Name : Pathri Vidya Praveen
Roll No. CS24BTECH11047

Homework 1

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 = x19 + 24

Solution:

a) addi x8 , x5 , -5

   Explanation:

   addi instruction adds immediate (-5 here) to register x5 and store it in register x8. Subtracting 5 is the same as adding -5.

b) slli x5 , x3 , 3

   Explanation:

   multiplying x3 by 8 is the same as doing left shift by 3 bits(2^3 = 8) so we do this with slli instruction and store the result in register x5.

c) add x19 , x19 , x10

   Explanation:

   x19 = x19 + x10. add values in registers x19 and x10 and store result in x19 using add instruction.

d) addi x15 , x15 , 1

Explanation:

x15 = x15 + 1. addi instruction adds immediate 1 to register content x15 and store it in x15.

e) Case 1 : Unsigned division

srli x9 , x15 , 2

Explanation:

Dividing by 4 is equivalent to right shifting by 2. So we use srli instruction for this.

Case 2 : Signed division

srai x9 , x15 , 2

Explanation:

Similar to srli instruction but here using srai instruction preserving the sign , we perform division by 2^n .

f) addi x12 , x19 , 24

Explanation:

addi instruction adds immediate value 24 to x19 register content and stores it in register x12.

---

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

Solution :

a) ld x6 , 160(x5) # x6 = M[20]
   addi x6 , x6 , 100 # x6 = M[20] + 100
   sd x6 , 96(x5) # M[12] = x6 = M[20] + 100

b) ld x6 , 160(x5) # x6 = M[20]
   addi x6 , x6 , 1 # x6 = M[20] + 1
   sd x6 , 160(x5) # M[20] = x6 = M[20] + 1

c) ld x6 , 40(x5) # x6 = temp1 = M[5]
   ld x7 , 96(x5) # x7 = temp2 = M[12]
   sd x7 , 40(x5) # M[5] = temp2
   sd x6 , 96(x5) # M[12] = temp1

d) ld x6 , 32(x5) # x6 = M[4]
   # AND it with no. 000….0(32 bits)111….1(32 bits)
   addi x7 , x0 , -1 #x7 = 111……1(64 bits)
   srli x7 , x7 , 32 # x7 = 000……0(32 bits)111……1(32 bits)
   and x6 , x6 , x7 # this makes upper 32 bits 0
   sd x6 , 32(x5) # stores back the result

e) ld x6 , 16(x5) # x6 = M[2] = AB
   # Let M[2] = AB where A is the most significant 32 bits of M[2] and B is the least
   # significant 32 bits of M[2]
   slli x7 , x6 , 32 # x7 = B000…..0(32 bits)
   srli x8 , x6 , 32 # x8 = 000….0(32 bits)A
   or x6 , x7 , x8 # x6 = BA (swapped)
   sd x6 , 16(x5) # store back

---

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

Solution:

a) +23 is valid in the range of -128 to 127 and is positive.
   +23 = 1x2^4 + 0x2^3 + 1x2^2 + 1x2^1 + 1x2^0 = 16 + 0 + 4 + 2 + 1
   Two's complement representation → 00010111

b) -1 is valid in the range of -128 to 127 and is negative.
   +1 in binary is 0000001 (binary)
   One's complement of +1 = 1111110
   Add 1 to ones complement i.e 11111111
   Two's complement representation → 11111111

c) +255 is not valid in the range of -128 to 127. So this cannot be represented in 8-bit two's complement form.

d) -128 is valid in the range of -128 to 127.
   Two's complement representation → 10000000

---

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

Solution:

a) MSB is 1 so the number is negative.
   Invert all bits → 00101011
   Add 1 → 00101100
   Convert to decimal → 1x2^2 + 1x2^3 + 1x2^5 = 32 + 8 + 4 = 44
   This is the negative of the given negative number
   Equivalent decimal number of 11010100 → - 44

b) MSB is 0 so the number is positive.
   1x2^0 + 1x2^1 + 1x2^3 + 1x2^5 = 1 + 2 + 8 + 32 = 43 .
   Equivalent decimal number of 00101011 → + 43

c) MSB is 1 so the number is negative.
Invert all bits → 00000001
Add 1 → 00000010
Convert to decimal → 1x2^1 = 2
This is the negative of the given negative number
Equivalent decimal number of 11111110 → - 2