Name [PRINT CLEARLY]:_____

GT email address:_____

CS 2200: Systems and Architecture
Prof. Conte Fall 2016
EXAM II
VERSION A

This exam is given under the Georgia Tech Honor Code System. Anyone found to have submitted copied work instead of original work will be dealt with in full accordance with Institute policies.

Georgia Tech Honor Pledge: "*I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community.*"

[MUST sign:]_____

- THIS IS AN CLOSED BOOK, CLOSED NOTES EXAM
- NO CALCULATORS!
- This examination handout has 8 pages
- Do all your work in this examination handout.
- Use the back of the exam sheets if necessary.
- SHOW ALL YOUR INTERMEDIATE RESULTS TO RECEIVE FULL CREDIT

Problem	Points	Score
1	20	
2	20	
3	25	
4	15	
5	20	
TOTAL	100	

GOOD LUCK!

Useful Facts:
1. 1 KB = 2^{10} bytes, 1 MB = 2^{20} bytes, 1 GB = 2^{30} bytes,

1. [20 pts] Answer the following short questions:

(a) The Elmo computer system has 16GB of physical memory, a virtual address size of 44 bits and a page size of 16KB. (1) How many page frames are in physical memory? (2) A simple, one-level page table is used. How many page table entries does this table have? (3) Instead, an inverted page table is used. How many page table entries does this inverted page table have?

YOU CAN IGNORE THIS QUESTION: WE HAVEN'T COVERED THIS MATERIAL YET

(b) A page size is 1024 B (1KB), a level-one cache is (C = 15, B = 5, S) and is pipelined into two stages. What is the value of S required so that the TLB may be accessed in parallel with the first stage of the cache pipeline?

YOU CAN IGNORE THIS QUESTION: WE HAVEN'T COVERED THIS MATERIAL YET

(c) A particular pipelined processor has a 40 stage pipeline. When branches are executed, there are 20 stalls inserted in the pipeline. Every fourth instruction is a branch. You want to speed this processor up somehow. You **reduce** the cycle time (CT) by a **factor of 3** (i.e., new CT = CT/3), and, **add a branch predictor** that predicts branches with 90% accuracy (a correctly predicted branch produces no stalls). How many times faster is the new processor as compared to the old processor?

(d) Answer the following true/false questions by ***circling*** the correct answer:

TRUE or FALSE	Using more registers can remove anti-dependencies but cannot remove output dependencies from a program.
TRUE or FALSE	Assume the Acme pipelined processor has a branch target buffer to handle control dependencies. The CPI of this processor decreases if the branch target buffer is more accurate in predicting branch behavior.
TRUE or FALSE	The Kermit benchmark calculates the volume of various frogs and reports a metric of frogs per second. The right way to combine benchmark results from various runs of this benchmark is a harmonic mean.
TRUE or FALSE	Two reasons to use a direct-mapped cache over a fully-associative cache is that it uses lower power and costs less hardware.
TRUE or FALSE	A translation fault is when the requested page is not in physical memory. YOU CAN IGNORE THIS QUESTION: WE HAVEN'T COVERED THIS MATERIAL YET

2. [20 pts] The LC-2200 pipelined processor uses the following pipeline:

IF ID EX MEM WB

Assume that:

- all source registers of an operation are read during the second half of the ID stage's clock cycle.
- all destination registers are written during the first half of the WB stage's clock cycle.
- there is register bypass (forwarding) between ID and the EX output and Mem output pipeline buffers.
- Any RAW hazards that are not solved by register bypass result in stall(s).
- the CPI = 1 cycle per instruction when no hazards occur.

Consider the following sequence of code, where the loop A thru E iterates twice (A,B,C,D,E,A,B,C,D,E):

```
A. LW      R4, 0(R7)
B. ADD     R5, R5, R4
C. ADD     R7, R5, R3
D. ADDI    R1, R1, -1
E. BREQ    R1, R2, A
```

(a) Branch behavior and target location are known at the end of the ID stage. IF stalls when a branch occurs—there are no delay slots or branch predictors. Show how long this code takes to execute by completing this timing diagram (instruction A's behavior is filled in already). **WRITE NEATLY:**

Cycle	IF	ID	EX	MEM	WB
0	A				
1		A			
2			A		
3				A	
4					A
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					

(b) Now assume that there is NO register bypass (forwarding) between ID and the EX output pipeline buffer, nor between ID and Mem output pipeline buffer. Repeat part (a):

Code shown again for convenience:

```
A. LW      R4, 0(R7)
B. ADD     R5, R5, R4
C. ADD     R7, R5, R3
D. ADDI    R1, R1, -1
E. BREQ    R1, R2, A
```

Cycle	IF	ID	EX	MEM	WB
0	A				
1		A			
2			A		
3				A	
4					A
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					

3. [25 pts] Consider the following accesses to a cache (addresses are in hex):

(1) `read  xA523`
(2) `write  xC7EF`
(3) `read  xA600`
(4) `write  xC8DD`
(5) `read  xA743`
(6) `read  xA823`

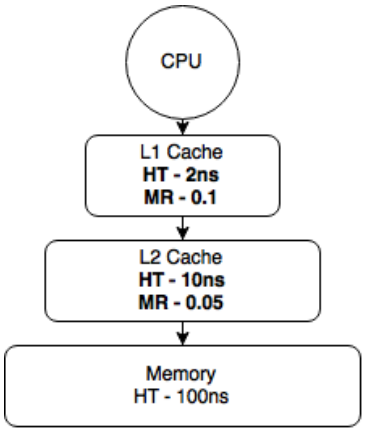
- The cache is **empty initially** and the **entire block** is fetched on a miss.
- The cache uses the WBWA write policy.

(a) Assume the access sequence, (1)-(6), occurs **5 times** for a total of **30 accesses.** What is the total number of misses for a byte addressable, direct mapped cache which has a data store of 4 KB and block size of 256 bytes?

(b) How many **write backs** occur over the access sequence in part (a) (**30 total accesses**)?

(c) Given the same access sequence, cache size and block size as part (a), what is the smallest value of **S** required such that there are only compulsory misses?

(d) Assume the following cache configuration, and compute the average access time of the memory hierarchy.



4. [15 pts] Consider the following processes scheduled by a **round-robin, pre-emptive** operating system scheduler.

Three processes, X, Y and Z, have the following run-time behavior:

Process X: CPU burst for 3ms, I/O burst for 1ms, CPU burst for 4ms, halt
Process Y: CPU burst for 2ms, I/O burst for 2ms, CPU burst for 4ms, halt
Process Z: CPU burst for 4ms, I/O burst for 2ms, halt

The round-robin, pre-emptive scheduler has the following characteristics:

- **The quantum is 2ms**.
- A process leaving I/O enters the ready queue at the **END** of the ready queue, in the next cycle.
- A process can enter I/O in the next quantum after leaving the run state.
- For a given cycle, pre-empted processes entering the ready queue get priority over processes entering the ready queue after finishing I/O.
- I/O is non pre-emptive, FIFO-scheduled, and new requests enter at the end of the I/O queue

Show the run state per quantum, ready queue and the I/O queue occupancy for the three processes, X, Y and Z. Use '**R**' for running, and '**I**' for in I/O. Leave a cell blank when a process is waiting.

You may not need all columns. Times 0ms, 1ms and 2ms are filled in for you.

Time=	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Process X	**R**	**R**																			
Process Y			**R**																		
Process Z																					
Ready queue contents (first to last)	Y Z	Y Z	Z X																		
Waiting for I/O queue contents (first to last)																					

5. [20 pts] Consider a single level paged virtual memory system that includes a **TLB**, and has the following characteristics:
   - A page size is **4 KB**
   - Virtual Addresses and physical Addresses are **24 bits** each

Below is the current state of the TLB and of Process 1's page table (All values are in hex):

TLB

VPN	PFN	V
ACE	0BB	0
BAD	0FF	1
123	0BE	0
264	568	1

P1 – Page Table

VPN	PFN	V
123	0BE	0
264	568	1
ACE	450	1
ADF	567	0
BAD	0FF	1
CAD	965	1
DED	444	1
DFE	332	0

(a) Translate the following virtual addresses to physical addresses. If the address results in a page fault, mark it as "Page Fault".

Virtual Address	Physical Address
ACE343	
CAD432	
123AEE	
BAD655	
264574	

(b) How many **translation** misses are there?

(c) There is a context switch, and now the CPU has to run Process 2. Does the TLB have to be flushed before Process 2 starts executing on the CPU? Why or why not?