EVENT BUILDER EXPANSION MODULE
August 18, 1993

1.0 OVERVIEW OF EVENT BUILDER EXPANSION MODULE 2
1.1 Description of the Event Builder Expansion Module 2
1.2 Block Diagram of the Event Builder Expansion Module 3

2.0 TEST PROCEDURE 7

3.0 EVENT BUILDER EXPANSION MODULE SCHEMATICS 8

APPENDIX A: ABEL PROGRAM FOR 22V10 10

1.0 OVERVIEW OF EVENT BUILDER EXPANSION MODULE

This section gives a brief overview of the EBX Module (Event Builder EXPANSION). Argonne National Laboratory's ECT Department designed and developed the EBX Module. The EBX Module is for LBL (Lawrence Berkeley Laboratory) to be used in the GammaSphere Data Acquisition System.

1.1 Description of the Event Builder Expansion Module

The EBX Module is a single wide B size VME Module. Its main function is to allow the Event Builder to interface with more than four CES Memory Modules. Each EBX Module allows the Event Builder to transfer data into four additional Memory Modules. The Event Builder controls the EBX Module over the P2-A VME connector. The EBX Module uses the signals over the P2 connector to generate the WSI, BUSY, and VETO signals. These signals control the data transfer into the CES Memory Modules. Include in the P2 signals is the MEM_OVF signal from the selected memory module, and the BRD_ACK signal from the selected EBX Module. For more details see the EBX Block Diagram Section.

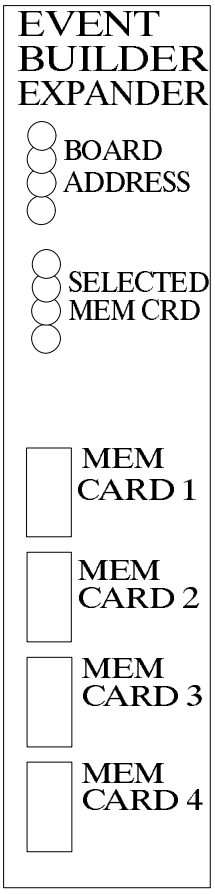
Described below are the LEDs and connectors that can be found on the front panel of the EBX Module.

BOARD ADDRESS LEDs - Shows which Memory Modules the Event Builder Expansion Module is configured to control. When the upper 4 bits of the Memory Module Address matches the Board Address LEDs, then one of the four Memory Modules connected to the EBX Module is selected. The Memory Module Address comes from the Event Builder over the P2 connecter. The Board Address is controlled by four dip switches on the EBX Module

LED on = 1.

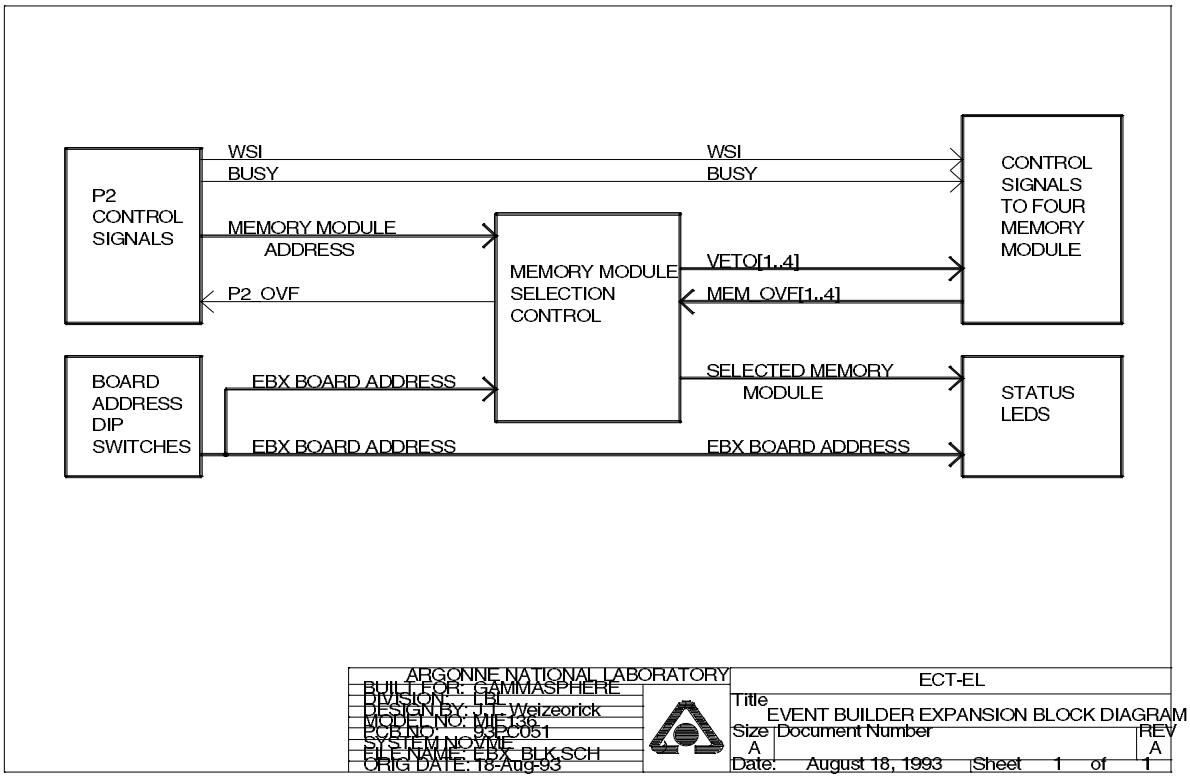
SELECTED MEM CRD LEDS - When an LED is on it indicates that the corresponding Memory Module is selected. Only one of these LEDs may be on at a time, but when the Event Builder is searching for available memory it may appear that more than one LED is on.

MEM1 -> MEM4 Connectors - Connect to the four CES Memory Modules controlled by the EBX Module. These connectors provide the control signals needed by the memory modules to clock in the ECL Data. The SELECTED MEM CRD LEDs determine which memory module is receiving the event data.



1.2 Block Diagram of the Event Builder Expansion Module

Below is a block diagram of the Event Builder Expansion Module. 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 Expansion Module consist of the following subsections listed below.

- **P2 Interface Signals**
- **Board Address Dip Switches**
- **Memory Module Selection Control**
- **ECL Control Signals to Memory Modules**
- **Status LEDs**

The **P2 Interface Signals** are used by the EBX Modules to generate the ECL control signals required by the memory modules. Below are detail descriptions each of the P2 Interface Signals.

- · 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 which is used to prevent the data from a signal event becoming split over two memory modules. As long as the BUSY signal is active the selected memory module will not generate a memory overflow signal. The EBX modules pass this signal to all of its 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.

The **Board Address Dip Switches** are used to select the addresses of the memory modules that the EBX module will interface with. The EBX Module compares the dip switches with the upper four bits of the selected memory module address. If they match then the EBX Module will use the lower two bits of the memory module address to determine which one of its' four memory modules will be selected. The position of the dip switches can be read by looking at the Board Address LEDs.
(LED ON = 1; LED OFF = 0)

The **Memory Module Selection Control** consist of a 22V10 PAL and some discrete ECL logic. The 22V10 compares the dip switch selection with the memory module address controlled by the Event Builder. When the dip switches match the upper four bits of the memory module address then the EBX Module becomes enabled. When enabled the EBX Module uses the two least significant memory module address bits to determine which of its four memory modules is to receive the event data. The chosen memory module has its VETO signal lowered and its corresponding led on the EBX front panel light. The discrete ECL logic routes the selected memory module's overflow signal back to the Event Builder over the P2 connector. The multiplexed overflow signal is called OVF_P2.

The **ECL Control Signals to the memory modules** are on a separate ribbon cables connected to each CES memory module from the EBX Module. The EBX Module like the Event Builder can interface with up to four memory modules. If additional memory modules are needed then more 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 connected 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.

There are a total of eight **Status LEDs** on the front panel of the EBX Module. The four Board Address LEDs match the position of the DIP switches and show the base address of the Memory Modules that the EBX Module will control. The four Select LEDs show which Memory Module is currently selected to receive the Event Data.

2.0 TEST PROCEDURE

The EBX Module can only be tested through the Event Builder. There are two methods to test the EBX Module.

> **Manual Test** - Involves only the Event Builder, EBX Module, and a VME Master. Does not test data moving at full speed, but will test that everything is hooked up properly and help the operator to isolate errors in the system test.
>
> **System Test** - The system test requires many more boards and if an error occurs it may be difficult to decide which board is at fault. The system test is described in detail in the Master Readout Module Test Procedure.

The following describes the Manual Test which can be performed with the EBX modules.

????????????????? Need To Complete ?????????????????

3.0 EVENT BUILDER EXPANSION MODULE SCHEMATICS

APPENDIX A: ABEL PROGRAM FOR 22V10

```
module EBX_U1C
title 'EVENT BUILDER EXPANDER - MEMORY SELECT LOGIC'

        EBX_U1C device 'P22V10';

        SEL2            pin 1;  "Select inputs from dip switches
        SEL3            pin 2;
        SEL4            pin 3;
        SEL5            pin 4;
        I5              pin 5;  "Not used
        I6              pin 6;  "Not used
        AD_MEM0         pin 7;  "Address of selected memory card (0-64)
        AD_MEM1         pin 8;
        AD_MEM2         pin 9;
        AD_MEM3         pin 10;
        AD_MEM4         pin 11;
        AD_MEM5         pin 13;
        VETO4           pin 14; "Veto signals are inputs to the four memory
        VETO3           pin 15; "cards that the EBX_BRD interfaces to
        VETO2           pin 16; "The Veto of the addressed memory card is low
        VETO1           pin 17;
        MEM4_LED        pin 18; "Mem led outputs are used to show the operator
        MEM3_LED        pin 19; "which memory card is currently active.
        MEM2_LED        pin 20;
        MEM1_LED        pin 21;
        BRD_ACK         pin 22; "This signal is low when SEL2->5 = AD_MEM2->5
        BRD_EN          pin 23; "Same as BRD_ACK except always enabled

        H,L,X,Z,C =   1,0,.X.,.Z.,.C.;
        OUT0      =   ((SEL2 !$ AD_MEM2) & (SEL3 !$ AD_MEM3) &
                      (SEL4 !$ AD_MEM4) & (SEL5 !$ AD_MEM5));

        VETO4, VETO3, VETO2, VETO1, MEM4_LED, MEM3_LED,
        MEM2_LED, MEM1_LED, BRD_ACK ISTYPE 'com';
```

```
equations
        BRD_ACK     = L;
        BRD_ACK.OE  = !BRD_EN;

        BRD_EN      = !((SEL2 !$ AD_MEM2) & (SEL3 !$ AD_MEM3) &
                      (SEL4 !$ AD_MEM4) & (SEL5 !$ AD_MEM5));

        MEM1_LED    = OUT0 & (AD_MEM1 !$ 0) & (AD_MEM0 !$ 0);
        MEM2_LED    = OUT0 & (AD_MEM1 !$ 0) & (AD_MEM0 !$ 1);
        MEM3_LED    = OUT0 & (AD_MEM1 !$ 1) & (AD_MEM0 !$ 0);
        MEM4_LED    = OUT0 & (AD_MEM1 !$ 1) & (AD_MEM0 !$ 1);

        VETO1       = !MEM1_LED;
        VETO2       = !MEM2_LED;
        VETO3       = !MEM3_LED;
        VETO4       = !MEM4_LED;

TEST_VECTORS
([SEL5, SEL4, SEL3, SEL2, AD_MEM5, AD_MEM4, AD_MEM3, AD_MEM2,
  AD_MEM1, AD_MEM0] ->
 [VETO4, VETO3, VETO2, VETO1, MEM4_LED, MEM3_LED, MEM2_LED,
  MEM1_LED, BRD_ACK, BRD_EN])

[0,0,1,0, 0,0,0,0,0,0]  ->  [1,1,1,1, 0,0,0,0, Z, 1];
[0,0,1,0, 0,0,1,0,0,0]  ->  [1,1,1,0, 0,0,0,1, 0, 0];
[0,0,1,0, 0,0,1,0,0,1]  ->  [1,1,0,1, 0,0,1,0, 0, 0];
[0,0,1,0, 0,0,1,0,1,0]  ->  [1,0,1,1, 0,1,0,0, 0, 0];
[0,0,1,0, 0,0,1,0,1,1]  ->  [0,1,1,1, 1,0,0,0, 0, 0];
[0,0,1,0, 0,0,1,1,0,0]  ->  [1,1,1,1, 0,0,0,0, Z, 1];

end EBX_U1C
```