B02 - Trigger algorithm development, performance, and demonstration

Nhan Tran
HL LHC CMS CD-1 Review
October 22-24, 2019
Technical design
   Algorithms overview and physics performance
   Algorithm design
   Firmware demonstration

Management aspects
   Risk, quality assurance
   Milestones and progress
   Cost and schedule (see talk later by Jeff)
Nhan Tran

Wilson Fellow (Fermilab)
- L3 Manager: Correlator trigger
- Development of Particle Flow and PUPPI in L1 trigger
- Lead on hls4ml: high level synthesis for machine learning

Postdoc (Fermilab)
- Track trigger ASIC development and testing for Vertically Integrated Pattern Recognition Associative Memory (VIPRAM)
- Development of PUPPI algorithm
TECHNICAL DESIGN
Trigger Scope Overview

- **DOE Trigger/DAQ scope**
  - Endcap Calorimeters
  - Barrel Calorimeters
  - Outer Tracker Detector
  - Endcap Muon System
  - Barrel Muon System
  - Pixel Tracker
  - MIP Timing Detector

- **NSF Trigger/DAQ scope**
  - Barrel Calo Trigger RCT (24 Boards)
  - Endcap Muon Track Finder (12 boards)
  - BMU BE

- **Other US CMS scope**
  - Barrel Calo Trigger GCT (10 Boards)
  - Endcap Muon Track Finder (12 boards)
  - BMU BE
  - Overlap Muon Track Finder

- **Correlator Trigger Layer-1** (42 Boards)
  - 24 kHZ to HLT

- **Correlator / Global Trigger Layer-2**
  - 750 kHZ to HLT
  - 7.5 kHZ to Offline

**BE+L1 System**
- 40,000 kHz event data processing
Design Considerations

Maintain performant trigger under high luminosity conditions

Upgrade L1 trigger accept rate: 750 kHz
Upgrade L1 trigger total latency: 12.5 μs

Detector/Trigger Upgrades

Tracking trigger for tracks with pT > 2 GeV
New high granularity endcap calorimeter
Full crystal readout of barrel ECal
New muon detectors for improved high η coverage and higher granularity readout

DOE trigger scope

402.6.3: Calorimeter (regional barrel and forward calorimeter, global)
402.6.5: Correlator trigger (combining muon, calorimeter, tracker inputs)
General Algorithm Strategy

Deliver a suite of algorithms which cover both robustness and optimized physics performance

**Single system triggers***

*Robust, simpler algorithms*

- Global Calorimeter Trigger objects
- Track-only Trigger objects

**Multi-system optimized reconstruction**

*More complex, performant algorithms*

- Track + muon + calorimeter correlated (particle flow and PUPPI) trigger objects

* muon system only triggers in NSF scope
Trigger overview

- First ~5 ms spent on regional processing, track finding, and CE clustering
- Next ~4.5 ms consumed by triggerable object matching and reconstruction (2.5 ms), followed by global trigger decision (1 ms), and finally TCDS communicating decision to front ends (1 ms)
- +3 ms latency is held as contingency
- Critical path is through Particle Flow and its latest inputs (CE, Track Finder)
- Our design is already saturating our 9.5 ms budget; latency contingency will have to be carefully managed globally hereafter!
Starting from the lower left, we have a ~5 ms phase spent on regional processing, track finding, and CE clustering.

Next, we have a ~4.5 ms phase devoted to triggerable object matching and reconstruction, which takes approximately 2.5 ms. This is followed by a global trigger decision taking ~1 ms, and finally, TCDS communicates the decision to the front ends in ~1 ms.

Additional latency of ~3 ms is held as a contingency.

The critical path is through Particle Flow and its latest inputs (CE, Track Finder).

Our design is already saturating our 9.5 ms budget; latency contingency will have to be carefully managed globally hereafter!
Deliverables

Devise algorithms and study physics performance for the correlator (Particle Flow) and calorimeter triggers

Implement and integrate algorithms into firmware
   *Within latency, bandwidth, and resource requirements*

From *robust*, single-system algorithms to *optimized* multi-system algorithms
   *Calorimeter and Track-only objects*
   *Particle Flow with PUPPI*
Functional algorithm diagram
Functional algorithm diagram

- **Calorimeter clustering and ID and calibration**
- **Track propagation**
  - Particle Flow algorithm
  - PUPPI algorithm
- **Vertexing**
  - Track jets
  - Track MET
- **Calo Jets, MET, EG objects, taus**
- **Track jets**
- **Track MET**
- **PF PHYSICS OBJECTS**
- **INTER-OBJECT CORRELATION AND GLOBAL TRIGGER**
- **TRACK+MUON OBJECTS**
- **MUON-ONLY OBJECTS**
- **ENDCAP CALORIMETER**
- **TRACK TRIGGER**
- **MUON SYSTEMS**
Steps of algorithm development path:

Understand physics case
Define algorithm and interfaces (inputs & downstream)
Develop firmware, estimate latency and resources
Integrate within full trigger architecture
Demonstrate full implementation feasibility

n.b. difficult to present all steps for all algorithms

1) **Calorimeter clustering** example
2) **Vertexing** example
2) **Particle flow** example larger focus, most complex in terms of algorithm size and input interfaces
FPGA development of algorithms in languages like VHDL or Verilog (RTL) have long development cycles and require a lot of engineering support.

New tools: **HLS, high level synthesis**
C-level programming with specialized preprocessor directives which synthesizes optimized firmware.

**Particle flow example:**
Core first version of firmware developed in 2-3 months using HLS, only physicists.

Engineering firmware support still required (of course!) — our experience: system interfaces, infrastructure, and signal routing, etc.
Barrel **Regional Calorimeter Trigger** takes as input single ECal crystals and HCal towers to build clusters.

36 RCT Cards to cover barrel calorimeter

- **$\eta$**
- **$\phi$**

2x4 and 3x4 ECAL regions = total 216 regions

17$\eta \times$ 4$\phi$

1x1 ECAL region (tower) 1x1 HCAL region (tower)

0.0174 $\eta \times$ 0.0174 $\phi$ → 1 of 25 crystals

at the trigger

0.0872 $\eta \times$ 0.0872 $\phi$

at the trigger
Calorimeter clustering

- **Input:** 17x4x5x5 ECAL crystals and 16x4 HCAL towers.
- 1-RCT card covers 17$\eta$ x 4$\varphi$ towers
- Divide card in regions of 3$\eta$ x 4$\varphi$ towers to make clusters.

- Building clusters in 3$\eta$ x 4$\varphi$ region:
  - Search for seed crystal > 1 GeV
  - Make 3x5 clusters at crystal level
  - Select maximum of 5 highest ET cluster in 3$\eta$ x 4$\varphi$ region

- Move to next 3x4 towers and then do the merging around the neighbors if cluster is at the boundary of the tower

- For 1-RCT card, there are 5 – (3$\eta$ x 4$\varphi$) regions = 30 clusters
- Sort and send a maximum of 12 highest ET clusters

- To these 12 highest ET clusters, if there is a HCAL tower behind the ECAL tower, HCAL ET is also added to the cluster.

- **Output:** 12 Clusters (ECAL + HCAL)
First Barrel RCT algorithm has reasonable resource usage and latency of 40 clock cycles @ 240 MHz for large resource FPGA (VU9P)
Two 17x2 blocks split across SLRs for better floor planning

Candidate for smaller form factor FPGA
Vertexing algorithms

Vertexing can be done in parallel to particle flow but is needed for pileup mitigation techniques.

First “fast histogramming” algorithms implemented as a baseline.

Slide: Keith Ulmer

Board: VCU118
FPGA: VU9P
Clock: 320MHz (HLS)
Configuration 1 (tested):
  TMUX: 6
  Phi Segmentation: 12
  Binning: ~64 clock cycles
  Vertex Finding: 17 clock cycles
Configuration 2 (possible):
  TMUX: 18
  Phi Segmentation: 9
  Binning: ~148 clock cycles
  Vertex Finding: 17 clock cycles

The vertex is ready in plenty of time to pass it on to a PF/PUPPI board.
Vertexing algorithms

Algorithm implemented in HLS with VHDL wrapper to support feeding in tracks and reading out results

<table>
<thead>
<tr>
<th>VU9P</th>
<th>FF</th>
<th>LUT</th>
<th>BRAM</th>
<th>DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertexing</td>
<td>117735</td>
<td>154979</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Available</td>
<td>2364480</td>
<td>1182240</td>
<td>2160</td>
<td>6840</td>
</tr>
<tr>
<td>% used</td>
<td>5</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Algorithm finished in time to pass to PF+PUPPI
Particle Flow Engine

Use inspiration from offline reconstruction for best performance

**Particle Flow:**

efficient combination of complementary detector subsystems

**particle interpretation** of the event, improves any single system energy/spatial resolution

---

Detector | $p_T$-resolution | $\eta/\Phi$-segmentation
---|---|---
Tracker | 0.6% (0.2 GeV) – 5% (500 GeV) | 0.002 x 0.003 (first pixel layer)
ECAL | 1% (20 GeV) – 0.4% (500 GeV) | 0.017 x 0.017 (barrel)
HCAL | 30% (30 GeV) – 5% (500 GeV) | 0.087 x 0.087 (barrel)
Large gains from PF on jet and MET resolutions

arXiv:1706.04965 [PF paper]

**Particle flow impact**

**Improved jet pT resolution**

- **CMS Simulation**
  - Anti-\(k_T\), \(R = 0.4\)
  - \(|\eta^{\text{Ref}}| < 1.3\)

**Improved missing pT resolution**

- CMS Simulation
  - Calibration
  - Calculation

**Figure 13:** Jet energy resolution as a function of difference to gluon response for Calo jets (left) and PF jets (right). The lines, added to guide the eye, correspond to fitted functions with ad hoc parametrizations.

**Figure 14:** Absolute difference in jet energy response between quark and gluon jets as a function of regions. The lines, added to guide the eye, correspond to fitted functions with ad hoc parametrizations.

**Figure 15:** Relative resolution and resolution on the angular resolution, obtained with a Gaussian fit in each bin of \(\phi\).
**PF+PUPPI schematic**

- **Tracks**
- **Calo clusters**
- **Muons**
- **EM clusters**

**Vertexing**

**TK particles**

**PU estimate**

**PUPPI calculation**

**PF+PUPPI Cands**

- **μ - Tk linking**
- **EM Calo - Tk linking**
- **Calo - Tk linking**
- **PF Cands**

PF inherently local, Event regionalized into ~1.0η x 0.7φ blocks
Particle flow regions

**Barrel**
- HCal
- Ecal e/γ
- 3x3 tower clusters
  - Tracks
  - PV
  - PF (PFAlgo3)
  - Puppi

**Endcap**
- HGC 3D
  - PF (PFAlgoHGC)
  - Puppi

**Forward**
- HF
  - 3x3 tower clusters
  - PF (PFAlgo3)
  - Puppi
PF+PUPPI algorithms bring significant improvement for hadronic trigger objects

Continual improvements to algorithms
Jet algorithms still offline style, work in progress
Resources

Implementation, VU9P, 240 MHz, 18 FPGAs (barrel)

<table>
<thead>
<tr>
<th>Region Size</th>
<th>Initiation Interval</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>2</td>
<td>[25,20,15]</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>[26,20,14]</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>[24,17,13]</td>
</tr>
<tr>
<td>54</td>
<td>2</td>
<td>[22,15,13]</td>
</tr>
</tbody>
</table>

The particle-flow and PUPPI firmware assumes that each input object is encoded in 64 bits. Different information is associated with each input type. Every input encodes $p_T$ using 16 bits, $h$ using 10 bits, and $f$ using 10 bits. The additional information for each input type and the number of bits used to store it is given below.

- **Tracks**
  - $s(p_T)$ - 16 bits
  - $z_0$ - 10 bits
  - Pass tight quality - 1 bit

- **EM clusters**
  - $s(p_T)$ - 16 bits

- **Combined clusters**
  - $p_{EM}$ - 16 bits
  - Is EM? - 1 bit

- **Muons**
  - $s(p_T)$ - 16 bits

The resulting particle-flow and PUPPI particles are also allotted 64 bits, and the information is stored with the same precision as for the input objects. The PUPPI weight $a$ is stored using 10 bits. This scheme leaves at least 10 bits unused for output objects. This allows additional...
Architecture (Exploded View)
Latency

Stream in tracks + inputs
TM(18) * 25 ns

First track

Regionizer

+ Stream in PF regions

Last track

1 μs

~250 ns

PUPPI

200 ns

Transmit

550 ns

First region

PF

Last region

+100 ns

120 ns

450 ns
Demonstration Firmware

(More in Sridhara’s talk)

- **Gen-0 (in operation at UW and CERN)**
  - CTP7-based (3 cards, 96 total active links @ 10.0G available)
  - HLS interface grafted onto Phase 1 8b10b link infrastructure
  - 64-bit link interfaces

- **Gen-1 (in operation at UW)**
  - APd1 single card setup
  - Iridis-style 64b66b transport
  - Early APx shell functionality
  - Asynchronous processing clock

- **Gen-2 (underway)**
  - Multicard test setup
  - Iridis-style signaling and APx shell environment
  - Common emulated TCDS timebase in ATCA/MicroTCA crates using CTP7

*Versioned in Github/Gitlab*
Demonstration

Note: The APx firmware shell will support decoupling of algorithm clock from link or LHC clock and thus significantly relax algorithm timing constraints (Vivado HLS) and optimize algorithm latency.
Demonstration

APx Engineering Update

A. Svetek, U. Wisconsin, November 15, 2018

Note: The APx firmware shell will support decoupling of algorithm clock from link or LHC clock and thus significantly relax algorithm timing constraints (Vivado HLS) and optimize algorithm latency.
Bitwise simulation
Utilization within project requirements

Less than 50% (70%) for algorithm (total) firmware

Recent milestone: all elements integrated and meeting timing with full place & route
Demonstration

Infrastructure

Regionizer

Particle Flow and PUPPI
## Summary of algorithm status

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Baseline Algo</th>
<th>Firmware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calibration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Track prop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vertexing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PUPPI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trk jet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>τ's</td>
<td></td>
<td>in progress</td>
</tr>
<tr>
<td>calo e/γ</td>
<td></td>
<td>done</td>
</tr>
</tbody>
</table>

**Legend**
- **done**
- **in progress**
- **unstarted**

Suite of algorithms to meet physics needs (menu) demonstrated

Firmware for most resource intensive algorithms within system requirements to meet mission need.
R&D status

Achieved

- Full simulation framework for studying algorithm physics performance
- Algorithm development with High Level Synthesis (HLS) tools and bitwise validation of firmware/software
- Integration of HLS IP blocks into firmware infrastructure in multiple modes (direct to infra, through intermediate VHDL)

Needed before production

- Final specification of algorithm interfaces for board-to-board communication (with USCMS and iCMS technologies)
Institutions and contributed labor

Contributing institutions

Clustering and ID: UW
Calibration: MIT, Fermilab, UIC
Track propagation: TAMU
Muon-track correlation: UCLA, UF, TAMU, Fermilab
Vertexing and track-based objects: CU Boulder, Rutgers
Particle Flow and PUPPI: MIT, Fermilab, UIC
Calo-based objects: UW
<table>
<thead>
<tr>
<th>Threat</th>
<th>RT-402-6-02-D</th>
<th>TD - Board or parts vendor non-performance (DOE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>RT-402-6-03-D</td>
<td>TD - I/O performance does not meet requirements (DOE)</td>
</tr>
<tr>
<td>Threat</td>
<td>RT-402-6-04-D</td>
<td>TD - Additional board redesign is required (DOE)</td>
</tr>
<tr>
<td>Threat</td>
<td>RT-402-6-05-D</td>
<td>TD - Additional firmware development is required (DOE)</td>
</tr>
<tr>
<td>Threat</td>
<td>RT-402-6-06-D</td>
<td>TD - Baseline FPGA does not satisfy requirements (DOE)</td>
</tr>
<tr>
<td>Threat</td>
<td>RT-402-6-90-D</td>
<td>TD - Key Trigger or DAQ personnel need to be replaced (DOE)</td>
</tr>
<tr>
<td>Threat</td>
<td>RT-402-6-91-D</td>
<td>TD - Shortfall in Trigger or DAQ scientific labor (DOE)</td>
</tr>
</tbody>
</table>

**RT-402-6-05:** Risk Rank = Medium, firmware does not meet technical specifications
- Probability: 20%, Impact: $10-60k
- Mitigation: allocate more time to on-project engineering, or new hire

**RT-402-6-90:** Risk Rank = Low, due to lack of base funding
- Probability: 30%, Impact: $0-292k
- Mitigation: replace with costed labor, other USCMS or iCMS institutions
Quality assurance plan (cms-doc-13093,cms-doc-13318)
Flows down from the engineering requirements
Acceptance plans based on QC activities
For algorithms, basic requirements on resource and latency met with each firmware release
Important: SW and FW stored and maintained in repositories

<table>
<thead>
<tr>
<th>TD-QC-004</th>
<th>Design Verification and Measurement/Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The total (algorithm) firmware will be designed to execute within the required latency budget and not occupy more than 75% (50%) of chip resources involves analysis of logic utilization estimates during firmware synthesis.</td>
</tr>
</tbody>
</table>

Acceptance of each firmware release will involve validation activities that verify that the firmware fits within the chip resource limits and executes within the required latency.
• As with entire project, we follow the Integrated Safety Management Plan (cms-doc-13395) and have documented our hazards in the preliminary Hazard Awareness Report (cms-doc-13394)
Algorithm performance and firmware have progressed since 2018 CD1

Algorithms for: barrel calorimeter trigger, global calorimeter trigger, correlator (including vertexing, track-based objects)

Full demonstration system for algorithm firmware progressing

First demonstration performed

In sync with iCMS milestones for TDR in 2019
ADDITIONAL MATERIAL