

DISCIPLINA: LABORATÓRIO DE LOAC

RELATÓRIO: LABORATÓRIO 01

(2022)

ALUNO: Edilson do Nascimento Costa Júnior

Prof. Dr.: Misael Elias de Moraes

CAMPUS CAMPINA GRANDE, abril de 2021

RESUMO

Este relatório tem como objetivo expor os conhecimentos adquiridos nas aulas da disciplina de Laboratório de Organização e Arquitetura de Computadores, utilizando a linguagem Assembly no simulador de microcontrolador MCU 8051 IDE.

INTRODUÇÃO

A partir do código disponibilizado para o experimento teve-se o objetivo de estudar a realização de programas no microcontrolador MCU 8051. Destinchando cada linha do programa, com a finalidade de compreender a interação entre os comandos, foram mostrados os modos de endereçamento, indicando as alterações na memória de dados e o que foi gravado na memória do programa, as mnemônicas e os operandos.

MATERIAL E MÉTODOS:

Na produção deste relatório fez se o uso do simulador de microcontrolador MCU 8051, além de um código de programa que foi disponibilizado.

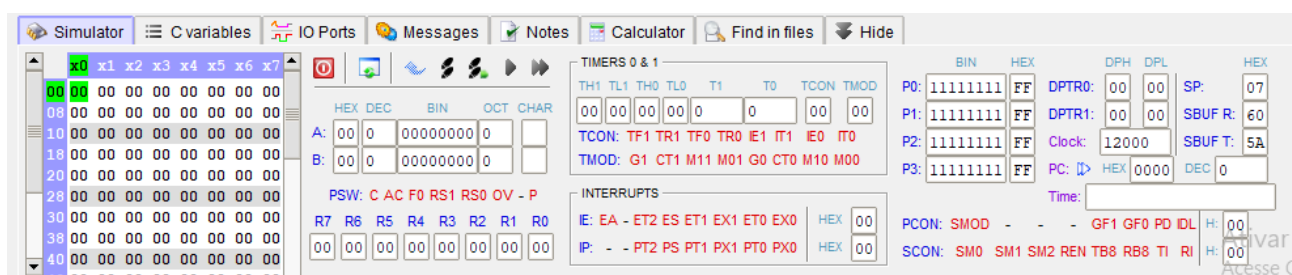
```
1 ORG      0000H
2 INICIO:
3          MOV     A, #051h
4          MOV     R2, #02h
5          MOV     28h, 02h
6          MOV     R1, #42
7          MOV     51h, #12
8          MOV     @R1, #33h
9          SWAP    A
10         XCH     A, 2Ah
11         MOV     32, 01h
12         MOV     DPTR, #TAB
13         MOV     A, 28h
14         MOVC    A, @A+DPTR
15         LJMP    INICIO
16 TAB:     DB      37h, 12, 47, 99
17
18 END
```

RESULTADOS E DISCUSSÃO

A partir da compreensão da estrutura do código e dos objetivos propostos, podemos avançar para os experimentos solicitados no tópico 4.2 referente às Atividades. Abaixo estão apresentados os comentários indicando os modos de endereçamento de cada linha, além das mnemônicas e os seus operandos. Posteriormente será mostrado também as devidas alterações na memória de dados e o que foi gravado na memória do programa.

```
1 ORG 0000H ;endereço inicial do programa
2 INICIO: ;rótulo da operação
3 MOV A, #051h ;A constante é movida para o acumulador . Modo de endereçamento
4 imediato. Mnemonica: MOV. Operando: A,#051h.
5 MOV R2, #02h ;O valor #02h é movido para o registrador R2. Modo de
6 endereçamento registrador e imediato. Mnemonica: MOV. Operando: R2,#02h
7 MOV 28h, 02h ;O conteúdo da memória em 02h é movido para memória de 28h. Modo
8 de endereçamento direto. Mnemonica: MOV. Operando: 28h, 02h
9 MOV R1, #42 ;O valor 42 é movido para o registrador R1. Modo de endereçamento
10 registrador e imediato. Mnemonica: MOV. Operando: R1,#42.
11 MOV 51h, #12 ;O valor 12 é movido para a memória 51h. Modo de endereçamento
12 direto e imediato. Mnemonica: MOV. Operando: 51h, #12
13 MOV @R1, #33h ;O numero 33h é movido para a memória ram endereçada no R1. Modo
14 de endereçamento imediato e indireto. Mnemonica: MOV. Operando @R1, #33h.
15 SWAP A ; Troca os nibbles do acumulador. Mnemonica: SWAP. Operando: A.
16 XCH A, 2Ah ;O conteúdo da memória de 2Ah é trocado com acumulador. Modo de
17 endereçamento direto. Mnemonica: XCH. Operando: A,2Ah.
18 MOV 32,01h ;O conteúdo da memória 01h é enviado para a memória de 32. Modo
19 de endereçamento direto. Mnemonica: MOV. Operando: 32,01h.
20 MOV DPTR, #TAB ;O valor de TAB é movido para o Registrador DPTR. Modo de
21 endereçamento imediato. Mnemonica: MOV. Operando: DPTR,#TAB
22 MOV A,28h ;O valor 28h é movido para o Acumulador. Modo de endereçamento
23 direto. Mnemonica: MOV. Operando: A,28h.
24 MOVC A,@A+DPTR ;o valor A é somado com o registrador DPTR e movido para o
25 acumulador. Modo de endereçamento: Indexado. Mnemonica: MOVC. Operando: A,@A+DPTR.
26 LJMPL INICIO ; Retorna a posição de memória armazenada no rótulo do INICIO.
27 Mnemonica: LJMPL. Operando: INICIO
28 TAB: DB 37h, 12, 47, 99 ; posiciona 4 bytes de forma ordenada a partir da posição atual.
29 Mnemonica: TAB. Operando: DB 7h,12,47,99.
30 END ;informa ao compilador o final da execução
```

Ao inicializar o programa, a interface que temos é:



Como fica na memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	00	00	00	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

A memória do programa:

[illegible]

Na **linha 04**, o código está movendo o valor de #02h para o registrador R2.

A memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	00	00	00	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

A memória do programa:

[illegible]

Na **linha 05** move-se o valor de 02h para o registrador 28h.

A memória de dados e a memória do programa ficam:

[illegible][illegible]

Na **linha 06** do programa, o código está movendo o valor de #42 para o registrador R1. A memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	00	02	00	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	02	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

A memória do programa:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
																	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	tQz...	(y*	uQ.	w3...												
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*	(...	7.	/									
0020	63																c															

Na **linha 07**, o código move o valor de #12 para o registrador 51h. A memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	2A	02	00	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	02	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

A memória do programa:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	tQz...	(y*	uQ.	w3..												
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*	(...	7.	/									
0020	63																c															

Na **linha 08**, move-se o valor de #33h para a posição de memória R1.

	x0	x1	x2	x3	x4	x5	x6	x7
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	00	00	00	00	00	00
50	00	0C	00	00	00	00	00	00
58	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00
68	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00

A memória do programa:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	t	Q	z	...	(y	*	u	Q	.	w	3
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*	(...	7
0020	63																c															

Na **linha 09**, o código está fazendo uma troca entre as nibbles do acumulador A.

A memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	2A	02	00	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	02	00	33	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

A memória do programa:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	tQz...	(y*uQ.w3..														
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*..	(...7./													
0020	63																c															

Na **linha 10**, faz-se uma troca entre o conteúdo do acumulador e do registro.

A memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	2A	02	00	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	02	00	33	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	tQz...	(y*uQ.w3.														
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*..	(....7./													
0020	63																c															

Na **linha 11**, está movendo o valor de 01h para o registrador 32.

A memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	2A	02	00	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	02	00	15	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

A memória do programa:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	tQz...	(y*uQ.w3.														
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*.	(...	7./												
0020	63																c															

Na **linha 12**, o código está movendo o valor de #TAB para o registrador DPTR.

A memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	2A	02	00	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	2A	00	00	00	00	00	00	00
28	02	00	15	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

A memória do programa:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
																	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	t	Q	z	...	(y	*	u	Q	.	w	3
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*
0020	63																c															

Na **linha 13**, move-se o valor de 28h para o registrador A.

	x0	x1	x2	x3	x4	x5	x6	x7
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	00	00	00	00	00	00
50	00	0C	00	00	00	00	00	00
58	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00
68	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00

A memória do programa:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
																	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	t	Q	z
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*
0020	63																c															

Na **linha 14**, somando DPTR + o acumulador forma um endereço de memória do programa e o conteúdo do endereço é movido para o acumulador.

A memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	00	2A	02	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	2A	00	00	00	00	00	00	00
28	02	00	15	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

A memória do programa:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	tQz...	(y*uQ.w3..														
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*..	(...	7./										
0020	63																c															

Na **linha 15**, o código informa ao compilador um endereço de memória. A memória de dados:

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	00	2A	02	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	2A	00	00	00	00	00	00	00
28	02	00	15	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	74	51	7A	02	85	02	28	79	2A	75	51	0C	77	33	C4	C5	tQz...	(y*uQ.w3..														
0010	2A	85	01	20	90	00	1D	E5	28	93	02	00	00	37	0C	2F	*	(...	7	.	/								
0020	63																c															

Na **linha 16**, tem-se o DB que serve para informar ao compilador para reservar memória com certo conteúdo.

Conteúdo do acumulador quando o conteúdo do PC for zero, pela terceira vez é:

	HEX	DEC	BIN	OCT	CHAR
A:	2F	47	00101111	57	/
B:	00	0	00000000	0	

Na **linha 17**, o programa é finalizado.

CONCLUSÃO

O experimento foi realizado com sucesso diante de suas propostas, visto que os modos de endereçamento possuem grande importância para o microcontrolador, bem como seus operandos, constituindo partes imprescindíveis para a compilação ser executada sem erros pelo computador. Cada etapa de sua execução foi entendida e analisada, e considerada então uma prerrogativa básica para o entendimento da linguagem Assembly.

REFERÊNCIAS BIBLIOGRÁFICAS

MORAIS, Misael. Organização e arquitetura de computadores. [S. l.], 16 mar. 2021. Disponível em:

<https://classroom.google.com/u/0/c/MTQxNTk2Mzk0MDQz/m/Mjk5NDIxMjYxOTA1/details>.

Acesso em: 20 maio. 2022.

Uma Introdução a Instruções Assembly. **cin.ufpe**, 2021.

Disponível em:

<https://www.cin.ufpe.br/~arfs/Assembly/apostilas/Tutorial%20Assembly%20-%20Gavin/ASM3.HTM>.

Acesso em: 20 maio. 2022..

SCHMIDT, Gerhard. Introdução para o iniciante à Linguagem Assembly dos Microprocessadores ATMEL-AVR. Dezembro de 2003. Disponível em:

https://www.ic.unicamp.br/~ducatte/mc404/2009/docs/beginner_pt.pdf. Acesso em: 20 maio. 2022.