Universidade Federal do Rio Grande do Norte

ELE1717 - SISTEMAS DIGITAIS

Datasheet do Processador

Projeto e Documentação de um processador de 70 instruções

Discentes:

Ana Karoline Pontes Machado
Camila Barbosa Gomes de Araújo
Gabriel Teixeira Vantuil

Docente:
Samaherni Morais Dias

17 de abril de 2018

Esse documento descreve e documenta o projeto de um processador de 8 bits e 69 instruções para a disciplina de Sistemas Digitais (ELE1717). O processador foi implementado em VHDL, simulado com o auxílio do Quartus e também executado na FPGA Cyclone II.

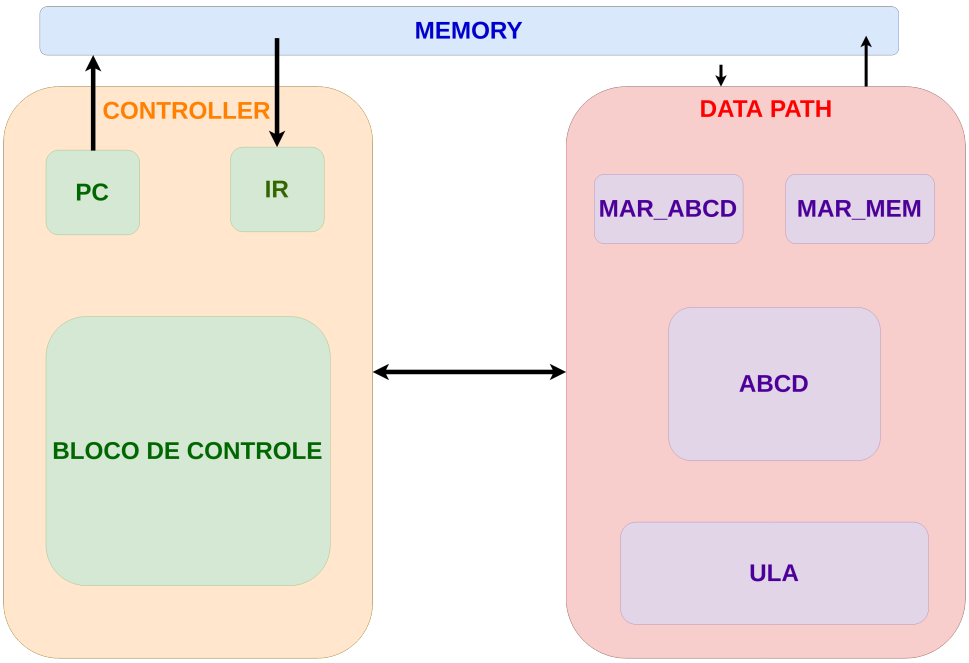
O Processador implementado possui arquitetura Von-Neumann (RAM 256x8-Bits), com 4 registradores de uso geral (A,B,C,D) de 8 bits cada, com um ponteiro para pilha (SP) de 8 Bits (na configuração SP-1 com SP máximo no endereço 231), com 2 bits de sinalização Z (True se o resultado da ULA for zero) e C (True se a ULA apresentar um carry de saída), com 24 posições de I/O representadas pelos endereços de 232 até 255 da memória RAM.

Nesse documento será explicado:

- A escolha dos opcodes para as 69 instruções
- A descrição dos componentes do Data Path
- A descrição dos componentes do Controlador
- A listagem e especificações dos sinais de controle
- Máquina de Estados
- Desenho completo da modelagem do Processador

1 Visão geral do Processador

Dividimos o processador em data path e em controller. O data path contém a ULA e o banco de registradores. Já o controle é onde se executa a máquina de estados e onde acontece a manipulação dos opcodes, onde se localiza o ponteiro para pilha e o contador do programa. Abaixo segue um esquemático simplificado do processador desenvolvido.



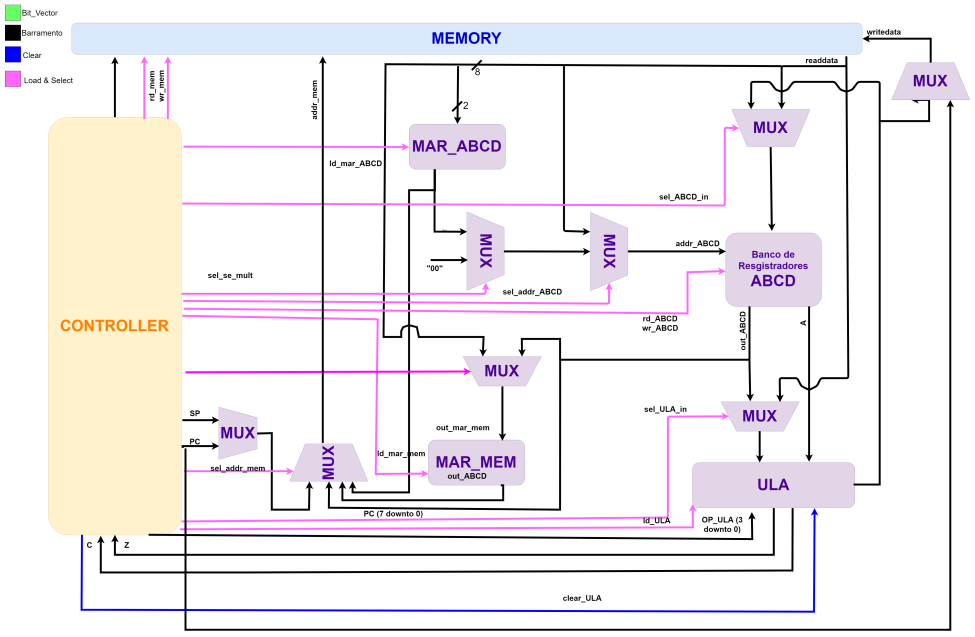
2 Escolha de Opcodes

A escolha dos opcodes foi estratégica para tentar diminuir o máximo possível o número de estados distintos na MDE. Logo, instruções com a mesma sequência de estados, foram padronizadas para facilitar a implementação. A tabela de opcodes está anexada a esse documento.

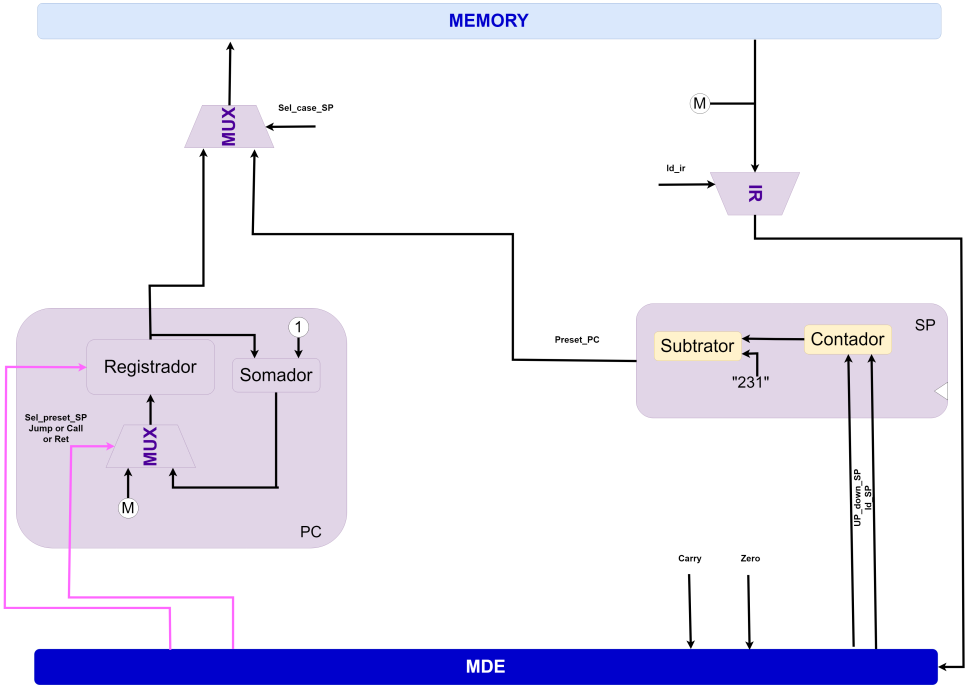
TABELA DE OPCODES				
opcode(7)	opcode(6 downto 3)		opcode(2 downto 0)	Parâmetros
0	”0000”	ADD	”000”	reg, regX
	”0001”	SUB		
	”0010”	AND	”001”	reg, [regX]
	”0011”	OR		
	”0100”	XOR	”010”	reg,[address]
	”0101”	SHL		
	”0110”	SHR	”011”	reg, constant
	”0111”	CMP		
	”1000”	PUSH	”100”	reg
	”1001”	POP		
	”1010”	MUL	”101”	[reg]
	”1011”	CALL		
	”1100”	INC	”110”	address
	”1101”	DEC		
	”1110”	NOT	”111”	constant
	”1111”	JMP		
1	”0000”	MOV	”000”	reg, regX
			”001”	reg, [regX]
			”010”	reg, [address]
			”011”	reg, constant
			”100”	[reg], regX
			”101”	[address], reg
			”110”	[address], constante
	”0001”	JC	X	
	”0010”	JNC		
	”0011”	JZ		
	”0100”	JNZ		
	”0101”	JA		
	”0110”	JNBE		
	”0111”	JAE		
	”1000”	JNB		
	”1001”	JB		
	”1010”	JNAE		
	”1011”	JBE		
	”1100”	JNA		
	”1101”	JE		
	”1110”	JNE		
	”1111”	RET	”000”	
		HLT	”111”	

3 Data Path

Nosso data path possui 2 registradores auxiliares, o MAR_MEM e o MAR_ABCD. Além disso, possui um banco de registradores, o ABCD, com apenas 4 registradores. Tal banco possui apenas 1 entrada de endereço e uma saída apenas. Todavia, existe uma saída permanente do registrador com o valor do registrador A. Tal saída é usada somente na multiplicação.



4 Controlador



5 Sinais de Controle

Podemos classificar os sinais que saem da máquina de estados em dois grupos, os que irão controlar o fluxo de ações do data path e os que vão controlar o fluxo de ações do controlador. Abaixo seguem os sinais saídos da máquina de estados para o data path:

- Sel_mar_mem: bit de seleção para o mux da entrada do registrador MAR_mem. '0' = saída da memória RAM; '1' = saída do banco de registradores;
- Ld_mar_mem: load do registrador MAR_mem
- Sel_se_mult: bit de seleção do mux que seleciona o registrador A para entrada de endereço do banco de registradores (usado para o caso da operação MUL). '0' = caso não seja operação de multiplicação; '1' = caso seja multiplicação;
- Wr_ABCD: bit de controle que escreve no banco de registradores ABCD
- Ld_mar_ABCD: load do registrador MAR_ABCD
- Sel_addr_ABCD: bit de seleção para o endereço do banco de registradores ABCD. Seleciona entre a saída do mux "se_mult"e a saída da RAM. "00"= saída do mux PC ou SP; "01"= saída do registrador MAR_mem; "10"= saída do banco de registradores ABCD; "11"= saída do registrador MAR_ABCD;
- Sel_addr_mem: bits de seleção da entrada de endereço na memória RAM. . "00"= PC ou SP; "01"= saída do MAR_mem; "10"= ; "11"= saída ;
- Wr_mem: bit de controle que escreve na memória RAM
- Sel_mem_in: bit de seleção do mux da entrada de dados na memória, seleciona entre a saída da ULA e o PC. '0' = saída da ULA; '1' = PC;
- Sel_ABCD_in: bit de seleção do mux da entrada de dados no banco de registradores, seleciona entre a saída da ULA e a saída da memória. '0' = saída da memória; '1' = saída da memória;
- Sel_ULA_in: bit de seleção do mux da entrada da ULA, seleciona entre a saída da memória e a saída do banco de registradores. '0' = saída do banco de registradores; '1' = saída da memória;
- Ld_ULA: load da ULA
- Clr_ULA: clear da ULA

Os sinais que vão para o controlador são, em sua maioria, para controlar o apontador para a pilha (SP) e para o contador do programa (PC). Abaixo segue a lista desses sinais e suas respectivas especificações:

- Sel_case_SP: bit de seleção do mux para seleção de addr na memória, caso '1', a saída é o SP, caso '0', a saída é o PC
- Ld_IR: load do IR
- Ld_PC: load do PC
- Ld_SP: load do SP

- Up_down_SP: Seleciona se o contador presente na estrutura do SP irá contar up ou down. UP='1', DOWN='0';
- Sel_preset_PC: Seleciona o mux de entrada para o preset do registrador PC. Caso '1', o preset do PC é ativado (jmp ou instruções de pilha), caso '0', não há preset;

6 Máquina de Estados

A máquina de estados é relativamente complexa devido ao extenso número de instruções. Para cada estado, dependendo do opcode, o valor de várias variáveis de controle serão diferentes. Segue a tabela com tais variáveis e em que estado elas mudam. É válido lembrar que o valor padrão de tais variáveis sempre será '0', logo, só serão apresentadas na tabela os valores de tais mudanças.

	Opcodes	Instrução	Sel_mar_mem	Ld_mar_mem	Sel_se_mult	Wr_ABCD	Ld_mar_ABCD	Sel_addr_ABCD	Sel_addr_mem	Wr_mem	Sel_mem_in	Sel_ABCD_in	Sel_ULA_in	Ld_ULA	Clr_ULA	OP_ULA (3 downto 0)	Ld_PC	Sel_case_SP	Ld_IR	Up_down_SP	Ld_SP	Sel_preset_PC
0	00000000	ADD reg, regX	(0)	(0)	(0)	6	2	4(1)	(00)	(0)	(0)	(1)	(0)	4	0	0 0 0 0	0, 2, 4	(0)	0	(0)	(0)	(0)
1	00000001	ADD reg, [regX]	4(1)	4	(0)	7	2	4(1)	5(01)	(0)	(0)	(1)	6(1)	6	0	0 0 0 1	0, 2, 4	(0)	0	(0)	(0)	(0)
2	00000010	ADD reg, [address]	(0)	4(1)	(0)	7	2	(0)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
3	00000011	ADD reg, constant	(0)	(0)	(0)	5	2	(0)	(00)	(0)	(0)	(1)	4(1)	4	0		0, 2, 4	(0)	0	(0)	(0)	(0)
8	00001000	SUB reg, regX	(0)	(0)	(0)	6	2	4(1)	(00)	(0)	(0)	(1)	(0)	4	0	0 0 1 0	0, 2, 4	(0)	0	(0)	(0)	(0)
9	00001001	SUB reg, [regX]	4(1)	4	(0)	7	2	4(1)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
10	00001010	SUB reg, [address]	(0)	4(1)	(0)	7	2	(0)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
11	00001011	SUB reg, constant	(0)	(0)	(0)	5	2	(0)	(00)	(0)	(0)	(1)	4(1)	4	0		0, 2, 4	(0)	0	(0)	(0)	(0)
16	00010000	AND reg, regX	(0)	(0)	(0)	6	2	4(1)	(00)	(0)	(0)	(1)	(0)	4	0	0 1 0 0	0, 2, 4	(0)	0	(0)	(0)	(0)
17	00010001	AND reg, [regX]	4(1)	4	(0)	7	2	4(1)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
18	00010010	AND reg, [address]	(0)	4(1)	(0)	7	2	(0)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
19	00010011	AND reg, constant	(0)	(0)	(0)	5	2	(0)	(00)	(0)	(0)	(1)	4(1)	4	0		0, 2, 4	(0)	0	(0)	(0)	(0)
24	00011000	OR reg, regX	(0)	(0)	(0)	6	2	4(1)	(00)	(0)	(0)	(1)	(0)	4	0	0 1 0 1	0, 2, 4	(0)	0	(0)	(0)	(0)
25	00011001	OR reg, [regX]	4(1)	4	(0)	7	2	4(1)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
26	00011010	OR reg, [address]	(0)	4(1)	(0)	7	2	(0)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
27	00011011	OR reg, constant	(0)	(0)	(0)	5	2	(0)	(00)	(0)	(0)	(1)	4(1)	4	0		0, 2, 4	(0)	0	(0)	(0)	(0)
32	00100000	XOR reg, regX	(0)	(0)	(0)	6	2	4(1)	(00)	(0)	(0)	(1)	(0)	4	0	0 1 1 0	0, 2, 4	(0)	0	(0)	(0)	(0)
33	00100001	XOR reg, [regX]	4(1)	4	(0)	7	2	4(1)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
34	00100010	XOR reg, [address]	(0)	4(1)	(0)	7	2	(0)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
35	00100011	XOR reg, constant	(0)	(0)	(0)	5	2	(0)	(00)	(0)	(0)	(1)	4(1)	4	0		0, 2, 4	(0)	0	(0)	(0)	(0)
40	00101000	SHL reg, regX	(0)	(0)	(0)	6	2	4(1)	(00)	(0)	(0)	(1)	(0)	4	0	0 1 1 1	0, 2, 4	(0)	0	(0)	(0)	(0)
41	00101001	SHL reg, [regX]	4(1)	4	(0)	7	2	4(1)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
42	00101010	SHL reg, [address]	(0)	4(1)	(0)	7	2	(0)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
43	00101011	SHL reg, constant	(0)	(0)	(0)	5	2	(0)	(00)	(0)	(0)	(1)	4(1)	4	0		0, 2, 4	(0)	0	(0)	(0)	(0)
48	00110000	SHR reg, regX	(0)	(0)	(0)	6	2	4(1)	(00)	(0)	(0)	(1)	(0)	4	0	1 0 0 0	0, 2, 4	(0)	0	(0)	(0)	(0)
49	00110001	SHR reg, [regX]	4(1)	4	(0)	7	2	4(1)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
50	00110010	SHR reg, [address]	(0)	4(1)	(0)	7	2	(0)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
51	00110011	SHR reg, constant	(0)	(0)	(0)	5	2	(0)	(00)	(0)	(0)	(1)	4(1)	4	0		0, 2, 4	(0)	0	(0)	(0)	(0)
56	00111000	CMP reg, regX	(0)	(0)	(0)	6	2	4(1)	(00)	(0)	(0)	(1)	(0)	4	0	1 0 0 1	0, 2, 4	(0)	0	(0)	(0)	(0)
57	00111001	CMP reg, [regX]	4(1)	4	(0)	7	2	4(1)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
58	00111010	CMP reg, [address]	(0)	4(1)	(0)	7	2	(0)	5(01)	(0)	(0)	(1)	6(1)	6	0		0, 2, 4	(0)	0	(0)	(0)	(0)
59	00111011	CMP reg, constant	(0)	(0)	(0)	5	2	(0)	(00)	(0)	(0)	(1)	4(1)	4	0		0, 2, 4	(0)	0	(0)	(0)	(0)

	Opcodes	Instrução	Sel_mar_mem	Ld_mar_mem	Sel_se_mult	Wr_ABCD	Ld_mar_ABCD	Sel_addr_ABCD	Sel_addr_mem	Wr_mem	Sel_mem_in	Sel_ABCD_in	Sel_ULA_in	Ld_ULA	Clr_ULA	OP_ULA (3 downto 0)	Ld_PC	Sel_case_SP	Ld_IR	Up_down_SP	Ld_SP	Sel_preset_PC
64	01000000	PUSH reg	X	(0)	(0)	(0)	(0)	1(1)	2(00)	2	(0)	X	1(0)	1	0	0000	0, 4	2 (1), 4(0)	5	3(1)	3	(0)
65	01000001	PUSH [reg]	X	(0)	(0)	(0)	(0)	1(1)	1(10)	3	(0)	X	2(1)	2	0		0, 5	3(1), 5(0)	6	4(1)	4	(0)
66	01000010	PUSH address	1(0)	1	(0)	(0)	(0)	X	1(00), 2(01)	3	(0)	X	2(1)	2	0		0, 5	3(1), 5(0)	6	4(1)	4	(0)
67	01000011	PUSH constant	X	(0)	(0)	(0)	(0)	X	1(00)	2	(0)	X	1(1)	1	0		0, 4	2(1), 4(0)	5	3(1)	3	
76	01001100	POP reg	X	(0)	(0)	2	1	(0)	(00)	(0)	(0)	2(0)	X	(0)	0		0, 4	2(1), 4(0)	5	3(0)	3	(0)
84	01010100	MUL reg	(0)	(0)	3(1), 4(1)	4	(0)	2(1)	(00)	(0)	(0)	3(1), 4(1)	(0)	2	(0)	0011	0, 2	(0)	0	(0)	(0)	(0)
85	01010101	MUL [reg]	2(1)	2	5(1)	5	(0)	2(1)	3(01)	(0)	(0)	5(1)	3(1), 4(1)	4	(0)		0, 2	(0)	0	(0)	(0)	(0)
86	01010110	MUL [address]	(0)	2	5(1)	5	(0)	(0)	3(01)	(0)	(0)	5(1)	3(1), 4(1)	4	(0)		0, 2	(0)	0	(0)	(0)	(0)
87	01010111	MUL constant	(0)	(0)	3(1)	3	(0)	(0)	(00)	(0)	(0)	3	2(1)	2	(0)		0, 2	(0)	0	(0)	(0)	(0)
94	01011110	CALL address	(0)	1	(0)	(0)	(0)	X	1, 2 (00), 4(01)	2	(1)	X	X	(0)	0	00000	0	2(1), 3(0)	5	3(1)	3	4(1)
100	01100100	INC reg	(0)	(0)	(0)	2	1	(0)	(0)	(0)	(0)	(1)	(0)	2	0	10100	0, 3	(0)	(0)	(0)	(0)	(0)
108	01101100	DEC reg	(0)	(0)	(0)	2	1	(0)	(0)	(0)	(0)	(1)	(0)	2	0	10111	0, 3	(0)	(0)	(0)	(0)	(0)
116	01110100	NOT reg	(0)	(0)	(0)	2	1	(0)	(0)	(0)	(0)	(1)	(0)	2	0	11000	0, 3	(0)	(0)	(0)	(0)	(0)
126	01111110	JMP address	(0)	1	X	(0)	(0)	X	1(00), 2(01)		(0)	X	X	(0)	0	00000	0	(0)	4	X	(0)	3(1)
128	10000000	MOV reg,regX	(0)	(0)	(0)	5	2	(0)	4(1)	(0)	(0)	5(1)	(0)	4	(0)	0000	0,2	(0)	0	(0)	(0)	(0)
129	10000001	MOV reg, [regX]	(0)	(0)	(0)	5	2	4(10)	4(1)	(0)	(0)	(0)	(0)	(0)	(0)		0,2	(0)	0	(0)	(0)	(0)
130	10000010	MOV reg [address]	(0)	4	(0)	7	2	(0)	5(01)	(0)	(0)	7(1)	6(1)	6	(0)		0,2	(0)	0	(0)	(0)	(0)
131	10000011	MOV reg, constant	(0)	(0)	(0)	4	2	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)		0,2	(0)	0	(0)	(0)	(0)
132	10000100	MOV [reg], regX	2(1)	2	(0)	5	(0)	2(1), 4(1)	4(01), 5(01)	(0)	(0)	(0)	(0)	4	(0)		0,2	(0)	0	(0)	(0)	(0)
133	10000101	MOV [address],reg	(0)	2	(0)	5	(0)	4(1)	4(01), 5(01)	(0)	(0)	(0)	(0)	4	(0)		0,2	(0)	0	(0)	(0)	(0)

	Opcodes	Instrução	Sel_mar_mem	Ld_mar_mem	Sel_se_mult	Wr_ABCD	Ld_mar_ABCD	Sel_addr_ABCD	Sel_addr_mem	Wr_mem	Sel_mem_in	Sel_ABCD_in	Sel_ULA_in	Ld_ULA	Clr_ULA	OP_ULA (3 downto 0)	Ld_PC	Sel_case_SP	Ld_IR	Up_down_SP	Ld_SP	Sel_preset_PC
136	10001000	JC	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0	0000	0	(0)	4	X	(0)	3(1)
144	10010000	JNC	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
152	10011000	JZ	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
160	10100000	JNZ	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
168	10101000	JA	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
176	10110000	JNBE	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
184	10111000	JAE	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
192	11000000	JNB	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
200	11001000	JB	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
208	11010000	JNAE	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
216	11011000	JBE	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
224	11100000	JNA	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
232	11101000	JE	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
240	11110000	JNE	(0)	1	X	(0)	(0)	X	1(00), 2(01)	(0)	(0)	X	X	(0)	0		0	(0)	4	X	(0)	3(1)
248	11111000	RET	(0)	(0)	(0)	(0)	(0)	X	(00)	(0)	X	X	X	(0)	0		3	1(1), 3(0)	4	2(0)	2	1
255	11111111	HLT																				