Table of Contents

1. The Trigger Supervisor - Overview

1.1 Introduction

The trigger supervisor is the interface between the experiment specific triggering system and the data acquisition system. Acting as the central control point for acquisition activity, the CEBAF trigger supervisor performs the following functions:

(1) accepts and prescales multiple sources of triggers, both physics and calibration types,
(2) maintains system busy while an input trigger is being processed,
(3) generates signals for gating and timing of front end electronics,
(4) couples the levels of multi-level trigger systems,
(5) communicates triggering information to the system's read out controllers.

A diagram of the Trigger Supervisor (TS) and its connections to the trigger system, front end electronics, and readout controllers (ROC) are shown in Figure 1.1. The TS allows for three logical trigger levels that can be associated with three stages of gating of the front end electronics. (Any number of physical trigger levels may make up a logical level.)

Up to 12 independent level 1 trigger streams can be accepted simultaneously. The level 1 triggers can be required to be in coincidence with a common strobe. Each trigger input can be individually disabled within the TS. Eight of these inputs are prescalable. Inputs 1-4 can be prescaled by up to a factor of 1 million, while inputs 5-8 can be prescaled by up to 16K. Each input is programmed as one of 3 possible trigger classes:

Class 1 - no higher level trigger decision is required to have event readout. Level 2 and level 3 trigger logic is not started. For calibration type triggers, or systems with no higher level trigger logic.

Class 2 - only level 2 pass is required to have event readout. Level 3 trigger logic is not started. For triggers not requiring a 3rd level of logic.

Class 3 - both level 2 and level 3 pass are required for readout. For complex physics triggers.

1.2 TS Operation

We now discuss the operation of the TS for a class 3 trigger. Differences for class 1 and class 2 will be described after this.

The TS is ready to accept triggers when all of the following conditions are satisfied:

- it is user enabled,
- no TS cycle is currently active,
- no input trigger is currently latched,
- "front end busy" is not asserted, and
- "external inhibit" is not asserted.

When the TS is ready to accept triggers a "lev 1 trig" that passes through the prescale circuitry will lock up the TS, latch the values of the trigger lines, and generate up to 8 simultaneous "lev 1 accept" signals out. The pattern of "lev 1 accept" signals asserted are a function of the latched trigger pattern, with this dependence being programmed by the user. The accept signals asserted will have a precise time relationship to the input trigger, and will be used by external gate circuitry to generate ADC gates and TDC starts/stops. At the same time the TS will issue "lev 2 start" signal to begin the level 2 trigger system. The TS will wait for a pass or fail response from this trigger system. If a "lev 2 fail" is returned, the TS will issue a "clear" signal, which will be used to generate (externally) a fast clear to the front end electronics. In this case, the TS will remained locked up until the front end is no longer busy ("front end busy"). Then a new level 1 trigger may be accepted.

If a "lev 2 pass" is returned instead, a "lev 2 accept" signal is issued by the TS along with "lev 3 start". "Lev 2 accept" is used to generate additional gating to the front end (e.g. Digitize data), and "lev 3 start" initiates level 3 logic. The TS will wait for a pass or fail response from this trigger system. If "lev 3 fail" is returned, the TS will issue "clear" and remain locked up until "front end busy" is no longer asserted. Then a new level 1 trigger may be accepted by the TS.

If "lev 3 pass" is returned the event is to be read out. "Lev 3 accept" is issued by the TS to generate additional gating signals if required (e.g." buffer data"). Trigger information for the event must be made available to the read out controllers. Like data from the front end modules, this trigger data is presented to the ROC's in a buffered fashion. The TS supports a total of 32 ROC's on its 4 independent ROC branches (see Fig 1.1). Each branch consists of an 8-deep buffer memory (FIFO), buffer counter, and read sequencer located on the TS, with an external cable that links up to 8 ROC's. The trigger information carried along these branches is in the form of a 4-bit readout code determined by the latched trigger pattern, with the code programmed by the user. Thus up to 16 independent readout control functions are permitted. Two additional data bits that will be described later also appear on the branch cable for the event.

When "front end busy" is no longer asserted, the TS will load the code for the event into the next available location in each of the buffers. All of the buffer counters are incremented upon this load. If none of the buffers is now full (buffer count = 8), the TS drops "lev 1 accept", "lev 2 accept", "lev 3 accept", and re-arms itself to accept new level 1 triggers. Otherwise the TS will hold its state and wait for space to become available in all buffers before dropping the accept signals and re-arming itself.

The following describes the communication of trigger information on one of the branches. The same activity occurs concurrently and independently on all 4 branches. Once the read sequencer on the branch notices that its buffer is not empty (buffer count > 0), the transmission of trigger information for the event to the ROC's along the branch begins. The read sequencer places the readout code from the first valid buffer location onto data lines "data(2-5)" and strobes these with "strobe". The ROC's along the branch receive this data and proceed to read out the event fragments according to the function defined by the readout code. Each ROC is assigned a unique "ack" line on the branch cable. When a ROC on the branch is finished processing the event it asserts its "ack" line. Upon receipt of "ack" from all ROC's on the branch, the read sequencer resets "data(2-5)", drops "strobe", and decrements its buffer counter. Each ROC lowers "ack" upon detecting the negation of "strobe". The read sequencer cycle is completed when it detects that "ack" has been lowered by all ROC's on the branch. These cycles on the branch will continue as long as there is valid trigger data available in the buffer.

For class 1 and class 2 trigger types the activity of the TS is somewhat different. For class 1, no higher level trigger decision is required so the "lev 2 start" and "lev 3 start" signals are never issued. The TS must generate the signals "lev 2 accept" and "lev 3 accept" that may be necessary for front end gating. The delays for these signals (after "lev 1 accept") are programmable and are loaded into the TS at run start time. Once "front end busy" is no longer asserted, the TS will behave as described earlier. The TS will load the code for the event into the next available location in each of the branch buffers. All of the buffer counters are incremented upon this load. If none of the buffers is now full, the TS drops "lev 1 accept", "lev 2 accept", "lev 3 accept", and re-arms itself to accept new level 1 triggers. Otherwise the TS will hold its state and wait for space to become available in all buffers before dropping the accept signals and re-arming itself.

For class 2 triggers the TS generates "lev 2 start" and waits for the level 2 trigger decision. If "lev 2 fail" is returned the TS asserts "clear" and gets itself ready for a new trigger. If "lev 2 pass" is returned the TS issues "lev 2 accept", but not "lev 3 start". The signal "lev 3 accept" is generated by the TS at a programmed time as described for class 1 triggers. Once "front end busy" is no longer asserted, the TS will behave as described earlier. The TS will load the code for the event into the next available location in each of the branch buffers. All of the buffer counters are incremented upon this load. If none of the buffers is now full, the TS drops "lev 1 accept", "lev 2 accept", "lev 3 accept", and re-arms itself to accept new level 1 triggers. Otherwise the TS will hold its state and wait for space to become available in all buffers before dropping the accept signals and re-arming itself.

For systems that exclusively have front end modules with nobuffering capability, the depth of the TS branch buffers can be set to 1. In this situation the TS will be able to accept new triggers only when the previous event has been completely read out.

The TS also supports a mixed system of buffered and non-buffered front end modules. One branch (branch 4) can be set to have a buffer depth of 1 independent of the other branches. All non-buffered front end modules must have their ROC reside on branch 4.

1.3 Synchronization

In our system the ROC's will be reading data from front end module buffers and may be several events behind the trigger. Only the position in the buffers link the pieces of data as an event. There lies a danger that if a hardware error occurs resulting in a misalignment of data (e.g. a module misses a gate), subsequent events will be corrupted.

To avoid this potentially large loss of data the TS-ROC system is designed to periodically test for synchronization automatically. This is done as follows. The TS will be programmed with a sync count at run start time. When the TS has accepted this number of events, it asserts and writes the sync bit along with the readout code to the branch buffers. As long as the sync bit remains asserted the TS will hold its state, unable to accept new triggers. The read sequencers on the branches continue to send trigger information for events that remain in the branch buffers, and the ROC's continue to read the event fragments from the front end modules in normal fashion. Just as always, a "no data" response from any front end module indicates loss of synchronization, and the block of events since the last successful synchronization must be noted as corrupt. The last event in all buffers is the synchronization event, and the sync bit appears as "data(0)" on the branch cable when the read sequencer reaches this event. The ROC's read this event from the front end modules as normal, but do not yet respond with "ack". At this point all front end module buffers should be empty. Each ROC attempts at least one more event read to assure that no additional data is found, also indicative of a synchronization problem. When this process and any bookkeeping associated with it is completed, the ROC finally issues "ack". When all ROC's have responded, the sync bit and the sync counter are reset, and the TS cycle is completed with the re-arming of triggers.

The synchronization can also be forced to be done at any time by the user. When the request is made the TS disables new triggers and waits for any current cycle to finish. Then the TS establishes a zero readout code as a tag, and asserts and writes the sync bit along with the readout code to the branch buffers. The read sequencers on the branches continue to send trigger information for events that remain in the branch buffers, and the ROC's continue to read the event fragments from the front end modules in normal fashion. When the ROC's recognize the zero readout code along with "data(0)" asserted, they know that the front end module buffers should be empty. As for the scheduled synchronization described above, each ROC attempts at least one more event read to assure that no additional data is found. When this process and any bookkeeping associated with it is completed, the ROC finally issues "ack". When all ROC's have responded, the sync bit and the sync counter are reset, and the user requested synchronization cycle is completed with the re-arming of triggers.

The ability to synchronize may be disabled by the user.

1.4 Compatability with existing front end modules

The basic scheme illustrated in Figure 1 assumes that the front end modules will be told explicitly when to load the digitized results into its buffer. However, some commercially available buffered front end modules do not operate in this way. Consider as an example the LeCroy 1882F FASTBUS ADC. When a gate is issued to this device it will sample and hold the analog input. The hold time is programmable, and when it elapses the data will be digitized and loaded into the buffer. Only a "clear" signal during the hold time will successfully purge the data. The "clear" issued by the TS results from a "lev 2 fail" or "lev 3 fail" received from the trigger system. If such a fail signal comes after the hold interval elapses the data is destined for the buffers, and the TS must declare this event as accepted to the ROC.

To do this the TS has a programmable timer. Started on the issuance of "lev 1 accept", the TS will behave as described earlier as long as this timer has not expired. If this period elapses before the highest level decision required of the trigger occurs, "clear" is disabled and the TS continues to wait for a decision. If it is a "pass" there is no difference from what was described earlier. If it is "fail" the TS tags the event as a late failure by asserting and writing a late fail bit along with the readout code to the branch buffers. This bit appears as "data(1)" on the branch cable when the read sequencer reaches this event in the buffer. The ROC's can use the late fail tag to flush the data for this event from the front end module buffers if desired.

This timer may be disabled by the user.

1.5 Simultaneous triggers

The case may arise where the latched trigger pattern indicates several triggers of different classes have occurred simultaneously. The TS must be able to decide what sequence to perform in this case. This is accomplished using the trigger class memory bits. Acting as a look-up table the memory is addressed by the latched trigger pattern, and the user programmed output bits will select the action of the TS for this pattern. For unacceptable trigger patterns a fast trigger clear feature will reset the TS with no issuance of "lev 1 accept".

With this solution the user has full control over the action of the TS for any trigger pattern.

Table 1.1 - Trigger Supervisor Vital Statistics

Trigger inputs		12
Prescalable inputs		8
Prescale factor:	inputs 1-4	1 to 2^{24}
	inputs 5-8	1 to 2^{16}
Prescale input bandwidth		> 50 Mhz
Minimum input pulse width		4 ns
Insertion time		40 ns (min)
Variation of insertion time across input channels		< 0.5ns
Time jitter of L1 Accept (relative to trigger in)		< 40 ps (RMS)
Simultaneous tigger resoulution (user selectable)		10 ns (min)
Maximum event rate		3 Mhz
Programming interface		VME A24/D32 slave
Base address (programmable IC)		(hex) ED0000
Form factor		'D' size VXI module (340 x 367 mm)

2. Summary

2.1 Summary of Input/Output Signals

Figure 2.1 identifies the trigger supervisor front panel connectors. Connectors A and B mate with 50 ohm ribbon coax cables. Tables 2.1 and 2.2 identify the signals carried and their pin assignments. Connectors C,1,2,3,4 mate with 100 ohm twisted pair ribbon cable. Tables 2.3 and 2.4 identify the signals carried and their pin assignments.

Signals are defined to be ECL levels (HIGH = -0.9 V, LOW = -1.75 V) and are considered asserted when HIGH.

Single-ended ECL signals are transmitted referenced to ground. Inputs are terminated in 50 ohms to -2V, and outputs should be terminated at their destination in the same fashion.

For differential RS-485 signals both the signal (Q) and its complement (/Q) are transmitted. Differential inputs are terminated passively with 100 ohms between Q and /Q. Outputs should be terminated at their destination in the same fashion.

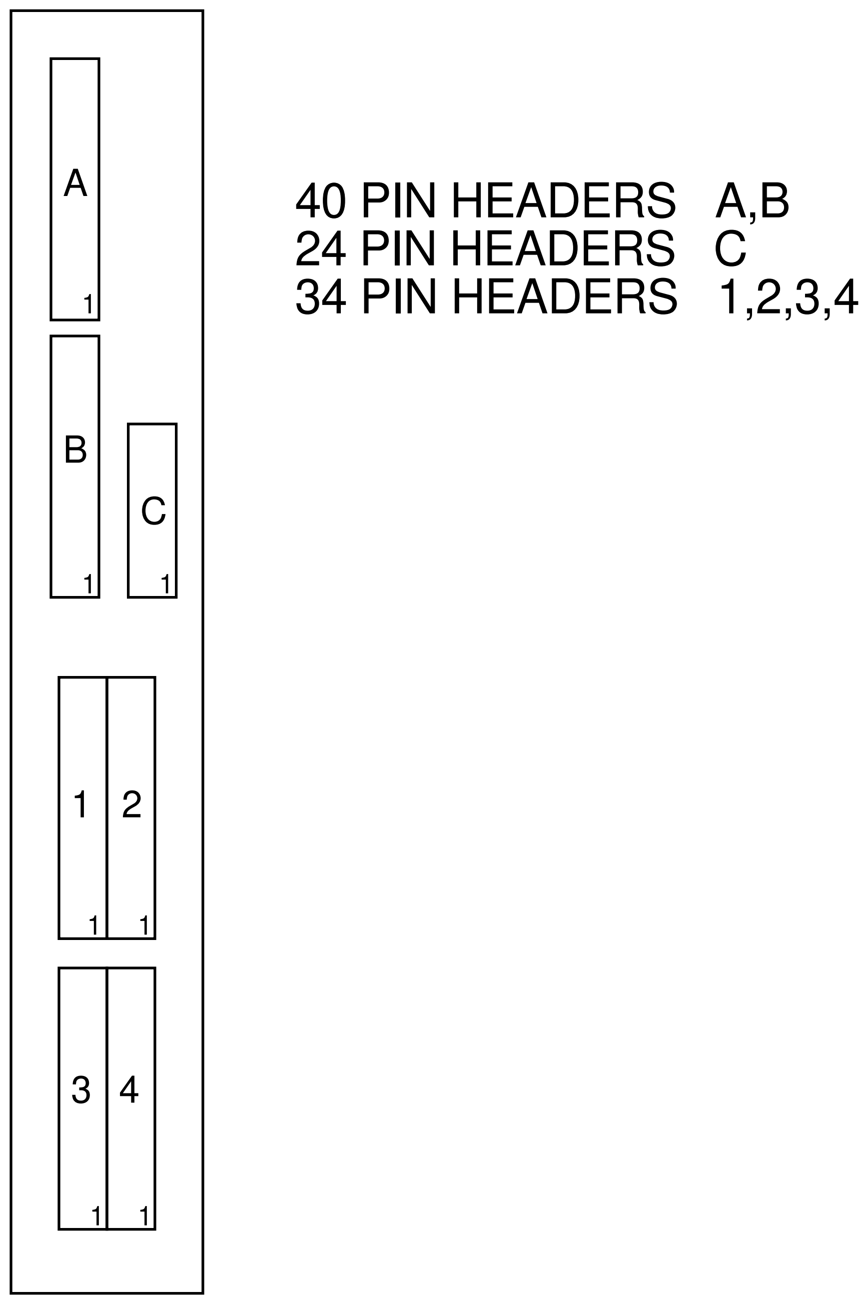

Figure 2.1 - Trigger Supervisor Front Panel Connectors

Table 2.1 Connector A signal definition

- All signals are single ended ECL
- All ODD # pins are signal ground

Signal name	Direction	Pin #
Common Strobe	Input	40
Trigger 1	Input	38
Trigger 2	Input	36
Trigger 3	Input	34
Trigger 4	Input	32
Trigger 5	Input	30
Trigger 6	Input	28
Trigger 7	Input	26
Trigger 8	Input	24
Trigger 9	Input	22
Trigger 10	Input	20
Trigger 11	Input	18
Trigger 12	Input	16
Level 2 Pass	Input	12
Level 2 Fail	Input	10
Level 3 Pass	Input	8
Level 3 Fail	Input	6
Front End Busy	Input	4
External Inhibit	Input	2

Table 2.2 Connector B signal definition

- All signals are single ended ECL
- All ODD # pins are signal ground

Signal name	Direction	Pin #
Level 1 Accept (1)	Output	40
Level 1 Accept (2)	Output	38
Level 1 Accept (3)	Output	36
Level 1 Accept (4)	Output	34
Level 1 Accept (5)	Output	32
Level 1 Accept (6)	Output	30
Level 1 Accept (7)	Output	28
Level 1 Accept (8)	Output	26
Level 1 OK	Output	24
Level 2 Accept	Output	22
Level 3 Accept	Output	20
Clear	Output	18
Level 2 Start	Output	16
Level 3 Start	Output	14
Enable level 1	Output	12
TS Live	Output	10
TS Busy	Output	8
OR Level 1	Output	6

Table 2.3 Connector C signal definition

- All signals are differential ECL

Signal name (Q)	Direction	Pin # (Q,/Q)
Prescaled Trigger 1	Output	1,2
Prescaled Trigger 2	Output	3,4
Prescaled Trigger 3	Output	5,6
Prescaled Trigger 4	Output	7,8
Prescaled Trigger 5	Output	9,10
Prescaled Trigger 6	Output	11,12
Prescaled Trigger 7	Output	13,14
Prescaled Trigger 8	Output	15,16
Prescaled Trigger 9	Output	17,18
Prescaled Trigger 10	Output	19,20
Prescaled Trigger 11	Output	21,22
Prescaled Trigger 12	Output	23,24

Table 2.4 Connectors 1,2,3,4 signal definition

- All signals are differential RS-485

- The four connectors represent the four ROC branches, all having the same basic signal assignments shown below (ROC n represents the nth ROC on the given branch)

Signal name (Q)	Direction	Pin # (Q,/Q)
Strobe	Output	1,2
Sync	Output	3,4
Late Fail	Output	5,6
ROC Code Bit 0	Output	7,8
ROC Code Bit 1	Output	9,10
ROC Code Bit 2	Output	11,12
ROC Code Bit 3	Output	13,14
ROC 0 Acknowledge	Input	19,20
ROC 1 Acknowledge	Input	21,22
ROC 2 Acknowledge	Input	23,24
ROC 3 Acknowledge	Input	25,26
ROC 4 Acknowledge	Input	27,28
ROC 5 Acknowledge	Input	29,30
ROC 6 Acknowledge	Input	31,32
ROC 7 Acknowledge	Input	33,34

2.2 <u>Summary of Trigger Supervisor Registers</u>

The following diagrams identify the registers of the device and serve as a reference to the programmer. The local address of each register is given, and all register bits are identified. A more complete discussion of the use of the registers is contained in a subsequent section (Section 3).

The absolute address of a register is

ADDRESS = LOCAL ADDRESS + BASE ADDRESS.

The base address of the board as supplied is (hex) ED0000. This can only be changed by replacing a socketed Programmable Logic Device (PLD) on the board.

3. The Trigger Supervisor Registers

The Trigger Supervisor is programmed by the user through VME bus protocols (ANSI/IEEE STD1014-1987). The device meets all VME bus standards except in terms of board size. The form factor is that of a 'D' size (340 x 367 mm) VXI module (VME extension for instrumentation). This configuration allows the device to plug directly into the VXI based trigger system of one experimental hall as well as into standard 32-bit VME backplanes (with suitably modified card cage) of other experimental facilities.

Officially the trigger supervisor is categorized as an A24, D32 VME bus slave. All storage locations can be accessed as both Standard Supervisory and Standard Nonprivileged data. The base address as supplied is ED0000 (hex), which can be modified by replacing a socketed Programmable Logic Device (PLD) on the board.

A brief discussion of the types of register access through VME is presented here because it impacts of the correct use of the trigger supervisor. The user is referred to to full VME bus specification for a complete description. The smallest addressable unit of storage is the byte location, to which a unique binary address is assigned. There are four categories of bytes distinguished by the least significant two bits of their address:

```
xx...x00     BYTE(0)
xx...x01     BYTE(1)
xx...x10     BYTE(2)
xx...x11     BYTE(3).
```

A set of byte locations differing only in the two least significant bits defines a four byte BYTE(0-3) group. The VME data transfer protocols allow a master to access 1,2,3 or 4 byte locations from the same group simultaneously:

```
Single byte access   - BYTE(0), BYTE(1), BYTE(2), BYTE(3)
Double byte access   - BYTE(0-1), BYTE(1-2), BYTE(2-3)
Triple byte access   - BYTE(0-2), BYTE(1-3)
```

```
Quad byte access     - BYTE(0-3).
```

For example, the 32-bits of data stored in a 4-byte group of a D32 slave may be accessed by a D32 master in a single bus cycle via the BYTE(0-3) transfer, or in 4 bus cycles via the BYTE(n) transfer. However, a 16-bit master would require a minimum of 2 bus cycles (BYTE(0-1) & BYTE(2-3)) to accomplish the same data transmission.

We now turn to a general discussion of the trigger supervisor's registers in the above terminology. Each individual register or memory location of the trigger supervisor is defined to be a 4-byte group. This is somewhat inefficient usage of address space as the actual storage locations contain from 1 to 4 bytes of data. However, the decision results in a simplification of board design and possibly in programming (since all of the above access types are legal for all locations). The local address given in this document for each register is the address of BYTE(0) for that register. Data is stored in the 4-byte registers as follows:

```
D(31)-D(24)     BYTE(0)
D(23)-D(16)     BYTE(1)
D(15)-D(08)     BYTE(2)
D(07)-D(00)     BYTE(3).
```

Certain considerations concerning a register's function must be understood by the user before employing the different access schemes described above. For example, data written to some registers has to be loaded into counters. For simplicity we have coupled this loading action with the actual write of the register. But as discussed above, the entire data word may be assembled as several writes by user choice or hardware constraints. We therefore adopt the convention that only a write of BYTE(3) triggers such a loading process. This requires that BYTE(3) must included in the group of bytes that is written last.

Similarly, some data to be read is dynamic in nature. Such data must be latched as a unit and then read. For simplicity we have coupled the latching action with the actual read of the register. But as discussed above, the full data word may be assembled only after several reads. We therefore adopt the convention that such dynamic data is latched only upon a read of BYTE(3). Reading of the remaining bytes causes no latching action to occur. This requires that BYTE(3) must be included in the group of bytes that is read first.

Of course, as long as the mode of access includes all bytes of interest, the above requirements are automatically met.

Many of the trigger supervisor's registers are designated as protected against writes while the device is active. The trigger supervisor is defined to be active if any of the following conditions hold:

- the GO bit of the Control/Status Register (CSR) is set,
- the TS main sequencer is active,
- the TS user synchronization sequencer is active,
- any of the 4 ROC buffer read sequencers is active.

The write protection prevents the user from changing the TS operating conditions while events are being processed. A status bit in the CSR is set if such a write is attempted.

The trigger supervisor's memory is also protected against reads while the device is active. Reading the memory while the device is active would interfere with event driven memory accesses. A status bit in the CSR is set if such a read is attempted.

Below we describe in detail the registers of the trigger supervisor.

3.1 CONTROL/STATUS REGISTER (CSR)

As the name suggests, the Control/Status register is the user's main command point for the trigger supervisor activities. It is used to start and stop the run, initialize the device, as well as configure most of the trigger supervisor's operating options.

The 32-bit register is designed so that individual bits may be set or cleared without the read-modify-write process. Writing 1 to bit N (N=0,...,13) sets the function associated with that bit, while writing 1 to bit N+16 clears that same function.

A bit-by-bit description follows. Bits are read/write unless otherwise indicated.

(0) GO - setting this bit starts the run. When asserted the prescalers become operational and the TS is enabled to latch triggers. Clearing the bit (e.g. writing 1 to bit 16) allows no more triggers in, but the sequencing activity of the TS will continue until completed. The GO bit may also be cleared by other control bits (see bits (1),(2)).

(1) PAUSE ON NEXT SYNC - setting this bit arms the TS to stop the run when the next scheduled synchronization event occurs. When the sync occurs both the GO bit and this bit (1) are cleared.

(2) PAUSE AND SYNC - setting this bit starts the "user" request synchronization sequencer" (see bit (3)). This holds off latched triggers until completed, at which time the GO bit and this bit (2) are cleared. The effect is to force a synchronization and immediately end/pause the run.

(3) FORCE SYNC - setting this bit starts the "user request synchronization sequencer". This sequencer holds off latched triggers and waits until any current trigger has been processed. It then inserts a "user sync" event (SYNC = 1, ROC code = 0000) into the ROC branch buffers. The sequencer then waits until all 4 ROC branch buffers are empty, signifying that the synchronization task has been completed. The sequencer then resets this bit (3) and re-enables triggers to be latched.

(4) ENABLE SYNC - setting this bit enables synchronizations (both scheduled and user initiated) to occur.

(5) ENABLE LEV 1 HARDWARE - setting this bit can be used to enable the external Level 1 trigger systems, if it is required. (Drives pin 12 of connector B)

(6) USE CLEAR PERMIT TIMER - setting this bit allows the CLEAR PERMIT TIMER to become operational.

(7) USE FRONT BUSY TIMER - setting this bit allows the FRONT BUSY TIMER to become operational.

(8) USE CLEAR HOLD TIMER - setting this bit allows the CLEAR HOLD TIMER to become operational.

(9) ROC LOCK - setting this bit defines the ROC buffer depth on all 4 ROC branches to be 1. This effectively defeats the buffering capability of the system.

(10) ROC LOCK 4 - setting this bit defines the ROC buffer depth on ROC branch 4 to be 1, while not affecting the other 3 branches.

(11) OVERRIDE INHIBIT - setting this bit does not allow the EXTERNAL INHIBIT signal to prevent triggers from being latched.

(12) HOLD STATE - setting this bit freezes the TS in its current state. It is useful for debugging, especially in conjunction with the SINGLE STEP mode of the TEST register.

(13) TEST MODE - setting this bit enables the test features of the TS. External trigger inputs are blocked and simulated input signals can be generated by writing to the TEST register.

(14) RESET - (Write only) asserting this bit generates a pulse that resets all state sequencers and counters. The contents of most register bits are left intact. CSR bits 0,1,2,3 are cleared.

(15) INITIALIZE - (Write only) asserting this bit generates a pulse that resets all state sequencers, counters, and registers.

(31) CLEAR LATCHED STATUS - (Write only) asserting this bit clears the LATCHED STATUS bits.

The following are Read only LATCHED STATUS bits. These bits indicate that certain conditions have occurred since the GO bit has been asserted (or since the last CLEAR LATCHED STATUS has been invoked).

(16) INHIBIT OCCURRED - at least one EXTERNAL INHIBIT signal has been received.

(17) LATE FAIL OCCURRED - at least one higher level trigger fail has been received outside the time window set by the CLEAR TIMER.

(18) SYNC OCCURRED - at least one SYNCHRONIZATION (scheduled or user requested) has occurred.

(19) WRITE ACCESS ERROR OCCURRED - at least one attempt has been made to write to a protected register or memory while the TS is operational.

(20) READ ACCESS ERROR OCCURRED - at least one attempt has been made to read the memory while the TS is operational.

(21) VME CYCLE ERROR OCCURRED - at least one VME cycle error (handshake lock failure) has occurred.

3.2 <u>TRIGGER CONTROL REGISTER</u>

The TRIGGER CONTROL REGISTER allows the user to enable the individual trigger inputs, and to define whether the enabled inputs must be in coincidence with a common strobe. This register is protected from writes while the TS is active (except when the TEST MODE bit of the CSR is asserted).

All bits are Read/Write.

(0) COMMON STROBE MODE - when set the overlap of the COMMON STROBE with an enabled TRIGGER input becomes the effective trigger input for that channel.

(1),...,(12) INPUT ENABLES - when set they enable level 1 trigger inputs (1),...,(12).

(15) OPEN PRESCALES - setting this bit allows the prescalers to count without the CSR GO bit being asserted. This is an alternate and valid way of running during data taking. It is also useful in determining prescale factors for the input channels, as rate measurements may be made using the onboard scaler before a run begins.

3.3 TRIGGER PRESCALE REGISTERS (8)

The TRIGGER PRESCALE REGISTERS allow the user to fix the scaledown coefficients for the 8 prescalable trigger inputs. PRESCALE REGISTERS 1-4 define 20-bit prescale factors, while PRESCALE REGISTERS 5-8 define 14-bit prescale factors.

The scaledown coefficient for each input is given by:

```
SCALEDOWN = 1 / ( PRESCALE FACTOR + 1 ).
```

All bits of the registers are Read/Write. These registers are protected from writes while the TS is active. The prescale counter for a given channel is loaded when the low order byte of the PRESCALE REGISTER is written to. All counters are loaded upon the assertion of the CSR RESET bit.

3.4 ROC ENABLE REGISTER

The ROC ENABLE REGISTER allows the user to define the configuration of Read Out Controllers (ROC) used in the system. This is necessary as the TS must know which ROC's on each branch are required to respond to its actions. The four ROC branches each support up to eight ROC's (0,...,7).

All bits of the 32-bit register are Read/Write. This register is protected from writes while the TS is active.

```
(0)-(7)   ENABLE ROC 0-7 on branch 1

(8)-(15)  ENABLE ROC 0-7 on branch 2

(16)-(23) ENABLE ROC 0-7 on branch 3

(24)-(31) ENABLE ROC 0-7 on branch 4
```

3.5 SYNCHRONIZATION REGISTER

The SYNCHRONIZATION REGISTER allows the user to define the frequency of scheduled synchronizations.

The 16-bit register is Read/Write. This register is protected from writes while the TS is active.

(0),...,(15) SYNC INTERVAL - the number of events between scheduled synchronizations.

3.6 TIMER REGISTERS (5)

The TIMER REGISTERS and their associated timers allow the user to specify time delays or time windows during which some action may occur.

TIMER REGISTERS (1)-(4) are 16-bits, with each count representing 40 ns. TIMER REGISTER (5) is 8-bits, with each countrepresenting 20 ns. All TIMER REGISTERS are Read/Write. These registers are protected from writes while the TS is active.

(1) CLEAR PERMIT TIMER - this timer is optionally used to define a time window in which a FAIL response from a higher level trigger results in a CLEAR being issued by the TS.

(2) LEVEL 2 TIMER - this timer is required for use by Class 1 triggers to define a time delay between the level 1 accept and level 2 accept signals generated by the TS.

(3) LEVEL 3 TIMER - this timer is required for use by Class 1 and Class 2 triggers to define a time

delay between the level 1 accept and level 3 accept signals generated by the TS.

(4) FRONT BUSY TIMER - this timer is optionally used to define a time period that can represent the conversion or dead time of the front end module inputs.

(5) CLEAR HOLD TIMER - this timer is optionally used to define a time period during which the TS generated CLEAR signal will be held asserted. This time can represent the clearing time of the front end modules.

3.7 MEMORY

The 4096 x 16-bit MEMORY allows the user to define the trigger class, ROC code, and level 1 accept output pattern for any input pattern of level 1 triggers (after prescale).

The local MEMORY address is related to the input trigger pattern as follows:

```
ADDRESS = 4 * TRIGGER PATTERN + 4000      (all hex)
```

where TRIGGER PATTERN (0)-(11) are level 1 trigger inputs (1)-(12).

All bits of the MEMORY are Read/Write. The memory is protected from reads and writes while the TS is active.

(0) LEVEL 1 OK - the input trigger pattern is acceptable and the level 1 accept signals will be issued as programmed. The main TS operating sequence will be started. Otherwise a fast reset of the TS front end logic will occur.

(1) CLASS 1 TRIGGER - the trigger pattern is to be treated as a Class 1 trigger (i.e. no higher level trigger decision is required for event readout).

(2) CLASS 2 TRIGGER - the trigger pattern is to be treated as a Class 2 trigger (i.e. only trigger level 2 pass is required for event readout).

(3) CLASS 3 TRIGGER - the trigger pattern is to be treated as a Class 3 trigger (i.e. both trigger levels 2 and 3 pass are required for event readout).

(4)-(7) ROC CODE - Read Out Controller code bits (0)-(3) defining the ROC function.

(8)-(15) LEVEL 1 ACCEPT - pattern of signals level 1 accept (1)-(8) to be asserted when LEVEL 1 OK is asserted.

3.8 SCALER REGISTERS (2)

The SCALER REGISTERS are two 32-bit Read/Write registers that are used in conjunction with the two onboard 32-bit scalers. Reading the low order byte (bits 0-7) of a SCALER REGISTER latches the current count of the associated scaler into the register. Writing to the low order byte of a SCALER REGISTER clears the associated scaler count.

Scaler_0 counts events to be read out, while Scaler_1 is multiplexed across several signals. The signal assigned to Scaler_1 is selected by the SCALER_1 ASSIGN REGISTER.

Each SCALER REGISTER can accessed by two independent addresses. Let A=(A0,A1) and B=(B0,B1) define two address setsthat each access SCALER REGISTERS 0 and 1. When using the addresses of set A the scalers are independently latched (read) or cleared (write). When using the addresses of set B the scalers are simultaneously latched or cleared by read/write operations on Scaler_0.

3.9 SCALER_1 ASSIGN REGISTER

The SCALER_1 ASSIGN REGISTER defines which signal will be counted by Scaler_1.

This is a single byte Read/Write register. The assignment data is as follows (hex):

```
00 = OR PRESCALED TRIGGER
01-0C = PRESCALED TRIGGER 1-12
0D = LATCHED TRIGGER
0E = LEVEL 1 ACCEPT
0F = FAST RESET
1x = LEVEL 2 FAIL
2x = LEVEL 3 FAIL
3x = LATE FAIL
5x = SYNC.
```

3.10 TEST REGISTER

The TEST REGISTER allows the user to simulate a set of input signals to test the operation of the TS and other components of the data acquisition system. All test signals are generated by writing to the test register while the TEST MODE bit of the CSR is asserted. The TEST MODE bit blocks the appropriate external input signals.

The TEST REGISTER is 16-bits. Bits 0-7 are Write only, while bits 8-15 are Read/Write. A bit-by-bit description follows.

(0) LEVEL 1 PASS - writing this bit generates a pulse that simulates level 1 input triggers. The pattern of level 1 input triggers produced will be identical to the pattern of inputs enabled by the TRIGGER CONTROL REGISTER.

(2) LEVEL 2 PASS - writing this bit generates a pulse that simulates the level 2 pass input signal.

(3) LEVEL 3 PASS - writing this bit generates a pulse that simulates the level 3 pass input signal.

(4) LEVEL 2 FAIL - writing this bit generates a pulse that simulates the level 2 fail input signal.

(5) LEVEL 3 FAIL - writing this bit generates a pulse that simulates the level 3 fail input signal.

(7) SINGLE STEP - while the HOLD bit of the CSR is set, a write to this bit enables a single positive edge of the system clock (50 Mhz) to be applied to all synchronously clocked components. Thus the activity of the TS and its state transitions may be observed in a controlled fashion.

(8) FRONT BUSY - setting or clearing this bit simulates the condition of the input signal front end busy.

(9) EXTERNAL INHIBIT - setting or clearing this bit simulates the condition of the input signal external inhibit.

(10) BRANCH 1 ROC ACKNOWLEDGE - setting or clearing this bit simulates the collective condition of the ROC acknowledge signals on ROC branch 1. Bit (14) of

this register must be asserted to obtain this action.

(11) BRANCH 2 ROC ACKNOWLEDGE - setting or clearing this bit simulates the collective condition of the ROC acknowledge signals on ROC branch 2. Bit (14) of this register must be asserted to obtain this action.

(12) BRANCH 3 ROC ACKNOWLEDGE - setting or clearing this bit simulates the collective condition of the ROC acknowledge signals on ROC branch 3. Bit (14) of this register must be asserted to obtain this action.

(13) BRANCH 4 ROC ACKNOWLEDGE - setting or clearing this bit simulates the collective condition of the ROC acknowledge signals on ROC branch 4. Bit (14) of this register must be asserted to obtain this action.

(14) ENABLE ROC TEST ACKNOWLEDGE - setting this bit enables the action of the ROC test acknowledge signals generated by bits (10)-(13). With this bit cleared the external ROC acknowledge signals are enabled according to the data contained in the ROC ENABLE REGISTER.

3.11 STATE REGISTER

The STATE REGISTER allows the user to observe the state of selected Trigger Supervisor signals. When used in conjunction with the TEST REGISTER, programmed controlled diagnostic sequences may be executed. The STATE REGISTER is also useful in tracing errors that hang up the Trigger-TS-ROC system.

The 32-bit STATE REGISTER is Read only. A bit-by-bit description follows:

(0) LEVEL 1 ACCEPT (SYNC) - the level 1 accept signal synchronized to the 50 Mhz system clock.

(1) LEVEL 2 START - the level 2 start signal.

(2) LEVEL 2 PASS (SYNC) - the latched level 2 pass signal synchronized to the 50 Mhz system clock.

(3) LEVEL 2 FAIL (SYNC) - the latched level 2 fail signal synchronized to the 50 Mhz system clock.

(4) LEVEL 2 ACCEPT - the level 2 accept signal.

(5) LEVEL 3 START - the level 3 start signal.

(6) LEVEL 3 PASS (SYNC) - the latched level 3 pass signal synchronized to the 50 Mhz system clock.

(7) LEVEL 3 FAIL (SYNC) - the latched level 3 fail signal synchronized to the 50 Mhz system clock.

(8) LEVEL 3 ACCEPT - the level 3 accept signal.

(9) CLEAR - the clear signal.

(10) FRONT END BUSY (SYNC) - the front end busy signalsynchronized to the 50 Mhz system clock.

(11) EXTERNAL INHIBIT (SYNC) - the external inhibit signal synchronized to the 50 Mhz system clock.

(12) LATCHED TRIGGER (SYNC) - the latched trigger signal synchronized to the 50 Mhz system clock.

(13) TS BUSY - the TS busy signal.

(14) SEQUENCER ACTIVE - the signal indicating that the main sequencer of the TS is performing a cycle.

(15) READY - the ready signal indicating that the TS is able to latch input triggers.

(16) ACTIVE - the signal indicating that some process is under way in the TS.

(17) SYNCHRONIZATION - the synchronization signal.

(18) BUFFERS EMPTY - the signal indicating that all the ROC branch buffers are empty.

(19) BUFFER FULL - the signal indicating that at least one of the ROC branch buffers is full.

(20) ROC BRANCH 1 STROBE - the strobe signal on ROC branch 1.

(21) ROC BRANCH 1 AND_ACK (SYNC) - the AND of all ROC acknowledge signals on branch 1, synchronized to the 25 Mhz system clock.

(22) ROC BRANCH 1 OR_ACK (SYNC) - the OR of all ROC acknowledge signals on branch 1, synchronized to the 25 Mhz system clock.

(23) ROC BRANCH 2 STROBE - the strobe signal on ROC branch 2.

(24) ROC BRANCH 2 AND_ACK (SYNC) – the AND of all ROC acknowledge signals on branch 2, synchronized to the 25 Mhz system clock.

(25) ROC BRANCH 2 OR_ACK (SYNC) – the OR of all ROC acknowledge signals on branch 2, synchronized to the 25 Mhz system clock.

(26) ROC BRANCH 3 STROBE – the strobe signal on ROC branch 3.

(27) ROC BRANCH 3 AND_ACK (SYNC) – the AND of all ROC acknowledge signals on branch 3, synchronized to the 25 Mhz system clock.

(28) ROC BRANCH 3 OR_ACK (SYNC) – the OR of all ROC acknowledge signals on branch 3, synchronized to the 25 Mhz system clock.

(29) ROC BRANCH 4 STROBE – the strobe signal on ROC branch 4.

(30) ROC BRANCH 4 AND_ACK (SYNC) – the AND of all ROC acknowledge signals on branch 4, synchronized to the 25 Mhz system clock.

(31) ROC BRANCH 4 OR_ACK (SYNC) – the OR of all ROC acknowledge signals on branch 4, synchronized to the 25 Mhz system clock.

4. Trigger Section

A simplified schematic representation of the input trigger logic of the trigger supervisor is shown in Figure 4.1. There are 12 level 1 trigger inputs, the first 8 of which are prescalable.

An input trigger signal is admitted by the TS only if the input channel has been enabled. When operating in the common strobe mode, the level 1 trigger input signals must be in coincidence with the common strobe signal.

Before entering the prescale circuitry the input trigger is regenerated (on occurrence of its rising edge) as a 12 ns pulse. The prescale circuitry requires a pulse width of at least 10 ns to function properly. The regeneration rather than an input specification is necessary because when the common strobe mode is used the resulting overlap pulse that is fed to the prescale circuitry may be narrow enough to violate this requirement.

The prescaling for an input channel is done out-of-line. That is, the current pulse determines the pass thru status of the next pulse. The prescale circuitry for each channel is composed of a count-down counter and some logic. An input pulse will pass thru the circuit only if the count upon its arrival is zero. The rising edge of the regenerated input pulse serves as a clock to count down or load the counter. If the count is non-zero at the arrival time of this rising edge, the counter will count down. If the count is zero upon its arrival, the programmed prescale factor will be loaded into the counter. In either case the new count value is quickly resolved and its zero status is latched by the falling edge of the same pulse. This determines the pass thru status of the next pulse.

Channels 9-12 have no prescaling feature but are otherwise identical to channels 1-8. This insures that the propagation delays will be the same for all input channels.

After prescaling the 12 trigger channels are collected into a 12-fold OR signal. The leading edge of the OR signal sets a latch if the TS is ready. The TS is ready if:

- the GO bit of the CSR is set,
- no TS cycle is currently active,
- front end busy is not asserted,
- external inhibit is not asserted.

From this latched trigger signal a trigger gate signal is generated. Trigger pulses from the 12 channels that are in coincidence with this gate are individually latched. The latch pattern determines the address that is applied to the look-up memory. The data from this memory location fixes the status of the trigger (accept or fast TS reset). If the trigger is accepted the data also defines the trigger class, the pattern of level 1 accept signals generated, and the ROC code for the event.

The trigger gate width for the TS as supplied defines a simultaneous trigger resolution time of 10 ns. That is, if an input trigger signal on any channel has a leading edge within 10 ns of the earliest such trigger signal, it is included among those that are latched to determine the look-up memory address. Changing jumpers on the board permits the user to select a resolution time of 15 ns, or >15 ns (see Table 5.1). The choice of >15 ns also requires a timing resistor adjustment.

When the data from the memory is valid it is strobed by a delayed signal derived from the trigger gate. If the trigger pattern is acceptable (Level 1 OK bit asserted) the level 1 accept signals (1)-(8) are driven out as programmed along with the level 1 OK signal. These 9 signals are in time with each other, and are issued 38 ns later than the input trigger's leading edge (10 ns simultaneous trigger resolution time selected). The Level 1 OK signal also starts the main sequencer (50 Mhz) of the TS. The level 1 accept signals asserted will remain asserted until the sequencer has completed its cycle. The Latched trigger is then reset and the TS is receptive to new triggers.

If the trigger pattern is not acceptable (Level 1 OK bit not asserted), the TS resets its logic with no level 1 acceptsignals issued. The total time from the rejected input trigger to the trigger being re-enabled is 50 ns.

4.1 Optional Trigger Configurations

There are optional ways to configure input channels 9-12 that adds some flexibility in the use of the trigger supervisor. These are accomplished by changing jumpers on the board (see Table 3.1).

Option 1 - Channels 9-12 are configured as a group not to contribute to the OR trigger signal. Input signals on these channels cannot themselves generate a latched trigger and start a TS cycle. However, they continue to be latched when in coincidence with a latched trigger originating from channels 1-8. This allows the signals of channels 9-12 to serve as in-time vetoes (by using the TS fast reset capability), or simply as additional in-time information with which to determine the level 1 accept pattern and ROC code for the event.

Option 2 - In addition, channels 9-12 can be individually configured to bypass pulse regeneration. For a channel so configured, an input level (rather than an in-time edge) may be used to tag latched triggers when Option 1 is used.

5. On-board Jumper Configuration

Trigger Supervisor configuration options that were expected to be changed often by the user were assigned to register bits. Wire wrap jumpers are used to select operating features that would not normally be changed once the board is in use.

Table 5.1 summarizes the trigger supervisor's configuration via jumpers.

Table 5.1 - Trigger Supervisor Jumper Configuration

Notes: * defines the default configuration
NC = no connection

Location	pin connection	Function
J1	* 3-4	- 10 ns trigger resolution
	4-5	- 15 ns trigger resolution
	1-4	- >15 ns trigger resolution (with timing resistor choice)
J2	* 1-2	- trigger 9-12 in OR trigger
	NC	- trigger 9-12 not in OR trigger
J3	* 1-2	- regenerate input 9 pulse
	2-3	- no regeneration of input 9 pulse
J4	* 1-2	- regenerate input 10 pulse
	2-3	- no regeneration of input 10 pulse
J5	* 1-2	- regenerate input 11 pulse
	2-3	- no regeneration of input 11 pulse
J6	* 1-2	- regenerate input 12 pulse
	2-3	- no regeneration of input 12 pulse
J7	* 1-2	- ROC branch 1 - 40 ns data deskew
	2-3	- 80 ns data deskew
J8	* 1-2	- ROC branch 2 - 40 ns data deskew
	2-3	- 80 ns data deskew
J9	* 1-2	- ROC branch 3 - 40 ns data deskew
	2-3	- 80 ns data deskew
J10	* 1-2	- ROC branch 4 - 40 ns data deskew
	2-3	- 80 ns data deskew