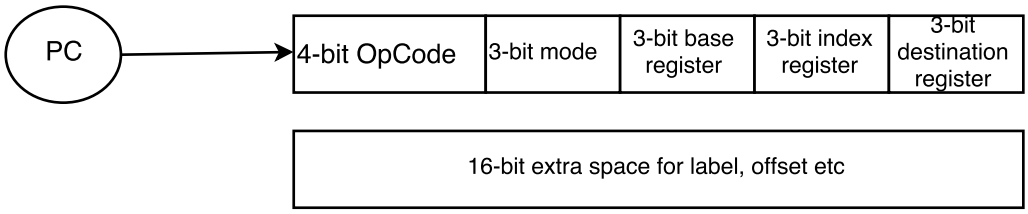
16-bit instruction



Sr. no	Op-code(4 bit)	Operation Implemented		Operand2(src)		Register(Operand1)	Extra(16 bit)	COMMENTS
			Mode(3 bit)	Rb(3 bit)	Rx(3 bit)	Rdst(3 bit)		
1	1 0 0 0	ADDI	0 0 0	x x x	x x x	Address(dest)	d	Rdst <= Rdst + d (add immediate)
2	1 0 0 0	ADDR	0 0 1	(src reg)	x x x	Address(dest)		Rdst <= Rdst + Rb (register addressing)
3	1 0 0 0	ADDX	0 1 0	(base reg)	(index reg)	Address(dest)	Offset	Rdst <= Rdst +M[rb+rx+Offset] (Base index addressing)
4	1 0 0 0	ADDN	0 1 1	(Address reg)	(index reg)	Address(dest)	Offset	Rdst <= Rdst + M[M[rb+rx+Offset]] (Indirect memory addressing)
5	1 0 0 1	SUBI	0 0 0	x x x	x x x	Address(dest)	d	Rdst <= Rdst - d (add immediate)
6	1 0 0 1	SUBR	0 0 1	(src reg)	x x x	Address(dest)		Rdst <= Rdst - Rb (register addressing)
7	1 0 0 1	SUBX	0 1 0	(base reg)	(index reg)	Address(dest)	Offset	Rdst <= Rdst - M[rb+rx+Offset] (Base index addressing)
8	1 0 0 1	SUBN	1 0 0	(Address reg)	(index reg)	Address(dest)	Offset	Rdst <= Rdst + M[M[rb+rx+Offset]] (Indirect memory addressing)
9	1 0 1 0	ANDI	0 0 0	x x x	x x x	Address(dest)	d	Rdst <= Rdst & d (add immediate)
10	1 0 1 0	ANDR	0 0 1	(src reg)	x x x	Address(dest)		Rdst <= Rdst & Rb (register addressing)
11	1 0 1 0	ANDX	0 1 0	(base reg)	(index reg)	Address(dest)	Offset	Rdst <= Rdst & M[rb+rx+Offset] (Base index addressing)
12	1 0 1 0	ANDN	0 1 1	(Address reg)	(index reg)	Address(dest)	Offset	Rdst <= Rdst & M[M[rb+rx+Offset]] (Indirect memory addressing)
13	1 0 1 1	ORI	0 0 0	x x x	x x x	Address(dest)	d	Rdst <= Rdst | d (add immediate)
14	1 0 1 1	ORR	0 0 1	(src reg)	x x x	Address(dest)		Rdst <= Rdst | Rb (register addressing)
15	1 0 1 1	ORX	0 1 0	(base reg)	(index reg)	Address(dest)	Offset	Rdst <= Rdst | M[rb+rx+Offset] (Base index addressing)
16	1 0 1 1	ORN	0 1 1	(Address reg)	(indexreg)	Address(dest)	Offset	Rdst <= Rdst | M[M[rb+rx+Offset]] (Indirect memory addressing)
17	1 1 0 0	MNSI	0 0 0	x x x	x x x	Address(src1)	d	Rdst - d (add immediate)
18	1 1 0 0	MNSR	0 0 1	(src reg)	x x x	Address(src1)		Rdst - Rb (register addressing)
19	1 1 0 0	MNSX	0 1 0	(base reg)	(index reg)	Address(src1)	Offset	Rdst - M[rb+rx+Offset] (Base index addressing)
20	1 1 0 0	MNSN	0 1 1	(Address reg)	(index reg)	Address(src1)	Offset	Rdst – M[M[rb+rx+offset]] (Indirect memory addressing)
21	1 1 0 1	CMP	x x x	x x x	x x x	Address(dest & src)		Rdst <= !Rdst (2'c complement of Rdst)
22	0 0 1 0	JALR	1 x x	x x x	x x x	Address(dest)	Offset	Rdst <= PC, PC <= PC + Offset
23	0 0 1 1	JR	1 x x	x x x	x x x	Address(dest)		PC <= Rdst
24	0 0 0 0	Store	X X X	Base register	Index register	Address(dest)	Offset	Rdst = M[Rb+Rx-Offset] (Base addressed)
25	0 0 0 1	Store	X X X	Base register	Index register	Address(dest)	Offset	Rdst = M[M[Rb+Rx-Offset]] (Indirect)
26	0 1 0 0	Branch	1 0 0	X X X	X X X	Address(dest)	Label	j (Jump unconditionally)
27	0 1 0 0	Branch	1 0 1	X X X	X X X	Address(dest)	Label	jz (Jump on zero)
28	0 1 0 0	Branch	1 1 0	X X X	X X X	Address(dest)	Label	jnz (Jump on not zero)
29	0 1 0 1	Branch	1 0 0	X X X	X X X	Address(dest)	Label	jc (Jump on carry)
30	0 1 0 1	Branch	1 0 1	X X X	X X X	Address(dest)	Label	jnc (Jump on not carry)
31	0 1 0 1	Branch	1 1 0	X X X	X X X	Address(dest)	Label	jv (Jump on overflow)
32	0 1 1 0	Branch	1 0 0	X X X	X X X	Address(dest)	Label	jnv (Jump on not overflow)
33	0 1 1 0	Branch	1 0 1	X X X	X X X	Address(dest)	Label	jm (Jump on minus)
34	0 1 1 0	Branch	1 1 0	X X X	X X X	Address(dest)	Label	jnm (jump on not minus)
35	0 1 1 1	Load	0 0 0	X X X	X X X	Address(dest)	mmediate valu	li (Load immediate)
36	0 1 1 1	Load	0 0 1	Base register	X X X	Address(dest)		lr (Load register)
37	0 1 1 1	Load	0 1 0	Base register	X X X	Address(dest)	Offset	lm (Load base addressed)
38	0 1 1 1	Load	0 1 1	Base register	Index register	Address(dest)	Offset	lx (Load base index addressed)
39	0 1 1 1	Load	1 0 0	Base register	Index register	Address(dest)	Offset	ldn (Load indirect)