HCAL
Trigger Primitive Generator and Readout System

CMS week
March, the 3rd 2001

Tullio Grassi / Drew Baden
University of Maryland
Outline

• System Overview

• Trigger-Path

• DAQ-Path

• Status

• Schedule

Algorithms
Hardware constraints
Present schemes
Possible modifications
Readout System Overview

Data from:
HB, HO (no TP), HE, HF

1 channel in Mantissa/ADC format
1 depth of Trigger Tower

1 Trig. Tower = \[ \begin{cases} 1 \text{ HCAL tower for } \eta < 1.83 \\ \frac{1}{2} \text{ HCAL tower for } 1.83 < \eta < 3 \end{cases} \]

Need of a map of depth segmentations vs electronic channel
L1-Path Latency estimation (in BXs)

FE Data → RX stage G-link Rx, X-Clks, fiber-to-fiber → Linearizer LUT, $E_T$ and Sum → Filters (Energy and BCID) → Non-Lin Compr. (LUT ?) → TX stage SyncTx/Rx Vitesse, 20m cable → Trigger Primitives

| Theoretic | 6 | 2 | 8 (or 2?) | 1 | 5 | Total=22 |
| Contingent | 7 | 3 | 8 (or 2?) | 2 | 7 | Total=27 |

Requirement from TDR = 23 BX !!

⇒ Complexity on trigger path should be kept to a minimum
Input Stage

- Optical rx
- G-link rx
- RXData
- Recov. CK
- System CK
- Clock Synchro nization
- Fiber to Fiber Alignment
- Pat. Gen. RAM
- Mux
- Channel A
- Channel B
- Cap ID
- 3 BX
- 1 BX
- 2 BX
- Conceptually belongs to VFE and detector link
- Hard to implement in less than 25 ns
Output Stage
SyncTx/Rx - Link - Cable
-- similar to ECAL --

TP Data

SyncTx/Rx circuit (from ECAL)

Vitesse TX Link Board (from ECAL)

20m copper Serial Link

1.5 BX

~3 BX

66ns ~3 BX
Which is the maximum number of channels (= depths ?) that I need to sum?

Strong impact on the design: the more channels the more expensive and the longer latency!

Compression Algorithm?
L1-TPG Filter

**Energy filter:** baseline subtraction and amplitude extraction in presence of noise.

**Time filter:** Bunch Crossing IDentification.

Proposed weights: 
\[-1.5, -1.5, 1, 1, 1\]. OK more precise: 
\[n \times \frac{1}{2}, \frac{1}{4}, \frac{1}{8} \ldots\]

What about 4 weights?

Do we really need a peak finder (MAX) for BCID?
Can we use different algorithms? (e.g. thresholds, zero-crossing…)}
Trigger path - variations

Do I sum channels with $\neq \eta$? In this case the $E_T$ conversion should be before L1-Filter

Design idea:

- Linearizer + $E_T$ calculation on the same Look-Up Table
  - $\Rightarrow$ faster, denser

Problem:

- $\mu$ bit should be in Trigger Tower Energy. Could it be on raw energy of the bigger depth?

Diagram:

- Lineariz. & $E_T$
- $E_T$
- Sum
- L1-TPG Filter
- Compr. LUT?
- $E_T,\text{compr}$
- $\mu$ bit
- Output stage
- to DAQ Path
- to CRT
Possible Requirements:

- Option to send everything without Zero Suppression?
- At which trigger rate?
- Split the filter in 2: a simple one before ZS, another one after
Hi all;

I don't need to tell anyone that leaving [the energy extraction algorithm] to Level-2 is the most flexible solution. And certainly, when we first turn on we will (anyway) do everything in Lvl-2 even if the hardware can do the job -- as a cross-check that is. The question I suppose is whether we should decide, today, that the hardware should *never* have this functionality. Could we not envisage mounting the FPGA (I assume it's not an ASIC...) on an elevated plug-in board so that we can upgrade to the latest greatest FPGA as they become available? Already, today's chips make the stuff of three years ago look/feel like ancient history. It's not far from the current extrapolation line to imagine that there will be enough flexibility to execute fairly complicated algorithms in an FPGA.

Just a thought...

Best, Paris
L2 Filter

No constraints on latency ⇒ 10-weight filter is possible

We do not have clear specifications

L2 Filter need more studies for high luminosity (run in 2008)

Proposal:
• No L2-filter for 2005
• Only reserve resources ⇒ How?
• Impact on Zero-Sup?

Reserve resources on 2005 programmable circuits
⇒ Low cost

Foresee to upgrade the programmable circuits in 2008
⇒ Extra costs and potentialities
Testing

Demonstrator project at UMD:
FE-Emulator + HTR + DCC + CPU + TTC
Focus on hardware compatibility, data transmission, protocols hardware/software interfaces.
Goal: inject an event, assign it to the correct BX, trigger on it and read it out.

FNAL integration of FrontEnd + Readout Electronics.
Focus on calibration with source: no energy extraction algorithms.
Status

• Just received the HTR board; some debugging has started

• FPGA design still very rough

• Some re-use of ECAL designs is in progress, but should be more systematic ⇒ would allow re-use of ECAL simulations and VME software

• So far minimal interaction with Salavat and Sarah
  ⇒ future collaboration can improve the implementation of the algorithms
## Schedule

<table>
<thead>
<tr>
<th>Period</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 2001</td>
<td>Design of FPGAs and testing of HTR and Emulator cards.</td>
</tr>
<tr>
<td>April-June 2001</td>
<td>Demonstrator Readout integration at UMD</td>
</tr>
<tr>
<td>July-Sept. 2001</td>
<td>Source +FEE + Readout at FNAL (additional task)</td>
</tr>
<tr>
<td>Summer 2001</td>
<td>Start Prototype, should be very close to the final version ⇒ <em>I need more refined specifications</em></td>
</tr>
</tbody>
</table>

⚠️ We are behind on the original schedule