STC DESIGN DOCUMENTATION

Alexandre Zabi

May 15th, 2002

Abstract

This document provides a complete description of the STC design. The structure of the VHDL code, the functional simulation as well as the fitting procedure into a Xilinx Virtex E chip are presented. Each STC daughterboard has one Xilinx FPGA mounted on it in order to perform the clustering and the filtering algorithm. Three PCI Altera chips establish the connections with the three motherboard PCI buses. The Altera chips are not touch on here, however the interfaces between the Xilinx FPGA and the PCI chips are detailed.

1 INTRODUCTION

STC stands for Silicon Trigger Card. It is part of the L2STT (Level-2 Silicon Track Trigger) which is a sub-system of the D0 experiment trigger framework. It performs a precise reconstruction of charged particule tracks in the Central Fiber Tracker (CFT) and the Silicon Microstrip Tracker (SMT). Each track candidate from the CFT, defines a road in the SMT. These roads are used to sort out the hit clusters formed from the raw SMT data. The fit of the track includes the CFT information and the retained hits in the SMT. The L2STT outputs the fitted $r\phi$ tracks to the Level-2 global processor.

The STT covers all of the 12 sectors of the SMT with six VME crates holding each the following boards (a picture of the rack is provided figure 1):

- 1 FRC (Fiber Road Card) which receives the information from the Level-1 CFT trigger and broadcasts it to the STC and TFC.
- 9 STC that convert the FRC roads into range limits in the SMT. They also find the clusters in SMT data from which centroids (center of "mass" of the clusters) are computed. The $r\phi$ centroids are filtered, retaining only the ones that fall within the range limits defined by the roads (fig 2). These centroids become hits in the SMT and are transmitted to the TFC. The centroids calculated from the 90 degrees strips data of the detector, are sent out to the ZVC.
- 2 TFC (Track Fit Card) perform the track fitting.
- 1 ZVC (Z Vertex Card) determines the z-position of the interaction vertex.

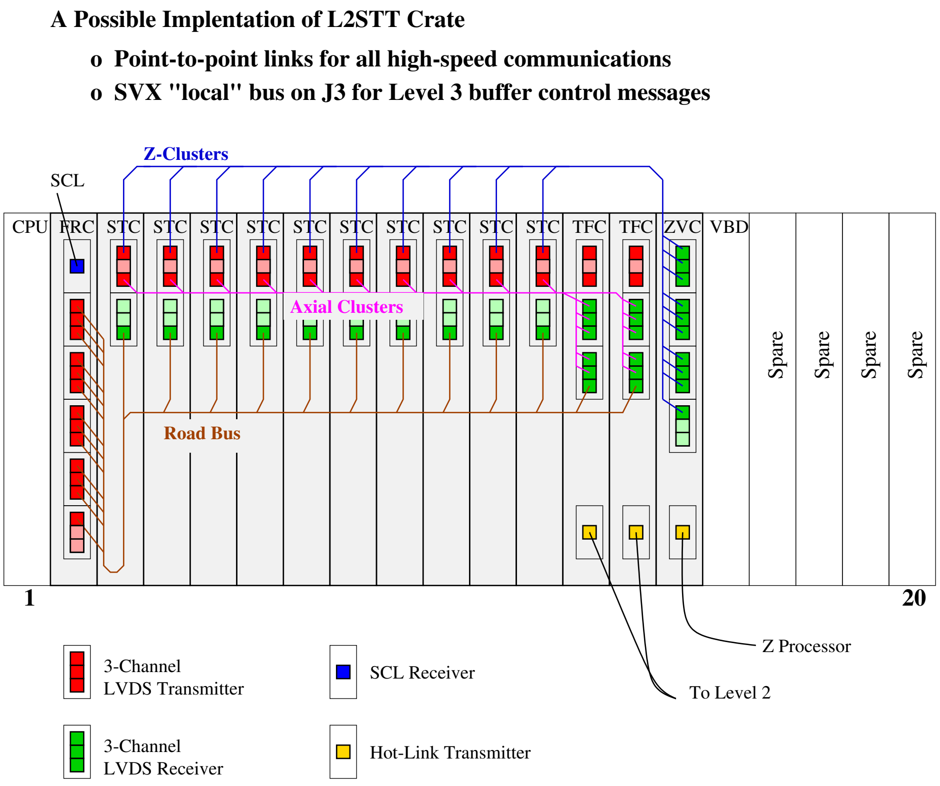

Figure 1: STT crate

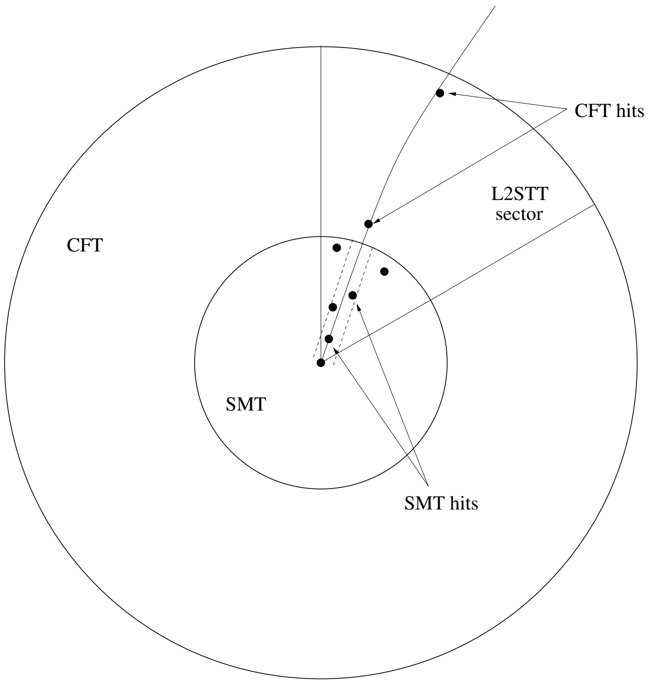

Figure 2: L2STT Track definition

In addition to generating hits and Z-centroids, the STC produces data to the Level-3 trigger. This information consist of intermediate data produced while processing an event for example: preprocessed SMT data, clusters, centroids etc.

The data processing takes place in a single Xilinx FPGA programmed with the VHDL language. The implementation of the VHDL code will be documented in detail but an overview of the design is given in the first place. The functionnal simulation as well as the fitting procedure of the VHDL design will conclude this documentation.

2 OVERVIEW OF THE STC DESIGN

Now that the role of the STC in the L2STT has been described, its hardware implementation can be presented. The following section will give an overview of the design. The inputs and outputs are described as well as the structure of the VHDL code. The clustering algorithm and the definition of a centroid as well as the filtering process are reviewed. The reasons that motivated such approach for the implementation of the STC are also argued.

2.1 The boards:

The STC daughtercard is a 6U board that sits on a 9U STT motherboard. The STC logic board is inserted where PMC-0 and PMC-1 are located (figure 3).

Motherboard:

The motherboard is common to all STT cards. It provides the connections to the following devices:

- VRB Transition Module (VTM):

  The VTM boards are used to get input data from the SMT and the CFT through optic fibers.

- VME bus: Communication with External CPUs.

- Serial Link boards:

  Two type of board are implemented:

  - Link Transmitter Board (LTB).
  - Link Receiver Board (LRB).

  The STT boards communicate through serial link using the LTBs and LRBs.

The motherboard also provides 3 PCI buses to transfer data among the different devices. They operate at a frequency rate of 33MHz.

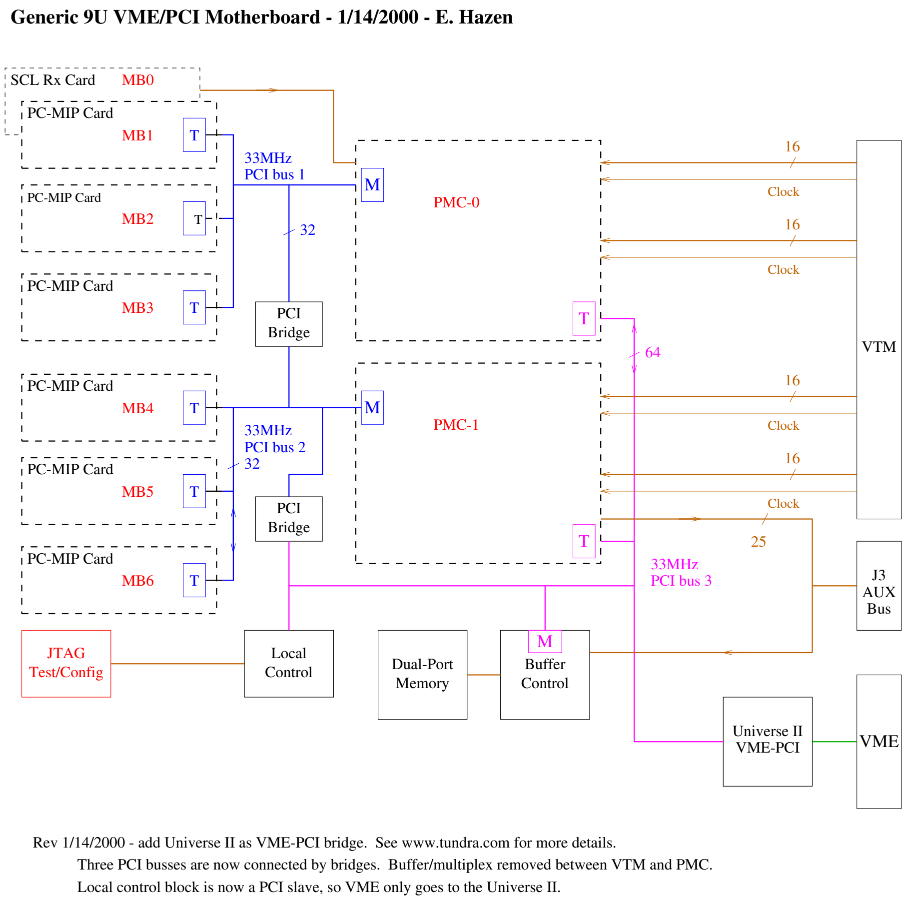

Figure 3: Motherboard

STC card:

The STC card inserted into the STT rack consists of a motherboard and an STC daughterboard. It implements the following devices (figure 4):

- 1 LRB: This is used to receive the road data from the FRC board.
- 2 LTBs : These two cards are used to sent out the following data:
  - Hits to the TFC.
  - Z-Centroids to the ZVC.
- 1 VTM used to receive SMT data.

The Annex 1 shows the STC layout schematic.

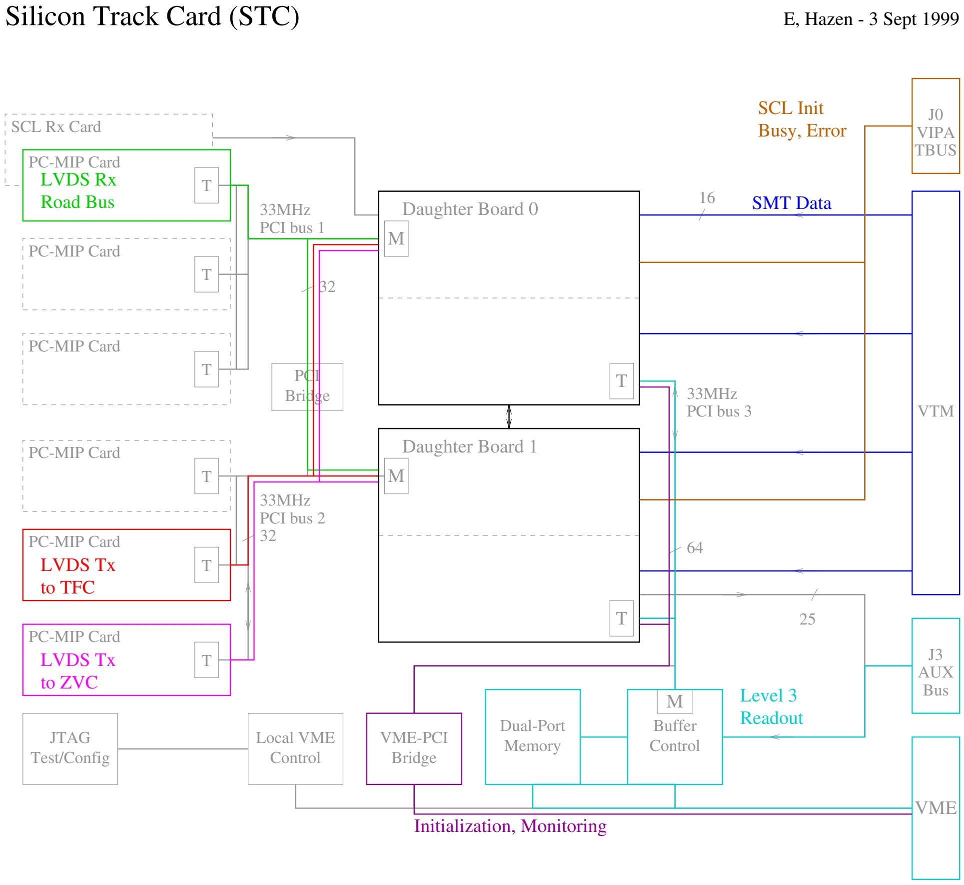

Figure 4: STC board

2.2 Inputs and Outputs

The inputs and outputs of the STC are summarized in figure 5.

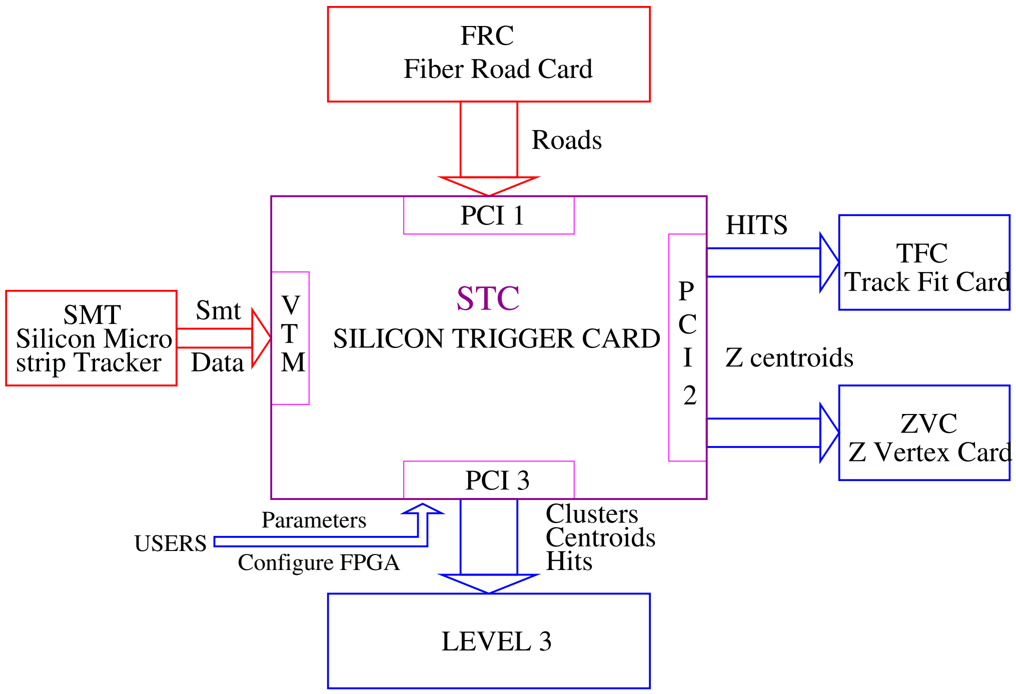

Figure 5: STC inputs and outputs

- INPUTS:
  - Road information:
    The STC receives the road information from the FRC is received through PCI1.
  - SMT data:
    the SMT raw data are transmitted from the front-end electronics to the VTM by optic fibers. Four optic fibers are plugged into the VTM card located on the back side of the crate. Each of them carries the data from 2 silicon detectors at a frequency rate of 53MHz. Thus each STC is processing the data from 8 detectors of the SMT.
  - Downloading information:
    The firmware executable and the parameters as well as the look up tables are downloaded through the PCI3 interface.
- OUTPUTS:
  - Hits and Z centroids:
    After running the clustering and the filtering, hits and Z-centroids are being produced. The hits are sent to the TFC and the Z-centroids to the ZVC (if implemented), both using the PCI2 interface. The output formats for the HITS and the Z-CENTROIDS are given in ANNEX 3.

– Level-3 data:

The second utilization of PCI3 is to output Level-3 data such as Centroids, Clusters and Hits. The data are actually sent to the buffer control that sits on the motherboard and then the VBD is accessed to transmit those to the Level-3 trigger.

2.3 Structure of the design

The structure of the VHDL design is shown on figure 6.

The STC is organized into 8 CHANNEL-LOGIC, 4 SMT-IF and a CONTROL-LOGIC:

- SMT-IF:

  The SMT-IF modules are used to receive SMT data and split them to feed two CHANNEL-LOGIC. They also store test SMT data for testing the STC.

- CHANNEL-LOGIC:

  The STC processes the data of 8 silicon detectors by implementing 8 idential CHANNEL-LOGIC. Each of them is in charge of extracting the clusters from the raw SMT data and computing their centroids. The filtering is activated as soon as the road information is available in the road fifos. The Hits, Z-Centroids and Level-3 data produced by the STC channels are stored into buffers for later read out by the CONTROL-LOGIC.

- CONTROL-LOGIC:

  The CONTROL-LOGIC manages the process of an event through the STC. It provides the interfaces with the PCI buses and issues the control signals to the channels. The FRC roads are converted into range limits by consulting an external SDRAM which stores the 128kX22 road LUT. These range limits are loaded into the STC channel road fifos. The hits and z centroids produced by the channels are collected and sent out to the TFC and ZVC respectively. The CONTROL-LOGIC is also responsible for gathering all the Level-3 data generated by the 8 channels and storing them in another external SDRAM. The data read out from this SDRAM will be formatted and transferred to the Level-3 trigger.

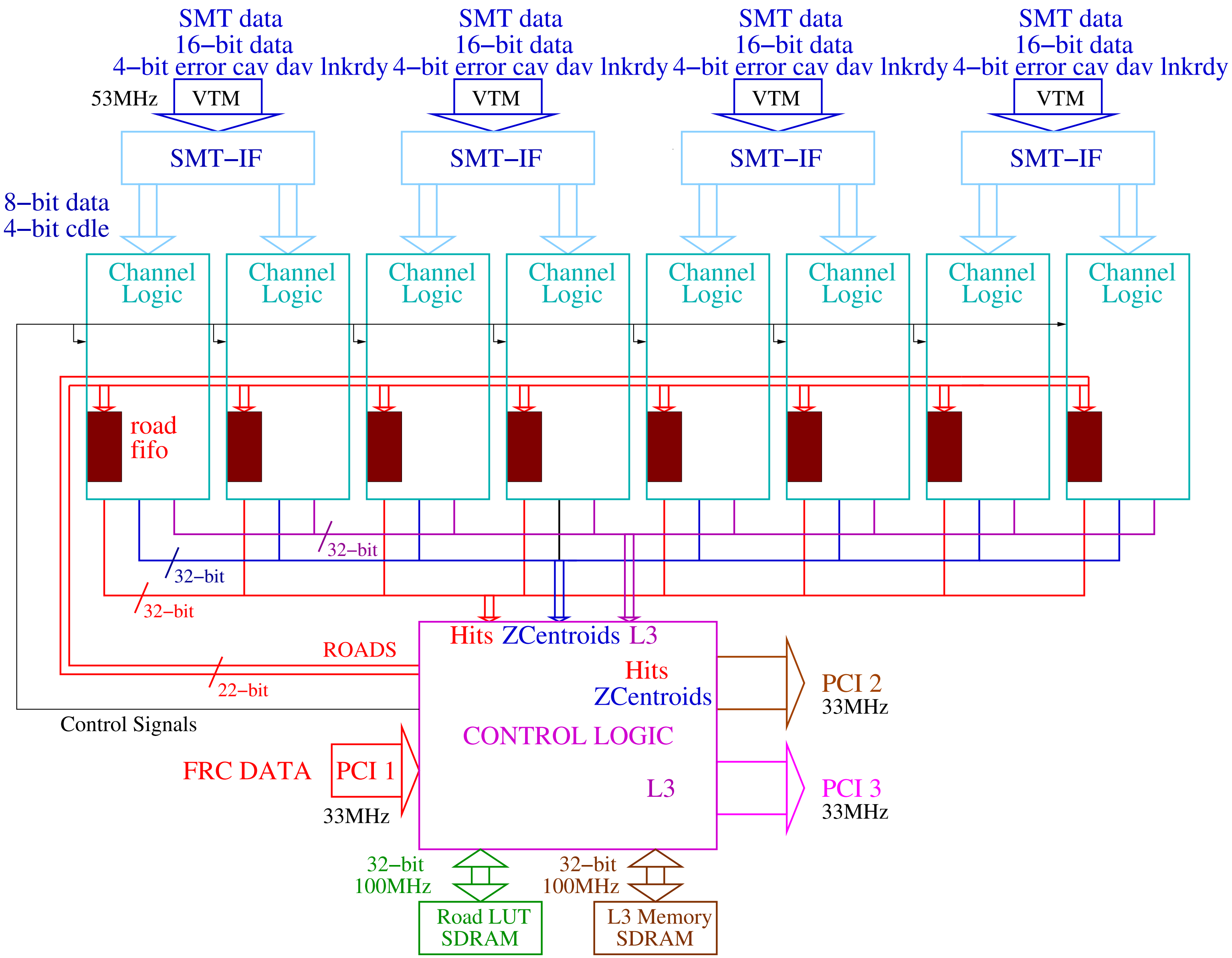

Figure 6: STC Logic Design

2.4 Clustering and Centroid definition:

2.4.1 The clustering algorithm:

The clustering process is one of the main functionnality of the STC. It involves two thresholds that must be downloaded as parameters. After calibration by the STRIP-READER module, every ADC value of each strip must be compared with THRESHOLD1 and THRESHOLD2 where THRESHOLD1 $<$ THRESHOLD2. The first strip having its ADC value above (or equal to) THRESHOLD1 starts a cluster. The strip with data underneath THRESHOLD1 are ignored. The following strips are included into the cluster as long as these three conditions are met:

- The ADC value of these strips remain above THRESHOLD1.
- Their addresses (CHIP-ID + STRIP-ID) are consecutives.
- The ADC counts are from the same type of strips (either axial, stereo or z)

Note that the SVXII read out procedure does not include non active strips. So the SMT output may not have consecutive strip numbers. If any of these conditions is not met, the cluster ends and a new one starts with the next strip data above THRESHOLD1.

Although group of strips have been formed, not all of them can be retained as cluster. Only the group of strips that have at least one ADC value above THERSHOLD2 are kept as clusters, the others are discarded. Every strips that have their data above THRESHOLD2 are eligible as cluster centers. Now, if several values are found above THRESHOLD2 within the same cluster, the strip that has the highest ADC pulse is considered as cluster center.

Depending on the STC operating mode, the clusters consist of either 5 strips or 3 strips maximum. In case of the 5 strips mode, two strips on both sides of the cluster centers remain. The figure 7 illustrates the clustering algorithm of the STC (Some examples are also given in Annex 2).

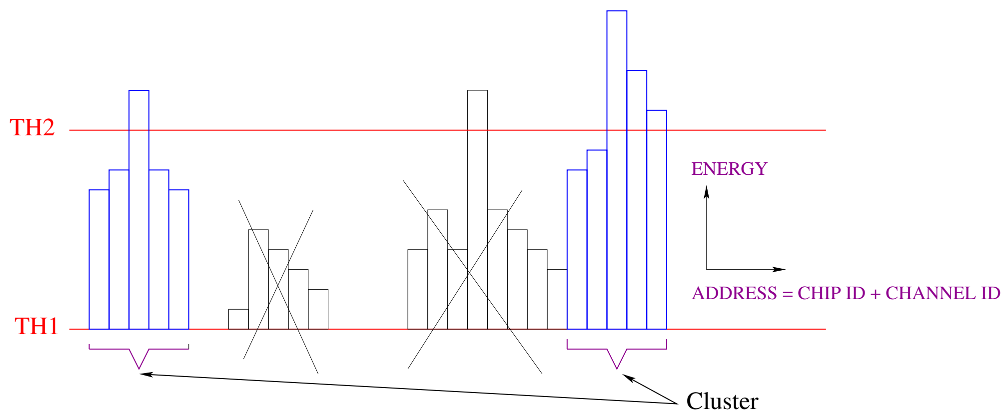

Figure 7: Cluster Definition

Note that if more than one cluster center are found because their ADC counts are equal and both above THRESHOLD2, the strip that is received the latest from the SMT, becomes the cluster center:

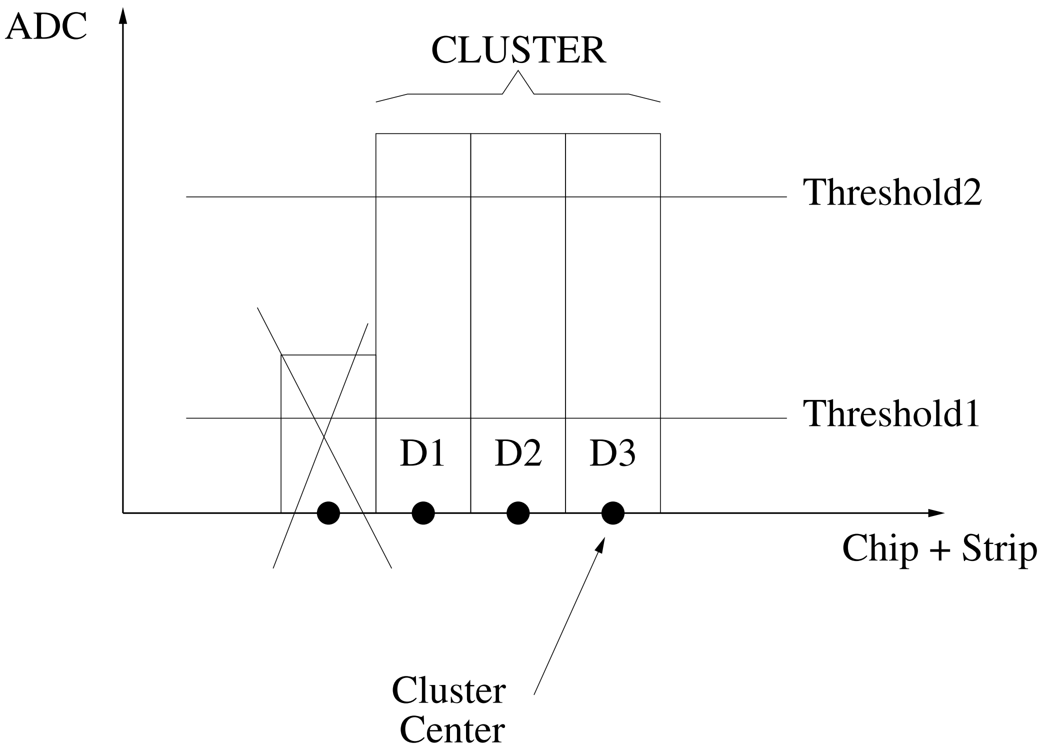

Figure 8: Cluster Definition

2.4.2 Centroid definition:

Definition:

The centroid is calculated from the cluster. It is a more precise definition of the cluster center. As defined above, the cluster center is the strip that has the most energy deposited on it. Therefore, the rest of the energy of the cluster may not be equally distributed on each sides of the center. A pulse-height weighted effective cluster center can be calculated in order to take into account this distribution and give a more precise location of the hit of the particle in the ladder. The improvement in resolution of the cluster center requires the addition of two more bits to the original 7-bit STRIP-ID.

Calculation:

The ADC values of the cluster can be called D1, D2, D3, D4 and D5, where D3 is the ADC count of the cluster center. The two precision bits are evaluated by looking at the following division:

$d = \frac{NUMERATOR}{DENOMINATOR} = \frac{-2\times D1 - D2 + D4 + 2\times D5}{D1+D2+D3+D4+D5}$

The two precision can be determined using this table:

	Precision bits (binary)
$0 \leq \lvert d \rvert < 0.25$	00
$0.25 \leq \lvert d \rvert < 0.50$	01
$0.50 \leq \lvert d \rvert < 0.75$	10
$0.75 \leq \lvert d \rvert < 1$	11

These two bits are either added or subtracted to the 11-bit cluster center address by looking at the sign of the NUMERATOR.

In case of a 3-strip mode, the ADC value D1 and D5 are ignored in the division above.

An example of a centroid and its calculation is given below (The Annex 2 gives also some examples of centroid calculation):

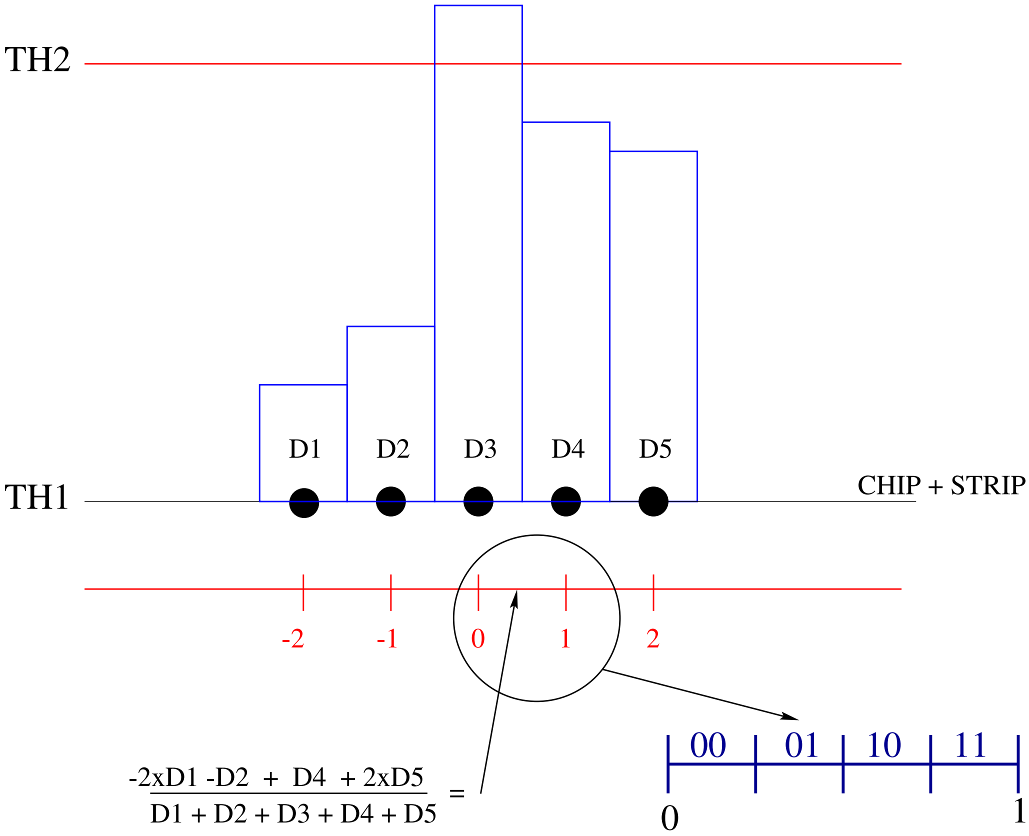

Figure 9: Centroid Calculation

2.5 Production of the Hits

Serial design for the filtering:

This design is implementing a serial filtering algorithm. Each centroid is compared with one road at a time. The filtering process is achieved when all the centroids of an event have been compared with all the roads of the same event. On one hand, this approach involves the implementation of only 2 comparators (one for each road limit) per channel but on the other hand the timing process is lengthened. But the few resources (or logic gates) required, made the utilization of a single Xilinx chip possible. A parallel approach of the filtering design would implement as many comparators as the maximum number of roads to handle by the STT (48 per sector for the time being). This option would be rather expensive in term of resources of the chip.

Filtering process:

So because the filtering is the most time consuming process, the STC has been designed in such way that all the other tasks are executed as fast as possible. The accent has also been put in optimizing the logic by introducing numerous registers in order to reach a maximum system clock about 100MHz. Such internal clock makes the filtering process taking place at a rate of one comparison every 10ns.

The tasks we are referring to here are the clustering and the conversion of FRC data into road limits. The clustering algorithm is running as data are coming through and centroids are calculated on the row. No buffers are inserted on the data path to avoid delays which obviously increase the processing time. Therefore, a FIFO is needed before the logic that performs the filtering. This buffer is large enough to store several events. The advantage of having a buffer after the clustering algorithm rather than at the input of the design is that the amount of data is smaller so the storage of several events needs less memory resources.

While the clustering algorithm is being performed, the CONTROL-LOGIC extracts the road information from the FRC data and converts it into road limits. Each channel receives different road limits according to the geometry of the SMT detector. This operation may be taking some time. An additional delay has to be taken into account: the road information and the SMT data are released from two different parts of the D0 detector. These blocks of data are most likely to be received with a timing skew. By being able to store several events of centroids and roads, the filtering can be launched consecutively. Hits are also being stored by the channels and before being sent out to the PCI2 in order to reduce latency. In average, the number of roads is around 1 and the number of centroids is around 5 in a 60 degrees sector for only 1 interaction (ref). The fitering process would take $0.05\mu s$ in this case, which represent a relatively small amount of time. The hits distribution is around 4 on STT track (ref). The events with 30 or 40 roads can be handled by this design but precautions must be taken when several of these events are through. This point will be clarified when describing the VHDL implementation. Note that the maximum number of hits that can be produced by the STC is 256 per event.

2.6 Level3 and monitoring information

In addition to its primary function of finding the hits, the STC must create Level-3 material as well as monitored data.

Level 3 data:

The Level 3 produced are the following:

- L3 Raw data: data coming in form the SMT.
- L3 Corrected data: data preprocessed by the channels.
- L3 Cluster: clusters found in the input SMT data.
- L3 centroids: Centroids calculated from the clusters.
- L3 hits: hits found after filtering the centroids with the roads (same information as for PCI2).

The CONTROL-LOGIC determine which kind of data are to be produced per event by looking at a downloaded parameter called L3-CONFIG (L3 configuration word). Once gathered, the data are formatted and sent out to the Level-3 trigger. The amount of Level-3 data generated by the STC may be very big and it would be impossible to handle the storage of several events inside the Xilinx chip itself. For example, Level-3 raw data are the data as they are received by the CHANNEL-LOGIC. A block of raw Level-3 data can easily reach 200 to 300 words. The Xilinx chip can definitely not provide storage for several events about this size (even though only one of the STC channel is suppose to output l3 raw data at a time). Furthermore, the Level-3 data must be formatted before sending them out. This operation needs time and implies the implementation of large buffers. Besides, the output to PCI3 is done at a 33MHz rate which can be considered as a bottle-neck. To solve this problem, an external SDRAM is implemented to store all of the Level-3 information. The Level-3 data produced by the STC are not formatted but compacted to increase the bandwidth. The advantage of using such SDRAM is that they can be run as fast as 200 MHz which suits the design requirements of 100MHz. The design has plenty of time to read out the data and format them before sending them out to the PCI3 interface.

Monitoring:

The STC provides also the monitoring of an event. This operation consist of counting the number of errors that occurs while processing the data. It is also used to check the activity of the board. Each Channel logic is provided with a monitor. These modules implements large counters that can be consulted through PCI3 in order to check on the status of the STC.

2.7 Set up and downloading procedure:

After configuring the FPGA, the set up of the STC logic board must be performed. This operation is executed trough the PCI3 interface. Control registers are implemented in the CONTROL-LOGIC module in order to set up the Firmware. Selecting the operating mode of the STC as well as the online checking of the status can be performed by addressing those registers.

Once the code is loaded into the chip, the parameters needed to process the data are downloaded through PCI3. Several tables are stored into memories in the STC channels and the CONTROL-LOGIC. A read back from these memories is provided.

The information on the downloading and setup of the STC is given in the memory map specification.

2.8 Test mode

To be able to test the STC without removing it from the crate, test LUTs are implemented. Memories have been allocated to store SMT test data as well as FRC test data. These test files are downloaded into the SMT-IF for the test SMT data and in the PCI1-Altera-chip for the test road data. The test roads are stored in the PCI1 Altera interface chip for convenience (save memory ressources). These files can be written through PCI3 at the same time as the rest of the downloaded parameters. When the STC is set up in TEST mode a bit in the control registers can be set to launch the testing procedure. The output can be checked to determine the functionality of the STC. Various option can be sectioned to proceed the test:

- Timing skew between the FRC data and the SMT data
- using only SMT test data and road data coming from the FRC.
- using SMT data from the VTM and FRC test data.

Those setups are also described in the memory map.

3 VHDL IMPLEMENTATION

This section describes the VHDL code in details. The SMT-IF, the CONTROL-LOGIC and the CHANNEL-LOGIC functionalities and implementation are documented.

3.1 SMT-IF

The SMT-IF module provides an interface between the VTM and the STC logic. Its main function is to receive the SMT data and transmit them to the STC channel logic. This mode is called the data taking mode. This module can also be used to store test SMT data. The testing mode consists of sending through test data instead of real VTM data.

3.1.1 Data Tacking Mode:

The SMT data consist of a series of double byte words plus 4 control signals : CAV, DAV, LNKRDY and an error bit ERR. The data are transmitted to the STC in conjunction with a 53MHz clock called STRBOUT. In this mode, the SMT-IF is an asynchronous fifo used to output data to two channels as a 16-bit bus FIFO-DATA in conjunction with a DATA-AVAILABLE signal. Each channel gets 8-bit data words and 4 control signals (10). Note that the control signals are identical for two STC channels.

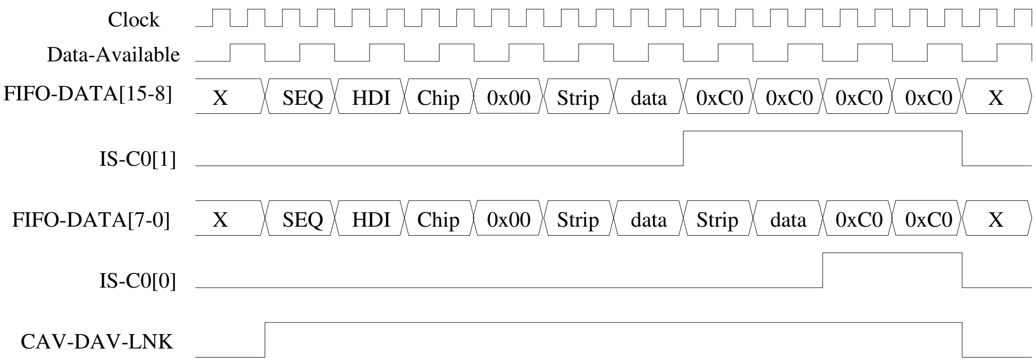

Figure 10: SMT-IF output

Two counters are implemented to generate the write and read addresses for this fifo. Their values are compared every system clock to determine whether data are present or not. If data are stored, a READ signal is issued to send them out to the channels. This READ signal is also used as a DATA-AVAILABLE signal.

Other output signals:

- CAV-DAV-LNK:

  This signal is high when the data from the VTM are valid. SMT data are valid when the CAV bit is high and both DAV and LNKRDY bits are low. Any other combinations set the CAV-DAV-LNK to zero.

- IS-C0:

This signal is high when the data word output to the channel is a 0xC0. The 0xC0 are used to spot the SMT data trailer as described later in the STRIP-READER section.

3.1.2 Test Mode:

In test mode, this module is a 512x22 look up table. This memory space is dedicated to store SMT test data. When the TEST mode is selected as operating mode for the STC, test data are injected into the design instead of VTM data. This function allows the user to proceed a test of the firmware design at any time.

The SMT-IF is able to store up to 512 words which is enough space to hold several events. The test SMT data look exactly like real data from the detector. They consist of a double byte word of data plus 4 control bits. Two additional control bits are used here to spot the end-of-events and end-of-file words. The last word of an event must have the bit 21 set to one and the last word of the test data file must have both bits 21 and 20 set to one as shown figure 11.

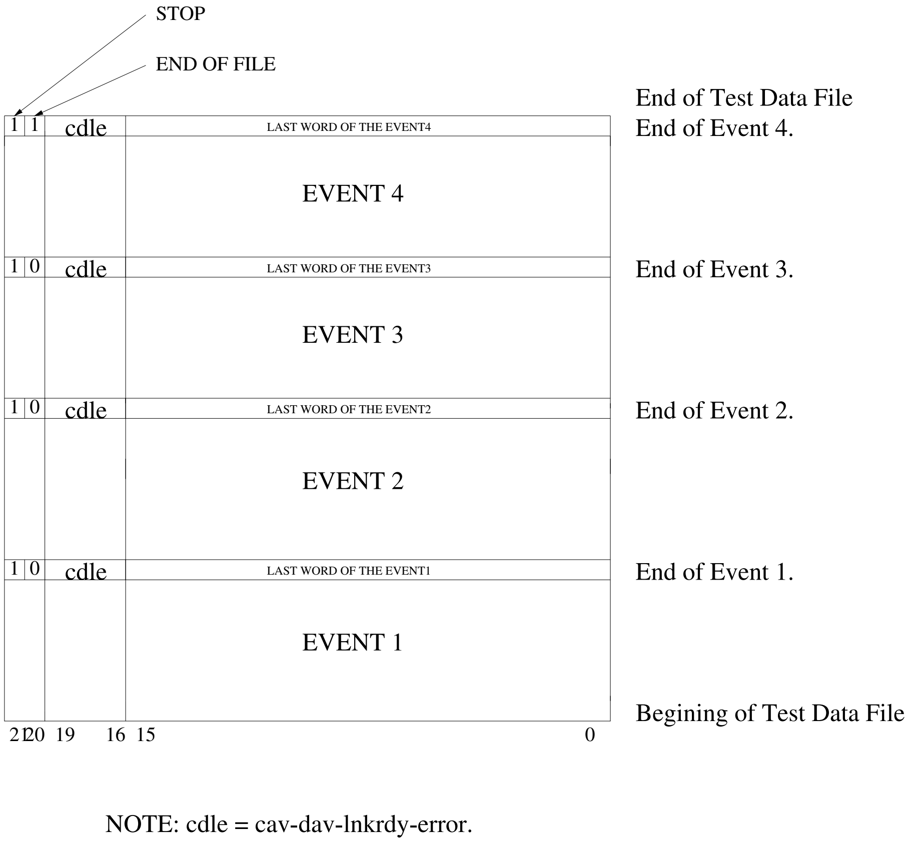

Figure 11: SMT test data file

The test starts when the CONTROL-LOGIC of the STC issues a TEST-START signal. A counter is then incremented to read out the content of the test LUT until the end-of-event is detected (bit 21). The TEST-START bit may be set again to read out the next event. When the end-of-file is reached, the counter is reset and points to the beginning of the test data file. The next TEST-START will result in outputting the first event again. Several loops can be performed while testing the hardware.

Note: If the STC is offline and the STRBOUT clock is not available for testing, the INT-TEST-CLK configuration bit can be set. This signal enables an internal clock of 50MHz derived from the system clock.

The INT-TEST-CLK and the TEST-START signals are explained in the CONTROL-LOGIC section.

3.2 CHANNEL-LOGIC

The CHANNEL-LOGIC performs the clustering and the hit-filtering of the SMT data. Eight identical channels are implemented in the STC design. They are composed of three modules as shown on figure 12:

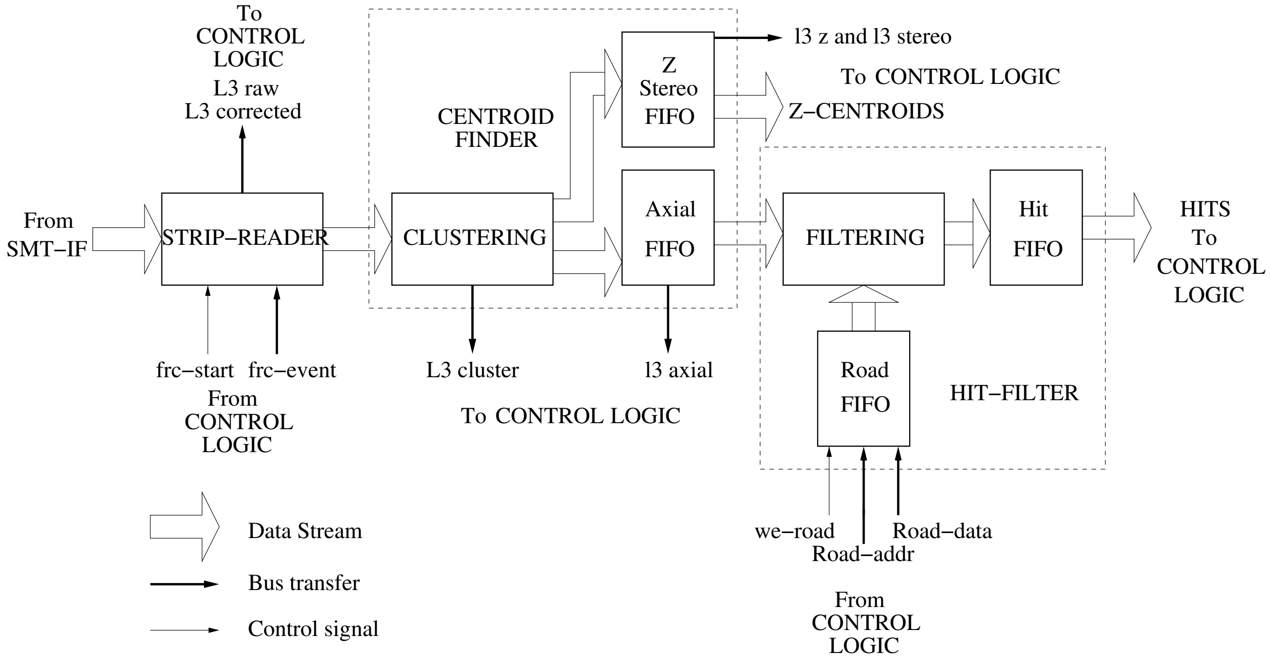

Figure 12: CHANNEL-LOGIC

- STRIP-READER:

  The STRIP-READER receives the data from the SMT-IF and corrects them for offset and scale. This operation is done consulting downloaded Look up tables. Another function of this module is to add an event number to the SMT data block. The preprocessed data are then passed on to the CENTROID-FINDER module.

- CENTROID-FINDER:

  The CENTROID-FINDER forms the clusters from the STRIP-READER data and calculates the centroids (center of mass of the clusters). The centroids in the $r\phi$ plan (axial centroids) are stored into a fifo for later read out by the HIT-FILTER module. The other types of centroids (z and stereo) are stored in a separate buffer for read out by the CONTROL-LOGIC.

- HIT-FILTER:

The HIT-FILTER is used to keep only the centroids that fall into at least one of the range limits defined by a road. The roads are written into a ROAD-FIFO by the CONTROL-LOGIC. The filtering process starts as soon as all the roads and at least one centroid are present into their respective buffers. The hits produced are stored and transmitted to the CONTROL-LOGIC.

The CHANNEL-LOGIC is also in charge of generating Level-3 data. These data consist of preprocessed information produced by the STC while an event is through:

- Raw data: Data as they are received from the VTM.
- Corrected data: SMT data corrected for offset and scale by the STRIP-READER.
- Centroids: Axial, Stereo and Z centroids produced by the CENTROID-FINDER module.
- Hits: SMT hits found by the HIT-FILTER module.

The CHANNEL-LOGIC has been designed to process data up to the clustering with minimum control by the CONTROL-LOGIC module. Centroids are calculated and stored as VTM data are being received. Therefore, the CONTROL-LOGIC intervenes when adding an event number to the SMT data block and when filtering centroids.

3.2.1 STRIP-READER

The STRIP-READER is the first module on the data path. It is designed to correct SMT data for gain-offset and eliminate information from defective strips. The calibration must be performed before running the clustering algorithm. The SVXII chips mounted on each silicon detector have different gains and pedestals, the data recorded on their strips must be corrected before being compared to the clustering thresholds. Furthermore, some of the strips have mechanical problems and their ADC values must be discarded. These functions are performed by consulting the gain-offset and bad-channel tables.

The STRIP-READER is organized in eight sub-modules as shown on figure 13. The STRIP-READER-CONTROL manages the process of the data. Memories are also implemented to store the required parameters and tables. These memories must be downloaded through PCI3 prior to run the STC. This module is also in charge of monitoring an event. The preprocessed data are fromatted and stored into a small fifo for later read out by the CENTROID-FINDER. Another feature of the STRIP-READER is to output raw and corrected data for L3 purposes. Each of those sub-modules are described in the following sections.

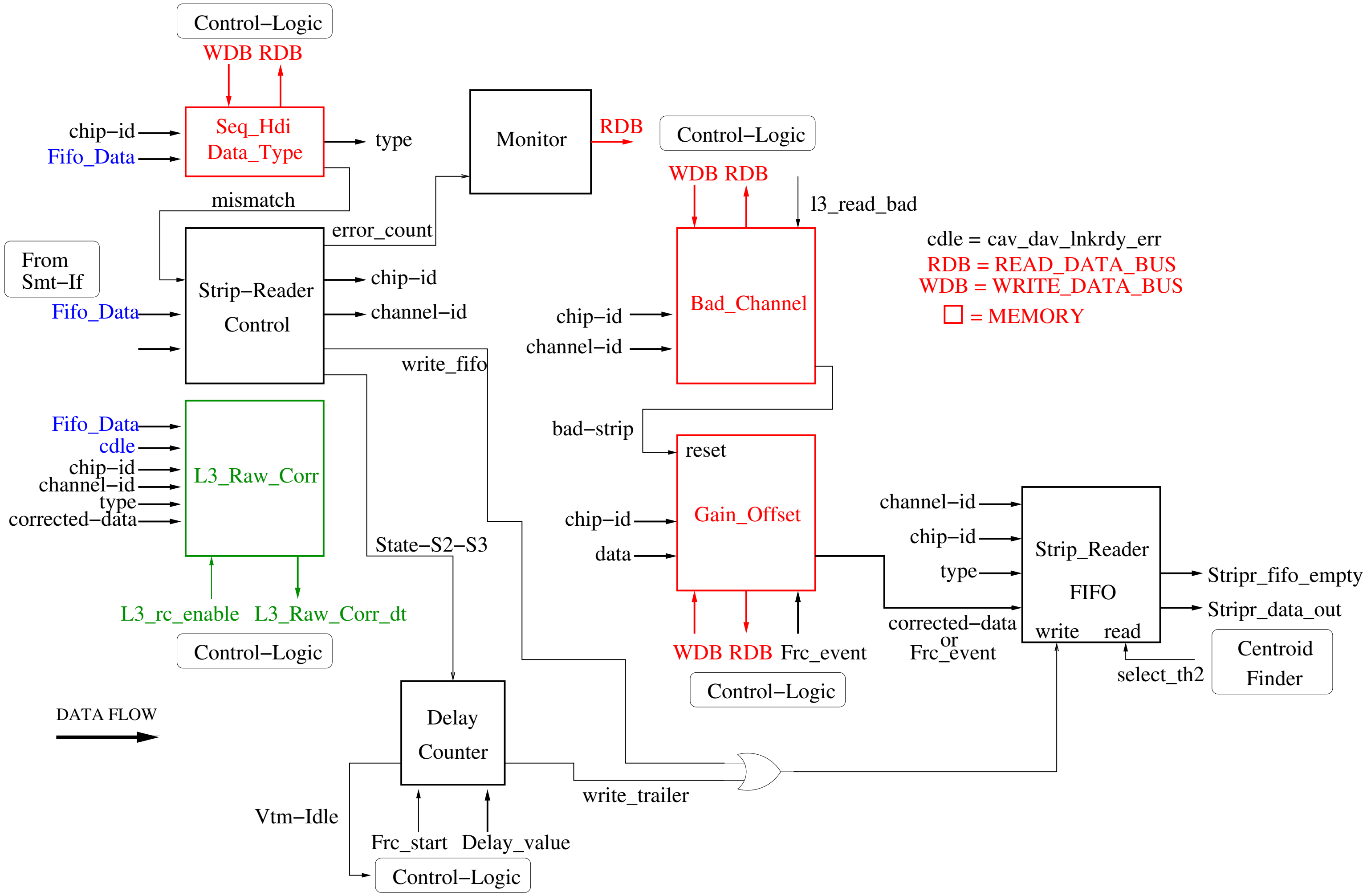

Figure 13: STRIP-READER

STRIP-READER-CONTROL:

Definition:

The STRIP-READER-CONTROL manages the process of the SMT data through the STRIP-READER. It implements a state machine with 4 states. Each of these states executes a different task according to the incoming data word. The input SMT data format is the following:

cav	dav	lnk	err	7 6 5 4 3 2 1 0	
1	0	0	err	sequencer id	0 to 255
1	0	0	err	hdi id	hdi id = 0 to 7
1	0	0	err	1 | chipid	chip id, 0 to 8
1	0	0	err	hex 00	
1	0	0	err	0 | channel id	
1	0	0	err	data	
1	0	0	err	0 | channel id	strip number, 0 to 127
1	0	0	err	data	

. . .

1	0	0	err	c0	trailer
1	0	0	err	c0	trailer
1	0	0	err	c0	optional filler bytes

. . .

Figure 14: SMT data

The input data from the SMT-IF are treated through the state machine when both signals DATA-AVAILABLE and CAV-DAV-LNK are high. Note that the CAV-DAV-LNK signal remains high as long as data from the same event are driving the optic link.

Description of the state machine:

The machine is structured into 4 states shown in figure 15. The transitions between states are also illustrated on the same figure. The state-s0 is the idle state, the system waits for data. Then the logic jumps to state-s2 and to state-s3. It jumps back and forth from state-s2 and state-s3 when data are being processed. When the trailer of two 0xC0 is reached, the machine goes back to state-s0.

The content of each state is shown figure 15 and figure 16.

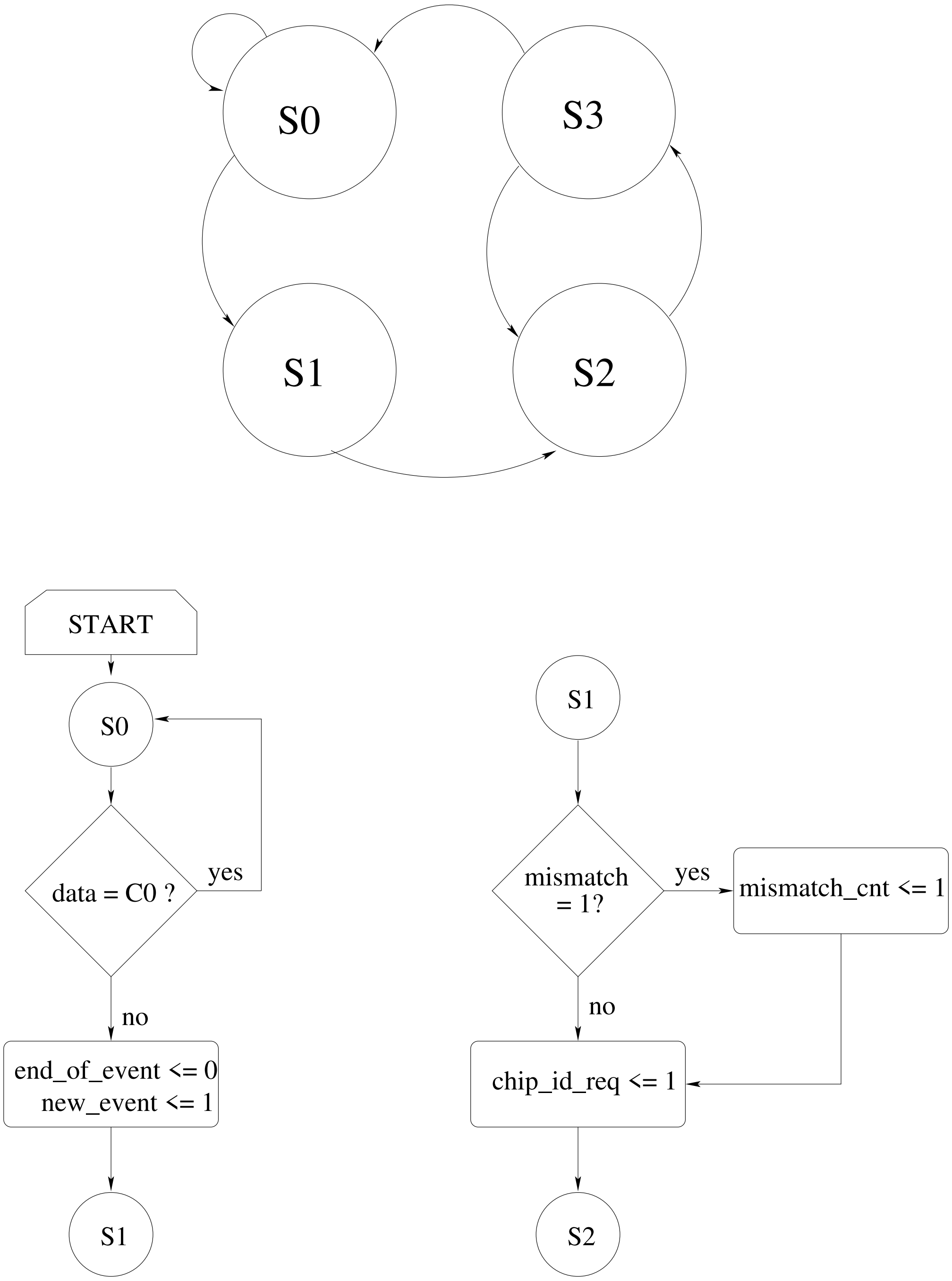

Figure 15: Strip Reader Control

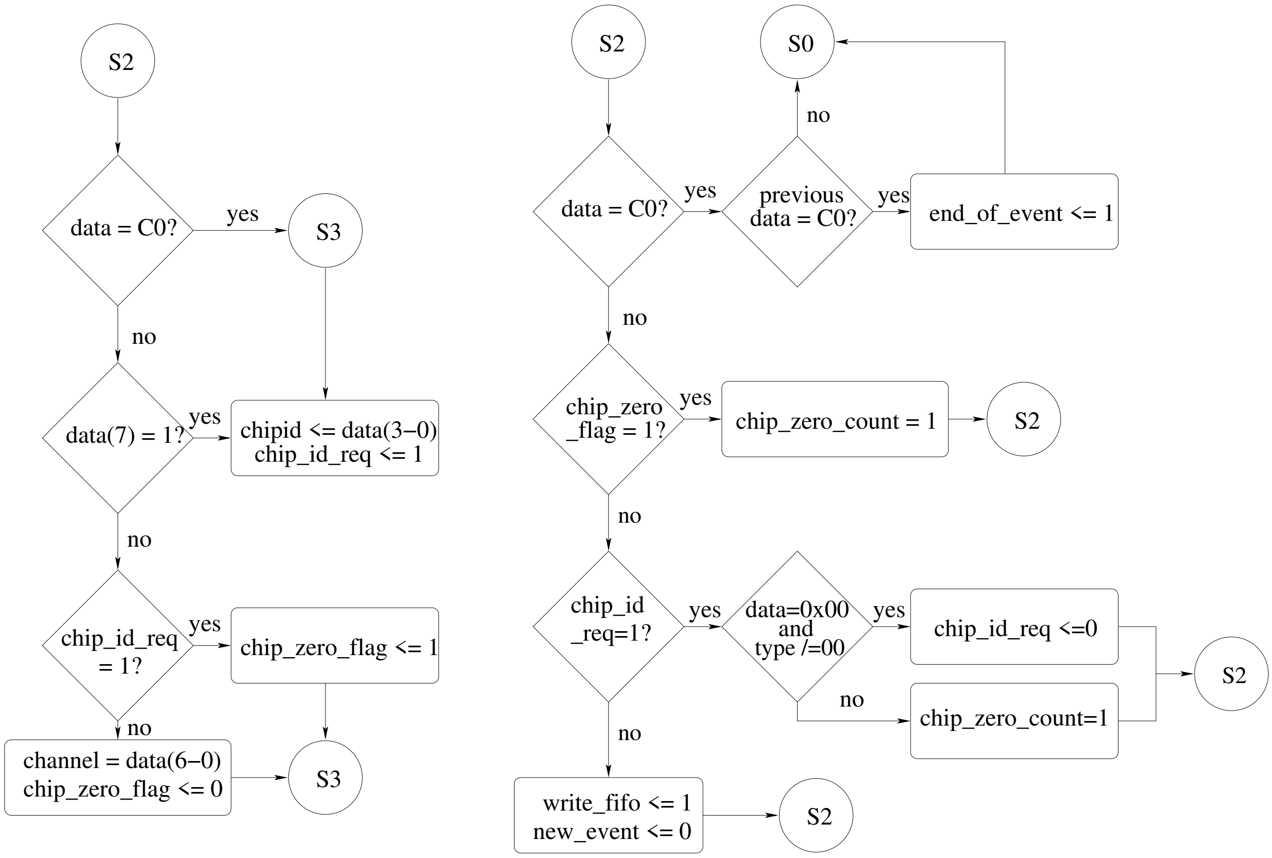

Figure 16: Strip Reader Control

<u>State-s0:</u>

The State s0 is considered as the idle state. The process of an event starts as soon as the first word having its CAV-DAV-LNK signal high is received. This word must be interpreted as the SEQUENCER-ID and the logic has to proceed to the next state.

Three signals are propagated:

- NEW-EVENT-FLAG: This bit informs the STRIP-READER logic that a new event is starting.
- END-OF-EVENT: This signal must be reset at this point.
- STATE-S0: Remains high as long as the system remains idle.

Situations when the system remains idle:

- CAV-DAV-LNK signal low:

  When CAV-DAV-LNK is zero, the state machine is forced into an idle state. This measure is taken because of the situations when either the trailer is missing or not valid. In normal cases the CAV-DAV-LNK is expected to go low right after the second 0xC0C0 received from the optic fiber.

- 0xC0 being transmitted:

The size of an event may be different for the two silicon detector that share the same optic fiber. Thus, 0xC0 word are inserted after the second 0xC0 byte to get to an equal number of words from each ladder. The CHANNEL-LOGIC must remain in state-s0 as long as these extra 0xC0 are coming in.

- RESET or DISABLE-CHANNEL:

  In case of a system RESET, the logic goes back to idle. This is also the case if the channel is disabled (DISABLE-CHANNEL signal high).

<u>State-s1:</u>

This state performs the following tasks:

- Checks the SEQUENCER and HDI ids:

  The SEQUENCER and HDI ids received must be match the expected ones. This is done looking at the MISAMATCH-FLAG from the SEQ-HDI-DATA-TYPE module. In case of a MISMATCH-FLAG high, the STRIP-READER-CONTROL must inform the MONITOR by sending its MISMATCH-COUNT.

- Set the CHIP-ID-REQ bit:

  After receiving the SEQUENCER and HDI words, the CHIP-ID is next. This bit tells the next state that the following word is required to be a CHIP-ID.

- Set the STATE-S2-S3 signal:

  The STATE-S2-S3 signal is used to let the rest of the STRIP-READER logic knows that data are being processed.

- Send the ERROR-BIT-COUNT to the MONITOR:

  For monitoring purposes, when either the VTM error bit of the previous or the present input word is one, an ERROR-BIT-COUNT flag is sent to the MONITOR.

<u>State s2:</u>

At this point, data start being transmitted by the detector. Each block of data begins with the CHIP-ID and the list of active strips follows. Every strip is follwed by its ADC value.
The tasks executed in this state are described in the following:

From State s1:
The MSB of the CHIP-ID word is checked:

- If the MSB is high:

  The 4 LSBs of the input word are saved into a register as CHIP-ID. Although the MSB was set, the CHIP-ID is utterly identified when the following zero byte is received in the state s3. The CHIP-ID-REQ remains high to force this check in the following state.

- If the MSB is low:

  The system has encountered an error and the logic must proceed to state s3. The CHIP-ZERO-FLAG is pulled to tell state s3 that the CHIP-ID wasn't correctly detected.

From State s3:
If coming from State s3, different tasks are performed:

- Saving a STRIP-ID:

  The MSB must be zero in order for the logic to record the 7-bit STRIP-ID into a register. The CHIP-ZERO-FLAG is also reset at this point.

- Receiving a new CHIP-ID:

  If the MSB is one, a new CHIP-ID is saved. The CHIP-ID-REQ is pulled to force the check on the zero byte in state s3.

- Receiving the trailer:

  The trailer of the SMT data shows up with a first 0xC0 byte. The logic proceeds to the state s3 right away.

<u>State s3:</u>
The following tasks are performed in this state:

- End the event:

  If a first 0xC0 byte has been seen in the state s2 and second one is received in this state. The END-OF-EVENT bit is set to one and the logic goes back to idle. Nevertheless, even if the second 0xC0 word is not to be seen, the logic still jumps back to idle but the END-OF-EVENT bit is not set.

- Reset the STATE-S2-S3 signal:

  The STATE-S2-S3 signal is also reset to inform the rest of the logic that the process of the event is completed.

- Check on CHIP-ID validity:

  In state s2, the CHIP-ID has been received and saved. However, the CHIP-ID is valid only if the 0x00 byte is following (according to the SMT data format) and the corresponding DATA-TYPE is legal. The following checks must be performed to insure this:

  - CHIP-ZERO-FLAG :
    The CHIP-ZERO-FLAG may be set in state s2 if the MSB wasn't one. In the present case, it is not necessary to look at the 0x00 word or DATA-TYPE, the CHIP-ID is not valid.
  - Check on 0x00 word:
    The value received is compared with 0x00.
  - DATA-TYPE:
    The SEQ-HDI-DATA-TYPE memory sends a 2-bit DATA-TYPE according to the 4-bit CHIP-ID. If this type is neither axial, stereo or z, the value 00 is transmitted. This is an illegal type which involves an invalid CHIP-ID number.

  If any of this three checks fails, the CHIP-ID is considered not valid and the following data must be discarded. There is no gain-offset and bad-channel tables for these type of chips. The data cannot be analysed properly. The invalid CHIP-IDs must be monitored by sending a CHIP-ZERO-COUNT to the MONITOR module. The CHIP-ID-REQ is not reset to force the logic into an awaiting state. The next valid CHIP-ID will start the process of data.

NOTE: Only valid CHIP-ID get their following data to be processed unless the ACCEPT-BAD-CHIP bit is set. This is a different operating mode selected in the COONTROL-LOGIC. It allows data from invalid CHIP-ID to be processed by the CHANNEL-LOGIC. The CHIP-ID-REQ is reset in state s3 as for a valid CHIP-ID.

- Writing data into the STRIP-READER-FIFO:

  When coming from state-s2 after receiving a STRIP-ID, a DATA word is following. A WRITE-FIFO signal must be issued to the STRIP-READER-FIFO in order to save the corrected data word.

  The NEW-FLAG-FLAG is reset at this point.

- Send the ERROR-BIT-COUNT to the MONITOR:

  As state s1, an ERROR-BIT-COUNT is generated if the previous or the present word has its VTM error bit set.

<u>SEQUENCER-HDI and DATA-TYPE memories</u>:

The following memory spaces are allocated to store the SEQUENCER and HDI parameters as well as the DATA-TYPE of the SVXII chips. They are READ/WRITE memories, implemented in the same block called SEQ-HDI-DATA-TYPE.

SEQUENCER-HDI:

Definitions:

The CHANNEL-LOGIC is configured to process data from a particular silicon detector. Every detector can be identified with the SEQUENCER and the HDI identification numbers:

- SEQUENCER-ID:

  This an 8-bit word associated to every and each SEQUENCER boards of the SMT front end electronics.

- HDI-ID:

  The HDI-ID identifies the silicon detector connected to the SEQUENCER board. The HDI is the read out cable that carries the signals from the SVXII chips mounted on an SMT ladder.

These Ids are passed on to the STT as header words in the data stream 14. They are also downloqaded as parameters into the SEQUENCER-HDI memory. This must be done for the STC cards once the optic fiber cables have been plugged into the VTM.

Comparing IDs:

The process of the data must start with comparing these two downloaded parameters with the expected ones. This check is needed in order to avoid the process of data with the incorrect gain-offset and bad-channel tables.

If either the HDI-ID or the SEQUENCER-ID is not matching the downloaded value, a MISMATCH-BIT is generated. This bit is conveyed to the STRIP-READER-CONTROL module which informs the MONITOR. The Monitoring of this error makes it possible to point out cabling or set up problems.

DATA-TYPE:

Data-type definition:

There are up to 9 SVXII chips per silicon detector having each 128 strips. There are three different orientations of strips possible:

- Axial: The strip are oriented along the beam axis (z axis of the detector).
- 90 degrees (or z): Strips that make a 90 degrees angle with the axial ones.
- Stereo: The stereo strips have a 2 degrees angle with the beam axis.

The Silicon detectors have at least 5 chips with axial strips and either 4 chips with stereo strips or 3 chips with z strips. The type of data is then determined looking at the 4-bit CHIP-ID. The 2-bit DATA-TYPE is generated by this table according to the input CHIP-ID:

DATA-TYPE	Type of Strip
01	STEREO
10	AXIAL
11	Z
00	Undefined

Utilization:

This information is used by the CENTROID-FINDER to select the correct clustering thresholds. The centroids are sorted by types and stored in separate buffers. Only the axial centroids go the HIT-FILTER module.

A CHIP-ID that would give an undefined DATA-TYPE is considered as illegal. The DATA-TYPE is also sent to the STRIP-READER-CONTROL for monitor purposes.

<u>BAD-CHANNEL:</u>

Bad-Channel table:

The BAD-CHANNEL is a memory space allocated to store the strip status table. Some strips may have mechanical problems or may be noisy so their ADC values must be ignored. The data recorded on those strips are forced to a zero value in order to be thrown away by the clustering algorithm.

There are up to 9 SVXII chips per detector which leads us to a maximum number of 1152 strips. Only one bit could be stored to determine the status of each strip. The BAD-STRIP bit is 0 when the strip doesn't have any problem.

Download and Addressing:

The Bad-Channel LUT is downloaded through PCI3. Each 32-bit word from the PCI3 bus corresponds to 32 BAD-STRIP bit status.

While processing SMT data, this look up table is addressed with the 4-bit CHIP-ID and the 7-bit STRIP-ID. The BAD-STRIP bit is propagated to the GAIN-OFFSET table. In case of a defective strip, the GAIN-OFFSET is reset to force its output to a corrected value equal to zero.

L3 logic:

An other feature of this block is to output the entire table to the Level-3 logic of the CONTROL-LOGIC module. The access to the memory should be granted eventhough data are being processed. This isn't possible if the STC is in DOWNLOAD mode. When an L3-READ-BAD is received, an 8-bit binary counter is incremented to read out all the locations of the table and sent the data out to the CONTROL-LOGIC.

Output to L3 format

Note: When downloading, the data are manipulated to get to the L3 required format. The read back from this memory shows the 4-bit CHIP-ID and the 3-bit GROUP number in the upper double-byte word. There are 8 groups of 16 strips per chip.

<u>GAIN-OFFSET:</u>

Gain-Offset table:

The GAIN-OFFSET memory is a 2304X8 table. It is consulted while processing SMT data in order to correct the ADC values from good strips for offset and scale. This module allows a chip by chip correction of data.

Download and Addressing:

The look up table is downloaded through PCI3 prior to run the STC. It is addressed with the 4-bit CHIP-ID and the 8-bit DATA word. The 8-bit CORRECTED-DATA is then output to the STRIP-READER-FIFO. If a bad strip is seen in the incoming stream, the BAD-STRIP signal from the BAD-CHANNEL module is used as a reset flag. The output of the GAIN-OFFSET is forced to a zero value.

Inserting the Event Number:

A multiplexer has been inserted at the output of this module to select either the CORRECTED-DATA or the FRC-EVENT (event number) word. They are both 8-bit word and occupy the same bit location in the output format of the STRIP-READER.

The STATE-S2-S3 signal is used to make the selection:

- High:

  The STRIP-READER is processing data, the CORRECTED-DATA is selected as output of the GAIN-OFFSET.

- Low:

  The STRIP-READER finished processing data, the FRC-EVENT is output to the STRIP-READER-FIFO. The 8-bit event number is provided by the CONTROL-LOGIC.

STRIP-READER-FIFO:

Features and puprose:

The STRIP-READER-FIFO is a small 24X3 fifo. It has been inserted between the STRIP-READER and the CENTROID-FINDER for synchronization purposes.

The CENTROID-FINDER has been designed to run at a 50MHz frequency rate which is two times faster than the STRIP-READER output rate. For this reason a large fifo is not needed. The STRIP-READER-FIFO can only store up to 3 words.

A STRIPR-FIFO-EMPTY signal is generated by this module to inform the CENTROID-FINDER whether the fifo is empty or not:

- High:

  The STRIP-READER-FIFO is empty and the output word must be discarded.

- Low:

  A word is available in the fifo and is ready to be read out by the CENTROID-FINDER.

Output Format:

The STRIP-READER output format to the CENTROID-FINDER is shown here:

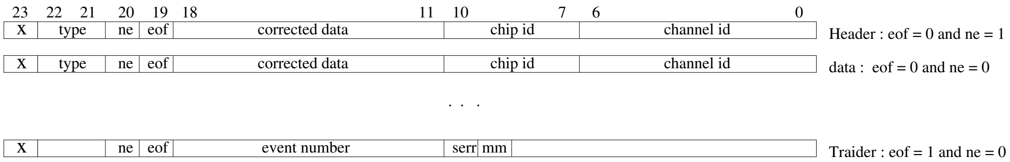

Figure 17: Strip reader output format

- bit 23 is not used for the time being.
- ne: NEW-EVENT-FLAG. This bit is high when the first word is being sent to the CENTROID-FINDER.
- eof: END-OF-EVENT. This bit is high when the end of event is reached. The trailer word must have this bit set to one. This is set by the WRITE-TRAILER signal from the DELAY-COUNTER.
- SERR and MM: Those bits are monitoring bits from the MONITOR module. They appear in the trailer word in order to be propagated to the following systems on the data path.

DELAY-COUNTER:

Definition:

The roads from the L1CTT and the data from the SMT are latched by an L1-ACCEPT signal from the trigger framework. By considering the cabling as well as the electronics involved in the process of transmisting, a significant timing skew may be introduced between these chunks of data.

The STC requires both blocks of data in order to produce the hits, a lost of synchronization may result in extracting SMT hits using the roads from the wrong event. This situation is a fatal error and must be fixed by pulling an SCL-INIT. The DELAY-COUNTER has been implemented to insure that the SMT data are processed with the correct set of roads. It is also used to assign an event number to the SMT data. A schematic of the DELAY-COUNTER is shown here:

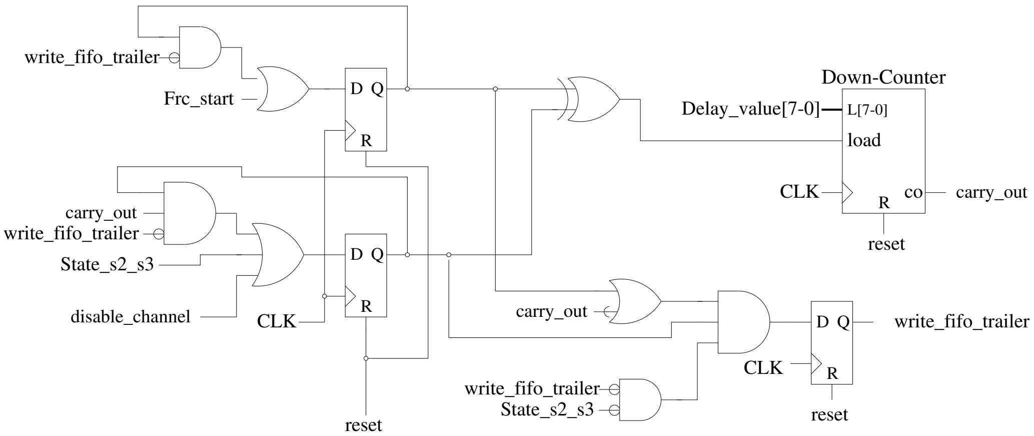

Figure 18: Delay Counter

In order to process SMT data with the roads of the same event, this module allows the timing skew between SMT and FRC data not to exceed a certain limit called DELAY-VALUE. This parameter is downloaded into the CONTROL-LOGIC when configuring the STC. It is the same for the 8 STC channels and at a 32 MHz clock rate, the maximum delay is about $8\mu s$.

Writing a trailer:

The logic of this block implements an 8-bit loadable binary counter and requires the following input signals:

- STATE-S2-S3:

  This signal is high when SMT data are being processed by the STRIP-READER. It can be used by this module to detect the receiving of SMT data.

- FRC-START:

  This signal is issued by the CONTROL-LOGIC when roads are being received. It also latches the FRC-EVENT (event number).

When either STATE-S2-S3 or FRC-START is high, the counter starts decrementing. The counter is stopped as soon as both of them are high. The DELAY-VALUE is also loaded at this point.

Two situations may be considered when writing the trailer:

- FRC data leads:

  In case of the FRC data coming first, the counter starts decrementing and stops as soon as the SMT data are received. The logic waits until the end of the event (STATE-S2-S3 low) to issue a WRITE-TRAILER to the STRIP-READER-FIFO. The WRITE-TRAILER signal writes the SMT data trailer that contains the FRC-EVENT as well as the SERR and MM bits form the MONITOR.

- SMT data leads:

  If the SMT data come first and are still being processed when the FRC block arrives, the logic waits until the end of the event to write the trailer. If the process is done before the FRC data are received, the trailer is written as soon as the FRC-START is high.

Keeping synchronization:

These situations are ideals and things get more complicated when one of the block of data is missing or when the delay expires:

- Delay expires:

  - SMT data late:
    If an FRC-START has been seen and the corresponding SMT data arrive more than $8\mu s$ later, a trailer is written as soon as the counter expires. The STC considers that no SMT data where received for that event. This block of SMT data will be associated with the next event.
  - FRC data late:
    If the process of SMT data is done and the FRC-START is received after the maximum delay has expired, no trailer is written. This block of SMT data is merged with the one that belongs to the next event. Both of them are processed with the same set of roads. Note that if the FRC data of the next event come on time, the logic is not able to recover. It is a problem of the trigger system and the SCL-INIT must be pulled.

- SMT data missing:

  In the situation where the SMT data are missing, a trailer is written when the delay expires. No hits are going to be generated with the associated roads.

- FRC data missing:

  No trailer is written eventhough the delay expires. This block of SMT data is then merged with the following one. Thus, both blocks are treated with the same set of roads.

Other features:

- Case of a disabled channel:

  The CONTROL-LOGIC expects 8 trailers overall, one from each of the STC channels no matter wich one are disabled. For this reason, a WRITE-TRAILER is generated when an FRC-START is received.

- VTM-IDLE signal to the CONTROL-LOGIC:

  This signal is used by the CONTROL-LOGIC to determine whether the channels are done processing SMT data. Each STC channel sets this signal to zero when receiving an FRC-START and to one when writing the trailer.

<u>L3-RAW-CORR:</u>

L3 data from Strip-Reader:

This module has been designed to generate the L3-CORRECTED-DATA and L3-RAW-DATA for Level-3 trigger:

- L3-RAW-DATA:

  The l3 raw data are the data as they are received by the STRIP-READER from the SMT

- L3-CORRECTED-DATA:

  The l3 corrected data are the data produced by the STRIP-READER.

Only one STC channel is selected to output l3 raw and corrected data. These data are sent out to the CONTROL-LOGIC which stores them. The l3 corrected and raw data have been merged at the STRIP-READER level so that only one buffer is needed. This option also simplifies the logic needed to transmit these data to the L3SDRAM.

Output Format:

The data are output as a 32-bit L3-RAW-CORR bus in conjunction with a VALID signal which is low-active. The lower 20 bits are dedicated to the l3 raw data and the remaining upper bits are for the l3 corrected data. The chosen format is the following (details are given in ANNEX 4):

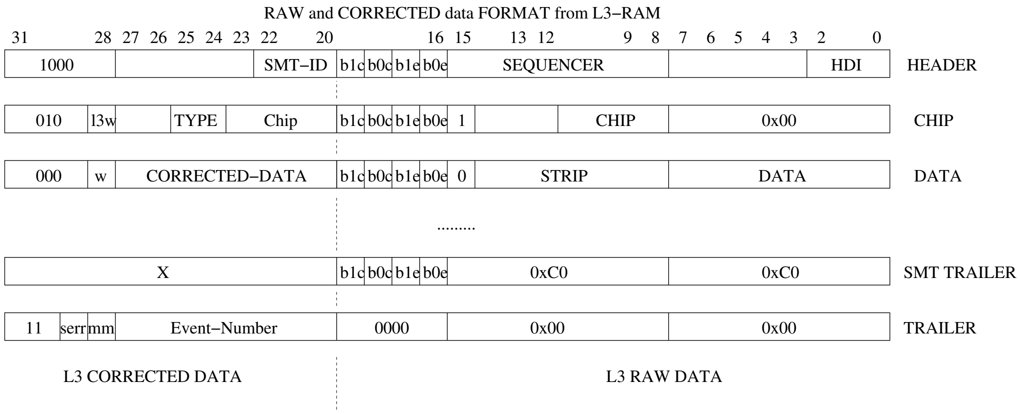

Figure 19: L3 Raw and Corrected data format

The format used here is of course a temporary one. The required format is reestablished before sending out to Level-3. The CONTROL-LOGIC is in charge of this task.

<u>MONITOR:</u>

Definition:

The MONITOR is used to provide monitoring information for one or several events. It implements a set of 24-bit binary counters to keep track of the different error flags generated and check on the activity of the system.

- MISMATCH-COUNT: Use to count the number of mismatch SEQ or HDI.
- ERROR-BIT-COUNT: Use to count the number of VTM-error bit set.
- CHIP-ZERO-COUNT: Use to count the number of

  - MSB unset when a CHIP-ID word is received.
  - Illegal CHIP-ID.
  - Zero byte missing after the CHIP-ID word.

- WRITE-FIFO: Use to count the number of strip data received from a particular CHIP-ID. There are 9 counters for each possible chip number (from 0 to 8).
- CENTROID-COUNT: Use to count the number of centroids produced in this event. The 2-bit CENTROID-TYPE comes along with this flag to determine whether this is an axial, stereo or z centroid. Three counters are implemented, one for each type.

Available counters:
Each type of monitored data is encoded into a 4-bit word as shown in the following table:

ERROR	ADDRESS	ERROR	ADDRESS
Mismatch	1010	Chip 6	0110
Error	1001	Chip 7	0111
Chip zero	1011	Chip 8	1000
Chip 0	0000	Undef centroid	1100
Chip 1	0001	Axial centroid	1110
Chip 2	0010	Stereo centroid	1101
Chip 3	0011	90deg centroid	1111
Chip 4	0100		
Chip 5	0101		

These 4-bit words are addresses that point to memory spaces allocated to every and each type of monitored data. One can notice that a memory space is storing the number of undefined centroids. These centroids can be produced in case of an ACCEPT-BAD-CHIP bit set to one in the CONTROL-LOGIC. These addresses are chosen to be consecutive in order to provide a read out of the monitored data in burst mode rather than as a series of individual data transfers.

Monitoring an event:

As soon as a monitoring flag is received, the corresponding address is generated by a 4-bit encoder and stored into a small fifo. The centroid information is stored into a separate fifo. Note that in order for this logic to work, these flags must be received at different clock cycles.

The next step is to read out a 4-bit word from one of the fifos and address the REGISTER1. The REGISTER1 is a memory that stores the 24-bit word count for each monitored data. The current value is incremented by one using a 24-bit adder. The result is then stored back into the REGISTER1. At this point an another word may be read out. The combination of the REGISTER1 and the adder provide a unique 24-bit binary counter used to count the 16 different types of monitored data. When latching the MONITOR, the content of REGISTER1 is transfered to REGISTER2 while REGISTER1 is cleared. The MONITOR is not stopped during this time because the fifos keep on storing the information. This principle allows the read out of the monitoring data through REGISTER2 at anytime without interrupting the process of an event.

The MONITOR is latched when the folloeing conditions are both met:

- CENTROID-DONE is received from the CENTROID-FINDER. This signal is high when all centroids have been processed.

- MONITOR-LATCH from the CONTROL-LOGIC.

Two other bits are generated by the MONITOR:

- SERR: high when one or several VTM error bit where set while receiving SMT data.
- MM: high when either a MISMATCH-COUNT or a CHIP-ZERO-COUNT has been detected during the process of an event.

These bits are generated with 2 simple registers that are reset after a WRITE-TRAILER high. These two bits are passed on to the CENTROID-FINDER and L3-RAW-CORR in the trailer word.

3.3 CENTROID-FINDER

The CENTROID-FINDER has been designed to find the clusters in the SMT and compute their centroid. The description of the clustering algorithm and the definition of a centroid are given in the introduction of this documentation. The CENTROID-FINDER gets its input data from the STRIP-READER module. These data are preprocessed SMT data (Bad channels rejected and data corrected).

Structure:

The CENTROID-FINDER is composed of eight sub-modules as shown on figure 20. This design uses a pipelined logic in which each data from the STRIP-READER is treated one at a time. The COMPARATORS and the PIPE-ACC are used to find the cluster and calculate their DENOMINATOR and NUMERATOR which we defined as the following:

- NUMERATOR = -2 D1 - D2 + D4 +2 D5
- DENOMINATOR = D1 + D2 + D3 + D4 + D5

Then the DIVIDER computes the two precision bits adds them to the existing 11-bit cluster center address (CHIP-ID + STRIP-ID of the center of the cluster).

The energy of the cluster is sent to the PULSE-AREA memory to determine the 3-bit pulse height. As the STRIP-READER, these two memories should be loaded prior to run the STC. The 13-bit cluster center address plus the 3-bit PULSE-AREA form a 16-bit CENTROID.

According to their types, the CENTROIDS are stored into the following fifos:

- AXIAL-CENTROID-FIFO for axial type of centroids .
- ZSTEREO-FIFO for the z and stereo type of centroids.

The PIPE-ACC provides also L3 information to the L3-CLUSTER-DATA module which passes them on to the CONTROL-LOGIC. The centroids stored into the fifos are output as L3 data to the CONTROL-LOGIC as well.

The sub-modules are detailed in the following sections.

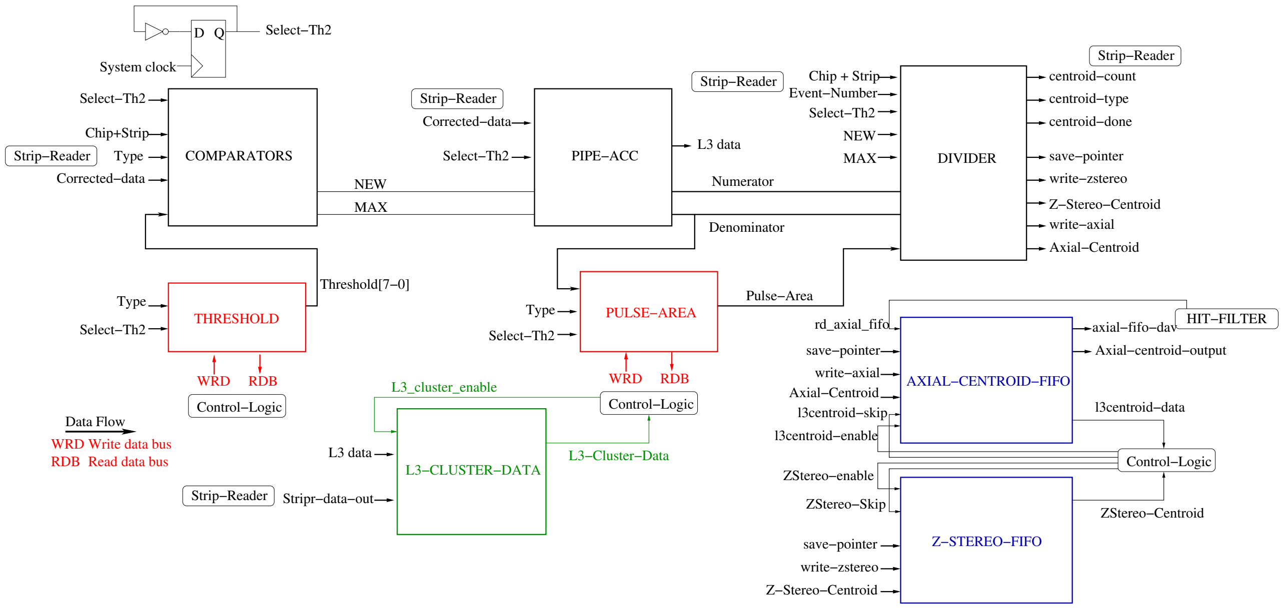

Figure 20: Centroid Finder

Important Signals:

In order to speed up the centroid calculation process, this module has been designed to run twice as fast as the STRIP-READER output rate ($\simeq$ 25MHz). Thus, the CENTROID-FINDER logic needs the following signals to be able to deal with a slower input rate:

- SELECT-TH2:

  This signal is generated from the system clock. It has a frequency rate of 50MHz. It is used to select which thresholds must be output from the THRESHOLD memory and also as internal clock for the CENTROID-FINDER module.

- FREEZE:

  When the STRIP-READER-FIFO is empty, the word that persists at the output is not valid and must be discarded. The STRIPR-FIFO-EMPTY is used as a valid signal (fig 21). When the fifo is empty the most of the system must be frozen especially the pipeline. The reason is that the pipeline must only store valid data for the accumulators. The CENTROID-FINDER is then stopped completely when the FREEZE signal is pulled.

- SELECT-TH2p:

  The SELECT-TH2p is the same signal as SELECT-TH2 but it is only high if STRIPR-FIFO-EMPTY is low. This signal is sent to every registers in order to stop updating their values if a FREEZE is high. Of course if the STRIP-READER-FIFO is empty but the end of event word has been read out already, the FREEZE signal cannot be set, the design is in a free-running mode (SELECT-TH2 = SELECT-TH2p).

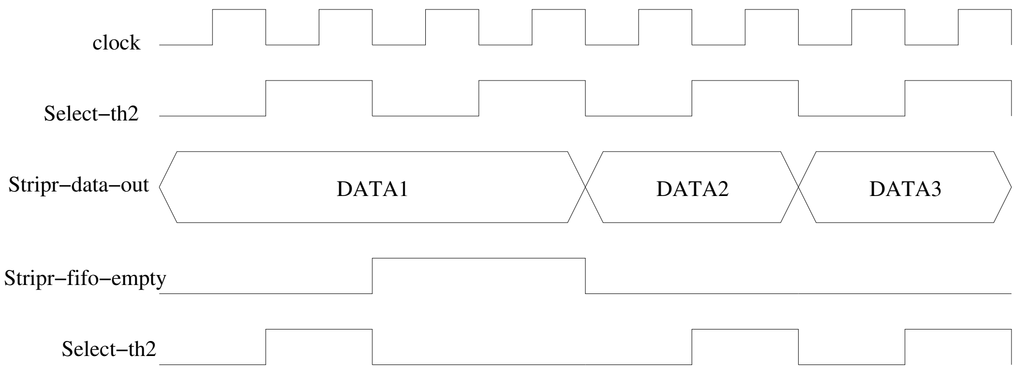

Figure 21: Strip Reader Output

THRESHOLD:

This memory stores the two thresholds involved in the clustering algorithm. There are six 8-bit thresholds in total, two for each type of data. The THRESHOLD1 is lower than the THRESHOLD2.

Download:

This memory is READ/WRITE and it is loaded through PCI3. When downloading the thresholds, a copy is also stored into in the CONTROL-LOGIC. Thus, the read back is showing the same information in the upper double-byte word from the PCI3.

Addressing:

While processing the STRIP-READER data, the memory is addressed with the 2-bit DATA-TYPE and the SELECT-TH2 signal. The DATA-TYPE determines which pair of thresholds are to be selected. The SELECT-TH2 is high when THRESHOLD2 must be output to the COMPARATORS module.

PULSE-AREA:

The PULSE-AREA memory stores the seven 11-bit thresholds used to generate the 3-bit PULSE-AREA. Each STC channel receives data from one silicon detector which can only output two types of data: axial and stereo or axial and z. Thus, 14 thresholds have to be downloaded overall. A schematic of the PULSE-AREA memory is presented figure **??**.

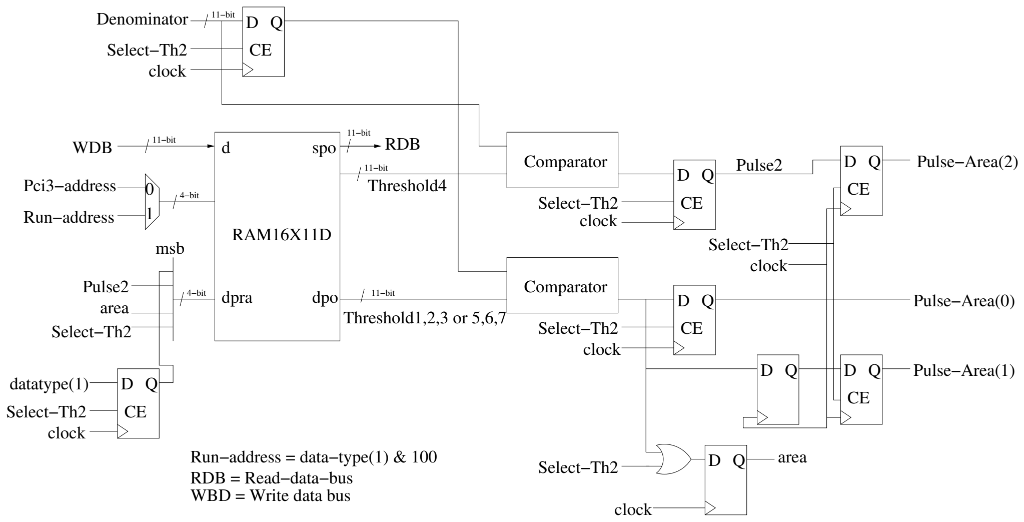

Figure 22: Pulse Area Memory

Download:

This memory space is also READ/WRITE. The thresholds are downloaded through PCI3.

Pulse height calculation:

While running the STC, the PULSE-AREA memory is addressed with the 11-bit DENOMINATOR and the 2-bit DATA-TYPE:

- DENOMINATOR:

  The DENOMINATOR is the sum of the ADC values within a cluster. So it is equivalent to the total energy of this cluster.

- DATA-TYPE:

  The MSB of the 2-bit DATA-TYPE determines which set of 7 thresholds is to be used. If it is one, the axial thresholds are used.

Then, the calculation of the pulse height is done within three SELECT-TH2 clock cycles (figure 23):

1. The DENOMINATOR is compared with the THRESHOLD4 and result is stored into a register as PULSE2. This bit is high if the DENOMINATOR was above THRESHOLD4.

2. If PULSE2 is high, the THRESHOLD6 is output to the second comparator otherwise it is THRESHOLD2. The result of this comparison is also stored into a register as PULSE1.

3. The following table displays which threshold is to be used to calculate the LSB of the PULSE-AREA:

PULSE2	PULSE1	THRESHOLD
1	1	THRESHOLD7
1	0	THRESHOLD5
0	1	THRESHOLD3
0	0	THRESHOLD1

The result of the comparison is registered as well.

The 3 bits are joined together to form the 3-bit PULSE-AREA word that is sent out to the DIVIDER module.

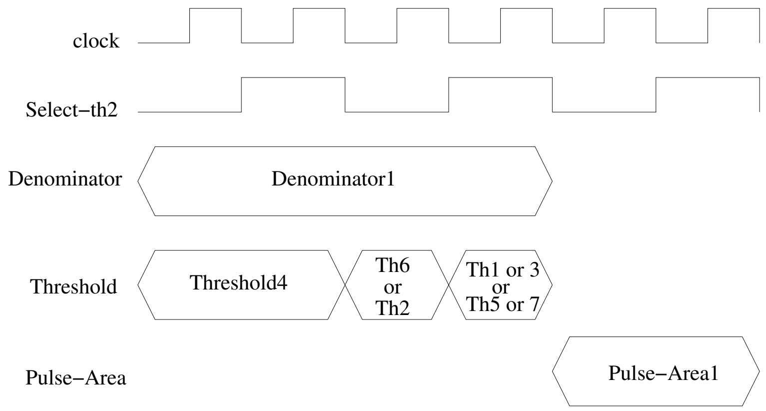

Figure 23: Pulse Area Calculation

<u>COMPARATORS:</u>

Definitions:

This sub-modules generates two bits that are associated to every and each input data word from the STRIP-READER.

- MAX: This strip is a possible cluster center.
- NEW: A new cluster must be started with this strip.

Conditions for setting NEW and MAX:

The conditions for setting a NEW or/and a MAX bit(s) are described in the following:

- NEW is set to one if:
  - The 8-bit CORRECTED-DATA is under THRESHOLD1.
    The corresponding strip cannot be included into a cluster. This strip is discarded and the cluster starts with the next strip. The NEW bit is high for the present strip and remains high for the next one.
  - END-OF-EVENT flag set to one.
    This word is the trailer and it should not be included in a cluster so its NEW bit is set to one. This also insure that the cluster that as in progress is forced to an end.

  - DATA-TYPE changes:

    In normal operation, the data from axial strips are received and then the data from z or the 90 degrees strips follow. When the change of DATA-TYPE occurs, a new cluster must be started.

  - The 11-bit address (CHIP-ID + STRIP-ID) is not contiguous with the previous one.

    This strip does not belong to the same cluster. A new cluster is started with this strip. Note that the algorithm treats every contiguous group of strips having their ADC value above THRESHOLD1 as a cluster.

  - NEW-EVENT-FLAG set to one:

    This is the first strip of the event so it obviously starts a cluster.

- MAX is set to one if:

  - CORRECTED-DATA is above THRESHOLD2 or LOCAL-MAX:

    If a strip has its ADC value above THRESHOLD2 it becomes the cluster center. The ADC value becomes the LOCAL-MAX which replaces the THRESHOLD2 as current limit. Thus the following words are compared with LOCAL-MAX instead of THRESHOLD2. If another strip has its CORRECTED-DATA above the LOCAL-MAX, it becomes the cluster center. The LOCAL-MAX remains until a NEW bit is high which indicates the beginning of a new cluster. The THRESHOLD2 is restored as upper limit at this point.

PIPE-ACC:

The PIPE-ACC sub-module is used to calculate the NUMERATOR and the DENOMINATOR of each cluster:

- NUMERATOR = -2 D1 - D2 + D4 +2 D5
- DENOMINATOR = D1 + D2 + D3 + D4 + D5

Where D1, D2, D3, D4 and D5 and the ADC values of each cluster strip. The strip associated with D3 is the cluster center. Note that if the cluster has less than 5 strips, the missing strips have their corresponding ADC value as zero. But it is required to have a cluster center.

The DENOMINATOR and the NUMERATOR are transmitted to the DIVIDER in order to generate the two precision bits added to the cluster center address.

The cluster are not relevant at this point, only their centroids are useful to the HIT-FILTER. Thus the clusters are sent out as Level-3 data to the CONTROL-LOGIC.

The pipeline:

The CENTROID-FINDER logic implements a 3-stage pipeline to calculate the NUMERATOR and the DENOMINATOR of the cluster. The number of stages is chosen by the fact that the weight that must be applied to each data for the calculation of the NUMERATOR for example, depends on its position in the cluster. The only relevant positions are the center and its first and second neighbors.

The 8-bit DATA as well as the NEW and MAX signals are stored into this pipeline. Each stage generates signals such as NEW1, NEW2, NEW3 and MAX1,MAX2 and MAX3 (figure 24). These signals will be used to control the accumulators. Each accumulator gets its input from the pipeline (DATA3, MAX3 and NEW3).

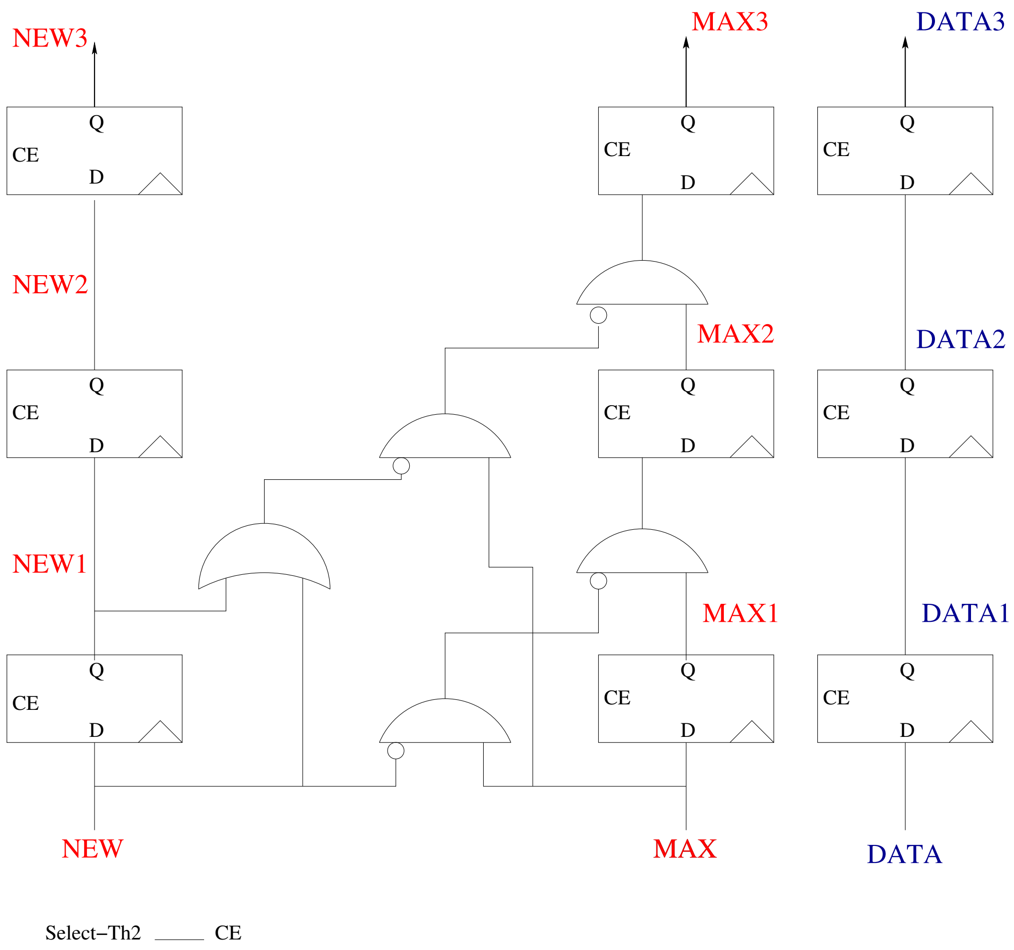

Figure 24: pipeline

Some combinatorial logic has been added to the pipeline in order to handle the situations displayed on figure 25

- Condition 1: A new maximum of the cluster is found. The MAX1 signal that is set for DATA1 must be reset.
- Condition 2: A new maximum is found after 2 words, the MAX2 signal has to be reset.
- Condition 3: A new maximum is found but not within the same cluster, the MAX2 signal should not be reset. Two single strip clusters are found here.

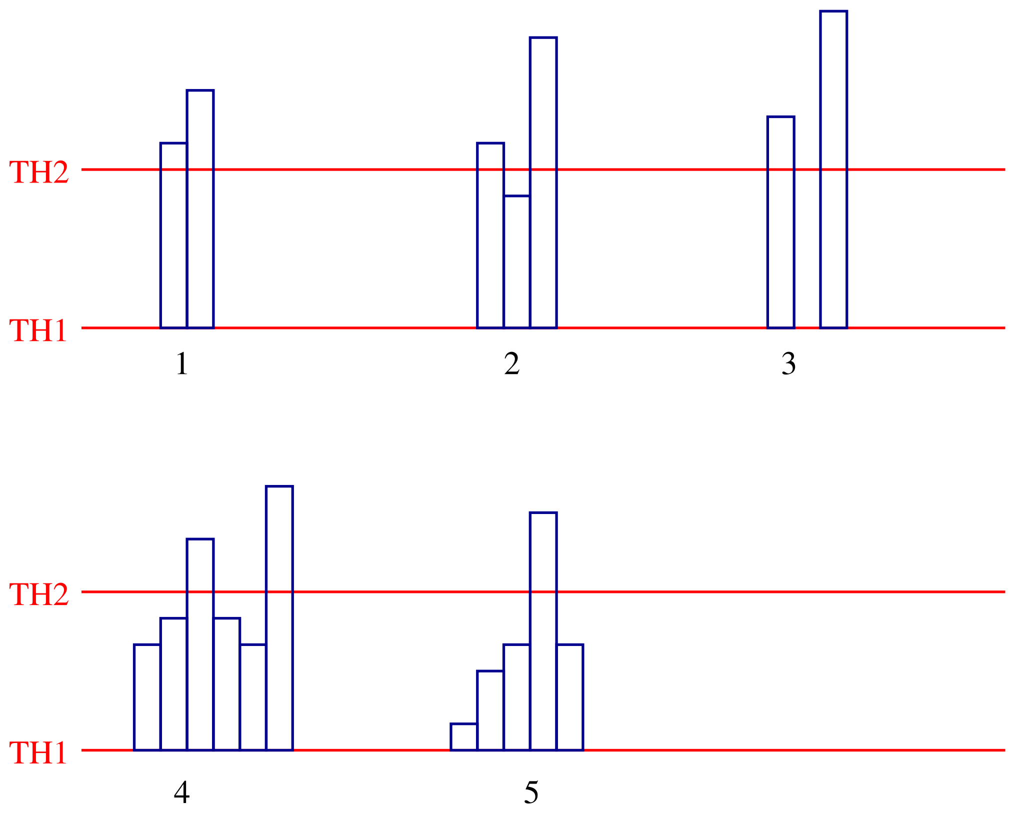

Figure 25: Possible Clusters

The accumulators:

The accumulators get their input from the pipeline. Two of them are implemented:

- ACC (accumulator): calculates the DENOMINATOR.
- WACC (weighted accumulator): generates the NUMERATOR.

NOTE: Details about the accumulators are given in ANNEX 5.

<u>ACC</u>:

The 11-bit DENOMINATOR is the sum the ADC values of each cluster strip: DENOMINATOR = D1 + D2 + D3 + D4 + D5. It takes as many SELECT-TH2 clocks as the number of strips in the cluster:

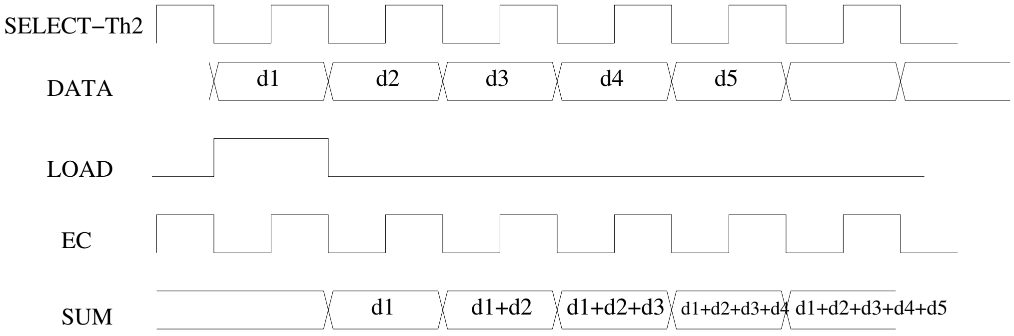

Figure 26: Denominator

When a cluster starts, the ADC value is loaded into ACC. The following data are added until the end is reached. The next cluster first ADC value is loaded into ACC which overwrite the previous value of SUM. The signals produced by the pipeline are used to determine the equations of the ACC ports. These equations are reviewed in detail in the ANNEX 6. Note EC, LOAD and SUM are the ports of ACC which are described in the annexe.

<u>WACC</u>:

The 11-bit NUMERATOR is a weighted sum of the ADC values of the cluster: NUMERATOR = -2 D1 - D2 + D4 +2 D5. The principle is the same as the previous accumulator but the complication comes with the weight 2. In order to produce the NUMERATOR in sync with the DENOMINATOR, it must operate within the same number of SELECT-TH2 clock. Thus, the WACC must run at a faster clock speed: the system clock. The WACC has to be able to handle all possible types of clusters so the equations used for its ports get very complicated, a special annexe is devoted to it (ANNEX 6). As the ACC accumulator, it takes as many SELECT-TH2 clock cycles as the number of strips in the cluster.

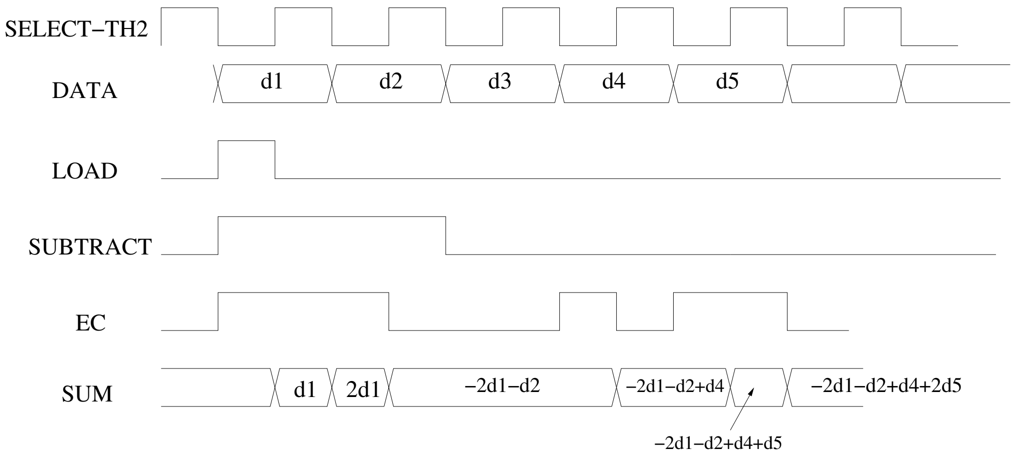

Figure 27: Numerator

Note that the LOAD, SUBTRACT, EC and SUM are the ports of the WACC accumulator which are described in the annexe.

These accumulators have the advantage of delivering the NUMERATOR and the DENOMINATOR right after the last word of the cluster. No additional delay is needed so the 2-bit precision can be evaluated right away. An alternative approach would have been to find the 5 data that compose the cluster and then calculate the NUMERATOR and the DENOMINATOR. In this case a pipeline is not needed which avoids complications like the fact that the pipeline must be frozen when no valid input data are available. But this approach has the advantage of implementing only two 11-bit adders which is not the case of the other method.

The 3 strips mode:

The 3 strips mode is another operating mode for the STC. It fixes the maximum size of the cluster to 3 strips. The DENOMINATOR and NUMERATOR words are truncated to the following expressions:

- DENOMINATOR = D1 + D2 + D3.
- NUMERATOR = - D1 + D3.

D2 is the ADC value of the cluster center strip.

L3 output:

The PIPE-ACC sub-module is also generating Level 3 cluster data. This sub-module is in charge of sending out the relevant l3 information to the L3-CLUSTER-DATA module. This includes the data from the pipeline as well as the signals used by the accumulators.

<u>DIVIDER:</u>

This sub-module is in charge of the 3 following tasks:

- Calculate the 2 precision bits from the DENOMINATOR and NUMERATOR.
- Store the cluster center address.
- Format centroids for the HIT-FILTER module.
- Write the centroids into the FIFOs.
- Monitoring information.

Division:

The logic implemented to calculate the two precision bits is shown here:

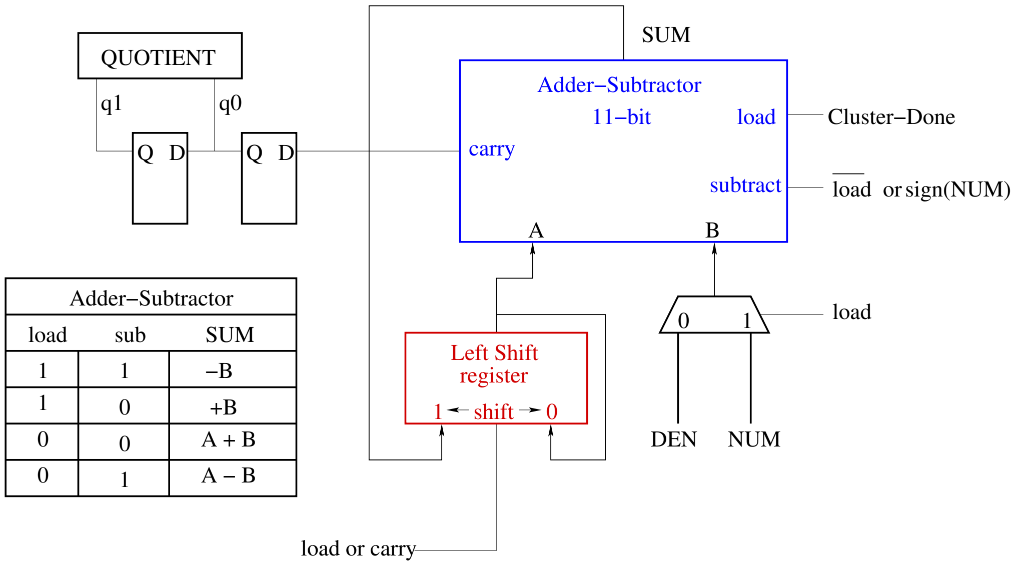

Figure 28: Divider

The DIVIDER implements an 11-bit adder-subtractor and some other combinatorial logic like a multiplexer, a couple of registers and a shift register. The 11-bit adder-subtractor used here is identical as the one implemented in the PIPE-ACC module. This peace of logic performs the calculation of the 2-bit resolution following the same rules as a simple binary division:

The division of A by B is done by subtracting B form A and look at the sign of the result:

- If it is positive: The first bit of the quotient is 1 and the result of A-B gets left-shifted.
- If it is negative: The first bit is 0 and B gets left-shifted.

After left-shifting the subtraction takes place again to evaluate the next quotient bit.

The DIVIDER reproduces exactly this procedure. It calculates the absolute value of the quotient which simplifies the logic. It keeps track of the sign of the NUMERATOR at the moment of adding the 2 precision bits to the cluster center address. The fact that the ratio of NUMERATOR and DENOMINATOR is always less than one makes it possible to generate these two bits within 3 system clocks.

Saving the cluster center address:

While the 2 precision bits are being calculated, the address of the cluster center is stored into a small fifo. The trailer containing the event number is stored into this fifo as well. When the quotient calculation is done, a word is read out from the fifo. The trailer is treated as a single strip cluster so that the same logic is used to read it out.

Format the data:

Once the division is done, the 2-bit QUOTIENT and the 11-bit cluster center address are sent to an adder-subtractor. By looking at the sign of the NUMERATOR of the corresponding cluster, the two words are either subtracted (if negative) or added. In case of a trailer as for a single strip cluster, the 11-bit word goes through untouched (the 2 precision bits are irrelevant). The 13-bit address and the 3-bit PULSE-AREA are joind together to form a 16-bit CENTROID.

Writing data out:

There are two types of centroids stored into two separate FIFOs:

- AXIAL-CENTROID which are stored into the AXIAL-CENTROID-FIFO.
- ZSTEREO-CENTROID which are stored into the ZSTEREO-FIFO.

The DIVIDER generates either a WR-AXIAL-FIFO or a WR-ZS-FIFO by looking at the MSB of the DATA-TYPE.

When the trailer is seen, it is written into both FIFOs. A SAVE-WRITE-POINTER is also issued at that point. This signal will be discussed in the section that describe the FIFOs.

Monitoring signals:

The DIVIDER is also responsible for sending the following information to the MONITOR sub-module:

- 2-bit CENTROID-TYPE is sent along with a CENTROID-COUNT pulse every time a CENTROID is being written into the FIFO. This is used to monitor the number of centroids produced and their type.
- CENTROID-DONE: This signal is issued when the trailer has been written. The end of event has been reached and the MONITOR can be latched by the CONTROL-LOGIC.

<u>AXIAL-CENTROID-FIFO and Z-STEREO-FIFO:</u>

The AXIAL-CENTROID-FIFO and the Z-STEREO-FIFO are used to store the axial centroids and z-stereo centroids respectively. The axial centroids are sent to the HIT-FILTER and the z and stereo centroids are sent to the CONTROL-LOGIC. All of them are also propagated as Level-3 data.

Storage size:

These fifos must be big enough to store several events because no storage have been implemented before on the data path. Furthermore, the STC must keep up with the following cases:

- Events that produce a lot of centroids (many active strips in the SMT).
- Several events of this type received consecutively.

These are the reasons for implementing fifos with the following features:

	AXIAL-CENTROID-FIFO	Z-SETEREO-FIFO
Size	2048X16	1024X16
Maximum number of Events stored	6	4 for z strips 5 for stereo strips

The number of centroids that are produced per event in the worst case are:

- Axial ⇒ 320.
- Z ⇒ 256.
- Stereo ⇒ 192.

Note that those numbers are produced when considering the clustering algorithm described in the introduction of this documentation.

In average 2 or 3 centroids are found in the incoming data per event (look at Meena's paper) so the storage is most likely to keep up.

Full signal:

Extra logic must be added to handle situations where the FIFOs are getting full. If the fifo is almost full (99%) a FIFO-FULL signal is pulled internally. This signal high will result in not storing any more centroids but only the trailers of each event. Up to 16 trailers can be stored which is the maximum number of events that the STC is able to handle (check if SCL is pulled). The data are being truncated because the lower level of the trigger (Level-1) rate can't be slowed down in order to limit the amount of incoming data.

Saving the trailer:

The CENTROID is a 16-bit word which includes the 13-bit centroid address and the 3-bit pulse height. There is no spare bits to identify the trailer word. To be able to recognize it among the centroid data, the 2 MSBs of its CHIP-ID are set to one. Note that the CHIP-ID does not have any meaning in the trailer word. The present option avoids the implementation of an END-OF-EVENT bit which allows to save up to 16 Dual-Port memories for the whole STC design. This END-OF-EVENT bit is added to the centroid word when reading out the FIFOs.

Level 3 logic:

The AXIAL-CENTROID-FIFO special feature:

A regular FIFO would use one port for writing and the other for reading. In this case, no port would be left for the Level-3 read out of axial centroids and an extra memory space would have been implemented for this purpose. This FIFO has been designed in such way that the same memory space is used for the Level2 and Level3 data. The same port is used to write and read axial centroids. The other port is then free to be used by the Level3 logic. Of course, this needs extra logic but this option saves a lot of memory ressources. An ordinary FIFO about this size would need two 11-bit binary counters to generate the write and read addresses. Here a special loadable counter plus an additional register are used to generate both write and read addresses for the Level2 port. The Level3 port implements a simple 11-bit binary counter.

Note that the Z-STEREO-FIFO is a regular fifo because the output goes only to the CONTROL-LOGIC which takes care of separating information for the ZVC and the Level-3 trigger.

Read out logic:

The CONTROL-LOGIC is in charge of collecting the Level3 axial, z and stereo centroids as well as the z centroids for the ZVC. It reads out data from the CENTROID-FINDER fifo's by sending the following signals:

- L3-CENTROID-ENABLE ⇒ AXIAL-CENTROID-FIFO.
- ZSTEREO-ENABLE ⇒ Z-STEREO-FIFO.

After consulting the L3-CONFIG word, some of the Level3 data are not needed for a particular event. The CONTROL-LOGIC should be able to read the l3 centroids for the next event if required without having to read out the data of the previous event. For this reason, the location of the beginning of each block of data is saved at the end of every event. The DIVIDER sub-module issues a SAVE-WRITE-POINTER in order to save the NEXT-EVENT-POINTER word into a small fifo. The CONTROL-LOGIC has to send the following signals in order to jump to the location of required data:

- L3-CENTROID-SKIP ⇒ AXIAL-CENTROID-FIFO.
- Z-STEREO-SKIP ⇒ Z-STEREO-FIFO.

This logic saves a lot of clock cycles needed for the read out of centroids in the STC.

Output format:

The axial centroids are sent to the HIT-FILTER and the z and stereo centroids go to the CONTROL-LOGIC. All types of centroid use the following output format:

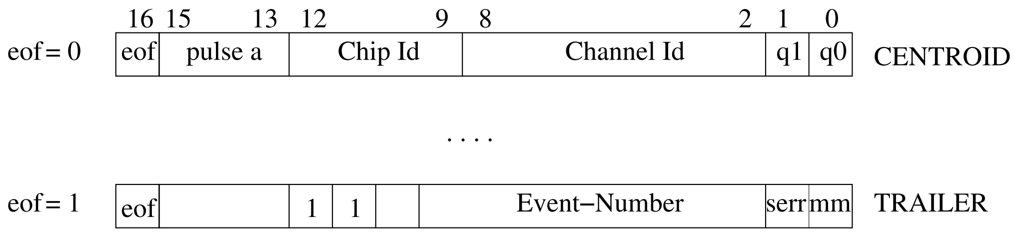

Figure 29: Centroid Format

The same format is used for the Level3 information.

L3-CLUSTER-DATA:

This sub-module is used to generate L3 cluster data to the CONTROL-LOGIC. Only one STC channel gets selected to output clusters in order to limit the amount of information to be stored. A single FIFO can then be implemented in the CONTROL-LOGIC rather than having 8 fifos in each CHANNEL-LOGIC. In order to facilitate the read out, the clusters are not reconstructed at the CHANNEL-LOGIC level. The present sub-module sends out the necessary information to form the clusters in the CONTROL-LOGIC and store them. This information consists of data from the PIPE-ACC sub-module. The format used is the following (details are given in ANNEX 4):

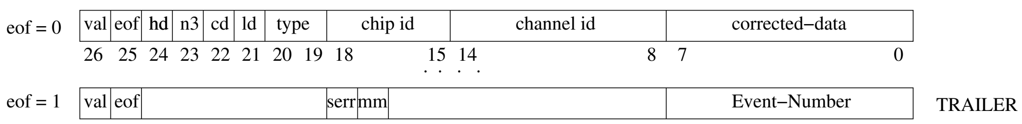

Figure 30: L3 Cluster format

3.4 HIT-FILTER

The hit filtering consists of comparing the axial centroids from the CENTROID-FINDER with the road limits from the CONTROL-LOGIC. Each road word is extracted by the PCI1-IF and used to address a 128kX22 look up table. Each CHANNEL-LOGIC obtains for each road word a lower and upper range limit. Each limit is an 11-bit word which correspond to a 4-bit CHIP-ID and 7-bit STRIP-ID on a particular silicon detector. The STC has been designed to treat as many as 48 roads per event. Each axial centroid is then compared with all of the range limits of the event. If it falls within one of the possible road limits, it is considered as a hit. A hit can be defined as a centroid in the SMT that is consistent with a road from the CFT. The hits are stored and passed on to the TFC via PCI2.

The HIT-FILTER module implements the following logic (also shown on figure 31):

- ROAD-FIFO:

  The ROAD-FIFO is used to store the range limits from the CONTROL-LOGIC. This fifo has been designed to store several events.

- Two 11-bit Comparators:

  A pair of comparators are provided to compare each centroid with both range limits. Note that the filtering takes place in a serial way which consists of comparing one centroid with all the roads of an event.

- HIT-FIFO:

  A fifo is implemented after the comparators in order to store the hits found. This block is connected to a bus which carries data up to the CONTROL-LOGIC.

Four STCs are connected to the same read out bus which goes to the CONTROL-LOGIC. The end of this section presents the read out procedure for the hits.

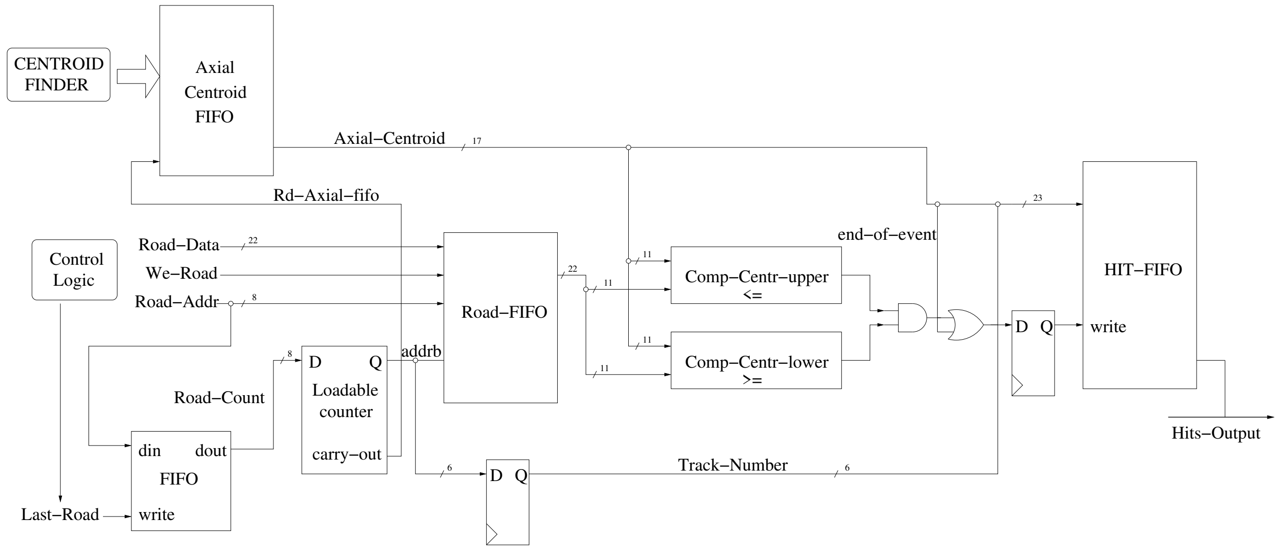

Figure 31: Hit Filter

3.4.1 Storing the road information:

The CONTROL-LOGIC provides to following information:

- ROAD-DATA: This is a 22-bit bus which carries the pairs of 11-bit limits.
- ROAD-ADDR: Each 22-bit word is sent along with an 8-bit ROAD-ADDR. This address indicates the location where the roads are to be written.
- WE-ROAD: This signal is a write enable signal.
- LAST-ROAD: This bit is high when the last road of an event is being transfered.

The ROAD-FIFO is a 256X22 fifo that is able to store up to 4 events. The memory space of this fifo is actually separated into 4 blocks of 64 words. Each block is dedicated to store the roads for one event. Note that the maximum number of roads can be moved to 64 without adding extra memory space or logic. The division of the memory space into 4 blocks makes it easier to spot the beginning of an event.

The LAST-ROAD bit is used to save the corresponding ROAD-ADDR into a small FIFO as ROAD-COUNT. The ROAD-ADDR of the last word is equal to the number of roads in the event. The ROAD-COUNT is loaded into the counter used to loop over the range limits of an event.

3.4.2 Filtering process:

The filtering is started when the following conditions are met:

- All the road limits of an event are stored into the ROAD-FIFO.
- At least one axial centroid is present in the AXIAL-CENTROID-FIFO.

1. The process starts by loading the ROAD-COUNT into the 6-bit counter designed to loop over all roads. The 2 MSBs of the ROAD-COUNT select which 64-word block contain the roads of this event. At this point the HIT-FILTER lowers its IDLE signal and waits for the first centroid.
2. An axial centroid is read out and injected into both comparators. Note that the 2-bit precision are not included when comparing.
3. The counter starts decrementing its pointer value. A new range limits is sent to the comparators every clock cycle (10ns).
4. If the centroid falls into one of the intervals, it is written into the HIT-FIFO. The 6-bit TRACK-NUMBER is added to the centroid word before storage. A TRACK-NUMBER is associated to every road and its value goes from 0 to ROAD-COUNT.

When the input centroid happends to be the trailer word, it is written into the HIT-FIFO regardless of the output of the comparators. An END-OF-EVENT bit is added to the centroid, a hit is then a 23-bit word. The HIT-FILTER goes back to an IDLE state unless the roads of the next event are available. At that point, if an axial centroid is ready, the filtering for the next event begins.

3.4.3 Storage:

The HIT-FIFO is a 512X23 fifo with the following features:

- HIT-FIFO-FULL signal:

  A full flag is implemented and goes high when 94 of the memory space is filled. This situation inhibits the reading out of centroids so that no more hits can be produced.

- HIT-WRD-CNT:

  The hit word count is used by the read out logic. It tells how many hits are present in the HIT-FIFO. It is used to generate the HIT-FIFO-FULL and the FIFO-EMPTY signals.

3.4.4 Hit read out procedure:

The strategy used to output the hits to the HIT-READ-OUT module of the CONTROL-LOGIC is described in the following section. A schematic is also provided on figure 32).

Definitions:

The STC is divided in two sides with 4 channels each. The four STC channels output hits on a common 26-bit bus called HITS-OUTPUT. This bus is shared between these channels which means that priority access must be defined. Every STC channel has to communicate with the three others using the following signals:

- BUS-GNT: bus granted $\Rightarrow$ Input signal.
- BUS-REQ: bus request $\Rightarrow$ Output signal.

Procedure:

Here is the hit read out procedure for a single STC channel:

1. HIT-FIFO not empty:

   The BUS-REQ is pulled and propagated to the other channels to let them know that the bus is requested.

2. The bus is granted:

   When the BUS-GNT signal comes back high the bus is not used by other channels and it is free to to be driven. The DRIVE-BUS signal is set and the hit is output.

3. Prepare next hit:

   A DRIVE-BUS high results in moving the pointer to the location of the next hit depending on the status of the HIT-FIFO:

   - Empty: The BUS-REQ goes away as well as DRIVE-BUS, the bus is free for other user $\Rightarrow$ go back to step 1.
   - Not Empty:
     The pointer has been moved and the next hit is ready to be sent out $\Rightarrow$ go to step 4

4. More hits to output:

   In case of more hits are to be output and if the BUS-GNT remains high, the DRIVE-BUS remains high as well and hits are sent out.

   Then two situations may occur:

   - BUS-GNT goes low ⇒ go to step2
   - fifo empty ⇒ go to step 1.

When the trailer is sent, a TRAILER-SENT signal is set to disable the DRIVE-BUS signal as well as the BUS-REQ signal. No more hits are being sent after the trailer until a HIT-DONE from the HIT-READ-OUT is received which means that all the hits of the event have been collected. At that point, the hit read out may start for the next event.

Priority access:

If the channels request the bus at the same time, priorities have to be defined:

- STC channel 0:

  The channel 0 gets the bus as soon as it has hits present in its fifo.

- STC channel 1:

  The channel 1 gets to output hits only if channel 0 is done or does not have hits for the time being.

- STC channel 2:

  If neither channel 0 nor channel 1 have hits, channel 2 can drive the bus.

- STC channel 3:

  This channel has the lowest priority and it ouputs hits only when the three other channels are not using the bus.

The priority goes down with the SMT-ID. The same scheme is repeated for the right side of the STC where channel 4 has maximum priority over the bus.

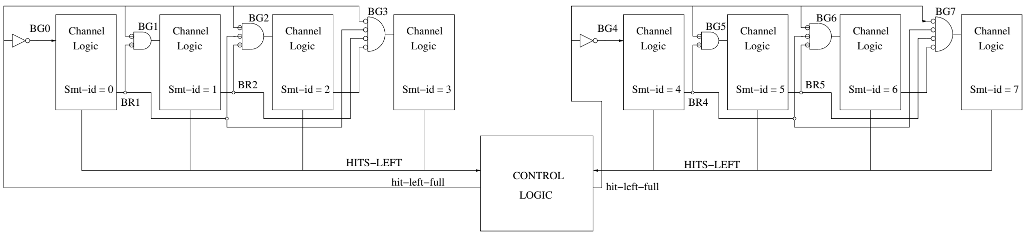

Figure 32: Hit read out

Maximum Number of hits per STC:

Only 256 hits can be produced by the STC to the TFC. So an extra feature must be added to the HIT-FILTER:

As soon as the hit count in the HIT-READ-OUT reaches 256, a YANK-HITS signal is pulled and sent to all STC channels. All HIT-FIFOs must be cleared of the hits belonging to this event. Only the trailers must be sent through because the HIT-READ-OUT logic uses them to produce a main hit trailer.

Ouput Format:

The hits are output to the HIT-READ-OUT module of the CONTROL-LOGIC with the following format (Detail in ANNEX 3):

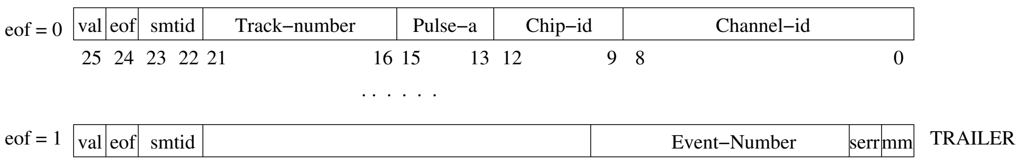

Figure 33: Hit Format

This format is of course used internally and the SMT-ID will be used to select the correct SEQUENCER and HDI ids to finalize the format. This way, information like SEQUENCER and HDI do not have to be carried every where using memory resources as well as precious horizontal lines in the FPGA.

4 CONTROL-LOGIC

The CONTROL-LOGIC manages the process of an event through the STC. It implements the 3 PCI interfaces and generates the control signals to the STC channels. A schematic of the CONTROL-LOGIC is shown figure 37. The PCI1-IF module is used to establish the interface with the PCI1-Altera-Chip. It converts the FRC data into road information consulting an external SDRAM. The road information is then sent to the STC channels. The interface logic with the Road lUT is implemented in the PCI1-IF as well. As the name suggests, the PCI3-IF is used to implement the interface with the PCI3-Altera-Chip. The PCI3 bus connects the STC with the user system (external CPU) and the Level-3 trigger(via VBD). All of the downloaded parameters are carried by this bus to every memory in the STC. The read back from each memory is provided. The other function of this module is to provide an access to the PCI3 when Level-3 data are to be output. The PCI3-IF has also control registers that are used to set up the STC in different operating modes. The logic used to output hits and z centroids is also implemented here. This part of the CONTROL-LOGIC is referred as Level-2 Logic. It has tow sub-modules used to store Hits and Z centroid. The PCI2-OUTPUT module outputs them to the PCI2-Altera-Chip. The Level-3 logic collects all Level3 data generated while processing an event. These data are thrusted to an external SDRAM. This L3 ram stores the data temporarily. The read out of this data is executed when the PCI3 is available. The data are also formatted as expected by the Level-3 trigger. While processing the data, the CONTROL-LOGIC is quite independent from the STC channels. The CONTROL-LOGIC does not intervene in the clustering process. It is used to ignite the filtering logic by sending out the roads and wait until Level-2 and 3 data are present in the buffers. The following sections will discuss of all the modules implemented in the CONTROL-LOGIC.

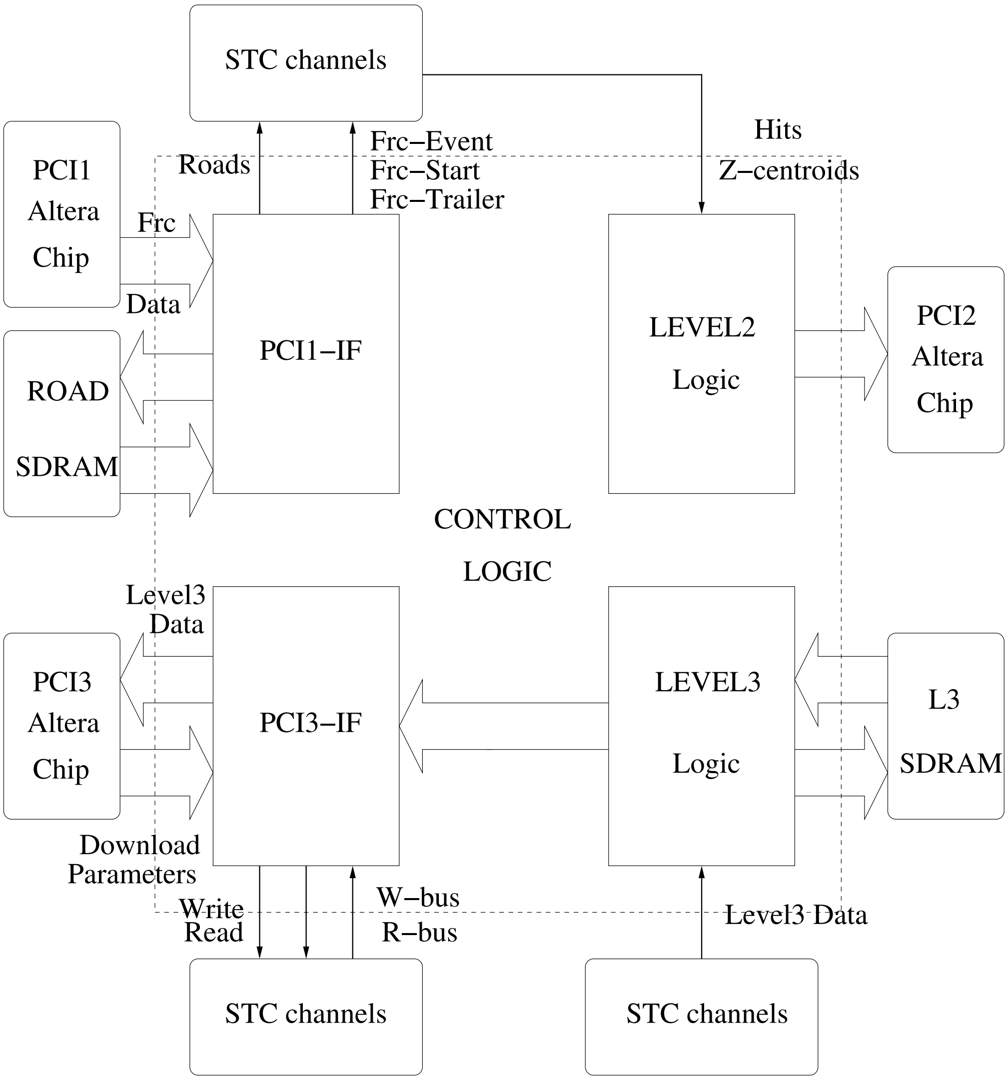

Figure 34: STC control logic

4.1 PCI3-IF

The PCI3-IF module provides the interface with the PCI3-Altera-chip. The necessary logic used to download and read back data from the parameter memories is implemented. The input PCI3 address is decoded to access the memory location. When running the STC, this PCI3-IF is also used to send out L3 data to the Level-3 trigger system. It also implements control registers used to set up the STC in different operating modes. Some extra logic is implemented for the testing mode of the STC. It includes also the logic for the SCL-INIT trigger framework procedure.

4.1.1 Memory access

The connection between the PCI3-IF and the PCI3-ALtera-Chip is established by a 32-bit STC-DATA bidirectionnal bus. The STC-CTRL bus carries the 12 available control signals: the bits 0 to 6 are input and the bits 7 to 11 are output to the PCI3-Altera-Chip. Although the PCI3 works at 33MHZ, the communication between the two chips is performed at a 50 MHz clock. When data are to be downloaded, the Altera chip transmits the address of the location followed by the data word through the 32-bit STC-DATA bus. The control bits STC-CTRL0 and 1 are used to identify the information that is being carried by the bus and determine the operation to be performed:

STC-CTRL[1-0]	**BUS CONTENT**	**OPERATION**
11	DATA	WRITE
10	ADDRESS	WRITE
01	ADDRESS	READ
00	X	NONE

The input data look like this:

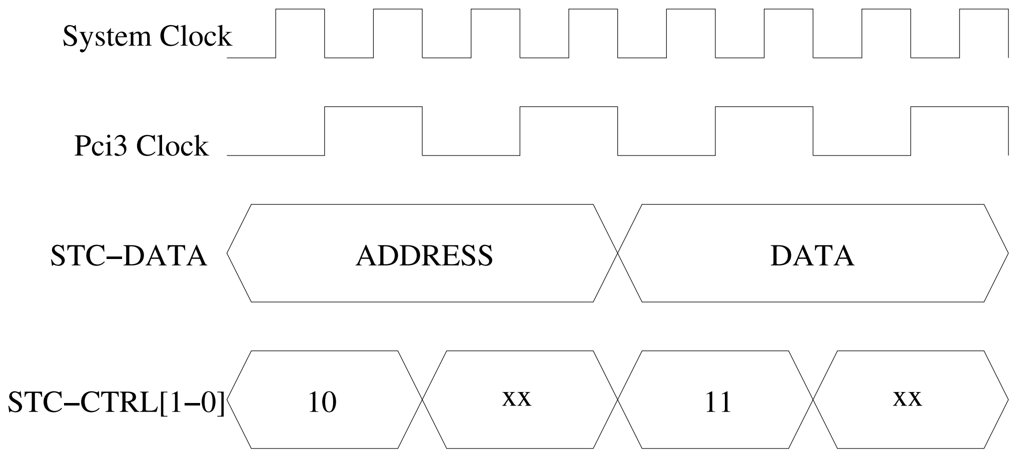

Figure 35: PCI3 input data

The address received from the PCI3-Altera-Chip is an 18-bit word. The bits 31 and 30 of the address word are used as control bits for the PCI3-IF:

- STC-DATA(31):

  This bit is set to one if a write operation is to be executed. The write operation is done when the corresponding data word is received. The use of this bit is justified by the fact that the access to certain memories is more complicated. The write operation may need the execution of subsequent commands. It is used here to inform the PCI3-IF that a write operation will be done with the input data.

- STC-DATA(30):

  This bit is high if a burst access mode is going to be executed.

When the PCI master accesses the STC, consider as a target here, the STC-CTRL(2) is set to one. This bit is the PCI-CYCLE that remains high as long as the PCI Master is accessing the STC. The PCI-CYCLE is sent out to the rest of the design as PCI3-CYCLE. While this signal is on, the address and the data from PCI3 are propagated to the STC. Note that the present design supports both write and read burst mode.

When the address is received, the 14-bit ADDR and ADDR-R address are generated. ADDR-R is the same address as ADDR but registered by on STROBE clock. This STROBE clock is a 50MHz clock generated from the system clock. The bits 17 down to 2 of the PCI3 address are used to generate the bits 14 to 0 of ADDR. The ADDR as well as the ADDR-R are sent out to the rest of design in conjunction with a STROBE signal and a IS-HIGH signal as shown on figure 36. An additional signal is sent, the write or read signal depending on the nature of the operation ongoing. The ADDR address is actually decoded to determine which memory is to be accessed. The corresponding read or write signal is sent in conjunction with the ADDR-R address.

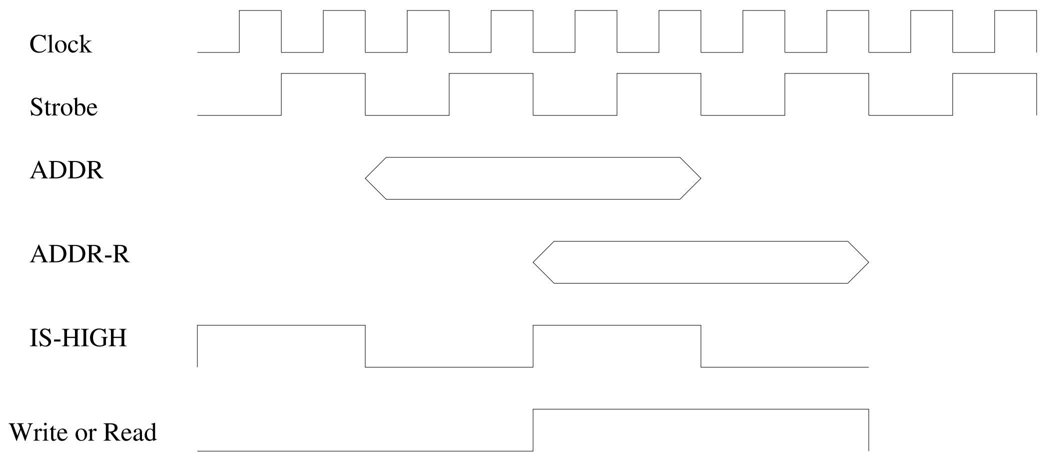

Figure 36: ADDR and ADDR-R addresses

Here are listed the write and read signals available:

WRITE/READ signals	**MEMORY**	**DESCRIPTION**
WR-GAIN-OFFSET RD-GAIN-OFFSET	GAIN-OFFSET	stores corrected data
WR-TEST-DATA RD-TEST-DATA	SMT-IF	stores the test SMT data
WR-BAD-CHANNEL RD-BAD-CHANNEL	BAD-CHANNEL	stores the bad channel status bit
WR-PULSE-AREA RD-PULSE-AREA	PULSE-AREA	stores the Pulse Area thresholds
WR-DATA-TYPE RD-DATA-TYPE	DATA-TYPE	stores the type of data
WR-THRESHOLD RD-THRESHOLD	THRESHOLD	stores two thresholds for clustering
WR-SEQ-HDI RD-SEQ-HDI	SEQ-HDI	stores the Sequencer and Hdi ids
RD-MONITOR	MONITOR	stores the monitored data
WR-ROAD-PA RD-ROAD-PA	PA register	stores ROAD LUT page address
WR-CSR0 RD-CSR0	RUN-CTRL register	stores STC control signals
CMD RD-STATUS	MISC-CSR	executes commands reads status bits
WR-INIT-TIME RD-INIT-TIME	INIT-TIME register	stores the INIT-TIME value for SCL-INIT
WR-DELAY RD-DELAY	DELAY-VALUE register	stores the DELAY-VALUE
WR-L3CONF RD-L3CONF	L3CONFIG	stores the Level 3 configuration word
WR-TEST-REG RD-TEST-REG	TEST-REG	This register stores the test options
RD-LRB-MON	LRB-MON	stores link receiver monitored data

A complete memory map of the design is also available 54.

The 32-bit downloaded data from the PCI3 cannot be sent to the memories as it is. The reason for that is the Xilinx Virtex familly provides Block Memory with a fixed size of 4096 bits. The size of the ports may varied from 1 to 16-bit. A 32-bit port can not be implemented. Thus every input PCI3 data is spited into two double-byte word and carried to the memory through a 16-bit W-DATA bus. Each double-byte word is then sent in conjunction with a IS-HIGH signal which is high when the W-DATA bus drives the upper 16 bits of the input PCI3 data. Another 16-bit bus R-DATA is implemented for the read back from these memories. A 32-bit word is reconstituted in the PCI3-IF before output to the PCI3-Altera-Chip.

For the Road LUT memory, things are slightly different: the RD-ROAD-DATA is not issued as the ADDR address is decoded. When a read command is received from the PCI3-Altera-Chip, the START-READ-SIGNAL is sent to the ROAD-RAM signal. Once data are available, ROAD-RAM-AVL is high, and the RD-ROAD-DATA is issued. A RD-ROAD-DATA high will provide the access to the

R-DATA bus to the ROAD-RAM read out logic. The difference with the other memories is that it is external and the access procedure is more complicated. In this case the burst mode is also supported but a wait state is needed in the PCI3-Altera-Chip to allow some access delay.

4.1.2 Control registers:

Some control registers are implemented to store the set up bits of the STC. These registers are downloaded the same way as the other memories and a read back is provided as well. The operating mode of the STC board is basically defined by those bits. For example, When downloading data into the STC parameters memories, the DOWNLOAD bit must be set to one in the RUN-CTRL register. The description of the existing registers is given below:

- **TEST REGISTER:**

  This register stores the 15-bit tests signals needed such as:

  - bit 9: TEST-LINK
  - bit 8: INT-TEST-CLK. This bit is set to one when the test is performed with an internal clock. This option is needed when proceeding a test with SMT data from the TEST LUT memory without optic fiber hooked up in the back plane. Without the input optic fiber, the VTM-CLOCK is not available.
  - bit 7-0: SKEW TIME. The 8-bit FRC-DELAY value stored here specifies the timing delay between the FRC and the SMT data. These block of data are not received at the same time when running the STC with detector data. This option allow the testing of the STC in more realistic conditions.

- **DELAY VALUE REGISTER:** This register stores the 8-bit DELAY-VALUE parameter for the STC channels. This parameter is used by the DELAY-COUNTERs to synchronize the input FRC and SMT data. These block of data must be received within the timing window defined by the DELAY-VALUE. This timing window is about $8\mu s$. With 8-bit and a 100MHz system clock, the timing window can only be about $2.56\mu s$ which is not enough. The DELAY-COUNTERs must be scaled with a CLK-DIV4 signal. This clock is a 4 time slower clock based on the system clock. The maximum delay can then be 10.24 μs

- **SCL INIT TIME REGISTER:**

  The 12-bit INIT-TIME word is stored here. When an SCL-INIT command is in progress this INIT-TIME value is decremented. When the timer expires, that indicates that the FRC data input has been idle for the specified time interval.

- **RUN-CTRL REGISTER:**

  This register is mainly used to set up the STC. It is accessed right after power up of the system. This is a 16-bit register used to store the following bits:

  - bits 15-8: DISABLE-CHANNEL:

    These bits are used to disable the STC channels. The 8 DISABLE-CHANNEL signal are sent to each CHANNEL-LOGIC. These bits are all one be default. Once written, they persist until a reset of the RUN-CTRL register is performed. This operation is explained below.
  - bit 7: BC-EN

- bit 6: THREE-STRIP

  This bit is set to one to set the maximum number of strip within a cluster to 3. When this bit is zero, the default value is 5.
- bit 5: ACCEPT-BAD-CHIP.

  The STRIP-READER module does not propagate date from an illegal CHIP-ID to the CENTROID-FINDER module. An illegal CHIP-ID is defined as a chip that has its identification number off the range limits fixed by the DATA-TYPE memory. When this bit is on, the data are allowed to go through the STC.
- bit 4: Z-PCI2-ENABLE.

  This bit is high when the ZVC is not implemented. The output of Z-centroids is inhibited. The Z-CENTROIDS are not even stored.
- bit 3: SCL-READY.

  This bit is set to one to perform an SCL-INIT procedure. The whole STC logic is reset and all the buffers are cleared out. After SCL-DONE has been set, SCL-READY will eventually be reset.
- bit 2: DOWNLOAD.

  The DOWNLOAD bit must be set to one when downloading parameters into the memories. This bit persists as long as the download procedure is going. It is used in several logic to select either local addresses or PCI3 addresses.
- bit 1: TEST.

  This bit is set to one when the STC is in test mode. When high, the test SMT data and test FRC data stored into the respective memories are injected through the STC. Several option can be chosen in the TEST-REGISTER.
- bit 0: RUN.

  The RUN bit is set to one when the STC has been configured and ready to take SMT and FRC data. Several counters, buffers and memories accesses are enable with this bit. If low, no valid data are generated.

When the WR-CSR0 signal is high and the IS-HIGH signal is low, the value carried by W-DATA is stored. Once a bit is set to one, it can not be reset because the 16-bit output of the register is "ORed" with the W-DATA. The bits that where zero can be set to one, but writing zero has no effect. When the IS-HIGH signal is one, the input W-DATA is used to reset the bits stored in the register.

When reading back from the RUN-CTRL register, the lower double-byte word of the PCI3 data contain the value actually stored. The upper double-byte has the version number of the code as bit 31 to 24 and SCL-DONE as bit 19. The version number is an 8-bit word defined in the PCI3-IF code. This value is incremented overtime a new version of the code is released. The SCL-DONE is checked after performing a SCL-INIT command. It is one when the SCL-INIT is completed.

- **MISC-CSR REGISTER:**

  The Miscellaneous control/status register is used to issue command to the STC and check on the status while processing data. Basically, when writing into the register, the command bits stored are pulsed and do not persist. Thus, when reading, the output bits determine the status of the

STC. The bit definitions when either writing or reading are given below:

<u>*command write:*</u>

- bit 0: RESET.

  This signal is used to reset all control and channel logic.
- bit 1: RESET road flags.

  When high, this signal reset the road memory error flags. The definition of these error flags is given in the status read part.
- bit 2: not used.
- bit 3: RESET event mismatch flags.

  This signal reset the register that keep track of the event number mismatch generated while processing data.
- bit 4: MONITOR-START.

  This control signal is set to one to latch all monitored data.
- bit 5: TEST-START.

  When the TEST bit mode is set in the RUN-CTRL register, the test data are used to debug the STC. The TEST-START is set to one to ignite the test procedure. This signal high results in sending the test data through.

<u>*status read:*</u>

- bit 0: not used.
- bit 1: road memory correctable error.

  This bit is high if an error has been detected when reading out data from the ROAD LUT. This bit is generated by the EDAC module in the ROAD-RAM block by applying the Hamming code.
- bit 2: road memory non-correctable error.

  This signal is generated as the previous one. In this case two errors have been detected and this situation is not recoverable.
- bit 3: not used.
- bit 4: MONITOR-DONE.
- bit 5: PCI2-HIT-FULL.

  This signal is coming from the PCI2-Altera-Chip. It is set to one when the buffer that stores the hits is full.
- bit 6: PCI2-Z-FULL.

  This signal is generated from the same PCI2-ALtera-Chip and inform that the Z buffer is full.
- bit 7: L3-FULL

  The Level3 buffer is full.

– bit 15-8: EVT-ERR.

  These bits are generated by the HIT-READ-OUT module. When processing data, the event number extracted in the FRC data is sent to the STC channels. The same event number is sent to the HIT-READ-OUT in order to generate the header and trailer word that precede and trail the hit information. The event number contained in SMT trailer of each STC channel is compared with the one stored locally. There are 8 registers implemented for each STC channel. If a mismatch occurs, it is stores in the corresponding register.

4.1.3 STC testing logic:

This module implements the necessary logic to issue to perform a test of STC. Once in TEST mode, the test can be started by pulling the TEST-START signal in the MISC-CSR register. When the TEST-START bit is set, both TEST-START-FRC and TEST-START-CHNL must be issued. The TEST-START-FRC signal is sent to the PCI1-IF module. The PCI1-IF module sends a control signal to the PCI1-Altera-Chip to start reading out the road test data memory. The input FRC data are replaced by the test data. The TEST-START-CHNL is sent to the SMT-IF modules. As the PCI1-IF, the SMT-IF starts sending out SMT test data to the channels instead of real VTM data. A delay can be specifed between these two signals. The timing interval value is stored in the TEST-REGISTER.

Two counters are implemented to generate these signals. The counter used to issue the TEST-START-CHNL has a delay value fixed. The TEST-START-CHNL is sent about $5\mu s$ after receiving a TEST-START. The TEST-START-FRC is generated by a 9-bit loadable binary counter. When a TEST-START is received, the 8-bit FRC-DELAY from the TEST-REGISTER is loaded into the counter. This delay is actually multiplied by two to get a maximum delay with a 50MHz clock of roughly $10\mu s$. When the delay expires, the carry out signal from the counter is used to generate the TEST-START-FRC signal. At the same time, the FRC-DELAY is reloaded into the counter in order to wait for the next TEST-START signal. When the real data are processed, the FRC block may come either earlier or later than the SMT block. This situation can be reproduced here by setting the FRC-DELAY to the following values:

- FRC-DELAY = 0x80 → skew is zero.
- FRC-DELAY = 0xff, frc lags about 5 μs
- FRC-DELAY = 0x00, frc leads about 5 μs

4.1.4 SCL-INIT logic:

The SCL-INIT command is pulled by the trigger framework when a fatal error has occurred. The trigger system must be reinitialized. An SCL-INIT flag is sent to the FRC card which transmits it to the STC cards. When the SCL-READY bit is set to one in the MISC-CSR register, the PCI3-IF must generate a clear signal to all buffers and a reset to all logic. The STC card must send a SCL-DONE back to the FRC at the end of the SCL-INIT procedure. The SCL-DONE is pulled when the STC did not receive any input data from the FRC and the VTM for a time defined by the 12-bit INIT-TIME parameter. This parameter is stored in the SCL-INIT-TIME register. At a 32MHz clock frequency, the timer has a maximum delay of $163\mu s$. This timer indicates that the initialization of the trigger system inhibited the transmission of any data for a certain amount of time.

A 12-bit loadable counter is used to implement the SCL timer. When the SCL-READY is low, the INIT-TIME value is loaded into the counter. When the SCL-READY bit is high and the INPUT-QUIET is high as well, the counter starts decrementing the INIT-TIME. A SCL-RST pulse is also generated to reset all logic and clear the buffers. This bit is ORed with the RESET command in the MISC-CSR register. The INPUT-QUIET signal has been sent by the STC channels and the PCI1-IF. When the STC channel remains in STATE-S0, no data are received from the VTM, though the INPUT-QUIET signal is pulled by each channel when their STATE-S0 signal is high. If one of the channel has its STATE-S0 low, the INPUT-QUIET signal remains low until this channel does not receive any more data. The PCI1-IF sets the INPUT-QUIET signal to one when no valid data are received from the PCI1-Altera-Chip. The INIT-TIME is loaded each time the INPUT-QUIET signal goes away. The delay expires when the carry out of the counter is seen. This carry out propagates an SCL-OVFL signal that sets the SCL-DONE to one. The SCL-DONE remains one as long as SCL-READY is high. The SCL-DONE can be checked in the MISC-CSR register.

Note that the INPUT-QUIET signal is not generated individually by each channels and the PCI1-IF. This implementation of this signal would need to propagate 9 signals to the PCI3-IF and a 9-bit AND gate would be required. In this design, each module is connected to the same horizontal line with tri-state buffers. This signal is then high (or high impedance "Z") when all module do not get any input data. If one of then output pulls this line to zero, the INPUT-QUIET is low.

4.2 PCI1-IF

The data from the FRC must be interpreted in order to extract the road information. The PCI1-IF uses these roads to address the road look up table and write the lower and upper limits into the ROAD-FIFOs of the STC channels (A sub-module called ROAD-RAM performs this function). Other information comes along with FRC data such as the event number that identifies the event that is being process and the L1-QUAL word that informs the STC whether the particular event is monitored or not. The L1-QUAL is also used to determine what kind of l3 data must be generated by the STC for this particular event. The PCI1-IF also stores the FRC data as they are received for l3 purposes and saves monitoring information from the LRB (link receiver board). Error bits from sub level of the trigger system must be propagated to higher levels. This part will explain how the roads are extracted from the FRC incoming stream and how the information such as event-number, l3 data selection are conveyed to the STC channels. But in the first place, the interface between the Pci1-Altera chip and the STC-Xilinx chip must be broken down as well as the FRC data format.

The PCI-IF and PCI1-Altera chip interface and input format:

The PCI1 Altera chip that sits on the STC daughter board, transmits the data from FRC to the PCI1-IF in conjunction with a PCI1-CLK. The PCI1-CLK is generated locally in the PCI1-IF and it is chosen to be a third of the system clock, so roughly 33 MHz. Eight control signals are also used between these two systems. There are two 4-bit buses: the PCI1-CTRL-I, which is an input port, and the PCI1-CTRL-O, which is an output port. The data are then input to the PCI1-IF as a 32-bit bus called PCI1-DATA. The two 4-bit control signals buses are specified bellow as well as the format of the FRC data.

- PCI1-CTRL-I:
  - Bit 0 : VALID-FLAG. This is bit is one when data on bus is valid. Note that if the valid signal is low the bus value is ignored by the logic.
  - Bit 1 : TRAILER-FLAG. It is high when the corresponding word is the FRC trailer.
  - Bit 2 : MONITORED-DATA. The LRB monitored information is propagated using a serial link.
  - Bit 3 : not used.

- PCI1-CTRL-O:
  - Bit 0 : RESET. When a reset of the STC is pulled, it is propagated to the PCI1-Altera chip.
  - Bit 1 : TEST-START. When the TEST-START signal is set, the smt test data are sent to the channels so the corresponding test roads must be issued from the PCI1-Altera-Chip.
  - Bit 2 : TEST. The test bit is set and persists as long as the STC is used in the test mode.
  - Bit 3 : monitor starts sychro

FRC data format:

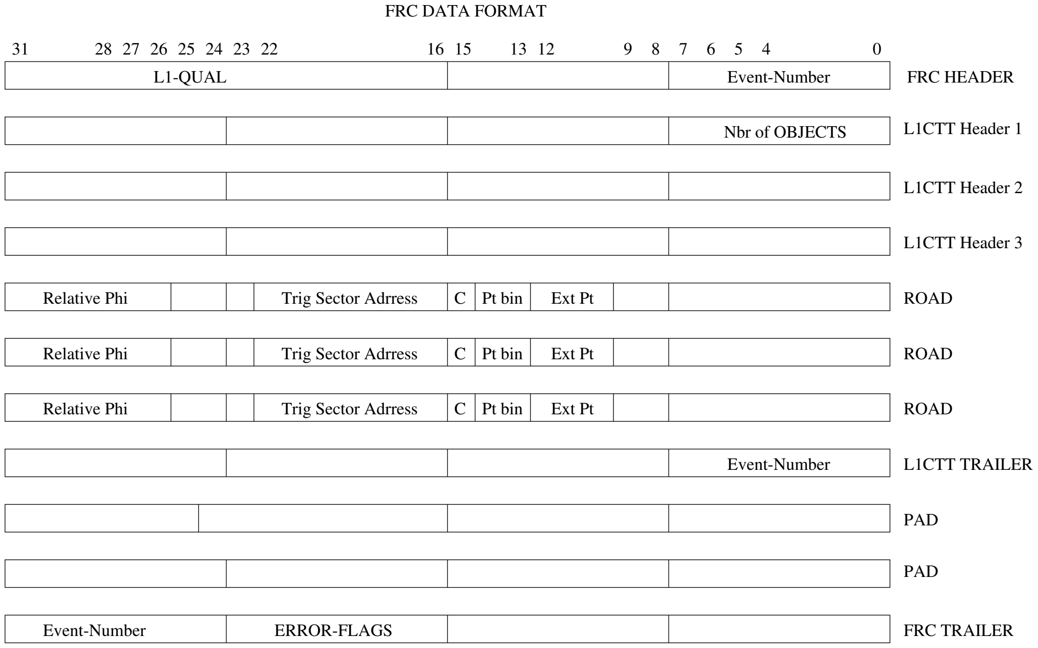

Figure 37: FRC DATA format

This figure only shows the bits that are relevant to the STC. A complete description of the format is given 54.

The FRC data are structured as the following:

- FRC header: This word is added by the FRC board. It contains the event number (L1-BX) and the L1-Qual word.

- 3 L1CTT headers: These headers are produced by subsequent trigger level. The first headers contains the number of objects which corresponds to the number of roads in the event.

- Roads information:
  - Bit 26 to 31: Relative phi. Six bits are allocated for the relative address of 4.5-degree sector where PS cluster found.
  - Bit 16 to 22 : Trig sector addr. This 7-bit word is the 4.5-degree sector where the track was found.

  - Bit 15: C. This bit is the sign of the Pt.
  - Bit 13 to 14: Ptbin. 2 bits giving the Ptbin of track.
  - Bit 10 to 12: Ext Pt. 3 bits of extended Pt information.

- L1CTT trailer: Contains also the event number.

- Padding words (if necessary). The FRC passes along the L1CTT information in 32-bit words as it is received. The number of 32-bit words in L1CTT data is required to be a multiple of four. Pad words that are valid but with contents equal to zero are added in order to meet this condition.

- FRC trailer. The FRC trailer contains the LRB error flags and the event number.

Interpreting FRC data:

The principal function of the PCI1-IF is to extract the roads from the L1CTT information. To perform this, it is inevitable to properly identify the headers, road words and trailers. The logic has been implemented in such a manner that this identification is accomplished as long as the format is respected. If not, an error bit is generated and added into a trailer word passed on to Level-3. The road information as it is defined for the STC point of view, is only a 17-bit word out of the 32-bit word from the FRC. The relevant bits will be specified in the road information part. Furthermore, information in the headers and trailers are relevant for the STC such as the event-number, the L1-QUAL word, the number of roads and the error flags. The use of these words brings us to the other important tasks of this module, adding an event number to the SMT data block and selecting the kind of L3 data that are to be produced. These two peaces of information are transmitted to the design as FRC-EVENT and L3-CONFIG both latched by an FRC-START (or FRC-TRAILER). The way these signals are produced will be explained in detail and conversion of roads into upper and lower limits from the road look up table is also described. The L3config word is a downloaded parameter and the storage of it must be clarified.

Generating an FRC-EVENT, L3-CONFIG and FRC-START:

There are four headers in the FRC incoming stream: one FRC header and 3 L1CTT headers. Only part of these words can be interpreted, the rest of them are just ignored. The FRC header contains the 7-bit event-number and the 16-bit L1-QUAL word. The event-number must be sent to the channels as FRC-EVENT. It is added into the trailer generated by each channel. This is because the SMT data do not have an event number and this one is suppose to be the same as the corresponding L1CTT data. The L1-QUAL is used to select the type of L3-CONFIG word to be assigned to the particular event. The L3-CONFIG word is a 10-bit word that specifies the kind of l3 data that are to be produced by the STC. The FRC-EVENT and the L3-CONFIG are latched by the FRC-START signal. Note that they are also latched by the FRC-TRAILER signal when sent to certain parts of the code.

FRC-START:

The FRC-START is one of the most important signals in this design. It is used to latch important information such as the FRC-EVENT and the L3-CONFIG word. It is also used to mark the beginning of a process of an event. It makes a useful flag for synchronization purposes. By synchronization, we mean here synchronization between the FRC data and the corresponding SMT data. For the 8 STC channels point of view, this signal is high when the FRC data are being received and the FRC-EVENT word is valid. In response of an FRC-START, the STC channels pull the VTM-IDLE signal low and start decrementing their DELAY-COUNTER. Once the SMT data are received, their DELAY-COUNTER is stopped. At the end of the event, the VTM-IDLE goes high again and a new FRC-START can be generated. If no SMT data are sent, the VTM-IDLE remains low until the delay is expired. An SMT trailer is still written to keep synchronization. In case of no FRC data, no FRC-START is issued and therefore no trailer is inserted at the end of SMT data. These measures insure the maintain of synchronization, this point being one of the most crucial issues here.

FRC-EVENT:

The FRC-EVENT is a 7-bit word extracted from the FRC header word. The first valid word that comes out of the PCI1-Altera-Chip must be the FRC header and it is actually identify as such. The first byte is saved into a 7-bit register as EVENT-NO. This EVENT-NO is output as FRC-EVENT to the rest of the STC logic design. After receiving the last L1CTT header and if the VTM-IDLE signal is high an FRC-START is produced to latch the FRC-EVENT. Note that the VTM-IDLE here is low when the channels have received an FRC-START. It remains low as long as the channels are processing SMT data. Time during which no other FRC-START can be issued.

L3-CONFIG word:

Storage of the L3-CONFIG and SEC-OFFSET:

They are two kinds of L3-CONFIG words: UNBIASED-L3 and NORMAL-L3. These words are stored into a small dual port memory RAM16X10 which implements distributed memory (LUTs). The location zero holds the NORMAL-L3 and the location two, the UNBIASED-L3. The 7-bit sector offset is also store into this memory as location three. The two ports are actually used when processing an event. When the DOWNLOAD bit is low the first port outputs the SEC-OFFSET and the second port is outputting the selected L3-CONFIG word. This memory is then addressed by the "OR" signal of the bit 16 and 18 of the FRC header. Note that when the L3 ram is full, the input address of this memory block points to a location where no data have been downloaded. This way the L3-CONFIG word selected is neither normal nor biased but zero. No L3 data are then produced.

L3-CONFIG:

The FRC header also contains a 16-bit L1-QUAL word. This one is used to select the type of L3-CONIFG word. If either bit 0 or bit 2 of the L1-QUAL word is high, the unbiased configuration word is used to generate the L3-CONFIG. The 10 bits of the L3-CONFIG word are the following:

- Bit 0: BAD. BAD-CHANNEL look up table.
- Bit 1: L3-90. 90 degrees centroids.
- Bit 2: L3-STEREO. Stereo centroids.
- Bit 3: L3-AX. Axial centroids.
- Bit 4: L3-CLSTR. Clusters.
- Bit 5: L3-CORR. Corrected data.
- Bit 6: L3-RAW. Raw data.
- Bit 7: L3-HITS. Hits.

- Bit 8: L3-ROTATE. Select only one channel at a time. A different channel is selected every event.
- Bit 9: L3-ROADS. Road information from the FRC stored into the PCI1-IF.

The actual L3-CONFIG word sent to the channels is a 13 bit word, the bit 0 to 9 being equivalent as the one downloaded. The complementary information is the following:

- Bit 10-11: L3-CHANNEL. When the L3-ROTATE bit is one, the selected channel is indicated by the 3-bit L3-CHANNEL word.
- Bit 12: MON-EVENT. This bit is extracted from the L1-QUAL word. It corresponds to the L2CollectStatus bit. it is used to determine whether or not monitoring should be performed while processing this event. If yes, the MON-EVENT bit is high and the MONITORs in the STC channels are latched.

These bits are devoted to different peaces of logic. When high, the corresponding data must be produced and stored. They are all latched by the FRC-START signal. The ROTATE bit is used to reduce the amount of L3 data produced. When high the number of STC channel that must output L3 data is restrained to one. This do not affect the L3 hits information. It is also important to add that only one channel gets to output level-3 cluster, raw and corrected data. A 3-bit counter is implemented to generate the L3-CHANNEL word used to decide which channel is affected. This counter is incremented at the end of the event. So it takes 8 events to cycle through all channels. In this module, the end of event is reached when the eight channels have written their SMT trailer. When a channel is disable, meaning that the DISABLE-CHANNEL bit is set for this particular channel, the L3-CHANNEL counter is incremented as long as a channel enabled is selected.

Note that the logic that generates L3 raw, L3 corrected and L3 cluster data is enable looking at the L3-CHANNEL word. This word must not change until the SMT trailer added by the STRIP-READER gets to the CENTROID-FINDER fifos. It must persist until the event is thoroughly finished otherwise the data would only be partially collected. After writing a SMT trailer, it takes 30 clock cycles to get this trailer to the CENTROID-FINDER fifos. SO when the VTM-IDLE goes high, the eight trailers have been written and a delay of 30 clocks is added before incrementing the L3-CHANNEL counter.

The number of roads, trailer detection and generation of FRC-TRAILER:

Trailer detection:

In order to check that the FRC data format has been interpreted correctly, the number of road in the first L1CTT header is used. The 7-bit number of object is inverted and stored as N-TOTAL into a register. The number of pad words can be determined looking at the N-TOTAL word.

- If the two LSBs are high, the number of roads is a multiple of four and no pad words need to be inserted. Remember that there are 3 L1CTT headers and 1 L1CTT trailers so the number of roads must be a multiple of four in order to have the total number of L1CTT words as a multiple of four.
- Any other combination of these two bits implies the presence of padding words.

The counter is incremented as soon as the last L1CTT header is received. When the count reaches 0xFF, the CARRY-OUT bit is set to one and is used to detect the last road. The following word is identify as the L1CTT trailer. If no padding words were expected, the next valid word is identify as the FRC-TRAILER and the TRAILER signal is set to one. Otherwise, the counter continues to increment until reaching a value for the two LSBs equal to the the two LSBs of the N-TOTAL word. The trailer bit is then set to one to expect the next word to be the FRC-TRAILER. To make sure we have done things right, the TRAILER bit is compared with the TRAILER-FLAG that comes along with the FRC-TRAILER. If these bits are identical, the FRC-DATA has been interpreted correctly or no error in the format has been detected. Otherwise a MT bit is set to one and added into the L3-TRAILER word.

ERROR FLAGS:

When the TRAILER-BIT is set to one, the bits 16 to 23 of the FRC trailer are registered as LRB-ERR. These error flags must be sent to the STC channels to be added to most of the trailers. This word is latched by a FRC-TRAILER flag which is issued as the FRC trailer is received. The FRC-TRAILER is used to latch the FRC-EVENT and the L3-CONFIG in some of the read out logic.

ROAD information:

Data words after the headers contain road information. Each road is a 17-bit word that is saved into a road fifo. These roads will be used to address the external road look up table. Within the 32-bit road word received only 19 are used to generate the 17-bit road address:

- Relative phi (6-bit).
- Trig Sector Address (7-bit).
- C (1-bit).
- Ptbin (2-bit).
- ExtPt (3-bit).

A 7-bit sector offset is subtracted to the 7-bit track sector address to reduce it to 5 bits. Thus these bits are reorganized to get the correct road address format 38. An END-OF-ROAD bit is added as MSB in order to mark the end of the road sample for the event. the content of the road address word must be discarded.

17	16	15 14	13 11	10 5	4 0
eor	C	Ptbin	Ext Pt	Relative Phi	Trk Sector address

Figure 38: Road Format

These roads are stored into a 1024X20 fifo. It is made of 4 RAMB4-S4-S4 and two 10-bit binary counters. Roads are written into the fifo as they are received from the PCI1-Altera-Chip. Note that the headers and the L1CTT trailer as well as the FRC trailer are not included into the road information of an event. Therefore, a trailer word with the END-OF-ROAD bit set to one is written into the fifo. This trailer is necessary to mark the end of event. It is done so to simplify and do not cause any troubles because the content is not relevant. The END-OF-ROAD is set to one when either the L1CTT-TR or TRAILER-FLAG or ROAD-LIMIT or PADDING is high. The word that comes after the last word is written into the fifo with the END-OF-ROAD bit set. This word can either be a padding word or the l1cttt trailer or the frc trailer if the l1ctt trailer has not been detected correctly. The condition

ROAD-LIMIT is because if the count of roads exceeds 46, the next word written into the fifo must be the trailer.

GETTING the Upper and lower limits:

As soon as roads are present into the fifo, the ROAD-RQST bit is pulled. The number of words stored is always known using a 10-bit subtractor, which outputs a ROAD-CNT word. At this moment, the first road is already at the output of the fifo. The ROAD-RAM module sends the road address out to the road look up table and generates a GET-ROAD signal. The GET-ROAD signal will produce a read signal that will output the next available road. It actually takes several clocks to get the 8 upper and 8 lower limits for the 8 sets of STC channels. The next road is read out before getting the limits of the previous roads to get the next limits right after the others. This point is discussed in more detail and the delays are specifies in the part that presents the ROAD-RAM module. The 8 upper and lower limits are sent to the STC channels as 22-bit ROAD-DATA word. The 8-bit road address, ROAD-ADDR, is also sent as well as a write signal WE-ROAD. There are 8 WE-ROAD signals for all of the channels. When the WE-ROAD is high for a particular channel, the ROAD-DATA word is written into its ROAD-FIFO at the location indicated by ROAD-ADDR. The ROAD-ADDR counter is incremented after writing the lower and upper limits of the last channel. This counter is a 6-bit binary counter that is used to generate the 6 LSBs of the ROAD-ADDR. An additional 2-bit binary counter updates the two remaining bits. The last word to be output is the trailer that has its bit 17 set. This bit is immediately checked and results in resetting the ROAD-RQST signal which tells the ROAD-RAM that no more roads are to be sent to the SDRAM. No more GET-ROAD are going to be issued so the logic must move the pointer to the next empty location in the fifo. The ROAD-ADDR counter is also disabled after a certain delay because the limits from the previous road address are still coming from the road table. This delay allows the logic to set a LAST-ROAD signal to one that needs to be sent to the STC channels with the ROAD-DATA. The LAST-ROAD indicates that the corresponding limits belong to the last road address of the event. The WE-ROAD that is output for the channel number seven is used to update the 2-bit binary counter that generates the 2 MSBs of the ROAD-ADDR. That is to start writing the limits for the next event into the next 64-word block of the STC channels ROAD-FIFOs (see the Channel logic for more details). One important point is to know what the logic is suppose to do when no roads are present in the FRC incoming data. The channel HIT-FILTERs required at least one road to ignite the filtering process. A fake road must be sent and the logic must prevent for any hits to be found. The trailer word is still written into the fifo and it is the only word present. The "and " of the GET-ROAD signal and the ENF-OF-ROAD signal will provoke the FAKE-ROAD signal to be set. FAKE-ROAD is then delayed to generate a other signal called FALSE-ROAD that is propagated to the ROAD-RAM. The limits output by the table are not interpretable and to make sure that no hits are going to be produced, the ROAD-RAM module sets the bit 10 of the lower limit to 1 and the bit 10 of the upper limit to zero. The FALSE-ROAD signal forces the ROAD-RAM to output lower limits above the upper limits so that no hits can be found. These fake limits are sent with the LAST-ROAD signal high and stored into the channel ROAD-FIFOs as real roads. This way, the HIT-FILTERs do not have to include the no-road-in-the-event condition and therefore simplify the logic. It is important to notice that the PCI1-IF resets the ROAD-RQST when the roads for two events have already been processed and all the hits that correspond to these two events haven't been sorted out. A simple 2-bit binary counter is implemented for this purpose. It is incremented after reading the trailer word and it is decremented when receiving a HIT-DONE from the HIT-READ-OUT logic that testified the end of the filtering process of an event.

Logic for Level 3:

The PCI1-IF is also assigned to produce l3 FRC data. Basically, the L3 data stored are the complete block of FRC data as transmitted. But a header and a trailer must be inserted and the information contained in those words needs extra logic to be coded. The data for l3 are stored into a 256X32 fifo implemented with 2 RAMB4-S16-S16 memory blocks. The input PCI1-DATA is stored with the same format with a WR-L3-ROAD signal that is high as long as the VALID-FLAG remains. The padding words are of course not stored to save memory space. Thus the WR-L3-ROAD signal is reset when PADDING is high. Two 8-bit binary counters are needed to generate the write address and the read address for the l3 fifo. One particularity of the write counter is that it is separated into a 6-bit binary counter and a 2-bit binary counter. The 6-bit counter is incremented as writing l3 data and the 2-bit counter updates the 2 MSBs of the write address when all the l3 data for this event have been saved. The l3 fifo is then divided into four 64-word block that are able to store the l3 data for one event each. The maximum number of word produced for an event, in normal conditions, is 56. This number includes the 4 headers and 2 trailers, the header and trailer added by the STC as well as the 46 roads (maximum number). An allocation of 64-word memory space is adequate for an event. This point will be explained when discussing the L3RAM module. An other feature of this l3 fifo is the reading mode. The read procedure has been designed in such a manner that a single read pulse, called GET-L3-ROAD from the L3RAM module, results in outputting the whole 64-word block of l3 data. This procedure is also explained in the L3-RAM part. The PCI1-IF informs the L3-RAM module that a 64-word block is present by setting the L3-ROAD-AVL. This signal can be coded as the output of a 2-bit non-equal comparator that has for input the 2 MSBs of the write and the read addresses.

A header must be inserted at the beginning the l3 data block. The number of words that are contained in this l3 block has to be displayed in this particular word. But the overall number of words is known after writing all of them into the l3fifo. For this reason, the write pointer is moved to the second location after resetting the entire STC. This free space will be used to insert the header once the end of event is reached. After writing the trailer word, the write counter is reset to move the pointer to this location and the count of the number of words is saved into a register as L3-WRD-CNT and is to be added to the header word. The header is then written and the write pointer is moved to the 2nd available location of the next 64-word block. In case of a L3-FULL signal high from the L3-RAM module, no more L3 data are produced and then no more data need to be stored.

5 ROAD-RAM

This module is used to establish the connection between the CONTROL-LOGIC and the Road LUT. Because of its size, this table is held in an external SDRAM. The Xilinx chip is not able to provide such memory space for both the road LUT and the rest of the logic. The ROAD-RAM implements an SDRAM controller and an error detection and correction sub-module called EDAC. The SDRAM controller issues the control signals needed to set up the SDRAM, download the table and access the memory while operating the STC. The EDAC receives the data from the SDRAM and checks for errors. The roads are sent to the PCI1-IF along with two error bits. The SDRAM features are presented as well as a complete description of the controller logic. The download procedure from PCI3 and the read operation while processing data are explained.

5.1 SDRAM features:

Some features of the SDRAM implemented for the STC are given here. Additional technical details can be found on the data sheet 54. The Synchronous DRAM used here provides a 64 Mb memory space with an access speed frequency up to 200 MHz. The frequency rate depends on the speed grade chosen.

A -7 speed grade correspond to a maximum frequency of 143MHz. This characteristic is suitable for a 100 MHz system clock. This memory is organized in 4 identical banks providing each 512KX32. Each bank is organized in 2048 lines and 256 columns. Every location of the banks has been designed to store 32-bit words. Thus, the raw address is an 11-bit bus (A0-A10) and the column address is 8-bit wide (A0-A7). The bank addressing is determined with 2-bit B0 an B1.

Some other features and their advantages are summarized here:

- Fully synchronous: all signals are registered on positive edge of the clock.
- Internal pipelined operations: column address can be changed every clock cycle.

  The conversion of the roads into range limits can then be performed at a 100MHz frequency rate: One interval is received every clock cycle.
- Programmable burst lengths: 1,2,4,8, or fully page.

  Each road is associated with 8 range limits. Each pair of limit is sent to a CHANNEL-LOGIC. This feature makes then possible the read out of 8 limits in a burst mode.
- Auto precharge.

  Precharging one bank while accessing one of the other 3 banks will hide the precharge cycle and provide high-speed random operation. While one banks is being accessed, an other bank can be precharched. The output from the SDRAM is continuous. This feature provides a full random access : Different raws and bank may be addressed while processing.
- Auto refresh mode.

  This Dynamic RAM implements CMOS transistors that loses their charge with the time. They need to be refreshed. This operation is executed by the internal refresh controller.
- CAS latency supported.

  The CAS latency is the delay, in clock cycle, between the registration of a READ command and the availability of the first output data. It can be set to 1,2, or 3.

This component offers useful options for the present design. The same SDRAM is used to store L3 data.

SDRAM controller.

The SDRAM controller consists of a state machine. Seven states are implemented in total to execute operations such as READ, WRITE, REFRESH, PRECHARGE, ACTIVE, NOP and LOAD MODE REGISTER. The architecture of the state machine is shown figure 39. The function of each state is described in the following.

Each command correspond to a different combination of the 3 signals RASN, CASN and WEN. The meaning of these signals are:

- RASN : Row Address Strobe.
- CASN : Column Address Strobe.
- WEN : Write Enable.

The following truth table provides a reference of the available commands:

COMMAND	RASN	CASN	WEN
NOP	1	1	1
ACTIVE	0	1	1
READ	1	0	1
WRITE	1	0	0
PRECHARGE	0	1	0
AUTO REFRESH	0	0	1
LOAD MODE REGISTER	0	0	0

The RASN, CASN and WEN signals are control signals sent to the SDRAM control logic. This module manages to execution of the command specified by those bits. The NOP (no operation) state is the idle state. After the completion of any command, the logic goes back to NOP.

The LOCKED signals forces the controller to an idle state. No access to the SDRAM are provided as long as this signal is low. When high, the memory can be initialized. All banks are being precharged and placed into an idle state. Once in the idle state, two REFRESH commands must be performed to set up the SDRAM in a mode register programming status. The Mode register in the SDRAM control logic will be power up in an unknown state, it should be loaded prior to applying any operational command.

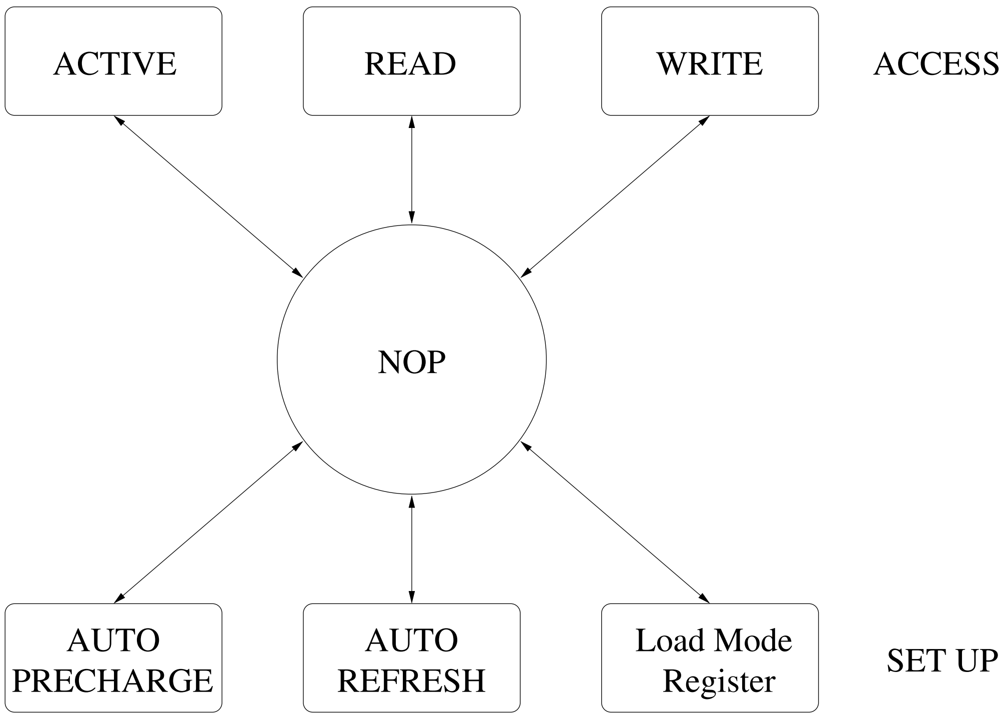

Figure 39: Road-RAM state machine

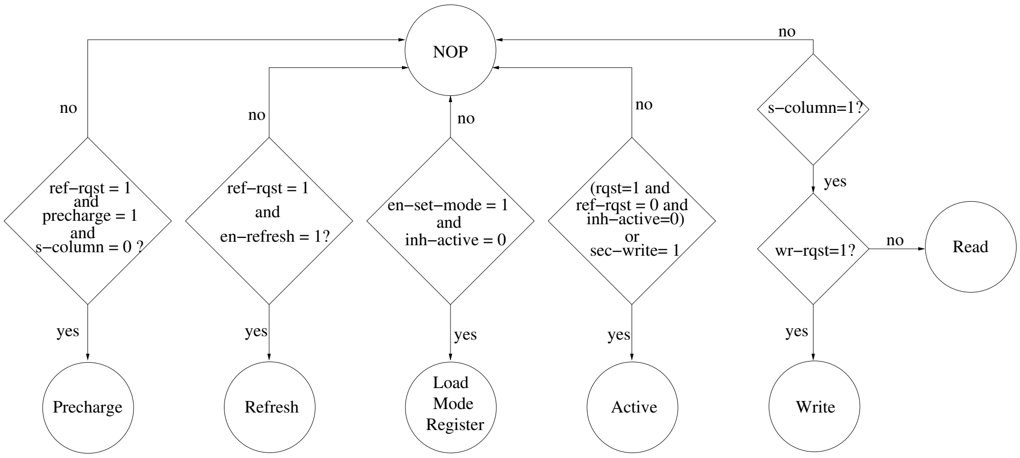

Figure 40: NOP state

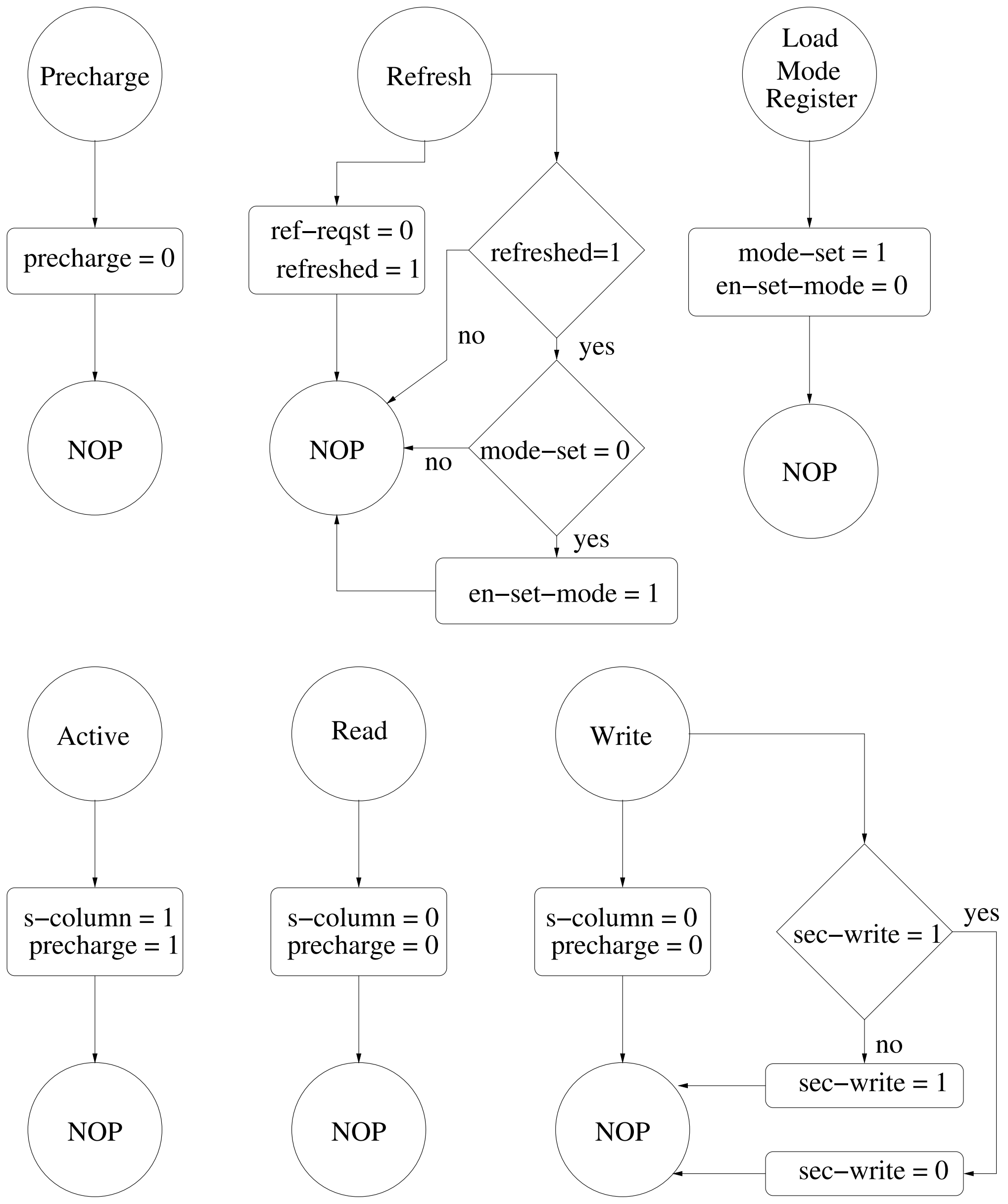

Figure 41: Road-Ram states

NOP (No operation):

The NOP state is used as a transition state, before of after performing commands, the system must come back to it. A scheme of this state is shown figure 40. The other states are shown figure 41. From this state, the logic can go to different state according to the control signal generated:

- goto precharge:

  After setting the Locked bit to 1, the system must be initialized. The PRECHARGE state is intended as soon as the REF-RQST signal is received. When a REF signal is sent from the REFESH module, the REF-RQST is set to one and persists. The logic proceed to the PRECHARGE state.

- goto auto refresh:

  When the EN-REFRESH signal is high and the REF-RQST is set to one, an AUTO REFRESH operation may be performed.

- goto set mode:

  This option is intended when the initialization is in progress. After two REF signal high, the MODE REGISTER in the SDRAM control logic is set up. The logic proceeds to LOAD MODE REGISTER.

- goto active:

  When the RQST signal is high, a READ or a WRITE operation has to be performed. If REF-RQST and INH-ACTIVE signals are low, the machine jumps to the ACTIVE state. In case a SEC-WRITE signal is high, the logic goes to ACTIVE as well.

  A 4-bit counter, INH-CNTR is implemented to inhibit the ACTIVE command for certain number of clock. cannot send two READ command: need set up time

- goto write or read:

  If the S-COLUMN signal is high, a READ or WRITE operation has to be executed. In case of a WR-RQST signal high, the WEN goes low and the logic goes to the WRITE state. Otherwise the WEN remains high and the READ state is attended.

<u>READ</u>

The column address is sent to the SDRAM as bits A0 to A7. Eight consecutive locations will be read out. The precharge bit that was set to one after the ACTIVE operation, is used to set the bit A10 of the input address. This bit set will force an automatic precharge operation at the end of the burst. This way an other raw in same or a different bank can be addressed. A read operation is shown on figure 42

The S-COLUMN and PRECHARGE signals are reset. The logic goes back to NOP.

<u>WRITE</u>

When the write state is reached, the column address is sent out in conjunction with a WRITE command. The WRITE operation is autoprecharged as well. The S-COLUMN and the PRECHARGE signals are reset. The SEC-WRITE signal is set to one to order a second WRITE operation. This is to write the same information to a mirror bank. When reaching the NOP state, the logic will proceed to the ACTIVE and the WRITE state to saved the same input data. The reason for this is explained in the download procedure part.

A write operation is sown on figure 43.

<u>ACTIVE</u>

The ACTIVE command is used to activate a row in a particular bank for a subsequent access. The bits BA0 and BA1 select the bank and the address provided on inputs A0 to A10 selects the row. This row remains active until a PRECHARGE command is issued to that bank. This command must be performed before opening a different raw in the same bank. The PRECHARGE can also be executed automatically.

The S-COLUMN bit is set to one to select to column address and force the logic to go to the READ or WRITE operation. the precharge signal is pulled in order to set up for a autoprecharge of the system.

<u>PRECHARGE:</u>

The precharge operation is sent to the SDRAM control logic in order to deactivate all the banks. The precharge signal is reset and the logic goes back to NOP. By resetting the precharge signal, the EN-REFRESH signal is enabled to allow a refresh operation.

AUTO REFRESH

During a refresh operation, the addressing is generated by an internal refresh controller. This logic disable the input address bits while the refresh is being performed. The REF-RQST signal is reset and the REFRESHED bit is set. The EN-SET-MODE signal stays zero until an other REF-REQST is received. Once two refresh are done, the MODE-SET remains one and the EN-SET-MODE is not pulled anymore.

LOAD MODE REGISTER

This state is used to set up the MODE REGISTER. The Mode Register is used to define the specific mode of operation of the SDRAM. This definition includes the selection of a burst length, a burst type a CAS latency, an operating mode and a write burst mode. The stored information are retained until it is programmed again or the device loses power. The input address bits A0 to A10 of the SDRAM are switched to Mode register bits M0 to M10 while this operation is being performed. These bits specify the following options (only the used ones are shown):

- M0-M3: Burst Length.

  The burst length is fixed to 8 here by setting bit 0 and 1. Eight sets of ranges limits are being read out one after the other.

- M4-M6: CAS Latency.

  The bit 4 and 5 aer set to one in order to program a latency of 3 clocks.

- M9: Write burst mode.

  This bit is set to one so that the burst length programmed via M0-M2 applies only to the READ bursts. The write accesses are single location accesses.

The MODE-SET signal is set to one to terminate the initialization procedure. The EN-MODE-SET is reset as well and the logic goes back to NOP.

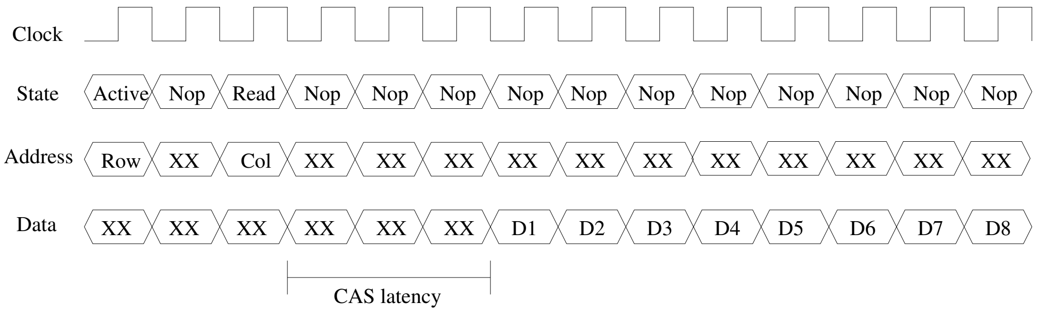

Figure 42: Read operation

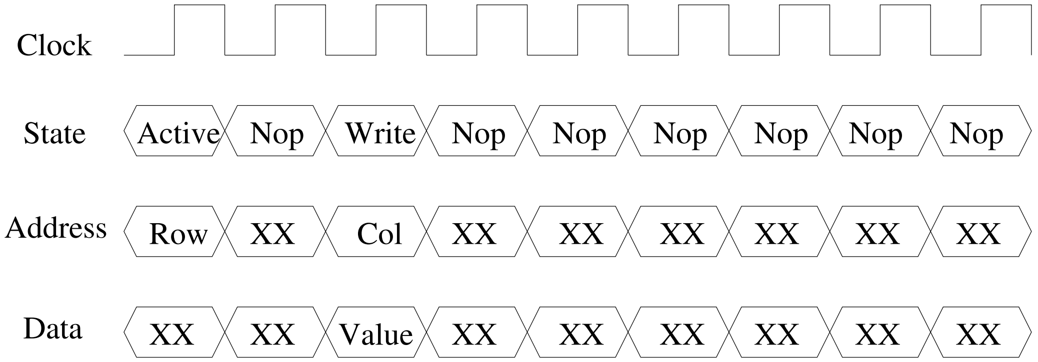

Figure 43: Write operation

ROAD LUT DOWNLOAD PROCEDURE:

The road LUT is a 128KX22 table. To each road, correspond 8 range limits so in total, the table needs a 1024KX22 memory space. This SDRAM provides a 2048KX32 memory space which more than twice the size required. Although the table fits in the SDRAM, a copy of the table must be loaded as well. Two copies of the table are stored into this SDRAM to be able to read out one range limits every clock cycle. The first table can be written on the first two banks and the other one on the remaining two. One bank is being accessed while the other one is being set up. Two banks are practically independent and they can be accessed after a minimum time (to finish the ongoing activity) This method speeds up the process of converting roads.

The memory organization is the following: Each bank holds 32 pages. The entire table needs 64 pages so the extra 64 are allocated for the copy. The page correspond to a memory size of 64 rows and 256 columns. Every location is able to store a 32-bit word. The table is downloaded via PCI3 and a read back is also provided. The write and read procedure are explained in the following sections.

Write procedure:

The SDRAM mode register is set up in a single WRITE mode. One 22-bit road (refer to range limits here) is written at a time. The logic has been designed in such way that a second write is generated automatically to make a copy of the same input data in the mirror memory. The road information is stored in the same exact location in the copy file.

Before writing the roads information into the SDRAM, the page address PA must be registered in the ROAD-RAM module. The 6-bit information received via W-DATA is saved into a page address register when the WR-ROAD-PA signal is high. A read back from this register is implemented via the R-DATA bus when a RD-ROAD-PA is set to one. Once the page address is specified, the roads may be loaded. The input data from PCI3 is transmitted from the PCI3-IF module as a 16-bit W-DATA bus in conjunction with a WR-ROAD-DATA signal. The address ADDR is a 16-bit bus that is being received one STROBE clock earlier than the corresponding data. The bit 13 to 0 of ADDR are loaded into a 14-bit counter. The output of this counter correspond to the MAR (memory address registered) address. This counter is not incremented during a write procedure, it is used as a register. The counter is disable because the write burst mode is not available. The MAR is then sent to a multipexer. The mutiplexer logic is used to select the row or column address according tp the S-COLUMN bit. This selection include another criteria which is the S-ROAD bit. When high, the FRC data are used to address the SDRAM and when low, the memory is being loaded. Thus the output of the multiplexer is a 12-bit MA address that is sent to the SDRAM.

The MA address is also used to set up the MODE REGISTER when initializing the SDRAM. When the MODE-SET bit is low, the SDRAM control logic is not programmed yet. The MAR address register is reset to force this value to zero. The output MA of the multiplexer has some of its bits set (See Load Mode Register state). The second REF flag received forces the SDRAM controller to the LOAD MODE REGISTER state. When this command is issued, the content of MA10-MA0 is recorded in the Mode Register.

The following table gives the MA address values according to the command:

PUT IT IN ANNEX

As the raw address is selected by the mutiplexer, the WR-ROAD-DATA signal brings high the WR-RQST signal which results in setting the RQST bit to one. This RQST bit informs the state machine that data are to be written and the row address must be set up. The registered MA address coincident with the ACTIVE command are used to select the bank and the row to be accessed. The bits MA12-MA6 are used to select the page address and bits MA5 to MA0 select the row within the page. The two MSB of the MA are in fact selecting the BANK.

The 22-bit road data that is stored into the memory are received from the PCI3-IF as two consecutive double-byte words. The first double-byte must be registered as well as the second one. The output of these registers are merged together to form the 32-bit word to be sent. While the ACTIVE command is being performed, the 16-bit W-DATA is saved into a first register when both WR-ROAD-DATA and IS-HIGH signals are high. The second 16-bit word is written into another register when IS-HIGH is low and WR-ROAD-DATA is high. The value of these registers are not updated until a new WR-ROAD-DATA is issued. The data need to be saved because they are stored into the copy as well. The same data value must be available when the second WRITE command is issued. These two registers are declared into a sub-module called ROAD-PADS. This module is used to implement the output pins for the SDRAM. The 32-bit MD-I declared is a bi-directional bus. The MA address uses a separate 13-bit port and each RASN, CASN and WEN is connected to one pin.

One clock after the ACTIVE command, the S-COLUMN bit is set to one in order to select the column address within the page. The bits MA7-MA0 are used to specify the 8-bit column address. The bit MA9 is pulled as well, in order to set the auto precharge mode. The eight consecutives column addresses must store the 8 range limits for the 8 STC channel.

After specifying the location, the WRITE command is issued to the SDRAM control logic. The coincident data word is thus written. A SEC-WRITE signal is set to one to get a copy of the same data. The second write is generated internally to avoid the user to issue too consecutive write command with the same address and data. The second WRITE signal cannot be issued right after the first one because the SDRAM requires some set up time. This time is needed to finish the ongoing process. The ACTIVE operation is then inhibited for .. clock cycles. When the ACTIVE command is finally issued, the bit MA12 is set to one in order to select the pair of banks storing the copy of the road LUT. The WR-RQST signal is reset when the second WRITE command is issued. The RQST goes low until a new WR-ROAD-DATA signal. The state machine remains in the NOP idle state until the next write procedure.

The write procedure of a single word needs several clock cycles. This delay forces the PCI3-Altera-Chip into a wait state. The PCI3-Altera-Chip is put in a wait state for about 6×30ns. During this time no WR-ROAD-DATA are issued to the ROAD-RAM module.

burst write possible

Read procedure:

A read back from this memory is possible via PCI3. This operation can be done in either single read mode or burst read mode. The single read mode is used to read the data back from a unique location and the burst mode reads out data from several consecutive location starting from a specified address. These processes require additional logic because the SDRAM has been set up in a read burst mode. Every READ access of a single location results in getting the data stored in the following 8 columns.

When a read procedure is required by the PCI3-Altera-Chip, the PCI3-IF issues a START-READ-RAM to the ROAD-RAM module. The PCI3-CYCLE signal is set to one and remains one during the read access of the memory. The input ADDR address coincident with the START-READ-RAM is loaded into a counter. The output MAR of this counter is sent to the same multiplexer invoked previously while discussing the write procedure. The S-ROAD signal is low because the download process is on. The START-READ-RAM sets the RD-RQST and then the RQST signals to one. The state machine proceeds to the ACTIVE state. The row address is sent to the SDRAM control logic. Then S-COLUMN is set to one to select the column address. The READ command is issued in conjunction with the MA address. After 3 clock cycle of latency, the data are received. Eight consecutive words are received and stored into a small 16X32 FIFO called R-FIFO. The output from the SDRAM are sent at a 100MHz frequency rate and the ROAD-RAM module is only able to output data to the PCI3-IF at 50MHz. For this reason the data read out must be stored temporarily. Then the scenario is different when accessing in a single read mode or in a burst mode:

- Single Read Mode:

  When the READ command is executed, a EC-BURST-CNTR signal is sent to a 4-bit BURST-CNTR counter. This counter is used to generate a READ-BURST signal to the R-FIFO. When the READ-BURST signal is high, the data received from the SDRAM are stored into the R-FIFO. In the single read mode, only the first data received must be output to the PCI3-IF. This data is already present at the output of the R-FIFO as a 32-bit word R-FIFO-O. A ROAD-DATA-AVL (road data available) signal is sent to the PCI3-IF to inform that data are ready to be output. A RD-ROAD-DATA is then received in response. The RD-ROAD-DATA signal is used to save the R-FIFO-O word into register. The upper 16 bits are sent through the R-DATA bus first and the lower 16 bits follow. When the PCI3-CYCLE bit goes low, the read operation is done and the RD-RQST is reset as well as the RQST signal . The R-FIFO is also flushed out. The next single read begins with START-READ-RAM.

- Burst Read Mode:

  When the burst mode is selected, the PCI3-CYCLE remains high until the whole memory space has been read out. The RD-RQST remains high so that READ operations are constantly required. The MAR address counter is loaded with the burst starting address when START-READ-RAM is received. This counter is incremented as data are received from the SDRAM. The MAR address has to be incremented of as many column address as read out. This counter is then used to indicate the location of the next burst operation. The eight words received every time are stored into the R-FIFO. When almost full, the RD-RQST goes low to stop the READ operations. This signal goes back high as soon as data can be stored into the R-FIFO again. The ROAD-DATA-AVL signal is issued to the PCI3-IF to get a R-DATA bus access. The data are sent out every STROBE cycle i.e. at a 50MHz rate. The R-FIFO is not flushed out as long as PCI3-CYCLE remains high.

The READ operation are sent one after the other but a minimum number of clock cycles delay is needed. This is the same delay as the write procedure.

Addressing with road information:

Once the road LUT has been configured, the STC may be switched to the run mode. The RUN bit in the RUN-CTRL register is set to one. The PCI1-IF extracts the road information out of the FRC data. The 17-bit road addresses are stored into a fifo. As soon as roads are available, the ROAD-RQST input is high. This signal will pull the RQST signal to ignite the SDRAM access procedure. The first road is already present at the input port ROAD-I of the ROAD-RAM module. The ROAD-RQST signal sets the S-ROAD signal as well. The S-ROAD signal is issued to the MA multiplexer to select the row addressing bits. While the state machine moves to the ACTIVE sequence, the MA address is set up with the ROAD-I bits (look at table). The bit MA12 is set to a 0 value and the bit MA11 correspond to the bit ROAD-I16 in order to select the correct bank. Bits MA10-MA0 are generated from the bits ROAD-I15-ROAD-I5 The ACTIVE command accesses the page and the corresponding row in the SDRAM. The S-COLUMN and the PRECHARGE signals are set to one as the state machine gets back to the idle state. The column address is selected: MA7-MA3 are set as the value of the ROAD-I4-ROAD-I0 bits and MA2-MA0 are set to zero. The 3 LSBs don't need to be specified because to each road address correspond 8 range-limits. The column address coincident with a READ command are sent to the SDRAM. The MA10 bit is also set to enable the auto precharge option. Three clocks after the READ command, 8 consecutive columns are read out in a burst mode. These data are sent to the EDAC sub-module to perform the error detection and correction. Then, after two additional clock cycles delay , the ROAD-DATA are sent out to the PCI1-IF in conjunction with a WE-ROAD signal. The WE-ROAD is in fact an 8-bit bus where each bit correspond to a write enable road to each of the STC channel. The bits are consecutively set every clock cycle as data are sent out figure(X). The READ command also sets the MSB of the MA address to one in order to select the copy of the road LUT when the next read access is performed. The S-COLUMN is reset as well. Two other error bits are sent out with the ROAD-DATA.

As the ACTIVE command is performed, the GET-ROAD signal is issued to the PCI1-IF. The GET-ROAD signal moves the read pointer to the location of the next available 17-bit road address. If only one road was present in the fifo, the ROAD-RQST signal is reset after sending the GET-ROAD signal. The RQST bit goes low as well and the state machine remains in the idle state.

If the PCI1-IF road fifo contains several words, the ROAD-RQST remains high after sending the GET-ROAD signal. A new input is received and the row address is set up. The state machine receives a RQST bit but no ACTIVE command can be performed if the INH-ACTIVE signal is high. This signal is generated by a 3-bit counter to inhibit any read operation for .. clocks. As data are being received from the previous read operation, the two other banks of the SDRAM are activated. The column address is sent and data are received right after the last word read out from the previous read operation. The read access is thus performed continuously and data flow in every clock cycle.

Error detection and correction:

This function is preformed by the EDAC sub-module. The input 32-bit MD-I bus is checked for errors. The bits go through equations that are used to determine symptoms. Those symptoms are used to set either the MEM-ERR is case of an error or the MEM-DERR if a double error had occurred in reading the road ram. These bits are registered so that they remain high. They are propagated to the PCI3-IF module which provides an external access to those status bits. A RST-ERR may be pulled to reset those registers. This is done setting a reset command in the MISC register (check). data are corrected, double errors cannot.

Hamming code 54

6 Level 2 Logic

6.1 HIT-READ-OUT:

The HIT-READ-OUT module is a sub-module of the CONTROL-LOGIC. It collects hits from all of the STC channels, formats them and stores them for Level 2 and Level3 purposes. The same hit information is actually stored for the PCI2 and for the Level3 logic but in two separated buffer. Information such as event number and L3 configuration words are needed from PCI1-IF. This module provides event number mismatch detection as well.

6.1.1 Information from PCI1-IF

The PCI1-IF sends information such FRC-EVENT, L3-HITS and LRB-ERR. The FRC-EVENT is the 8-bit event number that is inserted in the header and trailer word of the hit information to PCI2-OUTPUT. The LRB-ERR is an 8-bit word that contains the error flags from the FRC trailer word. This word must be inserted in the trailer word to be propagated to the next trigger level. L3-HITS is one of the bit of the L3 configuration word. If high, the hits produced by the treatment of the event have to be stored in a separate buffer. These information are latched by the PCI1-IF with a FRC-TRAILER. They are written into a 17X16 event number FIFO with an FRC-TRAILER signal high. This FIFO is then able to store information for 16 event maximum. Once these words are saved into the FIFO, the EMPTY-FIFO goes low and the system is activated and ready to receive hits.

6.1.2 HIT storage:

Once the EMPTY-FIFO flag goes low, a START pulse is issued and HIT-IDLE goes low the next clock. The HIT-IDLE flag is used to select either the header or the trailer word. With a HIT-IDLE high the header word is selected and sent to the MAIN-HIT-FIFO. The START pulse is actually used to generate a MAIN-HIT-WRITE in order to write the header into the MAIN-HIT-FIFO. In case of a L3-HITS high, an L3-HIT-WRITE is also generated by START to save the same header into the L3-HITS-FIFO. The header word is written way before any hits have time to reach the HIT-READ-OUT because the filtering process starts when all of the roads have been loaded into the ROAD-FIFO. The HIT-READ-OUT is then ready to receive the hits from the STC channels. Two buses converge in this module: HITS-LEFT and HITS-RIGHT. Each of these buses carries the hits generated by half of the STC. A small FIFO that can store up to 16 hits is implemented at the end of each buses. These FIFOs are named SRL-FIFO-LEFT and SRL-FIFO-RIGHT. When full, a HIT-LEFT-FULL (or HIT-RIGHT-FULL) is issued to force all BUS-GNT signals to zero so that no Channels is able to drive the HITS-LEFT (or HITS-RIGHT) bus. The output of these FIFOs are connected to a mutilplexer that is used to select a hit from each FIFO every clock cycle. The bits in each hit word are reorganized into a 32-bit word. Because the SEQUENCER and HDI are not available yet, the SMT-ID and zeros are filling the corresponding place. The 32-bit word is written into both MAIN-HIT-FIFO with a MAIN-HIT-WRITE and L3-HIS-FIFO with a L3-HIT-WRITE (if L3-HITS high). The number of hits written into the buffers is counted with an 8-bit binary counter. The 8-bit NUM-OF-HITS word will be added to the trailer word. This counter is reset every time a START pulse is received, i.e. when a new event starts. The trailers coming from the channels are not included in the counting. If the maximum number of hits is reached (256), MAX-HITS goes with to disable the counter and generate the YANK-HITS signal to all the STC channels.

Each STC issues a trailer that contains the SERR, MM error bits and the FRC-EVENT. The trailers word are not written into the buffers, a single main trailer is inserted at the end of the hit information to the PCI2-OUTPUT. The SERR and MM bits are extracted from the trailers and are registered. If one or more trailer had its SERR bit set, an other SERR bit is generated and added into the main hit trailer. The same thing is done with MM bit. Both registers are also reset with a START pulse. The event number contained into each trailer is input to an 8-bit comparator with the FRC-EVENT stored into the small FIFO. If one or more mismatch occur, an EERR bit is set to one and added into the main hit trailer. This register is reset with START as well. To be able to check the status of the event mismatch bit at any time, the EERR is registered for each STC channel. There are 8 EERR bits that can be consulted at any time through PCI3. These error flags can be read out after a long run to check whether an event number mismatch occurred and with which channel. These registers can be reset by pulling the RST-MM bit (reset bit in the MISC register).

The main hit trailer is generated after receiving the 8 individual trailers from each STC channel. A 3-bit counter is implemented to count the number of trailer seen while receiving hits. When the count reaches 7 and a trailer is seen, the signal HIT-DONE is set to one. This counter is reset at the beginning of a new event i.e. with a START high. A MAIN-HIT-WRITE and L3-HIT-WRITE (If L3-HITS high) are issued to write the main hit trailer into both buffer. HIT-DONE is also sent out to the STC channels to inform them that the event is finished and new hits may be released. The HIT-READ-OUT is brought back to an idle state with a HIT-IDLE high and the next word stored into the event number FIFO is read out. If no more words are present the EMPTY-FIFO signal is high which prevent any START pulse to be issued.

A HIT-AVL is sent to the PCI2-OUTPUT module if hits are available in the MAIN-HIT-FIFO. The MAIN-HIT-FIFO as well as the L3-HIT-FIFO can store up to 256 hits. When either the MAIN-HIT-FIFO is 87% full or the L3-HIT-FIFO is 70% full a FIFO-FULL signal is set to one. A FIFO-FULL signal high to stop the SRL-FIFO-LEFT and SRL-FIFO-RIGHT to be read. These two small FIFOs get full pretty quickly which lead to no more hits transmitted by the channels.

The Level3 buffer is structured as the other L3 buffer in the STC design: after writing the trailer, the read pointer is moved to the next 64-word block. If a 64-word block of hits is available a L3-HIT-AVL is set to one. When the L3-HIT-READ is received, the content of one 64-word block is sent to the L3-RAM module. The Level 3 hits information is identical to the hits sent to the TFC. The ROTATE bit of the configuration word is ignored in this peaces of logic. Both FIFOs are implemented with RAMB4-S16-S16 dual port memory and 8-bit binary counters.

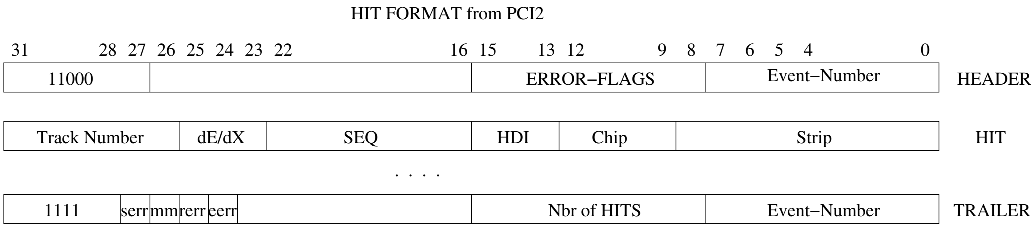

Figure 44: Hits format

6.2 ZSTEREO-READ-OUT

The CONTROL-LOGIC has a ZSTEREO-READ-OUT sub-module that gathers the z and the stereo centroids produced by the STC. The Z centroids are stored in a buffer for later read out by the PCI2-OUTPUT. The Z centroids are sent to the ZVC card if implemented. Another buffer is used to store the same Z centroid information as well as the stereo centroid information. The data for Level 2 and Level 3 are roughly the same but the trailers and headers are different. When a complete event is present in the STC channels buffers, the ZSTEREO-READ-OUT starts scanning over all of them to read out the centroid information. By consulting a memory and the L3-CONFIG word, the logic is able to select only the channels that need to output data. This avoids reading out data that are not needed and save time a lot of time. only one header and trailer are added for the whole z centroif information and one header and trailer is inserted for each level3 data block form one channel.

Information from the PCI1-IF such as the FRC-EVENT, LRB-ERR, L3-ROTATE, L3-CHANNEL L3-90 and L3-STEREO are latched by a FRC-TRAILER to the ZSTEREO-READ-OUT. These words are stored into a EVTNUM-L3CONFIG FIFO. This FIFO is able to store these data for 16 events. The output of the FIFO is identified as FRC-INFO. The information contain in this word is used to add an event number to trailers and headers generated and to determine which type of level 3 data must be saved for this particular event. The LRB-ERR is exactly the same 8-bit word as the one added into the main hit trailer in the HIT-READ-OUT module. L3-STEREO is high when stereo centroids are requested for level 3. It is the same when the L3-90 bit is set; the z centroids are stored in both buffers level3 and 2. To limit the amount of Level 3 data to be produced, the L3-ROTATE bit can be set to one. The logic will only store Level3 z and stereo centroids from the STC channel selected by the L3-CHANNEL word. When L3-ROTATE, the logic assume that all STC channels have to produce L3 data and the L3-CHANNEL word is discarded.

6.2.1 Downloaded parameters

The ZSTEREO-READ-OUT scans over all the STC channels in order to collect all the z and stereo centroid information. Each STC channel stores either z or stereo centroids. A 1-bit parameter Z-SN is associated with every channel: It is one when the channel stores z centroids and low when it generates stereo centroids. This memory is then addressed with the SMT-ID and the output is used to determine which kind of data the ZSTEREO-READ-OUT may expect from this STC channel. This memory is implemented with a RAM16X1D dual port. This parameter is stored at the same time that the DATA-TYPE information is being downloaded into the SEQ-HDI-DATA-TYPE memory of the channels. When a particular channel is selected and when the LSB of the DATA-TYPE is 1, the MSB of the DATA-TYPE is stored at the location defined by the bit 13 to 11 of the ADDR-R address. When the LSB of the DATA-TYPE is high, it can coincide with only two types of data, stereo 01 or z 11. Only the MSB is used to identify the type of data stored. The read back from this memory is performed also at the same time that the SEQ-HDI-DATA-TYPE memory is accessed. This bit is inserted into the higher double-byte word of the PCi3 output bus.

An other control signal is checked before activating the system, the Z-PCI2-ENABLE must be high if the z centroids have to be output through PCI2. If this bit is low the Z centroids are not stored into the Z-PCi2-FIFO. The FRC-EVENT, LRB-ERR and part of the L3-CONFIG are written into the EVTNUM-L3CONFIG fifo with a FRC-TRAILER high. This operation brings down the EMPTY-FRC-INFO signal. In order to activate the ZSTEREO-READ-OUT, a complete event has to be stored in all of the STC channels Z-STEREO-FIFO. An EVENT-DONE signal is issued by the PCI1-IF when the process of an event is done up to the CENTROID-FINDER level (the filtering may not be accomplished by that time). A 4-bit binary counter is implemented to count the number of event available in the buffers. This counter is incremented when an EVENT-DONE is received and decremented when the ZSTEREO-READ-OUT starts to process the centroids of an event. When the count is different than zero, an EVENT-AVL is set to one and the logic is activated. A START signal is issued and brings low the ZS-IDLE signal. The number of event counter is decremented and the counters used to count the number of centroids stored for L2 and L3 are reset. If the Z-PCI2-ENABLE is high in the RUN-CTRL register in PCI3, a Z-PCI2-WRT is generated with the START signal in order to write the z centroids header in the Z-PCI2-FIFO.

Once the ZS-IDLE signal is low, the SCAN-CNT counter is activated. SCAN-CNT is 3-bit binary counter used to execute the scanning over all the channel logic of the STC. The scan starts with SMT-ID zero until SMT-ID 7. For each SMT-ID, the RAM-ZSN is addressed and according to the output bit, the ZSTEREO-READ-OUT knows whether this particular channel generates z or stereo centroids. If the channel in question has z centroids and the Z-PCI2-ENABLE is high, a Z-ENABLE is set to one. Then the same checks are performed for level 3 data: The L3-CONFIG word is consulted and if either z or stereo are requested and the channel selected stores one of these type of data, an L3-ENABLE is set to one. When either L3-ENABLE or Z-ENABLE is set, ENABLE-Z is issued to the STC channels to launch the read out operation. ENABLE-Z is an 8-bit bus and each bit is used to enable the read operation for a particular channel (look at figure) In case of L3-ENABLE low and Z-ENABLE low, the particular channel does not store the relevant data and must be skipped. A SCAN signal is sent to the SCAN-CNT to increase its output value. Before reading out the data for Level3, an l3 header with the SMT-ID of the selected channel is written if the L3-ENABLE bit is set. A 4 to 1 multiplexer is implemented to insert the headers and trailers for both information. The selection is made among the Header word, the centroid word, and the trailer for z or the l3 trailer. The header word for z and l3 are very similar only the SMT-ID is an extra information added for L3. The 16-bit centroid information is reorganized into a 32-bit word and the SMT-ID is added to each word instead of the SEQUENCER and HDI ids (The same has been done for the hit information in the HIT-READ-OUT). The centroids are then stored into a Z-PCI2-FIFO and/or a L3-PCI3-FIFO. Two 8-bit binary counters are implemented to count the total number of z centroids stored into the Z-PCI2-FIFO and the total number of L3 centroids produced by the selected channel logic. The 8-bit NUM-OF-Z is inserted into the z trailer and the 8-bit NUM-OF-L3 is added to the L3 trailer. Both counters are reset when a new event is being processed .i.e when START is set to one. The maximum number of z as well as level 3 centroids must not exceed 256. When the count reaches the maximum value, the respective write signal is brought low and no more centroids are written into the FIFO. Only the headers and trailers are transmitted. When reading out the centroids from one channel logic, the trailers end the block of data. This trailers contains an event number that is compared with the one stored into EVTNUM-L3CONFIG FIFO. If it's a mismatch, the EERR bit is set to one in the L3 trailer. This bit is also registered and if one of the channel had a event number mismatch, the EERR bit of the Z trailer is set. This register will be reset when a new event is being processed. The same logic is implemented for the SERR and the MM bit. Basically, when the trailer word is received from the channel logic, the l3 trailer word is written and a SCAN is set in order to select the following STC channel. When the SCAN-CNT reaches the last SMT-ID and when the trailer is received, a SCAN-DONE signal is generated. The SCAN-DONE signal sets the ZS-IDLE back to one and generates a write signal to the Z-PCI2-FIFO to write the main z trailer word. A write signal is generated at the same time to the l3 FIFO, to write an extra word at the end of the Level 3 data (look at format). The next FRC-INFO word is read out form the EVTNUM-L3CONFIG FIFO. If this FIFO is empty EMPTY-FRC-FIFO is high and the ZSTEREO-READ-OUT remains in the idle state. Otherwise, the logic starts processing the next event as soon as EVT-AVL is high. In order to read out the data for the next event, the read pointer of all the Z-STEREO-FIFO must be moved to the location of the beginning of the following event.

6.2.2 storage

Two 256X32 fifos are implemented: Z-PCI2-FIFO that stores the z centroids and the L3-PCI3-FIFO that stores the Level 3 z and stereo centroids. The Z-PCI2-FIFO is built with 2 RAMB4-S16-S16 dual port memories and two 8-bit binary counters to generate the read and write addresses. A subtractor is also implemented to generate a WRD-CNT (word count). The WRD-CNT informs the logic of how many centroids are present in the FIFO. A Z-AVL is set to one when the WRD-CNT is not zero. This signal is then used to tell the PCI2-OUTPUT that z centroids are ready to be sent to the ZVC. The L3-PCI3-FIFO is implemented as the other L3 buffers, the memory space is divided into 4 64-word block of data. After writing the last word of an event, the write pointer is moved to the next 64-word block. The FIFO is implementing the same dual port and counters as the Z-PCI2-FIFO. When a 64-word block is present in the FIFO, a L3-ZS-AVL is issued to the L3-RAM module. A L3-ZS-READ is then received by the ZSTEREO-READ-OUT when the data are requested. The whole block of data is read out in once. Note that if the event generates less than 64 words, the L3-ZS-AVL is set to one right after writing the trailer word.

When the 95 % of the Z-PCI2-FIFO is occupied with data or 70% of the L3-PCI3-FIFO is used, a FIFO-FULL is set to one and no more centroids are being written into the fifos. The ENABLE-Z bus is forced to zero as long as no memory space is available.

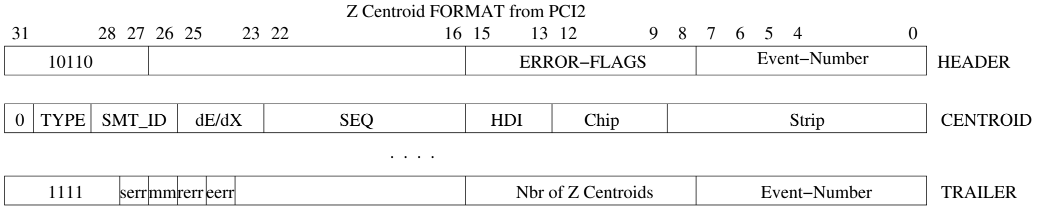

Figure 45: Z and Stereo centroids format

6.3 PCI2-OUTPUT

The PCI2-OUTPUT module provides the interface between the STC Xilinx chip and the PCI2-Altera-chip. This logic is used to output the 32-bit Z centroids and HITS to the ZVC and the TFC. Control signals are also provided to the PCI2-ALtera-Chip to identify the type of data sent as well as to determine whether the bus is driving valid data or not. A 50MHz connection is established between both chips. The HITS and Z-CENTROIDS are formatted as expected by the other card of the STT.

6.3.1 SEQ and HDI

The SEQUENCER and HDi ids are needed to achieve the final format for the HITS and Z-CENTROIDS. These words are actually stored locally into a mirror memory. The SEQUENCER and HDI are saved into two RAM16X8 at the same time that these parameters are being written into the SEQ-HDI-DATA-TYPE memory. The bits 13 to 11 of the ADDR-R address from PCI3-IF is used to store the SEQUENCER and the HDI ids according to their SMT-ID. The read back from this memory appears as bit 16 to 23 of the PCI3 output bus. The read back can actually be used to check whether the correct information had been downloaded into this memory. In order to finalize the format the HITS and the Z-CENTROIDS, the SMT-ID located as bit 22 to 20 on each 32-bit word is used to address this memory. The SEQUENCER and HDI replace the SMT-ID and the zeros between bits 22 to 13. Two RAM16X8 are needed because the SEQUENCER and HDI words are written on after the other into the SEQ-HDI-DATA-TYPE memory block. The SEQUENCER can be stored in one of them and the HDI in the other so that both information are delivered at the same time when addressing instead of one at a time with a unique memory space.

6.3.2 Sending HITS and Z-CENTROIDS out

The HITS are output to the PCI2 whenever the are available and Z-CENTROIDS are sent out when no HITS are present in the HIT-READ-OUT. This scheme gives the highest priority to HITS over the Z-CENTROIDS. This method can be modified if the implementation of the ZVC reveal problems. HITs and Z-CENTROIDS are stored into two separate buffers into the PCI2-Altera-chip. The status of these buffers is transmitted to the PCI2-OUTPUT module as PCI2-CTRL(0) and PCI2-CTRL(1). These bits are the full signal from the fifos: PCI2-H-FULL (PCI2-CTRL(0)) and PCI2-Z-FULL (PCI2-CTRL(1)). When a HIT-AVL is received form the HIT-READ-OUT and PCI2-H-FULL is low, a READ-HIT is immediately generated in order to start sending out hits. READ-HIT is issued every two clock cycle as long as HIT-AVL is high. When HIT-AVL goes low or the PCI2-H-FULL is high and if Z-AVL is high and PCI2-Z-FULL is low, READ-Z is sent to the ZSTEREO-READ-OUT. READ-Z is also issued every two-clock cycle. The HITS and the Z-CENTROIDS are input to a 2 to 1 32-bit multiplexer. The selection is done looking at the READ-Z signal delayed by one clock: when low the HITS are selected and when high the Z-CENTROIDS are sent out. The output of the mulstiplexer is registered as PCI2-DATA. The 7-bit SEQ and the 3-bit HDI replace bit 22 to 13 of PCI2-DATA if the word is not a trailer. The 32-bit PCI2-DATA is output in conjunction with a low active valid signal as PCI2-CTRl(4) and the selection bit of the multiplexer is registered and output as PCI2-CTRL(5). PCI2-CTRL(5) is high if the corresponding word is a z centroid. The informmation is output to the PCI2-Altera-chip at a rate of 50 MHz.

When the hit trailer word is detected, the READ-HIT is reset and remain low until EVT-FINISH (event finish) is issued. A HIT-FINISH signal is set to one and remains high as soon as the trailer is input. When the z trailer is received, the READ-Z is reset and the Z-FINISH signal is set to one as well. When both Z-FINISH and HIT-FINISH are high, all the hits and the z centroids of an event have been sent out. An EVT-FINISH is issued to reset HIT-FINISH and Z-FINISH and release the READ-HIT and READ-Z signals. The next event can now be treated. If the ZVC is not implemented, Z-PCI2-ENABLE is set to zero in the RUN-CTRL register and in this case, Z-FINISH remains high so that an EVT-FINISH is pulled as soon as HIT-FINISH is seen. The trailer words are sent out to the PCI2 without modifications (no SEQ and HDi are needed).

LEVEL3 LOGIC:

L3-CL-READ-OUT:

This module is used to collect and format the L3 cluster information. The data are stored in a buffer before sending them out to the L3 external SDRAM. Contrary to the other L3 data, only one channel is selected to output L3 cluster data. The cluster data are then received from the CENTROID-FINDER of the selected STC channel. The input data contains all the useful control signal to reconstruct the clusters.

In order to format data, information such as the thresholds used by the clustering algorithm and the channel select in the L3-CONFIG word must be available. The process of reconstructing the cluster is described as well as the storage.

Downloaded parameter:

In order to format the L3 cluster data, the thresholds used in the cluster finder must be stored. This thresholds are displayed in the strip count word (see the format). Instead of carrying this information through the L3-CLUSTER-DATA sub-module of the CENTROID-FINDER, these words are stored locally. The same method has been used for the SEQ and HDI identification numbers. The format is finalized just before the output. When downloading the THRESHOL memory of each STC channel, a copy of the thresholds is stored into a Dual port memory in the L3-CL-READ-OUT. Each 8-bit threshold is stored in the mirror using the SMT-ID as address. The read back from the THRESHOLD memory, contain the value stored in the mirror in the bits 23-16. Thus the content of this memory can be checked.

The memory is loaded when a WR-THRESHOLD is issued from the PCI3-IF. The 8-bit threshold in the W-DATA bus is written at the location pointed by the SMT-ID and the type. The second threshold of the same type is written in the following location. This is to output both thresholds of the same type at the same time when processing data. The read back is executed when the RD-TRESHOLD is received.

Another information is needed for the format: the CALCULATION-TYPE. This bit is provided by the RUN-CTRL register in the PCI3-IF. This bit selects the maximum number of strips within the cluster.

L3-CONFIG information

The STC channel that is to output L3 cluster data is selected by a 3-bit counter implemented in the PCI1-IF. The output of this counter is a 3-bit word L3-CHANNEL that is part of the L3-CONFIG word. This L3-CHANNEL is used to generate the L3-RC-CLSTR-ENABLE signal to the L3-CLUSTER-DATA module. The input 26-bit L3-CLUSTER-DATA bus carries only the data from the selected channel.

In this module the FRC-EVENT is not saved. This is because the L3-CL-READ-OUT may start receiving cluster data before the FRC block is input to the PCI1-IF. No buffers are implemented in each STC channel to store the cluster data. The amount of data produced by the STC channel can be very big and would need a buffer big enough for several event. Furthermore, each STC channel would need its buffer and the read out would have been more complicated. It is the reason only one channel can output L3 cluster data at a time. Because the amount of data is relatively important, they must be stored in the L3 SDRAM right away to avoid the implementation of large storage. The L3 cluster data are actually be transferred to the L3 external ram as they are produced. Sometimes, the FRC data can already be present so the event number can be inserted but some other time the FRC data can trail the SMT data and the FRC-EVENT would not be available. Thus, the event number is not added in the trailer and no information from the PCI1-IF is needed here. An event number is still added in the trailer but it is the one received from the STRIP-READ module.

L3 Cluster data:

Each word data received from the L3-CLUSTER-DATA contains the address (CHIP+STRIP), the data, the type and 4 control signals. The control signals are going to be used to reconstruct the cluster information. First of all, the input data is considered valid when its bit 25 is low. The valid bit is a low active signal. A small fifo is implemented in the first place to store the incoming data. The CLSTR-FIFO is a 16X21 implemented with distributed memory of the Xilinx chip.

The frist data word received has its NEW3 bit set to one. This means that a new cluster is starting. Though, this data is not stored in the fifo unless the coincident LOAD signal is high. The LOAD signal is high when a center has been detected. This signal set to one when the center is two data word away. The data that has its LOAD signal is the first strip of a cluster. The GOT-CLUSTER signal is then set to one to enable the writing of the following strips. The location of the first word of the cluster is saved into a 4-bit register as WP.

If the number of data exceeds 5, the HOLD signal is set to one. A HOLD bit high disable the writing of the strips in the fifo. No more data wre written until the cluster ends or a new maximum above the previous one is detected. If the cluster ends the data that starts a new cluster has its NEW3 set to one. The location of the beginning of the next cluster overwrite the previous one. The WP register is updated. Thus, the cluster is saved into the fifo and cannot be overwritten. If a new center is detected the LOAD signal is received and the corresponding data is saved into the fifo. The LOAD signal moves the pointer back to the location indicated by WP. The first cluster is overwritten.

As cluster are being reconstructed, the number of strips received in a cluster is calculated with a 10-bit counter. The counter is incremented continuously between two NEW3 signal received. This signal define the boundaries of a cluster. In this case the boundaries include all the strips received and can be greater than 5. The 10-bit STRIP-COUNT is also saved into a small fifo when the cluster ends. The CHANNEL-LOGIC receives data from at most 5 SVXII chip that have axial strips. The cluster can then reach a size of 640 strips according to this algorithm. So the STRIP-COUNT would need 10-bit.

The whole idea of having fifos to store the cluster and their STRIP-COUNT is to have enough time to format data. The Cluster could have been sent out right away followed by the STRIP-COUNT but the format needs the STRIP-COUNT before the cluster information. Then the cluster is reconstituted and then the STRIP-COUNT is available.

The STRIP-COUNT is read out and sent to the logic that formats data. The cluster information is written afterwards. The fifo are used to store the incoming cluster while the previous one is being formatted and stored.

When the END-OF-EVENT bit is set to one, the corresponding word is written in the CLSTR-FIFO. This word contains the event number instead of the 8-bit data. At the same time as saving this word, the WR-EOF signal issued. This signal sets the FINISH bit to one which remains one until the trailer is written.

FORMAT THE CLUSTER DATA.

The cluster data, as the other L3 data block, has a header and a trailer word. The formatted data are stored into a 256X32 fifo. The header with the 3 bit L3-CHANNEL is written when the first valid data is received from the selected channel. When the fifo that stores the STRIP-COUNT is not empty, a word is read out. At the same time the L3-CHANNEL and the DATA-TYPE of the cluster address the threshold mirror. The STRIP-COUNT, the DATA-TYPE, the calculation type and the two thresholds are written into the fifo. The CLSTR-FIFO is then read out to get the cluster information. The data are read out continuously and transferred to the fifo every clock cycle. The last word of the cluster enable the reading of the STRIP-COUNT fifo. If this fifo is not empty, the same procedure is repeated. After finishing writing the last cluster, if the FINISH signal is one the trailer word is sent to the FIFO.

The format is the following:

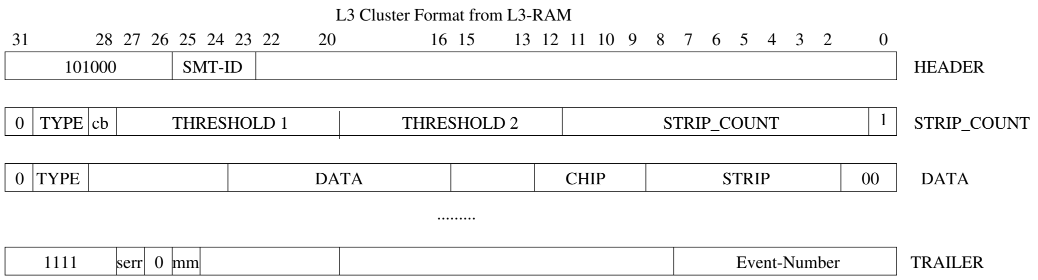

Figure 46: L3 Cluster Format

When a block of 64 words of level 3 cluster data is available, the L3-CLSTR-AVL signal is sent to the L3-RAM module. This signal is also set to one if less than 64 words have been written but the trailer is already present in the FIFO. The L3-READ-CLUSTER from the L3-RAM starts the read out of an entire block of 64-words to the L3-RAM. The maximum number of cluster can get to 320. the entire event can not be stored in this fifo. The L3-RAM modules needs to issue L3-READ-CLUSTER as soon as the L3-CLSTR-AVL is high. The transfer priority to the L3 ram is given to the L3 cluster and the L3 Raw Corrected data.

full signal

L3-RC-READ-OUT:

This module receives the level 3 raw and corrected data and store them. This module does not format the data it only inserts a header and a trailer word before and after the block of data. As the Level3 Cluster data, only one channel is to output L3 Raw and Corrected data. The channel is selected with the 3-bit L3-CHANNEL word fro the L3-CONFIG.

The L3 raw and Corrected data can be pretty big in size. As for the L3 cluster, the Raw and Corrected data cannot be stored in large buffers. This would need way too much storage space. So only a small buffer is implemented but the read out to the L3 RAM has the priority over the other type of L3 data. If all the strips of a detector are hit, the number of raw data can get to more than a thousand. The Raw and Corrected data are merged together in order to limit the amount of word produced by the STC channel. These data are compacted to save buffer space.

The channels are connected as shown in figure:

Storage of L3 raw and corrected data:

The L3 raw and Corrected data are received as series of 32-bit from the channel selected. The it 33 is used as a valid signal, it low active. The data are written into a 256X32 fifo as they are received. The trailer word from the channel is reorganized to look like the other blocks of L3 data.

the header

The format is actually almost identical as the input one:

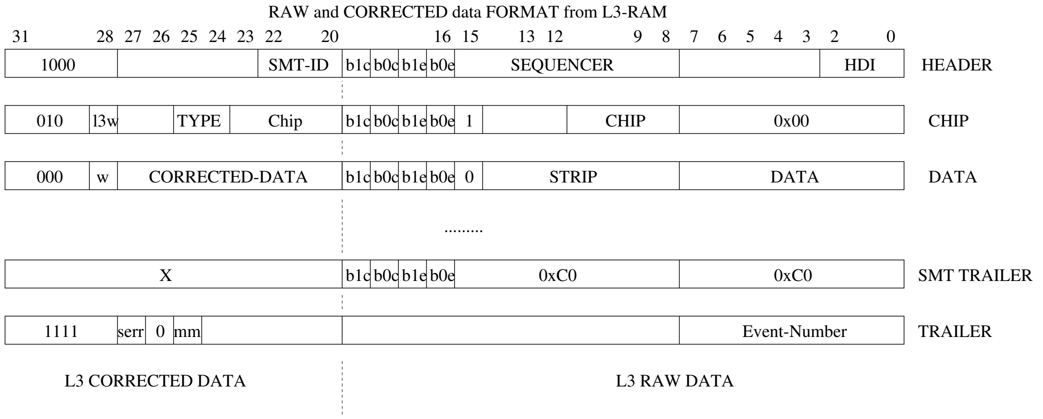

Figure 47: L3 Raw and Corrected data format

Once 64 words are there or the trailer, the L3-RC-AVL is sent out. The L3-READ-RC ignite the output of an entire block of 64 raw and corrected data words.

full signal

L3-AX-READ-OUT:

This module is used to collect the Level3 axial information. It implements the exact same logic as the ZSTEREO-READ-OUT logic. The L3-CONFIG word from the PCI1-IF determine whether L3 axial information must be produced and from which channel if the L3-ROTATE bit is one. The logic implements a scanner that selects the first STC channel, start reading out the data and move to the next channel once the trailer is received. The data are formatted with the FRC-EVENT from the PCI-IF. A header and a trailer word are added. The L3 axial data are stored into a 256X32 fifo. The read out procedure is implemented the same way as the othe L3 read out logic, the storage space is divided into 4 64-word block.

The format to the L3-RAM module is the following:

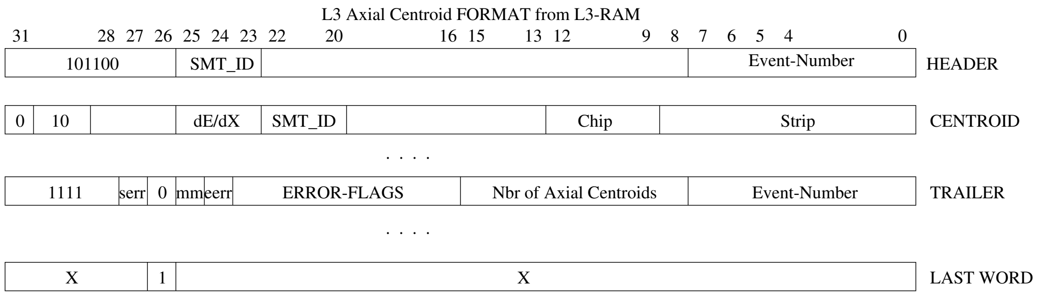

Figure 48: L3 Axial centroid format

7 ANNEX 1: STC Layout Schematic

The layout shown in figure 49 displays the physical organization of the STC logic board:

- Xilinx XCV812E chip (in the middle). This FPGA chip performs the clustering of SMT data and the filtering of centroids. It also produces the Level3 data.
- 2 SDRAMs (Above the Xilinx chip): One of them stores the Road LUT and the other one the Level3 data.
- 3 PCI Altera Chips (upper and lower right corners and left side): These Chips are used as interfaces to the 3 motherboard PCI buses.

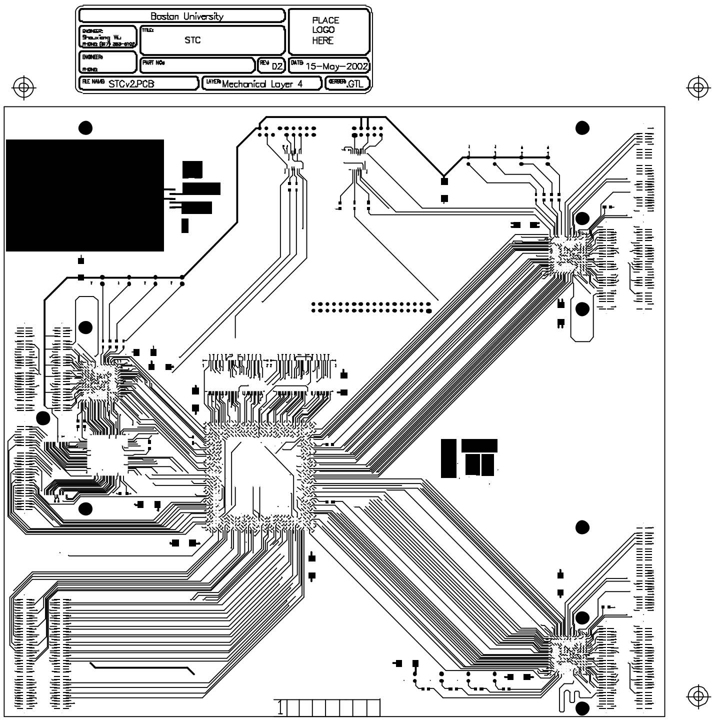

Figure 49: STC layout schematic

8 ANNEX 2: Clusters and Centroids

This annex gives the cluster algorithm output and the corresponding centroids assuming the following test vector:

Note that the Address here is the 4-bit CHIP-ID and the 7-Bit STRIP-ID. The DATA is the ADC value recorded on the associated STRIP-ID.

ADDRESS (Hex)	DATA (hex)	ADDRESS (Hex)	DATA (hex)
001	06	01C	06
002	08	01D	08
003	0A	01E	0B
004	08	01F	02
005	06	020	01
00C	02	022	08
00D	08	023	0B
00E	0A	024	0B
00F	0A	025	0C
010	0A	026	06
015	0B	027	08
017	06	028	0A
018	08	029	08
019	0A	02A	06
01A	08	030	0C
01B	06	031	06

The assumed thresholds are:

- THRESHOLD1 = 3
- THRESHOLD2 = 9

The clusters and their centroids found in this event are presented in this table (All values are in hex). The ADC values retained for each clusters are named D1, D2, D3, D4 and D5. For the case of the three strips operating mode, only D2, D3 and D4 ADC count are relevant.

D1	D2	D3	D4	D5	Cluster Center	q (bin)	CENTROID
06	08	0A	08	06	003	00	000C
0A	0A	0A			010	-11	0041
		0B			015	00	0054
06	08	0B			01E	-11	0079
0B	0B	0C	06	08	025	00	0025
		0C	06		030	01	00C1

Centroids in case of three strips mode:

q (bin)	CENTROID
00	000C
-10	0042
00	0054
-01	007B
00	0025
01	00C1

9 ANNEX 3: STC Output Format

9.1 HIT format to TFC

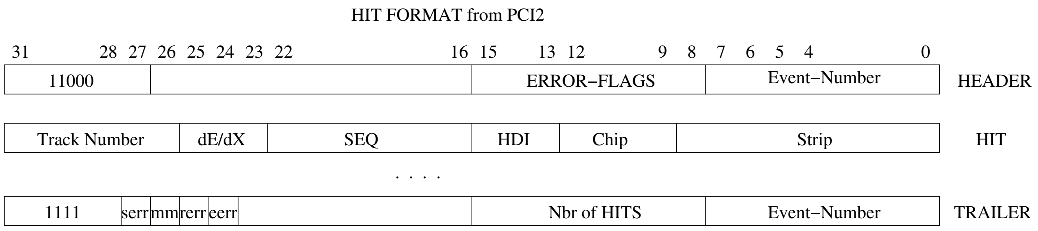

Figure 50: Hits format

- HEADER:
  - Bit 7-0: EVENT-NUMBER. The event number is in fact the 8-bit L1-BX (Bunch number within turn) word from the FRC header word.
  - Bit 15-8: ERROR-FLAGS. This 8-bit word is from the FRC trailer word.
- HIT:
  - Bit 8-0: STRIP-ID. This word contain the 7-bit STRIP-ID received from the VTM and the two precision bits added by the STC.
  - Bit 12-9: CHIP-ID. This word identify the SVXII chip from which the hit comes from.
  - Bit 15-13: HDI-ID.
  - Bit 22-16: SEQUENCER-ID. As received from the SMT detector read out, the SEQUENCER-ID is an 8-bit word. The bit 3 is omitted in the HIT output format.
  - Bit 25-23: DE/DX. This is the pulse height of the cluster from which the HIT has been computed.
  - Bit 31-26: TRACK-NUMBER. This word is used to identify the roads from the FRC. It tells which track is associated with this hit.
    NOTE: This number starts from 0.
- TRAILER:
  - Bit 7-0: EVENT-NUMBER. Same as the HEADER word.
  - Bit 15-8: Nbr of HITS. This word tells the number of hits found by the 8 channels of the STC for this event.
  - Bit 24: EERR. This error bit is high if the EVENT-NUMBER added in the SMT data differs from the FRC one.
  - Bit 25: RERR. High if any error flag was set in the ERROR-FLAGS word loacted in the HEADER.
  - Bit 26: MM. If any of the STC channel had this bit set to one.
  - Bit 27: SERR. If any of the STC channel had this bit set to one.

9.2 Z-Centroid format to ZVC

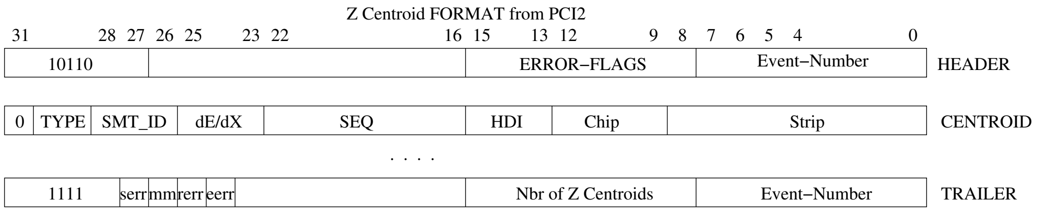

Figure 51: Z-Centroids format

- HEADER:
  - Bit 7-0: EVENT-NUMBER. The event number is in fact the 8-bit L1-BX (Bunch number within turn) word from the FRC header word.
  - Bit 15-8: ERROR-FLAGS. This 8-bit word is from the FRC trailer word.
- Z-CENTROID:
  - Bit 8-0: STRIP-ID. This word contain the 7-bit STRIP-ID received from the VTM and the two precision bits added by the STC.
  - Bit 12-9: CHIP-ID. This word identify the SVXII chip from which the centroid comes from.
  - Bit 15-13: HDI-ID.
  - Bit 22-16: SEQUENCER-ID. As received from the SMT detector read out, the SEQUENCER-ID is an 8-bit word. The bit 3 is omitted in the HIT output format.
  - Bit 25-23: DE/DX. This is the pulse height of the cluster from which the centroid has been computed.
  - Bit 28-26: SMT-ID. This is the STC channel number from which the centroid has been issued.
  - Bit 30-29: TYPE. This is the type of centroid: axial, stereo or z. It must be 11 here (z centroids).
- TRAILER:
  - Bit 7-0: EVENT-NUMBER. Same as the HEADER word.
  - Bit 15-8: Nbr of Z-CENTROIDS. This word tells the number of 90 degrees centroids found by the 8 channels of the STC for this event.
  - Bit 24: EERR. This error bit is high if the EVENT-NUMBER added in the SMT data differs from the FRC one.
  - Bit 25: RERR. High if any error flag was set in the ERROR-FLAGS word loacted in the HEADER.
  - Bit 26: MM. If any of the STC channel had this bit set to one.
  - Bit 27: SERR. If any of the STC channel had this bit set to one.

10 ANNEX 4: STC Internal Format

10.1 STRIP-READER output:

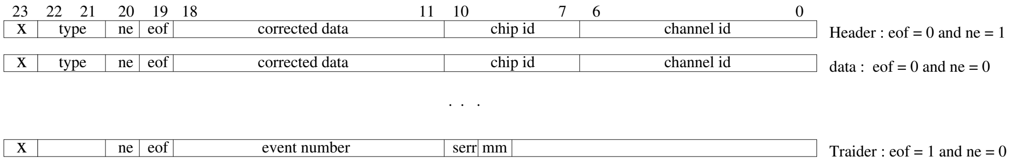

Figure 52: Strip reader output format

10.2 L3-RAW-CORRECTED data:

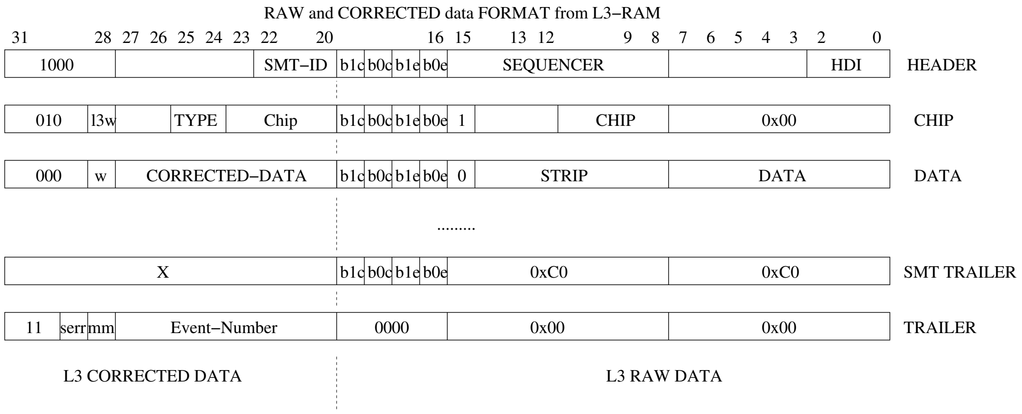

Figure 53: L3 Raw and Corrected data format

Bits description of the L3 RAW-CORRECTED-DATA

- HEADER:
  - Bit 22-20: SMT-ID. This the STC channel number.
  - Bit 19-18: b1c and b0c. These are the CAV-DAV-LNK bits of the byte 1 and 0 respectively.
  - Bit 17-16: b1e and b0e. This is the VTM error bit of byte 1 and 0 respectively.
  - Bit 15-8: SEQUENCER-ID. This is the SEQUENCER-ID received from the SMT.
  - Bit 2-0: HDI-ID. HDI-ID received from the SMT.
- CHIP word:
  - Bit 28: l3w. This bit is high if the CHIP-ID received is illegal.
  - Bit 25-24: DATA-TYPE.
  - Bit 23-20: CHIP-ID. This is the CHIP-ID saved by the STRIP-READER logic.
  - Bit 11-8: CHIP-ID. This is the CHIP-ID received from the SMT.
- DATA word:
  - Bit 28: w. This bit is one if the corresponding CORRECTED-DATA was passed on to the CENTROID-FINDER.
  - Bit 27-20: CORRECTED-DATA. Data corrected for gain and offset by the STRIP-READER.
  - Bit 14-8: STRIP-ID. The STRIP-ID received from the SMT.
  - Bit 7-0: DATA. This is the 8-bit data from the SMT.
- SMT TRAILER:
  - Bit 15-0: 0xC0C0. Those two 0xC0 correspond to the SMT trailer.
- TAILER:

- Bit 27: SERR.
- Bit 25: MM.
- Bit 7-0: FRC-EVENT. This is the event number from the CONTROL-LOGIC.

10.3 L3-CLUSTER-DATA format:

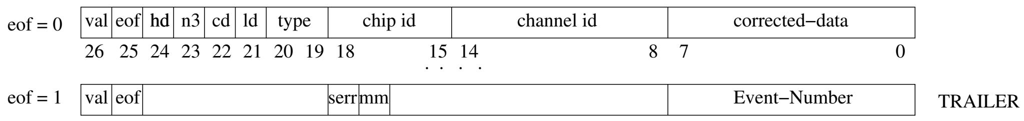

Figure 54: L3 Cluster format

11 ANNEX 5: Description of an Accumulator:

The accumulators used here are created from loadable adder-subtractor logic module. A simple adder-subtractor becomes an accumulator when its output is used as a feedback (figure 55). Its input and output ports as well as the truth table are described here:

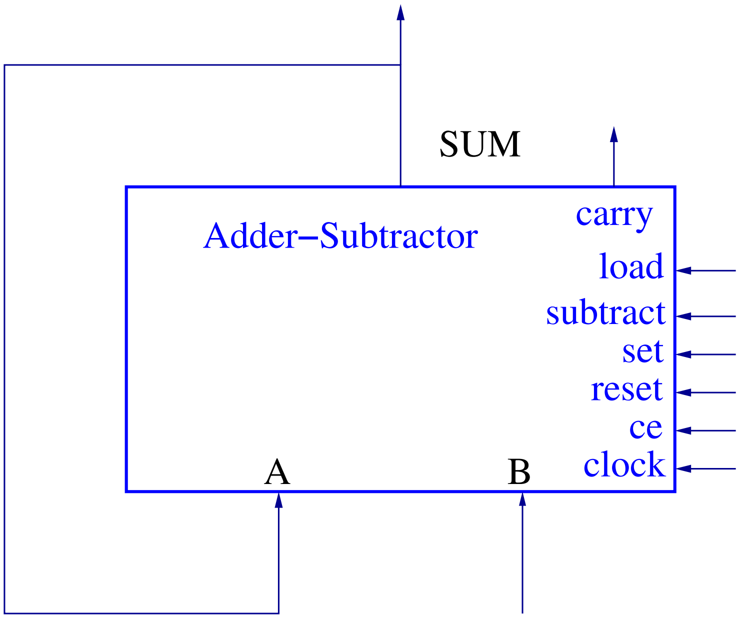

Figure 55: Accumulator

```
-- Input(s): a, b, subtract, load, reset, set, ec, clk
-- Output(s): sum, carry
-- a,b,sum all unsigned
-- truth table
-- reset        set     ec      load    subtract        sum
-- 1            x       x       x       x               others => '0'
-- 0            1       x       x       x               others => '1'
-- 0            0       0       x       x               no change
-- 0            0       1       1       0               b
-- 0            0       1       1       1               -b
-- 0            0       1       0       0               a + b
-- 0            0       1       0       1               a - b
-- subtract     carry   meaning
-- 0            0       no carry out
-- 0            1       carry out
-- 1            0       borrow out
-- 1            1       no borrow out
```

12 ANNEX 6: Accumulator Equations for the CENTROID-FINDER module.

This annex gives the equations used to determine the ports of both accumulators (ACC and WACC) implemented in the PIPE-ACC sub-module of the CENTROID-FINDER. As a remainder, these two accumulators are calculating the DENOMINATOR and NUMERATOR words of every clusters. These 2 components have to handle all kind of clusters according to the algorithm described in the introduction. Their input ports are defined by equations of signals generated by the pipeline such as NEW1, NEW2, NEW3, MAX1, MAX2, and MAX3. Two additionnal signals are also involved: HOLD and SUBTRACT which are described here as well. The ACC and WACC are run in 3-strip mode when the CH3 signal is set in the CONTROL-LOGIC (bit 6 RUN-CTRL REGISTER, see PCI3-IF module).

1. ACC accumulator equations:

   - SET $\Rightarrow$ ground (not needed)
   - RESET $\Rightarrow$ ground (not needed)
   - CLOCK $\Rightarrow$ system clock
   - SUBTRACT $\Rightarrow$ ground (This accumulator adds only)
   - LOAD $\Rightarrow$ ($NEW3$ and ($MAX3$ or $MAX2$)) or ($MAX1$ and $\overline{NEW1}$ and $\overline{NEW2}$) or ($MAX2$ and $\overline{NEW2}$ and $CH3$)
     - $NEW3$ and ($MAX3$ or $MAX2$):
       The ACC must be loaded when a cluster starts ($NEW3 = 1$) with a possible center stored into the pipeline ($MAX3 = 1$ or $MAX2 = 1$).
     - $\overline{NEW1}$ and $\overline{NEW2}$:
       By looking at condition 4 on fig 56, another maximum could be found above the first one so the ACC must be loaded if $\overline{NEW1}$ and $\overline{NEW2}$ is true.
     - $MAX2$ and $\overline{NEW2}$ and $CH3$:
       For the 3-strip mode, the cluster starts with the first strip before the maximum.
   - EC $\Rightarrow$ $SELECT - TH2$ and $\overline{HOLD}$:
     The ACC output is updated when $SELECT - TH2 = 1$. The control flag HOLD is used to stop the accumulator when five data have been already added and the cluster is not yet finished. The HOLD signal is high after 3 words added in case of 3-strip mode.

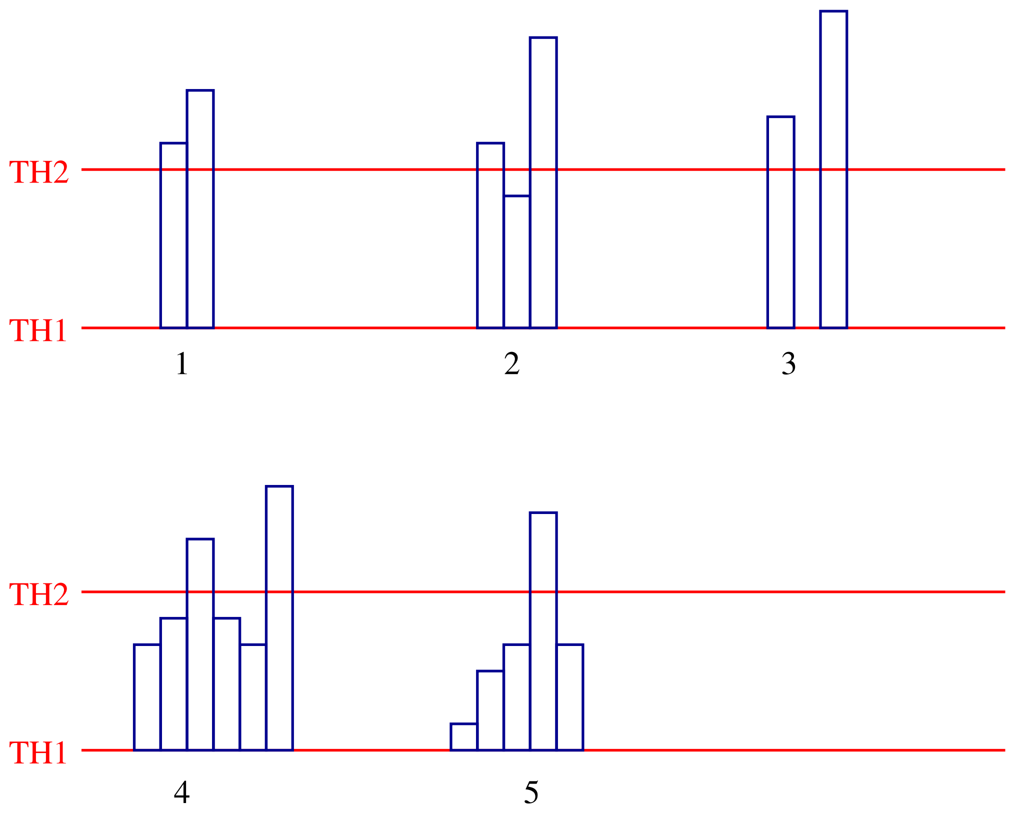

Figure 56: Possible Clusters

2. WACC accumulator equations:

- SET ⇒ ground (not needed)
- RESET ⇒ $MAX3$ and $NEW3$:
  If a single strip cluster is at the ouput of the pipeline, its NUMERATOR is null. So the ACC is reset to force the output to 0.
- CLOCK ⇒ system clock
- SUBTRACT ⇒ SUBTRACT
  The SUBTRACT signal is high before the maximum has been received and high after receiving it. The reason for that is the ADC values before the center must be subtracted and the ADC values after must be added in order to get the following result:$-2 \times d1 + -1 \times d2 + 0 \times d3 + 1 \times d4 + 2 \times d5$.
- LOAD ⇒ $\overline{SELECT - TH2}$ and ($NEW3$ or ($\overline{MAX2}$ and $SUBTRACT$) or ($\overline{MAX3}$ and $SUBTRACT$ and $CH3$))
  - $\overline{SELECT - TH2}$:
    The WACC has to be loaded within one system clock.
  - $NEW3$ or ($\overline{MAX2}$ and $SUBTRACT$):
    This accumulator is loaded when either a cluster starts ($NEW3 = 1$). The first strip of a cluster ($NEW3 = 1$) may not be included at the end because only 2 strips before

the center are kept. This situation is illustrated on fig 56. In condition 5, the first strip must be discarded.

- $\overline{MAX3}$ and $SUBTRACT$ and $CH3$:
  In 3-strip mode, the two first strips are discarded in condition 5 (fig 56).

- EC $\Rightarrow$ $\overline{MAX3}$ and $\overline{HOLD}$ and NOT($SELECT{-}TH2$ and (($\overline{SUBTRACT}$ and $\overline{LAST-CH}$) or ($SUBTRACT$ and $MAX2$)))
  - $\overline{MAX3}$:
    When a maximum is detected ($MAX3 = 1$, the accumulator is disabled in order to apply the weight 0 to its ADC value.
  - $\overline{HOLD}$:
    After five ADC values, HOLD is high so this accumulator is stopped. Only three if $CH3 = 1$.
  - $SELECT - TH2$ for synchronous reasons.
  - $\overline{SUBTRACT}$ and $\overline{LAST - CH}$:
    The LAST-CH signal is one when the fifth data is read out from the pipeline. This condition enables the accumulator for one clock after the maximum to add the fourth data to the NUMERATOR : +d4.
    For the calculation with three strips, LAST-CH is not used. When $CH3$ is active, and the signal SUBTRACT becomes null after the maximum and HOLD is set to 1. These condition, allowed to read out one more value after the maximum.
  - $SUBTRACT$ and $MAX2$:
    Before the maximum, the weight applied to the ADC value has to be -1 so the WACC is enabled.

HOLD and SUBTRACT:

- HOLD:

  In 5-strip mode this signal turns high when two strip ADC values have been read tou of the pipeline. In the 3-strip mode, it becomes high after reading one ADC value after the maximum.

- SUBTRACT:

  This signal is used to control the SUBTRACT port of the WACC accumulator. It is high to subtract input ADC values and it is low to add them.