7 The z-by-Timing Data Acquisition System

7.1 Overview

Local data acquisition of the z-by-timing data is supervised by a Readout Controller, one per crate. Its purpose is to read out and zero-suppress the data, perform calculations on the data (in particular SLT processing) and to pass the data onto the ZEUS Event Builder for subsequent processing. Timing control is provided by a single Master Timing Controller interfaced directly to the GFLT, and 16 Local Timing Controllers, one per z-crate.

The Timing Controllers have the following functions :

- To synchronize data from the 16 z-crates.
- To interface with the GFLT.
- To control the pipeline and buffers of the front-end electronics.
- To interface with the DAQ software.

Fig. 15 is a schematic diagram showing the functionality of the timing system and readout control. The Master and Local Timing Controllers provide timing to both the z and the CTD FLT systems. The design philosophy of MTC/LTC communication is one of simplicity and robustness; all timing is derived from a single clock and the only control signal returning from the LTC to the MTC is a 'Busy'.

7.2 The Readout Controller

The CTD DAQ and SLT processing system is based on Transputer-based ROCs [16, 17], each of which controls a single crate. The z-crate ROC is a single 9U card which contains two T425 Transputers with a total of 3 Mbytes of external memory and three 'TRAM' modules, each containing a T800 Transputer with 1 Mbyte of external memory [22]. The ROC operates a backplane bus to download parameters to and read data from the front-end cards. Each Transputer is equipped with a 10 MIPS processor, on-chip memory, and four bidirectional 20 Mbit/s serial links which make it ideal for the application of SLT parallel processing. In response to an interrupt from the LTC, readout of the crates is carried out by one of the T425 Transputers on each ROC, known as the 'Readout Transputer'. The Teradyne backplane is mapped directly onto the Readout Transputer memory, hence one can download and read data from each of the front-end cards by simply reading from and writing to locations in memory. The second T425 performs data merging and transport to the Event Builder. The three T800 Transputers are dedicated to running the software-based SLT algorithm [11, 21]. The interface between the front-end z-by-timing cards, the LTC, and the Readout Transputer via the address and Transputer buses of the crate backplane is shown in Fig. 15.

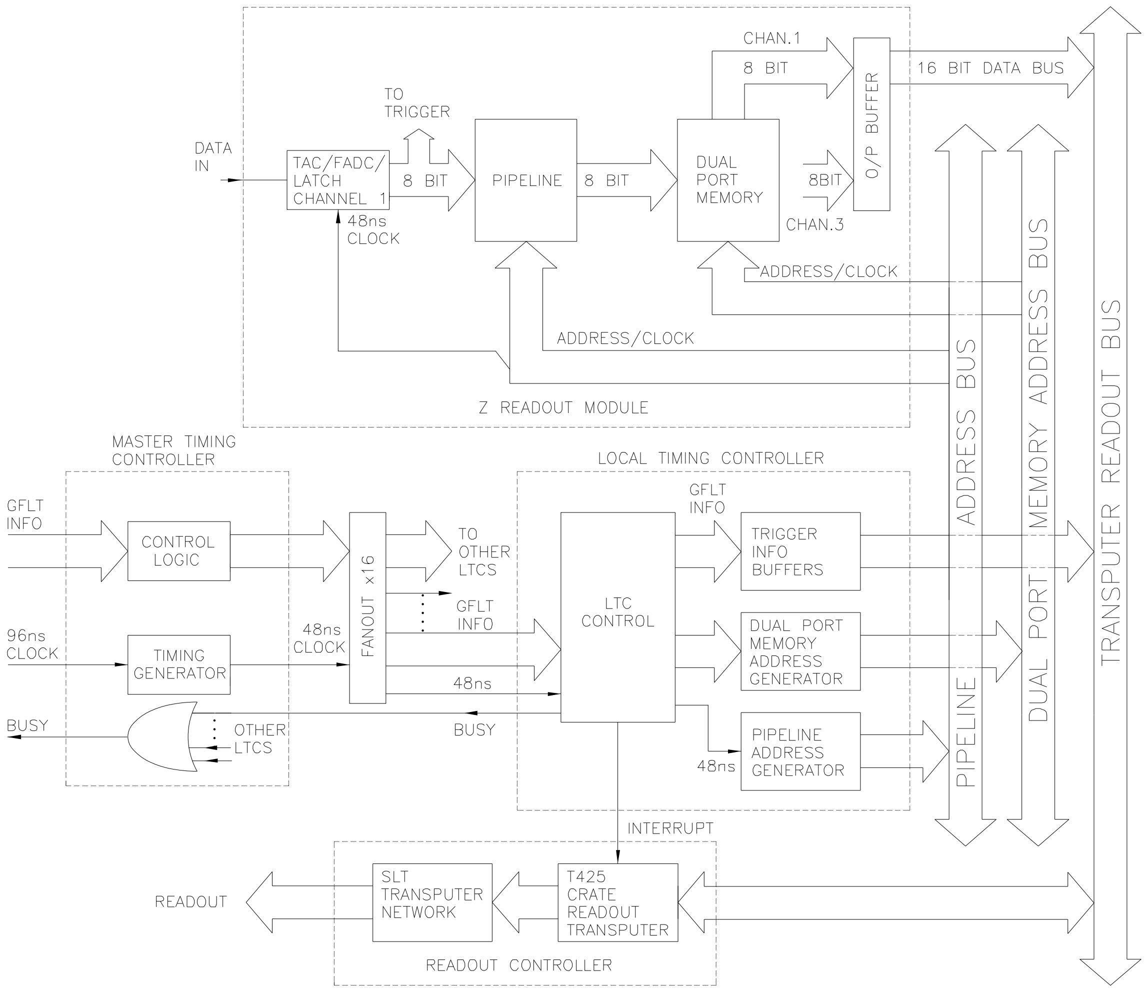

Figure 15: The local z-by-timing DAQ system, demonstrating the functionality of the Timing Controllers.

7.3 The Master Timing Controller

The main purpose of the MTC is to synchronize the CTD readout and FLT processing to the 96 ns HERA beam-crossing clock (and hence to the rest of the experiment), and to provide the interface by which the DAQ system is able to recognize a trigger Accept. The MTC resides in a dedicated crate (the so-called 'RBOX' Crate) which also houses MTC fanout units and the cards responsible for the final stages of the first level track trigger processing and the interface to the GFLT. Although its operation during normal data-taking is relatively simple, its test capabilities make the circuitry rather complex.

A functional diagram of the MTC is shown in Fig. 16. The MTC receives the 96 ns beam-crossing clock from the GFLT, derives a 48 ns clock using a phase-locked loop, and fans out this clock to each of the 16 LTCs. In addition to the clock, the MTC receives a number of signals from the GFLT:

- 'Beam Crossing Zero' (BCN0) is a signal which defines which clock pulse corresponds to the beam crossing numbered zero. Its role is to synchronize the pipelines throughout the system by resetting the LTC pipeline addresses to zero (see section 7.4). Timing accuracy of this signal is important, hence the MTC synchronizes this signal to the 48 ns clock before it is fanned out to the 16 LTCs. Although the pipeline address generation is done on the LTCs, there is also a pipeline counter on the MTC which serves two important functions. Firstly it enables the MTC to detect a 'pipeline jump' error if the BCN0 signal from the GFLT goes out of synchronization (in which case the MTC will flag an error bit). Secondly, when running in stand-alone test mode, it enables the MTC to generate its own BCN0 signal which is then sent to the LTCs.

- 'Accept' and 'Abort' are the signals that control the system. The GFLT Accept signifies an event trigger and hence initiates a pipeline copy. The Abort initiates an abort of the pipeline copy and on receipt of this signal the system becomes free to restart clocking data into the pipelines. The MTC decodes and time-synchronizes these signals with the 48 ns clock before passing them on to the LTCs. Likewise the Test Enable signal passes through the MTC and thence to the LTCs and Calibration Controllers. This signal instigates a calibration cycle.

- The 'Bunch Crossing Number', the 'Trigger (event) Number' and 'Trigger Type' signals are passively transferred from the MTC to the LTC. Each LTC subsequently stores this information in DPMs which are read out by the ROC. In this way the LTC tags the relevant z-by-timing card buffer with the Trigger Number and so synchronizes this block of data with the rest of the experiment. The Trigger Type contains information such as Initialize, Test Trigger, End of Run etc, which is accessed by the MTC and the ROCs.

When the GFLT issues an Accept, the MTC passes on all trigger information to each LTC. At any given time the 'State Machine' logic defines what state the system as a whole is in. Accept, Abort and LTC Busy are the signals which change this state. The MTC OR's 16 Busy signals, one from each LTC, the output of which is sent back via the State Machine logic to the GFLT. This Busy, OR'ed at the GFLT with Busy's from other ZEUS components, inhibits further data-taking until all Busy's are low.

The MTC provides essential stand-alone test functionality, both self-test and test of the LTCs. This test circuitry accounts for much of the MTC complexity. This capability provides the important GFLT functions and gives essential diagnostics in case of failure and during system installation. During data-taking, the MTC has circuits to detect various errors such as clock, address and synchronization faults, signalled by LEDs.

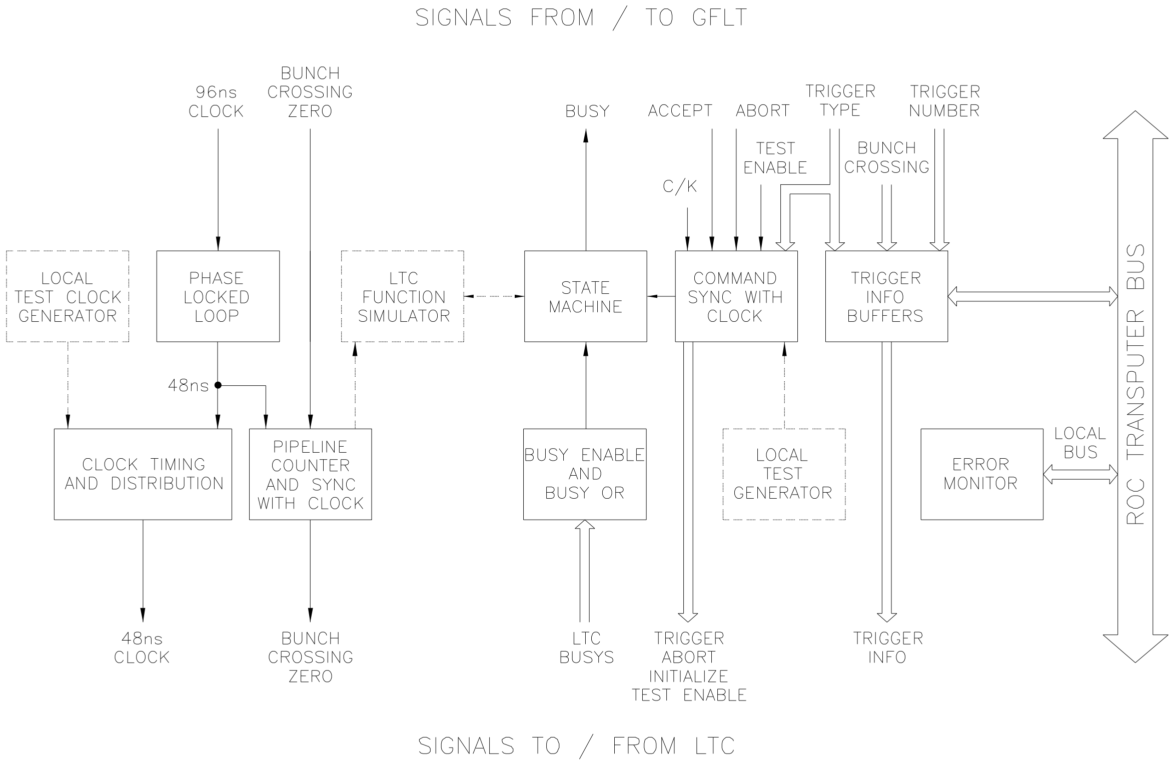

Figure 16: A functional diagram of the MTC. The test functionality, not operational during normal data-taking, is represented by dotted lines.

7.4 The Local Timing Controller

The main purpose of the LTCs is to control the pipelines and buffers of the front-end z-by-timing cards, and to interface with the DAQ system. In addition the LTCs clock z-by-timing data into and through the trigger processors. Each of the 16 z-crates contains one LTC; the LTC shares a 9U card with that crate's Calibration Controller module. The role of the LTC is demonstrated in Fig. 17 which shows a state transition diagram of its readout sequence. A description of how the state transitions are implemented in hardware is given below.

A functional diagram of the LTC is shown in Fig. 18. The LTCs receive the MTC clock pulses and use them to synchronously generate pipeline addresses; this process is carried out in parallel across all crates. The 'Pipeline Address Generator' provides the pipeline addresses to the z-by-timing and trigger cards via the crate backplane. While the data are being pipelined these addresses are derived from the 'Pipeline Counter', which is incremented every 48 ns (twice every beam crossing). The Pipeline Counter is reset to zero by Bunch Crossing Zero every 440 addresses (220 bunch crossings). Prior to a GFLT Accept being received, the data in the pipeline are simply overwritten by subsequent events after $\sim$21 μs. On receipt of the Accept, a window of data is transferred from the pipeline memory on each

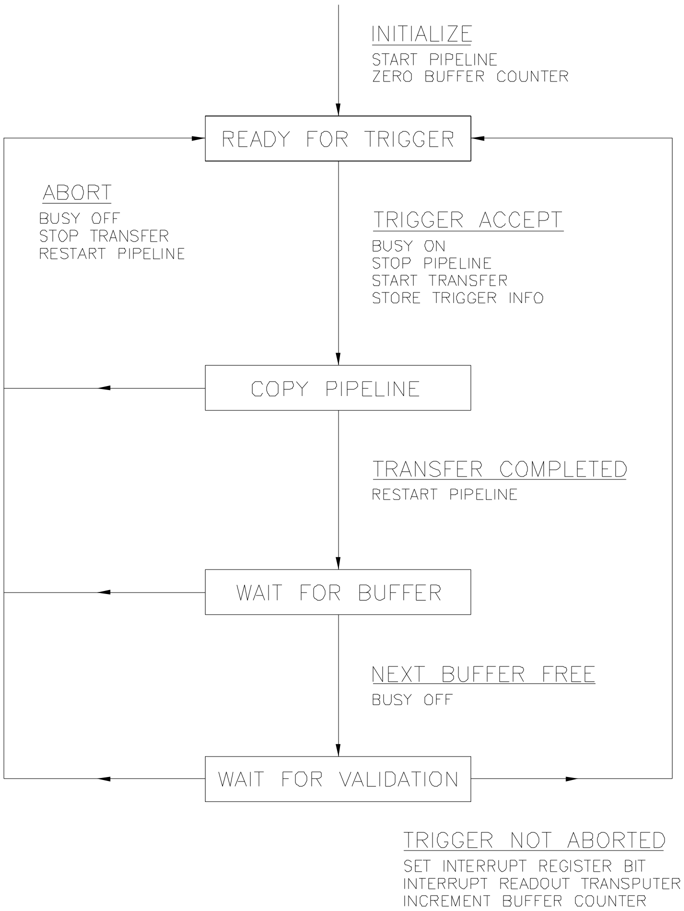

Figure 17: A state transition diagram of the LTC, demonstrating its basic operation during a run.

z-by-timing card into the corresponding DPM; the pipeline window addresses are provided by the Pipeline Address Generator. The DPM addresses are also derived from the LTC. During the transfer, the clocking of new data into the pipeline must be disabled, hence the system incurs deadtime (typically 1 μs per transfer for the z-by-timing system only, but 3 μs including the CTD FLT). Throughout the transfer procedure the Pipeline Counter keeps counting in order to maintain bunch crossing number integrity.

The interface between the Pipeline Address Generator and the software running on the ROC is via the 'Pipeline Registers', which are preloaded from the ROC prior to data-taking. Three registers control the pipeline length (normally set to 440 locations), the length of the window to be read out, and the window 'Jumpback' value (the position of the pipeline window to be read out relative to the arrival time of the Trigger Accept). The window length is approximately 50 locations in normal data-taking mode, i.e. 2.5 μs equivalent. The window has to be relatively wide since the z-by-timing and trigger cards have their data in a spread of pipeline locations

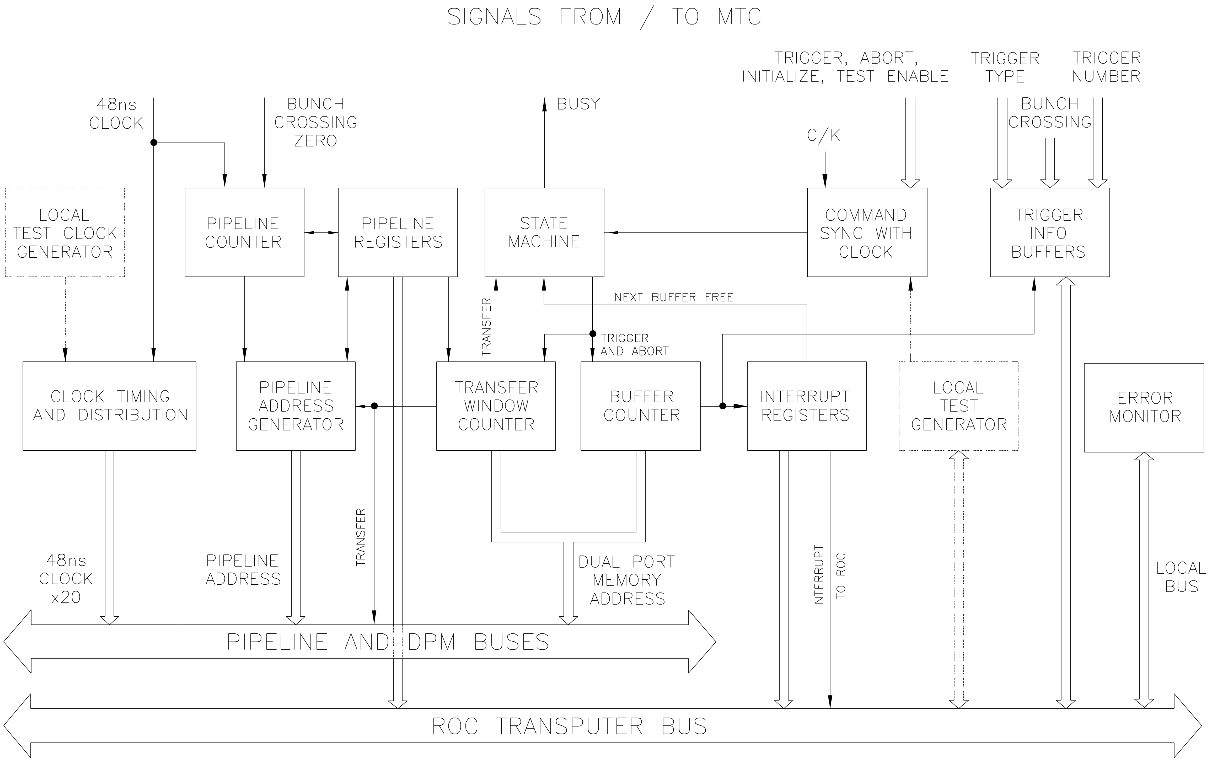

Figure 18: A functional diagram of the LTC. The test functionality, not operational during normal data-taking, is represented by dotted lines.

because of the finite processing time through the CTD trigger. The Jumpback is normally set to 105 locations to account for the timing of the GFLT Accept which occurs 46 bunch crossings after the interaction. This defines the start of pipeline readout, 12 time bins before the interaction crossing. An additional three registers store pipeline address information and include the 'Trigger Address', i.e. the pipeline address at which the last trigger was received.

The 'Transfer Window Counter' generates the seven least significant bits of the DPM Address during the transfer of data. It starts counting from zero after a trigger has been received and signals the end of the transfer when the count reaches the 'Window End Address'. When asserted, the 'Transfer' control signal enables the transfer of data from pipeline memory to DPM buffer (and is high for one additional clock period to allow the front-end cards time to change their memory read/write enables). The Transfer Window Counter and the Pipeline Address Generator change mode simultaneously from writing pipeline data to transferring pipeline data. The DPM is partitioned into ten 'event buffers' in order to reduce system deadtime. This means that a maximum of ten events can be stored for readout by the ROC. The buffer number, simply the four most significant bits of the DPM address, is counted by the 'Buffer Counter'. An example of a timing diagram which also demonstrates the relevant LTC address transitions is shown in Fig. 19.

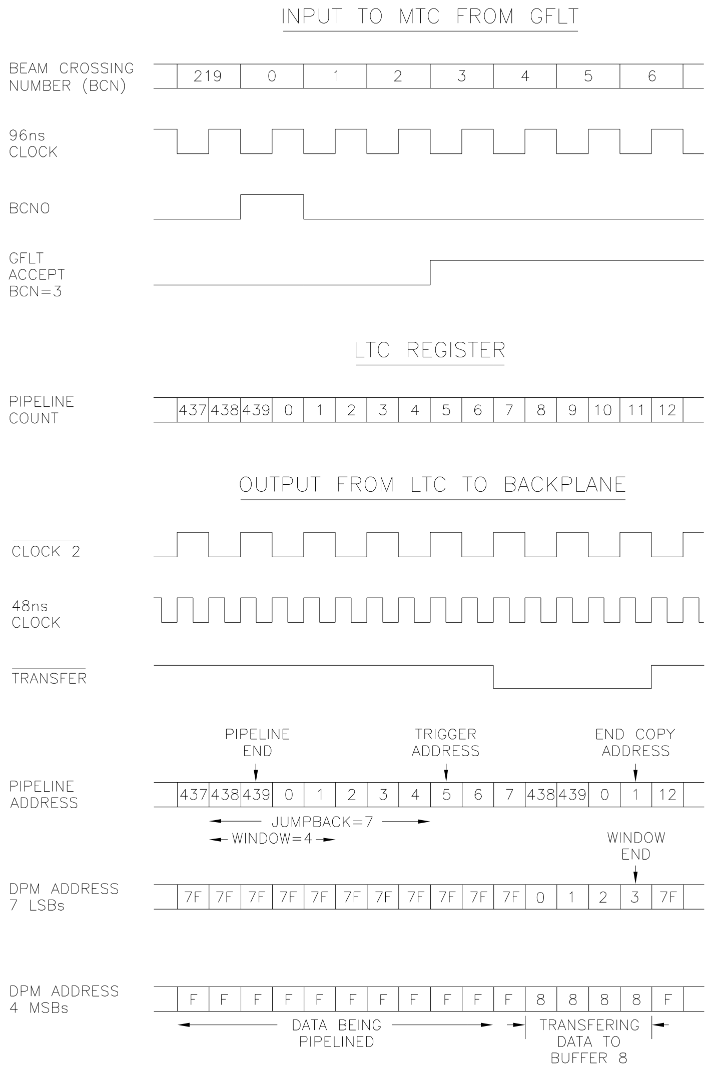

Figure 19: An example of a timing diagram also showing LTC address transitions. Note that for clarity of the diagram, this example uses non-typical values of Jumpback and window length values (7 and 4 respectively). The Buffer Count value has been chosen arbitrarily to 8.

The transfer of data from the DPM to the ROC is an asynchronous process. On command from the LTC, data are transferred from the DPM to the ROC (and hence to the SLT) under control of software running on the ROC Readout Transputer. The window of data transferred in normal data-taking mode is approximately 16 locations, i.e. ~800 ns equivalent. This is chosen to minimize the overall data volume through the system while allowing the complete chamber drift time (~500 ns) to be sampled. Interaction with the DAQ software running on the ROC occurs through an 'Interrupt Register' on the LTC. Each DPM event buffer is associated with a bit in the Interrupt Register, which is set from the Buffer Counter if that event buffer is awaiting readout. When an event is validated (i.e. the Trigger Accept is not aborted by the GFLT), the LTC sets the bit. In usual operation this is synchronous between all LTCs in the system. Any set bit results in an interrupt in the Readout Transputer, thereby notifying the DAQ software that an event is awaiting readout from the buffers. After the appropriate buffers on all z-by-timing cards have been read out, the software clears the Interrupt Register bit for that event, which frees the buffer. This is asynchronous between LTCs. Each buffered event is tagged with its Trigger Number as determined by the GFLT, which later allows data from all 16 LTCs to be re-synchronized.

Each LTC provides a Busy signal to the MTC until its pipeline copy is complete. The MTC presents a z-by-timing system Busy flag to the GFLT until all 16 LTC Busy's are lowered (note however that during this period the MTC continues to provide 48 ns clock pulses for the trigger processors). Additional deadtime will be incurred if any event buffer is full with ten events, with the LTC again maintaining its Busy signal. The LTC 'Next Buffer Free' control signal lowers the Busy via the LTC 'State Machine' logic.

Ideally for a given trigger, pipeline addresses should be the same throughout all 16 LTCs. The window location transferred from pipeline to buffer memory is determined by the arrival time of the GFLT Accept at the LTC Pipeline Address Generators. Similarly the setting to zero of the pipeline address is determined by the arrival time of Bunch Crossing Zero. Throughout the system, the relative t_0's need to be well controlled. The t_0's are affected by the following:

- The 'Jumpback' values loaded in the Pipeline Registers. These are usually the same throughout all crates.
- The timing of the HERA/GFLT 96 ns clock relative to the interaction time. This is affected by the details of the tuning of the HERA machine parameters.
- The delay from the GFLT to the MTC.
- The timing of the 48 ns clock from the MTC through the fanout and LTCs to the pipeline on the z-by-timing cards. This is controlled to about 2 ns across all crates.
- The cable and electronic delays from the CTD to the pipeline on the z-by-timing cards.

The RMS spread in wire-to-wire t_0's from chamber to readout is approximately 3.6 ns. The spread from clocks on the backplane is about 1 ns.