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

Document Type	**LIGO-T950012-00 -**	**C**	3/28/95

Pulse Damping Test Control System Description

R. Bork

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/

TABLE OF CONTENTS

1 OVERVIEW

This document describes the hardware and software developed for the 'Smart Lock Acquisition' or 'Pulse Damping' test performed on the LIGO 40M prototype. These tests were conducted to study and model lock aquisition and define methods to assist the analog servo system in lock acquisition i.e. speed up the lock acquisition time.

2 SYSTEM ARCHITECTURE

2.1. Hardware

This system was designed with the constraint of making use of existing LIGO hardware. If this system is adopted for future use, the hardware system should be re-designed to make use of more appropriate hardware.

The system consists of a single VME crate w/modules and a single NIM module. The basic layout is shown in the following figure. Detailed schematics and layouts appear in Appendix A.

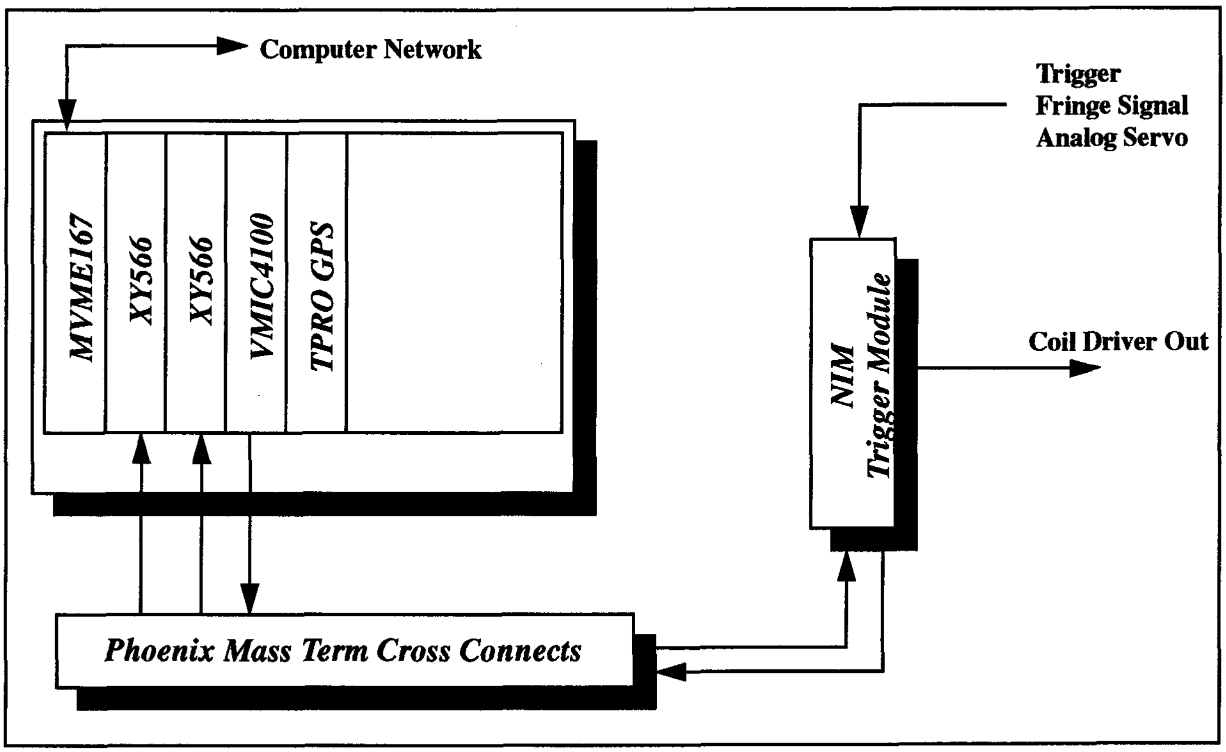


Figure 1: System Hardware Layout

MVME167: System real-time processor board with Motorola 68040 processor.

XY566: ADC modules, 12 bit resolution at 100KHz sample rate. Boards are setup to sample only one channel (0) at 100KHz in the manner of a waveform digitizer. ADC1 is connected to capture the fringe signal and ADC2 to capture the coil driver signal (analog servo signal).

VMIC4100: DAC module, 12 bit resolution at 100KHz. This module provides:

-> Trigger threshold adjust: Level to NIM trigger module comparator.

-> Control Pulse: voltage output to coil driver via NIM trigger module.

-> Blanking Output: Switches relay in NIM trigger module to direct either the control pulse output or the analog servo output to the coil driver.

TPRO GPS: This is a Global Positioning System (GPS) slave timing module, used in this case to provide precision time interval measurements.

This system requires the following base addresses to be set for proper operation:

Xycom 566: 0xffff9000 with data registers at 0xf0a00000
Xycom 566: 0xffff9040 with data registers at 0xf0a20000
VMIC 4100: 0xffff4100
TPRO Timing: 0xe0f00700
(NOTE: Due to an EPICS driver problem, a XY212 is also in the crate to avoid bus errors. This will be corrected at a later date).

2.2. Software

This code consists of three parts in Base Directory: /home/CDS/a/A3.11.4/VacuumApp. This is not the right place for this code to reside in the long term, but was setup and available when this project began in haste. If this code continues to be used, an EPICS application directory structure should be built for the 40M prototype area and this code moved into it.

The three software components are:

EPICS screens
EPICS database
State Notation Language (SNL) based real-time code
These software components are discussed in detail in the Software Detailed Description section of this document.

3 SYSTEM OPERATION

3.1. Start-up

3.1.1. Hardware Power up

The VME crate and NIM module reside in a rack near the vertex of the 40M prototype. This system is not normally cabled into the prototype except during testing. Therefore, cables should be connected in accordance with the 40M log book entries. The coil driver output should be left disconnected until all equipment has been powered up as the power up states of the VME modules are not controllable.

The VME and NIM crates should be powered on. The VME processor will automatically boot, load the software, and begin operation.

3.1.2. Operator Console

At present, the operator console can be any LIGO Sun workstation. The workstation actually performing the operator console functions, i.e. communicating with the VME processor, will be 'leopardess'. The workstation actually used by the operator will be simply an X term to that workstation. This is because, at present, this is one of the few workstations configured to run EPICS and also the only machine which can broadcast on the 40M subnet.

The first step to start operation from a workstation is to bring up the EPICS display manager and the pulse test operator displays. The sequence is:

1) X windows must be running on the workstation.
2) From a window, add 'leopardess' as an X host (xhost +).
3) Rlogin to 'leopardess'.
4) cd to /home/CDS/a/A3.11.4/VacuumApp/screens
5) setenv DISPLAY your_computer_name:0
6) execute 'medm -display your_computer_name:0 -cleanup&
The Motif EPICS Display Manager (medm) window should now come up on the screen.
7) From the medm 'FILE' menu, select 'OPEN'; a file browser will appear.
8) Scroll down in the browser and select 'PT_MAIN.adl'; the operator screen will now appear in a new window in EDIT mode.
9) Select 'EXECUTE' from the medm main window. The screen will become 'live' and should fill in the appropriate real-time data.

3.2. Display Description and Operation

3.2.1. PT_MAIN

This display is the primary one for operation of the system. It allows for:

- System Triggering
- Mode Selection
- System Settings

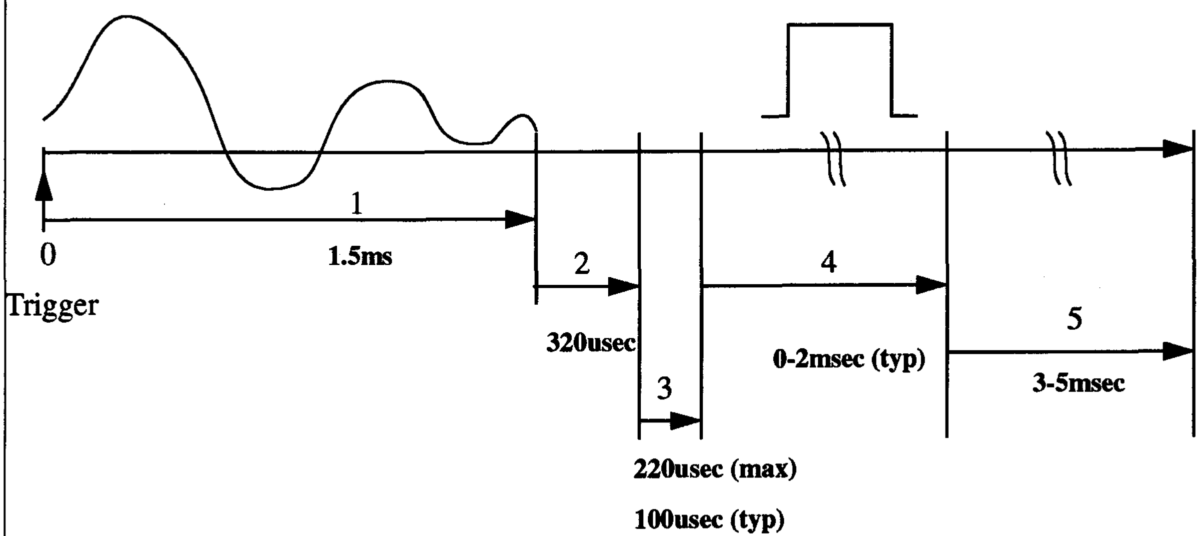
- Bringing up additional displays

3.2.1.1 Modes of Operation

The system presently has nine operating modes, selectable from the main display. These modes are primarily different combinations of two basic building blocks:

- Monitor: Acquire the fringe waveform and calculate velocities.
- Control: Issue a pulse to the coil driver in response to measurements made from the fringe signal.

A typical timing chart is shown in the following figure.



T0: Trigger ADC module from trigger module

1) Record fringe in ADC @100Ksamples/sec for 150 samples

2) Read ADC into uprocessor software variable (~2.1usec/sample)

3) Find zeros and calculate pulse width (variable to max 220usec if all 150 points must be scanned to find zero crossings)

4) Send control pulse to DAC for up to 6 msec (typically not greater than 2 to 3msec)

5) Release servo lockout relay (mechanical limitations of relay)

Figure 2: Typical Timing Chart

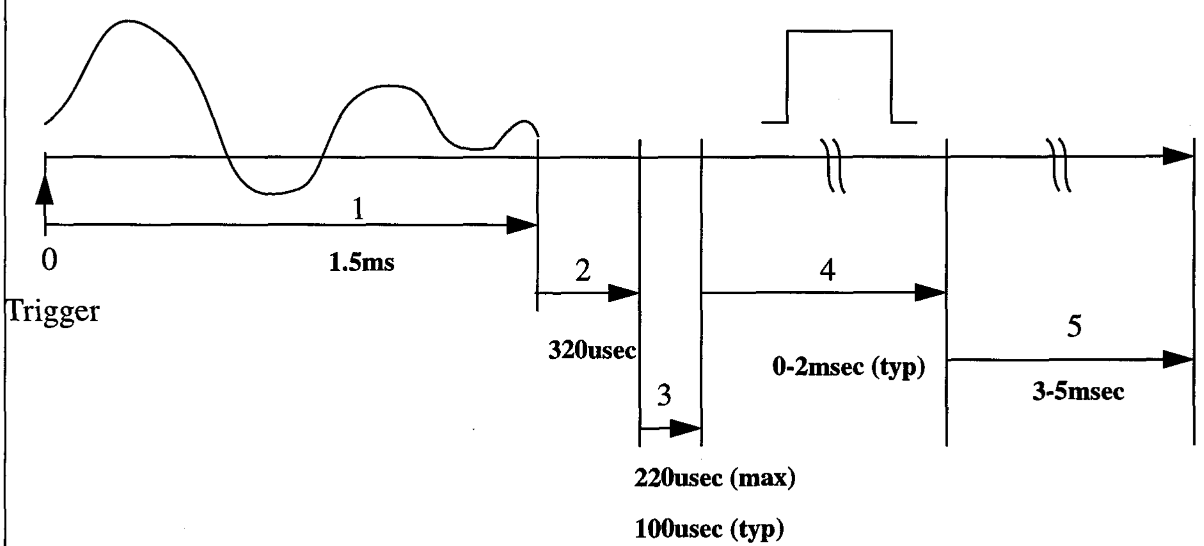
- Bringing up additional displays

3.2.1.1 Modes of Operation

The system presently has nine operating modes, selectable from the main display. These modes are primarily different combinations of two basic building blocks:

- Monitor: Acquire the fringe waveform and calculate velocities.
- Control: Issue a pulse to the coil driver in response to measurements made from the fringe signal.

A typical timing chart is shown in the following figure.



T0: Trigger ADC module from trigger module

1) Record fringe in ADC @100Ksamples/sec for 150 samples

2) Read ADC into uprocessor software variable (~2.1usec/sample)

3) Find zeros and calculate pulse width (variable to max 220usec if all 150 points must be scanned to find zero crossings)

4) Send control pulse to DAC for up to 6 msec (typically not greater than 2 to 3msec)

5) Release servo lockout relay (mechanical limitations of relay)

Figure 2: Typical Timing Chart

The individual operating modes are selected from a radio button selection in the middle section of the main display as described below.

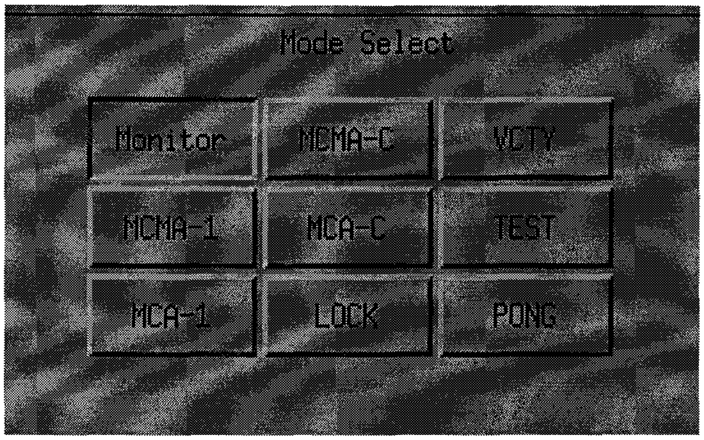


Figure 3: Mode Select of Main Display

These modes are:

Monitor: Analog servo is disconnected and system acquires fringe waveforms up to the 'Number of Waveforms' selected.

MCMA-1:

1) Analog servo is disconnected.
2) When first fringe detected, control pulse is calculated and sent to coil driver.
3) When second fringe is detected, it is acquired and analog servo output reconnected to coil driver.
4) Following fringes are only acquired by system as in Monitor mode until 'Number of Waveforms' setting is reached.

MCA-1: Same as MCMA-1 mode above, except analog servo is reconnected immediately after control pulse is sent to coil driver on first fringe.

MCMA-C:

1-3) Same as MCMA-1 steps 1 thru 3.
4) If a third fringe is detected, return to step one.
This process will continue until 'Number of Waveforms' setting is reached.

MCA-C: Same as MCA-1, except control pulses will be sent on every fringe detected until 'Number of Waveforms' setting is reached.

LOCK:

1) Disconnect analog servo.
2) When fringe detected, calculate and send control pulse; reconnect analog servo.
3) Wait time period as defined by 'Trigger Delay' setting.
4) If another fringe is detected, return to step 1.
This process will continue until 'Number of Waveforms' setting is reached.

VCTY: This mode acquires and calculates the velocity of fringes up to the 'Number of Waveforms' selection. This data is then plotted on the 'Velocity Plots' display.

TEST: Presently used to test and calibrate pulse width timing.

PONG: This mode uses a two pulse method to slow the test mass prior to reconnecting the analog servo. This mode is the same as the 'LOCK' mode, with the insertion of an additional step between 1 and 2 which calculates and sends a control pulse.

3.2.1.2 System Settings

Most system settings are entered via this display. This settings are shown in the following figure and defined below.

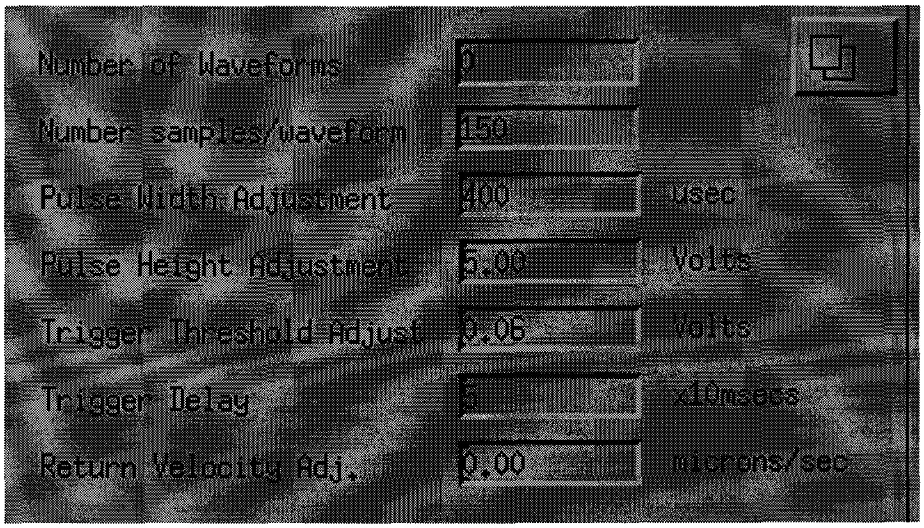


Figure 4: System Settings

Number of Waveforms: The number of fringes that the system is to acquire and/or monitor, depending on operating mode selected. (See next section) The range is 0-99.

Number samples/waveform: The number of samples to be taken by the VME ADC each time it is triggered by a fringe. The range is 0-255.

Pulse Width Adjustment: This is used in the calculation of the control pulse width shown at the end of this section.

Pulse Height Adjustment: This is the voltage of the system control pulse output.

Trigger Threshold Adjust: Level above which triggers will be passed to the acquisition ADC.

Trigger Delay: After control pulse has been applied and analog servo has been turned back on, time to allow second arm to lock before this system again looks for fringes.

Return Velocity Adj.: This setting is used to calculate the pulse width of the control pulse applied to the coil driver as calculated by:

$$pulsewidth = pulsewidth_{Adj} \times velocity_{Calc} + \frac{2.0}{pulseheight} \times velocity_{RVA}$$

3.2.1.3 System Triggering

Once the appropriate settings have been entered and mode selected, the system is triggered or turned on by selecting 'ARM' in the left hand corner of the main display as shown below. Once the system has completed running the sequence(s) selected by the operator, it will automatically stop and toggle the push-button back to the 'OFF' position.

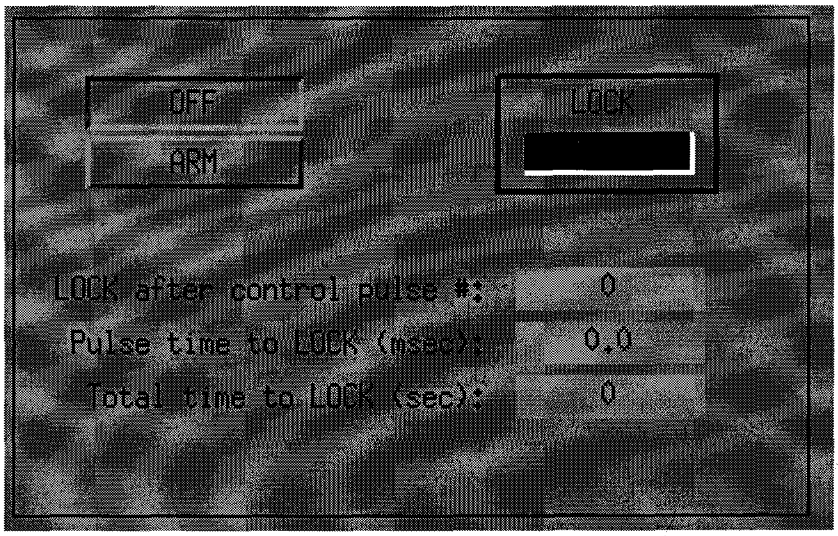


Figure 5: Main Display Triggering Section

Other items in this section of the display are:

LOCK: Indicates green when lock is acquired on the 40M secondary arm. Note that since there is no 'lock' signal from the interferometer to monitor by this system, LOCK is assumed if no fringes occur within a 10 second window. Therefore, lock may actually occur much sooner than the indicator responds and the system can be fooled by the interferometer being so far misaligned as no fringes are detected.

LOCK after control pulse #: The number of control pulses output by this system prior to its determining that lock has occurred. If lock has not occurred, this is simply the number of control pulses output by the system before the system stopped.

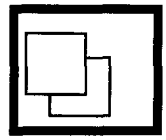
Pulse time to LOCK: Time in msec between fringes on which lock occurred.

Total time to LOCK: Time in sec from 'ARM' until lock was detected. If lock is not achieved, this is simply the total time the system attempted to reach a lock condition.

3.2.1.4 Calling Up Additional Displays

Additional operator screens may be displayed by selecting the menu icon in the upper right hand corner of the PT_MAIN display.

These displays are discussed in the following sections.



Histogram
Diagnostics
Standard Plots
Velocity Plots
PONG Plots

3.2.2. Standard Plots Display

This display is shown in Figure x. There are three primary parts:

- Waveform display area.
- Waveform select area.
- Data readout area.

3.2.2.1 Waveform Display Area

This lower section of the display will present up to three waveforms, depending on the 'Mode' selected on the main display. The first two (drawn on the screen in black and blue) display fringe signals. The third (drawn in white) displays the coil driver signal when 'LOCK' or 'MCMA-1' modes are selected.

3.2.2.2 Waveform Select Area

The upper left of this display has two sets of radio buttons for selecting which fringe is shown by the corresponding waveform trace. This allows for individual display selection of up to 100 stored fringe waveforms. These operate individually, except when in the 'LOCK' operating mode. When this mode is selected, use of the 'Trace 1' radio buttons will cause the waveforms to be displayed in groups i.e. Black trace shows the fringe on which the control pulse was issued, the Blue the fringe following the control pulse and the white the coil driver signal from the analog servo which correlates in time with the Blue fringe signal.

3.2.2.3 Data Readout Area

This area displays data associated with the two fringe trace plots. This includes:

Period: Time between first and third fringe zero crossings (usec)

ZX1 & ZX2: Time of first and third zero crossings relative to start of fringe waveform (usec).

Time -> Prev.: Time between this fringe signal and the previous fringe signal (msec).

Velocity: Calculated velocity from the fringe signal (microns/sec).

$$velocity = \frac{0.1112}{period^2} - \frac{0.0013}{period^3}$$

Note: Period is in msec units for this calculation.

Pulse width: Calculated response control pulse width sent out by system (usec).

3.2.3. PONG Plots

The 'PONG Plots' display is similar to the Standard Plots display, with the following changes made to accommodate this mode:

- Single Trace or Set Selection: All traces in the waveform display area are brought up in sets in accordance with the PONG operating mode.
- Settings area (Upper left, as shown below) has addition of:
  - Allowable time between fringes (msec): This is the amount of time allowed between first and second fringes on which control pulses are applied.
  - SAVE: Selecting this button with cause data from the current waveform presentation to be saved to file. This data includes all three displayed fringe data readouts and the waveform associated with the coil driver (in volts).
- Data Readout Area:
  - Row added to provide data readouts for three fringe signals.
  - Field added at end which displays the measured control pulse width.

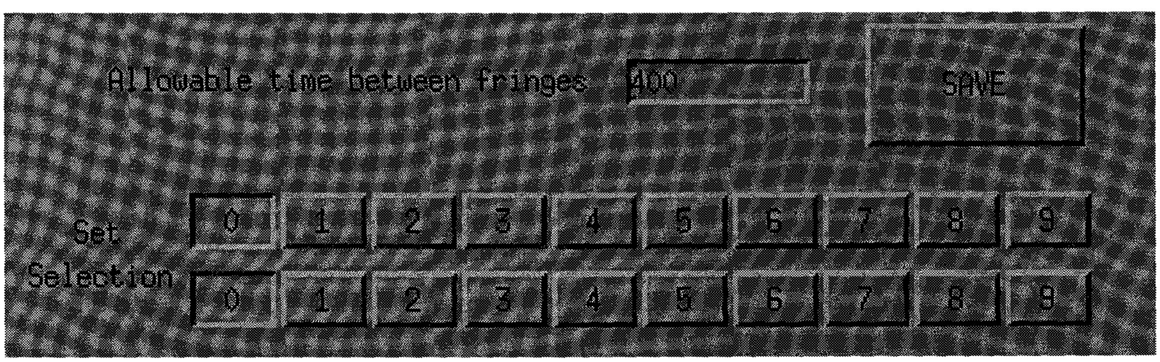


Figure 6: PONG Display Setting Area

3.2.4. Velocity Plot

This display is active when operating in the 'VELOCITY' mode. It has a single waveform presentation area, which will plot fringe velocities vs. time for the number of waveforms acquired.

3.2.5. Histogram

This display is used in conjunction with the 'Monitor' mode of operation. After an operator selected number of fringes have been monitored, selecting the 'HISTOGRAM' push-button on this screen will cause the plot area to display the peak value of all acquired fringes.

3.2.6. Diagnostics

This display is only for use for testing the system.

4 SOFTWARE DETAILED DESCRIPTION

4.1. Overview

Software for this project was developed using EPICS as the infrastructure for operator displays and data communications. C code was written and embedded within the EPICS SNL framework to perform the real-time processes in the VME system.

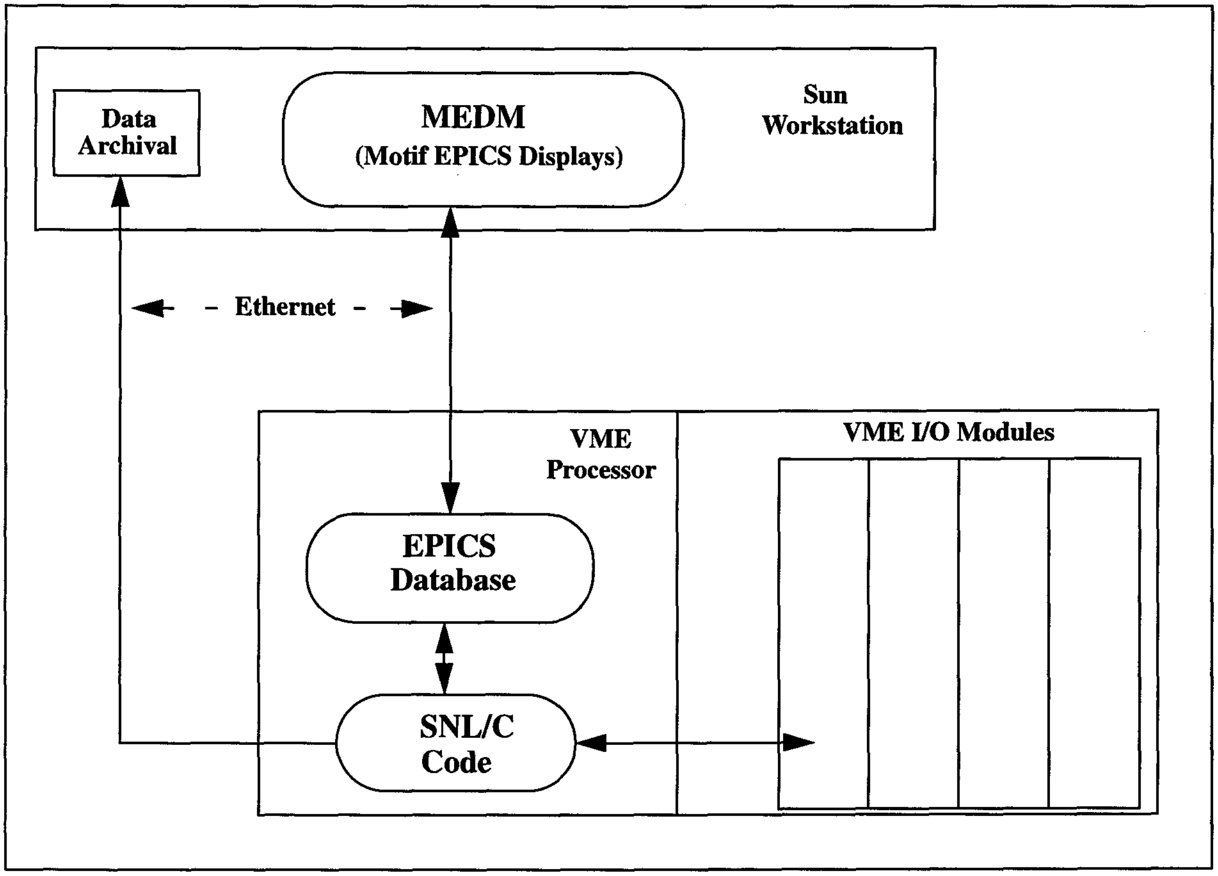


Figure 7: Pulse Damping Test Software Architecture

The figure above shows the basic architecture of the software and the data connections. EPICS is used for its display capabilities on the Sun workstation through medm and its real-time database in the VME to make use of EPICS communication capabilities. All real-time control/data acquisition and communication with the VME modules is done through custom C code embedded within the EPICS SNL framework. The SNL framework was used mainly for its calls to communicate with the EPICS database and is structured more as a standard piece of code rather than a "state machine". Also, since EPICS does not support the archival of array data, code was also developed to directly archive data across the network via the Sun workstation. This data archival is located on 'leopardess' in /usr/CDS/data.

4.2. Database in subdirectory ./Vac1Db

The EPICS database for this project was created using GDCT and saved in file “pulser”. This file unfortunately also contains components used for a PSL demo. Attempts were made to delete these extra items, but it ended up with a corrupted file (something wrong with GDCT??).

Due to this being a rush job, with daily spec. changes, the database is not well (at all) documented. To determine which records do what, it is best to reference the SNL code. A data dictionary should be developed if time permits.

All records used for the pulse damping test are “Passive” and are not linked, either for data or process sequencing (except one set for determining operator Trace select).

4.3. State Notation Language in subdirectory ./src

For this project, all real-time execution is done in a SNL program, comprised mostly of embedded C code. This file is ‘wv.st’. A makefile exists in the base directory to compile this into ‘wv.o’ in the ./mv167 directory for loading to the IOC. To make the file, execute: ‘make bldVx’ in the VacuumApp directory.

4.3.1. Start-up

This program is automatically loaded and executed whenever the VME crate is powered up or rebooted. It may also be dynamically loaded and unloaded, such as when changes are made, without rebooting the system. To do this:

rlogin to scipe1
execute: td “pulse”
execute: ld < wv.o (After file has been compiled on Sun and is ready to load)
execute: seq &pulse

On sequence start, the rlogin terminal should print a message “Pulse seq running” to indicate that the code has found all EPICS database records it needs and VME modules and is ready to continue.

4.3.2. Program Description

The various states of the SNL program are shown in the following diagram. For this particular project, the states are more subroutines than true states, as most of the code is C code which is not precompiled by the SNL system. SNL in this project was used more for its ability to monitor and interface with EPICS database records than for sequencing. Future projects will probably use stand-alone C code with Channel Access calls, but, in the short term of this project, this was the quickest method of implementation.

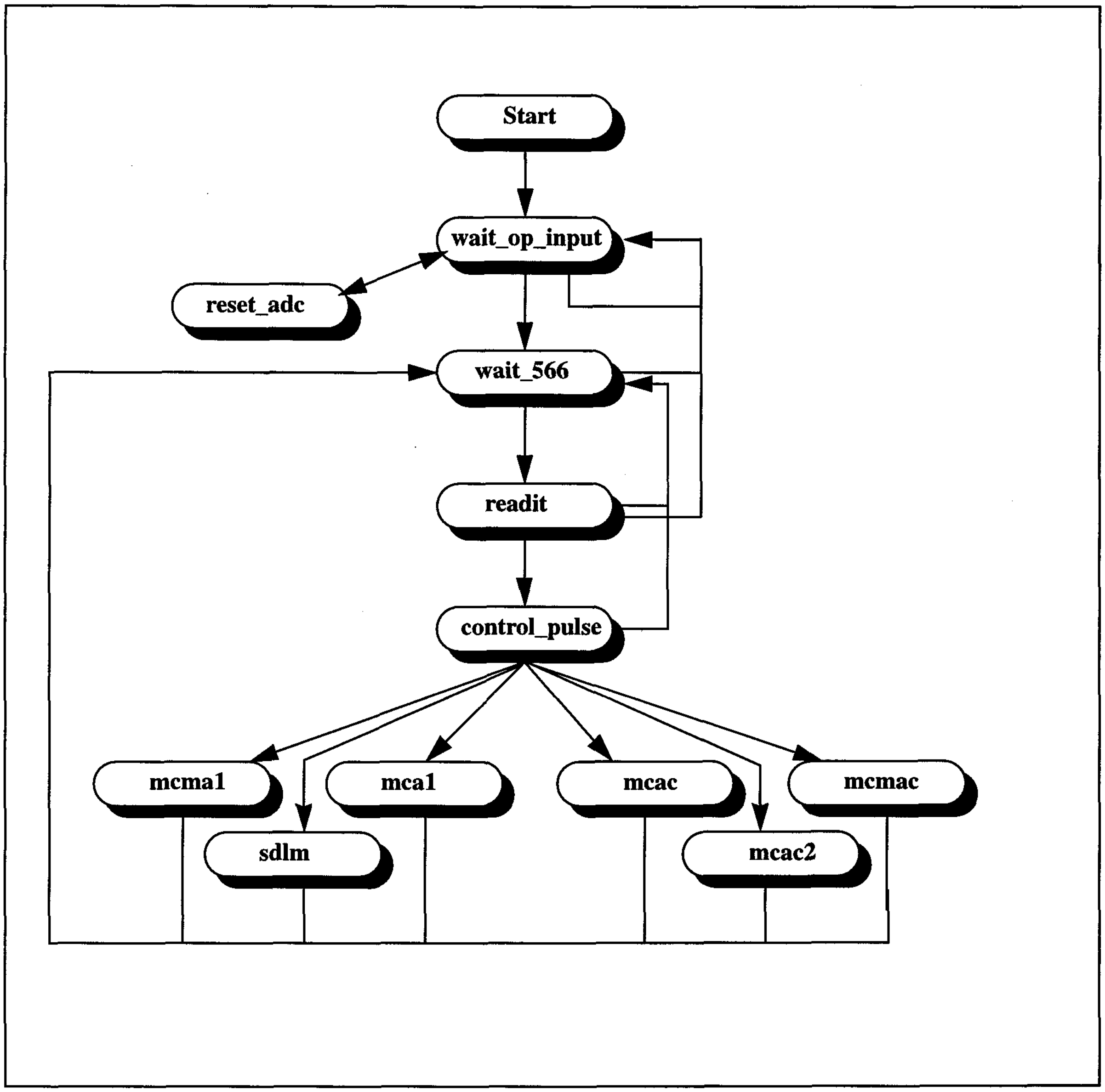


Figure 8: Pulser States

Figure 8: Pulser States is only a basic road map to the various SNL states. The following paragraphs only give a brief description of the states. Refer to the code itself for more information on the internal working of the states themselves.

4.3.2.1 STATE start

This state initializes the VME module address space pointers and sets a few start-up values to the DAC channels.

4.3.2.2 STATE wait_op_input

This is where the program typically idles when it is not actually performing data acquisition/control functions. It waits here for an input from the operator via the operator displays. If the operator request is simply a new pulse width, or similar, setting, it will reset its internal variables accordingly, then return to wait_op_input. This is also true when a new waveform trace, or similar data, is requested from the system.

Transition out of this state only occurs under two conditions, as shown in the following figure.

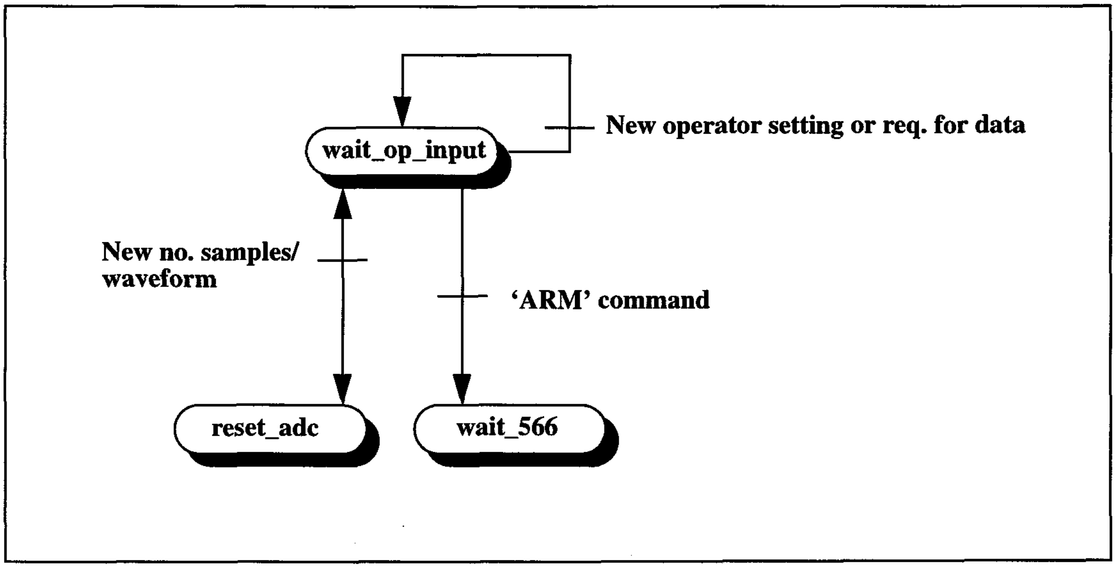


Figure 9: Transitions from wait_op_input (IDLE) state

4.3.2.3 STATE reset_adc

This state is called when:

Operator requests a manual reset of the ADCs
New 'Number of samples/waveform' is entered by the operator.

This state sends all setup information to the XY566 modules required for data acquisition, then returns to the wait_op_input state.

4.3.2.4 STATE wait_566

The primary functions performed by this state are:

- Arm one or both ADC modules and determine when acquisition is complete.
- Once acquisition cycle is complete, send data back to operator screens.

Transitions into/out of this state are shown in the following figure.

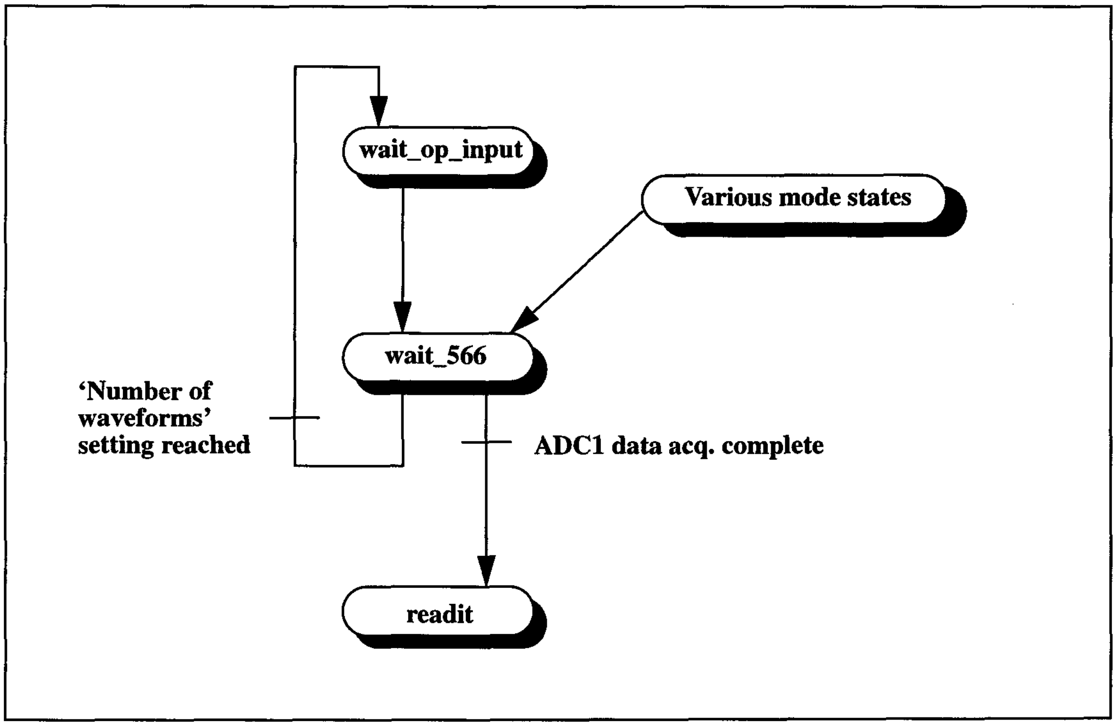


Figure 10: wait_566 Transition Diagram

The code processing within this state are shown in Figure 10. Primarily, this code arms ADC1 and waits for a fringe to be acquired. It then transitions to state readit. If operation is complete, denoted by the number of waveforms acquired equals the number of waveforms requested by the operator, this state transitions back to the wait_op_input state.

In addition to arming ADC1, this section of code will also arm ADC2 to acquire analog servo coil driver data when requested by other 'mode' states. In addition, when in MCMA-1 mode, ADC2 is software triggered to acquire the pulse output by the system. (This is for test only).

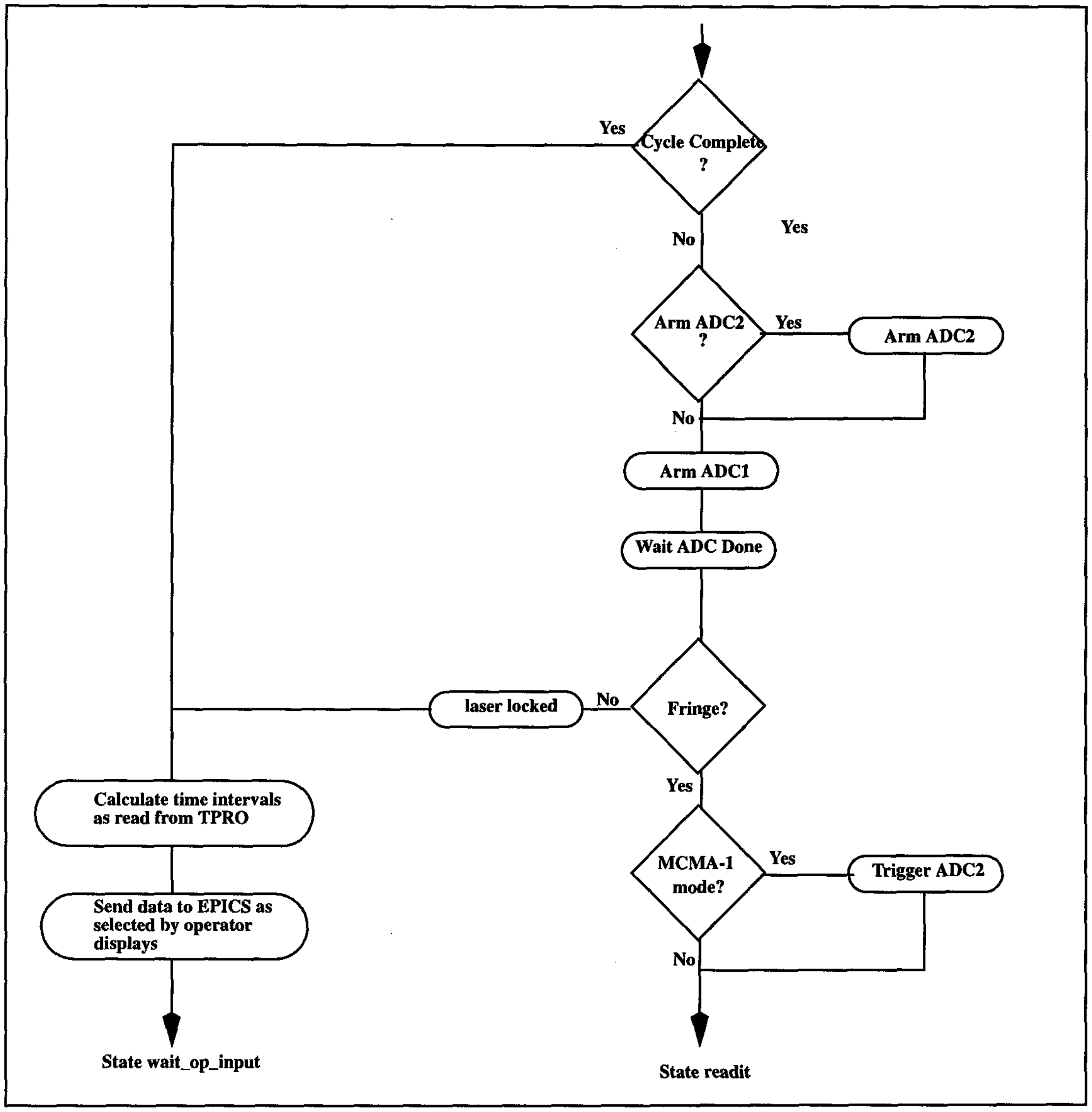


Figure 11: State wait_566 Flow Diagram

4.3.2.5 STATE readit

In this state:

- Data is read in from ADC1
- Zero crossings are located
- Velocity is calculated

State transition and flow diagrams appear in the following two figures.

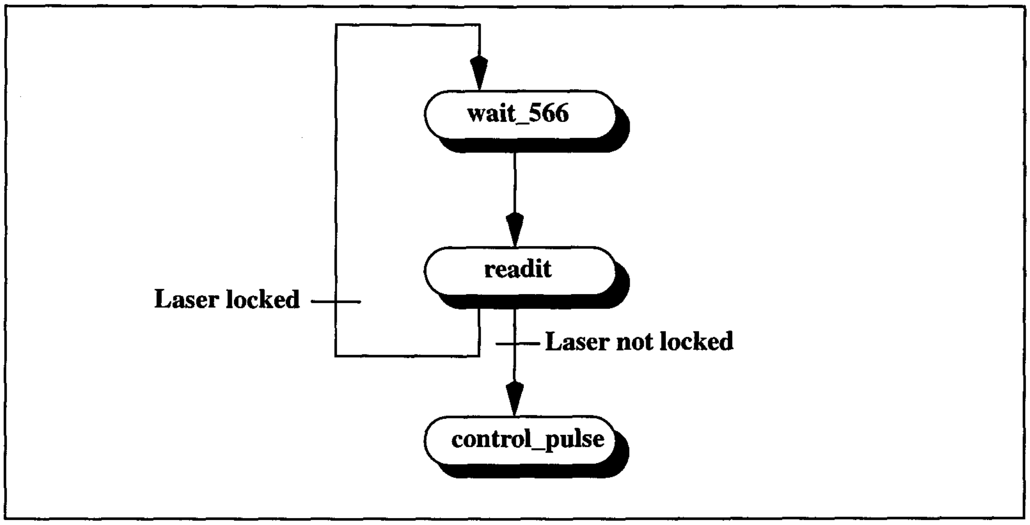


Figure 12: readit State Transition Diagram

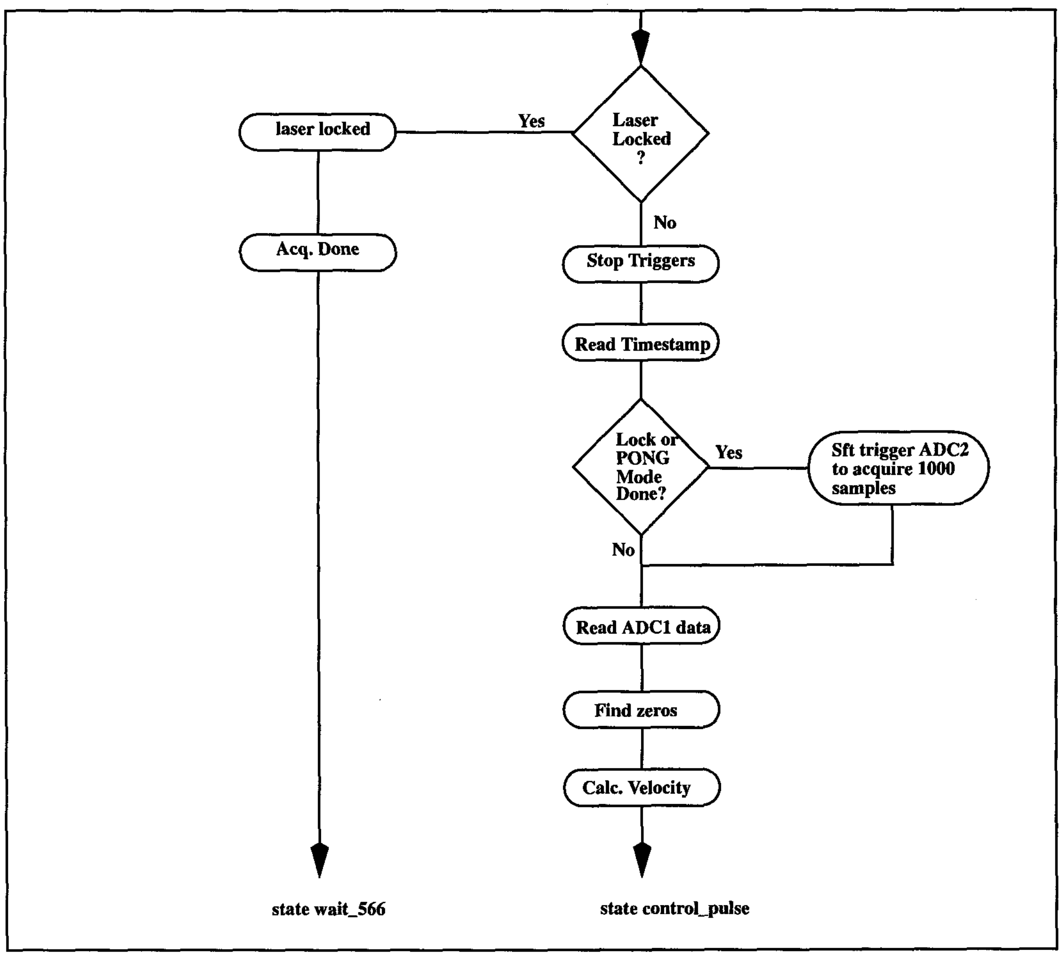


Figure 13: State readit Flow Diagram

4.3.2.6 STATE control_pulse

This section checks which control mode has been selected by the operator and calls the appropriate state. The only time this is not true is when 'TEST' mode has been selected. In this case, a dummy control pulse is fired via a spare DAC channel to calibrate the pulse width timing and returns to state wait_566.

4.3.2.7 Mode States

For each mode of operation which provides a output pulse to the coil driver, a state exists which supports that mode of operation. These modes were previously described in the Operations section. The only mode further described here in detail is the PONG mode. Other mode states are similar, though the sequences vary. All mode states transition back to wait_566 when they complete.

4.3.2.7.1 State sdlm (PONG Mode)

This state supports the PONG mode of operation by providing control pulses in groups of two. The logic diagram is shown below.

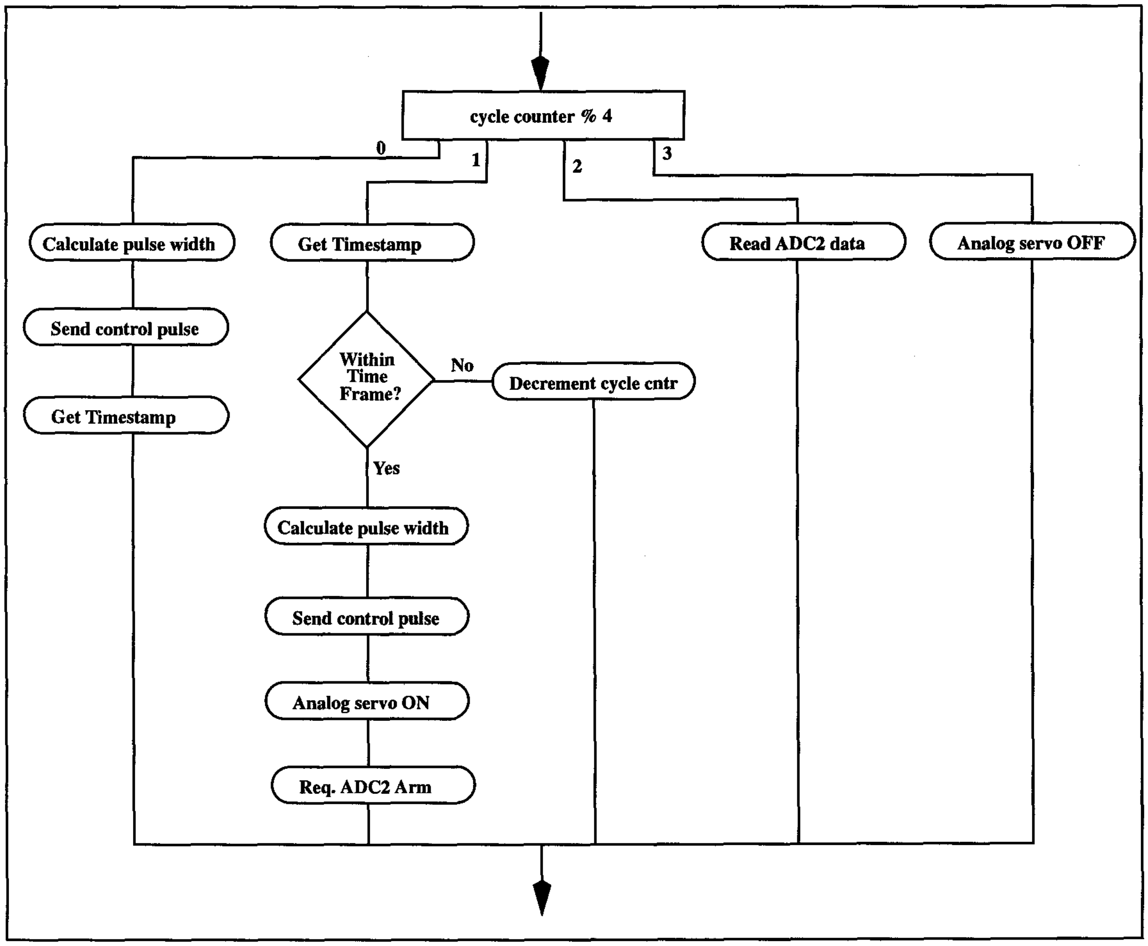


Figure 14: State sdlm (PONG) Flow Diagram

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

Technical Note	LIGO-T950012-00	-C	3/28/95
Document Type	*Doc Number*	*Group-Id*	*Date*

Pulse Damping Test

Control System Description

Title

R. Bork

Author(s)

This is an internal working note
of the LIGO Project

California Institute of Technology
LIGO Project - MS 102-33
Pasadena CA 91125
Phone (818) 395-2966
Fax (818) 304-9834
E-mail: info@ligo.caltech.edu
WWW: http://www.ligo.caltech.edu

TABLE OF CONTENTS

1 OVERVIEW

This document describes the hardware and software developed for the 'Smart Lock Acquisition' or 'Pulse Damping' test performed on the LIGO 40M prototype. These tests were conducted to study and model lock aquisition and define methods to assist the analog servo system in lock acquisition i.e. speed up the lock acquisition time.

2 SYSTEM ARCHITECTURE

2.1. Hardware

This system was designed with the constraint of making use of existing LIGO hardware. If this system is adopted for future use, the hardware system should be re-designed to make use of more appropriate hardware.

The system consists of a single VME crate w/modules and a single NIM module. The basic layout is shown in the following figure. Detailed schematics and layouts appear in Appendix A.

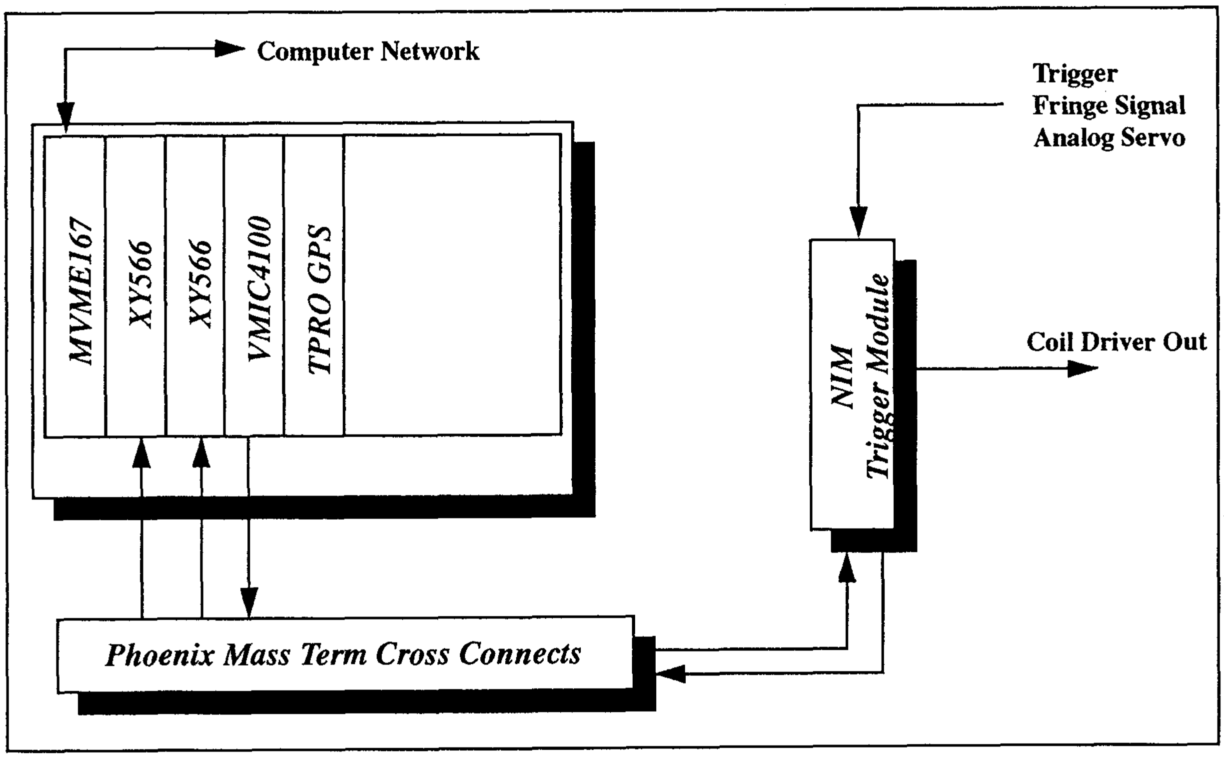


Figure 1: System Hardware Layout

MVME167: System real-time processor board with Motorola 68040 processor.

3 SYSTEM OPERATION

3.1. Start-up

3.1.1. Hardware Power up

The VME crate and NIM module reside in a rack near the vertex of the 40M prototype. This system is not normally cabled into the prototype except during testing. Therefore, cables should be connected in accordance with the 40M log book entries. The coil driver output should be left disconnected until all equipment has been powered up as the power up states of the VME modules are not controllable.

The VME and NIM crates should be powered on. The VME processor will automatically boot, load the software, and begin operation.

3.1.2. Operator Console

At present, the operator console can be any LIGO Sun workstation. The workstation actually performing the operator console functions, i.e. communicating with the VME processor, will be 'leopardess'. The workstation actually used by the operator will be simply an X term to that workstation. This is because, at present, this is one of the few workstations configured to run EPICS and also the only machine which can broadcast on the 40M subnet.

The first step to start operation from a workstation is to bring up the EPICS display manager and the pulse test operator displays. The sequence is:

1) X windows must be running on the workstation.
2) From a window, add 'leopardess' as an X host (xhost +).
3) Rlogin to 'leopardess'.
4) cd to /home/CDS/a/A3.11.4/VacuumApp/screens
5) setenv DISPLAY your_computer_name:0
6) execute 'medm -display your_computer_name:0 -cleanup&
The Motif EPICS Display Manager (medm) window should now come up on the screen.
7) From the medm 'FILE' menu, select 'OPEN'; a file browser will appear.
8) Scroll down in the browser and select 'PT_MAIN.adl'; the operator screen will now appear in a new window in EDIT mode.
9) Select 'EXECUTE' from the medm main window. The screen will become 'live' and should fill in the appropriate real-time data.

3.2. Display Description and Operation

3.2.1. PT_MAIN

This display is the primary one for operation of the system. It allows for:

- System Triggering
- Mode Selection
- System Settings

The individual operating modes are selected from a radio button selection in the middle section of the main display as described below.

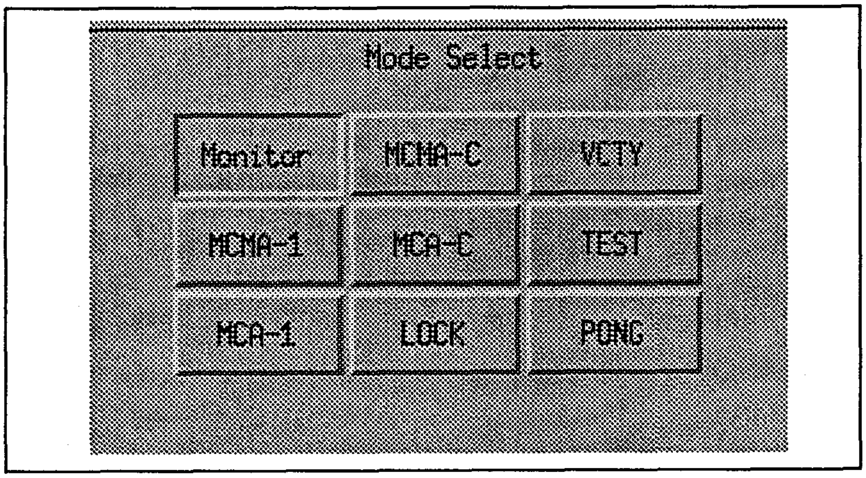


Figure 3: Mode Select of Main Display

These modes are:

Monitor: Analog servo is disconnected and system acquires fringe waveforms up to the 'Number of Waveforms' selected.

MCMA-1: 1) Analog servo is disconnected.

2) When first fringe detected, control pulse is calculated and sent to coil driver.
3) When second fringe is detected, it is acquired and analog servo output reconnected to coil driver.
4) Following fringes are only acquired by system as in Monitor mode until 'Number of Waveforms' setting is reached.

MCA-1: Same as MCMA-1 mode above, except analog servo is reconnected immediately after control pulse is sent to coil driver on first fringe.

MCMA-C: 1-3) Same as MCMA-1 steps 1 thru 3.

4) If a third fringe is detected, return to step one.
This process will continue until 'Number of Waveforms' setting is reached.

MCA-C: Same as MCA-1, except control pulses will be sent on every fringe detected until 'Number of Waveforms' setting is reached.

LOCK: 1) Disconnect analog servo.

2) When fringe detected, calculate and send control pulse; reconnect analog servo.
3) Wait time period as defined by 'Trigger Delay' setting.
4) If another fringe is detected, return to step 1.
This process will continue until 'Number of Waveforms' setting is reached.

VCTY: This mode acquires and calculates the velocity of fringes up to the 'Number of Waveforms' selection. This data is then plotted on the 'Velocity Plots' display.

TEST: Presently used to test and calibrate pulse width timing.

Return Velocity Adj.: This setting is used to calculate the pulse width of the control pulse applied to the coil driver as calculated by:

$$pulsewidth = pulsewidth_{Adj} \times velocity_{Calc} + \frac{2.0}{pulseheight} \times velocity_{RVA}$$

3.2.1.3 System Triggering

Once the appropriate settings have been entered and mode selected, the system is triggered or turned on by selecting 'ARM' in the left hand corner of the main display as shown below. Once the system has completed running the sequence(s) selected by the operator, it will automatically stop and toggle the push-button back to the 'OFF' position.

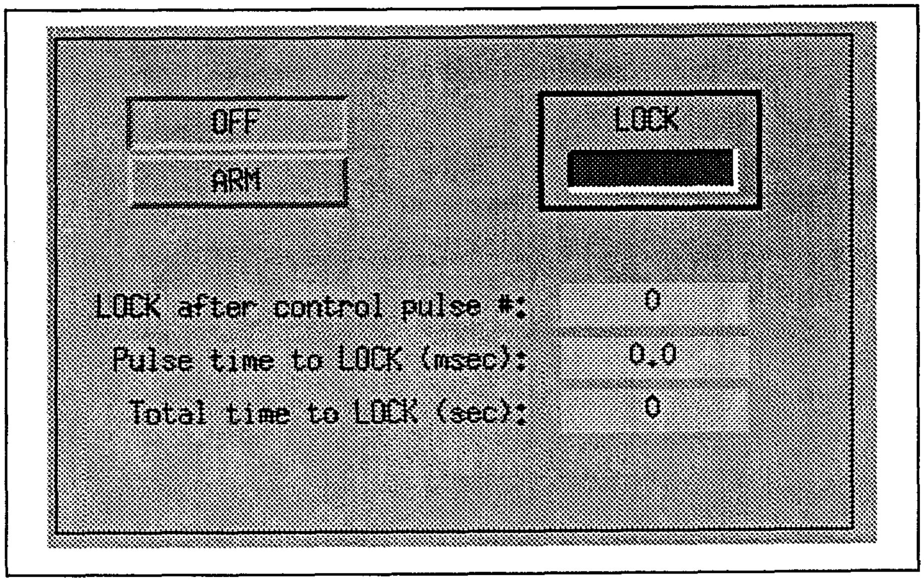


Figure 5: Main Display Triggering Section

Other items in this section of the display are:

LOCK: Indicates green when lock is acquired on the 40M secondary arm. Note that since there is no 'lock' signal from the interferometer to monitor by this system, LOCK is assumed if no fringes occur within a 10 second window. Therefore, lock may actually occur much sooner than the indicator responds and the system can be fooled by the interferometer being so far misaligned as no fringes are detected.

LOCK after control pulse #: The number of control pulses output by this system prior to its determining that lock has occurred. If lock has not occurred, this is simply the number of control pulses output by the system before the system stopped.

Pulse time to LOCK: Time in msec between fringes on which lock occurred.

$$velocity = \frac{0.1112}{period^2} - \frac{0.0013}{period^3}$$

Note: Period is in msec units for this calculation.

Pulse width: Calculated response control pulse width sent out by system (usec).

3.2.3. PONG Plots

The 'PONG Plots' display is similar to the Standard Plots display, with the following changes made to accommodate this mode:

- Single Trace or Set Selection: All traces in the waveform display area are brought up in sets in accordance with the PONG operating mode.
- Settings area (Upper left, as shown below) has addition of:
  - Allowable time between fringes (msec): This is the amount of time allowed between first and second fringes on which control pulses are applied.
  - SAVE: Selecting this button with cause data from the current waveform presentation to be saved to file. This data includes all three displayed fringe data readouts and the waveform associated with the coil driver (in volts).
- Data Readout Area:
  - Row added to provide data readouts for three fringe signals.
  - Field added at end which displays the measured control pulse width.

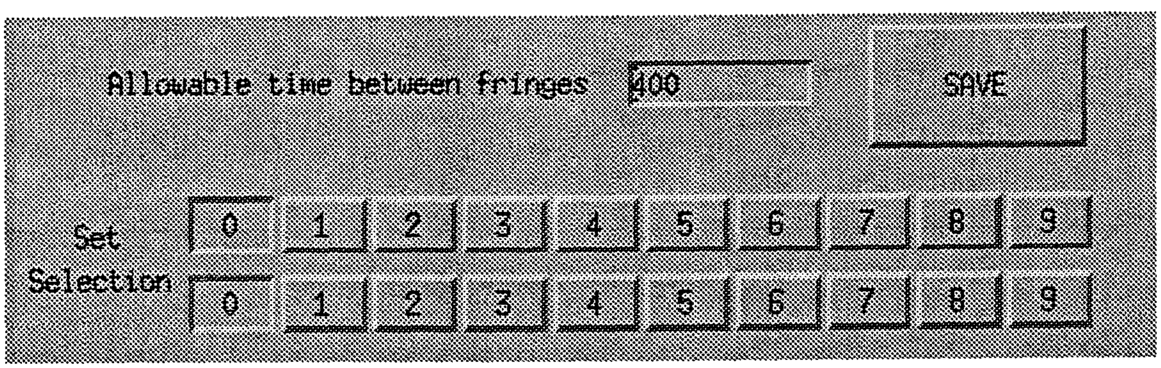


Figure 6: PONG Display Setting Area

3.2.4. Velocity Plot

This display is active when operating in the 'VELOCITY' mode. It has a single waveform presentation area, which will plot fringe velocities vs. time for the number of waveforms acquired.

4 SOFTWARE DETAILED DESCRIPTION

4.1. Overview

Software for this project was developed using EPICS as the infrastructure for operator displays and data communications. C code was written and embedded within the EPICS SNL framework to perform the real-time processes in the VME system.

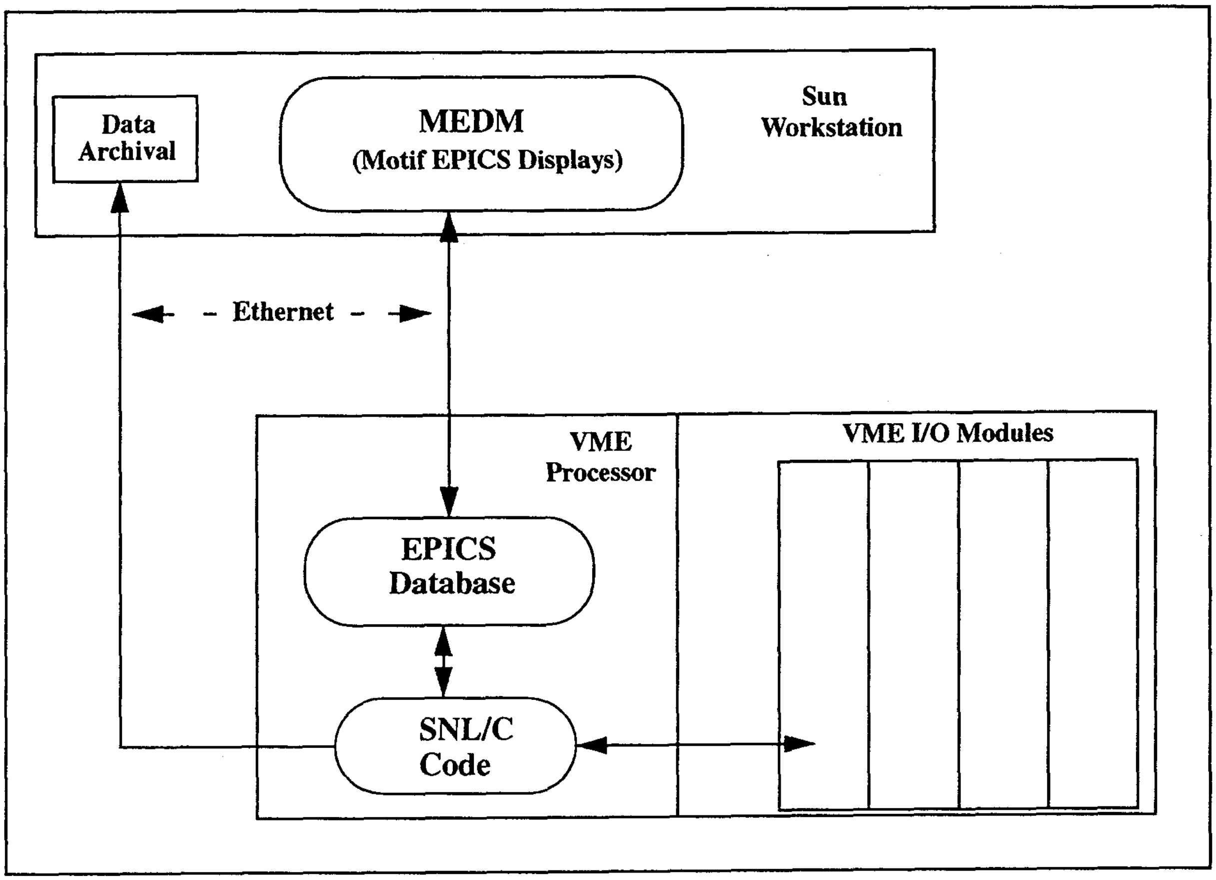


Figure 7: Pulse Damping Test Software Architecture

The figure above shows the basic architecture of the software and the data connections. EPICS is used for its display capabilities on the Sun workstation through medm and its real-time database in the VME to make use of EPICS communication capabilities. All real-time control/data acquisition and communication with the VME modules is done through custom C code embedded within the EPICS SNL framework. The SNL framework was used mainly for its calls to communicate with the EPICS database and is structured more as a standard piece of code rather than a "state machine". Also, since EPICS does not support the archival of array data, code was also developed to directly archive data across the network via the Sun workstation. This data archival is located on 'leopardess' in /usr/CDS/data.

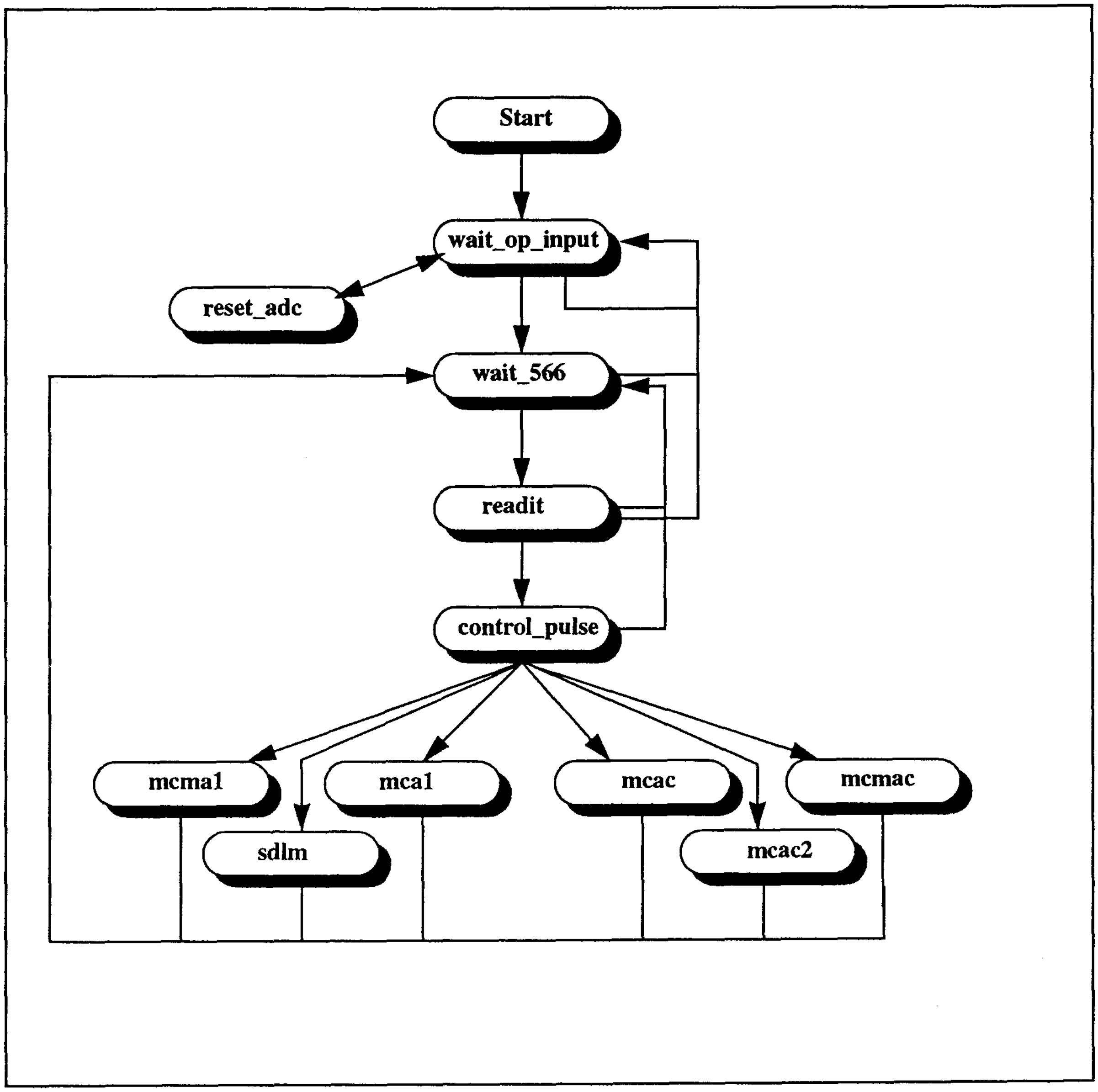


Figure 8: Pulser States

Figure 8: Pulser States is only a basic road map to the various SNL states. The following paragraphs only give a brief description of the states. Refer to the code itself for more information on the internal working of the states themselves.

4.3.2.1 STATE start

This state initializes the VME module address space pointers and sets a few start-up values to the DAC channels.

Transitions into/out of this state are shown in the following figure.

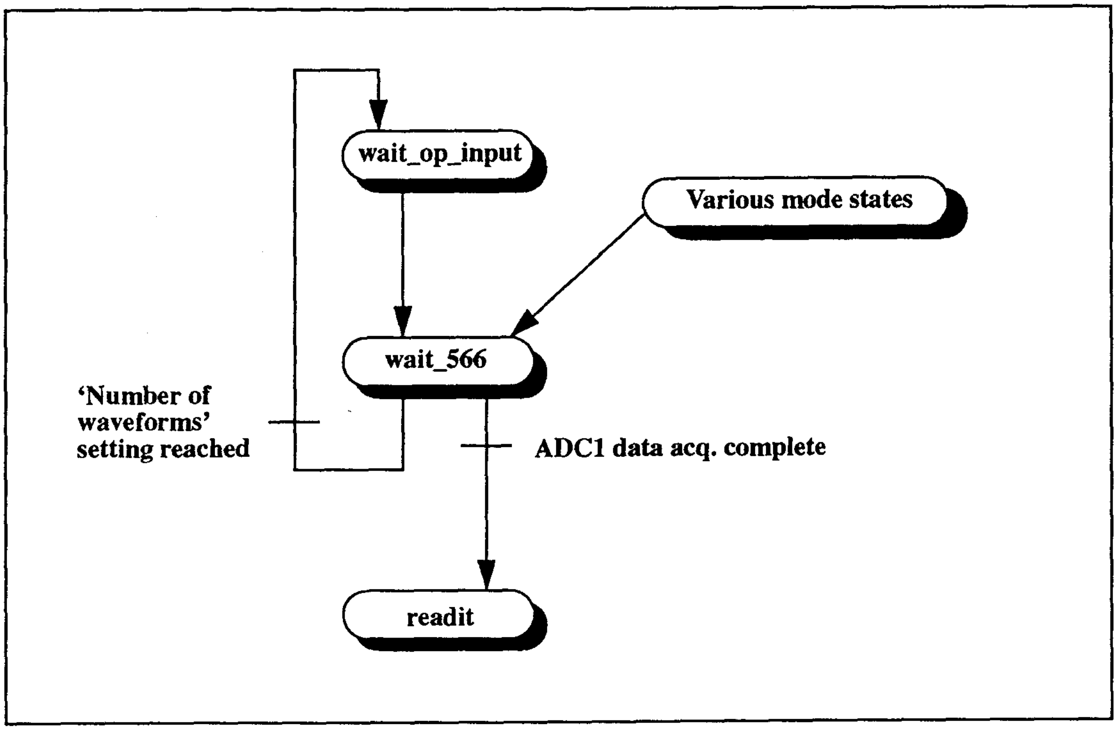


Figure 10: wait_566 Transition Diagram

The code processing within this state are shown in Figure 10. Primarily, this code arms ADC1 and waits for a fringe to be acquired. It then transitions to state readit. If operation is complete, denoted by the number of waveforms acquired equals the number of waveforms requested by the operator, this state transitions back to the wait_op_input state.

In addition to arming ADC1, this section of code will also arm ADC2 to acquire analog servo coil driver data when requested by other 'mode' states. In addition, when in MCMA-1 mode, ADC2 is software triggered to acquire the pulse output by the system. (This is for test only).

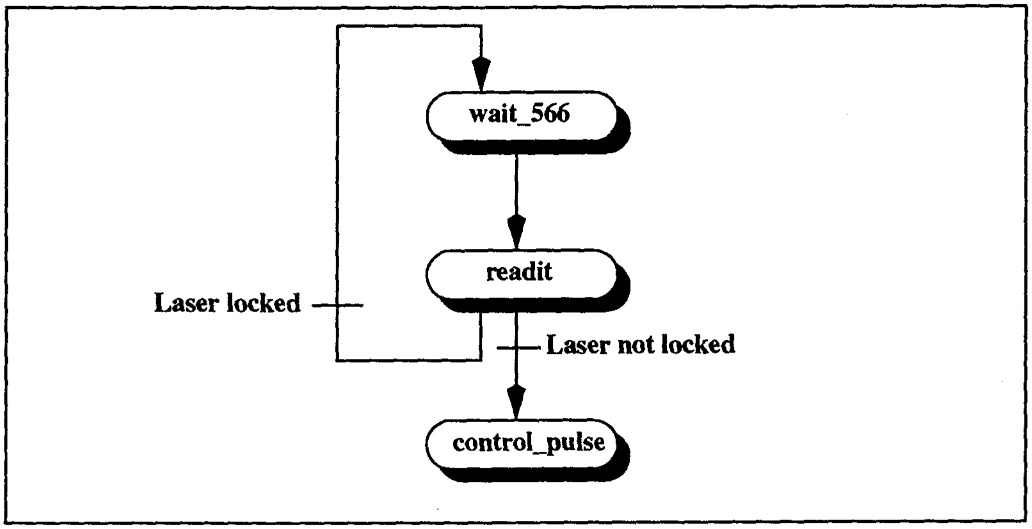


Figure 12: readit State Transition Diagram

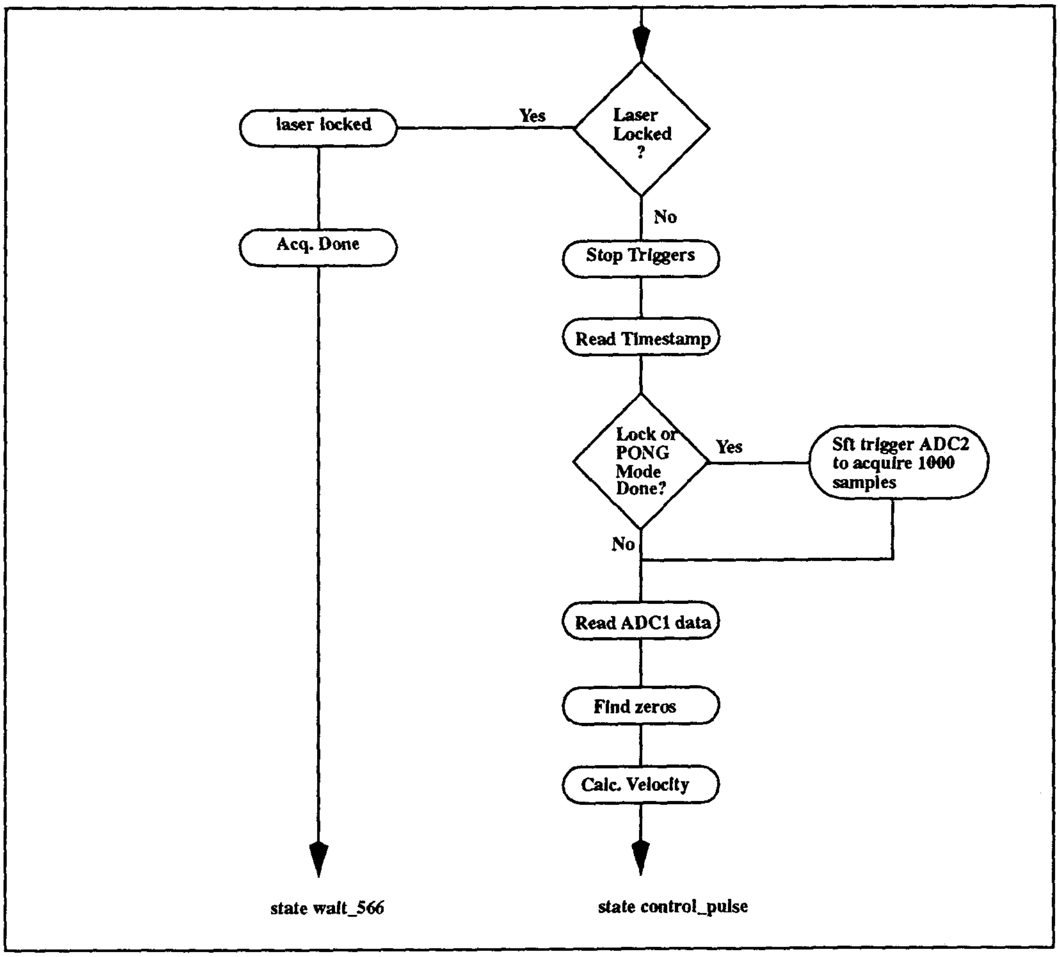


Figure 13: State readit Flow Diagram