

# Timing and Synchronization in the LHC Experiments

Varela, J.

LIP/IST, 1000 Lisbon, Portugal and CERN, 1211 Geneva 23, Switzerland  
Joao.Varela@cern.ch

## Abstract

Data synchronization is an important aspect in the operation of the trigger and readout systems of the LHC experiments. In this paper we discuss the current understanding of the problem as developed by the LHC collaborations. Some of the main synchronization issues, namely, the assignment of bunch crossing to data, the overall alignment of the trigger system and the synchronization of the front-end readout pipelines, are covered in some detail. We discuss the tools required for distribution of timing and control signals and for the fast collection of front-end status, as well as the functions performed by the central trigger control unit. Methods to determine and monitor the timing parameters in the experiments, the sources of synchronization losses and the recovery procedures are briefly surveyed.

## I. INTRODUCTION

The design of the front-end readout and level 1 trigger in the LHC experiments follows a synchronous pipeline model: the detector data are stored in pipeline buffers at LHC frequency waiting the L1 decision, while data from the calorimeters and muon detectors are processed by a distributed, pipelined, tree-like processor that computes the L1 decision. The L1 latency must be constant and shall match the pipeline buffer length. The whole system behaves synchronously.

Synchronization at different levels and in different contexts has to be achieved and monitored for proper operation of the system. In order to fix definitions, we list in Table 1 the various synchronization types that we refer to along this paper.

The trigger system is based on the assumption that at the input of every processing stage data are synchronized and belong to the same bunch crossing. The monitoring of the bunch number of trigger data flowing in the trigger pipeline is of the greatest importance. The system operates with a single clock frequency, the LHC frequency. However, the phase of the clock signal, after distribution to tens of thousands destinations, is unpredictable at the level of a few ns. When transmitting data between sub-systems re-synchronization of the data to the clock phase at the reception is in order. Much care has to be taken to avoid missing or adding one clock cycle to the transmission

latency. After the L1 accept signal, event fragments have to be collected to form complete events. Careful checking of event identifiers, recovery from event synchronization losses and management of buffer overflows are some of the issues in this context.

The setting of the experiments timing implies that programmable delays are adjusted to achieve synchronization and correct bunch crossing identification. The number of different delay parameters in the experiments is very large. A clear procedure to determine these parameters needs to be defined.

The sources of synchronization loss are numerous. The assignment of a pulse to a bunch crossing depends on the shape and jitter of the signals. Variations in the signal shape, large signal jitter, jitter of the clock phase, all can be at the origin of bad identification of the signal timing. The detector signals in a given crossing are distorted by eventual pile-up of signals from previous or following crossings. This effect will not only deteriorate the pulse height measurement but its timing as well.

For the first time a so large number of high-speed optical serial links will be put in synchronous operation. Any loss of link synchronization will have immediate consequences in the trigger and in the readout synchronization.

Table 1: Synchronization types.

Type	Description
Sampling Synchronization	Synchronization of the detector signals with the clock phase
Serial Link Synchronization	Recovery of parallel data words from the serial bit stream.
Bunch Crossing Identification	Assignment of a bunch crossing number to data in the trigger path
Trigger Data Alignment	Alignment of trigger data at the input of the trigger pipeline processors
Sub-Triggers Synchronization	Alignment of trigger data from different sub-systems at the Global Trigger input.
L1 Accept Synchronization	Synchronization of L1A signal with data in the readout pipelines
Event Synchronization	Assignment of bunch and event number to data fragments in the DAQ path

Synchronization errors due to single event upsets in the front-end pipelines, due to buffer overflows or due to errors in signals transmission are likely to occur. Efficient procedures to recover from synchronization losses need to be defined.

In Section II we discuss the tools needed for timing and trigger control, in particular the central trigger control unit, the trigger and timing distribution system (TTC system) and the fast monitoring network. Section III addresses the issue of system partitioning and Section IV covers the control and distribution of calibration and test triggers. The synchronization of the first level trigger is discussed in some detail in Section V. Finally, Section VI is dedicated to the timing procedures without beam and Section VII to the synchronization monitoring and diagnostic procedures.

## II. TOOLS FOR TIMING AND TRIGGER CONTROL

An overview of the trigger and timing context in the LHC experiments is shown in Figure 1. In this paper we refer to Trigger Control as the entity that supervises the distribution of trigger and timing signals.

The main task of the Trigger Control is to control the delivery of L1 Trigger Accepts, depending on the status of the sub-detectors readout electronics and data acquisition. This status is derived from direct information transmitted back by the subsystems as well as from state machines that emulate the front-end buffers occupancy [1].

The Trigger Control is also responsible to generate the Bunch Crossing Zero and other synchronization commands, as well as to control the delivery of test and calibration triggers. The distribution of the clock, trigger and fast signals to the experiments sub-systems is performed by the Trigger Timing and Control (TTC) network [2].

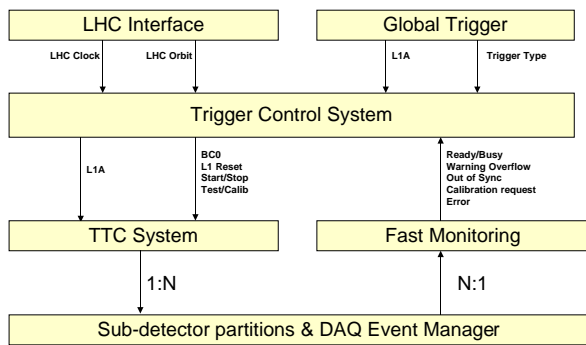


Figure 1: Overview of the trigger and timing control in the LHC experiments.

A deadtime monitor that tracks the disposition of each crossing, i.e. whether accepted, rejected, or lost due to downtime of trigger and/or DAQ should also be part of the central Trigger Control.

An interface to the LHC RF clock and orbit signal, provided by TTC components, and a beam backup interface to communicate with the beam pickup electronics is available to each experiment.

In summary, the functions of the Trigger Control can be organized as follows:

1. Throttling of the L1 Accept trigger signal.
2. Generation of fast commands distributed by the TTC network.
3. Handling of fast monitoring feedback from the sub-systems.
4. Generation of calibration and test trigger sequences.
5. Monitoring of the experiment dead time.

### A. TTC System

The LHC Clock and Orbit signals are distributed from the Preveessin Control Room to the LHC experiments through single-mode optical fibers. At the experiments Counting Room, the clock and orbit signals are recovered by circuitry (LHCrx module) in the TTC Machine Interface crate (TTCmi) [3]. At this level the clock jitter is expected to be of the order of 10 ps rms. Fanout modules (TTCcf) in the TTCmi crate are used to distribute the clock and orbit signals to the Trigger Control and to the sub-detectors TTC master crates (see Figure 2).

The TTC network provides for distribution of clock, trigger and fast control signals using two time division multiplexed channels transmitted over an optical passive distribution network. The channel A is used to transmit the

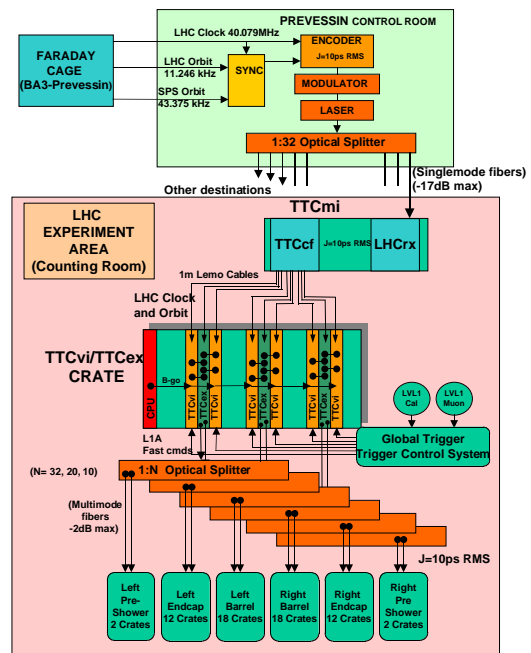


Figure 2: Architecture of the TTC System and interface to Trigger Control

L1A signal whereas channel B transmits other fast commands. Both channels have a bandwidth of 40 Mbit/s.

### 1. TTC VME Interface and Transmitter

The top elements of the each sub-detector TTC partition are the TTCvi and TTCex modules [4]. The TTCvi propagates the L1A and distributes various types of programmable commands. The commands are synchronous with the Orbit signal or triggered by front panel signals received from the Trigger Control (B-Go signals). The B-Go signals are associated with FIFOs where the actual commands coding are programmed. Each FIFO can be operated in repetitive mode. Asynchronous commands initiated by VME control are also possible. The data consists of 8 bits of broadcast commands and 32 bits of individually addressed commands. All of the data has Hamming error detection and correction codes.

The TTCex module encodes both channels A and B from the TTCvi and has a transmitter laser with sufficient power to drive the optical splitters. The clock from the TTCmi crate is plugged directly into the TTCex modules. At reception, optical receivers with a pin diode and an amplifier generates an electrical signal at the TTCrx chip input.

### 2. TTC Receiver chip

The TTC signals are received by the TTCrx chip which provides as its output the 40 MHz LHC clock, both raw and deskewed, the Level 1 Accept (L1A) trigger and fast commands data [5]. Deskewing is provided for the clock, the L1A and the broadcast commands. The coarse deskew is provided in 16 steps of 25 ns each and the fine deskew is provided in 240 steps of 104 ps each. The TTCrx tests to have a clock jitter less than 100 ps. The TTCrx was revised into a radiation hard version, which will have a hardwired ID at startup, boundary scan, an I2C interface with all registers read/write, and 64 possible user broadcast commands. The new TTCrx will have latency reduced from 100 ns to 70 ns.

### 3. Interface to Trigger Control

The Trigger Control sends the L1A and B-Go signals to each subsystem TTCvi module on a separate cable. In order to allow independent operation of the sub-detector partitions during setting-up and calibration phases, independently programmable trigger signals can be made available by the Trigger Control (see Section III). In normal operation a single trigger is distributed to all partitions.

### B. L1A Throttling

The Trigger Control must guarantee that sub-systems are ready to receive every L1 Accept delivered. This is essential to prevent buffers overflows and/or trigger signals missed when the sub-systems are not ready to receive them. In either case, the consequence would be a loss of synchronization between event fragments.

Warning signals sent from the sub-systems through the fast monitoring network, indicating that some of its buffers are almost full, may be received centrally. However this

feedback signal can take several microseconds to reach the Trigger Control, which meanwhile could have delivered a number of L1A signals that originate a buffer overflow. This problem is particularly acute in the front-end de-randomizers which have a small storage capacity.

According to the front-end electronic logical model, the front-end derandomizers after the L1 latency pipelines are the first devices to overflow when the L1A rate is too high. Space and power constraints in the front-ends imply small derandomizer depth and hence these queues are very sensitive to bursty L1A. In general, the derandomizers behave like a first-in-first-out queue: the input/output frequency is directly the L1A rate and the event readout time (detector dependent) ranges from 3 $\mu$ s to 7 $\mu$ s.

Simulations have been made showing the relation between the derandomizer depth, the event readout time and the probability to overflow [6]. The overflow probability is strongly dependent on the ratio between the service time and the buffer depth. A derandomizer with capacity for 8 events and a service time of 7  $\mu$ s per event, as in the tracker, has an overflow probability of  $2 \cdot 10^{-3}$ . The data acquisition inefficiency is negligible but the buffer would overflow every 5 msec. To reset the buffers at this frequency it is obviously not a good solution.

For a given sub-detector, all front-end de-randomizers behave identically. Therefore, its occupancy depends only on the L1A rate and on the service time. A state machine receiving the L1A signals can emulate the de-randomizer behavior and determine its occupancy at each new L1A. If a new L1A is estimated to cause a de-randomizer overflow, this L1A is throttled.

In general, it would be very difficult to guarantee that the state machine reproduces exactly the buffer status at every time. However in the present case the L1A accept signals are synchronous with the clock and the write and read latencies are measured in multiples of the clock period. It is this time quantization that makes the de-randomizer emulation really possible.

A complementary solution to the same problem is to oblige the delivery of L1A signals to comply with a set of trigger rules. These rules take the general form 'no more than n L1A signals in a given time interval'. Suitable rules, inducing a negligible dead time, would minimize the buffer overflow probability.

The occupancy of buffers following the de-randomizers in the data acquisition chain depends on the event fragment sizes after data selection (zero suppression or selective readout). Potentially, every buffer has a different occupancy at a given time. Therefore its occupancy can not be emulated in a central place. The trigger rules can reduce the overflow probability but are insufficient to completely prevent overflows.

The strategy to avoid overflows in this case could be based on the following points:

- a local buffer handler controls its occupancy and detects when the buffer is filled above a certain warning level;
- at this point a warning message is sent to the Trigger Control which stops the trigger;
- meanwhile the buffer handler stores 'empty events', a small data block containing just the event identification and a buffer overflow error flag;
- the storage of complete events is resumed when the buffer occupancy gets below the warning level.

The buffer safety region above the warning level can easily store a very large number of 'empty events', giving enough time for the fast monitoring feed-back loop to take place. On the other hand, the probability of reaching the buffer safety level times the number of buffers should not contribute significantly to the data acquisition inefficiency. These probabilities can be estimated with system simulation tools [7].

### C. Fast Controls

In addition to the L1A signal, the Trigger Control must be able to send to the sub-systems a number of other fast control signals for synchronization, fast reset, calibration or test purposes. The distribution of fast control signals is organized by sub-detector partition in order to allow independent operation of the partitions in test or calibration mode.

#### 1. Bunch Crossing Zero

The Trigger Control sends broadcast commands synchronous with the LHC orbit signal, to be used by the sub-systems as a basic synchronization tool. The Bunch Crossing Zero (BC0) command is used by the trigger sub-systems to flag the bunch zero data flowing in the trigger processing pipeline. It can also be used by the sub-detectors to localize the orbit gap when particular actions are needed in that period (e.g. sub-detector specific reset or test procedures). The BCR command is a special TTC command used to reset the bunch counter in the TTCrx chip. In most applications it is synchronous with data at the end of the front-end pipelines. The BC0 and BCR commands differs only by a constant phase of the order of the trigger latency and could be issued both centrally for sub-systems convenience. Detectors that do not participate in the trigger will need in principle only one of the two commands.

The phases of the BC0 and BCR commands relative to the Orbit signal are globally adjusted per sub-detector partition with a programmable delay in the TTCvi, and locally adjusted at the level of the receiver TTCrx in a smaller range (16 clock periods).

#### 2. Fast Reset

We can anticipate a number of circumstances during data tacking, in particular synchronization errors due to single event upsets in the front-end pipelines, due to buffer

overflows or due to errors in signals transmission, that will require a reset in order to recover normal functioning. The standard recover procedure involving a new run start may translate in an unacceptably high data acquisition inefficiency.

For this reason it is wise to foresee the possibility that the central Trigger Control distributes reset commands through the fast control network to the sub-systems. The reset command defines a time interval without triggers that can be used by the sub-systems to re-synchronize all readout components to the same event identifier or to recover from some hardware faults. These commands can be issued on a periodic basis, independently of the status of the readout electronics, or in response to a loss of synchronization identified in some sub-system. A reset request may be generated by a subsystem as a result of the detection of an error condition that affects a significant part of the subsystem. In parallel with the fast reset request, sub-systems log the error condition for monitoring purposes.

Different fast reset commands can be foreseen. One command could be interpreted as a re-synchronization of all sub-systems readout to the same event. Event and bunch counters as well as readout memories and pointers are reset. Another could be issued to recover from electronics malfunction. Its range of action depends on the particular module that receives it, but it could involve the reset of the readout state machines or the fast reconfiguration of programmable devices. None of these fast resets should involve a reprogramming of parameters controlled by software.

The reset sequence consists of the following steps. After receipt of an external request or the internal generation of a reset command, Level 1 Accepts are shut off. The reset command is broadcast via the TTC system. Upon receipt of the reset command, the subsystems assert the busy signal, read pending data in the readout buffers, reset pipeline and buffer pointers. The local event number counter is set to zero or the current event number is broadcast by the Trigger Control. When this operation is concluded the sub-systems drop the busy signal. After the reset procedure, the subsystems should be in a state to receive BC0 and L1A signals as at run start. The Trigger Control resumes triggers at the start of the next orbit.

Special reset procedures affecting one sub-system or part of it can be initiated by the sub-systems provided it preserves the ability to respond to central issued TTC commands. Tracking detectors may need to reset the front-end pipelines quite often due to the high probability of radiation induced SEUs. In order to allow for readout pipeline resets without events loss, the main orbit gap may be artificially extended to a size slightly larger than the trigger latency so that a reset can be issued at the end of the gap after the readout of remaining events of the previous orbit. The pipeline reset should not affect the TTCrx chips, and in particular the local Event Number and Bunch Number.

#### D. Fast Monitoring

The Trigger Control is able to receive status information from the sub-detectors readout and from the data acquisition system. These feedback status, collected by a fast monitoring network, could include ready, busy, out of sync and error signals.

The ready status is applied continuously to know that the system is connected and working. It must be different from the signal received when cables are unconnected or the electronics switched-off. The busy signal means that the system is preparing to take data and can not yet receive triggers. The Trigger Control inhibits LIAs during this time. Also when the readout buffers are close to overflow a warning could be sent to the Trigger Control so that further LIAs are inhibited until the buffers recovers enough free space. The out of sync signal indicates that event fragments collected in a given partition do not correspond to the same front-end pipeline position or have different event identifiers. The Trigger Control can then send a reset command to recover event synchronization.

The DAQ event manager sends back to the Trigger Control the same status signals as the sub-detector partitions. The busy signal is interpreted by the Trigger Control as a trigger inhibit. The trigger pipeline logic may run but does not send any LIA until the DAQ releases the busy signal to allow broadcasting of LIA signals. During the run the DAQ may send a busy signal to allow DAQ buffers to recover. Dead time counters in the Trigger Control are incremented until the end of the busy signal.

The granularity at which fast monitoring feedback is collected by the central Trigger Control is limited for practical reasons. It is reasonable to assume that one input per sub-detector partition is provided. The sub-systems should have programmable logic to determine when to issue the feedback signals (e.g. more than  $n$  modules in error implies that the error signal is send to the central Trigger Control).

The collection of fast status can be implemented according to two different models. The first is a tree-like structure with point-to-point connections and hardware state machines in the nodes combining the status of individual components. This model can be used by the sub-detector partitions since the latency of the feedback signals is short as required. An implementation of this model was developed by the ATLAS collaboration [8]. The second is a bus-like, message passing channel supported by software. This structure is more appropriate for the collection of feedback from the DAQ units where the timing constraints are not severe.

### III. PARTITIONING

It is a general requirement of the LHC experiments that sub-detectors and sub-detector main components have the possibility of operating in stand-alone mode, as

independently as possible, during setting-up, test or calibration phases. These components are installed and tested by different teams that need the largest possible autonomy to start and stop runs, reset the electronics, change high voltages, change programmable timings or trigger conditions, etc.

It is clear that sub-detector commissioning will need a non-negligible beam time during which sub-detectors may need different and mutually incompatible run conditions. Test and maintenance of some components may need to be done in parallel with stable physics running on the rest of the detector. In practice, this signifies that the Trigger and Data Acquisition systems should allow flexible configuration of independent partitions.

Partitioning into subsystems may be accomplished in two ways. The principal method is to maintain the full TTC tree and perform separate L1A for different subsystems in the global trigger. Each partition receives its own L1 trigger and the DAQ event manager requests readout of the partitions receiving the L1A signal. The Trigger Control is able to deliver fast commands to independent partitions, and is able to handle the fast feedback received from independent partitions. We call this mode of operation DAQ-Partition mode.

A second method is to shut off the link between the central Trigger Control and the subsystem TTC master crate. The DAQ processors supervising the Trigger Control and the subsystem TTC crates accomplish this shutoff. The subsystem TTC master crate then takes over the function of generating control data over the subsystem's TTC network. This method has a restricted use since it is only employed with local triggers and local DAQ readout. We call this mode of operation Standalone mode.

The L1 Trigger presents a special case in partitioning. Conceived as a subsystem of its own, alongside the various front-end DAQ subsystems, it nonetheless has components within it that are closely associated with each of the DAQ subsystems. For some types of testing, it is desirable to include both the front-end DAQ crates and their associated L1 Trigger crates in the same test partition. To accommodate this situation, the L1 Trigger system, along with the selected DAQ subsystems, can be controlled from the Trigger Control as a limited configuration. The remaining subsystems that are not in the configuration may be operated as separate partitions as described above.

#### A. Trigger Control Partitioning

As an example of possible implementation of Trigger Control partitioning we describe briefly the design developed by the CMS experiment [9]. The architecture of the CMS Trigger Control System (TCS) provides control of 32 physical partitions. Identical units implement the L1A and fast command generation, as well as the trigger throttling and calibration functions, for each sub-detector partition (see Fig. 3).

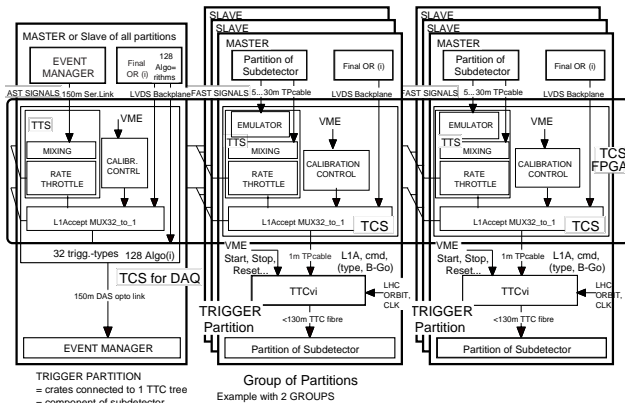


Figure 3: Partitioning of the CMS Trigger Control System.

Each unit can select one out of 8 different L1A signals provided by the Global Trigger. Each of these L1A signals combines a maximum of 128 trigger conditions. The TCS units send an L1A, a trigger type word and the B-Go signals to the respective TTCvi board.

The TCS units can be combined to form partition groups that wish to run common test and calibration procedures. In normal data taking all subsystems are connected to one group. Within one group one of the units is programmed as master and the other as slave. The master provides the L1A and the event number of the group.

The TCS unit for the DAQ Event Manager (EVM) runs either as the group master or as slave for all partitions. In normal data taking mode the EVM unit is master. As slave the EVM unit sends an OR of all partitions L1As to the Event Manager. The trigger type words tell the EVM to which subsystems the current L1A has been sent and how to read the next event.

In summary, the foreseen implementation of the CMS Trigger Control allows the following possibilities:

- i) The subsystem partitions may run either in single partition mode or combined in partition groups.
- ii) Partitions or group of partitions may run in parallel for calibration, tests and commissioning with beam.
- iii) The partitions are configured by the Run Control both in the TCS and in the DAQ Event Manager. The TCS and the EVM count the Event Number per partition or partition group. The DAQ system combines data from the subsystems of a group to build events.
- iv) Private readout of events triggered by L1As is possible: a partition may want L1A to read events by the local DAQ and does not require global DAQ resources.

#### IV. CALIBRATION AND TEST TRIGGERS

Calibration and test triggers can be delivered in several different contexts. A better understanding of the test and calibration requirements of the LHC detectors is now being developed by the collaborations. However we can anticipate that some of the scenarios listed below will be implemented:

1. Sub-detectors in Standalone mode: some detectors are able to generate test and calibration sequences using their own resources and capturing the data with the sub-detector local DAQ.
2. Sub-detectors in DAQ-Partition mode: the central Trigger Control generates test and calibration triggers at the rate required by the sub-detector partition and the data is collected by the central DAQ. In this mode the sub-detectors have access to a fraction of the CMS on-line computing and storage resources.
3. Periodic test and calibration triggers during a Physics Run: the triggers are issued centrally, the Event Number is incremented and all sub-systems deliver an event data block in order to keep the event synchronization (the event block can be empty if the sub-detector does not require test data). The data is collected by the central DAQ.
4. Test and calibration mini-runs during a Physics Run: the Trigger Control send fast commands to the sub-systems to mark the beginning and the end of a calibration mini-run. At the beginning and at the end of the mini-run some time is reserved for the sub-systems to set their front-ends in test/calibration mode and to restore the normal mode.
5. Test and calibration triggers handled at the sub-system level during a Physics Run: the sub-systems perform test, calibration or monitoring activities during Private Test periods defined by the Trigger Control.

##### A. Central Calibration Triggers

Test and calibration triggers issued centrally follow a well defined protocol. The Trigger Control sends to all subsystems a Test Enable signal a fixed number of crossings before the crossing when the test is to be done. This enables subsystems to prepare for the test (e.g. laser pulse generation). The subsystems set up their tests so that the test data will be contained in the exact crossing indicated by the Test Enable signal. The TCS then sends out a Level 1 Accept for the appropriate crossing to read in the test data.

The timing of the test and calibration triggers can be programmed in order to happen at pre-defined bunch crossing numbers in the LHC orbit. During physics runs, these triggers will occur during the LHC main gap but they can also be delivered in superposition to beam crossings to study pile-up effects.

If different types of tests are needed (e.g. laser pulse, electric pulse, pulses send to restricted geographical regions in the detector, etc.) sequences of different test commands are pre-loaded in the relevant TTCvi.

### B. Local Calibration Triggers

Some sub-systems have the ability to generate test and calibration signals by themselves. This feature is already being used in the test beams and will naturally be integrated in the experiment for tests in Standalone mode. In normal physics data tacking, these systems may be used provided that they not interfere with the main DAQ path.

For this purpose, the central Trigger Control may issue periodically the command Private Test that marks the start of a fixed period without triggers or other commands. The sub-systems may use this time interval to generate test/calibration signals at the top of the sub-system TTC. The Event Number is not incremented and the data is collected by the sub-system local DAQ. This possibility is used when the volume of test or calibration data is small and can be easily handled locally.

## V. TRIGGER SYNCHRONIZATION

### A. Synchronization of the Detector Signals with the Clock

All level 1 trigger and DAQ pipelines must be driven with a common clock synchronized to the LHC crossing frequency. This clock, distributed centrally, is phase locked to the LHC machine clock and has a 25-ns cycle. The processing of the Level 1 decision is driven by this cycle.

The phase difference between the LHC 40 MHz clock and the arrival of detector signals from collisions at the front-end electronics must be determined, adjusted for and monitored. The methods used to convert the analog detector pulses to digital information are detector dependent but all rely on the clock signal. The determination of pulse amplitude and timing, in particular the assignment of pulses to bunch crossings, depends critically on the clock phase initial adjustment and stability.

### B. Bunch Crossing Assignment of Trigger Primitives

The trigger primitive data (amplitude or pattern) for each trigger channel, and for each crossing, are transmitted to the regional trigger logic in digital form, the signals having a width equal to the LHC clock period (approximately 25 ns). These data is transmitted for every crossing, synchronously with the clock, even if most of the crossings it will be zero.

By bunch crossing assignment we mean the process that assigns the trigger primitive digital data to a certain clock cycle. In most cases, this process involves the treatment of detector pulses than span several crossings. The

performance of these algorithms is crucial for the trigger synchronization. The main requirement is that the offset between the time of a given beam crossing and the generation of the correspondent trigger primitives is constant.

In the calorimeter trigger path, pulse digital samples are processed by a digital filter in order to assign a pulse to a clock tick. The filter performs a weighted sum of a number of consecutive pulse samples and in parallel searches for a peak in the filtered pulse. Samples out of the peak are put at zero (see Figure 4).

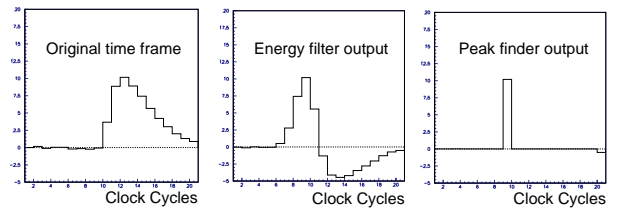


Figure 4: Digital filtering of calorimeter pulses in the trigger path.

The efficiency of clock tick assignment depends on the relative amplitude of signal and noise. It was shown that in the CMS electromagnetic calorimeter the efficiency is close to 100% for energy sums above 1 GeV. The efficiency is also affected by pile-up. For a signal of 5 GeV the efficiency remains 100% for pile-up energies up to 1.5 GeV [10].

In the muons Resistive Plate Chambers, the total signal propagation time in the upstream part, before bunch crossing assignment is done, has four components: the time of flight, the propagation time along the strips, the time of intrinsic RPC phenomena and the preamplifier and discriminator propagation delay. Some of these components have large irreducible jitter. The combined jitter of the time of flight and strip propagation time is of the order of 6 ns in the worst cases. The contribution from the front-end electronics is negligible, but the time of RPC response has a quasi-gaussian distribution with  $\sigma_{RPC} < 3.0-3.5ns$  as measured in recent RPC prototypes tests [11]. These data indicates that the correct bunch crossing can be assigned with 99% efficiency.

The timing structure of the drift muon chambers used in the CMS muon trigger has several components similar to those of RPC. The difference is that the chamber response time includes the drift time, which makes the distribution as wide as 50-70ns at the base, in the case of Cathode Strip Chambers, and about 400 ns in the case of the Drift Tubes. Because of that the bunch crossing identification is more difficult.

In the CSC case, a local trigger is based on a coincidence of at least 4 out of 6 layers within 75ns gate. The bunch crossing is identified by the second (in time) hit of those

contributing to the coincidence. Prototype tests indicate that the distribution of the second hit arrival time is fully contained within 20ns. In the Drift Tubes, bunch crossing recognition is performed by the Bunch and Track Identifier (BTI) circuit, using a generalized meantimer technique [12].

## C. Trigger Alignment

### 1. Trigger Pipeline Alignment

The basic architecture for the Level 1 trigger is that of a fully pipelined structure with a 25 ns clock. The result is a complex structure in which raw trigger data flows to the L1 Trigger logic from a number of front-end subsystems, each at a different offset with respect to the absolute bunch phase. Each trigger decision subsystem in turn has its own offset with respect to the front-end data as well as the other trigger decision subsystems. At the global L1 Trigger, the remaining offsets between trigger decision data streams are reconciled to a single offset [13].

The L1 Trigger logic has the capability to provide a programmable multi-clock buffer delay on data that they transmit to or receive from other logic. This delay is necessary to compensate for the different inherent processing latencies in the different logical units and different cable lengths. With these capabilities, it is possible to adjust the timing delays of convergent data streams as necessary to guarantee the proper alignment of data for trigger decision calculations.

When the signals are sent to another board they might have a constant shift in phase in respect to the local clock at the destination. The rule to be followed is to synchronize the phase of the signal at the destination to the local clock.

### 2. Alignment of TTC

Offsets exist between individual subdetector crates for the distribution of TTC data. These offsets reflect mainly the difference in cable interconnection lengths between those crates on the detector and those in the counting room. Thus, each crate is assigned offsets which reflect both its position in the trigger decision pipeline as well as its distance from the central Trigger Control.

The TTC system has various adjustable delays. At the top of the TTC partition, the TTCvi module allows to set programmable delays on all fast commands except on the L1A signal for latency reasons. In particular, these delays allow to adjust globally the timing of the BC0 command, as well as the calibration and reset commands distributed to the front-ends.

On the front-ends, the TTCrx receiver provides for fine adjustment of the clock phase and for coarse adjustment (in bx steps) of the L1A, BC0 and other fast commands.

## D. Alignment with LHC Crossings

One LHC orbit consists of 3564 periods. They are often called “bunches” although some of them do not contain protons. The proton bunches are grouped in trains, 72

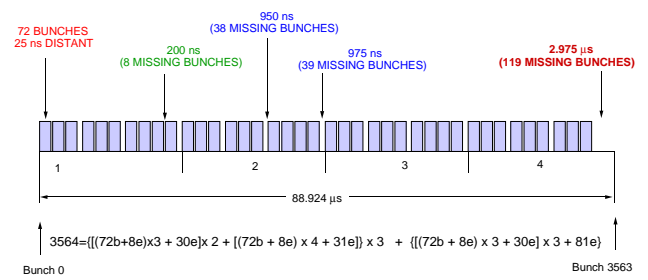


Figure 5:LHC beam structure

bunches each (Figure 5). The structure of gaps between them can be used for the absolute synchronization of trigger and DAQ data. Histograms of channel or group of channels occupancy per bunch crossing number are used for this purpose. Empty bins in the histogram are made to correspond to the gaps of the LHC beam structure by adjusting the necessary delays [14].

In the calorimeters, energy deposition above the noise threshold is clearly associated to particle collisions so that the histogramming technique provides very reliable absolute synchronization. In the CMS case, the histograms are incremented at LHC frequency using dedicated synchronization circuits in the ECAL and HCAL readout and trigger primitive boards [15]. The content of the histogram is accessed by the crate CPU where the correlation function between data and the expected bunch profile is computed. Misalignments are compensated by a corresponding number of steps in synchronization delays.

The threshold for histogram incrementing is set at 1GeV, in order to guarantee efficient bunch crossing assignment by the digital filter. It is estimated that at low luminosity the determination of the timing alignment constants for all the ECAL channels will take about 2 hours of beam time [10].

In the muon detectors, the bunch profile histograms are based on track segments identified within a muon station rather than on single hits, because of an important neutron induced background that is not time correlated with the interactions. Provided relative synchronization is achieved within each station, the absolute synchronization of the muon detectors can be achieved with the histogramming method.

The case of RPC is more difficult, because there is no local coincidence within one muon station. The only place where the neutron background can be suppressed is the pattern comparator processor, looking for coincidence of RPC planes. This implies that the RPC synchronization with real data must involve the trigger. That, in turn, means that the first iteration, done without the beam, should be precise enough to enable trigger to work with at least 10% efficiency.

The rate of muons, especially in the barrel, is relatively low. At luminosity  $10^{33} \text{cm}^{-2} \text{s}^{-1}$  the muon rate is about 30Hz per chamber in the worst case, i.e.  $\sim 1800$  muons / minute /



chamber. In one hour of running a bunch profile histogram with about 40 counts per bin can be accumulated [16].

## E. Alignment with Readout

### 1. Alignment of L1A with Readout Pipelines

In the readout chain, we are concerned with the synchronization of the L1A signal with the pipeline data. The L1 trigger signal is broadcasted to all channels through the TTC system. The TTC receivers assign to each trigger a double identifier: the event number and the bunch crossing number. The event number is counted since the last event counter reset and the bunch crossing number is counted since the last bunch counter reset.

Three parameters can be adjusted to achieve pipeline synchronization: programmable delays on the L1A and BCR signals and the pipeline length. The goal is that the trigger and the data extracted from the pipeline correspond to the same bunch crossing.

The value of the delays can be established using one of the following methods.

#### 1. Multi-crossing readout of real data

This method is especially suitable for low occupancy detectors participating in the trigger. A given detector region is read out if there was a L1 Accept caused by the data from this region. Several consecutive bunch crossings are read out in order to discover a possible misalignment of data with respect to LV1 Accept.

#### 2. Histogramming real data

This method is similar to the trigger bunch crossing synchronization with real data. Whenever there was a trigger, the data are stored in a histogram according to the bunch number given by the trigger. Possible misalignment can be detected comparing the histograms to the LHC bunch structure.

The last method can be used by any detector, not necessarily participating in the trigger, e.g. by the tracker detectors. High occupancy is of advantage, because needed statistics can be collected faster. Background not correlated in time with the collisions (e.g. loopers in the tracker due to magnetic field) is an obstacle to the precision that is finally achieved.

Various subsystems may need a special fill of the LHC in order to establish synchronization. A special fill would have a series of empty crossings before a single full crossing in order clearly identify a specific point in the pipelined samples. This pattern would be repeated as much as possible.

### 2. Front-End Event Identifier

The sub-detector event fragments delivered to the Data Acquisition are identified with a standard Front-End Event Identifier. This identifier contains at least the Event Number

and the Bunch Number. It could also contain the Orbit Number and the Event Time.

The Event Counter in the TTCrx has 24 bits and completes a new cycle every 168 sec (at 100 kHz L1A rate). This time interval is larger than the readout latency of the entire trigger and data acquisition chain. The Bunch Counter with 12 bits counts the periods within one LHC orbit (3564 periods). The Event Time defines the absolute time of the L1A and is used to correlate the event data with the slow control data.

### 3. Synchronization of Event Fragments

The event fragments are synchronized after all the necessary timing constants are adjusted correctly and hence corresponding event fragments are labeled by identical event and bunch crossing numbers, where the emphasis has to be put on the “and”. This consistency is checked during the transfer of the data towards the data acquisition system.

## F. L1 Latency

The trigger latency must remain within a well defined boundary set by the smallest pipeline buffer in the readout system.

There are many contributions to the latency. These include time of flight to the detector; propagation of signals within the sensitive elements of the detector; signal processing and trigger primitive generation times; cable runs within the detector hall and from the detector hall to the control room; time to form regional trigger components; time to make the global trigger decision; and time to distribute the L1 trigger accept signal back to the electronics in the front end.

An appreciable portion of the latency is in fixed cable delays, in particular the cables between the detector cavern and the electronics room. In order to minimize the cable latency dedicated straight paths for trigger cables are foreseen in the underground layouts. A good understanding of the cable lengths is crucial.

## VI. TIMING SETUP WITHOUT BEAM

The final synchronization of the trigger and sub-detector readout systems will be achieved using beam collisions as described in the previous section. However, before colliding beams are available, various timing setup procedures are necessary.

### A. Cable Lengths

All cables and fibers, including those of laser or test pulse monitoring systems, shall be measured before installation and their length stored in a database. The programmable timing parameters (deskewing parameters, pipeline lengths, etc.) will be initially adjusted to compensate differences in cables lengths. It should not be too difficult to achieve a precision better than 5ns, which corresponds to ~1m of cable.

### B. Timing of TTC distribution

Special care should be taken with TTC fibres. Good knowledge of their length will facilitate synchronization with test data. The initial adjustment of the BC0 deskewing is based on the knowledge of the cable lengths, both for data and control signals distribution, to the various levels (front-ends, trigger primitive generators, regional triggers, global trigger). The goal is that the BC0 timing at different levels is synchronous with data from bunch crossing zero flowing in the system.

### C. Timing of Trigger Pipeline

The timing of trigger pipeline, that is the adjustment of the delays needed for the synchronous behavior of the trigger electronics, will be made initially with test patterns.

Test patterns are generated at the source (e.g. front end boards) and transmitted on a request broadcasted by the TTC. At the destination (e.g. trigger processor board) they are compared to the generated ones. Let us consider a simple example — a sequence (00100) sent through all channels. The same sequence should be observed at the destination, namely the “1” should come at the same time, defined by the bunch crossing number provided by TTC.

Generated test patterns should unambiguously mark one bunch crossing. Let us denote by  $N$  its number given by the TTC at the source. Again one can use the sequence (00100) as an example. The “1” is sent in bunch crossing  $N$ . The timing of the test signal at the destination is adjusted in such a way, that the “1” is received in bunch crossing  $N$ , according to the local TTC.

Obviously, this timing procedure is limited by the ability to generate test patterns synchronously at different points. Before using real data this can be done only approximately, taking the bunch crossing number provided by TTC at front end as a reference. Precision of the method is limited by the knowledge of all delays in the TTC network.

### D. Timing of Readout Pipelines

Detectors participating in the trigger (calorimeters and muons) are first aligned at the level of the electronics chain. A test pattern generated at the input of the trigger boards propagates through the system and originates a L1A signal, which after distribution will trigger the readout of a time frame (multi-crossing sequence) in the readout pipelines. The identification of the position of the test pattern in the time frame allows to measure the trigger latency and to adjust in consequence the pipeline length or the L1A deskewing. This procedure is repeated for all channels, in order to check that the trigger latency is the same irrespective of the trigger geographical origin.

Detectors not participating in the trigger are aligned at the level of the electronics chain using test triggers delivered by the Trigger Control. The timing of the test pulse is adjusted, relative to BC0, in such a way that the test pulse simulates data produced at bunch crossing  $N$ . The latency of

the corresponding L1A is adjusted to reproduce the latency of the physics trigger capturing data at crossing  $N$ .

## VII. MONITORING AND DIAGNOSTIC PROCEDURES

### A. Data and Trigger Links

The synchronization of the detector links is monitored in permanence by the deserializer circuits at reception. If the link desynchronizes the circuit sets a flag that accompanies the data in the readout pipeline. This flag stays up until synchronization is recovered, so that all data produced during that interval is flagged.

In case of the trigger links, when a loss of synchronization is identified by the deserializer a bit is set which causes the corresponding trigger channel to be masked.

In parallel the local board controller at reception is requested to log the error and to initiate a synchronization procedure. A command is sent to the transmitter side which instructs the serializer to send synchronization patterns during a fixed interval of time. This procedure can be executed during data taking without the need for a full system reset.

### B. Bunch Crossing Identification

The bunch crossing identification is monitored in permanence with the bunch profile histograms accumulated on dedicated circuits or on the crate controller using spying events. The monitoring can be done channel by channel or per group of channels.

The status of bunch crossing identification is stored periodically in a database. The recovery from bunch crossing mis-assignments implies an adjustment of the BC0 deskewing in the relevant TTCrx circuit or a reprogramming of the concerned synchronization pipeline length. This operation requires a system reset controlled by software.

### C. Synchronization between L1A and pipeline data

The synchronization between L1A and pipeline data is monitored with real events verifying the matching between global trigger data and detector data. Matching data from different sub-detectors provides an important cross-checking of the detector synchronization.

On the other hand, profiting from the readout of time frames,  $Z^0$  candidates triggered by the single lepton trigger could be reconstructed in space-time in order to check possible misalignments between detector regions. This method is potentially quite powerful but requires that the detector is already calibrated, at least at a level of precision sufficient for good  $Z^0$  reconstruction.

The recovery of L1A-pipeline synchronization implies an adjustment of the L1A deskewing at the level of the

TTCrx circuits concerned or an adjustment of pipeline lengths. This operation requires a system reset controlled by software.

#### D. Synchronization of event fragments

A number of transient faults can be at the origin of a loss of synchronization between event fragments collected by the readout and data acquisition systems. An error is detected when an inconsistency between event identifiers of different event fragments collected at a given point is found. The important principle is that the checking of the synchronization is done in the same direction as the dataflow. The individual front ends do not receive the total crossing and level 1 accept numbers, but instead count these quantities and forward the phases of these counters to the subsequent levels which then check them.

Recovery from this synchronization loss can be achieved with a L1 reset signal distributed by the TTC system, as described in Section II. A safe set of actions to be followed in these cases are to insert an error flag in the event fragments transmitted to DAQ, to log the occurrence for monitoring by the local crate processor and to send a L1 reset request through the fast monitoring network.

### VIII. CONCLUSIONS

Timing and synchronization is a major issue for the LHC experiments. A substantial amount of R&D was carried out by the collaborations to understand the problem and to develop the tools that will be need. A common timing and trigger distribution system developed at CERN shows the required performance and will be used by the four experiments. Methods to set the timing of the experiments and to monitor the trigger and data readout synchronization were developed. A better understanding and more practical experience on synchronization is now being obtained with the operation of large detector components in the LHC-like beam at the CERN SPS [17].

### IX. ACKNOWLEDGMENTS

I would like to acknowledge J. Christiansen, S. Cittolin, D. Edmunds, Ph. Farthouat, A. Marchioro, A. Racz, J. C. Silva, W. Smith, A. Taurok and G. Wrochna for a number of useful discussions on the subject of this paper.

### X. REFERENCES

- [1] W. Smith, "CMS Synchronization Workshop Conclusions", CMS IN-1999/016.
- [2] B. Taylor, "TTC Distribution for LHC Detectors", IEEE Trans. Nuclear Science, Vol. 45, 1998.
- [3] B.G. Taylor, LHC machine Timing Distribution for the Experiments, these Proceedings.
- [4] Ph. Farthouat, P. Gallno, "TTC-VMEbus Interface TTCvi", RD12 Working Document, CERN.
- [5] J. Christiansen, A. Marchioro, P. Moreira and T. Toifl, TTCrx Reference Manual, RD12 Working Document, CERN.
- [6] A. Racz, "Trigger Throttling System for CMS DAQ", these Proceedings.
- [7] C. Tully, J. Varela, J.C. Silva, G. Varner, "Study of the ECAL Data Concentrator", CMS IN-1999/012.
- [8] P. G. Gallno, "Timing, trigger and control distribution and dead-time control in ATLAS", these Proceedings.
- [9] W. Smith, A. Taurok, J. Varela, C. Wulz, "The CMS Trigger Control System", CMS Note in preparation.
- [10] R. Benetta et al., "Synchronization of the ECAL data", CMS IN-2000/006.
- [11] A. Ranieri, "A New Front-End Board for RPC Detector of CMS", CMS NOTE-1999/047.
- [12] R. Cirio, R. Martinelli, L. Ventura, C. Willmott, P. Zotto, "A synchronization procedure for the muon drift tubes detector", CMS IN-2000/031.
- [13] A. Taurok, "Phase Synchronization of Trigger Data in the CMS Level 1 Global Trigger", CMS IN-1999/010.
- [14] J. Varela, "A method for synchronization of the trigger data", CMS NOTE/1996-011
- [15] L. Berger, R. Nóbrega, J.C. da Silva, J. Varela, "Trigger synchronization circuits in CMS", in proceedings of 'Electronics for LHC experiments', Oxford, Sept 97.
- [16] G. Wrochna, "Synchronization of the CMS Muon Detector", CMS CR-1998/017.
- [17] J. Varela, "Using an LHC-like Test Beam to Study the Trigger and Front-End Readout Synchronization of the CMS Detector", CMS IN 1998-012.