Indian Institute of Technology Kharagpur

Department of Computer Science and Engineering

Exam-3, Autumn 2021-22

Computer Organization and Architecture (CS31007)

Students: 133 **Date**: 15-November-2021

Full marks: 60 **Time**: 12:00 noon – 2:00 PM

Credit: 40%

INSTRUCTIONS: This is an OPEN-BOOK, OPEN-NOTES test. The questions are such that they require either numerical answers or very short answers (in a few sentences or a few lines of code). Please submit ONLY THE ANSWERS. It might help if you typeset your solution using software like MS Word/LibreOffice Writer, convert it to PDF, and then upload the PDF file on Moodle. DO NOT FORGET TO WRITE YOUR NAME AND ROLL NUMBER AT THE TOP OF YOUR ANSWER SHEET. You may use calculators if required. ANSWER ALL QUESTIONS.

1. (a) Consider a 64-byte cache with 8 byte blocks, an associativity of 2 and LRU block replacement. Virtual addresses are 16 bits. The cache is physically tagged. The processor has 16KB of physical memory. What is the total number of tag bits? [**3**]

   (b) Assume that the processor in part-(a) is part of a computer where each page is 64 bytes. How large would a single-level page table be given that each page requires 4 protection bits, and entries must be an integral number of bytes? [**3**]

   (c) Consider a data cache which follows the `Write Back with Write Allocate` policy, to handle read misses and write misses. The cache has a 95% read hit rate, 90% write hit rate, zero clock cycle hit latency, and 50 clock cycles latency to write/read a single block to/from main memory. The CPU is executing a program in which 5% instructions are `load` instructions, and 0% instructions are `store` instructions. On the occasions when a cache read miss happens, 10% of times the target cache block is found to be `dirty`. Calculate the `effective CPI` when the above-mentioned program is executed, assuming all instructions execute in one clock cycle when interaction with memory is not required, or when cache hit happens. [**4**]

2. Consider a CPU with only one level of cache memory.

   (a) For a data cache with a 92% hit rate and a 2-cycle hit latency (unlike zero cycle hit latency in most processors), calculate the *Average Memory Access Time* (AMAT). Assume that latency to memory and the cache miss penalty together is 124 cycles. Note: the cache must be accessed after memory returns the data. [$\mathbf{2\frac{1}{2}}$]

   (b) Calculate the performance of a processor taking into account stalls due to data cache and instruction cache misses. The data cache (for loads and stores) is the same as described in part-(a) above and 30% of instructions are loads and stores. The instruction cache has a hit rate of 90% with a miss penalty of 50 cycles. Assume the base CPI using a perfect memory system is 1.0. Calculate the CPI of the pipeline, assuming everything else is working perfectly. Assume the load never stalls a dependent instruction and assume the processor must wait for stores to finish when they miss the cache. Finally, assume that instruction cache misses and data cache misses never occur at the same time.

      i. Calculate the additional CPI due to the instruction cache stalls. [$\mathbf{2\frac{1}{2}}$]

      ii. Calculate the additional CPI due to the data cache stalls. [$\mathbf{2\frac{1}{2}}$]

      iii. Calculate the overall CPI for the machine. [$\mathbf{2\frac{1}{2}}$]

3. (a) Explain two differences between a process in `wait` state and a process in `ready` state. [**2**]

   (b) A 32-bit CPU receives an interrupt with an interrupt code 0x04. It is known that the interrupt vector table for this computer starts at address 0x0000ab00. The value of the relevant entry in the interrupt vector table for the raised interrupt (say, VT) is related to the address of the entry (say A) by:

$VT = A + 0x00010000$. The interrupt service routine to handle this interrupt is of length 8 KB, and the entire interrupt service routine is stored contiguously in memory. Calculate the (starting) address of the last instruction of the interrupt service routine. Note: 1 KB = 2^{10} bytes. [**4**]

(c) Suppose we have a magnetic disk with the following parameters: average seek time 12 ms; rotation rate 3600 RPM; transfer rate 3.5 MB/second (with 1 MB $\equiv 2^{20}$ bytes); number of sectors per track 64; sector size 512 bytes; controller overhead 5.5 ms.

i. Calculate the average time to read a single sector. [**2**]

ii. Calculate the average time to read 8 KB in 16 consecutive sectors on the same cylinder. [**2**]

4. (a) Consider a regular single-cycle data path implementation of a non-pipelined MIPS processor that implements only a small subset (`lw, sw, add, addi, slt, beq`) of instructions. Write down the appropriate values 0, 1, x (don't care) of the following control signals that are needed to execute the MIPS instruction **sw $t1, -64($t2)**:
RegDest, ALUSrc, MemtoReg, RegWrite, MemRead, MemWrite, Branch, ALUOp1, ALUOp0, Function Fields (6-bit value). [**2**]

(b) Now suppose *one wire* of the 32-bit bus that leaves the Sign-Extension module, has become defective with a stuck-at-0 fault (the wire is always fixed at logic-0 regardless of the actual signal arriving on it). Which of the following statement(s) is(are) **true**? Justify your answer in brief. [**2**]

- A. The execution of some *lw* and *sw* instructions might be affected, but other instructions will be fine.
- B. The execution of some *lw*, *sw* and *beq* instructions might be affected but other instructions will be fine.
- C. The execution of some *sw* instructions, some *addi* instructions, and all *beq* instructions will be affected.
- D. The execution of all add instructions and all branch instructions of type **beq $t1, $t2, Target** with ($t1) $\neq$ ($t2), will remain unaffected.
- E. None of the above.

5. Consider a regular single-cycle data path implementation of a non-pipelined MIPS processor as in Question–4(a). The operation-times for the major functional units are given as follows:

- Instruction and Data Memory Units: 2 ns each;
- Register File Read/Write: 1 ns each;
- ALU: 3 ns; Adder for PC+4: 2 ns; Adder for branch-address computation: 2 ns;
- MUX block: 1 ns each; Sign-Extend and Shift-Left: 1 ns each;
- other units: negligible.

(a) Calculate the maximum clock frequency with which this CPU can run. [**2**]

(b) The same machine is now implemented as 5-stage pipeline where the delay through each of the inter-stage pipeline-latches is 1 ns. Estimate the maximum speed-up that is achievable by the pipelined machine compared to the non-pipelined one (assume no hazard is present). [**2**]

6. Consider the following sequence of actual outcomes in a program for a branch instruction that is iteratively executed 20 times. "T" means "the branch is taken", whereas "N" means "the branch is not taken". Assume that this is the only branch instruction in the program.

T T T N T N T T T N T N T N T N N N T N

The pipeline controller uses a dynamic branch predictor based on **2-bit branch history**. Assume that the predictor state is initialized to "N". Calculate the number of mispredictions. [**3**]

7. Consider a standard 5-stage pipelined implementation of MIPS processor, where each of the stages takes the same amount of time (i.e., 1 clock cycle). The PC-Update is being implemented in the Mem-Stage. Given a program P, assume that all machine instructions in P can be executed without any pipeline stall, except those which follow a *beq* instruction. In P, the frequency of branch instructions is 30% and out of those, 60% are not taken. The pipeline controller works on statically predicting that a branch is always "*not taken*". Assume that the pipeline overhead due to inter-stage latches is negligible.

   (a) Calculate the maximum speed-up achievable by this pipelined machine over non-pipelined single-cycle implementation. [**2**]

   (b) If you are allowed to do certain hardware modification concerning zero-checking and PC-Update, what will be the speed-up in the modified machine? Describe the scheme for modification and justify your answer. [**2**]

8. (a) Consider the execution of the following code P in a standard five-stage MIPS pipelined machine (IF, ID/RR, EX, Mem, WB), without any additional forwarding hardware. However, Register-Read (Write) is always accomplished in the second (first) half-cycle of the system clock pulse. Assume that the first stage (IF: Instruction Fetch) of the first instruction (`I0`) is being executed during clock cycle #1. Also assume that there will be no arithmetic overflow while running P.
   P:
   ```
   I0: add $t3, $t1, $t2
   I1: sub $t5, $t4, $t3
   I2: lw $t6, 100($t5)
   I3: add $t1, $t5, $t6
   I4: xor $t8, $t2, $t3
   I5: addi $t8, $t8, 100
   ```

      i. Identify all data-hazards in P; [**5**]

      ii. In the worst case, how many clock cycles are needed to complete P? [**3**]

   (b) We now modify the machine by using additional resources such as hazard detector and forwarding hardware (EX $\rightarrow$ EX, and Mem $\rightarrow$ EX). Also, a smart compiler is available to reshuffle the code if needed. Write the modified sequence of instructions which would minimize the number of pipeline stalls under this arrangement. Also, write the estimated number of clock cycles needed to complete P now. [**5**]

   (c) If there is a data hazard encountered by an instruction that immediately follows a load instruction, how is that detected by the hardware? [**2**]