Max Shi

CS 383 C

Professor Barbalace

I pledge my honor that I have abided by the Stevens Honor System.

2.1

SUB X9, X2, #5  
ADD X0, X1, X9

2.3

SUB X9, X3, X4 // X9 = i-j  
LSL X9, X9, #2 // X9 = index shifted by size of int (4 bytes)  
ADD X10, X9, X6 // X10 = memory index of element  
LDUR X9, [X10, #0] // load element in X9  
LSL X10, #8, #2 //get byte index as index 8 \* 4 bytes  
ADD X10, X10, X7 // get position in array  
STUR X9, [X10, #0] //store X9 in array

2.4

f = A[f]  
B[g] = A[f+1] + f

X10 is B[g]  
X11 is &A[f+1]  
X9 is A[f+1]  
X9 is now A[f+1] + f  
stored in B[g]

2.5

LSL X9, X0, #3  
ADD X9, X6, X9  
LSL X10, X1, #3  
ADD X10, X7, X10  
LDUR X0, [X9, #0]  
LDUR X9, [X9, #8]  
ADD X9, X9, X0  
STUR X9, [X10, #0]

2.9

A[1] = &A[0]  
f = &A[0] + &A[0]

2.10

1. opcode = 1001000100, immediate = 000000010000, Rn = 00110, Rd = 01001

2. opcode = 10001011000, Rn = 00110, Rm = 11111, Rd = 01010

3. opcode = 11111000000, Rn = 01001, Rd = 01010

4. opcode = 11111000010, Rn = 01001, Rd = 01001

5. opcode = 10001011000, Rn = 01001, Rm = 01010, Rd = 00000

2.22

X1 = 2

2.25.1

X0 = 20

2.41

Original program:

Assume 1 \* 10-9 seconds per clock

2.42.1.

Original frequency \* 1.1 = 1.1\*10-9 seconds per clock

This would not be a good design choice as this would increase the overall time the program takes to run by about 6.4%.

2.42.2.

Double the arithmetic performance = 50% of the arithmetic instructions.

Assume 1 \* 10-9 seconds per clock

10 times performance = 10% of the arithmetic instructions

Assume 1 \* 10-9 seconds per clock

2.42

1. Weighted average = 0.7 \* 2 + 0.1 \* 6 + 0.2 \* 3 = 2.6 average CPI

2. 25% improvement in performance

An arithmetic instruction must take 1.07 clocks per instruction on average.