SDC-92-243

SDC
SOLENOIDAL DETECTOR NOTES

PRELIMINARY SPECIFICATION OF THE ERROR DETECTION AND MONITORING FUNCTIONS OF THE SDC WIRE CHAMBER DATA COLLECTION CIRCUIT

January 28, 1992

A. Holscher, G. Stairs and P.K. Sinervo
University of Toronto

UOFT-DCC-05

Preliminary Specification of the Error Detection and Monitoring Functions of the SDC Wire Chamber Data Collection Circuit

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

January 28, 1992

Abstract

A preliminary description of the monitoring and error detection functions required in the SDC tracking system Data Collection Circuit is presented. This note also discusses a number of issues related to overall monitoring and error detection at a system level involving both the Front End circuits, data collection circuits and the data acquisition system.

1 Introduction

A description of the functionality and architecture of the Data Collection Circuit (DCC) for the SDC wire tracking detectors is currently under development, and a preliminary conceptual design has been proposed [1]. As currently designed, the DCC is a data-driven system that transports digital information from the Front End (FE) chips located on the Front End Board (FEB) of the wire tracking device to the DAQ interface card located off the detector (the FEB instruments up to ~ 200 channels of electronics; for the proposed SDC straw tube tracker, this implies that there will about 1000 such FEB boards).

We define the DCC to be the system between the FE chips on the FEB and the DAQ Interface module residing in a crate located off the detector. We assume that data will be read out of the FE chips asynchronously and huffered in a Front End Readout Controller (FERC) that is located either on the FEB or at the other end of digital data path located between the FEB and the DAQ interface crate. A proposal for the hardware architecture has heen outlined [2], and a sketch of this system is shown in 1.

We also have assumed that the DCC system will he performing a certain amount of control, monitoring and error detection of the system located on the FEB. The DCC will distribute the

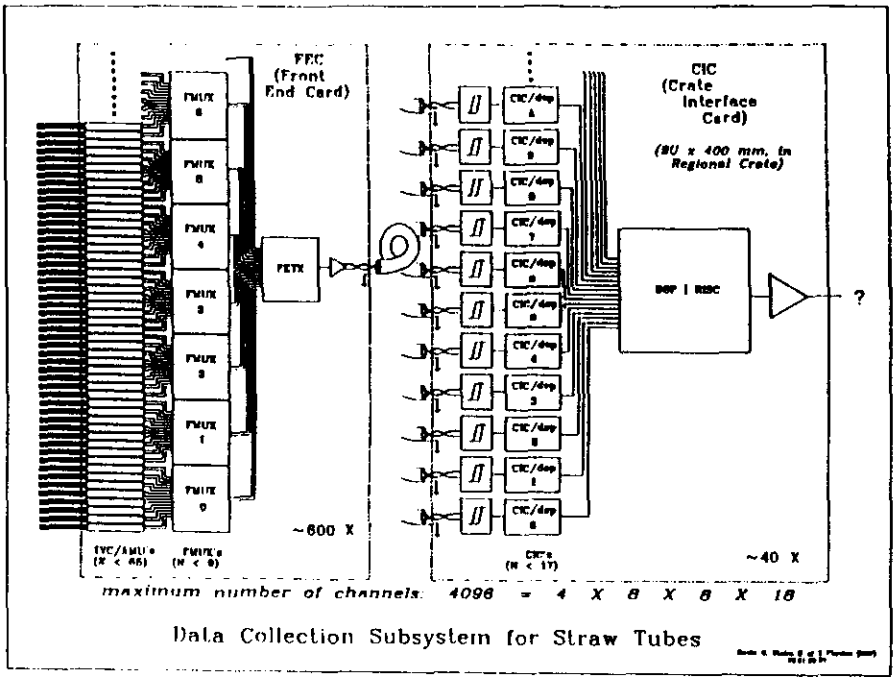

Figure 1: The proposed hardware architecture on the Front End Board (FEB) to read out a set of FE chips. The Front End Multiplexor (FMUX) funnels data from about 8 FE chips, buffers it and passes it to the Front End Transmitter (FETX). The FETX writes the data into the input buffer of a Front End Readout Controller (FERC), which can be located either on the FEB or at the other end of the data link driven by the FETX.

high- speed signals required to drive the FE chips (this includes the L1 and L2 trigger signals and beam clock), as well as the slow control signals required for system reset, initialization and monitoring. The degree of monitoring, the level of sophistication required in the overall system, and the communications network required for this task have not been specified in detail. This note is intended to promote further dicussion and more detailed system design.

2 System Level Considerations

The error detection capabilities of the overall system will determine the ease by which the system is prototyped, commissioned and maintained. We therefore believe it necessary to lay out an overall framework for how specific status or error conditions are detected by the FE readout, data collection circuits and data acquisition system, how they are reported, and what processes are responsible for reporting these error conditions to the physicists monitoring the experiment. We view the overall monitoring and error detection system as consisting of three separate functions: 1) error detection, 2) status monitoring and 3) error reporting.

We assume that a *monitoring task* will exist for each subsystem in the detector, and that these monitoring tasks will communicate with an overall supervisory task that is used by the

physicists to operate and control the detector. The DCC's will be responsible for communicating with this monitoring task over either a slow control network, or using the normal data path. In either case, the primary role of the DCC will be to monitor and report status conditions to the monitoring task.

3 Types of Status and Error Conditions

It is useful to discriminate between different categories of status and error conditions.

3.0.1 Monitoring

The DCC system must be able to perform some low-level monitoring of the behaviour of components that comprise it. Examples of this may be 1) the occupancy of individual channels (to allow the detection of dead or hot channels) and 2) the average time required to readout an event. Such information could be collected locally by each DCC and occassionally read out by a monitoring task.

3.0.2 Warnings

We define *warnings* to be those error conditions that taken individually will have little if any effect on the overall performance of the detector. For example, in the case of the tracking systems, where one has of order 10^5 individual channels of electronics, the failure of any single channel could probably be considered a "soft" error condition that should be noted by the DCC, but would not require the DCC to immediately report it to some monitoring task.

3.0.3 Alarms

The next level of error conditions are what we consider *alarms*, *i.e.* conditions that warrant immediate reporting, but do not automatically halt operation of the system. For example, a readout failure that affects an entire straw-tube module (of order 200 channels) or a large number of channels may be handled in this way. The error detection and monitoring would probably be very similar for this class of errors and the preceding one, but the error reporting would probably differ. These sorts of conditions should be reported immediately to the monitoring process. The DCC may continue normal data taking.

3.0.4 Severe Errors

We define a *severe error* to be a situation where the DCC encounters a condition that requires the immediate notification of the monitoring task and a cessation of data-taking activity. An example of such a situation is the case where the DCC system detects an invalid control signal

or the absence of a signal like the bean clock. In such cases, the DCC must still be responsive to control signals (such as a global reset), but will refuse to acquire data from the FE chips.

4 DCC Communication

Many of the error or status conditions detectable by the DCC will be a result of information generated by the FE chip. To avoid the need for a second output data path from each FE chip, the format used for data exchange between the FE chip and DCC will consist of $\sim$ 6 byte packets, with each packet containing a flag field identifying the type of information in the packet. The DCC will intercept those data packets that contain error or monitoring information generated by the FE chip, and will strip them from the data stream. The error/status information will then be interpreted by the DCC, which will respond accordingly.

A similar mechanism will be used by the DCC to generate a higher level of error/status information. The DCC generates normal data in data *blocks*, where each event block consists of all the data associated with a specific L2 trigger from the channels read out by the DCC. Any error/status information could be inserted into the normal data stream as a separate *status block*, either on demand by the monitoring task (using the slow control link), on a regular basis without a specific request, or immediately when an alarm or severe error is detected. One could also consider making use of the slow control network to communicate between the monitoring task and the DCC, but the number of DCC's in the system (of order 10^3) could cause a severe bottleneck on what is considered a relatively low-bandwidth network.

The primary disadvantages with using the normal data stream for communication of status information are

1. the possibility of a failure in the link itself (detected perhaps as an output buffer overflow in the DCC), and
2. that the monitoring task must now have access to the datastream to be able to intercept the status blocks.

However, the first problem can be easily solved by allowing the DCC to signal the monitoring task of this condition, whereas the second can only be addressed as the overall DAQ architecture is developed.

5 Categories of Status and Error Conditions

There are a variety of status and error conditions that the DCC will have to detect and monitor or report. We can categorize them as follows:

1. status and error conditions initially detected by the FE circuits,

2. failure in FE-DCC communication detected by the DCC,
3. conditions detected by the DCC (for example, invalid data),
4. errors in communications protocols with high-speed or slow control system, and
5. failure to communicate with DAQ system.

5.1 Errors Detected by the Front End Electronics

There are a number of error conditions that may occur on the FE chip that in principle can be detected by the FE chip:

1. L1 buffer overflow—the condition where the buffer used to store data prior to a L1 decision fills and the FE chipd has to reject valid data.
2. L2 buffer overflow—the condition where the buffer used to store data awaiting a L2 decision fills and the FE chip has to reject valid data associated with triggers passing the L1 trigger.
3. Readout synch error, where the interface used to read data off the FE chip has been left in an invalid configuration. This occurs if the readout of the data associated with a hit has not completed (a number of data cycles are required to read all the hit information).
4. An L2 synchronization error, where the FE chip is unable to correctly process the next L2 decision, either because the chip is still digitizing data from a previous L2 decision or the buffer used to hold subsequent L2 decisions is filled.
5. L1 counter synch error, where the FE chip detects that the contents of the L1WR register is invalid.
6. L2 counter synch error, where the FE chip detects that the contents of the L1RA register is invalid.

We propose that these error conditions be signalled by the FE chip by having special "data" added to the output buffer used to special error information. A flag in each datum identifies whether the data is hit information or status information.

5.2 Errors in FE-DCC Communication

Errors in the transmission of information between the FE and DCC will most easily be detected by a simple checksum, CRC check or other modest protocol. Assuming that the error rate is low, it is sufficient to only detect the occurence of a data transmission error at the DCC and monitor the rate of such errors. We do not believe it necessary to develop a more elaborate protocol that requires the overhead of a handshake between the FE and DCC.

5.3 Errors Detected by the DCC System

The DCC is responsible for collecting all the information associated with a given L2 trigger from a FEB together into a block of data that is then presented to the DAQ system. Thus, it can validate the integrity of the data as it is collected. The types of validation checks it can perform are:

1. The appearance of an invalid checksum or CRC check.
2. The appearance of an invalid trigger ID (either L1 or L2).
3. The use of an invalid channel address or chip ID.
4. Verification of the size of the event, or the number of hits/channel (indication of one or more "hot" channels or a failure in the FE-DCC data path).
5. The absence of data from a given channel (indicating a dead channel).

Most of these failure modes may generate only warning conditions. In these cases, the DCC would monitor and store these error conditions, and would only provide this information to the monitoring task when requested. However, the DCC probably should be programmed to generate a severe error if the number or rate of these errors exceeds a certain threshold.

The DCC also must be robust enough to detect other types of error conditions internal to the circuit itself. Examples of these errors are:

1. Buffer or memory overflows at either input or output.
2. The presence of "stale" data in the DCC.

Some of these events may generate warning conditions, while others may generate severe errors.

The DCC may also be required to perform some level of monitoring or manipulation of the data itself. For example, it may be desirable for the DCC to monitor mean drift times, look for threshold shifts, etc. These will clearly depend on the detailed design of the system and readout and should be kept in mind as the design of the DCC evolves. The advantage of doing this form of data manipulation at this level is that it is done at a point where the net bandwidth through any one system is still low enough that a modest processor will have some CPU power available to perform such tasks.

5.4 Errors in DCC-DAQ Communication

Errors in the communication between the DCC and DAQ can also be easily detected by employing some form of data validity check in the data transmission format (checksum, CRC, etc.). The best choice presumably depends on the detailed architectural choice for the hardware at the DCC-DAQ interface. At this level, it may be useful to employ a layered protocol for data transmission to recover from occasional errors in data transmission.

6 Summary

The above specification would require the DCC to have a certain amount of intelligence and flexibility. It almost certainly has to have some level of programmibility though it is not clear that a general purpose CPU is required to perform any of these functions. Given the relatively low data flow through each DCC system, it is possible that an inexpensive DSP would be adequate.

This specification also does not place strong contraints on the location of this intelligence. It appears possible to have most of the data manipulation and validation take place either on the FEB or at the other end of the data path in a crate, where one could more conveniently locate a high-level digital device.

References

[1] G. Stairs and P. K. Sinervo, *Design Options for an SSC Wire Chamber Data Collection System*, UOFT-DCC-01, Oct. 1991 (unpublished).

[2] G. Stairs, *Front End Collector Bus Design Description*, UOFT-DCC-03, Jan. 1992 (unpublished).