

Sistemas Microcontrolados

Aula 4B – Microcontrolador PIC16F628 Linguagem de Máquina

Prof. Esp. Pedro Luís Antonelli
Anhanguera Educacional



Cronograma de Aulas

Semana n.º	Tema
1	Apresentação da Disciplina. Introdução à Linguagem de Máquina.
2	Estrutura interna PIC 16F628A
3	Memória de programa e memória de dados. Conceituações e utilizações.
4	Introdução a registradores. Conceituação, classificação
5	Registradores específicos: contador de programa, pilha e registradores de funções especiais (SFR). Conceito e funcionamento. Exemplos.
6	Registradores TRISA, TRISB, PORTA, PORTB
7	Introdução a Interrupções. Conceito, funcionamento, categorias, utilizações
8	Introdução a Linguagem de máquina. Conceituação. Instruções em binário e hexadecimal. Exemplos e exercícios.
9	Atividades de Avaliação.
10	Introdução a Programação em linguagem Assembly.
11	Programação em linguagem Assembly.
12	Programação em linguagem Assembly.
13	Programação em linguagem Assembly.
14	Aula pratica em laboratório de hardware.
15	Comandos e funções em linguagem C para manipulação de registradores. Revisão de operadores lógicos e aritmético em C. Diretivas de compilação. Retornos de funções,
16	Continuação da abordagem da linguagem C
17	Aula pratica em laboratório de hardware.
18	Prova Escrita Oficial
19	Exercícios de revisão.
20	Prova Substitutiva



Formato das Instruções

As instruções em Linguagem de Máquina (Assembly) segue o padrão:

LABEL: OPERAÇÃO OPERANDO(S) ; COMENTÁRIOS

Observações:

- O campo **LABEL** é facultativo e indica posições particulares no programa;
- O campo **OPERAÇÃO** (OP CODE) indica qual a operação a ser realizada;
- O campo **OPERANDO(S)** existirá se for necessário à instrução;
- O caracter “ ; ” indica ao montador que se trata de um comentários

Convenções

As seguintes convenções serão utilizadas, pois são reconhecidas pelos **Montadores Assembly MPASM** e **MPASWIN** que podem ser utilizados na linha de comando do Windows ou através da **IDE MPLAB**.

Campo :	Descrição:
f	registrador (endereço) de 00H a 0E0H
w ou W	registro W (acumulador)
b	bit utilizado pela operação (de 0 a 7)
k	contante (literal) ou rótulo (label)
d	destino do resultado
	- se d = 0 , o resultado será armazenado em W;
	- se d = 1 , o resultado será armazenado no próprio endereço (padrão).

Notação para constantes (Literais)

Por padrão as constantes (literais) são expressas em **HEXADECIMAL**, mas podem ser explicitamente declaradas nas formas:

0x12 ; número hexadecimal 12

12H ; número hexadecimal 12

0AFH ; número hexadecimal AF

0x0AF ; número hexadecimal AF

Constantes **DECIMAIS** podem ser declaradas nas forma:

D'20' ; número decimal 20

d'20' ; número decimal 20

Constantes **BINÁRIAS** podem ser declaradas nas forma:

B'01011101' ; número binário 01011101

b'01011101' ; número binário 01011101

MOVLW k

Descrição: Carrega o literal em W (acumulador)

Sintaxe: MOVLW k

Flags afetados: nenhum

Exemplo: MOVLW 0x5A ; carrega o acumulador (W) com 5AH

MOVWF f

Descrição: Carrega o conteúdo do acumulador (W) em um registrador (indicado por seu endereço).

Sintaxe: MOVWF f

Flags afetados: nenhum

Exemplo:

MOVLW	0x5A ; carrega o acumulador (W) com 5Ah
MOVWF	29H ; carre o endereço 29H com o conteúdo de W

MOVF f, d

Descrição: Carrega o conteúdo de um registrador (indicado por seu endereço) no acumulador (W) ou em seu próprio endereço.

Sintaxe: MOVF f, d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados: ' Z ' (Flag de Zero)

Exemplo 1: MOVF 29H ; recarrega o conteúdo do endereço 29H e
; altera o Flag de Zero, se seu conteúdo for 0H.

Exemplo 2 : MOVF 29H,0 ; carrega o conteúdo do endereço 29H em W
; altera o Flag de Zero, se seu conteúdo for 0H.

ADDLW k

Descrição: Soma a constante k com o conteúdo acumulador (W) e armazena o resultado no acumulador (W).

Sintaxe: ADDLW k

Obs. - k pode variar de 0H a FFH (0 a 255).

Flags afetados:

- ' Z ' (Flag de Zero)
- ' C ' (Flag de “vai um” do 7º bit)
- ' DC ' (Flag de “vai um” do 3º bit)

Exemplo :

```
MOVLW 10H ; carrega W com o valor 10H
ADDLW 15H ; soma o valor de W com 15H
          ; W fica com o valor 25H
```

ADDFW f , d

Descrição: Soma o conteúdo acumulador (W) com o conteúdo do endereço de f.

Sintaxe: ADDFW f , d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados:

- ‘ Z ‘ (Flag de Zero)
- ‘ C ‘ (Flag de “vai um” do 7º bit)
- ‘ DC ‘ (Flag de “vai um” do 3º bit)

Exemplo 1: ADDFW 15H ; soma W com o conteúdo do endereço 15H
; e coloca o resultado no endereço 15H

Exemplo 2: ADDFW 15H , 0 ; soma W com o conteúdo do endereço 15H
; e coloca o resultado em W

SUBLW k

Descrição: Subtrai o valor do conteúdo acumulador (W) do valor da constante k, e armazena o resultado no acumulador (W), ou seja $W \leftarrow k - W$.

Sintaxe: SUBLW k

Obs. - k pode variar de 0H a FFH (0 a 255).

Flags afetados:

- ' Z ' (Flag de Zero)
- ' C ' (Flag de "vai um" do 7º bit)
- ' DC ' (Flag de "vai um" do 3º bit)

Exemplo 1 :

```
MOVLW 12H ; carrega W com o valor 12H
SUBLW 20H ; subtrai o valor de W
          ; W fica com o valor 08H
          ; flag C = 1 ( positivo )
```

Exemplo 2 :

```
MOVLW 20H ; carrega W com o valor 20H
SUBLW 12H ; subtrai o valor de W
          ; W fica com o valor- 08H
          ; ; flag C = 0 ( negativo )
```

SUBFW f, d

Descrição: : Subtrai o valor do conteúdo acumulador (W) do valor do endereço f, e armazena o resultado no acumulador (W), ou seja $W \leftarrow f - W$.

Sintaxe: SUBFW f, d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados:

- ' Z ' (Flag de Zero)
- ' C ' (Flag de “vai um” do 7º bit)
- ' DC ' (Flag de “vai um” do 3º bit)

Exemplo 1 :

MOVLW	12H	; carrega W com o valor 12H
MOVWF	10H	; carrega o valor de W no endereço 10H
MOVLW	01H	; carrega W com o valor 01H
SUBLW	10H	; subtrai o valor de W do valor do endereço 10H
		; o resultado é armazenado no endereço 10H

Exemplo 1 :

MOVLW	12H	; carrega W com o valor 12H
MOVWF	10H	; carrega o valor de W no endereço 10H
MOVLW	01H	; carrega W com o valor 01H
SUBLW	10H,0	; subtrai o valor de W do valor do endereço 10H
		; o resultado é armazenado em W

BCF f , b

Descrição: Zera (nível lógico “0”) o bit “b” no endereço de f.

Sintaxe: BCF f , b

Obs. - b pode variar de 0 a 7.

Flags afetados: nenhum

Exemplo : BCF 10H , 3 ; zera o bit 3 no endereço 10H

BSF f , b

Descrição: Seta (nível lógico “1”) o bit “b” no endereço de f.

Sintaxe: BSF f , b

Obs. - b pode variar de 0 a 7.

Flags afetados: nenhum

Exemplo : BCF 10H , 3 ; seta o bit 3 no endereço 10H

Descrição: Desvia, incondicionalmente, a execução do programa para o endereço k.

Sintaxe: GOTO k

Flags afetados: nenhum

Exemplo : GOTO 55H ; desvia para o endereço 55H

BTFSC f, b

Descrição: Pula a instrução seguinte se o bit “b” no endereço de for zero (“0”).

Sintaxe: BTFSC f, b

Obs. - b pode variar de 0 a 7.

Flags afetados: nenhum

Exemplo : BTFSC 10H, 3 ; testa o bit 3 no endereço 10H
GOTO label1 ; se for “1” executa a instrução (“GOTO label 1”)
GOTO label2 ; se for “0” salta para a próxima instrução (“GOTO label 2”)

BTFSS f, b

Descrição: Pula a instrução seguinte se o bit “b” no endereço de for um (“1”).

Sintaxe: BTFSC f, b

Obs. - b pode variar de 0 a 7.

Flags afetados: nenhum

Exemplo : BTFSC 10H, 3 ; testa o bit 3 no endereço 10H
GOTO label1 ; se for “0” executa a instrução (“GOTO label 1”)
GOTO label2 ; se for “1” salta para a próxima instrução (“GOTO label 2”)

CALL k

Descrição: Salva o endereço da **próxima instrução** (PC + 1) na Pilha (Stack) e desvia a execução do programa no endereço k.

Sintaxe: BTSFC f , b

Flags afetados: nenhum

Exemplo : CALL 57h ; desvia para o endereço 57h

RETURN

Descrição: Retorna a execução do programa para o endereço recuperado do topo da Pilha (Stack).

Sintaxe: RETURN

Flags afetados: nenhum

Exemplo : RETURN ; desvia para o endereço recuperado do topo da pilha

RETLW k

Descrição: Retorna a execução do programa para o endereço recuperado do topo da Pilha (Stack) e carrega o acumulador (W) com valor de k.

Sintaxe: RETLW k

Flags afetados: nenhum

Exemplo : RETLW 18H ; desvia para o endereço recuperado do topo da pilha
; e carrega o acumulador (W) com o valor 18H

RETFIE

Descrição: Retorna a execução do programa para o endereço recuperado do topo da Pilha (Stack) após a execução da rotina de tratamento de uma interrupção.

Sintaxe: **RETFIE**

Flags afetados: O bit GIE é setado (sinalizador de interrupção)

Exemplo : **RETFIE ; desvia para o endereço recuperado do topo da pilha armazenado no
 ; momento do atendimento da interrupção;**

CLRF f

Descrição: Zera o conteúdo do registrador no endereço f.

Sintaxe: CLRF f

Flags afetados: ' Z ' (Flag de Zero)

Exemplo : CLRF 13H ; zera o conteúdo do endereço 13H

Descrição: Zera o conteúdo do acumulador (W).

Sintaxe: CLRW

Flags afetados: ‘ Z ‘ (Flag de Zero)

Exemplo : CLRW ; zera o conteúdo do acumulador (W)

COMF f , d

Descrição: Complementa (“inverte”) os bits do endereço f.

Sintaxe: COMF f , d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados: ‘ Z ‘ (Flag de Zero)

Exemplo 1: ACONF 15H ; inverte os bits do endereço 15H
; e o resultado é colocado em 15H

Exemplo 2: ADDFW 15H , 0 ; inverte os bits do endereço 15H
; e o resultado é colocado em W

DECF f , d

Descrição: Diminui de 1 unidade o valor do conteúdo do endereço f.

Sintaxe: DECF f , d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados: ' Z ' (Flag de Zero)

Exemplo 1: DECF 15H ; decrementa o conteúdo do endereço 15H
; e o resultado é colocado em 15H

Exemplo 2: DECF 15H , 0 ; decrementa o conteúdo do endereço 15H
; e o resultado é colocado em W

DECFSZ f , d

Descrição: Diminui de 1 unidade o valor do conteúdo do endereço f e salta a próxima instrução se conteúdo de f for zero.

Sintaxe: DECFSZ f , d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados: nenhum

Exemplo 1: DECFSZ 15H ; decrementa o conteúdo do endereço 15H
; e o resultado é colocado em 15H
; e salta a próxima instrução se o conteúdo for zero

Exemplo 2: DECFSZ 15H , 0 ; decrementa o conteúdo do endereço 15H
; e o resultado é colocado em W
; e salta a próxima instrução se o conteúdo for zero

INCF f , d

Descrição: Aumenta de 1 unidade o valor do conteúdo do endereço f.

Sintaxe: INCF f , d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados: ' Z ' (Flag de Zero)

Exemplo 1: INCF 15H ; incrementa o conteúdo do endereço 15H
; e o resultado é colocado em 15H

Exemplo 2: INCF 15H , 0 ; incrementa o conteúdo do endereço 15H
; e o resultado é colocado em W

INCFSZ f , d

Descrição: Aumenta de 1 unidade o valor do conteúdo do endereço f e salta a próxima instrução se conteúdo de f for zero.

Sintaxe: INCFSZ f , d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados: nenhum

Exemplo 1: INCFSZ 15H ; incrementa o conteúdo do endereço 15H
; e o resultado é colocado em 15H
; e salta a próxima instrução se o conteúdo for zero

Exemplo 2: INCFSZ 15H , 0 ; incrementa o conteúdo do endereço 15H
; e o resultado é colocado em W
; e salta a próxima instrução se o conteúdo for zero

Descrição: Reseta o registro do Watch Dog Timer (“temporizador de cão de guarda”).

Sintaxe: CLRWDT

Observação:

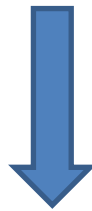
Se o circuito do Watch Dog Timer estiver habilitado, essa instrução deve ser executada em intervalos de tempo menor que o “tempo de estouro” (**típicamente 18 ms**), afim de que o sistema não seja forçado a reiniciar automaticamente.

Flags afetados:

- ‘ TO ‘ (Flag de “Timer-Out”)
- ‘ PD ‘ (Flag de “Power Down”)

Exemplo : INICIO: xxxx ; início do programa

; corpo do programa



CLRWDT ; reseta Watch Dog Timer

GOTO INICIO ; salta para o início do programa

ANDLW k

Descrição: Realiza a operação lógica “AND” (“ E “)entre uma constante k e o conteúdo de W,
e armazena o resultado em W.

Sintaxe: ANDLW k

Obs. - k pode variar de 0H a FFH (0 a 255).

Flags afetados: ‘ Z ‘ (Flag de Zero)

Exemplo : ANDLW 10H ; realiza a operação AND entre W com o valor 10H

ANDFW f, d

Descrição: Realiza a operação lógica “**AND**” entre conteúdo de W e o conteúdo de f.

Sintaxe: ANDFW f, d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados: ‘ Z ‘ (Flag de Zero)

Exemplo 1: ANDFW 15H ; realiza a operação AND entre W e o conteúdo
; de 15H e o resultado é colocado em 15H

Exemplo 2: ANDFW 15H , 0 realiza a operação AND entre W e o conteúdo
; de 15H e o resultado é colocado em W

IORLW k

Descrição: Realiza a operação lógica “OR” (“ **OU Inclusivo** “) entre uma constante k e o conteúdo de W, com o resultado armazenado em W.

Sintaxe: IORLW k

Obs. - k pode variar de 0H a FFH (0 a 255).

Flags afetados: ‘ Z ‘ (Flag de Zero)

Exemplo : IORLW 10H ; realiza a operação OR entre W com o valor 10H

IORFW f , d

Descrição: Realiza a operação lógica “**OR**” entre conteúdo de W e o conteúdo de f.

Sintaxe: IORFW f , d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados: ‘ Z ‘ (Flag de Zero)

Exemplo 1: IORFW 15H ; realiza a operação OR entre W e o conteúdo
; de 15H e o resultado é colocado em 15H

Exemplo 2: IORFW 15H , 0 realiza a operação OR entre W e o conteúdo
; de 15H e o resultado é colocado em W

XORFW f, d

Descrição: Realiza a operação lógica “XOR” (“ Ou exclusivo “) entre conteúdo de W e o conteúdo de f.

Sintaxe: XORFW f, d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Flags afetados: ‘ Z ‘ (Flag de Zero)

Exemplo 1: XORFW 15H ; realiza a operação XOR entre W e o conteúdo
; de 15H e o resultado é colocado em 15H

Exemplo 2: XORFW 15H , 0 realiza a operação XOR entre W e o conteúdo
; de 15H e o resultado é colocado em W

XORLW k

Descrição: Realiza a operação lógica “**XOR**” entre uma constante k e o conteúdo de W, com o resultado armazenado em W.

Sintaxe: XORLW k

Obs. - k pode variar de 0H a FFH (0 a 255).

Flags afetados: ‘ Z ‘ (Flag de Zero)

Exemplo : XORLW 10H ; realiza a operação XOR entre W com o valor 10H

NOP

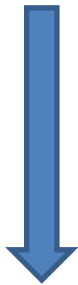
Descrição: Não realiza nenhuma operação, servindo apenas para que a CPU utilize um ciclo de máquina (4 pulsos do clock).

Sintaxe: NOP

Flags afetados: nenhum

Exemplo1 :

INICIO: NOP ; início do
; programa



GOTO INICIO ; salta para o
; início do
; programa

Exemplo 2 : (Clock 4 MHZ)

BSF 05H,3 ; seta pino RA4

NOP

NOP

NOP

NOP

NOP

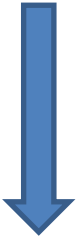
BSC 05h,3 ; reseta pino RA4
; após 5 µs

Descrição: Coloca a CPU no modo **SLEEP** (“dormindo “) sendo que o oscilador de clock pára e o Watch Dog Timer é desativado.

Sintaxe: SLEEP

Flags afetados: ‘ TO ‘ (Flag de “Timer-Out”)
‘ PD ‘ (Flag de “Power Down”)

Exemplo 1:

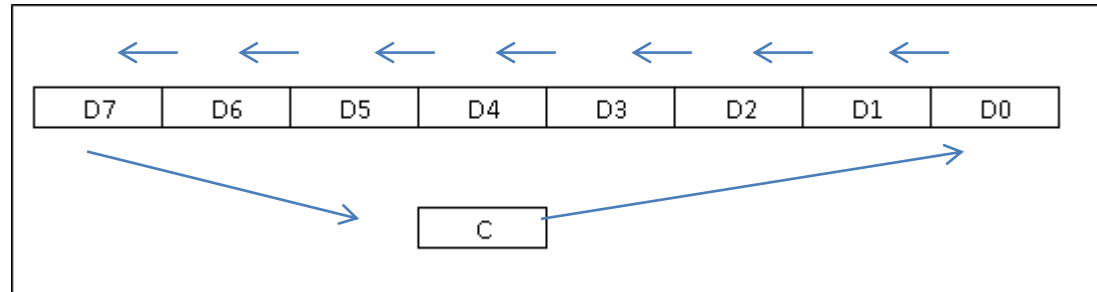


; processamento normal

SLEEP ; a partir desse ponto a CPU só voltará a “acordar”
; se uma evento externo (ntterupção) ocorrer.

RLF f, d

Descrição: Rotaciona à **esquerda**, o conteúdo do endereço de f de uma posição binária, passando pelo Flag “C” (“vai um”).



Sintaxe: RLF f, d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

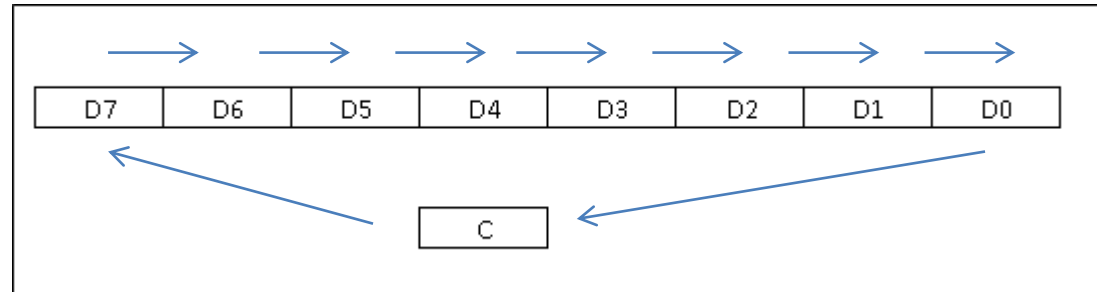
Flags afetados: C ‘ (Flag de “vai um” do 7º bit)

Exemplo 1 : RLF 15H ; rotaciona à esquerda o conteúdo do endereço 15H
; e coloca o resultado no endereço 15H

Exemplo 2 : RLF 15H,0 ; rotaciona à esquerda o conteúdo do endereço 15H
; e coloca o resultado em W

RRF f, d

Descrição: Rotaciona à **direita**, o conteúdo do endereço de f de uma posição binária, passando pelo Flag “C” (“vai um”).



Sintaxe: RLF f, d

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

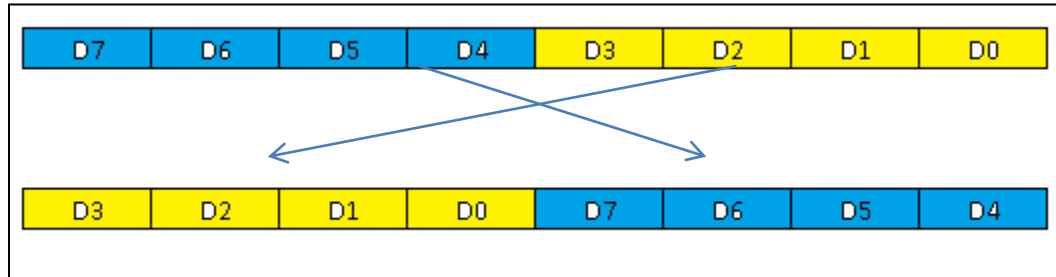
Flags afetados: C ‘ (Flag de “vai um” do 7º bit)

Exemplo 1 : RRF 15H ; rotaciona à esquerda o conteúdo do endereço 15H
; e coloca o resultado no endereço 15H

Exemplo 2 : RRF 15H,0 ; rotaciona à esquerda o conteúdo do endereço 15H
; e coloca o resultado em W

SWAPF f, d

Descrição: Realiza uma **troca** (“ swap “) entre os nibbles do da parte alta (“ High”) e da parte baixa (“Low ”) do conteúdo do endereço f.



Sintaxe: SWAP f

Flags afetados: nenhum

Obs.

- se d = 0, o resultado será armazenado em W;
- se d = 1, o resultado será armazenado no próprio endereço (padrão).

Exemplo 1 : SWAPF 15H ; troca o nibble baixo pelo alto no conteúdo do endereço 15H
; e coloca o resultado no endereço 15H

Exemplo 2 : SWAPF 15H,0 ; troca o nibble baixo pelo alto do conteúdo do endereço 15H
; e coloca o resultado em W

Bibliografia Básica Padrão

1) SOUZA, David Jose de. Desbravando o PIC : ampliado e atualizado para PIC 16F628A. 6ª ed. São Paulo: Érica, 2003.

Bibliografia Básica Unidade: Faculdade Anhanguera de Rio Claro (FRC)

1) ZANCO, Wagner da Silva. Microcontroladores PIC : Técnicas e Software e Hardware para Projetos de Circuitos Eletrônicos. 1ª ed. São Paulo: Érica, 2006.

2) SOUZA, David Jose de; LAVINIA, Nicolás César. Conectando o PIC16F877A : recursos avançados. 4ª ed. São Paulo: Érica, 2008.

Bibliografia Complementar: Faculdade Anhanguera de Rio Claro (FRC)

1) STALLINGS, W.. Arquitetura e Organização de Computadores. 1ª ed. São Paulo: Pearson - Prentice Hall, 2002.

2) NICOLOSI, Denys Emílio Campion. Microcontrolador 8051 : Detalhado. 8ª ed. São Paulo: Érica, 2003.

3) GIMENEZ, S.P.. Microcontroladores 8051. 2ª ed. São Paulo: Pearson, 2005.

4) HENNESSY, J. L.; PATTERSON, D. A.. Arquitetura de Computadores : Uma Abordagem Quantitativa. 4ª ed. Rio de Janeiro: Campus - Elsevier, 2008.

5) ZANCO, Wagner da Silva. Microcontroladores PIC : Técnicas de Software e Hardware para Projetos de Circuitos Eletrônicos. 2ª ed. São Paulo: Érica, 2008.