# COA LAB RISC PROCESSOR DESIGNING

Name: T.V.S.S.Sripad

Roll Number: 18CS01008

# **Encoding Scheme for Instructions:**

1)

MSB LSB

XX (13 bits)	Source2 (5-bits)	Source1 (5-bits)	Destination- address (5 bits)	Opcode (4 bits)
19-31 bits	14-18 bits	9-13 bits	4-8 bits	0-3 bits

2)

Immediate Value (16 bits)	XX (Dont care) (2 bits)		Destination-a ddress (5 bits)	Opcode ( 4 bits)
16-31 bits	14-15 bits	9-13 bits	4-8 bits	0-3 bits

# Assembler Table:

Assembly Instruction	OP-Code	
Move (MOV)	0000	
Move Immediate (MVI)	0001	
Load	0011	
Store	0101	
Add	0110	

Add Immediate (ADI)	0111
Subtract (SUB)	1000
Subtract Immediate (SUI)	1001
AND	1010
And Immediate (ANI)	1011
OR	1100
OR Immediate (ORI)	1101
HALT	1110

### Addresses of Registers:

The Register file consists of 32 Registers. Hence 5-bits are required to denote address of registers.

Examples for addresses of registers:

- 1) Register R1 has an address of 00000.
- 2) R2's address is 00001 and so on.

#### **Encoding for Given Sample Codes**:

1)

INSTRUCTION	Machine Code	
MOVE R1, 00	0000 0001	
LOAD R2, 0(R1)	0000 0013	
MOVE R3, R2	0000 4020	
ANI R2, R2, 1	0001 021B	
STORE R2, 1(R1)	0001 0015	
MOVE R2, R3	0000 8010	

ANI R2, R2, 2	0002 021B
STORE R2, 2(R1)	0002 0015
MOVE R2, R3	0000 8010
ANI R2, R2, 4	0004 021B
STORE R2, 3(R1)	0003 0015
MOVE R2, R3	0000 8010
ANI R2, R2, 8	0008 021B
STORE R2, 4(R1)	0004 015
MOVE R2, R3	0000 8010
ANI R2, R2, 16	0010 021B
STORE R2, 5(R1)	0005 0015
MOVE R2, R3	0000 8010
ANI R2, R2, 32	0020 021B
STORE R2, 6(R1)	0006 0015
MOVE R2, R3	0000 8010
ANI R2, R2, 64	0040 021B
STORE R2, 7(R1)	0007 0015
MOVE R2, R3	0000 8010
ANI R2, R2, 128	0080 021B
STORE R2, 8(R1)	0008 0015
HLT	0000 000E
-	

Output of Code: The code gives 40, 80 in hexadecimal which are 64, 128 in decimal

Logisim: Hex Editor	
ulate Window Help	
0000 00000c0 0000000 0000000 0000000 000000	0 00000000 00000000 00000000 00000000
0.010 00000000 00000000 00000000 00000000	
$0020$ 00000001 00000013 00004020 0001021b 00010015 00008010 0002021b 00020015 00008010 0004021b $\underline{00030015}$ 0000801	
0030 00050015 00008010 0020021b 00060015 00008010 0040021b 00070015 00008010 0080021b 00080015 $00000000$ 000000000000000000000000000	0 00000000 00000000 00000000 00000000
0040 00000000 00000000 00000000 00000000	
$\it 0050$ 00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000

#### Note:

The above program was loaded at location 0020 and the data is located at 0000

2)

<b>~</b> )	
Instruction	Machine Code
MOVE R1, 00	0000 0001
LOAD R2, 0(R1)	0000 0013
LOAD R3, 2(R1)	0002 0023
ADD R4, R2, R3	0000 8236
SUI R4, R4, 1	0001 0639
STORE R4, 1(R1)	0001 0035
ADI R1, R1, 3	0003 0007
LOAD R2, 0(R1)	0000 0013
LOAD R3, 2(R1)	0002 0023
ADD R4, R2, R3	0000 8236
SUI R4, R4, 1	0001 0639
STORE R4, 1(R1)	0001 0035
MOVE R1, 00	0000 0001
LOAD R2, 1(R1)	0001 0013
LOAD R3, 4(R1)	0004 0023
OR R4, R2, R3	0000 823c
STORE R4, 6(R1)	0006 0035
HLT	0000 000E

Input:- 7 numbers (1,2,3,4,5,6,7) are given from Location 0000 - 0007 Output :- 1,3(1+3-1), 3, 4, 9(4+6-1), 6, B (3 or 9)

Note:- The program is loaded from 0080 location onwards. The operations done are: e1=e0+e2-1

e4=e3+e5-1 e6=e1 or e4

# Logisim Output:

Logisim: Hex Editor
ulate Window Help
0000 00000001 00000003 00000003 00000004 00000009 00000006 00000000 <u>0</u> 00000000 00000000 00000000 00000000
0010 0000000 0000000 0000000 0000000 000000
$\it 00220\ 00000000\ 00000000\ 00000000\ 0000000$
0030 00000000 00000000 00000000 00000000
0040 0000000 0000000 0000000 0000000 000000
0050 0000000 0000000 0000000 0000000 000000
0070 0000000 0000000 0000000 0000000 000000
0080 00000001 00000013 00020023 00008236 00010639 00010035 00030007 00000013 00020023 00008236 00010639 00010035 00000001 00010013 00040023 0000823c
$0090\ 00060035\ 0000000e\ 00000000\ 00000000\ 00000000$
20a0 0000000 0000000 0000000 0000000 000000