CMP SCI 635: Modern Computer Architecture

An Evaluation of the TRIPS Computer System

Name: Kunjal Panchal

Student ID: 32126469

Date: 17th Sept, 2019

Paper: 4 – Experience [Gebhart09]

Strengths:

1. EDGE model preserves sequential memory semantics and exposes greater instruction level concurrency without requiring explicit software parallelization. Which is huge from a programmer's perspective as one of the big challenges in programming is to structure your code in a way that exploits the underlying computing architecture in an optimal way.

2. The block-atomic execution model is efficient as it logically fetches, executes, and commits each block as a single entity. Blocks amortize per-instruction bookkeeping and reduce branch predictions, providing latency tolerance to make distributed execution practical. Blocks communicate through registers and memory. Within a block, direct instruction communication delivers results from producer to consumer instructions in dataflow fashion. This supports distributed execution by eliminating accesses to a shared register file.

Weaknesses:

1. As mentioned in the Raw paper [Agarwal04], we need a new ISA for TRIPS too. There are already many established ones and telling programmers to learn and code in new ISA in the competing environment is infeasible and less likely to happen. Even if the new ISA [EDGE in this case is better], the legacy code management for the programs written in assembly e.g. for some embedded computing device, will be much harder with migrating to new environment. Even with a layer of compilation to convert one language code to TRIPS ISA, we are talking about extra overhead.

2. TRIPS didn't perform up to the expectations with single-threaded benchmarks available in market. It performed well on hand-optimized code, which might be biased towards exploiting TRIPS because of its internal structure being known. If it cannot glean out the concurrency possibilities out of a sequential commercial code, how good it will perform with other common commercial applications, which can profit from parallelism?

3. EDGE ISA requires more code space and space to store a graph structure.

Questions/Assertions:

1. "*…TRIPS use an ASIC technology and lacks some hardware needed for an operating system.*" What does author mean by this?

2. What is the utility of "*lessons learned block predictor*"?

3. A wide question maybe, but are micro-coded and hard-coded architectures the one and the same? A micro-coded architecture should simplify the design of each instruction, is it more difficult to design a full hard-coded processor? Or maybe hard-coded architecture is suitable for microcontrollers or smaller design?

4. Does having a large L1 cache instead of L1 and L2 cache makes computation faster? If L2 cache is accessed in parallel with L1 but with higher latency, is it an L2 cache or part of a non-uniform cache architecture (NUCA) L1, especially if some cache blocks are never allocated to the smaller portion of the cache? Is that why NUCA allows transfers between slow and fast portions of L2?