

# L1CTOC

## **Technical Design Report**

ver 1.03

7 April 2003

**Jamieson Olsen**  
**DZERO Experiment**  
**Fermi National Accelerator Laboratory**

# 1. INTRODUCTION

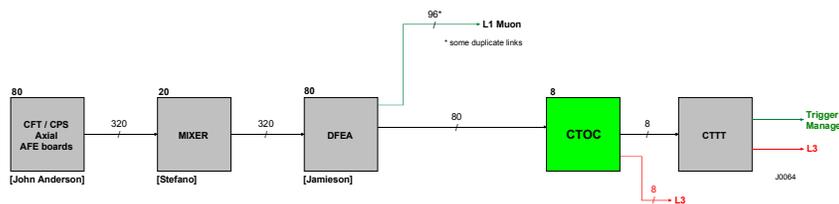
The CTOC is a double-wide daughterboard that is mounted to a Digital Front End Motherboard (DFEM). CTOC stands for *Central Tracker Octant* board or something like that. The CTOC board processes L1 and L2 records from the DFEA boards and provides data to three customers: CTTT (makes L1 trigger decisions), CTQD (quadrant board, processes L2 records), and L3 (collects raw L1 records for diagnostics).

There are three FPGAs on the CTOC board: one for L1 processing and two for L2 processing. These FPGAs operate completely independently. This document focuses on the L1 FPGA, but also includes information about the other two FPGAs on the CTOC.

## 2. NOMENCLATURE

*Event* and *crossing* are used interchangeably in this document. Each event is 132ns.

## 3. L1CTT CHAIN DIAGRAM



Data in the L1CTT chain begins at the AFE, where CFT and CPS axial fibers are sent through discriminators and converted to bits. The discriminator bits are sent to the Mixer, which rearranges them into sectors (each sector being 4.5 degrees in phi). The DFEA daughterboards find CFT tracks and CPS clusters.

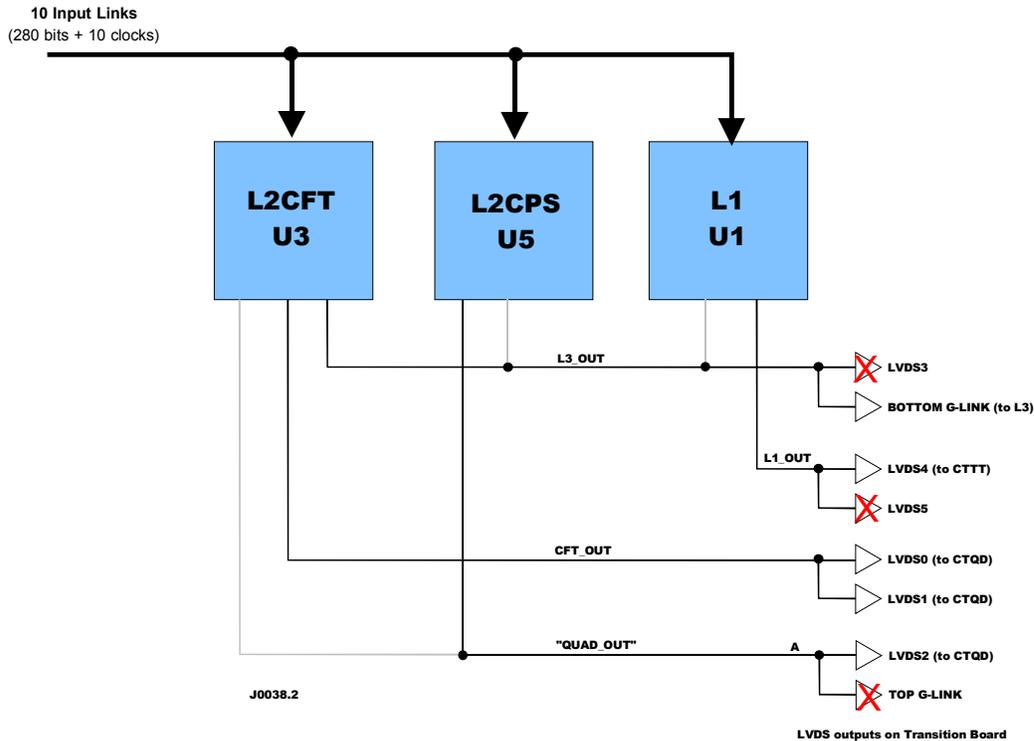
The DFEA board has three customers: L1muon, CTOC, and STOV/STSX (not shown). With each beam crossing the DFEA sends a list of the six highest Pt tracks to L1muon. The CTOC and STOV/STSX boards receive identical copies of the same data. In normal running the DFEAs continuously produce L1 records, which are 7 frames long and contain counts of the found tracks, counts of clusters, number of fibers hit, etc. However, when a L1\_ACCEPT signal arrives at the DFEA it jumps back in its memory and builds up two L2 records: L2CFT and L2CPS. L1 record transmission is momentarily suspended and the two L2 records are sent to the CTOC.

There are three FPGAs on each CTOC board. One FPGA looks only at L1 records, another only looks at L2CFT records, and the last FPGA only looks at L2CPS records from the DFEA. The L1 FPGA is reference number U1 and is the topic of this design report.

With each beam crossing the eight CTOC boards process 80 DFEA L1 records and send eight L1 records to the CTTT. The CTTT processes these eight records and makes trigger decisions and passes those trigger bits onto the D0 trigger manager.

## 4. CTOC BUS STRUCTURE

Ten 28-bit LVDS links feed each CTOC. These ten busses are expected to be aligned to within  $\pm 1$  18.8ns clock tick of each other. Each FPGA on the CTOC sees all ten links. Each FPGA on the CTOC has at least one output bus, which drives an LVDS transmitter or a G-LINK daughterboard (fiber optic transmitter).



The DFE motherboard provides a 53MHz clock and reset; the output busses should be synchronous to this clock. The motherboard also provides a path for FPGA configuration data. After the FPGAs are initialized the motherboard provides a path for setting parameters in the FPGAs and collecting status information.

Some of the output busses go to more than one FPGA – in the diagram above these “extra” data paths are shown in gray lines. These busses should not be used. For example, U3 should not drive the “Quad Out” bus.

## 5. RECORD TYPES AND DEFINITIONS

see the Protocols Document<sup>1</sup> version 7.00

## 6. L1CTOC FUNCTIONS

### 6.1.1. L1 OUTPUT

The L1CTOC must continuously process L1 records sent to it by the DFEA boards. Fields inside these records must be added up, truncated or otherwise processed according to the algorithms described in CORE section of this document. Output records should be formatted in accordance to the DFE Protocols Document<sup>1</sup>.

When the DFEAs interrupt their L1 record transmission to insert L2CFT and L2CPS records the L1CTOC must handle this situation gracefully and alert the CTTT that a L1\_ACCEPT just occurred. The L1CTOC must also pass along Trigger framework control bits (First\_Crossing and CFT\_RESET) which are embedded into DFEAs L1 records.

The L1CTOC device is also responsible for producing a L3 record. Every time a L1\_ACCEPT occurs, the L1CTOC must jump back  $n$  events in its memory (pipeline) and dump out the ten L1 records to the L3 hardware for verification.

Several fake L1 records must be made available so the L1CTOC can send known stable data to the CTTT for checkout and debug. The L1CTOC must be able to switch into any of these fake modes at any time without recompiling the firmware.

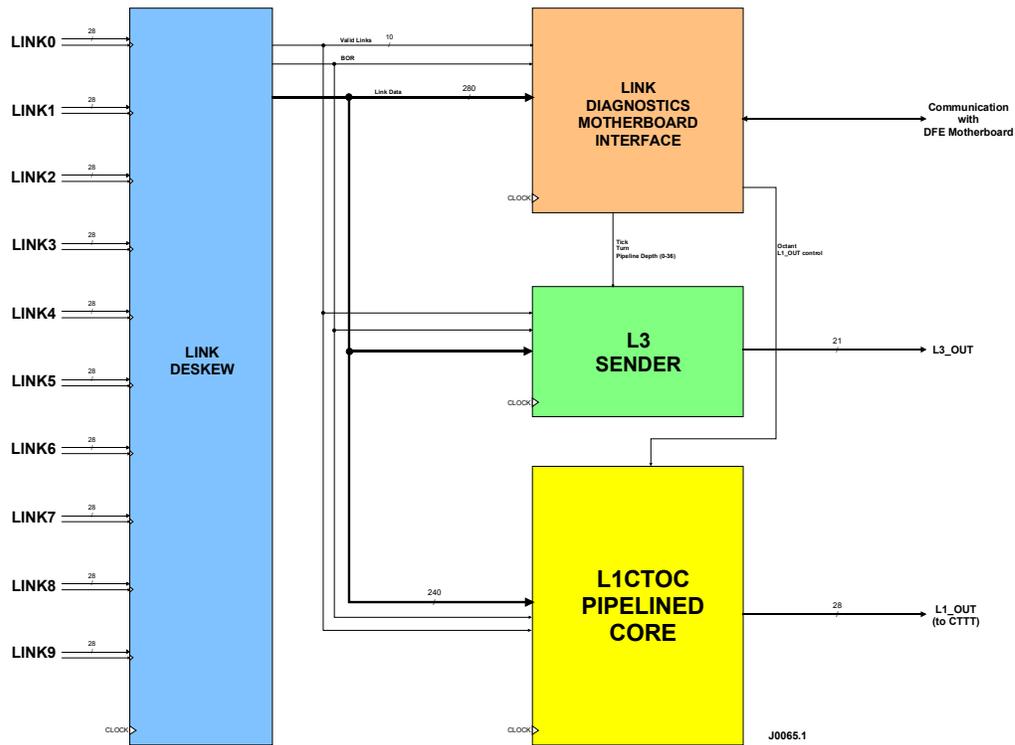
### 6.1.2. L3 OUTPUT AND DIAGNOSTICS

When a L2CFT record comes in the L1CTOC must jump back  $n$  events and extract the ten DFEA L1 records, format them into a L3 record and sent it over the G-LINK transmitter. The length of this L3 record is fixed. The value for pipeline depth  $n$  must be dynamically adjustable at any time without recompiling the design. The pipeline depth must go up to 36 events.

The L1CTOC must collect status information about the health of the input links and the data they are delivering to the L1CTOC. Maintain a history of parity errors, link synchronization errors, missing link error bits, and control bit disagreement errors. Make this status information available to the rest of the world via the slow monitor network. The L1CTOC must also accept *post-initialization* parameters from the DFEC such as octant number, L3 pipeline depth, fake event selection bits, etc. These parameters should be *writable and readable* at any time after initialization.

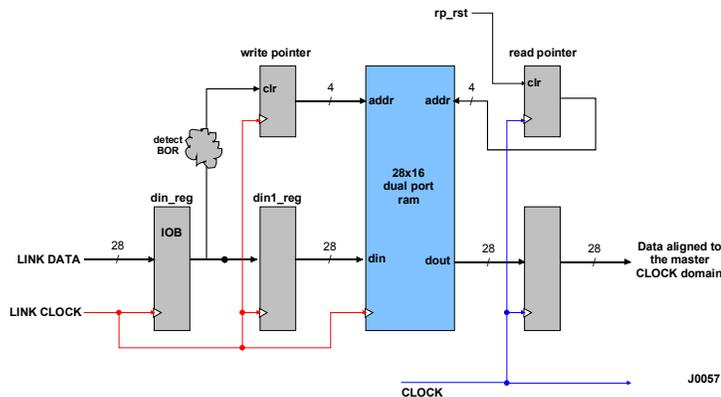
The L1CTOC must be able to detect a special link test pattern sent to it by the DFEA and report the error free reception of this information to the outside world. Likewise, the L1CTOC must be able to drive this special pattern to the CTTT so that it can check for cable errors, stuck bits, etc.

# 7. L1CTOC BLOCK DIAGRAM



## 7.1. LINK DESKEW FRONT END

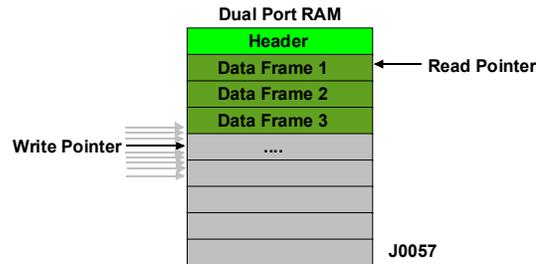
The main purpose of the front end is to cleanly and reliably cross the link clock domains so that the rest of the design operates in the master clock domain. Small dual port rams are used as FIFOs to cross the clock domains. These dual port rams are constructed from distributed RAMs which are smaller than the true dual port BlockRAMs found on the Xilinx FPGAs. As a result, the distributed RAMs can be pushed closer to the input bus pins, thus reducing clock delays and clock skews on non-global clock nets (Links 3-9 use non-global clocks). Below is deskew module for one link:



The registers and write pointer logic in the link clock domain insure that the first frame of a record (L1 or L2) always goes into address 0 of the RAM. The link data is written into the RAM on the rising edge of link\_clock. As the record comes in, subsequent frames are stored in increasing memory locations in the RAM.

After a few frames have been written into the RAM the read pointer is set to zero synchronous to the master clock (called clock from here on out). As the read pointer increments it pulls frames out of the RAM and delivers them to the rest of the design. All link modules have their read pointers reset at the same time.

The read pointer must always trail the write pointer by a few memory locations in order for this to work:



Because the read pointer is trailing a few locations behind the write pointer the design is not sensitive to skew and jitter in the link clock domains. The farther apart the read and write pointers are the more tolerant the design is to link skews, however this also increases latency through the front end. Currently the design separates the pointers by two memory locations, allowing for link skews of about  $\pm 2$  clock ticks. (In reality the CTOCs have measured link skews of less than 5ns.)

With all the ugly clock domain crossing logic hidden in the front end, the rest of the design sees a clean link interface: ten 28-bit busses and a master beginning of record (BOR) marker, all signals are synchronous to the master clock. When BOR is asserted, each of the ten busses should be displaying the first frame of a record (whether it is a L1 or L2 record is doesn't matter). Input link number two was chosen as the master link – this module's BOR output is derived from input link2. *This also means that link2 must be working properly otherwise the link deskew module will break.*

The front end also checks for missing links. A link is considered missing if a BOR marker is not seen within 64 link clock ticks. A ten bit output bus called valid\_link provides this information to the rest of the design. For example, if link 3 is not plugged in, valid\_link(9..0)= 11 1111 0111.

## 7.2. STATUS AND MOTHERBOARD INTERFACE

This functional block handles all communication with the DFEA motherboard. After the CTOC board is initialized the DFE motherboard basically acts as a conduit for low speed status information and parameter setting. For example, after CTOC initialization the user wants to change the L3 pipeline depth. The user runs a DFE\_Ware macro which sends the parameter to the DFE crate controller (DFEC). The DFEC writes the parameter over the DFE backplane to the specified DFE motherboard. The DFE motherboard serializes the parameter and sends it to the specified FPGA. The FPGA converts the parameter back to parallel and make it available to the L3 sender module – now the pipeline depth has been changed.

This module also sends status information back down to the motherboard one “page” at a time. The motherboard passes this information back to the DFEC over the slow monitor bus. This bus runs continuously and is completely independent of the main DFE backplane bus. The DFEC continuously collects this slow monitor information and makes it readable from the 1553 bus. DFE\_Ware regularly scans the DFEC and extracts this status information.

After synchronization the input link busses are checked for things like parity errors, synchronization errors, control bit errors, etc. These status bits can be passed down to the motherboard. A counter that keeps track of the turn and crossing numbers is also instantiated in this module.

The communication between the CTOC FPGAs and the DFEM re-uses the SelectMAP bus, which is normally used to send data to the FPGAs during device initialization. Once the FPGAs initialize (DONE=1) then the SelectMAP pins *must* become general purpose I/O pins. In order for this motherboard interface to work the SelectMAP bus `persist=false` must be set in the Xilinx Bitgen options.

### 7.2.1. INPUT LINK DIAGNOSTICS

**Missing Link Errors.** If an input link module does not see a beginning of record marker within 64 clock ticks it considers that input link to be missing. This logic is actually contained in the Link Deskew module, but the ten bit bus is sent to the motherboard interface module where the missing link history register is maintained. If a link goes missing – even for a brief moment— the corresponding bit in the missing link history register is set, and remains set until the history is manually cleared. [history\\_reg.vhd](#)

**Parity Errors.** L1 records have both a horizontal and vertical parity bits. The vertical parity bits were added to enable parity detection and correction for the L1 data. Currently the L1CTOC design ignores these vertical parity bits and simply checks the first 6 frames of the L1 record for horizontal parity errors. No effort is made to correct the parity error. If a link has a parity error the corresponding bit in the link parity history register is set, and stays set until manually cleared. [parity\\_check.vhd](#), [parcheck\\_module.vhd](#)

**Synchronization Errors.** The Link\_Dskew module attempts to align the link data and move everything into the master clock domain. If everything is working correctly when the Link\_Deskew module asserts the BOR output then all output busses should be displaying a L1 (or L2) header. The synchronization error logic checks for conditions when this isn't the case. Only valid (non-missing) input links will be considered by the synchronization logic. [sync\\_detect.vhd](#)

**Test Pattern Detection.** The DFEA can be placed in a mode where it repeatedly sends a L1 test pattern. This is basically a L1 record with a walking 1's pattern inside of it. Logic in the L1CTOC checks that this pattern is received error free. If not, it sets a bit in the pattern history register. This is only intended to be used for DFEA to CTOC link tests. [pattern\\_checker.vhd](#), [patt8.vhd](#), [patt\\_detect.vhd](#)

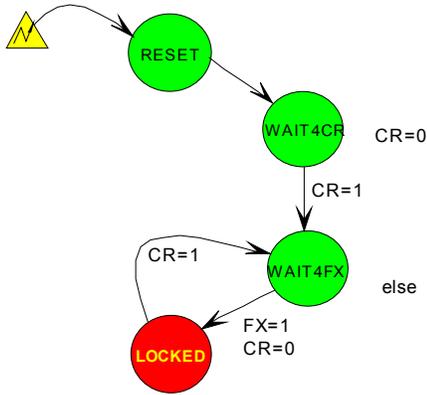
## 7.2.2. CONTROL BIT DIAGNOSTICS

**FX XOR.** Each DFEA inserts a First Crossing (FX) control bit into the first live crossing of a turn. If the DFEAs are aligned in time then the L1CTOC device should see the FX bits arrive on each link at the same time. Thus under normal conditions the XOR of all ten FX bits should be zero. If the XOR value goes high that means that there was a disagreement amongst the input links. This bit is a history bit, controlled by the clear history command.

**L2.** This bit is set when a L2 header comes in on Link2. Once set, it remains set until the history is cleared.

## 7.2.3. TICK AND TURN COUNTER

The L1CTOC's L3 sender needs to know the tick (crossing) and turn numbers so that it can insert these values into the L3 header. Based on the FX and CFT\_RESET (CR) control bits received on link2 the counters are locked onto the incoming data stream. The tick and turn counter logic is based on the simplified state diagram:



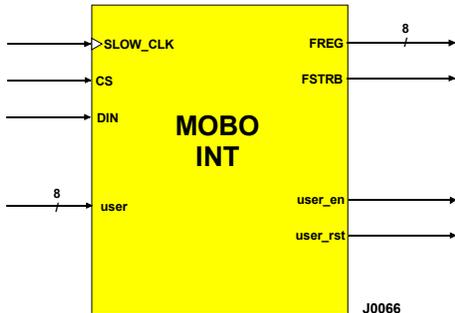
CR happens infrequently (few times per hour) and is often asserted for a second or so. This active high signal is a global system reset (also called SCL\_INIT). On the first FX after the CR goes low the counters are reset to turn=1, tick=7. The tick counter counts up to 158 and is then reset to zero, and the turn counter is incremented. If all is proceeding normally the tick and turn counter module will drive the LOCKED output high.

When the tick counter equals 6 it *expects* to see the next FX bit in the next L1 record. Since the FX bits are not present in the L2CFT or L2CPS records it's possible that one of these records has displaced the L1 record containing the FX bit. So the tick and turn counter must be able to "ride out" a missing FX bit without dropping the LOCKED bit. If FX comes at an unexpected time, however, the tick and turn module will pulse the BAD\_FX output.

Both LOCKED and BAD\_FX status bits are available to the outside world through the slow monitor interface. BAD\_FX is run through a history register – once it goes high it stays high until the history is cleared. The LOCKED status bit is passed straight through. [tick\\_turn.vhd](#)

## 7.2.4. DFE MOTHERBOARD INTERFACE

The module *mobo\_int.vhd* forms the core of the DFE Motherboard Interface. It handles the low-level serialization and de-serialization of data between the L1CTOC FPGA and the DFE motherboard.



When the DFE motherboard writes to the L1CTOC device the data byte is shifted in on the DIN input MSb first. The CS input is an active low control line that is low for the entire time it takes to shift in the eight bits. Shortly after CS goes high the data byte is driven out on the FREG bus and the FSTRB line is pulsed high for one slow\_clk period. The DIN and CS inputs are sampled by slow\_clk.

In the other direction the eight bits applied to the USER input are continuously serialized and sent to the motherboard over the USER\_EN and USER\_RST lines. The DFE motherboard collects this user byte from the *current device* and sends it over the slow monitor bus as the slow monitor user bits[7..0]. The three devices on the CTOC daughterboard have separate data paths to send their status bytes down to the motherboard, however the motherboard can only read the status information from one device at a time.

### 7.2.4.1. SETTING PARAMETERS AFTER INITIALIZATION

After the L1CTOC device is initialized some additional arguments must be passed to it. These are:

- **L3 Pipeline Depth.** This is adjustable from 0-36 crossings deep. The default is 34.
- **Home Octant.** The firmware needs to know what its home octant is. Range is 0-7, default is 0.
- **Fake L1 record select bits.** For diagnostic purposes, the L1 CTOC can send fake L1 records to the CTTT. Default is zero.
- **Status Page.** The L1 CTOC has a lot of status information, but it can only be viewed in one byte-wide “page” at a time. Default is page 0.

As previously mentioned, *mobo\_int* supports writing and reading an 8-bit value to/from each device on the CTOC. This is does not provide enough bits to do everything the L1CTOC needs it to do, so a simple protocol is imposed on this interface. The feature register now has four independent registers, selected by bits 7 and 6:

7	6	5	4	3	2	1	0
0	0		CH	Page Select			
0	1	L3 Pipe Depth (0-36)					
1	0				Octant		
1	1			Fake L1			

2003-01-06

So to change the L3 Pipeline depth to 25 do the following:

```
set DFE 3
set device 2
# 25 = 0x19 Now OR 0x19 with 0x40 to set bit 6.
cmd writebyte 3 0x59
```

And to change the Octant to 5 do the following:

```
set DFE 3
set device 2
# 5 = 0x05 Now OR 0x05 with 0x80 to set bit 7.
cmd writebyte 3 0x95
```

### 7.2.4.2. STATUS PAGES

The L1CTOC supports 16 pages of status information. Each page is a single byte of status information. The current status page becomes the slow monitor user bits for the CTOC board. Previous versions of the status information used logic to stretch error pulses so that spurious errors had a better chance of being sampled by the slow monitor system. However, this method had shortcomings and has been replaced with a status “history” scheme.

Status history works like this: after reset all of the L1CTOC status pages are set to 0x00. On every 53MHz clock edge the contents of the status pages are updated. If a bit goes high, it stays high until the status page history is cleared. Setting Bit 4 of the Page Register clears the status history for the status pages.



- *After power up or board reset the history bits may contain garbage. So it is good practice to clear the history after these conditions.*
- *The history is cleared on the 0 to 1 transition of the CH bit. When clearing the history it's best to explicitly toggle it by writing 0x00, 0x10, and 0x00.*

Below are the 16 status pages specific to the L1CTOC design.

page	S7	S6	S5	S4	S3	S2	S1	S0
0		Bad Master FX	FX XOR	L2	Parity Err	Missing Link	Sync Err	Tick/Turn Locked
1						Home Octant Number [2..0]		
2						Fake L1 Mode		
3						L3 Pipeline Depth [S..0]		
4								
5	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0
7								
8				Miss-link4	Miss-link3	Miss-link2	Miss-link1	Miss-link0
9				Miss-link9	Miss-link8	Miss-link7	Miss-link6	Miss-link5
10				Sync-err4	Sync-err3	Sync-err2	Sync-err1	Sync-err0
11				Serr-err9	Sync-err8	Sync-err7	Sync-err6	Sync-err5
12				Parity-err4	Parity-err3	Parity-err2	Parity-err1	Parity-err0
13				Parity-err9	Parity-err8	Parity-err7	Parity-err6	Parity-err5
14				Patt-err4	Patt-err3	Patt-err2	Patt-err1	Patt-err0
15				Patt-err9	Patt-err8	Patt-err7	Patt-err6	Patt-err5

2003-01-06

**NOTE:** the orange cells are cleared by the clear history feature.

**Parity Err.** logical OR of the ten Parity Error bits in pages 12 and 13.

**Missing Link:** logical OR of the ten Missing Link bits in pages 8 and 9.

**Sync Err.** logical OR of the ten SYNC\_ERR bits in pages 10 and 11.

**L2.** Set when a L2 record comes into the L1CTOC on link2.

**Tick/Turn Locked.** The L1CTOC extracts two trigger framework bits (First Crossing, CFT\_RESET) from the DFEA L1 records delivered on link 2. From these control bits the L1CTOC determines the beam timing and maintains a tick and turn counter. If this LOCKED bit is set it means that the L1CTOC has observed a CFT\_RESET followed by an FX and it is incrementing the tick and turn counters normally. Tick and Turn counts are only used for L3 readout – they do not affect the L1 output of the CTOC in any way.

**Bad Master FX:** Once the tick and turn counters are locked they expect the FX bit to come in at a specific time. If FX comes in at another time this bit is set, and it stays set until the history is cleared.

**FX XOR:** Every time a L1 record comes into the L1CTOC all of the FX bits are compared against the master FX bit extracted from link2. This bit is the logical XOR of the individual FX bits— if the FX bits disagree, this bit is set and stays set until the history is cleared.

**Status Pages 1-3** always returns the value of their associated registers.

**Status Pages 5-6.** 0x55 and 0xAA, respectively. Used for debugging the interface.

**Status Pages 8 and 9.** The Missing Link error bit for each input link is listed here. If a link has not seen a Beginning of Record (BOR) transition in at the last 63 clock cycles, the corresponding bit is set and stays set until the history is cleared.

**Status Pages 10, 11.** After the links have been synchronized, a circuit checks that each link's BOR signal is aligned with the master BOR signal. If there is a synchronization problem with a link, the corresponding bit is set and stays set until the history is cleared. Only considers non-missing input links.

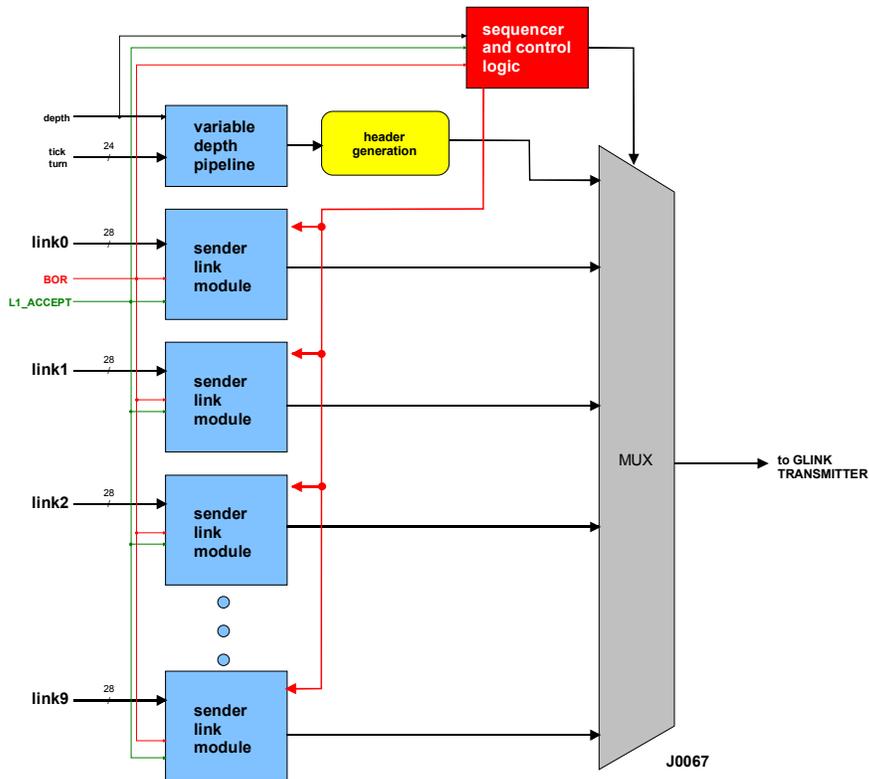
**Status Pages 12 and 13.** As L1 records come into the L1CTOC, the horizontal parity bits are checked for frames 1-6. If there is a parity error in the first 6 frames of the L1 record then the corresponding bit will be set and stays set until the history is cleared. Only considers non-missing input links.

**Status Pages 14 and 15.** The L1CTOC contains test record detection circuitry. The DFEA can be placed into a special mode where it sends a L1 test record to the L1CTOC. *Used only for DFEA to L1CTOC LVDS link tests.*

## 7.3. L3 SENDER

The purpose of the L3 sender is to wait for a L1\_Accept to occur, then jump back  $n$  crossings and concatenate all ten L1 records, wrap the raw link data in a L3 record and send it to the L3 processor via a G-LINK fiber optic link. The value of  $n$  is known as the L3 pipeline depth and it can be changed dynamically via the motherboard interface module after initialization. The L3 pipeline depth can range from 0 to 36 crossings deep, with the default being 34.

### 7.3.1. L3 SENDER BLOCK DIAGRAM



### 7.3.2. SENDER LINK MODULE

Most of the complexity in the L3 sender is pushed into ten sender link modules. These modules are comprised of two BlockRAMs, some counters, and glue logic. The two block rams are configured as 32x256 for writing, and 16x512 for reading. Normally these modules accept 28-bit link data and write it into the RAM and increment the write pointer on each clock tick. When a L2 header comes in (L1\_ACCEPT occurred) the first seven frames are written into the RAM and then writing is suspended. This is “readout mode”. The read pointer is then calculated as:

$$\text{read pointer} = \text{write pointer} - (7 * (\text{depth} + 1))$$

The read pointer now points to the header of the target L1 event. Now the sequence and control logic block is responsible for extracting the link data from each of the sender link modules 16 bits at a time. To the sequencer and control logic block each sender link module looks like a 16x16 RAM. For example, if the control logic drives 0000 on the 4-bit address bus all of the sender link modules will display the L1 header frame bits [27..16]. 0001 = L1 header frame bits [15..0], etc.

A separate pipeline stores the tick and turn numbers for later retrieval. The depth of this pipeline can be dynamically adjusted from 0 to 48 crossings – it is adjusted so that when the sender link modules transition to “readout mode” the turn and crossing number of the target event is present on the outputs of this pipeline.

Most of the fields in the L3 header are constant, except for the tick and turn numbers. The sequencer and control logic controls a large mux that selects which 16-bit word to pass to the G-LINK transmitter. The G-LINK data bus bits[19..16] are used to mark the beginning (0x5) and end (0xA) of the L3 record. The total length of the L3 record is 6 headers + 140 data words + 2 trailers + 4 padding = 152 words, arranged as follows:

```

0101  L3 header1 (0x0310)
0000  L3 header2 (0x418c)
0000  L3 header3 (tick(7..0)+0x01)
0000  L3 header4 (turn(15..0))
0000  L3 header5 (firmware version, 0x0002)
0000  L3 header6 (L3 status bits)
0000  0000 + link0 L1 header[27..16]
0000  link0 L1 header[15..0]
0000  0000 + link1 data frame[27..16]
:      : : :
0000  0000 + link9 trailer frame[27..16]
0000  link9 trailer frame[15..0]
0000  L3 trailer1
0000  L3 parity word (0x0000)
0000  padding word (0xdead)
0000  padding word (0xbeef)
0000  padding word (0xbabe)
1010  padding word (0x0000)

```

After the complete L3 record has been transmitted, the sequencer and control logic block asserts the RESUME signal, and all of the sender link modules return to “write mode” and continue to store the L1 data in their RAMS. The period L1\_ACCEPT period is equal to the depth of the L3 pipeline – nominally 34 132ns crossings. Also note that in “396 mode” there is a live crossings one out of every three crossings – the other two crossings don’t have any data in them. The L1CTOC design is 132ns mode compliant – it treats each crossing as through it contains valid data.

### 7.3.3. L3 SENDER TEST MODE

If the L3 depth is set to zero the L3 sender will produce a L3 record but instead of containing ten L1 records it will contain the first 7 frames of the L2CFT record received on each link. This is only intended to be used for diagnostic purposes only.

For more information on the L3 record fields, refer to the [Protocols Document<sup>1</sup>](#).

***Currently the L1CTOC design drives the shared L3\_OUT[27..0] bus all the time. U3 and U5 must tri-state off the L3\_OUT bus or else bus contention and garbled data will occur.***

### 7.3.4. L3 RECORD DIAGNOSTIC BITS

The L1 CTOC inserts status bits into the L3 record. These 16 bits go into L3 header frame 5, which is normally reserved for the “Firmware Version Major/Minor” bytes.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
										FXD	FXX	PE	ML	SE	TFW	Header 5
ML						L9	L8	L7	L6	L5	L4	L3	L2	L1	L0	Header 6

2003-04-07

#### 7.3.4.1. L3 HEADER FRAME 5

*This frame does NOT agree with the protocols!!!*

- FXD** This bit is set if the master FX bit (which comes from input link2) arrives at the wrong time.
- FXX** This bit is the exclusive OR of all the FX bits that come into the CTOC. If there is a disagreement this bit is set.
- PE** Parity Error. Set if any input link detects a horizontal parity error in a L1 record.
- ML** Missing link error. A link is considered missing if the link\_deskew front end fails to detect a beginning of record (BOR) marker within 64 clock cycles.
- SE** Sync Error. Checks that that all input links are aligned after leaving the link\_deskew front end. If this bit is set it means that the input links were so badly skewed that the front end could not correct it.
- TFW** Set if the tick and turn counter has locked onto the trigger frame work control bits.

NOTE: These bits are special HISTORY bits. They are automatically cleared immediately after the L3 record is sent. Once one of these bits is set, it stays set until the next L1\_ACCEPT. These bits are similar to, but separate from the slow monitor status bits, which are history bits but the “Clear History” control is manual.

#### 7.3.4.2. L3 HEADER FRAME 6

*This frame agrees with the protocols.*

**ML and L9..L0.** Missing link. If an input link is missing the appropriate L9..L0 bit is set. ML is the logical OR of bits L9 through L0.

This Missing link information does not have any “history” characteristics – it is merely a snapshot of the missing link information at the time the L3 record is created.

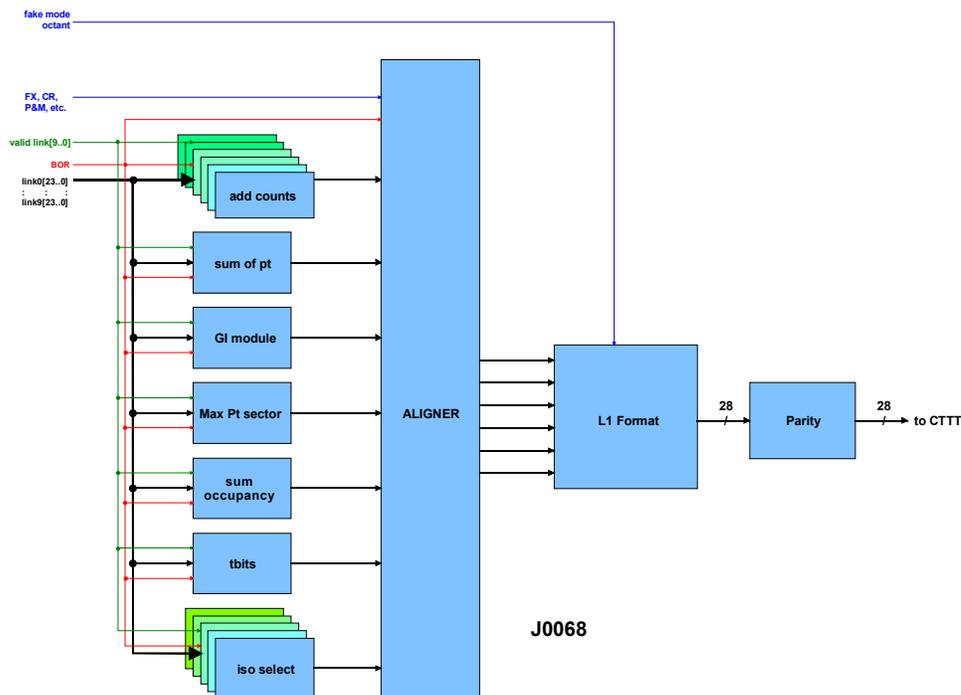
## 7.4. L1CTOC CORE

This is where all of the work gets done in the L1CTOC. Basically the L1CTOC is just a large adder and comparator. Previous incarnations of the L1CTOC did parity correction on the L1 records before they were applied to the core equations. Parity correction requires that the design *wait* until the entire L1 record was received, then the records are flattened into a wide bus and the parity bits are corrected. After the parity correction the wide bus is applied to the glob of core equations in parallel and *some time later* the valid results are latched and clocked out in the proper order. This type of asynchronous design can be made to work, but it's prone to race conditions and strange characteristics that may only show up in timing simulation (or not in any simulation!). Also, the wide parallel busses chew up routing resources in the FPGA.

This version of the L1CTOC doesn't do parity correction, and thus it doesn't need to wait for the entire L1 record to come in before starting to process data fields in the L1 records – a latency gain of a least one crossing (132ns). The core design is 100% synchronous and fully pipelined: L1 records are clocked into the core one frame at a time.

If an input link is missing, the core excludes that link from all calculations, thus preventing a bad link from corrupting the output data.

### 7.4.1. L1CTOC CORE BLOCK DIAGRAM



### 7.4.2. L1 MODULES

The modules listed in this section perform simple calculations on the fields in the ten L1 records. Their outputs correspond to fields in the L1CTOC output record. If an input link is missing (not valid) then its fields are not considered in the calculations.

### 7.4.2.1. TBITS

This module has two outputs T1 and T2. T1 is set if any sector has an isolated track with a tight preshower cluster match and its PSC[1..0]=1. T2 is set if any sector has an isolated track with a tight preshower cluster match and its PSC[1..0]=2

### 7.4.2.2. GI MODULE

GI is set if a sector has an isolated track and that sector's occupancy level is less than 16 doublets hit.

### 7.4.2.3. ADD COUNTS

There are six instantiations of the Add\_Counts module in the core. Each module is a pipelined adder that accepts ten 3-bit values and sums them. The result is truncated to three bits ("111" means seven or more). Since this adder is pipelined, the same circuit is used to concurrently add up the Maximum, High, Medium and Low track counts. The Maximum counts are clocked in on timeslice 1, High on timeslice 2, etc. The four final sums are clocked out one at a time, starting with the Max sum.

For example, the instantiation called **plc\_adder** adds up all of the XPT+LC, HPT+LC, MPT+LC and LPT+LC fields from all ten input records. On the first timeslice ten XPT+LC sums are clocked in. On the next timeslice ten HPT+LC sums are clocked in, etc. It sequentially produces four 3-bit sums which become "XPT+LC[2..0]", "HPT+LC[2..0]", "MPT+LC[2..0]", and "LPT+LC[2..0]" in the output record.

### 7.4.2.4. SUM PSC

Sum\_PSC adds up all ten of the PSC[1..0] fields and truncate the final sum to three bits. "111" means seven or more clusters. The output field is "#PSC".

### 7.4.2.5. MAX PT SECTOR

This module determines the sector with the highest total Pt [SUM\_ABS] field. If there is a tie always choose the lower link number. The number reported here is tied to the input link number: "0000" means that input link0 had the highest total Pt tracks. The output field is "RA HTPTS[3..0]".

### 7.4.2.6. SUM OCCUPANCY

This four bit field (OL\_OCT) is equal to the total doublet occupancy of the octant divided by 64. If the total occupancy is greater than 42.6% then all four bits will be set. For example, if OL\_OCT[3..0]=0001 then the occupancy of the octant is  $\geq 2.6\%$  and  $< 5.3\%$ .

### 7.4.2.7. SUM OF PT

Add up the ten sum of Pt fields (SUM\_ABS[PT]) and truncate to 6 bits. If the final sum is greater than 63 report it as 63. The output field is "SUM\_ABS[PT][5..0]".

### 7.4.2.8. ISO SELECT

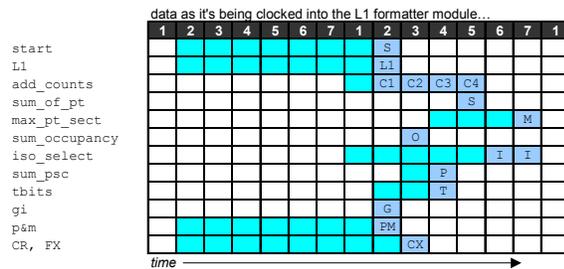
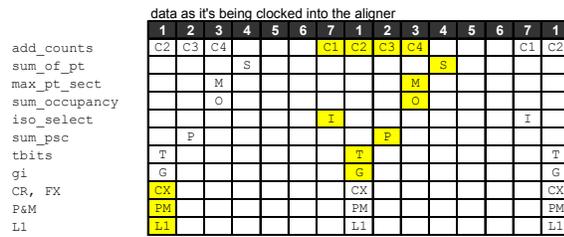
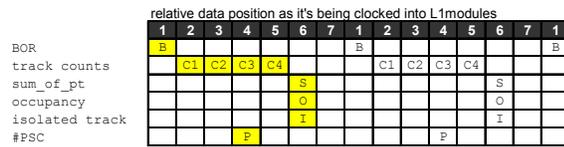
Each module looks at the isolated track bytes for two adjacent sectors. By definition only one of these sectors can have a valid isolated track. If this occurs the module simply passes that valid byte through. The fields in the output record are as follows:

Frame 5[23..16]	= isolated track byte from link0 or link1
Frame 5[15..8]	= isolated track byte from link2 or link3
Frame 5[7..0]	= isolated track byte from link4 or link5
Frame 6[15..8]	= isolated track byte from link7 or link6
Frame 6[7..0]	= isolated track byte from link8 or link9

### 7.4.3. DATA ALIGNMENT

The order in which L1 fields are clocked into the L1 modules are not necessarily the in the order that it needs to be in for the output fields. Additionally, the latency of each L1 module is different, so after the modules the pieces of data will be skewed across multiple timeslices. The aligner module consists of shift registers used to delay various pieces of data so that they're in the proper timeslices as they get clocked into the L1 formatter module. The aligner's shift registers are constructed from SRL16E primitives. Each SRL16E takes up one CLB and is an adjustable 1 to 16 deep shift register.

The timing diagrams below follow one L1 event though the L1 modules and aligner, all the way to the inputs of the L1 formatter module. The light blue squares represent how much the aligner had to delay the data to make everything line up properly for the formatter.



## 7.4.4. L1 FORMATTER

The L1 formatter accepts the pieces of data after it has been aligned into the proper timeslices. If the data is valid L1 data (if the L1 bit is set) the formatter packs the fields into the correct position within the 28-bit frame and clocks it out. Since the data is already aligned into the order that it will appear in the output record there is no need for the L1 formatter to buffer up any data frames – what comes in goes out one clock cycle later.

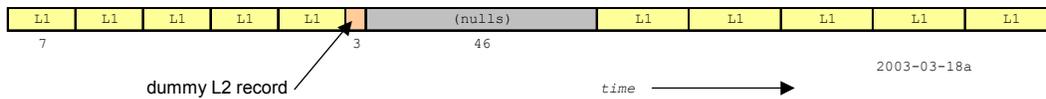
The L1 formatter is also responsible for dealing with L2 data. When a L2CFT or L2CPS record makes its way through the core the output of the L1 modules will be complete garbage. The L1 formatter must recognize this situation and not attempt to repackage it into a L1 record. Rather, it will send a dummy L2 header to the CTTT. This alerts the CTTT that a L1\_ACCEPT happened upstream.

The L1 formatter can also replace the normal L1 data with one of several fake L1 records used for chain diagnostics. These fake records can be selected dynamically after the L1CTOC device has been initialized.

### 7.4.4.1. L1 AND L2 SWITCHOVER

Normally, when no L2 data is coming into the CTOC the L1 formatter will see the L1 input set whenever the START input is asserted. In this case the formatter will produce the L1 record and clock it out one frame at a time.

However, if the START bit is asserted and the L1 input is not asserted, the formatter interprets this as the start of a L2CFT header coming in (note that L2CPS records do not generate a START bit). When this happens the L1 formatter transmits a dummy L2 header (2 frames) followed by 47 null frames. The parity module replaces one of the null frames with a parity frame, and thus the output of the L1CTOC looks like this:



After going through the parity module, the dummy L2 record looks like this:

```
0xE000028 (empty L2 header)
0x1123456 (turn=0x1234, tick=0x56)
0x012347E (parity word)
```

### 7.4.4.2. DIAGNOSTIC (FAKE) OUTPUT RECORDS

The L1CTOC can replace the normal L1 output record with a diagnostic or fake L1 record. When one of the fake records is selected, it is sent repeatedly with every crossing. If a L1\_ACCEPT occurs it will displace the fake L1 records as described in the previous section.

**NOTE: the normal L1 records include the FX and CR control bits, however the fake L1 records do not. If the L1CTOC is sending fake L1 records the CTTT's tick and turn counter (also based on FX and CR) may be invalid.**

The L1CTOC supports up to 15 fake L1 records:

mode	description
0	normal data (default)
1	empty L1 record
2	one max track, no matching clusters
3	one high track, no matching clusters
4	one medium track, no matching clusters
5	one low track, no matching clusters
6	2 max tracks, no matching clusters
7	3 max, 1 low, no matching clusters
8	isolated high track, no matching clusters
9	2 max, 1 matched to a cluster. #PSC=1
10	reserved
11	reserved
12	reserved
13	reserved
14	shutdown output link completely
15	link test pattern

*fake\_maker.xls*

The link test pattern is used for CTOC to CTTT link checks. The output record is a standard L1 header followed by a walking 1's pattern:

```
0xF000007  
0x0111111  
0x0222222  
0x0444444  
0x0888888  
0x0000000  
0x0000000  
0x0FFFFFF8
```

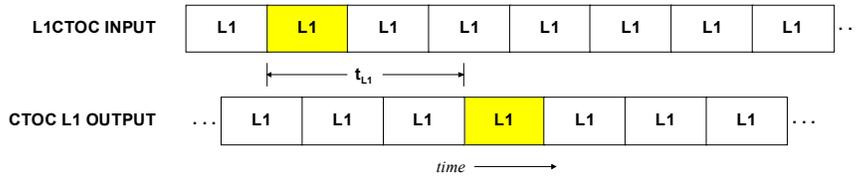
The CTTT front end has logic that checks for the error free reception of this record type.

### 7.4.5. PARITY GENERATOR

L1 records and dummy L2 records are clocked out of the L1 formatter one frame at a time. At this point the horizontal or vertical parity bits are zeroed out. As they pass through the parity generator the horizontal parity bits are added to the frames and clocked out on their way to the CTTT. The vertical parity word is added at the end of the record. The parity generator knows that L1 records are 7 frames long and that L2 records may be variable length – in the case of the L2 records the parity generator determines the length of the record by looking at the “number of objects” field in the L2 header. Latency though the parity generator is 3 clock ticks.

## 8. TIMING AND LATENCY

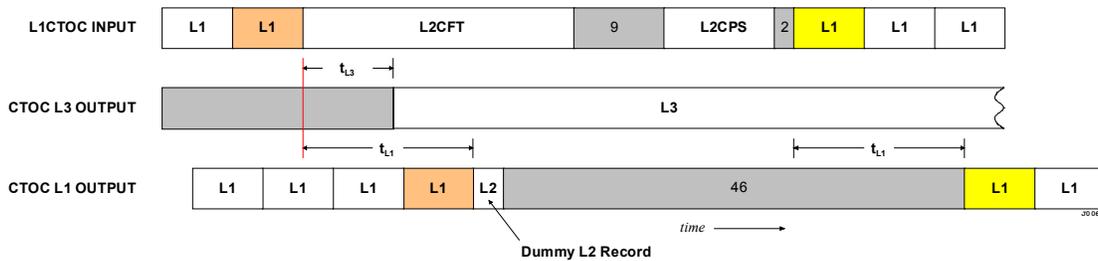
### 8.1. L1 LATENCY



Normal L1 record latency through the L1CTOC is  $t_{L1} = 320\text{ns}$  or 17 clock cycles.

### 8.2. L2/L3 LATENCY

When the DFEA sees a L1\_ACCEPT it sends a L2CFT and L2CPS record to the CTOC as shown below:



When the CTOC observes a L2CFT record arrive there is a delay of  $t_{L3} = 169\text{ns}$  (9 clock cycles) before the start of the L3 record. The L1CTOC also sends a dummy L2 record to the CTTT in response to a L2CFT record coming in. This delay is the same as  $t_{L1}$ .

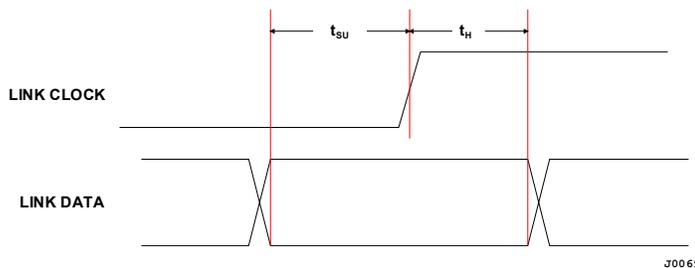
### 8.3. TIMING AND CONSTRAINTS

The master clock as well as all link clocks are 53.104MHz. The current L1CTOC FPGA meets and exceeds these clock frequencies.

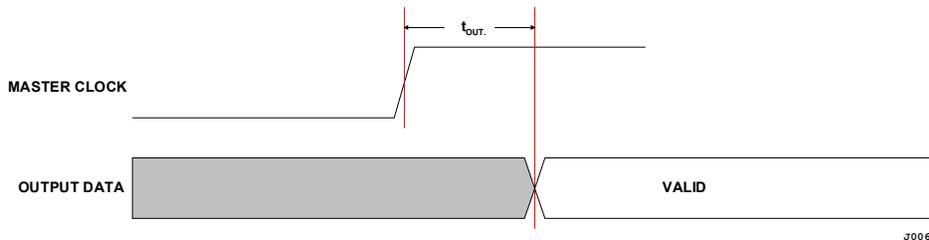
The data being clocked into the L1CTOC FPGA has nominal setup time of 10ns and a hold time of 8ns. The UCF constraint

```
net my_signal offset=in 8ns before link_clock
```

is used on all inputs. Hold time is not critical for FPGA inputs.



The FPGA outputs should be stable 8ns after the rising edge of the master clock:



The UCF constraint

```
net my_signal offset=out 8ns after clock;
```

is used on all high speed outputs such as the L1OUT bus and the L3OUT bus.

Several of the input link clocks must be routed on local (non-dedicated) traces in the FPGA. The clock skews on these nets are carefully checked, as data corruption has been observed when the clock skews become excessive above  $\sim 3$ ns.

In order to minimize local clock skew and clock delay the following tricks were used:

- use distributed rams in the deskew circuit and physically constrain them to be located near where the input link bus enters the die.
- deskew modules for links3-link9 were *not* packed into the IOBs (helps minimize clock delay)
- UCF constraint MAXSKEW=500ps was used on link3\_clk through link9\_clk to prevent the place and route tools from using the “secondary low skew lines”. (This constraint was not met, but it forced the router to pack the logic tightly.)

## 9. DEVICE RESOURCES

The L1CTOC design fits into a Xilinx Virtex 600 device:

### Device utilization summary:

Number of External GCLKIOBs	4 out of 4	100%
Number of External IOBs	349 out of 404	86%
Number of LOCed External IOBs	349 out of 349	100%
Number of BLOCKRAMs	21 out of 24	87%
Number of SLICES	2809 out of 6912	40%
Number of GCLKs	4 out of 4	100%

The design is written entirely in VHDL. It has been functionally simulated in Active-HDL and Synthesized using Xilinx XST. The design was placed-and-routed using Xilinx ISE 5.x backend tools.

## 10. REVISION HISTORY

- 0.01 11 March 2003 created.
- 0.02 18 March 2003 added motherboard interface stuff, L3 sender, Core Design
- 1.00 19 March 2003 added more core design, L1formatter, latency diagrams, timing stuff.
- 1.01 20 March 2003 fixed typo in L3 sender link module section
- 1.02 25 March 2003 changed occupancy OL\_OCT definition.
- 1.03 07 April 2003 added L3 record status bits.

## 11. REFERENCES

1. DFE protocols document, maintained by Levan Babukhadia. Currently available at:  
<http://d0server1.fnal.gov/projects/VHDL/General/ctt-protocols-v07-00.pdf>