Available on CMS information server

CMS CR 1997/017

CMS Conference Report

22 October 1997

Trigger synchronisation circuits in CMS

J. Varela* [1], L. Berger [2], R. Nóbrega[3], A. Pierce[2], J.C. da Silva[3]

[1] *CERN, Geneva, Switzerland*

[2] *TECMIC, Lisbon, Portugal*

[3] *LIP, Lisbon, Portugal*

Abstract

We present the principles of the CMS method for synchronizing the trigger data at LHC. The method makes use of the LHC bunch gap and allows for a re-synchronisation of the data every LHC orbit (88 μs). It relies on the distribution by the TTC system of a signal synchronous with the first bunch in the orbit, and is implemented by dedicated synchronisation circuits in each trigger link. We report on the test of a prototype FPGA implementation.

Presented at 3rd Workshop on Electronics for LHC Experiments, London, September 1997

* email: joao.varela@cern.ch

1 Introduction

The CMS L1 trigger system can be viewed as a massive parallel processor that computes local trigger objects, followed by a tree like structure that selects the highest rank objects in the detector or performs global energy sums. The entire system works in synchronous pipelined mode, processing events at the 40 MHz LHC rate.

The system is based on the assumption that, at every processing stage, the data are synchronized and belong to the same bunch crossing. This goal has to be achieved without making use of bunch crossing identifiers attached to the data, which would imply a non acceptable complication of the trigger system design.

Due to different flight paths to different regions of the detector, the various front-ends are not necessarily synchronized with each other. In addition, the length of the links from the detector to the trigger processors can be different. Changes in length due to mechanical stretching during installation or replacement of optical fibers can occur. Temperature variations are also expected to induce variations in the signals timing. As a consequence, data corresponding to the same crossing have an unpredictable phase at the trigger inputs.

The use of test pulses to derive the delays needed in the trigger inputs to achieve synchronisation is subject to uncertainties in the timing of the distributed signals, specially when optical fibers are used to distribute the test signals.

In reference [1] we have proposed a new method for the synchronisation of the trigger data which takes profit of the gaps in the LHC bunch structure. In this paper, after a summary of the method, we describe the main characteristics of the synchronisation circuits SyncTx/Rx and we present preliminary results of the test of a prototype FPGA implementation.

2 Synchronisation method

The LHC bunch structure has 3564 periods of the 40.08 MHz clock in one orbit. The structure has several gaps, that is sequences of missing bunches, of predefined length and location. We will concentrate in the LHC extraction gap (127 missing bunches), that for simplicity we will call the gap. By definition, the first bunch after the gap is called the 'bunch zero'. In addition to the 40 MHz clock, the LHC machine can provide a 'bunch 0 clock' at a frequency of 11.25 kHz, the frequency of the LHC orbits.

The TTC system [2] distributes to the front-end, trigger and readout electronics, through optical fibers, the timing and trigger control signals. In order to be able to perform the synchronisation operations, the TTC system is programmed to distribute to all trigger synchronisation circuits the Bunch Crossing 0 (BC0) signal[1]. The BC0 timing is adjusted to match the arrival of the bunch zero data at the trigger synchronisation circuits.

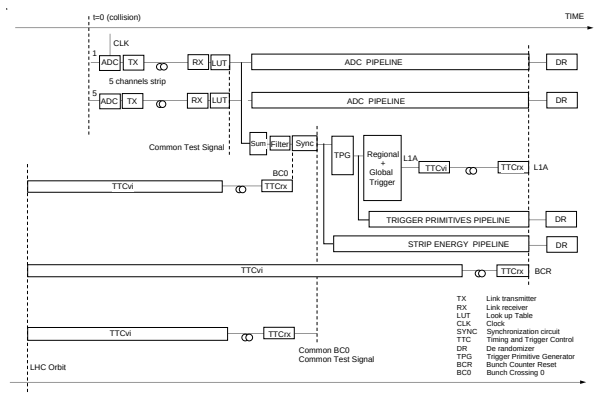

Figure 1 - Timing diagram of the calorimeter trigger

The synchronisation circuits placed at the input of the trigger processors make use of FIFOs to achieve synchronisation. The key points in the FIFOs operation are the following: i) after a clear FIFO executed during the gap, the first data entering every FIFO must correspond to the bunch zero crossing ; ii) the readout enable of the FIFOs is controlled by a common signal arriving simultaneously to every circuit. These resynchronisation operations are performed at every LHC orbit.

Phase adjustments between the TTC command BC0 and the trigger data are computed based on statistical data accumulated in the synchronisation circuits. A few minutes of running is in general enough to determine all time settings needed for a synchronous operation of the trigger. During normal running this information is used to monitor the synchronisation stability.

The circuits have monitoring mechanisms which allow to identify any loss of synchronisation in an orbit by orbit basis. When such error occurs, the events collected in the anomalous orbit are flagged by the readout system.

As an application example, we show in figure 1 a timing diagram of the CMS calorimeter trigger [3]. We assume that, at the output of the linearisers, the data corresponding to the 5 crystals of one strip [4] are synchronous since these data are transmitted through the same fiber ribbon. With this assumption there is no

[1] The BC0 signal and the Bunch Counter Reset (BCR) signal defined in the TTC documentation [2] have the same frequency but their phase differs by an amount of the order of the trigger latency.

need for a synchronisation step before the first summation.

The synchronisation circuits are placed after the L1 filter which extracts the energy and timing information out of the pulse samples. In this way data presents a sharp transition at the gap boundary that allows a clear identification of the bunch 0.

The TTC network distributes the BC0 command to all synchronisation circuits in the calorimeter trigger. Global adjustments of the BC0 phase are made at the top of the calorimeter's TTC sub-network (TTCvi module [5]). Individual adjustments are made in the TTC receivers (TTCrx [6]) associated to each synchronisation circuit.

An independent fan-out network distributes a common BC0 signal and a common clock to synchronise the FIFO's readout. The same network is used to distribute test signals synchronously to all trigger inputs. These test signals are used to measure in situ the trigger latency which is needed to adjust the readout pipelines length.

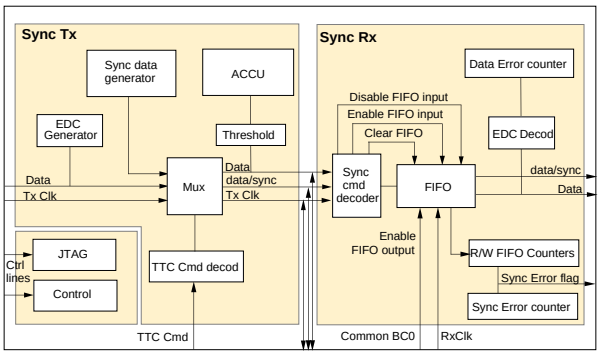

Figure 2 - Block diagram of the Synchronization Circuit.

3 Trigger synchronisation circuits

In each trigger data link, the synchronisation actions are performed by the circuit SyncTx/Rx (Figure 2). The circuit is divided in two main blocks. The SyncTx block is responsible for flagging the bunch zero data and for the accumulation of the bunch profile histogram. The SyncRx block contains the synchronisation FIFO and monitors the synchronisation errors. EDC generation and decoding functions are also provided.

The SyncTx/Rx circuit can be used in the two configurations represented in the figure 3. In the first one the functionalities of the SyncTx and Rx blocks are executed by two circuits located at the two ends of the trigger links and programmed in Tx and Rx modes, respectively. This configuration allows the monitoring of the trigger links: link identification and test functions available in the synchronisation circuits can be used. In the second configuration the synchronisation functions are executed by the same circuit programmed in Tx/Rx mode.

3.1 Flagging the bunch zero

The SyncTx block receives the input data frame together with the clock, receives the TTC commands and outputs the data together with the 'data/sync flag'. This flag is active during the gap.

The main blocks of the SyncTx, shown in figure 2, are the following:

- Synchronization Data Generator, which generates synchronisation data during the gap;

- Multiplexer, which allows to switch between real data and synchronisation data;

- Accumulator, which allows to build the bunch profile histogram (see §4);

- TTC Command Decoder, which decodes the commands BC0, Start, Stop and Reset.

The TTC command BC0, if properly adjusted in time, indicates that the data incoming in the next clock corresponds to the bunch 0. When this command is decoded, the circuit switches the Multiplexer to real data and sets the 'data/sync flag' in 'data' mode.

After reception of the BC0 the circuit counts the appropriate number of clock periods in order to identify the start of the gap (SOG). When SOG is reached, the circuit starts the Synchronization Data Generator (a simple counter), switches the Multiplexer to synchronisation data and sets the 'data/sync flag' in 'sync' mode.

3.2 Synchronisation FIFO

The SyncRx receives the data frame together with the 'data/sync flag' and the clock (Tx Clk). The SyncRx

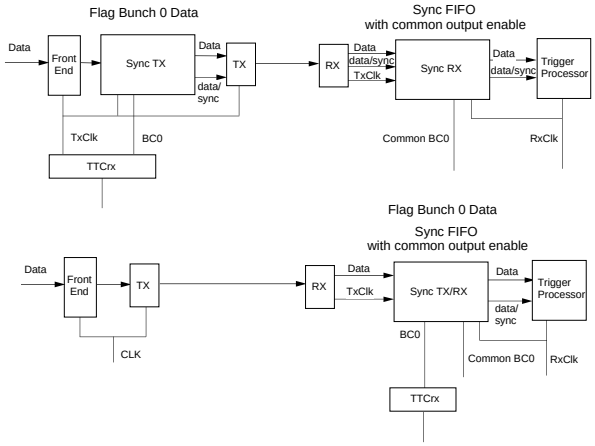

Figure 3 - Two configurations using the SyncTx/Rx circuit.

receives also the Common BC0 and the common clock (Rx Clk).

The main blocks of the SyncRx, shown in figure 2, are the following:

- The Synchronization Command Decoder (SCD);

- The Synchronization FIFO;

- The Monitor of Synchronisation Errors.

When the 'data/sync flag' at the input enter the 'sync' mode the SyncRx disables the FIFO input: one LHC orbit is completed and in consequence the input of data in the FIFO should be inhibited.

One of the synchronisation words received during the gap is recognized by the SCD as a clear FIFO command. The clear FIFO prepares the circuit to receive data from the next LHC orbit. The timing of this command shall be such that all the FIFO data from the previous orbit was already read.

When the 'data/sync flag' enters the 'data' mode, indicating that the next LHC orbit has started, the FIFO input is again enabled. Note that this event can happen at different times in different channels.

The input in the Synchronization FIFO is synchronous with the clock TxClk. The RxClk clock drives the FIFO readout. It is the responsibility of the fan-out network to distribute RxClk clock to all SyncRx circuits in the system with exactly the same phase. This guarantees that all the FIFOs in the system are read at the same moment and that the trigger data gets synchronised.

The read access to the Synchronization FIFO is controlled by the Common BC0 signal. When the Common BC0 is identified, the FIFO output is enabled and the next data extracted from the FIFO corresponds to the bunch 0. The timing of the Common BC0 must be adjusted in such a way that all the Sync FIFOs in the trigger system have already received data from the bunch 0. On the other hand it can not be issued much latter than the latest data to not affect significantly the trigger latency. Again, it is the responsibility of the fan-out network to distribute the Common BC0 command to all SyncRx with the same phase. In this way we guarantee that the all trigger data is synchronous and corresponds to the same bunch crossing.

Similarly, when the SOG condition is reached the circuit disables the FIFO output. During the synchronisation gap the circuit outputs zeros in the data bus.

The circuit counts the number of writes in and reads from the FIFO during one orbit. Synchronisation is lost if these two numbers are not equal to the number of periods between BC0 and SOG.

4 Timing adjustments

In order to have the trigger data well synchronized, the timing of the TTC BC0 commands distributed to the SyncTx must be adjusted. The method used to compute these time adjustments is based on the ideas pioneered by the RD12 collaboration [2]. The distribution of events as a function of the bunch crossing number (bunch profile histogram) is compared (correlated) with the expected LHC bunch structure to extract the time adjustment. The function of the Accumulator in the synchronisation circuits is to build the bunch profile histogram in-situ.

The operation of the Accumulator (ACCU) is the following:

- When the command BC0 is received, the ACCU address is set to zero;

- At each clock cycle, the ACCU content (at the current address) is incremented if the input data is above the noise threshold (programmable), and the ACCU address is incremented;

- When the SOG is reached, the circuit stops updating the ACCU content.

After a certain number of LHC orbits the ACCU contains the bunch profile histogram with enough statistics to be compared (in the DAQ controllers) with the expected LHC structure. In figure 4 we show typical examples of the bunch profile histograms. The first spectrum reproduces exactly the LHC bunch structure, which means that the command BC0 arrives to the SyncTx circuit in phase with the data that effectively is produced by the collisions occurring at bunch 0. The other two examples correspond to situations where the BC0 signal is either late or in advance relative to the data.

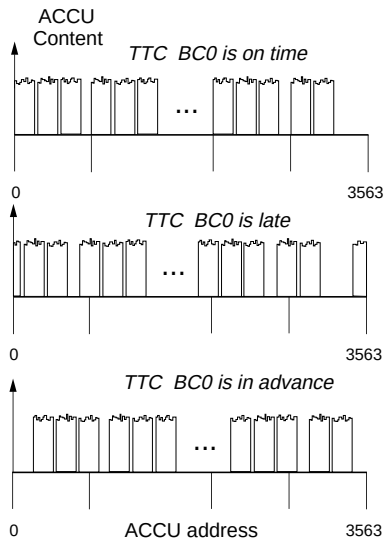

Figure 4 - Examples of expected bunch profile histograms.

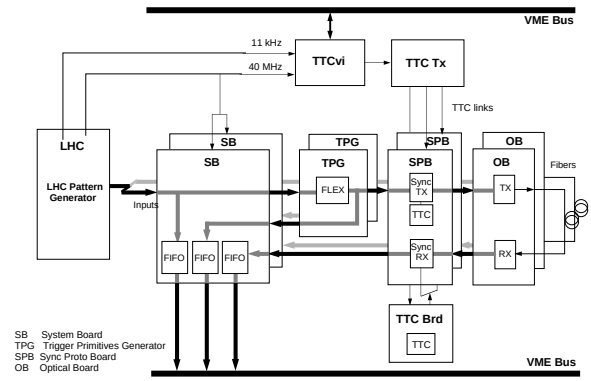

Figure 5 - The test setup of the CMS calorimeter trigger primitives system

The time interval needed to accumulate the histograms depends on the channel occupancy and on the statistics required. For example, a precise spectrum with about 1000 events per bin for a channel with occupancy 10^{-4} takes about 15 mn to accumulate.

The DAQ controllers can read the ACCU content without disturbing the circuits main operation. This facility allows to monitor the synchronisation during the data taking.

The time adjustment of the Common BC0 at the SyncRx is less critical. It is intended to verify that the BC0 doesn't arrive too soon, before the data from bunch 0 has arrived to all FIFOs, neither too late, in which case the FIFOs would reach the full state or the clear FIFO signal would happen before the end of the complete FIFO readout. In practice, the Common BC0 signal is advanced up to the point where one of the FIFO's reaches the 'empty FIFO' state. Delaying then the Common BC0 by one clock is enough to guarantee the operation point with the smallest latency.

5 Implementation

A prototype FPGA implementation of the synchronisation circuits was developed [7]. The SyncTx and SyncRx blocks were implemented in two XILINX circuits, the XC4006-E and the XC4013-E, respectively. The implementation of the Accumulators used a set of external RAMs.

The SyncTx and SyncRx circuits were mounted in a VME module, the Synchronisation Prototype Board (SPB), together with a TTCrx circuit. The SPB was included in the CMS Calorimeter Trigger Primitives System test set-up (figure 5).

The LHC Pattern Generator module generates 2x6 8 bits 40 MHz patterns which emulate the bunch structure of the LHC orbit. These patterns feed in parallel two trigger channels, composed of the Trigger Primitive Generator (TPG), the SyncTx, the Trigger Optical Link (1 Gbit/s) and the SyncRx. The TTC Board, housing a TTCrx circuit, distributes the Common BC0 to the SyncRx circuits. The System Board performs control and readout functions.

During the tests reported here the TTC system, not yet fully operational, was replaced by a simplified electrical version.

The first tests of the system were very successful. The circuits performed as expected and stable operation conditions were achieved. In figure 6 we show the bunch profile histograms collected in the Accumulators for different settings of the BC0 delay, demonstrating the system ability to flag the bunch 0. At the time of writing synchronisation stability tests are going on.

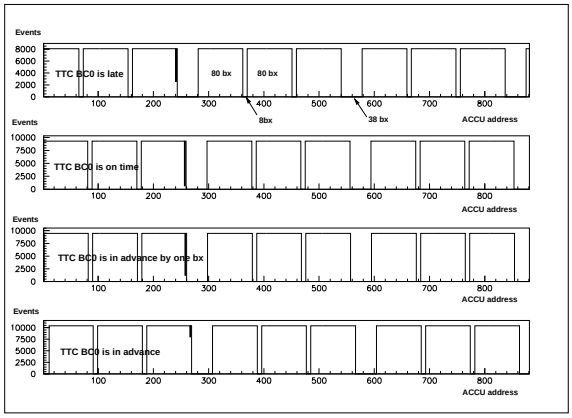

Figure 6 - Bunch profile histograms collected in the Accumulators for different settings of the BC0 delay.

6 Conclusions

We have presented the principles of the CMS scheme for synchronisation of the trigger data. The method makes use of the LHC bunch gap and allows a resynchronisation of the data every LHC orbit (88 ms). It relies on the distribution by the TTC system of a signal synchronous with the first bunch in the orbit, and is implemented by dedicated synchronisation circuits in each trigger input. Prototype circuits in a FPGA implementation were developed and integrated in the CMS Calorimeter Trigger Primitives System test set-up. The circuits performed as expected and stable operation conditions were achieved.

References

[1] Varela, J. A method for synchronisation of the trigger data. CMS TN/96-011.

[2] RD12 Status Report, CERN/LHCC 97-29 (1997) and http://www.cern.ch/TTC/intro.html

[3] G. Heath, these proceedings

[4] Heath, G. et al. Preliminary specifications of the baseline trigger algorithms. CMS TN/96-10.

[5] TTC-VMEbus Interface (TTCvi), RD12 Working Document.

[6] TTCrx Reference Manual, RD12 Working Document.

[7] Silva, J.C. et al. Specifications of the prototype trigger Synchronisation Tx/Rx Circuits, CMS IN 1997/009