Data Acquisition System for E-835 Silicon Detector

P. Musico [a], M. Pallavicini [b] and S. Minutoli [a]

[a] *INFN, Sezione di Genova, via Dodecaneso 33 I-16146 Genova, Italy*

[b] *Universitá di Genova and INFN, Sezione di Genova, via Dodecaneso 33 I-16146 Genova, Italy*

Abstract

In this paper we describe the DAQ system for the Silicon Pad Detector used in experiment E-835 at Fermilab. This system is completely custom designed and it has been installed on site in spring 1996. It is composed by three different electronic boards used to read-out the data coming out from the detector, to collect all the non-zero data, to store the data in a memory segment seen by the VME DAQ controller. The work, started in spring 1993, was carried out in parallel with the design of the detector. The system has been tested at CERN in november 1995 and is now fully installed and ready for data taking.

1 Introduction

Experiment E-835 scientific goal is to complete the study of charmonium spectrum by forming all resonant states in proton-antiproton annyhilation, using an internal jet target installed within the Antiproton Accumulator at Fermilab.

The experimental apparatus is a non-magnetic spectrometer composed mainly by a very compact tracking system, a threshold Cerenkov counter and an electromagnetic lead glass calorimeter.

In the next run one of the main challenges will be the detection of the η_c resonance in the $\Phi\Phi \longrightarrow 4K$ channel, mainly using the informations coming from the tracking system. For this reason we have built a Silicon Pad Detector (see [1] and [2]) with a fast and completely custom made read out system that is the subject of this paper. A more detailed technical description can be found in [5].

The main guidelines in the design have been dictated by several constraints: the high instantaneus luminosity (E-835 will run at 5.0 10^{31} $cm^{-2}s^{-1}$, so the

Preprint submitted to Elsevier Science *5 September 1996*

trigger rate will be as high as 10 kHz), the compatibility with the other part of the DAQ system (VME standard was requested), the availability of the ICAR16[1] chip as front-end amplifier and of the FEROS[2] chip for encoding and zero suppression.

disk1:[pallavicini.silici.nim.figure]system.ps

Fig. 1. Block diagram for the Silicon Pad Detector DAQ system. The detector, the 8 Slab Coding Board, the Feros Bus Interface and Read Out and Control module are schematically shown. Data are read and coded by each SCB and then transferred to the ROC via FBI and stored on a VME readable FIFO memory.

A block diagram is schematically shown in figure 1. The data handling system is composed by 8 readout boards, one local control board and one master VME card.

The detector is made by 24 SLABs assembled on a cylindrical structure. On each SLAB both front end electronics and two layers of silicon diode detectors are mounted; the two layer are kept completely independent, together with all read-out electronics, to reduce inefficencies due to malfunctions and from now on will be referred in the text as INNER and OUTER layers. Schematically (see figure 2) each silicon pad is connected to a charge preamplifier followed by a discriminator. The discriminators digital outputs are sent to the read out

[1] ICAR16 is supplied by Smart Silicon System, Avenue de Chailly, 23 CH 1012 Lausanne Switzerland.

[2] The FEROS is a gate array developed at CERN by G. Schuler for the WA92 experiment. In this project we do not use the internal circuit that implements a distributed adder.

boards (the SCBs). Practically, 16 silicon pads are connected to a 16 channels charge preamplifier with a serial output. The discriminator works serially on the 16 analog signals coming from the pads with a single programmable threshold.

`disk1:[pallavicini.silici.nim.figure]slab.ps`

Fig. 2. The SLAB Block Diagram

The signals coming out from each of the 24 SLABs are sent to the read out and coding boards called Slab Coding Board (SCB) via a 50 wires flat cable. The 8 SCBs needed to handle the 24 SLABs are housed in a VME crate whose distance from the detector must be within 5 m. Then all coded data are sent to the Read Out and Control Module (ROC) through the Feros Bus Interface (FBI) and stored on a VME readable memory segment.

In the following sections we describe the various part of the system and summarize its performance.

2 The Slab Coding Board

Each Slab Coding Board (SCB) permits readout and coding of non-zero data coming from 3 different SLABs ($192 \times 3 = 576$ channels). In figure 3 the block diagram of the SCB is shown. The SCB itself and one SLAB are shown in figure 4.

disk1:[pallavicini.silici.nim.figure]scb.ps

Fig. 3. The Slab Coding Board Block Diagram

disk1:[pallavicini.silici.nim.figure]scbslab.eps

Fig. 4. A SCB connected to one SLAB

It is made with three identical sections and some common parts. For each SLAB there is an interface section dedicated to receive data and to transmit control signals toward the ICAR16 and the DACs.

The received data converted in TTL levels feed the input of the 2 coding circuits (the FEROS) needed to handle independently the 2 layer of the detector. References for ICAR16 and FEROS can be found in [3]. Each FEROS is connected to 6 ICAR16 output signals: the total number of channel coded by a FEROS will be $16 \times 6 = 96$ pads. The FEROS code the non-zero incoming data into a number in the range 0 - 95, using a 7 bit output bus. An external circuit, built using PLD, generates the slab number in 5 bit. These bits are added to the FEROS output bus and a 12 bit bus is made.

The FEROS and the PLDs of each section are connected with the corresponding ones in the forthcoming section using a daisy chain scheme. The chaining connection allows to start the coding work in one section as soon as the preceding one ended its job.

Each SCB is daisy chained with the following one: in this way all FEROS (24 for each layer) are daisy chained together. The chaining is initiated by the local control board (the FBI). Again the FBI receive the end of the chain and detects the end of the data stream for both INNER and OUTER buses.

Each SLAB has its independent silicon bias voltage line: this voltage is generated by an external power supply unit and distributed to the SCBs via 24 dedicated lines on the crate backplane. Each SCB have three jumpers banks to select the appropriate bias lines from the backplane.

On the SCB is implemented also the circuit to distribute the data to the DACs on the SLABs. It is made with six 11 bit shift registers (serial in, serial out), with the 6 intermediate taps that are the data point connected to the DACs.

All digital signals exchanged between the SCB and the SLAB are transmitted using differential electrical protocols: ECL or RS422, in order to minimize the digital noise effects on the data.

To read the whole detector we need to use 8 SCBs. They are connected together using the VME auxiliary backplane (the J2/P2 connector)[3] cabled ad-hoc for this system. In particular, the 8 SCBs have in parallel the two 12 bit data buses (INNER and OUTER) and some control signals; other control signals, the FEROS related ones, are connected in daisy chain.

In the crate where the 8 SCBs are housed, there is the local control board

[3] The SCB and the FBI are mechanically compliant to the VME standards and housed in a VME crate.

called Feros Bus Interface (FBI). This unit handles the 8 SCBs located near behind and exchange data with the control board set in the counting room.

3 The Feros Bus Interface

It performs the following functions:

(i) For each trigger signal (either hardware or software) it generates the control signals to be sent to the ICAR16 on the SLABs (via the SCBs) and to the FEROS chips on the SCBs.
(ii) It manages two external asyncronous buses (called FEROS buses), one for each detector layer. These buses carry the silicon pads data to be read by ROC and the commands issued by the master controller.
(iii) It executes the commands issued by the read out controller. They are: software reset, software trigger, write DAC setting, set DACs, reset DACs settings.
(iv) It handles the SLABs discriminators threshold loading. It stores all data in a FIFO and then it can read them and generate a serial stream to the SCBs that retransmit the correct data to the SLABs.

A picture of the FBI board is reproduced in figure 5.

3.1 Generation of control signals

This circuitry is triggered by the level 0 `STROBE` signal. It must be generated within 240 nsec from the interaction to ensure that the charge preamplifiers (ICAR16) work in the correct peaking time range (up to 300 nsec). In addition to this signal, coming from trigger logic as a NIM level, this circuit can be triggered by two other sources: another NIM signal, available for test purpose, and a software trigger generated by the readout control card.

Figure 6 shows the control signals, whose meaning is the following:

- `TH_CS`: it is the hold signal for the charge preamplifier ICAR16. It is generated with minimum delay from the `STROBE`. In real life it arrives to the ICAR16 after 60 nsec from the `STROBE` (hence after about 300 nsec from the interaction). All the ICAR16 are feeded in parallel with this signal.
- `CK_LT`: After 100 nsec from the leading edge of `TH_CS` 17 clock pulses with 50% duty cycle are generated with 200 nsec period (5 MHz frequency). This signal has three functions: on the leading edge it extracts the analog data from the preamplifier (and the comparator starts its job); on the trailing edge it latches the comparators outputs; with the next leading edge the

disk1:[pallavicini.silici.nim.figure]fbi.eps

Fig. 5. A picture of a Feros Bus Interface

digital data are loaded into FEROS. Practically the first 16 pulses extract data from the ICAR16 and digitalize them; starting from the 2^{nd} to the 17^{th} the digitalized data are loaded into the FEROS.

This pipelined system is the most efficient to handle the data transfer between the detector and the SCBs. The limit of 5 MHz frequency is imposed by the propagation delay on the flat cable. We need about 3.5 μsec (200 nsec $\times$ 17) to read the whole detector.

- `XLOAD`: Is the signal used to indicate the end of loading data into the FEROS. After the assertion of this signal FEROS is ready to give the coded data at every read command.
- `XCLEAR`: Is generated at the beginning of the sequence and it is the RESET to the FEROS. In this way we are sure that FEROS is ready to receive the data.

At the end of the generation of this control signals sequence it is possible to read the coded data from FEROS. This is carried out generating a read disk1:[pallavicini.silici.nim.figure]icarfero.ps

Fig. 6. The ICAR16 and FEROS Control Signals

request on the external asyncronous buses from the ROC.

An additional asyncronous `CLEAR` input (at NIM level) is provided to have the possibility of resetting this sequence.

3.2 The Feros Bus

To handle the communication between the FBI (in the pit, near the detector) and the ROC (in counting room) we implemented an asyncronous, point to point, master-slave protocol. The reason why we need an asyncronous transmission system is the long distance between FBI and ROC that could be up to 30 meters.

It is always the master (ROC) that can start a bus operation; the slave (FBI) answers to the issued command with a status condition and the acknowledge. Both the status and the data lines (in a read cycle) are valid on the leading edge of the acknowledge signal.

All signals are transmitted using RS422 electrical interface specifications, using a 40 twisted-pairs flat cable. We use a balanced interface to reduce the noise effects on the cable.

We implemented two equivalent buses to handle data coming from the two detector layer in an independent way. In particular only the INNER bus is used to issue commands to the FBI, while the OUTER one is used only to read data. A more detailed description of these buses can be found in [5]. Every write operation on the bus needs a time depending on the flat cable length because we use a full handshake protocol. Considering a 30 m long cable, we could have a transaction every 750-800 nsec.

Read operations are performed faster. A sort of block transfer is implemented: when a read command is issued the slave respond with a new data and the corresponding acknowledge every 300 nsec until the `End Of Data` condition is reached. In this way 10 Mbytes/s can be transferred.

3.3 Discriminators Thresholds Handling

Some of the commands are related to the DACs settings. In particular it is possible to store the 8 bit threshold values in a FIFO writing data on the INNER feros bus with command code 3.

After that all data are loaded (288 values stored in the FIFO) issuing a command code 4, a sequential circuit is triggered: the data are read from the FIFO, for each byte a 3 bit address field is added and the resulting 11 bit word is serialized. The address field is related to the internal DAC channel. This data stream is sent to the SCBs together with some control signals. The SCBs retransmit these signals to the SLABs after a sort of demultiplexeing.

The loading operation that follows the command code 4 is carried out in about 5 msec (the DACs are loaded at 1 MHz frequency).

4 The Readout and Control Unit (ROC)

The ROC is a VME module specifically designed for the management of the entire data acquisition system of the Silicon Pad Detector.

It has an additional local control port to connect a personal computer via a parallel interface to its internal devices: with this facility, it is possible to debug easily the system also when all the DAQ programs are up and running.

In figure 7 a detailed block diagram of the ROC is shown.

The interface with the already described system (FBI + 8 SCBs) is made with the two FEROS buses, one for each detector layer. Practically the cables disk1:[pallavicini.silici.nim.figure]roc_nocopy.ps

Fig. 7. Detailed ROC Block Diagram

arrive to the rear of the crate, in the same position of the ROC, on a dedicated transceiver board used to handle level conversions and signals regeneration. The connection between the conversion unit and the ROC itself is made using 40 of the 64 user definable lines on the J2/P2 VME connector.

4.1 VME specifications

The ROC is a VME slave device, with no master capability.

Only extended (32 bit) addressing is supported. Standard and short addressing (24 and 16 bit) are not supported. The board will respond to privileged and non privileged data and program address modifier codes (AM codes: `09, 0A, 0B, 0D, 0E, 0F` [4]).

Only longword (32 bit) aligned data transfers are supported. Unaligned data transfers, word transfers and byte transfers (16 and 8 bit) are not supported.

Single VME cycles and pipelined single VME cycles are supported, block transfers are supported as well, during the readout of data buffer.

[4] In the following detailed description all numbers are represented using hexadecimal notation unless otherwise noted.

No interrupt capability is provided: the master must poll the status register to see if there are data to read.

4.2 ROC memory map

The lower 32 Kbytes (8 Klongwords, form `0000` up to `7FFC`) provide access to the data buffer, which is organised in a FIFO fashion. A range of addresses is provided so that the VME master can read data with a single block transfer cycle.

The first address (`0000`) contains the data header with some fields comprhending the number of data words associated and a trigger identification number.

The header contains a 12 bit wide field which stores the number of data longwords associated with this event. Word count equal zero is allowed for empty events. A 4 bit field contains a trigger identification number coming from the ESN module [4] [5].

The following addresses (`0004, 0008, ..., 7FFC`) contain data coming from the detector. The number of valid data migth differ in the 2 detector layers: if a 16 bit field contains the value `0FFF` means that there are no more data associated with this layer (it is an out of range flag). Note that the VME master can read only once the FIFO data, because when a read cycle has been carried out, the FIFO read pointer has been incremented.

At address `8000` is located an 8 bit register used to transfer DAC data to the FBI board: the master must write all DAC data in this location before issuing a `Set DAC Value` command.

At address `8004` is located the Status Register. Only the lowest 12 bits are used; bits from 12 to 31 are unused and their value is unknown.

At address `8008` is located the Control Register: it is used to set working conditions and to issue commands. Only the lowest 7 bits are used; bits form 7 to 31 are unused and writing these bits has no effect.

4.3 Local Control

A local control port is provided to easily debug the system also when the DAQ program is running. A personal computer can be connected to this port

[5] It is the module used to generate the trigger identification number in E-835 DAQ system.

via a 40 lines parallel interface. All ROC internal devices are accessible via this local port but with lower speed, due to the very simple parallel protocol implemented in software by the PC. When the PC is enabled (bit 6 of Control Register set to 1) all VME operation are discarded with a Bus Error condition except the operation of reading the Status Register: this is useful to monitor the bit 7 to see when the PC release the control of the board. Additional references for the PC interface can be found in [5].

4.4 ROC operations

After power up the DAQ system for the Silicon Pad Detector must be initialized loading the thresholds data for the SLABs comparators into the DACs registers. This must be done with the following sequence of operations:

(i) Disable the module for normal operations (writing bit 5 in the Control Register)
(ii) Issue a `Software Reset` command (writing bit 0 in the Control Register)
(iii) Issue a `Reset Dac Value` command (writing bit 3 in the Control Register)
(iv) Load all thresholds discriminator data (writing the 8 bit value as longword at address `8000`) waiting 1 μsec at least between successive write
(v) Issue a `DAC Set` command (writing bit 2 in the Control Register)
(vi) Wait at least 5 milliseconds: time to carry out the operation
(vii) Issue a `Software Reset` command (writing bit 0 in the Control Register)
(viii) Enable the module for DAQ operations (clearing bit 5 in the Control Register)

When the unit is enabled for data acquisition, it monitors the `STROBE` input signal (the same signal used to strobe the detector) coming from the 1^{st} level trigger logic. When an active `STROBE` is detected the device waits about 3 μsec so that the entire detector data are collected into the front-end and transferred into the SCBs. After this time-out if no `TRIGGER` is arrived (always from the 1^{st} level trigger logic) a `CLEAR FBI` signal is generated and the process is restarted. If a `TRIGGER` arrives within 3 μsec from the `STROBE` the unit generates read command on both FEROS buses in parallel and stores the data coming from FBI in the FIFO data buffer.

If a `TRIGGER` comes without an associated `STROBE` an empty event is recorded with the correct trigger identification number.

The `TRIGGER` signal is also used to latch the ESN trigger number into a register.

Each read operation on the external buses increments a counter. When both buses give the `End Of Data` condition an entire event is read and the header word made with the word count and the trigger identification is copied into the header FIFO. At this time an event is available by the DAQ system and the read-out process is restarted.

When the read-out process is in progress the `BUSY` signal is active: the trigger logic must use this signal as a veto in order to disable the generation of `STROBE` and `TRIGGER` while the ROC is reading out the data from the detector.

The VME processor can read an event as follows: it must monitor the Status Register to see if there are data in the header FIFO (bit $6 = 0$). When there are data it reads the header word at address 0000 and then reads the number of data word as read into the `WORD COUNT` field in the header at successive addresses (`0004, 0008, ...`). It is possible to use a block transfer at these addresses to speed up the operation.

There is a possibility that an event generates no data in the Silicon Pad Detector: in this case an header word with the `WORD COUNT` field $= 0$ is generated. In this case no data readout must occur.

All the buffers (data and header) can be asyncronously cleared using the `CLEAR FIFO` signal. The correct way to do this is to assert this signal when neither `STROBE` nor `TRIGGER` are generated, to ensure that the internal state machine is in the starting state. This is true when the `BUSY` signal is inactive.

In figure 8 a picture of a ROC connected with its level conversion unit is presented.

5 Performances

The VME system we used for testing was based on a FIC8234 controller: a 68040 cpu board, supplied by CES [6], equipped with 8 Mbytes of RAM, an hard disk, a floppy disk and an ethernet controller. The Microware OS9 operating system was installed on this unit.

This system has worked to test the detector on a test beam at CERN [7] in fall 1995. See [1] for test results. Now, in the final installation, the controller is a Motorola MVME167 with the VxWorks operating system [8].

[6] Creative Electronic System S.A., Route du Pont Butin CH-1213 Lancy (Switzerland)

[7] We used the Hall East T10 5 GeV/c π^- beam.

[8] Configuration and performances for the 2 CPU boards are very similar: the test program we used was recompiled without modification and ran at the first time.

disk1:[pallavicini.silici.nim.figure]foto_daq.eps

Fig. 8. A ROC connected with its conversion unit and a FBI

The system can read each couple of silicon pad data from FBI every 300 nsec. In this configuration we are able to read the whole detector ($96 \times 24 = 2304$ words) in less than 700 μsec [9].

In real life we expect to have events with about 50 words (on average), including good data and noisy channels. so we expect an averaged read out time of 20 μsec per event. Assuming the trigger frequency of 3 KHz (given by the trigger logic system) and events made by 50 words, the dead time introduced by our system will be less than 10%.

[9] This number refers to an event with all 4608 pads turned on.

Acknowledgement

We would mention here the persons that helped us in this job.

Nanni Darbo and Leo Rossi (INFN Genova) kindly helped us at the beginning of the project, giving us precious information about ICAR16 and FEROS.

Adriano Bevilacqua and Fabio Siccardi (INFN Genova) did the hard job of the designing the printed circuit boards used in the system and the assembling of the prototypes and the final versions of the electronic units. They worked at the test of the system both in lab and on the test beam, cabling the crates used and helped us in the modification needed on the boards.

Giancarlo Barisone, Marco Negri and Pasquale Poggi (INFN Genova) have worked principally at the detector assembly, mechanics and silicon crystals characterization, but they also did a precious job in the test beam set up at CERN.

We would also thank Prof. A. Santroni and dott. M. Macrì for their advices and encouragement.

References

[1] A. Buzzo et al., *A Silicon Pad Detector for E-835 Experiment at Fermilab* submitted to NIM A; Fermilab memo n. 357 (feb 1996).

[2] M. Pallavicini *Tesi di Dottorato di Ricerca in Fisica (Ph.D Thesis)* Genoa University (1995).

[3] A. Beer et al., *The beauty contiguity trigger of the BEATRICE experiment: detector, readout and processor overview* NIM A 337 (1994) 280-294.

[4] R. Majka *E791 ESN module* (26 jan 90) Fermilab Computing Division note.

[5] P. Musico et al., *The E-835 Silicon Pad Detector DAQ System* Experiment E-835 internal note n. 358 (1 Feb 1996).