reflected memory and stores it into frame structures.
10) Once the framebuilder has formatted a one second data frame, it is transferred to the DAQS server via a dedicated fibre channel link.

DAQS Operator Control Panel

Start

Stop

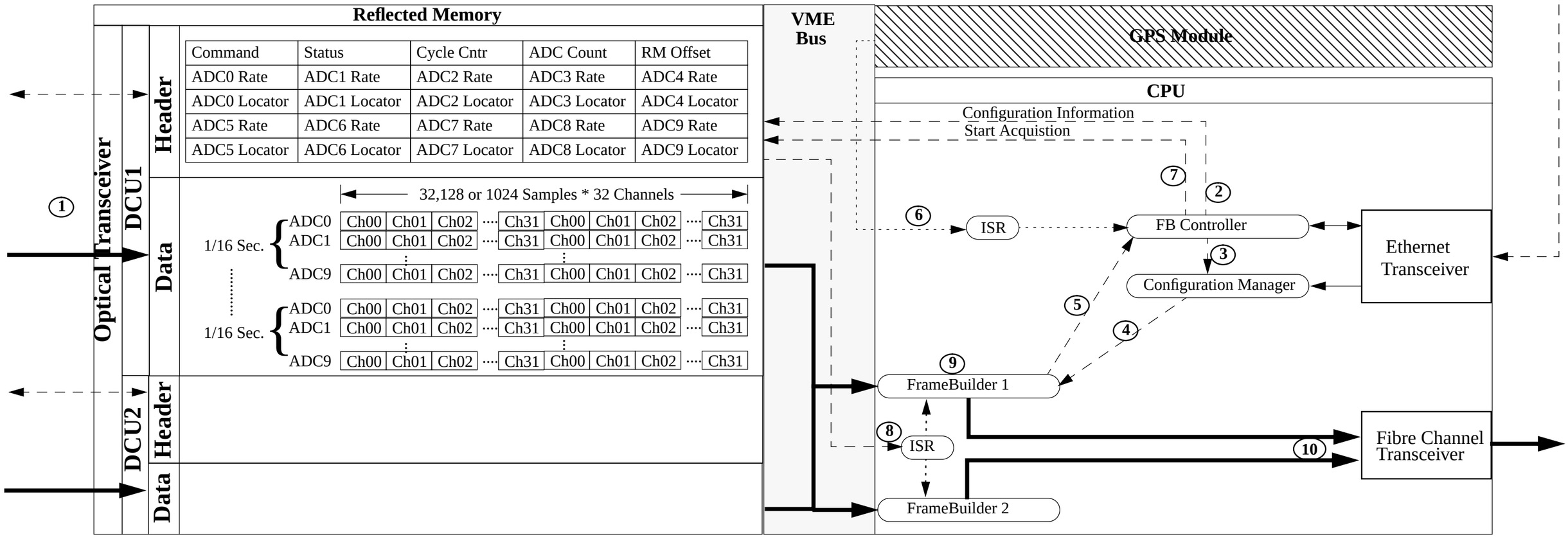
Configure

FB1 Status	Acquiring	**Run Number**	47
FB2 Status	Idle	**Frame Count**	1045678

FF File	/spa1/ff/Data0/H1-98_06_23_16_03_23
AF File	/spa1/af/Data0/H1-98_06_23_16_03_23.A
TF File	/spa1/tf/Data4/H1-98_06_23_16_03_00.T

Full Frame	/spa1/configuration/ff/format7
Analysis Frame	/spa1/configuration/af/format7
Trend Frame	/spa1/configuration/tf/format7

	Status	Cycle
DCU01	Acquiring	16730848
DCU02	Acquiring	16730848
DCU03	Acquiring	16730848
DCU04	Acquiring	16730848
DCU05	Acquiring	16730848
DCU06	Acquiring	16730848
DCU07	Acquiring	16730848
DCU08	Acquiring	16730848
DCU09	Acquiring	16730848
DCU10	Acquiring	16730848
DCU11	Acquiring	16730848



Appendix A
Figure 3
FrameBuilder