



UNIVERSIDADE ESTADUAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
BACHARELADO EM CIÊNCIAS
DA COMPUTAÇÃO

LUCAS DE LUCENA SIQUEIRA

DISCIPLINA: LABORATÓRIO DE ORGANIZAÇÃO E ARQUITETURA DE
COMPUTADORES

EXPERIMENTO 05

RELATÓRIO - INTERRUPÇÃO

CAMPINA GRANDE - PB

2021

SUMÁRIO

1. RESUMO.....	3
2. INTRODUÇÃO.....	3
3. MATERIAL E MÉTODOS.....	3
3.1. OBJETIVO.....	3
3.2. SOFTWARE NECESSÁRIO.....	3
3.3. HARDWARE NECESSÁRIO.....	3
3.4. PROCEDIMENTOS.....	4
3.4.1.LIGAÇÕES DO DISPLAY DE 7 SEGMENTOS.....	4
3.4.2. LISTAGEM DO PROGRAMA EM LM.....	4
3.4.3. FUNCIONAMENTO DO PROGRAMA.....	6
3.4.4. FUNCIONAMENTO DO PAINEL DE LED's.....	8
4. RESULTADOS E DISCUSSÃO.....	9
5. CONCLUSÕES.....	9
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	9

1. RESUMO

Neste trabalho em questão é possível observar alguns conceitos básicos e importantes sobre a programação do microcontrolador MCS51 em Assembly a partir do simulador MCU8051 IDE. Por fim, foram feitas as atividades propostas pelo professor e expostas neste relatório para a prática e conhecimento do conjunto de instrução do microcontrolador e alguns dos seus periféricos, como a utilização do painel de LED's e do KeyPad com a utilização da técnica de multiplexação.

2. INTRODUÇÃO

O presente trabalho tem como objetivo introduzir o estudo do microcontrolador MCS51 a partir do uso do simulador MCU8051 IDE. Tendo o Assembly como linguagem dominante na programação do referente microcontrolador, se faz necessário também o estudo da mesma a partir de consultas em seu *Datasheet*, que é nada mais do que uma folha com dados e especificações técnicas e de desempenho do produto levado em consideração, que no nosso caso é o microcontrolador MCS51.

3. MATERIAL E MÉTODOS

3.1. OBJETIVO

O experimento em questão tem como objetivo induzir o estudo e a prática da programação do microcontrolador MCS51 a partir do simulador MCU8051 IDE com o uso da linguagem Assembly. Tornando possível estudar o painel de LED's e o KeyPad com a técnica de multiplexação e interrupção.

3.2. HARDWARE NECESSÁRIO

- Computador com sistema operacional superior ou equivalente ao Windows 7.

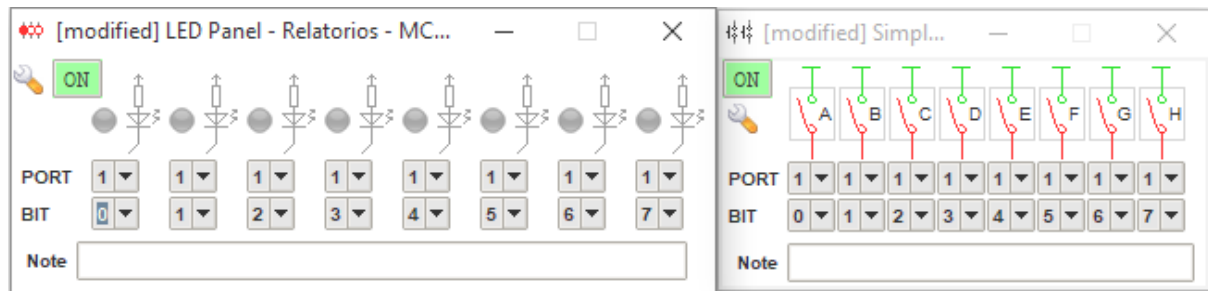
3.3. SOFTWARE NECESSÁRIO

- Simulador MCU 8051 IDE.

3.4. PROCEDIMENTOS

3.4.1. LIGAÇÕES DO DISPLAY DE 7 SEGMENTOS

Na figura abaixo estão as conexões realizadas no painel de LED “LED Painel” e no “Simple KeyPad”. Para a ligação dos bits de 0 a 7 inferiores foi utilizada a porta P1.



3.4.2. LISTAGEM DO PROGRAMA EM LM

- ORG 00H

É uma pseudo-instrução que define o endereço em que o programa irá começar ou continuar.

- LJMP inicio

Faz um salto para o label “inicio”.

- ORG 0BH

É uma pseudo-instrução que define o endereço em que o programa irá continuar.

- CPL P1.0

Define um temporizador.

- RETI

Gera interrupções.

- ORG 1BH

É uma pseudo-instrução que define o endereço em que o programa irá continuar.

- CPL P1.1

Define um temporizador.

- RETI

Gera interrupções.

- ORG 30H

É uma pseudo-instrução que define o endereço em que o programa irá continuar.

- INICIO: MOV SP,#2FH

Move para o registrador SP o valor 2FH. INICIO é uma label.

- MOV IE,#8AH

Move para o registrador IE o valor 8AH.

- MOV TMOD,#10H

Move o valor 10H e ativa o timer 1.

- SETB TR0

Define o TL0 através da porta TR0.

- SETB TR1

Define o TH0 através da porta TR1.

- SJMP \$

Retornar para a subrotina.

- END

Define o final do programa.

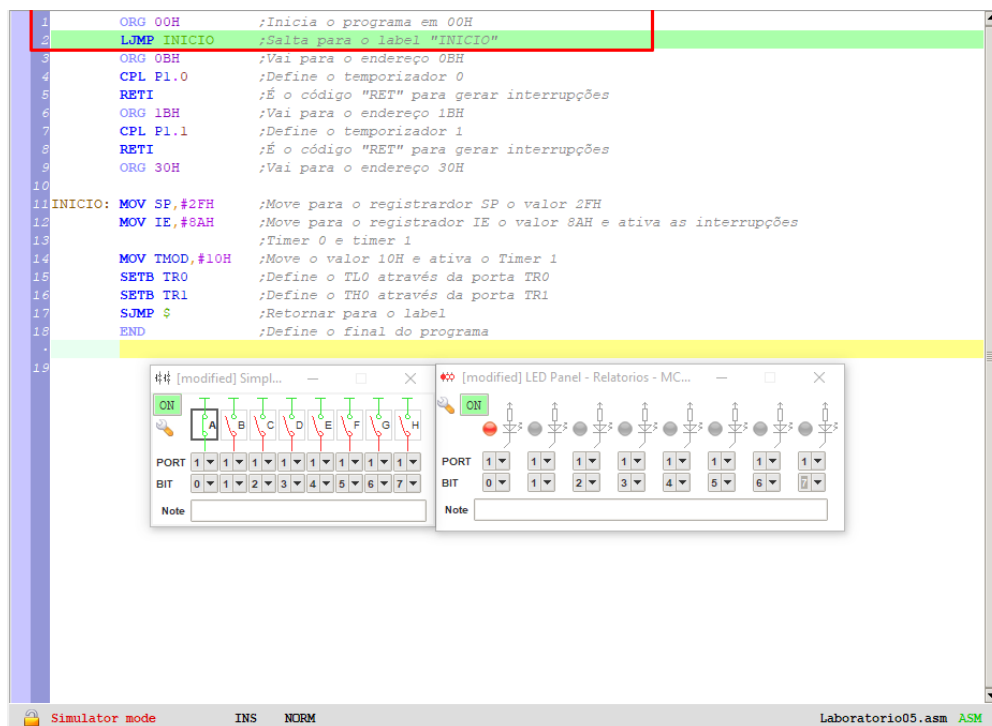
3.4.3. FUNCIONAMENTO DO PROGRAMA

Aqui estará presente toda a descrição detalhada de cada linha do programa utilizado como exemplo, apresentando em cada linha seu respectivo funcionamento comentado.

```
1      ORG 00H      ;Inicia o programa em 00H
2      LJMP INICIO  ;Salta para o label "INICIO"
3      ORG 0BH      ;Vai para o endereço 0BH
4      CPL P1.0      ;Define o temporizador 0
5      RETI          ;É o código "RET" para gerar interrupções
6      ORG 1BH      ;Vai para o endereço 1BH
7      CPL P1.1      ;Define o temporizador 1
8      RETI          ;É o código "RET" para gerar interrupções
9      ORG 30H      ;Vai para o endereço 30H
10
11 INICIO: MOV SP,#2FH ;Move para o registrador SP o valor 2FH
12         MOV IE,#8AH ;Move para o registrador IE o valor 8AH e ativa as interrupções
13         ;Timer 0 e timer 1
14         MOV TMOD,#10H ;Move o valor 10H e ativa o Timer 1
15         SETB TR0      ;Define o TL0 através da porta TR0
16         SETB TR1      ;Define o TH0 através da porta TR1
17         SJMP $         ;Retornar para o label
18         END           ;Define o final do programa
```

Comentários e execução do código acima:

No primeiro momento do programa, ao executar as duas primeiras linhas, o programa será iniciado em 00 em hexadecimal.



No segundo momento do programa, tem-se a declaração dos vetores, mas sabendo que as interrupções propostas funcionam de maneira vetorizada acaba sendo necessário fazer uma função para as mesmas. o programa vai até o endereço 0BH e declara o temporizador 0, logo após o complemento para o bit P1.0 é realizado caso o valor seja igual a #0, o que fará o complemento de 1 e consequentemente o LED irá acender. O mesmo ocorre para o temporizador 1.

```

1      ORG 00H           ;Inicia o programa em 00H
2      LJMP INICIO      ;Salta para o label "INICIO"
3      ORG 0BH          ;Vai para o endereço 0BH
4      CPL P1.0         ;Define o temporizador 0
5      RETI             ;É o código "RET" para gerar interrupções
6      ORG 1BH          ;Vai para o endereço 1BH
7      CPL P1.1         ;Define o temporizador 1
8      RETI             ;É o código "RET" para gerar interrupções
9      ORG 30H          ;Vai para o endereço 30H
10
11 INICIO: MOV SP,#2FH   ;Move para o registrador SP o valor 2FH
12 MOV IE,#8AH          ;Move para o registrador IE o valor 8AH e ativa as interrupções
13                      ;Timer 0 e timer 1
14 MOV TMOD,#10H        ;Move o valor 10H e ativa o Timer 1
15 SETB TR0             ;Define o TL0 através da porta TR0
16 SETB TR1             ;Define o TH0 através da porta TR1
17 SJMP $              ;Retornar para o label
18 END                  ;Define o final do programa
19

```

[modified] Simpl...

ON

A	B	C	D	E	F	G	H
PORT 1	1	1	1	1	1	1	1
BIT 0	1	2	3	4	5	6	7

Note

[modified] LED Panel - Relatorios - MC...

ON

1	1	1	1	1	1	1	1
BIT 0	1	2	3	4	5	6	7

Note

Simulator mode INS NORM Laboratorio05.asm ASM

No terceiro momento, o label INICIO vai definir o inicio no endereço 30H. Logo em seguido o valor 2FH é movido para o registrador SP, o que imperirá com que a pilha ultrapasse sua capacidade, fazendo com que os endereços sejam perdidos. Na próxima parte o valor 8AH é movido para o registrador IE que irá ativar as interrupções (timer 1 e timer 0). Na linha seguinte o valor 10H é movido para o TMOD que vai ativar o timer 0 para que se torne um contador e o timer 1 para que se torne um temporizador. Por fim, os controladores TL0 e TH0 são definidos pelas portas TR0 e TR1, respectivamente, que vão saltar para o início da execução do programa formando um loop.

The screenshot displays the 8086 simulator interface. The main window shows the following assembly code:

```

1      ORG 00H      ;Inicia o programa em 00H
2      LJMIP INICIO ;Salta para o label "INICIO"
3      ORG 0BH      ;Vai para o endereço 0BH
4      CPL P1.0      ;Define o temporizador 0
5      RETI          ;É o código "RET" para gerar interrupções
6      ORG 1BH      ;Vai para o endereço 1BH
7      CPL P1.1      ;Define o temporizador 1
8      RETI          ;É o código "RET" para gerar interrupções
9      ORG 30H      ;Vai para o endereço 30H
10
11 INICIO: MOV SP,#2FH ;Move para o registrador SP o valor 2FH
12         MOV IE,#8AH ;Move para o registrador IE o valor 8AH e ativa as interrupções
13         ;Timer 0 e timer 1
14         MOV TMOD,#10H ;Move o valor 10H e ativa o Timer 1
15         SETB TR0      ;Define o TL0 através da porta TR0
16         SETB TR1      ;Define o TH0 através da porta TR1
17         SJMP $         ;Retornar para o label
18     END              ;Define o final do programa
19

```

Below the code, two hardware status windows are visible:

- [modified] Simpl...:** Shows the status of ports A, B, C, D, E, F, G, and H. Port A is highlighted with a green box. The PORT and BIT values are shown for each port.
- [modified] LED Panel - Relatorios - MC...:** Shows the status of the LED panel. The PORT and BIT values are shown for each LED.

The simulator mode is indicated at the bottom as "Simulator mode". The file name "Laboratorio05.asm" is shown in the bottom right corner.

3.4.4. FUNCIONAMENTO DO PAINEL DE LED's

The screenshot displays the 8086 simulator interface. The main window shows the following assembly code:

```

8      RETI          ;É o código "RET" para gerar interrupções
9      ORG 30H      ;Vai para o endereço 30H
10
11 INICIO: MOV SP,#2FH ;Move para o registrador SP o valor 2FH
12         MOV IE,#8AH ;Move para o registrador IE o valor 8AH e ativa as interrupções
13         ;Timer 0 e timer 1
14         MOV TMOD,#10H ;Move o valor 10H e ativa o Timer 1
15         SETB TR0      ;Define o TL0 através da porta TR0
16         SETB TR1      ;Define o TH0 através da porta TR1
17         SJMP $         ;Retornar para o label
18     END              ;Define o final do programa
19

```

Below the code, two hardware status windows are visible:

- [modified] Simpl...:** Shows the status of ports A, B, C, D, E, F, G, and H. Port A is highlighted with a green box. The PORT and BIT values are shown for each port.
- [modified] LED Panel - Relatorios - MC...:** Shows the status of the LED panel. The PORT and BIT values are shown for each LED.

The simulator mode is indicated at the bottom as "Simulator mode". The file name "Laboratorio05.asm" is shown in the bottom right corner.

4. RESULTADOS E DISCUSSÃO

Após todo o experimento foi possível compreender algumas peculiaridades da linguagem a nível de máquina “Assembly”. Após a execução do código listado foi possível compreender a relação entre as operações realizadas, o painel de LED 's e os conceitos de multiplexação e interrupção.

5. CONCLUSÕES

Foi possível concluir que a linguagem Assembly em atuação ao microcontrolador MCU 8051 pode dar uma vasta biblioteca de opções e atuações diante de seus registradores e portas, assim como demonstrado no decorrer do relatório com o exemplo do painel de LED 's e conceituações de multiplexação e interrupção.

6. REFERÊNCIAS BIBLIOGRÁFICAS

NETO, Hugo Vieira. MICROCONTROLADORES MCS51. Curitiba, 2002. Disponível em: <https://pessoal.dainf.ct.utfpr.edu.br/hvieir/download/mcs51.pdf>. Acesso em: 24 jul. 2021.

MORAIS, Misael. Organização e arquitetura de computadores. [S. l.], . 2021. Disponível em: <https://drive.google.com/drive/folders/0Bwjlecok7TpyfnAtQmx6bE4yZ043amNsbnRxMkF4UF1pWVZhWmRfeVBSeHRRVi1xRzNOZnM?resourcekey=0-WmQ6S1i6hLYyCoGBLbMeGw>. Acesso em: 01 ago. 2021