

# CANOS Final Exam

5/7/2020

12:00 noon – 11:59 pm

Last name	First name	Student ID number	Email address
Ahmed	Saaif	661925946	ahmed.s7@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 May 7<sup>th</sup>, 2020 before 11:59 pm.
- If you have a question during the exam, you should email the instructor at [hameem2@rpi.edu](mailto: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: Ahmed Saaif

Please use separate blank sheets of paper to answer questions. Not enough space here.

## 1. (10 points) CPU Performance

You are the lead designer of a new processor. The processor design is complete, and now you must decide whether to produce the current design as it stands or spend additional time to improve it. You have the following options:

**Option MBASE:** Leave the design as it stands. Call this the base machine, MBASE. It has a clock rate of 1.6GHz and the following characteristics:

Instruction type	CPI	Frequency of use
A	2	30%
B	3	40%
C	4	20%
D	5	10%

**Option MOPT:** The hardware team claims they can improve the processor design to give a clock rate of 2.5GHz. Call this machine MOPT. The changes cause the CPI of some instructions to change.

Instruction type	CPI	Frequency of use
A	4	30%
B	4	40%
C	4	20%
D	5	10%

- a. (4 points) What are the CPIs of MBASE and MOPT?

$$CPI_{MBASE} = 2(.3) + 3(.4) + 4(.2) + 5(.1) = \boxed{3.1}$$

$$CPI_{MOPT} = 4(.3) + 4(.4) + 4(.2) + 5(.1) = \boxed{4.1}$$

Please use separate blank sheets of paper to answer questions. Not enough space here.

- b. (4 points) What are the MIPS ratings of MBASE and MOPT?

$$\text{MIPS}_{\text{MBASE}} = \frac{\text{instructions}}{\text{clock}} \cdot \frac{\text{cycles}}{\text{instruction}} = \frac{3.1}{4.1} \cdot \frac{1.6 \times 10^9}{1} \cdot \frac{1}{10^9} = 516.13 \text{ MIPS}$$

$$\text{MIPS}_{\text{MOPT}} = \frac{1}{4.1} \cdot \frac{2.5 \times 10^9}{1} \cdot \frac{1}{10^9} = 609.75 \text{ MIPS}$$

- c. (2 points) How much faster is MOPT compared to MBASE?

MOPT is 18.14% faster than MBASE

$$\frac{609.75 - 516.13}{516.13} \times 100 = 18.14\%$$

## 2. (10 points) Deadlock Detection and Resource Allocation Graph

Consider a system with five processes P0 through P4 and three resource types A, B, C. Resource type A has 10 instances, B has 5 instances and type C has 7 instances. Suppose at time  $t_0$  the following snapshot of the system has been taken:

Process	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	7	4	3	3	2
P1	2	0	0	3	1	2	2	3	4
P2	3	0	2	9	6	0	2	3	1
P3	2	1	1	2	0	2	1	4	5
P4	0	0	2	4	4	3	3	5	5

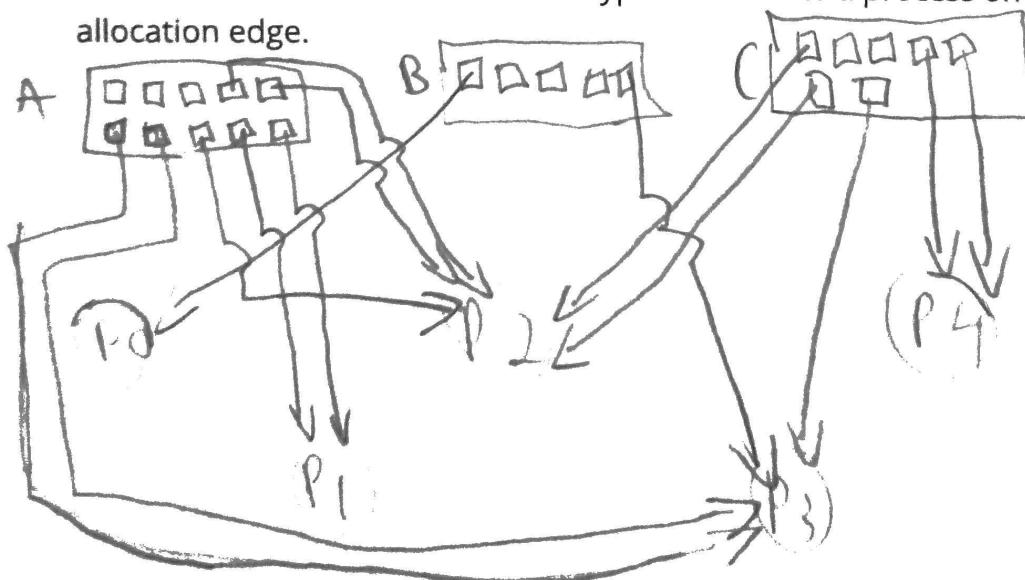
- a. (6 points) Determine whether the system has a Deadlock or is it safe. If it is safe, determine a safe process sequence. If there is Deadlock present, propose the Resource type instances that must be increased and by how much to prevent Deadlock.

Hint: Request[i,j] = Max[i,j] - Allocation[i,j]

A: no deadlock

every thing  
else can  
be done

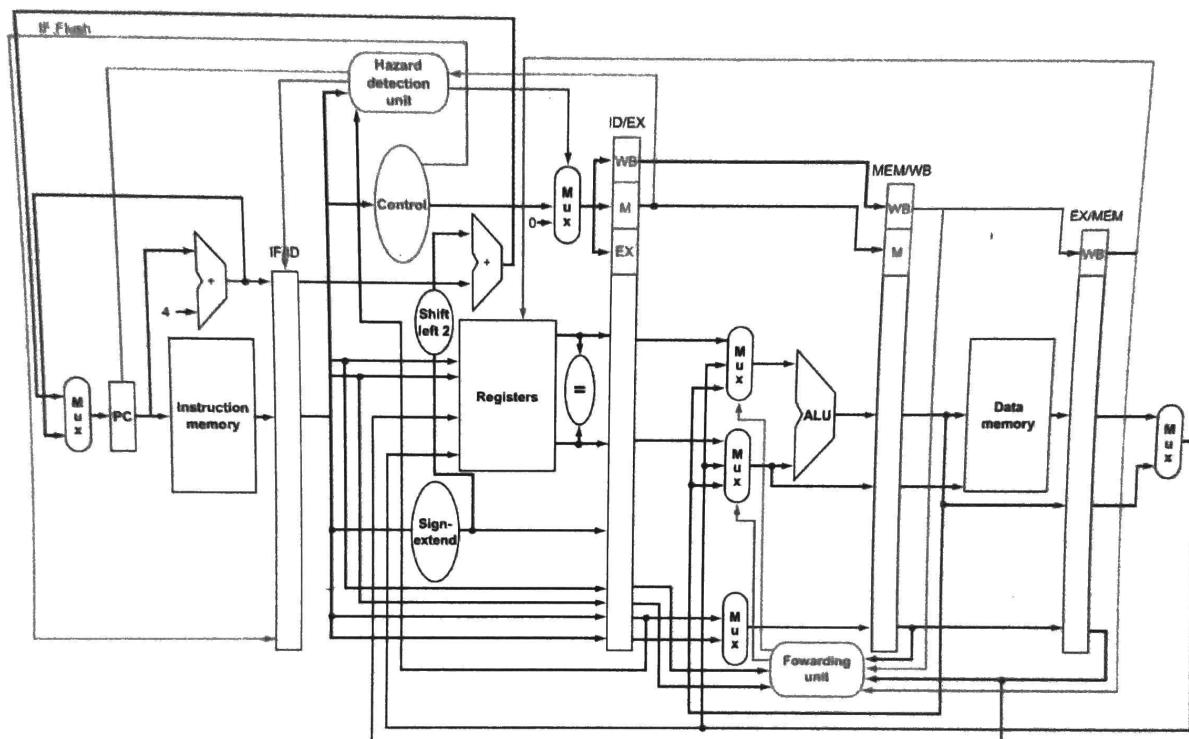
- b. (4 points) Sketch the resource allocation graph of the system at time  $t_0$ . You only need to show the allocated resources. You may choose to write the number of instances of a resource type allocated to a process on the allocation edge.



Please use separate blank sheets of paper to answer questions. Not enough space here.

### 3. (15 points) Datapath & Control, MIPS

Consider a machine with the datapath and control as in Figure 4.65 from P&H textbook, which is reproduced below. Notice that the machine has Forwarding and Hazard Detection Units, as well as necessary components to perform dynamic branch prediction. Assume that data and instruction memory can be read and written at the same cycle.



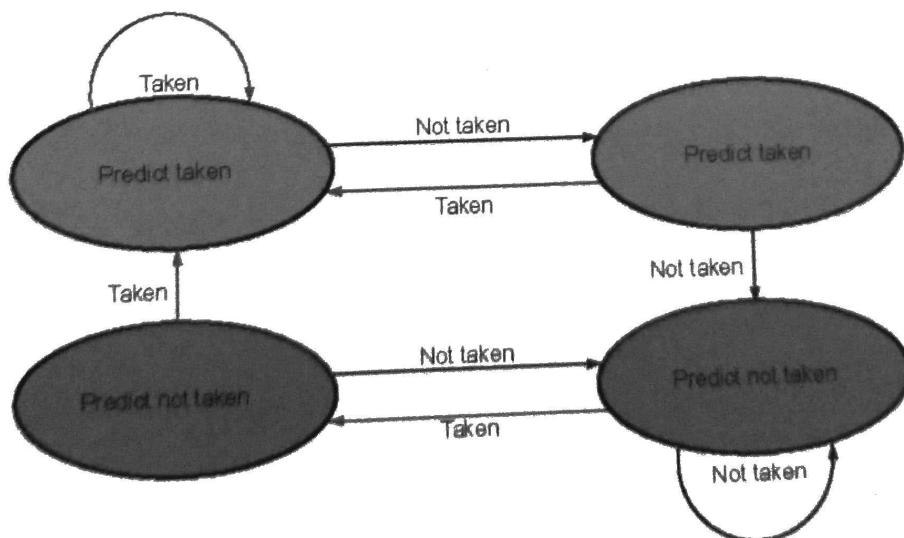
The last seven lines of a program are given as follows:

Instruction Address	MIPS Instruction
...	...
0x0000 011C	L1: add \$t0, \$zero, \$zero
0x0000 0120	L2: lw \$t1, 300(\$zero)
0x0000 0124	L3: subi \$t1, \$t1, 1
0x0000 0128	L4: lw \$t2, 304(\$t1)
0x0000 012C	L5: add \$t0, \$t0, \$t1
0x0000 0130	L6: bne \$t1, \$zero, L3
0x0000 0134	L7: add \$v0, \$t0, \$zero

Please use separate blank sheets of paper to answer questions. Not enough space here.

We also know that, right before the line L1, an integer variable **n=100** is placed at memory location **0x300** and an array **A[ ]** is placed in memory starting at location **0x304**.

- a. (2 points) Show the instruction at line L6 in hexadecimal format.
- b. (3 points) For the last seven lines of the program above, indicate the dependencies among the instructions by using the instruction labels (L1 to L7). If it is a hazard, identify the type of hazard and (if applicable) the register associated with the hazard.
- c. (3 points) What does this last seven lines of the program do?
- d. (3 points) Assuming that the datapath has no dynamic branch prediction hardware and always assumes branches to be not taken, indicate the execution time of the last seven lines of the program above in terms of clock cycles.
- e. (4 points) Assuming that the dynamic branch prediction technique shown in figure below is used, indicate the execution time of the above code in terms of clock cycles. Assume that dynamic branch prediction starts as "predict taken" as in upper left state of the state diagram below.



3 A: bne: 000101 → | S  
t<sub>1</sub>: 01001 → 2  
zero?: 0000 → 0  
offset: 0000000100100100 → 0124

0x15200124

B: L7 depends on L1, L5  
L5 depends on data hazard EX/MEM reg (L5)

L5 depends on L3, L1

data hazard EX/Mem reg (L3)

L4 depends on L2, L3

data hazard Mem/WB reg (L3)

data hazard EX/MEM reg (L2)

L3 depends on L2

data hazard EX/MEM reg (L2)

C2 From a counter stored in t1 we increment t0 by t1, t1 times. While this is happening t2 will have the 304th offset of t1 at each step of the way. Then t0 is stored into V0 and the program ends.

D: U: 5, L2: 6, L5 → L6 is 100 loops  
4 times 400 clocks → L2 → +

total clock cycles: 407

E: same as above + 2 cycle penalty for prediction on the first time, 2 cycle penalty for being in not taken state.

total clock cycles: 411

Please use separate blank sheets of paper to answer questions. Not enough space here.

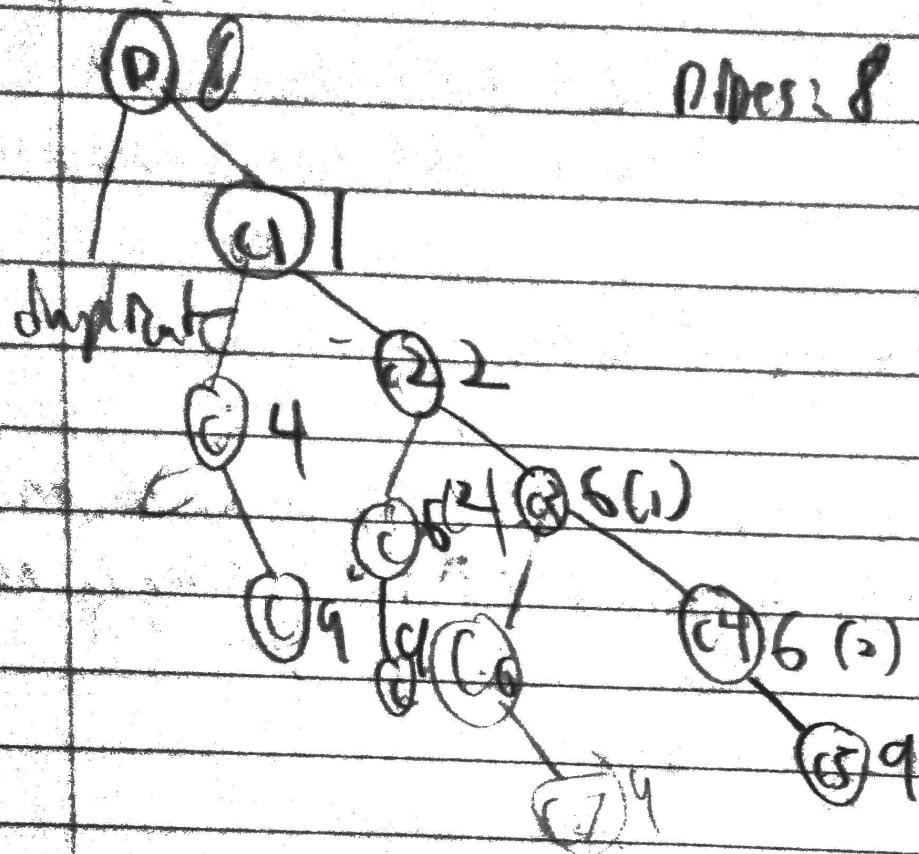
#### 4. (10 points) Process creation and communication

Given the code below, assume that all the fork/pipe operations succeed, and neglect any syntax errors you might identify.

```
int main( )
{
    1   fork();
    2   int pid=fork();
    3   if (pid > 0)
        4       fork();
    5   else
        6       {fork(); fork();}
    7   int pipefd[2];
    8   pipe(pipefd);
    9   pid = fork();
   10  if (pid==0) { int x = 16;
   11      write(pipefd[1], x, sizeof(x));
   12      x = 32;
   13      write(pipefd[1], x, sizeof(x));
   14      exit(0);
   15  }
   16  else { int y1, y2;
   17      read(pipefd[0], y1, sizeof(y1));
   18      read(pipefd[0], y2, sizeof(y2));
   19      y1 = y1 + y2;
   20      printf("PARENT: value = %d", y1; /* LINE A */
   21      exit(0);
   22  }
}
```

- a. (3 points) Including the initial parent process, how many processes are created?
- b. (3 points) How many pipes are created?
- c. (4 points) Explain what the output will be at Line A.

4:



A = 22 processes created

B = 14 pipes created

C = The output at Line A will be the  
number of pipes  $\times 32 \times 2 = 896$

## 5. (10 points) Process Scheduling

Consider the following processes:

Process	CPU Burst Time (ms)	Arrival Time (ms)
P1	2	0
P2	7	1.9
P3	5	5.1
P4	3	6.2
P5	2	7.3

- a. (3 points) Draw a Gantt chart showing the execution of these processes using Round Robin (RR) scheduling with a **quantum of 2 ms**. Assume that the currently executing process is at the head of a circular queue, and a newly arriving process joins the tail of the queue. The top row shows the time (in ms) at the beginning of the interval. Use columns as needed.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
-	-	P1	-	P2	P2	P3	P2	P4	P1	P5	P3	P2	P4	P3							

- b. (3 points) Now draw a Gantt chart showing the execution of these same processes using Shortest Remaining Time First (SRTF). Assume all changes occur exactly on the (next) millisecond. Use columns as needed.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
		P1					P2		P5				P4					P3			

A5

- S. t=0 P1 2  
t=1.9 P1 0.1 P2 7  
t=2 P2 7  
t=4 P2 5  
t=5.1 P2 3.9 P3 5  
t=6 P3 5 P2 3  
t=6.2 P3 4.8 P2 3 P4 3  
t=7.3 P3 3.7 P2 3 P4 3 P5 2  
t=8 P2 3 P4 3 P5 2 P3 3  
t=10 P4 3 P5 2 P3 3 P2 1  
t=12 P5 2 P3 3 P2 1 P4 1  
t=14 P3 3 P2 1 P4 1  
t=15 P2 1 P4 1 P3 1  
t=17 P4 1 P3 1  
t=18 P3 1  
t=19 —

SB:  $t=0$  P12

$t=1.4$  P1.1 P27

$t=2$  P27

$t=5.1$  P23.9 P35

$t=6.2$  P22.8 P35 P43

$t=7.3$  P21.7 P35 P43 P52

$t=9$  P52 P35 P43

$t=11$  P43 P35

$t=14$  P35

$t=19$  -

Please use separate blank sheets of paper to answer questions. Not enough space here.

- c. (4 points) The first two columns of the following table show measured (i.e., actual) CPU burst times (in milliseconds) of a process. Use exponential averaging with  $\alpha = 0.7$  to predict the next burst time  $\tau_6$  assuming that the prediction at 0 ( $\tau_0$ ) is 1ms. Write your answers in a table as shown below.

Burst Number	Measured Burst Time (ms)	Predicted Burst Time Using $\alpha = 0.7$
0	4	1ms
1	2	3.1ms
2	3	2.33ms
3	7	2.8ms
4	5	5.74ms
5	9	5.22ms
6		7.867ms

$$\tau_{n+1} = \alpha \tau_n + (1-\alpha) T_n$$

$$\alpha = 0.7$$

$$T_0 = 1\text{ms}$$

Please use separate blank sheets of paper to answer questions. Not enough space here.

## 6. (15 points) MIPS Code Optimization

Consider the following MIPS code.

20  $x \leftarrow L$

```
addi $t5, $0, 5 ] 2
addi $t6, $t5, 20
LOOP: beq $t5, $t6, END 5, 25  $(x+a) + (x-a)$ 
      lw $s2, 0($s0)
      lw $s3, 0($s1)
      add $t1, $s2, $s3
      sub $t2, $s2, $s3
      add $t3, $t1, $t2 $10
      add $t4, $t4, $t3
      add $t4, $t4, $s3
      addi $t5, $t5, 1
      addi $s0, $s0, 4
      addi $s1, $s1, 4
      j LOOP
END:   add $t1, $t4, $t4 ] 1
```

25  $2t$

add \$t1, \$t2, \$s2

add \$t3, \$t4, \$t1

- (3 points) Explain what this MIPS code does.
- (4 points) How many instructions does this code execute?
- (4 points) Rewrite code above to minimize the number of MIPS instructions executed.
- (4 points) How many instructions does your new code execute?

A: A loop with bounds 5 to 25 is set. The two words are loaded into the registers. t1 stores the addition of the two words and t2 stores the difference. t3 stores the sum of those two previous processes. Then t4 is increased by t3 and the second loaded word. Finally the counter increases and the offsets move to the next word. At the end  $2 * t4$  is stored in t1.

$$B: 2 + 20(12) + 1 + 1 = 244 \text{ instructions}$$

6 C: addi \$t5, \$0, 5

addi \$t6, \$t5, 20

Loop: beq \$t5, \$t6, END

lw \$t2, 0(\$t0)

lw \$t3, 0(\$t1)

add \$t1, \$t2, \$t2

add \$t3, \$t1, \$0

add \$t3, \$t3, \$t3

addi \$t5, \$t5, 1

addi \$t0, \$t0, 4

addi \$t1, \$t1, 4

; LOOP

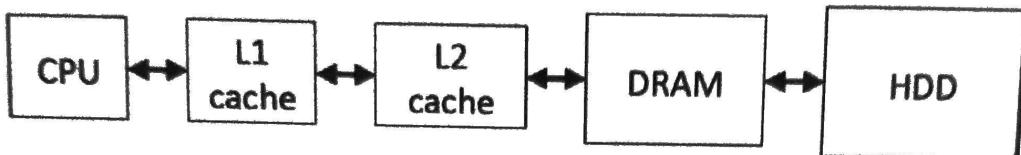
END: add \$t1, \$t3 \$t3

D:  $2 + 20(10) + 2 = 224$  instructions

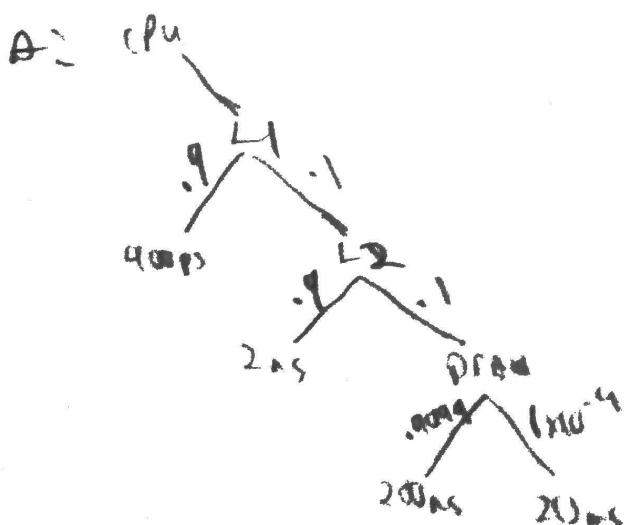
Please use separate blank sheets of paper to answer questions. Not enough space here.

## 7. (15 points) Memory Hierarchy and Access Time

A computer has the memory hierarchy as shown below. Let  $c_1$  and  $c_2$  denote the access time of L1 and L2 cache, and  $h_1$  and  $h_2$  denote the hit rate of L1 and L2 cache. Let  $c_m$  denote the access time of DRAM, and  $f_m$  denote the page fault rate of DRAM. Let  $c_{hdd}$  denote the access time of the 1TB HDD.



- a. (5 points) Suppose  $c_1 = 400\text{ps}$ ,  $c_2 = 2\text{ns}$ ,  $h_1 = 90\%$ ,  $h_2 = 90\%$ ,  $c_m = 200\text{ns}$ ,  $f_m = 0.01\%$ , and  $c_{hdd} = 20\text{ms}$ . Calculate the average access time of the overall memory system.
- b. (10 points) In order to reduce the average access time with the budget of \$1000.00, you have two options: (1) upgrade the DRAM to reduce  $c_m$  from 200ns to 100ns, and reduce  $f_m$  from 0.01% to 0.001%; (2) replace the 1TB HDD with a 1TB SSD whose access time is 200us. Which option is better? Explain.



$$0.9(400 \text{ ps}) + 0.1 \cdot 0.9(2 \text{ ns}) + 0.1 \cdot 0.9(2 \text{ ns} + 20 \text{ ms}) + 0.1 \cdot 0.1(20 \text{ ms}) = [2.25 \text{ ms}]$$

$$B = 1^2 + .00099(10^{-3}) + 1^2 \cdot 10^{-5} (200 \times 10^{-3}) = 3 \times 10^{-4}$$

vs

$$1^2 + .9999(200 \times 10^{-4}) + 1^2 \cdot 10^{-4} (200 \times 10^{-6}) = 2.2 \times 10^{-4}$$

Upgrading the to an SSD is the best because  
it's effect on the access time is the smallest.  
mean

Please use separate blank sheets of paper to answer questions. Not enough space here.

## 8. (15 points) Caching and Page Replacement

A TOY 8-bit computer has a 32-byte cache, which is completely empty before observing the following 8-bit memory access addresses (each 8-bit address is represented in HEX format):

A0, C0, A0, C0, 80, 00, 00, 20, C0, A0

- a. (5 points) If the 32-byte cache is direct mapped with one-byte blocks, how many cache hits will occur? You may find it useful to fill the table below first.

Number of cache hits = 5 hits

everything can be loaded

Addr	A0	C0	A0	C0	80	00	00	20	C0	A0
Tag										
Index										
Hit/miss	m	m	<u>h</u>	<u>h</u>	m	m	<u>h</u>	m	<u>h</u>	<u>h</u>

- b. (5 points) If the 32-byte cache is two-way set associative with one-byte blocks and uses LRU cache replacement policy, how many cache hits will occur?

Number of cache hits = 3 hits

all have 0 index : 2 frames

Addr	A0	C0	A0	C0	80	00	00	20	C0	A0
Tag										
Index										
Hit/miss	m	m	<u>h</u>	<u>h</u>	"	m	<u>h</u>	<u>n</u>	m	<u>m</u>

- c. (5 points) If the 32-byte cache is four-way set associative with one-byte blocks and uses LRU cache replacement policy, how many cache hits will occur?

Number of cache hits = 4 hits

8 block entries : 4 frames

Addr	A0	C0	A0	C0	80	00	00	20	C0	A0
Tag										
Index										
Hit/miss	<u>N</u>	<u>m</u>	<u>A</u>	<u>h</u>	<u>m</u>	<u>m</u>	<u>h</u>	<u>m</u>	<u>h</u>	<u>m</u>

X