

CANOS Midterm 2 – Take home exam
4/10/2020
12:00 noon – 11:59 pm

Last name	First name	Student ID number	Email address
Ahmed	Saaif	661925946	ahmeds7@rpi.edu

Exam Rules:

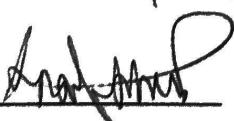
- Exam is open book, open notes.
- Please write your full name and RIN on every page of your submitted work.
- This is an individual assessment, work on it by yourself.
- No collaboration allowed.
- Exam should be submitted on Gradescope on April 10th, 2020 before 11:59 pm.
- If you have a question during the exam, you should email the instructor at hameem2@rpi.edu. Don't post it on Piazza. If it happens to be a common question or typo, the instructor will inform all students through a piazza post.
- Be sure to show your work in each problem.
- If you need to make specific assumptions to answer a question, make sure to state those assumptions explicitly.
- Write and sign honor pledge below before you start working on the exam.

Honor pledge for this exam:

"I pledge on my honor that I will not give or receive any unauthorized assistance on this exam."

Please hand write the pledge on the first page of submitted work and provide your signature.

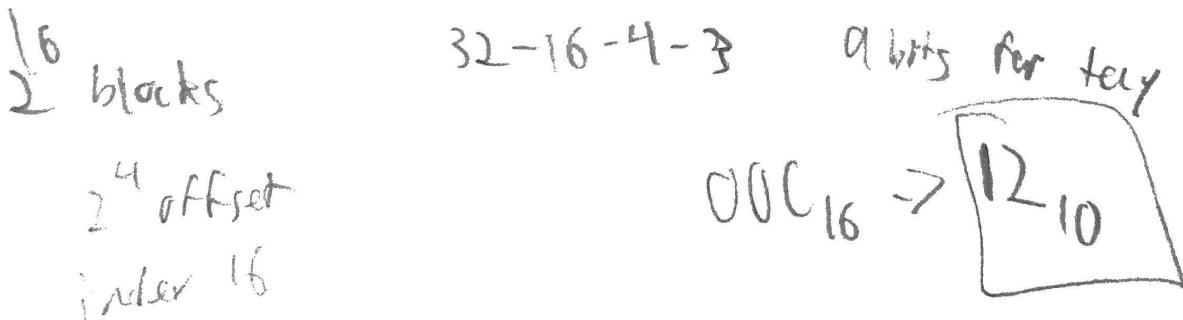
I pledge on my honor that I will not give or
receive any unauthorized assistance on this exam.

Signature: 

1. (20 points) Cache Systems

A computer has a 32-bit physical address space, and it is byte addressed. It has a direct-mapped 1MB cache with 32-byte blocks. The cache is mapped to physical addresses.

- a) (2 points) How many bits are required for the block offset? 5 bits
- b) (2 points) How many blocks does the cache hold? $2^{20} / 2^5 = 2^{15} \text{ blocks}$
- c) (2 points) How many bits are required for the cache index? $15 \text{ bits for index}$
- d) (3 points) How many bits are required for each tag? 12 bits per tag
- e) (2 points) How many total bits of storage are required to build the cache? Assume that each cache entry has 1 valid bit, and 1 dirty bit. $32 \text{ bits needed to build cache}$
- f) (2 points) What is the block offset for memory address FACE B00C₁₆?
 $5 \text{ bits: } 0C \rightarrow 12_{10}$
- g) (2 points) What is the cache index for address FACE B00C₁₆?
 $11110101100111_2 = 32103_{10}$
- h) (2 points) What is the tag for address FACE B00C₁₆?
 $010110000000_2 = 1408_{10}$
- i) (3 points) If the 1MB cache were organized as a 8-way set-associative cache of 16-byte blocks, what would the tag be for address FACE B00C₁₆?



2. (20 points) Page Replacement.

Suppose we have a virtual address space consisting of 5 pages, numbered 1 to 5, and physical memory to hold 4 frames. For each part, assuming no pages are in memory at the start,

- (i) Complete the table, indicating which pages are in memory in the each frame following each reference in the sequence, using the given page replacement algorithm.
- (ii) If a fault occurs, put a "y" in the "fault?" box. If a fault does not occur, put an "n".
- (iii) Finally, state clearly exactly how many page faults you get in each case.

If an algorithm needs to evict a frame, but could choose equally from among multiple frames (i.e., there is a tie according to the algorithm), choose the frame for the page that was referenced least recently.

2.a. (6 points) FIFO page replacement.

page reference	2	1	3	4	1	2	5	1	2	3	4	1
frame 1	2	2	2	2	2	2	5	5	5	5	5	5
frame 2		1	1	1	1	1	1	1	2	2	2	2
frame 3			3	3	3	3	3	3	3	3	3	1
frame 4				4	4	7	4	4	4	4	4	4
fault?	Y	Y	Y	Y	N	N	Y	N	Y	N	N	Y

7 page faults

2.b. (7 points) LRU page replacement.

page reference	2	1	3	4	1	2	5	1	2	3	4	1
frame 1	2	2	2	2	2	2	2	2	2	2	2	2
frame 2		1	1	1	1	1	1	1	1	1	1	1
frame 3		3	3	3	3	5	5	5	5	4	4	
frame 4			4	4	4	4	4	4	4	3	3	
fault?	Y	Y	Y	Y	N	N	Y	N	N	Y	Y	N

7 page faults

2.c. (7 points) OPT (optimal) page replacement.

page reference	2	1	3	4	1	2	5	1	2	3	4	1
frame 1	2	2	2	2	2	2	2	2	2	2	2	2
frame 2		1	1	1	1	1	1	1	1	1	1	1
frame 3		3	3	3	3	3	3	3	3	3	3	3
frame 4			4	4	4	5	5	5	5	4	4	
fault?	Y	Y	Y	Y	N	N	Y	N	N	Y	Y	N

6 page faults

3. (20 points) Cache and paging performance

Consider the 800MHz Shmegeggie-800 processor. It has one-level paging with the page table always in main memory. 70% of the memory accesses are for instructions, with a page-fault rate of 2×10^{-8} , 30% for data with a page-fault rate of 10^{-7} . The Translation Look-aside Buffer is checked before the cache, and if it misses, the cache is not checked before accessing the page table.

Main memory access time	20 ns
TLB access time	5 ns
Ave time to process a page fault	15 ms
Cache access time	8 ns
Cache hit rate	90% for instructions, 85% for data
TLB hit rate	80% for instructions, 70% for data

- a) (7 points) What is the average memory access time for instructions?
- b) (7 points) What is the average memory access time for data?
- c) (2 points) What is the average memory access time?
- d) (2 points) On average, how many CPU cycles does each memory access take?
- e) (2 points) Based on your answer to part 3.d above, is the Shmegeggie-800 processor running at full capacity? Answer "yes" or "no" and explain why.

$$\begin{aligned} & (5\text{ns} + 8\text{ns}) \cdot .9 + (5\text{ns} + 8\text{ns} + 20\text{ns}) \cdot .1 + .2(1 - 2 \times 10^{-8})(40\text{ns} + 5\text{ns}) + .2(2 \times 10^{-8})(15\text{ms}) \\ & = 2.106 \times 10^{-8}\text{s} \end{aligned}$$

$$\begin{aligned} & (5\text{ns} + 8\text{ns}) \cdot .7 \cdot .85 + (5\text{ns} + 8\text{ns} + 20\text{ns}) \cdot .7 \cdot (.5 + .3(1 - 10^{-7})(40\text{ns} + 5\text{ns}) + .3(10^{-7})(15\text{ms})) \\ & = 2.515 \times 10^{-8}\text{s} \end{aligned}$$

$$C: 2.106 \times 10^{-8} \cdot .7 + 2.515 \times 10^{-8} \cdot .3 = 2.23 \times 10^{-8}\text{s}$$

$$D: \frac{2.23 \times 10^{-8}\text{s}}{1 \text{ memory access}} \cdot \frac{800 \times 10^6 \text{ cycles}}{\text{second}} = 17.83 \text{ CPU cycles}$$

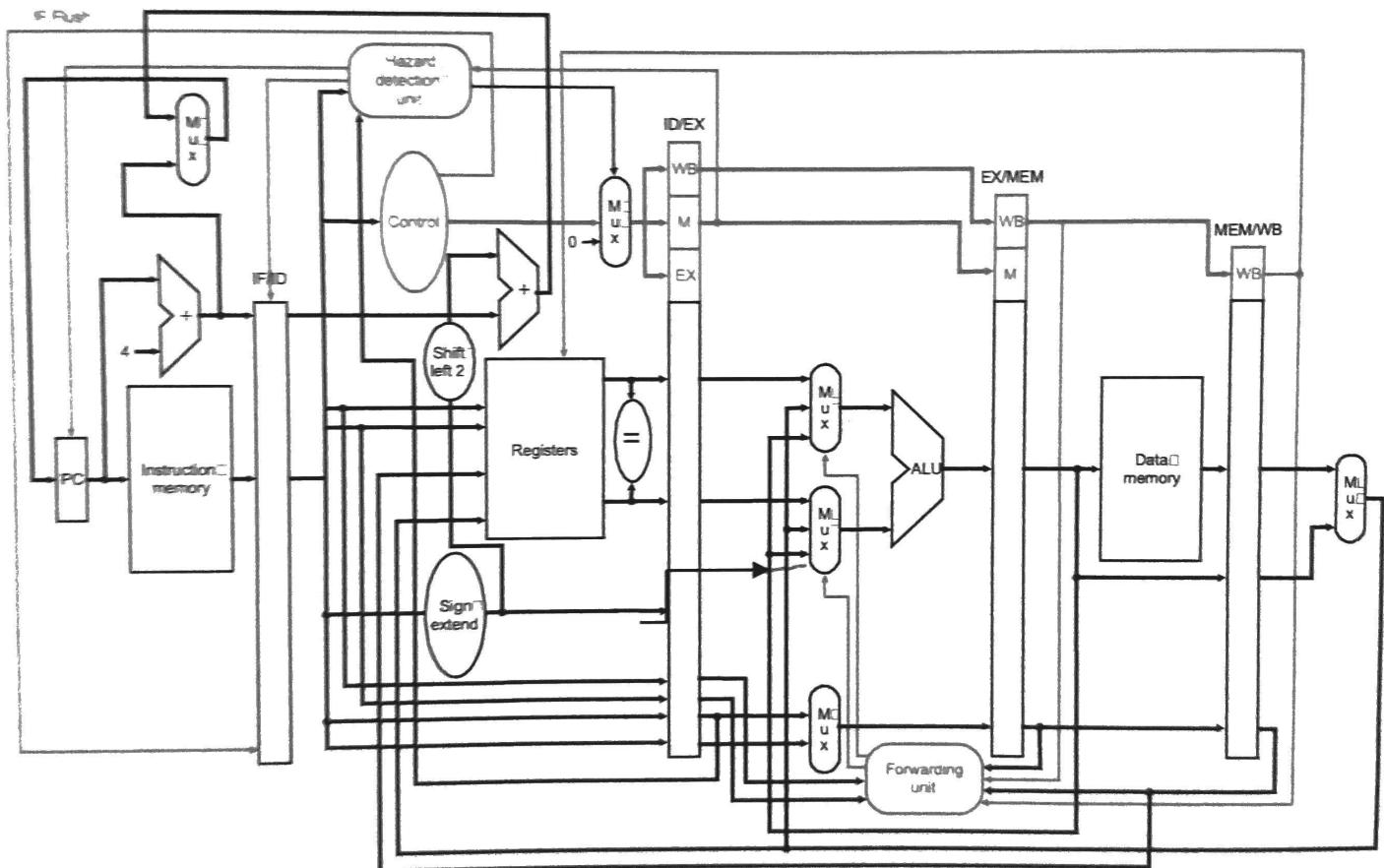
E: Yes it is. It takes more than 1 cycle to do 2 memory access. Which means the CPU is highly occupied.

4. (20 points) Pipelined Processor Design

Figure 4.65, page 325, from Patterson and Hennessy textbook is reproduced below.

Note that there is a typo in the textbook. The labels for the EX/MEM and MEM/WB inter-stage pipeline registers are incorrect. The labels need to be reversed, as indicated by the diagram shown below; EX/MEM first and then MEM/WB.

Suppose that the instruction set is limited to {lw, sw, add, sub, and, or, beq, slt}.



CHECK FIGURE

Also note that this figure is missing some parts such as ALsrc MUX from figure 4.57 and the multiplexor inputs from figure 4.51, as pointed out in the textbook.

The following code fragment is executed on this pipeline.

```

add $3, $2, $4
lw $2, 298($0)
or $1, $2, $3
sub $3, $1, $6

```

a) (5 points) Fill up the following table:

Instruction	Rd	Rs	Rt	Immediate operand (in hex)
add \$3, \$2, \$4	\$3	\$2	\$4	0
lw \$2, 298(\$0)		\$0	\$2	23
or \$1, \$2, \$3	\$1	\$2	\$3	0
Sub \$3, \$1, \$6	\$3	\$1	\$6	0

b) (7 points) By sketching a pipeline execution diagram, similar to figures 4.28 and 4.29 on page 279 of textbook, find out how many cycles it takes to execute this code on the above hardware. Explain any hazards, and how to deal with them.

c) (8 points) For the same code fragment, list the values of the following control signals. Assume that the MUX inputs are numbered from top to bottom, starting with 0.

Instruction	Top MUX in EX stage	Middle MUX in EX stage	Bottom MUX in EX stage
add \$3, \$2, \$4	0	0	0
lw \$2, 298(\$0)	3	3	0
or \$1, \$2, \$3	0	3	1
sub \$3, \$1, \$6	3	0	1

B: There is hazard detection and forwarding. Line 3 has a conflict with line 2 because of LW. So you add NOP. This is the only hazard because there is forwarding implemented. Thus it takes 5 cycles.

5. (20 points) Pipelined Processor Performance

Consider the following specifications for a processor implementing the reduced MIPS instruction set {lw, sw, add, sub, and, or, beq, j}:

Instruction usage:	
Instruction	Usage Frequency
lw	25%
sw	15%
add, sub, and, or	40%
beq	15%
j	5%

Speed of datapath components:	
Instruction Fetch	10ns
Instruction Decode/Register Read	2ns
ALU operation	6ns
Data Access	10ns
Register Write	2ns
All others	Negligible

- a) (5 points) What is the clock cycle time in a non-pipelined processor?

30 ns

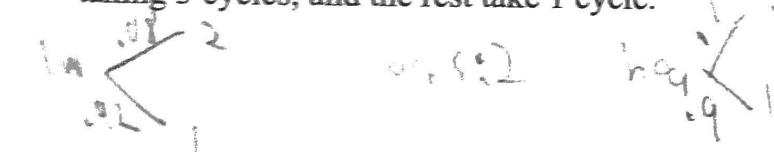
- b) (8 points) What is the CPI of a non-pipelined processor?

10 ns

- c) (5 points) What is the clock cycle time in a 5-stage pipelined processor?

t_{est} = 1

- d) (12 points) What is the CPI of a 5-stage pipelined processor? Assume: 8% of lw instructions encounter a load-use data hazard, requiring a 1-cycle stall; Unconditional jumps (j) take 2 cycles on average; 10% of conditional branches (beq) need to have the pipeline emptied, taking 3 cycles, and the rest take 1 cycle.



$$.25 \cdot .01 \cdot 2 + .25 \cdot .62 \cdot 1 + .15 \cdot 1 + .4 \cdot 1 + (.15 \cdot 1 \cdot 3 + .15 \cdot .9 \cdot 1 + .05 \cdot 2)$$

$$= 1.1 \text{ CPI}$$