1 The z-by-timing Data Aquisition System

1.1 Overview

Data aquisition of the z data is supervised by a 'Readout Controller' (ROC), one per crate. Its purpose is to read out and zero-suppress the data, perform calculations on the data (in particular to make a Second Level Trigger (SLT) decision) and to pass the data on to the ZEUS Event Builder for subsequent readout. Timing control is provided by a single 'Master Timing Controller' (MTC) interfaced directly to the Global First Level Trigger Box (GFLTB) and 16 'Local Timing Controllers' (LTCs), one per z crate. Between them they synchronize the CTD readout and FLT processing to the 96 ns HERA beam crossing clock (and hence to the rest of the experiment), control the pipelines and buffers of the front end electronics, and also drive the interface by which the data acquisition software is able to recognise a trigger accept. Fig. **??** shows a block diagram of 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 rhobustness; all timing is derived from a single clock and the only control signal a 'BUSY'.

1.2 The Readout Controller

The CTD data aquisition system and Second Level Trigger (SLT) processing system is based on Transputer Readout Controllers (ROCs) [**?**, **?**]. The ROC is a single 9U card, one per crate and contains two T425 Transputers with 1 Mbytes ? of external memory. The ROC operates as sole master of the readout operation; a simple backplane bus is used 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 our application of SLT parallel processing. The readout of the crates is controlled by one of the T425 Transputers on each ROC, known as the 'readout Transputer'. The Teradyne backplane is mapped directly into the readout Transputer memory which means that the front end memories and buffers in a given crate-slot appear as a dedicated area in the transputer memory. Hence one can download and read data from each of the front end cards by reading from and writing to locations in the Transputer memory. Figure **??** shows the interface between the front-end z cards, the LTC and the readout transputer via the address and transputer buses of the crate backplane. The second ROC Transputer is dedicated to running the software-based SLT algorithm [**?**].

1.3 The Master Timing Controller

The Timing Controllers have the following functions :

- Control the pipeline and buffers of the front-end electronics,
- Synchronize data from the 16 z crates,
- Interface with the Global First Level Trigger (GFLT),
- Interface with data acquisition software and run control.

The functionality of the of the Master Timing Controller (MTC) during a normal data taking run is rather simple. The MTC receives the 96 ns beam crossing clock from the GFLTB, derives a 48 ns clock using a phase lock loop, and fans this out to each of the 16 crate LTCs (Fig. **??**). In addition the MTC 'OR's 16 BUSY signals from each LTC and thus inhibits the global first level trigger box until all BUSYs are low. The MTC resides in the 'RBOX' crate which also contains the LTC fanout units. When the GFLTB issues a First Level accept, the MTC passes on trigger information to each LTC (what info??).

A functional diagram of the MTC is shown in fig **??**. Technical details to go here relating to the block diagram.

The MTC provides essential stand-alone test capabilities, of itself and the LTCs. More words here. Couple of sentances on MM box ?

1.4 The Local Timing Controller

The LTCs receive the MTC clock pulses and use them to synchronously generate pipeline addresses which are incremented every 48 ns (twice every beam crossing). The pipeline cycling address is reset to zero by Bunch Crossing Zero from the GFLT every 220 bunch crossings (440 addresses). These addresses are received by the z cards via the crate backplane. Similarly, the LTC clocks the z data into and through the trigger processors. A functional diagram of the LTC is shown in fig.**??**.

On receipt of a trigger (FLT Accept) the LTC stops the pipeline and copies the relevant window of the pipeline to the dual port memory. The length of the window to be read out is preloaded into an LTC register from the ROC and in normal data taking mode is approximately 50 locations, ie 2.5 μs. This window has to be wide because the z and trigger cards have their data in a spread of pipeline locations, due to finite processing time through the trigger system. It is the LTC which keeps track of current address locations of pipeline and DPM via registers which interface with the software running on the ROC. The transfer of event data is carried out in parallel across all z cards in a crate. Delays in distributing the trigger information to each LTC throughout the system is controlled to within a few ns. The clocking of new digitized z data from the chamber into the pipelines must be disabled whilst this transfer takes place, and so the system incurs deadtime. However the pipeline cycling address for writing continues clocking to maintain Bunch Crossing Number integrity. Until its copy is complete each LTC will provide a BUSY signal to the MTC. Since the system is not ready for another trigger, the MTC will present a CTD BUSY flag to the GFLTB until all 16 LTC BUSYs are lowered (note, however, that the MTC continues to provide 48 ns clock pulses to drive the trigger processors). If no GFLT is received, the data in the pipeline are eventually overwritten by subsequent events.

Fig.**??** shows a state transition diagram of the readout sequence (MTC and LTC STDs to be included as fig a) and b) ?). Fig.**??** shows a timing diagram (this diagram to be a 'Pete Shield'-type diagram, pipeline timing with respect to the beam clock and trigger). Words here to describe the diagrams in addition to the paragraph below?

On command from the LTC, data is transferred from the DPM to the ROC (and hence to the SLT) under control of software running on the ROC readout Transputer. Interaction with the Data Acquisition (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 if that event buffer is awaiting readout. Any bit set results in a 'look-at-me' (LAM) in the readout Transputer being raised and this notifies the DAQ software that an event is awaiting readout from the buffers. When an event is validated (ie not Aborted) the LTC sets the bit. This is usually synchronous between all LTCs in the system. The window of data transferred from the DPM to the ROC in normal data taking mode is approximately 16 locations, ie $\sim$800 ns. This is chosen to minimize the overall data volume through the system while allowing the complete chamber drift time ($\sim$500 ns) to be sampled. After the buffers on all z cards have been read out, the software clears that event's interrupt register. This is asynchronous between LTCs. Additional deadtime may be incurred if all event buffers are full with 10 events at any time, with the LTC maintaining its BUSY signal to the MTC. Buffers are required to reduce the deadtime because synchronization is lost as control passes to the data acquisition software.

Synchronization is required if the same time-zero is to be applied throughout the system. Ideally, pipeline addresses and the Trigger Address register should be the same in all crates. The window transferred from pipeline to buffer memory is determined by the arrival time of the Trigger (FLT Accept) at the LTC pipeline address generators, and not the GFLT Bunch Crossing Number (GBCN). The setting to zero of the pipeline address is determined by the arrival time of Bunch Crossing Number Zero (BCN0) at the LTC pipeline address generators. When the pipeline data is written by the ROC synchronization is seen relative to the pipeline address. The window copied is determined by the Trigger time and is seen relative to the pipeline address. When the pipeline data is from in-coming pulses synchronization is seen relative to the arrival time of the pulses. The window copied is determined by the Trigger time and is seen relative to the time of the pulses.

Paragraph on stand-alone testing capabilities (including under software control)?