Contribution ID: **117** Type: **Oral**

Using MaxCompiler for High Level Synthesis of Trigger Algorithms

Tuesday 27 September 2016 09:50 (25 minutes)

Firmware for FPGA trigger applications at the CMS experiment is conventionally written using hardware description languages such as Verilog and VHDL. MaxCompiler is an alternative, Java based, tool for writing FPGA applications and removes some of the need for electronics expertise. This provides potential to lower the barrier for contribution to firmware design. An implementation of the jet and energy sum algorithms for the CMS Level 1 Trigger upgrade has been written using MaxCompiler to benchmark against the production VHDL implementation in terms of accuracy, latency, resource usage, and code size.

Summary

The use of conventional hardware description languages (HDLs) presents a barrier to contribution to trigger algorithms, leading to a code-base which is understood by only a few expert contributors. By investigating other tools available for firmware design it may be possible to open algorithm development up to a greater number of designers, and to improve the maintainability of the code. In the context of ongoing trigger upgrades at the CMS experiment, an improvement in ease of development and maintainability of algorithms would be highly beneficial. The planned track trigger and endcap calorimeter upgrades, in particular, will have a significant complexity in their algorithms.

Here MaxCompiler is investigated as a tool for the design of trigger applications. The tool uses a Java based language for firmware development, providing a higher level of abstraction than HDLs. Language aspects that ease the development compared to HDLs are explored, as well as design patterns that yield a low latency and resource usage. Other tool features that benefit development are presented, including functional design simulation that simplifies testing an algorithm on data.

The CMS Level 1 Calorimeter Time Multiplexed Trigger jet and energy sum algorithms have been implemented using MaxCompiler to demonstrate the feasibility of using such a tool for an existing trigger application. These algorithms are part of the Run II calorimeter trigger upgrade in which all data from a bunch crossing is passed through a single processor node. The MP7 board, as used in the trigger upgrade and which houses a Virtex-7 FPGA, is targeted. MaxCompiler 'kernels' have been written and compiled to integrate with the rest of the board infrastructure. The latency and resource usage is compared to the original design written in VHDL, to investigate any difference associated with using a higher level language. The design equivalence is demonstrated by comparing the output of each implementation on simulated data.

Author: SUMMERS, Sioni Paris (Imperial College Sci., Tech. & Med. (GB))

Presenter: SUMMERS, Sioni Paris (Imperial College Sci., Tech. & Med. (GB))

Session Classification: Programmable Logic, design tools and methods

Track Classification: Logic