CMS Front-End Driver

PMC

Prototype

Test Bench Manual

Version 1.1,January 2001

J. Coughlan
Rutherford Appleton Laboratory

1 Contents

1 CONTENTS 2

2 FIGURES 3

3 FOREWORD 4

4 INTRODUCTION 4

5 TEST TEAM 4

6 HARDWARE COMPONENTS 5

7 SOFTWARE COMPONENTS 8

8 GETTING STARTED WITH FEDPMC.VI 9

9 MACINTOSH PARAMETERS 11

10 NI MXI PCI-VME LINK 11

10.1 HARDWARE 11
10.2 SETTING UP THE LINK 11
10.2.1 Running VXIinit application 11
10.2.2 Running resman application 12
10.2.3 Other comments 14

11 "VERSATERM PRO" TERMINAL EMULATOR 14

11.1 TROUBLESHOOTING 14

12 BOOTING RIO2 AND PPC_MON 15

13 RUNNING FED_MON 16

13.1 RUNNING FED_MON AS THE BOOT PROGRAM 16
13.2 RUNNING FED_MON FROM THE NETWORK 18
13.3 RIO2 BOOT PARAMETERS FOR FED_MON 19
13.4 STORING FED_MON IN RIO2 FLASH MEMORY. 20
13.5 RIO2 BOOT PARAMETERS FOR FED_MON 20

14 TFTPD FILE SERVER 22

15 RUNNING THE LABVIEW APPLICATION FEDPMC.VI 23

16 CHECKING FED-PMC FUNCTIONALITY WITH FED_MON 27

16.1 TESTING WITH INTERNAL CLOCK AND SOFTWARE TRIGGERS 27
16.2 TESTING WITH EXTERNAL CLOCK AND EXTERNAL TRIGGERS 28

17 PROGRAMMING THE FED-PMC 30

17.1 PROGRAMMING THE LATTICE CPLD 30
17.2 PROGRAMMING XILINX FPGA 31
17.2.1 Programming the Xilinx Flash EPROM for the first time 31
17.2.2 Re-programming the Xilinx Flash EPROM 31
17.3 PROGRAMMING THE PLX SERIAL EPROM 31
17.4 STORING THE FED-PMC SERIAL NUMBER 32

18 DOCUMENT SERVERS.....33

19 REFERENCES.....34

20 INDEX.....35

2 Figures

FIGURE 1 FED-PMC TEST BENCH..... 5
FIGURE 2 FED-PMC TEST BENCH BLOCK DIAGRAM..... 6
FIGURE 3 FED-PMC TEST BENCH CONNECTIONS..... 7
FIGURE 4 FEDPMC.VI MAIN WINDOW.....23
FIGURE 5 FEDPMC.VI RUN SETTINGS PANEL.....24
FIGURE 6 FEDPMC.VI EXAMPLE OF DATA CAPTURE.....25

3 Foreword

This documents describes the test bench set up used for commissioning CMS Front-End Driver FED-PMC prototype cards.

This document can be obtained from the following anonymous ftp site:

address : **ftp.te.rl.ac.uk**
directory: **cms/fed/fed_pmc/docs/manuals/ fedpmc_testbench.pdf**

Please send any comments on the contents of this document to the author J.Coughlan@rl.ac.uk

4 Introduction

The test bench for the CMS FED-PMC's Mk3 production is based on a VME crate system with a CES RIO2 as the PMC carrier. The VME crate is linked to a Macintosh running a LabView application which controls the FED-PMC and displays the captured data.

This system has been used in the commissioning of Mk1 and Mk2 FED-PMC's in 1999 and 2000.

5 Test Team

Project Manager & Test Software : John Coughlan
Test Engineer : Jonathan Godwin
Digital & Analogue tests : James Salisbury

6 Hardware components

Figure 1 shows the test bench setup (as of January 2001).

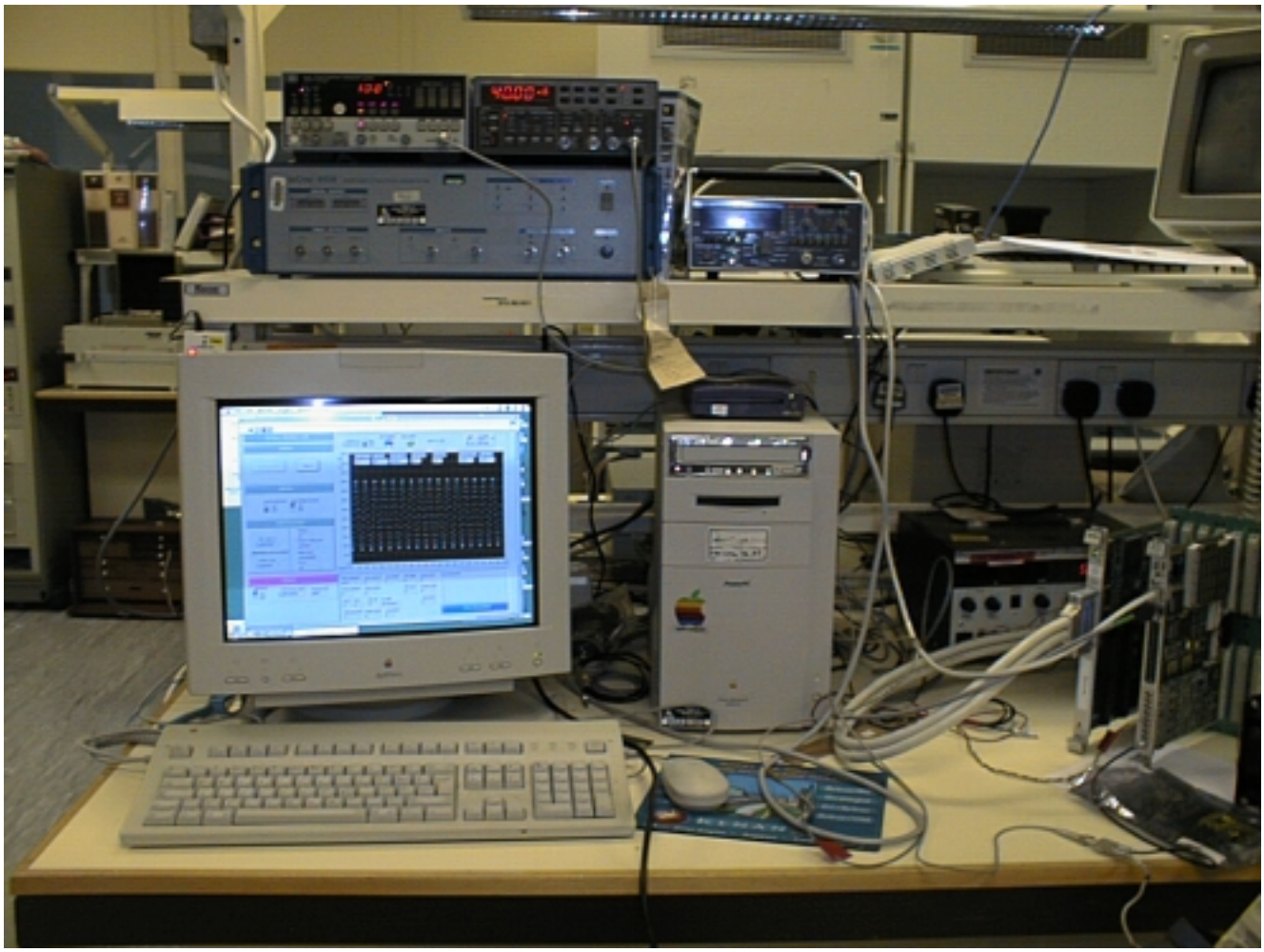

Figure 1 FED-PMC Test Bench

The major hardware components are :

• 6U VME backplane and power supply.
• Networked CES RIO2 VME PowerPC Single Board Computer. (RIO2 Models 8060/8061/ 8062)
[plus small external fan unit for cooling the RIO2, as PowerPC on RIO2 can get quite hot]

• Networked Power Macintosh 8500/120 (with internal PCI Bus slots).

VME crate is connected to Mac using the National Instruments MXIbus (section 10)
• NI VME-MXI-2 VME card (VME arbitor in slot 1)
• NI MXI PCI card in Mac PCI bus slot.

The Mac is connected to RIO2 by serial RS232 cable.

The Mac and RIO2 must be connected to Ethernet.

• Signal generator unit for 40 MHz Clock.

- Signal generator unit for external trigger.
- Signal generator unit for analogue data source (e.g. sine wave).

- Special PCB with TTL->LVDS converter chip (for clock and trigger signals) and associated power supply.
- Technobox VME PMC extender card

- PC with Lattice cable for loading Lattice CPLD.

A block diagram of the test bench is shown in Figure 2 and a list of connections in Figure 3.

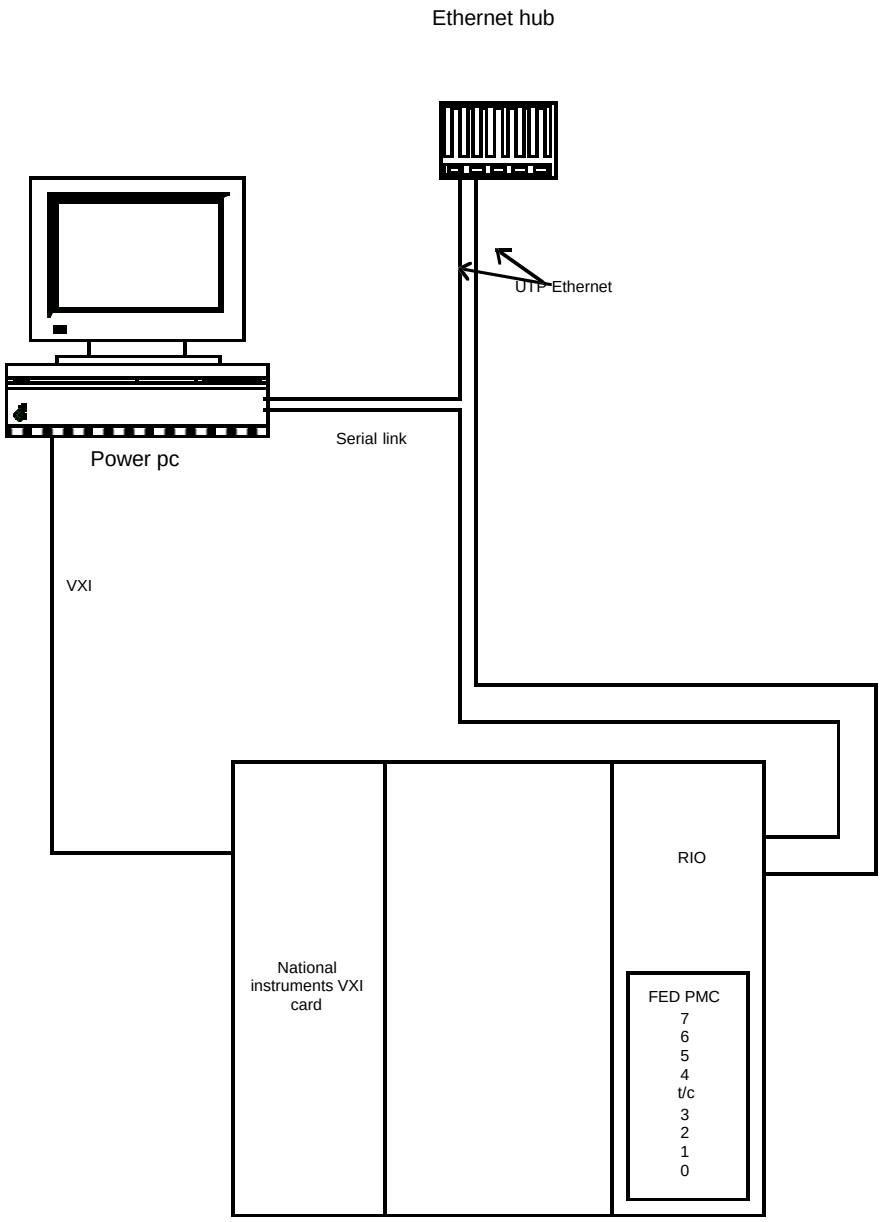

Figure 2 FED-PMC Test Bench Block Diagram

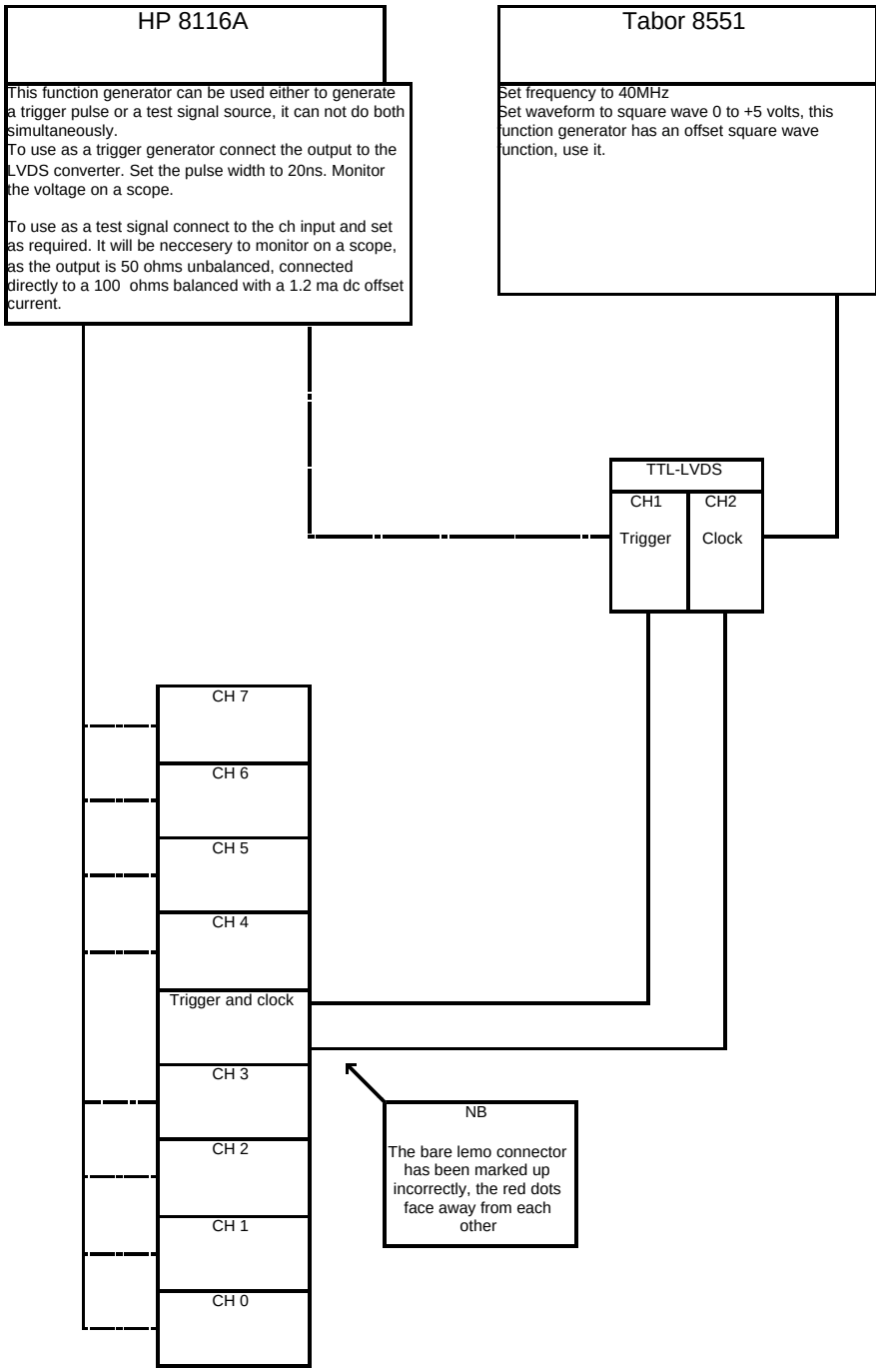

Figure 3 FED-PMC Test Bench Connections

7 Software components

The major software components are:

- PPC_Mon monitor and debugger running on RIO2.
- FED_Mon custom test program for FED-PMC running on RIO2.

- VersatermPro terminal emulator for Mac.
- LabView application FEDPMC.vi running on Mac.

- Lattice CPLD loading program running on PC.

8 Getting Started with fedpmc.vi

This section gives a simple step by step guide on how to set up to run LabView application FEDPMC.vi for data acquisition from the FED-PMC. For details of the programs involved see the relevant sections.

The LabView application is designed to provide a (relatively) user friendly means of controlling the FED-PMC and for visualising the captured data.

HINT: Before running the LabView application you can verify the basic functionality of the FED-PMC's and external clocks and triggers just using the FED_Mon program. See section 16.

IMPORTANT:
The following procedure assumes the FED-PMC Flash EEPROM is already loaded with the FPGA design: version = 0 ; revision = 2.
For instructions on how to bootstrap load the Flash EEPROM see section 17.

First ensure that at least one of the PMC slots of the RIO2 contains a FED-PMC.

• Power on the Macintosh (the boot up process takes a couple of minutes).

• At the end of the boot up process of the Macintosh the application **VXIinit** is run automatically (section 10.2.1) This initialises the PCI-MXI-2 card inside the Mac.
Choose Quit from the File menu to close the VXIinit application.

• Start the terminal emulator by opening the icon labelled "VersaTerm PRO" on the Macintosh desktop (e.g. by double clicking on it).

• Power on the VME crate (ensure the associated fan is on to cool the RIO2).

The output of the RIO2 boot process should appear on the VersaTerm screen.
The RIO2 should start up in the CES monitor program PPC_Mon (see section 12).

NOTE: The red LED on the FED-PMC (top right of Lattice CPLD) should come ON indicating power is reaching the PMC.

• Run the FED_Mon program (section 13).

FED_Mon must be run to set up the mapping from VME to PCI in order to allow access to the registers of the FED-PMC.

To run FED_Mon simply execute the PPC_Mon "boot" command (a copy of FED_Mon is stored inside the RIO2 Flash memory):

```
PPC_Mon>boot
```

When asked for the PMC slot e.g. answer **2** if the FED-PMC to test is on the lower PMC slot.

After a page or so of printout the following prompt should appear:

```
FED_Mon>
```

NOTE: At this point the red LED on the FED-PMC should have gone OFF confirming that the FPGA was successfully loaded.

Verify that the FPGA firmware is version 0 ; revision 2.

To change the FPGA firmware see section 17.

After this set up the FED-PMC can now be completely controlled from the LabView application (i.e. you can forget about FED_Mon until you power off or want to control a second FED-PMC in the other PMC slot).

• On the Macintosh run the **resman** program (section 10.2.2). This program initialises the NI/VXI card in the VME crate.
NOTE: This must be run each time the VME crate is powered in order for the LabView application to access the FED-PMC (c.f. **VXIinit** which only needs to be run once after powering on the Macintosh). After running the program you can quit out from it.

Now you are finally ready to run the LabView application **FEDPMC.vi** (see section 15).

• Open the LabView application by opening the icon **FEDPMC.vi** on the Mac desktop.

Labview should indicate it is loading the necessary libraries and then the main screen should appear.

• Run the application to do a data taking run (section 15).

9 Macintosh parameters

Model: Power Macintosh 8500/120

IP Address : 130.246.18.87
Subnet mask: 255.255.252.0
Router address: 130.246.16.254

Macintosh name: Aberdeen

Boot disk name: “Macintosh HD”

10 NI MXI PCI-VME link

10.1 Hardware

To connect the Power Macintosh to the VME crate the National Instruments MXI PCI-VME link is employed. It consists of:

A PXI-PCI-2 card in any of the Power Macintosh PCI slots.
A VME-MXI-2 card in left most slot (i.e. slot 1) in VME crate (Nb VME-MXI-2 auto configures as arbiter when located in slot 1)
A connecting MXI cable.

10.2 Setting up the Link

Two programs must be run in order to set up both ends of the MXI link.

10.2.1 Running VXIinit application

After each power on of the **Mac** the application **VXIinit** must be run to initialise the PCI-MXI-2 card inside the Mac.

NOTE: Normally this application is started automatically at the end of the Macintosh boot process.
(an alias of the application is stored in “Macintosh HD:System Folder:Startup Items” folder)
(However, if you need to run it manually just open the icon “VXIinit” on the desktop.)

The output of the application appears in a window called “VXIinit.out” and should look like the following.

```
-----------------------
PCI-MXI-2 Settings

Device Configuration
  Your PCI-MXI-2 is using the following regions of address space
  Driver Window:  Location - 0x80810000 Size - 0x8000
  User Window:    Location - 0x81000000 Size - 0x1000000
  Your PCI-MXI-2 has been assigned interrupt line 0
Logical Address Configuration
  Logical Address        : 0
  Device Type            : Message-Based
```

```
  Mem Space              : A16 Only
  Slave Pool Size        : 0x20000 bytes
VXI/MXIbus Configuration
  MXIbus Sys Controller  : Yes
  MXIbus Fair Requester  : Yes
  Sys Controller Timeout : 1000 usec

Your PCI-MXI-2 has been initialized successfully.
-----------------------
```

Choose Quit from the File menu to close VXIinit.

10.2.2 Running resman application

After each power on of the **VME crate** the program **resman** must be run to initialise the VME-MXI-2 card.

NOTE: After powering on the VME crate the red LED on the VME-MXI card labelled SYSFAIL appears for a few seconds and then goes off.
The resman application must be run manually.

HINT: One of the commonest mistakes is to power on&off the VME crate and then forget to do this step. The LabView application will later hang when it tries to access the VME crate!)

Open the icon “resman” on the desktop.

The output of the application appears in a window labelled “Resource Manager (c) National Instruments” and should look like the following:

```
-----------------------
>>>>>>>>> Resource Manager Operations Begun <<<<<<<<<

Resource Manager is a Nat'l Insts PCI-MXI-2 "ROOT" at Logical Address (LA)   0.

IDENTIFYING VXI/VME/MXIbus DEVICES:
Waiting (1 sec's) for SYSFAIL* to be removed from the backplane...done.
Configuring Extender   0:
  SC Message  Based device "ROOT" found at LA   0.
  SC Mainframe Extender device "VME-MXI-2" found at LA   1.
  Verifying Self Tests:
    LA   0 Passed Self-Test.
    LA   1 Passed Self-Test.
Configuring Mainframe   1:
          ****************************************************
          * WARNING:                                         *
          * There is no VXI Slot 0 device in this frame.     *
          * Assuming this is a VME frame.                    *
          ****************************************************
  SC Mainframe Extender device "VME-MXI-2" found at LA   1.
  Non-VXI (VME) device "CES_FIC" assigned at Pseudo-LA 258.
  Non-VXI (VME) device "CES_RIO" assigned at Pseudo-LA 257.
  Non-VXI (VME) device "CMSFED" assigned at Pseudo-LA 260.
  Verifying Self Tests:
          ****************************************************

CONFIGURING ADDRESS MAP:
A16 Address Map for Non-VXI (VME) devices:
  CES_FIC (Pseudo-LA 258) requests no memory.
  CES_RIO (Pseudo-LA 257) requests no memory.
  CMSFED (Pseudo-LA 260) requests no memory.
```

```
A24 Address Map:
  CES_FIC (Pseudo-LA 258) requests 0x100000 bytes at 0x600000.
  CES_RIO (Pseudo-LA 257) requests 0x10000 bytes at 0x850000.
  CMSFED (Pseudo-LA 260) requests 0x200000 bytes at 0x400000.
  ROOT (LA   0) requests no memory.
  VME-MXI-2 (LA   1) requests 0x4000 bytes (allocated at 0x700000).
A32 Address Map:
  CES_FIC (Pseudo-LA 258) requests no memory.
  CES_RIO (Pseudo-LA 257) requests 0x1000000 bytes at 0x8000000.
  CMSFED (Pseudo-LA 260) requests no memory.
  ROOT (LA   0) requests no memory.
  VME-MXI-2 (LA   1) requests no memory.
         ************************************************

CONFIGURING COMMANDER/SERVANT HIERARCHY:
Finding 'Commander' Message Based Devices:
Initial Commander/Servant Hierarchy is as follows:
  Resource Manager   ROOT (LA   0) is highest commander.
  Granting commander ROOT (LA   0) servant VME-MXI-2 (LA   1).
Known Servant Lists are as follows:
  ROOT (LA   0) has servants:
     VME-MXI-2 (LA   1)
  VME-MXI-2 (LA   1) has servants: none (It is a Mainframe Extender Device).
         ************************************************

ALLOCATING VXI/VME IRQ LINES:
Finding out protocols supported by Message Based Servants:
  No Immediate Message Based Servants that have PASSED self-tests.
  Mainframe 1:
    Assigning Interrupt lines to Static Non-VXI (VME) Handlers:
      CES_FIC (Pseudo-LA 258), assigned interrupt handlers : none.
      CES_RIO (Pseudo-LA 257), assigned interrupt handlers : none.
      CMSFED (Pseudo-LA 260), assigned interrupt handlers : none.
    Assigning Interrupt lines to Programmable VXI Handlers:
      ROOT (LA   0), PH #1, allocated interrupt line 7.
      ROOT (LA   0), PH #2, allocated interrupt line 6.
      ROOT (LA   0), PH #3, allocated interrupt line 5.
      ROOT (LA   0), PH #4, allocated interrupt line 4.
      ROOT (LA   0), PH #5, allocated interrupt line 3.
    Assigning Interrupt lines to Static Non-VXI (VME) Interrupters:
      CES_FIC (Pseudo-LA 258), assigned interrupt lines : none.
      CES_RIO (Pseudo-LA 257), assigned interrupt lines : none.
      CMSFED (Pseudo-LA 260), assigned interrupt lines : none.
    Assigning Interrupt lines to Programmable VXI Interrupters:
      No Programmable VXI Interrupters found.
         ************************************************

INITIATING NORMAL OPERATION:
  Done.
         ************************************************

         >>>>>>> Resource Manager Operations Completed <<<<<<<
-----------------------
```

NOTE: The green LED labelled MXI on the VME-MXI-2 card should flash a couple of times.

Choose Quit from the File menu to close the resman application.

Now the link between the Mac and the VME crate is ready to use.

10.2.3 Other comments

In fact the rule is that VXIinit and resman must both be run at least once between powering on the Mac and VME crate and running the LabView application for the first time. It doesn't hurt to run these applications more than once or in a different order as long as you follow the previous rule.

The original files for the NI-VXI system are located in the folder "Macintosh HD:LabView 4.0:NI-VXI:".
FOR EXPERTS: the application used to edit the VXI configuration files is also located there.

11 "VersaTerm Pro" Terminal emulator

The Macintosh "**VersaTerm Pro**" application allows the Mac to act as a dumb terminal for the RIO2. The program is located on the folder : "Macintosh HD:VersaTerm Pro:"
An alias to the program is located on the desktop labelled "VersaTerm Pro".
To run the program either double click on the icon or select it and choose Open from the File menu.

11.1 Troubleshooting

If there are problems getting terminal output from the RIO2 on the Macintosh try the following:

In the "Sessions" Menu ensure "Modem" is selected. (other sessions can be set to printer port or Telnet links)

Quit and restart the VersatermPro program.

Reset the RIO2.

Check the serial cable from the RIO2 is connected to the Modem port on the rear of the Macintosh.
(alternatively the Printer port can be used if it is selected in Sessions menu)

Check other VersaTerm menu settings:

In the "Sessions" Menu ensure "Modem" is selected. (other sessions can be set to printer port or Telnet links)

In the "Settings" Menu check the Baud parameters are:
- Baud = 9,600
- Data bits = 8
- Parity = none
- Stop bits = 1.0

In the "Emulation" Menu check that "DEC VT220" is selected

12 Booting RIO2 and PPC_Mon

After powering on or resetting the RIO2 the output from the boot sequence should look like the following: NOTE: The exact details of the terminal listings in the following sections may vary slightly depending on the actual version of PPC_Mon that is being used.

```
-----------------------
PPC_Mon>

PPC Boot Rev 3.42 created Mon Feb  1 13:45:20 1999

  Module Type: 8060BA  rev: C2  serial: 182
  Host CPU: PPC603   ver: 3.02   speed: 66 Mhz
  PCI Bridge: IBM27-82660 rev: 01
  Memory Size: 8+0+0+0+0+0+0+0 = 8 Mbytes
               STD RAM  60 ns   ECC enabled
  L2 CACHE:  not present
  FPROM: One bank of AMD29F016 installed
  Ethernet Address: 00:80:a2:00:80:c6

Entering boot diagnostics

 0    Check System Memory (0x00000000 - 0x006ffffc)          0  SKIPPED
 1    Check Interrupt Handler
*** Decrementer ***
0  OK
 2    Check PCI Bridge                                       0  OK
 3    Check Cache and MMU                                    1  OK
 4    Check MK48T08 RTC and NVRAM                            1  OK
 5    Check SIC6351 Interrupt Controller                     1  OK
 6    Check PC87312 Super IO                                 1  OK
 7    Check ZCIO8536 Micro Timers                            1  OK
 8    Check FIFO's (0 - 7)                                   1  OK
 9    Check DS1620 Digital Thermometer                       0  OK
10    Check PCI devices                                      0  OK
11    Check AM79C970 Ethernet Controller  (PCI slot 0)       0  OK
12    Check PMC #1 Extension  (PCI slot 2)                   0  OK
13    Check PMC #2 Extension  (PCI slot 1)                   0  NO DEVICE
14    Check VME Interface                                    0  OK

************************************************
*   PPC_Mon RTPC8060 monitor - version 3.4       *
*     CES SA Copyright 1995,1996,1997,1998       *
************************************************

PPC_Mon>
-----------------------
```

The RIO2 is now sitting in the standard CES PPC_Mon monitor program [section 12].

13 Running FED_Mon

FED_Mon is a low level debugger & monitor program [2] specially written for testing the FED-PMC.

FED_Mon comprises of a PowerPC XCOFF file which can be run on the RIO2 in a couple of ways described in the following sections.

NOTE: The exact details of the terminal listings in the following sections may vary slightly depending on the actual version of FED_Mon (and PPC_Mon) that are being used.

13.1 Running FED_Mon as the Boot program

A copy of the standard FED_Mon program is normally stored in the Flash memory of the RIO2.

To start FED_Mon type enter the "boot" command in PPC_Mon:

```
-----------------------
PPC_Mon>boot
loading from FPROM 0x00100000
at address:600000
xcoff file
starting code execution at address: 4b1c8

FEDMON Version of Jan 16 2001 at 10:40:54

* WARNING => RUNNING FEDMON TEST VERSION.
PPC regs: msr  = 0x00001030; hid0 = 0x81000000;
INFO => DOING VME SETUP...
FED PMC @ VME A32 base ; len = 2 MB
SYS MEM start @ VME A32 base + $200000 ; len = 4 MB
RIO A24 base for CSR           = $xx400000
RIO A32 base for System Memory = $08000000

QUESTION => Which PMC Slot is FED in ? [1 (upper) / 2 (lower = default = CR)] =>
2
INFO => OK Using PMC Slot 2...
* WARNING => PCI CFG is using RIO2 MAPPING.
fedmem: pci_cfg_base = $80802000
fedmem: pci_mem_base_abs = $01200000
fedmem: pci_device_number = $03

bridge config regs @ pci cfg base = $80802000

            vendor id = $fed010dc (PCI9080_PCIIDR)
bridge cmd/status reg = $00000006 (PCI9080_PCICR)
     dpm pci mem base = $01200000 (PCI9080_PCIBAR2)
reg/fifo pci mem base = $01300000 (PCI9080_PCIBAR3)
  bridge pci mem base = $01300100 (PCI9080_PCIBAR0)
fedbrg: default values were set.

bridge local regs: @ pci mem base = $c1300100

      dpm local base = $00000001 (PCI9080_LAS0BA)
     dpm window size = $fff00000 (PCI9080_LAS0RR)
```

```
      dpm descriptor = $030300c3 (PCI9080_LBRD0)
     regs local base = $00100001 (PCI9080_LAS1BA)
    regs window size = $ffffff00 (PCI9080_LAS1RR)
     regs descriptor = $00000243 (PCI9080_LBRD1)
  mode & arbitration = $01200000 (PCI9080_MARBR)
       eeprom cntrl = $98000070 (PCI9080_CNTRL)
         big endian = $00000084 (PCI9080_BIGEND)

INFO => Xilinx FPGA status at start:
fedxlst: Xilinx Load status : INIT = 1, DONE = 0
INFO => Loading Xilinx from Flash EEPROM (takes approx 5 secs)...
INFO => End Xilinx load from Flash after 4559 msecs
fedxlst: Xilinx Load status : INIT = 1, DONE = 1
fedbrg: default values were set.

bridge local regs: @ pci mem base = $c1300100

      dpm local base = $00000001 (PCI9080_LAS0BA)
     dpm window size = $fff00000 (PCI9080_LAS0RR)
      dpm descriptor = $030300c3 (PCI9080_LBRD0)
     regs local base = $00100001 (PCI9080_LAS1BA)
    regs window size = $ffffff00 (PCI9080_LAS1RR)
     regs descriptor = $00000243 (PCI9080_LBRD1)
  mode & arbitration = $01200000 (PCI9080_MARBR)
       eeprom cntrl = $98020070 (PCI9080_CNTRL)
         big endian = $00000084 (PCI9080_BIGEND)

INFO => Setting up FED specific registers with defaults...
ATTENTION => If we stop here the Xilinx is NOT Loaded.

fedinit: using default values.
fedinit: WARNING => If Bus errors here, the CLOCK may NOT be running.
fedinit: WARNING => If we stop here the Xilinx may NOT be loaded.

fedstat: fed register status only:-

fedstat: ATTENTION => If we stop here the Xilinx is NOT Loaded.
fedstat: reset default bridge configuration.
fedstat: reset default bridge configuration.
 fed regs: @ pci mem base = $c1300080

 Digitisation is >> DISABLED <<

    Test mode is >> DISABLED <<

     Throttle is >> DISABLED <<
 throttle threshold = $000000ab

   Clock source = 0
   Clock source => PCI CLOCK
    Clock delay = 0

 Trigger source = 0
 Trigger source => FRONT PANEL TRIGGER
   Trigger mode = 0
   Trigger mode => START DIGITISATION

 adc_chan_mask = $aa
 ADC chans 0&1 are << ENABLED >>
```

```
           2&3 are << ENABLED >>
           4&5 are << ENABLED >>
           6&7 are << ENABLED >>
 ADC Sample Freq = 0
 ADC Sample Size = 16

 Number of Filled Buffers = 0

INFO => The FED PMC is Initialized and ready to take data.
INFO => FED Serial Number = 004 (dec)
fedrev:  Version information :

FIRMWARE:
Xilinx : version = 0
       : revison = 2
       : prototype = 0

SOFTWARE:
FED_Mon Version of Jan 16 2001 at 12:01:13
FEDPMC s/w library version = '2.09'
INFO => Type 'fed' for a full list of FED_Mon commands.
FED_Mon>
-----------------------
```

FED_Mon does the following actions when it starts up:

Initialises the FED-PMC PCI bridge chip.
Programs the Xilinx FPGA from the Flash EEPROM on the FED-PMC.
Sets up the VME windows on the RIO2 to allow the Mac to access the FED_PMC.

NOTE: The FED-PMC serial number and Firmware versions are reported near the end of the above listing. The FED-PMC starts up by default using the internal PCI clock.

If FED_Mon fails to boot check the boot parameters of the RIO2 (see section 13.3).

The FPGA takes about 5 seconds to load. If this step fails the program will terminate.
To reload the FPGA see section 17.

NOTE: To return from FED_Mon to PPC_Mon (e.g. in order to check network or boot parameters) enter the "reset" command.

13.2 Running FED_Mon from the Network

The FED_Mon program can also be down loaded from the network from the tftp server running on the Mac. The following command loads FED_Mon into the RIO2 system memory (i.e. temporary memory).
IMPORTANT: The download address must be set to $10000 (FED_Mon is relocated to execute at this address).

```
-----------------------
PPC_Mon>load le fedmon 10000
ethernet load
local address = 00:80:a2:00:80:c6
using ARPA
file server address 00:a0:40:24:4f:31
loading file 'fedmon' from server 130.246.18.87
tftp packet number:      26d
transfer rate: 27 kbyte/sec
PPC_Mon>
-----------------------
```

FED_Mon is then run using the "exec" command:

```
-----------------------
PPC_Mon>exec  10000
starting code execution at address: 4b1c8

FEDMON Version of Jan 16 2001 at 10:40:54

* WARNING => RUNNING FEDMON TEST VERSION.
PPC regs: msr  = 0x00001030; hid0 = 0x81000000;
INFO => DOING VME SETUP...
FED PMC @ VME A32 base ; len = 2 MB
SYS MEM start @ VME A32 base + $200000 ; len = 4 MB

...etc  same output as in listing of section 13.1.

-----------------------
```

HINT: If the network load fails:

• Try it again.

• Check red LED labelled LRx on RIO2 is flashing indicating network activity.

• Check the RIO2 network parameters in PPC_Mon.
The network parameters should be as follows:

```
-----------------------
PPC_Mon>show  inet

INTERNET parameters

   inet_host [dotted decimal] : 130.246.17.229
   inet_bootserver [dotted decimal] : 130.246.18.87
   inet_gateway [dotted decimal] : 130.246.16.254
   inet_nameserver [dotted decimal] : 130.246.8.13
   inet_protocol [ arpa bootp ] : arpa
   inet_mount :
PPC_Mon>
-----------------------
```

The inet_host is the RIO2 address.
The inet_bootserver is the Mac address (for the tftp server).

Refer to the PPC_Mon documentation for an explanation of these settings.
These settings can be changed using the PPC_Mon “set” command.

• Check the FED_Mon program file is in the tftp server folder on the Mac (section 14). The server should report a file not found error in this case.
• Check the tftp server is running on the Mac (section 14).

• Check the Macintosh network connection is working (e.g. try running a Web browser).

13.3 RIO2 boot parameters for FED_Mon.

To enable FED_Mon to load on boot the boot parameters of the RIO2 should be as follows:

```
-----------------------
PPC_Mon>show  boot
```

```
BOOT parameters

   boot_flags [afmnN] : a
   boot_device [sd<0-7><a-d>, le, fp] : fp
   boot_file : 100000
   boot_rootfs [sd<0-7><a-d>, rd] :
   boot_delay : 3 sec
   boot_address : 10000
   boot_size : 0
PPC_Mon>
-----------------------
```

Refer to the PPC_Mon documentation for an explanation of these settings.
These settings can be changed using the PPC_Mon "set" command.

In addition the boot environment parameter should be selected to value [10] to stop the RIO2 boot process in the PPC_Mon monitor program. This can be checked by hitting CR in response to the following command:

```
-----------------------
PPC_Mon> setenv boot
Host boot mode [10] : <CR>
PPC_Mon>
-----------------------
```

13.4 Storing FED_Mon in RIO2 Flash memory.

The following command downloads the program fedmon on the tftp server into permanent storage in the RIO2 Flash memory.

WARNING: This procedure should only be done by experts. The CES boot software is also stored in this memory. Corrupting the Flash may damage the RIO2.
In particular, this command must **NOT** be used on the CES RTPC8067.

```
-----------------------
PPC_Mon>fprom load 100000 le fedmon 100000
ethernet load
local address = 00:80:a2:00:80:c6
using ARPA
file server address 00:a0:40:24:4f:31
loading file 'fedmon' from server 130.246.18.87
tftp packet number:      26d
transfer rate: 39 kbyte/sec
fprom_off = 100000   mem_addr = 100000  len = 4d8dc
 Sector 0x00100000 : Mapped 0x81200000 : Base 0x81200000 : Size 0x20000
 Sector 0x00120000 : Mapped 0x81220000 : Base 0x81220000 : Size 0x20000
 Sector 0x00140000 : Mapped 0x81240000 : Base 0x81240000 : Size 0x20000
loading at address : 0x8124d800 Got "Success!" from Eprom_Write
PPC_Mon>
-----------------------
```

13.5 RIO2 boot parameters for FED_Mon.

To enable FED_Mon to load on boot the boot parameters of the RIO2 should be as follows:

```
PPC_Mon>show boot

BOOT parameters
```

```
   boot_flags [afmnN] : a
   boot_device [sd<0-7><a-d>, le, fp] : fp
   boot_file : 100000
   boot_rootfs [sd<0-7><a-d>, rd] :
   boot_delay : 3 sec
   boot_address : 10000
   boot_size : 0
PPC_Mon>
```

Refer to the PPC_Mon documentation for an explanation of these settings.
These settings can be changed using the PPC_Mon “set” command.

In addition the boot environment parameter should be selected to stop the RIO2 boot in the PPC_Mon monitor program. This can be checked by the co:

```
PPC_Mon> setenv boot
Host boot mode [10] :
PPC_Mon>
```

14 TFTPd file server

The RIO2 monitor program uses the TFTP (“Trivial FTP”) protocol to download files from the network.

A TFTP server application **TFTPd** is installed on the Mac. This application is launched when the Mac boots up and runs as a “background application” (i.e. you can’t see it running on the desktop).
A copy of TFTPd is stored in the “Macintosh HD:System Folder: Startup Items:” folder.

IMPORTANT: All files to be downloaded to the RIO2 MUST be located in the folder “Macintosh HD:System Folder:Preferences:TFTPd:”. For convenience, an alias to this folder is kept on the desktop labelled “TFTPd *f*” (i.e. FED_Mon program and FPGA files are stored here).

If you really want to verify that **TFTPd** is active you need to run the application **Peek-a-Boo** which can be found in the “Macintosh HD: Utilities:” folder.
You should see TFTPd in the list of running process’s.
If it isn’t in the list start TFTPd by opening it’s icon (see above).

WARNING: Be careful when running Peek-a-Boo it gives you control to kill all running process’s on the Mac! Choose Quit from the File menu to close Peek-a-Boo.

15 Running the LabView application FEDPMC.vi

The LabView application **FEDPMC.vi** is used to control and readout the FED-PMC.

The application and it's associated files are located in the folder:
"Macintosh HD:FED-PMC:LabView:FEDPMC:".

The following shared libraries must be located in the "Macintosh HD:System Folder:Extensions:" folder:

fedpmc.shlb
fedvxi.shlb

These libraries contain the low level C functions which access the FED-PMC registers.

Ensure all the steps in section 8 have been carried out.

• Start the application by opening the alias icon **FEDPMC.vi** . on the desktop.
After a short while LabView v4.01 will start up and some messages indicating libraries are loading are displayed and finally the Main application window will appear.

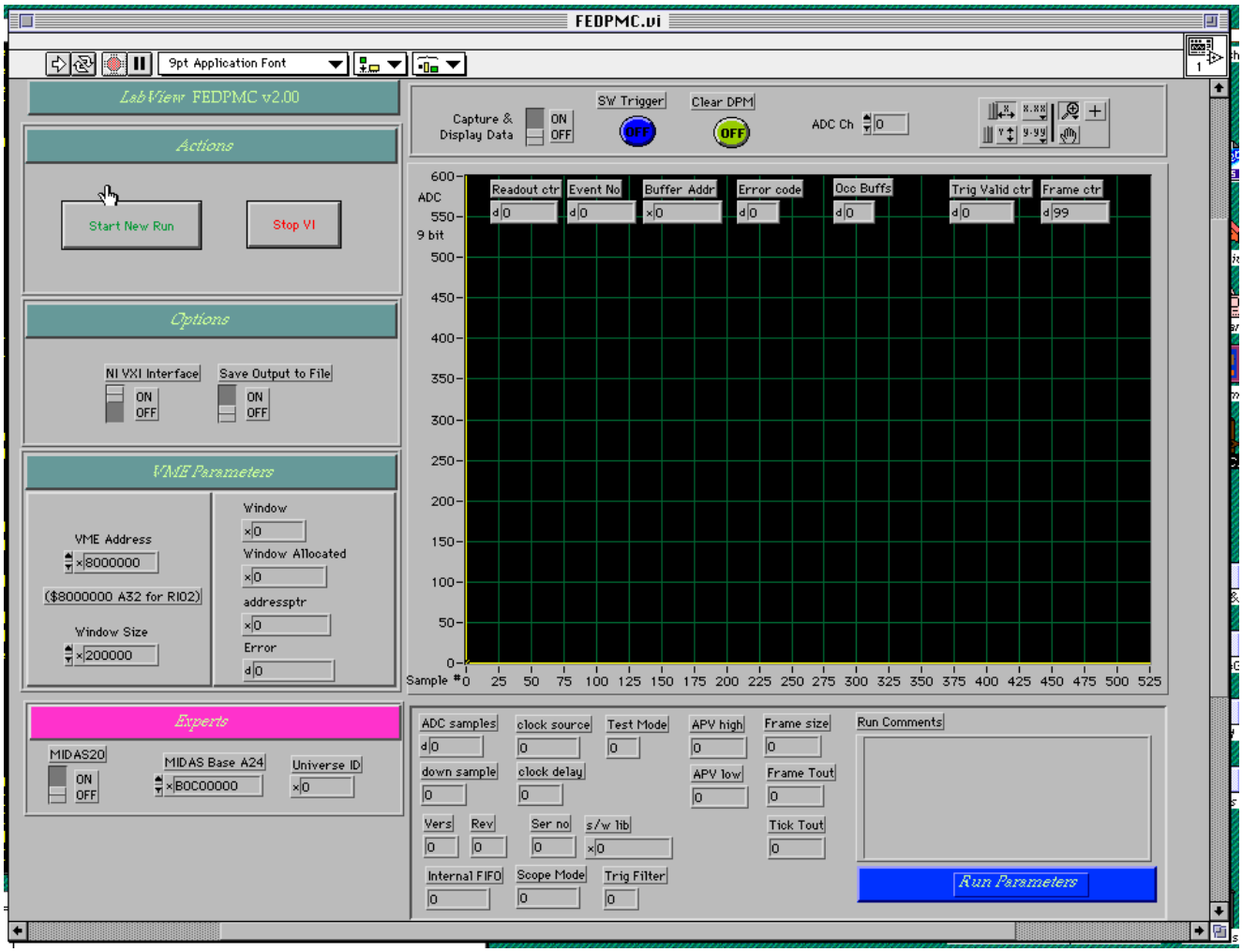

Figure 4 FEDPMC.vi Main Window

Ensure these panel options are set as follows (normally the default values):

NI/VXI Interface = ON
Save Output to File = OFF
VME Address = $8000000
Window Size = $200000
MIDAS20 = OFF

• Run the application by choosing “Run” from the “Operate” menu.

The VME Parameters panel should now display:

Window = 4
Window Allocated = $1FFFFF
Addressptr = $81000000
Error = 0

• Start a data taking run by pressing the “Start New Run” button.
The Run Settings panel will appear (Figure 5).

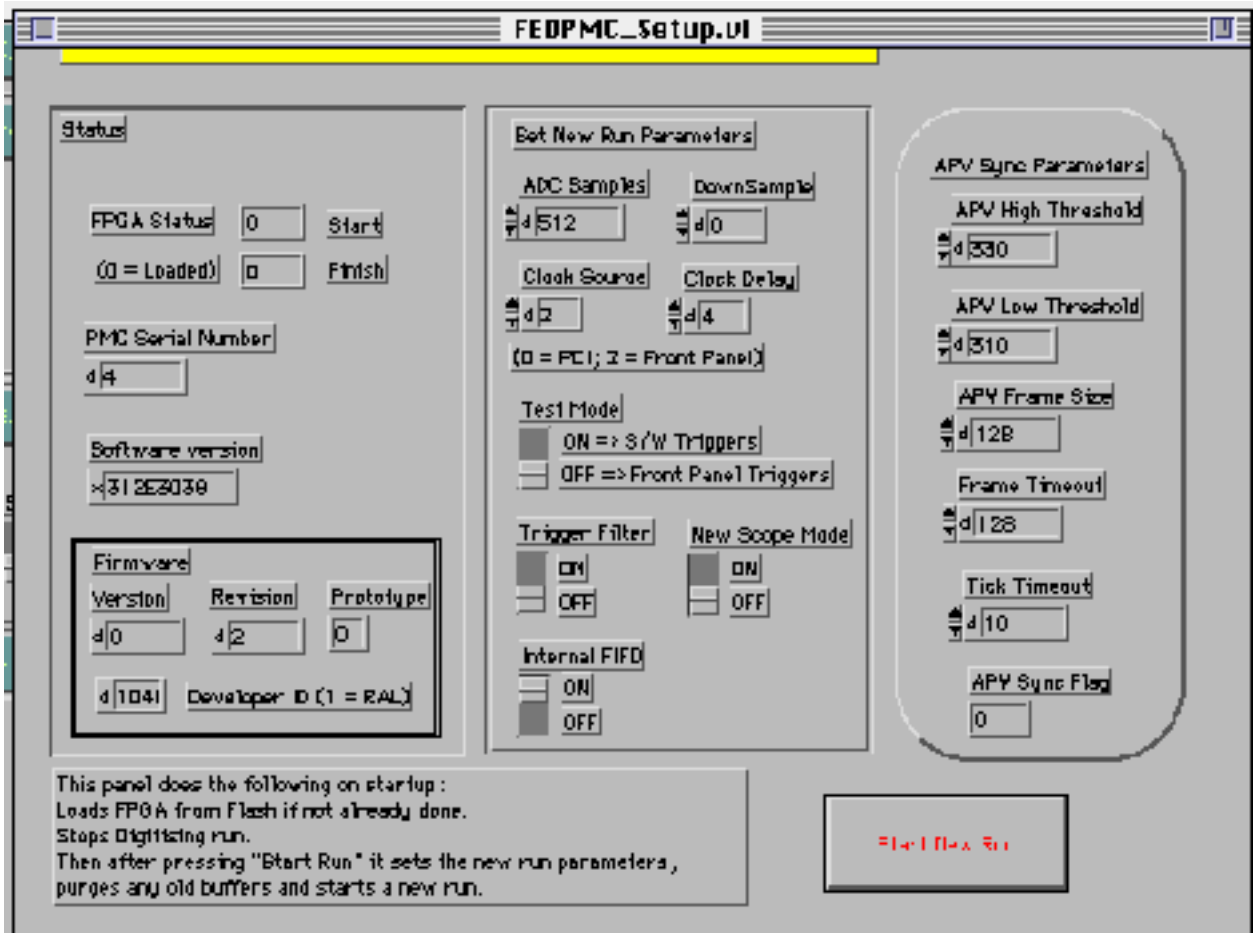

Figure 5 FEDPMC.vi Run Settings panel

The Status panel should indicate the serial number.
The Xilinx FPGA Firmware loaded should be version = 0 , revision = 2.

HINT: If the values here are nonsense it means there is a problem and the application cannot access the FED-PMC. Check the steps in section 8.

• Set up the settings for the Run. For an explanation of these parameters see [1].

For example to take a run capturing 512 samples following the trigger and using the Internal PCI clock (with a skew of 4 units) and software triggers and sampling on every clock set the following:

Clock Source = 0
Clock Delay = 4
Test Mode = ON
ADC Samples = 512
Downscale = 0

NOTE: The other settings are only used with Header Finding firmware (version 2) and can be ignored.

• Press the button “Start New Run” (on this panel) and you will be returned to the Main window.

• Set button “Capture & Display Data” ON

NOTE: The green LED labelled “MXI” on the VME-MXI-2 card and the red LED labelled “VSL” on the RIO2 should both come on continuously indicating the Mac is polling for data in the FED-PMC memory.

• Select the ADC channel number (0-7) on which you want to view data using the control “ADC Ch”.

• Send a trigger by pressing the button labelled “SW Trigger”

The data captured in the ADC channel selected should now appear in the window (Figure 6).

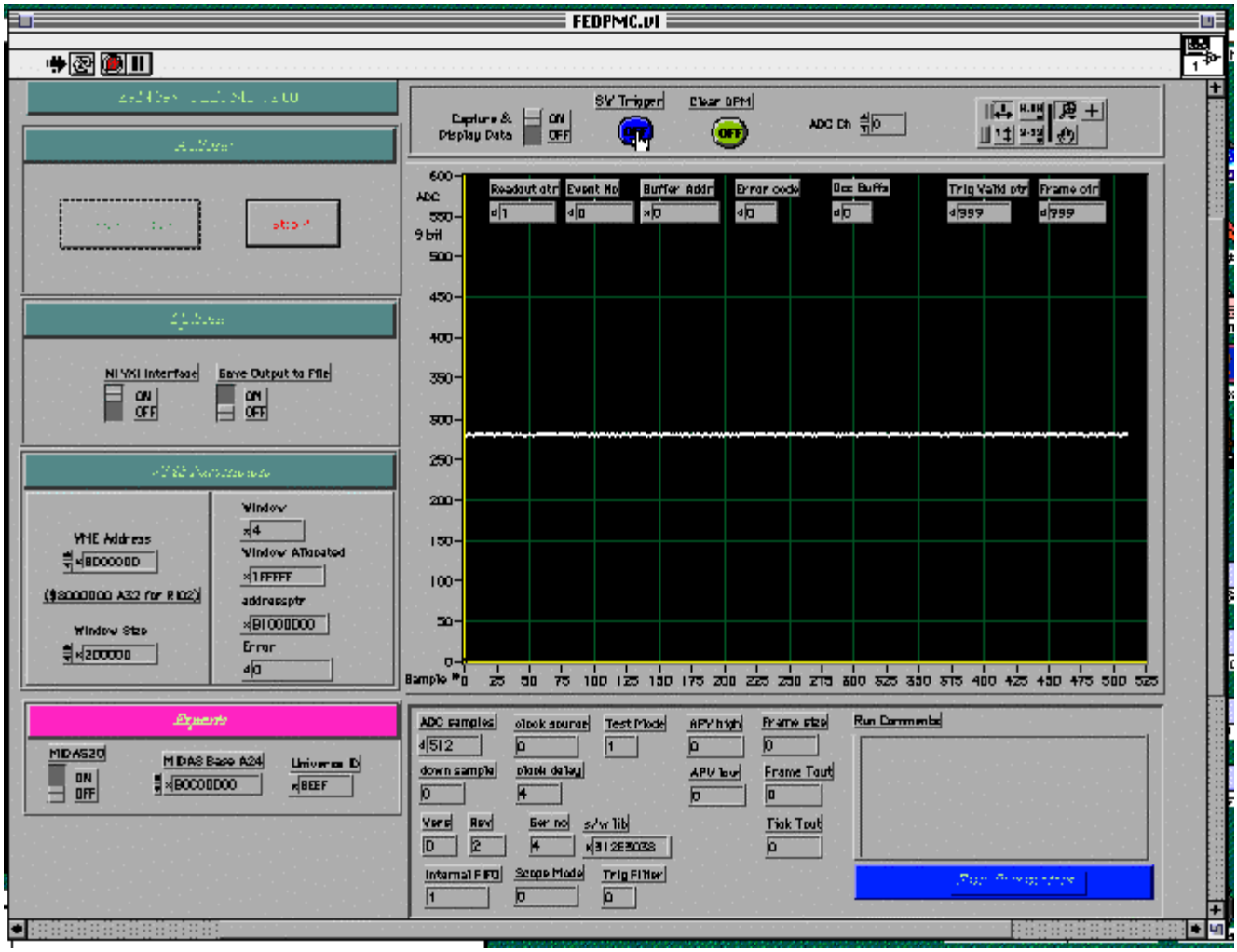

Figure 6 FEDPMC.vi Example of Data Capture

The indicators in the display should behave as follows:

readout ctr : starts as 1 for 1st trigger and increments by one after each trigger
event No : should always be one less than readout ctr
Buffer Addr : 0 for 1st event and then should increment by fixed amount after each trigger (depends on capture length).
Error code = 0
Occ Buffer : After each trigger should go to 1 and return immediately to 0
Trig Valid ctr & Frame ctr : for header finding only, ignore.

• To view data on another ADC channel change control "ADC Ch" and send another trigger.
NOTE: you can't see the data from more than one ADC channel for a given trigger.

• To change the run parameters you must Stop the run by pressing "Stop VI" button.
Then run the application again by choosing "Run" from the "Operate" menu (see above).

HINT: If there are problems running the application try first checking the basic FED-PMC operation as described in section 16. Try FED_Mon also if you want to check validity of external clock and trigger sources.

HINT: Stop the application before resetting the RIO2 or powering off the VME crate. Otherwise the application will hang.

HINT: If you select external clock or triggers ensure the cable carrying these signals is plugged into the FED-PMC. Otherwise the application will hang when you start capturing data.

16 Checking FED-PMC functionality with FED_Mon

The basic functionality of the FED-PMC can be quickly checked by doing a simple data taking run using FED_Mon only (i.e. without LabView application). This avoids any potential problems caused by the VME link or the application itself.
It is also useful for checking the validity of the external clock and triggers.

Run FED_Mon (section 13).
Firmware v0r2 should be loaded in FPGA.

16.1 Testing with internal clock and software triggers

Take some data using the internal PCI clock and software triggers by entering the following commands (see [2] for details):

```
-----------------------
FED_Mon>fedstart 1
fedstart: FED data acquisition Run is STARTED, using SOFTWARE triggers.
FED_Mon>

FED_Mon>fedswt
fedswt: sent software trigger #1.
fedswt: number of filled buffers = 1.
FED_Mon>

FED_Mon>fedrout
readout time for 16 samples (skip begin/end = 0/0) = 622 micro sec
fedrout: Readout Event data to RIO address $00300000

Event no = $00000000 ; Bunch no = 60844
Event size = 256 bytes; Number Samples = 16; Skipped samples begin/end = 0/0;
Buffer base in dpm = $00000000

Unformatted data follows:

ADC Chan# =>  0    1    2    3    4    5    6    7
Sample #
       000:  0280 0275 0274 0274 0277 0277 0277 0275   (Dec)
       001:  0281 0275 0274 0274 0277 0277 0277 0276   (Dec)
       002:  0281 0275 0274 0274 0277 0277 0277 0276   (Dec)
       003:  0281 0276 0274 0274 0277 0277 0278 0276   (Dec)
       004:  0281 0276 0274 0274 0277 0277 0278 0276   (Dec)
       005:  0281 0275 0274 0274 0277 0277 0277 0276   (Dec)
       006:  0281 0275 0274 0274 0277 0277 0278 0276   (Dec)
       007:  0280 0275 0274 0274 0277 0277 0278 0276   (Dec)
       008:  0280 0275 0274 0274 0277 0277 0277 0275   (Dec)
       009:  0280 0275 0274 0274 0277 0277 0278 0276   (Dec)
       010:  0281 0275 0274 0274 0277 0277 0278 0276   (Dec)
       011:  0280 0275 0274 0274 0277 0277 0278 0276   (Dec)
       012:  0281 0276 0274 0274 0277 0277 0278 0276   (Dec)
       013:  0280 0275 0275 0274 0277 0277 0277 0276   (Dec)
       014:  0280 0275 0274 0274 0277 0277 0277 0276   (Dec)
       015:  0281 0276 0274 0274 0277 0277 0278 0276   (Dec)
```

```
Number of events pending = 0
FED_Mon>

FED_Mon>fedstop
fedstop: Event & BX Counters are RESET
fedstop: FED data acquisition Run is STOPPED.
FED_Mon>
-----------------------
```

The contents displayed above are typical when there are no data inputs connected to the front panel lemo connectors. Expect ≈ 1/2x 511 (full scale) with differential inputs.

Testing with external clock and software triggers
Select the external clock.
Make sure the external clock & trigger cable is connected to the FED-PMC and that the clock is running.

```
-----------------------
FED_Mon>fedclk 2 4
fedclk: clock source = PCI CLOCK
fedclk:  clock delay = 1 (x 2.5 nsec)
fedclk: WARNING => If Bus errors after changing, new CLOCK may NOT be running.
FED_Mon>
-----------------------
```

Note: If the external clock is not present the FED-PMC may hang and will need a h/w reset (i.e. RIO2 reset) to recover.

Now do a data run by entering same commands as in section 16.1.

16.2 Testing with external clock and external triggers

Select the external clock as described in section 0.

NOTE: Before starting the run ensure the cable carrying the external clock & trigger is plugged into the FED-PMC. Otherwise the input levels float generating thousands of spurious triggers. This will cause the buffer logic to overflow and the FED-PMC will need a h/w reset to recover.

To take a data run with external triggers use the following commands:

```
-----------------------
FED_Mon>fedstart  0
fedstart: FED data acquisition Run is STARTED, using EXTERNAL triggers.
FED_Mon>
-----------------------

Now send a trigger from your generator.
And read out the triggered data…

-----------------------
FED_Mon>fedrout
readout time for 16 samples (skip begin/end = 0/0) = 622 micro sec
fedrout: Readout Event data to RIO address $00300000

Event no = $00000000 ; Bunch no = 60844
Event size = 256 bytes; Number Samples = 16; Skipped samples begin/end = 0/0;
Buffer base in dpm = $00000000
```

```
Unformatted data follows:

ADC Chan# =>  0    1    2    3    4    5    6    7
Sample #
       000:  0280 0275 0274 0274 0277 0277 0277 0275   (Dec)
       001:  0281 0275 0274 0274 0277 0277 0277 0276   (Dec)
       002:  0281 0275 0274 0274 0277 0277 0277 0276   (Dec)
       003:  0281 0276 0274 0274 0277 0277 0278 0276   (Dec)
       004:  0281 0276 0274 0274 0277 0277 0278 0276   (Dec)
       005:  0281 0275 0274 0274 0277 0277 0277 0276   (Dec)
       006:  0281 0275 0274 0274 0277 0277 0278 0276   (Dec)
       007:  0280 0275 0274 0274 0277 0277 0278 0276   (Dec)
       008:  0280 0275 0274 0274 0277 0277 0277 0275   (Dec)
       009:  0280 0275 0274 0274 0277 0277 0278 0276   (Dec)
       010:  0281 0275 0274 0274 0277 0277 0278 0276   (Dec)
       011:  0280 0275 0274 0274 0277 0277 0278 0276   (Dec)
       012:  0281 0276 0274 0274 0277 0277 0278 0276   (Dec)
       013:  0280 0275 0275 0274 0277 0277 0277 0276   (Dec)
       014:  0280 0275 0274 0274 0277 0277 0277 0276   (Dec)
       015:  0281 0276 0274 0274 0277 0277 0278 0276   (Dec)

Number of events pending = 0
FED_Mon>

FED_Mon>fedstop
fedstop: Event & BX Counters are RESET
fedstop: FED data acquisition Run is STOPPED.
FED_Mon>
-----------------------
```

17 Programming the FED-PMC

This section assumes the FED-PMC has passed all the visual and electrical tests as described in the test procedure instructions located on the ESDG Project FED-PMC web pages

The order of programming of the FED-PMC components is as follows:

1. Program the **Lattice CPLD** as described in section 17.1.

2. Load **Xilinx Flash EPROM** as described in section 17.2.1
The standard Xilinx design is contained in the file : **fpgav0r2.rbt** (Firmware: version 0 ; revision 2)

3. Program the **PLX serial EPROM** as shown in section 17.3.

4. Store the **FED-PMC serial number** in the Flash EPROM as described in section 17.4.

17.1 Programming the Lattice CPLD

The Lattice CPLD is the first device on the FED-PMC that must be programmed (i.e. before Xilinx FPGA is programmed).

Power up the old DAN PC and add the floppy disk named "FEDPMC LATTICE .JED & .DLD".
(cancel if you are asked for a networked drive as this pc is not on the network).

Select Start>Programs>Lattice Semiconductor>ispDCD – this launches the Lattice download program.

With the FED-PMC on the extender and the the power off:
Connect up the special Lattice download cable to the FED-PMC nipples. The connectors are dotted to show position. One dot indicates the LHS e.g. GND, seven dots indicates SDI and DVCC is the red connector.
Power on the FED-PMC (make sure the Fan unit is on to cool the RIO2).

On the PC:
Select File>Open : C:\CPLD_FEDPMC_FILES**ral.dld** – and OK.
(**ATTENTION:** All file names appear mangled because the PC is running an old version of Windows. So in the window the file name appears as “C:\CPLD_F~1” !!)

This script opens a browser window in which you can select the CPLD firmware JEDEC file.
Browse to **cpld_long.jed** (Nb. it looks like cpld_l~1.jed!)

Select Operation **Program & Verify** from the popup menu next to the file name window and then under the main menu bar select Command : Run Operation.
As long as the board has power the Lattice should now be loading.

Once completed change the popup menu selection Operation to **Checksum** and select Run Operation in the main menu bar and the code returned should be **7D28** (Hex) for **CPLD_LONG.JED** (this is the version modified for Mk3 FED_PMC’s to enable long PLX serial eprom loading).

NOTE: If you just want to check the CPLD load without changing it select the file you think should be loaded and choose Operation **Verify**.

Exit to leave the program. (answer No to dialog so as not to change settings).

Power Off the RIO2 and remove the Lattice cable from the PMC (otherwise the Xilinx FPGA loading can be disturbed).

Note: The file **ral.dld** is the default script file and contains the device type to be programmed info and the path of the JEDEC file to program it with e.g. 1032E C C:\CPLD_FEDPMC_FILES**CPLD_LONG.JED** – if a new .jed file is created then ral.dld should to be edited to reflect the new file name.
The file can also be selected by browsing through the files.

17.2 Programming Xilinx FPGA

The Xilinx FPGA is configured from a raw bit file (.rbt) stored in the Flash EPROM.

17.2.1 Programming the Xilinx Flash EPROM for the first time

In order to load the Flash EPROM for the first time a bootstrap procedure must be followed.
Before the Flash EPROM can be accessed the Xilinx FPGA must already be programmed.
This can be achieved by programming the Xilinx FPGA directly from file (i.e. not via Flash) using the FED_Mon command "**fedldxf**"
For further details refer to the FED_Mon manual [2].

Reset the RIO2 to enter PPC_Mon.

Clear a block of RIO2 system memory for the new file (to avoid odd characters at the end of the file).

```
PPC_Mon>fm.w 100000..200000 0
```

Load the new rbt file into this memory block. For example, if the Xilinx file name is fpgav0r2p.rbt.

```
PPC_Mon>load le fpgav0r2p.rbt 100000
```

Start FED_Mon (assuming FED_Mon is stored on the RIO2, see section 13.1).

```
PPC_Mon>boot
```

Once FED_Mon has started program the Xilinx FPGA from the .rbt file.

```
FED_Mon>fedldxf
```

Answer "yes" when prompted to proceed with the programming of the FPGA (or "no" to cancel the operation).

Once the FPGA is programmed the same file can be downloaded into the Flash EPROM.

```
FED_Mon>fedldxfep
```

Answer "yes" when prompted to proceed with the reloading of the Flash EPROM contents (or "no" to cancel the operation).

Power Off and On the RIO2 before rerunning FED_Mon in order to ensure the FED_PMC is running with the new FPGA firmware.

17.2.2 Re-programming the Xilinx Flash EPROM

If the Flash EPROM has **already** been loaded then its contents can be **changed** by using the FED_Mon command "**fedldxfep**" as described below.

Follow the instructions described in section 17.2.1 but skip the "**fedldxf**" operation.
For further details refer to the FED_Mon manual [2].

17.3 Programming the PLX serial eprom

Use the FED_Mon command "**fedplxepfl**" to store the standard values in the PLX serial EPROM.
The stored values can be read back with the "**fedplxeprd**" command.
For details refer to the FED_Mon manual [2].

17.4 Storing the FED-PMC serial number

Use the FED_Mon command "**fedidwr**" to store the FED-PMC serial number (labelled on PCB) in the Flash EPROM.
The stored value can be read back with the "**fedidrd**" command.
NOTE: The FPGA must be programmed before these operations can be carried out.
For details refer to the FED_Mon manual [2].

18 Document Servers

Latest versions of all FED-PMC documentation and software are available from the following web site:

hepwww.rl.ac.uk/CMS_fed/Default.htm

or alternatively directly at the following anonymous ftp server:

address : **ftp.te.rl.ac.uk**
directory: **cms/fed/fed_pmc/**

19 References

[1] FED-PMC User Manual ftp://ftp.te.rl.ac.uk/cms/fed/docs/manuals/fedpmc_um.pdf

[2] FED_Mon User Manual ftp://ftp.te.rl.ac.uk/cms/fed/docs/manuals/fedmon_um.pdf

[3] CES RIO8062 http://www.ces.ch/Products/Products.html

[4] CES PPC_Mon AWX 3317C http://www.ces.ch/Products/Products.html

20 Index

fedpmc.shlb.....23
fedvxi.shlb.....23
Lattice CPLD.....6, 8, 9, 30
TFTP file server.....22