

A 96-channel, 500 ps resolution TDC board for the BaBar experiment at SLAC

S. Minutoli¹ and E. Robutti^{1,2}

¹Istituto Nazionale di Fisica Nucleare, Sezione di Genova, 16146 Genova, Italy

²Università degli Studi di Genova, Dipartimento di Fisica, 16146 Genova, Italy

Abstract

A TDC board has been designed and built to complete the readout of the Instrumented Flux Return of the *BABAR* experiment at the Stanford Linear Accelerator Center. The board has 96 input channels and makes use of 3 general purpose TDC chips designed at CERN, with time resolution up to 500 ps and configurable via a Test Access Port (IEEE standard 1149). Data are stored before readout in a multi-Event Buffer. Communication with *BABAR* DAQ system is realized through 3 serial lines on the backplane connector. All the logic, including internal registers and the interfaces with the *BABAR* protocol and the TAP controller, is implemented in two fast FPGAs. The board is designed to work at 59.5 MHz clock frequency.

I. INTRODUCTION

The *BABAR* experiment [1] aims at the study of *CP* violation in the neutral *B* meson system. It consists of a large detector installed around the interaction point of the newly built electron-positron collider *PEPII* at the Stanford Linear Accelerator Center. The detector itself is made up of 5 concentric subdetectors, the outermost of which, the muons and neutral hadrons detector, is referred to as *Instrumented Flux Return (IFR)* [2].

The active part of the *IFR* consists of a set of *Resistive Plate Chambers (RPCs)* interspersed with the iron layers of the magnet return yoke and covering a total area of over 1000 m². Signals generated by charged particles crossing the detector are picked up by about 50,000 aluminum strips, connected in groups of 16 to a set of *Front End Cards (FECs)* [3]. Each FEC in turn provides to store the hit pattern for a fixed time window and send it, multiplexed, to a custom *IFR FIFO Board (IFB)* when a trigger is received.

Besides the strip pattern, FECs also provide a *FAST-OR* signal, promptly formed as the logical OR of the 16 inputs. Given the known good intrinsic time resolution of the RPCs (~1 ns), this signal is suitable for use in some time-critical items, such as prompt bunch crossing time determination, IFR pattern recognition, rejection of reflections and cross-talk noise on strips.

These considerations led to the decision of upgrading the IFR set of front-end boards with a series of TDC boards, whose features are briefly described in this paper.

II. REQUIREMENTS AND SPECIFICATIONS

A. The *BABAR* environment

The *BABAR* DAQ system [4] is based upon a set of custom *ReadOut Modules (ROMs)* designed at SLAC. They manage the distribution of commands coming from the *Fast Control and Timing System (FCTS)* to the front-end electronics of each subdetector, gather and pre-process raw data and build the event to be sent to the following stage of processing

(essentially the level 3 trigger). Connection between ROMs and front-end is realized through a GHz optical link (*G-link*).

A dedicated data transmission protocol [5] has been developed by *BABAR* for sending commands and data through the G-link. Incoming optical signals are demultiplexed to 16 parallel lines at 59.5 MHz clock frequency (and vice-versa for outgoing signals), so that a single G-link can serve up to 16 front-end boards in parallel. Each front-end board is thus connected to the DAQ system by 3 serial lines: one (CLK) carries the 59.5 MHz clock signal, the other two (DIN and DOUT) transport data to and from the board, respectively.

Each command is preceded by a start bit on the DIN line: the next 5 bits identify the command, followed by a 5-bit (optional) sub-address. 6 of the 32 possible command codes are defined as global (*BABAR*-wide) commands and are listed in the first part of Table 1; 16 are left for subsystem-specific needs.

The way each front-end board responds to the receiving of global commands is strictly encoded in the rules of *BABAR* protocol. Some restrictions are imposed to the relative timing of some commands. In particular, two subsequent *L1 Trigger Accept* or *Read Event* commands, or an *L1 Trigger Accept* and its correspondent *Read Event* command cannot be closer in time than 2.7 μ s. All other sequences must be dealt with by front-end boards no matter how close commands are inside them.

One inevitable consequence of the high interaction rate and the complexity of the trigger logic is a delay in the delivery of the level 1 trigger which is far too large (~12 μ s) to allow a real-time snapshot of the detector to be taken. To keep up with the rate without introducing large dead times, data buffering is required to all front-end electronics¹: it has to be able to store data for a time corresponding to at least a fixed trigger latency of 12 μ s plus half a trigger jitter window of 1 μ s (see Fig. 1).

On occurrence of a trigger, data corresponding to the appropriate time window must be transferred to an internal Event Buffer to be read by the next *Read Event* command. The buffer must be deep enough to store data from at least 4 events.

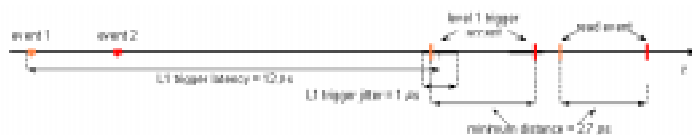


Figure 1: Timing of *L1 Trigger Accept* and *Read Event* commands.

B. *IFR*-Specific Requirements

IFR front-end electronics sits in 8 VME crates with custom backplane (*PDB*), located just outside the detector. A Controller Card (*ICC*) in each crate acts as an interface with the G-link and controls up to 16 read-out boards via point-to-point lines on the *PDB*.

¹ With an exception for the Electromagnetic Calorimeter, which sends data in continuous mode.

The total number of FAST-OR signals to be read by TDCs is approximately 3000. The number of FIFO Boards (*IFBs*) and Calibration Boards (*ICBs*) already hosted in front-end crates is such that a maximum of 5 TDC Boards (*ITBs*) is allowed per crate. This leads to the choice of a 96 channel board: a total of 36 *ITBs*, unevenly distributed among the crates, is then needed in order to read out the whole detector.

The required time resolution is dictated by a few figures. First of all, the intrinsic time resolution of the RPCs, which is known to be better than 1 ns. Then the close spacing of PEP-II bunches, which results in a bunch crossing interval of only 4.2 ns. Last, the precision needed by possible time-of-flight measurements, which has to be better than 1 ns to help in neutral hadron identification in the IFR. All of this leads to the requirement of a 500 ps time resolution. Moreover, the width of FAST-OR signals being ~ 70 ns, a double pulse resolution per single channel of 50 ns is required.

The dynamic range must be such that no ambiguities arise when converting read values to useful times. Since measurements are done with respect to the arrival of the *L1 Trigger Accept* command, the width of the trigger jitter window, 1 μ s, sets the minimum acceptable value.

Other requirements come from the general features of BABAR DAQ system. TDCs have to operate in continuous mode, i.e. without any dead time coming from other board activities, including response to *L1 Trigger Accept* and *Read Event* commands. Furthermore, a selective readout of digitized data must be possible in order to store into the Event Buffer only those pertaining to the appropriate time window. Also, channel masking capability is desirable to reduce the data size when noisy channels are present.

Finally, given the expected long lifetime of the experiment, long term stability and negligible temperature dependence is required for the TDC performances, including linearity.

C. Commands

A list of all commands recognized by the *ITB* is shown in Table 1.

Global commands (op-codes 00 ÷ 05) are the only ones allowed at run-time. They are broadcast by the FCTS to all ROMs in the partition during normal data taking. The *ITB* behavior in response to each of them follows the rules of BABAR protocol. Just a few remarks are in order here (other details are given later).

Table 1
ITB Commands.

OpCode (hex)	Command Name	Sub-Address	Action
00	<i>No Operation</i>	irrelevant	none
01(19)	<i>Clear Readout</i>	irrelevant	resets all counters
02(1A)	<i>Synchronize</i>	irrelevant	clears Event Buffer
03(1B)	<i>L1 Trigger Accept</i>	trigger tag	writes data into Event Buffer
04(1C)	<i>Read Event</i>	irrelevant	reads data from Event Buffer
05	<i>Calibration Strobe</i>	irrelevant	none (<i>Test Pulse</i> bit = 0); sends test pulse (TPb = 1)
11	<i>Write Register</i>	reg. Type and addr.	writes TDC register(s)
12	<i>Read Register</i>	reg. Type and addr.	reads a TDC register content
13	<i>En./Dis. Test Pulse</i>	0 (dis.) / 1 (enable)	enables/disables the <i>Test Pulse</i> control bit

The *Synchronize* command, besides resetting the 8-bit counter inside the Command Decoder FPGA, also resets all TDCs internal buffers and counters.

The response to a *Calibration Strobe* command depends on the value of a *Test Pulse* control bit inside the Command Decoder FPGA: if the bit is set, a pulse is sent to all 96 input pins of the TDCs; otherwise the command has no effect.

The *Synchronize* command, besides resetting the 8-bit counter inside the Command Decoder FPGA, also resets all TDCs internal buffers and counters.

The response to a *Calibration Strobe* command depends on the value of a *Test Pulse* control bit inside the Command Decoder FPGA: if the bit is set, a pulse is sent to all 96 input pins of the TDCs; otherwise the command has no effect.

For each of the commands with op-code 01, 02, 03, 04 there exists an *ITB*-specific copy, so that the same command can be sent “privately” even during non-run time.

Other two *ITB*-specific commands are used to read and write a set of internal registers used by TDCs. Here the sub-address specifies the register type (see next Subsection) and its address (i.e. the TDC number). The *Write Register* command must of course be followed by the data to be written to the register.

A further command enables setting/resetting of the *Test Pulse* control bit.

Any other op-code received by the board is not recognized as a valid command and raise a *False Command* flag, which is included in the next event header.

D. Registers

There are three sets of three registers each. They are located inside the Command Decoder FPGA and correspond to three different registers inside each of the three TDC chips.

Each *Configuration Register* is 128 bit wide; its last 127 bits map the 127-bit *Setup Register* of the corresponding TDC chip. This is intended to set up the chip general configuration, including operational modes and offset values (see [6] for details).

Mask Registers are 32 bit wide and correspond to bit 0-31 of the 33-bit *Control Registers*. They allow disabling of individual input channels (channel 32 is not used).

Both the Setup and the Control registers are meant to be write-only: thus, while the *Write Register* command first writes to the FPGA appropriate objects and then transfers their content to the corresponding TDC ones, the *Read Register* command only reads from the FPGA.

Finally, the 32-bit wide *Error Registers* store in their first 5 bits the content of the TDCs *Status Registers*. These are read-only registers and are thus actually read on each *Read Event* command.

E. Data Format

Data are stored into the Event Buffer in the form of 32-bit words. Each event is composed of a header, a number of data words varying from 0 to 96, and a trailer. Their format is shown in Fig. 2.

The header has the same format as for the other IFR front-end boards. The trigger tag is taken from the *L1 Trigger Accept* command, while the time stamp is set by an internal counter: they are used by the DAQ to check for possible “out

of sync” or “out of order” events. The crate and slot IDs are hardware selected through a set of jumpers on the board.

Other bits are used to identify the board type and the data type, to check the FIFO status and to signal a former occurrence of a No Operation or an unrecognized command.

Data words contain essentially all the information provided by TDCs for every hit recorded, i.e. the channel number², the 21-bit time record and an error flag. Bit 7 set to 1 enables to discriminate them unambiguously from header words.

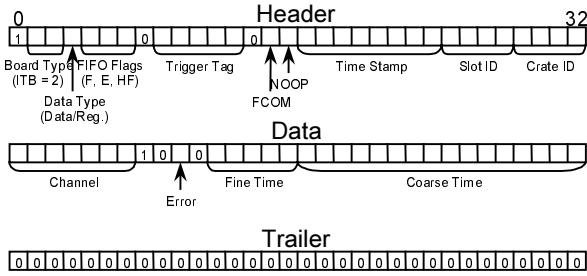


Figure 2: ITB data format.

III. BOARD DESIGN

A. General Design

The mechanical specifications for the ITB are those for a standard single-slot 6U VME board. Only the J1 connector is needed to access power supply and the three serial lines on the custom backplane.

ECL-differential input signals are fed through three high density, 68-pin front connectors. 12 LEDs on the front panel monitor the board status and activity, while other 13 inside are for testing/debugging purposes.

Let aside the input lines, standard TTL levels are used throughout the board. All components are SMD except for the two serial PROMs, which are mounted on sockets. A simplified block diagram of the board is shown in Fig. 3.

After conversion to TTL, input signals are read and stored (continuously) by the TDCs. Sequencing of all operation on the board is managed by the Command Decoder, which gets input from DIN.

On occurrence of an *LI Trigger Accept* command, header, data and trailer are written sequentially to the Event Buffer: therefore the same 32-line bus is driven by the Command Decoder and the three TDCs. All of them provide to tri-state their output buffers when not used, so that there is no need for external buffers. The same applies to the *Write Enable* line.

Read-out of the TDCs is controlled by a “token ring” protocol: a token is issued by the Command Decoder and sent to TDC0, which then begins transferring all available data to its output; when this is done, the token is passed to TDC1, and so on. After the last TDC (TDC2) have finished its job, the token is sent back to the Command Decoder to have the sequence completed.

Two other 32-line buses take data to the output unit, which performs a parallel-to-serial conversion. One is used on occurrence of a *Read Event* command to transfer a complete event data block from the Event Buffer; the other is used to

read registers content from the Command Decoder when a *Read Register* command is received.

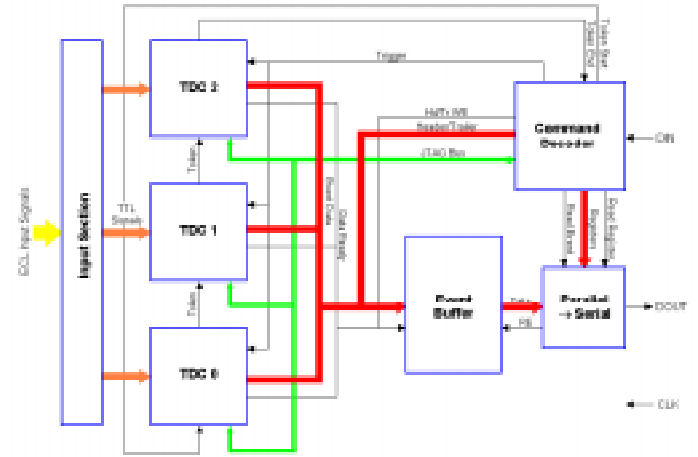


Figure 3: Simplified ITB block diagram.

Configuration of the TDCs is done via a standard *Test Access Port* and the JTAG protocol defined in [7]. A JTAG bus connects the Command Decoder to the TDCs and consists of one *TCK*, three *TMS*, one *TDI* and one *TDO* lines. Three separate *TMS* lines are used so that it is possible to write to any number of TDCs in parallel. The other lines are shared, while *TRST* is not used.

B. Clock Distribution / Input Section

The 59.5 MHz clock is supplied through the CLK line on the PDB. Given the high frequency, a good quality is necessary for a reliable behavior of the various components on the board: this is especially true for for the TDC chips, which require a signal with jitter lower than 200 ps and duty cycle between 45% and 55%.

This is accomplished by use of a CY7B9910 clock buffer, which re-shapes the input signal by means of an internal PLL and drives 8 clock lines.

Input signals come from the FECs as 96 ECL-differential pairs. A protection is applied against spikes before conversion to TTL levels: an SP720 “transient diverter” limits the tension values between two reference levels, in this case chosen to be VEE (−5.0 V) and GND.

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* command when the *Test Pulse* bit is set: to switch between external and test pulses, 6 74CBT16233 32-bit multiplexers are used.

C. TDC

A 32-channel general purpose TDC chip [6] developed at CERN for LHC experiments has been chosen. Its main specifications are listed in Table 2

Time digitization is performed by means of a coarse time counter running at the supplied clock frequency (59.5 MHz in this case) and a chain of delay gates for the fine timing, providing a bin size of 1/32 of the clock period. A self-calibration scheme based on the use of a *Delay Locked Loop* (DLL) balances variations in the gates delays due to temperature and voltage drift.

² The 2 upper bits identify the TDC chip and are set externally.

Being conceived as a general purpose device, this IC features a variety of different operational modes and tunable parameters. The working configuration is set via the registers described in the previous Section, which are accessed through the Test Access Port.

A “trigger matching” procedure is used to select and read out only those stored data lying within a definite time window. To do this, appropriate values for the trigger time offset and window width are loaded on configuration. Besides that, the trigger time can be automatically subtracted before readout.

Each input channel can be assigned a different offset in “fine time” units (= 525 ps) or disabled, if necessary.

Table 2
TDC chip main specifications.

Number of channels	32
Input threshold voltage	TTL
Clock frequency	20 ÷ 60 MHz
Time bin size	$T_{clk} / 32$ (= 0.525 ns at 59.5 MHz)
Dynamic range	21 bit (= 11 ms at 59.5 MHz)
Double pulse resolution	15 ns
Max. hit rate on single channel	$f_{clk} / 80$ (= 750 MHz at 59.5 MHz)
Event Buffer size	256 words
Read-out buffer size	32 words
Enable/disable of single channels during normal operations	
Complete configurability via JTAG protocol	

D. Logic

The logic controlling all operations on the board is implemented on two ultra-fast Xilinx FPGAs, programmed on power-up with data stored in two serial PROMs.

All the logic is synchronous and runs at 59.5 MHz clock frequency (except for the TAP interface, which works at 14.9 MHz). Given the high frequency, careful optimization was required in the design: automatic synthesis from “high level” sources has proven not to be suitable, so that a “low level” approach, often at the level of elementary logic cells, has been necessary (again, with the exception of the TAP controller state machine, which has been written in *verilog* language).

An XC4006E-1 device is used for the largest block of logic. It features 128 I/O buffers and 256 *Configurable Logic Blocks (CLBs)*, individually configurable to work as synchronous RAMs. This capability has been exploited to realize the large registers needed.

A block diagram of the logic implementation is shown in Fig. 4. It includes three main blocks: a command decoder, which decodes the pattern transferred from DIN into a shift register; a set of registers (see previous Section), seen as $n \times 1$ RAMs; the TAP interface, which drives the JTAG bus at $\frac{1}{4}$ the clock frequency.

A separate FPGA does the parallel-to-serial conversion of data to be sent on the DOUT line. An XC3142A-1 has been chosen for this purpose: a simpler and very fast device that features 69 I/O buffers and 144 CLBs.

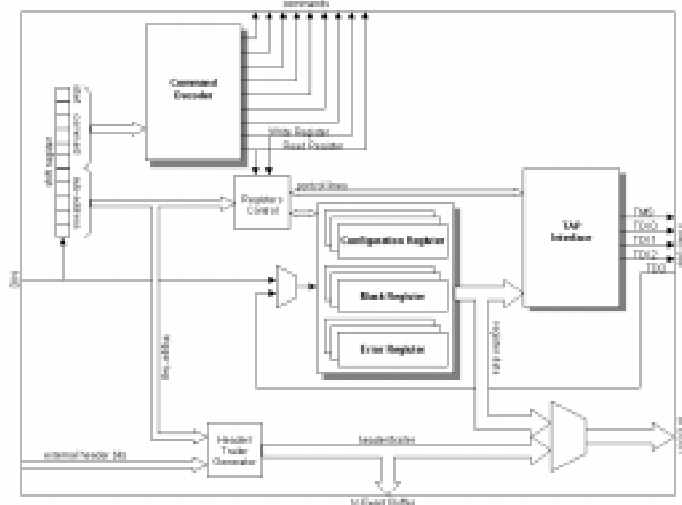
E. Event Buffer

The Event Buffer is an SN74ACT3631-15, 512×36 clocked FIFO memory, supporting clock frequencies up to 67 MHz. It makes use of two separate clock inputs (as well as two separate enable inputs) for read and write operations. Two

lines carrying the 59.5 MHz clock are used for that, with independent programmable digital delay lines on input to assure a correct data-vs.-clock timing.

The FIFO size is such that it can store at least 5 full sized events, the maximum event size being $1 + 32 + 32 + 32 + 1 = 98$ words.

Figure 4. Block diagram of the Command Decoder logic.



IV. CONCLUSIONS

After production and testing of a prototype, a pre-production board has been built and tested extensively and has proven to reproduce the required functionality and performances. Final stage testing involved integration with the complete *BABAR* data acquisition system during cosmic rays and physics runs. Production of the whole stock of 40 boards is now complete and the ITBs are ready to be installed on the experiment.

V. ACKNOWLEDGEMENTS

We gratefully acknowledge the help given by Dr. Paolo Musico and the whole staff at the electronics workshop of I.N.F.N. Genova in the development of the FPGA code and the production of the board layout.

VII. REFERENCES

- [1] D. Boutigny et al. (the *BABAR* collaboration), The *BABAR* Physics Book, SLAC-R-504, P.F. Harrison and H. R. Quinn, Editors, October 1998.
- [2] F. Anulli et al., The muon and KL detector for the *BABAR* experiment: physics requirements, final design and start of construction, *Nucl. Phys. B (Proc. Suppl.)* 61B (1998), 244-249.
- [3] N.Cavallo et al., Front-End Card Design for the RPC Detector at BaBar, INFN/TC-96/22.
- [4] R. Hamilton, The BaBar Data Acquisition System, paper contributed to these Conference Proceedings.
- [5] G. Haller and G. Oxoby, BaBar Note 281.
- [6] J. Christiansen, 32 Channel General Purpose Time to Digital Converter, CERN/ECP-MIC (WWW: http://pcvlsi5.cern.ch/MicDig/jorgen/tdc32_ma.pdf).
- [7] IEEE Computer Society, IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-1990.