Advanced Branch Predictors for Soft Processors

Di Wu, Jorge Albericio and Andreas Moshovos
Electrical and Computer Engineering Department
University of Toronto
peterwudi.wu@utoronto.ca, jorge@eecg.toronto.edu, moshovos@eecg.toronto.edu

Abstract—**The abstract goes here.**

I. Introduction

Field Programmable Gate Arrays (FPGAs) are increasingly popular to be used in embedded systems. Such designs often employ one or more embedded microprocessors, and there is a trend to migrate these microprocessors to the FPGA platform. Although these soft processors cannot match the performance of a hard processor, soft processors have the advantage that the designers can implement the exact number of processors to efficiently fit the application requirements.

Current commercial soft processors such as Altera's Nios II [1] and Xilinx's Microblaze [2] are in-order pipelines with five to six pipeline stages. These processors are often used for less computation-intensive applications, for instance, system control tasks. To support more compute-intensive applications, a key technique to improve performance is branch prediction. Branch prediction has been extensively studies, mostly in the context of application specific custom logic (ASIC) implementations. However, naïvely porting ASIC-based branch predictors to FPGAs results in slow and/or resource-inefficient implementations since the tradeoffs are different for reconfigurable logic. Wu et al. [3] have shown that a branch predictor design for soft processors should balance its prediction accuracy as well as the maximum operating frequency. They proposed an FPGA-friendly minimalistic branch prediction implementation *gRselect* for Altera's highest performing soft-processor Nios II-f.

Wu et al. limits the hardware budget of the gRselect predictor to one M9K Block RAM [4] on Altera Stratix IV devices, which is the same hardware budget as Nios II-f. Such a small hardware budget prohibits more elaborated and accurate branch prediction schemes such as perceptron [5] and TAGE [6]. This work looses the hardware budget constraint and investigates FPGA-friendly implementations of perceptron and TAGE predictors. This work also assumes a pipelined processor implementation representative of Altera's Nios II-f for comparison versus gRselect.

Specifically, this work makes the following contributions: (1) It studies the FPGA implementation of the perceptron predictor and TAGE predictor for branch direction prediction. It shows that comparing the accuracy versus gRselect, perceptron is 1% worse while TAGE is 1% better, assuming these predictors can be accessed in a single cycle.

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

II. Perceptron Predictor

Section **??** introduced that perceptron predictor maintains vectors of weights in a table. It produces a prediction through the following steps: (1) a vector of weight (i.e., a perceptron) is loaded from the table. (2) multiply the weights with their corresponding global history (1 for taken and -1 for not-taken). (3) sum up all the products, predict taken if the sum is positive, and not-taken otherwise. Each of these steps poses difficulties to map to the FPGA platform. The rest of this section addresses these problems.

A. Perceptron Table Organization

Each weight in a perceptron is typically 8-bit wide, and perceptron predictors usually use at least 12-bit global history [5]. The depth of the table, on the other hand, tends to be relatively shallower (e.g. 64 entries for 1KB hardware budget). This requires a very wide but shallow memory, which does not map well to BRAMs on FPGAs. For example, the widest configuration that a M9K BRAM on Altera Stratix IV chip is 36-bit wide times 1k entries. If we implement the 1KB perceptron from [5], which uses 96-bit wide perceptrons 12-bit global history

As Fig. **??** shows, 64-entry

B. Multiplication

C. Adder Tree

III. TAgged GEometric history length Branch Predictor (TAGE)

IV. Conclusion

The conclusion goes here.

References

[1] Altera Corp., "Nios II Processor Reference Handbook v9.0," 2009.

[2] *MicroBlaze Processor Reference Guide*, Xilinx Inc., July 2012.

[3] D. Wu, K. Aasaraai, and A. Moshovos, "Low-cost, high-performance branch predictors for soft processors," in *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*, Sept 2013.

[4] *TriMatrix Embedded Memory Blocks in Stratix IV Devices*, Altera Corp, Dec. 2011.

[5] D. A. Jimenez and C. Lin, "Dynamic Branch Prediction with Perceptrons," in *Intl' Symposium on High-Performance Computer Architecture*, January 2001.

[6] A. Seznec and P. Michaud, "A case for (partially) tagged geometric history length predictors," in *Journal of Instruction Level Parallelism (http://www.jilp.org/vol7)*, April 2006.