CSE 4305: Computer Organization & Architecture
Assignment

Namisa Najah Raisa 210042112

December 27th 2023

1 The provided code

```
int i = i + 10;
int j = j + 12; //ID:210042112
int BASE = BASE + 4;
int val = val + 30;
arr[3] = j
if(j < val) {
i = i + 15;
j = j + 20;
arr[4] = i + j;
} else {
i = i + 5;
j = j + 25;
arr[4] = j - i;
}
```

2 Converted Assembly Code

```
ADDI x1,x1,10//x1=i
ADDI x2,x2,12//x2=j
ADDI x3,x3,4//x3=BASE
ADDI x4,x4,30//x4=val
SW x2,24(x3)//the base of the array is x3,BASE
BGE x2,x4,ELSE//if j>= val
    ADDI x1,x1,15
    ADDI x2,x2,20
    ADD x5,x1,x2//temporary x5=i+j
```

```
10      SW x5,32(x3)
11  ELSE:
12      ADDI x1,x1,5
13      ADDI x2,x2,25
14      SUB x5,x2,x1//temporary x5=j-i
15      SW x5,32(x3)
```

Input your RISC-V code here:

```
1   ADDI x1,x1,10
2   ADDI x2,x2,12
3   ADDI x3,x3,4
4   ADDI x4,x4,30
5   SW x2,24(x3)
6   BGE x2,x4,ELSE
7       ADDI x1,x1,15
8       ADDI x2,x2,20
9       ADD x5,x1,x2
10      SW x5,32(x3)
11  ELSE:
12      ADDI x1,x1,5
13      ADDI x2,x2,25
14      SUB x5,x2,x1
15      SW x5,32(x3)
```

Reset | Stop | CPU: 32 Hz

```
[line 10]: SW x5,32(x3)
[line 12]: ADDI x1,x1,5
[line 13]: ADDI x2,x2,25
[line 14]: SUB x5,x2,x1
[line 15]: SW x5,32(x3)
No more instructions to run! Press Reset to reload the code!
```

3 Registers with the Updated Values

Init Value	Register	Decimal	Hex	Binary
0	x0 (zero)	0	0x00000000	0b00000000000000000000000000000000
0	x1 (ra)	30	0x0000001e	0b00000000000000000000000000011110
0	x2 (sp)	57	0x00000039	0b00000000000000000000000000111001
0	x3 (gp)	4	0x00000004	0b00000000000000000000000000000100
0	x4 (tp)	30	0x0000001e	0b00000000000000000000000000011110
0	x5 (t0)	27	0x0000001b	0b00000000000000000000000000011011
0	x6 (t1)	0	0x00000000	0b00000000000000000000000000000000
0	x7 (t2)	0	0x00000000	0b00000000000000000000000000000000
0	x8 (s0/fp)	0	0x00000000	0b00000000000000000000000000000000
0	x9 (s1)	0	0x00000000	0b00000000000000000000000000000000
0	x10 (a0)	0	0x00000000	0b00000000000000000000000000000000
0	x11 (a1)	0	0x00000000	0b00000000000000000000000000000000
0	x12 (a2)	0	0x00000000	0b00000000000000000000000000000000
0	x13 (a3)	0	0x00000000	0b00000000000000000000000000000000
0	x14 (a4)	0	0x00000000	0b00000000000000000000000000000000
0	x15 (a5)	0	0x00000000	0b00000000000000000000000000000000
0	x16 (a6)	0	0x00000000	0b00000000000000000000000000000000
0	x17 (a7)	0	0x00000000	0b00000000000000000000000000000000
0	x18 (s2)	0	0x00000000	0b00000000000000000000000000000000
0	x19 (s3)	0	0x00000000	0b00000000000000000000000000000000
0	x20 (s4)	0	0x00000000	0b00000000000000000000000000000000
0	x21 (s5)	0	0x00000000	0b00000000000000000000000000000000
0	x22 (s6)	0	0x00000000	0b00000000000000000000000000000000
0	x23 (s7)	0	0x00000000	0b00000000000000000000000000000000
0	x24 (s8)	0	0x00000000	0b00000000000000000000000000000000

0	x25 (s9)	0	0x00000000	0b00000000000000000000000000000000
0	x26 (s10)	0	0x00000000	0b00000000000000000000000000000000
0	x27 (s11)	0	0x00000000	0b00000000000000000000000000000000
0	x28 (t3)	0	0x00000000	0b00000000000000000000000000000000
0	x29 (t4)	0	0x00000000	0b00000000000000000000000000000000
0	x30 (t5)	0	0x00000000	0b00000000000000000000000000000000
0	x31 (t6)	0	0x00000000	0b00000000000000000000000000000000

Download Registers!

4 Memory with the Updated values

Memory Address 0x00000000 Go Download!

Memory Address	Decimal	Hex	Binary
0x00000000	0	0x00000000	0b00000000000000000000000000000000
0x00000004	0	0x00000000	0b00000000000000000000000000000000
0x00000008	0	0x00000000	0b00000000000000000000000000000000
0x0000000c	0	0x00000000	0b00000000000000000000000000000000
0x00000010	0	0x00000000	0b00000000000000000000000000000000
0x00000014	0	0x00000000	0b00000000000000000000000000000000
0x00000018	0	0x00000000	0b00000000000000000000000000000000
0x0000001c	12	0x0000000c	0b00000000000000000000000000001100
0x00000020	0	0x00000000	0b00000000000000000000000000000000
0x00000024	27	0x0000001b	0b00000000000000000000000000011011

[zooming needed]