



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 03

RELATÓRIO - ACIONAMENTO DE LED'S E LEITURA DE TECLAS

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.....	3
3.4.1. LIGAÇÕES DO PAINEL DE LED.....	4
3.4.2. LIGAÇÃO DO PAINEL DE TECLAS.....	5
3.4.2. LISTAGEM DO PROGRAMA EM LM.....	6
3.4.3. FUNCIONAMENTO DO PROGRAMA.....	7
3.4.4. FUNCIONAMENTO DO PROGRAMA FORNECIDO.....	9
3.4.5. IMPLEMENTAÇÃO DO CIRCUITO LÓGICO.....	12
4. RESULTADOS E DISCUSSÃO.....	12
5. CONCLUSÕES.....	12
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	13

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 o acendimento de LED's e leitura de teclas.

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 a rotação de led 's além da leitura de dados no simulador.

3.2. HARDWARE NECESSÁRIO

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

3.3. SOFTWARE NECESSÁRIO

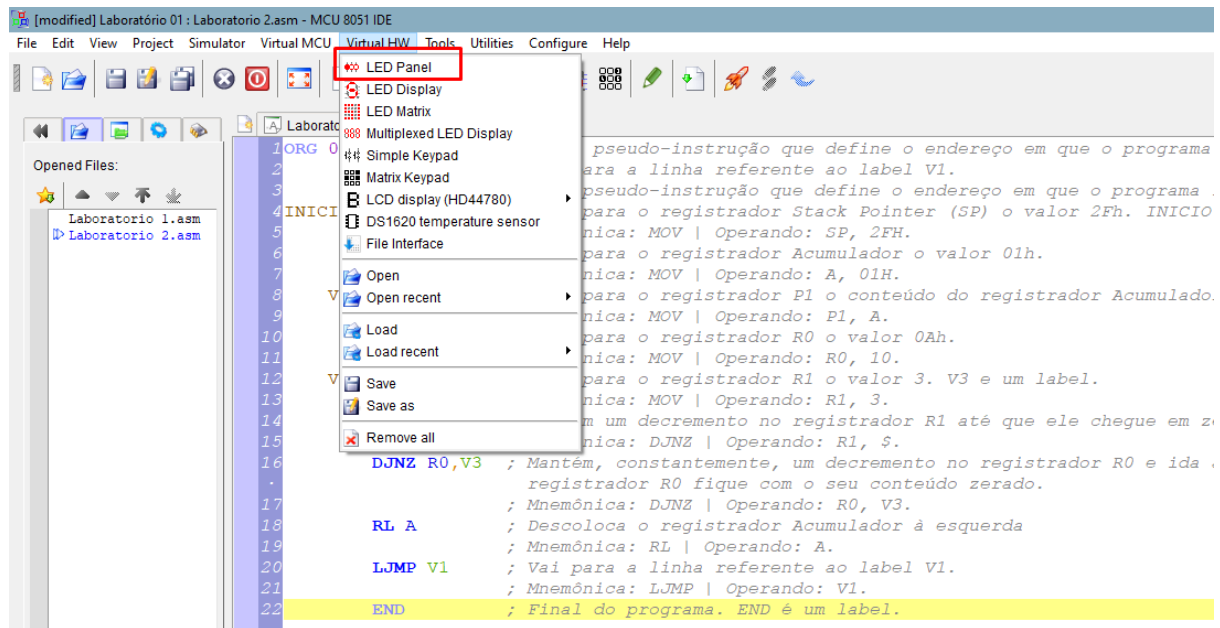
- Simulador MCU 8051 IDE.

3.4. PROCEDIMENTOS

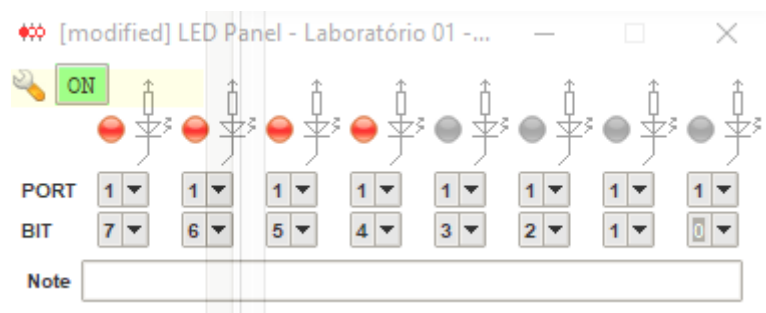
3.4.1. LIGAÇÕES DO PAINEL DE LED

Para iniciar o experimento é necessário fazer a ligação das portas e seus respectivos bits no painel de led's do simulador.

Localização do painel de led's:



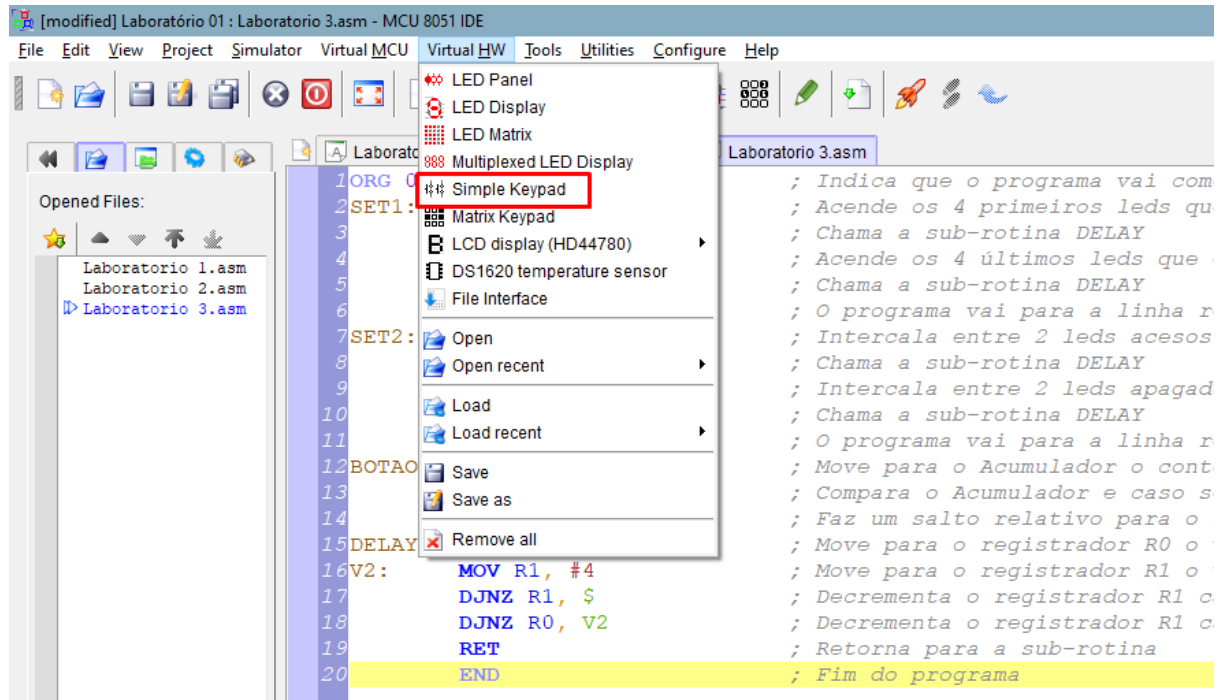
Para configurar o painel, basta ativá-lo e relacionar cada um dos oito led's a uma porta e a um bit do microcontrolador como mostra a imagem a seguir:



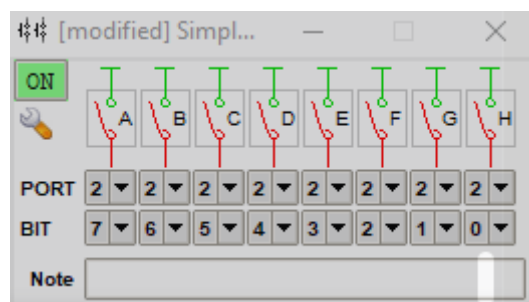
3.4.2. LIGAÇÃO DO PAINEL DE TECLAS

Para iniciar o experimento é necessário fazer a ligação das portas e seus respectivos bits no painel de teclas do simulador.

Localização do painel de teclas:



Para configurar o painel, basta ativá-lo e relacionar cada um dos oito led's a uma porta e a um bit do microcontrolador como mostra a imagem a seguir:



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.

- MOV P1, #00001111b

Move para o registrador P1 um determinado valor.

- LCALL DELAY

Chama a sub-rotina DELAY.

- INICIO: MOV SP,#2FH

Move para o registrador SP um determinado valor.

- MOV P1, #11110000b

Move para o registrador P1 um determinado valor.

- LJMP BOTAO

O programa vai para a linha referente ao label BOTAO

- MOV P1, #00110011b

Move para o registrador P1 um determinado valor.

- MOV P1, #11001100b

Move para o registrador P1 um determinado valor.

- MOV A, P2

Move para o Acumulador o conteúdo do registrador P2.

- CJNE A, #0FFH, SET2

Compara o Acumulador e caso seja diferente vai para o label SET2.

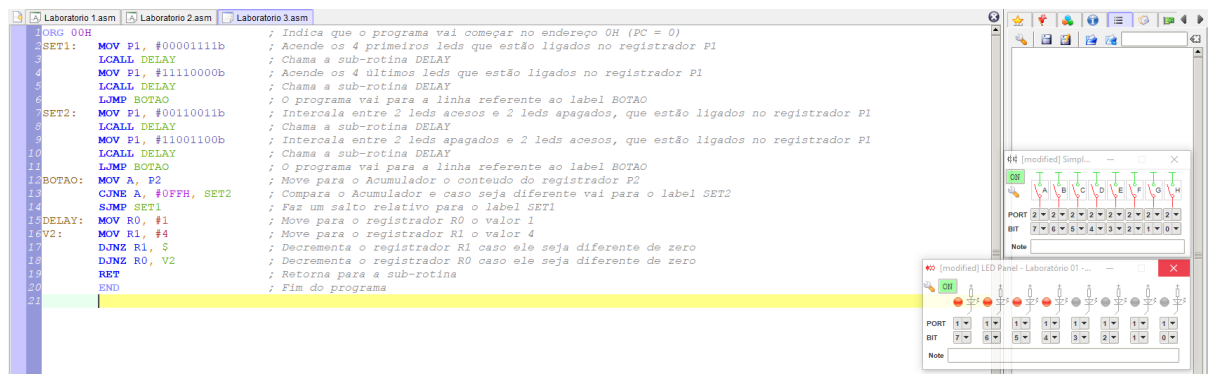
- SJMP SET1

Faz um salto relativo para o label SET1.

- **MOV R0, #1**
Move para o registrador R0 o valor 1.
- **MOV R1, #4**
Move para o registrador R1 o valor 4.
- **DJNZ R1, \$**
Decrementa o registrador R1 caso ele seja diferente de zero
- **DJNZ R0, V2**
Decrementa o registrador R1 caso ele seja diferente de zero
- **RET**
Retorna para a sub-rotina
- **END**
Define o final do programa.

3.4.3. FUNCIONAMENTO DO PROGRAMA

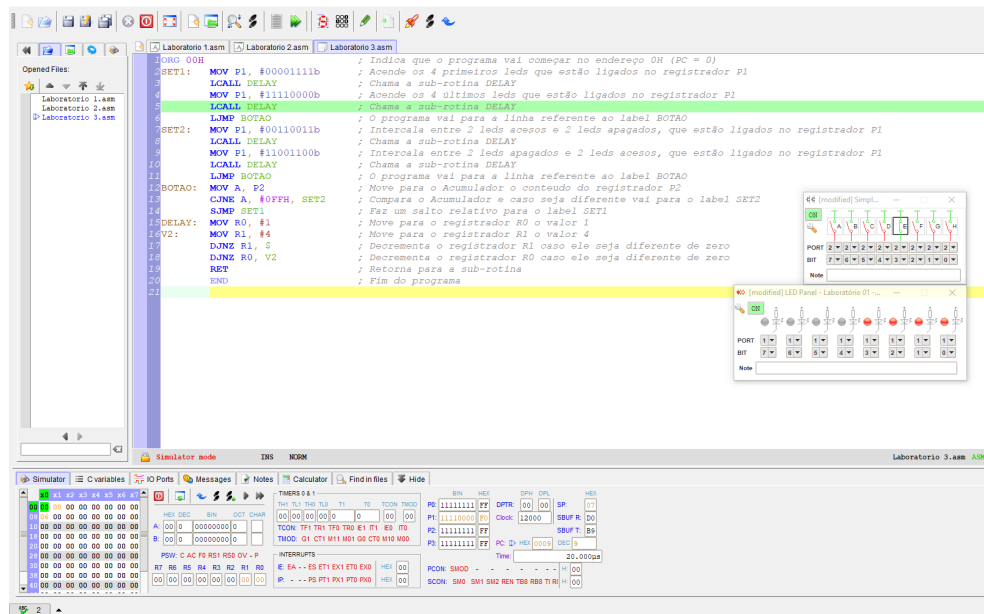
Aqui estará presente toda a descrição detalhada de cada linha do programa utilizado como exemplo, apresentando em cada linha seu respectivo funcionamento comentado.



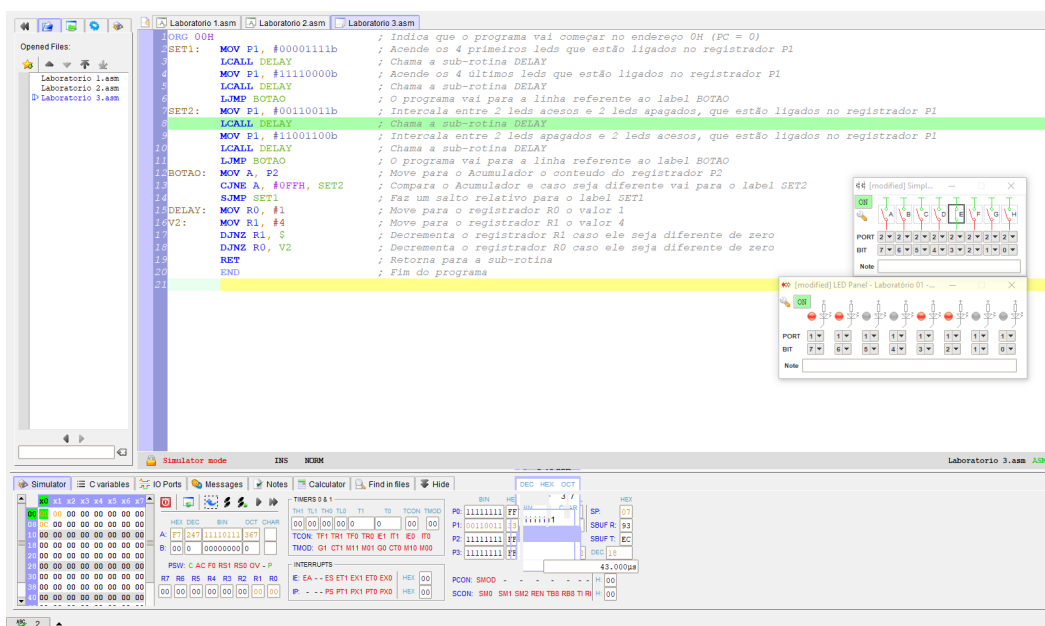
Comentários e execução do código acima:

No primeiro momento do programa, ao executar as duas primeiras linhas, o painel de led's irá

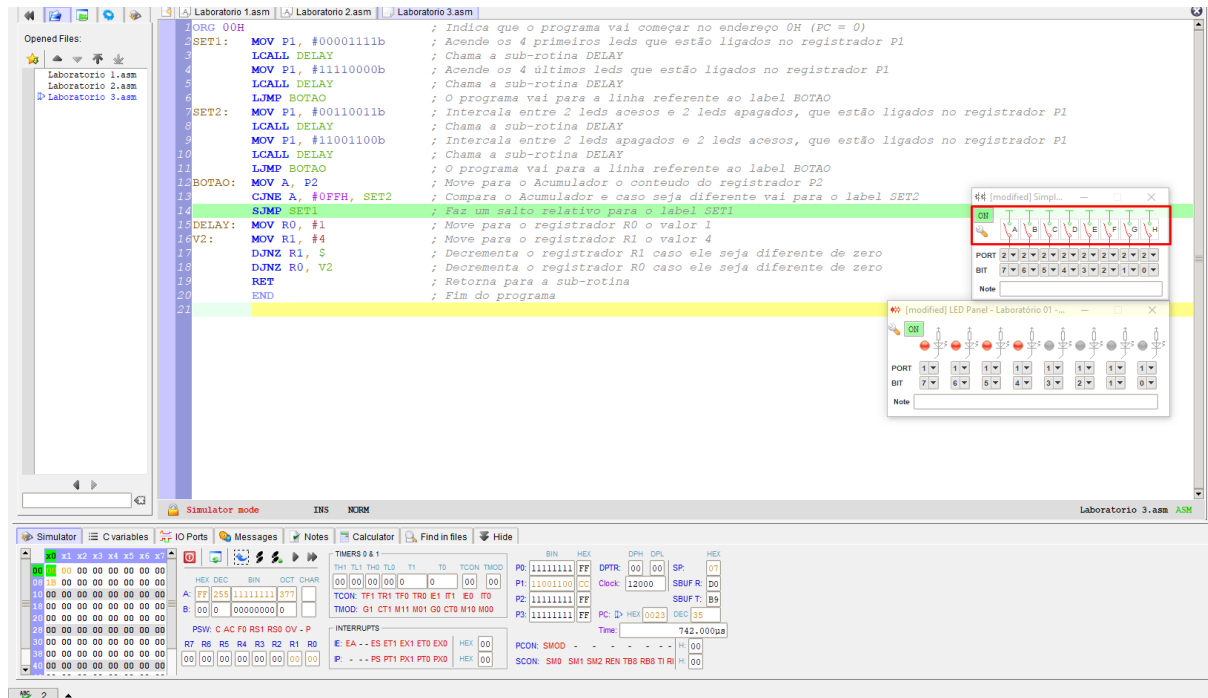
acender conforme descrito no comentário (linha 02). Segundamente o programa irá cair na sub-rotina definida no label “DELAY” que fará com que a próxima mudança no painel de led’s ocorra depois de um intervalo de tempo definido. Então o programa irá voltar para a sub-rotina que irá inverter o estado dos led’s ligados no registrador P1. Por fim, ao executar a linha 06 (LJMP BOTAO), será possível ter acesso ao label BOTAO.



No segundo momento do programa, ao chegar no label SET2, ocorre o comando MOV P1 #00110011b que tem como função alterar o padrão dos led’s acesos e desligados, em seguida a sub-rotina DELAY é acionada novamente, após o fim da sub-rotina o programa executa o comando MOV P1 #11001100b e vai novamente para a sub-rotina. Por fim, ao chegar no comando LJMP BOTAO, o programa irá acessar a posição do programa definida pelo label BOTAO.

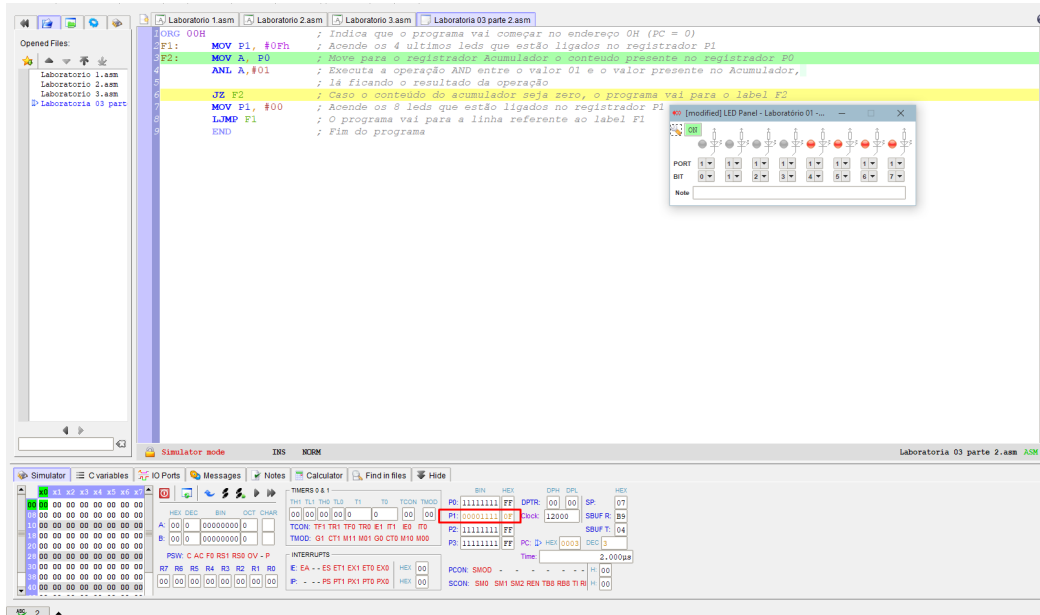


No terceiro momento, é possível ver o booleano do programa no label BOTAO (define a alternância de led's 4 em 4 ou 2 em 2. Primeiramente o programa pega o conteúdo presente no registrador P2 e o move para o registrador Acumulador. Então se o dado que tiver sido enviado for diferente de 0FFH, ou seja caso qualquer botão do teclado virtual tenha sido pressionado, o que vai deixar o valor diferente. A instrução “CJNE, A, SET2” faz com que o programa vá até o label definido como “SET2”, caso contrário, será passado para a instrução “SJMP SET1” que vai para o local do programa definido por “SET1”.

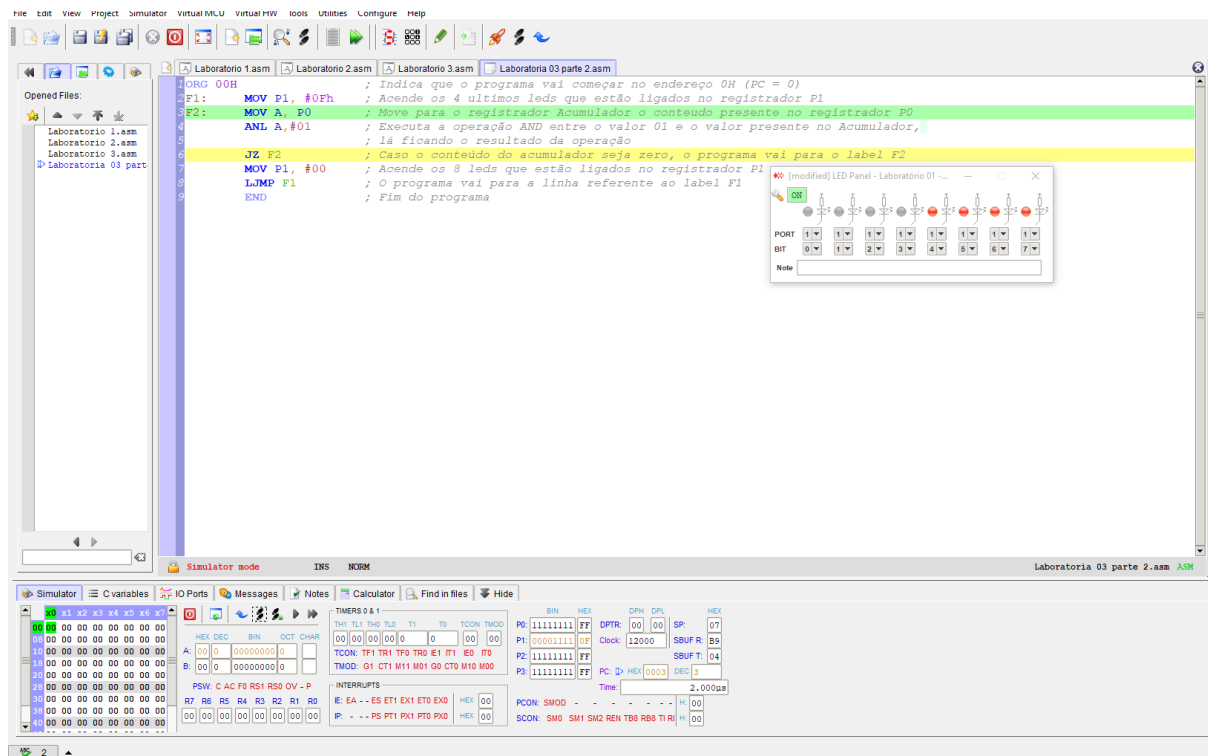


3.4.4. FUNCIONAMENTO DO PROGRAMA FORNECIDO

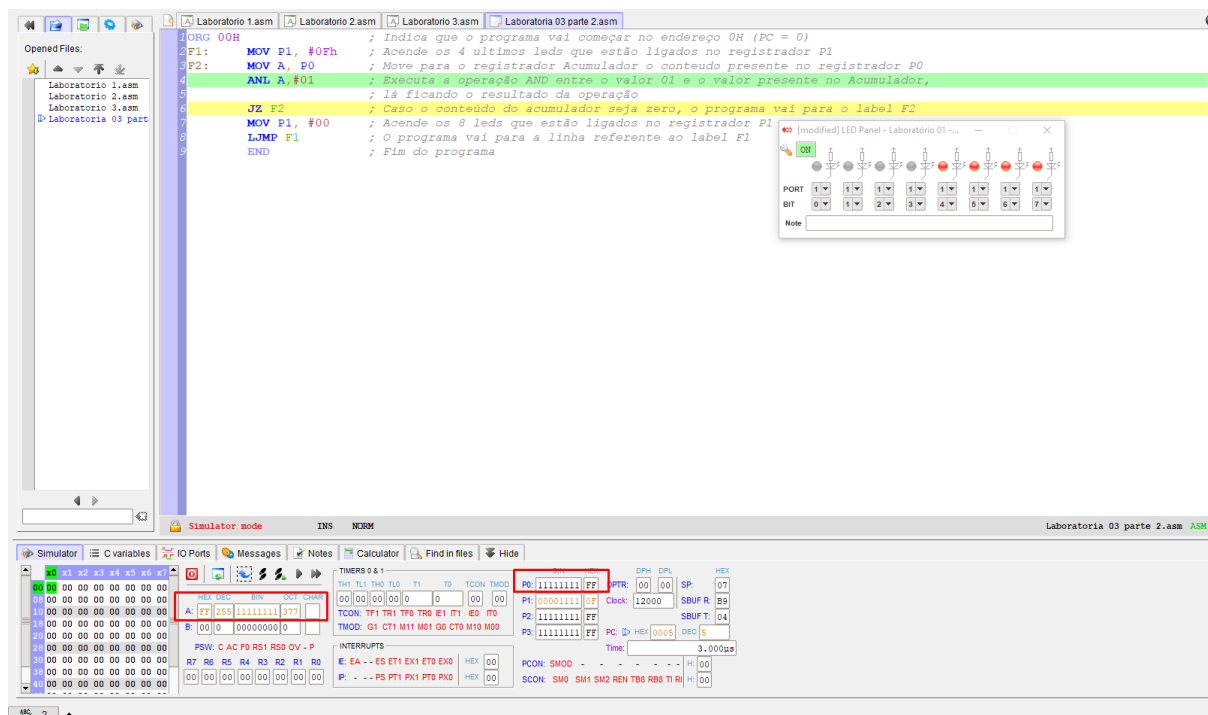
Segue abaixo o programa fornecido na descrição da atividade:



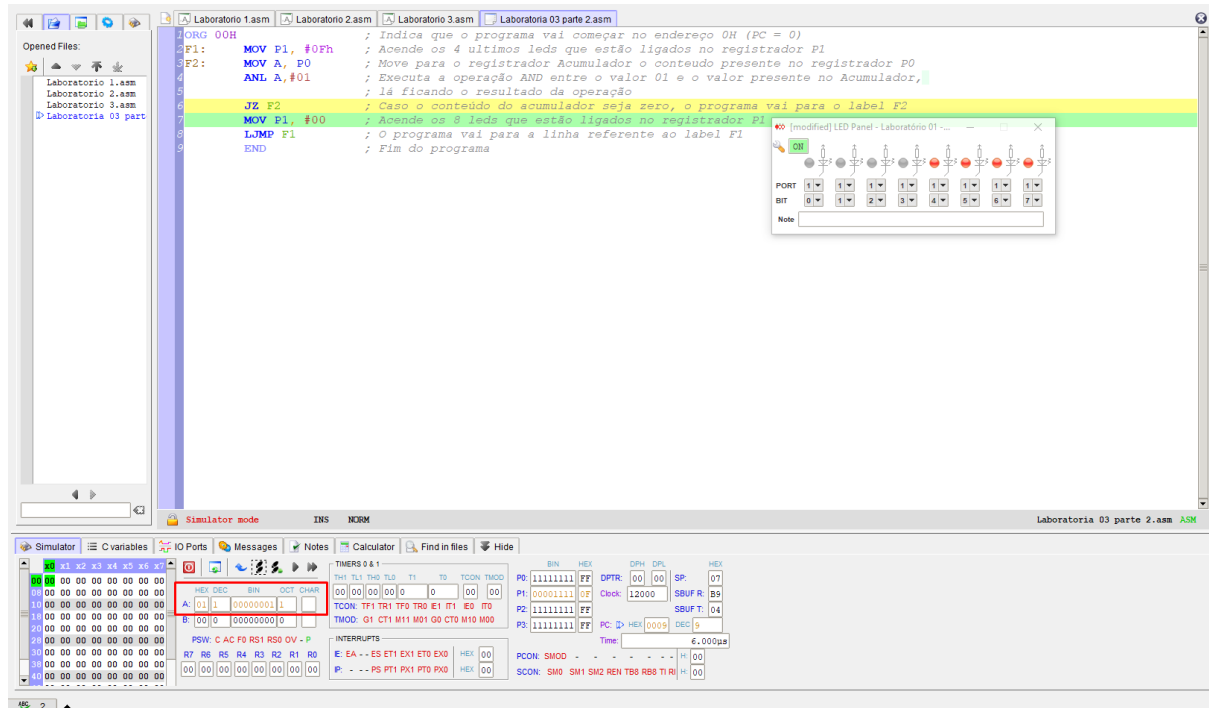
No primeiro momento, o programa inicia movendo o valor 00001111b para o registrador P1, que por sua vez tem como função ligar ou desligar os leds do painel. Quando iniciado é possível notar que os 4 últimos led's foram acesos, devido ao comando.



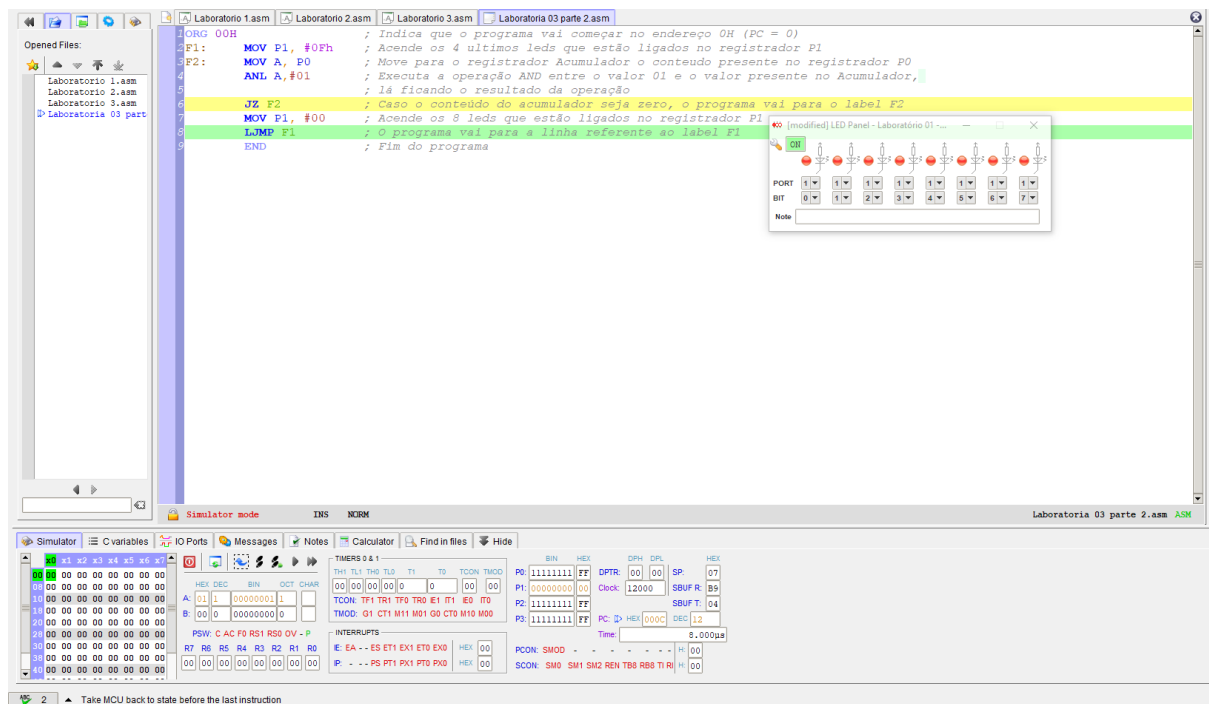
Na segunda linha é possível notar que o programa move para o registrador Acumulador o conteúdo presente no registrador P0, deixando o mesmo com 11111111b.



Na terceira e quarta linha, após a execução da operação AND entre o conteúdo presente no registrador Acumulador e o número 01 foi possível notar que o resultado da operação foi armazenado no próprio registrador Acumulador.

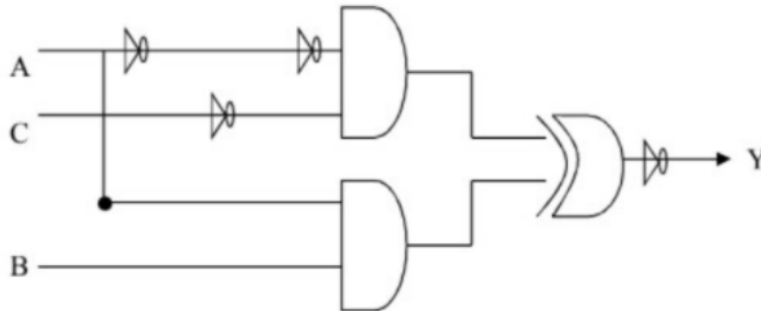


Na última linha de comando, o programa vai para o label definido como F1, após acender todos os led's do painel.



3.4.5. IMPLEMENTAÇÃO DO CIRCUITO LÓGICO

- ✓ Considere o **Y** para ligar um LED e as entradas (**A**, **B** e **C**) como sendo chaves.



```
1      ORG 00h
2TEST1:
3      JNB P0.0,TEST1
4TEST2: JNB P0.1,TEST3
5      JNB P0.2,LIGA
6TEST3: JB  P0.1,DESLIGA
7      JNB P0.2,DESLIGA
8      LJMP TEST1
9LIGA:  MOV P1, #00
10     LJMP TEST1
11DESLIGA:MOV P1, #01
12     LJMP TEST1
13     END
```

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, o teclado virtual e a atuação dos registradores entre si.

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 do teclado virtual.

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/0Bwjlecok7TpyfnAtQmx6bE4yZ043amNsbnRxMkF4UFfpWVZhWmRfeVBSeHRRVi1xRzNOZnM?resourcekey=0-WmQ6S1i6hLYyCoGBLbMeGw>. Acesso em: 01 ago. 2021