

"5000 Electronics and Software" revised November 6, 1995, received from Benoit Mours/VIRGO (mours@frcpn11.in2p3.fr).

Distributed by H. Yamamoto 2/23/96.

B. Barish
K. Blackburn
R. Bork
M. Coles
D. Coyne
A. Lazzarini
F. Raab
G. Sanders
D. H. Shoemaker
R. Vogt
R. Weiss
S. Whitcomb
H. Yamamoto
M. Zucker
Chronological File
Document Control Center

L160 - ~~9950024~~ - 00 - D

T 950153 - 00 - D

5000 ELECTRONICS AND SOFTWARE

(Rev. November 6, 1995)

INTRODUCTION.....	3
5000.1. THE ONLINE ARCHITECTURE.....	3
5000.1.1. Specifications.....	4
5000.1.2. The architecture.....	4
5000.1.3. The hardware options.....	5
5000.1.4. The software options.....	7
5000.1.5. Security.....	9
5000.1.6. The timing system.....	10
5000.1.7. The Slow monitoring network organization.....	11
5000.2. THE CONTROL OF THE INTERFEROMETER.....	13
5000.2.1. The suspension control.....	14
5000.2.2. Longitudinal signal detection.....	15
5000.2.3. Global Control.....	16
5000.2.4. Non-linear Global Control.....	17
5000.2.5. Laser control.....	17
5000.2.6. Input bench control.....	17
5000.2.7. Detection bench control.....	17
5000.3. THE CONTROL AND MONITORING OF THE INTERFEROMETER ENVIRONMENT.....	18
5000.3.1. The Tube Control Station.....	18
5000.3.2. The Tower Control Station.....	18
5000.3.3. Imaging system.....	19
5000.3.4. Calibrators control.....	19
5000.4. OFFLINE.....	19
5000.5. DETECTOR COMMISSIONING AND CALIBRATION.....	20
5000.5.1. Data acquisition commissioning.....	20
5000.5.2. Controls commissioning.....	20
5000.5.3. Calibration.....	20
5100. GLOBAL CONTROLS (ASSOCIATED SOFTWARE).....	1
5100.1. THE SUPERVISOR.....	1
5100.2. THE COMMUNICATION TOOL: CM.....	2
5100.3. THE ERRORLOGGER.....	3
5100.4. THE ONLINE DATABASE.....	3
5200. STANDARDIZED COMPONENTS.....	1
5200.1. STANDARDIZATION AND SOFTWARE RULES.....	1
5200.1.1. Language.....	1
5200.1.2. Coding rules.....	1
5200.1.3. File organization.....	1
5200.1.4. The list of Slow Monitoring Stations.....	3
5200.1.5. Data format on tape.....	3
5200.1.6. Information distribution.....	4
5200.2. SUMMARY OF THE CRATE CONTENT.....	4
5200.2.1. Control Building.....	4
5200.2.2. Main building.....	4
5200.2.3. Mode cleaner.....	5
5200.2.4. End building.....	5
5200.3. WORKSTATIONS IMPLEMENTATION.....	6
5200.3.1. Control Room & Computer Room.....	6
5200.3.2. Main Building.....	7
5200.3.3. Mode Cleaner or End Mirror Building.....	7
5200.4. THE VIDEO SYSTEM.....	8
5200.4.1. List of cameras.....	8

5200.4.2. Digital processing of the signals.....	9
5200.4.3. Monitor installation in the Control room.....	9
5200.4.4. Cabling.....	10
5200.5. STANDARDIZATION: HARDWARE.....	12
5300. NETWORKS.....	1
5400. DATA ACQUISITION AND ONLINE PROCESSING.....	1
5400.1. THE LOCAL READOUT SYSTEM.....	1
5400.2. THE FRAME BUILDER.....	2
5400.3. THE FRAME STRUCTURE.....	4
5400.4. THE DATA DISPLAY.....	5
5400.5. ONLINE PROCESSING.....	5
5400.5.1. Global Control Survey.....	6
5400.5.2. Data monitoring and quality.....	6
5400.5.3. Data Filtering or frame trigger.....	6
5500. DATA ARCHIVING STORAGE AND DISTRIBUTION.....	1
5600. SIMULATION: SIESTA.....	1

Introduction

This chapter describes all the needed activities starting from the output of the local electronic up to the data ready to be analysed for Gravitational Waves search. It covers four fields:

- The online activity. It concerns the controls of the interferometer, the acquisition, online processing, and monitoring of the data produced by the various detectors or control process.
- The detector simulation activity. It concerns the generation of gravitational events like coalescent binaries, pulsars, supernovae and the simulation of the response of the interferometer to such signals and to all possible noises.
- The offline activity. It concerns the basic tools for the refined analysis of the large amount of collected data.
- The detector commissioning and calibration activity. It concerns the strategy to check parts or all of the system and the calculation of all calibration parameters.

A coherent view of all these activities will optimize the design, the commissioning, the running and the understanding of the first results produced by the interferometer. In addition to the discussion of these four fields, the standardization and rules and the hardware implementation will be also presented.

Most of the solutions presented here have already been described in the VIRGO note PJT93-16 and in its updated version from September 94.

5000.1. The online Architecture

The various active parts of the detector which produce digitized information or 'data' are represented on figure 5000.1. Most of the signals produced by the different sensors are processed by a layer of local controls which compute and apply local corrections to keep the controlled elements (laser, mirror, vacuum pump...) within a given set of tolerances. These controls produce data like status information or feedback values. Several local controls may be supervised by a higher level control, the Global Control in charge of the locking or the alignment .

The data generated by these detectors and controls (the 'raw data') are collected by a local readout process, concentrated and structured by a Frame Builder and written to tape (Data archiving). They are processed ('reconstructed') to convert ADC counts and feedback signals to an h value, and 'filtered' to reduce their amount to a level manageable by the offline analysis. The data quality is permanently monitored by surveying the noise level and a known signal produced by a calibration device stimulating permanently the interferometer as a gravitational wave would do it. Finally, the 'filtered' data are sent to a storage and distribution system which is the experiment front end for the offline analysis. The system provides also tools for histogramming and data editing (including monitoring data).

All these controls are coordinated by a Supervisor and synchronized with a central Timing system.

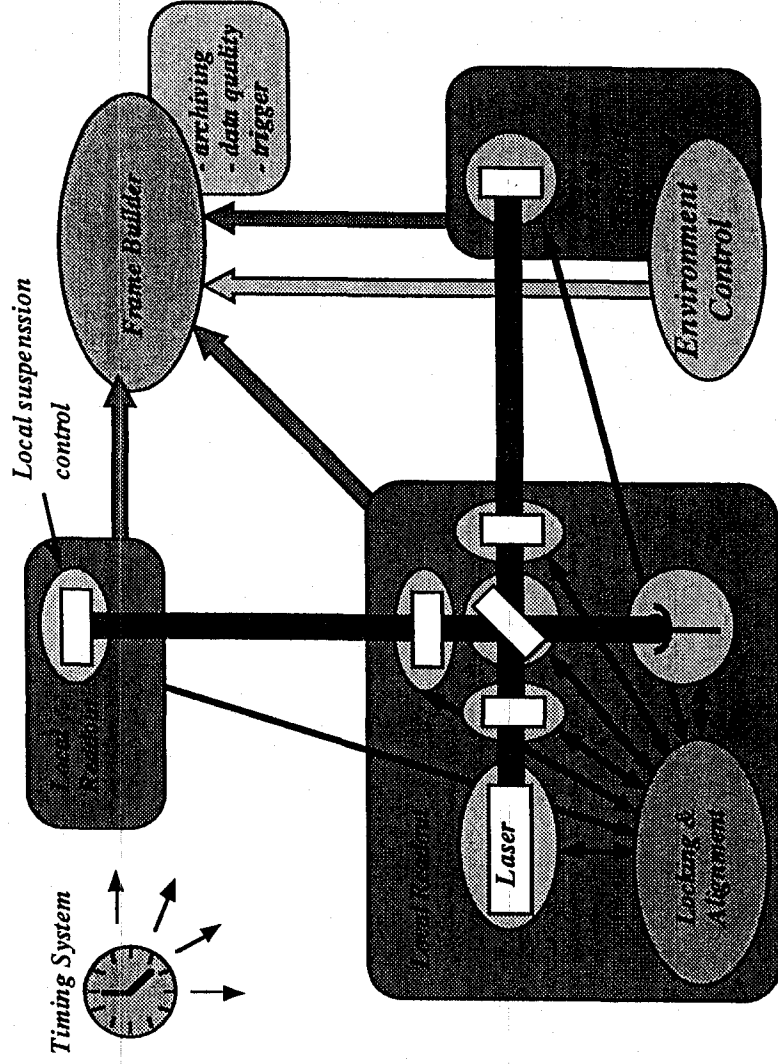


Figure 5000.1 The online architecture

5000.1.1. Specifications

The main specifications for the online system are:

- a real time control system with fast decision taking processes and correlation capability (up to 20 kHz) all over the site,
- a data acquisition system with a maximum sampling frequency of 20 kHz and a data rate up to 10 MBytes/s,
- a dead time between the user interface and the real time process lower than one second,
- a system easy to reconfigure and to use on small test bench,
- test capability at all levels.

5000.1.2. The architecture

The interferometer is kept at its working point by a set of controls which process data provided by sensors and adjust accordingly its main components: the laser, the two mirrors of each Fabry Perot cavity, the end mirror of the mode cleaner, the recycling mirror, the beam splitter, the injection and detection benches.

The architecture of the control and read out systems has to account for the large distances which may separate two components. To preserve flexibility in the interferometer setting up, the control systems are organized as much as possible into independent units in charge of the adjustment of a well defined component. They are run locally in standalone mode and accessed remotely by some high level control process. Similarly, the readout systems are implemented into independent units in charge of the concentration of the data produced near the components located in a same building: end mirror west and north, mode cleaner, and main buildings. A central data acquisition system, located in the control room, collects and assembles these data before writing them to tape.

The knowledge of the precise timing of the various measurements and actions performed around the interferometer is one of the key points of its operation as a

gravitational wave detector. This is implemented with a central timing system, located in the control room, and set up to distribute a well defined clocking sequence all over the site.

5000.1.3. The hardware options

To achieve a few kHz bandwidth on the interferometer sensitivity, one has to design its various servo loops with a much higher bandwidth in the range of 10 kHz. As a consequence, one has to implement the system with 'fast' sensors, processors and actuators connected with 'fast' links. To execute the various feedback loops within a constant time the system has to be built with conflict free accesses.

Sensors, processors, and their actuators sampled at those high rates are thus implemented within the commercially available VME standard. High transfer rate digital servo loops are implemented with a dedicated bus. Conflicting bus accesses are avoided by housing only exclusive controls in the same crate. To keep proper track of all the measurements only digital information can be exchanged. Higher level control processes transfer their information from building to building via a Digital Optical Link (DOL). Short distance transfers between different controls are performed through mirrored memories connected on a local vertical bus or using a DOL.

The environment status is generally measured with sensors sampled at a low rate. They are thus implemented in VME or G64, depending on their availability and cost. They are read out by slow monitoring systems and the data are exchanged using a dedicated ETHERNET network (the slow monitoring network) extending all over the site.

The data sampled at high rates are collected by the local fast readout systems. Then they are transferred to the main control room by a digital optical link.

All the processors are networked via ETHERNET or FDDI and accessible for control and file exchange by all workstations. Figure 5000.2 shows the networking which has to be implemented between the main building, an end mirror building and the control room. The basic rule is that hardware links are used for any real time connections while computer networks are used for state control and for slow monitoring.

The use of standard hardware is mandatory since it allows easy reconfiguration and long term maintenance.

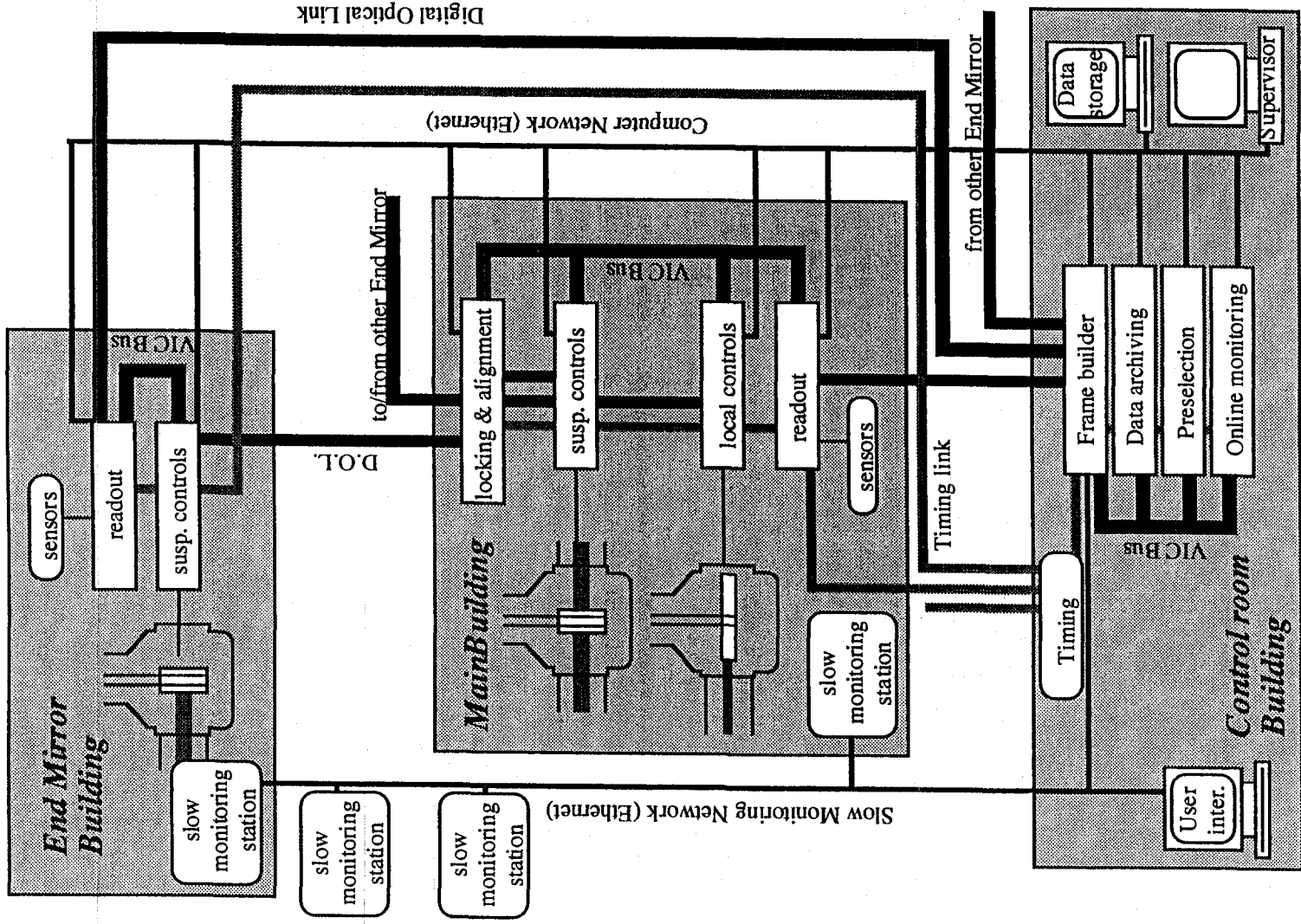


Figure 5000.2 The network architecture implemented between one end mirror building, the main building and the control room.

5000.1.4. The software options

The steady operation of the interferometer is implemented in the context of distributed processing synchronized by a 'central' timing system. The various control processes are designed as standalone tasks getting data from a dedicated local sensor and/or from another control process. The different readout tasks are organized to operate under the mastership of a central DAQ task.

Each control process (fig. 5000.3) is a server built in the framework of the client-server model. It is a real time program usually running on a VME CPU. It has direct access to the hardware required by its functionality (typically ADC's, servoloop,...). It should be able to run and to deal with its possible error conditions, minimizing its access to the computer network. For fast control processes, only data for test, configuration parameters and status information should be exchanged using the computer network.

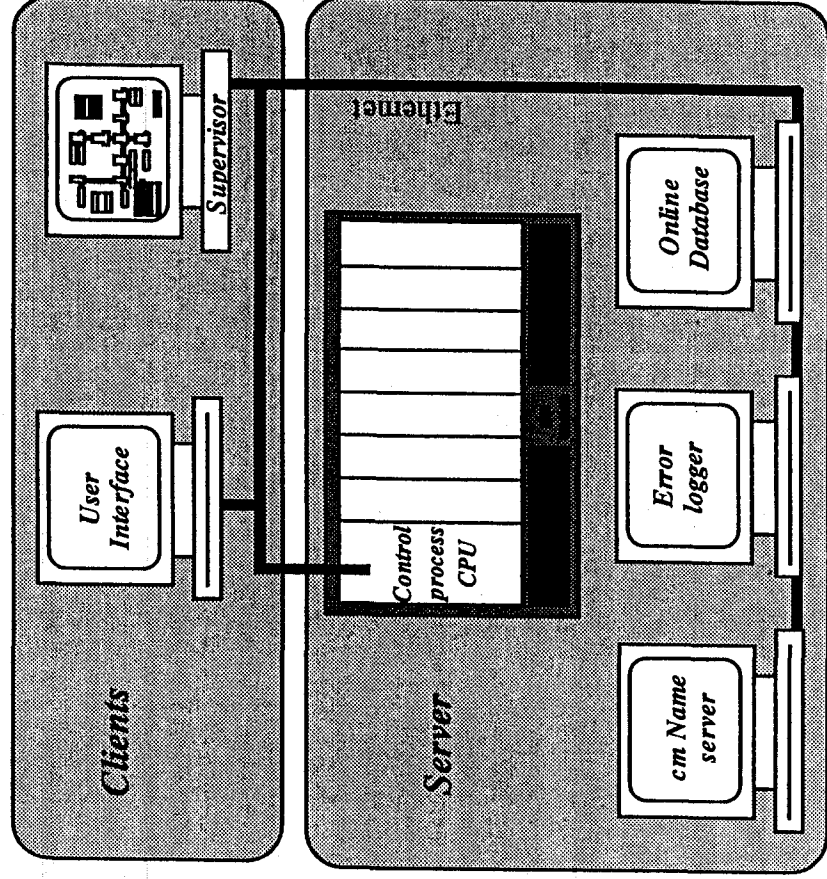


Figure 5000.3 client/server process organization

The user interface (a client of the control process) configures the server and/or monitor and displays the data provided by the server. Several user interface clients can be simultaneously used to monitor its information, but only one client at a time (the master client) has the privilege to configure a server. Possible conflicts are locally solved using standard rules and information provided by the online database. The user's interfaces are run on workstations and they do not need to be real time.

The Cm library is used for ALL the communications between processes on the computer network. This package provides platform independent communications tools and access to the processes using logical names. The correspondence between logical names and physical addresses is provided by the Cm name server. This server is needed for all applications using cm and therefore it should always be running. There is a unique Cm name server for the experiment.

To provide an easy control of all the process by a single user interface (The supervisor) all the running servers are able to respond to a standard set of messages. These messages allow the supervisor to control the basic server operations and some monitoring facility. The list of the standard messages is given in table 5000.1. Figure 5000.4 and 5000.5 give additional information about the exchanges messages. In addition to this standard list, messages specific to a type of server are available. They should always start with the two key letters of the process type.

Request		Expected return information	
Message type	Data send	Message type	Data send
SuGetStatus	none	SuServerStatus	status (see note 1)
SuBecomeStatusUser	none	SuServerStatus	status (see note 1)
SuNoMoreStatusInfo	none	SuServerStatus	status (see note 1)
SuLoadConfigFromDisk	file name	SuServerStatus	status (see note 1)
SuLoadConfigFromDB	file name	SuServerStatus	status (see note 1)
SuPutConfig	none	SuServerStatus	status (see note 1)
SuGetConfig	none	SuSendConfig	config file
SuSaveConfigOnDisk	file name	SuSaveConfigStatus	success or failed
SuLoadConfig	none	SuServerStatus	status (see note 1)
SuCheckConfig	none	SuCheckStatus	success or failed
SuRunServer	none	SuServerStatus	status (see note 1)
SuGoToLevel	an integer	SuServerStatus	status (see note 1)
SuStopServer	none	SuServerStatus	status (see note 1)
SuKillServer	none	SuServerStatus	status (see note 1)
SuWhoIsMaster	none	SuMasterIs	status (see note 1)
SuBecomeMaster	none	SuMasterIs	master name
SuDelegateMaster	client name	SuMasterIs	master name
		SuMasterIs	master name

note 1: The text gives the server status should start with one of the following words:

- 8 character for the current status: -stable-, runLv1nn (where nn is a number), configId, configAv, noConfig (see figure 5000.4):
- 6 characters for the time with the format: hhmmss
- 4 characters for the config checksum (computed with a short)

Then additional text could describe the changes like: waiting for DataBase, trying to load,...

Table 5000.1

The stable state is reached when the server and its hardware are operated in a mode which allows stable data taking. Running is for the cases where the system tries to reach the stable case like pumping down, trying to align, to lock,... Several running levels could be specified.

All the parameters needed to run a server are stored in configuration files which could be stored locally, by the server, or by the Online Data Base. These configuration files are just an ASCII text to allow easy understanding and debugging. In fact one of the main tasks of the user interface is to be an intelligent configuration editor. The Online Data Base stores also all the information about privilege access to the server. This is done by using the machine name and the user account as selection parameters.