Ramiro Gonzalez

Homework/Lab 03a

CSE 140
March 12, 2020

Using Ubuntu Linux

- Collaborators : Jamal Bills, Nikko Solon

1. Given the following MIPS code snippet

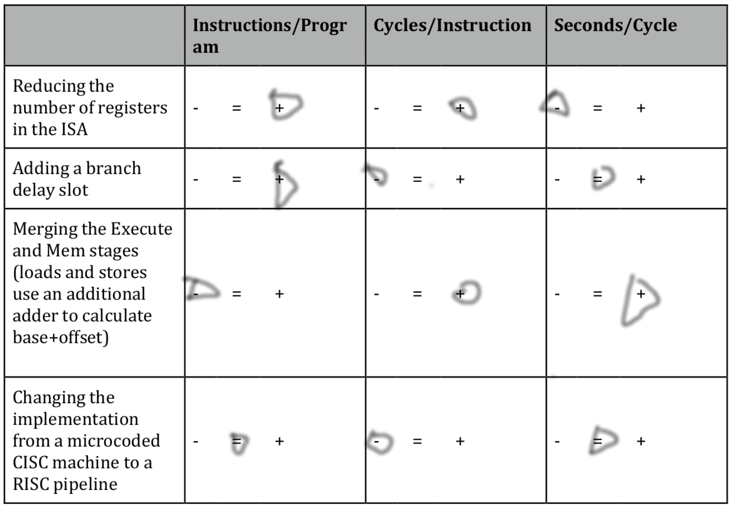
   (a) Hazards in lines 2 and 3; 6@2.5

   (b) Reorder/rewrite

```
        LOOP:
             addi $t0, $t0, 4
             lw   $s0, 60($t0)
             lw $v0, 0($t0)
             bne $s0, $0, LOOP
             sw  $v0, 20($t0)
```

   (c) The three reasons why we use caches in a pipelined processor. Is to increase clock speeds, reduce time required to access memory, and reduce interruptions. Pipelining increases CPU speed by making each instruction run proportionally to each other.

   (d) Circle whether each change will make something decrease (-)



	Instructions/Program	Cycles/Instruction	Seconds/Cycle
Reducing the number of registers in the ISA	- = [✓ +]	- = [✓ +]	[✓ -] = +
Adding a branch delay slot	- = [✓ +]	[✓ -] = +	- [✓ =] +
Merging the Execute and Mem stages (loads and stores use an additional adder to calculate base+offset)	[✓ -] = +	- = [✓ +]	- = [✓ +]
Changing the implementation from a microcoded CISC machine to a RISC pipeline	- [✓ =] +	[✓ -] = +	- [✓ =] +

Figure 1: Caption

2. Consider the following loop.

```
    loop:   lw r1, 0($r1)
            and r1, r1, r2
            lw r1, 0(r1)
            lw r1, o(r1)
            beq r1, r0, loop
```

(a) Show a pipeline execution diagram

```
2nd Cycle | lw r1, 0(r1)  | WB |
2nd Cycle | lw r1, 0(r1)  | EX | MEM | WB |
2nd Cycle | beq r1,r0,loop| ID | X   | EX | MEME| WB |
3rd Cycle | lw r1,0(r1)   | IF | X   | ID | EX | MEM | WB |
3rd Cycle | add r1,r1,r2  |    |     | IF | ID |  X  | EX | MEM| WB
3rd Cycle |lw r1, o(r1),r2|    |     |    |IF  | X  | ID  | EX | MEM|
3rd Cycle | lw r1, 0(r1)  |    |     |    |    |    | IF  | ID | X  |
3rd Cycle |beq r1, r0,loop|    |     |    |    |    |     | IF | X  |
```

(b) How often (as percentage of all cycles)

- 8 clock cycles in which all five pipeline stages are doing useful work, not all stages are used.

3. Caches are important to provide a high performance

(a) For each of these references, identify the binary address (in words, not bytes)

Word Address	Binary Address	Tag	Index	Hit(H)/Miss(M)
3	0000 0011	0	3	M
180	1011 0100	11	4	M
43	0010 1011	2	11	M
2	0000 0010	0	2	M
191	1011 1111	11	15	M
88	0101 1000	5	8	M
190	1011 1110	11	14	M
14	0000 1110	0	14	M
181	1011 0101	11	5	M
44	0010 1100	2	12	M
186	1011 1010	11	10	M
253	1111 1101	15	13	M

(b) For each of these references, identify the binary address (in words), the tag, and the index given a direct-mapped cache

Word Address	Binary Address	Tag	Index	Hit(H)/Miss(M)
3	0000 0011	0	1	M
180	1011 0100	11	2	M
43	0010 1011	2	5	M
2	0000 0010	0	1	H
191	1011 1111	11	7	M
88	0101 1000	5	4	M
190	1011 1110	11	7	H
14	0000 1110	0	7	M
181	1011 0101	11	2	H
44	0010 1100	2	6	M
186	1011 1010	11	5	M
253	1111 1101	15	6	M

(c) You are asked to optimize a cache design for the above references
C2 takes 3 cycles

(d) Below are listed parameters for different direct mapped cache designs

```
            Cache Data Size: 32kiB
            Cache Block Size: 2 words
            Cache Access Time: 1 cycle
```

4. For a direct-mapped cache design with 32-bit address

(a) What is the cache block size (in words)?
The number of words is 8.

(b) How many entries does the cache have?
There are 32 entries.

(c) What is the ratio between total bits required

data	tag	valid	cache/block
256bit	22bits	1bit	1.09

(d) How many blocks are replaced
Replace 4 blocks

(e) What is the hit ratio?
There are 4 hits, and thefore the ration is $4/12 = .33$