STC Data FORMAT

Alexandre Zabi

July 15th, 2002

Updated and Expanded: by Kevin Black on April 26, 2004

Abstract

This paper gives the output format of the STC. An example is given with one hypothetical data event. The stc output format can be found in [5]. This document repeats what is found in [5] and gives a simple sample example of the stc output.Additional information on the format can be found in the specification for one channel [1] and the control logic [2].

1 Hypothetical Data and Parameters

1.1 SMT data

The SMT data sent by the sequencer are sent over an optical fiber serial links using the HP G-link protocol, into receiver cards (VME Transition Module) or VTM located in the rear card cage of the VME64/VIPA crate that house the STT electronics. On each optical fiber, the data from two detectors is transmitted. The format is shown in the table below.

20 ··· 16	15 ··· 8	7 ··· 0	Bits
Control Bits	Channel 1	Channel 0	
Cav-Dav-LnkRdy-Error	Sequencer ID	Sequencer ID	
Cav-Dav-LnkRdy-Error	HDI ID	HDI ID	
Cav-Dav-LnkRdy-Error	Chip ID	Chip ID	
Cav-Dav-LnkRdy-Error	0x0	0x0	
Cav-Dav-LnkRdy-Error	Strip	Strip	
Cav-Dav-LnkRdy-Error	ADC Count	ADC Count	
Cav-Dav-LnkRdy-Error	···	···	
Cav-Dav-LnkRdy-Error	···	···	
Cav-Dav-LnkRdy-Error	HDI ID	HDI ID	
Cav-Dav-LnkRdy-Error	Chip ID	Chip ID	
Cav-Dav-LnkRdy-Error	0x0	0x0	
Cav-Dav-LnkRdy-Error	Strip	Strip	
Cav-Dav-LnkRdy-Error	ADC Count	ADC Count	
Cav-Dav-LnkRdy-Error	···	···	
Cav-Dav-LnkRdy-Error	···	···	
Cav-Dav-LnkRdy-Error	0xC0	0xC0	
Cav-Dav-LnkRdy-Error	0xC0	0xC0	

Here, the CAV, DAV, LNKrdy and error flags are control bits indicating the the status of the data transmission. For more detailed information see [1]. If they are in a valid state Dav (data-available is high) while the rest of the bits are low. There are a few important things to note

1. Two consecutive 0xC0's indicate the end of an event from one channel.
2. Bit 3 of the sequencer ID is omitted in the stc output since this bit is not needed to uniquely identify the detector.

3. Only bits 0-3 are used in the hdi id, the other bits represent the sequencer status but are ignored by the stc.
4. The msb of the chip ID is always 1, while the msb of the strip ID is always 0. The chip id range is 0-7 while the strip id is 0-127 (only using 7 bits). Therefore the msb can always be used to differentiate between a hdi id and a strip id.
5. For simplicity, in the above I have shown the two channels in the same fiber completely synchronous. However, that need not, and is generally not the case. If there are fewer strips fired on one detector then another, the two channels will continue to readout in the same format but can be on different types of words, and in general will be.
6. Once one channel has finished reading out all the data for that event, the channel continues to write out a 0xc0 word indicating that there is no more data on this channel.
7. In order to minimize the chances of a transmission error, the ADC counts are are gray encoded.

In order to demonstrate the data format and processing, we illustrate the data format and processing of an extremely small event. The SMT data file used for this example is the following (control bits always set to valid):

```
0x80505 - both channels have seq id 5
0x80002 - channel 1 has hdi id 0, channel 0 has hdi id 2
0x88586 - channel 1 chip 5, channel 0 has chip 6
0x80000 - zero byte
0x82325 - channel 1 strip 0x23, channel 0 strip 0x25
0x80d04 - channel 1 gray encoded ADC 0xd, channel 0 gray encoded ADC 0x4
0x8c0c0 - first  0xc0 on both channels
0x8c0c0 - second 0xc0 on both channels indicating end of event
```

1.2 FRC Data

The FRC data format to the stc is given by:

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
L1_QUAL	L1_QUAL	LTB channel ID.	L1_BX	FRC Header
Format	Object Length	Header Length	Number of Objects	L1 CTT Header 1
L1 TURN	L1 TURN	L1_BX	L2 Data Type	L1 CTT Header 2
L2 Status Bits	L2 Status Bits	Firmware Version (Minor)	Firmware Version Major	L1 CTT Header 3
H-Doublet(0-43) | 1| 0	D | Trigger Sector Address (0-79)	C | Pt Bin | Ext Pt | T | L	Err code | R | RA PSC	Track
...	...	...	...	Other Tracks
Vertical Parity	Vertical Parity	Data Type	L1_BX	L1 CTT Trailer
Pad 0x0	Pad 0x0	Pad 0x0	Pad 0x0	Padding Word
L1_BX	LTB Error Flags	FRC Error Flags	Word Count	FRC Trailer

The FRC data used to get the road information is shown below:

```
0x00000023 - FRC Header
0x21010305 - 1st CTT Header
0xafa423b5 - 2nd CTT Header
0x00000000 - 3rd CTT Header
0x9c4ec615 - 1st Track
```

```
0x9c4eca15 - 2nd Track
0x3e046010 - 3rd Track
0x148b7800 - 4th Track
0x108b7800 - 5th Track
0x1f70b523 - CTT Trailer
0x00000000 - Pad word
0x00000000 - Pad word
0x00000000 - Pad word
0x2300000e - FRC Trailer
```

Note that the protocol of the L1CTT is that all event records must have a number of words which is a multiple of 4. Hence, the extra three padding words The roads obtained from the road LUT have the upper and the lower limits as the maximum and minimum possible values. All the centroids generated are hits in these conditions. Of course, in the actual hardware we use roads that are propagated into the silicon geometry that have a width of $/pm$ 2 mm. For simplicity we assume all centroids are within the tracks.

1.3 Parameters

- DATA-TYPE: All chips are axial.
- Clustering Thresholds:
  - Threshold1 = 1
  - Threshold2 = 2

  The same thresholds are used for the three types of strips: axial, stereo and z.
- Pulse-Area Thresholds: 1,2,3,4,5,6 and 7. The same thresholds are also used for the three types of strips.
- For the SMT-ID 0: SEQ = 05 and HDI = 2 (STC channels are identified with a 3-bit SMT-ID from 0 to 7, and can be exchanged with STC channel).
- For SMT-ID 1: SEQ = 05 and HDI = 0.
- GAIN-OFFSET: gain = 1 and offset = 0.
- BAD-CHANNEL: all strips are OK.

2 PCI2 Output Format

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
11000000	LRB Error	00000000	L1_BX	Header
tttttttee	eqqqqqqq	hhhccccs	strip	Hit
tttttttee	eqqqqqqq	hhhccccs	strip	Hit
....	....	....	.....	Hit
1111smln	00000000	cccccccc	L1_BX	Trailer

where:

1. t: track number [5:0], starting from 1

2. e: dE/dX[2:0]
3. q: seq[7-4][2:0]
4. h: hdi [2:0]
5. c: chip [3:0], in hit word
6. strip [8:0], note 2 lsb are calculated fractional strip from centroid calculation, s in lsb of byte 1 stands for strip in Hit word
7. s: smt data error bit set,*in trailer word* (not to be confused with s in the header data word which stands for msb in the strip)
8. m: seq/hdi mismatch
9. n: event number mismatch
10. c: number of hits, *in trailer word* (not to be confused with the c in the hit word which stands for chip id.)

2.1 Hits

For event number 23:
(The order of the hits may be different)

```
0xC0000023  - stc header
0x07850A8C  - hit
0x0b850A8C  - hit
0x0f850A8C  - hit
0x13850A8C  - hit
0x17850A8C  - hit
0x07854C94  - hit
0x0b854C94  - hit
0x0f854C94  - hit
0x13854C94  - hit
0x17854C94  - hit
0xF0000a23  - stc trailer
```

2.2 Z-hits

The STC also has the capacity to output hits which are clustered in the z-direction or along the beam line. This was initially for a proposed z-fitter card which would fit z-position of the fit, however this card will not be built. The format is essentially the same as the axial hits.

3 L3 output

The STC l3 output is composed of an overall header, 8 status words, several different types of l3 blocks, a trailer word and a checksum. The L3 words that are available for readout are controlled via a register in the stc they are: Road Data, Hit Data, Z-Stereo Centroid Data, Axial Centroid Data, Cluster Data, Raw and Corrected Data, and Bad Data. Note that because of a limitation of the amount of data that can be read out and stored, only 1 of the 8 input channels l3 data is readout on any event. The hits and roads are global quantities so they are always read out completely, but the rest of the data blocks only come from one channel per event. In order to be able to get status for each of the channels a rotate bit can be set which will rotate which stc channel fills the l3 blocks every l1 accept.

3.1 Overall Header

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
0000wwww	wwwwwwww	ssssssss	L1_BX	Overall Header

here:

1. w: total word count [11:0] including everything
2. s: Source id from bits[15:8] of register 0x28014 (note that this gets written over by the buffer controllers source id, set by dip switch.

3.2 Status Word Format

The Status words give the status of the eight input channels from the smt the format is the following:

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
fdiiiqqq	qqqqqhhh	mescbxxx	xxxxxyyy	Status

where:

1. f: FRC data came after timeout error
2. d: SMT data come after timeout error
3. i: smt_id[2:0] (stc channel)
4. q: seq[7:0] downloaded
5. h: hdi[2:0] downloaded
6. x: seq[7:0] received
7. y: seq[2:0] received
8. m: multiple smt data streams received
9. e: missing eof (0xc0c0 missing)
10. s: seq_hdi mismatch
11. c: chip id error
12. b: bit error in smt data

3.2.1 Status words

here we did not specify the other channels, i will assume that the data received from the smt is from the first two channels (id 0 and 1) and fill the rest of the channels with dummy numbers:

```
0x2a002a   -- status word for chan 0
0x8280028  -- status word for chan 1
0x10290029 -- status word for chan 2
0x182a002a -- status word for chan 3
0x202b002b -- status word for chan 4
0x282c002c -- status word for chan 5
0x302d002d -- status word for chan 6
0x312e00ee -- status word for chan 7
```

3.3 Road Data Format

The road data in the L3 is essentially whatever is received from the FRC minus the padding words.

The format is

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
11100000	wwwwwwww	00000000	L1_Bx	L3 Road Block Header
Whatever	is received	from the	FRC	Data
11110000	mrrrrrrr	00000000	L1_Bx	Trailer

where:

1. w: [7:0] word count including header and trailer
2. r: LRBerr[7:0]
3. m: FRC data length error

3.3.1 Road Data

```
0xe00d0023 - L3 Road Block Header
0x00000023 - FRC Header
0x21010305 - 1st CTT Header
0xafa423b5 - 2nd CTT Header
0x00000000 - 3rd CTT Header
0x9c4ec615 - 1st Track
0x9c4eca15 - 2nd Track
0x3e046010 - 3rd Track
0x148b7800 - 4th Track
0x108b7800 - 5th Track
0x1f70b523 - CTT Trailer
0x2300000e - FRC Trailer
0xF0000023 - L3 Road Block Trailer
```

3.4 L3 hits

The L3 hit format is identical to the l2 output format

For event number 23:
(The order of the hits may be different)

```
0xC0000023 - STC L3 Hit Block Header
0x07850A8C - Hit
0x0b850A8C - Hit
0x0f850A8C - Hit
0x13850A8C - Hit
0x17850A8C - Hit
0x07854C94 - Hit
0x0b854C94 - Hit
0x0f854C94 - Hit
0x13854C94 - Hit
0x17854C94 - Hit
0xF0000a23 - STC L3 Hit Block Trailer
```

3.5 L3 Z and Stereo Centroids Format

The format is

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
10110000	wwwwwwww	00000iii	L1_Bx	L3 Z and Stereo Centroid Header
0tt000ee	eqqqqqqq	hhhccccs	ssssssss	Data
1111smln	LRB Error Flags	Z-Stereo Centroid Count	L1_Bx	Trailer

where:

- t: data type [1:0] “11” z, “01” stereo
- e: dE/dX [2:0]
- q: seq[7:4][2:0]
- h: hdi [2:0]
- c: chip[3:0]
- s: strip[8:0] note that the lowest two bits of the strip are the calculated “fractional strip”, the result of the centroid calculation

3.5.1 L3 Z and Stereo Data

For event number 23:
(recall the data is assumed to be from axial strips)

```
0xB0020023 - L3 Z and Stereo Data Header
0XF0000023 - L3 Z and Stereo Data Trailer
```

3.6 L3 Axial Centroid Data Format

The format is

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
10100000	wwwwwwww	00000iii	L1_Bx	L3 Axial Centroid Header
001000ee	eqqqqqqq	hhhccccs	ssssssss	Data
1111smln	LRB Error Flags	Z-Stereo Centroid Count	L1_Bx	Trailer

where:

- e: dE/dX [2:0]
- q: seq[7:4][2:0]
- h: hdi [2:0]
- c: chip[3:0]
- s: strip[8:0] note that the lowest two bits of the strip are the calculated "fractional strip", the result of the centroid calculation

3.6.1 L3 Axial Data

In this case, the centroid calculations are trivial since in both channels there is only one strip fired. Recall however, that except for the hits and roads, the l3 data is only read out for one channel per event. In this case, channel 0.

```
0XA0030023 -- L3 Axial Data header
0x83854c94 -- Axial Cluster from STC channel 0
0xf0000123 -- L3 Axial Data trailer
```

4 Cluster Data Format

The format is

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
10010000	wwwwwwww	00000iii	L1_Bx	L3 Cluster Header
0ttbllll	llllhhhh	hhhhrrrr	rrrrrrr1	Data_Cluster
0tt00000	vvvvvvvv	000ccccs	ssssss00	Data_Strip
1111sm00	00000000	00000000	L1_Bx	Trailer

where:

- w: cluster data word count including header and trailer [10:0]
- i: smt id (stc channel)
- t: data type [1:0]
- b: '1' three strip centroid, '0' five strip centroid
- l: cluster threshold 1 [7:0] , after pedestal subtraction the minimum value of a strip ADC required to be part of a cluster
- h: cluster threshold 2 [7:0], after pedestal subtraction the minimum value of the maximum value in a cluster, in order to be considered a real cluster (to further suppress noise clusters)

- r: strip count [10:0], number of strips included in this cluster
- v: ADC count, after pedestal subtraction of a strip in the cluster in binary [7:0]
- c: chip[3:0]
- s: strip[6:0] the strip number (*in the data strip*)
- s: *in the trailer* smt data error bit set
- m: seq/hdi mismatch

again, because of the simplicity of the hypothetical smt data, the cluster data shown here is quite small. In general, format is:
L3 cluster header
data_cluster: one for each cluster found
data_strip : for each strip in the cluster
L3 cluster trailer

4.0.2 Cluster Data

For this event the data would be

```
0x90010023 -- L3 Cluster Data Header
0x50102003 -- Data Cluster
0x50070C94 -- Data Strip
0xF0000023 -- L3 Cluster Data Trailer
```

4.1 Raw and Corrected Data

The format is:

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
10000www	wwwwwwww	00000iii	L1_Bx	L3 Raw Corrected Data Header
01100000	0sssklmn	eeeeeeee	oooooooo	Seq_HDI Data
010v00tt	cccckl mn	eeeeeeee	oooooooo	Chip_ID Data
00v0cccc	ccccklmn	eeeeeeee	oooooooo	Strip_ID Data
1111sm00	00000000	00000000	L1_Bx	Trailer

where:

- w: raw/corrected data word count [10:0] including header and trailer
- i: smt id (stc channel) [2:0]
- k: cav_dav_lnk ok for even byte
- l: cav_dav_lnk ok for odd byte
- m: even byte error bit set
- n: odd byte error bit set
- e: even byte (seq, chip, strip)

- o: odd byte (hdi, 0 byte, ADC value)
- c: corrected data

Note that in the raw data the smt data is left Gray encoded, however in the corrected data the data is the ADC count after pedestal subtraction and gain correction in binary.

4.1.1 Raw Corrected Data

For this example the data would be

```
0x80000523  -- L3 Raw and Corrected Data header
0x600C0502  -- Seq_HDI data
0x426C0600  -- Chip_ID data
0x007C2504  -- Strip, Raw and Corrected data
0XF0000023  -- L3 Raw and Corrected Data trailer
```

again, because of the simplicity of the hypothetical smt data, the cluster data shown here is quite small. In general, format is:
L3 Raw and Corrected Data
seq_hdi: One per event, since the stc only reads out one channel of raw and corrected data per event
chip_id : One for each chip the stc received data from, up to 9 for 9 chip detectors that had data from each chip
strip_data: For each strip on each chip, the strip and raw data received, and the corrected data L3 Raw and Corrected Data

4.2 Bad Data

The format is:

31 ··· 24	23 ··· 16	15 ··· 8	7 ··· 0	Bits
11010000	wwwwwwww	00000iii	00000000	Bad Data Header
00000000	0ccccggg	ssssssss	ssssssss	Data
11110000	00000000	00000000	00000000	Bad Data Trailer

where:

- w: bad data word count [7:0] including header and trailer
- i: smt id (stc channel) [2:0]
- c: chip[3:0]
- g: group[2:0]
- s: strip[15:0]

Bad data has a header and trailer for each smt channel. So if the rotate bit is not set, there will be 8 bad data blocks each having its own header and trailer. The "Bad Data" block is a listing of every chip and strip each bit marking if it has been downloaded as a bad strip or not. For the sake of brevity, I will not show all the strips marked as good here.

Following the l3 data is an total l3 word count, followed by a checksum word required by the buffer controller.

REFERENCES:

[1] http://www-ese.fnal.gov/SVX/Production/SVX_Web/VTM/VTM_Spec.PDF
[2] http://ohm.bu.edu/~hazen/my_d0/STC/channel_10Nov00.pdf
[3] http://ohm.bu.edu/~hazen/my_d0/STC/control_10Nov00.pdf
[4] http://www.nevis.columbia.edu/~evans/stt/docs/TR_Data.pdf
[5] http://physics.bu.edu/~wusx/download/stc_doc/L3formatn.doc