LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY

-LIGO-

CALIFORNIA INSTITUTE OF TECHNOLOGY

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Document Type	DCC Number **LIGO T060031-00-C**	February 3, 2006

LASTI Digital Control and Data System Design

R. Bork

Distribution of this draft:
This is an internal working note of the LIGO Laboratory

California Institute of Technology
LIGO Project – MS 18-33
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 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 Scope

The scope of this document is limited to the general design for the digital portion of the LASTI Control and Data System (CDS). Specifically excluded are the analog front end electronics and connection cabling required to interface with the LAST actuators and sensors. The latter is to be described in separate documentation.

2 Purpose

The purpose of this document is to give an overview of the CDS architecture to be implemented for Lasti controls over the period of the next year. Systems to be installed include controls for quad suspension, triple suspension and Internal Seismic Isolation (ISI) tests, Ponderomotive experiments, and an upgrade to the present HEPI and suspension control systems.

3 Document Overview

The following sections of this document give some background into the concepts that this design evolved from and an overview of the hardware and software used to implement these concepts. The hardware and software sections provide an overview of the components to be employed in all Lasti CDS subsystems. A final section provides a representative cost estimate.

4 Background

A meeting was held at Caltech last summer to discuss the design concepts for future Ligo control systems. This meeting was attended by representatives from all Ligo facilities. A number of ideas and/or recommendations came out of this meeting. The key items were:

- VME should not continue to be the primary format for front end controls. Several key issues were raised with VME. First is the age of the standard (over 20 years) and how much longer will it be supported. Versus newer technology, the bus speed is slow and the VME modules are typically much more expensive. Electrical and acoustic noise are also issues.
- Custom ADC and DAC modules are to be developed for future systems, particularly for Instrument Sensing and Control (ISC) systems. The driver here is the lack of commercially available modules which fully comply with LIGO noise and performance requirements.
- The CDS Input/Output (I/O) modules will reside in a separate enclosure remote from any CDS processors or processor buses. The proposal was that the I/O modules reside near the various Ligo sensors and actuators and use a serial fiber optic communication up link to data concentrators and/or CDS computers located outside of vacuum equipment areas. Initial discussions of the serial link centered on serial Front Panel Data Port (FPDP) or a custom implementation on fiber channel.
- Given the cost and reliability of the present Reflected Memory (RFM) network used to interconnect Ligo CDS realtime controls and data acquisition, perhaps this network should be replaced in the future with another network technology.

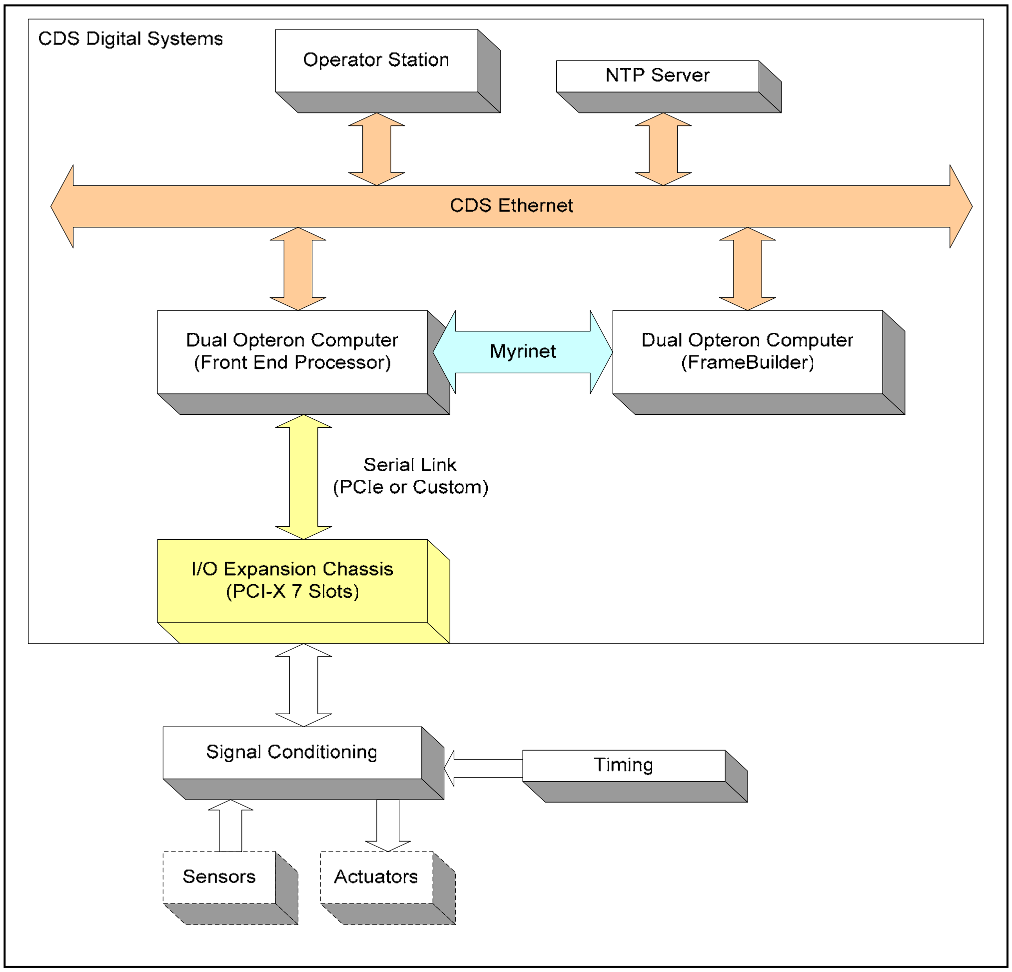
The CDS designs for Lasti, described in this document, attempt to move in the direction prescribed by the above concepts and other items raised in this meeting. However, given the installation time constraints for initial Lasti systems, the design proposed here is not fully in compliance with the concepts of advanced Ligo CDS designs. For example, custom ADC and DAC modules with serial links are at least a year away from realization of even a prototype. Therefore, the intent here is to incorporate as many of the desired new concepts into Lasti and test their capabilities and, at the same time, not preclude the replacement of some components along the way that are intended for advanced Ligo as they are developed.

5 Digital System Hardware

5.1 CDS Components

The following figure depicts a basic block diagram of the Lasti CDS design. Key components of this design include:

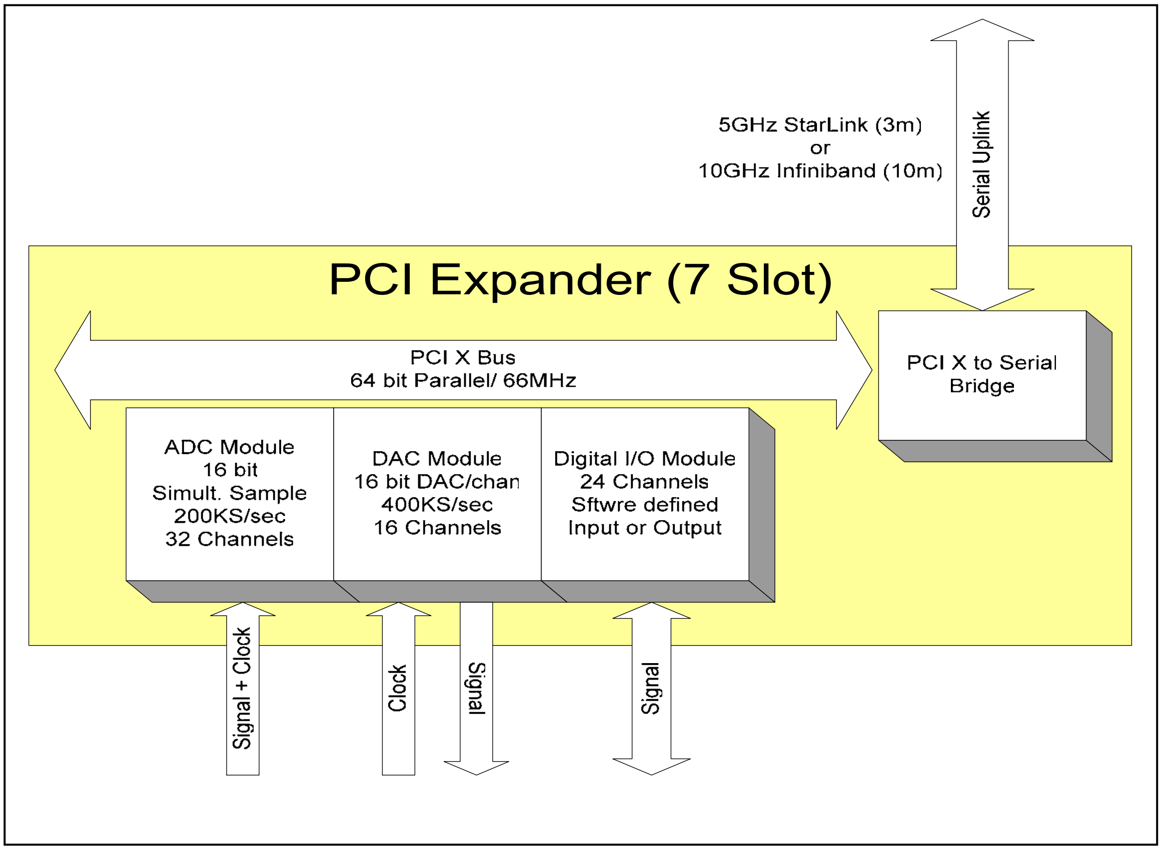
- PCI-X I/O module expansion chassis. This chassis houses the ADC, DAC and digital I/O modules necessary to interface to the analog signal conditioning modules. A serial link is used to communicate to the front end processors. This unit, and its serial link, is the primary unit to be replaced by a future Ligo custom design.
- Stand-alone, rack mount computers for front end control computations. This replaces the VME processors in present Ligo CDS implementations.
- A new commercial network (Myrinet) for realtime communications between front end processors and between front end processors and the data acquisition FrameBuilder.
- There are no longer separate computers for the Arbitrary Waveform Generator (AWG) or Test Point Manager (TPM). This functionality has been moved into the front end computers.



5.1.1 I/O Expander Chassis

The I/O expander chassis is a 4U rack mount unit with 7 PCI-X slots. A basic block diagram is shown in the following figure. Two such units are presently under test for installation at Lasti. One has a 64 bit, 66MHz PCI-X bus implementation and the other a 64bit, full 133MHz PCI-X bus implementation. The

66MHz version uses a 5Gbit/sec serial uplink to a PCI-X interface card in the front end computer. The 133MHz version employs a 10Gbit/sec Infiniband serial link to a PCI Express (PCIe) card in the front end computer.



PCI-X was chosen as the interim format for digital I/O modules for a number of reasons, among them:

- Current commercial standard with a future upgrade path (PCIe).
- Higher bus speeds, lower latency than VME.
- Lower cost than VME. Typically, I/O modules of the type required for Ligo controls, PCI modules run less than 50% of the cost of comparable VME modules.
- Relative ease of programming. All PCI modules must advertise certain useful information about themselves to the bus in a standard defined structure. This information is then readily available to user software in the implementation of hardware drivers. Unlike VME, there are no bus address jumper settings required.

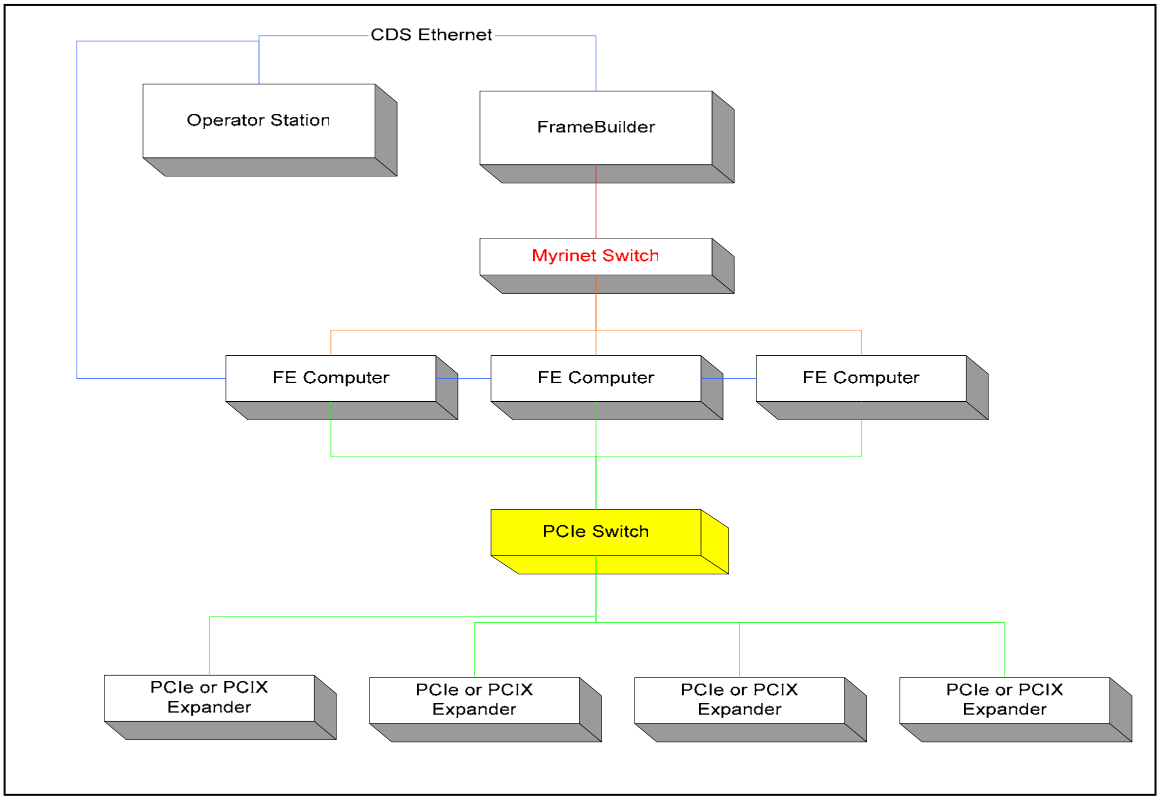
One emerging standard that this design is compatible with is PCIe. Starting to come to the market now are PCIe expander units, which use PCIe serial ‘lanes’ instead of the parallel PCI-X bus. Each PCIe lane provides for a 2.5GHz serial transmit line and a 2.5GHz serial receive line. At least one vendor is presently shipping a PCIe expander with 19 slots in a 4U enclosure. This uses a PCIe serial uplink to a PCIe card in the host computer. This standard also involves a PCIe switching mechanism, whereby multiple PCIe expansion crates can be linked to a switch and then the switch provide connections to multiple computers at the higher level. These serial data links between computers and I/O modules is at least in keeping with the advanced Ligo ideas, though at a higher data clocking rate.

The other plus side of PCIe is that the driver software at the computer end sees all I/O modules, be they PCI-X or PCIe, the same as if they were a local PCI card. The PCI standard registers are maintained in the new PCIe standard.

Unfortunately, while PCIe links and expansion units are starting to come to market, there are no ADC or DAC cards available to meet Lasti requirements in the near term. These are still 6 months to a year away. However, it may be possible to go to the new technology in the later stages of Lasti, if:

- The PCI-X implementation meets all requirements in early Lasti operations, particularly the ADC and DAC modules.
- More PCIe products become available in time.
- Fiber optic links become available for PCIe. At present, only cable links exist, which are usable in Lasti, but would preferably be fiber to remote the front end computers.

If all of this came into being, then the architecture may change somewhat to that shown below. One advantage of this architecture is that all FE computers would have direct access to all I/O modules. Also, if fiber PCIe comes about, the I/O crates could be further distanced from the FE computers.



5.1.1.1 ADC Modules

The initial ADC modules for Lasti are to be General Standards Corporation (GSC) model PMC66-16AI64SSA models. Several key features of these units are:

- 64 Single ended simultaneous sampled 16 bit input channels. In the Lasti application, the selectable differential processing mode is set to provide 32 differential channels.
- Software selectable ranges of +/-10V, +/-5V, +/-2.5V, 0/+5V or 0/+10V
- 200KS/sec per channel conversion rate

5.1.1.2 DAC Modules

The initial DAC modules for Lasti are to be GSC model PMC66-16AO16 models. Features of this module include:

- 16 high speed 3-wire balanced differertial output channels
- 16 bit resolution
- Separate D/A converter per channel
- Output update simultaneously
- Output ranges of +/-10V, +/-5V, +/-2.5V or +/-1.25V
- Data rates to 450KS/sec per channel

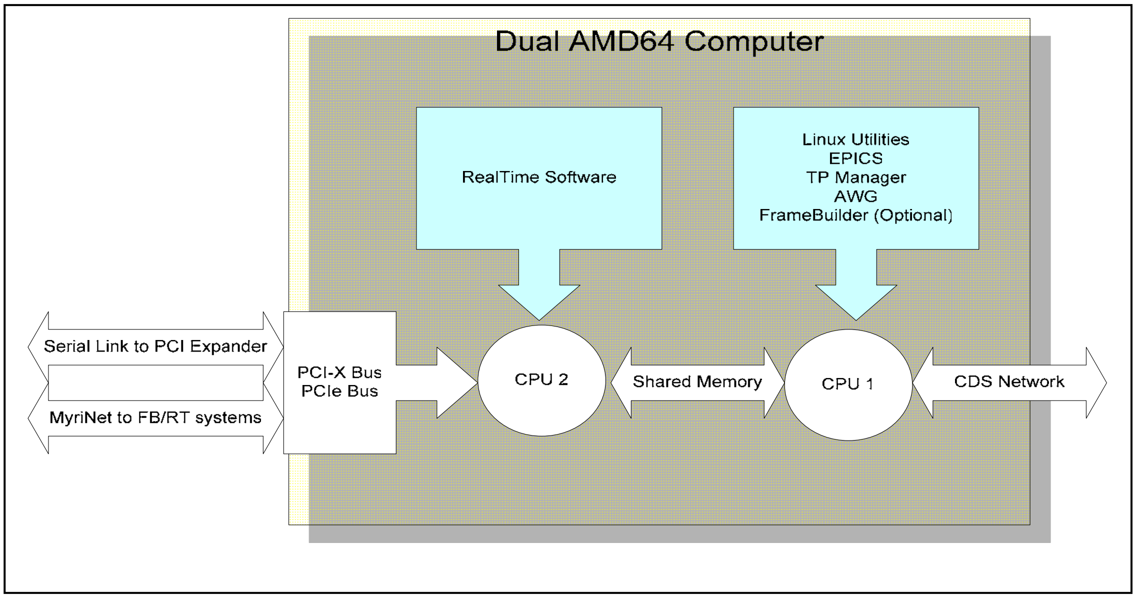
5.1.1.3 Digital I/O Modules

For digital I/O applications, such as setting or monitoring remote switch positions, an Access IO model PCI-DIO-24D module will be used. This module features 24 bits of TTL I/O, software configurable to set the I/O bits as inputs or outputs, in blocks of 8 bits.

5.1.2 Front End Computers

For front end computers, the option was to go with 64bit Operaton processor based units. The idea is to go with the latest technology presently readily available within a reasonable cost range.

For CDS front ends, dual Opteron computers will be used. Several models are presently on hand for development and testing. A basic block is shown in the following diagram.



The dual CPU provides for several features to be built into the front ends:

- Realtime control software can be set to run on one CPU without interference from additional standard operating system tasks running on the other CPU.
- Various CDS utilities, such as EPICS and GDS, are run on the non-realtime CPU. Some of the advantages of this are:
  - No separate computers required for these functions.
  - GDS TP/AWG no longer a shared resource among multiple front ends.
  - Reduced network traffic.
  - Faster response.

In some cases, as requirements dictate, the front end computers may have more than two processors. At present, up to 8 dual core AMD64 computers are available.

5.1.3 Realtime Networking

For the transmission of data between realtime front applications and between realtime front end applications and the DAQ Framebuiler, Myrinet replaces RFM networks in the Lasti design. Myrinet is a proprietary network, but can handle standard network protocols and be interconnected with standard Ethernet networks. Some advantages seen in this network over RFM networks:

- Cost: A Myrinet Network Interface Card (NIC) runs about ¼ the cost of a RFM NIC.
- Topology: Myrinet runs in a star fabric configuration as opposed to the serial loop configuration of RFM networks. This has reliability benefits and connection flexibility.
- Speed: Each Myrinet NIC has a full duplex 2.5GHz fiber connection. With RFM, a 2GHz serial link is shared by all processors on the network.
- Lower access time: Myrinet is implemented in full 133MHz PCI-X. In initial tests, the CPU time required to move data on and off the network have been significantly less.
- Direct memory access. Myrinet includes a feature which allows one computer to transmit data directly into the local memory of another computer on the network without CPU or software intervention at the receiving computer. This feature is highly useful in the movement of DAQ data from a front end computer to the FrameBuilder, as will be described later.

Like RFM, Myrinet is designed for low latency communications. As such, its latency numbers are similar to RFM.

5.1.4 FrameBuilder

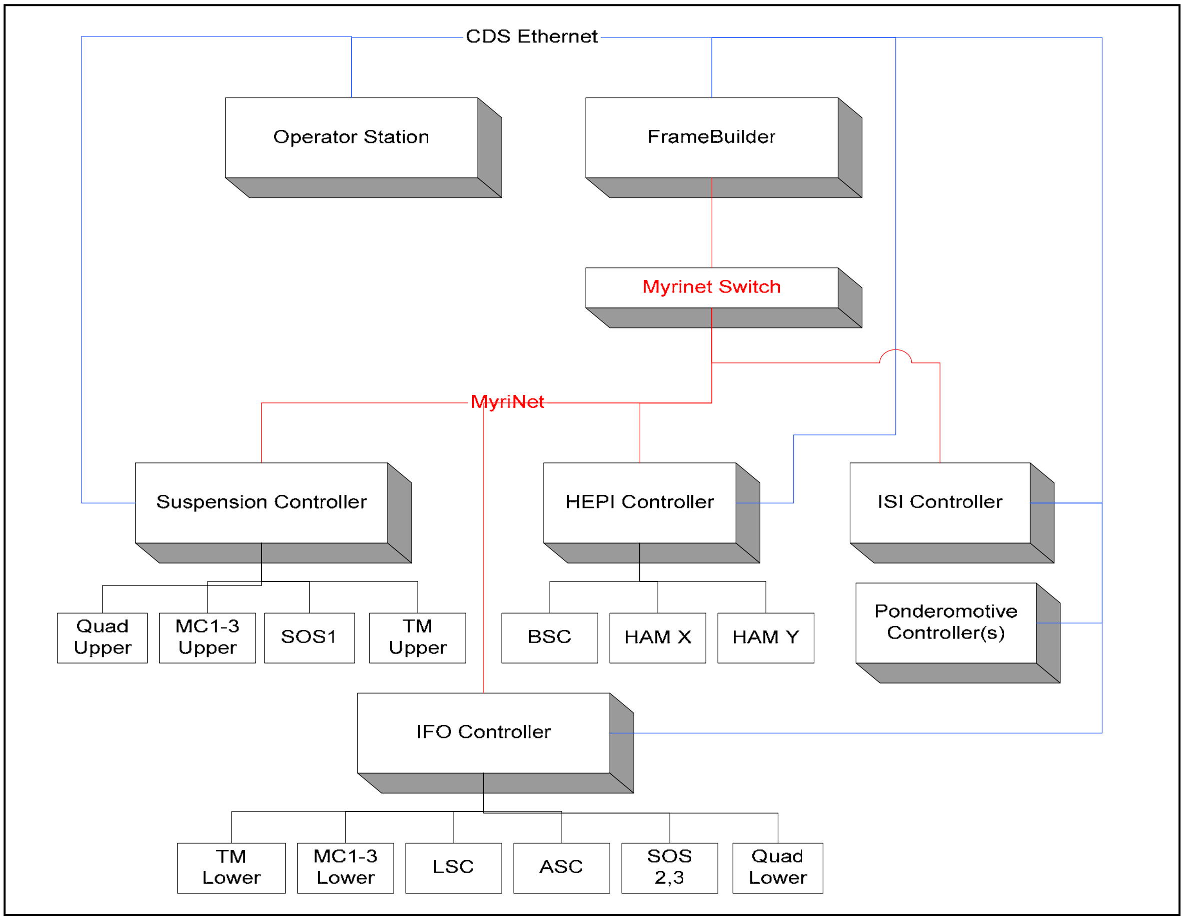
Data Acquisition and storage will be provided by a separate 2U rack mount, dual AMD64 computer. This computer will be equipped with a 73GB drive for storing CDS software, a 300GByte drive to store full data frames and two 300GB drives to save trend data. The latter two drives are set up in a RAID configuration to help prevent data loss on a drive failure. This computer has two SCSI drive bays empty, which may be used in the future to add more disk drives if requirements dictate.

5.2 Hardware Integration and Installation

Though not all of the details of the Lasti subsystems have been defined at this time, a concept of the Lasti subsystems and their integration is shown in the following figure. These subsystems are:

- Suspension Controller: On initial installation (April, 2006), this subsystem will provide all of the controls necessary for the Quad suspension testing. In a second phase (late 2006, early 2007), this unit will provide controls for the upper stages of the quad and triple suspensions, as well as small optics suspension 1. This second phase essentially provides for all suspension controls which do not require LSC/ASC inputs (see IFO controller below).
- HEPI subsystem controller: Present HEPI controls implemented in VME and DSpace will move to the new PCI based architecture. Three chambers will be controlled by this unit. In addition, there will be 12 Geophone devices monitored by HEPI controls. Installation is planned for April, 2006.
- Interferometer (IFO), or global, controller: This unit will provide for the LSC and ASC functions of the Lasti IFO. All processing involved with these global controls will run on this subsystem, including control of the lower quad/triple suspensions and SOS2 and SOS3. The reason suspension controls will be done here, as opposed to the LIGO model of separate computers for the LSC/ASC and suspension control functions, is primarily to avoid having to transmit the LSC/ASC data to the suspensions via the network. This single unit control will reduce network traffic and avoid the one control cycle delay seen in LIGO. In order to meet this goal, the computer used here will be a quad AMD64 computer. Installation is scheduled for late 2006 or early 2007.
- ISI Controller: This subsystem will provide all controls necessary for one chamber equipped with internal seismic isolation equipment. Installation is scheduled for early summer 2006.
- Ponderomotive subsystem: This subsystem is still in early design stages. It does have a few unique performance requirements. One is that at least a major portion of the control algorithms must run at 64KHz, as opposed to the more standard 16KHz or 2KHz encountered in the past. There may also be a requirement for a single 500KHz loop. Therefore, the number

of computers and number of processors per computer are still TBD. Installation of this subsystem is scheduled to begin during the summer of 2006.



6 Software

6.1 Front End Software

6.1.1 FE Realtime Software

The realtime control application runs in a realtime Linux kernel on the second processor of the front end computer. Once the realtime application is started, this second CPU is locked to only perform the realtime tasks. A basic block diagram of the realtime software is shown in the following figure.

All software for front ends is developed in C. The basic concept is to make as much of the realtime software as generic as possible for reuse in multiple front ends. Only the item labeled 'FE Application' in the diagram would be specific to a single controller. However, all four software components shown would be compiled into a single runtime object for performance reasons. (Note: In Lasti subsystems which require more compute power, more AMD64 processors would be added to run additional realtime threads).

6.1.1.1 CDS FE Controller

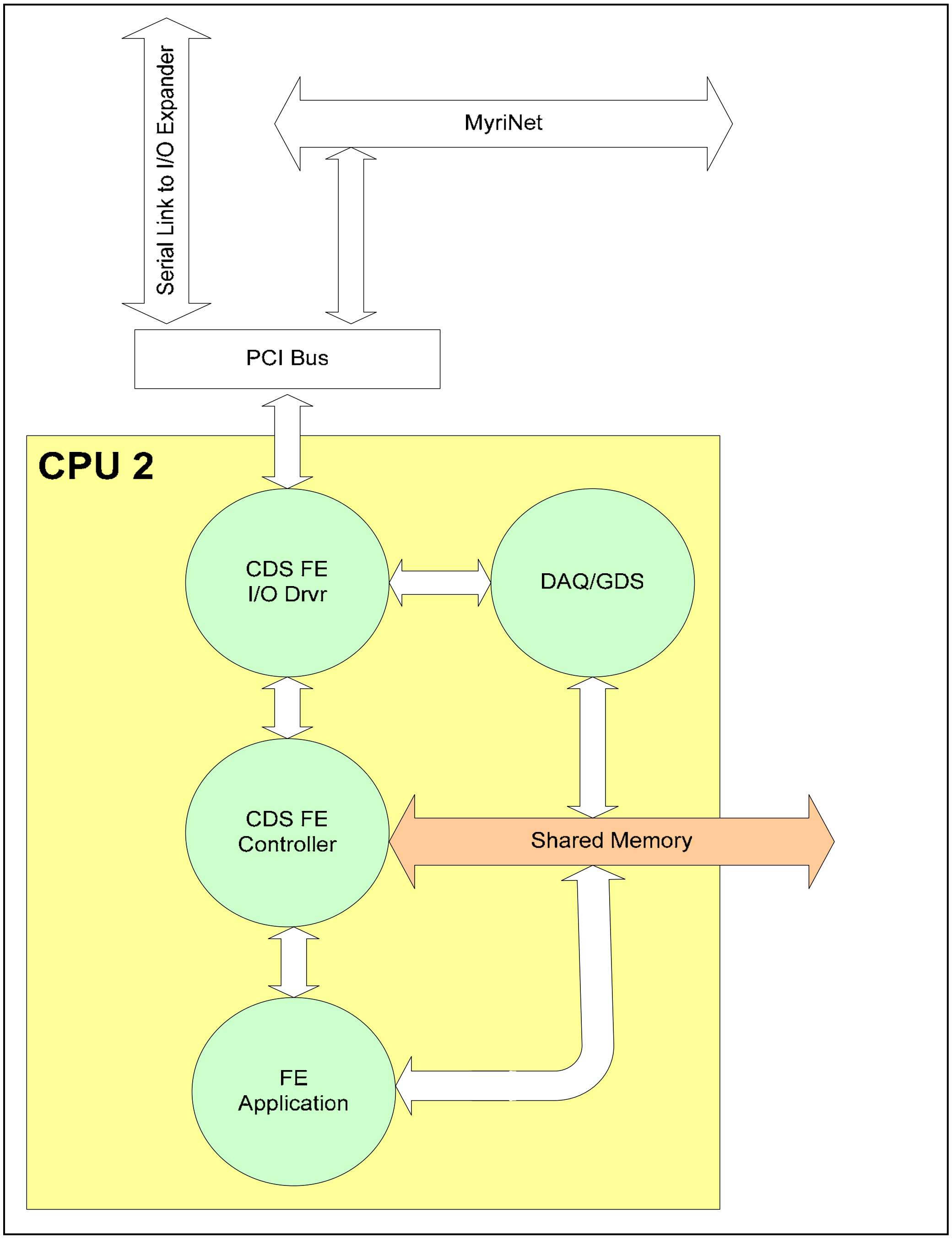
This software module controls the basic sequencing of front end tasks. Among these are:

- System initialization on startup. This includes verifying EPICS settings in shared memory are current, calling the DAQ/GDS software to get initial DAQ channel configuration information, and calling the I/O Driver to set up and start PCI interfaces.
- System synchronization. Once initialization is complete, the software will wait until it detects the 1pps timing pulse from channel 32 of its first ADC module. It will then begin the code sequence at the prescribed rate of up to 64KHz. The code will continue to check its timing against the 1pps signals during operation and is capable of resynchronizing itself. (***Note: This startup timing and timing verification scheme is temporary and may change when a new timing system is available***.)
- ADC and DAC overflow checking. This code provides the overflow counters to EPICS and also truncates any overflow settings being sent to the DAC channels.
- Provide ADC values to the front end application, as well as pointers to necessary EPICS information.
- Updates all CDS filter module information between the front end application and EPICS.

6.1.1.2 CDS FE I/O Driver Software

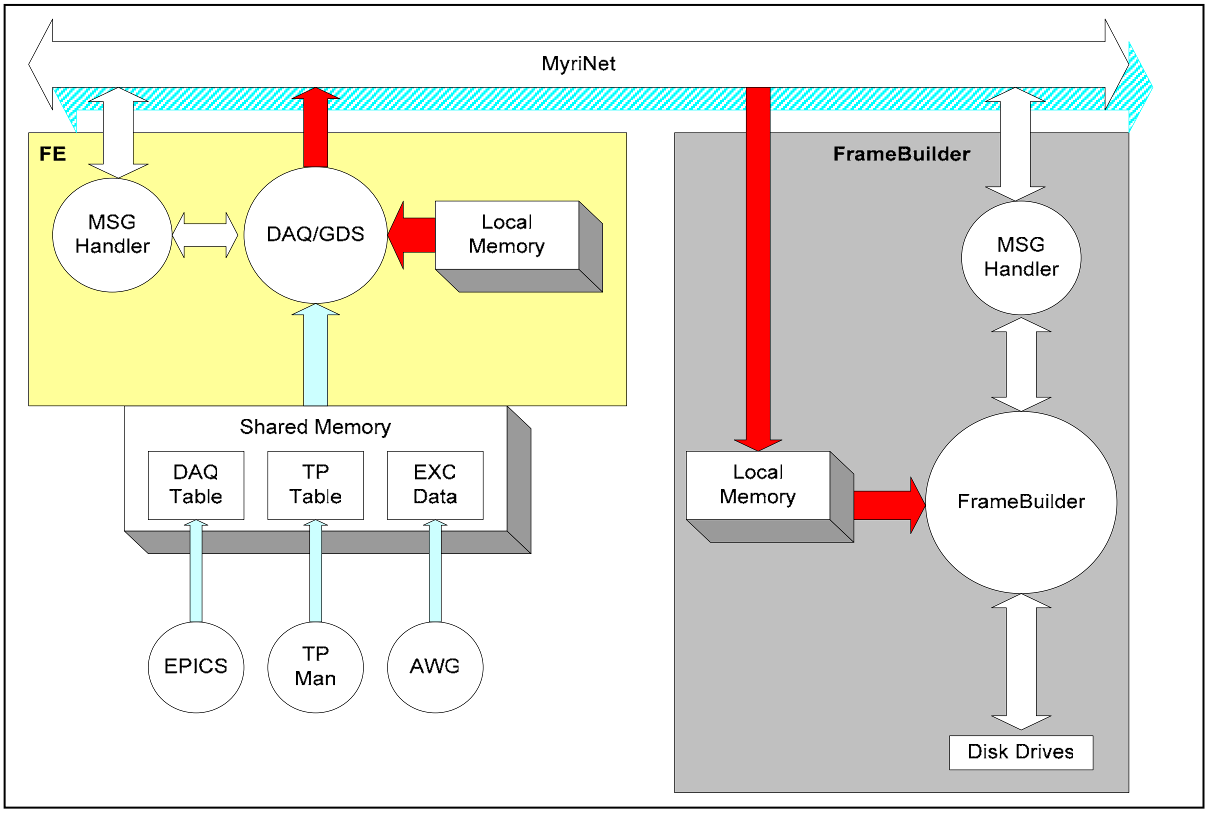
A single C code library contains all of the routines necessary for the front end to communicate with PCI I/O modules and the MyriNet fabric. Routines defined in this code are called by both the FE controller and DAQ/GDS software. Functionality included in this library are:

- PCI I/O initialization. This function will automatically search the PCI bus for the instances of modules that the software supports. Once supported modules are found, they are automatically mapped to user memory space and initialized to specified settings. Information about the modules found is kept in a defined structure for later use and diagnostics reporting.
- Data transmission routines between PCI I/O modules and the FE controller software. This includes Direct Memory Access (DMA) routines.
- Myrinet fabric communication routines to allow FE to FE communications and DAQ/GDS communications to the FrameBuilder.



6.1.1.3 FE DAQ/GDS Software

This software library provides all of the functions necessary for the front end to communicate its data to the CDS Framebuilder and to provide GDS excitation channels to the front end application software. This code operates in a similar fashion as the present Ligo daqLib and gdsLib front end software. It still sends defined DAQ channel data and testpoint (TP) data to the FrameBuilder and it still reads in GDS excitation (EXC) data and provides that data to the front end applications. However, there are some significant changes in the new software. An outline diagram of the new software configuration is shown below, with DAQ data flow shown in red.



There are two primary changes to the FE DAQ software in the new architecture:

- Since EPICS, TP Man and AWG now run locally on the first CPU, their channel configuration information is now conveyed to the FE code via a shared memory block. AWG also conveys EXC data via this shared memory.
- Data is transmitted to the FrameBuilder via the new Myrinet instead of an RFM network.

The first change is rather trivial in that the code simply points now to a local memory block instead of the RFM network memory block to get its configuration information and EXC data. The change to Myrinet is not so trivial, as it uses a more traditional network message passing scheme rather than the global memory scheme of the RFM network.

First, as previously mentioned, Myrinet provides the capability of a sending computer to transmit data directly into the local memory of a receiving computer. However, the receiving computer must define this memory and its location to the sending computer. So, on startup and whenever reconnection is required, the FE sends a message to the FrameBuilder requesting this memory information.

In the RFM network scheme, the DAQ/GDS code writes data on every front end clock cycle, attempting to balance the amount of data sent on each cycle to complete a 1/16th second data transmission within the 1024 or 128 cycles of a 16KHz or 2KHz system. This was done in an attempt to balance the loading of the shared serial RFM network and reduce the CPU load involved in these transmissions. With Myrinet, it is more efficient to send larger data blocks with less CPU load. Therefore, the new code transmits data to the FrameBuilder 16 times during a 1/16th second period or at 256Hz.

In Ligo controls, the FE sends DAQ data and TP data to different RFM locations and these are therefore received at different memory locations by the FrameBuilder. The AWG transmits its own data to the network in yet another location. In the new scheme, the DAQ routines of the FE send all of this data together to the FrameBuilder, DAQ data first, followed by TP data and finally EXC data. AWG no longer sends out data on the network. To handle this, a 4MByte block of memory is set up in the FrameBuilder for each FE to receive data. This represents one second of data or a maximum of 4MByte/sec from each FE processor. This maximum is for combined DAQ, TP and EXC data. Once the FE completes sending a 1/16th second block of data, a message is sent to the FrameBuilder to inform it that the transmission is complete. Contained in this message is:

- Timing information for the data block.
- The size of the data block and a CRC checksum.
- A list of the TP and EXC channels transmitted along with the DAQ data and the size and locations of those data within the 1/16th data block.

Once the FrameBuilder receives this message, it performs its data integrity checks and puts the DAQ data into the data frame. TP and EXC data are relayed to the CDS network via the Network Data Server (NDS) if they are being requested by a CDS operator station or as part of the broadcast frame to GDS DMT monitors.

6.1.1.4 FE Application Software

This software is unique to each Lasti CDS subsystem. However, this software will use all of the standard code modules previously developed for LIGO controls. The primary element of this is the CDS standard filter module software.

One desire in future code development is to provide a graphical modeling interface, such as SimuLink, to generate the application software. One of the drivers to develop the modular software, described in previous subsections, was that this graphical code generator would only need to develop the specific application. This application would not need to know about how to handle PCI I/O, interface with networks, etc. That would all be handled by the CDS infrastructure software.

6.1.2 FE Non-realtime Software

For the new architecture, the GDS TP manager and AWG software have been modified to run on the FE computer on the first CPU. In the past, these software modules have run on separate computers and communicated with the FE and FrameBuilder via the RFM network. In this case, these programs communicate their information via the shared memory block within the FE computer

Key changes required and implemented in the GDS software to support this new scheme include:.

- Porting the GDS software to run on the Linux operating system.
- Move the GDS software data read/writes pointers to internal, shared memory instead of the RFM network memory.
- Modify the GDS .param file. In Ligo, the GDS signal list, .param file, contained a list of all GDS TP/EXC channels within an interferometer. In the new instance, the .param file contains only those signals which apply to the specific FE.

EPICS will also run on the first CPU of the FE computer. The only significant change here is to once again set its pointers to read/write to/from local memory instead of the RFM network memory.

7 Cost Estimate

This section provides a cost estimate for the key components of the Lasti CDS subsystems. Individual subsystem documentation will later provide detailed cost estimates for those specific subsystems.

Costs for the components necessary to provide the Lasti CDS infrastructure are listed in the following table.

CDS Infrastructure Cost Estimate	
Component	**Cost**
Framebuilder Computer	$4000
Framebuilder Disk Drives	$1800
Framebuiler Myrinet Module	$550
Myrinet Switch	$6000
Fiber Optics	$500
Total Cost	$12850

The following table lists the components and costs for a typical subsystem (initial quad suspension controls). Cost numbers for a similar VME implementation are also given for comparison purposes. Note, however, that the comparison is not exact ie the VME costs list a much lower end CPU (Pentium 4 vs dual AMD64). Also, the VME version requires additional support in the form of separate computers and VME crates for the GDS AWG and EPICS function ($12000). These functions are built into the PCI system.

Cost Comparison for a basic 32 I/O channel Front End system				
VME Component	**Cost**		**PCI Component**	**Cost**
Single CPU board	$3500		Dual AMD64 Computer	$3500
VME Card Cage	$2000		PCIX I/O Expander	$3000
ADC Module (4x8Ch)	$16000		ADC Module (32 Ch)	$4000
DAC Module (4x8Ch)	$16000		DAC Module (2x16Ch)	$7000
RFM Network Module	$2200		MyriNet Module	$550
Digital I/O (32 Ch)	$2200		Digital I/O (24 Ch.)	$175
Total	$41,900			$18,225