**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 $-\frac{1}{3}$ 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. What address does the following code load?
   addi R3, R0, 12
   addi R2, R3, 12
   addi R1, R0, 4
   lw R1, 12(R2)
   a. 4
   b. 12
   c. 24
   d. 36

2. Why do we need both a register and an ALU to read data from memory?
   a. We don’t
   b. The ALU calculates the address and the register stores the data read
   c. The ALU calculates the data and the register stores the address
   d. The ALU always does something regardless of whether we use the results

3. In MIPS, what would happen if you try to write into the R0 register?
   a. Your program might crash
   b. Your operating system might crash
   c. The content of R0 is changed
   d. Nothing, this register always contains 0

4. In MIPS, what special meaning does the “exit:” label has compared to other labels?
   a. It has no special meaning
   b. It is used in branch instructions to jump to the end of a loop
   c. It is used in branch instructions to jump to the end of the program
   d. It marks the end of the current function

5. If R4 contains an unsigned integer, what does the instruction “sll R4, R4, 2” do?
   a. It multiplies the content of R4 by 4
   b. It shifts the bits in R4 2 positions to the left
   c. A and B
   d. Neither A nor B

6. Which of the following will NOT copy the contents of R2 to R1?
   a. addi R1, R2, 0
   b. sll R1, R2, 0
   c. sw R1, R0(12)
2. ISA 2
   1. Which statement is NOT true?
      a. J can only jump within its 256 MB (2^{23}) regions
      b. BEQ can sometimes jump (branch) further ahead than J can
      c. Either J or BEQ can always jump further than JR
      d. BEQ and BNE have the same restrictions for the jump target
   2. main() uses t0, t1, t2, s0, 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. 8
      b. 9
      c. 11
      d. 17
   3. Which of the following is NOT a valid statement?
      a. beq R1, R5, exit
      b. add RA,R0,T1
      c. sub R0,R0,R0
      d. addi T1,T2,63000
   4. What is written to $RA when the JR instruction is executed?
      a. Nothing
      b. PC
      c. PC + 4
      d. The previous value of the $RA register
   5. What is a disadvantage of a load-store-register machine compared to a memory-register machines?
      a. More complex instruction decoder
      b. Cannot be pipelined
      c. Less dense code
      d. Denser code
   6. When calling another procedure, a certain convention must be followed. Why?
      a. To minimize the stack size
      b. To avoid corrupting data from other programs.
      c. Procedures do not know which registers their caller and callees use
      d. There may be an arithmetic overflow, which must be checked beforehand

3. Computer Arithmetic
   1. Will the results be correct if you use an ALU designed for 32-bit two’s complement addition for 32-bit unsigned integers?
      a. Yes, however the overflow flag is calculated differently, and may be incorrect
      b. No, if MSB is 1 for any of the inputs, then the answer will be incorrect
      c. Yes, the same algorithm is used, regardless whether the input happen to be signed or not
      d. No, all signed numbers are subjected to the operation NOT(input) + 1 before an addition, which would produce garbage in the unsigned case
   2. Is it possible to complete hardware binary multiplication of two 32 bit numbers in a single clock cycle?
      a. Yes
      b. No, it would require too much power
      c. No, but it will likely be possible in the future, as chips get smaller
      d. No, you need registers to store partial products, which requires extra cycles
3. For a 7-bit normalized floating-point format: \((-1)^S \times (\text{FFFF}) \times (2^{EE})\) where \(\text{FFFF}\) is unsigned and \(EE\) is two’s complement, what is the smallest positive value you can represent?
   a. 0.5
   b. 0.25
   c. 0.125
   d. 0.28125

4. Compute 1010 - 1110 for 4-bit two's complement numbers?
   a. Overflow
   b. +4
   c. +7
   d. -4

5. What is the range of an 8-bit signed magnitude binary number?
   a. -64 to 64
   b. -64 to 63
   c. -128 to 127
   d. -127 to 127

6. Which of the following statements is true?
   a. A non-normalized floating-point number can represent the same number in several ways
   b. Signed-magnitude has a wider range than two's complement numbers
   c. To get the absolute value of a two's complement number, you remove MSB and negate the remaining bits
   d. You can detect arithmetic overflow on unsigned numbers by comparing carry in and carry out of the MSB

4. Logic

1. These two circuits do the same thing. If the inputs to all gates change at the same time, which of them will be faster?
   a. X output: Circuit A, Y output: Circuit A
   b. X output: Circuit B, Y output: Circuit A
   c. X output: Both the same, Y output: Circuit A
   d. X output: Both the same, Y output: Circuit B

2. Why are DRAMs slower than SRAMs?
   a. They only have small transistors to generate the output
   b. They only have a small amount of charge to output
   c. They have longer wires for the output
   d. They aren’t

3. What does the decoder do in a memory?
   a. Take in the row to read and produce the address
   b. Take in the address and choose the row to read
   c. Take in the address and select the columns to output
4. Why are binary circuits more immune to noise than analog circuit?
   a. Binary values are further apart in terms of voltage than analog values, so they are harder to confuse
   b. Binary circuits generate un-polluted outputs (e.g., 1 or 0 out) after each circuit, so the noise doesn’t propagate through
   c. Binary circuits interpret values close to a 1 or 0 as a 1 or 0 even if they are not exactly 1 or 0
   d. All of the above

5. In the above circuit, an adder takes 10ps, a multiplier 50ps, a latch 10ps, and a mux 1ps. What will happen if you run the clock at 25ps for the above circuit?
   a. It will always produce the wrong results
   b. It will always produce the right results
   c. It will only produce the right results if A=0
   d. It will only produce the right results if A=1

6. In the above latch each inverter takes 5ps. How much time before the clock rises does the data need to be ready such that when the clock rises there will be a stable signal to the second part of the latch, and how much time will it take after the clock rises before the new data is available on the output?
   a. 5ps before, 5ps after
   b. 10ps before, 5ps after
   c. 5ps before, 10ps after
   d. 10ps before, 10ps after
1. How many inputs can a 6-bit multiplexer with 4 control bits switch between?
   a. 4
   b. 6
   c. 16
   d. 64

2. If we were to replace the sign-extend module with a “zero-extend”, how would this affect I-type instructions?
   a. Not at all, the sign-extend is not used for I-type instructions
   b. Not at all, the operand is already two's complement encoded
   c. If the operand value is negative, then it changes value
   d. If the operand value is positive, then it will become negative

3. There is a hardware error in our processor chip. As a result none of the I-type instructions work as they should (the other types work fine). Where could the error be?
   a. The ALU mux input control signal (ALUSrc) is stuck at zero (always chooses Reg File input)
   b. The sign-extend module sometimes flips LSB, so that a 1 becomes 0, and vice versa
   c. MemToRg mux input control signal is stuck at one (always choose ALU output)
   d. All of the above faults could be responsible for the error

4. Which of these statements is NOT true?
   a. There are no hazards in a single-cycle processor
   b. Single-cycle processors have no use for a branch predictor
   c. The ALU generally does not need a clock input
   d. Clock cycle time for a single-cycle processor is longer than in a pipelined processor

5. A single-cycle processor has three types of instructions: A takes 1ns, B takes 2ns, and C takes 10ns. A program with an instruction mix of 25% A, 70% B, and 5% C will run at what clock speed?
   a. 1000 MHz (1 / 1ns)
   b. 465 MHz (1/2.15ns)
   c. 333 MHz (1/3ns)
   d. 100 MHz (1/10ns)
6. What are the control signals for BEQ?
   a. RegWrite=0, ALUSrc=Read Data 2, ALUOp=Sub, PCSrc=Zero
   b. RegWrite=0, ALUSrc=SignExtend, ALUOp=Sub, PCSrc=!Zero
   c. RegWrite=1, ALUSrc=Read Data 2, ALUOp=Sub, PCSrc=!Zero
   d. RegWrite=0, ALUSrc=SignExtend, ALUOp=Add, PCSrc=Zero

6. Pipelining

1. A processor is perfectly pipelined with 5 stages and runs at 5x the clock speed with no overhead, but the pipeline is only 80% full. What is the speedup over the original processor?
   a. 0.8x
   b. 4x
   c. 5x
   d. 20x

2. How does pipelining improve performance?
   a. It divides instructions into smaller steps
   b. It allows the processor to work on multiple instructions at the same time
   c. It allows shorter instructions to finish sooner
   d. It reduces the instructions’ latencies

3. What is the role of pipeline registers?
   a. They store the results of each stage so that the next stage can use them
   b. They allow the register file to store more values
   c. They reduce the number of NOPs we need to insert in the pipeline when executing a branch instruction
   d. All of the above

4. A single-cycle processor takes 100ns per instruction. It is perfectly pipelined into 10 stages with pipeline registers that take 2ns and can be kept 60% full. What change in instruction throughput can you expect compared to the single-cycle processor?
   a. 60% slowdown
   b. 60% speedup
   c. 5x speedup
   d. 10x speedup

5. A single-cycle processor takes 100ns per instruction. It is perfectly pipelined into 10 stages with pipeline registers that take 2ns and can be kept 60% full. What change in instruction latency can you expect compared to the single-cycle processor?
   a. 20% speedup
b. 20% slowdown  
c. 60% speedup  
d. 60% slowdown

6. What is the advantage of a multi-cycle CPU compared to a single-cycle CPU?  
a. The clock speed is no longer limited by the slowest instruction path  
b. The CPU can analyze the lengths of the instructions currently in the pipeline and adjust its clock speed accordingly  
c. Each instruction can take a different number of cycles to execute  
d. A and C

7. Hazards
1. What is needed to resolve the following hazard?  
and R1, R2, R1  
ld R2, 0(R3)  
add R2, R4, R1  
a. Forward from MEM to EX  
b. Forward from WB to EX  
c. A and B are both required  
d. Nothing, there is no hazard

2. A 5-stage MIPS pipeline has a double-pumped register file and forwarding from WB to EX but not from MEM to EX. How many NOPs will you need to add to make this code work?  
add R10, R12, R13  
sub R12, R13, R11  
add R13, R12, R11  
a. 3  
b. 2  
c. 1  
d. 0

3. An 8-stage pipeline processor requires NOPs to avoid hazards. The compiler is aware of this and inserts appropriate NOP instructions into the code. What would happen if the processed code now is run on a single-cycle processor?  
a. The program would execute correctly  
b. Conditional branches would not work correctly, due to the zero output from the ALU  
c. The code would not work correctly because the NOPs are in the wrong place  
d. The code would not work correctly because there are too many NOPs

4. What kind of hazard would an add instruction have in a pipeline that executes two instructions at a time (dual-issue) that it does not have in our standard (single-issue) pipeline?  
a. Structural hazard  
b. Data hazard  
c. Control hazard  
d. None

5. Which of the following is true about hazards?  
a. All hazards can be solved with bubbles  
b. The use of bubbles breaks the ISA promise of "Atomic execution" of instruction  
c. Using the branch delay slot ensures the promise of sequential execution is kept  
d. There are still control hazards in single-cycle processor

6. In a certain program run on a 5-stage pipeline processor with double-pumped register file but no forwarding, 50% of all instructions are loads and 10% of the instructions directly following the loads depend on it. What is the slowdown of the program due to these dependencies?  
a. 1.0x (no slowdown)
b. 1.05x
c. 1.1x
d. 1.2x

8. Branch Prediction and Exceptions and Interrupts

1. Why are branch predictors useful?
   a. They allow us to only execute the needed instructions when they are correct
   b. They allow us to squash instructions that should never have entered the pipeline
   c. They enable forwarding between pipeline stages to avoid branch dependencies
   d. All of the above

2. What is the advantage of 2-bit branch predictors compared to 1-bit predictors?
   a. They change decision more slowly
   b. They learn faster
   c. They always make better predictions
   d. They use less logic

3. A program contains 1 million instructions, of which 15% are branches. If we run it on a CPU with a 20 cycles branch penalty and a branch predictor that is 85% accurate, how many cycles will be wasted due to mispredictions?
   a. 100 000 cycles
   b. 200 000 cycles
   c. 450 000 cycles
   d. 1 000 000 cycles

4. How would the performance change if we ran the program described in the previous question on a CPU with a 30 cycles branch penalty but with a branch predictor that is 90% accurate?
   a. Need more information
   b. The performance would improve
   c. The performance would decrease
   d. The performance would remain the same

5. Consider the following code:
   ```c
   for (i = 0; i < 1000000; ++i)
   {
       for (j = 0; j < 1000; ++j)
       {
           m[i][j] = 0
       } // branch j-loop
   } // branch i-loop
   ```

   How many mispredictions would we see if we ran this program on a CPU with a 1-bit branch predictor (initialized to “taken”)?
   a. 0
   b. 1 000 001
   c. 2 000 000
   d. 1 000 005

   6. What if we ran the same code on a CPU with a 2-bit branch predictor? (initialized to “strongly taken”)?
   a. 0
   b. 1 000 001
   c. 2 000 000
   d. 1 000 005

9. Input/Output

1. Why have busses become a problem in modern systems?
1. a. Hard to make small wires  
   b. Hard to connect many wires  
   c. Hard to find space for long wires  
   d. Hard to keep the data on the wires synchronized

2. What is the benefit of DMA?  
   a. Faster to detect that data is ready than polling  
   b. Easier to program than interrupts  
   c. Processor can do other things at the same time  
   d. All of the above

3. How would you access a device if you wanted the lowest possible latency?  
   a. Polling  
   b. Interrupt  
   c. I/O instructions  
   d. DMA

4. Why do we not use Flash memory instead of DRAM?  
   a. Too expensive  
   b. Too slow  
   c. Doesn’t keep data without power  
   d. Can’t store enough data

5. How does Ethernet share a single serial connection?  
   a. Each device is assigned different times to transmit  
   b. Each device checks for data corruption and re-tries if needed  
   c. Uses encryption to keep data safe from corruption  
   d. Actually has separate wire for each device

6. How can memory-mapped IO accesses and virtual memory interact?  
   a. Virtual memory allows more I/O devices per application than the ISA’s address space  
   b. Page protection can limit which programs can access I/O devices directly  
   c. Copy-on-write is used to get new data each time you read from the I/O address  
   d. All of the above

10. Caches

   1. A program is reading every word (4 bytes) consecutively of a very large (4GB) array. Which of the following cache configurations can achieve a hit ratio of more than 50%?  
      Assume LRU replacement policy?  
      a. Cache size: 16 B. Block size 4 B, Direct mapped  
      b. Cache size: 8 MB. Block size 8 B. 2-way associative  
      c. Cache size: 2 KB. Block size 32 B. Fully associative  
      d. All of the above

   2. A program is reading a 4x4 word matrix in "the wrong way", e.g., addresses 0, 4*4, 8*4, .... What is the lowest associativity that will have no conflict misses for a 32B LRU cache with a block size of 8 B?  
      a. 2-way  
      b. 4-way  
      c. It is impossible to fully avoid conflict misses with this access pattern  
      d. There are no conflict misses with this memory pattern

   3. What data is evicted on a miss in a LRU cache?  
      a. The data that has been in the cache for the longest time  
      b. The data that has most recently been installed in the cache  
      c. The data that has not been used for the longest time  
      d. Misses do not evict data

   4. Which of the following statements are NOT true about caches?
a. Caches are faster than the DRAM used for main memory
b. Large cache lines improves spatial locality
c. High associativity decreases the number of misses when you first access data
d. LRU is not always the best replacement policy

5. What data will be in a 4-entry, fully-associative, LRU cache with one word per line after the following memory accesses? 0, 1, 2, 3, 4, 5, 3, 2, 1?
   a. 1 3 4 5
   b. 1 2 3 5
   c. 1 2 4 5
   d. 0 3 4 5

6. How many sets does a 4kB, 4-way set associative cache with an 8-byte line size have?
   a. 4
   b. 64
   c. 128
   d. 512

11. Virtual Memory
1. What does the TLB do?
   a. It caches recently-used data
   b. It caches recently-used address translations
   c. It caches recently-used pages
   d. It caches recently-used instructions

2. What would happen if two programs tried to read from the same virtual address?
   a. The two programs would likely crash
   b. One of them will read the correct data while the other would read random data
   c. They would read the same data
   d. They would read their own data

3. What are the benefits of virtual memory?
   a. It increases security by isolating each process’ memory
   b. It frees the applications from having to manage a shared memory
   c. It allows an application to use more memory than physically available
   d. All of the above

4. What can cause a page fault?
   a. Trying to read a virtual address not present in the TLB
   b. Trying to read from a page that is not in main memory but that exists on the disk
   c. Trying to read an address that has no page in the main memory and no page on the disk
   d. B and C

5. What is true if you have more physical address bits than your virtual address bits?
   a. The only way to use all the physical memory is to run multiple programs
   b. You cannot address all the virtual memory from one program
   c. You can address more memory from one program than you have physical memory
   d. All of the above

6. For a system with 4MB (2^{22} bytes) pages and 32GB (2^{35} bytes) of physical memory, figure out which bits of a 64 bits virtual address are used for the Virtual Page Number (goes into the TLB for translation), the Page Offset and how many bits of Physical Addresses you have?
   a. 22 bits for PO, 42 bits for VPN. 35 bits physical addresses
   b. 42 bits for PO, 35 bits for VPN. 22 bits for physical addresses
   c. 35 bits for PO, 42 bits for VPN. 22 bits for physical addresses
   d. 22 bits for PO, 35 bits for VPN. 42 bits physical addresses
12. Parallelism

1. A program takes 200s to run on a single CPU, and 35% of it can be run in parallel. How long will it take to run the same program on a multicore CPU with 8 cores?
   a. 112s
   b. 120s
   c. 138s
   d. 145s

2. Which of the following designs to improve performance does not require multiple PCs?
   a. Simultaneous multithreading (issuing instructions from multiple programs into the pipeline at the same time)
   b. Multicore processors (having multiple processors on one chip)
   c. Dual-issue processors (issuing multiple instructions from the same program into the pipeline at the same time)
   d. All of them require multiple PCs

3. Why do we need locks?
   a. To synchronize accesses to the same data by different processors
   b. To make sure only one thread can use the pipeline at the same time
   c. To avoid page faults
   d. None of the above

4. The Intel Haswell Processor (CPU) with the Iris Pro 5100 integrated Graphics Processor (GPU) has the following characteristics: 40 GPU cores that can each execute 20.8B floating point operations per second per core and 4 CPU cores that can each execute 56B floating point operations per second per core. Which is better for a program that is 20% parallel?
   a. Run on the CPU cores
   b. Run on the GPU cores
   c. Same
   d. Need more information

5. How much faster can my program run if I have 9000 cores and 10% of my program cannot be parallelized?
   a. 9000x faster
   b. 4500x faster
   c. 5x faster
   d. 10x faster

6. An image processing algorithm divides an image in 4 equally-sized parts and works on each part on a separate CPU in parallel. Each processor completes its work in 10 seconds, except one that completes in 12 seconds. If we run this program on a CPU with 8 cores, what is its execution time?
   a. 2 seconds
   b. 10 seconds
   c. 12 seconds
   d. 5.25 seconds

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