Data transmission protocol for the readout of Silicon Drift Detector.

Gianluca Alberici Gianni Mazza Paolo De Remigis

August 7, 1998

1 Summary

An Interface Board will be located at both ends of each Silicon Drift Detector ladder, and it has three fundamental goals. It is designed to provide the compression of the digital data stream incoming from the bank of detectors (reducing the amount of data to allow the writing on disk), the hardware implementation of the transmission protocol (managing data coming from eight half detectors to be sent on one optical fibre) and the implementation of the physical interface towards the Detector Data Link (DDL) (using the data link transmission protocol).

2 Brief Analysis of the System

The Silicon Drift Detector (SDD) layers are two coaxial barrels composed by ladders mounted one beside the other [1]. The silicon detectors are distributed along the ladder, and on each ladder end there is an Interface Board that hosts a DDL with its optical fibre. Each ladder manages eight silicon detectors. The detectors are symmetrical devices, with drift opposite to the middle line. Each half detector acts as an independent sensor. The half detector has 256 anodes and produces an analog signal every particle hit; this signal is a current whose amplitude depends on the energy released in the detector. The set of 256 signals coming out from each half detector represents a description of an event in two dimensions (the anode direction and the charge drift direction, i.e. time). Therefore, each Interface Board serves eight half detectors with more than 2000 channels.
The output current is first converted in a voltage and then sampled and held in an analog memory 256 samples deep. When the trigger is present this memory should be read out, by the Interface Board, in 200μs. A double-event buffering mechanism is employed to derandomize the events. In this way the read out can be performed in 2ms [2]. The analog signals are converted by an Analog to

Digital Converter (ADC) with an 8bit resolution that produces a throughput equal to:

$$Anode \cdot \frac{Sample}{Anode} \cdot \frac{Bit}{Sample}$$

$$256 \cdot 256 \cdot 8 = 64kByte$$

And then:

$$\frac{64kBytes}{2ms} = 32MByte/s$$

For modularity reasons each half detector has its 8bit wide data bus; at the fixed 40MHz clock frequency, the resulting 40MByte/s bandwidth is sufficient even with the overhead (less than 10%) due to the handshake.
In summary there are 8 half-detectors producing a throughput equal to 35Mbyte/s each for a total transfer rate of the input bus equal to 280MByte/s or around 2.2Gbit/s. The optical fibre bandwidth (of DDL) is 800 Mbit/s: so it is necessary to compress the data at least three times to match the two throughputs [3]. This compression ratio can be easily obtained with a relatively simple algorithm, but the critical constrain in the whole system is the final writing of the data. The global amount of data is 30 times larger than the space on disk which at present is allocated for the SDD's.
Leaving aside implementation details, the system implemented on the Interface Board has to provide the following features:

- the buffering of the eight streams incoming from detectors;
- hardware compression (compression parameter can be changed through the slow control system; it is also possible to read the raw data from detectors without any processing);
- multiplexing of the eight channels in a 32bit word stream to send to the DDL using the relative protocol.

The figure 1 shows a simple block diagram of our subsystem.

3 Bandwidth allocation

The eight input streams, of 8bit, coming from the detectors along the ladder are basically asynchronous: because of the compression processing they cannot be considered as eight homogeneous streams to send to the fibre. There are two well known standard approaches in allocating bandwidth:

- Dynamic allocation: when a channel has no data to send it frees its band which is allocated to another channel.
- Static allocation: each channel shares a fixed part of the whole bandwidth and when it has no data to send it sends dummy codes on the line, i. e. (in any case) it always uses its assigned bandwidth.

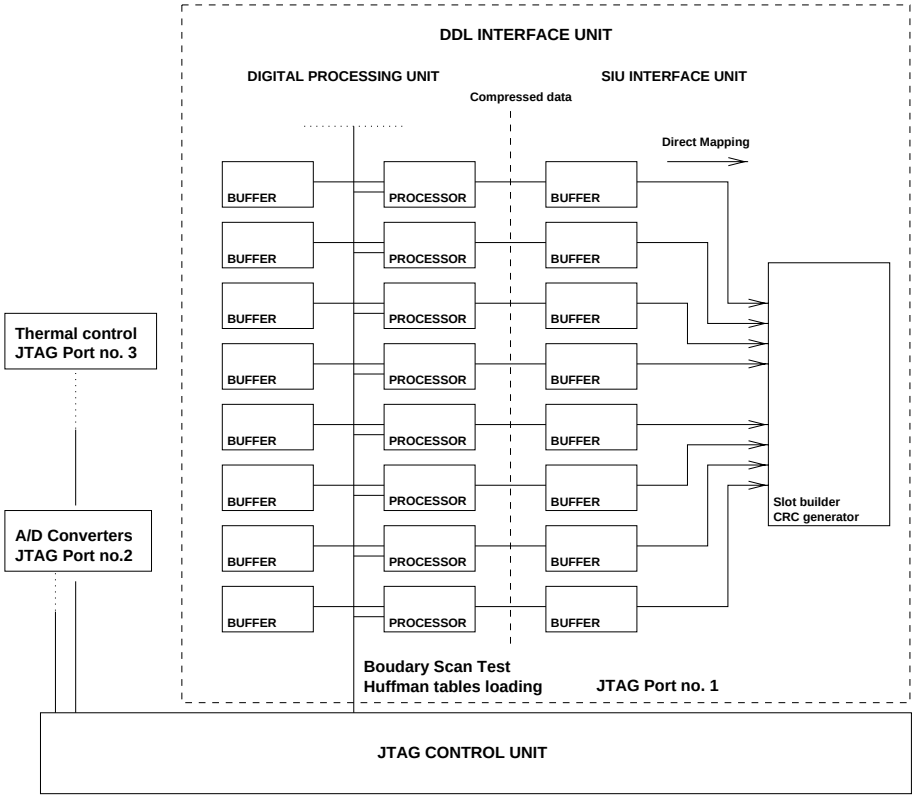

Figure 1: Block diagram of the described system.

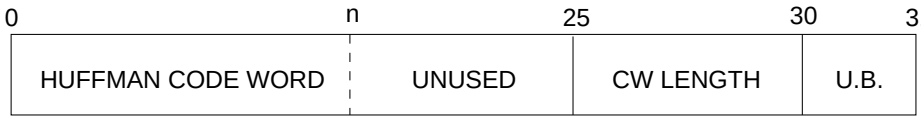

Figure 2: Bit mask of the compressors output word.

In the typical case, when the amounts of data coming from the detectors are roughly the same, the statistical multiplexing (or dynamic allocation) will become static multiplexing: for this reason and for its simplicity the static allocation has been chosen. To manage the little differences of data quantity between channels, the unused bandwidth will be filled with dummy data that will be discarded at destination after the optical fibre.

4 Data format out of the compression blocks

The output word format of the Compressor chip is crucial, since it is the starting point to get the 32bit wide word to send to the DDL module. A Huffman code with variable length is applied to the 8bit wide samples coming from the detectors. To get a better compression coefficient, the system foresees the possibility to apply also other algorithms (as threshold, tolerance, etc.) even though they imply information loss. These codes, after the compression algorithm is applied, are variable size words with a maximum length of 26bit. To the output format of the Compressor chip has been assigned a 32bit wide word: the structure is shown in figure 2. A transmission protocol has been defined for the DDL interface which should provide easily both error correction and static bandwidth allocation using dummy data. This data transmission protocol is described below.

One of the additional compression schemes that is implemented in the compression block is the Run Length coding for the symbol 'zero': this symbol is the most probable one for the given input source and the logic threshold scheme raises many times its occurrence. There are a lot of very long 'zero' streams in the precoded input data. These streams have been coded with the 'zero' symbol from the Huffman table followed by an 8bit wide field specifying the Run Length [4]. These specific words are transparently managed by the following stages because this sequence is represented as a 10bit wide Huffman word which will be interpreted in a particular manner from the destination decoder only. Anyway, while the compressor block calculates the run length dimension counting the 'zero' symbols, the significant data flow towards the next interface unit could slow down. Since the system is working as a pipeline, the data input rate must be equal to the data output rate. To maintain the synchronisation with the rest of the pipeline during this phase, the processor block will generate

a dummy codes stream setting a flag named the Underrun Bit. The idea is that the following stages will discard these selected code words. A buffer will help to maintain a more continuous data flow towards the DDL. In case that this buffer went in underrun state the transmission unit will generate dummy code words to send to the DDL in order to implement the static allocation described below.

5 The transmission unit buffer

The transmission unit needs to be equipped with a hysteresis buffer: in the case of a long unlikely source sequences and, as consequence, long output words (up to 26bit wide words) the instantaneous throughput could become very high. Given that the output clock, used also for DDL, will run at lower frequency than the input clock (at least a number of times equal to the expected compression ratio), a handshake will be necessary between the processing unit output and the DDL controller input to slow down the data flow from the compressor block in order to manage this kind of situation.

6 Transmission protocol proposal

The currently defined protocol has a 6.3% redundancy which allows both the multiplexing of the symbol of the eight input channels and the indication of the stream end for each channel. Furthermore this protocol allows the implementation of a Cyclic Redudancy Code (CRC) scheme on 8 bit. The decision of providing CRC on this system is suggested by the particular encoding scheme, and architectural structure. The implementation of the CRC comes without big overhead because, as it will be shown later, the necessary bits are encapsulated in the words used to signalling the stream end for each channel.
Codewords extracted from the output of the eight compressors, that are 32bit wide, are packed in fields of 4bit, then the eight channels are directly mapped in a 32bit wide word allocating eight slice of 4bit per channel. This will require a barrel shifter architecture as shown in figure 3. What is transmitted on optical fibre, via the DDL, is a sequence of slots made of 32 words of 32bit each. After that it will be sent two control words which will provide indication of both the last valid bit for each channel and the CRC implementation.

6.1 Indication of last valid bit

In the case of a channel with less data to send(the DDL controller buffer gets empty because, e.g. that channel has very long 'zero' runs that are compressed very well) it is necessary a correct indication of the last valid bit of each channel

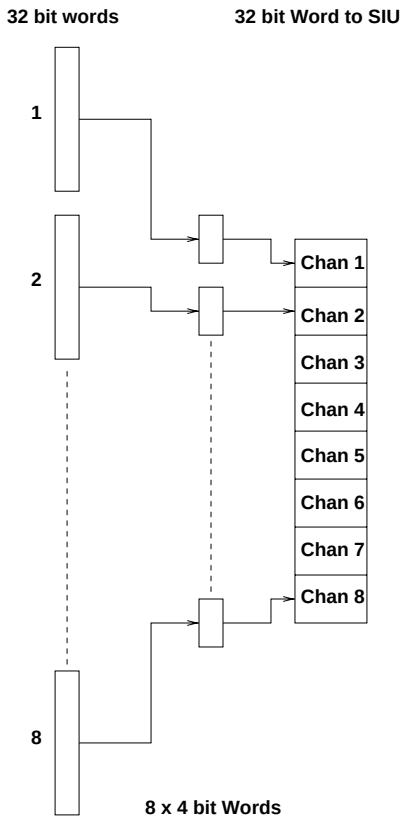

Figure 3: Building a 32 bit word from 8 channels.

in the slot under transmission. This is done specifying two fields for each of the eight channels:

- a pointer to the 32bit wide word of the current slot where the channel presents its stream end.
- a pointer to the last valid bit in the 4bit wide field of the word previously selected by the first pointer.

Using this solution 5bit are needed to index one word over 32, and 2bit to index one bit over 4. In summary this scheme needs 7bit in order to locate exactly the last valid bit of each stream, in a specific channel. Globally the structure has to reserve 56bit to select the last valid bit, for each of the eight channels.
If, for some reason, a channel stops to send data in a slot then it cannot start again transmission in the same slot but it has to wait for the next one (a buffer is required).
Each slot of (32+2) word contains eight channel related streams: if the stream is fully used it transports $4 \cdot 32 = 128bit$.

6.2 CRC implementation

The remaining 8bit in the two control words can be used for error recovery (CRC) and/or for future applications. Currently the choice of the specific polynomial, for the CRC implementation, has not been done yet. About the error recovery it must be investigated whether to implement a single CRC word for each channel or to generate one CRC for the stream resulting from concatenation of the eight channels.

7 JTAG protocol

The Interface Board design has to contain test structures (design for testability) as well as Boundary Scan Registers etc.... Test procedures, as slow control, will be implemented according to the JTAG protocol. The design of the JTAG control unit and of the JTAG interfaces is under development.

8 References.

[1] Alice collaboration. Technical proposal. CERN.LHCC 95.71.

[2] G. C. Bonazzola et al.. A possible SDD readout scheme for the Inner Tracking System of the Alice experiment. Alice.97.36.

[3] Cern ECP.ALD, RMKI RFFO.DB. Alice Detector Data Link. Alice.96.42.

[4] D. Cavagnino et al.. Data compression for the Alice Silicon Drift Detector. To be issued as Alice internal note.