**CS7290 in-class Quiz-I**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Submission guide:* you submit a written version at the class and then submit an individual version and a group discussion version all together by next week’s class time at T-square (only soft copy is accepted). Please indicate which version is yours and which version is after group discussion and also include your discussion members’ names. In class no open book. After class, you can use books/papers/etc. and include citations if that’s necessary. Do not exceed more than 1 page (total 4 questions) for each version.

*Grading algorithm:* in-class submission (10%) + individual submission (30%) + group discussion submission: 60% (no group size limit but it will affect the criteria for grading)

1 [Trace cache]

Trace cache can provide benefits in various aspects but it can also increase power consumption and complexity. Let’s discuss pros and cons of trace cache.

(a) Discuss pros and cons of instruction trace cache in terms of power. Can trace cache increase or decrease power consumption? Do we need I-cache with a trace cache?

(b) Discuss pros and cons of trace caches over compiler optimizations. Please show examples of code optimization effects in trace cache.

(c) Discuss pros and cons of trace caches and branch predictors. Is it a good idea to turn off branch predictor?

(d) Discuss the benefits of trace caches in terms of pipeline depth and width.

Source: From 2012 Fall PhD qualifying exam questions

2. [Dataflow Architectures and Block Structured ISAs]

Static dataflow architectures **classic** (MIT Tagged-Token) or **modern** (TRIPS) are alternatives to von Neumann architectures. Please answer the following questions to highlight the main bottlenecks, benefits, and progress of dataflow machines. Furthermore, for the last question please discuss the prospect of dataflow machines in the near-threshold voltage design domain. (Please note that no prior knowledge on NTV or 3D stacking is required to answer the last question.)

1. What are the main differences between the MIT Tagged-Token and TRIPS architectures?
2. How do dataflow machines deal with control flow? Is it possible to perform control flow speculation in classic (MIT Tagged-Token) and modern dataflow (TRIPS) machines? Is control flow speculation an advantage for out-of-order machines?
3. Which one is more energy-efficient, static dataflow execution or dynamic out-of-order execution? (Hint: remember that one machine is dataflow and the other is von Neumann.)
4. Static dataflow architectures try to maximize the operation (instruction)-level parallelism (ILP). What are the main limiting factors in extracting ILP in these machines?
5. When voltage is scaled to near threshold values, it has been shown that caches are 2× faster than processors. Imagine a system (combination of NTV and 3D stacking) that memory can be accessed in matter of cycle. In this system, energy is the main constraint. Which microarchitecture design paradigm is more suitable for this system, dataflow, in-order, out-of-order? Please discuss.

From 2013 PhD qual question