MuID FEM Mode Control Design Document

JSHicks

Inputs: From T&C Glink (~40 Mhz)

BIT #	Signal	
0	Mode Bit 0	
1	Mode Bit 1	
2	Mode Bit 2	
3	Mode Bit 3	
4	Mode Bit 4	
5	Mode Bit 5	
6	Mode Bit 6	
7	Mode Bit 7	(Alignment Bit)
8	Beam Clk (9.4Mhz)	
9	LVL1 Accept	
10	Timing Mode Enable	
11	EnDat0	
12	EnDat1	
13	User Bit 0	(0 0 1 – Reset ARCNET)
14	User Bit 1	(0 1 0 – FEM Self-test)
15	User Bit 2	
16	Endat2	
17	Endat3	
18	Reserved	
19	Reserved	

Other Inputs:

/CLR	Sets all outputs to low state - Halt state
/DAV	Data Available from GLINK
/LINKRDY	GLINK signal indicating locked operation
STAT0	GLINK signal indicating correct state machine recovery of the clk strobe

Outputs:

MODE_B_ERR
B_CLK
RUN
CLR_CNTR
CLR_FIFO
CLR_DLY
S_FUN1
S_FUN2
PULSER
ALIGN
LVL1_ACPT
EN_DAT
FEM_RST
FEM_SELF_TEST
ARCNET_RST

Design assumptions:

DAV* and LINKRDY* are used to determine if the GLINK is finished with its startup procedure and running in locked mode with valid data.

STAT0 is used to gate the Beam-Clock mode bit.

The Beam-Clock, B_CLK, mode bit is used to clock the mode-bit control logic. It is also assumed that the mode bit data D0-D7 and D9-D19 change on the rising edge of the B_CLK data bit D8. The input latch for mode control accepts the data from the GLINK and latches it on the falling edge of B_CLK. This ensures stable data for input to the mode-bit decoder block. Data is processed by the decoder asynchronously and then re-synchronized with B_CLK at the output register, which is triggered on the rising edge of B_CLK.

MODES of Operation:

Modes should be decoded and used only if Mode Enable is high. Else the FEM should be placed in the **HALT** state (RUN low state). Power up in the Halt state.

Run

Run is for normal operation, FEM is running & collecting data. Default to RUN mode when invalid mode combinations are received.

HALT (complement of RUN)

Halt is used to Stop all operations including data formatting & transmission to the DCM. This should allow a new serial data stream form ARCnet to be loaded without problems - for FEM configuration, monitoring and diagnostics.

Initialize Reset (CLR_CNTR, CLR_FIFO, CLR_DLY)

The Initialization Reset is used to place the FEM in an initialized state. This is the same state that the FEM should be in at power concerning counters and data FIFOs. All counters, FIFOs, etc. are reset to zero. Configure information should be saved. Data is lost – even LVL1 qualified data is lost if it has not been completely sent.

Resysnc Reset (CLR_CNTR)

Reset the BCLK & Event counters. This should take effect at the next valid LVL1_ACPT to avoid data loss.

Special Reset (CLR_DLY)

Clears the output (final) flip-flop in the DelayMux chip. Special Reset provides sub-system specific reset options.
For this and other commands requiring more than one beam clk cycle for execution, the scheduler should output the commands the appropriate number of beam clock cycles.

Special Function – 1 & 2

Not yet defined for MuID – for future use.

Special Function – 3 (Fire Test Pulses)

Fire the local pulser for testing purposes.

Table 1.

mB7	mB6	mB5	mB4	mB3	mB2	mB1	mB0	
0	0	0	0	0	0	0	0	HALT
0	0	0	1	0	0	0	0	RUN Assuming D10 = 1 (Mode Enable)
0	0	0	1	0	0	0	1	Resync Reset BCLK & Event counters LVL1_ACPT
0	0	0	1	0	0	1	0	Init Reset Zero all counters May Flush Data
0	0	0	1	0	0	1	1	Special Reset - (Delay MUX outputs)
0	0	0	1	0	1	0	0	Special Function - 1 (not implemented yet for MuID)
0	0	0	1	1	0	0	0	Special Function - 2 (not implemented yet for MuID)
0	0	0	1	1	1	0	0	Special Function - 3 (Fire local test Pulser)
	X	X						Extra bits – No current assignment
1	0	0	1	0	0	0	0	Alignment Bit
X	X	X	X	X	X	X	X	RUN

USER_BITS		
0 0 0		No function
0 0 1		Arcnet Reset
0 1 0		FEM Self Test
X X X		No function

Appendix A

```
--*****************************************************************************
-- Mode Control for MuID                                              modectrl.vhd
-- Author:  Steve Hicks
-- Date: APR 07, 1999
-- DataFlow design of mode bit decoder
--   with sequential synchronization of outputs
--
--
-- Revision History
-- 07-APR-99  Added DMux_rst to functions activated by InitReset
-- 11-MAY-98
-- 19-APR-98
-- Created: MAR 20, 1998
--*****************************************************************************

--  Library Clause
library ieee;
--  Use Clause
use ieee.std_logic_1164.all;
--Library Units

--  Entity Declaration

--IO Interface Declaration

entity modectrl is
  port
  (
   -- inputs
   clk:      IN STD_LOGIC;
     MODE_Bits:     IN STD_LOGIC_VECTOR(7 downto 0);
   USER_Bits:    IN STD_LOGIC_VECTOR(2 downto 0);
   LVL1_ACPT:    IN STD_LOGIC;

   -- outputs
   run:      out STD_LOGIC;
   clr_cntr:   out STD_LOGIC;
     clr_fifo:   out STD_LOGIC;
   DMux_rst:   out STD_LOGIC;
   s_fun1:     out STD_LOGIC;
   s_fun2:     out STD_LOGIC;
   pulser:     out STD_LOGIC;
   align:      out STD_LOGIC;
   mode_B_ERR:   out STD_LOGIC;
   ARCNET_rst:   out STD_LOGIC;
   Self_Test:    out STD_LOGIC
  );
end modectrl;

-- Architecture body

architecture behaviorial of modectrl is

signal MBout_v:    STD_LOGIC_VECTOR(9 downto 0); -- signal for encoding MODE Bit
output lines
signal UBout_v:    STD_LOGIC_VECTOR(6 downto 0); -- signal for encoding USER Bit
output lines
SIGNAL set:        STD_LOGIC;
SIGNAL asyn_resyn:   STD_LOGIC;
SIGNAL resynch:    STD_LOGIC;
```

```
SIGNAL clr_cntr_init:   STD_LOGIC;


begin
  WITH MODE_Bits SELECT

  MBout_v <=
       "0000000000" WHEN "00000000",  -- HALT
       "0000000001" WHEN "00010000",  -- RUN
       "0000000011" WHEN "00010001",  -- ReSynch (BCLK, EVENT counters)
       "0000011101" WHEN "00010010",  -- InitReset (counters, FIFO's, Delay Mux
data)
       "0000010001" WHEN "00010011",  -- Special Reset0 (Delay MUX outputs)
       "0000100001" WHEN "00010100",  -- Special Function1 ( )
       "0001000001" WHEN "00011000",  -- Special Function2 ( )
       "0010000001" WHEN "00011100",  -- Special Function3 (Pulser)
       "0100000001" WHEN "10010000",  -- Alignment
       "1000000001" WHEN OTHERS;      -- Default RUN Set Mode Bit Error Flag

       run         <=   MBout_v(0);
       asyn_resyn    <=  MBout_v(1);
       clr_cntr_init  <= MBout_v(2);
       clr_fifo    <= MBout_v(3);
       DMux_rst    <= MBout_v(4);
       s_fun1      <= MBout_v(5);
       s_fun2      <=  MBout_v(6);
       pulser      <=  MBout_v(7);
       align       <=  MBout_v(8);
       mode_B_ERR  <=  MBout_v(9);


  WITH USER_Bits SELECT

  UBout_v<= "0000001" WHEN "001",     -- Reset the ARCNET
            "0000010" WHEN "010",     -- FEM Self-Test
            "0000000" WHEN OTHERS;    -- Default

---------------------------------------------
  seq_output:
  PROCESS (MBout_v, UBout_v)
  BEGIN
  WAIT UNTIL clk = '1';

    -- Signal Assignment Statement
    ARCNET_rst  <=   UBout_v(0);
    Self_Test <=  UBout_v(1);

  END PROCESS seq_output;
---------------------------------------------

  PROCESS (clk)
    BEGIN
     IF (clk'EVENT AND clk = '1') THEN
      CASE set IS
        WHEN '0'=>
          IF asyn_resyn = '1' THEN
           set <= '1';           -- S set
          END IF;
          IF LVL1_ACPT = '0' THEN
           --resync_out <= '0';
          END IF;
        WHEN '1'=>
          IF LVL1_ACPT = '1' THEN
```

```
            set <= '0';           -- R reset
            --resync_out <= '1';
           END IF;
          WHEN others =>
           set <= '0';
       END CASE;
     END IF;

     END PROCESS;

  resynch  <= LVL1_ACPT AND set;
  clr_cntr <= resynch  OR clr_cntr_init;

end behaviorial;
```