Indian Institute of Technology
Hyderabad

CS2323 HW2

Mihir Divyansh E, EE23BTECH11017

August-November, 2025

1 Question 1

Write equivalent machine code (in hexadecimal) for the given assembly instructions, by highlighting the various fields in the 32-bits of the instruction:

1. addi x15, x22, -45
   This instruction uses I-Format encoding. The imm value here is -45, which in 2's complement form is 111111010011.

Immediate	rs1	Funct3	rd	Opcode
111111010011	10110	000	01111	0010011
		0xFD3B0793		

2. and x23, x8, x9
   This instruction uses R-Format encoding.

Funct7	rs2	rs1	Funct3	rd	Opcode
0000000	01001	01000	111	10111	0110011
			0x00947BB3		

3. blt x2, x11, 240
   This instruction uses B-Format encoding. The imm value here is 240, which in binary is 000011110000. Rearranged, it becomes 0000001111000000. We take the 12th bit to be imm[12], the next 6 bits (000011) to be imm[10:5] and the last 5 bits (11000) to be imm[4:1,11] and the last bit is imm[11].

Immediate[12,10:5]	rs2	rs1	Funct3	Immediate[4:1,11]	Opcode
00001110	01011	00010	100	10000	1100011
			0x0EB14863		

4. sd x19, -54(x1)
   This is a S-Format instruction. -54 in 2's complement format : 111111001010.

Immediate[11:5]	rs2	rs1	Funct3	Immediate[4:0]	Opcode
1111110	10011	00001	011	01010	0100011
			0xFD30B523		

5. jal x3, -10116
   This is a J-Format instruction. -10116 in 2's complement format for 21

bits : 111111101100001111100.

Immediate[20,10:1]	Immediate[11,19:12]	rd	Opcode
10000111110	111111101	00011	1101111
	0x87DFD1EF		

2 Question 2

For various pseudo instructions shown below, write their equivalent using a maximum of 2 real instructions.

1. li x5, 0xFFFFFFFFFFFFFFFF

```
addi x5, x0, -1     # Sign extension automatic
```

2. li x5, 132

```
addi x5, x0, 132
```

3. li x5, 2134

```
addi x5, x5, 2047
addi x5, x5, 87
```

4. li x5, 0x000000002345abcd

```
lui x5, 0x2345a      # Load upper 20 bits
addi x5, x5, 0xbcd  # load lower 12 bits
```

3 Question 3

Convert the given instructions in hex to their corresponding assembly code

1. 0x0019F233 : This in binary is 0000000_00001_10011_111_00100_0110011
    - 0110011 is opcode
    - So, the instruction is R format.
    - Funct3 and Funct7 are 111 and 0000000 respectively, which means the operation is or.
    - rs2 is 1, rs1 is 19, rd is 4.
    - Therefore the instruction would be

      and, x4, x19, x1

2. 0x06B4D763 : This in binary is 0000011_01011_01001_101_01110_1100011
    - 1100011 is opcode
    - So, the instruction is B format.
    - The immediate value is 0000001101110, which is 110 in decimal.
    - Funct3 is 101, which means the operation is bge.
    - rs2 is 11, rs1 is 9.
    - Therefore the instruction would be

      bge, x9, x11, 110

3. 0x0169CF93 : This in binary is 000000010110_10011_100_11111_0010011
    - 0010011 is opcode
    - So, the instruction is I format.
    - Funct3 is 100, which means the operation is xori.
    - rs1 is 19, rd is 31.
    - The immediate value is 0000000010110, which is 22 in decimal.
    - Therefore the instruction would be

      xori, x31, x19, 22