Arquitetura de Computadores

Aula 4 – Codificação de instruções. Descodificador de instruções

1. Instruções assembly.

Operação		Instrução assembly	Instrução exemplo
Load	rz = M[rx]	ld rz, M[rx]	R1 = M[R0]
Store	M[rx] = ry	st ry, M[rx]	M[R0] = R1
Soma	rz = ry + rx	add rz, ry, rx	R3 = R3 + R1
Subtração	rz = ry - rx	sub rz, ry, rx	R3 = R3 - R1
Incremento	rz = ry + 1	inc rz, rx	R3 = R3 + 1
Decremento	rz = ry - 1	dec rz, rx	R3 = R3 – 1
Soma lógica	rz = ry or rx	or rz, ry, rx	R3 = R3 or R2
Multiplicação lógica	rz = ry and rx	and rz, ry, rx	R3 = R3 and R2
Salto condicional C		jmpC addr	C = 1 ? PC = addr: PC = PC + 1
Salto condicional Z		jmpZ addr	Z = 1 ? PC = addr: PC = PC + 1

2. Codificação da instrução

Instrução assembly	opcode				operandos					
	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
load rz, M[rx]	1	0	0	1	rz		-		rx	
st ry, M[rx]	1	0	1	0	-		ry		rx	
add rz, ry, rx	0	0	0	1	rz		ry		rx	
sub rz, ry, rx	0	0	1	0	rz		ry		rx	
inc rz, rx	0	0	1	1	rz		-		rx	
dec rz, rx	0	1	0	0	rz		-		rx	
or rz, ry, rx	0	1	1	0	rz		ry		rx	
and rz, ry, rx	0	1	1	1	rz		ry		rx	
jmpC addr	1	0	0	0	addr					
jmpZ addr	0	0	0	0	addr					
		OP_ALU			AD		AB		AA	

Exemplo:

Operação	Instrução assembly	Instrução codificada
R2 = M[R1]	ld r2, M[r1]	1001 10 -- 01
M[R2] = R3	st r3, M[r2]	1010 -- 11 10
R3 = R3 + R1	add r3, r3, r1	0001 11 11 01
R3 = R2 – R0	sub r3, r2, r0	0010 11 10 00
R1 = R0 + 1	inc r1, r0	0011 01 -- 00
R2 = R2 – 1	dec r2, r2	0100 10 -- 10
R0 = R1 or R2	or r0, r1, r2	0110 00 01 10
R1 = R2 and R0	and r1, r2, r0	0111 01 10 00
C = 1 ? PC = 5: PC = PC + 1	jmpC 5	1000 000101
Z = 1 ? PC = 3: PC = PC + 1	jmpZ 3	0000 000011

3. Descodificação da instrução

   Circuito combinatório:

   Entradas: instrução (opcode, rx, ry, rx, addr)

   Saídas: palavra de controlo

opcode	Z	C	selA	selB	selD	WE	opALU	SD	WR	RD	EP	SF	addr
0000	0	-	--	--	--	0	---	-	0	0	0	0	------
0000	1	-	--	--	--	0	---	-	0	0	0	1	b_{5-0}
0001	-	-	b_{1-0}	b_{3-2}	b_{5-4}	1	b_{8-6}	1	0	0	1	0	------
0010	-	-	b_{1-0}	b_{3-2}	b_{5-4}	1	b_{8-6}	1	0	0	1	0	------
0011	-	-	b_{1-0}	--	b_{5-4}	1	b_{8-6}	1	0	0	1	0	------
0100	-	-	b_{1-0}	--	b_{5-4}	1	b_{8-6}	1	0	0	1	0	------
0101	-	-	--	--	--	-	---	-	-	-	-	-	------
0110	-	-	b_{1-0}	b_{3-2}	b_{5-4}	1	b_{8-6}	1	0	0	1	0	------
0111	-	-	b_{1-0}	b_{3-2}	b_{5-4}	1	b_{8-6}	1	0	0	1	0	------
1000	-	0	--	--	--	0	---	-	0	0	0	0	------
1000	-	1	--	--	--	0	---	-	0	0	0	1	b_{5-0}
1001	-	-	b_{1-0}	--	b_{5-4}	1	---	0	0	1	0	0	------
1010	-	-	b_{1-0}	b_{3-2}	--	0	---	-	1	0	0	0	------
1011	-	-	--	--	--	-	---	-	-	-	-	-	------
1100	-	-	--	--	--	-	---	-	-	-	-	-	------
1101	-	-	--	--	--	-	---	-	-	-	-	-	------
1110	-	-	--	--	--	-	---	-	-	-	-	-	------
1111	-	-	--	--	--	-	---	-	-	-	-	-	------

4. Implementação do descodificador de instruções

SelA = b_{1-0} (rx), SelB = b_{3-2} (ry), SelD = b_{5-4} (rz), opALU = b_{8-6}, addr = b_{5-0}

opcode	Z	C	WE	SD	WR	RD	EP	SF	Conteúdo ROM
0000	0	-	0	-	0	0	0	0	0x00
0000	1	-	0	-	0	0	0	1	0x01
0001	-	-	1	1	0	0	1	0	0x32
0010	-	-	1	1	0	0	1	0	0x32
0011	-	-	1	1	0	0	1	0	0x32
0100	-	-	1	1	0	0	1	0	0x32
0101	-	-	-	-	-	-	-	-	0x00
0110	-	-	1	1	0	0	1	0	0x32
0111	-	-	1	1	0	0	1	0	0x32
1000	-	0	0	-	0	0	0	0	0x00
1000	-	1	0	-	0	0	0	1	0x01
1001	-	-	1	0	0	1	0	0	0x24
1010	-	-	0	-	1	0	0	0	0x08
1011	-	-	-	-	-	-	-	-	0x00
1100	-	-	-	-	-	-	-	-	0x00
1101	-	-	-	-	-	-	-	-	0x00
1110	-	-	-	-	-	-	-	-	0x00
1111	-	-	-	-	-	-	-	-	0x00