Group 17

Kaustubh Hiware 14CS30011
Rameshwar Bhaskaran 14CS30027

GENERAL INSTRUCTION SET

16 bit instruction varies depending

```
    _ | _ _ _ | _ _ _ | _ _ _ | _ _ _ | _ _ _   ( 16 bit)
type  sub-type A.M     dst     op1      op2
```

A.M = Addressing Mode

First bit decides if it is an ALU operation or not

The remaining 3 bits encode the general instruction like ld,st, add etc

The next 3 bits encode the addressing mode

The next 3 bits encode the dst register

The next 6 bits contain the 2 registers, presence of which is decided based on addressing modes

In some cases, 16 bit displacement is also required

NON-ALU OPERATIONS

1. li r5, #100

```
 1 | 00x | 000 | --- | xxx | xxx
                    register
```

2. lr r5, r7

```
 1 | 00x | 000 | --- | xxx | xxx
                    register
```

3. lx r5, 10(r1,r7)

```
 1 | 00x | 001 | --- | --- | ---
```

reg5 reg1 reg7

4. ldn r5, @10(r1,r7)

1 | 00x | 010 | --- | --- | ---
reg5 reg1 reg7

5. stx -5(r2), r3

1 | 01x | 000 | --- | --- | xxx
reg2 reg3

6. stn @-5(r2),r3

1 | 01x | 001 | --- | --- | xxx
reg2 reg3

7. stx @-5(r2),r3

1 | 01x | 010 | --- | --- | xxx
reg2 reg3

8. j addr

1 | 10x | 000 | 0xx | xxx | xxx

16 bit address as displacement

9. jz addr

1 | 10x | 000 | 1xx | xxx | xxx

16 bit address as displacement

10. jnz addr

1 | 10x | 001 | 0xx | xxx | xxx

16 bit address as displacement

11. jc addr

1 | 10x | 001 | 1xx | xxx | xxx

16 bit address as displacement

12. jz addr

1 | 10x | 010 | 0xx | xxx | xxx

16 bit address as displacement

13. jc addr

1 | 10x | 010 | 1xx | xxx | xxx

16 bit address as displacement

14. jnc addr

1 | 10x | 011 | 0xx | xxx | xxx

16 bit address as displacement

15. jv addr

1 | 10x | 011 | 1xx | xxx | xxx

16 bit address as displacement

16. jnv addr

1 | 10x | 100 | 0xx | xxx | xxx

16 bit address as displacement

17. jm addr

1 | 10x | 100 | 1xx | xxx | xxx

16 bit address as displacement

18. jnm addr

1 | 10x | 101 | 0xx | xxx | xxx

16 bit address as displacement

SUBROUTINE CALLS

19. jal r5, sub

1 | 110 | xxx | xxx | xxx | xxx

16 bit address as displacement

20. jr r5

1 | 111 | xxx | xxx | xxx | xxx

16 bit address as displacement

ALU OPERATIONS

ADD:

21. addi r1, #43

0 | 000 | 000 | 001 | xxx | xxx
reg1

16 bit displacement

22. addr r5,r7

0 | 000 | 001 | 101 | 111 | xxx | xxx
reg5 reg7

23. addx r5, 10(r1,r7)

0 | 000 | 010 | 001 | 111 | xxx | xxx
16 bit displacement

24. addn r5, @10(r1,r7)

0 | 000 | 011 | 001 | 111 | xxx | xxx
16 bit displacement

SUB:

25. subi r1, #43

0 | 001 | 000 | 001 | xxx | xxx
reg1

16 bit displacement

26. subr r5,r7

0 | 001 | 001 | 101 | 111 | xxx | xxx
reg5 reg7

27. subx r5, 10(r1,r7)

0 | 001 | 010 | 001 | 111 | xxx | xxx
16 bit displacement

28. subn r5, @10(r1,r7)

0 | 001 | 011 | 001 | 111 | xxx | xxx
16 bit displacement

AND:

29. andi r1, #43

0 | 000 | 000 | 001 | xxx | xxx
reg1

16 bit displacement

30. andr r5,r7

0 | 010 | 001 | 101 | 111 | xxx | xxx
reg5 reg7

31. andx r5, 10(r1,r7)

0 | 010 | 010 | 001 | 111 | xxx | xxx
16 bit displacement

32. andn r5, @10(r1,r7)

0 | 010 | 011 | 001 | 111 | xxx | xxx
16 bit displacement

OR:

33. ori r1, #43

0 | 011 | 000 | 001 | xxx | xxx
reg1

16 bit displacement

34. orr r5,r7

0 | 011 | 001 | 101 | 111 | xxx | xxx
reg5 reg7

35. orx r5, 10(r1,r7)

0 | 011 | 010 | 001 | 111 | xxx | xxx
16 bit displacement

36. orn r5, @10(r1,r7)

0 | 011 | 011 | 001 | 111 | xxx | xxx
16 bit displacement

MNS:

37. mnsi r1, #43

0 | 100 | 000 | 001 | xxx | xxx
reg1
16 bit displacement

38. mnsr r5,r7

0 | 100 | 001 | 101 | 111 | xxx | xxx
reg5 reg7

39. mnsx r5, 10(r1,r7)

0 | 100 | 010 | 001 | 111 | xxx | xxx
16 bit displacement

40. mnsn r5, @10(r1,r7)

0 | 100 | 011 | 001 | 111 | xxx | xxx
16 bit displacement

CMP:

41. cmpi r1, #43

0 | 101 | 000 | 001 | xxx | xxx

reg1
16 bit displacement

42. cmpr r5,r7

0 | 101 | 001 | 101 | 111 | xxx | xxx
reg5 reg7

43. cmpx r5, 10(r1,r7)

0 | 101 | 010 | 001 | 111 | xxx | xxx
16 bit displacement

44. cmpn r5, @10(r1,r7)

0 | 101 | 011 | 001 | 111 | xxx | xxx
16 bit displacement