

Group 17

Kaustubh Hiware 14CS30011

Rameshwar Bhaskaran 14CS30027

GENERAL INSTRUCTION SET

32 bit instruction varies depending

 _ | _ _ _ | _ _ _ | _ _ _ | _ _ _ | _ _ _ | (16 bit displacement extra needed for immediate addressing etc.)

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

Next 16 bits are for displacement is also required

JUSTIFICATIONS:

1. We have made all the instructions 32 -bit to prevent another cycle to read in the displacement
2. We have removed base indexed addressing to make all instructions work in the single cycle.

NON-ALU OPERATIONS

1. li r5, #100

1 | 00x | 000 | --- | xxx | xxx | y | yyy | yyy | yyy | yyy | yyy
register

(yyy... - 16bit operand)

2. lr r5, r7

1 | 00x | 000 | --- | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000
register

5. stx -5(r2), r3

1 | 01x | 000 | --- | --- | xxx | 0 | 000 | 000 | 000 | 000 | 000
reg2 reg3

6. stn @-5(r2),r3

1 | 01x | 001 | --- | --- | xxx | 0 | 000 | 000 | 000 | 000 | 000
reg2 reg3

7. stx @-5(r2),r3

1 | 01x | 010 | --- | --- | xxx | 0 | 000 | 000 | 000 | 000 | 000
reg2 reg3

8. j addr

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

16 bit address as displacement

9. jz addr

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

16 bit address as displacement

10. jnz addr

1 | 10x | 001 | 0xx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

11. jc addr

1 | 10x | 001 | 1xx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

12. jz addr

1 | 10x | 010 | 0xx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

13. jc addr

1 | 10x | 010 | 1xx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

14. jnc addr

1 | 10x | 011 | 0xx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

15. jv addr

1 | 10x | 011 | 1xx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

16. jnv addr

1 | 10x | 100 | 0xx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

17. jm addr

1 | 10x | 100 | 1xx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

18. jnm addr

1 | 10x | 101 | 0xx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

SUBROUTINE CALLS

19. jal r5, sub

1 | 110 | xxx | xxx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

20. jr r5

1 | 111 | xxx | xxx | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

16 bit address as displacement

ALU OPERATIONS

ADD:

21. addi r1, #43

0 | 000 | 000 | 001 | xxx | xxx | y | yyy | yyy | yyy | yyy | yyy

reg1

(yyy... - 16bit operand)

16 bit displacement

22. addr r5,r7

0 | 000 | 001 | 101 | 111 | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

reg5 reg7

SUB:

25. subi r1, #43

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

reg1

16 bit displacement

26. subr r5,r7

0 | 001 | 001 | 101 | 111 | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

reg5 reg7

AND:

29. andi r1, #43

0 | 000 | 000 | 001 | xxx | xxx | y | yyy | yyy | yyy | yyy | yyy

reg1

(yyy... - 16bit operand)

16 bit displacement

30. andr r5,r7

0 | 010 | 001 | 101 | 111 | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

reg5 reg7

OR:

33. ori r1, #43

0 | 011 | 000 | 001 | xxx | xxx | y | yyy | yyy | yyy | yyy | yyy

reg1

(yyy... - 16bit operand)

34. orr r5,r7

0 | 011 | 001 | 101 | 111 | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

reg5 reg7

MNS:

37. mnsi r1, #43

0 | 100 | 000 | 001 | xxx | xxx | y | yyy | yyy | yyy | yyy | yyy

reg1

(yyy... - 16bit operand)

38. mnsr r5,r7

0 | 100 | 001 | 101 | 111 | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

reg5 reg7

CMP:

41. cmpi r1, #43

0 | 101 | 000 | 001 | xxx | xxx | y | yyy | yyy | yyy | yyy | yyy

reg1

(yyy... - 16bit operand)

42. cmpr r5,r7

0 | 101 | 001 | 101 | 111 | xxx | xxx | 0 | 000 | 000 | 000 | 000 | 000

reg5 reg7