CS2323: Computer Architecture, Autumn 2023

Homework-1: RISC-V Assembly

---

1. Write an assembly instruction to achieve the given functionality, defined using C-language syntax (only 1 instruction to be used). Explain in brief. [6 marks]

   a. x8 = x5 - 5
   b. x5 = x3 * 8
   c. x19 += x10
   d. ++x15
   e. x9 = x15/4
   f. x12 = 24

2. Consider an array M consisting of 8 byte integers. The base address of M is stored in register x5. Write the assembly code that achieves each operation given below. [1+1+1+2+2 marks]

   a. M[12] = M[20] + 100
   b. M[20] ++
   c. swap M[5] and M[12]
   d. Make the first 32-bits (from MSB side) of M[4] as 0
   e. Swap the most significant 32-bits of M[2] with its least significant 32-bits

3. Write the following decimal numbers in their 2's complement representation, using 8-bits. Show your calculations. [4 marks]

   a. +23
   b. -1
   c. +255
   d. -128

4. Write the equivalent decimal number for given numbers in 2's complement format. Show your calculations. [3 marks]

   a. 11010100
   b. 00101011
   c. 11111110

Submission instructions:

1. Create a pdf file (write and scan) for the questions asked above. Be to the point without too much of explanation. Ensure that the scan/picture is readable.
2. The submission should be entirely your work
3. The pdf file should be named YOUR_ROLLNUM.pdf (e.g., CSYYBTECHXXXXX.pdf)
4. Submit the pdf file
5. Deadline: 23-Sep-2023, 11.59 pm

Solutions :

1. a) x8 = x5 - 5 ⇒

This instruction can be seen as adding -5 to x5 and updating that to the value of x8.
So , the assembly instruction would be,

addi x8 , x5 , -5

b) x5 = x3 * 8 ⇒

This instruction can be seen as multiplying 8 to x3 and updating that to the value of x5.
Multiplying 8 to x3 would mean shifting the bits in x3 to the left 3 times.
So , the assembly instruction would be,

slli x5 , x3 , 3

c) x19 += x10 ⇒

This instruction is adding x10 to x19.
So, the assembly instruction would be,

add x19 , x19 , x10

d) ++x15 ⇒

This instruction means adding 1 to x15
So , the assembly instruction would be,

addi x15 , x15 , 1

e) x9 = x15/4 ⇒

This instruction can be seen as dividing x15 by 4 and updating that to the value of x9.
Dividing x15 by 4 would mean shifting the bits in x15 to the right 2 times.
So , the assembly instruction would be,

srli x9 , x15 , 2

f) x12 = 24 ⇒

This instruction can be seen as adding 24 to x0 and updating the value of x12.
So , the assembly instruction would be,

addi x12 , x0 , 24

2.

The register x5 should be used as given in the solution. Using any other register numbers at other places may be okay. There is a possibility of other correct answers in a few cases.

a) M[12] = M[20] + 100

```
ld x6 160(x5)
addi x6 x6 100
sd  x6 96(x5)
```

b) M[20] ++

```
ld x6 160(x5)
addi x6 x6 1
sd x6 160(x5)
```

c) swap M[5] and M[12]
(the ordering might be different in different submissions, so it may be okay)

```
ld x6 40(x5)
ld x7 96(x5)
sd x6 96(x5)
sd x7 40(x5)
```

d) Make the first 32-bits (from MSB side) of M[4] as 0

```
sw x0 36(x5)
```

e) Swap the most significant 32-bits of M[2] with its least significant 32-bits

solution-1:

```
lw x6 16(x5)
lw x7 20(x5)
sw x6 20(x5)
sw x7 16(x5)
```

Solution-2:

```
ld x6 16(x5)            //load M[2] in x6
addi x7, x6, x0         //copy x6 to x7
slli x7, x7, 32         //The lower bits of x7 are now in MSB position
srli x6, x6, 32         //the upper bits of x6 are in LSB position
ori x6, x6, x7          //logical or of x6 and x7
sd x6, 16(x5)
```

3.

a) 23 = 2^4 + 2^2 + 2^1 + 2^0
Which means the binary representation in 8 bits is 00010111
As the given number is positive , it is the 2's complement itself.

b) 1 = 2^0
Which means the binary representation in 8 bits is 00000001
Since the given number is negative , we have to invert it and one.
So , the 2's complement representation is 11111110 + 1 = 11111111.

c) 255 = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0
8-bit 2's complement can support -128 to +127 only. The given number exceeds the supported range.

d) 128 = 2^7
Which means the binary representation in 8 bits is 10000000
Since the given number is negative , we have to invert it and one.

So , the 2’s complement representation is 01111111 + 1 = 10000000.

4.

a) 11010100 = -1*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0 = -44

b) 00101011 = 0*2^7 + 0*2^6 + 1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 43

c) 11111110 = 1*2^7 + 1*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0 = -2