Name:_____

CSE 30321 – Computer Architecture I – Fall 2008
Final Exam
December 18, 2008

Test Guidelines:

1. Place your name on EACH page of the test in the space provided.
2. Answer every question in the space provided. If separate sheets are needed, make sure to include your name and clearly identify the problem being solved.
3. Read each question carefully. Ask questions if anything needs to be clarified.
4. The exam is open book and open notes.
5. All other points of the ND Honor Code are in effect!
6. Upon completion, please turn in the test and any scratch paper that you used.

Suggestion:

- Whenever possible, show your work and your thought process. This will make it easier for us to give you partial credit.

Question	Possible Points	Your Points
1	10	
2	15	
3	15	
4	15	
5	20	
6	10	
7	15	
Total	100	

Name:_____

Problem 1: (10 points)
Assume that you have a 4-way, *set-associative* cache with:

- 8192 *total blocks*
- 32 words per block
- 64 bits per word
- 64 bits per physical address

Question A: (7 points)
If addresses are *to the word* what bits of the address will comprise the index, tag, and offset?

Answer
- Index:
  - # of sets: 8192 / 4 = 2048 = 2^{11}
  - Therefore 11 bits of index
- Offset:
  - # of words per block = 32
  - 2^5 = 32
  - Therefore 5 bits of offset
- Tag
  - 64 – 5 – 11 = 48

Therefore, bits 0-4 are offset, bits 5-15 are index and bits 16-63 are tag.

Question B: (3 points)
If addresses are to the byte, how – if at all – does the above change?

Answer
- There are 8 bytes per word and 32 words. We need to address each byte and therefore need an additional 3 bits of offset.
- Thus:
  - Offset: 8 bits
  - Index: 11 bits
  - Tag: 45 bits

Name:_____

Problem 2: (15 points)

Question A: (7 points)

Assume you have space on your chip to make a cache that is direct mapped and 32 KBytes. The processor that you are designing will be used primarily for handling memory requests that occur randomly (e.g. not sequentially). In most cases, the amount of data required will only be 2-4 bytes. Given these requirements, how might you organize your cache? Why?

Answer

- Make the cache "long and skinny" – e.g. have more blocks that store less data. This will reduce conflict misses. A short/fat cache has good spatial locality, but will lead to more conflicts assuming cache size is fixed.

Question B: (8 points)

Assume that you have a direct mapped cache with 4 blocks; each block holds 4 words. It is initially empty. Addresses are to the word. Now, assume you get the following pattern of address requests (in hex):

- A B C
- A B D
- A B B
- A B F
- B B D
- A B B
- B B F
- A B C

Which requests are hits and which are misses?

Answer:

-	A B C	I	11 00	Miss	(compulsory miss)
-	A B D	I	11 01	Hit	(spatial locality)
-	A B B	I	10 11	Miss	(compulsory miss)
-	A B F	I	11 11	Hit	(temporal locality + spatial locality)
-	B B D	I	11 01	Miss	(compulsory miss)
-	A B B	I	10 11	Hit	(temporal locality)
-	B B F	I	11 11	Hit	(temporal locality + spatial locality)
-	A B C	I	11 00	Miss	(conflict miss)

Name:_____

Problem 3: (15 points)

Question A: (3 points)

Is the following statement true or false?

"A miss in the TLB implies a page fault has occurred."

Answer
False. We could have a page fault, but a TLB miss only implies that the VPN number is not cached.

Question B: (3 points)

Assume that your CPU supplies a virtual address that is 40 bits long:

F 2 5 D 3 F 0 8 0 1 (hex)

In this system:

- Physical addresses are 32 bits
- Page sizes are 4 KBytes
- The Page Table Register = 0 0 0 0 0 0 0 F (hex)

What address in the page table should contain the Virtual Page Number that we need?

Answer

0 F 2 5 D 3 F 0 + 0 0 0 0 0 0 0 F = 0 F 2 5 D 3 F F

Question C: (3 points)

For the address in Question B, how many bits of data should the Page Table Entry hold? (You can ignore valid bits, dirty bits, etc.)

Answer

32 – 12 = 20.

Name:_____

Question D: (3 points)

After we calculate a physical address, how many entries down in the page will we find the data we are actually looking for?

Answer

- Offset: 8 0 1 (hex) = 1000 0000 0001
- Therefore 2^{11} + 20 = 2049.

Question E: (3 points)

Assuming a clock rate of 1 GHz, which answer below best describes the number of ***clock cycles*** required to process a page fault (circle your answer).

1 10 1000 1,000,000 1,000,000,000,000

Answer

1,000,000. This is equivalent to 1 ms and is about right for the time to go to disk. The next higher answer is 16 minutes which we hope is not the case!

Name:_____

Problem 4: (15 points)

This question considers the basic, MIPS, 5-stage pipeline. (For this problem, you may assume that there is full forwarding.)

Question A: (7 points)
Show how the instructions will flow through the pipeline:

	1	2	3	4	5	6	7	8	9	10	11	12
Lw $5, 0($7)	F	D	E	M	W							
Add $19,$7,$6		F	D	E	M	W						
Xor $5,$19,$3			F	D	E	M	W					
Sub $2,$10,$11				F	D	E	M	W				
Sw $2, 0($1)					F	D	E	M	W			

Question B: (2 points)
What are the data dependencies in this instruction mix?

Answer
Between (Add, XOR) and (Sub, Sw)

Question C: (2 points)
How many stall clock cycles arise due to hazards in this mix?

Answer
0. Forwarding helps us avoid all potential hazards.

Question D: (4 points)
Where does the XOR instruction get its data? (Be very specific)

Answer
One operand comes from the pipe latch between the logic for the execute state the data memory stage. It will be forwarded back to the ALU, to a mux, and the appropriate input will be selected. The other operand comes from the register file. It too will pass through a mux and then go on to the ALU.

Name:_____

Problem 5: (20 points)

This question considers the basic, MIPS, 5-stage pipeline. (For this problem, you may assume that there is full forwarding.)

Question A: (7 points)
Show how the instructions will flow through the pipeline:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Sub $17,$15,$12	F	D	E	M	W											
Sub $9,$8,$4		F	D	E	M	W										
Lw $10, 0($5)			F	D	E	M	W									
Add $1,$10,$2				F	D	D	E	M	W							
Add $6,$1,$3					F	F	D	E	M	W						
Lw $20, 0($21)							F	D	E	M	W					

Question B: (5 points)
What is the CPI of this instruction sequence if it is executed 1000 times?

Answer
The next sequence of instructions can start after 7 CCs. Accounting for the time to "drain" the pipeline, 7000 + 4 or 7004 CCs are required. The CPI is thus 7004 / 6000 = 1.1673.

Question C: (2 points)
How does the performance of this code compare with the ideal? Quantify your answer.

Answer
The ideal CPI is 1. This is about 16.73% worse.

Name:_____

Question D: (3 points)
How many clock cycles would 1000 iterations of the above sequence of instructions take if it were executed on a *multi-cycle* machine? (Assume loads take 5 CCs and adds take 4 CCs.)

Answer

[(4 x 4) + (2 x 5)] x 1000 = 26,000 CCs

Question E: (3 points)
If the clock cycle time of the multi-cycle machine is 80% of that of the pipelined machine, which machine is faster? By how much?

Answer

- time (multi-cycle)
  - 26000 * 0.8 (time-pipe) = 20,800 (time-pipe)
- time (pipe)
  - 7004 (time-pipe) = 7,004 (time-pipe)

Therefore, the pipelined machine is ~2.97X faster than the multi-cycle machine.

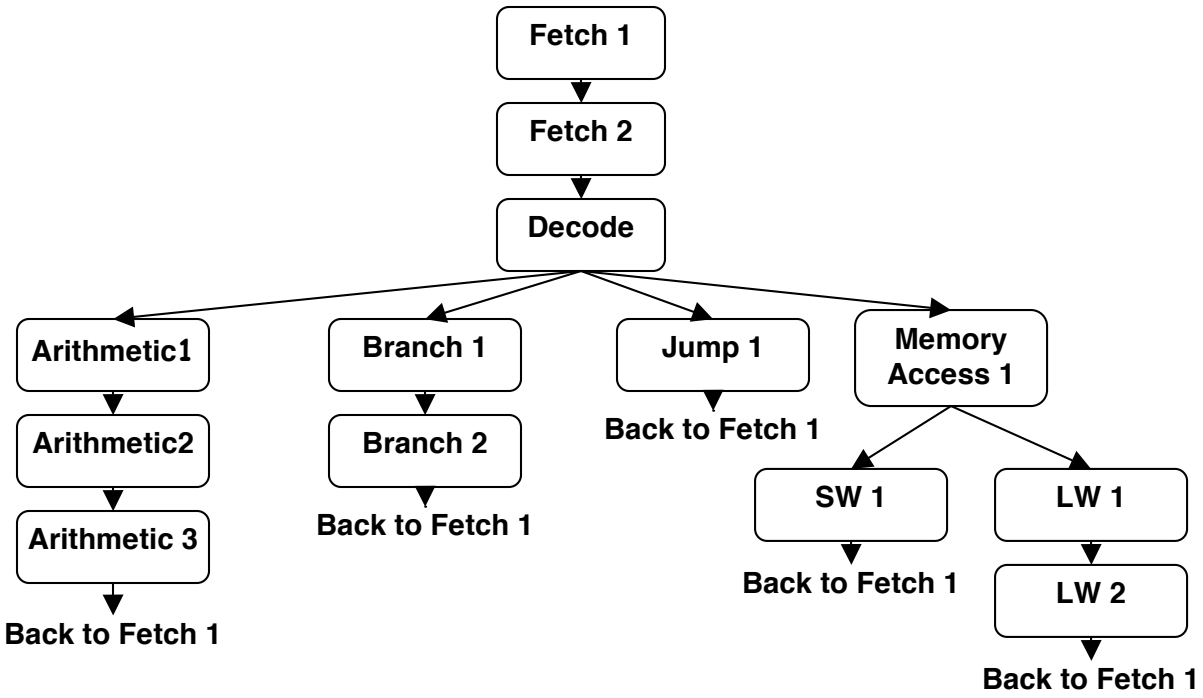
Name:_____

Problem 6: (10 points)

Part A: (5 points)
Assume that we have developed a datapath for a multi-cycle machine that can execute the following instructions: ADD, ADDI, AND, BEQ, BNEQ, JUMP, LW, OR, SUB and SW. The instructions are broken down into classes as follows:

Instruction Class	Instructions	Frequency of Class
Arithmetic	ADD, ADDI, AND, OR, SUB	50%
Branch	BEQ, BNEQ	20%
Jump	JUMP	10%
Memory Access	LW, SW	20% (10% loads, 10% stores)

The state machine for this particular datapath is shown below:



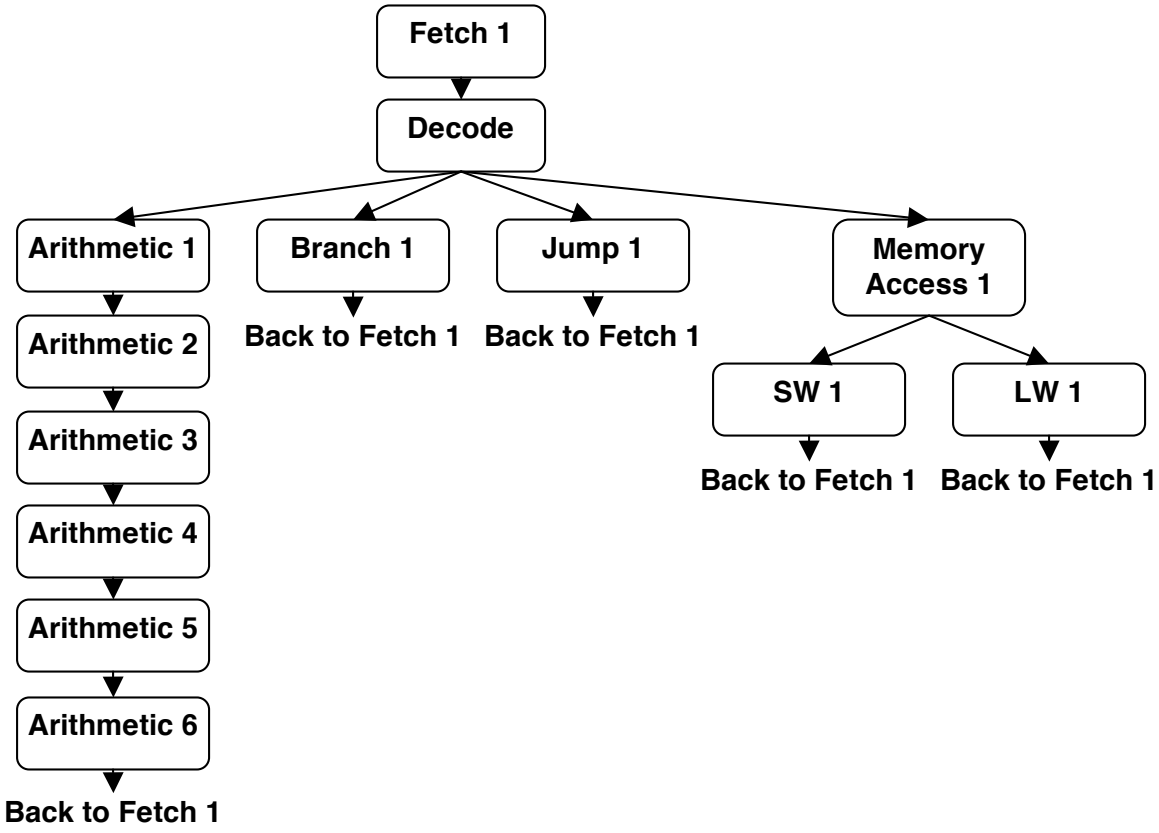
Calculate the average CPI:

Answer:

CPI = (.5)(6) + (.2)(5) + (.1)(4) + (.1)(5) + (.1)(6)
= 3 + 1 + .4 + .5 + .6
= 5.5 clock cycles/instruction

Name:_____

Part B: (5 points)
We have made some changes to our datapath causing some changes in the state machine. The new state machine appears below. (Note that the compiler has not changed and instruction frequencies remain the same).



Were these changes smart to make? Justify your answer (and note that a simple 'yes' or 'no' answer will earn you 0 points for this part of the problem).

Answer:

New CPI = (.5)(8) + (.2)(3) + (0.1)(3) + (.1)(4) + (.1)(4)
= 4.0 + .6 + .3 + .4 + .4
= 5.7 clock cycles per instruction

The new CPI is higher (5.7 vs. 5.5). Therefore this is not a good thing to do.

Name:_____

Problem 7: (15 points)
You have a dual core microprocessor that needs to execute 5 tasks. The instruction count and CPI for each task is shown in the table below. (You may assume a clock rate of 1 GHz).

Task	Instruction Count	CPI
1	400,000	1.7
2	2,100,000	2.1
3	3,000,000	1.4
4	10,000,000	1.2
5	6,000,000	1.5

Your job is to figure out which tasks should run on each core. However, there is a catch. Tasks 4 and 5 need to communicate with each other. If they are placed on separate cores, the CPI of Task 4 will rise to 2.3 and the CPI of Task 5 will rise to 3.0.

Which tasks should run on each core? Why?

Answer:

Task	IC	CPI	CCs	Comment
1	400,000	1.7	680,000	Core 2
2	2,100,000	2.1	2,100,000	Core 1
3	3,000,000	1.4	4,200,000	Core 2
4	10,000,000	1.2	12,000,000	Core 1
5	6,000,000	1.5	9,000,000	Core 2
4a	10,000,000	2.3	23,000,000	
5a	6,000,000	3	18,000,000	

If this is no communication penalty, having Tasks 2 and 4 on one core and 1, 3 and 5 on the other core makes the most sense (2 and 4 will take 14,100,000 CCs and 1, 3, and 5 will take 13,880,000 CCs – about the same time).

However, now Tasks 4 and 5 are separated and the higher CPI comes into play. With the distribution above, Tasks 2 and 4 would take 25,100,000 CCs and 1, 3 and 5 would take 22,880,000 CCs.

Compare this to having Tasks 4 and 5 run on one core. Yes, the time would be 21,000,000 CCs versus 6,980,000 CCs – but both are lower than the 2 alternatives above. Therefore, its just best to have 4 and 5 be on the same core.