SDC-92-239

SDC
SOLENOIDAL DETECTOR NOTES

DESIGN OPTIONS FOR ANS SSC WIRE CHAMBER DATA COLLECTION SYSTEM

October 21, 1992

G. Stairs and P.K. Sinervo
University of Toronto

SDC-92-239

UOFT-DCC-01

Design Options for an SSC Wire Chamber Data Collection System

G. Stairs and P. K. Sinervo
University of Toronto
Toronto, Canada

October 21, 1992

Abstract

Various design options for the architecture of the data collection system for a wire chamber tracking system are discussed.

1 Introduction

The design of the readout electronics of a wire tracking chamber for the SDC detector has evolved to the point where it is now salient to consider the architecture for the interface between this system and the rest of the detector readout system. The next stage in the design effort is the development of the data collection system that moves the digitized hit information from the individual front end circuits to the data acquisition system (which we define as the point in the system where buffers of data are transferred from the detector (or a shielded counting room) to the third level trigger).

2 System Functionality

The function of the data collection system is:

- to collect data from the individual front end electronics (FE) circuits, and format it into event data blocks, with one block of data corresponding to all the information for one event from a set of FE chips,
- to collect and record error/status information from the FE chips and internal sources and add the pertinent data to the event data block, and

- to provide the distribution of the necessary set of fast signals required to drive the FE chips and the data collection system. These fast signals include
  - Beam Clock
  - L1 Accept
  - L1 Synch
  - L2 Accept
  - L2 Synch

3 Overall System Design

3.1 Front End Chip

We will assume that the overall channel count of the wire chamber system is 200,000. We also assume that each FE chip instruments 8 wire chamber channels, implying an overall FE chip count of approximately 25,000. We further assume that the front end chips are mounted on readout cards in proximity to the tracking detector. Each card provides high voltage and electronic readout for approximately 200 channels. Each card would therefore carry 25 FE chips.

We are going to consider the Data Collection Circuit (DCC) architecture sketched in Fig. 1. The function of each of the components in this schematic are described in detail below.

3.2 Front End Readout Controller

The readout of the FE chips has to have a significant degree of parallelism as the instantaneous data rates can be quite large. For example, if we assume a L2 trigger rate of $r_{L2} = 10^4$, an occupancy/interaction (ie. the average number of hits read out for each event) of $f_o = 0.30$, and suppose that 4 bytes of information must be transferred for each datum, then the required bandwidth into the DAQ system is

$$N_{chan} r_{L2} f_o * 4 = (2 \times 10^5)(10^4)(\overset{0.30}{\cancel{0.10}})(4) \quad (1)$$

$$= \overset{24}{\cancel{8}} \times 10^8 \text{ byte/s.} \quad (2)$$

If we suppose that the average bandwidth from any one FE chip should be less than of order 10^6 bytes/s, then the number of parallel readout paths must be about 1000. Since this is

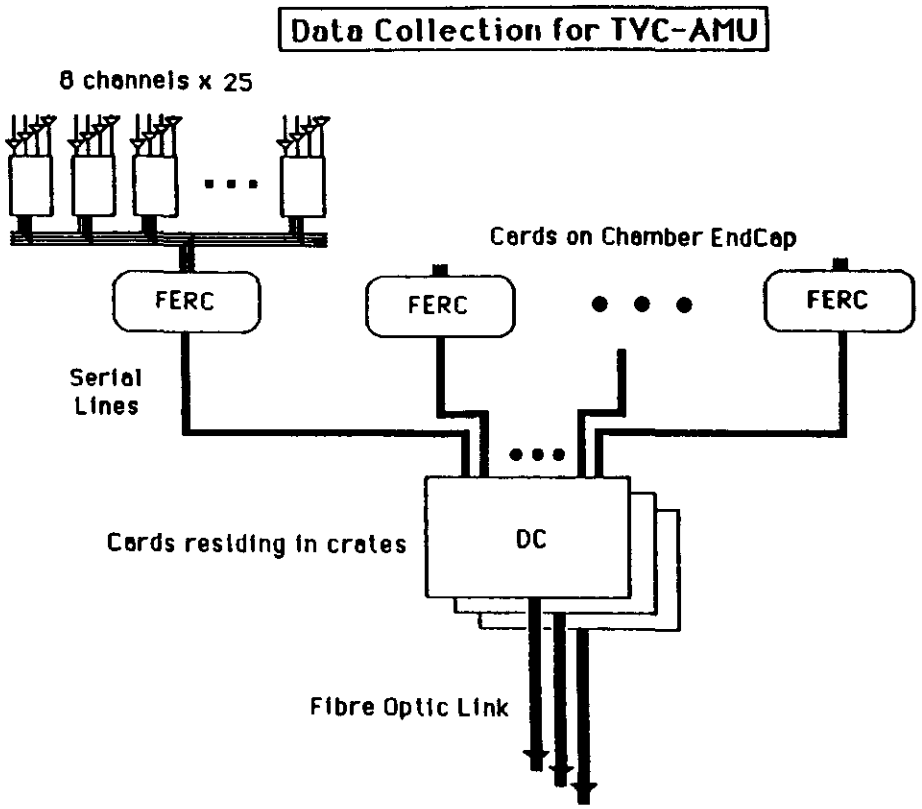

Figure 1: Schematic of the overall data collection system for the wire chamber readout.

approximately the number of readout cards, we will consider an architecture in which the readout of the FE chips takes place independently on each card.

We consider a specific architecture were the FE chips on a readout card are connected to a single device called the FE Readout Controller (FERC). The purpose of the FERC is to coordinate the readout of the FE chips, and organize the data for a given trigger into a sub-block of information. The FERC also does a modest amount of monitoring and error detection. We envision the FERC to be a VLSI circuit (or a set of chips) that is mounted on the readout card holding the FE chips. In this design, we aim to keep the complexity and functionality of the FERC to the necessary minimum, partly because we believe it will be located in proximity with the FE circuits and be relatively inaccessible.

3.3 The Data Collecter

The Data Collector (DC) is responsible for reading out the FERC's and providing the interface to the DAQ. Each DC is connected via a parallel or serial bus to a number of the FERC's. For example, if we have 32 FERC's connected to each DC, this would require having approximately 30 DC's for the readout of the tracking chamber. This implies that each DC will be reading out on average 2000 hits/trigger.

The DC may be rather complex and could be implemented as a general-purpose computer. We are making a conscious design decision to keep the FERC and FE chip as simple as possible, and trade that off with increased functionality in the DC itself. The DC should be able to exist as a "standalone" system that can perform the readout of a set of FERC's under the control of a workstation or personal computer.

4 FERC and FE Chip Interaction

There are several fundementally different ways the FERC could communicate and read out the FE circuits. The FERC could "pull" data from each of the FE chips, the FE chips could send data to the FERC as it becomes available, or the interaction between the two could be a mixture of these two schemes. In all these cases, we have assumed that the data coming into the FERC would not be ordered on the basis of trigger number (as data from different FE chips will be "jumbled" together). One of the tasks of the FERC would be to organize all the data for a given trigger into a single complete block of data. This has an impact on the functionality of the FERC that we will comment on later.

We will now briefly discuss the merits of the various schemes for communicating between the FERC and the FE chips.

4.1 The Push Option

In this scheme the FE chip would transfer data to the FERC whenever it becomes ready. This would require the communication between the FERC and FE to take place over either dedicated serial lines or a parallel bus structure shared by several FE chips. Given the number of FE chips instrumented by a single FERC, the use of dedicated serial lines is probably precluded.

As a datum becomes available, the FE chip would have to arbitrate for mastership of the parallel bus, and having obtained it would then transfer the data to the FERC. Some

provision would have to be made for detecting situations where the FERC is unable to accept more data. In addition, some form of arbitration control would have to be provided for the parallel bus, either by the FERC or by a separate device.

The advantage of this scheme is:

- it is a totally data-driven scheme where the transfer of information (including signals to manage bus protocol) are kept to an absolute minimum,
- the buffering requirements on the FE chips are reduced, and
- the FERC is not involved in polling the FE chips for data and can be devoted entirely to the processing of data.

The disadvantages of this scheme are:

- each card requires some form of arbitration control for the parallel bus (this could be part of the FERC),
- avoiding buffer overflows in the FERC requires some form of handshake with the FERC, and
- the FERC is no longer able to easily determine the state of the FE circuits.

4.2 The Pull Option

In this scheme, the FERC would actively control the transfer of data from the FE chips to the FERC. Perhaps the simplest protocol is for the FERC to poll each FE chip in turn, requesting any available data. For 25 FE chips and the standard maximum data and trigger rates, an average bandwidth of

$$N_{chan} \times N_{FE} \times f_o \times N_{byte} \times r_{L2} = \overset{8\times}{25} \times 0.30 \times 4 \times (\cancel{1} \times 10^4) = \overset{2.4}{\cancel{12}} \times 10^6 \text{ byte/s} \qquad (3)$$

is required. However, since only one-third of the FE chips will have data for a given interaction, this would require the polling to be relatively efficient to keep up with the dataflow.

An efficient implementation could be to have dedicated "chip select" lines between each FE chip and the FERC. This would require of order 6 signals dedicated to this addressing task (5 address signals and one synchronization signal) if this data were multiplexed onto an

address bus (it would require of order 25 lines if a dedicated chip select line was allocated to each FE chip, which we believe is an unreasonable use of signal lines) Each FE chip would respond to its own address by placing any valid data on a parallel data bus that all FE chips were connected to. A special data pattern could be used to signal an "end of transmission". The rate of data transfer would be under the control of the FERC.

The benefits of this scheme are:

- the FERC can quickly detect a FE chip that is misbehaving by monitoring its responses (each chip is interrogated for each L2 trigger).
- the electronics required to implement this scheme is modest both at the FE chip and the FERC, with the majority of the intelligence residing in the FERC.
- The system is still data-driven to the extent that data is transmitted when it exists. The potential problem of buffer overflows in the FERC can be treated within the FERC (it can throttle its polling if necessary).

The disadvantages are:

- it requires a dedicated set of "chip select" signals, and
- the FERC must poll each FE chip to interrogate it rather than have the FE simply push data to the FERC.

It is possible to have the "pull" scheme implemented in a somewhat different manner. However, each variation that we have considered implies more complexity in either the FERC or FE without an attendant increase in either reliability or robustness. One scheme would be to have the FERC ask the FE chips for data from a given trigger only (this would simplify the reformatting the FERC must do). However, it implies that the FE chip knows how to separate the data from adjacent triggers that may actually *share* some data hits (depending on the number of crossings the detector must integrate over).

4.3 Data Formats

The format of data transferred between the the FE chips and the FERC have to provide the following functionality:

- the channel address (for example, the FE chip and channel number),

Flag Value	Meaning
0	Valid data
1	Channel address
2	End of transmission
3	Monitoring information
4	Error information

Table 1: The flags used to identify the type of information transferred in each 4-byte packet.

- datum for a given channel (we assume that 24 bits is sufficient), and
- allow "end of transmission," "error states" and other monitoring status to be communicated from the FE chips to the FERC.

This information transfer can be easily implemented by having each datum transferred between the FE chip and FERC by identified by a "flag" consisting of order 8 bits that identifies the type of datum being transferred. For example, the codes listed in Table 1 would provide most of the necessary functionality. The information would be organized into 4-byte packets with the first byte reserved for the flag, and the remaining 3 bytes reserved for the datum being transmitted in the packet.

4.4 The Readout Cycle for a Pull Scheme

The FE chip readout cycle would begin after the receipt by the FERC of the L2 accept signal that is distributed to the FE chips. The FERC begins polling each of the FE chips for data. This data poll has the following sequence:

1. Place address of chip on chip select bus and raise Address Synch (AS).
2. Relevant chip responds with Address Acknowledge (AK). Note: This handshake is not necessary.
3. FE chip now responds to data cycles controlled by the FERC. The FERC strobes the Data Request (DR) line requesting the first packet of data. The FE chip will respond by placing the data on the bus and raising the Data Acknowledge (DK) line. The FERC will repeat this cycle until the FE chip responds with a status word indicating "End of Transmission".

4. The FERC will then drop AS and repeat this cycle on the next FE chip.

While polling each FE chip for data, the FERC must keep track of the information already received from the FE chips. When it decides that all of the data for a given L2 trigger has been received, it reformats this information and prepares an "event sub-block." The additional information added to this event sub-block will be the FE chip ID's (in practice added to the channel ID provided by the FE chip) and any additional status or error information detected by the FERC.

If we assume that each cycle on the data bus takes of order 500 ns to complete, this implies that the address/L1ID protocol (4 cycles), data transmission (on average less than 2 cycles) and end of transmission (2 cycles) will require of order 5 μs to complete. The readout of 25 such devices would then require typically 125 μs from start to completion.

5 FERC Processing Requirements

The data that comes from the FE chips to the FERC will in all of the cases outlined above by disordered, both by channel ID and by event trigger number. One of the tasks of the FERC will be to reformat this data into event sub-blocks, where each sub-block contains the information from all of the circuits controlled by the FERC for a given event trigger.

An immediate issue that must be addressed is how the system should deal with data that could be shared between two adjacent triggers. For example, this occurs when two interactions that trigger the detector take place within the maximum drift time of the detector. The SDC muon system is expected to have a maximum drift time of order 800 ns, which implies that of order 1% of all events will contain data from the muon system that could be associated with two or more triggers (this assumes a maximum L1 trigger rate of 10^5 Hz).

Several strategies can be employed to handle this situation. One approach is to associate the potentially shared data with one or other trigger and to understand how to properly deconvolve the information at a subsequent stage in the data acquisition system. However, this would require the system to be able to keep the information for the two events together, which is difficult if this data is separated between two event records. An alternative approach is to "duplicate" the shared data and associate each datum with both event records. This then provides all the information necessary to unravel the event in either data record. However, it does require additional capability in the FERC to perform this task. In particular, it requires the FERC to know about the time structure of the subsequent L2 triggers

and to correlate this information with the timing information associated with each hit.

6 DC Requirements

The primary function of the DC is to read out a set of FERC's. The readout of the FERC can be done in several ways. We prefer to focus on a method where the DC is in control of the data transfer and coordinates it in order that the system is in a well-defined state at any given time (that is the DC knows what FERC's have already been read out). This requires a "handshake" protocol at a higher level, but a handshake would be necessary in any scheme in order that FERC does not over run the capabilities of the DC.

The DC will collect all data relevant to a given L2 trigger and present it as an event data block to the DAQ system. We assume that the different DC's do not have to coordinate their activities with respect to each other, but that event data is prepared in monotonic order. A system in which the DC simply "pushes" data into the DAQ system is possible, but so is a system where the DC receives a control signal that instructs it to begin the data transmission of the next data packet.

We assume that the DC will provide data for every L2 trigger, even when there is no real data available. The data will be in a format similar to that proposed in the "strawman" calorimeter readout design developed by the LBL group.

We assume that each FERC has a dedicated serial communication line with which it can communicate with its DC (this could be an RG-58 cable?). The question of how to synchronize the FERC readout with the completion of the FE chip readout is an open question. Several schemes are possible:

- Have the FERC send a "data ready" signal to the DC when the FE readout has completed for a given L2 trigger. This requires either a dedicated line from each FERC to the DC or having a data path from the FERC to the DC that the FERC can arbitrate for and send a message to the FERC. This requires additional complication and functionality in the FERC that may be undesirable.
- Have the DC poll the FERC's checking to see whether the FERC has data to transfer. This would presumably be done on a shared parallel bus or individual serial links. This places all of the control in the DC. It also implies that the DC may spend alot of its time polling FERC's that may be busy.

In either case, the DC has to perform the necessary bookkeeping to ensure that all of the

FERC's have been read out.