Arquitetura de Computadores

Turma LI21N/LT21N

AULA 11
Arquitectura de Von Neumann

TÓPICOS A ABORDAR

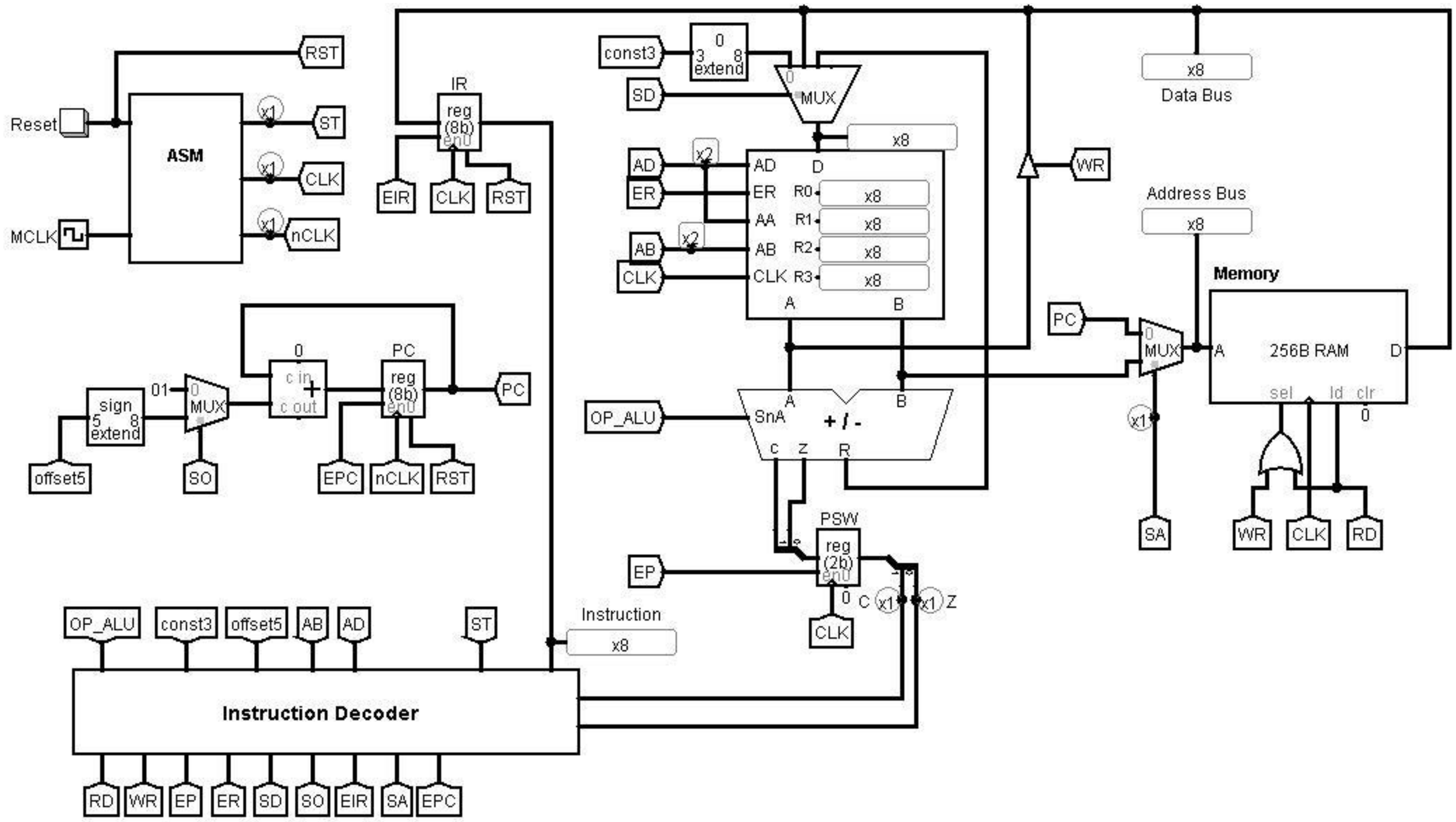
- *Características principais*
- *Diferenças relativas à arquitectura de Harvard*
- *Alterações à microarquitectura para implementação no modelo de Von Neumann*

Ano Lectivo 2019/2020

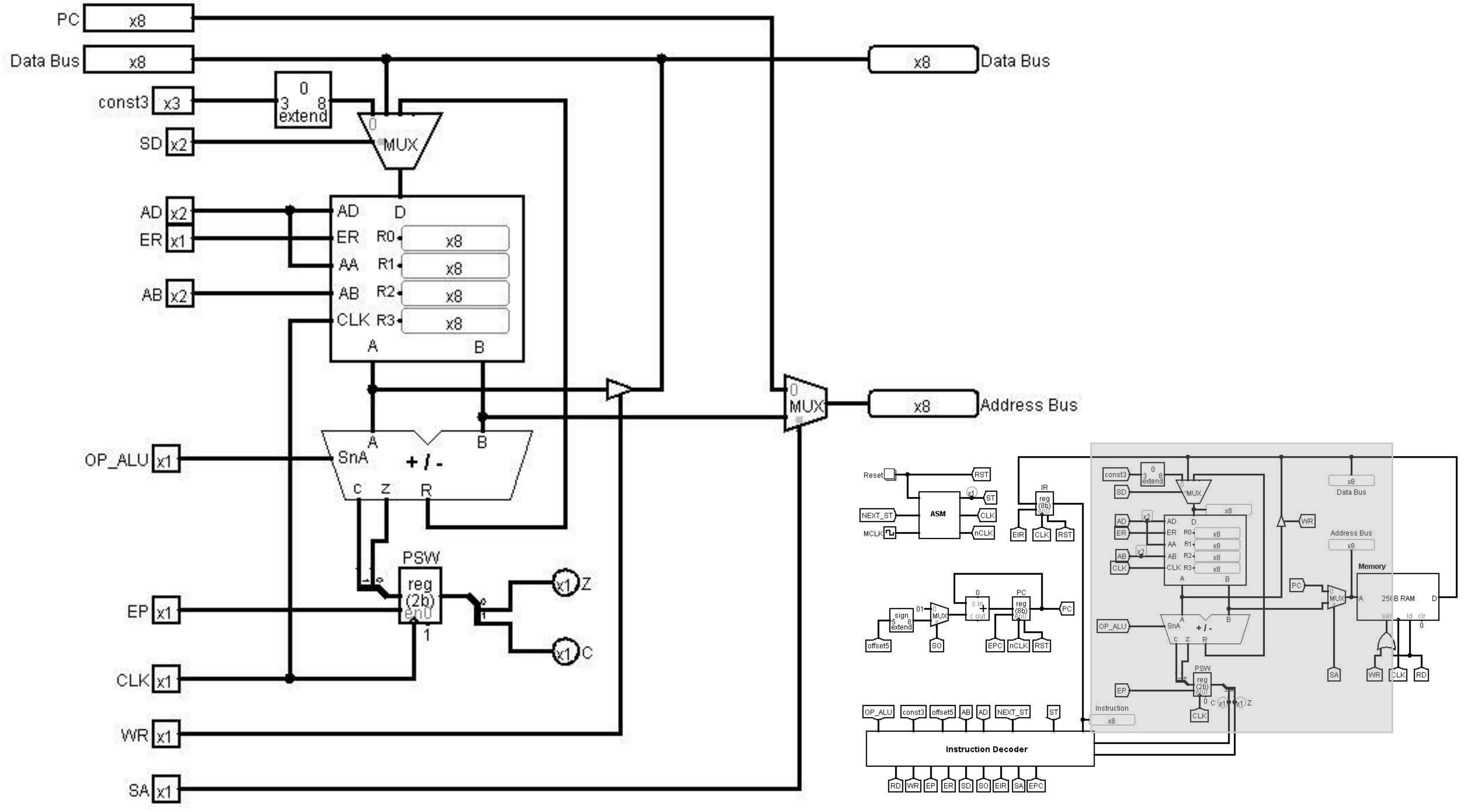
2º Semestre

Prof. Jorge Fonseca

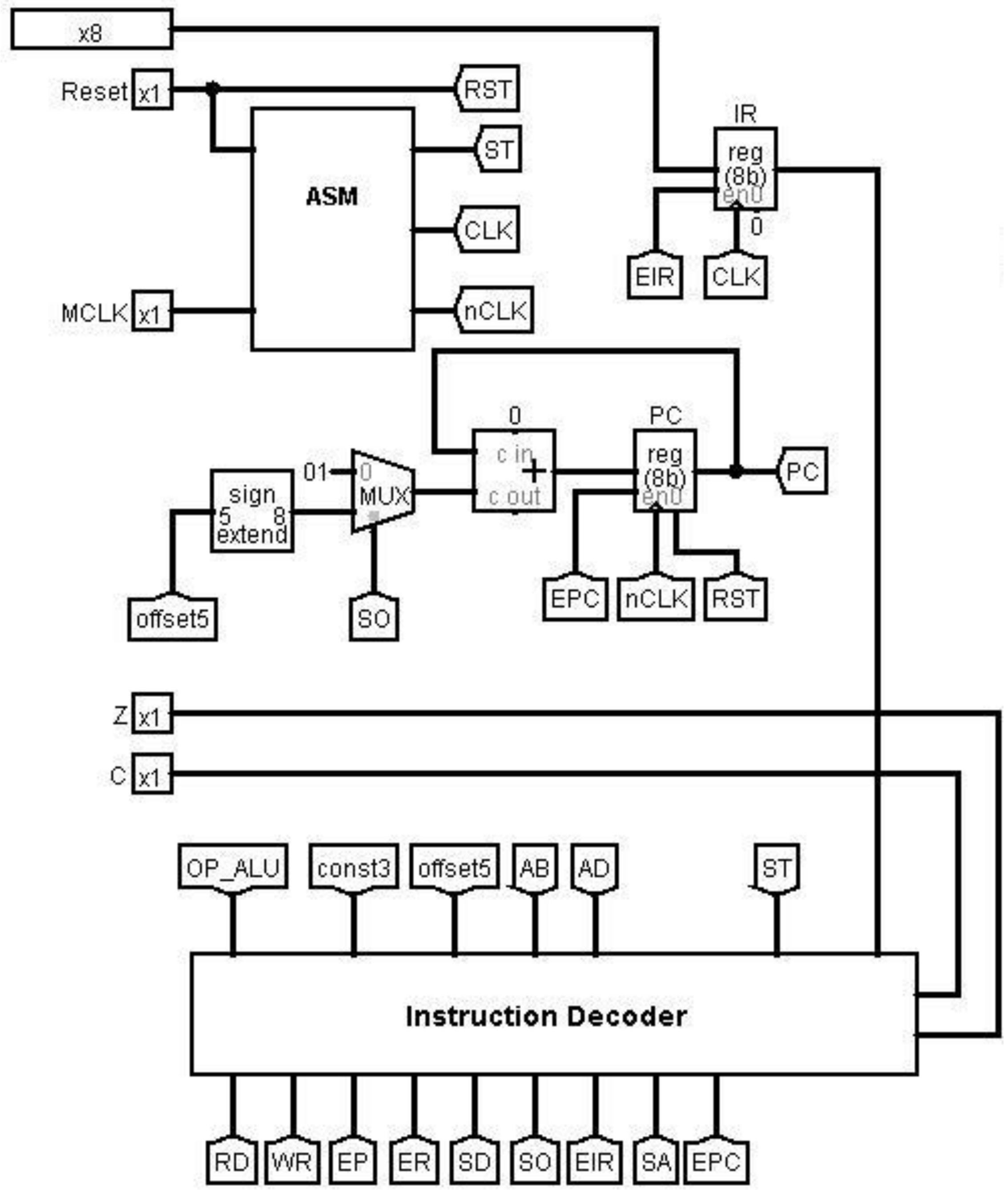
ARQUITECTURA DE VON NEUMANN

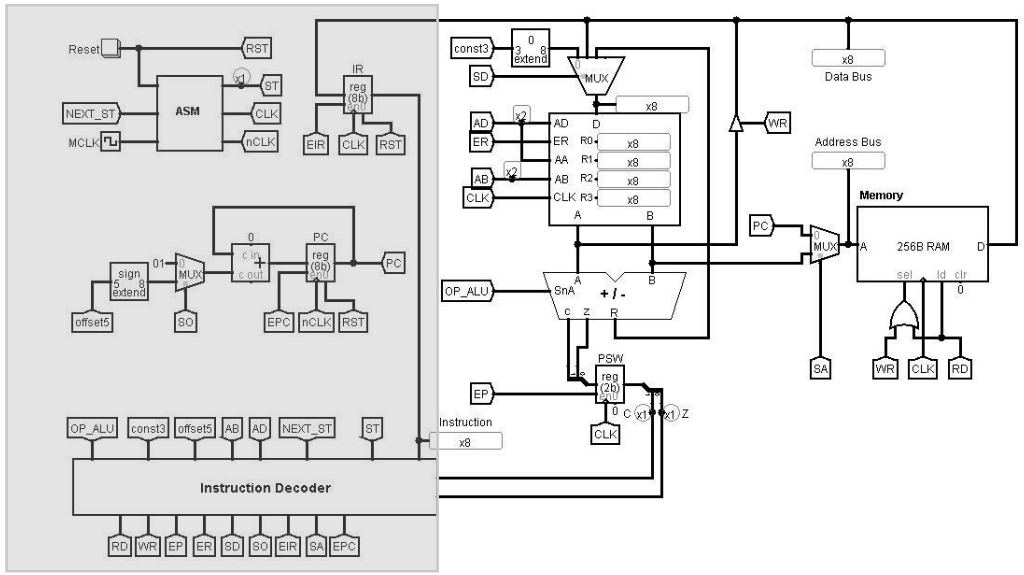


CPU – UNIDADE PROCESSAMENTO

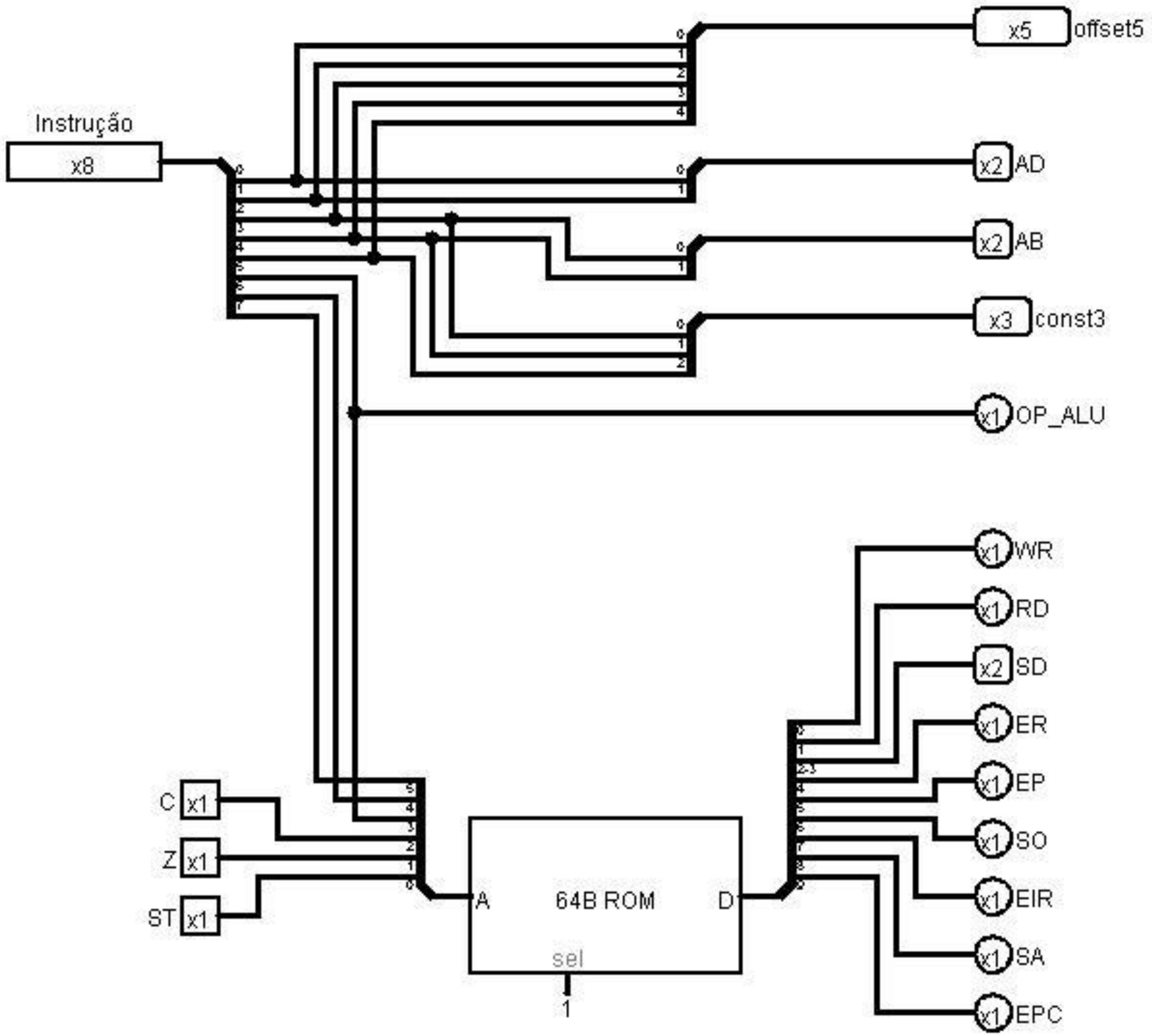


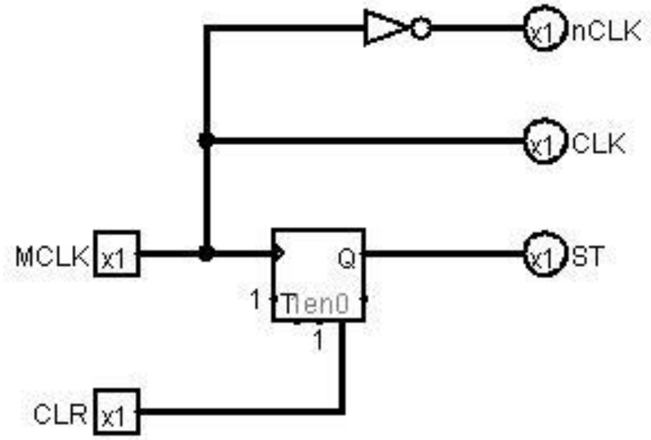
CPU – UNIDADE CONTROLO





INSTRUCTION DECODER




nCLK
CLK
MCLK
ST
CLR
Q
T
Ten0
1
1
1
x1

ISA – INSTRUCTION SET ARCHITECTURE

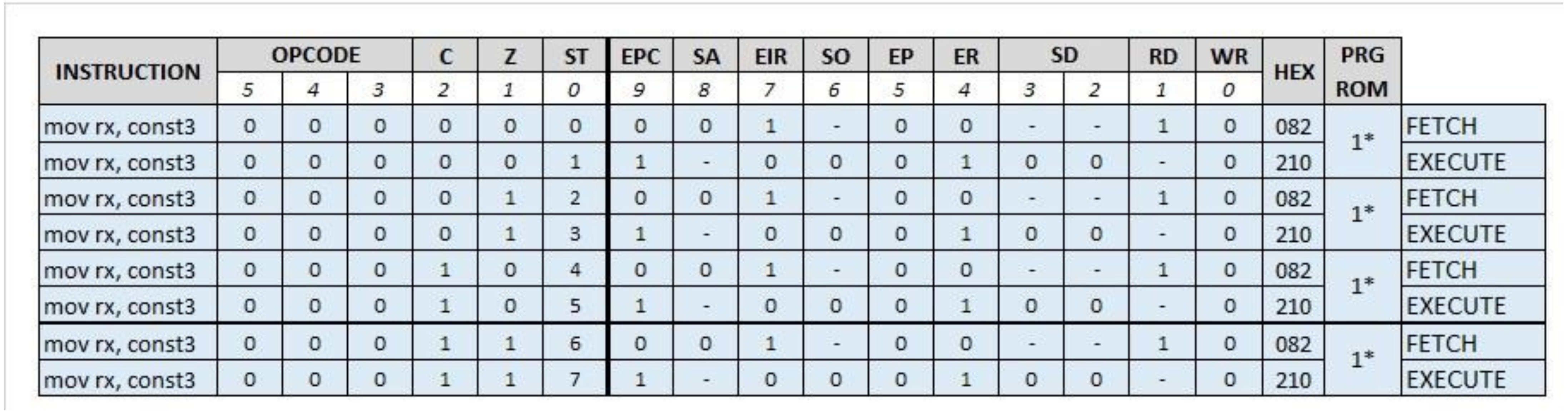
			OP_ALU					
INSTRUCTION	OPCODE				AB		AA/AD	
	7	*6*	*5*	*4*	*3*	*2*	*1*	*0*
ldi rx, const3	0	0	0	const3			rx	
ld rx, [ry]	0	0	1	-	ry		rx	
st rx, [ry]	0	1	1	-	rz		rx	
add rx, rz	1	0	0	-	rz		rx	
sub rx, rz	1	0	1	-	rz		rx	
bcc offset5	1	1	0	offset5				
bzs offset5	1	1	0	offset5				
b offset5	1	1	1	offset5				

INSTRUCTION DECODER

INSTRUCTION	OPCODE			C	Z	ST	EPC	SA	EIR	SO	EP	ER	SD		RD	WR	HEX	PRG ROM	
	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0			
mov rx, const3	0	0	0	-	-	0	0	0	1	-	0	0	-	-	1	0	082	4*	FETCH
mov rx, const3	0	0	0	-	-	1	1	-	0	0	0	1	0	0	-	0	210		EXECUTE
ld rx, [ry]	0	0	1	-	-	0	0	0	1	-	0	0	-	-	1	0	082	4*	FETCH
ld rx, [ry]	0	0	1	-	-	1	1	1	0	0	0	1	0	1	1	0	316		EXECUTE
bcc offset5	0	1	0	0	-	0	0	0	1	-	0	0	-	-	1	0	082	2*	FETCH
bcc offset5	0	1	0	0	-	1	1	-	0	1	0	0	-	-	-	0	240		EXECUTE
bcc offset5	0	1	0	1	-	0	0	0	1	-	0	0	-	-	1	0	082	2*	FETCH
bcc offset5	0	1	0	1	-	1	1	-	0	0	0	0	-	-	-	0	200		EXECUTE
st rx, [ry]	0	1	1	-	-	0	0	0	1	-	0	0	-	-	1	0	082	4*	FETCH
st rx, [ry]	0	1	1	-	-	1	1	1	0	0	0	0	-	-	0	1	301		EXECUTE
add rx, ry	1	0	0	-	-	0	0	0	1	-	0	0	-	-	1	0	082	4*	FETCH
add rx, ry	1	0	0	-	-	1	1	-	0	0	1	1	1	0	-	0	238		EXECUTE
sub rx, ry	1	0	1	-	-	0	0	0	1	-	0	0	-	-	1	0	082	4*	FETCH
sub rx, ry	1	0	1	-	-	1	1	-	0	0	1	1	1	0	-	0	238		EXECUTE
bzs offset5	1	1	0	-	0	0	0	0	1	-	0	0	-	-	1	0	082	1*	FETCH
bzs offset5	1	1	0	-	0	1	1	-	0	0	0	0	-	-	-	0	200		EXECUTE
bzs offset5	1	1	0	-	1	0	0	0	1	-	0	0	-	-	1	0	082	1*	FETCH
bzs offset5	1	1	0	-	1	1	1	-	0	1	0	0	-	-	-	0	240		EXECUTE
bzs offset5	1	1	0	-	0	0	0	0	1	-	0	0	-	-	1	0	082	1*	FETCH
bzs offset5	1	1	0	-	0	1	1	-	0	0	0	0	-	-	-	0	200		EXECUTE
bzs offset5	1	1	0	-	1	0	0	0	1	-	0	0	-	-	1	0	082	1*	FETCH
bzs offset5	1	1	0	-	1	1	1	-	0	1	0	0	-	-	-	0	240		EXECUTE
b offset5	1	1	1	-	-	0	0	0	1	-	0	0	-	-	1	0	082	4*	FETCH
b offset5	1	1	1	-	-	1	1	-	0	1	0	0	-	-	-	0	240		EXECUTE

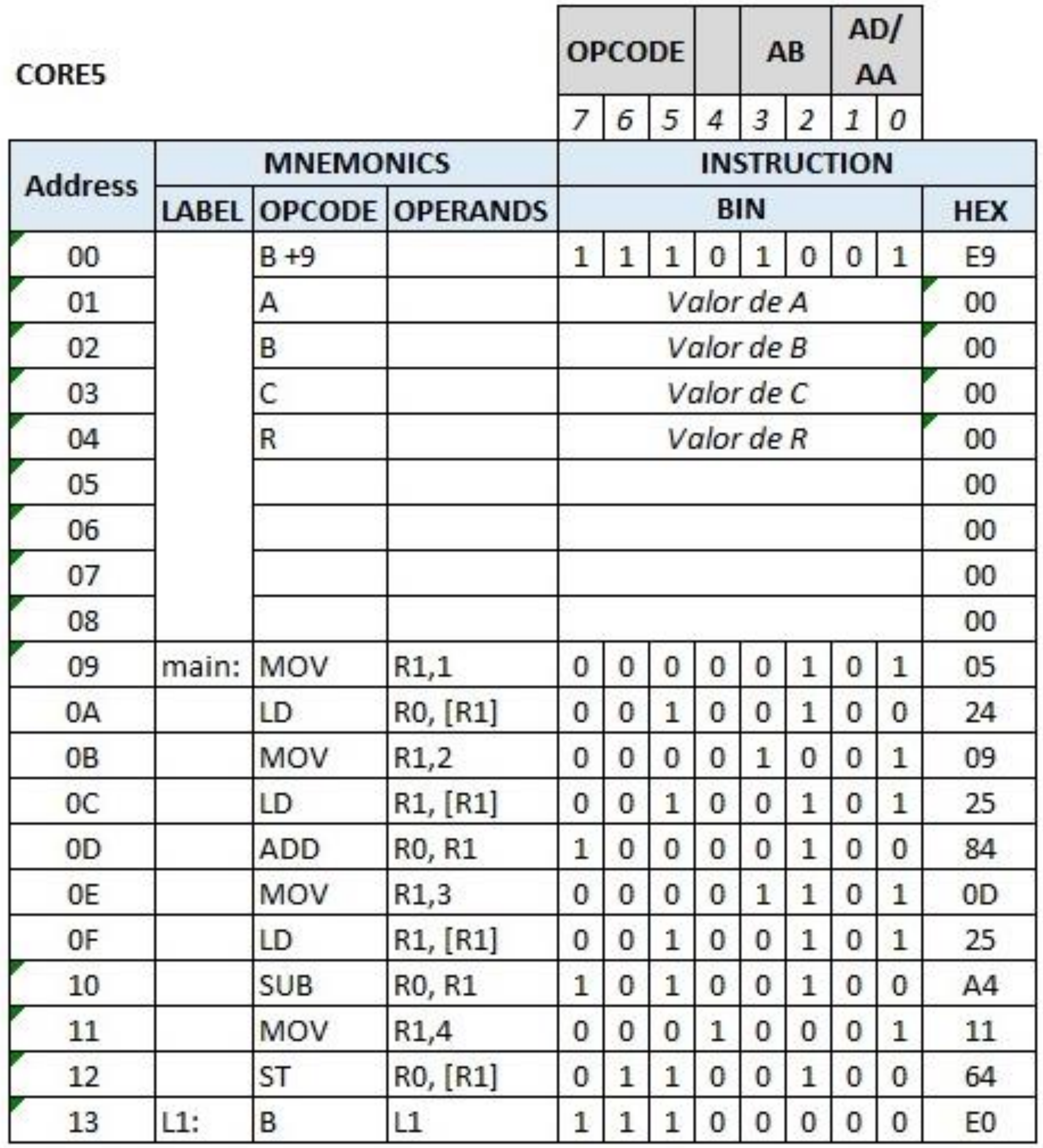
FICHEIRO PROGRAMAÇÃO ROM INSTRUCTION DECODER

```
v2.0 raw
082 210 082 210 082 210 082 210
082 316 082 316 082 316 082 316
082 240 082 240 082 200 082 200
082 301 082 301 082 301 082 301
082 238 082 238 082 238 082 238
082 238 082 238 082 238 082 238
082 200 082 240 082 200 082 240
082 240 082 240 082 240 082 240
```



INSTRUCTION	OPCODE			C	Z	ST	EPC	SA	EIR	SO	EP	ER	SD		RD	WR	HEX	PRG ROM	
	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0			
mov rx, const3	0	0	0	0	0	0	0	0	1	-	0	0	-	-	1	0	082	1*	FETCH
mov rx, const3	0	0	0	0	0	1	1	-	0	0	0	1	0	0	-	0	210		EXECUTE
mov rx, const3	0	0	0	0	1	2	0	0	1	-	0	0	-	-	1	0	082	1*	FETCH
mov rx, const3	0	0	0	0	1	3	1	-	0	0	0	1	0	0	-	0	210		EXECUTE
mov rx, const3	0	0	0	1	0	4	0	0	1	-	0	0	-	-	1	0	082	1*	FETCH
mov rx, const3	0	0	0	1	0	5	1	-	0	0	0	1	0	0	-	0	210		EXECUTE
mov rx, const3	0	0	0	1	1	6	0	0	1	-	0	0	-	-	1	0	082	1*	FETCH
mov rx, const3	0	0	0	1	1	7	1	-	0	0	0	1	0	0	-	0	210		EXECUTE

EXERCÍCIO



CORE5

				OPCODE				AB		AD/ AA		
				7	6	5	4	3	2	1	0	
Address	MNEMONICS			INSTRUCTION								
	LABEL	OPCODE	OPERANDS	BIN								HEX
00		B +9		1	1	1	0	1	0	0	1	E9
01		A		Valor de A								00
02		B		Valor de B								00
03		C		Valor de C								00
04		R		Valor de R								00
05												00
06												00
07												00
08												00
09	main:	MOV	R1,1	0	0	0	0	0	1	0	1	05
0A		LD	R0, [R1]	0	0	1	0	0	1	0	0	24
0B		MOV	R1,2	0	0	0	0	1	0	0	1	09
0C		LD	R1, [R1]	0	0	1	0	0	1	0	1	25
0D		ADD	R0, R1	1	0	0	0	0	1	0	0	84
0E		MOV	R1,3	0	0	0	0	1	1	0	1	0D
0F		LD	R1, [R1]	0	0	1	0	0	1	0	1	25
10		SUB	R0, R1	1	0	1	0	0	1	0	0	A4
11		MOV	R1,4	0	0	0	1	0	0	0	1	11
12		ST	R0, [R1]	0	1	1	0	0	1	0	0	64
13	L1:	B	L1	1	1	1	0	0	0	0	0	E0

Exercício1 : Escrever em Código Máquina um programa para determinar o A + B – C. Considere o operando A na posição de memória 1H, o operando B na posição de memória 2H e o operando C na posição de memória 3H. O resultado deve ser colocado na posição de memória 4H.

FICHEIRO COM A CODIFICAÇÃO DO PROGRAMA

```
v2.0 raw
E9
00
00
00
00
00
00
00
00
05
24
09
25
84
0D
25
A4
11
64
E0
```