Instructions

Format: This exam consists of 5 (or 4) multiple choice questions for 12 lectures in the course. You may have to do some calculations in order to determine the correct answer. The motivation for having so many questions is to get a more accurate assessment of your knowledge of the material.

Be sure to circle the answers on the page titled "answers page" or your answers will not be graded. If you choose not to answer a question do not circle a letter.

YOU MUST RETURN ALL PAGES.

Material: You are allowed a calculator and a single page of help

Good luck!

Answers page.
Only answers circled here will be graded.

1. ISA 1

1.1.	A	B	C	D
1.2.	A	B	C	D
1.3.	A	B	C	D
1.4.	A	B	C	D
1.5.	A	B	C	D

2. ISA 2

2.1.	A	B	C	D
2.2.	A	B	C	D
2.3.	A	B	C	D
2.4.	A	B	C	D
2.5.	A	B	C	D

3. Computer Arithmetic

3.1.	A	B	C	D
3.2.	A	B	C	D
3.3.	A	B	C	D
3.4.	A	B	C	D
3.5.	A	B	C	D

4. Logic

4.1.	A	B	C	D
4.2.	A	B	C	D
4.3.	A	B	C	D
4.4.	A	B	C	D
4.5.	A	B	C	D

5. Processor Control and Datapath

5.1.	A	B	C	D
5.2.	A	B	C	D
5.3.	A	B	C	D
5.4.	A	B	C	D
5.5.	A	B	C	D

6. Pipelining

6.1.	A	B	C	D
6.2.	A	B	C	D
6.3.	A	B	C	D
6.4.	A	B	C	D
6.5.	A	B	C	D

7. Hazards

7.1.	A	B	C	D
7.2.	A	B	C	D
7.3.	A	B	C	D
7.4.	A	B	C	D
7.5.	A	B	C	D

8. Branch Prediction and Exceptions and Interrupts

8.1.	A	B	C	D
8.2.	A	B	C	D
8.3.	A	B	C	D
8.4.	A	B	C	D
8.5.	A	B	C	D

9. Input/Output

9.1.	A	B	C	D
9.2.	A	B	C	D
9.3.	A	B	C	D
9.4.	A	B	C	D
9.5.	A	B	C	D

10. Caches

10.1.	A	B	C	D
10.2.	A	B	C	D
10.3.	A	B	C	D
10.4.	A	B	C	D
10.5.	A	B	C	D

11. Virtual Memory

11.1.	A	B	C	D
11.2.	A	B	C	D
11.3.	A	B	C	D
11.4.	A	B	C	D
11.5.	A	B	C	D

12. Parallelism

12.1.	A	B	C	D
12.2.	A	B	C	D
12.3.	A	B	C	D
12.4.	A	B	C	D

1. ISA 1

1. What is the minimum number of simultaneous reads and writes needed for a register file to work with the MIPS ISA?
   a. 0 writes, 1 read
   b. 1 write, 1 read
   c. 1 write, 2 reads
   d. 2 writes, 2 reads
2. How does the register file change after executing "addi R0, R1, 1"?
   a. R0 == 0
   b. R0 has R0+R1+1
   c. R1 has R0+1
   d. R0 has R1+1
3. What instructions go at XXX and YYY to accomplish: if (R3==R4) then R5=2 else R5=1?
   XXX R3, R4, labelA
   addi R5, R0, 1
   YYY R0, R0, labelB
   labelA:
   addi R5, R0, 2
   labelB:
   a. XXX=bne YYY=j
   b. XXX=beq YYY=j
   c. XXX=bne YYY=beq
   d. XXX=beq YYY=beq
4. What address does the following code load?
   addi R2, R0, 12
   addi R3, R2, 24
   lw R2, 12(R3)
   a. 12
   b. 24
   c. 36
   d. 48
5. The standard MIPS ISA contains how many usable general-purpose (not special) registers?
   a. 24
   b. 29
   c. 31
   d. 32

2. ISA 2

1. Why is the instruction "bne R2, 5, loop" not possible in the MIPS ISA?
   a. Because R-type instructions only have enough immediate field bits for one constant
   b. Because J-type instructions don't have any bits for registers
   c. Because I-type instructions need a constant offset, not a string like "loop"
   d. Because I-type instructions don't have enough immediate field bits for two constants
2. What is stored into $ra when a jal is executed?
   a. PC
   b. PC+4
   c. The address specified by the jal instruction
   d. The previous value of the $ra register
3. A J-type instruction has 6-bits for the opcode and 26-bits of immediate field. It works by:
   a. Using the immediate as a signed offset to the current PC (imm <<2 + PC)
   b. Using the immediate as a signed offset to the next PC (imm<<2 + PC+4)
   c. Replacing the lower bits of the next PC directly (imm <<2 overwrites bits 2 to 28 of PC+4)
   d. Directly using the immediate value for the new PC (next PC is the immediate value)
4. What would happen if the ori instructions used a sign-extended value?
   a. The upper 16 bits would sometimes be all 0s
   b. The upper 16 bits would sometimes be all 1s
   c. The lower 16 bits would sometimes be all 0s
   d. The lower 16 bits would sometimes be all 1s
5. Why do we only need a few bits for relative branches?
   a. Branches are usually to nearby code
   b. Longer branches can be done with jump instructions
   c. The branch address is signed so we can jump forwards and backwards
   d. All of the above

3. Computer Arithmetic

1. What is the range of a 6-bit signed magnitude binary number?
   a. -31 to +31
   b. -32 to +32
   c. -64 to +64
   d. -63 to +63
2. What is the output of a 3-bit adder when adding the unsigned numbers 6 and 5?
   a. 3
   b. 7
   c. 11
   d. undefined
3. What is the range of a 5-bit two's complement number?
   a. -16 to +15
   b. -16 to +16
   c. -32 to +32
   d. -32 to +31
4. How do you negate a 2's complement number?
   a. Invert
   b. Invert and add 1
   c. Invert and add 2
   d. Shift right and add a 1 to the most significant digit
5. How many additions are needed for a serial multiplier to multiply 1010 times 1000?
   a. 1
   b. 2
   c. 4
   d. 16

4. Logic

1. What do you use to transform 8 one-hot inputs into a 3-bit binary number?
   a. Decoder
   b. Encoder
   c. Multiplexer
   d. Demultiplexer
2. A memory array uses what to choose the right row based on the address?
   a. Decoder
   b. Encoder
   c. Multiplexer
   d. Demultiplexer
3. What does the combinational logic in a finite state machine do?
   a. Determine when to stop
   b. Calculate the current state from the state output and the inputs
   c. Determine the next state from the inputs and current state
   d. None of the above
4. How many control bits are needed to control a Multiplexor with 8 inputs?
   a. 2
   b. 3
   c. 8
   d. 2^8
5. Which is faster, and why? DRAM or SRAM?
   a. DRAM: each bit is stored in a powered feedback loop so it can be read quickly
   b. DRAM: each bit is stored in a passive capacitor so it can be read quickly
   c. SRAM: each bit is stored in a powered feedback loop so it can be read quickly
   d. SRAM: each bit is stored in a passive capacitor so it can be read quickly

5. Processor Control and Datapath
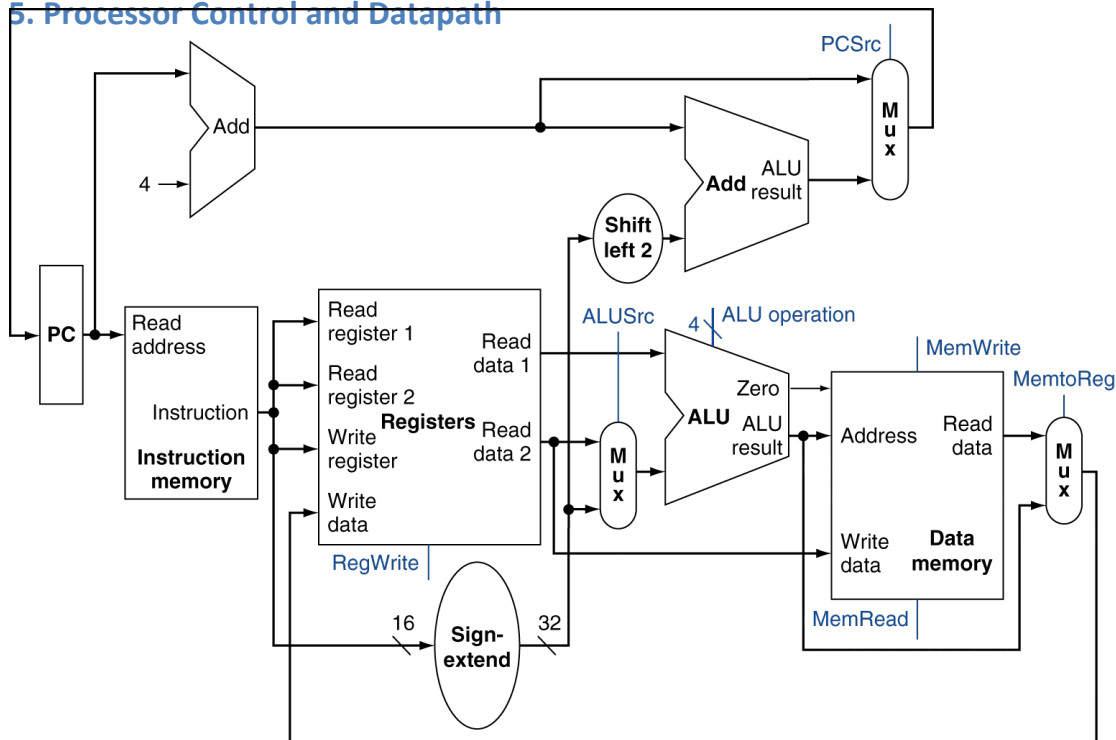
1. If the ALUSrc mux was removed, and Register Read data 2 was connected directly to the ALU, what instructions could no longer work?
    a. R-type (register-register instructions)
    b. I-type (instructions with immediate operands)
    c. J-type (jump/branch instructions)
    d. All of the above
2. Why do we have the "shift left 2" before the branch adder?
    a. Because branch offsets are specified in bytes
    b. Because branch offsets are specified in instructions
    c. Because we can only jump in multiples of 4 instructions.
    d. None of the above
3. Which parts of the processor need a clock signal?
    a. PC
    b. Instruction memory
    c. Register file
    d. All of the above
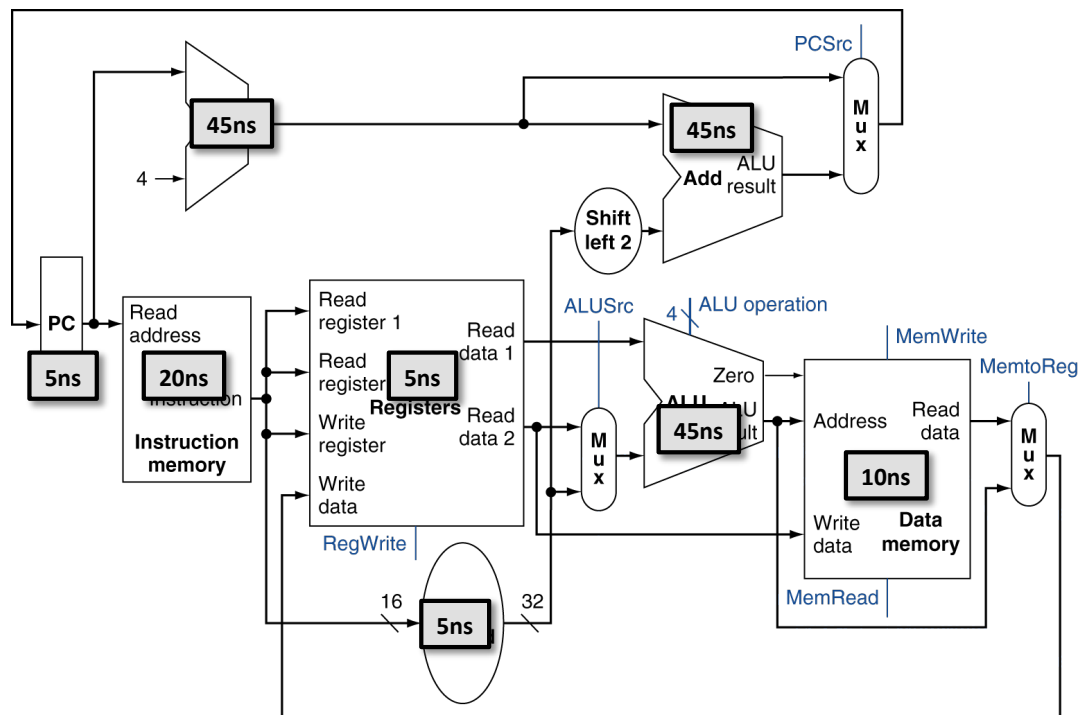
4. What is the maximum clock speed of the processor above? (The times listed for the state elements are the time it takes to read or write.)
    a. 85ns
    b. 90ns
    c. 100ns
    d. 200ns
5. What is the ALU operation for a store word instruction?
    a. Subtract
    b. Add (from register file)
    c. Add (from immediate)
    d. Write to memory

6. Pipelining

1. A multi-cycle processor has 3 instruction classes that take 1, 2, and 4 cycles to complete. This processor is converted into a pipelined processor. How long do the instruction classes take to complete in the pipelined version?
   a. 1 cycle because a pipelined processor finishes one instruction per cycle
   b. 4 cycles because all instructions take the same time in a pipeline
   c. 5 cycles because you need a write back stage
   d. None of the above

2. What is the key to getting performance out of a pipeline?
   a. Predicting branches
   b. Avoiding branch delay slots
   c. Keeping the pipeline full
   d. Re-ordering instructions

3. A processor takes 100ns for the longest instruction and pipeline registers that take 2ns. What is the lowest cycle time you could get for the processor if the processor can be divided up into an infinite number of equal stages?
   a. ~0ns
   b. ~2ns
   c. ~50ns
   d. ~100ns

4. A processor takes 180ns for the longest instruction. The processor is pipelined with 6 (equal) stages using pipeline registers that take 10ns. How much more or less time does each instruction take compared to the un-pipelined processor? (You can assume there are 6 pipeline registers for 6 stages.)
   a. 140ns shorter
   b. same
   c. 10ns longer
   d. 60ns longer

5. What do load word instructions do during the EX stage of the MIPS pipeline?
   a. Nothing
   b. Add
   c. Forward pipeline registers
   d. Set the MemWrite control

7. Hazards

1. What did double-pumping the register file fix?
    a. The need to read the register file early for branches
    b. The need to get data to the register file early for all dependent instructions
    c. The need to read and write the register file at the same time
    d. All of the above
2. Why do we have problems with data hazards
    a. Because writeback happens at the end of the pipeline
    b. Because the ISA promises atomic execution
    c. Because we only have one register file
    d. Because the ISA promises sequential execution
3. How many register values need to be forwarded in the following code?
    add R3, R4, R5
    sub R2, R3, R4
    beq R2, R3, done
    a. 1
    b. 2
    c. 3
    d. 4
4. Why do we need a load delay slot here?
    lw R7,12(R5)
    sub R2, R7, R5
    a. The data from MEM is available at the end of the cycle but needed by EX at the start of the cycle
    b. The data from MEM may take much longer due to a cache miss
    c. The second instruction needs the data from first instruction in the cycle before it accesses the memory
    d. All of the above
5. Why can't we eliminate the branch delay slot by moving the branch computation further forward in the pipeline?
    a. We need to read the register file first, and that is in the ID stage
    b. We need to do a comparison, and that is in the EX stage
    c. We need to decode the instruction, and that is in the ID stage
    d. All of the above
        We need to read the register file first to get the data to compare, and that doesn't happen until the ID stage.

8. Branch Prediction and Exceptions and Interrupts

1. What happens if you have more branches in your program than you have spaces in your branch predictor?
   a. Can't make a prediction
   b. Can't run your program
   c. May predict incorrectly
   d. Always predict incorrectly
2. What is the benefit of a 2-bit predictor over a 1-bit predictor?
   a. Predicts loop jumps better
   b. Predicts all jumps better
   c. Doesn't get fooled by a single wrong prediction
   d. None of the above
3. Which is better: a processor with a 16-cycle branch penalty and a branch predictor that is 90% accurate or a processor with an 8-cycle branch penalty and a branch predictor that is 80% accurate? (assume you can otherwise keep the pipeline full.)
   a. 16-cycle/90% accurate
   b. 8-cycel/80% accurate
   c. Same
   d. Can't tell without knowing the percentage of branches
4. What happens to later instructions in the pipeline when an exception occurs?
   a. They are paused until after the exception is handled
   b. They are killed
   c. They continue
   d. They continue unless they also have an exception
5. Was the decision to have a branch delay slot in MIPS a good one?
   a. Yes, because it enabled early MIPS processors to be easier to build
   b. Yes, because compiler technology can easily fill it
   c. No, because it is a limitation for newer MIPS processors
   d. No, because compilers can't reliably fill it

9. Input/Output

1. What are the two key performance metrics for I/O?
   a. Throughput and latency
   b. Throughput and longevity
   c. Latency and longevity
   d. Density and longevity
2. Direct Memory Access (DMA) allows the I/O device to do what?
   a. Directly trigger interrupts when data is available
   b. Directly get access to the bus master
   c. Directly transfer data to the CPU's memory
   d. Directly modify the program's data
3. Why don't we use DRAM for data storage instead of disks and flash?
   a. Too expensive
   b. Too slow
   c. Too unreliable
   d. All of the above
4. What is the benefit of DMA?
   a. Improves the speed of data transfers
   b. Reduces the workload of the processor
   c. Simpler to use than polling
   d. Lower overhead compared to interrupts
5. What is the benefit of polling?
   a. Simpler to program
   b. Uses more the processor for I/O
   c. Higher data throughput
   d. All of the above

10. Caches

1. Why do we have a memory hierarchy?
    a. To make the memory look big
    b. To make the memory look fast
    c. Because we can't afford tons of SRAM
    d. All of the above
2. What data is stored in an LRU cache?
    a. The most important data
    b. The most recently used data
    c. The fastest data
    d. The register data
3. What is the relative growth of memory speed vs. processor speed?
    a. Processors are getting faster more rapidly than memory
    b. Memory is getting faster more rapidly than processors
    c. They're both getting faster at the same rate
    d. Depends on the processor
4. What data will be in a 4-entry, direct-mapped cache with one byte per line after the following memory accesses?
    address: 0, 1, 2, 3, 4, 5, 3, 2, 1
    a. 1, 2, 3, 4
    b. 1, 2, 3, 5
    c. 1, 3, 4, 5
    d. 1, 2, 4, 5
5. What data will be in a 4-entry, fully-associative, LRU cache with one word per line after the following memory accesses?
    address: 0, 1, 2, 3, 4, 5, 3, 2, 1
    a. 1, 2, 3, 4
    b. 1, 2, 3, 5
    c. 1, 3, 4, 5
    d. 1, 2, 4, 5

11. Virtual Memory

1. How does virtual memory make software more portable?
    a. It allows programs to be put at different places in memory
    b. It allows programs to use more memory than is available
    c. It allows programs to separated from each other
    d. All of the above
2. What do you do when you can't find a PTE in the TLB?
    a. Load the page from the cache
    b. Load the page from disk
    c. Look in the full page table in memory
    d. Cause a memory protection exception
3. Which data is shared between processes 1 and 2?

   Process 1 PT (VA→ PA)   Process 2 PT (VA→ PA)
   1 → 12   14 → 4
   2 → 13   13 → 6
   3 → 24   3 → 3
   7 → 5   4 → 25
   8 → 3   5 → 7

    a. P1 8 shares with P2 3
    b. P1 3 shares with P2 3
    c. P1 7 shares with P2 5
    d. No sharing
4. If a machine does not translate 13 bits of the virtual address, how large are its pages?
    a. 4kB
    b. 8kB
    c. 2MB
    d. 2GB
5. Do page tables need a dirty bit?
    a. No, they don't store data, just VA-to-PA mappings
    b. No, they are read only
    c. Yes, the bit indicates if the data is on disk
    d. Yes, the bit indicates if the page has been changed since it was loaded from disk

12. Parallelism

1. Why do we do parallelism?
    a. Easier
    b. Faster
    c. Fewer transistors
    d. All of the above
2. Why don't we just increase the clock speed?
    a. Can't build faster circuits
    b. Can't afford more power
    c. Can't make memory faster
    d. All of the above
3. How much faster can my program run if I have 2000 cores and 10% of the program cannot be parallelized?
    a. 10x
    b. 90x
    c. 200x
    d. 2000x
4. How do locks help with parallel programs?
    a. They make sure data is updated across all processors
    b. They serve as an indication that another processor is changing the data
    c. They force the cache to keep the data synchronized
    d. Something else

Make sure you circled the answers on the answer page. Answers on other pages will NOT be graded.