

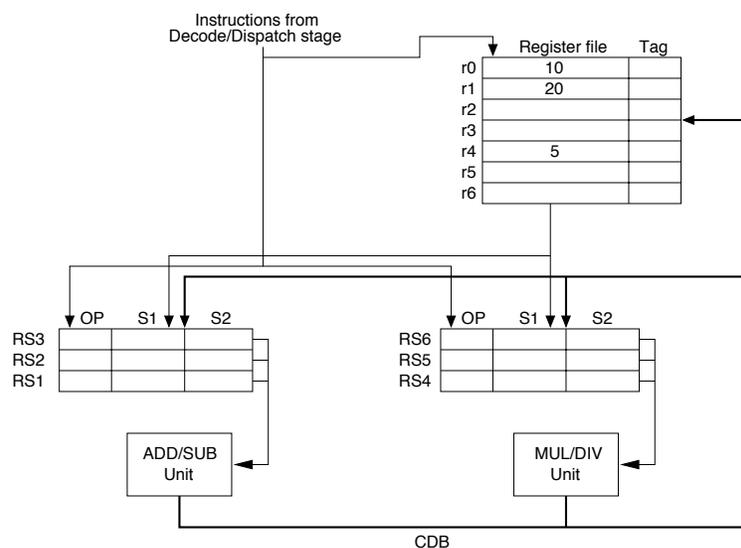
Exam Advances in Computer Architecture

Thursday 24 October 2013
D1.113, 9.00 am - 12.00 am

It is not allowed to use any literature (books, papers, etc.) or lecture notes during the exam. Provide an explanation with all your answers. Good luck!

- 1a (8%)** You, as a processor architect, need to design a data cache. To this end, you need to determine the block size of the cache: 64 or 128 bytes. After simulations, it seems that a block size of 64 bytes yields a cache hit-ratio of 95%, while 128-byte cache blocks yield a hit-ratio of 98%. Moreover, it appears that 25% of all instructions are memory accesses. A memory access that hits the cache has a latency of 1 cycle, while a cache miss has a latency of 40 cycles to fetch a 64-byte cache block from main-memory and 72 cycles to fetch a 128-byte cache block from memory. All other instructions have an average execution latency of 1.2 clock cycles. Which cache block size, 64 or 128 bytes, gives better performance?
- 1b (6%)** Explain the advantages and disadvantages of a virtually addressed cache as compared to a physically addressed cache.
- 1c (6%)** Explain how virtually indexed, physically tagged caches address the problems as discussed in question 1b. Also describe how these virtually indexed, physically tagged caches introduce a new problem.

Assume the below architecture based on Tomasulo scheduling (with register status):



Further assume that the architecture can dispatch 2 instructions per cycle: one to the reservation station of the ADD/SUB unit and one to the reservation station of the MUL/DIV unit. ADD/SUB instructions take 1 cycle to execute while MUL/DIV instructions take 5 cycles to execute.

2a (10%) Show for the next 4 instructions how they are scheduled/executed:

```
instruction i:    MUL R2, R0, R1    // R2 = R0*R1
instruction i+1:  MUL R3, R2, R0    // R3 = R2*R0
instruction i+2:  ADD R2, R4, R1    // R2 = R4+R1
instruction i+3:  SUB R0, R3, R2    // R0 = R3-R2
```

Draw the status of the reservation stations and the register file during the important cycles.

2b (5%) Describe what false dependencies are and show – by giving an example – how these dependencies are automatically taken care of in Tomasulo scheduling.

2c (5%) Explain how false dependencies are handled in *non-Tomasulo processor architectures* (like those using a scoreboard and/or ROB).

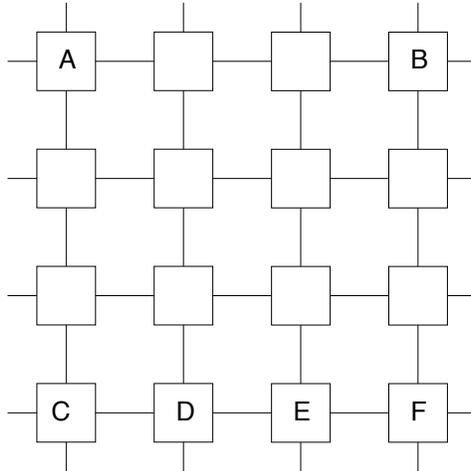
3a (5%) Explain why traditional VLIW processors use static branch prediction instead of dynamic branch prediction.

3b (5%) Explain what *predication* (NOTE: not prediction!) is and describe several advantages and disadvantages of this technique.

3c (10%) Why is code compression important for VLIW processors? Also describe two (2) alternative locations in the architecture where a VLIW processor can perform decompression of instructions (give advantages and disadvantages of both alternatives).

4a (8%) XY routing is a deterministic routing method that avoids deadlocks in networks without physical cycles. So, this does not hold for, for example, torus networks. Show, using a simple example, that XY routing in a torus network may lead to deadlock.

4b (12%) West-first routing is a partial-adaptive routing method that allows more freedom than XY routing. It routes packets first to the west, after which it can adaptively route the packets in any of the three other directions. Given the mesh network below,



- Describe what happens –in the case of XY routing– when nodes C, D, E and F all send a message at the same time to node B. Also explain whether or not there is a difference with the situation in which nodes C, D, E, en F send their messages to node A instead of B.
- Do the same as in the previous question, but now for west-first routing.

5a (10%) Describe two types of implementations of directories in directory-based cache coherency protocols (explain how they work).

5b (10%) The MESI coherency protocol uses 4 states for cache blocks: M(odified), E(xclusive), S(hared) and I(nvalid). Consider a 2-processor multiprocessor system with write-back caches, MESI snoopy-bus coherency and a write-invalidate protocol. Fill in the states for cache block x in the table below (copy the table onto your exam paper). Assume that cache block x initially is not present in either of the caches (x therefore initially does not have any state ('—') in the caches).

Processor action	P1 state	P2 state
P1 write x		
P2 write x		
P1 read x		
P2 read x		

P1 = processor 1, P2 = processor 2