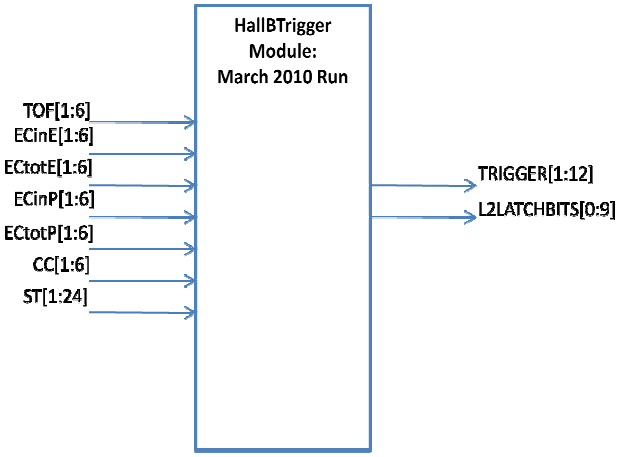
HallB Trigger Module – Run March 2010 Description

Implemented on CAEN V1495 Programmable VME Module



6x Sector Inputs:
TOF, ECinE, ECtotE,
ECinP, ECtotP, CC, ST

Trigger Outputs:
12 Programmable, TRIG[1:12]
10 Bit Prescaler
0-35ns Pulse Stretcher

Programmable Delays:
All Sector inputs have individually controllable delays: 0 to 155ns in 5ns steps

Trigger Output Jitter:
5ns

Scalars:
16/32bits 100MHz Count Rate
On all inputs, outputs, and most intermediate stages in trigger logic

Trigger Module Delays Paths (nominal)

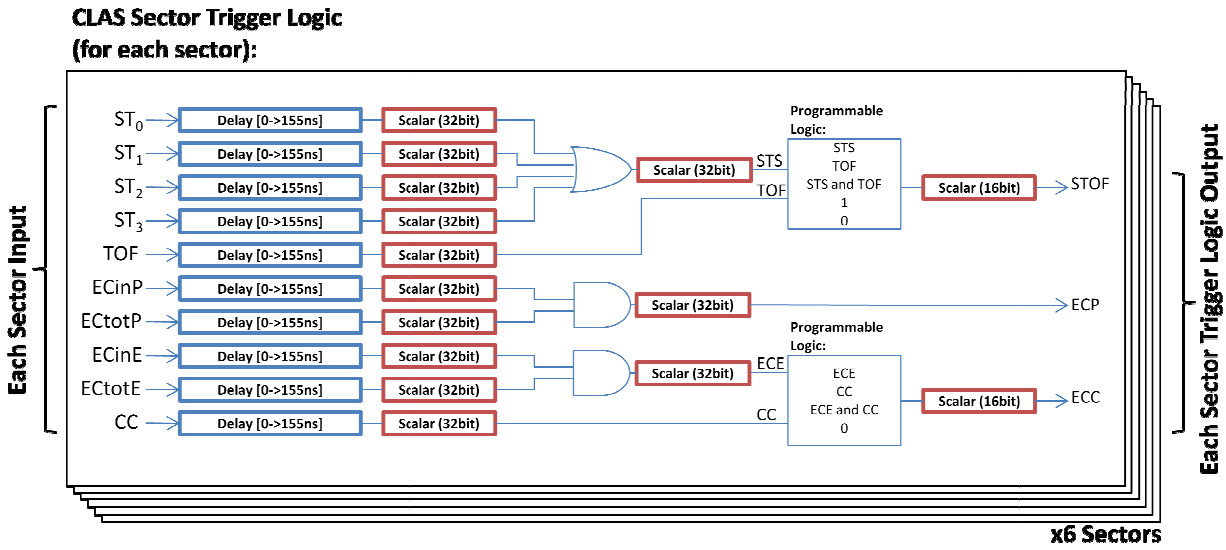
Path		Delay
From	To	
ST[1:24]	TRIG[1:12]	52ns
TOF[1:6]	TRIG[1:12]	59ns
ECPin[1:6],ECPtot[1:6]	TRIG[1:12]	52ns
ECEin[1:6],ECEtot[1:6]	TRIG[1:12]	52ns
CC[1:6]	TRIG[1:12]	52ns
MORA, MORB	TRIG[1:12]	52ns
TRIG[1:12]	L2LATCHBITS	35ns

Notes:

1) All input->output delays measured with all programmable delays set to their minimum values. For ST, TOF, ECPin, ECPtot, ECEin, ECEtot, CC, MORA, MORB the minimum delay setting is 0ns. For L2LATCHBITS the minimum delay setting is 10ns.

Sector Trigger Logic Block Diagram:

Shown below is the block diagram computed on each sector subsystem of CLAS. The resulting 6 sector triggers are supplied to each of the 12 programmable global trigger bits and level 2 latch bits logic.

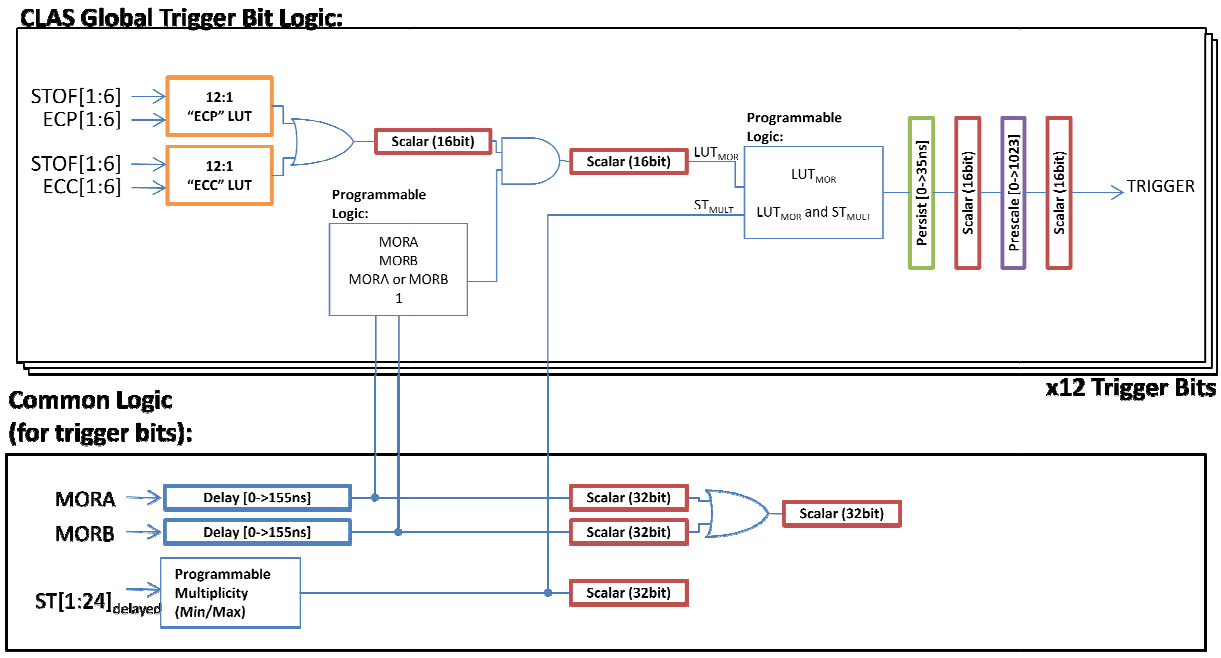


Notes:

1) Programmable delays shown are independently set from one another.
2) Scalars must be disabled during readout as discussed in register section.=
3) Scalars can count at a rate of 100MHz

Global Trigger Bit Logic Block Diagram:

Shown below is the block diagram computed for each of the 12 trigger output bits of this trigger module. These trigger decisions are delivered to the level 2 latch bit logic as well as output as differential NECL signals to interface with the Trigger Supervisor.



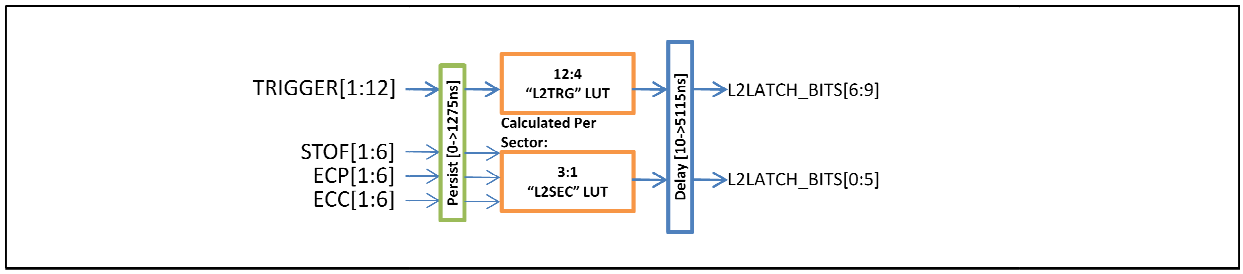
Notes:

1) See LUT structures below for details on how bit mapping is performed to ensure LUTs are programmed properly.
2) Scalars must be disabled during readout as discussed in register section.
3) Scalars can count at a rate of 100MHz
4) Persistence blocks will stretch the trigger decisions by amount programmed.
5) Setting prescalers to '0' will disable the trigger output (it will be set to a differential logic '1' permanently)
6) Common logic block shows "$ST[1:24]_{delayed}$" signals, which are the 24 start counter signals from all 6 sectors delayed as programmed in the sector configuration logic. Delay paths are internally compensated for such that if the ST signals are delayed appropriately for the sector logic to create a trigger, the ST multiplicity calculation will also be in time with the trigger signal.
7) **TRIGGER output signals are inverted with respect to V1495 connector documentation to accommodate Trigger Supervisor cable connections**

Level 2 Latch Logic Bits:

Shown below is the block diagram computed for the level 2 latch bit input signals, which are based on sector logic and global logic bits as shown.

L2 Sector Latch Logic:



Notes:

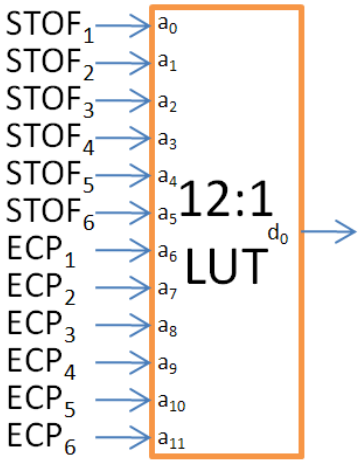
1) See LUT structures below for details on how bit mapping is performed to ensure LUTs are programmed properly.
2) Persistence blocks will stretch the input signals by amount programmed. In this case it can stretch signals by adding anywhere from 0 to 1275ns in steps of 5ns.
3) Programmable delay ranges from **10 to 5115ns**.

LUT Structure/Bit Mappings:

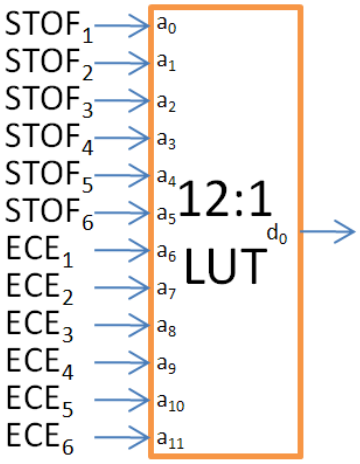
Care must be taken to follow LUT input/output convention when creating custom LUTs.

Trigger Bit LUTs:

ECP LUT Bit Mapping:

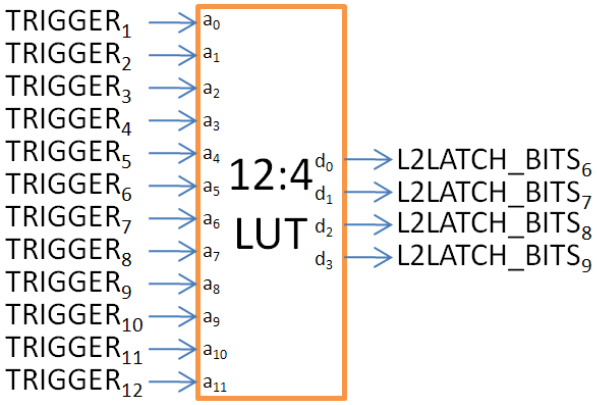


ECE LUT Bit Mapping:

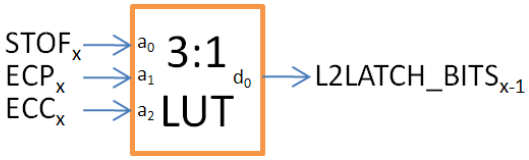


Level 2 LUTs:

L2TRG LUT Bit Mapping:

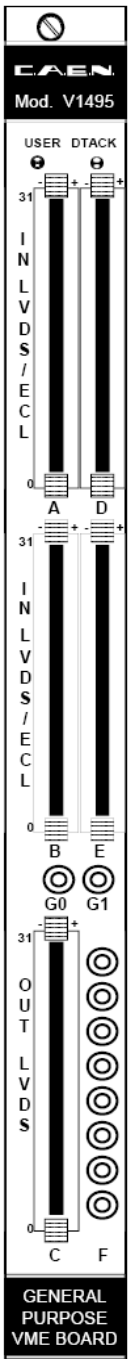


L2SEC LUT Bit Mapping:

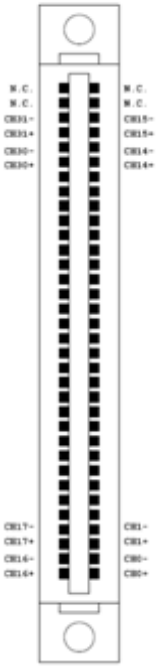


Note: 'x' represents a CLAS sector# (1-6)

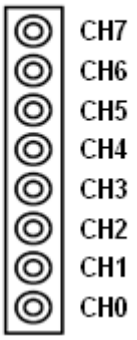
HallBTriggerMarch2010 Module Pinouts



32 Channel Input/Output Module Pinout



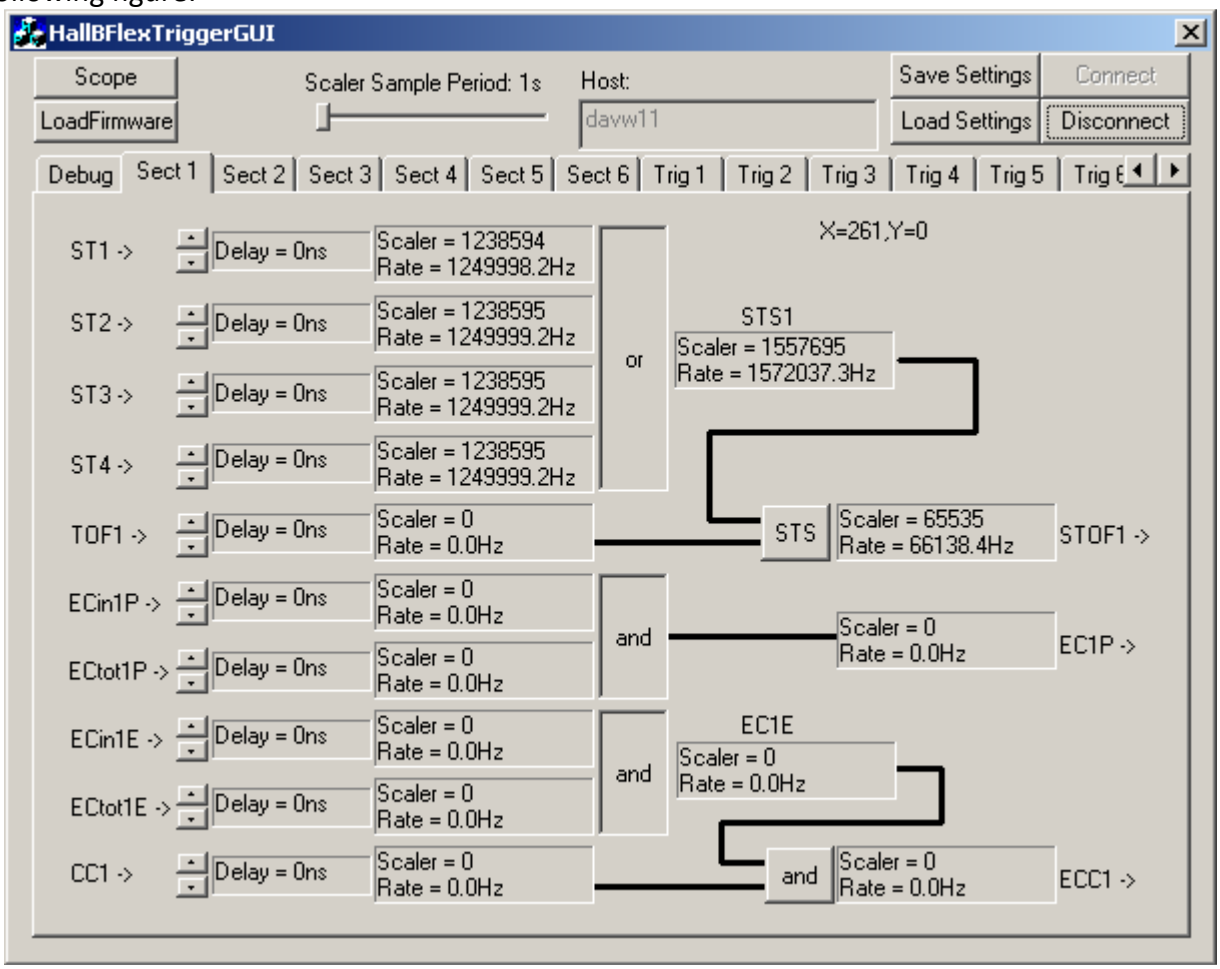
LEMO I/O NIM/TTL Module Pinout



Ch	PortA	PortB	PortC	PortD	PortE (A395C)	PortF(A395D)
0	IC Total Clusters Bit 0	ECinP1	unused	unused	unused	MORA
1	IC Total Clusters Bit 1	ECtotP1	unused	unused	TRIG1	MORB
2	IC Total Clusters Bit 2	ECinE1	unused	unused	TRIG2	TOF1
3	IC Total Clusters Bit 3	ECtotE1	unused	unused	TRIG3	TOF2
4	IC Total Clusters Bit 4	ECinP2	unused	unused	TRIG4	TOF3
5	IC Total Clusters Bit 5	ECtotP2	unused	unused	TRIG5	TOF4
6	IC Total Clusters Bit 6	ECinE2	unused	unused	TRIG6	TOF5
7	IC Hodo Clusters Bit 0	ECtotE2	unused	unused	TRIG7	TOF6
8	IC Hodo Clusters Bit 1	ECinP3	unused	unused	TRIG8	
9	IC Hodo Clusters Bit 2	ECtotP3	unused	unused	TRIG9	
10	IC Hodo Clusters Bit 3	ECinE3	unused	unused	TRIG10	
11	IC Hodo Clusters Bit 4	ECtotE3	unused	unused	TRIG11	
12	IC Hodo Clusters Bit 5	ECinP4	unused	unused	TRIG12	
13	IC Hodo Clusters Bit 6	ECtotP4	unused	unused	unused	
14	IC Veto Clusters Bit 0	ECinE4	unused	unused	unused	
15	IC Veto Clusters Bit 1	ECtotE4	unused	unused	unused	
16	IC Veto Clusters Bit 2	ECinP5	unused	unused	L2-SEC1	
17	IC Veto Clusters Bit 3	ECtotP5	unused	unused	L2-SEC2	
18	IC Veto Clusters Bit 4	ECinE5	unused	unused	L2-SEC3	
19	IC Veto Clusters Bit 5	ECtotE5	unused	unused	L2-SEC4	
20	IC Veto Clusters Bit 6	ECinP6	unused	unused	L2-SEC5	
21	unused	ECtotP6	unused	unused	L2-SEC6	
22	unused	ECinE6	unused	unused	L2-LUT0	
23	unused	ECtotE6	unused	unused	L2-LUT1	
24	unused	CC1	unused	unused	L2-LUT2	
25	unused	CC2	unused	unused	L2-LUT3	
26	unused	CC3	unused	unused	unused	
27	unused	CC4	unused	unused	unused	
28	unused	CC5	unused	unused	unused	
29	unused	CC6	unused	unused	unused	
30	unused	unused	unused	unused	unused	
31	unused	unused	unused	unused	unused	

Hall B Trigger Logic & GUI

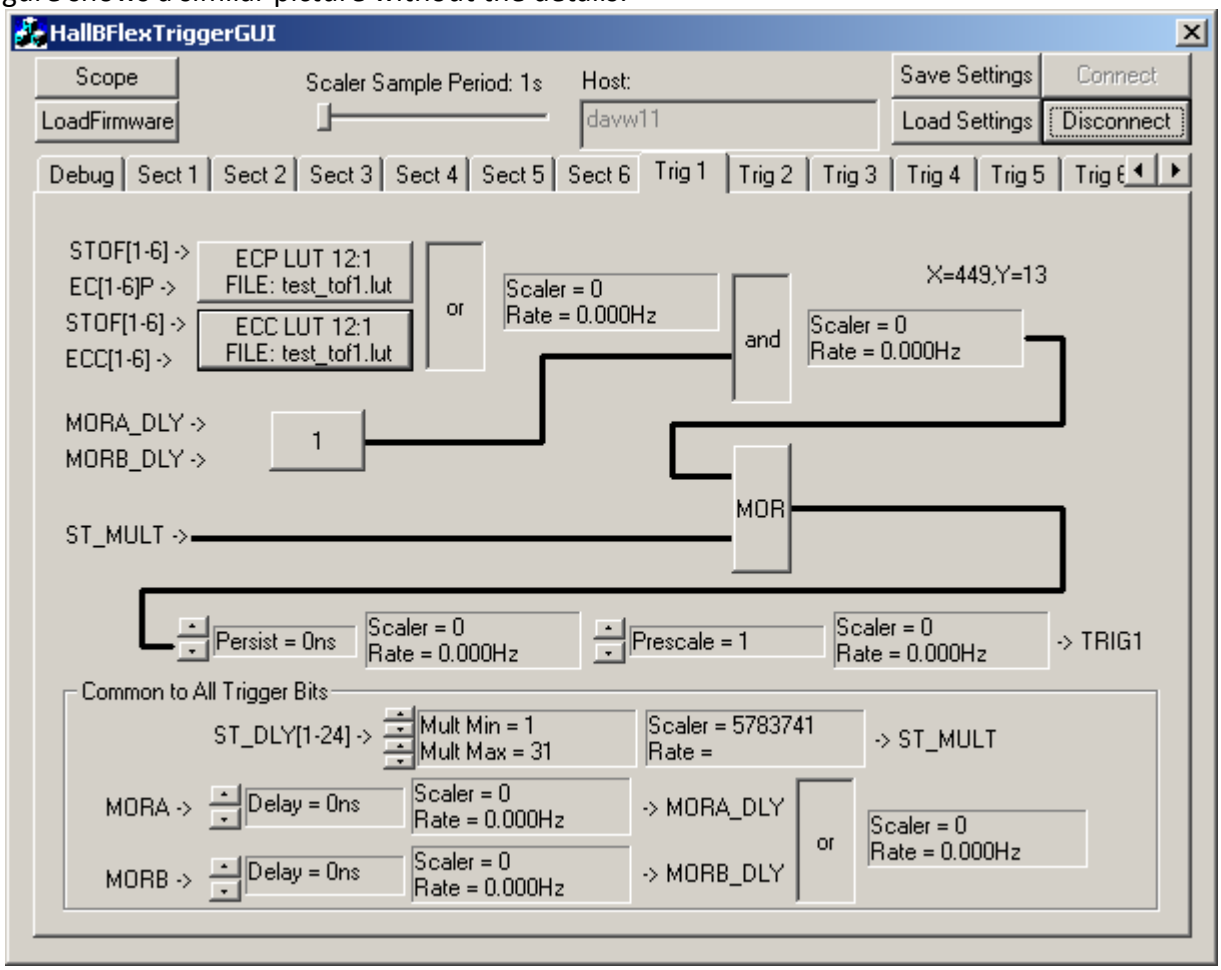
A GUI has been developed to help in commissioning and debugging trigger logic. An array of GUI tabs are provided that allow configuration of sector level signals. This is shown in the following figure:



The sector input signals (TOF, ECinP, ECtotP, ECinE, ECtotE, CC, ST) have individual delay controls and scalers as shown above. These signals go through some logic and form the final sector output signals STOF, ECP, and ECC.

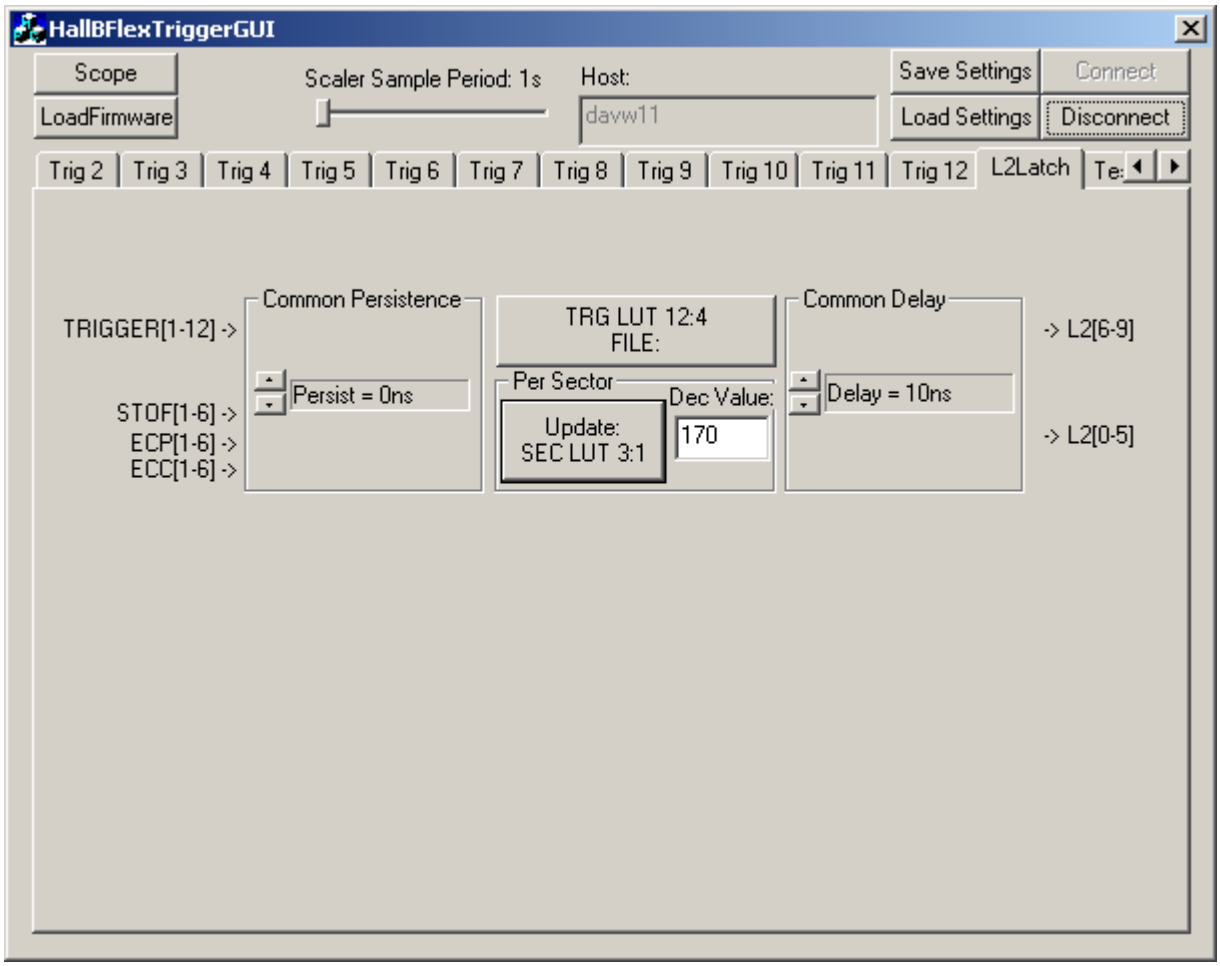
Sector Output Signal		**Equation**
STOF	Mode = AND	∑ST · TOF
	Mode = STS	∑ST
	Mode = TOF	TOF
	Mode = 1	1
	Mode = 0	0
ECP		ECinP · ECtotP
ECC	Mode = AND	ECinE · ECtotE ·CC
	Mode = CC	CC
	Mode = ECE	ECinE · ECtotE
	Mode = 0	0

The signals (STOF, ECP, ECC) from each sector are presented to each of the 12 programmable trigger outputs. TRIG1 to TRIG12 use a LUT approach to provide flexibility in changing triggers without having to recompile firmware. The trigger logic was presented earlier, but the next figure shows a similar picture without the details:



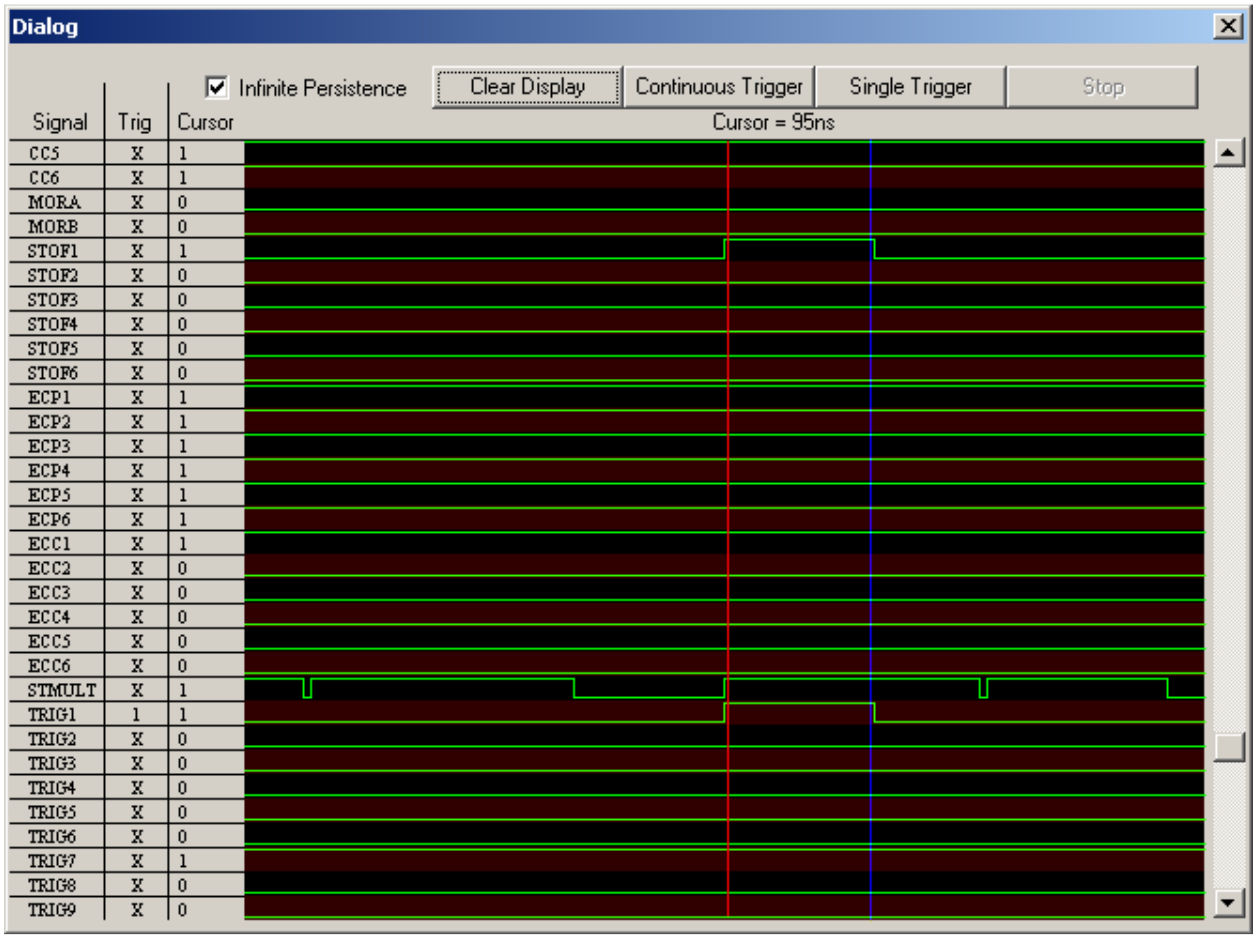
Each trigger bit uses 4 LUTs which have all sector output signals as inputs. In the GUI the LUTs can be reprogrammed at anytime by loading a properly format LUT file. Each trigger output bit includes a 10bit prescaler. The trigger bit can be disabled (which sets the output high) be setting the prescaler value to 0. Scalers are placed at each LUT output to monitor the intermediate stages in the trigger logic.

The L2Latch tab contains the controls for the 10bit latch input pattern used to be captured for readout using an external latch for each trigger (used to decide on whether to start level 2).



Scope Interface

The trigger board and GUI have a scope feature that allow nearly all signals in the trigger board to be monitored without interferring with the normal trigger logic. This scope feature allows single stage triggering on any combination of these signals and captures all signals with 5ns resoltion +/- 300ns around the trigger. The following example shows where TRIG1 is programmed for certain ST patterns, a scope trigger was formed by setting the trigger pattern on the left. There are 103 signals on this scope that can be seen by scrolling down. The signals in the scope are processed at different points in time of the trigger board. These have all been deskewed in the hardware to be consistence with the trigger equations in any column.



VME Interface & Registers

```
#ifndef HallBTriggerBoardRegs_H
#define HallBTriggerBoardRegs_H

/* Board Supports VME A32/A24 D32 Accesses (BLT32 only in address range 0x0000-0x0FFC */

#define CLOCK_PERIOD_NS                         5
#define      MAX_PRESCALE                       1023
#define MAX_DELAY_LONG                          1023
#define MAX_DELAY                               31
#define MAX_STMULT                              24
#define MAX_PERSIST_LONG                        255
#define MAX_PERSIST                             7

#define TS_BOARD_ADDRESS                        0x08510000

#define A_BOARDIDS                              0x1300

#define BOARDID_A395A                           0x00    // 32CH IN LVDS/ECL INTERFACE
#define BOARDID_A395B                           0x01    // 32CH OUT LVDS INTERFACE
#define BOARDID_A395C                           0x02    // 32CH OUT ECL INTERFACE
#define BOARDID_A395D                           0x03    // 8CH I/O SELECT NIM/TTL INTER

#define TS_REVISION                             0x1304
#define TS_ERRORS                               0x1308

/***********************************************/
/************** BEGIN SCALER REGISTERS **********/
/***********************************************/
/* Notes:
  1) Scalers are all 32bits, BIG-ENDIAN.
  2) TS_REFCLK_SCALER is a reference scaler which contains gate time of all scalers (in 25ns ticks)
  3) Set TS_ENABLE_SCALERS to '1' to enable scalers. Set to '0' to stop scalers for readout.
     Setting back to '1' will clear all scalers and allow them to count again.
  4) Scalers are capable of counting at 100MHz, which is about 43sec before overflowing at this high rate
*/
#define TS_ENABLE_SCALERS                       0x130C

#define TS_MORA_SCALER                          0x1000
#define TS_MORB_SCALER                          0x1004
#define TS_TOF1_SCALER                          0x1008
#define TS_TOF2_SCALER                          0x100C
#define TS_TOF3_SCALER                          0x1010
#define TS_TOF4_SCALER                          0x1014
#define TS_TOF5_SCALER                          0x1018
#define TS_TOF6_SCALER                          0x101C
#define TS_ECin1P_SCALER                        0x1020
#define TS_ECin2P_SCALER                        0x1024
#define TS_ECin3P_SCALER                        0x1028
#define TS_ECin4P_SCALER                        0x102C
#define TS_ECin5P_SCALER                        0x1030
#define TS_ECin6P_SCALER                        0x1034
#define TS_ECin1E_SCALER                        0x1038
#define TS_ECin2E_SCALER                        0x103C
#define TS_ECin3E_SCALER                        0x1040
#define TS_ECin4E_SCALER                        0x1044
#define TS_ECin5E_SCALER                        0x1048
#define TS_ECin6E_SCALER                        0x104C
#define TS_ECtot1P_SCALER                       0x1050
#define TS_ECtot2P_SCALER                       0x1054
#define TS_ECtot3P_SCALER                       0x1058
#define TS_ECtot4P_SCALER                       0x105C
#define TS_ECtot5P_SCALER                       0x1060
#define TS_ECtot6P_SCALER                       0x1064
#define TS_ECtot1E_SCALER                       0x1068
```

```
#define TS_ECtot2E_SCALER                    0x106C
#define TS_ECtot3E_SCALER                    0x1070
#define TS_ECtot4E_SCALER                    0x1074
#define TS_ECtot5E_SCALER                    0x1078
#define TS_ECtot6E_SCALER                    0x107C
#define TS_CC1_SCALER                        0x1080
#define TS_CC2_SCALER                        0x1084
#define TS_CC3_SCALER                        0x1088
#define TS_CC4_SCALER                        0x108C
#define TS_CC5_SCALER                        0x1090
#define TS_CC6_SCALER                        0x1094
#define TS_ST1_SCALER                        0x1098
#define TS_ST2_SCALER                        0x109C
#define TS_ST3_SCALER                        0x10A0
#define TS_ST4_SCALER                        0x10A4
#define TS_ST5_SCALER                        0x10A8
#define TS_ST6_SCALER                        0x10AC
#define TS_ST7_SCALER                        0x10B0
#define TS_ST8_SCALER                        0x10B4
#define TS_ST9_SCALER                        0x10B8
#define TS_ST10_SCALER                       0x10BC
#define TS_ST11_SCALER                       0x10C0
#define TS_ST12_SCALER                       0x10C4
#define TS_ST13_SCALER                       0x10C8
#define TS_ST14_SCALER                       0x10CC
#define TS_ST15_SCALER                       0x10D0
#define TS_ST16_SCALER                       0x10D4
#define TS_ST17_SCALER                       0x10D8
#define TS_ST18_SCALER                       0x10DC
#define TS_ST19_SCALER                       0x10E0
#define TS_ST20_SCALER                       0x10E4
#define TS_ST21_SCALER                       0x10E8
#define TS_ST22_SCALER                       0x10EC
#define TS_ST23_SCALER                       0x10F0
#define TS_ST24_SCALER                       0x10F4

#define TS_LUT_SCALER_TRIG1                  0x1128
#define TS_LUT_SCALER_TRIG2                  0x112C
#define TS_LUT_SCALER_TRIG3                  0x1130
#define TS_LUT_SCALER_TRIG4                  0x1134
#define TS_LUT_SCALER_TRIG5                  0x1138
#define TS_LUT_SCALER_TRIG6                  0x113C
#define TS_LUT_SCALER_TRIG7                  0x1140
#define TS_LUT_SCALER_TRIG8                  0x1144
#define TS_LUT_SCALER_TRIG9                  0x1148
#define TS_LUT_SCALER_TRIG10                 0x114C
#define TS_LUT_SCALER_TRIG11                 0x1150
#define TS_LUT_SCALER_TRIG12                 0x1154
#define TS_LUT_MOR_SCALER_TRIG1              0x1158
#define TS_LUT_MOR_SCALER_TRIG2              0x115C
#define TS_LUT_MOR_SCALER_TRIG3              0x1160
#define TS_LUT_MOR_SCALER_TRIG4              0x1164
#define TS_LUT_MOR_SCALER_TRIG5              0x1168
#define TS_LUT_MOR_SCALER_TRIG6              0x116C
#define TS_LUT_MOR_SCALER_TRIG7              0x1170
#define TS_LUT_MOR_SCALER_TRIG8              0x1174
#define TS_LUT_MOR_SCALER_TRIG9              0x1178
#define TS_LUT_MOR_SCALER_TRIG10             0x117C
#define TS_LUT_MOR_SCALER_TRIG11             0x1180
#define TS_LUT_MOR_SCALER_TRIG12             0x1184
#define TS_TRIG_PERSIST_SCALER_TRIG1         0x1188
#define TS_TRIG_PERSIST_SCALER_TRIG2         0x118C
#define TS_TRIG_PERSIST_SCALER_TRIG3         0x1190
#define TS_TRIG_PERSIST_SCALER_TRIG4         0x1194
#define TS_TRIG_PERSIST_SCALER_TRIG5         0x1198
#define TS_TRIG_PERSIST_SCALER_TRIG6         0x119C
```

```
#define TS_TRIG_PERSIST_SCALER_TRIG7                    0x11A0
#define TS_TRIG_PERSIST_SCALER_TRIG8                    0x11A4
#define TS_TRIG_PERSIST_SCALER_TRIG9                    0x11A8
#define TS_TRIG_PERSIST_SCALER_TRIG10                   0x11AC
#define TS_TRIG_PERSIST_SCALER_TRIG11                   0x11B0
#define TS_TRIG_PERSIST_SCALER_TRIG12                   0x11B4

#define TS_TRIG_PRESCALE_SCALER_TRIG1                   0x11B8
#define TS_TRIG_PRESCALE_SCALER_TRIG2                   0x11BC
#define TS_TRIG_PRESCALE_SCALER_TRIG3                   0x11C0
#define TS_TRIG_PRESCALE_SCALER_TRIG4                   0x11C4
#define TS_TRIG_PRESCALE_SCALER_TRIG5                   0x11C8
#define TS_TRIG_PRESCALE_SCALER_TRIG6                   0x11CC
#define TS_TRIG_PRESCALE_SCALER_TRIG7                   0x11D0
#define TS_TRIG_PRESCALE_SCALER_TRIG8                   0x11D4
#define TS_TRIG_PRESCALE_SCALER_TRIG9                   0x11D8
#define TS_TRIG_PRESCALE_SCALER_TRIG10                  0x11DC
#define TS_TRIG_PRESCALE_SCALER_TRIG11                  0x11E0
#define TS_TRIG_PRESCALE_SCALER_TRIG12                  0x11E4

#define TS_STS1_SCALER                                  0x11E8
#define TS_STS2_SCALER                                  0x11EC
#define TS_STS3_SCALER                                  0x11F0
#define TS_STS4_SCALER                                  0x11F4
#define TS_STS5_SCALER                                  0x11F8
#define TS_STS6_SCALER                                  0x11FC

#define TS_STOF1_SCALER                                 0x1200
#define TS_STOF2_SCALER                                 0x1204
#define TS_STOF3_SCALER                                 0x1208
#define TS_STOF4_SCALER                                 0x120C
#define TS_STOF5_SCALER                                 0x1210
#define TS_STOF6_SCALER                                 0x1214

#define TS_EC1P_SCALER                                  0x1218
#define TS_EC2P_SCALER                                  0x121C
#define TS_EC3P_SCALER                                  0x1220
#define TS_EC4P_SCALER                                  0x1224
#define TS_EC5P_SCALER                                  0x1228
#define TS_EC6P_SCALER                                  0x122C
#define TS_EC1E_SCALER                                  0x1230
#define TS_EC2E_SCALER                                  0x1234
#define TS_EC3E_SCALER                                  0x1238
#define TS_EC4E_SCALER                                  0x123C
#define TS_EC5E_SCALER                                  0x1240
#define TS_EC6E_SCALER                                  0x1244

#define TS_ECC1_SCALER                                  0x1248
#define TS_ECC2_SCALER                                  0x124C
#define TS_ECC3_SCALER                                  0x1250
#define TS_ECC4_SCALER                                  0x1254
#define TS_ECC5_SCALER                                  0x1258
#define TS_ECC6_SCALER                                  0x125C
#define TS_REF_SCALER                                   0x1260
#define TS_MORAB_SCALER                                 0x1264
#define TS_STMULT_SCALER                                0x1268
/***********************************************/
/************** END SCALER REGISTERS ************/
/***********************************************/

/***********************************************/
/************** BEGIN PRESCALER REGISTERS *******/
/***********************************************/
/* Notes:
  1) TS_PRESCALE_TRIGx sets the prescale value for trigger bit x
  2) Range is from 0 to 1023, where 0 disables output.
```

```
*/
#define TS_PRESCALE_TRIG1                                   0x2200
#define TS_PRESCALE_TRIG2                                   0x2204
#define TS_PRESCALE_TRIG3                                   0x2208
#define TS_PRESCALE_TRIG4                                   0x220C
#define TS_PRESCALE_TRIG5                                   0x2210
#define TS_PRESCALE_TRIG6                                   0x2214
#define TS_PRESCALE_TRIG7                                   0x2218
#define TS_PRESCALE_TRIG8                                   0x221C
#define TS_PRESCALE_TRIG9                                   0x2220
#define TS_PRESCALE_TRIG10                                  0x2224
#define TS_PRESCALE_TRIG11                                  0x2228
#define TS_PRESCALE_TRIG12                                  0x222C
/***********************************************/
/************** END PRESCALER REGISTERS **********/
/***********************************************/

/***********************************************/
/************** BEGIN DELAY REGISTERS **********/
/***********************************************/
/* Note: Delays occupy bits(4:0) of the register.
      Each count is a 5ns delay.
                        Range is 0ns (reg = 0) to 155ns (reg = 31)
*/
#define TS_MORA_DELAY                                       0x2000
#define TS_MORB_DELAY                                       0x2004
#define TS_TOF1_DELAY                                       0x2008
#define TS_TOF2_DELAY                                       0x200C
#define TS_TOF3_DELAY                                       0x2010
#define TS_TOF4_DELAY                                       0x2014
#define TS_TOF5_DELAY                                       0x2018
#define TS_TOF6_DELAY                                       0x201C
#define TS_ECin1P_DELAY                                     0x2020
#define TS_ECin2P_DELAY                                     0x2024
#define TS_ECin3P_DELAY                                     0x2028
#define TS_ECin4P_DELAY                                     0x202C
#define TS_ECin5P_DELAY                                     0x2030
#define TS_ECin6P_DELAY                                     0x2034
#define TS_ECin1E_DELAY                                     0x2038
#define TS_ECin2E_DELAY                                     0x203C
#define TS_ECin3E_DELAY                                     0x2040
#define TS_ECin4E_DELAY                                     0x2044
#define TS_ECin5E_DELAY                                     0x2048
#define TS_ECin6E_DELAY                                     0x204C
#define TS_ECtot1P_DELAY                                    0x2050
#define TS_ECtot2P_DELAY                                    0x2054
#define TS_ECtot3P_DELAY                                    0x2058
#define TS_ECtot4P_DELAY                                    0x205C
#define TS_ECtot5P_DELAY                                    0x2060
#define TS_ECtot6P_DELAY                                    0x2064
#define TS_ECtot1E_DELAY                                    0x2068
#define TS_ECtot2E_DELAY                                    0x206C
#define TS_ECtot3E_DELAY                                    0x2070
#define TS_ECtot4E_DELAY                                    0x2074
#define TS_ECtot5E_DELAY                                    0x2078
#define TS_ECtot6E_DELAY                                    0x207C
#define TS_CC1_DELAY                                        0x2080
#define TS_CC2_DELAY                                        0x2084
#define TS_CC3_DELAY                                        0x2088
#define TS_CC4_DELAY                                        0x208C
#define TS_CC5_DELAY                                        0x2090
#define TS_CC6_DELAY                                        0x2094
#define TS_ST1_DELAY                                        0x2098
#define TS_ST2_DELAY                                        0x209C
#define TS_ST3_DELAY                                        0x20A0
#define TS_ST4_DELAY                                        0x20A4
```

```
#define TS_ST5_DELAY                                    0x20A8
#define TS_ST6_DELAY                                    0x20AC
#define TS_ST7_DELAY                                    0x20B0
#define TS_ST8_DELAY                                    0x20B4
#define TS_ST9_DELAY                                    0x20B8
#define TS_ST10_DELAY                                   0x20BC
#define TS_ST11_DELAY                                   0x20C0
#define TS_ST12_DELAY                                   0x20C4
#define TS_ST13_DELAY                                   0x20C8
#define TS_ST14_DELAY                                   0x20CC
#define TS_ST15_DELAY                                   0x20D0
#define TS_ST16_DELAY                                   0x20D4
#define TS_ST17_DELAY                                   0x20D8
#define TS_ST18_DELAY                                   0x20DC
#define TS_ST19_DELAY                                   0x20E0
#define TS_ST20_DELAY                                   0x20E4
#define TS_ST21_DELAY                                   0x20E8
#define TS_ST22_DELAY                                   0x20EC
#define TS_ST23_DELAY                                   0x20F0
#define TS_ST24_DELAY                                   0x20F4
/***********************************************/
/************** END DELAY REGISTERS *************/
/***********************************************/

/***********************************************/
/************** BEGIN SCOPE REGISTERS ***********/
/***********************************************/
#define TS_TRIG_STATUS                                  0x3000
#define TS_TRIG_VALUE3                                  0x3004
#define TS_TRIG_VALUE2                                  0x3008
#define TS_TRIG_VALUE1                                  0x300C
#define TS_TRIG_VALUE0                                  0x3010
#define TS_TRIG_INGORE3                                 0x3014
#define TS_TRIG_INGORE2                                 0x3018
#define TS_TRIG_INGORE1                                 0x301C
#define TS_TRIG_INGORE0                                 0x3020

#define TS_TRIG_BUFFER_DMA                              0x0000
#define TS_TRIG_BUFFER                                  0x7FFC
/***********************************************/
/************** END SCOPE REGISTERS *************/
/***********************************************/

/***********************************************/
/****** BEGIN TRIG PULSE STRETCH REGISTERS ******/
/***********************************************/
/* Notes:
        1) 0->7 value, trigger pulse is extended by: value * 5ns
*/
#define TS_PERSIST_TRIG1                                0x2100
#define TS_PERSIST_TRIG2                                0x2104
#define TS_PERSIST_TRIG3                                0x2108
#define TS_PERSIST_TRIG4                                0x210C
#define TS_PERSIST_TRIG5                                0x2110
#define TS_PERSIST_TRIG6                                0x2114
#define TS_PERSIST_TRIG7                                0x2118
#define TS_PERSIST_TRIG8                                0x211C
#define TS_PERSIST_TRIG9                                0x2120
#define TS_PERSIST_TRIG10                               0x2124
#define TS_PERSIST_TRIG11                               0x2128
#define TS_PERSIST_TRIG12                               0x212C
/***********************************************/
/************** END SCOPE REGISTERS *************/
/***********************************************/

/***********************************************/
```

```
/******** BEGIN SECTOR TRIGGER REGISTERS ********/
/************************************************/
/* Notes:
        1) bit 0->sector 1, bit 1->sector 2, etc...
        2) STS <= ST0 or ST1 or ST2 or ST3
        3) STOF <= (STS or STS_DIS) and (TOF or TOF_DIS) and STOF_EN
        4) ECP <= ECinP and ECtotP
        5) ECE <= ECinE and ECtotE
        6) ECC <= (ECE or ECE_DIS) and (CC or CC_DIS) and ECC_EN
*/
#define TS_STOF_EN                                      0x2300
#define TS_TOF_DIS                                      0x2304
#define TS_STS_DIS                                      0x2308
#define TS_ECE_DIS                                      0x230C
#define TS_CC_DIS                                       0x2310
#define TS_ECC_EN                                       0x2314
/************************************************/
/********* END SECTOR TRIGGER REGISTERS *********/
/************************************************/


/************************************************/
/********* BEGIN MOR TRIGGER REGISTERS **********/
/************************************************/
/* Notes:
        1) bit 0->trigger 1, bit 1->trigger 2, etc...
        2) MOR <= (MORA_EN and MORA) or (MORB_EN and MORB) or MOR_DIS
        3) MOR is used in trigger signal
*/
#define TS_MORA_EN                                      0x2408
#define TS_MORB_EN                                      0x240C
#define TS_MOR_DIS                                      0x2410
/************************************************/
/********** END MOR TRIGGER REGISTERS ***********/
/************************************************/


/************************************************/
/**** BEGIN ST MULTIPLICITY TRIGGER REGISTERS ***/
/************************************************/
/* Notes:
        1) bits 5:9 => ST Multiplicity Max
        2) bits 0:4 => ST Multiplicity Min
*/
#define TS_STMULT_THRESHOLD                             0x2414
#define TS_STMULT_DIS                                   0x2418
/************************************************/
/**** END ST MULTIPLICITY TRIGGER REGISTERS *****/
/************************************************/

/************************************************/
/******* BEGIN L2 SECTOR TRIGGER REGISTERS ******/
/************************************************/
/* Notes:
        1) L2_SECLOGIC is a 3:1 LUT with sector inputs:
           Address Input 0: STOF
                    Address Input 1: ECP
                    Address Inptu 2: ECC
        2) L2OUTPUTDELAY delays L2 10b Latch signals in 5ns
           steps. Values: min. 2*5ns to max. 1023*5ns
        3) L2_OUTPUTWIDTH reshapes width from 1*5ns to 255*5ns
*/
#define TS_L2_SECLOGIC                                  0x2500
#define TS_L2_OUTPUTDELAY                               0x2504
#define TS_L2_OUTPUTWIDTH                               0x2508
/************************************************/
```

```
/******** END L2 SECTOR TRIGGER REGISTERS *******/
/************************************************/


/************************************************/
/************** BEGIN LUT CONTROL REGISTERS *****/
/************************************************/
/* Notes:
  1) See document "HallBTriggerMarch2010_Description.pdf" on how these LUTs are wired for triggers.

     LUT:   LUT Size:

     TS_CFG_LUT_ECC[1-12]    128Word x 32bit -> 4096 x 1bit LUTs
     TS_CFG_LUT_ECP[1-12]    128Word x 32bit -> 4096 x 1bit LUTs
     TS_CFG_LUT_L2                    512Word x 32bit -> 4096 x 4bit LUT

                    ECC LUT Address Mapping (4096 x 1bit):
                    Address Bit                Signal Mapping
                    -----------    --------------
                    0                          ECC Sector 1
                    1                          ECC Sector 2
                    2                          ECC Sector 3
                    3                          ECC Sector 4
                    4                          ECC Sector 5
                    5                          ECC Sector 6
                    6                          STOF Sector 1
                    7                          STOF Sector 2
                    8                          STOF Sector 3
                    9                          STOF Sector 4
                    10                         STOF Sector 5
                    11                         STOF Sector 6

                    ECP LUT Address Mapping (4096 x 1bit):
                    Address Bit                Signal Mapping
                    -----------    --------------
                    0                          ECP Sector 1
                    1                          ECP Sector 2
                    2                          ECP Sector 3
                    3                          ECP Sector 4
                    4                          ECP Sector 5
                    5                          ECP Sector 6
                    6                          STOF Sector 1
                    7                          STOF Sector 2
                    8                          STOF Sector 3
                    9                          STOF Sector 4
                    10                         STOF Sector 5
                    11                         STOF Sector 6

                    L2 LUT Address Mapping (4096 x 4bit):
                    Address Bit                Signal Mapping
                    -----------    --------------
                    0                          Trigger 1
                    1                          Trigger 2
                    2                          Trigger 3
                    3                          Trigger 4
                    4                          Trigger 5
                    5                          Trigger 6
                    6                          Trigger 7
                    7                          Trigger 8
                    8                          Trigger 9
                    9                          Trigger 10
                    10                         Trigger 11
                    11                         Trigger 12

 */
#define TS_CFG_LUT_BASE                                    0x4000
```

```
#define TS_CFG_LUT_ECC1                                 0x4000
#define TS_CFG_LUT_ECC2                                 0x4200
#define TS_CFG_LUT_ECC3                                 0x4400
#define TS_CFG_LUT_ECC4                                 0x4600
#define TS_CFG_LUT_ECC5                                 0x4800
#define TS_CFG_LUT_ECC6                                 0x4A00
#define TS_CFG_LUT_ECC7                                 0x4C00
#define TS_CFG_LUT_ECC8                                 0x4E00
#define TS_CFG_LUT_ECC9                                 0x5000
#define TS_CFG_LUT_ECC10                                0x5200
#define TS_CFG_LUT_ECC11                                0x5400
#define TS_CFG_LUT_ECC12                                0x5600
#define TS_CFG_LUT_ECP1                                 0x5800
#define TS_CFG_LUT_ECP2                                 0x5A00
#define TS_CFG_LUT_ECP3                                 0x5C00
#define TS_CFG_LUT_ECP4                                 0x5E00
#define TS_CFG_LUT_ECP5                                 0x6000
#define TS_CFG_LUT_ECP6                                 0x6200
#define TS_CFG_LUT_ECP7                                 0x6400
#define TS_CFG_LUT_ECP8                                 0x6600
#define TS_CFG_LUT_ECP9                                 0x6800
#define TS_CFG_LUT_ECP10                                0x6A00
#define TS_CFG_LUT_ECP11                                0x6C00
#define TS_CFG_LUT_ECP12                                0x6E00

#define TS_CFG_LUT_L2                                   0x7000
/***********************************************/
/*************** END LUT CONTROL REGISTERS ******/
/***********************************************/

#endif
```