

EECS 470 Winter 2024

Homework 1

Due Thursday, January 18 by 10pm.

Homework turned in late but by the 19th at noon will get 75% credit.

Name: _____ unique name: _____

Assignments that are difficult to read will lose at least 50% of the possible points and we may not grade them at all. This is an individual assignment; all work should be your own.

- If you use references other than the text and class notes, be sure to cite them!
- This assignment is worth about 2% of your grade in the class and is graded out of 100 points.
- Remember you may drop one homework assignment or quiz score.
- Please state any assumptions.

-
1. **Understanding Pipelines.** Consider the following code segment. Assume the branch is taken more than 99.9% of the time. **[16 points]**

```
Loop: LD R1, 0(R2)      ; R1=MEM[R2+0]
      DADDI R1, R1, #1  ; R1=R1+1
      SD 0(R2), R1      ; MEM[R2+0]=R1
      DADDI R2, R2, #4  ; R2=R2+4
      DSUB R4, R3, R2   ; R4=R3-R2
      BNEZ R4, Loop    ; if(R4!=0) goto Loop
```

- Show the timing of the above instruction sequence on the RISC pipeline in Appendix C of our text. Assume there is no forwarding but that a read and write in the same clock cycle “forwards” through the register file. Your answer should be drawn similarly to figure C.10 on page C-21. Memory accesses take one cycle and there is separate instruction and data memory. Branches are predicted not-taken. Branches are resolved in the decode stage. **[3]**
 - As part “a” but assume normal forwarding and bypassing. Stalls due to dependencies occur in the ID stage. **[5]**
 - Assume the RISC pipeline with a single-cycle delayed branch (a.k.a. “branch delay slot”) and normal forwarding and bypassing hardware. Schedule the instruction *including* the branch delay slot as *efficiently as possible*. You may reorder the instructions and modify the individual instruction *operands*, but do not change the number or opcode of the instructions. Again provide a timing diagram and indicate the number of cycles needed for a single iteration of the loop. **[8]**
2. **Cache Design.** Consider an 4 MB 8-way set-associative cache with 32-byte lines. Both the virtual and physical address spaces are 32 bits in size. **[8 points]**
- How many bits are used for the tag, set index, and byte offset respectively? **[2]**
 - How many bytes (total) are used to store the tags for this cache? **[2]**
 - What would your answer to b) be if the cache were fully-associative? Direct-mapped? **[4]**
3. **Cache Details.** Consider a 1KB cache with a 16-byte block size. Assume all entries in the cache start as “invalid” and the address spaces are 16-bits. **[13 points]**
- Assuming the cache is 2-way associative, for each address in the (load) address stream indicate if the access is a hit or miss. If it is a miss, indicate which type of miss (from the 3C’s) it is. Assume all requests are for 1 byte. **0x4001, 0x400A, 0x4017, 0x1000, 0x2000, 0x4000, 0x2005, 0x4008.** **[5]**
 - Identify as short a memory access pattern as possible (specific addresses) for the cache described above where a direct-mapped cache will get a better hit-rate than a two-way associative cache. Make it so that the sum of the addresses is as small as possible. **[4]**
 - Identify as short a memory access pattern as possible (specific addresses) for the cache described above where a two-way associative cache will get a better hit-rate than a direct-mapped cache. Make it so that the sum of the addresses is as small as possible. **[4]**

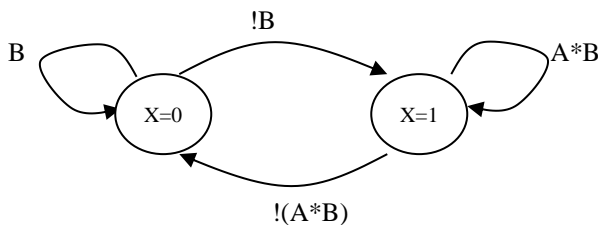
4. **ISA Design [15 points]**

- Give two examples of characteristics that distinguish RISC from CISC architectures. [2]
- For each of the following sets of terms define each term and write a sentence that explains how they relate. [6]
 - Register pressure, register spill, register fill.
- Say you have an “addi” instruction with the format “addi rx, ry, constant” where rx and ry are register numbers and the constant is a 16-bit 2’s complement number. Assuming instructions are all 32 bits in size and there are 32 registers, what percent of all possible instruction encodings is used by this instruction? [4]
- State **two disadvantages** of having very large (1000’s of entries) register files and one advantage. [3]

5. **Multicore and power.** Suppose that the code to run a transaction is 80% perfectly parallelizable (such that performance scales linearly with the number of cores), while the remaining 20% is purely serial (can only run on one core) and only one transaction can be run at a time. The company is evaluating three chips: [11 points]

- a single-core chip that draws 100W of power;
 - a 135W quad-core chip where each core has a “speedup” of 0.7 over the single-core chip
 - a 150W 8-core chip where each core has a “speedup” of 0.5 over the single-core chip
- Suppose the application runs at 1000 transactions per second on the single-core chip. How many transactions per second does it achieve on quad-core? The 8-core? [6]
 - On average how much energy (Joules) per transaction is required by each chip? Which chip is most *energy* efficient? [3]
 - Redo part b assuming that the transactions are 100% parallelizable. [2]

6. **Digital Logic Design.** Consider the following Moore-type state-machine: [6 points]



Draw a circuit diagram which implements this state-machine. You are to use only 2-input AND, OR and XOR gates, inverters, and D flip-flops. The inputs are not available in inverted form. *Be sure to include a clock input.*

7. **Parallel programming** Your friend has been tasked with writing a function called “increment()”. The function is suppose to increment a global variable called “count” *that may be shared by multiple threads/processes*. He’s written the following in a C-like language. The idea is that any thread/process that wants to change count will have to call this function. Assume “lock” is also a global (and volatile) variable. [15 points]

```

int increment()
{
    while(lock); //This stays here until lock==0
    lock=1;
    count=count+1;
    lock=0;
}
  
```

- Why is a lock needed? What is your friend *trying* to do? [3]
- Why won’t the above code work? (you can assume “lock” is initialized to zero) [3]
- Say we have a function called TAS that implements an atomic test-and-set (see http://en.wikipedia.org/wiki/Test_and_set). It takes a pointer to the variable to be used and returns a 0 if it succeeds in getting the lock. Rewrite your friend’s code to use TAS. (prototype is “int TAS(int * A)”). Explain why your solution solves the problem(s) you found in your friend’s code. [6]
- Answer the following: (Each can be answered in just a few words). [3]
 - Why is it important that lock and count be declared “volatile”?
 - What impact, if any, does the “volatile” declaration have on how we cache things?
 - What impact, if any, does the “volatile” declaration have on how compiler optimizations?

8. **Misc stuff.** The following questions are not part of any prerequisite material, nor will it be covered in class before the homework is due, but related material *will* show up. Each should be fairly easy to answer given a few minutes of on-line research. [16]
- a. In nanoseconds, what the period of a 2GHz clock? In picoseconds? How far does light (in a vacuum) travel in that time? [1]
 - b. Power [4]
 - Finish the following rhyme : “Twinkle Twinkle little star, power is I squared _____”
 - What is the difference between dynamic and static power?
 - Dynamic power is generally listed as being proportional to CV^2f .
 1. What are C, V and f in this context?
 2. Why *might* one say that dynamic power is proportional to performance cubed? (Guess if you must, you’ll only lose points if you don’t make a reasonable try.)
 - c. What are SPECint and SPECfp? Explain their relevance to computer architecture. [3]
 - d. Describe how the resistance of a wire changes (by how much and in what direction) as [4]
 - The wire gets twice as long
 - The (round) wire has twice the diameter.
 - The (rectangular) wire has twice the width but the same height
 - The wire changes from tin to copper.
 - e. Read all of [https://en.wikipedia.org/wiki/Spectre_\(security_vulnerability\)](https://en.wikipedia.org/wiki/Spectre_(security_vulnerability)). Write a one paragraph summary of what the attack is about. It is fine if you don’t fully understand, just read it and do your best—we will accept any reasonable effort. [4]