ATLAS Level-1 Calorimeter Trigger  

Local Trigger Module (LTM)  

A module to convert CMM LVDS outputs into NIM levels for local triggering use.  

Design Specification and User Guide  

Version: 1.05  
Date: 22\textsuperscript{nd} February 2010  

Richard Staley  
University of Birmingham
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>Document History</td>
<td>2</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>3</td>
</tr>
<tr>
<td>2 Requirements</td>
<td>3</td>
</tr>
<tr>
<td>3 Implementation</td>
<td>5</td>
</tr>
<tr>
<td>3.1 LVDS Input Stage</td>
<td>5</td>
</tr>
<tr>
<td>3.2 BUSYN Input</td>
<td>5</td>
</tr>
<tr>
<td>3.3 Simple Dead-time Veto</td>
<td>6</td>
</tr>
<tr>
<td>3.4 Complex Dead-time (Leaky Bucket algorithm)</td>
<td>6</td>
</tr>
<tr>
<td>3.5 NIM outputs</td>
<td>6</td>
</tr>
<tr>
<td>3.6 Front Panel</td>
<td>7</td>
</tr>
<tr>
<td>3.7 Switch settings</td>
<td>8</td>
</tr>
<tr>
<td>3.7.1 Input Logic functions</td>
<td>8</td>
</tr>
<tr>
<td>3.7.2 Simple Dead-Time Veto</td>
<td>8</td>
</tr>
<tr>
<td>3.7.3 Complex Dead-Time Veto</td>
<td>8</td>
</tr>
<tr>
<td>3.7.4 Output Delay</td>
<td>8</td>
</tr>
<tr>
<td>3.8 Module Layout</td>
<td>9</td>
</tr>
<tr>
<td>4 Project Management</td>
<td>10</td>
</tr>
<tr>
<td>4.1 Deliverables</td>
<td>10</td>
</tr>
<tr>
<td>4.2 Personnel</td>
<td>10</td>
</tr>
<tr>
<td>4.3 Design and Verification</td>
<td>10</td>
</tr>
<tr>
<td>4.4 Manufacturing</td>
<td>10</td>
</tr>
<tr>
<td>4.5 Test</td>
<td>10</td>
</tr>
<tr>
<td>4.6 Costs</td>
<td>10</td>
</tr>
<tr>
<td>Appendix A LTM User Guide</td>
<td>11</td>
</tr>
<tr>
<td>5 Creating Memory file with Quartus</td>
<td>11</td>
</tr>
<tr>
<td>Appendix B CMM Front-panel connector</td>
<td>14</td>
</tr>
<tr>
<td>Appendix C Other versions / revisions</td>
<td>15</td>
</tr>
</tbody>
</table>

## Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Main changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Production version (previously called LTN)</td>
</tr>
<tr>
<td>1.02</td>
<td>Switch settings added and BUSY timing corrected.</td>
</tr>
<tr>
<td>1.03</td>
<td>FPGA configure LED added</td>
</tr>
<tr>
<td>1.04</td>
<td><strong>Delay Stage moved after Veto circuit</strong> + L1A variant created</td>
</tr>
<tr>
<td>1.05</td>
<td>BUSYN input synchronised.</td>
</tr>
</tbody>
</table>

Updates to this document “LTM_Specification.pdf” and other information can be found at:

http://www.ep.ph.bham.ac.uk/user/staley/LTM/
1 Introduction

This document describes the specification for a proposed Local Trigger Module.

1.1 Overview

For commissioning of the L1 Calo Trigger Processor it is useful to self trigger the local system without involving the CTP. This module will translate the LVDS signals from a CMM into NIM level trigger signals required by the LTP/TTCvi sub-system, while performing some simple logic and veto functions.

The outputs from a CMM are defined by the type of crate this module is used in, so the LTM’s design must take account of this.

2 Requirements

• Receive 32* LVDS signals and clock from CMM via 68 way MDR type cable connector. Parity of input data will be checked and errors indicated.

• Output 12 NIM signals as a selection of, or as simple logic functions of, the LVDS inputs:

  o Individual inputs passed through 0 to n-1
  o Individual inputs passed through n to 2n-1
  o Logic OR of groups of 3 bits (for CP hits) i.e. non-zero count.
  o Logic OR of groups of 4 bits (for energies)

• Output 1 NIM version of LVDS clock input

• Provide adjustable coarse delay to all output signals
  o Delay of N BC periods (N = 1 to 15)

• Apply veto to each output signal
  o Simple dead-time of T BC periods (25ns) where (T = 1 to 50)
  o Complex dead-time for N triggers within 80us where (N = 1 to 8)

• Receive BUSY signal from ROD (active low)
  o CTP ‘TTL’ signal but reconfigurable to accept NIM signal.

*There are actually 33 LVDS channels on the cable, including a clock signal and a parity bit
Block Diagram showing order of functions.
3 Implementation

Constructed to occupy 1 slot of a 6U high VME crate. This version will not have a VME interface but only use the backplane to provide power.

The module will use a single FPGA, an Altera Cyclone II FPGA EP2C5, to perform all the logical and delay functions. The design uses pipelines for the delay and deadtime functions and requires a 25ns clock from the CMM. Incoming signals are sampled by the falling edge of the CMM clock, outgoing signals are clocked out on the rising edge.

The Incoming LVDS signals are received directly by the FPGA, but the outputs of the FPGA need translation to NIM levels. The translation will be done using a number of small daughter cards, one per NIM output. This gives extra flexibility to the module as it may be easily adapted to output other logic standards, or even to receive extra signals back into the FPGA, by changing the daughter card.

The grouping of the inputs from each type of CMM are shown by the table in Appendix A.

3.1 LVDS Input Stage

Note that a clock duty-cycle constraint requires the clock low time \( t_{cl} \) to be > 8 ns.

3.2 BUSYn Input

The BUSYn input (active low) can be configured to accept either TTL signals (as sent from a ROD), or NIM level signals. This input will go inactive if unconnected.

The hit outputs are gated with the inverted Busy signal just before being converted to NIM. The Busy Signal is sampled by the BC clock to prevent glitches on the outputs.

For setting the timing of the incoming BUSYn signal, a NIM level clock output is provided. BUSYn will be sampled by the rising edge of the incoming CMM clock.
3.3 Simple Dead-time Veto

This algorithm will apply to each output individually. Once triggered by an input, this circuit will veto any further activity until the set delay has passed.

3.4 Complex Dead-time (Leaky Bucket algorithm)

This algorithm will apply to each output individually. A counter generates a time-stamp value for each hit which will be written into a FIFO memory. Once the earliest time-stamp stored in the FIFO matches the wrapped-around time it is unloaded from the FIFO. The Veto will be active whenever the FIFO is full, the Veto will be active, and the FIFO will not take anymore time-stamps.

The counter has a modulus equal to the dead-time in BCs ($3200 = 80\mu s$), and the FIFO depth defines the number of hits allowed within this time. The FIFO depth may be set to values from 1 to 8. A depth of ‘0’ will disable the Complex Dead-time veto.

There will be additional logic to recover from error states where the FIFO holds corrupted count value.

3.5 NIM outputs

The NIM signals will be driven negative by a 16mA current source for an active level, and 0mA for an inactive level. With a 50 ohm load this gives the standard -0.8V NIM signal for an active output.

Maximum open-circuit voltage is -1.6V.
3.6 Front Panel

- Power +5V, -5V
- Inputs from CMM – 33* LVDS (31 Hit bits + Parity + Clock)
- 12 NIM Outputs (to LTP) with activity LEDs
- Output Delay set in BCs 0 - 15
- Input Logic Groups / mode 0 – 3,7
- Complex Veto Depth 1 - 8
- BUSY LED
- BUSY Input – TTL (or NIM)
- Monitor Output (clock) – NIM

FPGA Configured OK
CMM Input Parity Error

Note: Simple dead-time switches are mounted inside the module.
3.7 Switch settings

### 3.7.1 Input Logic function groups

<table>
<thead>
<tr>
<th>Switch value</th>
<th>Function</th>
<th>NIM Outputs (0..11) = f (LVDS inputs 0..30)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pass through</td>
<td>0   1  ... 7 8 9 10 11</td>
</tr>
<tr>
<td>1</td>
<td>Shift</td>
<td>12  13 ... 19 20 21 22 23</td>
</tr>
<tr>
<td>2</td>
<td>3 input OR</td>
<td>0,1,2 3,4,5 ... 21,22,23 * * * *</td>
</tr>
<tr>
<td>3</td>
<td>4 input OR</td>
<td>0,1,2,3 4,5,6,7 ... 28,29,30,P * * * *</td>
</tr>
<tr>
<td>7</td>
<td>Test mode</td>
<td>Pulse0 Pulse1 ... Pulse7 * * * *</td>
</tr>
</tbody>
</table>

* = output disabled

P = Parity bit

### 3.7.2 Simple Dead-Time Veto

<table>
<thead>
<tr>
<th>Switch value</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 – 79 (decimal)</td>
<td>Value = Length of veto in BC ticks after previous output</td>
</tr>
</tbody>
</table>

### 3.7.3 Complex Dead-Time Veto

<table>
<thead>
<tr>
<th>Switch value</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>All outputs disabled</td>
</tr>
<tr>
<td>1..8</td>
<td>Value = number of triggers allowed in 3200 BC ticks (80us)</td>
</tr>
<tr>
<td>F</td>
<td>Disable complex dead-time veto</td>
</tr>
</tbody>
</table>

### 3.7.4 Output Delay

<table>
<thead>
<tr>
<th>Switch value</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..15</td>
<td>Value = number of BC clock ticks trigger is delayed by</td>
</tr>
</tbody>
</table>

\[
\text{Delay} = n \cdot \text{BC} + t_c \\
\text{where } t_c = 10 \text{ ns}
\]
3.8 Module Layout

- **LVDS in**
- **FPGA Configured**
- **Logic Mode**
- **Simple Delay**
- **Complex Size**
- **Power +5V, -5V**
- **NIM out**
- **BUSY**

**VME**
- +5V Power Only

**FPGA**
- QFP208
- TP3, TP4, TP5, TP7
- 555
- TP2
- JTAG (test use only)
- Firmware Load
- EPC
- Tens Units
  - Simple Dead-time
- Polyfuse

**Power**
- +5V, -5V

**FPGA Configured**
- 10, 11
- 8
- 9
- 6
- 7
- 4
- 5
- 2
- 3
- 0
- 1

**NIM out**
- 10, 11
- 8
- 9
- 6
- 7
- 4
- 5
- 2
- 3
- 0
- 1

**BUSY**
- TTL/NIM
4 Project Management

4.1 Deliverables

- Specification (this document).
- 1 prototype LTM followed by 6 production modules.
- Firmware for programmable logic.
- Design documentation (schematics, layout information, component data-sheets etc.)

4.2 Personnel

The LTM is designed at Birmingham University to specifications set by members of the Level-1 Calorimeter Trigger Collaboration.

4.3 Design and Verification

- PCB Schematics and layout using OrCAD 9.2 Software.
- FPGA design and verification using Altera Quartus 7.2 Software

4.4 Manufacturing

The module is based on a 4 layer PCB manufactured by PCB-Pool.

The modules will be assembled by technicians at Birmingham University. 1 prototype module will be made and tested before the remaining ones are produced.

4.5 Test

The modules will go through a number of stages before being released to CERN:
- Using pulse generator and oscilloscope to check signal levels and basic operation.
- Module will be placed into Birmingham test rig and tested with signals from CMM.

4.6 Costs

<table>
<thead>
<tr>
<th></th>
<th>NRE costs</th>
<th>Cost each in UKP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 PCB - 4 Layer prototype / 8 day</td>
<td>£22 silk screen stencil</td>
<td>£245</td>
</tr>
<tr>
<td>6 PCBs - 4 Layer production / 8 day</td>
<td>£22 silk screen stencil</td>
<td>£148</td>
</tr>
<tr>
<td>Components</td>
<td>-</td>
<td>£277</td>
</tr>
<tr>
<td>Assembly</td>
<td>-</td>
<td>Birmingham Technicians</td>
</tr>
</tbody>
</table>

Costs given above exclude VAT.
Appendix A LTM User Guide

5 Creating Memory file with Quartus

The following shows the correct settings within the Device and Pin options menu in Quartus (V7.2) prior to generating the FPGA configuration memory file.
The compiler output has to be converted, and compressed for fitting into the EEPROM:

So from File menu choose

Convert Programming Files:
The generated memory image is downloaded using the Altera programming software, with a ByteblasterII cable plugged into the shrouded header pin connector (NOT JTAG).

View of the Programmer window:
## Appendix B - CMM Front-panel connector

Pinout for various CMM firmware revisions

<table>
<thead>
<tr>
<th>Pin Pair</th>
<th>CP sk#1</th>
<th>JET sk#1</th>
<th>JET sk#2</th>
<th>Energy sk#1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Thr0 b0</td>
<td>Thr0 b0</td>
<td>FwdLthr0 b0</td>
<td>Esum b0</td>
</tr>
<tr>
<td>2</td>
<td>Thr0 b1</td>
<td>Thr0 b1</td>
<td>FwdLthr0 b1</td>
<td>Esum b1</td>
</tr>
<tr>
<td>3</td>
<td>Thr0 b2</td>
<td>Thr0 b2</td>
<td>FwdLthr1 b0</td>
<td>Esum b2</td>
</tr>
<tr>
<td>4</td>
<td>Thr1 b0</td>
<td>Thr1 b0</td>
<td>FwdLthr1 b1</td>
<td>Esum b3</td>
</tr>
<tr>
<td>5</td>
<td>Thr1 b1</td>
<td>Thr1 b1</td>
<td>FwdLthr2 b0</td>
<td>Etmiss b0</td>
</tr>
<tr>
<td>6</td>
<td>Thr1 b2</td>
<td>Thr1 b2</td>
<td>FwdLthr2 b1</td>
<td>Etmiss b1</td>
</tr>
<tr>
<td>7</td>
<td>Thr2 b0</td>
<td>Thr2 b0</td>
<td>FwdLthr3 b0</td>
<td>Etmiss b2</td>
</tr>
<tr>
<td>8</td>
<td>Thr2 b1</td>
<td>Thr2 b1</td>
<td>FwdLthr3 b1</td>
<td>Etmiss b3</td>
</tr>
<tr>
<td>9</td>
<td>Thr2 b2</td>
<td>Thr2 b2</td>
<td>FwdRthr0 b0</td>
<td>Etmiss b4</td>
</tr>
<tr>
<td>10</td>
<td>Thr3 b0</td>
<td>Thr3 b0</td>
<td>FwdRthr0 b1</td>
<td>Etmiss b5</td>
</tr>
<tr>
<td>11</td>
<td>Thr3 b1</td>
<td>Thr3 b1</td>
<td>FwdRthr1 b0</td>
<td>Etmiss b6</td>
</tr>
<tr>
<td>12</td>
<td>Thr3 b2</td>
<td>Thr3 b2</td>
<td>FwdRthr1 b1</td>
<td>Etmiss b7</td>
</tr>
<tr>
<td>13</td>
<td>Thr4 b0</td>
<td>Thr4 b0</td>
<td>FwdRthr2 b0</td>
<td>*</td>
</tr>
<tr>
<td>14</td>
<td>Thr4 b1</td>
<td>Thr4 b1</td>
<td>FwdRthr2 b1</td>
<td>*</td>
</tr>
<tr>
<td>15</td>
<td>Thr4 b2</td>
<td>Thr4 b2</td>
<td>FwdRthr3 b0</td>
<td>*</td>
</tr>
<tr>
<td>16</td>
<td>Thr5 b0</td>
<td>Thr5 b0</td>
<td>FwdRthr3 b1</td>
<td>*</td>
</tr>
<tr>
<td>17</td>
<td>Thr5 b1</td>
<td>Thr5 b1</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>18</td>
<td>Thr5 b2</td>
<td>Thr5 b2</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>19</td>
<td>Thr6 b0</td>
<td>Thr6 b0</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>20</td>
<td>Thr6 b1</td>
<td>Thr6 b1</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>21</td>
<td>Thr6 b2</td>
<td>Thr6 b2</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>22</td>
<td>Thr7 b0</td>
<td>Thr7 b0</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>23</td>
<td>Thr7 b1</td>
<td>Thr7 b1</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>24</td>
<td>Thr7 b2</td>
<td>Thr7 b2</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>25</td>
<td>*</td>
<td>JetEt b0</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>26</td>
<td>*</td>
<td>JetEt b1</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>27</td>
<td>*</td>
<td>JetEt b2</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>28</td>
<td>*</td>
<td>JetEt b3</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>29</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>30</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>31</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>32</td>
<td>CLK</td>
<td>CLK</td>
<td>CLK</td>
<td>CLK</td>
</tr>
<tr>
<td>33</td>
<td>Parity</td>
<td>Parity</td>
<td>Parity</td>
<td>Parity</td>
</tr>
<tr>
<td>34</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
</tr>
</tbody>
</table>
Appendix C - Other versions / variants

L1A Version (Also PCB/2)

- Input from CMM – 32* LVDS (30 Hit bits + Parity + Clock)
- Logic Delay set in BCs 0 - 15
- Complex Dead-time Depth 1 - 8
- Power +5V, -5V
- DSS L1An- Input – ECL
- TTCvi L1An Synch - Output – ECL
- TTCvi Clock - Input – NIM
- L1An Gated - Output – NIM
- 8 Hit Outputs (to LTP) - NIM with activity LEDs
- BUSY LED
- BUSYn Input – TTL (or NIM)
- Monitor Output (TTCvi clock) – NIM

Prototype Version (using PCB/1)

As Production version, except for:

- FPGA Configured OK
- CMM Input Parity Error

The L1A variant is for use at Birmingham, and will provide a veto of the L1A trigger (generated by the DSS) by the ROD BUSY signal, without having to add a whole NIM crate for this. The L1An input (active low) is terminated by 50Ω to -2.0 V. The L1An output to the TTCvi has an additional 220 pull-down to -5V.

To get an ECL inputs on the LTM, a NIM output daughter cards was modified and the receiving pin on the FPGA being reprogrammed as an input.