Jacob Howard

ELEC 5200/6200 (Fall 2021)
Homework 2
Assigned 09/01/21, due 09/08/21

Note: This homework will be easier if you have completed the assigned reading for Chapter 2. You also might find the RISC-V reference data sheet useful (located on the green insert inside your textbook).

Question 1: Show how the value 0x EFBEADDA would be arranged in the memory of a big-endian machine. Assume that the data is stored starting at address 0. Show your answer in hexadecimal. (2 pts)

Address	0x03	0x02	0x01	0x00
Byte	0xDA	0xAD	0xBE	0xEF

Question 2: Show how the binary value 1111 1010 1100 1110 1011 1110 1010 1101 would be arranged in the memory of a little-endian machine. Assume that the data is stored starting at address 0. Show your answer in hexadecimal. (2 pts)

Address	0x03	0x02	0x01	0x00
Byte	0x FA	0x CE	0xBE	0xAD

Question 3: For the following C statement, assume that variable **f** is already stored in register x7, **g** in register x6, and **h** in x5.

```
f = g + (h - 11)
```

a) Write the corresponding RISC-V assembly code. (4 pts)

ADD x5, x5, -11

ADD x7, x6, x5

b) Write the machine language for the assembly code you came up with in part a. (4 pts)
Note: write the binary in groups of 4 for legibility. Ex: 0000 1111 0000 1111

1. 1111 1111 0101 0010 1000 0010 1001 0011

2. 0000 0000 0101 0011 0000 0011 1011 0011

Question 4: Consider the following assembly code written using the RISC-V instruction set:

```
        addi    x29, x0, 100
        add     x18,  x0, x0
        addi    x28, x18, 1
        add     x19,  x18, x28
        sw      x18, 0(x11)
        sw      x19, 4(x11)
        addi    x11,  x11,  8
        addi    x10,  x0,  1

LOOP:   add     x10,  x10,  x28  // ++n
        lw      x5,  -4(x11)     // x5 = result [n-1]
        lw      x6,  -8(x11)   //x6 = result [n-2]
        add     x7,  x5,  x6
        sw      x7, 0(x11)       // x7 = result[n]
        addi    x11,  x11,  4   // result++
        blt     x10,  x29,  LOOP
```

Translate the code above into C. Assume that the integer `n` is the loop variable and is located in register x10. Register x11 is initialized to the base address of the integer array named `Result`. Further assume that 32-bit integers are used. (5 pts)

```
int result[100], n;        // define array and loop var
result[0]=0;               // -sd x18 0(x11)
result[1] = 1              // - sd x19, 8(x11)

for (n=2; n<100; n++){
  result [n] =result [n-1]+ result[n-2];
}
```

- Sets first ele lement in array to 0 and Second element to 1
- loop starts from 3rd element in array and goes to 100th
- for each element from 3rd on, the Progam would Set the element to the sum of the last 2 elements
- Stops once counter reaches 100

Question 5: If the current value of the PC is 0x01027B08 can you use a single RISC-V **jal** instruction to get to PC address 0x12345678? Explain your answer. (5 pts)

Jal x1, #0x 0x12345678 - 0x01027B08 = 0x1131DB70

Jal permits target to be within ± 1MByte only

So Jal cannot be performed here

Only 20 bits are allowed to specify displacement

Question 6: If the current value of the PC is 0x0FFFFFF0 can you use a single RISC-V **branch** instruction to get to PC address 0x10000080? Explain your answer. (5 pts)

Jalx1, #0x24

0x10000080 - 0x0FFFFFF0 = 0x00000090

Should branch to 0x90 bytes ahead

So target address is 0x24 (0x90/4=0x24)

Only [illegible] to