THE CERN LHC CENTRAL TIMING, A VERTICAL SLICE

Julian Lewis, Pablo Alvarez, Jean-Claude Bau, Stephane Deghaye, Ioan Kozsar, Javier Serrano
CERN Geneva

Abstract

The design of the LHC central timing system depends strongly on the requirements for a Collider type machine. The accelerators in the LHC injector chain cycle in sequences, each accelerator providing beam to the next as the energy increases. This has lead to a timing system where time is divided into cycles of differing characteristics. The LHC timing requirements are completely different: there are no cycles, and machine events are linked to machine processes such as injection, ramping, squeezing, physics, etc. These processes are modelled as event tables that can be played independently; the system must also provide facilities to send asynchronous events for punctual equipment synchronization and a real-time channel to broadcast machine information such as the beam type and its energy. This paper describes the implementation of the LHC timing system, and gives some details of the synchronization of its injector chain, which manufactures various beams for LHC**.**

INTRODUCTION

This paper describes the LHC slow timing, it should be noted that there is a second Beam Synchronous Timing system [1] BST that distributes timing using the LHC revolution frequency and the RF bunch crossing frequency. The BST is not within the scope of this paper.

The LHC mode determines what the machine is doing; *injecting-beam*, *ramping-up*, and *physics*, are examples of modes. The LHC *mode* is controlled via a state machine [2] through the LHC Software Architecture LSA by the operations team. Multiple timing event tables for each mode are downloaded from the LSA timing service into the LHC central timing, and activated according to the machine mode and operational requirements. Each event table corresponds to a task that runs on a hardware Multitasking event generator module, the MTT [3]. Event table tasks can be loaded, unloaded, started, stopped and synchronized independently to produce an event stream that is distributed around the LHC machine to timing receivers located in end user equipment. The LHC event stream distributes the UTC time; a snapshot of the machine state in the form of a *telegram*, the 1KHz slow clock, and machine events such as *start-ramp*, *dump-ring*, *post-mortem*. There are also external asynchronous events that encode the beam energy and intensity in each ring, and other machine safety critical information such as the safe beam flags. As the LHC timing distributes safety critical information, it must be continuously monitored so that any detected failure will provoke a beam dump through the Beam Interlock System [4].

HARDWARE

A Symmetricom Xli Time and frequency system disciplined by a GPS input provides two basic clocks, namely the Pulse Per Second PPS and the 10MHz, all other clocks are derived from these throughout the CERN central timing. These clocks are fed to synchronization modules in the LHC master timing generator and its hot standby, where the PPS is delayed by (1s–100us) to effectively produce a PPS advanced by 100us. The synchronization module also multiplies the 10MHz via a Phase Locked Loop PLL to produce a 40.00MHz clock, which is used by the MTT to Manchester encode the event stream at 500Kbits per second, to produce eight 32bit event frames per millisecond. The MTT encodes and transmits the UTC time at the advanced PPS tick. The 100us offset together with the transmission delay will be compensated for at the CTR timing receivers [5] to realign the output PPS tick with UTC. A portable Symmetricom CS4000 Caesium atomic clock measures the real PPS coming from the Xli source. It is then moved to remote locations to measure the transmission delay at the timing receiver outputs. The CTR timing receivers recover the encoding 40MHz clock with 25ps RMS short term jitter via a hybrid PLL [6] and use this recovered clock to ensure the local PPS is within 25ns of the original Xli PPS. This method of one time calibration will not take into account long term jitter caused by temperature variations. In exceptional circumstances, where continuous calibration is required, an extra fibre may be used to measure the round trip time; hence providing continuous calibration measurements. Studies are underway to obtain a method of continuous calibration on a single fibre.

The Controls Timing receiver CTR

A full description of the CTR timing receiver can be found elsewhere [5] [6]. The CTR implements timing reception logic and fully programmable 50MHz counters capable of producing output pulses for end user equipment and bus interrupts to synchronize host processor tasks. An onboard hybrid PLL [6] is used to recover the 40.00MHz encoding clock, which is used to count 25ns increments within the UTC second. Logic in the FPGA permits the CTR to configure counters according to the incoming timing event frames and telegram, and to maintain UTC time to provide time stamps with 25ns precision. When required, a CERN High Precision Time to Digital Converter chip HPTDC [5] can be installed on the CTR board to divide the 25ns time slice into 32 equal slices, providing time stamps of higher precision.

Reflective Memory

Failures in the LHC timing should be rare and of short duration, hence two master timing VME crates provide a live timing system and a hot-standby. A manually operated switch permits switching between them without disturbing the accelerator. A gateway computer implements a Front End Software Architecture FESA API across which the LHC timing is controlled by LSA. It is important that actions carried out on the two timing generator crates occur simultaneously to guarantee they are always in the same state. As we have had positive experiences with the VMIPMC-5565 reflective memory from GE Fanuc on the LHC injector chain central timing, and have already developed a driver, library, daemons and test programs, we use three of these high speed 2GBit/S memories connected via a VMIACC-5595 single mode hub for communications between the gateway and the timing generator crates. The hub permits, with the correct software, cutting power and rebooting the hot-standby or gateway crates without disturbing the reflective memory or its token ring. The token ring transmits data when memory is write accessed, thus when a memory node comes on line after a cold start, the host daemons must engage in a protocol to ensure the new memory node is correctly initialized. This is achieved by a DMA that copies memory onto itself on the nodes already running to invoke data transfers that update the new node.

Safe Machine Parameters SMP verification

The energy and intensity determine the damage that the beam can inflict on LHC equipment. Once the potential damage exceeds a threshold, a safe beam flag is set to false and masked machine interlocks can not be ignored. Any bad interlock provokes a beam dump via the Beam Interlock System [4]. The SMP acquires the beam energy and intensity per ring each 100ms. It calculates the safe beam flags for each ring and provides them as a 10Hz event stream, along with the beam intensities and energy. The stream is sent to the central timing generator crates, where they are forwarded over the general machine timing. The SMP continuously compares the values of the flags, energy and intensities it sent with those it receives back from the timing distribution. If the difference exceeds a threshold, or events are not received, the SMP signals the Beam Interlock System, which removes a beam permit flag and dumps the beam. The beam permit flags are provided as TTL inputs to the central timing. A negative transition will provoke a *beam-dumped* event and possibly trigger a post-mortem request.

Failures in transmission of critical data, such as a bad connection to the timing distribution, must be detected by the timing receivers. The CTR implements a monostable output that removes the flag when it does not receive events each 100ms. The CTR is also capable of driving two P2 connector bus lines to serially encode the beam energy for direct use by end user equipment.

Central Beam and Cycle Manager CBCM

The CBCM [7] provides sequencing and timing for all CERN accelerators, except for the LHC. In particular, it controls the machines that form the LHC Injector Chain LIC. The LIC currently consists of the proton and ion Linacs; the low energy ion ring LEIR, the 1Gev proton synchrotron booster PSB, the 26Gev proton synchrotron PS and the 450Gev super proton synchrotron SPS. In order to fill the LHC rings with beam, the CBCM must orchestrate the cycles of the LIC accelerators so that beam is delivered to an LHC ring. When not filling the LHC, the CBCM may pilot the LIC accelerators to provide beam for other users, such as the Grand Sasso Neutrino experiment CNGS, SPS fixed target physics and other CERN machines such as the antiproton decelerator ADE or the isotope facility ISOLD.

All beam transfers between the LIC and the LHC must first be requested by LSA via the FESA API on the LHC timing gateway. Therefore a small number of signals must be exchanged between the two timing systems. These exchanged signals determine the number of PS batches that will be injected into the SPS, the target LHC ring and the beam request.

Once the SPS ring has been filled with the requested PS batches, the SPS extraction forewarning event is converted into the LHC injection forewarning by the LHC timing to trigger injection and beam transfer.

The Master Timing multi Task module MTT

The MTT hardware is fully described elsewhere [3]. The MTT module implements a general purpose microprocessor capable of scheduling 16 co-routine tasks. The MTT hardware executes one instruction of each active task in turn; tasks that are not running do not use processor time slices.

Each task is provided with its own set of local registers to keep private variables in, for example loop counters. There is a set of global registers accessible by all tasks and a set of memory mapped IO registers that map specific hardware, such as the VME P2 connector, the timing event frame output register and the millisecond register. All registers, local, global, and mapped IO, are accessible across the VME bus from the host processor.

Each task is controlled via a Task Control Block TCB that contains the tasks start address in program memory, its current program counter PC and the current task and processor status words.

The MTT instruction set provides conditional branching, arithmetic and logical instructions, and relative and absolute wait on register instructions. There is also a variety of move instructions for literal values, register to register and indexed. An interrupt instruction allows a task to interrupt the host processor when it needs attention.

Post-mortem events

The post-mortem event freezes buffers in the front end processors which subsequently pump gigabytes of data for processing to the post-mortem system; this is a costly process that should only be triggered for unforeseen beam dumps.

The beam permit flags for both LHC rings are delivered as TTL signals to the master timing generators. A negative transition of either flag normally results in a *beam-dumped* event being sent out for that ring and post-mortem trigger event(s). Note that there is a single post-mortem event for both rings. Partial post-mortems for a single ring are not supported. The trigger for the MTT to send beam-dumped, or post-mortem events is a negative transition on one of four corresponding bits on the VME P2 connector. A task running on the MTT monitors these bits (2 for beam-dumped, and 2 for post-mortem) and dispatches the corresponding event. The post-mortem trigger bits cause the post-mortem event to be sent, so when both rings are dumped, the event is sent twice. For a deliberate beam dump, where the post-mortem should not be sent, the corresponding bit arriving on the P2 is conditioned. This is achieved via 2 channels of a CTR timing receiver. It conditions its response to a falling edge of a beam permit flag on its inputs via timing events, before passing its output onto the P2 connector. Enable and disable post-mortem events may be embedded as needed in a timing table to disable the post-mortem response in one or both rings in a defined time window. If a dump occurs in this time window no post-mortem will be triggered.

SOFTWARE

The LSA timing service accesses the LHC timing gateway across a FESA API. The gateway may modify reflective memory to effect changes in the live and hot-standby master timing generators, and hence control the event stream being sent to the LHC machine.

The requirements for the LHC timing API are, to control timing tables, to send out events promptly on request, to control the contents of the LHC telegram and to access status and diagnostic information.

The LHC machine state is controlled through the mode parameter. As the mode can change at any time it must be immediately sent as a real time event on change, however, as the mode may remain constant for minutes or even hours, it is important to retransmit this information regularly as a telegram parameter each second. There are many such parameters in the LHC telegram. The LHC is essentially a UTC machine and has no inherent fixed cycle duration or basic period. Telegram parameters are a snap shot of the LHC machine state taken each second, and each parameter is shadowed by an event broadcast at the exact moment the parameter changes. Only those events that describe machine state are shadowed by telegram parameters. Thus control processes around the LHC machine may subscribe to events for prompt actions and may also read the telegram to obtain the current machine state.

MTT Tasks and event tables

The MTT is able to schedule up to 16 tasks that comprise two permanently running system tasks and event table tasks under LSA control.

One system task monitors negative transitions on the P2 connector and sends post-mortem and beam-dumped events accordingly. A second task monitors global registers containing the LHC telegram. It sends an immediate event out on change and the telegram parameters each second.

Event tables are named and can be loaded, unloaded, started, stopped and aborted under LSA control. Each event has a *name*, for example *start-ramp*, and an event *code* that can be used to identify it to client software. The event frame encoded on the timing cable has two parts, a *header* that identifies which event it is to the hardware, and a *payload*. For example the mode event carries the mode in its payload; while the beam energy event payload carries the energy encoded in 120GeV units. Telegram parameters are encoded using the same payloads and telegram headers.

An event table is a list of entries containing: the event name, a relative time to wait and a payload. On update it is compiled into position independent object code and loaded into the MTT program memory. A task control block is initialised with its start address and PC. It is then added to the MTT task run list. The task now waits for its run count and a synchronization event to be set from LSA. It is possible to specify an indefinite run count so the task will continue to run until it is either stopped or aborted.

A task can be stopped by setting its run count to zero: when it loops back to the run count test it will wait until the run count is positive. Each time the task loops, it must wait until the *event-out* memory mapped IO register has its synchronisation event frame in it. A task can be aborted by removing it from the run list of MTT tasks. Aborted tasks stop dead, while stopped tasks continue to output events until they loop back to the run count test.

REFERENCES

[1] An FPGA based multiprocessing CPU for Beam Synchronous Timing in CERN’s SPS and LHC. ICALEPCS 2003

[2] Functional specification LHC Modes CERN EDMS 865881 Mike Lamont, Reyes Alemany, Stephen Page

[3] The LHC central timing hardware implementation ICALEPCS 2007

[4] The Beam Interlock System for the LHC. CERN EDMS 567256

[5] Nanosecond Level UTC Timing Generation and Stamping in CERN's LHC. ICALPECS 2003

[6] PLL Usage in the General Machine Timing System for the LHC ICALEPCS. 2003

[7] The Evolution of the CERN SPS Timing System for the LHC Era. ICALEPCS 2003