EVENT BUILDER
June 29, 1994

1.0 OVERVIEW OF EVENT BUILDER MODULE 2
1.1 Description of the Event Builder Module 2
1.2 Block Diagram of the Event Builder 3
1.4 Memory Map 10

2.0 TEST PROCEDURE 13
2.1 VME Interface Test 13
2.2 CES Memory Module Interface Test 14
2.3 EBX Interface Test 15
2.4 System Test and Master Readout Module Interface Test. 16

3.0 EVENT BUILDER BOARD1 SCHEMATICS 17

4.0 EVENT BUILDER BOARD2 SCHEMATICS 24

APPENDIX A: Test Software and Utility Programs 29

APPENDIX B: Parts List and Data Sheets 31

1.0 OVERVIEW OF EVENT BUILDER MODULE

This section gives a brief overview of the EB Module (Event Builder) which was designed and developed by Argonne National Laboratory's ECT Department (Electronics and Computing Technology). It is to be used by LBL (Lawrence Berkeley Laboratory) for data acquisition on the GammaSphere Project.

1.1 Description of the Event Builder Module

The EB (Event Builder) is a double wide B size VME Module consisting of two PC Boards. The main function of the Event Builder is to take data from the MRM (Master Readout Module) and to transmit it to one of the CES Memory modules. The MRM sends four serial bit streams to the Event Builder over four fiber optic lines. The Event Builder converts the 4 optical signals into a 32 bit differential ECL bus. The ECL bus is used to transmit data from the Event Builder into one of the CES Memory Modules. The Event Builder controls which CES Memory Module gets the data on the ECL bus. This is done through a number of control signals sent to each CES Memory Module from either the Event Builder Module or the Event Builder Expansion Module.

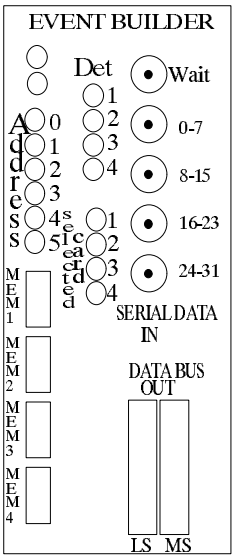
Described below are the LEDs and connectors which can be found on the front panel of the Event Builder.

Address LEDs - Shows which CES Memory Module the Event Builder is addressing. LED on = 1.

Det LEDs - Shows if the carrier signal is detected for the specified fiber optic input.

Selected Card LEDs - Shows which CES Memory Card is selected. Only one card may be selected at a time, and this should correspond to the address LEDs.

MEM1 -> MEM4 Connectors - Connect to the first four CES Memory Modules. Provides the control signals to these modules. The select LEDs indicate which card is selected to have the data transmitted to it.



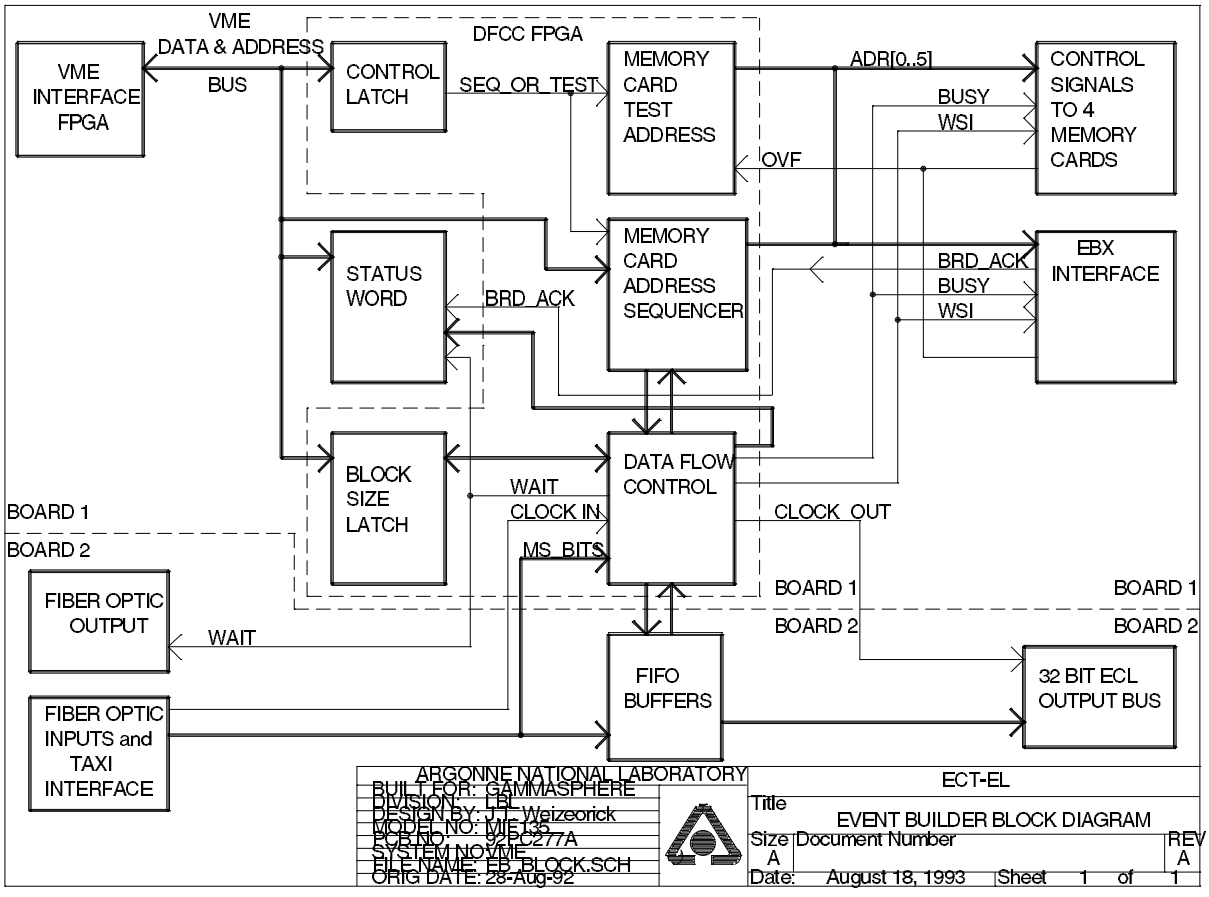
Ribbon Cable Connectors - The two ribbon cable connectors labeled Data Bus Out-LS and MS are for the ECL bus which connects to all of the CES Memory Modules. Note that the last CES Memory Module must be terminated.

Wait Fiber Optic Connector - This Fiber optic connector goes to the Master Readout Module and is used to signal when the memory cards are out of memory. The MRM must then stop sending data to the Event Builder.

Input Fiber Optic Connectors - The four fiber optic inputs are used to receive the four serial streams of data form the MRM. The serial streams of data are combined to form 32 bit words. The bits that each serial streams represent are written beside the connector.

1.2 Block Diagram of the Event Builder

On the following page is a block diagram of the Event Builder. Section 1.3 describes the function of each section shown in the block diagram.



1.3 Subsections of the Event Builder Module

The Event Builder consist of the following subsections listed below.

- VME INTERFACE FPGA
- DFCC (Data Flow Control Chip) FPGA
- FIBER OPTIC INTERFACE
- FIFO BUFFER
- ECL DATA BUS
- ECL CONTROL SIGNALS
- EBX INTERFACE
- MISCELLANEOUS TEST CIRCUITS

The **VME Interface FPGA** allows the VME Bus Master to setup, monitor, and control the Event Builder by reading and writing to memory locations. Details on the VME Interface FPGA can be found in the specifications for the VME Interface FPGA. The memory map for the VME Interface can be found in section 1.4 of this document.

The **DFCC (Data Flow Control Chip) FPGA** is used to control the flow of data through the Event Builder. The steps which occur when transferring an event through the Event Builder are outlined below.

- The data comes from the Master Readout Module over four fiber optic lines.

- The Event Builder stores the data in a set of FIFOs and counts the number of 32 bit words received. The beginning of an event is detected by reading a "10" in the two most significant bits of the 32 bit word. All other data words must start with a "0" in the most significate bit.

- At the end of an event the Event Size Counter value is latch and the DFCC starts clocking the data out of the FIFOs and into the memory modules. The DFCC sticks the event size into the least significant half of the first 32 bit word transmitted to the memory module.
  Note: The end of an event is detected by reading a "11" in the two most significate bits of the 32 bit word.

- Looks for the End of an Event out of the FIFO's to determine when to stop transferring data to the memory module.

- The DFCC also controls which memory board the data is flowing into. The DFCC logic performs the following activities.

  - Monitors the overflow signal from the selected memory module.
  - When an overflow occurs the DFCC sends a WAIT signal to the Master Readout Module and starts looking for a memory module with memory available. The DFCC uses a first available protocol to determine which memory module will be selected next. Note that the overflow signal only becomes active when the busy signal becomes inactive. Since the DFCC keeps the BUSY signal active while clocking out the data from an entire event the overflow can only occur at the end of an event. This forces the data from a signal event to be stored in only one memory module. The first available protocol starts checking for available memory at memory module 0 and continues cycling through the memory modules until it finds one with available memory.
  - The new card will receive the next event which was stored in the Event Builders' FIFO.

The **Fiber Optic Interface** consist of four optical receivers and one transmitter. Each receiver accepts a stream of serial data form the MRM. The bit stream enters the Taxi Chip where it is converts it to a stream of 8 bit words. The 8 bit words from each Taxi Chip is clocked a 8k x 9 **FIFO Buffer**. At the end of an event the DFCC transmits the data from the FIFOs as a stream of 32 bit words. The first word in the event has been modified to contain the number of 32 bit words in the event minus one. The 32 bit words are converted into ECL signals and are sent to the CES memory modules over the **ECL Data Bus**. The fiber optic transmitter sends the WAIT signal from Event Builder to the Master Readout Module. This signal notifies the Master Readout Module that the memory cards have no memory available and that it must wait before sending more data.

Possible Data into Event Builder

Memory Module Address	Bits 31-28	Bits 27-24	Bits 23-20	Bits 19-16		Bits 15-12	Bits 11-8	Bits 7-4	Bits 3-0
0x0	10xx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x4	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x8	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0xc	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x10	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x14	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x18	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x1c	11xx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx

Data out of Event Builder into CES Memory Module

Memory Module Address	Bits 31-28	Bits 27-24	Bits 23-20	Bits 19-16		Bits 15-12	Bits 11-8	Bits 7-4	Bits 3-0
0x0	10xx	xxxx	xxxx	xxxx		0000	0000	0000	0111
0x4	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x8	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0xc	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x10	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x14	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x18	0xxx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx
0x1c	11xx	xxxx	xxxx	xxxx		xxxx	xxxx	xxxx	xxxx

In the tables on the preceding page each character represents a bit. Listed below is the meaning of each character.

x = User programmable by writing to the Channel A DPR in the Master Readout Module or data from the ADC in the GeBGO modules.
1 = Must be 1 to indicate to the Event Builder that this is the beginning or end of an event. When the MS bit of the 32 bit words is a 1 if the next bit is a 0 then it is the beginning of an event, and if the next bit is a 1 then it is the end of an event.
0 = Must be 0 to indicate to the Event Builder that this is not the beginning or end of an event.

The least significant half of the first 32 bit word into the CES Memory Modules contains the event size which equals the number of 32 bit words in the event - 1.

The **ECL Control Signals** are on a separate ribbon cable connection to each CES memory module from the Event Builder or the EBX (Event Builder Expansion) Module. The Event Builder can interface with up to four memory modules. If additional memory modules are needed then EBX Modules must be used. Each EBX Module interfaces with up to four memory modules. The EBX modules must be plugged into the same VME Rack as the Event Builder and the outer two rows of the P2 connector must be bussed between the EBX boards and the Event Builder. This is because the control signals to the EBX boards from the Event Builder come over the VME P2 lines.

Below is a detail description of each control signal transmitted over the ribbon cable to the memory modules.

- BUSY - An output form the Event Builder which tells the memory module when it can notify the Event Builder about a memory overflow. The Event Builder keeps this signal high while data from an event is being stored into a memory module. After the entire event has been stored the Event Builder lowers the Busy line and allows the memory module to drive MEM_OVF if an overflow has occurred.

- VETO - This output from the Event Builder is used to select which memory card the data is to be stored in. When this signal is high the memory card ignores the data on the ECL Bus. When the VETO signal is low the memory card clocks the data into its memory and reports if an overflow occurs when the BUSY signal goes low.

- WSI - This output from the Event Builder is used to clock the data from the ECL Data Bus into the memory module. Data is stored in the memory module on the rising edge of WSI.

- MEM_OVF - This input from the selected memory module indicates when a memory overflows. The MEM_OVF signal can only be active while the busy signal is low. MEM_OVF goes high when a memory overflow has occurred. The programmer can prevent any data from being lost by setting the overflow size in the memory module larger than the biggest possible event.

he **EBX Interface** is done over the P2-A VME connector. The signals over the P2 connector are used by all of the EBX Modules to generate the WSI, BUSY, and VETO signals. The P2 signals also return the MEM_OVF signal from the selected memory module, and provide a BRD_ACK signal. Below is a detail description of all of the signals transmitted over the P2 connector.

- WSI - This is a differential ECL clock signal used to clock data into the memory modules. The EBX modules pass this signal to all of the memory modules and allow the VETO signal to determine which memory module will actual clock in the data.
- BUSY - This is a differential ECL signal who's function has been described above. The EBX modules pass this signal to all of the memory modules and allow the VETO signal to determine which memory module will use this signal.
- AD_MEM (0-5) - These six address lines are used to select one of a possible 64 memory modules to store data in. The EBX module drivers the selected memory module's VETO signal low.
- OVF - This is a differential ECL signal which comes form the selected memory module. The EBX module decodes the address lines and determines if one of its four memory modules are being addressed. If they are the the EBX allows the selected memory module to drive the OVF signal being returned to the Event Builder.
- BRD_ACK - This is an open collector signal which is driven low by the EBX module with the selected memory module. It can be used to determine how many EBX modules are plugged into the system.

Listed below are the **Miscellaneous Test Circuits** which can be found in the Event Builder.

- Status Latch
  Gives information about the state of the Event Builder. For details see address 0x10 and 0x20 in the memory map.
- Read FIFOs
  Hardware which allows the VME Master to read data stored inside the FIFOs. Note that once the data is read out of the FIFOs it will not be transmitted to the memory module when the end of event occurs.
- Write Memory Module
  Hardware which allows the VME Master to write data to the selected Memory Module.
- Control Registers inside the FPGAs
  - Test Mode Bit in Control Bits Register
    While the test mode bit is set VME Master may:
    - read data out of the FIFOs
    - write data into the selected Memory Module
    - write into the AD_MEM register to select a memory module.
  - VME Busy Bit in Control Bits Register
    This bit is used to manual control the Busy signal going to the memory modules while the Test Mode Bit is set.

1.4 Memory Map

The Event Builder has 64 bytes of address space in A16/D16 mode. The absolute VME address will be determined by two dip switches located on the board. The two dip switches are compared to address bits 14 and 15, and bit 12 is forced high. The table below shows how to calculate the Base Address of the Event Builder.

A16 ADDRESSING FORMAT

Base Address

SW15 SW14 0 1	0 0 0 0	0 0 0 0	0 0 0 0

The four possible base address relative to the dip switch positions are listed below.

SW15 SW14	ADDRESS
0 0	0x1000
0 1	0x5000
1 0	0x9000
1 1	0xD000

Relative Address	Name	Function
0x00	AD_MEM	READ/WRITE 5-Bit Latch which determines the address of the selected memory module. Can be read anytime. Can set while in test mode.
0x02	MAX_BRD	READ/WRITE 5-Bit Latch which stores the number of memory modules being used in the system.
0x04	CONTROL BITS	READ/WRITE 3-Bit Latch which controls the following control bits. Bit 0 = TEST_MODE Test mode = 1. While in test mode VME master can Read FIFOs, Write Memory Module, and select a Memory Module. Bit 1 = Not Used Bit 2 = VME-BUSY Manual control of Busy Signal to memory modules while in test mode
0x06	EOE COUNTER	READ - Reads 8-Bit Counter on D0-D7. The counter is incremented every time an EOE is detected coming into the FIFOs. WRITE - Resets Counter.
0x08	RESET FIFOs	READ/WRITE 1-Bit Latch which drive the FIFO resets. 0 = Reset FIFOs 1 = Normal Operation
0x0A	R/W LS HALF OF DATA BUS	READ - While in test mode the LS half of the 32 Bit FIFOs are read. WRITE - While in test mode the data is latched and will be used for the LS half of the 32 Bit Word written to the Memory Card.
0x0C	READ MS HALF OF DATA BUS	READ - Reads the MS half of the 32 Bit FIFOs. WRITE - Writes data to MS half of memory card and writes latched data to LS half of memory card.
0x0E	RESET DFCC	READ - Not Used WRITE - Generates Reset Pulse to DFCC.

Relative Address	Name	Function
0x10	STATUS BITS	READ - Reads 16-Bit Buffer with the following status bits. BIT0-3 = FIFO 1-4 EMPTY FLAGS BIT4-7 = FIFO 1-4 HALF FULL FLAGS BIT8-11 = FIFO 1-4 FIFO FULL FLAGS BIT12 = EOE - End of Event->FIFOs BIT13 = BOE - Beginning of Event->FIFOs BIT14 = VME_OVF BIT15 = BRD_ACK
0x20	STATUS BITS	READ - Reads 3-Bit Buffer with the following status bits. BIT0 = ERROR BIT 1 FROM DFCC BIT1 = ERROR BIT 2 FROM DFCC 00 - No Error 01 - No Data and one EF is not 0 10 - FF4 goes to 1 11 - First Word not BOE (10->ms bits) BIT2 = EOE - End of Event -> FIFOs

2.0 TEST PROCEDURE

The sections below give a step by step procedure used to test out the Event Builder. All of the test procedure require the following cards to be plugged into a VME Create.

SLOT 0	MVME147 Processor Board running VxWorks
SLOT 1-2	CES HSM 8170 Memory Modules
SLOT 3	Event Builder
SLOT 4	Event Builder Expansion Board

When setting up the CES HSM 8170 Memory Modules start their base address at 0x10000000. Also be sure that the termination resistors on the 32-bit ECL inputs are removed on all but the last module. When using the EBX board it must have its outer P2 pins connected to the Event Builder.

The MVME147 board should be running VxWorks and should be a node on a SUN's Ethernet Network. If the VXIcpu-030 board is properly configured on power up it will load VxWorks from the network and execute a start up routine. The start up routine will load all of the programs needed to test the Event Builder and the EBX board.

In the test procedures below, commands which an operator must enter are written in **BOLD**. The computer output is a smaller font and comments are in the normal font. The commands are entered by an operator on a terminal logged into the MVME147 Module. This terminal will be running a C shell under VxWorks.

2.1 VME Interface Test

The following program test the VME Interface in the Event Builder by performing reads and writes to the internal latches on the VME_INF FPGA. For example the following addresses have bits which are R/W if the proper procedure is used.

0x00 - AD_MEM = 5 BIT LATCH
0x02 - MAX_BRD = 5 BIT LATCH
0x04 - CNT_BITS = 3 BIT LATCH

To execute the test enter the following command while logged into the VME network node running VxWorks.

mallard:vxworks->**sp vme_inf_test** ; vme_inf_test assumes EB address at 0xffff9000
task spawned: id = 0x3c11b0, name = t3
value = 3936688 = 0x3c11b0
mallard:vxworks->
Testing AD_MEM REG. Watch Address and Card Select LEDs.
PASSED!! - AD_MEM REG
PASSED!! - MAX_BRD REG
PASSED!! - CNTL REG
PASSED!! - RESET FIFO REG

2.2 CES Memory Module Interface Test

This section describes how to run the test program which test the interface between the Event Builder and the CES Memory Module. The test program performs the following steps.

- Puts the Event Builder into test mode ad prepares it to make transfers into the CES Memory Module.
- Sets up the CES Memory Module to accept data from the Event Builder
- Writes data into the Event Builder which is transferred over the 32-bit Differential ECL Bus into the CES Memory Module.
- Verifies that the block of data has been properly written into the CES Module and that the status and control registers on the CES Module and Event Builder are correct.

To execute the test enter the following command while logged into the VME network node running VxWorks.

mallard:vxworks->**sp vme_ces_test,0,0x11000000**

NOTE: The first parameter is the Event Builder's Address of the CES Memory Card that is to be filled with the test data. The second parameter is the Base Address of the CES Module.

```
task spawned: id = 0x3c11b0, name = t3
value = 3936688 = 0x3c11b0
mallard:vxworks->

Please Wait Writing Data into Event Builder!
Note: Memory overflow bit set before Busy = 0
Please Wait Testing Data In CES Module!
TEST PASSED!!

For Information I will execute vme_reg_dump.
CES MEMORY MODULE REGISTER VALUES. Base address = 0x110000000
Interrupt Reg     - 0x2  =  0xfff00fc
Control Reg       - 0x4  =  0xffff3092
Address Pointer - 0x8  =  0xffff8000
Word Counter     - 0xc  =  0xfff001ff

Event Builder Register Values.    Base address = 0xffff9000
Selected CES Module    - 0x0 = 0xffc0
Max CES Modules        - 0x2 = 0xffc1
EB Control Reg         - 0x4 = 0xfff9
EB EOE Counter         - 0x6 = 0xff00
EB FIFO RESET Latch    - 0x8 = 0xffff
EB FIFO STATUS         - 0x10 = 0x400f
EB LS FIFO VALUE       - 0xa = 0xf0d3
EB MS FIFO VALUE       - 0xc = 0xb84a
```

2.3 EBX Interface Test

This section describes how to test the interface between the Event Builder and the EBX (Event Builder Expansion) Module.

The EBX Module must be plugged into the same VME crate as the EB Module and also have its P2-A row connector tied to the P2-A of the Event Builder. Recall that the dip switches on the EBX Module determine the EB addresses that the EBX Module will interface to. For details see the EBX specifications.

To test the EB to EBX interface, repeat the CES Memory Module Interface test using a CES Memory Module connected to the EBX Module. This will mean that the EBX module will have a an EB address larger than 3.

For example if the EBX Module is configured to interface with EB Addresses 4-7 and the CES Memory Module with Base Address of 0x12000000 is connected to EB Address 4 then the following command may be given

mallard:vxworks->**sp vme_ces_test,4,0x11000000**

NOTE: The first parameter is the Event Builder's Address of the CES Memory Card that is to be filled with the test data. The second parameter is the Base Address of the CES Module.

```
task spawned: id = 0x3c11b0, name = t3
value = 3936688 = 0x3c11b0
mallard:vxworks->

Please Wait Writing Data into Event Builder!
Note: Memory overflow bit set before Busy = 0
Please Wait Testing Data In CES Module!
TEST PASSED!!

For Information I will execute vme_reg_dump.
CES MEMORY MODULE REGISTER VALUES. Base address = 0x110000000
Interrupt Reg     - 0x2  =  0xfff00fc
Control Reg       - 0x4  =  0xffff3092
Address Pointer - 0x8  =  0xffff8000
Word Counter    - 0xc  =  0xfff001ff

Event Builder Register Values.    Base address = 0xffff9000
Selected CES Module    - 0x0 = 0xffc0
Max CES Modules        - 0x2 = 0xffc1
EB Control Reg         - 0x4 = 0xfff9
EB EOE Counter         - 0x6 = 0xff00
EB FIFO RESET Latch    - 0x8 = 0xffff
EB FIFO STATUS         - 0x10 = 0x400f
EB LS FIFO VALUE       - 0xa = 0xf0d3
EB MS FIFO VALUE       - 0xc = 0xb84a
```

2.4 System Test and Master Readout Module Interface Test.

The best way to test the Interface between the Master Readout Module and the Event Builder is to run the Data Acquisition System Test as described in the GammaSphere Data Acquisition Overview. Not only will this test verify the interface between the MRM and the EB but it will also retest the EB to CES Interface at full speed. Please read the GammaSphere Data Acquisition Overview for details on how to execute the system test.

3.0 EVENT BUILDER BOARD1 SCHEMATICS

4.0 EVENT BUILDER BOARD2 SCHEMATICS

APPENDIX A: Test Software and Utility Programs

This appendix contains all of the source code used to test the EB. It also includes the start up script which is executed when the VME is powered up. Listed below are all of the files included in this appendix.

File Name Description

vme_inf_test.c - Used to test the VME interface in the Event Builder. Test assumes that the Event Builder has its base address at 0xffff9000. To execute enter the command shown below.

```
->sp vme_inf_test
```

vme_ces_test.c - Used to test the interface between the Event Builder and the CES Memory Modules. To execute enter the following command:

```
->sp vme_ces_test, eb_address, ces_base_address
```

eb_address = The EB address of the CES Memory Module. Note: This is controlled by which connector the control signals are plugged into.

ces_base_address = Base Address of the CES Memory Module.

UTILITIES

SCRIPTS:

c - Used to compile c programs to be executed on the VxWorks Nodes.

Utility Routines:

vme_ma16d16.c - Used to read and modify A16 D16 addresses on the VME Bus.

```
->ma16d16 hex_address
```

hex_address - 16 Bit hex address to read and possibly write.

Routine provides a way to read and write A16, D16 address on the VXI Bus. While running, this routine accepts the following commands.

<enter>	=	Read the next address.
hex_num<enter>	=	Modify the data at the current address and reads it back.
^<enter>	=	Read the previous address.
.<enter>	=	Exit the program.

vme_ma32d32.c - Used to read and modify A32 D32 addresses on the VME Bus.

->ma32d32 hex_address
hex_address - 32 Bit hex address to read and possibly write.

Routine provides a way to read and write A16, D16 address on the VXI Bus. While running, this routine accepts the following commands.

<enter>	=	Read the next address.
hex_num<enter>	=	Modify the data at the current address and reads it back.
^<enter>	=	Read the previous address.
.<enter>	=	Exit the program.

vme_wca16d16.c - Used to continuously write the given data to the given address.

->sp wca16d16, hex_address, hex_data

hex_address	=	16 Bit hex address to write continuously to.
hex_data	=	16 Bit hex data to write continuously.

vme_rca32d32.c - Used to continuously read a given address. When ever the data changes the address and data will be typed on the screen.

->sp rca16d16, hex_address

hex_address	=	16 Bit hex address to read continuously to.

STARTUP.CMD - Executed on VME power up

```
shellPromptSet("mallard:vxworks-> ")
hostAdd ("llama", "146.137.180.16")
hostAdd ("walleye", "146.137.180.23")
hostAdd ("pike", "146.137.180.20")
cd "cougar:/home/cougar/WEIZEOR/gamma_utl"
ld < vme_ma32d32.o
ld < vme_ma16d16.o
ld < vme_wca16d16.o
ld < vme_rca32d32.o
ld < vme_inf_test.o
ld < vme_ces_test.o
cd "cougar:/home/cougar/WEIZEOR/gamma_sys_test"
ld < vme_init_mem.o
ld < vme_init_eb.o
ld < tst_data_ver.o
ld < vme_stubs.o
ld < /home/cougar/WEIZEOR/gamma_utl/vme_ces_test.o
ld < /home/cougar/WEIZEOR/gamma_rpc/svc.o
sp gen_svc
```

APPENDIX B: Parts List and Data Sheets

EVENT BUILDER Revised: February 25, 1993
L8825-9230-DB-00 Revision: B
Bill Of Materials February 25, 1993 11:48:06 Page 1

Item	Quantity	Reference	Part	Notes
1	35	C1,C2,C3,C4,C5,C6,C7,C8, C9,C10,C11,C12,C13,C14. C15,C16,C17,C18,C19,C20, C21,C22,C23,C24,C25,C26, C27,C28,C29,C30,C31,C32, C36,C38,C39	0.1	
2	4	C33,C34,C35,C37	33/10V	
3	17	D1,D2,D3,D4,D5,D6,D7,D8, D9,D10,D11,D12,D13,D14, D15,D16,D17	LED	To Front Panel
4	2	F1,F2	FUZE	Fuze Socket
5	4	H1,H2,H3,H4	HEAD9	
6	6	JP1,JP2,JP3,JP12,JP16, JP17	JUMPER	
7	8	JP4,JP5,JP6,JP7,JP13, JP14,JP15,JP18	HEADER 11X2	
8	4	JP8,JP9,JP10,JP11	HEADER 6X2	Rt Angle
9	2	JP20,JP19	HEADER 22X2	Str. Up to Daughter Brd. using a Viking connectors
10	6	P1C,P1B,P1A,P2C,P2B,P2A	DIN96	2 DIN96 Connectors
11	3	RP1,RP2,RP3	SIP10	390 ohms
11.5	3	RP6,RP7,RP8	SIP10	4.7k
12	3	RP4,RP5,RP10	SIP8	330 ohms
13	1	RP9	SIP6	390 ohms
14	4	R1,R2,R8,R9	0	
15	2	R3,R10	1K	
16	5	R4,R5,R6,R7,R11	100	
17	2	R12,R13	5K	
18	1	R14	120	
19	1	R15	75	
20	1	S1	SW DIP-2	
21	4	U1,U2,U3,U4	74HC245	
22	2	U5,U17	XC3042	84 pin PLCC Socket
23	2	U6,U26	XC1736A	8 pin DIP Socket
24	2	U7,U8	74HC374	
25	4	U9,U10,U12,U15	10124	
26	2	U16,U11	10125	
27	1	U13	96F173	
28	2	U23,U14	74LS240	
29	2	U18,U19	74LS645	
30	3	U20,U24,U25	74HC244	
31	1	U21	74LS00	
32	1	U22	74S38	
33	1	U27	MC7905.2	
34	1	U28	LM337T	
35	1	U29	74LS241	
36	1	X1	XO43B	10Mhz Crystal Osc.

EVENT BUILDER DAUGHTER BOARD Revised: December 1, 1992
L8825-9233-DB-00 Revision: A
Bill Of Materials February 25, 1993 11:38:41 Page 1

Item	Quantity	Reference	Part
1	35	C1,C3,C5,C6,C7,C9,C11, C13,C14,C15,C17,C18,C19, C20,C21,C22,C24,C25,C26, C28,C30,C31,C32,C34,C36, C38,C39,C40,C42,C43,C44, C45,C46,C47,C48	0.1
2	13	C2,C4,C8,C10,C12,C16,C23, C27,C29,C33,C35,C37,C41	33/10V
3	4	ISO1,ISO2,ISO4,ISO5	259003-1
4	1	ISO3	HFE4211-15
5	2	JP1,JP2	HEADER 22X2
6	2	J1,J2	CON34A
7	16	L1,L2,L3,L4,L5,L6,L7,L8, L9,L10,L11,L12,L13,L14, L15,L16	2.7uH
8	8	R1,R2,R8,R14,R18,R19,R26, R27	270
9	8	R3,R4,R9,R10,R20,R21,R28, R29	510
10	8	R5,R11,R12,R13,R22,R23, R30,R31	180
11	8	R6,R7,R15,R16,R24,R25, R32,R33	470
12	1	R17	100 1/2W
13	4	U1,U4,U11,U14	AMD7969
14	8	U2,U3,U5,U6,U12,U13,U15, U16	MC10H124
15	1	U7	74S00
16	4	U8,U9,U17,U18	IDT7205L35 TP
17	1	U10	74LS240
18	1	X1	XO43B