Indian Institute of Technology
Hyderabad

CS2323 HW1

Mihir Divyansh E, EE23BTECH11017

August-November, 2025

1 Question 1

Write an assembly instruction to achieve the given functionality, defined using C-language syntax

1. x8 = x5 - 5
   This can be achieved using the instruction:
   ```
   sub x8, x5, 5
   ```
   It subtracts 5 from the value in register x5 and stores the result in register x8.

2. x5 = x3 * 8
   Use
   ```
   mul x5, x3, 8
   ```
   x3 is multiplied by 8, and the result is stored in x5.

3. x19 += x10
   Use
   ```
   add x19, x19, x10
   ```
   This adds the value in register x10 to the value in register x19 and stores the result in register x19.

4. ++x15
   This is an increment operation is done using:
   ```
   add x15, x15, 1
   ```
   It adds 1 to the value in register x15 and stores the result in register x15.

5. x9 = x15/4
   This is a shift operation.
   ```
   srl x9, x15, 2
   ```
   It performs a logical right shift on the value in register x15 by 2 bits, effectively dividing it by 4, and stores the result in register x9.

6. x12 = x19 + 24

   We can use add with immediate instruction:

   addi x12, x19, 24

2 Question 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. M[12] = M[20] + 100

```
addi x6, x5, 160    # Load address of M[20] into x6
ld x7, 0(x6)        # Load M[20] into x7
addi x7, x7, 100    # Add 100 to x7
addi x6, x5, 96     # Load address of M[12] into x6
sd x7, 0(x6)        # Store result into M[12]
```

2. M[20] ++

```
ld x7, 160(x5)      # Load M[20] into x7
addi x7, x7, 1      # Increment x7
sd x7, 160(x5)      # Store result back into M[20]
```

3. swap M[5] and M[12]

```
ld x6, 40(x5)
add x7, x0, x6      # copy from x6 to x7
ld x6, 96(x5)
sd x7, 96(x5)
sd x6, 40(x5)
```

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

```
ld x6, 32(x5)
slli x6, x6, 32      # Shift left by 32 bits
slri x6, x6, 32      # Shifting back will -
sd x6, 32(x5)        # make the first 32 bits 0
```

5. Swap the most significant 32-bits of M[2] with its least significant 32-bits

```
ld x6, 16(x5)
slli x7, x6, 32
srli x6, x6, 32
or x6, x6, x7
sd x6, 16(x5)
```

3 Question 3

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

1. +23
   The unsigned binary representation of 23 is $\{\lfloor\frac{23}{2^7}\rfloor, \lfloor\frac{23}{2^6}\rfloor, \lfloor\frac{23}{2^5}\rfloor, \lfloor\frac{23}{2^4}\rfloor, \lfloor\frac{23}{2^3}\rfloor, \lfloor\frac{23}{2^2}\rfloor, \lfloor\frac{23}{2^1}\rfloor, \lfloor\frac{23}{2^0}\rfloor\} = \{0,0,0,1,0,1,1,1\}$ which is the same as it's 2's complement representation as it is positive.

2. -1
   The binary representation of 1 is $\{0,0,0,0,0,0,0,1\}$. To find the 2's complement, we invert the bits and add 1 to it to get $\{1,1,1,1,1,1,1,0\} + 1 = \{1,1,1,1,1,1,1,1\}$.

3. +255
   255 requires 8 unsigned bits to represent. Since we have 8 bits signed, we cannot represent 255 in 8-bit signed.

4. -128
   in 2's complement, -128 is represented as $\{1,0,0,0,0,0,0,0\}$.

4 Question 4

Write the equivalent decimal number for given numbers in 2's complement format. Show your calculations

1. 11010100
   $-1\cdot 2^7 + 1\cdot 2^6 + 0\cdot 2^5 + 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 0\cdot 2^0 = -44$

2. 00101011
   $0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 43$

3. 11111110
   $-1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = -2$