BABAR IFR TDC Board (*ITB*): system design

Version 1.1

12 december 1997

G. Crosetti, S. Minutoli, E. Robutti

I.N.F.N. Genova

1. Introduction

TDC readout of the IFR will be used during *BABAR* data taking to exploit the excellent time resolution of the RPCs [1,2]. *IFR TDC Boards* (*ITBs*) will be placed in the 8 IFR DAQ crates (3 or 4 per crate, depending of the part of the detector to be read) next to the other DAQ boards (*IFB* and *ICB*).

Each board will read 96 input signals coming from the *FAST-OR* output of the RPCs Front End Cards (*FECs*). Communication with the *BABAR* data acquisition system will take place accordingly to the general specifications in [3] through the *CK60*, *DIN* and *DOUT* lines on the IFR custom backplane (*PDB*).

2. General board design

The board will have the dimensions of a single-slot 6U standard VME card. Except for the input signals, which are of differential ECL type, the whole board is designed for working with TTL levels.

The input signals will be provided through 3 68-pin front connectors, while the power supplies and control signal will come from the PDB through 2 standard VME connectors (*J*1 and *J*2); time measurements are accomplished by means of 3 general purpose TDC chips controlled via the JTAG protocol; all the logic related to command decoding, data transmission and board configuration will be managed by 2 FPGA devices; the event buffer will consist of a 36-bit-wide FIFO; an additional single bit FIFO will temporarily store the TDC configuration data. Fig. 1 represents a block diagram of the ITB.

All details concerning accepted commands and data format are already described in [2].

3. Clock distribution

The 60 MHz clock is supplied to the board through the CK60 line on the backplane[*]. The signal is already of TTL type, so it has not to be converted. Nevertheless, given the high frequency, a good quality is necessary for a reliable behaviour of the various components on the board: this is especially true for the TDC chips, which require a signal with jitter lower than 200 ps and duty cycle between 45% and 55%.

For this reason a CY7B9910 clock buffer will be used on the board. This device drives up to 8 output lines, re-shaping the input clock signal by means of an internal PLL. The duty cycle of the output signal is 50%, its delay is zero and its jitter lower than 200 ps. Three of the output lines will feed the TDCs, two of them will feed the FPGAs and other two will feed the event buffer FIFO.

[*] This line is driven by the *IFR Crate Controller* (*ICC*) which distributes the 60 MHz *BABAR* clock signal coming through the optical link from the ROM.

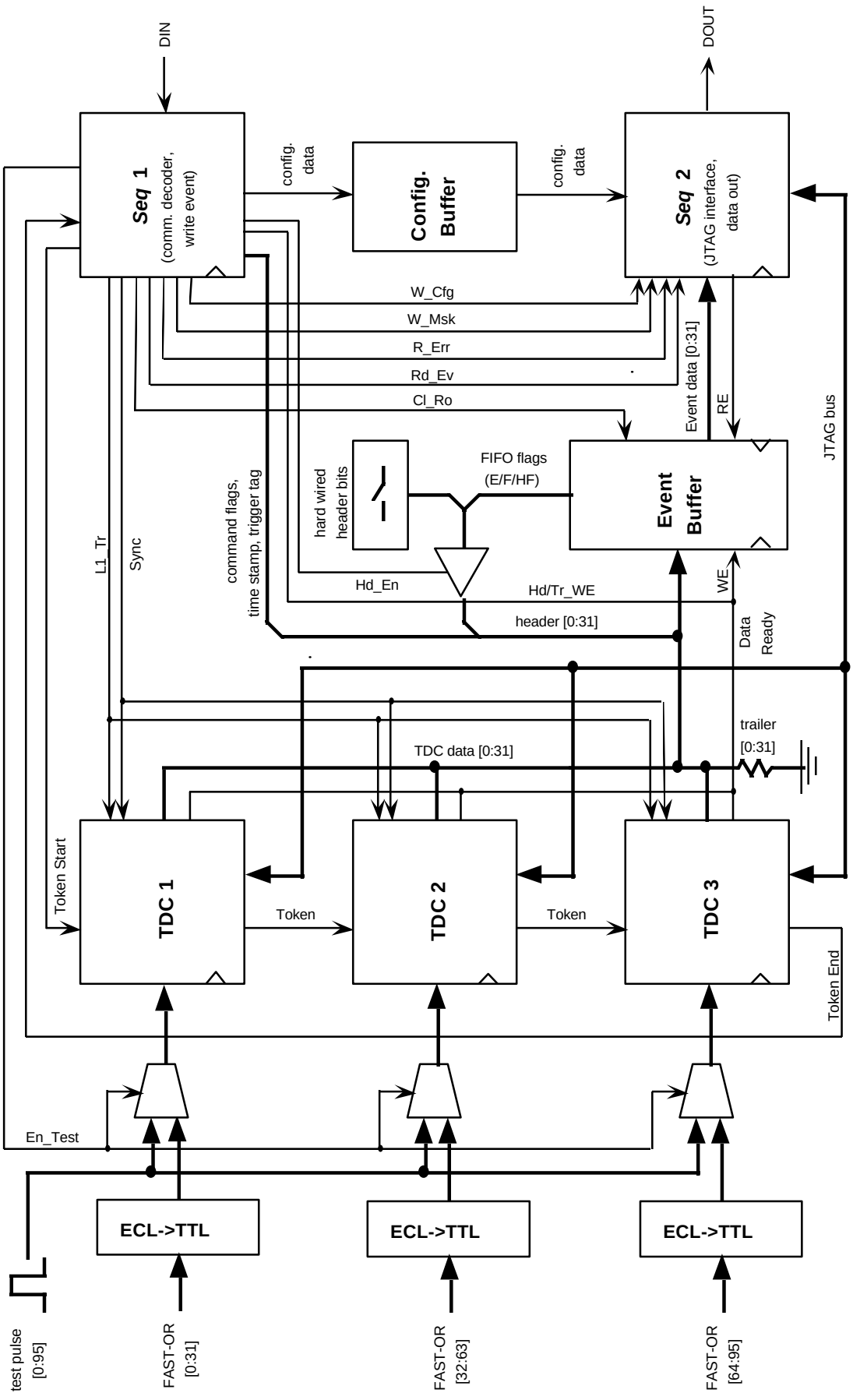

Figure 1: *ITB* Block Diagram

4. Input section

Input signals will come from the FECs as 96 differential ECL pairs. They are transmitted from the detector using 6 flat cables (34 wires each) and fed into the board through 3 68-pin connectors.

A protection is applied to the input against spikes before conversion to TTL levels. This is realized in the same way as for the IFB and ICB [4]: an SP720 "transient diverter" (with protection resistors on the input) limits the tension values between two reference levels, in this case chosen to be VEE (–5.0 V) and GND. Conversion to TTL is made by 16 6-channels ECL $\rightarrow$ TTL translators (F100325).

An internal test pulse can be sent to the TDCs input to check for dead or bad channels. The pulse is generated by the command decoder in response to a *Calibration Strobe 2* command (see [2]) and fanned-out (to 12 pulses) by a 74FCT16244T buffer/ line driver. The selection between FEC signals and test pulses is made by means of 6 74CBT16233 32-bit multiplexer; each of these gets the selection input from a line also driven from the command decoder and fanned out (twice) by the same driver used for the pulse.

5. TDC

A 32-channels general purpose TDC chip developed at CERN for LHC experiments has been chosen for the ITB for the reasons already explained in [2]. There its main features are also listed, while a complete description of the chip characteristics, including inner architecture and operational modes can be found in [5].

Each board will be equipped with 3 such chips. They will always be used in '*trigger matching*' mode (as opposed to '*START/STOP*' mode), i.e. input pulses will be continuously accepted[*] and their time recorded until those lying within a proper, selectable time window are stored in the readout buffer on occurence of a trigger signal.

The TDC contains several internal registers which are used to configure operational mode and to perform tests or status check. They are accessed through a *Test Access Port* (*TAP)* complying to IEEE Standard 1149.1 for boundary scan [6]. This consists of 5 pins (4 input, 1 output) [†] which receive/send signals from/to a dedicated bus according to the rules specified by the *JTAG* (*Joint Test Action Group*) protocol. Three of these registers will be accessed by subsystem commands: the setup register, which is used to set the configuration switches and parameters and the offset values, the control register, containing the channel enable/disable mask, and the status register, which returns the error flags status. These operations will be performed in a daisy-chain fashion, by connecting the JTAG *TDO* pin of a chip to the *TDI* pin of the following one.

---

[*] Provided they match the chip specifications for double pulse resolution (> 15 ns) and hit rate/channel (< 750 MHz), which are far beyond the system requirements.

[†] Namely, *TRST* (*Test Reset*), *TCK* (*Test Clock*), *TMS* (*Test Mode Select*), *TDI* (*Test Data In*), *TDO* (*Test Data Out*).

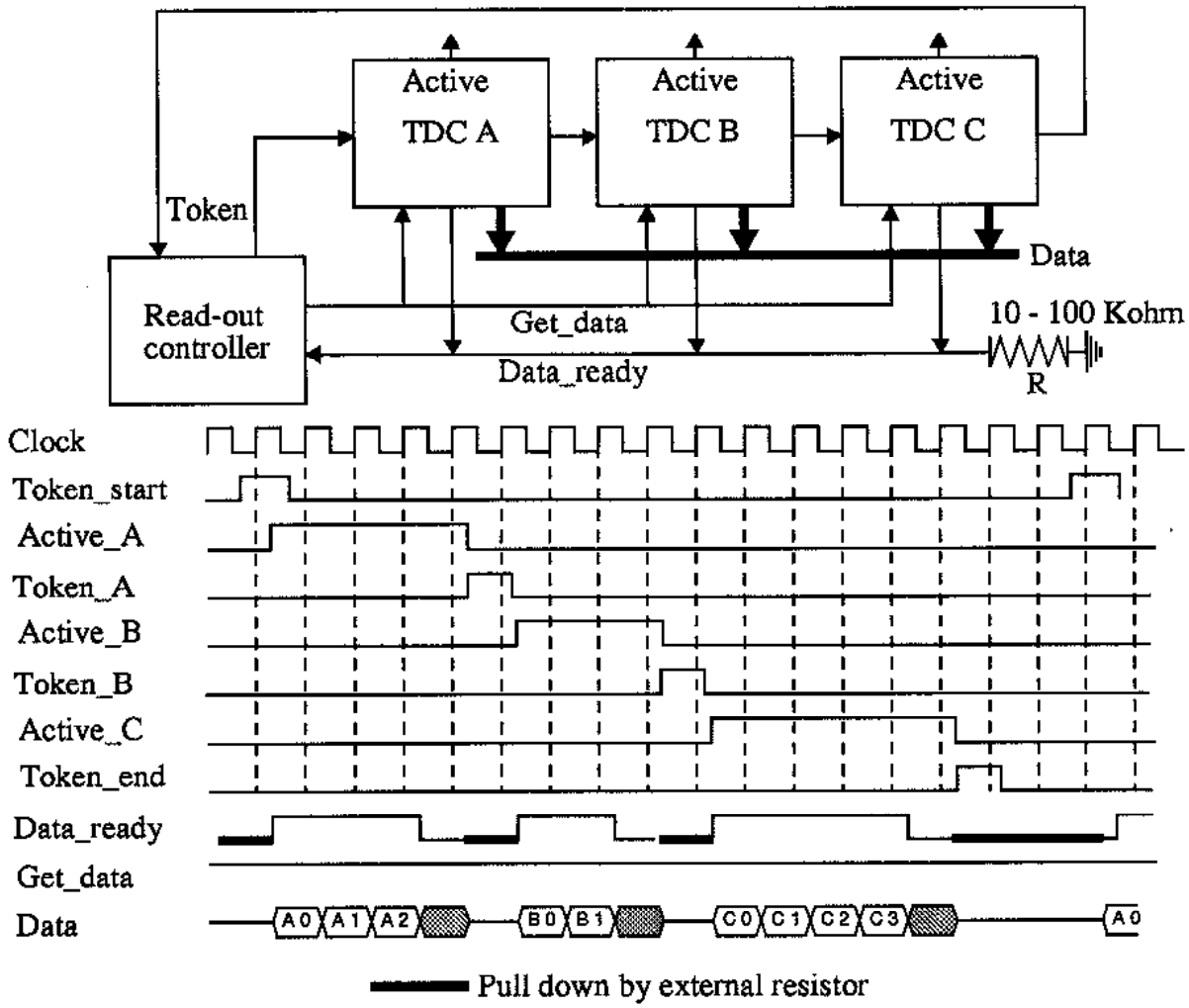

Figure 2: token based read-out protocol.

Each of the 32 input channels can be assigned a different offset value: the appropriate values will be determined by calibration to account for different cable length, speed of the FECs electronics etc.. A global and a trigger offset will also be set to properly line up times with the *Sync* command and to select the correct latency/jitter window, respectively. These operations are performed while writing the chip setup configuration by means of the JTAG protocol.

Data will be read out from the TDC and written to the event buffer immediately following an *L*1 *Trigger*. They consist of a 21-bit time record and a 5-bit channel ID, plus an error bit[*]. The read-out sequence will make use of a '*token ring*' protocol, by which the 3 chips can be daisy-chained and polled sequentially (see Fig. 2). Data transfer will occur at 60 MHz frequency by using the system clock.

[*] An additional start bit and a 2-bit TDC ID will be added to form the complete 32-bit event word.

6. Logic

Three different tasks can be clearly identified for the logic driving the board operations: (*a*) decoding the commands arriving serially through the DIN line and appropriately distributing data coming with them; (*b*) managing the JTAG protocol with the 3 TDC chips; (*c*) sending out data serially on the DOUT line. A single programmable device will handle (*b*) and (*c*) to limit the components number and to save space (also, this is sensible, as the DOUT line has to be accessed in both cases), while another will handle (*a*).

Two Xilinx FPGA devices of the 3100 family have been chosen to manage the logic because of their dimension (in terms of logic cells) and speed. Both of them are programmed at power-up by means of two serial PROMs of the 17000 series, and are externally clocked by the 60 MHz board clock.

6.1. Command decoder

The first FPGA (*Seq*1) is an XC3142A-2PC84 device, containing 144 CLBs (Configurable Logic Blocks) and featuring up to 74 input/output pins. It gets data coming serially through the DIN line and decodes them as commands in *BABAR* standard format *zsccccaaaaa*[*d…d*] (*z* = leading 0; *s* = start bit; *c* = command; *a* = sub-address; *d* = optional data). The decoding recognizes all *BABAR* run-time commands and the 3 ITB-specific commands (see [2]). For each of the decoded commands the following actions are taken:

- *No Op* (*Op_Code* = x00): the *No Op* flag is raised to be written in the next event header;
- *Clear Readout* (*Op_Code* = x01): a RESET signal is sent to the event buffer to clear its content;
- *Sync* (*Op_Code* = x02): the 8-bit free running counter[*] is reset; a RESET signal is sent to all TDCs: internal coarse time counters are reset, event and trigger buffers are emptied;
- *L1 Trigger Accept* (*Op_Code* = x03): the 8-bit *time stamp* register is latched with the current counter content; the 5-bit *trigger tag* register is written with the content of the sub-address field; a *Trigger* pulse is sent to all TDCs; all header lines are taken to low impedance[†] and a *Write Enable* pulse is sent to the event buffer (after that the *No Op* and *False Command* flags are cleared); a *Token_start* signal is sent to the first TDC in the chain; a *Token_end* signal is waited for and then another *Write Enable* is sent to the event buffer to write the trailer;
- *Read Event* (*Op_Code* = x04): a *Rd_Ev* signal is sent to the 2nd FPGA;

[*] This is an internal counter incrementing its content at each clock rising edge; reset on occurence of a *Sync* command and latch on occurrence of a *L*1 *Trigger* are tuned so that the same time stamp is associated to each event as the one given by the IFB.

[†] In particular, for the 16 driven by the FPGA itself, tri-state output buffers are enabled, while for the other 16 (hard-wired or from the event buffer FIFO) an external buffer is enabled.

- *Calibration Strobe 1* (*Op_Code* = x05): no action is taken;
- *Calibration Strobe 2* (*Op_Code* = x06): an *Enable_Test* signal and a test pulse are sent to the input multiplexers;
- *Write Configuration* (*Op_Code* = x12): 3×128 bits are read serially from the data field of the command and shifted to output to be written on the configuration buffer: a *Write_Enable* and a *Write Clock* signal, sent to the buffer, control the write operation; a *W_Cfg* signal is also sent to the 2nd FPGA to start the JTAG programming sequence;
- *Write Mask* (*Op_Code* = x13): the same as the previous one, save the data size is 3×32 (and a separate, *W_Msk* command is sent to the 2nd FPGA);
- *Read Error* (*Op_Code* = x14): the same as the previous ones, save there are no data (and a separate, *R_Err* command is sent to the 2nd FPGA);
- *False Command* (start bit followed by non-recognized command code): the *False Command* flag is raised to be written in the next event header.

The whole implementation of the logic has been realized starting from schematic design: this makes use of macros from XC3000 (*XACT*) library and, for time critical parts, of "manual" CLB mapping.

6.2. JTAG interface and data output

The second FPGA (*Seq*2) is an XC3160A-2PC84, the same physical size of *Seq*1 but with a higher number of CLBs (244) and a slightly lower number of input/output pins (up to 70). It acts as a JTAG interface between the command decoder and the TDCs, controlling the programming sequences, and gets data from the event buffer on a *Read Event* command to send them out serially on the DOUT line.

On occurence of a *Write Configuration*, *Write Mask* or *Read Event* command, the finite state machine which manages the JTAG protocol is started and the appropriate signals asserted on *TMS* and *TDI* lines, synchronously with the clock activated on the *TCK* line. For write operations, the data just written by *Seq*1 on the configuration buffer are read out and transmitted serially to the *TDI* line. The finite state machine is written in *verilog* language and synthesized for XC3100 components by the *Synergy* tool of *Cadence* software.

As the maximum frequency accepted by the TDC for *TCK* is 20 MHz, the 60 MHz clock timing the FPGA is divided by 4 and the resulting 15 MHz signal used to time all operation involving the JTAG protocol.

Conversely, all logic related to the event readout must work at 60 MHz frequency, and has been designed using schematics to improve speed performances. On occurrence of a *Read Event* command, 32 bit words are read in parallel from the event buffer, stored in an internal shift register and transmitted serially to the DOUT output until a trailer (x00000000) is found. The same DOUT pin is also used for three previous ("JTAG")

commands to output the content of registers being accessed[*].

7. Event and configuration buffers

The event buffer is a 512 × 36 clocked FIFO memory (SN74ACT3631) supporting clock frequencies up to 67 MHz. It uses two separate clock inputs (as well as two separate enable inputs) for read and write operations. The 60 MHz board clock will be used for both of them, with 2 programmable digital delay lines (3D7105) on input to assure a correct data-vs.-clock timing.

The *Read Enable* input will be driven by *Seq*2, while the *Write Enable* will be driven either by the *Hd_Tr_WE* output of *Seq*1[†] (for header and trailer) or by the *Data_Ready* outputs of the TDCs (for data). All of these lines are tri-stated when not used, so there is no conflict.

The 32 data lines also have more than one driving source: it can be the output of either one of the TDCs, the header or the trailer. All TDC data lines are tri-stated when the token is not there, and so are the 16 header bit coming from *Seq*1, except immediately after the decoding of a *L*1 *Trigger* command. The other 16 header lines go through a buffer which is only enabled at the time of header writing. As for the trailer, a simple pull-down on the bus is enough.

As the maximum size of an event is 1 + 32 + 32 + 32 + 1 = 98 words, the event buffer can store at least 5 full events.

The configuration buffer is a 512 × 1 synchronous FIFO (74ACT2229). It also has separate inputs for *Write*/*Read Clock* and *Write*/*Read Enable*: these are controlled by *Seq*1 and *Seq*2 (for *Write* and *Read* respectively); clock frequencies are 60 MHz and 15 MHz respectively, as appropriate for the *DIN* and *TDI* lines.

This FIFO is non-empty only during write operations on the internal TDC registers: it is only used as a temporary store for configuration data. Its size is enough to store the content of the 128 bit setup registers for all 3 TDCs.

[*] Interleaved by filling '1' to match the two different read-out frequency values.
[†] With an additional delay line in-between.

8. References

[1] L. Morelli, M.G. Pia, *Determination of the bunch crossing with the IFR*, *BABAR* note.

[2] G. Crosetti, S. Minutoli, E. Robutti, *IFR TDC Board (ITB): design requirements and specifications*, I.N.F.N. Genova, WWW: http://www.ge.infn.it/babar/ITB/ITB_req.ps.

[3] *BABAR* DAQ Group, *BABAR note* 281

[4] V. Masone, L. Parascandolo, P. Parascandolo, *A transient voltage protection network for ECL data transmission*, I.N.F.N. Napoli, internal note.

[5] J. Christiansen, *32 Channel general purpose Time to Digital Converter*, CERN/ECP-MIC, WWW: http://pcvlsi5.cern.ch:80/MicDig/jorgen/tdc32_ma.pdf.

[6] IEEE Computer Society, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, IEEE STd 1149.1-1990.