

The Central Trigger Processor Monitoring module (CTP_MON) in the ATLAS Level-1 Trigger System

N. Ellis, P. Farthouat, P. Gallno, G. Schuler, R. Spiwoks, R. Teixeira

CERN, 1211 Geneva 23, Switzerland

H. Pessoa Lima Jr, J. M. de Seixas

Universidade Federal do Rio de Janeiro / COPPE, Rio de Janeiro, Brazil

Abstract

The Central Trigger Processor (CTP) receives a 160-bit trigger pattern from the Calorimeter and Muon Trigger sub-systems at 40 MHz update rate, corresponding to the 25 ns bunch-crossing period of the LHC. This information is evaluated against a trigger menu to form the general Level-1 Accept (L1A) signal. The CTP Monitoring module (CTP_MON) monitors the trigger inputs of the CTP on a bunch-by-bunch basis in order to monitor trigger rates per bunch and to detect any pathological bunches in the LHC machine. This information can also be used to estimate the luminosity per bunch. The monitoring system is able to build histograms of the input data at the maximum possible rate, with one 30-bit counter per input bit for each of the 3564 possible bunch positions in the LHC giving at least 26 hours of free running without overflow condition. The design and simulation of the CTP_MON will be presented.

I. FUNCTIONAL DESCRIPTION

The CTP Monitoring module is one of six types of modules that form the Central Trigger Processor [1]. Its task is to monitor the selected 160-bit pattern coming from the trigger sub-systems at the 40 MHz bunch clock. This monitoring is performed bunch-by-bunch (for the 3564 bunches in the beam) and each trigger input is histogrammed in a 30-bit counter. An overall architecture of the CTP_MON, illustrating its functional blocks, is shown in Figure 1.

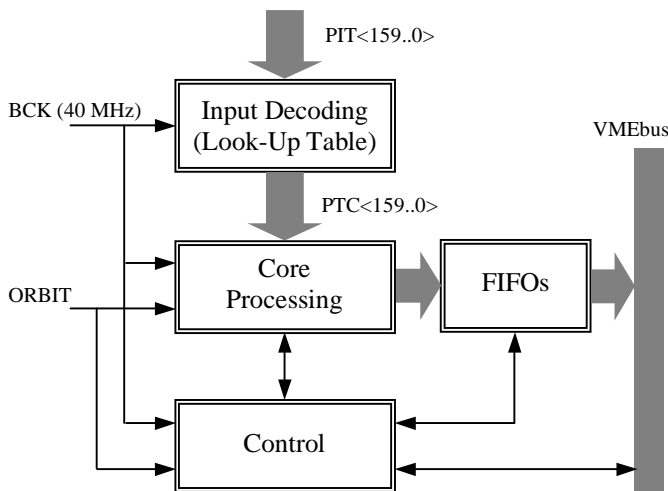


Figure 1: CTP_MON overall architecture

A. Input Decoding

The PIT bus (*Pattern In Time*), containing the 160 trigger signals, already aligned in time and synchronized to a local bunch clock, is first applied to an input decoding block. These inputs, consisting of multiplicity or energy information, are decoded into the PTC bus (*PIT Decoded*). The input decoding is based on several Look-Up Tables (LUT) implemented in memory blocks inside an FPGA. These LUTs provide the following flexibilities over the PIT bus:

- **Routing:** all the PIT inputs should be able to be connected with respect to any output;
- **Grouping:** all the PIT inputs should be able to be grouped in any configuration of groups, e.g. 2, 3 or 4;
- **Decoding:** the logic must decode the groups according to a large number of possibilities (keeping the same number of inputs at the output). Considering groups of multiplicity type, Figure 2 illustrates two possibilities of decoding.

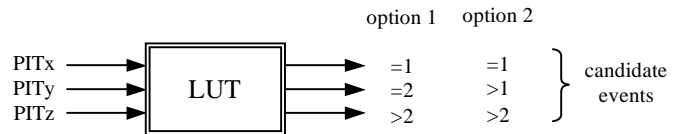


Figure 2: One example of decoding using the LUT

The LUTs are implemented in ROM memory blocks and each one contains data previously stored during the FPGA configuration.

In order to increase the design flexibility, when routing different groups of PIT, two layers of registers are used at the input and two at the output of the LUT. Concerning the input side, the first layer (the one that receives the PIT bus) is responsible for keeping the minimum and constant setup time (t_{su}) for all the inputs, no matter what very different routing takes place. The second layer in the data path is used to increase the flexibility of the physical routing inside the FPGA, thus avoiding timing restrictions due to a new configuration. On the output side, the same idea applies for the first layer in the data path. The last layer of registers is useful to keep a minimum and constant clock to output parameter (t_{co}). This extensive use of registers is only possible because the CTP_MON unlike other modules of the CTP is not critical in latency.

Due to the registers in the data path, the input decoding circuitry presents 4 clock cycles of latency from input to output. This is compensated inside the Core Processing, using an

address generator with programmable offset.

The input decoding is fully implemented in one FPGA. The design presents a maximum clock frequency of 101 MHz. The setup time related to the PIT bus is 2.7 ns and the clock to output is 6.3 ns in the last layer of registers.

B. Core Processing

This block contains the histogramming circuitry, besides the following implementations: almost overflow control, local BCID (Bunch-Crossing Identification) generator and readout control. The complete design is implemented in four high-speed/high-density FPGAs due to the massive use of memory required.

The histogramming circuitry must be able to build histograms of the PTC bits at the maximum possible rate. Figure 3 illustrates the basic mechanism designed to carry on this task.

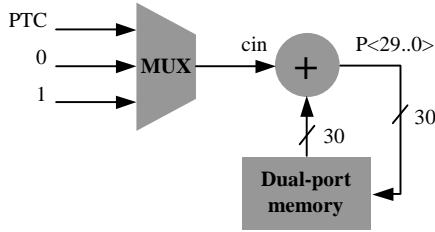


Figure 3: Histogramming cell

Each PTC input is accumulated in a 30-bit data word ($P<29..0>$), which is stored in a true dual-port memory. Data read from the memory is incremented through a 4-clock latency adder and written back to the memory. One should note that two different addresses are needed for reading and writing at the same time. In this way, it was possible to achieve zero deadtime for the histogramming process. Besides the adder-memory pair, a simple three-input multiplexer is used in this scheme, providing input selection ability for testing and calibration purposes.

An Almost Overflow control circuit, shown in Figure 4, is also designed in the Core Processing to generate an interrupt request on the VME bus whenever a P data word passes a programmable threshold ($TH<3..0>$). This comparison is performed using the 4 most significant bits in P, through a comparator (CMP) with 1 clock latency.

The comparison results of the 160 comparators are applied to an OR gate and stored in a register for each bunch crossing clock (BC). A high-level in the AO flag (see Figure 4) signals to the control circuit that an interrupt request must be issued. In order to deal with timing constraints, the 4-bit comparators work with an output latency of one clock cycle, which does not matter for the functionality.

Both the Core Processing and the Control contain a local BCID generator. This circuit is based on 12-bit counters and comparators, providing a programmable offset to take into account the total latency from the ORBIT reception to the effective use of the BCID in the histogramming process.

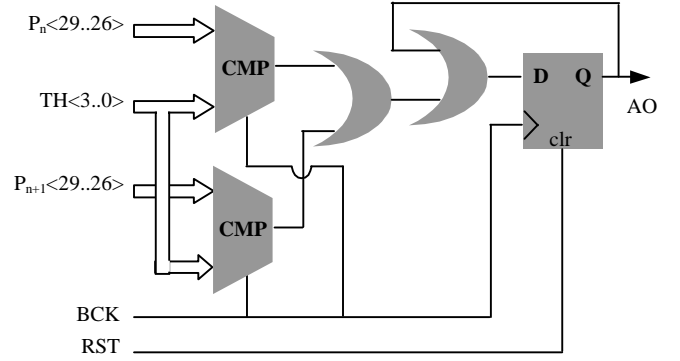


Figure 4: Almost Overflow control (n varying from 0 to 158)

One important functionality implemented inside the Core Processing is the readout control, responsible for the data transfer from the histogramming memory to the FIFOs. This circuit allows the VME cpu to access the Core memory at any time during normal operation (including the integration period). The readout control block makes use of a large 31-bit, 40×1 multiplexer to transfer one P data word at a time to the FIFOs. This multiplexer presents a latency of 3 clock cycles to meet timing requirements. Two registered-output finite state machines [2] are used to control the readout process (from the Core FPGAs to the FIFOs) and to generate the write enable signal for the external FIFOs.

The Core Processing requires four FPGAs to be implemented. Each one processes 40 trigger inputs decoded (PTC) in a parallel way and presents a maximum clock frequency of 66 MHz.

C. Control

This circuit is responsible for storing 20 status and control registers, and for implementing the VME interface.

The control registers give the flexibility needed to operate the module, including: Global Reset, Mode of Operation, Start Readout, Input Selection, BC Offset, Almost Overflow threshold, etc. The status registers provide detailed information about processes and devices in the module. This includes: Readout Status, FIFOs status (full/empty/almost), Number of turns during integration, Power Good, etc. All the registers are accessed using 32-bit single R/W cycles in the VMEbus.

The CTP_MON is a slave module type A32:D32:BLT based on the VME64x specification [3]. Decoding and interfacing circuitry is completely implemented in a block called VMEDEC, in the Control FPGA. This block makes use of a Finite State Machine, which is responsible for the protocol required between the module and the VMEbus. Three types of cycles are allowed by the module: Single R/W, Block Transfer (BLT) and Interrupt cycle. The first type is used only for reading and writing operations on the registers. The BLT cycle is used to read the histogramming data from the FIFOs. Interrupt requests may be generated due to an almost overflow condi-

tion in one of the 160 channels (as described in Section B). It is also foreseen to use the geographical addressing capability if requested in the final CTP specification.

D. FIFOs

Four IDT FIFO [4] memories (130,072 x 40) are used in order to simplify and speed up the data transfer from the CTP_MON. These memories provide an easy way to carry on the block transfers.

Besides the IDT FIFOs, a small FIFO (12k) is also necessary inside the Core FPGAs to complete the required total size of 142,680 words.

II. DATA FORMAT

Data words transmitted from the CTP_MON to the VME cpu are divided in four types: Header 1, Header 2, Header 3 and P word. Figure 5 shows the data words organization in the system.

Header 1:

31	29	28	21	20	13	12	11	0
100		X		PIT code		0		BCID

Header 2:

31	29	28	16	15	0
101		X		Turn Count (LSW)	

Header 3:

31	29	28	16	15	0
110		X		Turn Count (MSW)	

P word:

31	30	29	0
0	ov		P word

Figure 5: Data format in the CTP_MON

Each data word in this scheme is primary defined by D31. If this bit is high-level, the word is a header, otherwise it is a P word. Furthermore, there are three types of header and, for each one, the three most significant bits define its content. Header 1 is defined by the pattern 100 and contains the PIT code of the first P word transmitted, and the first BCID. Headers 2 and 3 are defined by 101 and 110, respectively, and contain the Turn Count value, which means the number of turns since the beginning of the integration process. The P word is 30-bit wide and corresponds to the histogramming data for the PIT inputs. For each P word transmitted, the bit D30 in the data bus represents the overflow information for that PIT input (high-level meaning that overflow has occurred).

From the Core FPGAs to the FIFOs, as well as from the FIFOs to the VMEbus, 32-bit words are always transmitted for simplicity. Using the data format described above, the total amount of data in the CTP_MON is 2.2 MByte, which takes on average about 72 ms to be transmitted from the module to the VMEbus.

III. MODES OF OPERATION

On the power-up, all the CTP_MON registers are initialized with specific values and the module goes into an idle state. This idle state remains until some command is issued by the VME cpu.

In order to increase the flexibility during operation, and also to test the CTP_MON prior to its use in the real experiment, two modes of operation are defined for the PIT integration:

- **NORMAL mode:** the user defines start and stop commands for integration of the trigger inputs. The integration is always synchronized with the local BCID 0.
- **WINDOW mode:** the user defines the number of beam turns for integration (maximum = $2^{30}-1$). In this mode, the integration window is also synchronized with local BCID 0.

In both modes, the integration period is a multiple of beam turns, so that it begins with the BCID 0 and finishes in the last BCID (3563). When using the Window mode, the R/W register called 'Number of Turns' (30-bit wide) defines how many turns will be integrated after the start.

IV. IMPLEMENTATION

VHDL and schematic approaches have been used as design entry and simulation methods for the FPGAs. The system level schematics, simulations and layout design were realized using Cadence [5] tools.

The first CTP_MON prototype, shown in Figure 6, is fully implemented in one 9U VME board with 10 layers of copper.

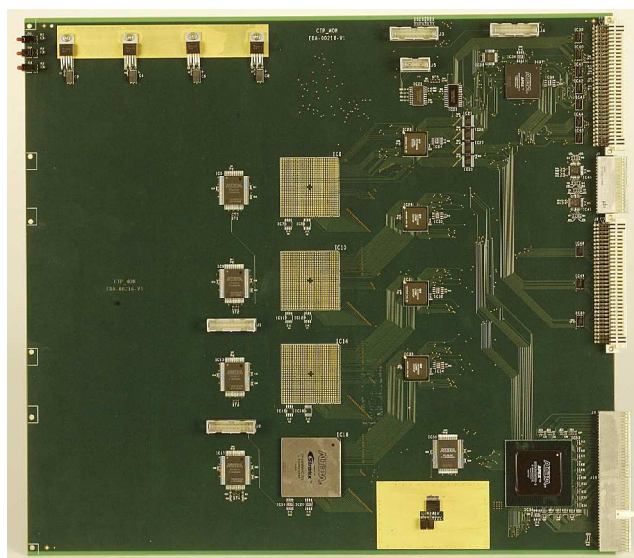


Figure 6: CTP_MON prototype

Dedicated planes are used to route the clock lines and the PIT bus, besides Ground and Power planes. The clock signal (LVPECL standard) is distributed from a clock driver IC using a star topology, thus reducing skew and jitter on the board. Most of the digital devices are connected in a JTAG chain to allow devices programming and Boundary Scan testing. In this

first prototype only one out of four Core FPGAs was mounted due to cost reasons. This keeps the same functionality of the module but for only 40 trigger inputs (PIT bus).

Three different FPGAs from Altera [6] were used in the module: one APEX EP20K160E to implement the Control block, one APEX EP20K300E for the Input Decoding and four Stratix EP1S80 to implement the Core Processing. The choice for this last FPGA, the largest one available from Altera at this moment, was based on the amount of memory blocks required for the histogramming circuitry. The usage of memory for 40 PIT inputs is 5,583,488 bits, which means 75 % of the total available in the FPGA. In the case of the Input Decoding FPGA (LUT), the number of I/O pins required is determinant for the choice. From a total of 408 pins available in the FPGA, 345 (84 %) pins are used to implement the PIT and PTC buses, besides the clock and control signals.

A Test Bus Controller (TBC) IC [7], from Texas Instruments, is used in order to provide in-system programmability (ISP) of the configuration devices through the VMEbus. A two-position jumper on the board selects between the use of a JTAG cable or the TBC to download the configuration files.

Three LEDs are mounted on the front panel of the module to provide information about the status of the power supplies regulated on the board: 1.5V, 1.8V and 2.5V. If the voltage level falls below 91 % of nominal value, the respective LED turns on.

V. SIMULATION RESULTS

Complete simulations have been done to evaluate two basic operations in the CTP_MON: the PIT integration and the readout process. Both simulations were realized using the tools Quartus II (Altera) and NCSim (Cadence). The first tool was mainly used during the FPGA design and the second one for simulating at system level, including also the IDT FIFOs.

One should note that, to save simulation time, we have always set the “BCID max” register to 7, which means that instead of 3564 bunches, as in the LHC, the simulations have been performed for 8 bunches.

A. PTC integration using Quartus II

The PIT integration might be realized in two different modes as described in Section III. The following simulations made use of the Window mode, where the PTC bits are integrated over a specific number of turns, defined in a control register. The purpose of this simulation is to verify the correct functionality of the histogramming process in the Core FPGA design. The initialization procedure consists of:

- 1) Reset the Core FPGA memory;
- 2) Define the Window operation mode in the “General Control” register;
- 3) Define the PTC as 0xFFFFFFFF (in order to integrate all the channels) and;
- 4) Define the number of turns for integration as 5 in the “Number of Turns” register.

At 887.5 ns of simulation the start integration command is issued and an integration window is created synchronized with the ORBIT signal. Finally, the readout is started and we can check that the contents are 5 in all the P words, as expected.

B. PTC integration using NCSim

In this simulation we repeat the same operations as the previous case, but this time all the signals, except VME control signals and the PTC bus, are generated by the Core and Control FPGA as well as by the IDT FIFO. The PTC is defined as a pattern changing each 25 ns between 0xAAAAAAAAAA and 0x5555555555 in order to integrate the channels in an alternate manner. Before 4000 ns there is an integration window of 5 LHC turns. After the integration period, the readout starts and we can verify that the output of the IDT FIFO correspond to 5 and 0 successively, due the PTC pattern applied. Considering the total simulation period, the worst setup time measured in relation to the FIFO is 6.0 ns. Since the minimum setup time specified is 3.0 ns, the circuit should work properly, even taking into account propagation delays due to the tracks. Other simulations were realized by applying different PTC patterns and in all the cases the results demonstrated a correct behavior of the histogramming circuitry.

C. Readout using Quartus II

The objective in this simulation is to evaluate the circuit designed to control the data transfer between the CTP_MON and the VMEbus. It includes the evaluation of functionality, timing parameters and circuit performance. Data transferred represent the integration results over the PTC bits and are formatted as described in Section II. In the scope of Quartus II simulations, only the “Readout Control” circuit inside the Core FPGA is evaluated. The following features are verified:

- functionality of the state machine that controls the readout process;
- functionality of the small FIFO inside the Core FPGA;
- generation and timing of the write enable signal for the IDT FIFO.

The total duration of the simulation was 25 μ s and the results demonstrated the expected behavior of the design. The period corresponding to the data transfer between the Core FPGA and the IDT FIFO is called ‘state = 1’.

Parameter	Specified	Measured during ‘state = 1’	Measured during ‘handshake’
Setup time for write enable	3.0 ns (min)	16.0 ns	16.0 ns
Hold time for write enable	0.5 ns (min)	9.1 ns	9.1 ns
Operation frequency	250.0 MHz (max)	40.0 MHz	6.7 MHz

Table 1: Timing parameters for the IDT FIFO

The second stage, when the VME cpu reads the IDT FIFO and the Core transfers the remaining data, is called ‘handshake’. Table 1 shows some results measured after simulation.

D. Readout using NCSim

This simulation includes the Control FPGA, the Core FPGA and the FIFO. The following points have been verified:

- write operation in the control registers (Control);
- operation of the state machine used to control the readout (Core);
- operation of the state machine used for the VME controller (Control);
- timing of the DTACK signal coming from the Control;
- operation of the FIFO inside the Core FPGA;
- operation and timing of the IDT FIFO;
- timing of the write enable signal for the IDT FIFO (Core).

The necessary VME control signals are created as stimulus based on the VME64x specifications. From the beginning up to 26.5 μ s, the operations are the same as realized in the Quartus II simulation. From this instant to the end, we can check the data transfer between the Core, the IDT FIFO and the VMEbus. Again, this period is called 'handshake' and the first one is called 'state = 1'.

Parameter	Specified	Measured during 'state = 1'	Measured during 'handshake'
Setup time for write enable	3.0 ns (min)	13.3 ns	13.3 ns
Hold time for write enable	0.5 ns (min)	11.7 ns	11.7 ns
Setup time for read enable	3.0 ns (min)	-	6.3 ns
Hold time for read enable	0.5 ns (min)	-	18.7 ns
Data setup time	3.0 ns (min)	7.0 ns	131.0 ns
Data hold time	0.5 ns (min)	15.0 ns	14.2 ns
Data access time	4.5 ns (max)	-	1.8 ns

Table 2: Timing parameters for the IDT FIFO

The simulation verified the correct behavior in all the transactions. Table 2 illustrates timing parameters measured after

simulation.

Some initial tests of the first prototype had been carried through and verified the correct functioning of the VME controller, concerning the access to the control and status registers, as well as the turn counter mechanism.

V. CONCLUSION

The CTP Monitoring module has been designed and implemented. Extensive simulations were carried out during the design and the results verified the expected functionality and timing. A first prototype is ready and is currently being tested in the laboratory.

It is planned to evaluate the CTP_MON module in large scale tests of the ATLAS trigger system at the test beam in the Summer of 2004.

VI. ACKNOWLEDGMENTS

We would like to thank the support provided by CERN and the brazilian agency CAPES.

VII. REFERENCES

- [1] ATLAS Collaboration, First-level Trigger Technical Design Report, CERN/LHCC/98-14, June 1998.
- [2] Registered-output FSMs synchronize outputs to state transitions, Richard A. Johnson, Boeing, EDN June 4, 1998.
- [3] American National Standard for VME64 Extensions, ANSI/VITA 1.1-1997.
- [4] <http://www.idt.com/products/pages/FIFOs-72T40118.html>
- [5] <http://www.cadence.com>
- [6] <http://www.altera.com>
- [7] <http://focus.ti.com/docs/prod/productfolder.jhtml?genericPartNumber=SN74LVT8980A>