

John T. Anderson Engineering Note

Date: October 15, 1999
Rev Date: ~~October 15, 1999~~ **November 2, 1999**

Project: CFT
Doc. No: A991015A

Subject: SVX and Virtual SVX operation in the Analog Front End Board

Introduction

A recent deluge of questions regarding the Virtual SVX impels me to write this note describing, in as painful detail as possible, exactly how the VSVX is supposed to work. Of course, since most of the VSVX is implemented in programmable logic, some change is allowable and probably inevitable; none the less, this document describes the design's intent and current implementation.

Following discussions with Kin Yip, Andre Turcot and Mike Fortner, the BIFIFO readout has been modified from the previous issue of this note. See areas with change bars.

System Overview

In the standard Central Fiber Tracker Axial Analog Front End board with eight multi-chip modules (hereafter referred to by the acronym 8-MCM AFE), and in the Central Fiber Tracker Preshower/Stereo Analog Front End board (12-MCM AFE), every input channel is sampled by a discriminator (the SIFT, ~~and no, I don't know what that stands for!~~ or **SIFT¹ Trigger¹ chip**) and an eight-bit ADC, the SVX IIe (from Silicon VerteX). The discriminator patterns are analyzed every crossing and those indicative of an interesting event are further analyzed by enabling the digitization and readout of the SVX chips as well. An analog pipeline in the SVX chip holds the charge data for up to 32 crossings of delay, to allow the trigger system time to reach its decision. Only upon acceptance of an event does the SVX actually digitize and convert the charge into values. If a trigger is not issued within 32 crossings of an event, the charge is lost and replaced by charge collected in a new event.

To ensure correct correlation of data, the discriminator patterns are stored in FIFO buffers on the AFE and, should a SVX readout ensue, the discriminator pattern which caused the SVX readout is appended to the SVX data as a 'fake' or 'virtual' SVX chip. Thus, the 8-MCM AFE appears to have nine SVX chips, not eight. To minimize readout delay, the SVX chips on the 12-MCM AFE are split into two readout streams. The first stream has four 'real' and one 'virtual' SVX for a total of five, and the second stream has eight 'real' and one 'virtual' SVX for a total of nine.

SVX Data Format and Initialization

The data provided by a 'real' SVX chip is byte-oriented. Readout of a chip starts when it receives a token (PRIORITY_IN) and a new data value is asserted onto the data bus with each rising edge of the SVX clock. The chip continues to assert data until it is empty, at which point the chip asserts PRIORITY_OUT, which is presumably the PRIORITY_IN of the next chip in sequence. When the entire chain is done the last PRIORITY_OUT goes back to the readout controller, signifying that all data has been sent. Every data word is associated with a transition in the DVALID line. This line is used by the readout controller as the strobe to latch the data.

A fixed format is utilized by the SVX chip in its data as shown in Table 1. The first byte is a CHIP_ID value which uniquely identifies which chip in the chain this data comes from. The second byte is a STATUS value which indicates some internal chip parameters. Following the CHIP ID/STATUS pair, the data is asserted as a CHANNEL (or ADDRESS) byte followed by a DATA byte.

¹ Acronym definition thanks to Fred Borcharding.

Index	Interpretation
1	CHIP_ID
2	STATUS
3	CHANNEL #
4	DATA
5	CHANNEL #
6	DATA
7...n	Same pattern

Table 1

The SVX chip can operate in different readout modes. In READ_ALL mode, every channel of the chip reads out regardless of data value. If the READ_ALL is not set the chip operates in a zero suppression mode where only those channels above a programmed threshold show up. Another mode called READ_NEIGHBOR exists which, during zero suppressed readout, also enables those channels adjacent in channel number to those channels who are above the suppression threshold.

Note that because the MCM on the AFE uses only some of the channels in the SVX – only 72 of the 128 channels are actually connected, and they’re not adjacent – the READ_NEIGHBOR bit is essentially useless in the AFE.

Setup of the SVX chip is done through a serial procedure where 190 bits of configuration data are downloaded to each chip in turn. A group of SVX chips looks like one big shift register of N*190 bits, where ‘N’ is the number of SVX chips in the chain. The bits are identified in Table 2, taken from the SVX IIe “Beginner’s Guide” available on the Internet at <http://d0server1.fnal.gov/projects/silicon/www/svx2e/svxe.html>.

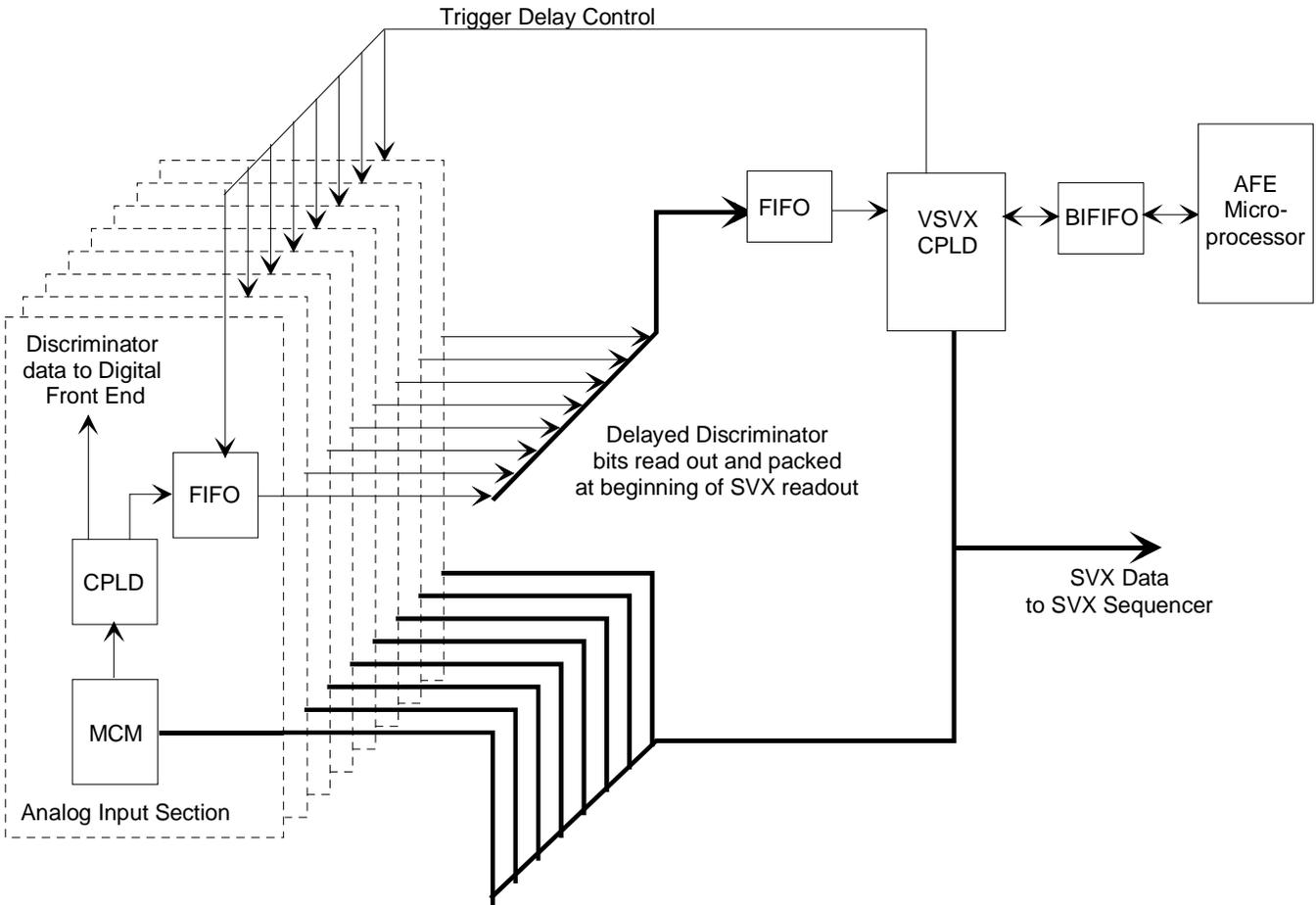
Bit(s)	Function
183-190	Counter Modulo sets the stop value for the A/D Gray Code counter.(bit 183 = MSB, bit 190= LSB)
175-182	Threshold set in digital comparator (bit 175 = MSB)
164-174	Ramp Trim adjusts the ramp capacitor value over a wide range by adding binary weighted capacitors (bit 164= largest capacitor, bit 174= smallest capacitor)
163	Sets flag for last chip in the chain
162	Read Neighbor bit (1 = Read nearest neighbors)
161	Read All Channels bit (1 = Read all Channels)
160	Comparator polarity bit (use 1 for ramp up)
156-159	Fine adjustment of ADC pedestal value
155	Selects polarity of A/D ramp (1 = up, 0 = down)
152-154	Sets value of internal calibration voltage
149-151	Sets scale of chip autobias currents
144-148	Binary Pipeline Depth number to set analog storage delay from 0 to 31 samples (bit 144 = MSB, bit 148 = LSB)
137-143	Binary coded chip ID number. (bit 137 = MSB, bit 143 = LSB)
131-136	Adjusts preamp bandwidth for different detector capacitances and interaction time. Bits add binary weighted capacitors to the preamp. (bit 131 = 1 adds smallest cap)
130	Pipeline-Select adds an offset voltage to the pipeline amplifier reset point maximum dynamic range with bidirectional inputs.

	(1 = negative detector input current, 0 = positive detector input current)
129	Select the polarity of the input test pulse. (1 selects a positive input charge equal to $(AVDD-CAL)*Ct$, 0 selects a negative input charge equal to $(CAL-AGND)*Ct$ is the test input capacitor)
1-128	Test input mask, bit 1 controls test input for channel 1 at top of chip, (1 = test, 0 = no test input)

Table 2

Block Diagram of Virtual SVX Implementation on the AFE

Figure 1 shows a sketch of how the Virtual SVX is connected to the other portions of the Analog Front End board. The sketch obviously ignores much of the circuitry on the AFE in order to more clearly show the functioning of the Virtual SVX.



8 Input Sections
(in 8-MCM AFE, eight * 64 = 512 bits)
(In 12-MCM AFE, eight * 64 = 512 or eight * 128 = 1024 bits)

Figure 1

The Virtual SVX has the basic function of capturing the discriminator data from the MCMs and delaying it in the FIFOs in each Analog Input Section such that the data available at the output of each of those FIFOs is synchronized to the pipeline delay in the real SVX chips in the MCMs themselves. The trigger decision typically occurs some 3.8 usec (29 crossings) after the event actually occurs. The SVX has a 32-crossing-deep analog pipeline, and so the decision must occur

before the analog data falls off the end. To match the SVX analog pipeline depth, each Analog Input Section implements a FIFO which can store 72 bits * 32 crossings = 288 bytes of data. Actually, in the 8-MCM AFE, no MCM has more than 64 active bits, so a 256-element FIFO may be used to save a little cash.

The CPLD (Complex Programmable Logic Device) has the job of determining when the SVX chip has been reset and controlling the event delay FIFOs appropriately. The event delay FIFOs are reset at board power-up and whenever the SVX is read out. The CPLD monitors the system for these two conditions and, following each, for when the SVX chip is placed back into the acquire mode. The event delay FIFOs are re-enabled when the SVX goes back into acquire mode such that the FIFOs are buffering at the same time that the internal analog pipeline of the SVX is buffering. A counter within the CPLD counts the number of crossings after the FIFO has been re-enabled. When this count reaches the pipeline depth that was programmed into the SVX chips during initialization, the event delay FIFO read clock is started, so that the data at the output pins of the FIFOs always corresponds to the event which will be read out of the SVX should a trigger occur.

Of course, should a trigger erroneously occur before the SVX pipeline depth is reached, the SVX chips won't have data but the Virtual SVX will. In this situation the Virtual SVX will simply return its Chip ID and a Status word indicating that the trigger occurred at an incorrect time, and no SIFT data will be read out.

The BIFIFO (an acronym for BI-directional FIFO) provides two-way communication between the CPLD and the AFE board's local microprocessor. In normal operation the BIFIFO will contain board status and event environment information loaded by the microprocessor, such as reference voltage settings, operational mode (e.g. 132 ns vs. 396 ns) or run ID data previously stored into the AFE from the 1553 bus. The Virtual SVX may append this status data to the discriminator data in order to form a more complete event record. However, since most of this information should already be available to the analysis through the slow monitoring, the user will have the option to minimize the readout time by skipping this data.

Alternatively, the BIFIFO may be used by the Virtual SVX to store the data from the discriminators, for readout by the microprocessor. This mode is intended to provide a diagnostic path to allow the bench test software access to the SIFT data without requiring the LVDS links to the Digital Front End board. Since the SVX data bus is bidirectional as well, this functionality also allows a slow diagnostic path by which the SVX data may be monitored, possibly useful to detect bit errors in transmission to the SVX Sequencer.

Setup and Control of the Virtual SVX

The Virtual SVX is normally controlled by the SVX Sequencer through the initialization string sent to the SVX chips. The initialization data flows through the Virtual SVX on the TNBR and BNBR pins, and the bits of interest to the VSVX may sample those of interest. ***To conserve programmable logic resources, the Virtual SVX does not implement most of the bits in the initialization string. Only those bits listed in Table 3 will be implemented and all other bits will read back zero regardless of the data programmed.***

Bit(s)	Function in 'real' SVX	Function in Virtual SVX
183-190	Counter Modulo sets the stop value for the A/D Gray Code counter.(bit 183 = MSB, bit 190= LSB)	Reserved for internal 8-bit register. Virtual SVX Control Register 2, described below.
175-182	Threshold set in digital comparator (bit 175 = MSB)	Reserved for internal 8-bit register. Virtual SVX Control Register 1, described below.
144-148	Binary Pipeline Depth number to set analog storage delay from 0 to 31 samples (bit 144 = MSB, bit 148 = LSB)	Binary Pipeline Depth number to set digital storage delay In Event Delay FIFOs from 0 to 31 crossings (bit 144 = MSB, bit 148 = LSB)
137-143	Binary coded chip ID number. (bit 137 = MSB,)	Binary coded chip ID number. (bit 137 = MSB, bit 143 = LSB)

Table 3

The Virtual SVX appears to the sequencer as just one more SVX chip in the initialization string; for example, in the 8-MCM AFE the SVX Sequencer will actually see (9 * 190) bits. However, the Virtual SVX cannot be loaded with arbitrary data in any of the bits as insufficient data storage exists within the SVX CPLD to implement the full 190 bit register. During subsequent Initialization downloads, the VSVX will simply issue 142 zeroes, real data for bits 137-148, 27 more zeroes, and then the real data for bits 175 through 190. ***External software will have to be aware of this limitation and neither try to load data into those bits nor flag errors if the unused bits don't store the data erroneously written to them.***

The Virtual SVX always assumes that it is the last chip in the chain, as if bit 163 were set.

Internal Registers Implemented in the Virtual SVX

Bits 175 through 190 of the initialization string presented to the Virtual SVX are used to load two internal 8-bit control registers. These registers modify the operation of the Virtual SVX for diagnostic and analysis purposes.

Virtual SVX Control Register 1 (Initialization bits 175 through 182)

Control Register 1 dictates the direction of data flow and the type of data presented by the Virtual SVX. It is a bitmap control register as detailed in Table 4.

Initialization bit position	Bit Name	Function if set (1)	Function if clear (0)
175	ENABLE_STAT ²	Enables readout of board status data from BIFIFO as part of Virtual SVX data.	Disables readout of BIFIFO during SVX readout.
176	ENABLE_SIFT	Enables readout of SIFT discriminator data from Event Delay FIFOs as part of Virtual SVX data.	Disables readout of SIFT discriminator data from Event Delay FIFOs.
177	ZERO_SUPPRESS	Virtual SVX will strip any bytes of value 0x00 from readout of Event Delay FIFOs such that data is compressed.	All data values read from Event Delay FIFOs are passed through to SVX readout, including zeroes.
178	COMPACT_DATA	Virtual SVX will not provide data as address/data pairs (normal SVX format) but instead will only issue data values. A unique data value will be issued with every DVALID, effectively doubling the data rate. <u>Setting this bit disables Zero Suppress, as bit position now determines interpretation.</u>	Readout format will follow SVX normal standard where every data value is preceded by an identifying address value.
179	MONITOR_READOUT	Virtual SVX will capture all SVX data from other SVX chips in string to BIFIFO during next readout, for alternate readout via microprocessor, in addition to normal readout activities.	Data from other SVX chips will not be captured.
180	TEST_PATTERN	Overrides all other bits in this register, forcing Virtual SVX to issue a fixed test pattern of address/data for cable diagnostics.	Readout occurs as controlled by other bits in this register.
181	SKIP_OTHER_CHIPS	The Virtual SVX uses the PRIORITY_IN from the Sequencer as it's PRIORITY_IN, not the PRIORITY_OUT of the previous chip, thus shortening the length of the initialization and readout chain to only one chip – itself.	The chain is allowed to act normally such that the Virtual SVX and all other chips show up as expected.
182	RESERVED	Reserved for future use.	Reserved for future use.

Table 4

A few words are probably required to explain some of these bits. During shakedown of the system it will be useful to have a known source of consistent data to insure that the SVX readout is reliable. By setting the SKIP_OTHER_CHIPS and TEST_PATTERN bits, the AFE board transforms into a one-chip-long SVX string in which the data is exactly the same every readout cycle. This is most useful for exercising the bit error rate of the SVX readout system, and requires no analog input to the board in order to be effective. Once the readout is believed accurate, data from the other SVX chips can be enabled by clearing the SKIP_OTHER_CHIPS bit and, as desired, the TEST_PATTERN bit. During actual system running, should bit errors in a string be suspected, the TEST_PATTERN bit all by itself may be used to bury test data in along with the real data.

² If software fails to set either the ENABLE_STAT or the ENABLE_SIFT bits, the Virtual SVX will only issue a CHIP_ID and a STATUS word, and no other data, unless the TEST_PATTERN bit is set.

Prior to installation in the experiment, the AFE board will be bench tested. A method of accessing SVX data without having to install an SVX Sequencer and associated cabling will simplify the bench test, and so the MONITOR_READOUT bit will allow access to SVX data in the most simplistic setups. This bit can also be used as a slow redundant data path during commissioning if the SVX readout would be adversely affected by the excess data created by the TEST_PATTERN bit.

Since it is already evident that the total amount of SVX data coming from the AFE will be a critical determinant of experiment deadtime, the ENABLE_STAT, ENABLE_SIFT, ZERO_SUPPRESS and COMPACT_DATA bits allow the users to ‘tune’ the amount of data coming from the Virtual SVX. Presumably, the correct selections of these bits will allow for the best compromise between bandwidth and reliability.

Virtual SVX Control Register 2 (Initialization bits 183 through 190)

~~As of this writing, Control Register 2 is reserved for engineering diagnostics. It may, in the future, allow for selection of various test patterns when the TEST_PATTERN bit is set in Control Register 1.~~

Control Register 2 is used by the Virtual SVX to store the Chip ID for the BIFIFO readout. When BIFIFO data is enabled, the Virtual SVX precedes the BIFIFO data with a new Chip ID/Status pair so that readout software may differentiate BIFIFO information from SIFT discriminator information.

Details of Virtual SVX Readout Format

The Virtual SVX follows the same generic readout format as given in Table 1 for the SVX IIE chip. Specific rules are followed:

- The CHIP_ID returned will be the CHIP_ID as loaded in bits 137-143 of the initialization string.
- The STATUS word will be a bitmap status of the Virtual SVX as given in Table 5:

Bit #	Interpretation if set (= 1)	Interpretation if clear (= 0)
7	BIFIFO data enabled (<u>expect a following Chip ID, Status, then the BIFIFO data</u>)	BIFIFO data not enabled
6	Discriminator data enabled	Discriminator data not enabled
5	Data is test pattern	Data is ‘real’ data
4	SKIP_OTHER_CHIPS is set	SKIP_OTHER_CHIPS not set
3	COMPACT_DATA is set	COMPACT_DATA not set
2	ZERO_SUPPRESS is set	ZERO_SUPPRESS not set
1	Pipeline Depth Error (insufficient number of crossings in Event Delay FIFOs)	Pipeline Depth OK
0	Reserved	Reserved

Table 5

Some explanation of bit 1 is probably in order. Once a readout has occurred, the SVX chips must all be reset and the pipelines restarted. If a trigger error causes a Level 1 Accept to occur too quickly after this reset, the pipeline depths as programmed in bits 144-148 of the Initialization string will not have been met. This means that the real SVX charge data will be wrong, and also means that the discriminator bits available to the Virtual SVX are also wrong. The trigger latency should insure this never happens, but it’s always good to check.

Bits 2 through 7 of the Status byte identify the Virtual SVX mode of operation, as was programmed via the Initialization string. These bits are used to allow the later data analysis to correctly decode the rest of the data. Bit zero is reserved for future use.

- The ADDRESS portion of the two-byte address/data pair will indicate the source of the data (whether BIFIFO or discriminator data, and if discriminator data, from which MCM, i.e., which MCM and which byte from that MCM) according to Table 6. The general algorithm used is that data from the Event Delay FIFOs will have an ADDRESS field which is, in binary, 0mmmddd where the mmm is a three bit value indicative of which MCM on the board sourced this data (MCM #0- #7) and the dddd field indicates which byte number (0-8) of the bytes of data from that MCM this comes from.

Each MCM may possibly source as many as 72 bits of discriminator data, which would require 9 bytes of data per MCM. The ADDRESS field algorithm above thus reserves 72 of the possible 128 addresses for MCM data in eight blocks, leaving 56 addresses in eight other blocks for BIFIFO or other data unused. Although nothing physically prevents the Virtual SVX from using addresses 128 through 255, the SVX convention is that the address byte never has the most significant bit set. Some readout software may use this for validity checking and so the Virtual SVX conforms to this convention, *except when the COMPRESS_DATA bit has been set.*

This convention thus limits the BIFIFO status data to no more than 7 bytes. If the readout software does not use the most significant bit of the ADDRESS for validity checking, then the preferred solution would be to use address locations 128 through 255 for the BIFIFO data. This will be determined as software is finalized.

Address Range	Data Source
0x00 – 0x08	MCM #0, data bytes 0 through 8.
0x09 – 0x0F	Reserved addresses
0x10 – 0x18	MCM #1, data bytes 0 through 8.
0x19 – 0x1F	Reserved addresses
0x20 – 0x28q	MCM #2, data bytes 0 through 8.
0x29 – 0x2F	Reserved addresses
0x30 – 0x38	MCM #3, data bytes 0 through 8.
0x39 – 0x3F	Reserved addresses
0x40 – 0x48	MCM #4, data bytes 0 through 8.
0x49 – 0x4F	Reserved addresses
0x50 – 0x58	MCM #5, data bytes 0 through 8.
0x59 – 0x5F	Reserved addresses
0x60 – 0x68	MCM #6, data bytes 0 through 8.
0x69 – 0x6F	Reserved addresses
0x70 – 0x78	MCM #7, data bytes 0 through 8.
0x79 – 0x7F	BIFIFO data (board status, etc.). Reserved addresses

Table 6

- The DATA field has arbitrary meaning for the BIFIFO data and is dependent upon the information loaded there by the microprocessor. This portion of the information will be covered in a separate note. For discriminator data, eEach bit of the DATA field corresponds to one discriminator, and thus, to one SVX channel. Other CPLDs will actually pre-format the data into the Event Delay FIFOs, and the order in which the bits correlate to SVX channels are controlled by their programming. Two different CPLDs are used for the resorting per MCM. Because of routing considerations, each CPLD ‘sees’ half of the MCM’s discriminators. One CPLD drives bits 0-3 of the Event Delay FIFO, whereas the other drives bits 4-7. This split results in the channel mapping shown in Table 7, where each bit of each byte is correlated to the SVX channel to which it belongs.

Byte	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
0	Channel 71	Channel 68	Channel 67	Channel 66	Channel 7	Channel 6	Channel 5	Channel 2
1	Channel 77	Channel 74	Channel 73	Channel 72	Channel 13	Channel 12	Channel 11	Channel 10
2	Channel 83	Channel 82	Channel 81	Channel 78	Channel 21	Channel 20	Channel 17	Channel 16
3	Channel 91	Channel 88	Channel 87	Channel 84	Channel 27	Channel 26	Channel 23	Channel 22
4	Channel 98	Channel 97	Channel 94	Channel 92	Channel 36	Channel 32	Channel 29	Channel 28
5	Channel 104	Channel 103	Channel 102	Channel 99	Channel 42	Channel 41	Channel 38	Channel 37
6	Channel 110	Channel 109	Channel 108	Channel 107	Channel 48	Channel 47	Channel 46	Channel 43
7	Channel 118	Channel 117	Channel 114	Channel 113	Channel 56	Channel 53	Channel 52	Channel 49
8	Channel 124	Channel 123	Channel 120	Channel 119	Channel 62	Channel 59	Channel 58	Channel 57

Table 7

As an example, if the Virtual SVX spits out address 0x63 with the data value 0xA8, this means that SVX channels 91, 87 and 27 of MCM #6 had their corresponding discriminator bits set.

BIFIFO Status Data

The format of BIFIFO data is entirely controlled by what the microprocessor loads into the BIFIFO for readout. As microprocessor firmware for the AFE is finalized, a separate note describing the BIFIFO data will be issued. However, general observations are available now:

- BIFIFO data will be preceded by a CHIP ID byte whose value is determined by the secondary Chip ID value loaded in the Initialization string.
- BIFIFO data will be preceded by the STATUS byte whose value will be zero, except in the collision condition where an SVX readout occurs while the microprocessor is updating the BIFIFO information. In that case, the STATUS byte shall be nonzero and indicate what portion, if any, of the BIFIFO data is valid.
- There will never be more than 256 bytes of BIFIFO data, and usually far fewer than that.
- In general, BIFIFO data will *not* follow the ADDRESS/DATA pair convention, but shall be all DATA.
- The length of the BIFIFO data is variable, but will change very infrequently. Changes in the length of the BIFIFO data can be synchronized to major startup/initialization sequences, such that BIFIFO data length should be consistent throughout a given run.

Deadtime Analysis

Readout of the SVX data in the CFT system is a significant contributor to experiment deadtime. To insure that the Virtual SVX doesn't cause a big problem here, the deadtime contribution of the Virtual SVX in various configurations need be explored.

'Normal' mode, no zero suppression or data compaction enabled

In this mode, the Virtual SVX simply grabs the data from the Event Delay FIFOs and spits it out unprocessed to the SVX Sequencer during readout. The Virtual SVX learns that the event has been accepted by either sensing the L1_ACCEPT signal from the SVX Sequencer or by seeing the SVX mode change from Acquire to Digitize. Either way, the Virtual SVX will have a leadtime equal to the time it takes for the SVX chips to digitize, plus the readout time of the SVX chips prior to the Virtual SVX in the chain.

A reasonable estimate for this lead time is to assume that the SVX will be allowed half of the maximum number of counts for digitization, about 1 usec for FIFO collapse, and that each of four SVX chips prior to the Virtual SVX will read out two bytes only (CHIP_ID and STATUS), plus one more clock tick for passing the token. Thus, the conservatively estimated lead time is

$$T_{lead} = (127 * 18.9ns) + 1000ns + [4 * (3 * 18.9ns)] = 3627ns$$

If one assumes that the Virtual SVX runs at 53 MHz and it takes two clock ticks to fetch each byte from the Event Delay FIFOs, this allows the Virtual SVX to prefetch approximately 95 bytes before it can ever see the token to start readout. The densest string of Event Delay information is found in the 8-MCM string of the 12-MCM AFE, where the Virtual SVX must read out 9 bytes from each of 8 MCMs, or 72 bytes. Thus, it would appear that the Virtual SVX will always be ready before it can get the token, and does not contribute anything more than its data readout time to the deadline.

Zero Suppression and/or Compact Data modes

In the Zero Suppression mode the Virtual SVX must completely pre-read the data from the Event Delay FIFOs in order to strip bytes of all zeroes prior to engaging readout. So long as the timing analysis of the above section is valid, this should not be a concern. In the Compact Data mode, the prefetch time is the same, only the actions of the Virtual SVX are different in that the data from the intermediate buffer FIFO is not interspersed with address data from the CPLD during the Readout cycles.