COMPSCI 2GA3 Fall, 2021

Assignment/Homework 4, Nov. 21st 2021
Assignment due date: Dec. 8th, 23:59:59.
Note: Please work on this assignment individually. Students copying each other's answer will get a zero and will perform poor on midterm and final.

Written Exercises

Complete the following questions from *Computer Organization and Design: The Hardware Software Interface: Computer Organization and Design The Hardware/Software Interface: RISC-V Edition*

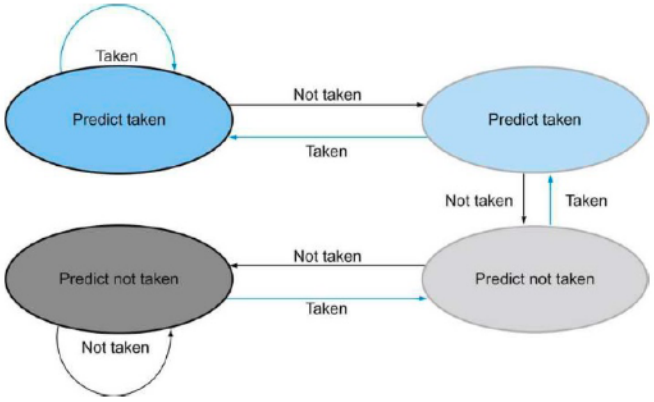
1. Exercise 4.27.1-3 (5 Marks) Problems in this exercise refer to the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath:

```
add x15, x12, x11
ld  x13, 4(x15)
ld  x12, 0(x2)
or  x13, x15, x13
sd  x13, 0(x15)
```

   4.27.1 If there is no forwarding or hazard detection, insert `NOP`s to ensure correct execution.
   4.27.2 Now, if possible, change and/or rearrange the code to minimize the number of `NOP`s needed. You can assume register `x17` can be used to hold temporary values in your modified code.
   4.27.3 If the processor has forwarding, but we forgot to implement the hazard detection unit, what happens when the original code executes?

2. Exercise 4.29.1-3 (5 Marks) This exercise examines the accuracy of various branch predictors for the following repeating pattern (e.g., in a loop) of branch outcomes: T, NT, T, T, NT.

4.29.1 What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?
4.29.2 What is the accuracy of the 2-bit predictor for the first four branches in this pattern, assuming that the predictor starts off in the bottom left state from figure below (Figure 4.61) (predict not taken)?
4.29.3 What is the accuracy of the 2-bit predictor if this pattern is repeated forever?



3. Exercise 5.11.1-2 (5 Marks) This exercise examines the effect of different cache designs, specifically comparing associative caches to the direct-mapped caches from Section 5.4. For these exercises, refer to the sequence of word address shown below.

```
0x03, 0xb4, 0x2b, 0x02, 0xbe, 0x58, 0xbf,
0x0e, 0x1f, 0xb5, 0xbf, 0xba, 0x2e, 0xce
```

5.11.1 Sketch the organization of a three-way set associative cache with two-word blocks and a total size of 48 words. Your sketch should have a style similar to Figure 5.18, but clearly show the width of the tag and data fields.

5.11.2 Trace the behavior of the cache from Exercise 5.11.1. Assume a true LRU replacement policy. For each reference, identify

- the binary word address
- the tag
- the index
- the offset

- whether the reference is a hit or a miss, and
- which tags are in each way of the cache after the reference has been handled

4. Exercise 5.17.1-2 (5 Marks) There are several parameters that affect the overall size of the page table. Listed below are key page table parameters.

Virtual Address Size	Page Size	Page Table Entry Size
32 bits	8 KiB	4 bytes

5.17.1 Given the parameters shown above, calculate the maximum possible page table size for a system running five processes.

5.17.2 Given the parameters shown above, calculate the total page table size for a system running five applications that each utilize half of the virtual memory available, given a two-level page table approach with up to 256 entries at the 1st level. Assume each entry of the main page table is 6 bytes. Calculate the minimum and maximum amount of memory required for this page table.