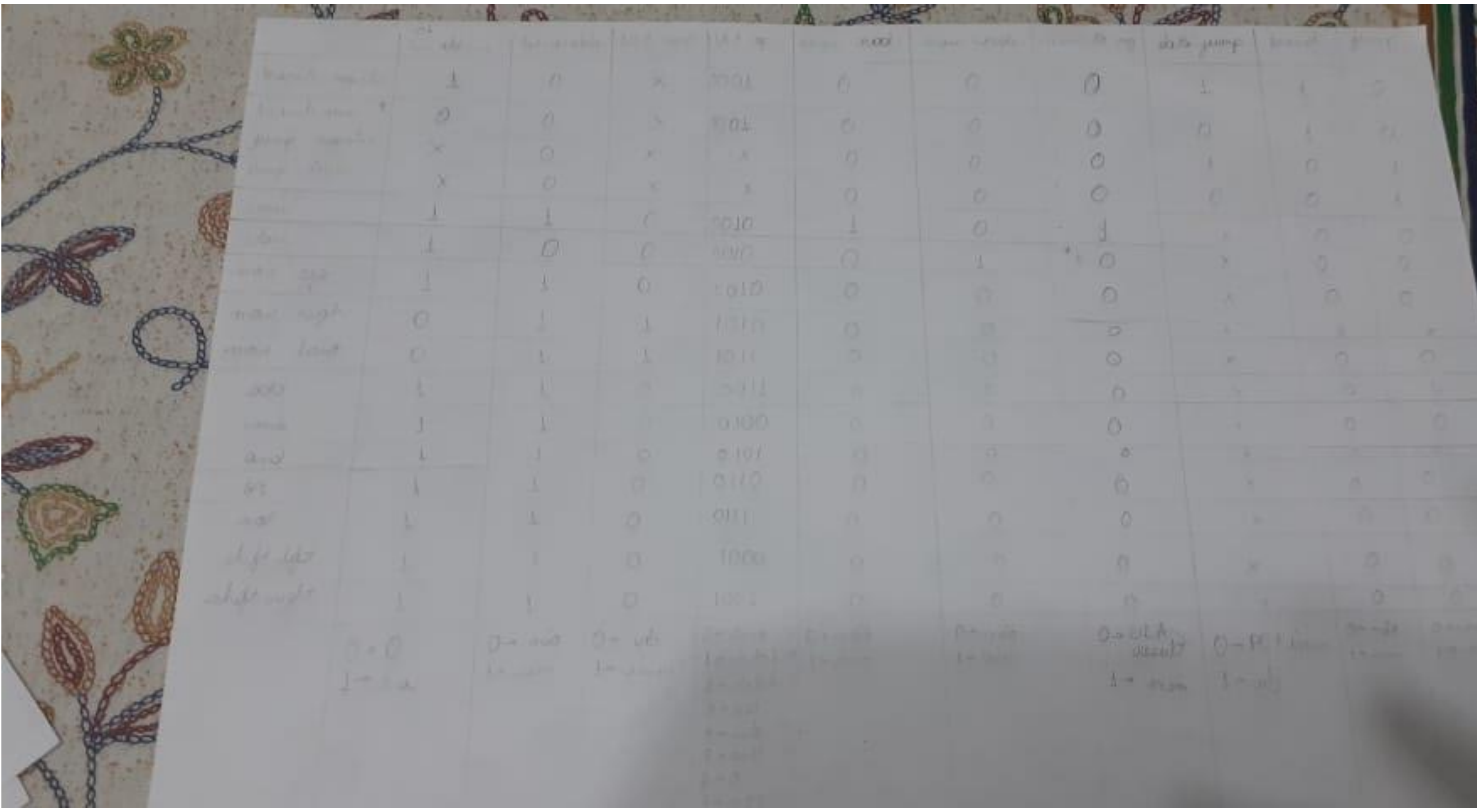
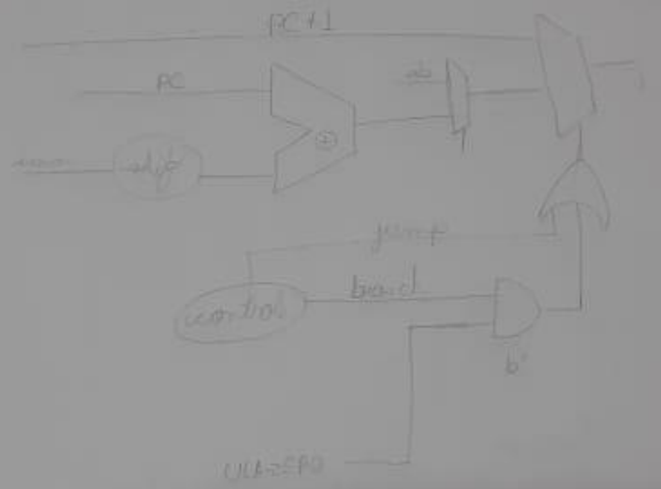
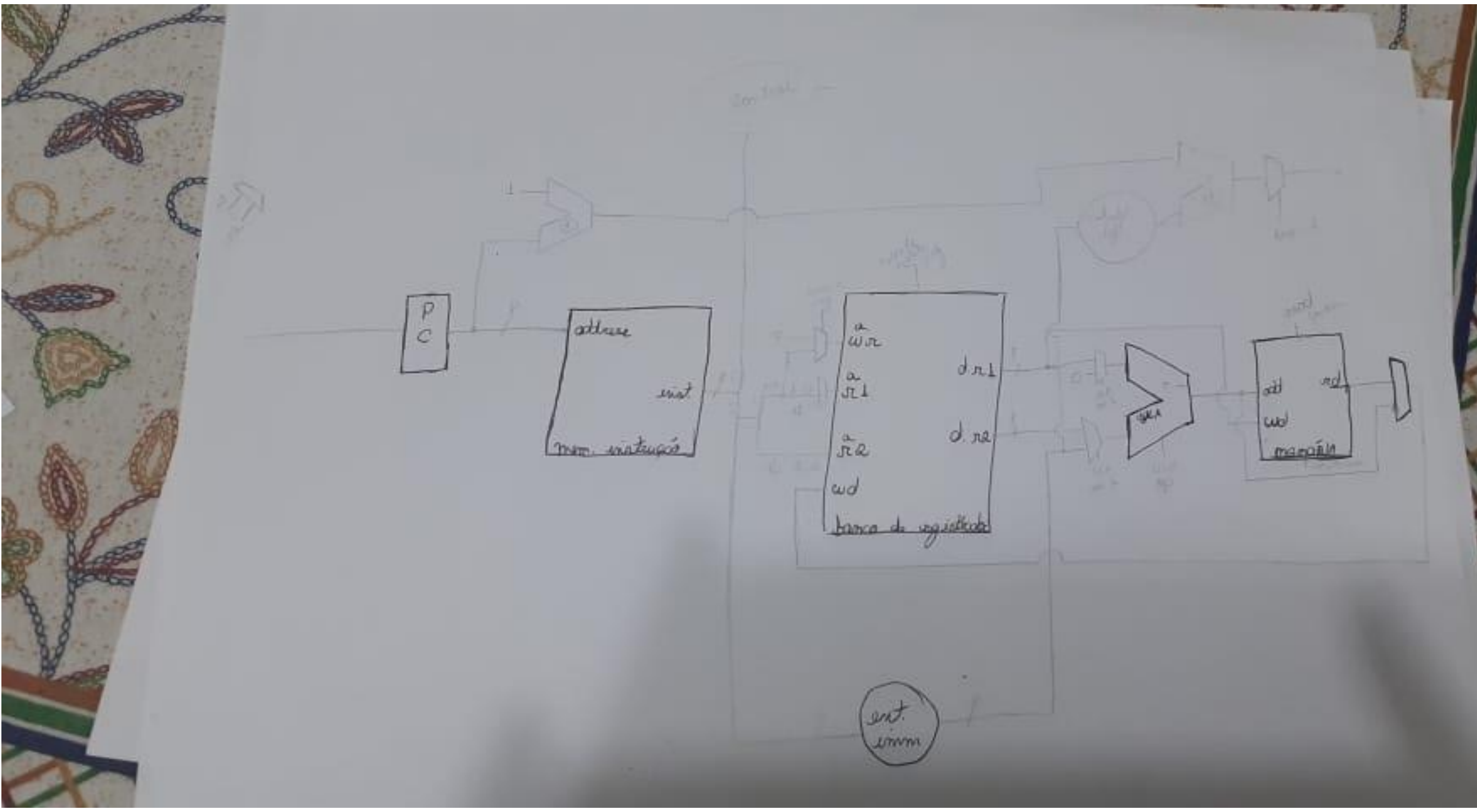
Trabalho Arquitetura – Sagui

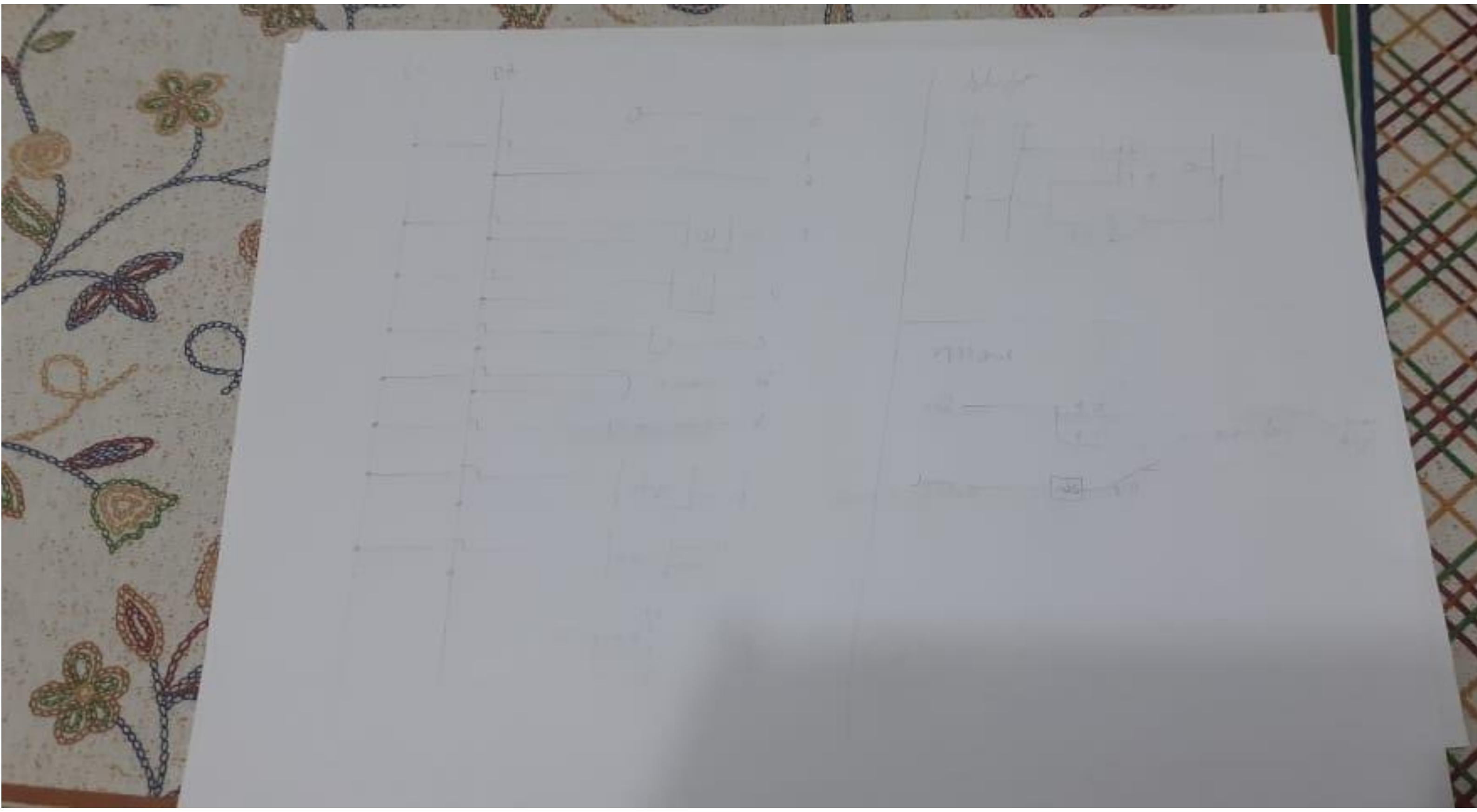
Heloísa Dias Viotto – GRR20231942

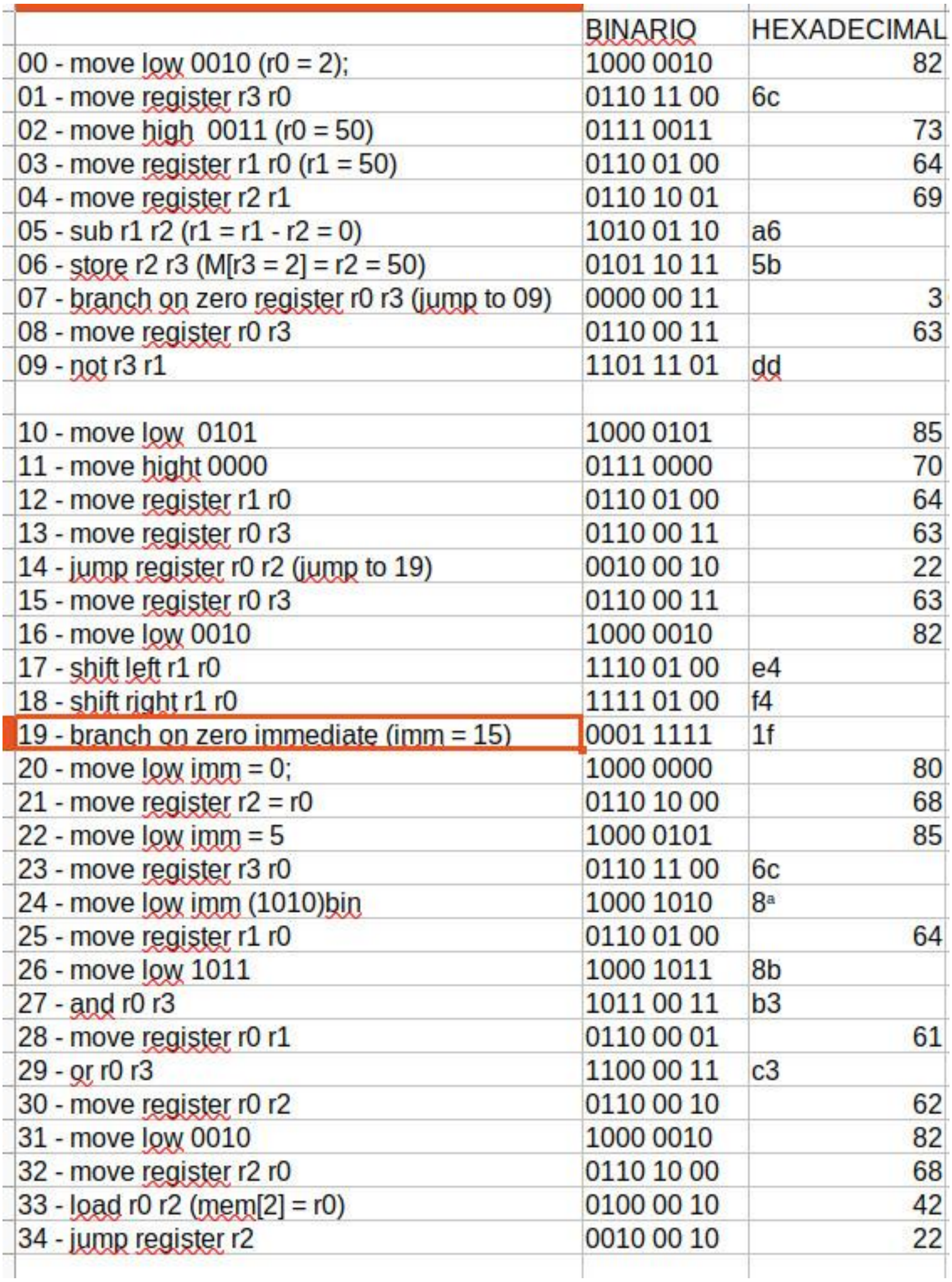


	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	data jump	[illegible]	[illegible]
branch equal	1	0	x	0001	0	0	0	1	1	0
[illegible]	0	0	x	0001	0	0	0	0	1	[illegible]
jump [illegible]	x	0	x	x	0	0	0	1	0	1
jump [illegible]	x	0	x	x	0	0	0	0	0	1
[illegible]	1	1	0	0010	1	0	1	x	0	0
[illegible]	1	0	0	0000	0	1	0	x	0	0
[illegible]	1	1	0	0010	0	0	0	x	0	0
mov right	0	1	1	1010	0	0	0	x	x	x
mov left	0	1	1	1011	0	0	0	x	0	0
add	1	1	0	0011	0	0	0	x	0	0
[illegible]	1	1	[illegible]	0100	0	0	0	x	0	0
and	1	1	0	0101	0	0	0	x	[illegible]	[illegible]
or	1	1	0	0110	0	0	0	x	0	0
not	1	1	0	0111	0	0	0	x	0	0
shift left	1	1	0	1000	0	0	0	x	0	0
shift right	1	1	0	1001	0	0	0	x	0	0
	0 → 0 1 → [illegible]	0 → [illegible] 1 → [illegible]	0 → [illegible] 1 → [illegible]	[illegible]	[illegible]	[illegible]	0 → ULA result 1 → [illegible]	0 → [illegible] 1 → [illegible]	[illegible]	[illegible]

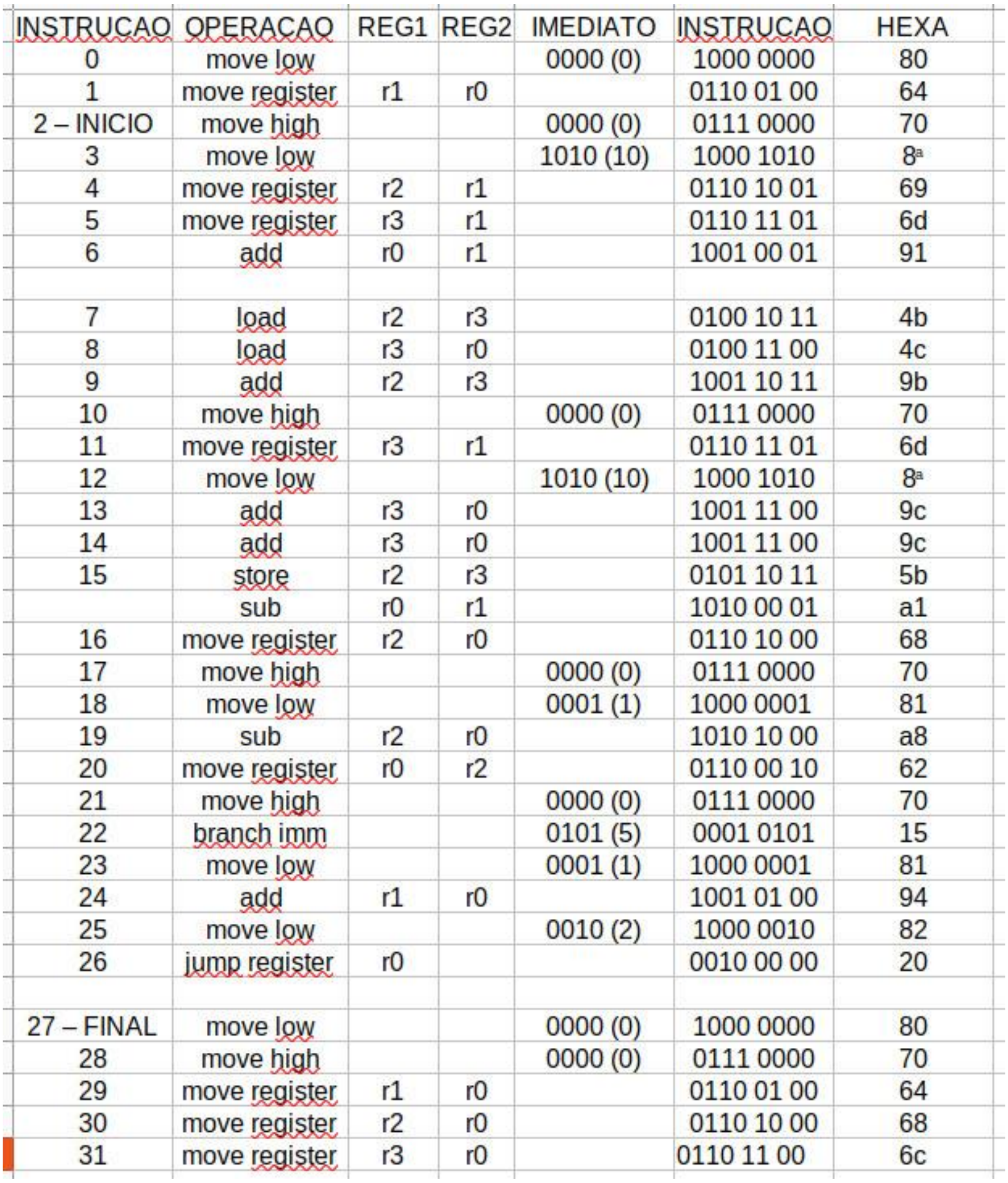

PC+1
PC
shft
⊕
jump
control
branch
b'
ULA=ZERO

control
1
P C
address
inst
mem. instruções
a wr
a r1
a r2
wd
d.r1
d.r2
banco de registradores
ULA
add
wd
rd
memória
int. imm





	BINARIO	HEXADECIMAL
00 - move low 0010 (r0 = 2);	1000 0010	82
01 - move register r3 r0	0110 11 00	6c
02 - move high 0011 (r0 = 50)	0111 0011	73
03 - move register r1 r0 (r1 = 50)	0110 01 00	64
04 - move register r2 r1	0110 10 01	69
05 - sub r1 r2 (r1 = r1 - r2 = 0)	1010 01 10	a6
06 - store r2 r3 (M[r3 = 2] = r2 = 50)	0101 10 11	5b
07 - branch on zero register r0 r3 (jump to 09)	0000 00 11	3
08 - move register r0 r3	0110 00 11	63
09 - not r3 r1	1101 11 01	dd
10 - move low 0101	1000 0101	85
11 - move hight 0000	0111 0000	70
12 - move register r1 r0	0110 01 00	64
13 - move register r0 r3	0110 00 11	63
14 - jump register r0 r2 (jump to 19)	0010 00 10	22
15 - move register r0 r3	0110 00 11	63
16 - move low 0010	1000 0010	82
17 - shift left r1 r0	1110 01 00	e4
18 - shift right r1 r0	1111 01 00	f4
19 - branch on zero immediate (imm = 15)	0001 1111	1f
20 - move low imm = 0;	1000 0000	80
21 - move register r2 = r0	0110 10 00	68
22 - move low imm = 5	1000 0101	85
23 - move register r3 r0	0110 11 00	6c
24 - move low imm (1010)bin	1000 1010	8ª
25 - move register r1 r0	0110 01 00	64
26 - move low 1011	1000 1011	8b
27 - and r0 r3	1011 00 11	b3
28 - move register r0 r1	0110 00 01	61
29 - or r0 r3	1100 00 11	c3
30 - move register r0 r2	0110 00 10	62
31 - move low 0010	1000 0010	82
32 - move register r2 r0	0110 10 00	68
33 - load r0 r2 (mem[2] = r0)	0100 00 10	42
34 - jump register r2	0010 00 10	22



INSTRUCAO	OPERACAO	REG1	REG2	IMEDIATO	INSTRUCAO	HEXA
0	move low			0000 (0)	1000 0000	80
1	move register	r1	r0		0110 01 00	64
2 – INICIO	move high			0000 (0)	0111 0000	70
3	move low			1010 (10)	1000 1010	8ª
4	move register	r2	r1		0110 10 01	69
5	move register	r3	r1		0110 11 01	6d
6	add	r0	r1		1001 00 01	91
7	load	r2	r3		0100 10 11	4b
8	load	r3	r0		0100 11 00	4c
9	add	r2	r3		1001 10 11	9b
10	move high			0000 (0)	0111 0000	70
11	move register	r3	r1		0110 11 01	6d
12	move low			1010 (10)	1000 1010	8ª
13	add	r3	r0		1001 11 00	9c
14	add	r3	r0		1001 11 00	9c
15	store	r2	r3		0101 10 11	5b
	sub	r0	r1		1010 00 01	a1
16	move register	r2	r0		0110 10 00	68
17	move high			0000 (0)	0111 0000	70
18	move low			0001 (1)	1000 0001	81
19	sub	r2	r0		1010 10 00	a8
20	move register	r0	r2		0110 00 10	62
21	move high			0000 (0)	0111 0000	70
22	branch imm			0101 (5)	0001 0101	15
23	move low			0001 (1)	1000 0001	81
24	add	r1	r0		1001 01 00	94
25	move low			0010 (2)	1000 0010	82
26	jump register	r0			0010 00 00	20
27 – FINAL	move low			0000 (0)	1000 0000	80
28	move high			0000 (0)	0111 0000	70
29	move register	r1	r0		0110 01 00	64
30	move register	r2	r0		0110 10 00	68
31	move register	r3	r0		0110 11 00	6c