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!

Note: Please take this exam seriously and pass it so you do not have to take future makeup exams! While it is essential to give students appropriate chances to demonstrate their knowledge, producing and grading an exam consumes significant teaching resources that could be spent otherwise improving the course. For example, this exam will cost the IT department roughly 10,000SEK in teaching time to prepare, administer, and grade.
0. Optional anonymous background questions
0.1. A B C
0.2. A B C
0.3. A B C D
0.4. Program: ________

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.

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

2. How many practice exams did you do?
   a. None
   b. One
   c. Several

3. How much time did you spend studying for this exam?
   a. None
   b. <2 hours
   c. 2-8 hours
   d. >8 hours

1. ISA 1

1. Why do processors do all their computations on registers?
   a. Main memory is too slow
   b. Easier to address fewer locations
   c. A and B
   d. None of the above

2. Can you write to the PC register?
   a. Yes, same as any other register
   b. Yes, but you have to use special instructions
   c. Yes, but not in a general purpose manner
   d. No

3. What is this code equivalent to?
   \[ \text{ld R1, (R2)} \]
   \[ \text{sll R1, R1, 24} \]
   \[ \text{slr R1, R1, 24} \]
   a. \[ \text{ld R1, 24(R2)} \]
   b. \[ \text{ld R1, 3(R2)} \]
   c. \[ \text{lbu R1, 3(R2)} \]
   d. Need to know the value in R2 to determine

4. How many memory locations are needed to fill the register file?
   a. 31
   b. 32
   c. 124
   d. 128

5. What is the difference between instructions and data stored in memory?
   a. They are stored in different memories
   b. They can only be accessed with different instructions
   c. Instructions cannot be written while data can
   d. There is no difference

6. Why is the PC incremented by 4?
   a. Each instruction is 4 bytes long and the memory is word-addressed
   b. Each instruction is 4 bytes long and the memory is byte-addressed
   c. The increment depends on the address for conditional jumps
   d. The last two bits of the instruction are always 00
2. ISA 2

1. Why is it good to have consistent instruction encoding?
   a. Smaller instructions
   b. Easier to handle constants
   c. Easier to decode instructions
   d. All of the above

2. A designer adds a “branch if register is zero” instruction to MIPS. (bzero REG, address) How far can this instruction jump if it uses a modified I-format encoding?
   a. \(-2^{14} \text{ to } 2^{14} - 1\)
   b. \(-2^{16} \text{ to } 2^{16} - 1\)
   c. \(-2^{20} \text{ to } 2^{20} - 1\)
   d. \(-2^{21} \text{ to } 2^{21} - 1\)

3. A designer adds a “load big constant to register” instruction to MIPS. (lbc REG, immediate) This instruction uses a modified I-format encoding. How much does this speed up loading a single 32-bit immediate?
   a. No change
   b. 31% faster
   c. 65% faster
   d. 2x faster

4. Why does the MIPS ISA define a stack?
   a. So procedures do not overwrite each other’s registers and lose data
   b. So procedures can overwrite each other’s registers without losing data
   c. Jump and link can’t work without a stack
   d. All of the above

5. A designer comes up with a new instruction: branch-equal-constant that uses a modified I-encoding where the 2nd register is used to encode the branch address and the immediate field is used to encode the constant. What does this graph tell you about how useful this instruction would be? (floating point is the bar that is highest at 2.)
   a. No difference
   b. Each constant branch will be twice as fast
   c. ~85% of floating point and ~62% of integer code could use it to save 1 cycle for each branch
   d. Can’t tell because we don’t know how many of the branches compare to constants

6. Does a procedure need to save $a1 and $v1 on the stack when it calls a sub-procedure?
   a. Yes, they are caller save
   b. Depends on whether they are needed again after the sub-procedure returns
3. Computer Arithmetic

1. What value can the binary number 10011 represent?
   a. 19
   b. -3
   c. -13
   d. All of the above

2. A company develops two new types of transistors: LF transistors are very large, but very fast, and SS transistors are very small but slow. How would you design a multiplier for each type of transistor to match its characteristics?
   a. LF: serial, SS: serial
   b. LF: serial, SS: parallel
   c. LF: parallel, SS: parallel
   d. LF: parallel, SS: serial

3. How many bits do you need to store the weight in a scale that can measure up to 2kg with an accuracy of 0.01kg?
   a. 11 bits
   b. 2.7 bits
   c. 8 bits
   d. You can’t represent 0.01 perfectly with binary numbers

4. What would the following 2’s compliment math compute: A+!B
   a. A-B
   b. A+B-1
   c. A-B-1
   d. Garbage

5. Why do you shift the mantissa in floating point addition?
   a. To make the exponents match
   b. To line up the binary points
   c. To make the mantissas have the same magnitude so you can add them
   d. All of the above

6. Why do computers use binary numbers instead of tertiary (3 levels) or analog (infinite levels) signals?
   a. Can’t design non-binary circuits
   b. Can integrate lots of circuits without noise adding up
   c. A and B
   d. None of the above

4. Logic

1. What is true about a logic circuit derived using a Karnaugh map vs. one derived from a truth table?
   a. Karnaugh maps always produce simpler circuits
   b. Karnaugh maps usually produce simpler circuits
   c. Karnaugh maps always produce the same circuit
   d. Truth tables always produce simpler circuits

2. Which of the following instructions will store 0 in R3?
   a. and R3, R3, R3
   b. or R3, R3, R3
   c. xor R3, R3, R3
   d. None of the above

3. What would happen if you replaced the decoder in the SRAM design with an encoder?
   a. It would activate different rows, but it would still work
b. It would activate different rows, and would not work

c. It would activate multiple rows, but would still work

d. It would activate multiple rows, and would not work

4. What signals do you put into A and B to build a latch that stores data on the falling edge of the clock?
   a. A=CLK, B=!CLK
   b. A=CLK, B=CLK
   c. A=!CLK, B=CLK
   d. A=!CLK, B=!CLK

5. A designer makes a mistake on a positive-edge triggered flip-flop and makes the inverters in the second latch 10x bigger (10x more powerful, and about 3x faster) than the first latch. What happens? (Assume the clocks are connected correctly.)
   a. The latch works the same as before, but it is a faster due to the more powerful inverter
   b. The first latch cannot overwrite the value in the second latch so the flip-flop doesn’t work
   c. The second latch cannot overwrite the value in the first latch so the flip-flop doesn’t work
   d. None of the above

6. Why are DRAMs so much cheaper than SRAMs?
   a. They do not need decoders to select the data because they do not share output wires since the charge on the capacitors is so small
   b. Each bit is slower because it only has a small amount of charge from a capacitor which has to drive the output
   c. Each bit is smaller because it is stored in a trench deep down into the silicon instead of several transistors spread out on top
   d. All of the above
5. Processor Control and Datapath

1. Why is it acceptable (performance-wise) to be able to do two reads and one write at the same time to the register file, but the memory can only do one read or one write at once?
   a. The memory is not used by most instructions, so it is okay to share the read and write ports
   b. The register file is small, so the complexity of having many read and write ports is manageable
   c. The memory is not read and written by the same instruction, so it is okay to share the read and write ports
   d. All of the above

2. Which MUX would you disable (that is, permanently set to only one input) to break all I-type instructions but not affect R-type or J-type ones?
   a. The top MUX
   b. The middle MUX
   c. The right MUX
   d. Disabling one MUX is not enough

3. The R-format instructions use the third register address to determine where to write their results into the register file. The I-format instructions use the second register address to determine where to write their results into the register file. What is needed in the design to make this work?
   a. Nothing. Just send the register address bits to the Write register input and the correct register will be selected for each instruction type.
   b. A DEMUX to send the write data bus to the correct register depending on the instruction type.
   c. A MUX to choose which register address bits should be sent into the write register input depending on the instruction type.
   d. Control logic to delay the RegWrite signal on I-type instructions to make sure the data is in the right place.

4. What is the logic that determines whether PCSrc should select PC+4 (0) or the ALU result (1)?
   a. (Zero AND bne) OR (Zero AND beq)
b. \((!\text{Zero AND bne}) \text{ OR } (\text{Zero AND beq})\)

5. Which instruction(s) could not be supported by the datapath as shown above. (Note that the control logic is not shown in the figure.)
   a. \(\text{jr}\)
   b. \(\text{bne}\)
   c. \(\text{A and B}\)
   d. Both of the above are supported

6. Which of the following statements are true?
   a. \(\text{MemtoReg always selects the ALU result for R-type instructions}\)
   b. \(\text{RegWrite is always true for I-type instructions}\)
   c. \(\text{PCSrc always selects the ALU result for I-type instructions}\)
   d. All of the above

6. Pipelining

1. What is needed to get better performance from a pipelined processor?
   a. A higher clock speed
   b. Executing multiple instructions in parallel
   c. Keeping the pipeline full
   d. All of the above

2. What limits the clock speed for a non-pipelined processor?
   a. The fastest instruction path
   b. The average instruction path
   c. The slowest instruction path
   d. The slowest logic unit

3. A processor is pipelined 10 times and runs at 10x the clock speed with no overhead, but only manages to keep the pipeline half full. What is the speedup?
   a. 5x
   b. 10x
   c. 50x
   d. 100x

4. A single-cycle processor takes 100ns per instruction and can be perfectly divide up into pipeline stages and the pipeline can be kept completely full. However, each pipeline register has an overhead of 1ns. What degree of pipelining gives the best \textit{instruction throughput}?
   a. 1 stage (1 pipeline register at the end)
   b. 10 stages
c. 100 stages  
d. 1000 stages  
5. A single-cycle processor takes 100ns per instruction and can be perfectly divide up into pipeline stages and the pipeline can be kept completely full. However, each pipeline register has an overhead of 1ns. What degree of pipelining gives the best instruction latency?  
a. 1 stage (1 pipeline register at the end)  
b. 10 stages  
c. 100 stages  
d. 1000 stages  
6. How many stages do processors that run at multiple GHz have these days?  
a. 1-10  
b. 11-25  
c. 26-50  
d. 50+  

7. Hazards  
1. What would be a structural hazard?  
a. Needing a result that is in another part of the processor  
b. Needing a result that can’t be computed because the hardware is busy  
c. Computing a result for an instruction that should not have been executed  
d. Any of the above  
2. What would happen to performance if a pipeline could only read or write to the register file on each cycle?  
a. No change  
b. Better performance because the register file would be simpler and faster  
c. Worse performance because some instructions would have to stall  
d. Terrible performance because most instructions would have to stall  
3. The bcopy routine does a bit copy from one memory location to another. The code consists of the following loop. What forwarding paths are needed in the standard 5-stage MIPS pipeline to avoid data hazards in this code?  
   loop:  
   addi R4, R4, 4  # increment source address  
   addi R5, R5, 4  # increment destination address  
   lw R6, 0(R4)  # load source data  
   sw R6, 0(R5)  # store data to destination  
   bne R4, R7, loop # check if we are done with the copy  
   a. WB-EX, MEM-EX  
   b. WB-EX, WB-MEM  
   c. MEM-EX, WB-MEM  
   d. WB-EX, MEM-EX, WB-MEM  
4. Which of the following did we not have to add/move to reduce the branch delay slots from 3 to 1?  
a. An extra adder for the PC  
b. An extra comparator for the branch true logic  
c. An extra register in the IF/ID pipeline register for the PC  
d. An extra MUX in the fetch stage to select the next PC  
5. If a program has 20% branches with 1 branch delay slot per branch, and the compiler can find an instruction to put in the branch delay slot every other time, what is the performance slowdown due to the branch delay slot?  
a. 5%  
b. 10%  
c. 20%
6. Are branch delay slots a good idea?
   a. Yes, the compiler can find instructions to fill multiple slots
   b. Maybe, the tradeoff makes sense with simple processors and people can always recompile their code for new processors
   c. No, no one re-compiles software, so you have to emulate them in all future processors whether you need them or not
   d. It’s not clear which one of these is most true

8. Branch Prediction and Exceptions and Interrupts
   1. Branch prediction can solve control hazards by doing what?
      a. Predicting the future
      b. Forwarding the result
      c. Replicating the branch calculation hardware
      d. All of the above
   2. What prediction strategy does our initial MIPS pipeline use?
      a. No strategy
      b. Predict taken
      c. Predict not taken
      d. Randomly choose taken/not taken
   3. What do you have to do to instructions in the pipeline that are executed due to a missprediction?
      a. Nothing if they are memory instructions
      b. Prevent them from changing anything
      c. Insert instructions to reverse their effects
      d. All of the above
   4. Which statement is true?
      a. Dynamic predictors can adapt to program behavior
      b. Static predictors rely on the compiler to know what branches will do
      c. Dynamic predictors can behave worse than static predictors for some code
      d. All of the above
   5. A processor has a 1000-entry dynamic branch predictor table. What happens for a program that has 1500 branches?
      a. Roughly 500 of the branches will not get branch predictions
      b. Roughly 1000 of the branches will mess up each other’s predictions
      c. The program will crash
      d. The operating system will have to clean up the misspredicted branches
   6. If we add a branch target buffer (BTB) to the branch predictor, do we still need an adder to calculate the branch address?
      a. Yes, but only if we predict the branch incorrectly
      b. Yes, to verify that the target was correct even if we predict correctly
      c. Only the first time we see the branch so we can store the address in the BTB
      d. No, the instruction provides the address

9. Input/Output
   1. What is the benefit of DMA?
      a. Faster than polling
      b. Easier than interrupts
      c. Processor can do other things at the same time
      d. All of the above
   2. What is hard about building communications busses?
      a. Using enough wires to get the bandwidth
      b. Synchronizing the signals on all the wires
c. Finding space in servers
d. All of the above

3. Why do serial communications send two versions of the same signal?
   a. To check against each other to make sure the final signal is correct
   b. To subtract from each other to cancel out noise that affects both signals
   c. To reduce energy by sharing noise between the wires
   d. All of the above

4. Ethernet networks are a serial link that is shared by multiple computers at the same time. How does Ethernet make sure that other computers do not send at the same time and corrupt the data?
   a. The receiver checks the checksum to verify the data hasn’t been corrupted
   b. The sender checks the sent data to see if it is what was expected
   c. The sender automatically retransmits if the data was corrupted
   d. All of the above

5. What are the benefits of hard disks over flash memory?
   a. Faster
   b. Cheaper
   c. Smaller
   d. None

6. How do memory-mapped IO accesses and virtual memory interact?
   a. VM can protect IO devices from programs that shouldn’t access them
   b. VM can map physical IO device locations to any virtual address
   c. VM can keep certain devices out of a program’s virtual address space to free up memory for it
   d. All of the above

10. Caches

1. Which of the following statements are true about caches?
   a. Caches are smaller and faster than DRAM
   b. Caches increase the chance of reusing data
   c. Caches are cheaper than DRAM
   d. All of the above

2. What data is stored in a cache with a most recently used replacement policy?
   a. The most important data
   b. The most recently used data
   c. The least recently used data
   d. The data that is difficult to get from memory

3. How many lines will each way have in a 256kB, 4-way set associative cache with a 32-byte line size?
   a. 8192
   b. 2048
   c. 1024
   d. 512

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, 2, 5
   a. 1, 2, 3, 4
   b. 0, 2, 4, 5
   c. 2, 3, 4, 5
   d. 0, 1, 3, 4

5. What percentage of the data in the array “data” will be reused at least once from a 32kB fully-associative cache with a 64-byte line?
int data[1024]; // each int is a word
for (int i=0; i<1022; i++) {
    data[i] += data[i+1];
    data[i+1] -= data[i]
}
a. 100%
b. 75%
c. 50%
d. 25%

6. Calculate the cache miss ratio for the previous code (number of hits/number of cache accesses) assuming a 32kB fully associative cache with a 64-byte line (i.e. all the data fits in the cache and we don’t have replacements).
   a. Around 8%
   b. Around 6%
   c. Around 4%
   d. 0%

11. Virtual Memory
1. What is not a benefit of virtual memory?
   a. Memory is protected from other applications
   b. Memory accesses are faster
   c. Memory appears larger than the installed DRAM
   d. Memory does not have to be used in contiguous chunks

2. What size virtual memory pages should you choose to address the most physical memory without a page fault on a system that supports the following TLB configuration: 128 pages with 8kB pages, or 8 pages with 256MB pages, or 1 page with 1GB pages.
   a. 8kB pages
   b. 256MB pages
   c. 1GB pages
   d. Page size doesn’t affect the amount of physical memory you can address without a page fault

3. With which TLB/cache arrangement you can have trouble with multiple applications sharing the same cache, and why?
   a. Virtually indexed, virtually tagged. Applications can access the same virtual address and the address cannot distinguish which one owns each cache line.
   b. Virtually indexed, virtually tagged. Applications can share memory addresses that are already cached and this can cause problems with the replacement policy.
   c. Physically indexed, virtually tagged. Need a translation to start looking in the cache and it is very slow to translate for multiple applications.
   d. Physically indexed, physically tagged. We don’t need a translation to start looking in the cache and addresses for both programs can be shared, which means we don’t know who is accessing the data.

4. If two programs want to share data using virtual memory, they can do so by mapping some pages to the same physical memory. On which type of architectures does this not work?
   a. On machines with a virtually indexed, physically tagged cache
   b. On machines with a virtually indexed, virtually tagged cache
   c. On machines with a physically indexed, virtually tagged cache
   d. It will work on all of the above

5. What does it mean if we have more bits for the virtual page number than we do for the physical page number?
   a. We have very large pages
   b. We have run out of physical memory
c. We have less physical memory than our ISA supports

d. We have more physical memory than our ISA supports

6. Why would we prefer a single-level page table rather than a multi-level one?
   a. To reduce the total size of the page table
   b. To make translations faster
   c. To reduce the in-memory size of the page table
   d. None of the above

12. Parallelism
   1. If we have a program that is 80% parallel and I have 1000 cores, how much faster can we make the program?
      a. 1000x
      b. 800x
      c. 5x
      d. 2.5x

2. What do you need to add to the pipeline to support executing load/store instructions in parallel with other instructions?
   a. Extra data memory
   b. Extra data memory ports
   c. Extra register file ports
   d. All of the above

3. Which of the following statement is incorrect about locks and a particular piece of data?
   a. Locks don’t prevent any other processor from changing the data while it is locked.
   b. Locks don’t prevent any other processor from accessing the data while it is locked.
   c. Locks enable each processor to safely update the data at the same time.
   d. Locks enable the program to keep track of when other processors are accessing the data.

4. What could happen if we do not use atomic instructions while implementing locks?
   a. Hazards in the pipeline.
   b. The appearance of shared data in the cache that shouldn’t be there.
   c. The data could be changed during the process of acquiring the lock.
   d. A segmentation fault.

5. A processor can do any three ALU operations (add/sub/etc) at the same time if the instructions are independent. A program has 60% loads/stores and 40% ALU operations. 15% of the instructions depend on a previous instruction and can only execute one at a time. What is the best IPC you could expect?
   a. 2.7
   b. 2.1
   c. 1.7
   d. 1.5

6. An Intel Xeon Sandy Bridge processor has 4 processors (cores) and 32kB of L1 cache for each processor, 256kB of L2 cache for each processor, and 20MB of L3 cache shared between all 4 processors. Intel is considering changing the processor to reduce the L3 cache to 13MB, keep the L2 the same, and increase the L1 cache to 64kB. What would you expect to happen when you run 4 programs that each use 3MB of data on the new design compared to the Sandy Bridge? (Assume the only difference is the cache sizes.)
   a. No difference
   b. The programs will run slower on the new design because the L3 cache will be almost full when executing, while on the Sandy Bridge there will be plenty of extra space.
   c. The programs will run slower on the Sandy Bridge because it is an older generation.
   d. The programs will run faster on the new processor because the L1 size is bigger
Make sure you circled the answers on the answer page. Answers on other pages will NOT be graded.