Name: Georges Hatem

CS 550 Homework Assignment 2

Date: October 17, 2021

Problem 1:

As a note: we have not learned the usage of MUL yet. The book used MUL in Section 2.8 example 2.8.2, and it said that “we assume a multiply instruction is available even though it is not covered until COD Chapter 3 (Arithmetic for Computers):” The else statement contains a multiplication. So, I can use the MUL, but the issue is that it is multiplied by a constant so it may be MULI. Since we are multiplying by 2, I just used addition as you can see below.

```
SUBS XZR, X19, X20
B.GE L1
ADD X19, X19, X20
L1: ADD X20, X20, X20
```

Problem 2:

As a note: considering i is in register X19.

```
      ADDI X19, XZR, #0
Loop: SUBIS XZR, X19, #100
      B.GE Exit
      LSL X10, X19, #3
      ADD X10, X10, X20
      LDUR X9, [X10, #0]
      ADDI X9, X9, #1
      STUR X9, [X10, #0]
      ADDI X19, X19, #1
      B Loop
Exit:
```

Problem 3:

As a note: considering a and b are in registers X1 and X2, respectively and f, g, and y are in registers X4, X5, and X6, respectively. Also, when calling BL sum, I am calling BL X0+1000. I am calling the address for the sum function. And since main is at address X0+800, BR LR returns the value at address X0+804 (the BL instruction saves PC+4 in register LR) and hence the reason why I am loading (LDUR) its value from X0+804.

```
sum: SUBI SP, SP, #8
        STUR X19, [SP, #0]
        ADD X19, X1, X2
        ADD X3, X19, XZR
        LDUR X19, [SP, #0]
        ADDI SP, SP, #8
        BR LR

main: ADDI X4, XZR, #2
         ADDI X5, XZR, #3
         BL sum
         LDUR X20, [X0, #804]
         ADD X6, X20, XZR
```

Problem 4:

Instruction Type:

To figure out the instruction type and write down the assembly language instruction and binary representation of instruction, we need to get the decimal value of the Op. Op is in hexadecimal. Let's first change it from hexadecimal to binary representation and then change it to decimal.

$$Op = 0X7c2$$

2 in 4 bits binary number is 0010, c in 4 bits binary number is 1100, and 7 in 4 bits binary number is 0111. So, the 12-bit binary representation of Op is as follows:

$$Op = 011111000010$$

Computing Op in decimal gives me:

$$Op = (2^1) + (2^6) + (2^7) + (2^8) + (2^9) + (2^{10}) = 1986$$

The load register (LDUR) has an Op of 1986.

So, the type of the instruction is a D-type instruction.

Assembly Language Instruction:

This is a load register with Rn = 11, Rt = 4, and address = 0x4

0x4 is 4 in decimal, which means #4 as you can see below. Rn is 11 means X11 and Rt is 4 means X4. So, the following is the assembly language instruction for the instruction:

LDUR X4, [X11, #4]

Binary Representation of Instruction:

OpCode, address, Op2, Rn, and Rt must have 11 bits, 9 bits, 2 bits, 5 bits, and 5 bits respectively. So, the only task left is to convert the decimal and hexadecimal numbers into binary numbers to get the binary representation of the instruction.

The binary representation of 0X7c2:

1) 7 is 0111 represented as a 4-bit binary number
2) c is 1100 represented as a 4-bit binary number
3) 2 is 0010 represented as a 4-bit binary number

So, the binary representation of 0X7c2 in 11 bits is: 11111000010

The binary representation of 0x4 in 9 bits is:

1) 0x4 is 4 in decimal. So, it will be represented as 000000100 in 9 bits

The binary representation of Op2 in 2 bits is 00 since Op2 is 0

The binary representation of Rn in 5 bits is:

1) 01011 since Rn is 11 in decimal

The binary representation of Rt in 5 bits is:

1) 00100 since Rt is 4 in decimal

So, the binary representation of the instruction is as follows below and in the table: (11111000010000000100000101100100)

Op	Address	Op2	Rn	Rt
0X7c2	0x4	0	11	4
11111000010	000000100	00	01011	00100

Problem 5:

As a note: shift left number 2 by 3 (doubleword – 8 bytes) and this will give you the #16 below

```
LDUR X9, [X11, #16]
LSL X9, X9, #4
```

Problem 6:

```
for (int i = 0; i < 100; i++){

        result = result + MemArray[i];

}
```