

Ld op3 000000

Format (3):

11	rd	op3	rs1	i=0	asi	rs2
31	29	24	18	13	12	4 0
11	rd	op3	rs1	i=1	simm13	
31	29	24	18	13	12	0

Exemplo:

Ld [%i1], %i0

11	11000	op3 = 0	11001	i=1	simm13 = 0
31	29	24	18	13	12 0
%i0 = %r24			%i1 = %r25		

Add op3 000000

Format (3):

10	rd	op3	rs1	i=0	unused(zero)	rs2
31	29	24	18	13	12	4 0
10	rd	op3	rs1	i=1	simm13	
31	29	24	18	13	12	0

Exemplo:

Add %g0, 0, %i7

10	11111	op3=0	rs1=0	i=1	simm13=0
31	29	24	18	13	12 0
%i7 = %r31			%g0 = %r0		

AND op3 010001

Format (3):

10	rd	op3	rs1	i=0	unused(zero)	rs2
31	29	24	18	13	12	4 0
10	rd	op3	rs1	i=1	simm13	
31	29	24	18	13	12	0

Exemplo:

And %i1, 1, %i2

10	11010	010001	11001	i=1	0 0000 0000 0001
31	29	24	18	13	12 0
%i2 = %r26			%i1 = %r25		

XOR op3 000111

Format (3):

10	rd	op3	rs1	i=0	unused(zero)	rs2
31	29	24	18	13	12	4 0
10	rd	op3	rs1	i=1	simml3	
31	29	24	18	13	12	0

Exemplo:

Xor %L0, %L2, %L0

10	01000	000111	01000	i=0	unused(zero)	01010
31	29	24	18	13	12	4 0
%L0 = %r16			%L0 = %r16			%L2 = %r18

SRL op3 100110

Format (3):

10	rd	op3	rs1	i=0	unused(zero)	rs2
31	29	24	18	13	12	4 0
10	rd	op3	rs1	i=1	unused(zero)	shcnt
31	29	24	18	13	12	4 0

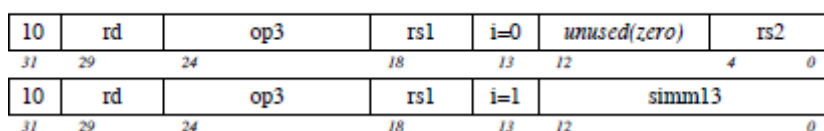
Exemplo:

Srl %L1, 1, %L1

10	10001	100110	10001	i=1	unused(zero)	00001
31	29	24	18	13	12	4 0
%L1 = %r17			%L1 = %r17			desloca 1

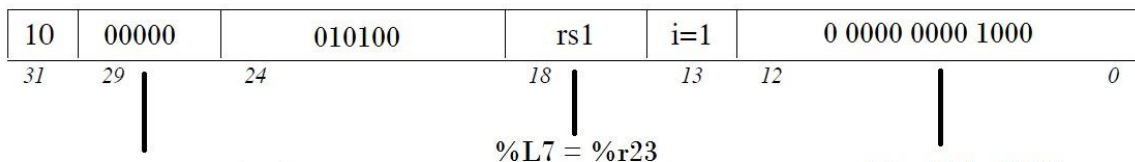
Cmp = subcc op3 010100

Format (3):



Exemplo:

Cmp %L7, 8 = subcc %L7, 8, %g0



não vai usar o resultado
%g0 = %r0

%L7 = %r23

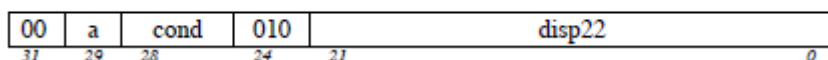
subtrai 8 do %L7

Observação Importante: subcc altera os bits icc. Então, se %L7 = 4, temos:

- Z = 0
- N = 1
- V = 0 não houve overflow? O número pôde ser representado em complemento de dois. Olhar pág 29 do manual v8.
- C = 0 não houve carry out (soma) do bit 31 nem borrow in (subtração) do bit 0.

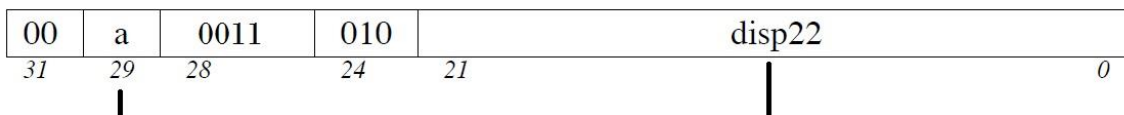
BL cond=0011

Format (2):



Exemplo:

Bl [address]



indiferente, não
implementado

PC = PC + 4 x sigext(disp22)

Observação importante: o bl usa os bits icc para decidir se fará o branch. Branch = N xor V (Negativo ou-exclusivo overflow). Se branch for verdadeiro, desvia.

LEGENDA:

Rd – reg destino. 5bits com address dele

Rs1 – reg operando. 5bitd com address dele

Rs2 - reg operando. 5bitd com address dele

Simm13 – imediato de 13bits; pode ser estendido para até 32 bits

i – i=0, operando é rs2; i=1, operando é simm13 ou shcnt

asi – para load/save alternae instrucion

shcnt – valor a ser deslocado

disp22 – 22bits para word-aligned, sign-extended, PC-relative displacements em uma branch

Registradores

Windowed Register Address	r Register Address
in[0] – in[7]	r[24] – r[31]
local[0] – local[7]	r[16] – r[23]
out[0] – out[7]	r[8] – r[15]
global[0] – global[7]	r[0] – r[7]