

- (a) (1 Point total) Team Alpha has come up with a CPU that has a clock frequency of 3GHz. All instructions take 1 clock cycle, except the floating point division that takes 13 cycles.
- (0.5 points) Team Beta has come up with an alternative design that takes 1 cycle for all instructions except the floating point division that takes only 8 cycles. However, their CPU runs at a clock frequency of 2GHz. What percentage of the instructions in a program must be floating point divisions for Beta to show better performance than Alpha?
  - (0.5 points) Team Gamma has a design similar to Team Alpha's design, except that their floating point division takes 11 cycles. What percentage of the instructions in a program must be floating point divisions for Beta to show better performance than Gamma?
- (b) (2 Points total) We have a 5-stage pipeline processor {IF, ID, EX, MEM, WB}. Consider the following piece of code:
- ```
I1: SW R16, 10 (R6)
I2: LW R4, 18 (R16)
I3: ADD R5, R4, R4 ; R5 is the destination
I4: OR R1, R2, R3
I5: AND R2, R1, R4
I6: OR R1, R1, R2
```
- (0.6 points) Indicate data dependences and their types.
  - (0.6 points) Assuming that no forwarding is implemented, insert NOP instructions to avoid hazards. How many instruction cycles does it take to run the above code from the IF stage of the first one to the WB stage of the last one?
  - (0.6 points) Assuming that full forwarding is implemented, insert NOPs to eliminate hazards. Full forwarding includes ALU output to the EX stage of the next instruction without hazard, but loads cannot forward to the EX stage of the next instruction. How many instruction cycles does it take to run the above code from the IF stage of the first one to the WB stage of the last one?
  - (0.2 points) If implementing forwarding results in 20% increased cycle time, is it worth it for the code segment discussed in the previous parts?
- (c) (1 point total) Consider a single-level virtual memory system in which the page size is 4KB, physical DRAM is 16GB and the virtual address is 43 bits. Each page table entry (PTE) also has

bits for valid, dirty and protection. Round up PTE sizes to be multiples of bytes for easy indexing.

- i. (0.5 points) How much physical memory is needed for storing the page table?
- ii. (0.2 points) What is the disadvantage of using a single-level virtual memory system in our example?
- iii. (0.3 points) Suppose the architecture team has decided to change the TLB to harness a 2-level fully associative cache. A designer in the team makes the argument that they have to implement the tag search algorithm in hardware because the bottleneck in handling page faults is to find out if a page is missing from the physical memory. Is that argument valid? Explain.