Fermi National Accelerator Laboratory

Conf-911106--46

FNAL/C--91/315

DE92 004809 1/315

Commissioning of the Control and Data Acquisition Electronics for the CDF Silicon Vertex Detector

S. Tkaczyk, K. Turner, C. Nelson, T. Shaw and T. Wesson

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

M. Bailey and M. Kruse

Purdue University
West Lafayette, IN 47907

A. Castro

University of Padova
I-35100 Padova, Italy

November 1991

* Presented at the *IEEE Nuclear Science Symposium*, Santa Fe, New Mexico, November 2-9, 1991.

Commissioning of the Control and Data Acquisition Electronics for the CDF Silicon Vertex Detector

S. M. Tkaczyk, K. J. Turner, C. A. Nelson, T. M. Shaw, T. R. Wesson
Fermi National Accelerator Laboratory*, P. O. Box 500, Batavia, IL 60510
M. W. Bailey, M. C. Kruse
Purdue University, West Lafayette, IN 47907
A. Castro
University of Padova, I-35100 Padova, Italy

Abstract

The SVX data acquisition system includes three components: a Fastbus Sequencer, an SVX Rabbit Crate Controller and a Digitizer. These modules are integrated into the CDF DAQ system and operate the readout chips. The results of the extensive functional tests of the SVX modules are reported. We discuss the stability of the Sequencers, systematic differences between them and methods of synchronization with the Tevatron beam crossings. The Digitizer ADC calibration procedure run on the microsequencer is described. The microsequencer code used for data taking and SVX chip calibration modes is described. Measurements of the SVX data scan time are discussed.

I. INTRODUCTION

We discuss some specific aspects of the operation of the readout components of the Silicon Vertex Detector[1] for the CDF '92 run. The SVX DAQ system handles 46,080 channels divided into 24 sectors. This number represents a 50% increase over the number of CDF channels in the last run. Each sector is read out within a limited time determined by the other CDF subsystems. The hardware modules include a Fastbus Sequencer, an SVX Rabbit Crate Controller and a Digitizer, which have been designed at Fermilab.

Included are the results of various tests used to quantify the performance of the system and the methods by which the DAQ modules have been implemented into the rest of the CDF control system. Event data are temporarily stored in the Sequencer's memory and are then transferred to a Slac Scanner Processor[2] (SSP), a standard interface to the CDF Data Acquisition. Different schemes of reading and buffering the event data will be described along with the scan times of each. The calibration results for the Digitizers will be discussed. A description of other features of the modules which were introduced to enhance the performance of the readout will also be included.

* Operated by the Universities Research Association under Contract with the US Department of Energy.

II. DESCRIPTION AND TESTING OF THE DAQ MODULES

The boards have been developed in phases, and different amounts of time were spent for checkout in each phase. Various tests have been performed using specified test programs. The changes indicated by the tests have been included in the design of the boards before the next iteration was made. While the system was being integrated, new tests were developed and used for the identification of problems.

A detail description of the design of the SVX DAQ readout components is presented in another contribution to this conference[3]. Here only a brief summary, relevant for the commissioning of the system, is presented. Some of the test results of the importance of the operation of the system are described for each of the modules.

A. Sequencer

The Sequencer consists of a 16K data memory, a 16K program memory, a microsequencer chip with peripheral conditional control logic, and a 100 MHz clock. The clock frequency is divided by 8 to create the 80 ns period clock which drives the execution of microinstructions. Each microinstruction has three logic sections; control of the pattern, control of the SVX Rabbit crate and control of microsequencer. The first of them defines timing signals used by the SVX Rev D integrated circuit [4] (SVXD), strobe signals to cause the Digitizer to latch data from the front end into its data registers, and the state of two sets of four signals accessible through front panel Lemo connectors. The timing and latch signals and one set of front panel output signals are timed off a pattern clock which is formed by delaying the main clock. A vernier adjustment to control the length of the delay is provided by 3 bits of this word. These bits and the other set of front panel signals are timed by the main clock. The second section sends data and control signals to the Controller, and is timed by the main clock. The third section controls the order in which the microsequencer executes these instructions; specifying the branch address, the condition code, and the 4-bit microinstruction code. The states of 32 conditions are available for branch control.

The data space is divided into 8 2K blocks. There is an End Of Block (EOB) register associated with each of these blocks. When data words are sent from the front end, the

Sequencer directs the data to the next open location in a particular block based upon 3 bits of the data words, and writes the address to the associated EOB register. The Sequencer can be read out by Fastbus block read of one of the Event Memory blocks. When the EOB content for that block is reached, a Fastbus SS=2 code is generated, and no more data words are sent by the Sequencer.

The earliest stage in testing the Sequencer was to run tests to identify design problems and errors in stuffing the board. To this end, a menu-driven program written in Fortran, called SVX_Control[5], was designed. It allows communication with the Sequencer from VAX through QPI, using CDF Fastbus primitives. Reads, writes, and comparisons to all registers, as well as the interactions among registers can be checked by selecting automated test options. Microsequencer tests can also be selected, along with tests of the front panel indicator lights and verification of restricted or undefined CSR registers. Additionally some of the Sequencer's write data cables could be connected to read data inputs to test the integrity of the front end crate data handling capabilities.

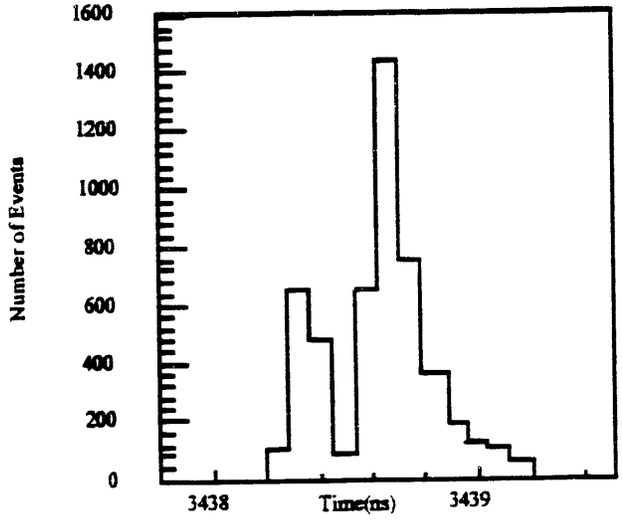

Fig. 1. Main clock stability.

Table 1

Typical delay between main clock and the delayed clock for each vernier setting.

vernier setting	measured delay (ns)	differential delay(ns)
0	30.80	3.71
1	34.51	5.53
2	40.04	4.18
3	44.22	4.88
4	49.10	6.47
5	55.57	3.87
6	59.44	5.16
7	64.60	

After the board was debugged, the next step was to test the timing stability of Sequencer instructions. The time required to execute a fixed number of instructions was measured multiple times to determine the variation. Of particular interest was the 43-instruction timing, since this is the length of the SVXD chip reset and integration cycles, which demand high stability in order that the voltage offset produced by the integration of leakage current can be accurately subtracted. The result of this test is shown in Figure 1.

We also measured the stability of the pattern clock and found that it was as stable as the main clock. This is to be expected, since any instability would be due to the vernier delay chip and was found to be small. However, we still had to measure what the magnitude of the delay between the two clocks was for each of the eight possible settings. These values are listed in Table 1.

After verifying that each of the Sequencers separately had acceptable timing stability, it was necessary to determine the homogeneity of the timing among the four Sequencers plus spares that would be used in the whole system. While the previous test allowed the internal clock to run continuously, this test provided a common start clock signal to all four Sequencers. The difference in the time between the sending of the start signal and the arrival of two different Sequencers at instruction 43 is shown in Figure 2.

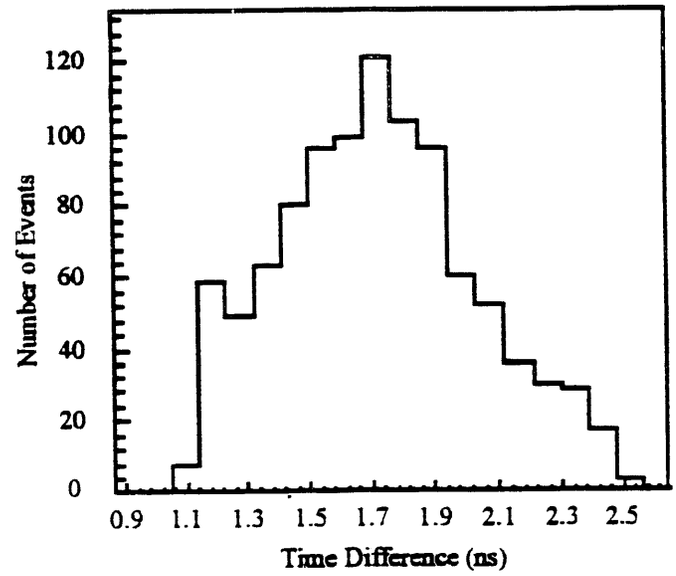

Fig. 2. Time difference for 2 Sequencers to execute 43 instructions.

B. Controller

The Controller translates function codes from the Sequencer and drives the SVXD timing signals onto the SVX Rabbit crate backplane. It can be directed to write or read individual Digitizers, or all Digitizers in the crate. The Priority line coming from each wedge to the Digitizer is sent to the Rabbit crate backplane. While scanning the Digitizers' front end data registers, the Controller interprets this signal as an indication of the presence of data in at least one Digitizer. It also contains registers that can be used to mask Digitizers or hold the data lines in a DC state.

The Controller is responsible for sending two control signals to the Sequencer: Read_Done and Data_Ready. If the

Read_Done signal is low, it indicates that the Controller is reading data from the crate which will soon be sent to the Sequencer. In this condition, the Sequencer will not accept any signals from other elements in the system. While Read_Done is low, the signal Data_Ready may be sent. This is a pulse that latches the data on the lines into the Sequencer's Event Memory. When Read_Done goes high, it increments the Sequencer's Event Memory Address Counter so that the next data word read will go into the next data space location. This also indicates that the Controller is now asserting the priority lines from the detector.

By writing data to the Controller's registers, reading the data back, and checking the data space locations to which the data should have been written, we are able to simultaneously test the functioning of the Controller registers and the timing of the Read_Done and Data_Ready signals. Other features of the Controller, such as its ability to mask off selected Digitizers or hold the data lines in a DC state, are tested in symphony with the Digitizer tests that follow.

C. *Digitizer*

The Digitizer board picks up the SVXD timing signals from the backplane and drives them to the wedge to which it is connected. It contains one 12-bit analog to digital converter (ADC), which accepts signals from the detector in the range between +5V and -5V. It also has three digital to analog converters (DAC). Two of the 12-bit DACs provide charge injection for the detectors. The third DAC is used to subtract a given offset from the analog signal from the detector, which is then sent through a variable gain of 0.625 to 8.125, adjustable in 0.625 increments. Finally, the signal is digitized by the ADC and latched into a data register. Digital signals received from the wedge, such as hit position, are also latched into the data registers. To each data word sent to the Controller, the Digitizer also appends an indication of its position in the crate.

The first Digitizer tests, which also were automated in the SVX_Control environment, were write, read, and comparison of data for all its registers. Registers could be accessed either through Fastbus writes to addresses mapped into the Sequencer's CSR space, or through execution of microsequencer programs using the 32-bit wide front end crate control registers. Function codes sent to the Controller can call for a Digitizer to be addressed individually by slot number, or as part of a scan of all the Digitizers in a crate. All combinations of these features are selectable from the SVX_Control test menu.

The process of understanding the calibration of the SVX readout system began with the tests of the ADC using the voltage generated by the on board DACs. The offset and gain of the converters have been trimmed to the minimal values. The difference between the average digitized value and the DAC setting was checked as a function of the DAC setting. As an input to the ADC we selected each of the three available DACs. All data points lie on straight lines indicating that the ADC and DAC operate in a linear way. The differences between the digitized values and the DAC setting for all three DACs used as input to the ADC were contained in the range between 0 and 1 LSB, indicating that the offset and gain have been correctly selected and the digitized values are equal to the DAC setting within 1 LSB. The average value of the error of the individual settings was found to be less than 0.02 LSB indicating high stability of the converters.

There will be more than 24 Digitizers used in the full readout system. It is very important that all of them operate with a similar precision. Based on a comparison of the measurements of offsets and gains of 5 Digitizers modules we confirmed linearity and stability of the system.

III. System Implementation and Operation

A. *Event Acquisition*

The design of the SVX DAQ system was determined by two major factors: the structure of the SVX detector and the architecture of the existing data acquisition scheme in CDF. The SVX will increase the existing number of channels in the CDF DAQ by about 50%. The SVX detector consists of two identical barrels, which contain four coaxial layers of silicon strip detectors that are read out at one end by the SVXD chips. Each layer is divided into 12 30-degree sectors containing 1920 channels, called wedges. Each wedge is

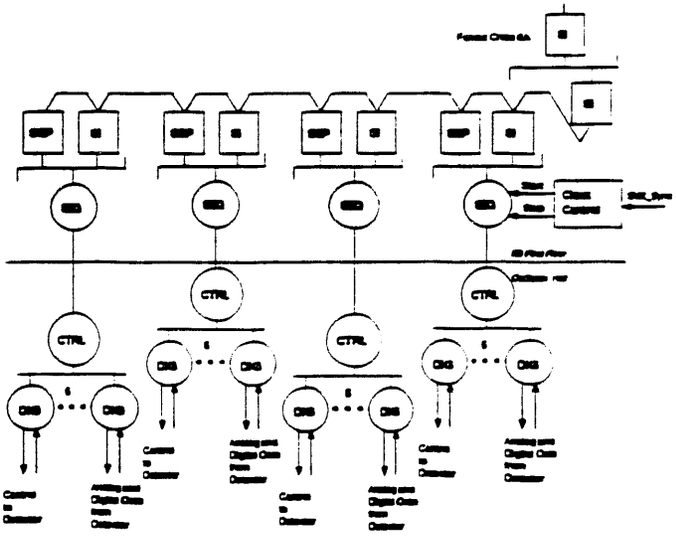

Fig. 3. SVX DAQ System Diagram

connected to a Digitizer module. The SVX data acquisition chain and its integration with the existing CDF DAQ system is shown in Figure 3. The Digitizers are evenly distributed in 4 Rabbit crates mounted on the CDF detector. Each crate has one Controller module which interfaces the front end system with the Fastbus Sequencer module in the counting room. In order to minimize the event readout times each Sequencer module is operated with an accompanying SSP scanner in a separate Fastbus crate. All SVX Fastbus crates are connected to a single cable segment in a Fanout crate. It should be noted

that the SVX data from the next run will not be included in the Level 1 or Level 2 trigger algorithms.

The architecture of the current CDF DAQ acquisition system is described in detail elsewhere[6]. A brief description of the components which are relevant to the integration of a new silicon tracking device will be presented. Most of the existing tracking detectors in CDF use the SSP. The scanners can buffer up to four events in their memories. A Fastbus module called the Trigger Supervisor controls the operation of the scanners. The Event Builder, a custom designed Fastbus module, reads the data from the scanners and sends them into the Level 3 system. The Buffer Manager manages the data flow between the scanners and the online computers.

A process called Run_Control performs event readout, detector calibrations and hardware diagnostics. It controls the data flow and management of hardware devices. Independent "consumer processes" access the data for monitoring, diagnostics or data analysis purposes. The calibration of the subsystems takes place between the runs and is performed in two modes: D_Mode (data taking) or X_Mode (scanner based). The calibration data are stored in the CDF calibration data base.

Upon receiving the Start Scan message from the Trigger Supervisor each scanner begins execution of the code and sets a Done signal on an external connector to a logical false level. The Start Scan message is written into the CSR address 8000000E and carries information such as event number and internal buffer number to which the data are to be written. When data collection is completed Done signal is set to a logical level true. This is a signal that the event is stored in one of the four buffers and the scanner is ready to be read out and to accept the next Start Scan message.

The algorithms for the readout of the SVX Sequencer include data acquisition and data calibration modes. During data acquisition and D_Mode calibration, the SSPs read data from each Sequencer's buffers until an SS=2 code is returned. Then a check is made between the number of data words transferred and the actual number of words written to the Sequencer's block. In case of a mismatch, an error condition is reported to the system. The raw data words are then sent through the DAQ chain and are written to disk.

In X_Mode calibration the raw data words from the Sequencers are sent only to the SSPs. In the Runtype Database the number of triggers and the values of charge injection are stored. Prior to start of the calibration task the Run_Control program downloads these values into the SSP. For each channel the scanner accumulates the sum of ADC and squared ADC counts as well as the number of events read. After collection of the predefined number of triggers, the data accumulated by the SSP are sent the rest of the way up the DAQ chain to be further analyzed by calibration consumer processes. The next set of values is then downloaded to the SSP and the process repeats until all desired settings are exhausted. This method of calibration runs much faster than the D_Mode because it reduces the number of data transfers and uses the computing power of the SSP.

A compact data format was used to describe the digital and digitized analog output from the SVXD chip. Each hit is described by one 32-bit word. The lowest 12 bits are reserved for the ADC information, and 4 bits are not used in the lower half word. In the upper half word 7 bits are used for decoding the channel address, 4 bits are used for the chip identification number and top 5 bits are used for the crate and Digitizer number descriptions. The 16-bit control registers are read from the Digitizer or Controller modules using the same format and the chip id bits all set high. Data from all the Digitizer and Controller registers are read out and appended to the raw data for diagnostics and calibration purposes.

B. Synchronization with the CDF DAQ

The 53MHz CDF Master Clock generates timing signals correlated with the bunch crossing times. The signals are called Clear&Strobe or C&S, Start C&S and Stop C&S. Their edges are timed with the 1 ns resolution relative to the bunch crossing. The Master Clock is synchronized with the Tevatron Marker once per turn. One turn takes 1113 clock cycles or 21 μs. The Tevatron operates with 6 bunches alternately spaced by 185 and 186 clock cycles. The C&S signal is sent to the Gate Selector and the Start C&S and Stop C&S signals are sent to the TS module. From the Gate Selector the C&S signal is sent to the front end devices. At

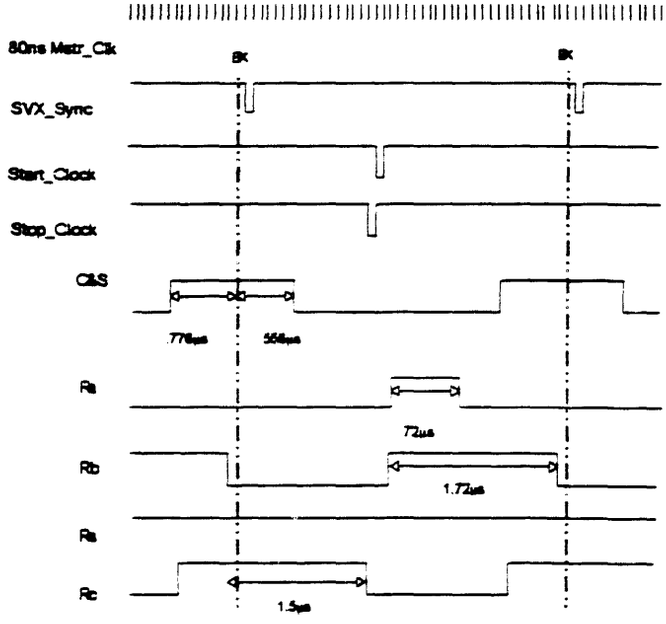

Fig. 4a. Synchronization of the Sequencer's operations with the SVX_Sync signal.

the Gate Selector the C&S signal is gated off by the signal produced by the TS of the Start C&S and the Stop C&S signals. The TS module generates C&S gate signals until a Level 1 Accept signal is received. From this moment the C&S Gates are not generated. Upon arrival of the Level 2 Accept signal a Start Scan message is generated and after Done signals are received from the scanners the C&S Gates are

to be generated at the LRS4222 by control of a Sequencer's signal called Inhibit Clocking during the threshold cycle. When the threshold is restored, the synchronization mechanism is enabled again.

In the next step a test is done of the Level 2 trigger decision by checking the Start Scan Latch signal. Its presence initiates the readout cycle and its absence directs control to the synchronization subprocess. The Start Scan Latch signal is automatically reset when is it checked by the execution of a conditional branch microsequencer instruction. During the readout cycle the synchronization mechanism is disabled again until the cycle is finished. To indicate that readout is finished, the Sequencer sets a particular bit in one of its CSR space registers. Since the SSP continually polls this register after sending the Start Scan message, it will recognize the signal and begin to read out the Sequencer within ~2μs of the bit being set. For non-triggered events, a separate integration cycle is performed in the very next Start_Clock/Stop_Clock cycle.

IV. Integration with the SVXD Chip

The SVXD chip is operated by selection of specific switch timing sequences generated by an external source, eg. the Sequencer. The Sequencer's programs are written in HiLevel Assembly Language Environment (HALE) macro meta-assembler. Separate modules as shown in Fig. 5 were compiled and linked together to form microsequencer's programs. The structure of the code used for an event acquisition, which includes pedestal, charge injection, leakage current monitoring and threshold determination, is described in another contribution to this conference[5].

After verifying that the communication among the Sequencer, Controller, and Digitizers was fully functional, in the next step the Digitizer was connected to a Port Card[1] and a set of SVXD chips. It was confirmed that the timing signals are being driven out the cable and the digital and analog data from the SVXD chips are being latched into the Digitizer registers correctly. It was also verified that charge injection DACs met the specified output range and the Priority logic is handled properly. Once this integration step was satisfactory, a Digitizer was connected to a ladder structure containing the SVXD chips microbonded to silicon detectors. Calibration data were taken and compared to similar data taken with the old Camac system. This verified that the components did not introduce additional noise into the system.

V. Scan Times

One of the very important constraints imposed on the SVX DAQ system is the 2 ms limit on the maximum scan time. The problem is not easy to solve in a system which consists of 46,080 channels. The proposed solution is based on partitioning the SVX channels into four independently read branches (see Fig. 3). The scan time is defined as a sum of two components: the time required to read SVXD chips in six wedges by the Sequencer and the time to move data from the Sequencer's Event Memory to the SSP's buffer. In other words it is the time between the broadcast of the Start Scan message by the TS and the moment the TS receives the Done signal from a scanner.

The Sequencer scan time is proportional to the number of hits multiplied by the sum of the duration of the basic chip data cycle, called HiLo, the response of the front end modules, signal propagation delays and the time required by the Sequencer to check if the scan is completed and all the data are read out from wedges. The Digitizer has two types of data registers where the digitized values are stored. The first register, A0, is called the sequential data register. It is operated in a sequence where each HiLo cycle is followed by a Read Done condition check before the next HiLo sequence is generated. The duration of this type of a readout cycle includes such components as cable delays, ADC conversion time and crate scan time. The second register, A1, is called a pipelined data register. It is operated in a sequence where a HiLo cycle is followed by the crate scan instruction and followed by another HiLo cycle, before the previous scan is completed. The duration of the pipelined readout is determined mainly by the length of the HiLo cycle plus a small overhead associated with checking if the previous scan was finished and all the wedge data were read out.

Measurements of the Sequencer scan times have been made using a 6-chip ear board and a 15-chip wedge structure. The front end components were read out in the pipelined and the normal modes. The HiLo cycle was selected to be equal to 2.08μs when reading the A1 register and 2.24μs for the register A0. The small difference between the duration of the HiLo cycles reflects the optimization of the pipeline readout code, that was not possible to be implemented in the normal readout mode. The total length of the HiLo signal was selected to accommodate the chip response time and the risetimes of signals on the cable. The scan time values that correspond to two methods of reading the front end modules by the Sequencer for different number of channels are presented in Table 2. The difference of the scan times corresponding to the time needed to read out one channel was understood in terms of the 250' cable propagation delays (800ns), ADC conversion cycle (500ns), crate readout time (100ns) and the length of the HiLo cycles. Based on these numbers an estimate of the readout time of a crate with 6 Digitizers was made assuming even occupancy. The values which correspond to 100% and 40% occupancies with 1 Digitizer in a crate are the measured values. The other values of the Sequencer scan times were found by extrapolation. In case of the pipeline readout method the scan times are independent of the number of Digitizers read out. This conclusion is valid only if the HiLo cycle is longer than the sum of the cable delays, ADC conversion and the crate readout times, as will be the case for the upcoming run.

The time required to move data to the SSP as a function of the number of hits can be parametrized by a linear formula. The offset represents the SSP overhead before the data transfers begin and the time required for making a Fastbus connection between the modules, multiplied by number of blocks to be read out. The slope value is a product of the word transfer rate

restored. The C&S signal is used by CDF subsystems for synchronization purposes as well as an indication of the L1 trigger decisions.

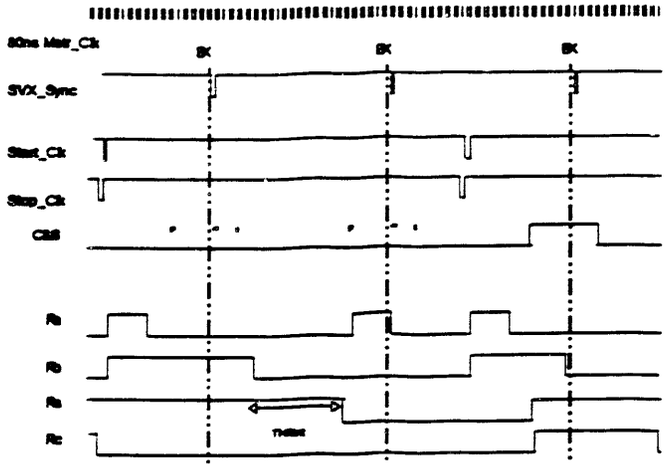

Fig. 4b. Threshold restoration process asynchronous to the beam crossing. The dashed lines show the time periods where the C&S was gated off by the TS.

The operation of the SVXD chip is performed by an ensemble of timing signals generated by the Sequencer module in phase with the bunch crossing times regardless of the uneven spacing of alternate bunches. For this reason the Sequencer has to be synchronized with every beam crossing. The method also minimizes the impact of the jitter of the Sequencer's internal clock on the precise timing of the event and threshold integration times. The existing C&S signal provides this functionality but its timing was tuned to the requirements of the calorimetric subsystems. The operation of the SVXD chip requires longer times prior to the beginning of the integration cycle than the calorimeters. For this reason a new synchronization signal, called SVX_Sync, is generated by the Master Clock. The SVX_Sync is related to the next beam crossing and is generated shortly after the previous crossing.

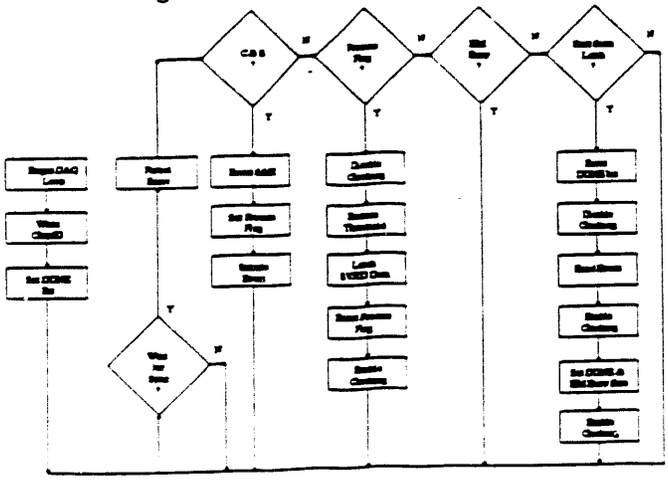

Fig. 5. The SVX DAQ Block Diagram

The process of the Sequencer's synchronization with the beam crossing is performed in the following way (See Figure 4). The signals: Ra, Rb, Rc, Rs control the reset and integration operations in the SVXD chip. The BX vertical lines indicate beam crossing times. The SVX_Sync signal is sent to a LRS4222, a programmable delay module to generate Start_Clock and Stop_Clock signals. These signals control the operation of the Sequencer's internal clock. The relative time difference selected in the LRS4222 is adjusted to guarantee that the same number of instructions are executed between the beam crossings. The moment at which the synchronization process takes place will be adjusted to a specific pattern of the timing signals. It is possible to implement a scheme where all the transitions of the switches on the SVXD chip are timed off the same edge of the synchronization signal. The accuracy of this timing was discussed in section IIA.

In Figure 5 an example of the microsequencer program block diagram is presented, performing an event acquisition during the collider run. The program starts with the downloading of the chip numbers to the chips' memories and reading them back for the diagnostic purposes. This part of the program is executed only once. Then the Sequencer's clock is shut off, waiting to be synchronized with the next beam crossing. After receiving the Start_Clock signal, the Partial Reset procedure is executed in which only the chip integrator is reset and the events sampled in the previous crossing are kept on the Sample&Hold capacitor. Then a check is made of the level of the C&S signal.

The presence of the C&S signal is an indication to integrate a new sample. When the integration cycle is finished the Sequencer's internal clock is stopped by the Stop_Clock signal and the synchronization with the next beam crossing takes place. The following Start_Clock signal enables the operation of the internal clock again and the Partial Reset is performed. There are 43 80 ns microsequencer's instructions executed between the Start_Clock and Stop_Clock sequence. The total number of instructions in that cycle is shared between the Partial Reset and the Integration subprocesses and the exact numbers depend on a particular implementation. In one of them the integration time requires 19 of these instructions, and low noise constraints applied to the reset operation demand another 22 instructions. In the remaining two instructions the Start_Clock and C&S signals are checked. The Sequencer's ability to select and check the separate conditions on each microsequencer instruction allows us to execute the full event sampling cycle.

The absence of the C&S signal indicates that the event stored during the previous crossing passed the L1 trigger and the event may be prepared for readout. In this case a check is done of the microsequencer's Process Flag which ensures that the threshold restoring process is executed only once per event. The event sampling and the threshold cycles must be executed with the full knowledge of the beam crossing. In the first case the beam crossing should appear inside the integration window and in the second one the integration must be performed between the beam crossings. The threshold restoring subprocess, shown in Fig. 4b, requires two beam crossings to be performed with the constraints defined above. The microsequencer program inhibits the Stop_Clock signal

and the number of wedges. Based on the measurements of the connection time and the word transfer rate, we expect for 6 wedges the offset to be 140 μs and the slope to be 1.2 μs per 6 words. In Fig. 6 and Table 2 the results of the measurements of the SSP average scan times as a function of the number of hits per wedge are shown. The scan time is defined as the length of time that the SSP Done signal is in the false state. The measurements agree with the estimates and confirmed the transfer rate of 200 ns/word.

Table 2

Predicted (Plain) and Measured (*Italic*) Values of the Sequencer and SSP Scan Times. [ms].

Occupancy	Sequencer Scan Time				SSP Scan Time
	A0 Readout		A1 Readout		
	1 Block	6 Blocks	1 Block	6 Blocks	6 Blocks
10%	0.81	0.91	0.41	0.41	*0.36*
20%	1.53	1.72	0.81	0.81	*0.59*
40%	*2.97*	3.35	*1.61*	*1.61*	*1.04*
100%	*7.28*	8.24	*4.02*	*4.02*	*2.40*

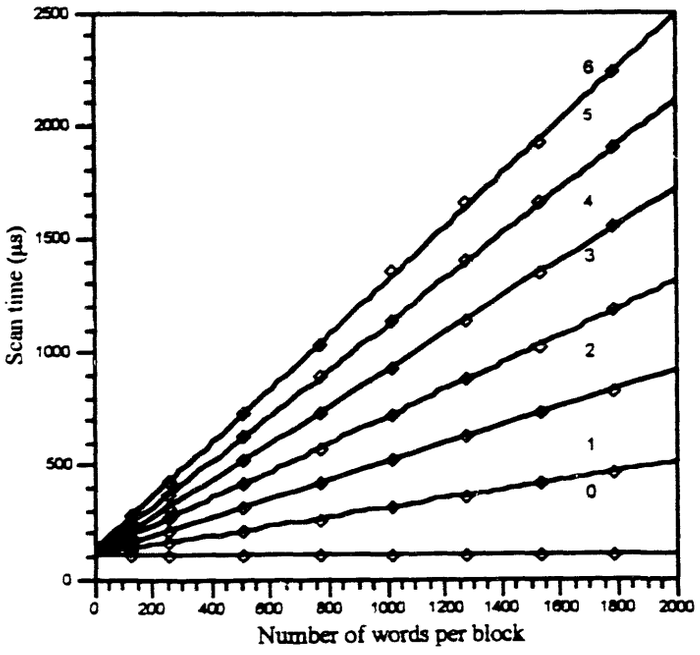

Fig. 6. The SSP scan time of the Sequencer Event Memory blocks as a function of the number of words. The lines correspond to the readout times of different numbers of blocks.

The SVXD chip has the ability to sparsify the data and read only channels above the threshold. The number of hits per interaction is a function of the following factors: event multiplicity, noise fluctuations on the threshold settings, number of low momentum spiral tracks and beam gas interactions. Simulation procedures for all these processes don't exist at this time, but we estimate that the average occupancy should not exceed 10% of channels. We also expect this number to increase slowly as a function of the absorbed radiation dose by the detector components, mainly due to the increased number of noisy channels with high leakage current value that will result.

If the total scan time exceeds 2 ms, there is an alternate scheme in which the SSP issues Done signal to the system when an event is stored in the Sequencer's memory. While the SSP moves data from the Sequencer's memory, the microsequencer can execute further integration cycles to acquire new events. If the Level 2 trigger arrives before data are moved from the Sequencer the event can be stored on the chip capacitors. Application of this method reduces the total scan time by hiding the scan time of the SSP. The number of channels which can be read within 2 ms is equal to 30% in the first scheme and 50% in the second one.

VI. Experience from the Development and Testing of the SVX Readout Electronics

Design, production and integration processes of the readout system of this scale are major tasks. They involved the acquaintance with the detector geometry and the architecture of the CDF DAQ environment. The first task was to prepare the specification followed by the design of the readout components. The boards evolved through as many as 5 revisions. Each revision offered the possibility of making changes to the design and adding features that can enhance the overall system performance. In the development of systems such as SVX it was important to use each opportunity in different phases to the fullest. Many of the system's capabilities were not a part of the original design specifications and are present only through a close collaboration between physicists and engineers. The people who are commissioning the SVX readout were involved in the project from the design specification stages. Some of them are now evaluating and measuring the performance of the SVX detector.

The other aspect of the integration process was to move the readout system from the test stand environment to the CDF DAQ system. The testing and diagnostic environment is usually a subset of the system that will be used for data acquisition. For the SVX project there were two types of test stands, both with a single Fastbus crate outfitted with a QPI, which were available in addition to the real DAQ system. The first one was used for testing the Sequencer, Controller and Digitizer modules. The other was a more complex system and was operated in the full data acquisition mode. Even with the special attention paid to the problems of moving from the test stands to the multi-crate Fastbus network in the detector building, several elusive problems were identified and the design revised.

Problems also showed up during the integration of all the SVX system components. As it is very easy to guess, but

difficult to implement, some of these problems could have been avoided or diminished by more frequent if not daily communication between the collaborating institutions or individuals. This experience is of particular importance for new supercollaborations which are now being formed that express interest in close contacts with industry on the design and production work of the future detectors.

VII. CONCLUSIONS

The commissioning process of the Fastbus data acquisition system for the CDF Silicon Vertex Detector has been presented. Different readout components of this system have been characterized. The methods of integration with the CDF DAQ system and the synchronization with the Tevatron beam crossings were described. The performance of the SVX readout system was evaluated by measurements of the scan times and the stability of the chip control signals. It was shown that by introduction of four parallel readout chains and sparsification of the data on the chip the SVX detector can be read out within the limits imposed by the CDF dead time.

The timing signals generated by the Sequencer's internal clock can be adjusted by use of a vernier delay utility in steps smaller than the clock period. The timing and stability of the delay settings were measured and the results indicate that this function performs very well. The standard deviation of the signals was better than 1ns. The stability of the main clock was checked. The number and timing of the microsequencer instructions which can be executed within beam crossing time were established. The basic reset and integration cycles can be implemented in 43 instructions and it takes 3438.630±0.003ns to execute them. The system functionality allows the users to perform automated, precise and fast calibration and monitoring of the detector.

VIII. REFERENCES

[1] W. Wester et al.,"The CDF Silicon Vertex Detector: An Overview and Test Results", these proceedings.

[2] KineticSystems Corporation, 11 Maryknoll Dr., Lockport, IL60441, Model F820 Scanner Processor, October 1987.

[3] K. J. Turner et al., "Control and Data Acquisition Electronics for the CDF SVX Design", these proceedings.

[4] S. A. Kleinfelder et al., "A Flexible 128 Channel Silicon Strip Detector Instrumentation Integrated Circuit with Sparse Data Readout", *IEEE Trans. Nucl. Sci.* NS-35 171 (1988).

[5] M. W. Bailey and S. M. Tkaczyk,"System Software Design for the CDF Silicon Vertex Detector", these proceedings.

[6] F. Abe et al., "The CDF Detector Overview", *Nucl. Instr. and Methods*, A268 387-403 (1988).

E. Barsotti at al., "Fastbus Data Acquisition for CDF", *Nucl. Instr. and Methods*, A268 82-92(1988) .

END

DATE FILMED

2/05/92