

## EEL7030 Microprocessadores – Laboratório 2

### Prof. Raimes Moraes

### Compilação de Código Assembly

Como pôde ser visto na aula anterior, a inserção direta de código na memória do simulador do 8051 é trabalhosa, principalmente devido à necessidade de se consultar o manual do 8051 para obter o código binário que o processador interpreta. Existem ferramentas de software para simplificar tal tarefa.

É possível criar arquivo texto (arquivo fonte) utilizando os mnemônicos do 8051 e diretivas para orientar o compilador (Apêndice B da apostila). Compiladores têm a tarefa de proceder a interpretação destes arquivos e gerar novo arquivo com a extensão .hex. Há aplicativos que realizam a transferência do arquivo .hex do microcomputador para a memória de programa dos microcontroladores. Estes programas atuam, geralmente, em conjunto com circuitos eletrônicos que permitem a gravação do código na memória de programa (OTPROM, EPROM, EEPROM).

O arquivo .hex pode ser aberto no *notepad* possuindo formato similar ao apresentado na Tabela 1.

Tabela 1 – Exemplo de conteúdo de arquivo hex resultante da compilação de arquivo fonte em Assembly.

```
:102000003180203E1B30211020FB3A1020C30A20D3
:0120100000CF
:0320CE0034FBC917
:00000001FF
```

Cada campo da primeira linha do arquivo .hex foi destacado com diferentes cores para permitir que se faça a sua correspondência com a Tabela 2.

Tabela 2 – Função de cada campo de linha do arquivo hex resultante da compilação de arquivo fonte em Assembly.

Campo		Nro. De Nibbles	Descrição (Valores hexadecimais)
1	Caractere inicial	1	":"
2	Nro de Bytes	2	Nro de bytes no campo de dados
3	Endereço	4	Endereço para carregamento dos dados na memória
4	Tipo	2	00 (Linha de dados) ou 01 (Linha de encerramento)
5	Dados	2	0 a n bytes de código ou de dados. n é geralmente menor ou igual a 32 (20H).
6	Checksum	2	Byte menos significativo (LSB) do complemento de dois da soma dos valores representados pelos bytes da linha, excetuando o caractere inicial e o próprio checksum.

A partir da Tabela 3, tem-se que a linha de encerramento do arquivo .hex é composta pelos seguintes caracteres:

: Caractere inicial  
00 Número de dados.  
00 Campo de endereços: MSB  
00 Campo de endereços: LSB  
01 Linha de encerramento.  
FF Checksum da linha de encerramento.

Crie projeto no compilador Keil que contenha arquivo texto (extensão .asm) com o conteúdo abaixo. Compile e observe o arquivo .hex gerado. Simule o programa no ambiente Keil.

```

; Programa    FIRST.asm
INICIO EQU    0H
                ORG    INICIO
                MOV    02H,#3
VOLTA:          MOV    A,R2
                ADD    A,32H
                MOV    32H,A
                JMP    VOLTA
                END

```

Tendo compreendido o procedimento, crie um novo projeto com o programa abaixo, compile-o e simule-o no Keil.

```

;Programa    ADDVECT.asm
RESET EQU    0H
VETOR EQU    60H
                ORG    RESET        ; PC = 0000H ao se resetar o 8051
                MOV    DPTR,#NRO    ; endereco nro parcelas a ser somado
                MOV    A,#0
                MOVC    A,@A+DPTR
                MOV    R1,A          ; R1 = nro parcelas a ser somado
                MOV    DPTR,#DADOS  ; end. vetor de dados a ser somado
                MOV    R2,#0        ; guarda resultado das somas realizadas
                MOV    R0,#0        ; especifica parcela a ser lida do vetor de dados
VOLTA:          MOV    A,R0
                MOVC    A,@A+DPTR    ; le parcela
                ADD    A,R2
                MOV    R2,A
                INC    R0
                DJNZ   R1,VOLTA
FIM:            JMP    FIM

                ORG    VETOR
NRO:            DB    03H
DADOS:          DB    01H,03H,05H,36H,0DAH,0E2H
                END

```

Exercícios:

Modifique o programa anterior para que:

- 1) o resultado seja salvo na posição de memória externa x:01.
- 2) o programa seja encerrado caso o nro. de parcelas a ser somado seja 0.
- 3) somar mais parcelas. Neste caso, o resultado pode exceder FFh; armazene o byte mais significativo da soma na posição de memória

x:00. Sugestão: utilizar a instrução ADDC, conforme o trecho de código abaixo.

```
....  
MOV    R3,#0      ; armazena byte mais significativo da soma  
VOLTA: MOV    A,R0  
        MOVC   A,@A+DPTR ; le parcela  
        ADD    A,R2  
        MOV    R2,A  
        MOV    A,#0  
        ADDC   A,R3  
        MOV    R3,A  
        INC    R0  
        DJNZ   R1,VOLTA  
...
```

- 4) Fazer um programa que escreva os números de 1 a 5, em posições consecutivas de memória, na memória interna a partir de 50H e na memória externa, a partir de 2200H.
- 5) Fazer um programa que copie os dados da região de memória externa de 2100H a 210FH para a região de memória externa que se inicia em 2300H. Inicialize os conteúdos destas posições de memória com valores quaisquer no início do programa.