A.1 SIGNALS FROM THE MAIN CONTROL MODULE

Start

Download the memory data

Enable the channel

Issue event start
A.2 SMT DATA FILTER

Start
Channel enable

The first 8bit read in as upper byte and next 8 bits as lower byte
The error bits are tagged to the 16 bits to form 18 bit stream

Is the byte C0

Write the last word errors bits, C0 and event number

Give the end of event signal to the strip reader control block

Write word in FIFO and go to S0

A
A.3 STRIP READER CONTROL

Start
Event_start

This is the first word

Check for Seq_id and Hdi_id, increment mis-match counter if not matching
Write raw data to raw data L3 buffer

Check the chip id for the data type

Check for the chip id and a byte of zeros, if byte of zeros not present - increment the zero -error counter.
Write raw data to the raw data L3 buffer

Check for Seq_id and Hdi_id, increment mis-match counter if not matching
Write raw data to raw data L3 buffer
A.3 STRIP READER CONTROL (CONTINUED....)

Write

Write the data in output FIFO

Read_FIFO

Read the input FIFO

Previous State - First read

Previous State - Seq_hdi

Previous State - Chip_zero/fill_output_data

Fill_output_data

Get the corrected data from the gain_offset LUT. Format data in 23 bit register.

Increment chip counters.

Wait

"C0" encountered

Set end of event

A

D

 Chan_value

Write the channel id and the corresponding data. Check for 'C0'

B

C

E
A.4 CLUSTER FINDER

**Start**
Event start
Return state - Init

**Read**
Read the 23 bit word from the FIFO

**Check for end of cluster**
1. Current type different
2. Data value < threshold1
3. Current address not in sequence

Yes: Send the available data values for calculation

**Calculate**
Peak of the cluster > threshold2

No: Write
Write the cluster into L3 buffer.
Write the centroid from the Calculator block into the output FIFO and L3 buffer.

**Init**
Load first data value and address in data3 and add3. Go to read.
Return state - main

**Main**
If ndata > data 3

Yes: data 1 <---- data 2
data 2 <---- data 3
data3 <---- ndata
same for add. Return state - main

No: data4 <---- ndata, same for add
Return state - next

**Note:** To write cluster data into L3 buffer, takes 6 cycles. The machine is in wait state for these cycles.
A.4 CLUSTER FINDER (continued)

Next

Yes

If ndata > data 3

data 1<--- data3
data 2 <---- data 4
data3 <----- ndata, same for add.
Go to read. Return state - main

No

Return state - outpeak

data5 <--- ndata
Shadow registers-
data_shadow1 <-- data4
data_shadow2 <--- ndata, same for add
Return state - outpeak

Outpeak

Yes

If ndata > data 3

data1 <--- data_shadow1
data2 <--- data_shadow2
data3 <--- ndata,
data4 and data5 <--zero
same for add
Return state - main

No

data_shadow1<--- data_shadow2
data_shadow2 <---- ndata
, same for add.

D

D

D
A.5 HIT FILTER – Hit filter control logic

Start

Event start

Wait for the road_write signal, or the last_road signal indicating no roads

Road_write

Wait for road_write to go low

Hits_busy = 1

Road_event count.

Next

Decide

First_load

Write the road data into comparator. Increment count

This is axial centroid.

Comp_read

Read the comparators output

What is the data type

Data type = 10

Data type = 11

This is z- axis centroid

Centroid_write

Write the centroid in the required format into the z- axis FIFO.

Init

Last_road

Last_road

Is Fifo_empty

No

Yes

Read the centroid. Hits_busy = 1 Road_event count.

A

B

C

D
A.5 HIT FILTER – Hit filter control logic (continued)

- Hitreg_read
  - Read the masked output
  - Wait
- Is done = 1
  - Yes: Wait in this state
  - No:
    - fifo_empty
    - fifo_empty and not end of event
    - fifo not empty
- Data_wait
  - Is fifo_empty
    - Yes: G
    - No: A
A.6 HIT FILTER - Hit filter hit_format logic

1. Latch the hitreg.
2. Latch the grouped signals

Point_select

Select the starting point to read the hit register

Ready

Are hits_present

Yes

Read the hits from the starting point

select the bit from the register the pointed out by the counter

No

Set the done bit

Read_bits

Is bit high

Yes

Write_hit

No

Next_hit
Write_hit
Format the data into 32 bit word

Output_hit
Write the word to L3 buffer and output FIFO

Next_hit
Is counter \leq \text{total count}

Yes
Increment the counter.

No
Centroid done. Reset Counter

Is it end of event

Yes
Write the trailer to L3 buffer and the output FIFO.

No
Go to Init and wait for the hit register for the next centroid.

NOTE: Total count here is the number of roads / comparators loaded. This the upper limit for the hit search.