Instructions
Format: This exam consists of 6 multiple choice questions for each of the 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. Feel free to detach the answers page from the remainder of the exam and only turn it in.

Grading: To discourage guessing, each incorrect answer will be worth $-1/3$ point, while each correct answer is worth $+1$ point. I.e., if you guess randomly, the expected score is 0.

Material: You are allowed a calculator and one double-sided, hand-written A4 sheet of notes during the exam.

Good luck!
0. Optional anonymous background questions

0.1. A B C
0.2. A B C
0.3. A B C

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
1.6. 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
2.6. 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
3.6. 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
4.6. 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
5.6. 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
6.6. 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
7.6. 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
8.6. 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
9.6. 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
10.6. 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
11.6. 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
12.5. A B C D
12.6. A B C D

Feel free to detach the remaining pages of the exam and turn in only your answers.
0. OPTIONAL anonymous background questions
(These are not graded and will not affect your grade on the exam. They are for me to be able to correlate what helps students prepare for the exam. Feel free to not answer these questions if you do not want to.)

1. How much of the book did you read?
   a. All
   b. Some
   c. None

2. How many of the practice sessions did you miss/turn in late?
   a. Several
   b. One
   c. None

3. How many of the previous years’ exams did you look at?
   a. All
   b. Some
   c. None

1. ISA 1

1. Which of the following addresses is word-algined?
   a. 0110 1010 0110 1000 0110 1010 0110 1001
   b. 0000 0000 1000 1010 0110 1000 1010 1111
   c. 0000 0000 0000 1010 0110 1000 1010 1110
   d. 0000 0000 1000 1010 0110 1010 1010 1100

2. What is the value in R1 after the following program? (starts with R0=0, R1=2, R2=2)
   add R1, R1, R2
   add R0, R0, R1
   add R1, R0, R1
   a. 0
   b. 2
   c. 3
   d. 6

3. Why do we want our processor to appear to be atomic and sequential?
   a. Faster
   b. Easier to build
   c. Easier to understand
   d. Smaller

4. What does the following code put in R3?
   lw R1, 12(R0)
   lw R2, 16(R0)
   sll R1, R1, 16
   slr R2, R2, 16
   or R3, R1, R2
   a. Zero
   b. The ORed sum of the data at addresses 12 and 16
   c. The ORed sum of the data at addresses 12 and 16 multiplied by $2^{16}$
   d. The word starting at address 14

5. What address does the following code load?
   addi R2, R0, 12
   addi R3, R2, 24
   lw R2, 12(R3)
   a. 12
6. 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

2. ISA 2
   1. If we added a new instruction format to MIPS that only specified one register, how large a constant could it hold?
      a. 16 bits
      b. 17 bits
      c. 21 bits
      d. 32 bits
   2. 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
   3. 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
   4. main() calls procedure A(), which runs for a while and then calls procedure D(). A() uses registers s0, s1, s2, t0, t1, and t2. Which registers need to be saved right before A() calls D()?
      a. ra, s0, s1, s2, t0, t1, t2
      b. s0, s1, s2, t0, t1, t2
      c. ra, t0, t1, t2
      d. t0, t1, t2
   5. main() uses t0, t1, s0, s1, B() uses t4, s3, s4, C() uses t1, t2, t3, t4, s0, s5. How many words on the stack are needed when main() calls B() and B() calls C()?
      a. 10
      b. 13
      c. 15
      d. 16
   6. What is an advantage of a memory-register machine over load-store-register machines?
      a. Easier to build
      b. Simpler for the compiler
      c. Simpler assembly code
      d. Denser code

3. Computer Arithmetic
   1. How many additions are needed in a serial multiplier to multiply 1101 by 1000?
a. 1  

b. 3  

c. 4  

d. 16  

2. What is the biggest problem with signed magnitude numbers?  
   a. Subtraction is messy  
   b. We have two zeros  
   c. We can’t detect overflow  
   d. We use a bit for the sign  

3. What is the result of (−4)+(-6) for 4 bit two’s complement numbers?  
   a. 10111  
   b. 10110  
   c. 0111  
   d. Overflow  

4. What are the largest and smallest values you can represent with a 5-bit unsigned fixed-point number?  
   a. 31 to 0.3125  
   b. 31 to 1  
   c. 0.96875 to 0.03125  
   d. 15.75 to 0.25  

5. Why do we use normalized numbers in floating point?  
   a. To force only one representation of each number  
   b. To provide an exponential scale  
   c. To force the mantissa to start with a 1  
   d. To avoid negative mantissas  

6. What do you expect the result to be of the following in a floating point format with a less than 28 bits for the mantissa? (2x10^{30}+5x10^{2})-2x10^{30}  
   a. 0  
   b. 2x10^{30}  
   c. 5x10^{2}  
   d. -2x10^{30}  

4. Logic  
   1. How can we use an XOR gate to determine overflow in two’s complement math?  
      a. XOR all output bits together: 1 = overflow  
      b. XOR carry in to MSB and carry out from MSB: 1 = overflow  
      c. XOR sign bit of input 1 and sign bit of input 2: 0 = overflow  
      d. XOR the carry bits together: 1 = overflow  

   2. How many inputs would a MUX have that has 3 select bits?  
      a. 3  
      b. 6  
      c. 8  
      d. 9  

   3. Why is the D0 input to an encoder not connected to the logic that determines the output?  
      a. D0 is a dummy input  
      b. If D0 is true, then we want the outputs to be all false  
      c. D0 is an illegal input
4. Why does a master-slave latch need two latches?
   a. To prevent the output from going through to the input
   b. To only capture the input on the edge of the clock
   c. To only make the output visible on the edge of the clock
   d. To use feedback to keep the value

5. What kinds of circuits are needed for a counter?
   a. Combinational
   b. State register
   c. Full adder
   d. All of the above

6. What determines the clock speed of a circuit?
   a. The slowest logic gate
   b. The slowest path through the combinational logic
   c. The slowest path through the combinational logic and the register
   d. The number of logic gates in the next stage logic

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
   b. I-type
   c. J-type
   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 don’t want to change the last two bits of the instruction address
   d. Because we can only jump in multiples of 4 instructions.

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. What are the settings for the ALUSrc and MemtoReg MUXes for a load word instruction?
   a. ALUSrc = Read data 2, MemtoReg = Read data
   b. ALUSrc = Read data 2, MemtoReg = ALU result
   c. ALUSrc = Sign-extended, MemtoReg = Read data
   d. ALUSrc = Sign-extended, MemtoReg = ALU result

6. Pipelining
   1. How much faster does a processor run if we divide each instruction into 7 parts and run the clock 7x faster?
      a. 1x
      b. 7x
      c. 49x
      d. Depends on the branch predictor
   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 180ns for the longest instruction. The processor is pipelined with 6 (equal) stages using pipeline registers that take 10ns. What percentage of the resulting cycle time is used for computation?
   a. 25%
   b. 75%
   c. 85%
   d. 95%

4. 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

5. 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 N pipeline registers for N stages.)
   a. 140ns shorter
   b. same
   c. 10ns longer
   d. 60ns longer

6. 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 prefer to forward data instead of stalling the processor?
      a. Forwarding is faster than stalling
      b. Forwarding logic is simpler than stall logic
      c. Stalling requires changing the compiler
      d. Stalling requires defining special bubble instructions
5. 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

6. 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

8. Branch Prediction and Exceptions and Interrupts
   1. What is the CPI of a 1 CPI processor if it has 2 branch delay slots, 20% of the instructions
      are branches, and the branch delay slots can be filled only 50% of the time?
      a. 1.1
      b. 1.2
      c. 1.4
      d. 1.5
   2. 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
   3. What is the benefit of a 2-bit predictor over a 1-bit predictor?
      a. Faster
      b. Predicts backwards jumps better
      c. Doesn’t get fooled by a single wrong prediction
      d. Always works better
   4. 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-cycle/80% accurate
      c. Same
      d. Can’t tell without knowing the percentage of branches
   5. What decides to jump to the interrupt handler code on an interrupt?
      a. The operating system
      b. The user code
      c. The processor
      d. The I/O device
   6. 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
9. Input/Output

1. Why are busses a problem?
   a. Hard to make wires
   b. Hard to connect wires
   c. Hard to find space for wires
   d. Hard to keep the wires synchronized

2. What is replacing busses, and why?
   a. Serial links – because they can connect many devices
   b. Serial links – because they don’t need to be synchronized
   c. Serial links – because they are simpler
   d. Optical links – because they are easier to build

3. 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

4. What is the benefit of polling?
   a. Simpler to program
   b. Lower overhead
   c. Higher throughput
   d. All of the above

5. What is the benefit of interrupt-driven I/O?
   a. Simpler to program
   b. Lower overhead
   c. Higher throughput
   d. All of the above

6. How does Ethernet share a single serial bus?
   a. Has special times for each device
   b. Checks for data corruption and re-tries
   c. Uses encryption to keep data safe
   d. Really uses one wire per device

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
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

6. Which cache will have the highest percentage of data bits for a 32bit machine?
   a. 4-entry, direct-mapped, 64 byte line
   b. 4-entry, fully-associative, 64 byte line
   c. 128-entry, direct-mapped, 64 byte line
   d. 128-entry, fully-associative, 64 byte line

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 happens when the program with the following page table tries to write to address 14?

   VA → PA on disk access bits
   1 → 8  0 read/write
   2 → 9  0 read/write
   3 → 14 0 read only
   13 → 15 1 read/write
   14 → 7 1 read only
   a. The data is loaded from the disk
   b. The PTE is loaded from the page table into the TLB
   c. An memory protection exception is generated
   d. The data is written to physical address 7

3. Which TLB/cache arrangement allows you to know if you have a hit in the cache without checking the TLB?
   a. Virtually indexed, virtually tagged
   b. Virtually indexed, physically tagged
   c. Physically indexed, virtually tagged
   d. Physically indexed, physically tagged

4. Which TLB/cache arrangement allows you to access the TLB in parallel with the cache? (And requires it on every access.)
   a. Virtually indexed, virtually tagged
   b. Virtually indexed, physically tagged
   c. Physically indexed, virtually tagged
   d. Physically indexed, physically tagged

5. 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

6. Which data is shared between processes 1 and 2?
   Process 1 PT (VA → PA)       Process 2 PT (VA → PA)
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. If I have 100M numbers I want to add up. What is the difficult part about parallelizing this?
   a. Dividing up the work across the processors
   b. Dealing with the data that overlaps between processors
   c. Synchronizing when processors are done
   d. All of the above

4. 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

5. How do locks help with parallel programs?
   a. The protect a memory location so only one processor can change it at once
   b. They make sure data is updated across all processors
   c. They serve as an indication that another processor is changing the data
   d. They force the cache to keep the data synchronized

6. A processor can do two ALU operations (add/sub/etc.) at the same time on two subsequent instructions. A program has 40% load/stores, and 60% ALU operations. 10% of the instructions depend on a subsequent instruction. What is the best IPC you would expect?
   a. 1.5
   b. 1.6
   c. 1.8
   d. 1.9

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