EECS 470 *Final Exam*

Winter 2012

Name: _____ unique name: _____

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

_____

Scores:

#	Points
Page 2	**/11**
Page 3	**/13**
Page 4	**/10**
Page 5	**/7**
Page 6	**/7**
Page 7	**/7**
Page 8	**/10**
Page 9	**/10**
Page 10	**/10**
Page 11	**/15**
Total	**/100**

NOTES:

1. Open book and Open notes
2. There are ***11*** pages to this exam (including this one). Please check you have them all.
3. Calculators are allowed, but no PDAs, Portables, Cell phones, etc.
4. Don't spend too much time on any one problem.
5. You have about 120 minutes for the exam.
6. **Be sure to show work and explain what you've done when asked to do so.**

____

Section I—Objective Section [32 points]

1. Circle the best answer. **[11 points, -2 for each wrong or blank answer, minimum of zero]**
   a) In an n-way super-scalar processor, the number of checks required to determine if there are dependencies between any of the n instructions being issued is on the order of
   n / n log(n) / n^2 / n^2log(n) / n^3 / 2^n

   b) If the multiplier is in the critical path for the processor's clock period, decreasing the number of pipeline stages used to do a multiply can be expected to ***increase /decrease / not affect*** the clock period while ***increasing / decreasing / not affecting*** the CPI.

   c) Consider the RAT used in the P6 scheme discussed in class. If the ROB has 64 entries, the ARF has 32 entries and the RS has 8 entries, the RAT will have a total of
   48 / 128 / 192 / 256 / 316 / 512 bits (not including valid bits and the like).

   d) Say a colleague has proposed an architectural change that will increase performance by about 3 percent. In order to do as well or better than voltage scaling, her change shouldn't use more than about an extra ***1 / 2 / 3 / 6 / 9*** percent power.

   e) Comparing a stack-based architecture to a load-store machine, the instruction in a stack-based machine generally encoded using ***fewer / more / about the same*** number of bits as the load-store machine. The total number of instructions to implement an algorithm in a stack-based machine is generally ***greater than / less than / about equal to*** a load-store machine.

   f) If you are using a 4KB four-way associative cache where the tag of the cache is found in parallel with the virtual-to-physical address translation, you would expect that the page size of the machine is ***less than or equal to 4 KB / less than or equal to 1 KB / greater than 8 KB / greater than 4 KB / greater than or equal to 4KB / greater than or equal to 1 KB***.
   (You are to pick the most restrictive answer that is true).

2. Which of the following are true of a hash cache? *LIST all correct answers*. [**3 points, -2 per wrong/missing answer, minimum of 0]**
    a) It effectively increases the degree of associativity of the cache.
    b) It reduces non-random conflicts often associated with striding.
    c) It has a slightly longer access time due to the hash function having to be computed.

    _____ ← Place your answer here

3. Which of the following are true of a skew cache? *LIST all correct answers*. [**3 points, -2 per wrong/missing answer, minimum of 0]**
    a) It effectively increases the degree of associativity of the cache.
    b) It reduces non-random conflicts often associated with striding.
    c) It has a slightly longer access time due to the hash function having to be computed.

    _____ ← Place your answer here

4. Say we are designing an LSQ on a system where we don't double-check for data mis-speculation (that is, we may only supply the data for a load when we are certain we have the right data). Say we've got 3 instructions, A, B and C. A is the first in program order, then B and then C. C is a load, A and B are stores. *For each problem,* **circle** *the* **best** *answer.* **[7 points, -3 for each wrong or blank answer, minimum of zero]**

    a. If store "A" has no known address and store "B" has an address that is the same as C's address, we should (ideally)
        - Forward the data from B to C and put that data on the CDB
        - Wait for A's address to resolve before taking any action
        - Send a request to memory for C's data in case we need it later and then wait for A to resolve.

    b. If store "B" has no known address and store "A" has an address that is the same as C's address, we should (ideally)
        - Forward the data from A to C and put that data on the CDB
        - Wait for B's address to resolve before taking any action
        - Send a request to memory for C's data in case we need it later and then wait for B to resolve.

    c. If store "A" has no known address and store "B" has an address that is ***not*** the same as C's address, we should (ideally)
        - Forward the data from B to C and put that data on the CDB
        - Wait for A's address to resolve before taking any action
        - Send a request to memory for C's data in case we need it later and then wait for A to resolve.

5. **[3 points, 2 for the fill-in-the-blank and 1 for the multiple choice]**
Arguing that "people are weird" (and perhaps more so, that marketing people have useful insights engineers should pay attention to), Bob Colwell gave the hamburger chain Wendy's as an example. Wendy's wasn't selling hardly any triple burgers, so they pulled them off the menu. Shortly after that they were returned to the menu. This was because

_____.

The point of the story was that marketing folks, for good reason, want to
(*circle the best answer*).

- have at least three products in a product line
- have every product that can reasonably be built be available.
- make low-cost options more attractive to the consumer.

6. When using the MESI coherence protocol and bus protocol described in class, which of the following are true? Assume the cache is write-back and write-allocate. *LIST all correct answers*. [**7 points, -2 per wrong/missing answer]**
   a) If a given cache line is held in the "E" state and that cache's processor wishes to do a **write**, a BRIL transaction will be placed on the bus.
   b) If a given cache line is held in the "M" state and that cache's processor wishes to do a **read**, the transaction need not be sent on the bus.
   c) If a given cache line is held in the "S" state and that cache's processor wishes to do a **write**, a BIL transaction is placed on the bus.
   d) If a given cache line is held in the "E" state and another processor does a BRL on the bus, that given cache line will change to the "S" state.
   e) The only state which indicates that the data is "dirty" is the "M" state.
   f) When a cache-line held in the "E" state is evicted, a BWL transaction is placed on the bus.

   _____ ← Place your answer here

Section II—Short Answer [33 points]

1. Consider a four-core processor with individual L1 caches and a shared L2 cache where coherence of the L1 caches is managed by using a snoopy bus using the MESI protocol. You have been tasked with evaluating locking code which will be running on this hardware platform. One locking mechanism uses "test and set" to protect the critical section, while the other uses "load-lock, store-conditional". The critical section in question is fairly long and quite memory intensive (lots of loads and stores though none to the locking memory location or its cache line).

Test and set scheme:

```
AQUIRE:
      R1=Test_and_set(MEM[R2])      // Atomic read/modify/write
      If(R1==1) goto AQUIRE

CRITICAL_SECTION:
   //Critical section code goes here, branches to FREE when done.

FREE: R3=0
      MEM[R2]=R3
```

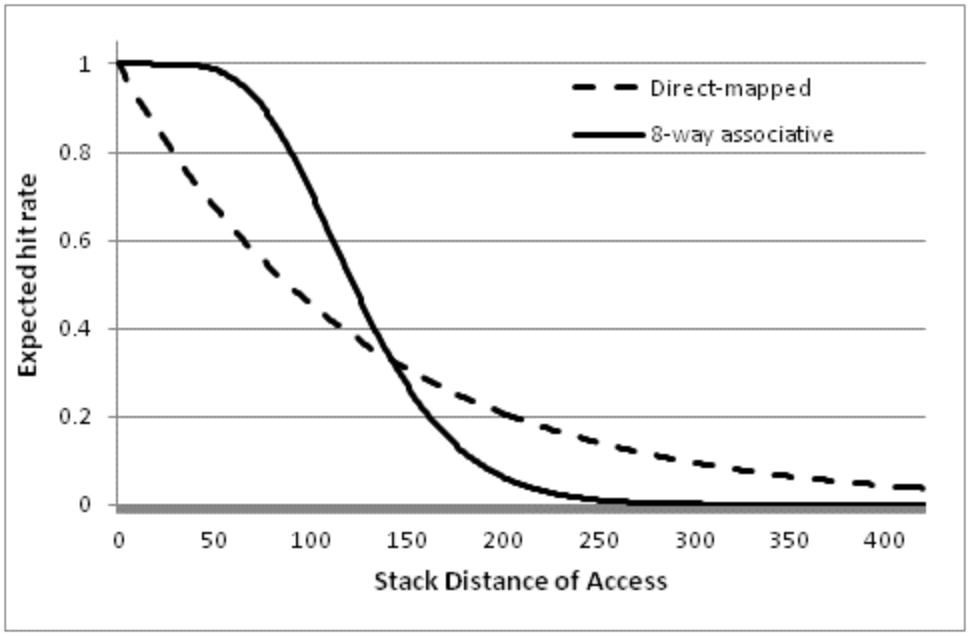
Load-lock, store-conditional scheme:

```
AQUIRE:
      R3=1
      R1=MEM[R2]                    // This is a load lock
      If(R1==1) goto AQUIRE
      MEM[R2]=R3                    // This is a store conditional
      If(R3==0) goto AQUIRE

CRITICAL_SECTION:
//Critical section code goes here, branches to FREE when done.

FREE:
      R3=0
      MEM[R2]=R3              // This is a normal store
```

Which of these two locking schemes would generate less bus traffic? **Clearly** explain your answer. **[7 points]**



2. In the above graph, we see the expected hit rate of a memory access with a given stack distance for two caches with the same number of lines (128) but different degrees of associativity. Both lines have the same area under the curve (again, 128). Using this graph, explain why more associative caches tend to get higher hit rates than less associative caches. **[7 points]**

3. Draw the state transition diagram that corresponds to the following Verilog code. **[7 points]**

```
module arbiter (clock, reset, req_0,
                req_1, gnt_0, gnt_1);
input clock, reset, req_0, req_1;
output gnt_0, gnt_1;
reg  gnt_0, gnt_1;
parameter  IDLE=2'b01;
parameter  GNT0=2'b10;
parameter  GNT1=2'b00;
reg [1:0] state;
reg [1:0] next_state;

always @*
begin
  next_state=state;
  case(state)
  IDLE :  if (req_1)
            next_state = GNT0;
          else if (req_0)
            next_state= GNT1;
          else
            next_state = IDLE;
  GNT0 :  if (req_0 == 1'b1)
            next_state = GNT0;
          else
            next_state = IDLE;
  GNT1 :  if (req_1 == 1'b1)
            next_state = GNT1;
          else
            next_state = IDLE;

  default:  next_state = IDLE;
  endcase
end
```

```
always @ (posedge clock)
begin
  if (reset == 1'b1)
 state <= IDLE;
  else
 state <= next_state;
end


always @*
begin
  case(state)
  IDLE : begin
           gnt_0 = 1'b1;
           gnt_1 = 1'b1;
         end
  GNT0 : begin
           gnt_0 = 1'b0;
           gnt_1 = 1'b1;
         end
  GNT1 : begin
           gnt_0 = 1'b1;
           gnt_1 = 1'b0;
         end
  default :
         begin
           gnt_0 = 1'b1;
           gnt_1 = 1'b1;
         end
  endcase
end
endmodule
```

4. Consider processor which, when running a given application performs 10 billion loads and 10 billion stores per second. Assume the following is true: **[10 points]**

    - The processor's multi-level cache system gets a 90% hit rate on both loads and stores.
    - Cache lines are 64-bytes in size
    - There is no prefetching and the instruction cache never misses.
    - 25% of all lines evicted from the last level of the cache are dirty.
    - The cache is write-back and write-allocate.
    - There are no coherence misses.
    - All loads and stores are to 4-byte values.
    - The bus only supports an operation size of 64-bytes.

    a) What is the read bandwidth (bytes/second) on the bus? Show your work. **[5]**

    b) What is the write bandwidth (bytes/second) on the bus? Show your work. **[5]**

Section III—Little Boxes [35 points]

1. Consider the following C code:

```
int SIZE, STRIDE;
short A[SIZE];   // shorts are 2 bytes on this computer

// Initialize SIZE and STRIDE here
for(j=0;j<N;j++)
   for(i=0;i<SIZE;i=i+STRIDE)
     X+=A[i];
```

Assume N is a very large number. What approximately what *hit rates* would you expect to get on a 1KB, direct-mapped data cache with 32-byte lines given the following values for STRIDE and SIZE? You are to assume that every value other than the array A is kept in registers and that shorts are 2 bytes in size.
[10, -3 per wrong or empty box, min 0]

	STRIDE=4	STRIDE=32	STRIDE=48
SIZE=512			
SIZE=2048			

2. Consider a case of having 2 processors using a snoopy MESI protocol where the memories can snarf data. All three have a 2 line direct-mapped cache with each line consisting of 16 bytes. The caches begin with all lines marked as invalid. Fill in the following tables indicating
   - If the processor gets a hit or a miss in its cache
   - If a HIT or HITM (or nothing) occurs on the bus during snoop.
   - What bus transaction(s) (if any) the processor performs (BRL, BWL, BRIL, BIL)
   - For misses only, indicate if the miss is compulsory, capacity, conflict, or coherence. A coherence miss is one where there would have been a hit, had some other processor not caused an invalidation of that line.

   In the event more than one bus transaction occurs due to a given memory read/write indicate the response for each bus transaction. Finally, indicate the state of the processor after all of these memory operations have completed. The operations occur in the order shown.
   [10 points, -0.5 per wrong or blank, minimum of 0]

Processor	Address	Read/Write	Bus transaction(s)	Hit/Miss	HIT/ HITM	"4C" miss type (if any)
1	0x100	Read				
1	0x110	Write				
1	0x210	Read				
1	0x110	Read				
1	0x100	Write				
2	0x110	Read				
1	0x110	Write				
2	0x210	Read				

Final state:

	Proc 1	
	Address	**State**
Set 0		
Set 1		

	Proc 2	
	Address	**State**
Set 0		
Set 1		

3. In this problem we are using the R10K algorithm with an RRAT. Say the ROB, RS, RAT, PRF, and RRAT are as follows:

RAT	
Arch Reg. #	**Phys. Reg. #**
0	1
1	2
2	11
3	10
4	9
5	8

ROB				
Buffer Number	**PC**	**Done with EX?**	**Dest. Arch Reg #**	**Dest. Phy. Reg #**
0				
1				
2				
3				
4				
5				
6				
7				

RRAT	
Arch Reg. #	**Phys. Reg. #**
0	1
1	2
2	11
3	10
4	9
5	8

Head of ROB=0
Tail of ROB=0

RS							
RS	**Op type**	**Op1 ready**	**Op1 PRN/value**	**Op2 ready**	**Op2 PRN/value**	**Dest. PRN**	**ROB**
0							
1							
2							
3							

Phy. Register File			
Phy. Reg	**Value**	**Free**	**Valid**
0	1	Y	N
1	2	N	Y
2	3	N	Y
3	4	Y	N
4	-1	Y	N
5	-2	Y	N
6	-3	Y	N
7	-4	Y	N
8	8	N	Y
9	-9	N	Y
10	0	N	Y
11	-5	N	Y

Key:

- **Op1 PRN/value** is the value of the first argument if "Op1 ready?" is yes; otherwise it is the Physical Register Number that is being waited upon.
- **Op2 PRN/value** is the same as above but for the second argument.
- **Dest. PRN** is the destination Physical Register Number.
- **ROB** is the associated ROB entry for this instruction.
- **Free/Valid** indicates if the PRF entry is currently available for allocation and if the valid in it is valid

The following instructions have been issued.

```
top: R1=R2*R3              // A
     R3=R4+R5              // B
     if(R3==R4) goto top   // C
     R5=R1+R2              // D
     R3=R5+R3              // E
```

A has committed, D has completed execution and the other instructions (B,C,E) have issued but have not yet started execution. The branch is predicted not-taken. Fill the different structures. The PRF will always allocate the lowest numbered free PRF entry. The PC of instruction A is 20 and each instruction is 4 bytes in size. **[15 points]**

____