

Aula 8: Instruções de Transferência de Dados e Modos de Endereçamento no 8051 - EXERCÍCIOS

Professor: Eraldo Silveira e Silva

eraldo@ifsc.edu.br

1 Objetivos da Aula

- fixar os modos de endereçamento no 8051;
- fixar o modelo de memória do 8051;

2 Introdução

A fim de podermos realizar alguns exercícios mais complexos usando os diversos modos de endereçamento do 8051 vamos inicialmente apresentar algumas instruções de apoio:

```
INC DPTR           ; incrementa em uma unidade o registrador
DJNZ Rn,rotulo    ; salta para rotulo se Rn é zero. n=0,1
ADD A,#imediato   ; adiciona imediato ao A e coloque o resultado em A
```

3 Exercícios

3.1 Exercício 1

Usando endereçamento indireto zerar a memória RAM interna de 20H a 7FH.

3.2 Exercício 2

Faça um trecho de código para zerar 200 posições de memória RAM externa a partir da posição F000H.

3.3 Exercício 3

Um display LCD de 2 linhas 16 posições está mapeado em memória externa de dados na posição 7000H (primeira linha - primeiro caracter) e 7010H(segunda linha primeiro caracter. Copiar a mensagem a “Entrar com a senha:” que se encontra na memória de código a partir da posição 8765H para a primeira linha do display.

3.4 Exercício 4

Considere o trecho de programa abaixo. Faça o papel do Assembler e traduza o programa abaixo para a linguagem de máquina colocando-o a partir da posição 0 na memória de programa.

```
MOV DPTR,#F8A5H
MOV A,#85
MOVX @DPTR,A
MOV 45H,A
MOV 45H,#45H
```

3.5 Exercício 5

A palavra “INCONSTITUCIONALISSIMAMENTE” está na memória RAM externa a partir da posição 7000H. Copie esta palavra para a RAM interna a partir da posição 40H transformando-a para letras minúsculas. Lembre que somando 20H a um ASCII de uma letra maiúscula obtemos a minúscula.

ANEXO - Tabela de Opcodes do 8051

Code	Mnemonic	Operands	Code	Mnemonic	Operands	Code	Mnemonic	Operands	Code	Mnemonic	Operands
00	NOP		40	JC	offset	80	SJMP	offset	C0	PUSH	direct
01	AJMP	addr11	41	AJMP	addr11	81	AJMP	addr11	C1	AJMP	addr11
02	LJMP	addr16	42	ORL	direct,A	82	ANL	C,bit	C2	CLR	bit
03	RR	A	43	ORL	direct,#immed	83	MOVC	A,@A+PC	C3	CLR	C
04	INC	A	44	ORL	A,#immed	84	DIV	AB	C4	SWAP	A
05	INC	direct	45	ORL	A,direct	85	MOV	direct,direct	C5	XCH	A,direct
06	INC	@R0	46	ORL	A,@R0	86	MOV	direct,@R0	C6	XCH	A,@R0
07	INC	@R1	47	ORL	A,@R1	87	MOV	direct,@R1	C7	XCH	A,@R1
08	INC	R0	48	ORL	A,R0	88	MOV	direct,R0	C8	XCH	A,R0
09	INC	R1	49	ORL	A,R1	89	MOV	direct,R1	C9	XCH	A,R1
0A	INC	R2	4A	ORL	A,R2	8A	MOV	direct,R2	CA	XCH	A,R2
0B	INC	R3	4B	ORL	A,R3	8B	MOV	direct,R3	CB	XCH	A,R3
0C	INC	R4	4C	ORL	A,R4	8C	MOV	direct,R4	CC	XCH	A,R4
0D	INC	R5	4D	ORL	A,R5	8D	MOV	direct,R5	CD	XCH	A,R5
0E	INC	R6	4E	ORL	A,R6	8E	MOV	direct,R6	CE	XCH	A,R6
0F	INC	R7	4F	ORL	A,R7	8F	MOV	direct,R7	CF	XCH	A,R7
10	JBC	bit,offset	50	JNC	offset	90	MOV	DPTR,#immed	D0	POP	direct
11	ACALL	addr11	51	ACALL	addr11	91	ACALL	addr11	D1	ACALL	addr11
12	LCALL	addr16	52	ANL	direct,A	92	MOV	bit,C	D2	SETB	bit
13	RRC	A	53	ANL	direct,#immed	93	MOVC	A,@A+DPTR	D3	SETB	C
14	DEC	A	54	ANL	A,#immed	94	SUBB	A,#immed	D4	DA	A
15	DEC	direct	55	ANL	A,direct	95	SUBB	A,direct	D5	DJNZ	direct,offset
16	DEC	@R0	56	ANL	A,@R0	96	SUBB	A,@R0	D6	XCHD	A,@R0
17	DEC	@R1	57	ANL	A,@R1	97	SUBB	A,@R1	D7	XCHD	A,@R1
18	DEC	R0	58	ANL	A,R0	98	SUBB	A,R0	D8	DJNZ	R0,offset
19	DEC	R1	59	ANL	A,R1	99	SUBB	A,R1	D9	DJNZ	R1,offset
1A	DEC	R2	5A	ANL	A,R2	9A	SUBB	A,R2	DA	DJNZ	R2,offset
1B	DEC	R3	5B	ANL	A,R3	9B	SUBB	A,R3	DB	DJNZ	R3,offset
1C	DEC	R4	5C	ANL	A,R4	9C	SUBB	A,R4	DC	DJNZ	R4,offset
1D	DEC	R5	5D	ANL	A,R5	9D	SUBB	A,R5	DD	DJNZ	R5,offset
1E	DEC	R6	5E	ANL	A,R6	9E	SUBB	A,R6	DE	DJNZ	R6,offset
1F	DEC	R7	5F	ANL	A,R7	9F	SUBB	A,R7	DF	DJNZ	R7,offset
20	JB	bit,offset	60	JZ	offset	A0	ORL	C,/bit	E0	MOVX	A,@DPTR
21	AJMP	addr11	61	AJMP	addr11	A1	AJMP	addr11	E1	AJMP	addr11
22	RET		62	XRL	direct,A	A2	MOV	C,bit	E2	MOVX	A,@R0
23	RL	A	63	XRL	direct,#immed	A3	INC	DPTR	E3	MOVX	A,@R1
24	ADD	A,#immed	64	XRL	A,#immed	A4	MUL	AB	E4	CLR	A
25	ADD	A,direct	65	XRL	A,direct	A5	reserved		E5	MOV	A,direct
26	ADD	A,@R0	66	XRL	A,@R0	A6	MOV	@R0,direct	E6	MOV	A,@R0
27	ADD	A,@R1	67	XRL	A,@R1	A7	MOV	@R1,direct	E7	MOV	A,@R1
28	ADD	A,R0	68	XRL	A,R0	A8	MOV	R0,direct	E8	MOV	A,R0
29	ADD	A,R1	69	XRL	A,R1	A9	MOV	R1,direct	E9	MOV	A,R1
2A	ADD	A,R2	6A	XRL	A,R2	AA	MOV	R2,direct	EA	MOV	A,R2
2B	ADD	A,R3	6B	XRL	A,R3	AB	MOV	R3,direct	EB	MOV	A,R3
2C	ADD	A,R4	6C	XRL	A,R4	AC	MOV	R4,direct	EC	MOV	A,R4
2D	ADD	A,R5	6D	XRL	A,R5	AD	MOV	R5,direct	ED	MOV	A,R5
2E	ADD	A,R6	6E	XRL	A,R6	AE	MOV	R6,direct	EE	MOV	A,R6
2F	ADD	A,R7	6F	XRL	A,R7	AF	MOV	R7,direct	EF	MOV	A,R7
30	JNB	bit,offset	70	JNZ	offset	B0	ANL	C,/bit	F0	MOVX	@DPTR,A
31	ACALL	addr11	71	ACALL	addr11	B1	ACALL	addr11	F1	ACALL	addr11
32	RETI		72	ORL	C,bit	B2	CPL	bit	F2	MOVX	@R0,A
33	RLC	A	73	JMP	@A+DPTR	B3	CPL	C	F3	MOVX	@R1,A
34	ADDC	A,#immed	74	MOV	A,#immed	B4	CJNE	A,#immed,offset	F4	CPL	A
35	ADDC	A,direct	75	MOV	direct,#immed	B5	CJNE	A,direct,offset	F5	MOV	direct,A
36	ADDC	A,@R0	76	MOV	@R0,#immed	B6	CJNE	@R0,#immed,offset	F6	MOV	@R0,A
37	ADDC	A,@R1	77	MOV	@R1,#immed	B7	CJNE	@R1,#immed,offset	F7	MOV	@R1,A
38	ADDC	A,R0	78	MOV	R0,#immed	B8	CJNE	R0,#immed,offset	F8	MOV	R0,A
39	ADDC	A,R1	79	MOV	R1,#immed	B9	CJNE	R1,#immed,offset	F9	MOV	R1,A
3A	ADDC	A,R2	7A	MOV	R2,#immed	BA	CJNE	R2,#immed,offset	FA	MOV	R2,A
3B	ADDC	A,R3	7B	MOV	R3,#immed	BB	CJNE	R3,#immed,offset	FB	MOV	R3,A
3C	ADDC	A,R4	7C	MOV	R4,#immed	BC	CJNE	R4,#immed,offset	FC	MOV	R4,A
3D	ADDC	A,R5	7D	MOV	R5,#immed	BD	CJNE	R5,#immed,offset	FD	MOV	R5,A
3E	ADDC	A,R6	7E	MOV	R6,#immed	BE	CJNE	R6,#immed,offset	FE	MOV	R6,A
3F	ADDC	A,R7	7F	MOV	R7,#immed	BF	CJNE	R7,#immed,offset	FF	MOV	R7,A