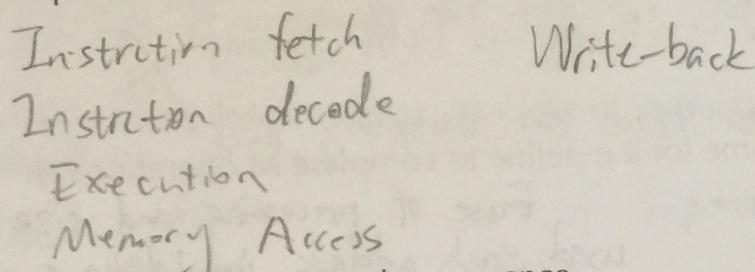


Answer each of the following problems on the paper provided. Neatness and organization will influence the grading of your exam. If you feel information you need to solve a problem is missing from the problem, state that and state the assumptions under which you will solve the problem. If the assumptions are reasonable and correct, the problem will be graded under those assumptions. You may use a calculator. Devices with communication capabilities, e.g. cell phones, smart phones, tablets, and laptops are not calculators. Otherwise, Closed books, closed computers, and closed neighbors.

Carefully read the questions, take your time, think, and then answer.

1. In our simplified MIPS CPU model, what are the five (5) pipeline stages?



2. Consider the following instruction sequence.

		Instruction Index
ADD	R1, R2, R3	(1)
SUB	R4, R5, R1	(2)
AND	R1, R5, R4	(3)
ADD	R8, R9, R10	(4)
XOR	R10, R11, R12	(5)

- (A) Which pair(s) of instructions exhibit a read-after-write hazard? (3) → (1)
 - (B) Which pair(s) of instructions exhibit a write-after-write hazard? (2) → (1)
 - (C) Which instruction(s) could execute at any time (out of order)? (3) (5)
 - (D) Which pairs(s) of instructions exhibit a write-after-read hazard? (2) → (1)
 - (E) Which pair(s) of instructions exhibit a read-after-read hazard? (2) → (1) X
- Your answer for A, B, D and E should look like (2) --> (1). -2

3. A computer architecture implements 32-bit virtual addresses. The page size is 4 KiB (2^{12} B). There are eight (8) different sub-address spaces. How many bits are available for the virtual page number?

$$\frac{2^{32}}{2^{12} \cdot 2^3} = 2^{17} \text{ bits}$$

17 bits.

4. The address trace files we used for the CacheSimulation homework assignment stored 32-bit addresses in binary. Explain why the binary format was used rather than ASCII (text).

binary format is related to caches that have 32-bit. Caches read and write CPU that is 0x86 system. X

ASCII (text) takes time on transferring bits.

5. The conventional time for a pipeline to complete M operations is:

$$Time_B = N_p \cdot t_p + (M-1) \cdot t_p$$

Where:

$Time_B$ is the time to complete M operations.

N_p is the number of pipeline stages.

t_p is the time to execute one stage in the pipeline.

M is the number of operations

Assume t_p is one time unit.

However, as we've seen in the simplified MIPS CPU pipeline model, some instructions cause stalls in the pipeline. Modify the conventional time for a pipeline to complete M operations if 25% of the instructions cause a single t_p stall. That is, 25% of the instructions take two (2) t_p rather than one t_p .

$$Time_B = N_p \cdot \frac{1}{2} t_p + \frac{3}{4} (M-1) \cdot t_p$$

$$Time_B = N_p \cdot t_p + \frac{1}{4} (M-1) \cdot 2t_p + \frac{3}{4} (M-1) \cdot t_p$$

$$Time_B = N_p \cdot t_p + (M-1) \cdot t_p + \frac{M}{4} t_p$$

A small cache for the page table that stores a few direct translation from virtual address to physical address.

6. What is a translation look-aside buffer?

A translator look-aside buffer is a buffer that translates look-aside. X

7. Explain loop unrolling.

The loop of rolling does not work. The loop do not execute in the next time.
Loop unrolling inlines a loop body multiple times to avoid branching back to the beginning of the loop as frequently, which aids pipelining

8. When can a compiler re-arrange the order of instructions to improve performance?

before the next time of executing programs.

Whenever their execution order doesn't affect the final review.

9. Suppose you have the following page table for a process. The virtual page numbers (VPN) are 4-bits and the physical page numbers (PPN) are 8-bits. The page size is 4096 bytes (12-bits).

4-bit	8-bit	12 bits
VPN	PPN	
0x0	0x77	
0x1	0x12	
0x2	0x45	
0x3	0xF3	
0x4	0x87	
0x5	0x99	
0x6	0xA3	
0x7	0x34	
0x8	0xD2	
0x9	0x11	
0xA	0xC3	
0xB	0x77	
0xC	0x55	
0xD	0xB8	
0xE	0xAA	
0xF	0xFF	

0x077
0x127
77

Pagesize. 4096
 2^{12}

Given the following virtual addresses (16-bits) what is the physical address (20-bits) in hexadecimal?

- (A) 0x4777 ==> 0x87343 ~~0x87777~~ ~~0x87343434 / 2^{12}~~
- (B) 0xC345 ==> 0x55F38 ~~0x55345~~ ~~0x55F38799 / 2^{12}~~
- (C) 0x8888 ==> 0xD2D2D2 ~~0xD2888~~ ~~0xD2D2D2D2 / 2^{12}~~