CS385

Homework 02

Julia Nelson

9/20/19

PLEDGE: "I pledge my honor that I have abided by the Stevens Honor System" J Nelson

PROBLEMS: 2.1, 2.3, 2.4, 2.5, 2.9, 2.10 (w/ table), 2.22, 2.25.1, 2.41.1, 2.41.2, 2.42.1, 2.42.2

2.1:

C variables f, g, h placed in registers X0, X1, X2

Use minimal LEGv8 assembly to write LEGv8 assembly code

```
f = g + (h-5);

SUB   X3, X2, 5      // register X3 holds h-5
ADD   X0, X1, X3     // f = g + X3 (h-5) stored in X0
```

2.3:

f, g, h, i, j stored in X0, X1, X2, X3, X4

Base address of arrays A, B stored in X6, X7

```
B[8] = A[i-j];

SUB   X5, X3, X4     // stores i-j in X5
LSL   X5, X5, #3     // X5 = (i-j) * 8 ← offset 8
ADD   X5, X5, X6     // A[i-j] stored
LDUR  X8, [X5, #0]   // X8 = A[i-j]
STUR  X8, [X7, #64]  // store A[i-j] in B[8]
                        #64 from 8*8 offset
```

2.4:

f in X0, h in X2, j in X4,
g in X1, i in X3,

```
LSL  X9, X0, #3      // X9 = f * 8
...
ADDI X11, X9, #8     // X11 = &A[f+8]
LDUR X9, [X11, #0]   // X9 = A[X11 + 8]
ADD  X9, X9, X0      // X9 = A[f] + A[f+1]
STUR X9, [X10, #0]   // X10 = X9 → B[g] = A[f] + A[f+1]
```

C statement: B[g] = A[f] + A[f+1]

2.9:

F x0
g x1
h x2
i x3
j x4
Arrays A x6
B x7

// Reads through #s of A
// sums the elements indiv.
// Stores results in x9 and x10

```
int f, g, h, i, j;
int A[2];
f = A[0] + A[1] + A[2]
```

2.10:

- Show value of opcode for each LEGv8 inst. 2.9, source register, target Register fields. (Op, Rn, Rd or Rt)
- For I-type inst show val of imm. field
- R-type inst show val of 2nd source register (Rm)

Instruction:	Opcode:	I → immediate R → Rm	Rn	Rd or Rt
ADDI X9, X6, #8	1001000100	immediate → 000000001000	00110	Rd: 01001
ADD X10, X6, XZR	10001011000	Rm → 00000	00110	Rd: 01001
STUR X10, [X9, #0]	11111000000		01001	Rt: 01010
LDUR X9, [X9, #0]	11111000010		01001	Rt: 01001
ADD X0, X9, X10	10001011000	Rm → 01010	01001	Rd: 00000

2.22:

X0 holds 0x0000000000101000
X1 = ? After inst.

```
CMP X0, #0      // X0 compared to 0
B.GE ELSE       // if X0 ≥ 0 → go to ELSE
B DONE          // → go to DONE
ELSE: ORRI X1, XZR, #2  // X1 = XZR (bitwise) OR 2 → 0 OR 2
DONE:
```

X1 = 2 because OR is b/w 0 and 2, so we take value of 2

2.25.1:

X1 initialized to value 10
Whats final val of X0 when initially 0

```
LOOP: SUBIS X1,X1,#0      // X1 = X1 - 0
      B.LE DONE           // IF X1 ≤ 0 → go to DONE
      SUBI X1,X1,#1       // X1 = X1 - 1
      ADDI X0,X0,#2       // X0 = X0 + 2
      B LOOP              // → go to B LOOP (recursive)
DONE:
```

X1 = 10 → X1 = 9	X0 = 0 → X0 = 2
→ X1 = 9 → 8	X0 = 4
→ X1 = 8 → 7	X0 = 6
→ X1 = 7 → 6	X0 = 8
→ X1 = 6 → 5	X0 = 10
→ X1 = 5 → 4	X0 = 12
→ X1 = 4 → 3	X0 = 14
→ X1 = 3 → 2	X0 = 16
→ X1 = 2 → 1	X0 = 18
→ X1 = 1 → 0	X0 = 20

Final value X0 = 20

2.41.1:

INITIAL CPI:

$$\rightarrow 10 * \frac{300}{900} + 1 * \frac{500}{300} + 3 * \frac{100}{900} = \frac{30}{9} + \frac{5}{9} + \frac{3}{9}$$

INITIAL TIME: $38/9 * 900 * f = \underline{3800f} = \underline{4.22}$

* New More Powerful Instruction *

reduce # arithmetic inst. by 25% w/ only 10% time increase
Good or No? Why?

(100 - 25%)

$0.75 * 500 = 375$ → New # arithmetic inst.

New CPI → $10 * \frac{300}{775} + 1 * \frac{375}{775} + 3 * \frac{100}{775} = \frac{3675}{775} = 4.742$

New TIME → $\frac{3675}{775} * f * 1.1 * 775 = 4042.5f$

Not a good choice because of large time increase

2.41.2 :

INITIAL CPI: 4.22 INITIALTIME = 3800f

New CPI :

$10 * \frac{3}{9} + \frac{1}{2} * \frac{5}{9} + 3 * \frac{1}{9} = \frac{35.5}{9} = 3.944$

4.22/3.944 = 1.069 speedup

* Improve arithmetic inst. by 10 *

New CPI =

$\frac{30 + \frac{1}{2} + 3}{9} = \frac{33.5}{9} = 3.722$

38/33.5 = 1.134 speedup

2.42.1

Avg CPI: 70% Arith Inst takes 2 cycles
10% load/store takes 6 cycles
20% branch inst takes 3 cycles

$(0.7 * 2) + (0.1 * 6) + (0.2 * 3) = 2.6$

2.42.2

25% improvement performance

Avg # cycles of arithmetic take?
if load/store + branch not improved

CPI = $1.25 * 2.6 = 3.25$

New CPI = $3.25 = (0.7 * X) + (.10 * 6) + (.2 * 3)$

~~$3.25 = 2x + 1.2 \rightarrow 2x = 2.05$~~

$2.05 = 0.7X \rightarrow X = 2.928$

= 2.93
cycles of Arithmetic