A Probability-Optimized Fast Timing Trigger for the Belle II Time of Propagation Detector

L. Macchiarulo, X. Gao
Department of Electrical Engineering
University of Hawaii

K. Nishimura, G. Varner
Department of Physics
University of Hawaii
Agenda

- Introduction
- Basic Idea of the Trigger Algorithm
- Firmware Implementation and Test
- Conclusion
Introduction

- The Belle II detector operating at the future upgrade to the KEKB accelerator will help to search for physics beyond the Standard Model.
- Charged hadron identification is a vital element of the experiments success.
- Timing of Propagation (TOP) counter has been chosen as the primary particle identification device (PID) in the barrel region of Belle II because of its intrinsically fast timing.
- Sub 100 ps time resolution is usually only obtained in offline reconstruction.
- A trigger is needed to help reduce the data volume from out-of-time hits in Silicon Vertex Detector.

- Requirement of the trigger:
  - *Processing time (a couple of us).*
  - *Timing resolution (a couple of ns).*
16 detector sectors

- 16 logical staves:
  - 16 only for 1-bar case
  - 2x16 for 2-bar case

Each logical stave has 8 fiber links

"front-end trigger"

Combine hits:
- time resolution
- back-back, mult., etc.

1x global bPID

2x "combiners"

16x modules
Agenda

- Introduction
- Basic Idea of the Trigger Algorithm
- Firmware Implementation
- Conclusion
Basic Idea of the Algorithm

- Estimation of interaction point and time based on the received photon pattern is not straightforward:
  - A small separation between the horizontal borders of the counters folds the Cherenkov cone into degenerate patterns.
  - The estimation must be done in a couple of us.
- Our solution: taking advantage of the information contained in the PDF (probability density function) of the time and space data.
Basic Idea of the Algorithm (cont.)

Timing Information from the photon detector

200 PDFs are stored in the trigger.

Judge which PDF best matches the distribution of the received timing information

Report event time and position
Basic Idea of the Algorithm (cont.)

Both Time and Space information is used
Time quantization: 1ns

Both Time and Space information is used
Time quantization: 2ns

Only Time information is used
Time quantization: 1ns

Only Time information is used
Time quantization: 1ns
Background noise: 4x
Agenda

• Introduction
• Basic Idea of the Trigger Algorithm
• Firmware Implementation and Test
• Conclusion
Firmware Implementation and Test

- Firmware Implementation
- Firmware Test
Firmware Implementation

- Implemented in a Virtex 4 FPGA (XC4VFX40-10FFG672I).

![Diagram of firmware implementation]
Firmware Implementation (cont.)

- Pipelined-sorter is based on the merge-sorting algorithm.

- Resource usage (32-bit width):
  - No. of Slices: 360 (1%).
  - No. of Slice FFs: 310 (0%).
  - No. of 4 input LUTs: 663 (1%).
Firmware Implementation (cont.)

- Basic implementation of trigger

- Resource usage:
  - No. of Slices: 9453 (50%).
  - No. of Slice FFs: 14364 (38%).
  - No. of 4 input LUTs: 9766 (26%).
  - No. of RMB16s: 100 (69%).

- LUTs: 200 x 64 x 20 bits
- To save resource usage, only 100 correlators are used to perform the 200 correlation operations.
- The running frequency of the trigger block is twice the frequency of the sorter to avoid throughput bottleneck

Sorted timing info → Correlate with 200 LUTs → MAX → event time and position → Trigger → FIFO → Aurora TX stream interface control → Aurora Core

Diagram showing the flow of data through correlators, sorters, FIFOs, and interface controls.
Firmware Implementation (cont.)

- Resource usage of other parts

- Resource usage of the Aurora cores and the Aurora RX stream interfaces at the receiving interface:
  - No. of Slices: 3200 (17%).
  - No. of Slice FFs: 4344 (11%).
  - No. of 4 input LUTs: 5152 (13%).

- Resource usage of the Aurora core and the Aurora TX stream interface at the sending interface:
  - No. of Slices: 416 (2%).
  - No. of Slice FFs: 534 (1%).
  - No. of 4 input LUTs: 689 (1%).
Firmware Implementation (cont.)

- Implemented in a Virtex 4 FPGA (XC4VFX40-10FFG672I).

- Overall resource usage:
  - No. of Slices: 14255 (76%);
  - No. of Slice FFs: 21112 (56%);
  - No. of 4 input LUTs: 16923 (45%);
  - No. of RAMB16s: 110 (76%).
Firmware Implementation and Test

- Firmware Implementation
- Firmware Test
Firmware Test

- RTL (register transfer level) simulation test
- In-chip test (via Xilinx Chipscope)
RTL simulation test

- 5000 timing patterns are generated by Monte Carlo simulation.
- The whole design (including all the aurora cores, stream interfaces, sorter, trigger, FIFOs) is tested simultaneously.
- 100% code coverage is obtained.
- Function coverage is under construction. (needs a detailed description of the protocol).
Firmware Test

- RTL simulation test
- In-chip test (via Xilinx Chipscope)
In-Chip Test (via Xilinx Chipscope)

- Since currently we only have one board, Aurora interfaces and other parts are tested separately.
- **Aurora interface test**
- **Sorter-trigger test**
Aurora Interface Test

- Test covers: Aurora core, Aurora RX stream interface and Aurora TX stream interface.

- Throughput: 2.4 Gbps
- Latency: 0.6 us
In-Chip Test (via Xilinx Chipscope)

- Since currently we only have one board, Aurora interfaces and other parts are tested separately.
- Aurora interface test
- Sorter-trigger test
Sorter-trigger Test

- Test covers: sorter logic, trigger logic and FIFOs.
- Throughput: 75M time words per second
- Latency: 0.8 us
Firmware Test (summary)

- Overall Performance
  - Throughput: *75M time words per second*;
  - Latency:
Agenda

- Introduction
- Basic Idea of the Trigger Algorithm
- Firmware Implementation and Test
- Conclusion
Conclusion

- We have prototyped a probability-optimized fast timing trigger for the Belle II TOP detector.
- The algorithm has been implemented and tested in a Xilinx Vertex 4 FPGA.
- Future work will optimize the trigger performance under various background noise and experimental conditions, including overlapping multiple track bits.
Thanks!