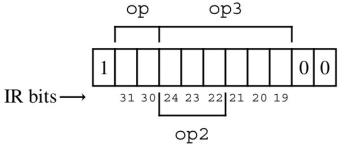


op	Format	op2	Inst.	op3 (op=10)	op3 (op=11)	cond	branch
00	SETHI/Branch	010	branch sethi	010000 addcc 010001 andcc	000000 ld 000100 st	0001	be bcs
10	Arithmetic	100	seciii	010001 andcc	000100 SC	0110	bneg
11	Memory			010110 orncc		1000	bvs ba
				111000 jmpl			100.000

n z v c

PSR

$F_3 F_2 F_1 F_0$	Operation	Changes Condition Codes
0 0 0 0	ANDCC (A, B)	yes
0 0 0 1	ORCC (A, B)	yes
0 0 1 0	NORCC (A, B)	yes
0 0 1 1	ADDCC (A, B)	yes
0 1 0 0	SRL (A, B)	no
0 1 0 1	AND (A, B)	no
0 1 1 0	OR (A, B)	no
0 1 1 1	NOR (A, B)	no
1 0 0 0	ADD (A, B)	no
1 0 0 1	LSHIFT2 (A)	no
1 0 1 0	LSHIFT10 (A)	no
1 0 1 1	SIMM13 (A)	no
1 1 0 0	SEXT13 (A)	no
1 1 0 1	INC (A)	no
1 1 1 0	INCPC (A)	no
1 1 1 1	RSHIFT5 (A)	no
	op	on3



```
0: R[ir] \leftarrow AND(R[pc],R[pc]); READ;
                                                   1762: R[temp0] ← SEXT13(R[ir]);
                                                   1763: R[pc] \leftarrow ADD(R[rs1], R[temp0]);
   1: DECODE;
                                                         GOTO 0;
      / sethi
                                                          / 1d
1152: R[rd] ← LSHIFT10(ir); GOTO 2047;
                                                   1792: R[temp0] ← ADD(R[rs1],R[rs2]);
      / call
                                                         IF R[IR[13]] THEN GOTO 1794;
1280: R[15] \leftarrow AND(R[pc],R[pc]);
                                                   1793: R[rd] \leftarrow AND(R[temp0], R[temp0]);
1281: R[temp0] \leftarrow ADD(R[ir], R[ir]);
                                                         READ; GOTO 2047;
1282: R[temp0] \leftarrow ADD(R[temp0], R[temp0]);
                                                   1794: R[temp0] ← SEXT13(R[ir]);
1283: R[pc] \leftarrow ADD(R[pc], R[temp0]);
                                                   1795: R[temp0] \leftarrow ADD(R[rs1], R[temp0]);
      GOTO 0;
                                                         GOTO 1793;
       / addcc
                                                          / st
1600: IF R[IR[13]] THEN GOTO 1602;
                                                   1808: R[temp0] \leftarrow ADD(R[rs1], R[rs2]);
1601: R[rd] \leftarrow ADDCC(R[rs1], R[rs2]);
                                                         IF R[IR[13]] THEN GOTO 1810;
      GOTO 2047;
                                                   1809: R[ir] ← RSHIFT5(R[ir]); GOTO 40;
1602: R[temp0] \leftarrow SEXT13(R[ir]);
                                                     40: R[ir] ← RSHIFT5(R[ir]);
1603: R[rd] \leftarrow ADDCC(R[rs1], R[temp0]);
                                                     41: R[ir] ← RSHIFT5(R[ir]);
      GOTO 2047;
                                                     42: R[ir] \leftarrow RSHIFT5(R[ir]);
      / andcc
                                                     43: R[ir] ← RSHIFT5(R[ir]);
1604: IF R[IR[13]] THEN GOTO 1606;
                                                     44: R[0] ← AND(R[temp0], R[rs2]);
1605: R[rd] \leftarrow ANDCC(R[rs1], R[rs2]);
                                                         WRITE; GOTO 2047;
      GOTO 2047;
                                                   1810: R[temp0] ← SEXT13(R[ir]);
1606: R[temp0] \leftarrow SIMM13(R[ir]);
                                                   1811: R[temp0] ← ADD(R[rs1], R[temp0]);
1607: R[rd] \leftarrow ANDCC(R[rs1], R[temp0]);
                                                         GOTO 1809;
      GOTO 2047;
                                                         / Branch instructions: ba, be, bcs,
      / orcc
                                                                                   bus, breg
                                                   1088: GOTO 2;
1608: IF R[IR[13]] THEN GOTO 1610;
                                                      2: R[temp0] ← LSHIFT10(R[ir]);
1609: R[rd] \leftarrow ORCC(R[rs1], R[rs2]);
                                                      3: R[temp0] ← RSHIFT5(R[temp0]);
      GOTO 2047;
                                                      4: R[temp0] ← RSHIFT5(R[temp0]);
1610: R[temp0] \leftarrow SIMM13(R[ir]);
                                                      5: R[ir] ← RSHIFT5(R[ir]);
1611: R[rd] \leftarrow ORCC(R[rs1], R[temp0]);
                                                      6: R[ir] ← RSHIFT5(R[ir]);
      GOTO 2047;
                                                      7: R[ir] ← RSHIFT5(R[ir]);
       / orncc
                                                      8: IF R[IR[13]] THEN GOTO 12;
1624: IF R[IR[13]] THEN GOTO 1626;
                                                         R[ir] \leftarrow ADD(R[ir],R[ir]);
1625: R[rd] \leftarrow NORCC(R[rs1], R[rs2]);
                                                      9: IF R[IR[13]] THEN GOTO 13;
      GOTO 2047;
                                                         R[ir] \leftarrow ADD(R[ir],R[ir]);
1626: R[temp0] \leftarrow SIMM13(R[ir]);
                                                     10: IF Z THEN GOTO 12;
1627: R[rd] \leftarrow NORCC(R[rs1], R[temp0]);
                                                         R[ir] \leftarrow ADD(R[ir],R[ir]);
      GOTO 2047;
                                                     11: GOTO 2047;
       / srl
                                                     12: R[pc] ← ADD(R[pc],R[temp0]);
1688: IF R[IR[13]] THEN GOTO 1690;
                                                         GOTO 0;
1689: R[rd] \leftarrow SRL(R[rs1],R[rs2]);
                                                     13: IF R[IR[13]] THEN GOTO 16;
      GOTO 2047;
                                                         R[ir] \leftarrow ADD(R[ir],R[ir]);
1690: R[temp0] \leftarrow SIMM13(R[ir]);
                                                     14: IF C THEN GOTO 12;
1691: R[rd] \leftarrow SRL(R[rs1], R[temp0]);
                                                     15: GOTO 2047;
      GOTO 2047;
                                                     16: IF R[IR[13]] THEN GOTO 19;
      / jmpl
                                                     17: IF N THEN GOTO 12;
1760: IF R[IR[13]] THEN GOTO 1762;
                                                     18: GOTO 2047;
1761: R[pc] \leftarrow ADD(R[rs1], R[rs2]);
                                                     19: IF V THEN GOTO 12;
      GOTO 0;
                                                     20: GOTO 2047;
                                                   2047: R[pc] \leftarrow INCPC(R[pc]); GOTO 0;
```

$C_2$ $C_1$ $C_0$	Operation
0 0 0	Use NEXT ADDR
0 0 1	Use JUMP ADDR if $n = 1$
0 1 0	Use JUMP ADDR if $z = 1$
0 1 1	Use JUMP ADDR if $v = 1$
1 0 0	Use JUMP ADDR if $c = 1$
1 0 1	Use JUMP ADDR if IR [13] = 1
1 1 0	Use JUMP ADDR
1 1 1	DECODE

nemonic	Meaning
ld	Load a register from memory
st	Store a register into memory
sethi	Load the 22 most significant bits of a register
andcc	Bitwise logical AND
orcc	Bitwise logical OR
ornec	Bitwise logical NOR
srl	Shift right (logical)
addcc	Add
call	Call subroutine
jmpl	Jump and link (return from subroutine call)
be	Branch if equal
bneg	Branch if negative
bcs	Branch on carry
bvs	Branch on overflow
ba	Branch always

