SDC-92-241

SDC
SOLENOIDAL DETECTOR NOTES

FRONT END COLLECTOR BUS (FECbus) DESIGN DESCRIPTION

January 20, 1992

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

Front End Collector Bus (FECbus) Design Description

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

January 20, 1992

Abstract

A design is proposed for the connection of front end electronics modules (e.g., the U. Penn TVC/AMU) to the Data Collection system on the Front End Boards of a straw tube system. The design employs ASICs for level shifting, data buffering and multiplexing of data signals in order to produce a single high speed data stream from up to 256 TVC/AMU channels.

1 Introduction

The FECbus connects the analog/digital front end chips (FEC's) to the next upstream data processors. The output of the FEC's is assumed to be a 24 bit datum organized as a nibble (4 bit) wide serial stream with a simple handshake, producing data at a relatively slow throughput (eg., 960 events/channel/s), and a higher burst rate (eg., 24 bits/500ns). The upstream processor is assumed to be a large word device (eg., 24 or 32 bits word size) capable of accepting data at a large burst rate (eg., 40Mword/s). The FECbus provides the synchronization logic for the FEC's data transfer, low capacitance, level-shifting buffers for the FEC's to isolate them from the bus, and a small amount (10 words or more) of data buffering capacity. The FEC's are organized as up to 50 chips on a front end card, with four channels per chip, or a total of up to 200 channels per card. The data rate out of a card is therefore order of 10.7×10^3 events/s, or order of 6×10^6 bits/s. This data rate is consistent with a single optical fibre channel, or a coaxial copper wire channel.

2 Data Formats

The incoming data word consists of 24 bits, comprising flags, location tag, T1 tag, T1 crossing clock, a crossing clock offset, and fine timing with error detect/correct. The outgoing word has inserted an additional channel ID field which identifies channels over a whole card of up to 256 channels.

Bits	Function
0-1	Bunch Counter
2-9	Level 2 ID
10-11	Channel ID (Front end chip)
12-15	Fine timing data error check
16-23	Fine timing data

Table 1: FEC incoming data word organization

Bits	Function
0-1	Bunch Counter
2-9	Level 2 ID
10-17	Channel ID (Front end card)
18-21	Fine timing data error check
22-29	Fine timing data

Table 2: FEC outgoing data word organization

3 Buffer Chip Organization

Each FEC requires four data signals, plus two handshakes. Since the signals are differential, that makes a total of 12 pins. The output bus is ~32 bits wide with two handshakes. The total pin count is therefore 68 + 12n + m, where n is the number of FEC's per buffer chip and m is the number of power/ground pins.

Number of Buffer Chip Signal Pins	Number of FEC's / Buffer Chip
332	22
300	19
257	15
220	12
178	9
155	7
138	5
108	3

Table 3: Fujitsu AU Series CMOS gate array signal pinouts

Table 3 shows that CMOS gate arrays are available with sufficient density to permit on card multiplexing to be accomplished on a small number of chips. This is desirable for reasons of power dissipation, noise and card real estate.

Assume 8 (or more) Front end MUtipleXer (FMUX) chips and 1 Front End board data Transmitter (FETX).

The FMUX's must buffer the data lines from up to 8 FEC's, provide intermediate storage in the form of a FIFO, insert the correct channel ID extension, execute the channel transmit selection algorithm, and transmit selected data on request.

The FETX performs the output interface function at the card boundary. It signals the output interface when data is ready, determines when transmission is permitted, transmits data, and so forth. It also performs all or part of the physical signal translation and buffering required by the transport medium (eg., coaxial wire or optical fibre).

If the FETX is a processor like a DSP, then it is conceivable for it to do some real processing. Possible tasks include low level monitoring, calibration, download of calibration constants, startup configuration, and event packet formation. Note that some of these functions imply two-way communication.

4 Front End Chip Selection

If an FEC has data which is ready to be transmitted, it will raise a flag (eg., TVC/AMU Cdata_present). This means that one of its four channels has data ready. This data event will normally happen synchronously with a number of channels together (expected 9% occupancy == expected 31% FEC occupancy). Approximately two thirds of the FEC's are expected to be vacant in any given physics event. If there are approximately 200 channels on the card, then each physics event will comprise something like 16 FEC's and 18 data words. If the transfer rate is 0.5 Mword/s (which is consistent with a 15 Mbps output channel burst rate), then each event will take about 36×10^{-6} s to clear. Since the event rate is expected to be about $11 * 10^3$ events/s, this implies that the data channel will be idle for about 2/3 of the time. However, bursts may occupy the full bandwidth. In order to provide as much data bandwidth as possible, is desirable to have a scheduling algorithm which is fair and has low time and logic overhead.

As each FEC raises its data flag, the data will be transferred to a FIFO on the corresponding FMUX. This FIFO will provide additional tolerance of data rate fluctuations, and will bridge the FMUX readout delay.

Let the FETX generate a signal which is propagated to each FEC channel in turn throughout the system, and which returns to the FETX.

The rising edge of this signal signifies a request that the receiving FIFO prepare to transmit data. If there is data ready, the FIFO changes state to *ready*. If there is no data ready, the FIFO changes state to *not_ready* and the rising edge is passed to the next FIFO.

The falling edge of this signal signifies permission to transmit data. If the falling edge reaches a FIFO in the *not_ready* state, the edge is passed through immediately. If the falling edge reaches a FIFO in the *ready* state, the leading edge is passed to the next FIFO, and the selected FIFO begins to execute the transfer protocol. When the transfer is complete, the falling edge is passed through, and the FIFO switches to *not_ready* state.

On reset, all channels are in the *not_ready* state.

The possessor of permission to transmit can hold it for a single transfer, or for as many as it wishes. However, it must pass permission whenever it runs out of data. The transmission strategy can be switched according to the state of some control register.

If no FIFO's have data, then the rising edge will propagate back to the FETX after some number of gate delays. If the edge were then permitted to pass through freely, the system would constitute a ring oscillator, which would be wasteful of power and would generate unnecessary noise. Accordingly, the rising edge will be delayed until the lapse of a timing counter before being permitted to proceed.

The falling edge could conceivably catch up to the rising edge. In this case, the falling edge will be delayed a standard delay after the rising edge. This delay will be some number of gate delays.

One of the advantages of this scheme is that there is only one high speed signal, and therefore relatively slight noise.

5 Data Switching

The data from each FIFO is routed through a switching network to the FETX. Since there is only one valid data route at any time, the selection of the switch states can be accomplished by a single asserted data line originating at the selected FMUX. What happens next depends on the most efficient implementation in the given technology. There are three possibilities: bussing, switching and hybrid.

Bussed Interconnection

Each FIFOresides on a common data bus. Assertion of the select for any given FIFO enables its bus drivers. When not selected, the bus drivers are in a high impedance state, and constitute a small but possibly significant capacitative load on the bus. The faster the bus is required to run, the more significant this load.

Switched Interconnection

Each FIFO stands as a terminal node in a tree the root of which is at the FETX. The tree consists of data pathways connected in pairs by switches to the next level. and so on. The selected FIFO asserts its control line which traverses the tree from terminal to root. All junctions in the traverse path are switched to match.

Hybrid

Bus together as many FIFO's as bus loading permits. Connect the busses into

trees with similar busses at all of the switching nodes.

6 Level Shifting (TVC/AMU)

The logic I/O levels of this chip {are I will be} low voltage differential. The FMUXs conform to this interface. The FETX may or may not, according to what type of chip it is. For example, if it is a standard DSP, it will have industry standard pin configurations. If it is an ASIC, it may have differential receivers and drivers for the FMUXs, and industry standard I/O for the transmit side.

7 Off Card Data Channel

This structure is not yet well defined. The following description may be considered a preliminary proposal.

The data rate in the output channel is consistent with a single moderate speed channel, such as a single optical fibre or coaxial cable. There are a number of industry signalling standards which can be applied.

In the case of optical fibre transmitters / receivers, there are proprietary devices available from firms such as Burr-Brown, Mototrola and Hewlett-Packard which could be used, and which can be coupled directly to available VMEbus I/O cards, for example. In this case, the data format is dictated by the logic supplied on the VMEbus card.

Similar de facto proprietary and more general standards exist for coaxial copper data links.

In both cases above, the choice is governed by the following considerations:

i) Availability of hardware for both ends.
ii) Suitability of the signalling protocol for execution at the front end.
iii) Noise coupling and generation at the front end.
iv) DC decoupling over long links.
v) Data noise rejection and link integrity.
vi) Power requirement at the front end.
vii) Radiation tolerance (integrated flux to 1Mrad)

2 Counter Stream Communication

The transmission of control words to the FEC's and to their corresponding FMUXs can be accomplished by making the forward data structures bi-directional, or by providing an alternate pathway. If the data rate required is not great, then a single daisy-chain serial channel will suffice, and will entail less general complication than generalizing the main data pathways. This implies the existence of some modest local intelligence to operate

this channel. This could be the DSP referred to elsewhere, or it could be a more modest controller not connected to the general data stream.

On the other hand, the off card data channel may very well provide duplex communication. If this is the case, then the return channel can be used for this purpose.

Stairs File name 92011401.DOC