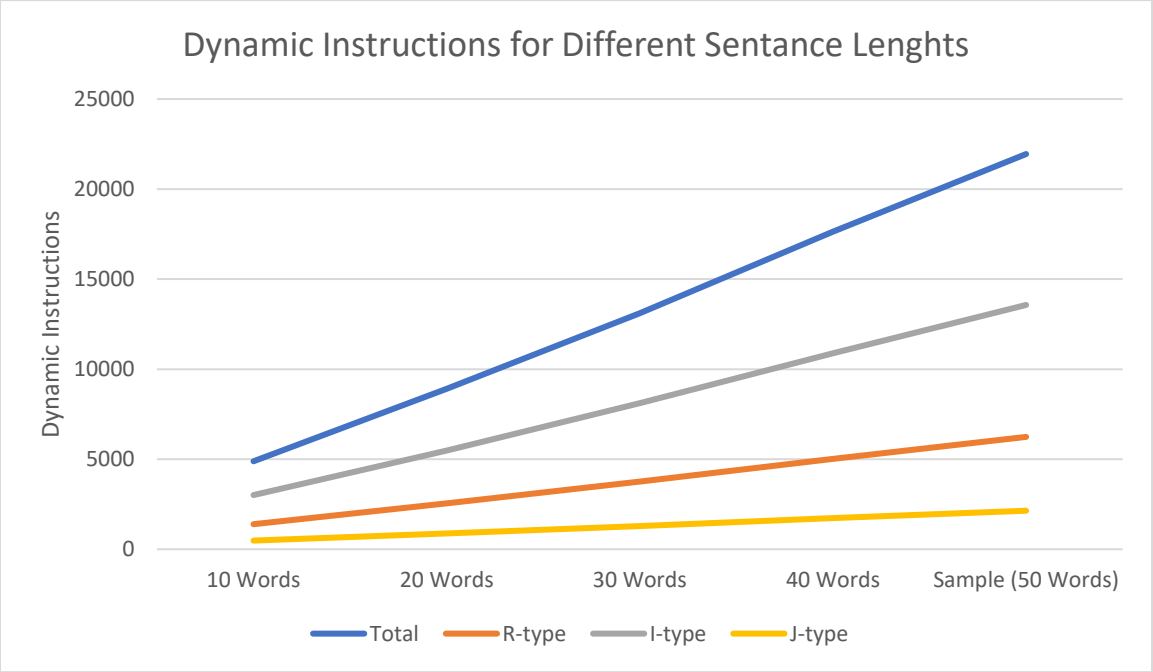
Bradley Grose

CSCE 212 Fall 2020

Dr. Wang

Project 3

1.0 Dynamic Instruction Count and CPI



CPI Calculations:

For all lengths of sentences tested, the resulting percentage of R-type, I-type, and J-type were the same, with it being 28%, 61% and 9% respectively. Given that R requires 2 clocks, I requires 3 clocks, and J requires 4 clocks. These will all have the same CPI for each type, however using different values. I will be using the following equation:

$$CPI = \frac{rClock(rCount) + iClock(iCount) + jClock(jCount)}{Total\ Count}$$

$$10\ Word\ CPI = \frac{2(1393) + 3(3010) + 4(482)}{4885} = 2.813\ CPI$$

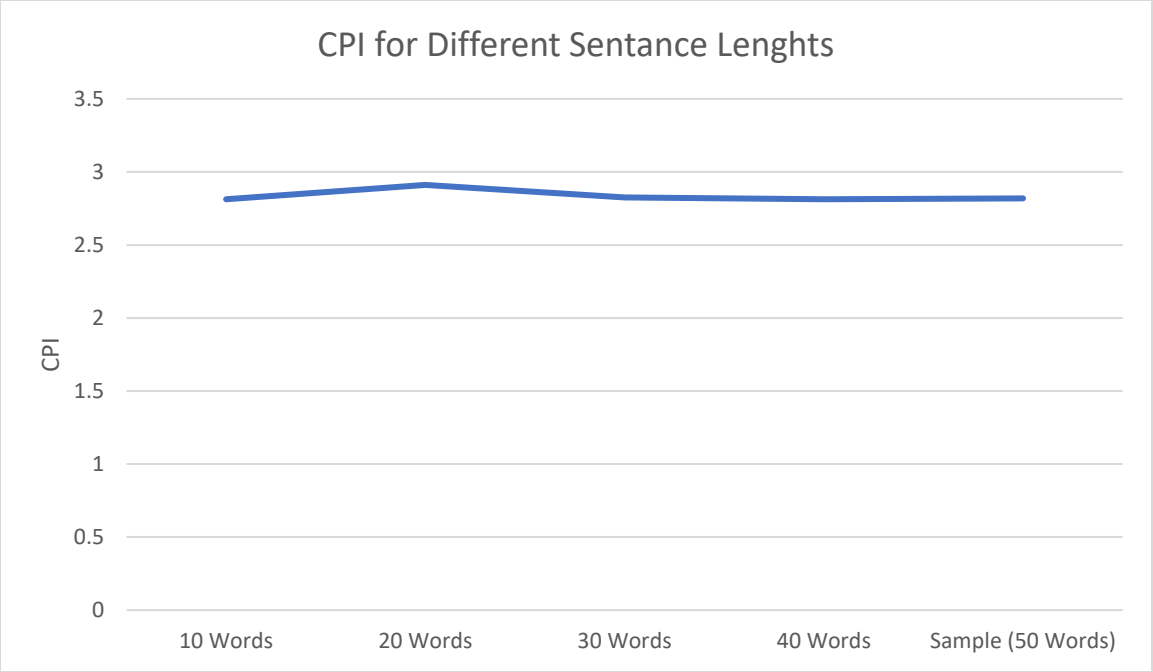
$$20\ Word\ CPI = \frac{2(2545) + 3(5482) + 4(878)}{8905} = 2.911\ CPI$$

$$30\ Word\ CPI = \frac{2(3756) + 3(8114) + 4(1291)}{13105} = 2.825\ CPI$$

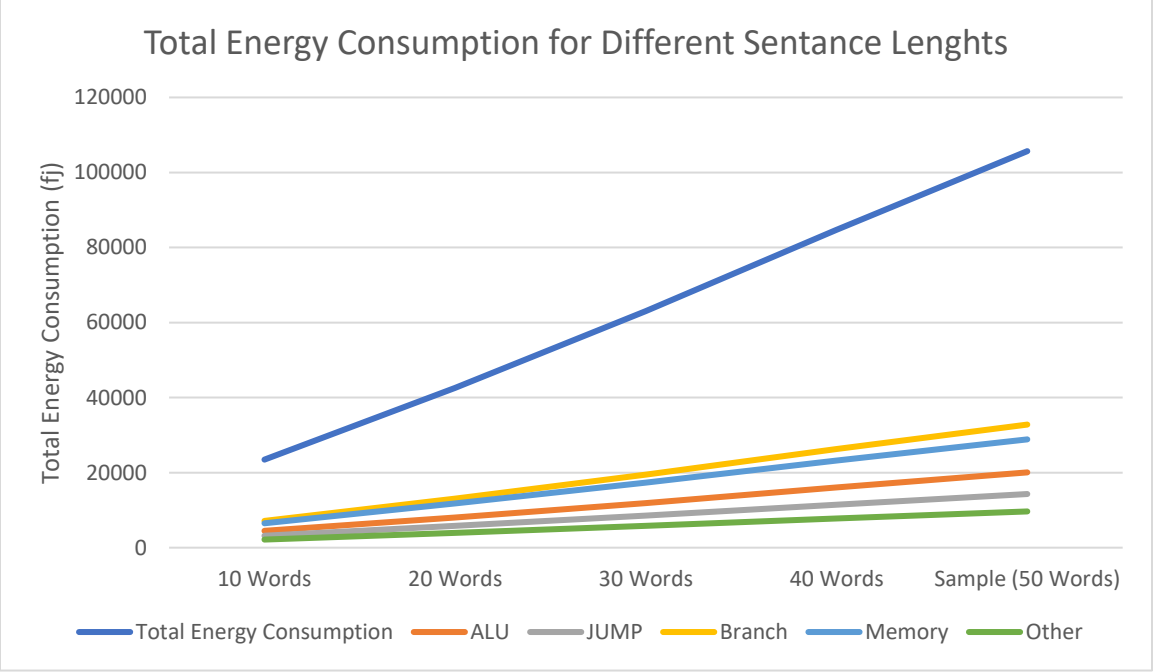
$$40\ Word\ CPI = \frac{2(5020) + 3(10886) + 4(1725)}{17631} = 2.813\ CPI$$

$$50\ Word\ CPI = \frac{2(6293) + 3(13565) + 4(2141)}{21945} = 2.818\ CPI$$

Therefor the CPI for all work lengths is roughly around 2.813-2.911 CPI based on the above calculations.



2.0 Energy Consumption



Calculations used: Most of the math was done using excel functions to solve, but this is the equation used that the numbers were plugged into

$$Energy\ Consumption = 3(ALU) + 4(Jump) + 6(Branch) + 20(Memory) + 2(Other)$$

Optimize for Energy Consumption

```
.data
        statement: .asciiz “#Put Statement Here”
        prompt1: .asciiz "Please input first word: "
        prompt2: .asciiz "Please input second word: "
        colonspace: .asciiz ": " #For Output of String and Result
        firstWord: .space 10       #Storage for Word 1
        secondWord: .space 10    #Storage for Word 2
        newline: .asciiz "\n"       #Creates a new line of seperation

.text
Main:
        #Prompt and Collect words
        # 16 Char limit is in A1 (17)
        li $v0,4
        la $a0,prompt1             #Print Prompt 1
        syscall
        li $v0,8
        la $a0, firstWord
        li $a1, 17
        syscall                    # reads string to firstWord
        li $v0,4
        la $a0,prompt2             #Print Prompt 2
        syscall
        li $v0,8
        la $a0, secondWord
        li $a1, 17
        syscall                    # reads string to secondWord
        #Set Counter and Load Statement
        la $t0, statement          #Statement
        li $t3, 0           #Counter
start1:  #Used to call firstword into $t1
        la $t1,firstWord
searchFirstWord:
        lb $t4, ($t0) #Statement
        lb $t5, ($t1) #Word One
        beq $t5, '\n', increase1 #tells to move forward
        beqz $t5, increase1 #tells to move forward
        beqz $t4, end1  #End of statement
        move $a0,$t4       #Makes Statemnt UpperCase
        jal toUpper
        move $t4,$v0       #Restores
        move $a0,$t5       #Makes Search UpperCase
        jal toUpper
        move $t5,$v0       #Restores
        bne $t4, $t5, nextChar1 #Goes to Next Char if not equal
        addi $t0, $t0, 1           #Increase Position in statement
        addi $t1, $t1, 1           #Inceases Position in firstWord
        j searchFirstWord          #jumps back to start
nextChar1:
```

```
        la $t6, firstWord       #loads word into t6
        bne $t6, $t1, start1    #if original not equal to current
        la $t1, firstWord       #sets original to correct
        addi $t0, $t0, 1        #Moves forward in statement
        j searchFirstWord       #Jumps to loop
increase1:
        addi $t3, $t3, 1        #match found
        la $t1, firstWord       #reset the word looking
        j searchFirstWord       #continue searching
end1:
        li $v0,4
        la $a0,firstWord                #Print Prompt 1
        syscall
        la $a0, colonspace      #Prints the ": "
        li $v0, 4
        syscall
        move $a0, $t3           #Prints the count Found
        li $v0,1
        syscall         # print new line
        la $a0, newline
        li $v0,4
        syscall
#Start of Second Word
        li $t3, 0       #Reset Counter
        la $t0, statement       #Reloads statement
start2:
        la $t2,secondWord       #loads in second word
searchSecondWord:
        lb $t4, ($t0) #Statement
        lb $t5, ($t2) #Word Two
        beq $t5, '\n', increase2   #increase count
        beqz $t5, increase2     #if a value increase
        beqz $t4, end2          #if statement is done
        move $a0,$t4      #Makes Statemnt UpperCase
        jal toUpper
        move $t4,$v0
        move $a0,$t5     #Makes Search UpperCase
        jal toUpper
        move $t5,$v0
        bne $t4, $t5, nextChar2 #Goes to Next Char if not equal
        addi $t0, $t0, 1        #Increase Counter
        addi $t2, $t2, 1        #Increase index
        j searchSecondWord

nextChar2:
        la $t6, secondWord      #loads word into t6
        bne $t6, $t2, start2    #if original not equal to current
        la $t2, secondWord      #sets original to correct
        addi $t0, $t0, 1        #Moves forward in statement
        j searchSecondWord      #Jumps to loop
increase2:
```

```
        addi $t3, $t3, 1        #match found
        la $t2, secondWord      #reset the word looking
        j searchSecondWord      #continue searching
end2:
        li $v0,4
        la $a0,secondWord               #Print Prompt 1
        syscall
        la $a0, colonspace      # prints ": "
        li $v0, 4
        syscall
        move $a0,$t3            # prints the count
        li $v0,1
        syscall         # print new line
        li $v0, 10              #tells code to exit
        syscall
toUpper:
  move $v0,$a0
  blt $a0,'a',return
  bgt $a0,'z',return
  addi $v0,$a0,-32      #Makes it Upper Case
return:
        jr $ra
```

Calculations

$$Energy\ Consumption = 3(ALU) + 4(Jump) + 6(Branch) + 20(Memory) + 2(Other)$$
$$Original - New = X\ Energy\ Optimization$$

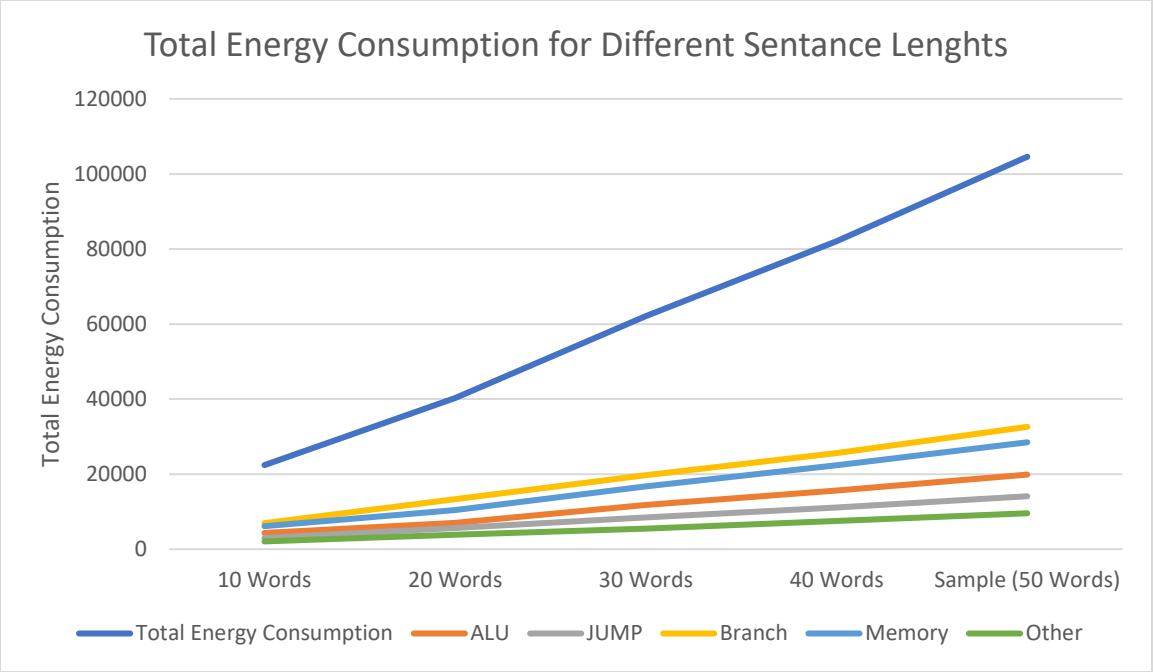
$$10\ Word = 3(1438) + 4(746) + 6(1153) + 20(306) + 2(1029) = 22394\ fJ$$
$$23466 - 22394 = 1072fj\ Energy\ Optimization$$

$$20\ Word = 3(2342) + 4(1403) + 6(2217) + 20(834) + 2(2732) = 40222\ fJ$$
$$42590 - 40222 = 2368fj\ Energy\ Optimization$$

$$30\ Word = 3(3921) + 4(2117) + 6(3281) + 20(834) + 2(2732) = 62061\ fJ$$
$$63030 - 62061 = 969fj\ Energy\ Optimization$$

$$40\ Word = 3(5213) + 4(2763) + 6(4264) + 20(1116) + 2(3746) = 82087\ fJ$$
$$84750 - 82087 = 2663fj\ Energy\ Optimization$$

$$50\ Word = 3(6620) + 4(3523) + 6(5432) + 20(1424) + 2(4779) = 104582\ fJ$$
$$105684 - 104582 = 1102fj\ Energy\ Optimization$$



3.0 MIPS/mW

Below is the math solving for the MIPS/mW for different sentence lengths. (Note 1fJ = $1*10^{-15}$ J)

$$\frac{MIPS}{mW} = \frac{Instruction\ Count}{Energy(fJ) * 10^{-15} * 10^{9}} = \frac{Instruction\ Count}{Energy(fJ) * 10^{-6}}$$

$$10\ Word\ \frac{MIPS}{mW} = \frac{4885}{22394 * 10^{-6}} = 218138$$

$$20\ Word\ \frac{MIPS}{mW} = \frac{8905}{40222 * 10^{-6}} = 221396$$

$$30\ Word\ \frac{MIPS}{mW} = \frac{13105}{62061 * 10^{-6}} = 211163$$

$$40\ Word\ \frac{MIPS}{mW} = \frac{17631}{82087 * 10^{-6}} = 214784$$

$$50\ Word\ \frac{MIPS}{mW} = \frac{21945}{104582 * 10^{-6}} = 209835$$

As seen in my calculations above, all of the MIPS/mW are within a similar range of 209835-221396 MIPS/mW. By using the simplified equation I made it so that I could input the fJ value and it would convert it to J. Then this could be used by combining it with the 10^9 that was already needed using the original formula. The resulting answers are what came when using the numbers for each sentence length.