



Contribution ID: 225

Type: Poster

A New Way to Implement High Performance Pattern Recognition Associative Memory in Modern FPGAs

Wednesday, 30 September 2015 17:24 (1 minute)

Pattern recognition associative memory (PRAM) devices are parallel processing engines which are used to tackle the complex combinatorics of track finding algorithms. PRAM implementation has been mostly done with ASIC for high pattern density. However, implementation of PRAM in FPGAs allow for quick iterations, making it an ideal hardware platform for designing and evaluating new PRAM features before committing to silicon. For example, modeling in FPGAs can bring the system interface to maturity much sooner and minimize the ASIC design cycles. In this talk we present our FPGA-based PRAM design that is optimized for modern FPGA architectures.

Summary

The strength of the FPGA PRAM implementation lies not in pattern density (for in this area the ASIC implementation is clearly superior) but in the inherent flexibility of firmware. Using an FPGA to model a subset of a PRAM device enables circuit designers to quickly design, implement, and evaluate new features such as pipelined operation, high speed serial I/O, etc. The effect these changes have on the overall system can then be evaluated quickly.

In our experience we have found that logic blocks which have been optimized for fine-grain ASIC architectures often do not implement efficiently in coarse-grain FPGA logic cells. For example, the CAM cell is fully programmable and requires SRAM cells to store the pattern bits and comparator circuit for each bit. These CAM bits (which take up just a few transistors in an ASIC) would consume many logic cells in an FPGA and this leads to an inefficient use of logic resources. Our solution to this is to implement equivalent CAM cell functionality in such a way to take advantage of the Xilinx UltraScale SLICE-M logic cell architecture. (A side benefit of this new CAM cell design is that full ternary bit matching is supported with no additional logic resources consumed in the FPGA.) Likewise, cascaded Fischer trees used in ASIC PRAM backend for sorting and zero-suppression readout logic are large and slow when implemented in the FPGA. In this case, the FPGA backend logic was completely redesigned to use a multi-stage pipeline in order to meet our performance goals.

Despite extensive optimizations, large modern FPGAs are only able to implement subset of what could be achieved in an ASIC-based PRAM design. We have successfully implemented 1k (32x32) to 4k (64x64) associative memory pattern arrays and these designs consume 22% and 78% of a Kintex UltraScale KU040 FPGA. In the slowest speed grade device we are able to achieve 250MHz operation with a fixed output latency of 7 clock cycles.

The UltraScale FPGA-based PRAM design allows us to test new system interfaces such as single-ended parallel DDR buses, serialized LVDS, and high speed CML serial (GTX/GTH transceivers) up to 16 Gbps. A new mezzanine card, which is compatible with our Pulsar IIb board, has been designed to support both the ASIC and FPGA designs side-by-side for silicon based Level-1 tracking trigger R&D.

Primary authors: OLSEN, Jamieson (Fermilab); HOFF, Jim (Fermilab); WU, Jinyuan (Fermi National Accelerator Lab. (US)); LIU, Tiehui Ted (Fermi National Accelerator Lab. (US)); XU, Zijun (Peking University (CN))

Presenter: OLSEN, Jamieson (Fermilab)

Session Classification: Poster

Track Classification: Trigger