

Arquitetura de Computadores

Turma LI21N/LT21N

AULA 8 UNIDADES DE CONTROLO MICROPROGRAMADAS

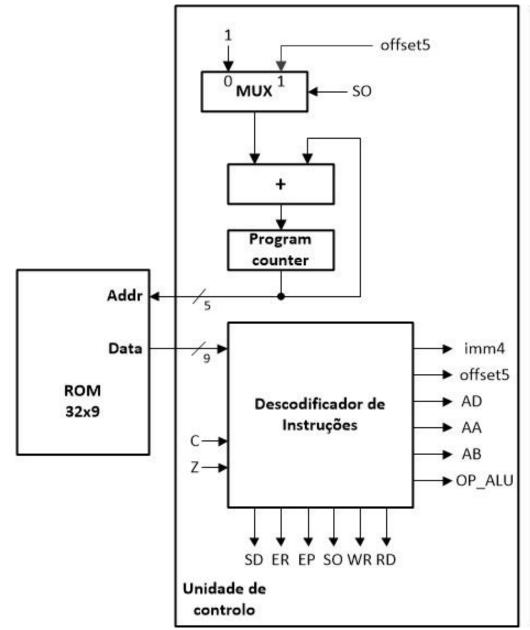
TÓPICOS DE REVISÃO

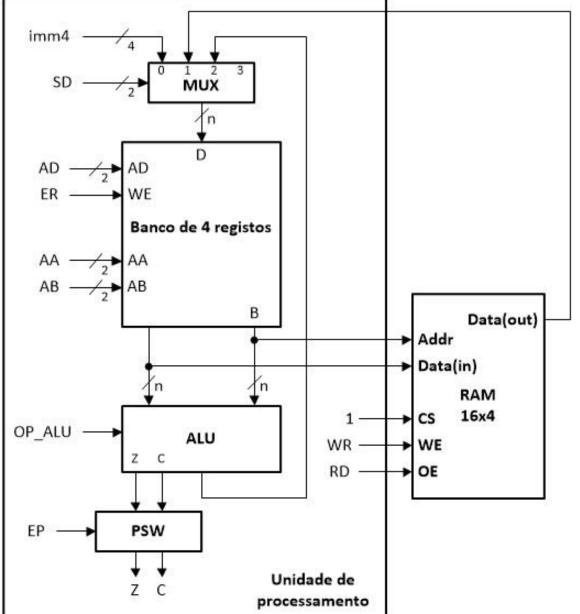
- Definição de conjunto de instruções
 Noção de mnemónica
 Operandos de uma instrução
- Tipos de instrução Transferência de dados Processamento de dados Controlo de fluxo de execução
- Definição / caracterização do ISA (Instruction Set Arquitecture)

Ano Lectivo 2019/2020

2º Semestre

Prof. Jorge Fonseca





Conjunto de instruções suportado pela arquitetura

Instrução			
mov rx, imm4	Carrega o valor imediato imm4 no registo rx.	rx = imm4	
Id rx, [ry] Transfere para o registo rx o conteúdo da posição de memória cujo endereço é definido pelo conteúdo de ry.		rx = mem[ry]	
st rx, [ry] Transfere o conteúdo do registo rx para a posição de memória cujo endereço está definido pelo conteúdo de ry.		mem[ry] = rx	
sub rx, ry, rz	Subtrai rz a ry e coloca o resultado em rx e atualizando o registo PSW	rx = ry - rz atualiza PSW	
add rx, ry, rz	Adiciona rz a ry e coloca o resultado em rx e atualizando o registo	rx = ry + rz atualiza PSW	
bae offset5	Quando a flag C apresenta o valor 0, muda a execução para o endereço resultante da adição ao PC do deslocamento offset5.	PC = (C == 0) ? PC + offset5 : PC + 1	
b offset5	Muda a execução para o endereço resultante da adição ao PC do deslocamento offset5.	PC = PC + offset5	
cmp rx, ry	Subtrai ry de rx e atualiza a flag C em conformidade com o resultado, que é descartado.	rx - ry	

ISA	0	PCOD	E	А	D	A	Α	AB		
ISA	8	7	6	5	4	3	2	1	0	
	Insti	uçõe	de T	ransfe	rênci	a		9	W.	
mov rx, imm4	0	0	0	r	x		im	m4		
ld rx, [ry]	0	0	1	r	X	28	848	ry		
st rx, [ry]	0	1	0	-	- rx				ry	
	Instruçõe	es de p	rocess	amen	to de o	dados			(8)	
sub rx, ry, rz	0	1	1	r	x	r	У	r	z	
add rx, ry, rz	1	0	0	r	x	ry		rz		
cmp rx, ry	1	1	1			r	x	ry		
	Instr	uções	de con	trolo	de flux	Ю			18	
bae offset5	1	0	1	3.5		(offset5	5		
b offset5	1	1	0	-2	13		offset5	5		

Numa instrução é preciso codificar:

- OPCODE: código único que distingue uma instrução de outra
- Operandos: pode ser operação da ALU, constante, Offset e/ou registos
- a codificação da operação da ALU pode pertencer ao OPCODE

Características:

- A constante para expressar um literal do tipo inteiro fica limitada a 4 bits, portanto ao valor 15.
- O offset5 é codificado com 5 bits e representa um inteiro com sinal ficando limitado a um salto baseado na instrução corrente para +15 ou -16 instruções.

Exercício1: Escrever em Código Máquina um programa para determinar o valor de A + B - C. Considere o operando A na posição de memória A0 operando A1 na posição de memória A1 na posição de memória A2 na posição de memória A4 na posição de memória A5 na posição de memória A6 na posição de memória A7 na posição de memória A8 na posição de memória A9 na posição de memória A

	Code Address	OP	со	DE	Α	D	Α	AA		В	HEX	
	Address	8	7	6	5	4	3	2	1	0		
MOV R1,0	00	0	0	0	0	1	0	0	0	0	010	
LD R0, [R1]	01	0	0	1	0	0	-	-	0	1	041	
MOV R1,1	02	0	0	0	0	1	0	0	0	1	011	
LD R2, [R1]	03	0	0	1	1	0	1	-	0	1	061	
ADD R2, R0, R2	04	1	0	0	1	0	0	0	1	0	122	
MOV R1,2	05	0	0	0	0	1	0	0	1	0	012	
LD R0, [R1]	06	0	0	1	0	0	ı	ı	0	1	041	
SUB R2, R2, R0	07	0	1	1	1	0	1	0	0	0	0E8	
MOV R1,3	08	0	0	0	0	1	0	0	1	1	013	
ST R2, [R1]	09	0	1	0	1	-	1	0	0	1	089	
B 0	10	1	1	0	-	0	0	0	0	0	180	

RAM							
ADDR							
0	Α						
1	В						
2	С						
3	R						

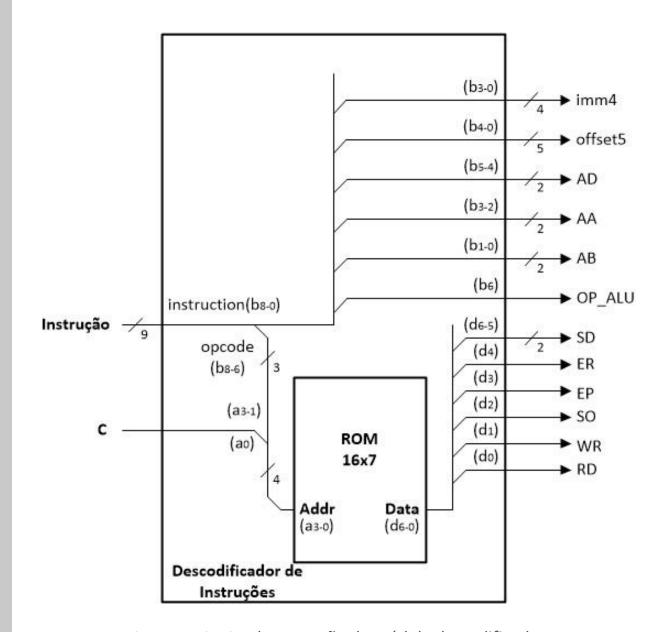


Imagem 1 – Implementação do módulo descodificador

Do ISA retira-se o controlo dos seguintes sinais:

- código da operação: B8..6

- operação da ALU: B6

- imm4: B3..0

- offset5: B4..0

- AA: B3..2

- AB: B1..0

- AD: B5..4

Estrutura interna do Core3

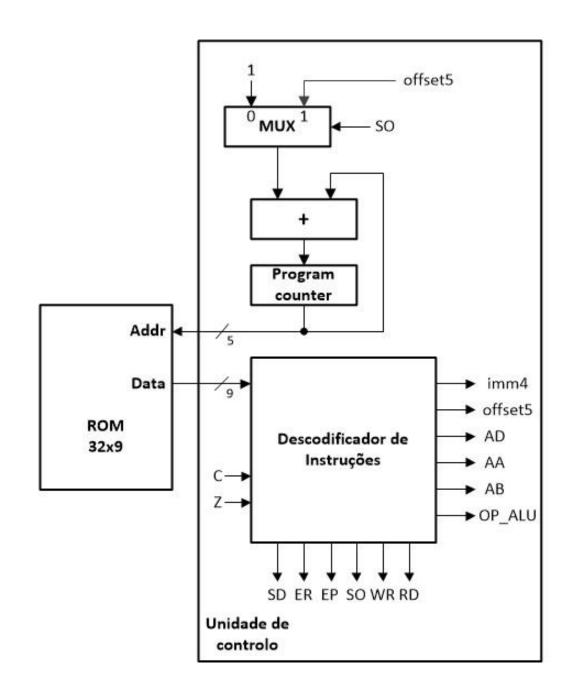
- Um programa é uma sequência de Códigos Máquina em memória para serem processados pelo CPU.
- Aspeto característico de arquitetura RISC: as instruções ocupam todas o mesmo espaço em memória.
- Processador de ciclo único → Arquitetura Harvard: separa a memória de código da memória de dados, ou seja, espaço de endereçamento distintos.

Instruction	C	PCOD	E	С	SD	10	ER	EP	SnA	SO	WR	RD	HEX	PRG
Decoder	3	2	1	0	7	6	5	4	3	2	1	0	ПЕХ	PNG
mov rx, immediate4	0	0	0	-	0	0	1	0	-	0	0	ı	020	2*020
ld rx, [ry]	0	0	1	-	0	1	1	0	-	0	0	1	061	2*061
st rx, [ry]	0	1	0	-	-	-	0	0	-	0	1	0	002	2*002
sub rx, ry, rz	0	1	1	-	1	0	1	1	1	0	0	ı	0B8	2*0B8
add rx, ry, rz	1	0	0	-	1	0	1	1	0	0	0	ı	0B0	2*0B0
bae offset5	1	0	1	0	-	-	0	0	-	1	0	ı	004	004
bae offset5	1	0	1	1	-	-	0	0	-	0	0	ı	000	000
b offset4	1	1	0	-	-	-	0	0	-	1	0	ı	004	2*004
cmp rx, ry	1	1	1	-	-	-	0	1	1	0	0	ı	018	2*018

Tabela 2 – Módulo descodificador

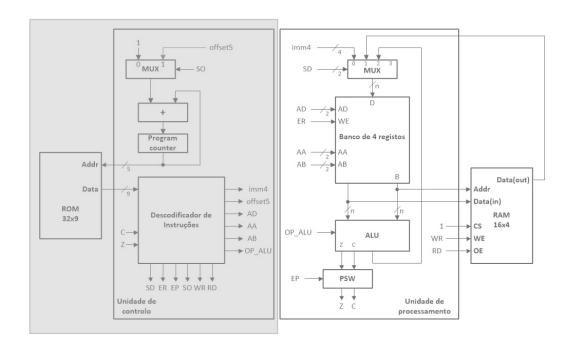
Descodificador de instruções:

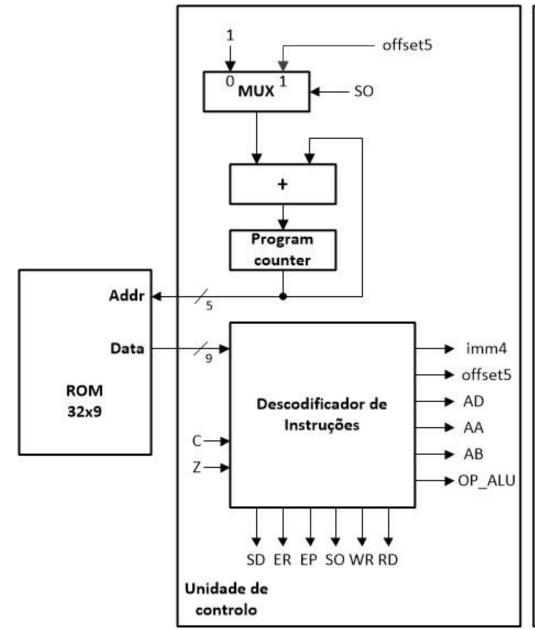
- Produz os sinais de controlo da unidade de processamento baseado no OPCODE e *flags C e Z* e controla a evolução do programa
- Micro-código implementado tipicamente numa ROM

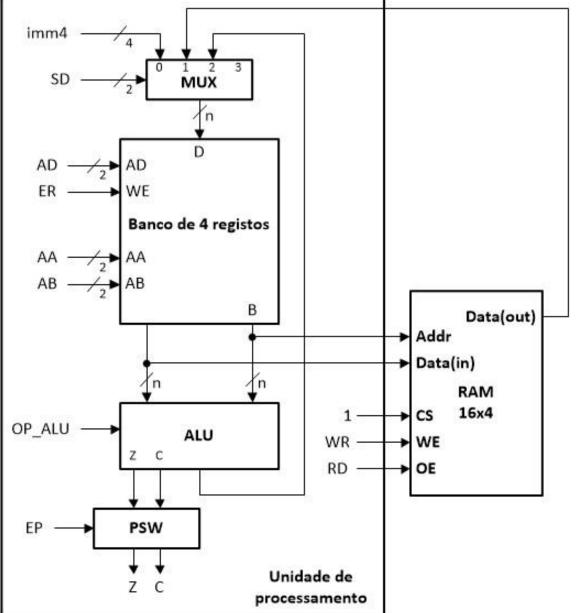


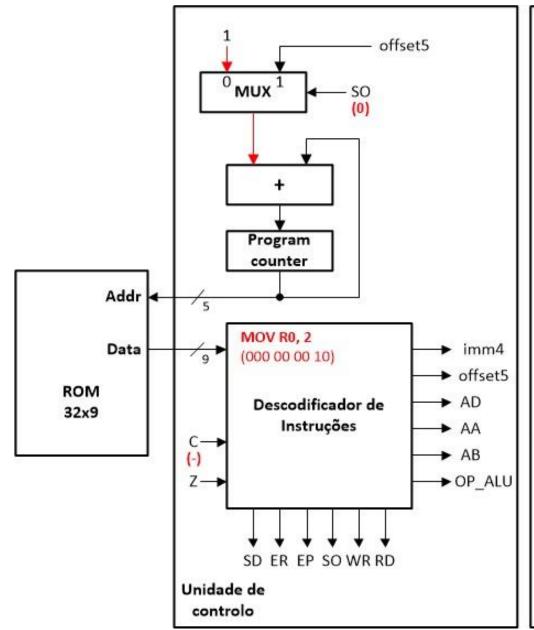
Aspectos relevantes:

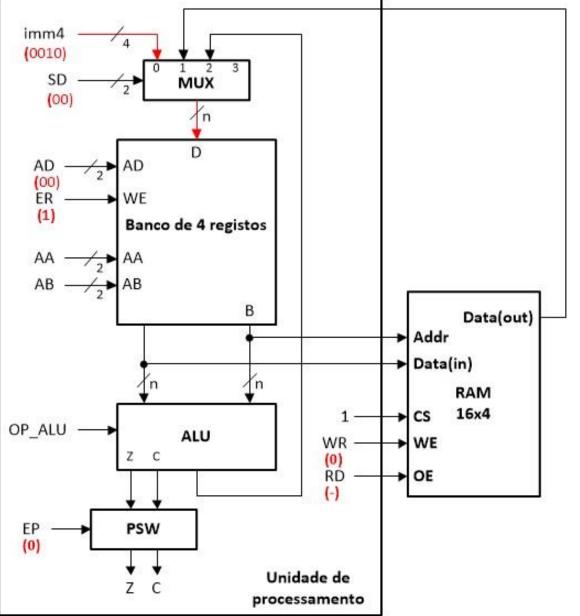
- Registo PC (Program Counter): retém a posição da instrução em execução.
- SO: controla o valor de incremento do PC.
- Clock da unidade de controlo em oposição de fase do clock da unidade de processamento.

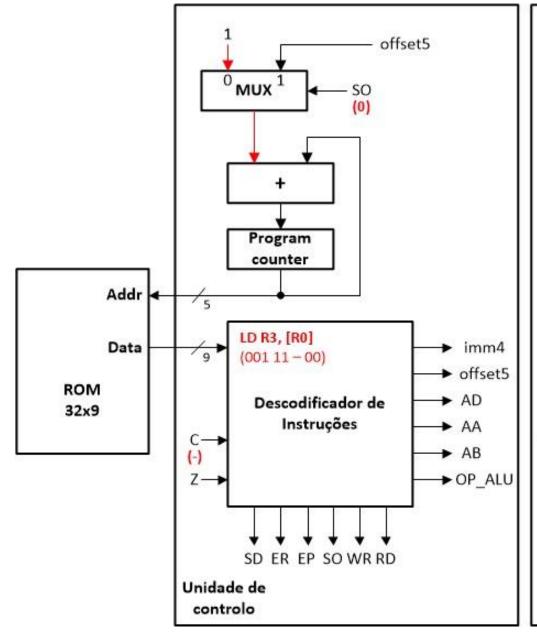


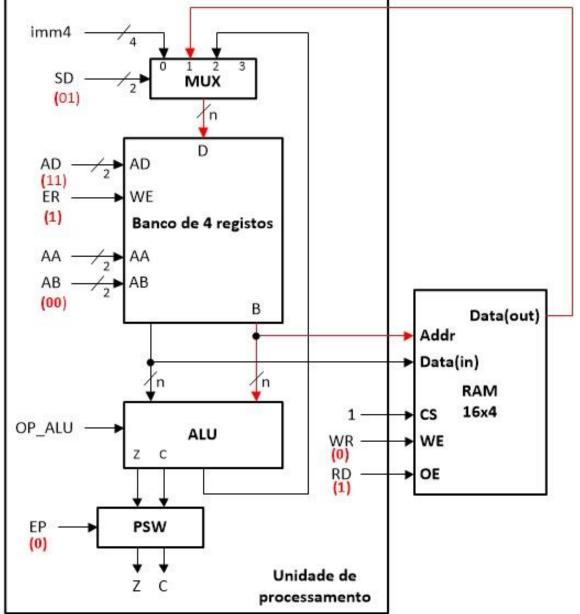


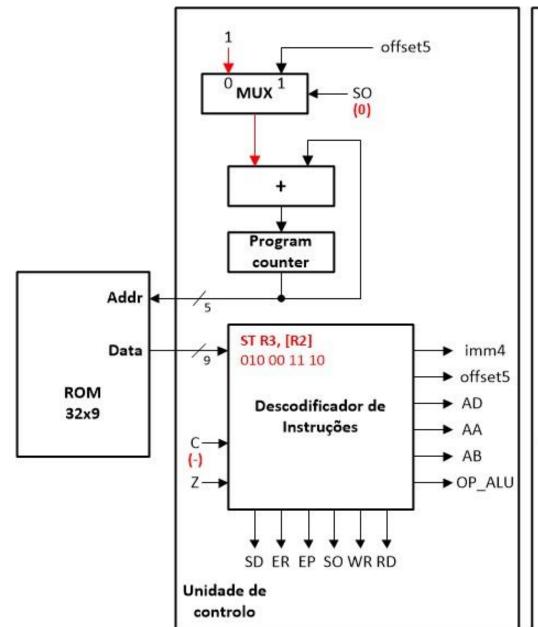


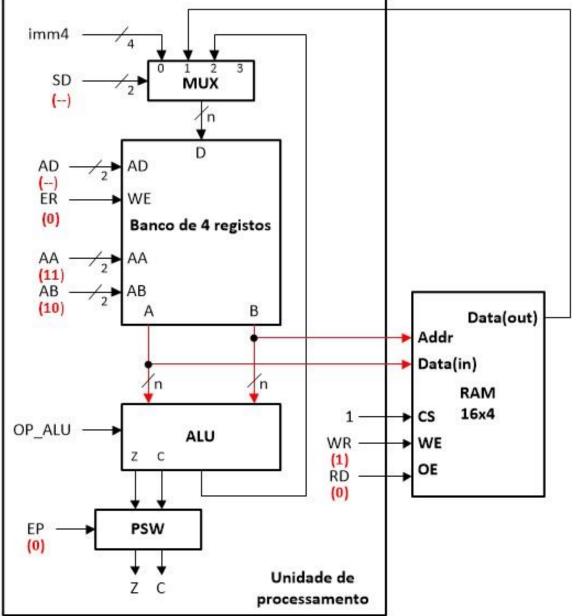


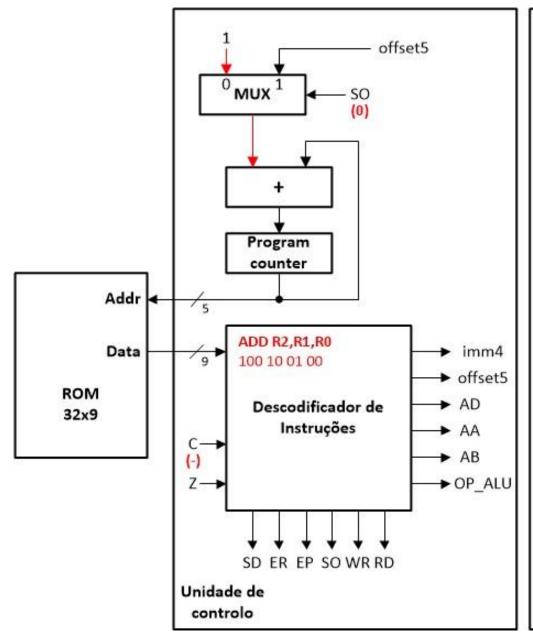


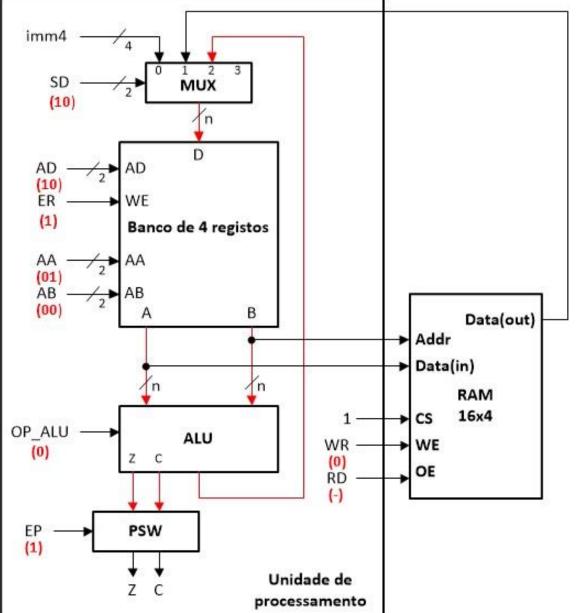


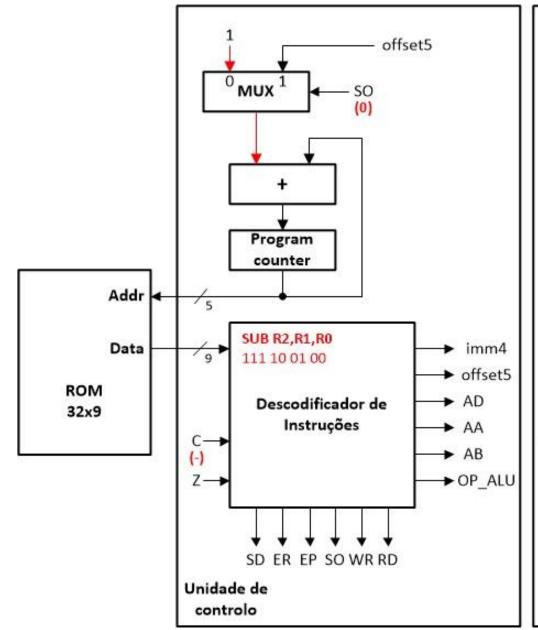


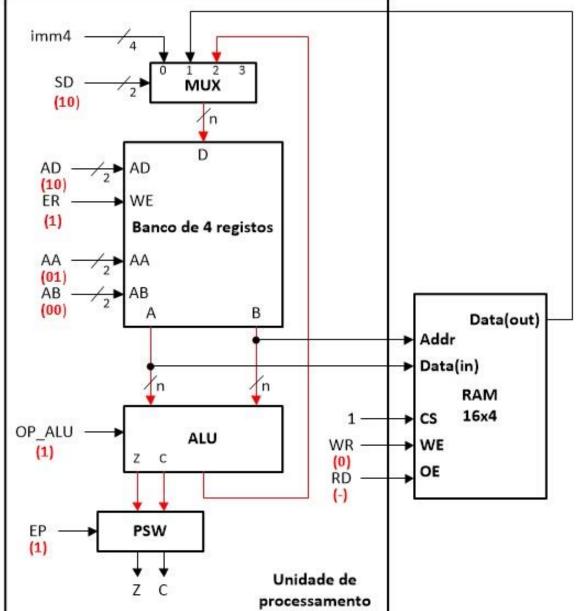


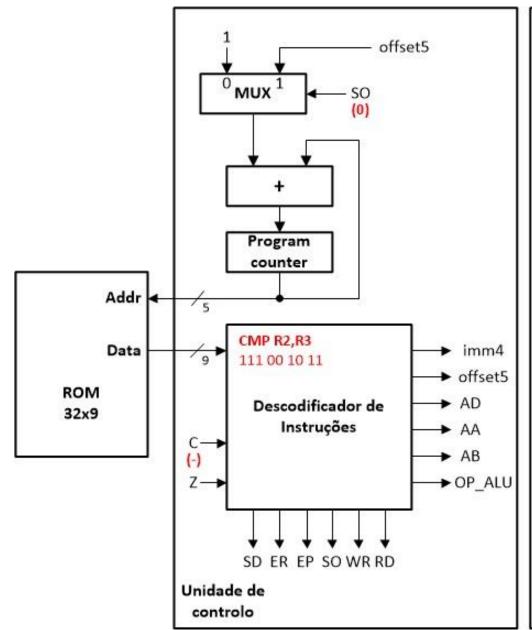


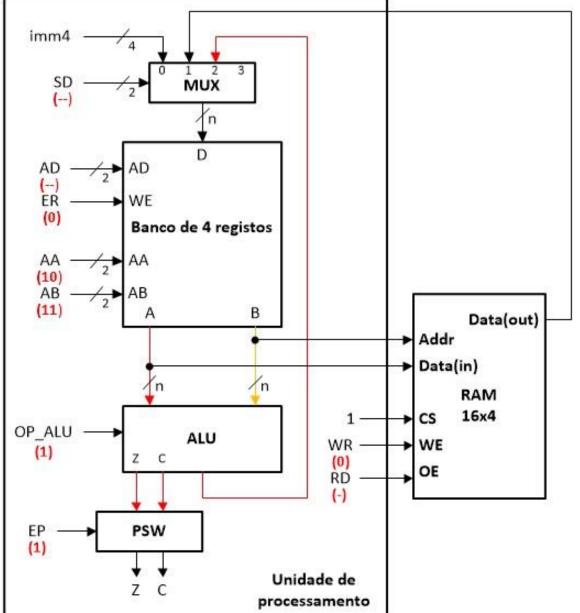


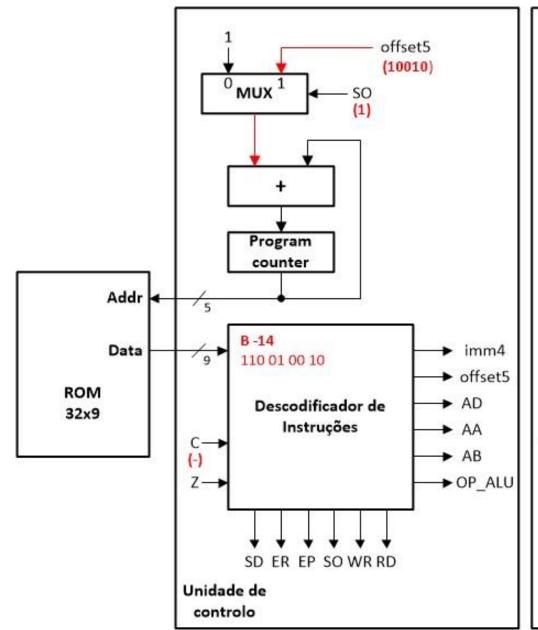


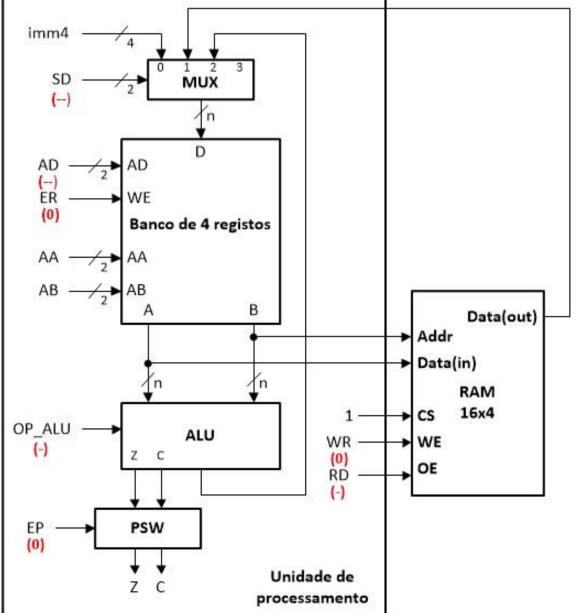


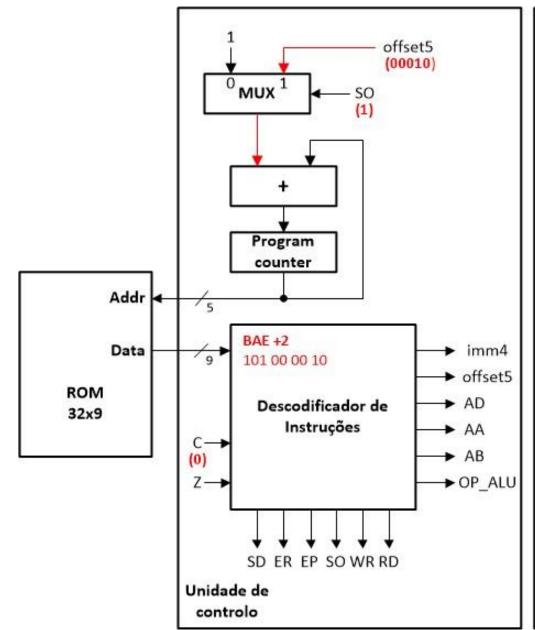


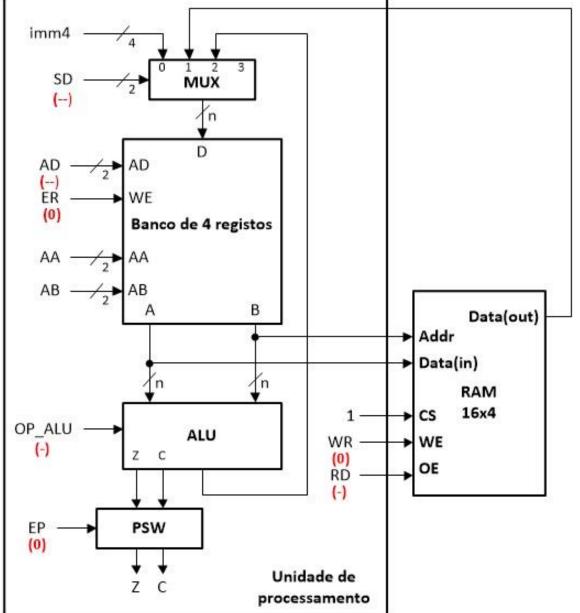


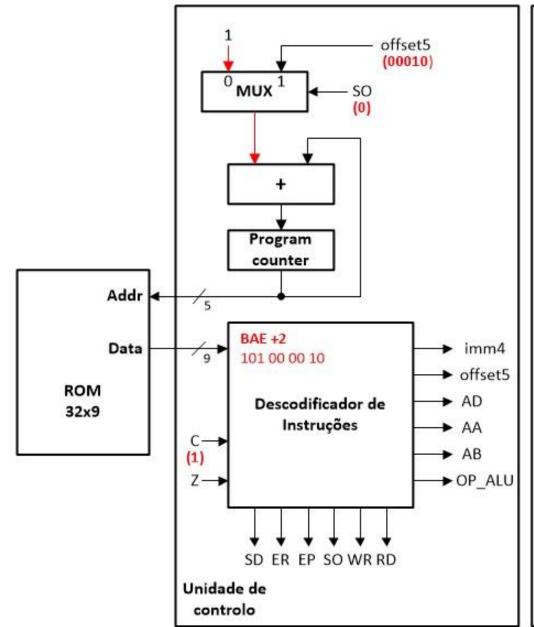


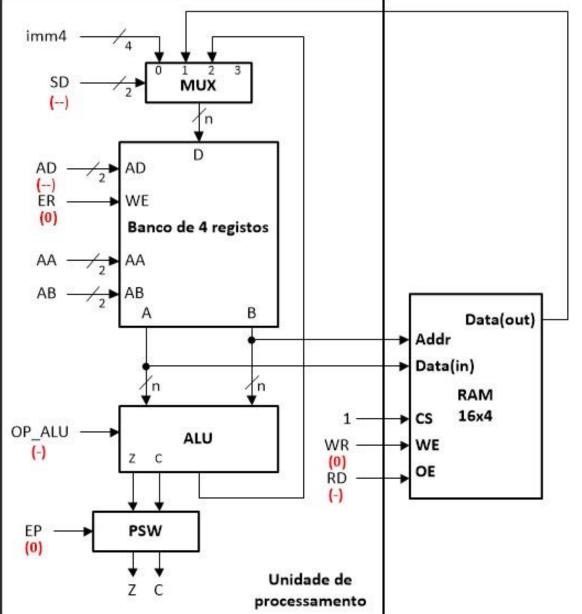












Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

	Code	OP	со	DE	Α	D	Α	Α	Α	В	PRG
	Address	8	7	6	5	4	3	2	1	0	(HEX)
MOV R1, 0A	00										000
LD R0, [R1]	01										000
MOV R1, 0B	02										000
LD R2, [R1]	03										000
CMP RO, R2	04										000
BAE +4	05										000
MOV R1, 0C	06										000
ST R2, [R1]	07										000
B +3	08										000
MOV R1, 0C	09										000
ST R0, [R1]	0A										000
B 0	OB										000

RAI	M
ADDR	
0A	X
OB	Υ
0C	M

Algoritmo
IF X >= Y {
M = X
} ELSE {
M = Y
}

ISA	OPCODE			Α	D	Α	Α	AB		
ISA	8	7	6	5	5 4		2	1	0	
mov rx, imm4	0	0	0	rx		imm4				
ld rx, [ry]	0	0	1	rx		1	1	r	у	
st rx, [ry]	0	1	0			rx		ry		
sub rx, ry, rz	0	1	1	r	X	ry		rz		
add rx, ry, rz	1	0	0	rx		r	у	rz		
bae offset5	1	0	1	1		offset5				
b offset5	1	1	0	-		offset5				
cmp rx, ry	1	1	1	-	_	rx		ry		

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

	Code	OF	co	DE	А	D	A	A	AB		PRG
	Address	8	7	6	5	4	3	2	1	0	(HEX)
MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
LD R0, [R1]	01					04.110/17			Marie		041
MOV R1, 0B	02										01B
LD R2, [R1]	03										061
CMP RO, R2	04										1C2
BAE +4	05										143
MOV R1, 0C	06										01C
ST R2, [R1]	07										089
B +3	08										183
MOV R1, OC	09	1									01C
ST R0, [R1]	0A	1									081
B 0	OB										180
											000
											000
	8			10 - 100 10 - 100		6 - 17 6 - 18		10 17 13 16		13 97 13 33	000
											000
		22 - 22 10 - 10		6 - 32 6 - 32		5 - 52 5 - 15		(5 - 32) (5 - 35)		6 - 32 6 - 35	000

X
Υ
M

Algoritmo	•
IF X >= Y {	
M = X	
} ELSE {	
M = Y	
}	

ISA	OI	PCO	DE	Α	D	Α	A	AB		
ISA	8	7	6	5	4	3	2	1	0	
mov rx, imm4	0	0	0	r	rx im		imn		m4	
ld rx, [ry]	0	0	1	r	х			ry		
st rx, [ry]	0	1	0	10	353	r	rx		ry	
sub rx, ry, rz	0	1	1	r	X	۲	у	rz		
add rx, ry, rz	1	0	0	r	X	r	у	rz		
bae offset5	1	0	1		6) - 1 S:	offset5				
b offset5	1	1	0	1.5		0	ffse	t5		
cmp rx, ry	1	1	1	-		r	x	r	у	

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

	Code	Code OPCODE AD		А	AA		В	PRG			
	Address	8	7	6	5	4	3	2	1	0	(HEX)
MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
LD R0, [R1]	01	0	0	1	0	0	- T	i z .	0	1	041
MOV R1, 0B	02	Г		783 - 300		76 - O.		98 - St			01B
LD R2, [R1]	03										061
CMP R0, R2	04									3	1C2
BAE +4	05										143
MOV R1, 0C	06									3	01C
ST R2, [R1]	07										089
B+3	08	1								30	183
MOV R1, OC	09	Ī									01C
ST RO, [R1]	0A	Ī								80	081
B 0	OB										180
	91.			8 38 8 33		6 36 6 36		6 36 6 36		8 83 8 33	000
											000
	81	10 70 12 73				4 17 4 16		9 77 9 78		6 93 8 38	000
											000
	34	15 - 37 6 - 15		6 - 37 6 - 39		6 - 97 6 - 19		6 - 10		6 - 32 6 - 39	000

ADDR	
0A	X
0B	Υ
0C	M

Algoritmo						
IF X >	>= Y {					
M:	= X					
} ELS	E {					
M:	= Y					
}						

ISA	O	PCO	DE	AD		AA		AB		
ISA	8	7	6	5	4	3	2	1	0	
mov rx, imm4	0	0	0	r	rx im		imn		m4	
ld rx, [ry]	0	0	1	r	Х			r	у	
st rx, [ry]	0	1	0	1.	353	rx		ry		
sub rx, ry, rz	0	1	1	r	X	r	у	rz		
add rx, ry, rz	1	0	0	r	х	r	у	rz		
bae offset5	1	0	1	!		offset5		t5		
b offset5	1	1	0	35		0	ffset	t5		
cmp rx, ry	1	1	1		-	r	X	r	у	

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

		Code	OF	co	DE	А	D	A	AA A		В	PRG
		Address	8	7	6	5	4	3	2	1	0	(HEX)
33 20	MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
	LD R0, [R1]	01	0	0	1	0	0	- T	17.	0	1	041
	MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B
	LD R2, [R1]	03		_								061
	CMP RO, R2	04									8	1C2
	BAE+4	05										143
	MOV R1, 0C	06									3	01C
	ST R2, [R1]	07										089
86	B+3	08									8	183
	MOV R1, 0C	09										01C
55	ST R0, [R1]	0A									30	081
	B 0	OB										180
35		84.			8 38 8 33		6 36 6 36		60 560 60 565		8 88 8 88	000
												000
35		8 t			6 - 55 6 - 55		6 17 8 18		9 33 9 33		0 - 30 0 - 33	000
												000
			200		(2		62 - 92 6 - 35		(2		(2	000

RAN	V1
ADDR	
0A	X
0B	Υ
0C	M

Algoritmo						
IF X >= Y {						
M = X						
} ELSE {						
M = Y						
}						

ISA	O	PCO	DE	Α	D	Α	AA		AB	
ISA	8	7	6	5	4	3	2	1	0	
mov rx, imm4	0	0	0	r	X	imm4				
ld rx, [ry]	0	0	1	r	X	4)ja	r	у	
st rx, [ry]	0	1	0	1.5	353	- rx		ry		
sub rx, ry, rz	0	1	1	r	X	ry		rz		
add rx, ry, rz	1	0	0	r	X	ry		rz		
bae offset5	1	0	1	ુ	offset5			t5		
b offset5	1	1	0			0	ffset	t5		
cmp rx, ry	1	1	1		-	rx		r	у	

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

		Code OPCODE AD		D	A	A	А	В	PRG			
		Address	8	7	6	5	4	3	2	1	0	(HEX)
32 23	MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
	LD R0, [R1]	01	0	0	1	0	0		12	0	1	041
	MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B
	LD R2, [R1]	03	0	0	1	1	0		12.	0	1	061
20	CMP R0, R2	04			/E - 31		V: 31		VII - 35			1C2
	BAE +4	05										143
6	MOV R1, 0C	06									0	01C
	ST R2, [R1]	07										089
	B +3	08									30	183
	MOV R1, 0C	09										01C
5	ST R0, [R1]	0A									30	081
	B 0	OB										180
5		8 L			6 55 6 55		6 55 5 55		8 33 8 33		6 55 6 55	000
												000
5		91			9 35 9 35		4 V		(4 - 32) (4 - 35)		9 35 9 35	000
												000
		3 4	12 77 10 10		(2 - 32) (5 - 15		(2 - 37 (c - 15		6 - 32 6 - 32		6 - 35 6 - 35	000

RAN	V1
ADDR	
0A	X
0B	Υ
0C	M

Algoritmo						
IF X	>= Y {					
M	= X					
} ELS	E {					
M	= Y					
}						

ISA	O	PCO	DE	AD		Α	A	AB	
ISA	8	7	6	5	5 4		2	1	0
mov rx, imm4	0	0	0	r	X	im		m4	
ld rx, [ry]	0	0	1	r	X			r	у
st rx, [ry]	0	1	0	e		r	х	ry	
sub rx, ry, rz	0	1	1	rx		ry		rz	
add rx, ry, rz	1	0	0	r	X	ry		rz	
bae offset5	1	0	1	ુ	- offset5				
b offset5	1	1	0			0	ffset	5	
cmp rx, ry	1	1	1		-	r	Х	r	у

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

	Code	OF	co	DE	Α	D	AA		AB		PRG	
	Address	8	7	6	5	4	3	2	1	0	(HEX)	
MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A	
LD R0, [R1]	01	0	0	1	0	0		=	0	1	041	
MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B	
LD R2, [R1]	03	0	0	1	1	0	87)	Œ.	0	1	061	
CMP RO, R2	04	1	1	1	2	-	0	0	1	0	1C2	
BAE +4	05										143	
MOV R1, 0C	06									0	010	
ST R2, [R1]	07										089	
B+3	08									30	183	
MOV R1, 0C	09										010	
ST R0, [R1]	0A									30	081	
B 0	OB										180	
	81.	16 36 16 35		8 88 8 88		6 56 6 56		6 56 8 36		8 (8) 8 (8)	000	
											000	
	91.			9 93 9 93		14 - 17 13 - 15		9 33 9 36		0 - 33 9 - 33	000	
											000	
10		12 - 77 10 - 12		6 - 32 6 - 32		(2 - 1) (0 - 1)		(2 – 37) (0 – 32)		5 - 35 5 - 19	000	

OB	ADDR	
18 3	0A	X
00	OB	Υ
UC	0C	M
I	15	

Algoritmo					
IF X >= Y {					
M = X					
} ELSE {					
M = Y					
}					

ISA	O	PCO	DE	AD		AA		Α	В
ISA	8	7	6	5	5 4		2	1	0
mov rx, imm4	0	0	0	r	X	im		m4	
ld rx, [ry]	0	0	1	r	X	1	-	ry	
st rx, [ry]	0	1	0	5.5	3 5 353		X	ry	
sub rx, ry, rz	0	1	1	rx		ry		rz	
add rx, ry, rz	1	0	0	r	X	r	у	rz	
bae offset5	1	0	1		- offset5				
b offset5	1	1	0	-		0	ffset	t5	
cmp rx, ry	1	1	1	2	-	r	X	r	у

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

		Code	Code OPCODE		А	D	A	AA		В	PRG	
		Address	8	7	6	5	4	3	2	1	0	(HEX)
22 23	MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
	LD R0, [R1]	01	0	0	1	0	0		=	0	1	041
(3) (3)	MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B
	LD R2, [R1]	03	0	0	1	1	0	37	i.e.	0	1	061
(3) (3)	CMP R0, R2	04	1	1	1	2	-	0	0	1	0	1C2
	BAE +4	05	1	0	1	270	0	0	0	1	1	143
(2) (3)	MOV R1, OC	06		_	12 00		10 00		10 00			01C
	ST R2, [R1]	07										089
88	B +3	08									8	183
	MOV R1, OC	09										01C
65	ST R0, [R1]	0A									30	081
	B 0	OB										180
50 50	3		12 SE		6 50 6 55		6 36 8 8		6 50 6 55			000
												000
88	3	8 t	S 37		0 00 0 00		0 - 00 0 - 00		16 - 17 18 - 18		9 - 33 9 - 33	000
												000
		3 8			(2 - 72) () ((2 - 27 (1)		(2 - 22) (1 - 1)		2 - 92 2 - 19	000

ADDR	
0A	X
OB	Υ
0C	M
:5	
13	

	Algoritmo					
IF X >	·= Υ {					
M =	= X					
} ELSE	Ε{					
M =	= Y					
}						

ISA	O	PCO	DE	AD		Α	Α	Α	В
ISA	8	7	6	5	5 4		2	1	0
mov rx, imm4	0	0	0	r	X	im		m4	
ld rx, [ry]	0	0	1	r	X)ja	ry	
st rx, [ry]	0	1	0	(E. 1986)		rx		ry	
sub rx, ry, rz	0	1	1	rx		ry		rz	
add rx, ry, rz	1	0	0	rx		ry		rz	
bae offset5	1	0	1	ુ	- offset5				37
b offset5	1	1	0	÷-		0	ffset	t5	
cmp rx, ry	1	1	1		-	r	rx		у

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

		Code	Code OPCODE		DE	А	D	AA		A AB		PRG	
		Address	8	7	6	5	4	3	2	1	0	(HEX)	
20	MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A	
	LD R0, [R1]	01	0	0	1	0	0		æ	0	1	041	
6	MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B	
	LD R2, [R1]	03	0	0	1	1	0	ST	13.	0	1	061	
6	CMP R0, R2	04	1	1	1	2	-	0	0	1	0	1C2	
	BAE +4	05	1	0	1	. T.	0	0	0	1	1	143	
0	MOV R1, 0C	06	0	0	0	0	1	1	1	0	0	01C	
	ST R2, [R1]	07										089	
8	B+3	08									30	183	
	MOV R1, OC	09										01C	
5	ST R0, [R1]	0A									33	081	
	B 0	OB										180	
5		81			6 56 6 35		6 55 5 55		66 56 6 55		8 88 8 88	000	
												000	
55	3	81.			10 - 10 13 - 15		12 - 17 13 - 18		,		(4 - 77) (4 - 73)	000	
												000	
			12 77		(2 - 37) (5 - 32)		12 - 32 10 - 15		2 - 27 2 - 12		6 - 92 6 - 19	000	

ADDR	
0A	X
0B	Υ
0C	M
13	

Algoritmo					
IF X >= Y {					
M = X					
} ELSE {					
M = Y					
}					

ISA	O	PCO	DE	AD) AA		AB		
ISA	8	8 7 6 5 4		5 4 3 2		3 2		0		
mov rx, imm4	0	0	0	rx			im	m4		
ld rx, [ry]	0	0	1	rx		4	4	r	у	
st rx, [ry]	0	1	0	35 353		rx		ry		
sub rx, ry, rz	0	1	1	r	X	ry		r	Z	
add rx, ry, rz	1	0	0	r	rx		у	rz		
bae offset5	1	0	1	ું	-		ffset	t5		
b offset5	1	1	0	1-		0	offset		t5	
cmp rx, ry	1	1	1			rx		r	у	

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

		Code	OF	со	DE	A	D	A	A	А	В	PRG
		Address	8	7	6	5	4	3	2	1	0	(HEX)
(3) (3)	MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
	LD R0, [R1]	01	0	0	1	0	0		=	0	1	041
83 83	MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B
	LD R2, [R1]	03	0	0	1	1	0	37		0	1	061
	CMP R0, R2	04	1	1	1	2	-	0	0	1	0	1C2
	BAE +4	05	1	0	1	· •	0	0	0	1	1	143
	MOV R1, OC	06	0	0	0	0	1	1	1	0	0	01C
	ST R2, [R1]	07	0	1	0	<u>(35</u> ()	i i	1	0	0	1	089
	B +3	08					-					183
	MOV R1, 0C	09										01C
55	ST R0, [R1]	0A									30	081
	B 0	OB										180
55					6 56 6 65		6 55 6 55		6 55 6 35		6 K	000
												000
35	S.	8 t	S 37		0 - 00 0 - 00		10 - 10 13 - 16		10 - 10 13 - 16		0 - 10 9 - 16	000
												000
		3 4	10 TO		0 - 17 0 - 19		(2 - 17) (0 - 15)		(5 – 37) (6 – 35)		0 - 25 0 - 19	000

RAN	••
ADDR	
0A	X
OB	Υ
0C	M
- 8	
-	

Algoritmo					
IF X >= Y {					
M = X					
} ELSE {					
M = Y					
}					
<u></u>					

ICA	ISA OPCODE AD		AA		AB						
ISA	8	7	6 5 4		3 2		1	0			
mov rx, imm4	0	0	0	rx			im	m4			
ld rx, [ry]	0	0	1	rx		1	-	ry			
st rx, [ry]	0	1	0	10	353	rx		ry			
sub rx, ry, rz	0	1	1	r	X	ry		rz			
add rx, ry, rz	1	0	0	r	X	ry		ry		r	Z
bae offset5	1	0	1		offse		ffset	t5			
b offset5	1	1	0	1-		off		t5			
cmp rx, ry	1	1	1			- rx		r	у		

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

	Code	Code OPCODE		AD		AA		AB		PRG	
	Address	8	7	6	5	4	3	2	1	0	(HEX)
MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
LD R0, [R1]	01	0	0	1	0	0		=	0	1	041
MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B
LD R2, [R1]	03	0	0	1	1	0	3		0	1	061
CMP RO, R2	04	1	1	1	2	-	0	0	1	0	1C2
BAE +4	05	1	0	1	· •	0	0	0	1	1	143
MOV R1, OC	06	0	0	0	0	1	1	1	0	0	01C
ST R2, [R1]	07	0	1	0	<u>(35</u> ()	: E	1	0	0	1	089
B+3	08	1	1	0	747	0	0	0	1	1	183
MOV R1, 0C	09							200			01C
ST R0, [R1]	0A									33	081
B 0	OB										180
	8			66 56 6 65		6 50 9 55		6 50 6 55		8 K	000
											000
S	8			0 00 0 00		0 - 00 0 - 00		16 - 17 18 - 18		9 35 9 35	000
											000
				(A - 37) (C - 15)		(2 - 32) (4 - 15		0 - 37 0 - 35		6 - 35 6 - 35	000

V
X
Υ
M

Algoritmo	•
IF X >= Y {	
M = X	
} ELSE {	
M = Y	
}	

ISA	OF	CO	DE	AD		AD AA		AB			
ISA	8	7	6	5	5 4		2	1	0		
mov rx, imm4	0	0	0	rx		rx			im	m4	
ld rx, [ry]	0	0	1	rx				ry			
st rx, [ry]	0	1	0	35 358		rx		ry			
sub rx, ry, rz	0	1	1	r	X	ry		rz			
add rx, ry, rz	1	0	0	rx		ry		rz			
bae offset5	1	0	1	- offset		5					
b offset5	1	1	0	1-		offse		t5			
cmp rx, ry	1	1	1		-	rx		r	у		

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

		Code	Code OPCODE		AD		AA		AB		PRG	
		Address	8	7	6	5	4	3	2	1	0	(HEX)
92	MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
	LD R0, [R1]	01	0	0	1	0	0	200		0	1	041
22	MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B
	LD R2, [R1]	03	0	0	1	1	0	370	=	0	1	061
(2) (3)	CMP RO, R2	04	1	1	1	2	-	0	0	1	0	1C2
	BAE +4	05	1	0	1	ंग	0	0	0	1	1	143
82 85	MOV R1, 0C	06	0	0	0	0	1	1	1	0	0	01C
	ST R2, [R1]	07	0	1	0	(30)	7	1	0	0	1	089
50 55	B+3	08	1	1	0	14	0	0	0	1	1	183
	MOV R1, OC	09	0	0	0	0	1	1	1	0	0	01C
33	ST R0, [R1]	0A		_	(1-37)		W		VI 37			081
	B 0	OB										180
33		91.	16 SA 16 35		6 36 6 33		6 56 6 (6		6 56 6 6			000
												000
50 35		91.	15 77 16 35		2 32 2 33		6 - 17 6 - 16		6 - 17 6 - 16		14 92 14 35	000
												000
			6 - 32 6 - 32		5 - 32 5 - 32		62 - 32 61 - 15		62 - 92 61 - 15		6 92 6 19	000

RAN	1
ADDR	
0A	X
OB	Υ
0C	M

ISA	O	co	DE	AD		D AA		AE	
IJA	8	7	6	5 4		3	2	1	0
mov rx, imm4	0	0	0	rx			im		
ld rx, [ry]	0	0	1	rx		1		ry	
st rx, [ry]	0	1	0	. E		rx		ry	
sub rx, ry, rz	0	1	1	rx		ry		rz	
add rx, ry, rz	1	0	0	r	X	r	у	rz	
bae offset5	1	0	1	-			offset5		
b offset5	1	1	0			offse		t5	
cmp rx, ry	1	1	1			rx		r	у

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

		Code	OF	со	DE	А	D	A	A	А	В	PRG
		Address	8	7	6	5	4	3	2	1	0	(HEX)
32 20	MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
	LD R0, [R1]	01	0	0	1	0	0	200		0	1	041
	MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B
	LD R2, [R1]	03	0	0	1	1	0	370	=	0	1	061
82 85	CMP R0, R2	04	1	1	1	23	-	0	0	1	0	1C2
	BAE+4	05	1	0	1	- TO	0	0	0	1	1	143
82 60	MOV R1, OC	06	0	0	0	0	1	1	1	0	0	01C
	ST R2, [R1]	07	0	1	0	73 50	7	1	0	0	1	089
60	B +3	08	1	1	0	~ <u></u>	0	0	0	1	1	183
	MOV R1, OC	09	0	0	0	0	1	1	1	0	0	01C
33	ST R0, [R1]	0A	0	1	0	74.0	.=	0	0	0	1	081
	B 0	OB					3 3		2 3			180
55		81.	16 SA 16 35		6 36 6 33		6 56 6 (6		6 56 6 6		8 83 8 83	000
												000
60 60	100	8	9 33 9 35		9 35 9 35		9 33 9 36		9 33 9 33		9 35	000
												000
			(2 - 32) (2 - 12		5 - 32 5 - 32		6 - 32 6 - 15		10 - 20 10 - 15		6 - 32 6 - 19	000

ADDR	
0A	X
0B	Υ
0C	M

Algoritm	0
IF X >= Y {	
M = X	
} ELSE {	
M = Y	
}	

ISA	O	PCO	DE	AD		AD AA		Α	AΒ		
ISA	8	7	6	5 4		3	2	1	0		
mov rx, imm4	0	0	0	rx		rx			im	imm4	
ld rx, [ry]	0	0	1	r	rx)ji	ry			
st rx, [ry]	0	1	0	35 353		rx		ry			
sub rx, ry, rz	0	1	1	rx		ry		rz			
add rx, ry, rz	1	0	0	r	X	r	у	rz			
bae offset5	1	0	1			offset5					
b offset5	1	1	0	1-		offse		t5			
cmp rx, ry	1	1	1			rx		r	У		

Exercício2 : Escrever em Código Máquina um programa para determinar o maior valor contido nas posições de memória de endereço OAH e OBH.

	Code	OPCODE		А	AD		AA		В	PRG	
	Address	8	7	6	5	4	3	2	1	0	(HEX)
MOV R1, 0A	00	0	0	0	0	1	1	0	1	0	01A
LD R0, [R1]	01	0	0	1	0	0		13.	0	1	041
MOV R1, 0B	02	0	0	0	0	1	1	0	1	1	01B
LD R2, [R1]	03	0	0	1	1	0	S	13.	0	1	061
CMP R0, R2	04	1	1	1	-	-	0	0	1	0	1C2
BAE +4	05	1	0	1	870	0	0	0	1	1	143
MOV R1, OC	06	0	0	0	0	1	1	1	0	0	01C
ST R2, [R1]	07	0	1	0	73 78	10	1	0	0	1	089
B+3	08	1	1	0	14	0	0	0	1	1	183
MOV R1, OC	09	0	0	0	0	1	1	1	0	0	01C
ST RO, [R1]	0A	0	1	0	947	-	0	0	0	1	081
B 0	OB	1	1	0		0	0	0	0	0	180
8	81.			90 300 90 300		(2 - 32) (2 - 33)				6 97. 8 38	000
											000
	81.							6 - 55 9 - 55		12 - 37. 13 - 33.	000
											000
	37			12 - 22 6 - 15				64 - 97 4		6 - 15	000

ADDR	
ADDR	- 353
0A	X
OB	Υ
0C	M
- 13	
-	

Algoritm	10
IF X >= Y {	
M = X	
} ELSE {	
M = Y	
}	

ISA	O	PCO	DE	Α	AD AA		AB		
IJA	8	7	6	5 4		3 2		1	0
mov rx, imm4	0	0	0	rx			im	m4	
ld rx, [ry]	0	0	1	rx				ry	
st rx, [ry]	0	1	0	SE 358		rx		ry	
sub rx, ry, rz	0	1	1	rx		ry		rz	
add rx, ry, rz	1	0	0	r	X	r	у	rz	
bae offset5	1	0	1	- offset5					- 3
b offset5	1	1	0	35		offse		t5	
cmp rx, ry	1	1	1			rx		r	у

Exercício3: Escrever em Código Máquina um programa para determinar o maior valor contido no array x.

		Code	OP	со	DE	Α	D	Α	A	AB PR		PRG
		Address	8	7	6	5	4	3	2	1	0	(HEX)
			\$ 52 3 52		8 38 3 33		8 39 3 33		8 - 58 8 - 58		8 39 3 39	000
												000
			8 92 8 99		8 98 8 99		8 92 8 93		8 99 8 99		8 39 8 39	000
												000
			8 99 8 99		8 98 3 33		8 99 3 99		6 99 3 33		8 39 8 39	000
												000
(3) (a)			8 99		8 88 3 33		\$8 33 10 33		8 18 8 11		8 38 3 33	000
												000
82			8 98 8 99		8 (6) 3 (3)		88 33 10 34		8 18 3 11		8 - 18 3 - 33	000
												000
	9. 9.		20 - 200 21 - 325		9 93 9 33		6 - 77. 6 - 75.		6 - 72 6 - 68		5 - 55 6 - 55	000
												000
86	2. G		9 33 9 33		9 33 9 35		6 33 8 33		9 - 33 9 - 33		9 - 33 9 - 33	000
												000
55	5. Si		96 97. 98 88		9 93 9 35		9 93 9 33		9 77 9 8		9 - 22 9 - 33	000
												000
88	5. S		S 35		0 77. 0 75		8 98 8 9		9 93 9 33		9 77 9 88	000
												000
86	5. Si		96 - 325 96 - 365		0 77 9 85		0 33 9 33		6 - 22 8 - 33		9 - 22 9 - 33	000
												000
CV Els			, s = 32; g = 15		9 - 92 6 - 15		9 - 92 6 - 15		9 - 92 6 - 18		9—19 6—19	000

RA	M
ADDR	
0	i
1	maior
2	x[0]
3	x[1]
4	x[2]
5	x[3]
6	x[4]
7	x[5]

Algoritmo
nibble i, maior;
nibble x[6]; /* inteiros sem sinal */
maior=x[0];
for (i=1; i < 6; i++) {
if (x[i] > maior) maior = x[i];
}

ISA	OPCODE			AD		AA		AB	
	8	7	6	5	4	3	2	1	0
mov rx, imm4	0	0	0	r	X	imm4			
ld rx, [ry]	0	0	1	rx		1	3	ry	
st rx, [ry]	0	1	0		378	rx		ry	
sub rx, ry, rz	0	1	1	rx		ry		rz	
add rx, ry, rz	1	0	0	rx		ry		rz	
bae offset5	1	0	1	- offset5					
b offset5	1	1	0	-	- offset5				
cmp rx, ry	1	1	1	-	1	r	Х	ry	