EE380 Fall 2015 Final Exam

Your name is:

Your email address is:

The two questions you skipped are:

There are 22 questions; you are to answer any 20 questions. You will lose 5 points for each of the questions that you did not answer correctly, except the two questions that you said to skip (above). For each multiple choice question, mark only the □ in front of the single best answer. The other questions should get **short** answers; excessively long answers may be considered incorrect. The answer to the fifteenth question is that it should be zero to avoid the side effect of writing garbage to memory. The test is closed book, closed notes, closed minds (read neither tests nor minds of others ;-).

1. For this question, *mark all that apply*. Which of the following statements about the memory hierarchy is true?

☐ There are usually separate L1 caches for instructions and data

☐ A translation lookaside buffer is basically a cache for page table entries

☐ For the best performance, you want cache to have as few hits as possible

☐ Replacement policy is simpler to implement for a direct-mapped cache than for a fully-associative cache

☐ A program referencing the elements of an array in reverse order, `a[1000]`, `a[999]`, `a[998]`, ... would be said to have good spatial locality

2. For this question, *mark all that apply*. Which of the following terms is describing a hardware structure or algorithm that is primarily intended to eliminate **control dependences** or to reduce their negative impact on pipelined performance?

☐ ACM

☐ ALU

☐ BHB

☐ BTB

☐ LRU

3. For this question, *mark all that apply*. Which of the following are true statements about parallel computers?

☐ In SIMD computers, the same operation is simultaneously performed on multiple data

☐ Multi-core processors typically use coherent shared memory to communicate data values between cores

☐ Graphics processing units are important because they provide massively-parallel execution at very low cost

☐ A cluster computer uses a bunch of conventional processors to execute portions of a program in parallel

☐ For the interconnection network inside a parallel supercomputer, latency is not important as long as you have high bandwidth

4. A quadcopter is a popular type of drone aircraft. They are very maneuverable, but that's actually because they are quite unstable: the slightest variation in speed of one of the four rotors, or a gust of wind, is enough to send the drone shooting off in some direction. To make them easier to fly, most have an onboard computer that is continually monitoring the drone position and orientation, and it must immediately compensate for any unintended motion. Of course, every so often, the human pilot (on the ground) will send the computer a command via a digital radio link requesting that the drone move to a new position. How would you expect the onboard computer to handle receiving such a message: polling, an interrupt, or DMA? Explain your choice.

5. Given the following MIPS code, in which "..." refers to a sequence of instructions that do not alter the values of **$t0** nor **$t1**, would predicting the **beq** is always not-taken be better or worse than predicting always taken? Why?

```
    or   $t0,$0,$0
    li   $t1,1000
a:  beq  $t0,$t1,b
    ...
    addi $t0,$t0,1
    j    a
b:
```

6. Consider the pipelined MIPS subset implementation shown at the back of this test. Suppose that the propagation delay for each of the five stages is respectively 5ns, 2ns, 10ns, 5ns, and 2ns. Somebody noticed that this could instead be built as a simpler three-stage pipeline with propagation delays of 8ns, 10ns, and 8ns. What affect would this change have on the fastest clock frequency that could be used? Briefly explain.

7. For this question, *mark all that apply*. Which of the following 4-bit numbers has the exact same value in 1's complement, 2's complement, and sign+magnitude notations?

□ 0000

□ 0001

□ 1000

□ 1001

□ 1111

8. For this question, *mark all that apply*. Consider executing the MIPS instruction **`addu $t0,$t1,$t2`** and then any one of the following single MIPS instructions. Assume that the hardware being used looks like the **pipelined** MIPS implementation given at the end of this test. Mark each that would execute faster using **value forwarding** than without value forwarding.

□ **`sw $t3,0($t0)`**

□ **`lw $t4,0($t3)`**

□ **`and $t1,$t2,$t3`**

□ **`xori $t1,$t2,42`**

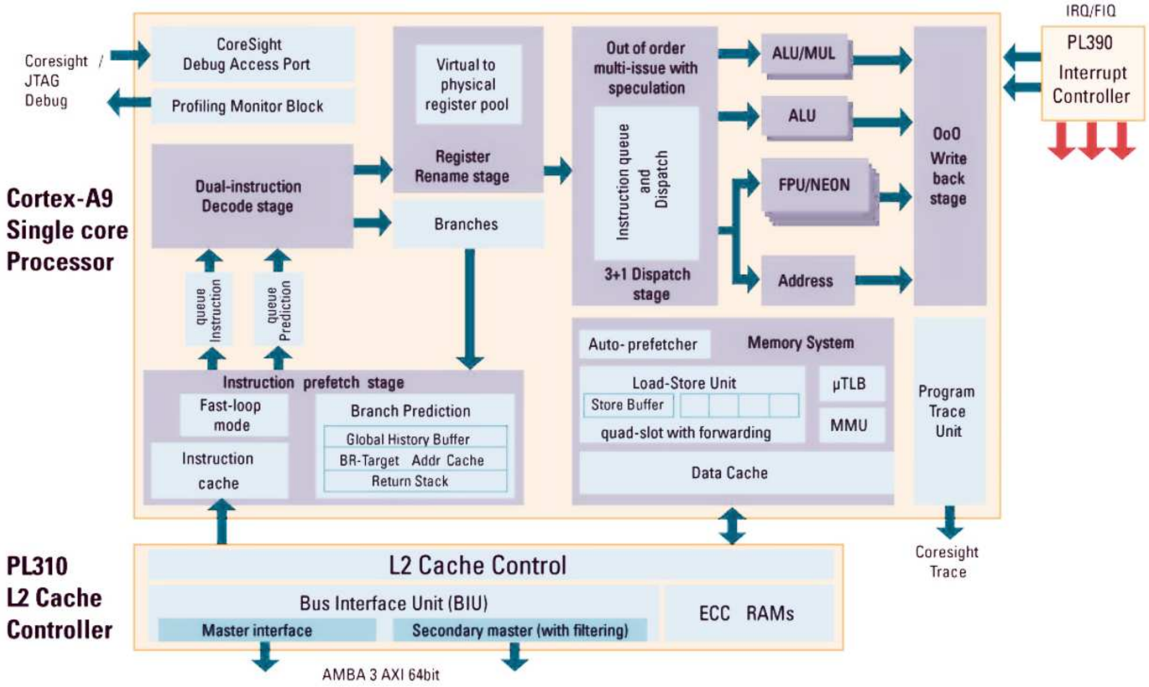
□ **`beq $t0,$0,place`**

9. Consider executing the following code sequence on the **pipelined** MIPS implementation given at the end of this test *without value forwarding*. Show any true dependences and reorder these instructions so that the same values are computed, but pipelined execution can be expected to take fewer clock cycles. (You don't need the fastest reordering, just one that is faster than the order given below.)

```
A: sw   $t0,0($t1)
B: addu $t3,$t0,$t2
C: slt  $t5,$t3,$t4
D: lw   $t0,4($t1)
E: and  $t3,$t4,$t2
```

10. Briefly explain what a **Page Table** is used for.

11. For this question, *mark all that apply*. The following diagram shows the internals of the ARM Cortex A9 processor. According to this diagram, which of the following techniques is used in this design?

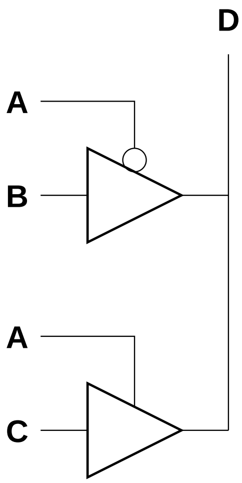


☐ BHB
☐ BTB
☐ Superscalar execution
☐ Separate L1 code and data caches
☐ Out-of-order instruction execution with register renaming

12. For this question, *mark all that apply*. Which of the following things is commonly stored in a stack frame?

☐ The return address
☐ The previous frame pointer value
☐ The machine code for the current function
☐ The heap for new/malloc memory allocation
☐ Local (**`auto`**) variables for the current function

13. Here's a simple circuit using tri-state drivers. In EE280, you did this without tri-state drivers. Give an equivalent logic formula for the output signal D as a function of inputs A, B, and C. You may use only AND, OR, and NOT operators in your logic formula.



14. For this question, *mark all that apply*. Which of the following statements about control logic in general is true?

☐ Outputs are enabled at the end of a clock cycle

☐ Horizontal encoding of control tends to use more bits than vertical encoding

☐ Gate array logic implementations are generally easier to modify than random logic

☐ The choice of how instructions are encoded as bit patterns can change the complexity of the control logic

☐ The control signals for single-cycle and pipelined processor implementations can be essentially the same

15. In the pipelined MIPS architecture discussed here, how should the **MEMwrite** control signal be set within a **pipeline bubble** (or nop instruction): 0, 1, or don't care? Why?

16. A particular system has a cache that takes 1 clock cycle to access and a main memory that takes 2000 cycles. Suppose the **hit ratio** (aka, **hit rate**) is 99%. Approximately how long would the average reference take to execute? No calculators here, so don't worry about giving a precise number value — an approximate answer and/or a formula will do.

17. A particular program, which takes a total of 1000 seconds to run on a single processor, is to be sped-up by running it in parallel on a cluster computer. However, the program starts by reading an input file which must be done by a single processor and takes 10 seconds. Assuming the rest of the program can be perfectly parallelized and arbitrarily many cluster processors are available, what is the maximum speedup one night obtain? Why?

18. One of the little quirks of the MIPS instruction set is that there are no conditional jump instructions, just **`j`** and **`jr`**. Of course, there are only conditional branches, **`beq`** and **`bne`**. Suppose that you need to conditionally jump to a location, **`L`**, which is too far away for **`beq $t0,$0,L`** to work. Give MIPS code that would implement a jump to **`L`** when the condition **`$t0==$0`** is true and fall through otherwise.

19. For this question, *mark all that apply*. Which of the following parameter values are roughly correct for modern high-end desktop PCs?

☐ A cache line contains about 32 bytes of data

☐ A disk drive can hold up to several gigabytes of data

☐ There are between 1-64 cores on the processor chip

☐ There are several billion transistors on the processor chip

☐ A page contains about 4K bytes of data (which is rather small!)

20. For this question, *mark all that apply*. Given that the system is using IEEE 754 floating-point arithmetic, which of the following will result in a value of 1 for **`r`**? (Note: 268435456 is 2^{28}.)

□ **`int a=1, b=268435456, r; r=a+(b-b);`**

□ **`int a=1, b=268435456, r; r=(a+b)-b;`**

□ **`float a=1, b=268435456, r; r=a+(b-b);`**

□ **`float a=1, b=268435456, r; r=(a+b)-b;`**

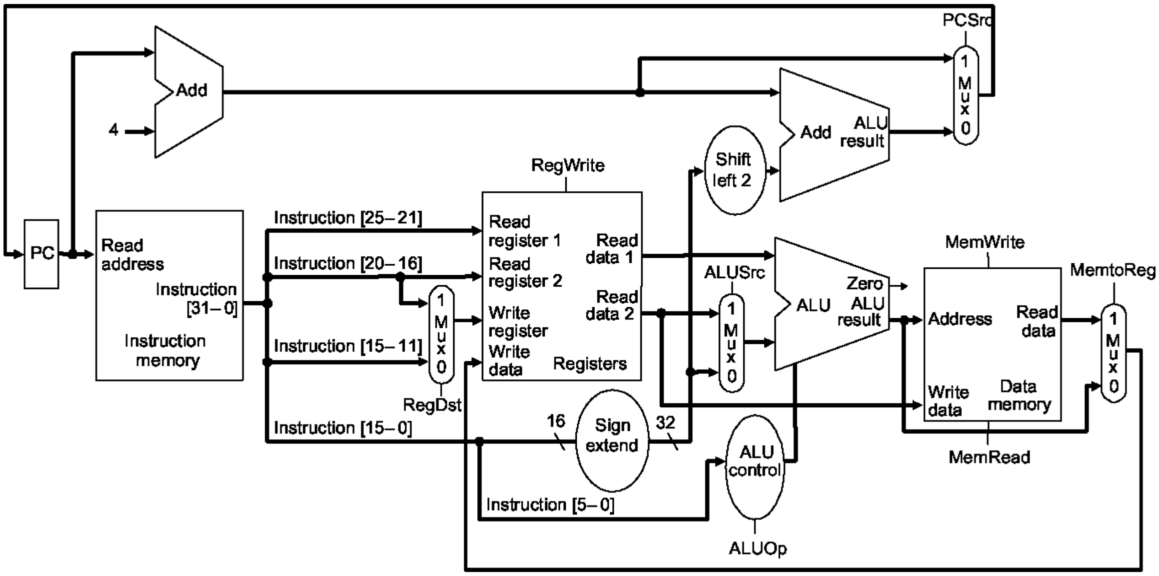
□ **`double a=1, b=268435456, r; r=(a+b)-b;`**

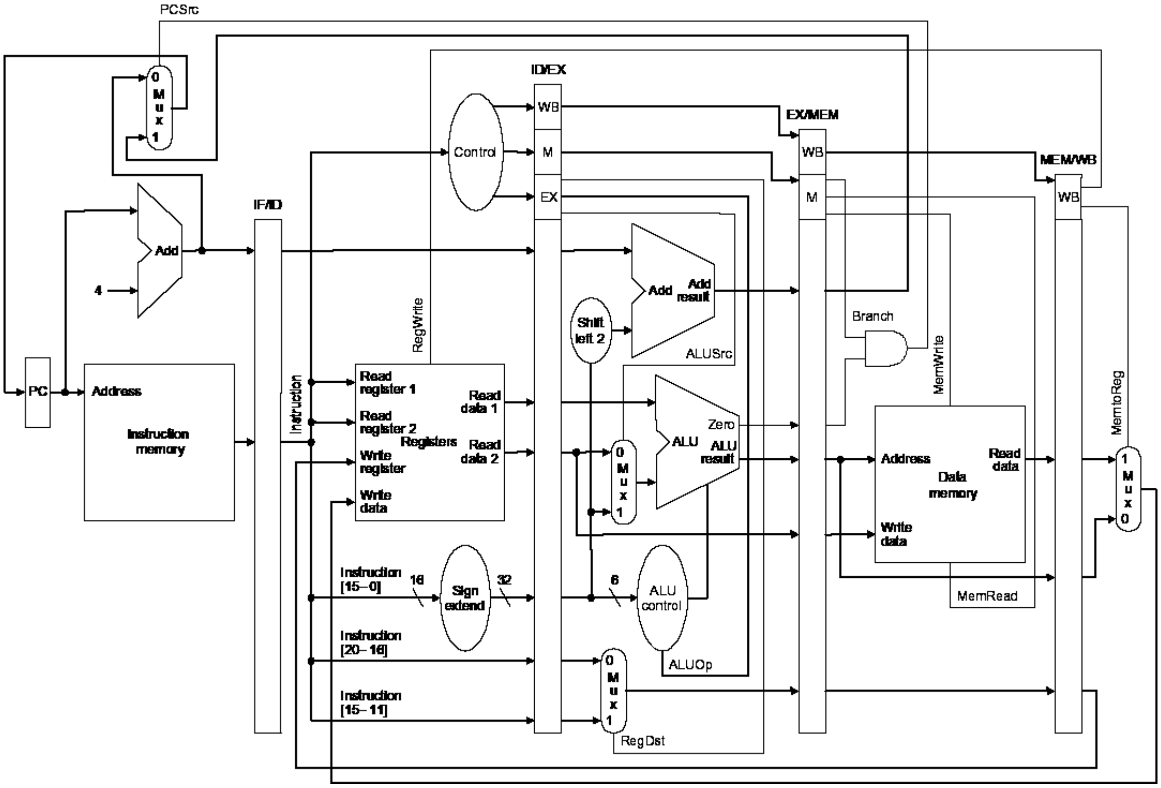
21. Given the declarations **`int a[N]; int i;`**, the following two C program fragments are found to execute **in almost exactly the same amount of time** using a modern PC. You suspect that the compiler may have optimized something away, but looking at the assembly code confirms that code (A) really does touch 8 times more data than code (B). Prof. Dietz says he isn't surprised because of how the cache works. Explain.

```
(A) for (i=0; i<N; i=i+1) a[i] = 0;
(B) for (i=0; i<N; i=i+8) a[i] = 0;
```

22. Given the declarations **`int array[N][N]; int i, j;`** your code originally is written to access **`array[j][i]`** inside loops over values of **`i`** and **`j`**. Your code runs a bit slower than you'd like, so a friend suggests maybe you can improve performance by swapping the index values, using **`array[i][j]`** instead. Why?

The following two figures show single-cycle and pipelined versions of the same basic MIPS subset implementation, essentially the same ones we used in class). Note that some signals have their sense flipped in one diagram as compared to the other.


PCSrc
Add
4
ALU result
Add
1 Mux 0
Shift left 2
RegWrite
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
1 Mux 0
RegDst
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
Sign extend
32
ALUSrc
1 Mux 0
ALU
Zero
ALU result
ALU control
Instruction [5–0]
ALUOp
MemWrite
Address
Read data
Write data
Data memory
MemRead
MemtoReg
1 Mux 0


PCSrc
0 Mux 1
IF/ID
ID/EX
EX/MEM
MEM/WB
Control
WB
M
EX
WB
M
WB
Add
4
PC
Address
Instruction memory
Instruction
RegWrite
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
Shift left 2
Add
Add result
ALUSrc
0 Mux 1
ALU
Zero
ALU result
Branch
MemWrite
Address
Read data
Write data
Data memory
MemRead
MemtoReg
1 Mux 0
Instruction [15–0]
16
Sign extend
32
6
ALU control
ALUOp
Instruction [20–16]
Instruction [15–11]
0 Mux 1
RegDst