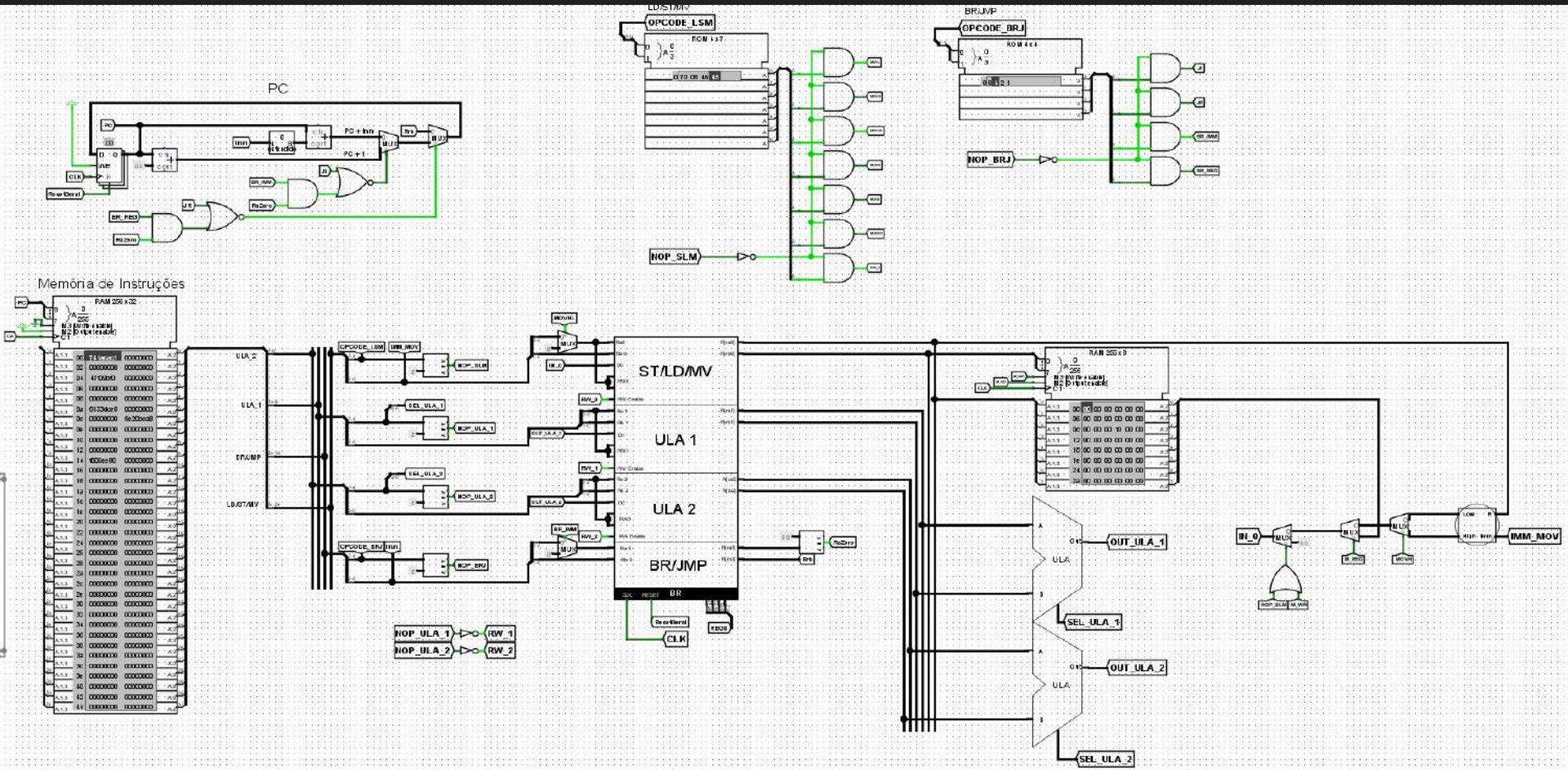
SAGUI
RABO-LONGO
VLIW

JORGE LUCAS VICILLI JABCZENSKI

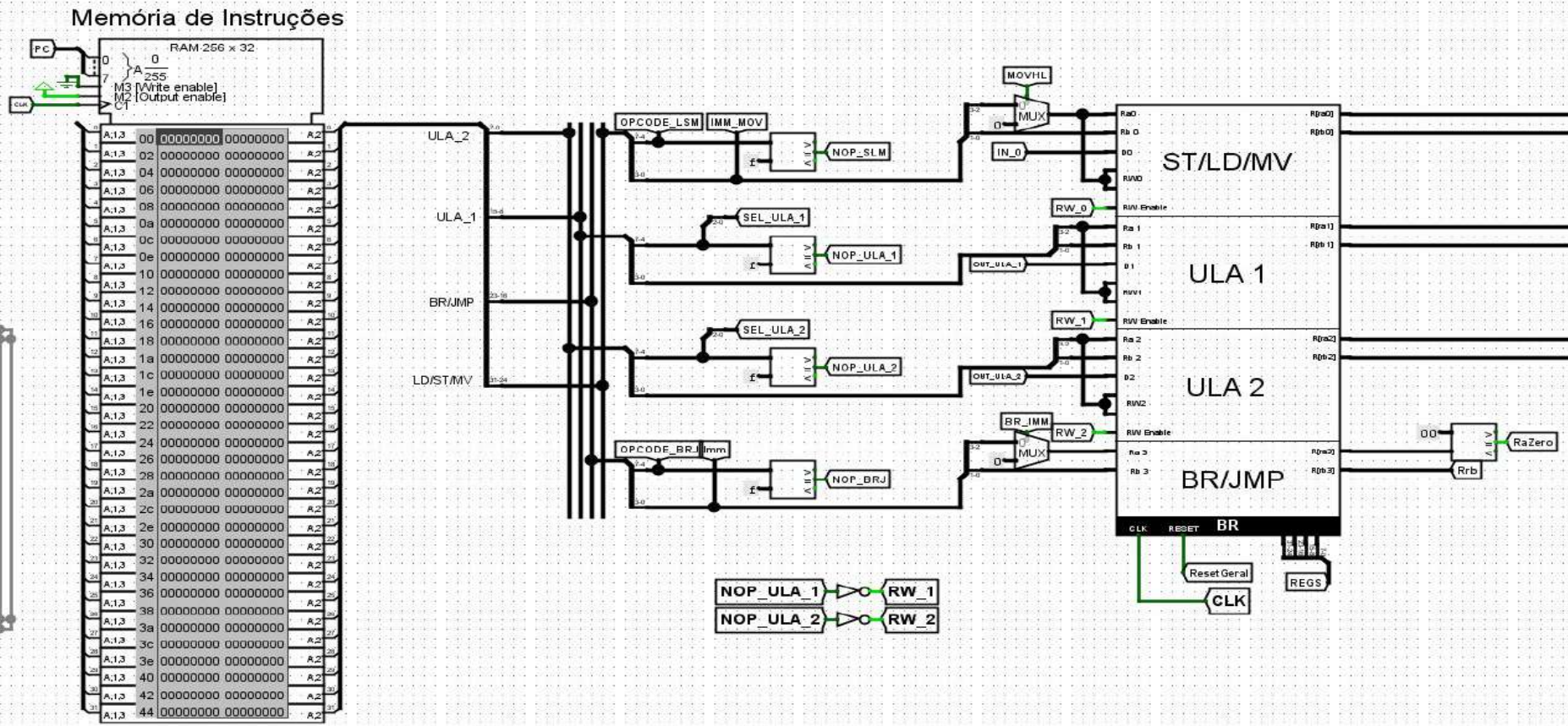
VISÃO GERAL


PC
LD/STMV
OPCODE_LSM
NOP_SLM
BR/JMP
OPCODE_BRJ
NOP_BRJ
Memória de Instruções
ULA_2
ULA_1
BR/JMP
LD/ST/MV
ST/LD/MV
ULA 1
ULA 2
BR/JMP
NOP_ULA_1
NOP_ULA_2
RW_1
RW_2
OUT_ULA_1
OUT_ULA_2
SEL_ULA_1
SEL_ULA_2
IMM_MOV
CLK

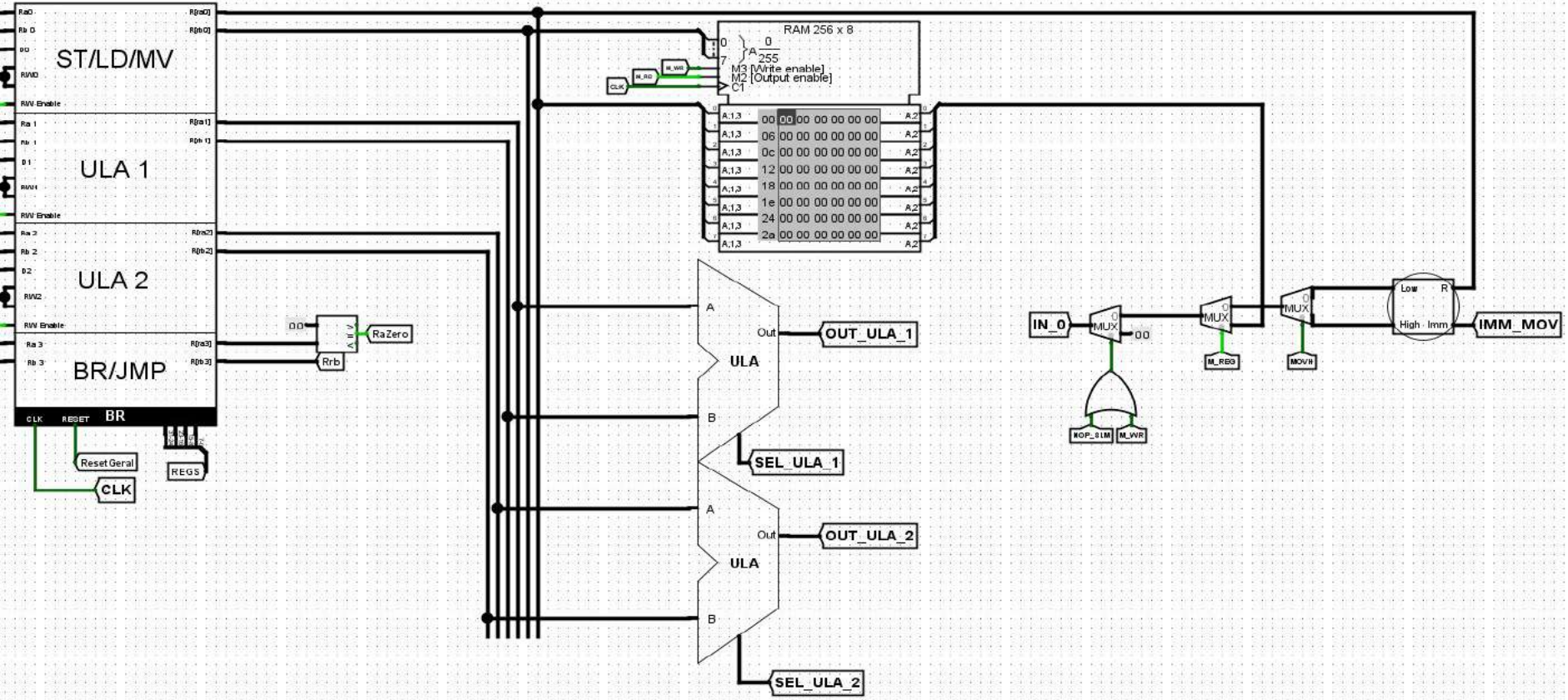
Memória de Instruções

Como as memórias no Logisim não podem ser lidas em mais de um endereço diferente simultaneamente, a Memória de Instruções agora possui 32 bits de largura ao invés dos 8.

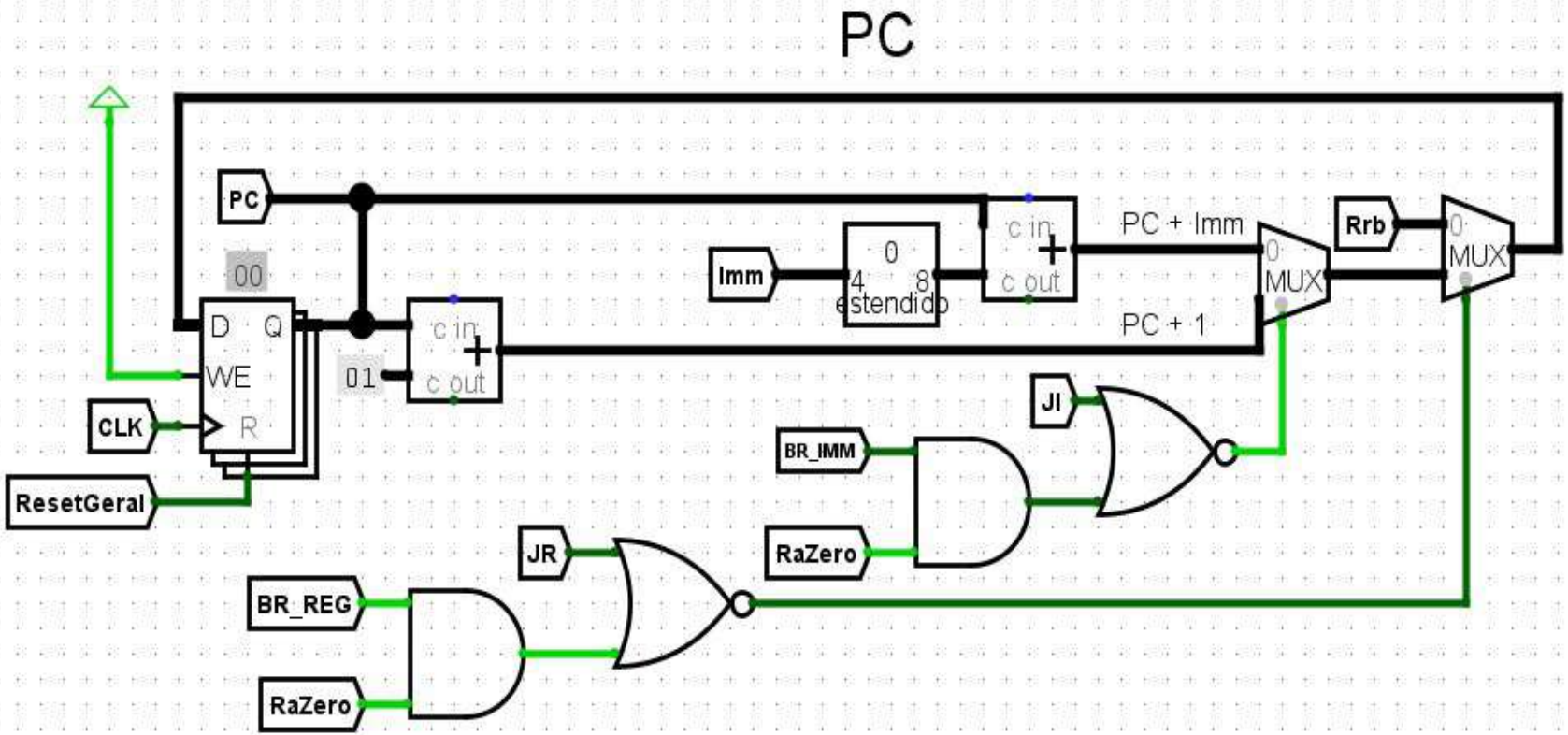
MEMÓRIAS DE INSTRUÇÕES + BANCO DE REGISTRADORES



UNIDADES FUNCIONAIS



PC

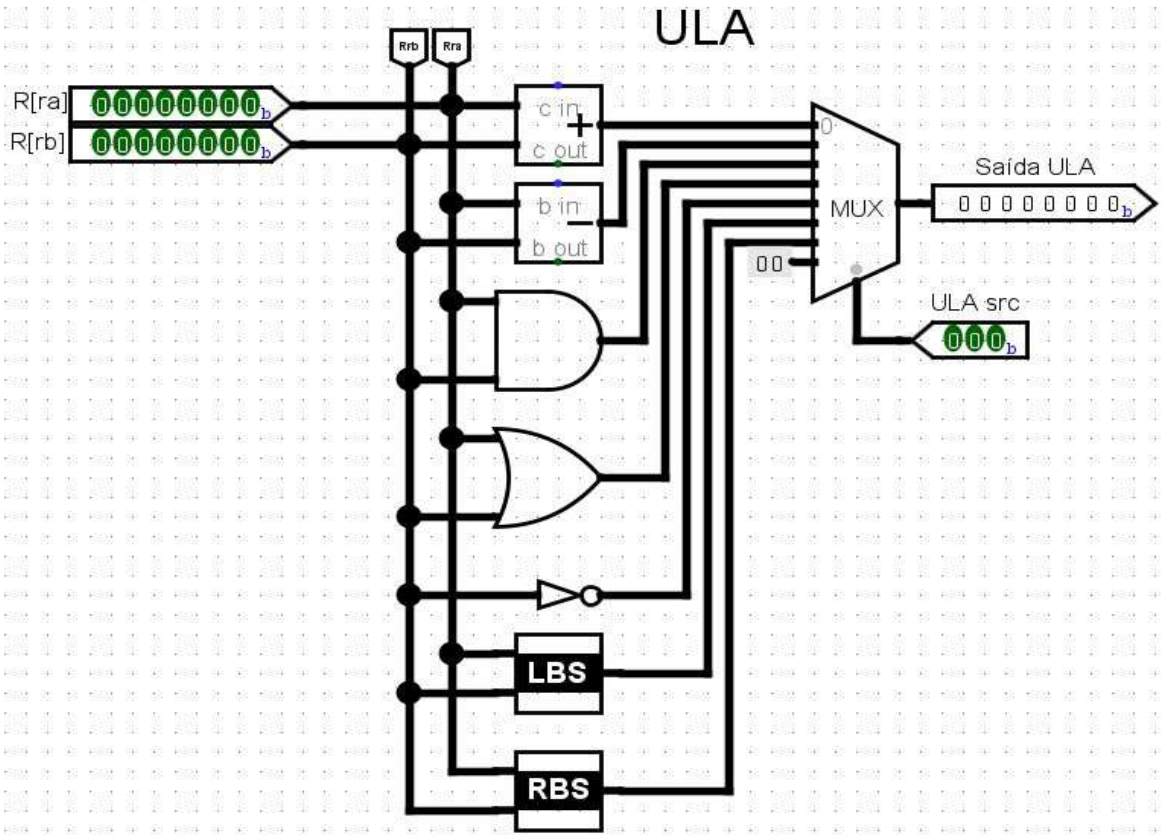

PC
PC
00
D
Q
WE
R
CLK
ResetGeral
01
c in
c out
Imm
0
4
8
estendido
c in
c out
PC + Imm
PC + 1
0
MUX
Rrb
0
MUX
JI
BR_IMM
RaZero
JR
BR_REG
RaZero

Unidade Lógica e Aritmética

Foram feitas duas alterações comparando com a ULA utilizada no projeto anterior:

- O sinal R[ra] == 0 foi retirado pois não é mais necessário
- A entrada 8 da ULA agora devolve 0 ao invés de R[ra] pois a instrução “Move Register” foi removida da ISA

UNIDADE LÓGICA E ARITMÉTICA



Banco de Registradores

Para conseguir realizar várias leituras e escritas simultâneas, foram adicionados DEMUX e portas OR no Banco de Registradores utilizado no projeto passado do Sagui, além da adição de 6 outros MUX de leitura.

BANCO DE REGISTRADORES

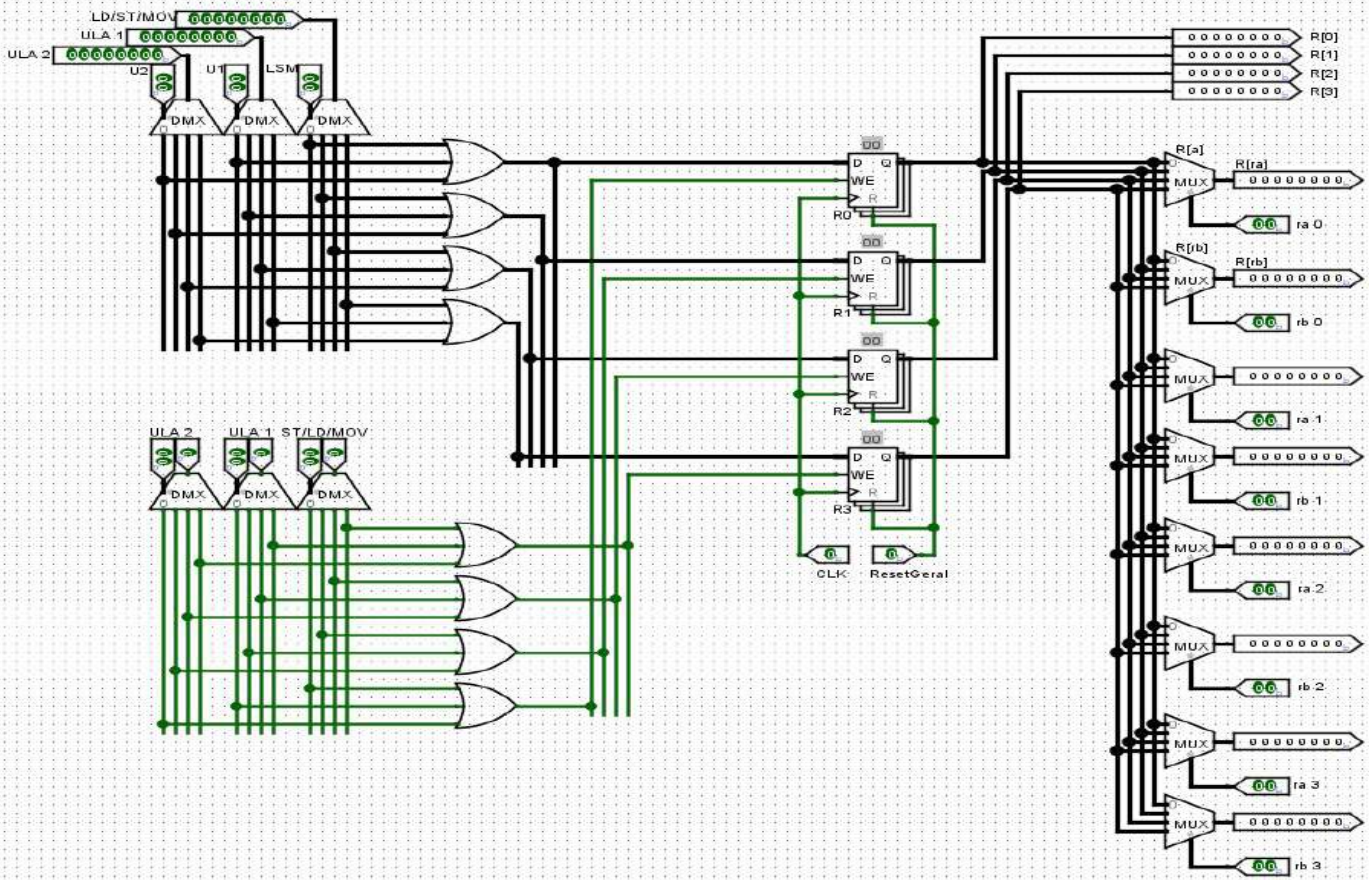


TABELA DE CONTROLE - UF DE BRANCHES E JUMPS

INSTRUÇÃO			BR_REG	BR_IMM	JR	JI
0	0000	BRZR	1	0	0	0
1	0001	BRZI	0	1	0	0
2	0010	JR	0	0	1	0
3	0011	JI	0	0	0	1

TABELA DE CONTROLE - UF DE ST/LD/MOV

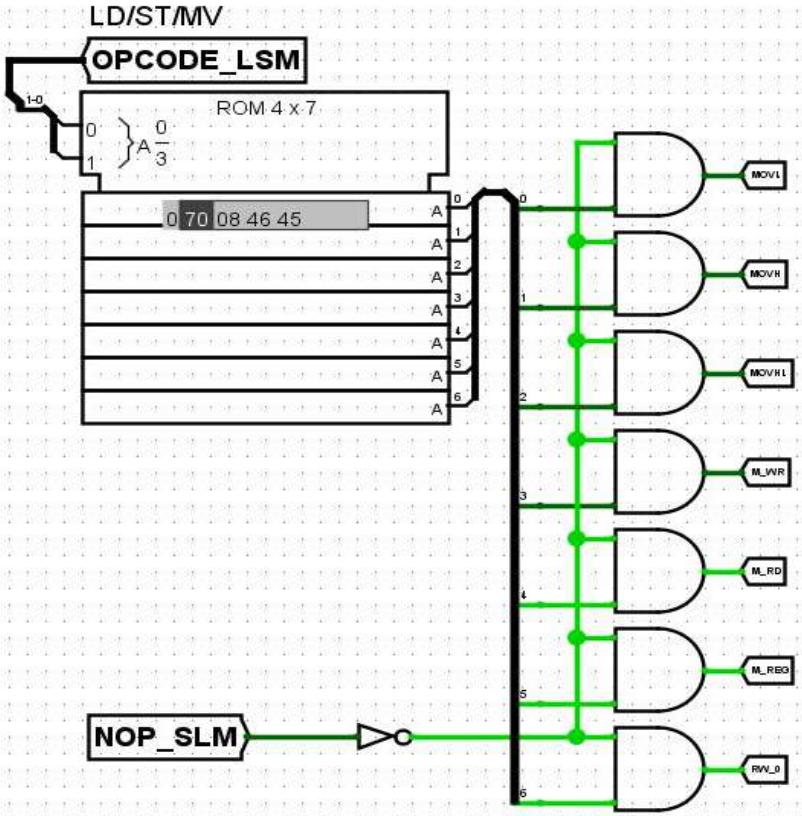
INSTRUÇÃO			RW	M_TO REG	M_RD	M_WR	MOVHL	MOVH	MOVL
4	0100	LD	1	1	1	0	0	0	0
5	0101	ST	0	0	0	1	0	0	0
6	0110	MOVH	1	0	0	0	1	1	0
7	0111	MOVL	1	0	0	0	1	0	1

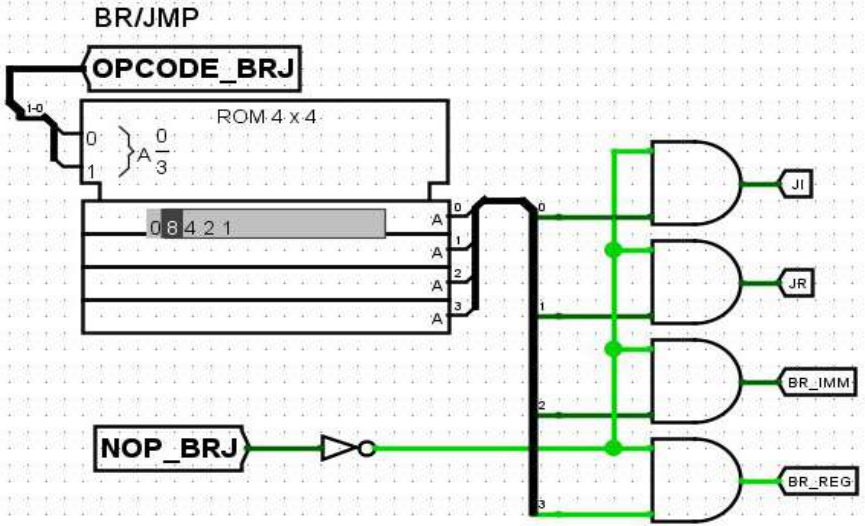
Tabelas de Controle

Como seriam utilizadas 4 memórias de controle usando 4 memórias grandes a maioria dos seus espaços seria inutilizado. Então foi decidido por usar 4 memórias pequenas de apenas 4 endereços, uma para cada UF (no final foram necessárias apenas 2, uma para BR/JMP e uma para LD/ST/MOV). Para fazer isso, são enviados apenas os dois bits menos significativos dos OPCODES.

Também foram adicionadas portas AND para zerar todos os sinais de controle caso a instrução passada para aquela Unidade Funcional for um NOP.

MEMÓRIAS DE CONTROLE

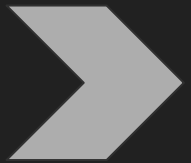




Controle da ULA

O controle das ULAs é feita de forma diferente das demais Unidades Funcionais. Foram utilizados os 3 bits menos significativos do OPCODE diretamente na seleção de operação da ULA, pois estes correspondiam exatamente a qual operação deveria ser realizada, o que evitou o uso de outras duas memórias de controle.

INSTRUÇÃO	
1000	ADD
1001	SUB
1010	AND
1011	OR
1100	NOT
1101	SLR
1110	SRR



CONTROLE ULA	
000	ADD
001	SUB
010	AND
011	OR
100	NOT
101	SLR
110	SRR