

*Curitiba, Brasil*

The diagram illustrates a 4-bit ALU circuit. It features two 4-bit inputs, A and B, each represented by a register with green circles. A 2-bit operation code (OP) is also provided. The circuit contains several functional blocks: an adder for 'c in' and 'c out', a subtractor for 'b in' and 'b out', an AND gate, an OR gate, a left shifter, and a right shifter. The outputs of these blocks are connected to a multiplexer (MUX) which produces the final 4-bit output D. The OP register is connected to the MUX to select the appropriate operation.

### III. Código de teste

Nº	Instrução	Descrição	Binário	Hexa
0.	movl 1	# R0 = 1	10000001	81
1.	movr \$R1 \$R0	# R1 = R0	01100100	64
2.	movh 4	# R0 = 1 + 4	01110100	74
3.	add \$R2 \$R0	# R2 = R2 + R0	10010001	98
4.	slr \$R0 \$R1	#R0 = R0 << R1	11100001	E1
5.	movr \$R3 \$R0	# R3 = R0	01101100	6C
6.	movl 7	# R0 = 0 + 7	10000111	87
7.	brzr \$R2 \$R3	# if(R2 == 0) goto R3	00001011	0B
8.	sub \$R2 \$R1	# R2 = R2 - R1	10101001	A9
9.	jr \$R0	# goto R0	00100000	20
10.	and \$R1 \$R1	# R1 = R1 && R1	10110101	B5
11.	or \$R1 \$R1	# R1 = R1    R1	11000101	C5
12.	movl 0	# R0 = 0 + 0	10000000	80
13.	not \$R1 \$R0	# R1 = !R0	11010100	D4
14.	movl 1	# R0 = 0 + 1	10000001	81
15.	movr \$R2 \$R0	# R2 = R0	01101000	68
16.	st \$R3 \$R0	# M[R0] = R3	01011100	5C
17.	ld \$R0 \$R2	# R0 = M[R2]	01000010	42
18.	srr \$R0 \$R2	# R0 >> R2	11110010	F2
19.	brzi 3	# if(R0 == 0) goto PC + 3	00010011	13
20.	sub \$R0 \$R2	# R0 = R0 - R2	10100010	A2
21.	ji -2	# goto PC + ( - 2)	00111110	3E
22.	finish	# end of program	00000000	00