CSC 656

Project 1

**Peitong Shi**

**915604994**

March 15, 2018

Problem 1:

Version 1:

Load Circle: IC \* (0.35 + 0.35 \* 0.55) = IC \* 0.5425

Store Circle: IC \* 0.1

ALU Circle: IC \* 0.35

Branches Circle: IC \* (0.2 + 0.2 \* 0.65) = IC \* 0.33

Total Circle: IC \* (0.5425 + 0.1 + 0.35 + 0.33) = IC \* 1.3225

Total Time: 400 \* Total Circle = IC \* 529 ps

Version 2:

Since 3 stalls on MEM

Load Circle: IC \* (0.35 + 0.35 \* 0.55 \* 3) = IC \* 0.9275

Store Circle: IC \* 0.1

ALU Circle: IC \* 0.35

Since 2 stalls on IF

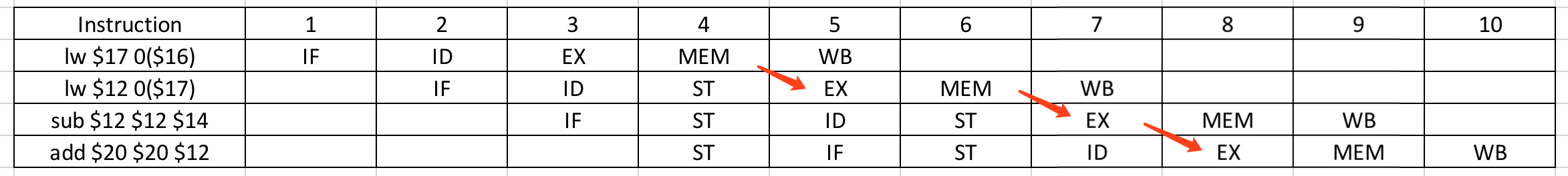
Branches Circle: IC \* (0.2 + 0.2 \* 0.65 \* 2) = IC \* 0.46

Total Circle: IC \* (0.9275 + 0.1 + 0.35 + 0.46) = 1.8375

Total Time = 250 \* Total Cycle = IC \* 459.375 ps

Problem 2:





First arrow: At circle 4, MEM -> EX, forward A = 01

Second arrow: At circle 6, MEM -> EX, forward A = 01

Third arrow: At circle 7, EX -> EX, forward B = 10

1. Original codes:  
   loop: lw $4, 0($16)   
    lw $7, 4($16)   
    add $4, $7, $4   
    sw $4, 0($23)   
    addi $16, $16, 8   
    addi $23, $23, 4   
    bne $16, $2, loop  
     
   Unrolling code:

loop: lw $4, 0($16)

lw $7, 4($16)

add $4, $7, $4

sw $4, 0($23)

lw $4, 8($16) // Increase by 8 because originally, address

lw $7, 12($16) // is added by 8 four each loop

add $4, $7, $4

sw $4, 4($23) //$23 is increased by 4 for each loop originally

lw $4, 16($16) //same

lw $7, 20($16) //same

add $4, $7, $4

sw $4, 8($23) //same

lw $4, 24($16)

lw $7, 28($16)

add $4, $7, $4

sw $4, 10($23)

addi $16, $16, 32

addi $23, $23, 16

bne $16, $2, loop

Instead of using one single register for each lw and sw, we use multiple register for each load/store instruction and then adjust the order of unrolling codes:

loop: lw $4, 0($16)

lw $7, 4($16)

//now we still load, change the $4 to $8 $9 $10, change

// the $7 to $11, $12, $13

lw $8, 8($16)

lw $11, 12($16)

lw $9, 16($16)

lw $12, 20($16)

lw $10, 24($16)

lw $13, 28($16) //then we make add instructions

add $4, $7, $4

add $8, $11, $8

add $9, $12, $9

add $10, $13, $10 //then store instructions

sw $4, 0($23)

sw $8, 4($23)

sw $9, 8($23)

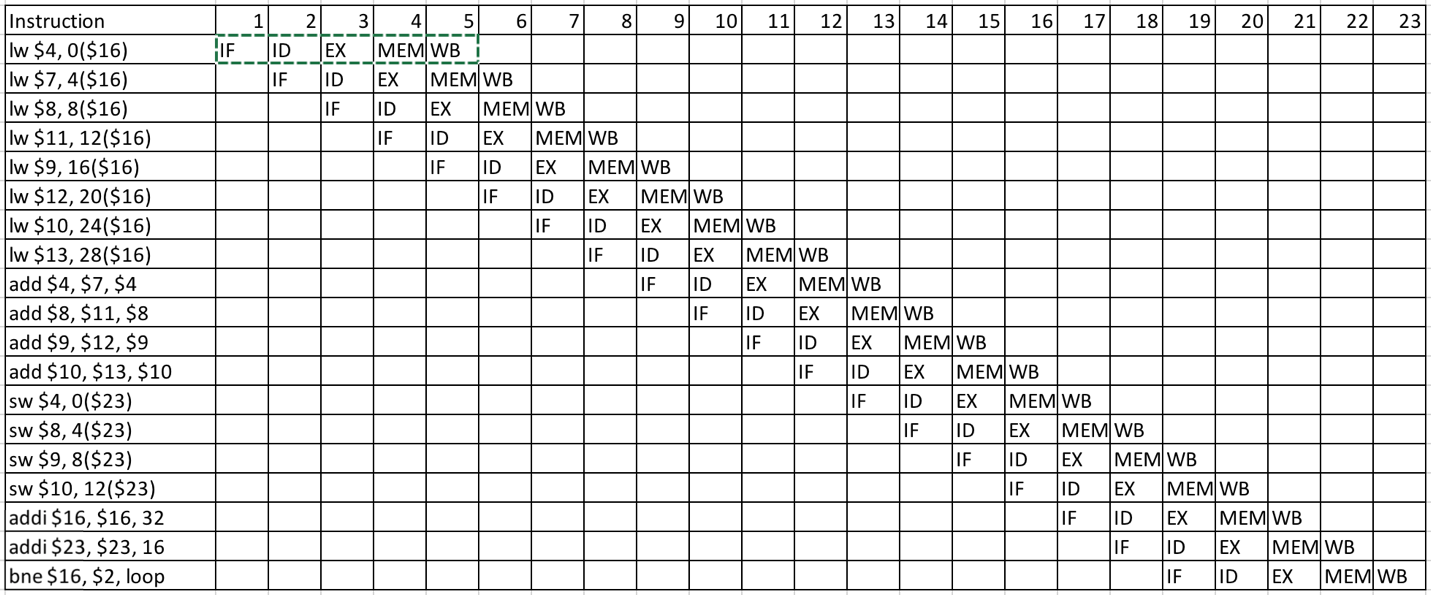
sw $10, 12($23) //Finally, addi instructions

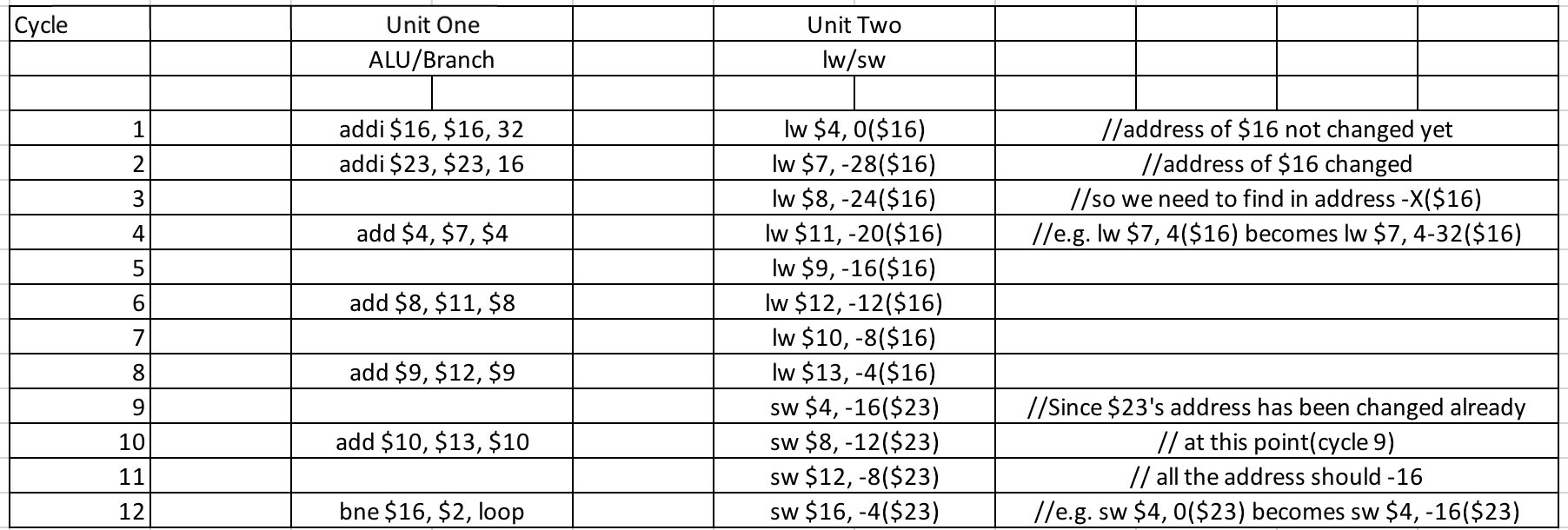
addi $16, $16, 32 //8+8+8+8 = 32

addi $23, $23, 16 //4+4+4+4 = 16

bne $16, $2, loop

5-Stage MIPS Pipeline Graph:



1.   
   According to the question, we have static 2-issue MIPS processor. One unit for ALU/Branch, and the other one for lw/sw.