			MUXAD	COND	ALU	DESP	RIDMM	RDMM	REWARD	WR	PERRC	C	A	B	DIR
0	rdm := cp; rd;	ciclo principal	x	00	xx	xx	0	1	1	0	0	0000	xxxx	0000	xxxx xxxx
1	cp := cp + 1; rd;	incrementa cp	0	00	00	00	0	0	1	0	1	0000	0110	0000	xxxx xxxx
2	ri := rim; **if** n **then goto** 28;	salva y decodifica rim	1	01	10	00	0	0	0	0	1	0011	xxxx	xxxx	0001 1100
3	rit := lshift(ri + ri); **if** n **then goto** 19;		0	01	00	10	0	0	0	0	1	0100	0011	0011	0001 0011
4	rit := lshift (rit); **if** n **then goto** 11;	000x o 001x?	0	01	10	10	0	0	0	0	1	0100	0100	xxxx	0000 1011
5	alu := rit; **if** n **then goto** 9;	0000 o 0001?	0	01	10	00	0	0	0	0	0	xxxx	0100	xxxx	0000 1001
6	rdm := ri; rd;	0000 = CARD	x	00	xx	xx	0	1	1	0	0	xxxx	xxxx	0011	xxxx xxxx
7	rd;		x	00	xx	xx	0	0	1	0	0	xxxx	xxxx	xxxx	xxxx xxxx
8	ac := rim; **goto** 0;		1	11	10	00	0	0	0	0	1	0001	xxxx	xxxx	0000 0000
9	rdm := ri; rim := ac; wr;	0001 = ALMD													
10	wr; **goto** 0;														
11	alu := rit; **if** n **then goto** 15;	0010 o 0011?													
12	rdm := ri; rd;	0010 = SUMD													
13	rd;														
14	ac := rim + ac; **goto** 0;														
15	rdm := ri; rd;	0011 = RESD													
16	ac := ac + 1; rd:	$x - y = x + 1 + not\ y$													
17	a := inv(rim);														
18	ac := ac + a; **goto** 0;														
19	rit := lshift(rit); **if** n **then goto** 25;	010x o 011x?													
20	alu := rit; **if** n **then goto** 23;	0100 o 0101?													
21	alu := ac; **if** n **then goto** 0;	0100 = ppOS													
22	cp := band (ri, mascd); **goto** 0;	realiza el salto													

23 alu := ac; **if** z **then goto** 22;	0101 = SCERO
24 **goto** 0;	no se produce el salto
25 alu :=rit; **if** n **then goto** 27;	0110 0 0111
26 cp := band (ri, mascd); **goto** 0	0110 = SALTA
27 ac :=band (ri, mascd); **goto** 0;	0111 = CARC
28 rit := lshift (ri + ri); **if** n **then goto** 40;	10xx o 11xx?
29 rit := lshift (rit); **if** n **then goto** 35;	100x o 101x?
30 alu := rit; **if** n **then goto** 33;	1000 o 1001?
31 a := ri + pp;	1000 = CARL
32 rdm := a; rd; **goto** 7;	
33 a := ri + pp;	1001 = ALML
34 rdm := a; rim := ac; wr; **goto** 10	
35 alu := rit; **if** n **then goto** 38	010 0 1011?
36 a := ri + pp;	1010 = SUML
37 rdm := a; rd; **goto** 13;	
38 a := ri + pp;	1011 = RESL
39 rdm := a; rd; **goto** 16;	
40 rit := lshift(rit); **if** n **then goto** 46;	110x o 111x?
41 alu := rit; **if** n **then goto** 44;	1100 p 1101?
42 alu := ac; **if** n **then goto** 22;	1100 = SNEG
43 **goto** 0;	
44 alu := ac; **if** z **then goto** 0;	1101 = SNCERO
45 cp := band (ri, mascd); **goto** 0	
46 rit := lshift (rit); **if** n **then goto** 50;	
47 pp := pp + (-1);	1110 = LLAMA
48 rdm := pp; rim := cp; wr;	

49 cp := band (ri, mascd); wr; **goto** 0

50 rit := lshift(rit); **if** n **then goto** 65;   1111000 = APILAI

51 rit := lshift(rit); **if** n **then goto** 59;

52 alu := rit; **if** n **then goto** 56;

53 rdm := ac; rd;

54 pp := pp + (-1); rd;

55 rdm := pp; wr; **goto** 10;

56 rdm := pp; pp := pp + 1; rd;   1111001 = DPILAI

57 rd;

58 rdm := ac; wr; **goto** 10;

59 alu := rit; **if** n **then goto** 62;

60 pp := pp + (-1);   1111010 = APILA

61 rdm := pp; rim := ac; wr; **goto** 10;

62 rdm := pp; pp := pp + 1; rd;   1111011 = DPILA

63 rd;

64 ac := rim; **goto** 0;

65 rit := lshift(rit); **if** n **then goto** 73;

66 alu := rit; **if** n **then goto** 70;

67 rdm := pp; pp := pp + 1; rd;   1111100 = RETOR

68 rd;

69 cp := rim; **goto** 0;

70 a := ac;   1111101 = INTERC

71 ac := pp;

72 pp := a; **goto** 0;

73 alu := rit; **if** n **then goto** 76;

74 a := band (ri, mascp);   1111110 = INCPP

75 pp := pp + a; **goto** 0;

76 a := band (ri, mascp);   1111111 = DECPP

77 a := inv (a);

78 a := a + 1; **goto** 76;

79 pp := pp + a; **goto** 0;