

ENGR xD52: Quiz 0

This quiz is to be done closed notes and without human interaction. You may use MARS and its help file.

Describe all work. You have 30 minutes. Email to comparch2012@gmail.com

The Best Slide Ever

Refer to The Best Slide Ever for this section:

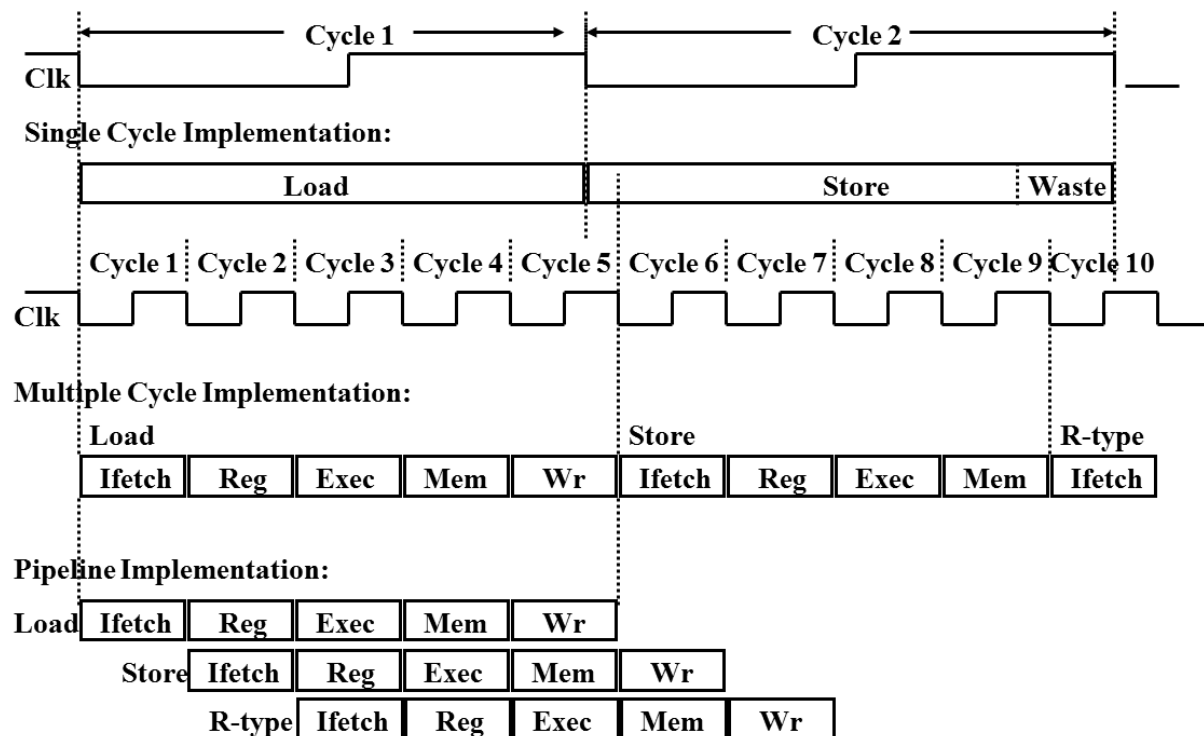


Figure 1: This is Just a Tribute

- Generally speaking, which implementation style optimizes:
 - Average Instruction Throughput
 - Average Instruction Latency
 - Maximum Instruction Latency
 - Functional Unit Re-use
- Rank the implementation styles in order of *increasing* instruction decoder complexity.
- What is the approximate Balancing Penalty of the Multi-Cycle Implementation? Give your answer as a percentage. Give your reasoning in 1 to 2 sentences.
- From the diagram it appears that the Multi-Cycle and Pipelined Implementations have similar clock periods. Is this a reasonable and/or safe assumption? Support your answer in 2-3 sentences.

The Best Program Ever

(From http://www.cs.bilkent.edu.tr/~will/courses/CS224/MIPS%20Programs/vowel_a.htm)

```
#-----
# vowelp - takes a single character as a
# parameter and returns 1 if the character
# is a (lower case) vowel otherwise return 0.
# a0 - holds character
# v0 - returns 0 or 1
#-----

vowelp:  li $v0,0
        beq $a0,'a',yes
        beq $a0,'e',yes
        beq $a0,'i',yes
        beq $a0,'o',yes
        beq $a0,'u',yes
        jr $ra
yes:     li $v0,1
        jr $ra

#-----
# vcount - use vowelp to count the vowels in a
# string.
# a0 - holds string address
# s0 - holds number of vowels
# v0 - returns number of vowels
#-----

vcount:
        sub $sp,$sp,16 # save registers on stack
        sw $a0,0($sp)
        sw $s0,4($sp)
        sw $s1,8($sp)
        sw $ra,12($sp)

        li $s0,0        # count of vowels
        move $s1,$a0     # address of string

nextc:   lb $a0,($s1)     # get each character
        beqz $a0,done    # zero marks end
        jal vowelp       # call vowelp
        add $s0,$s0,$v0  # add 0 or 1 to count
        add $s1,$s1,1    # move along string
        j nextc

done:    move $v0,$s0     # use $v0 for result

        lw $a0,0($sp)    # restore registers
        lw $s0,4($sp)
        lw $s1,8($sp)
        lw $ra,12($sp)
        add $sp,$sp,16
        jr $ra
```

5. Fill out the following chart. This may require you to make assumptions for missing data. Give explanation about any assumed information. Express Type Frequency as a percentage or as a fraction.

Instruction Type	Type Frequency	CPI (Single Cycle)	CPI (Multi Cycle)	CPI (Pipeline)
ALU		1	4	1
Load		1	5	2
Store		1	4	1
Branch		1	3	2
Jump (and Link)		1	2	1
Average CPI:				
Average Instructions / Second				

6. Compare the execution speed of The Best Program Ever on the 3 given processors. How much faster is the fastest option than the slowest?

Bonus:

What is overall speedup if the Pipeline processor implements:

Caching to reduce Load to 1 CPI?

Branch Prediction: Never Take?

Branch Prediction: Always Take?