

# Projeto de Sistemas Microcontrolados

## Aula 03 - Estudo de microcontroladores PIC – Parte I

# Apresentação

Nesta aula, você vai iniciar o estudo de microcontroladores. Especificamente estudará uma família de microcontroladores de uso extremamente simples, fabricados por uma empresa chamada Microchip, denominados de microcontroladores PIC. Você vai aprender um pouco sobre as características gerais dos microcontroladores da família 16F, bem como sua arquitetura interna. Será dada especial atenção sobre a organização da memória de programa e sobre as principais interfaces periféricas presentes em sua organização. Isso facilitará a programação e o projeto de sistemas digitais com o uso destes dispositivos. Durante a aula, você irá assimilar que unidades básicas presentes na arquitetura PIC são comuns a todos os microcontroladores PIC, bem como terá informações introdutórias sobre duas principais ferramentas de desenvolvimento no ambiente dos microcontroladores PIC. Enfim, você vai começar a ter uma visão geral sobre os microcontroladores PIC e suas principais características.

## Objetivos

Ao final desta aula você será capaz de:

- Descrever as principais características dos microcontroladores PIC.
- Descrever as unidades básicas presentes numa arquitetura PIC.
- Entender o que são e para que servem o detector de *brown out* e o *Watchdog* bem como, caracterizar os diversos tipos de memória presentes nos microcontroladores PIC.
- Descrever a organização da memória de programa dos Microcontroladores PIC da família 16F.

# Estudo de microcontroladores PIC

Na aulas anteriores, você aprendeu o que são e quais as áreas de maior atuação dos microcontroladores. Nesta aula, você terá a oportunidade de conhecer em detalhes a arquitetura e o conjunto de instruções de uma família específica de microcontroladores, a família PIC 16F, da Microchip, uma das mais populares atualmente no mercado. Especificamente, estudaremos os microcontroladores 16F84, 16F628 e 16F877. De antemão, também serão apontadas características da família 18F já que, em um dos kits a ser usado, o microcontrolador é desta família.

## Visão geral sobre os microcontroladores PIC

Como citado anteriormente, PIC é a abreviatura usada para *Peripheral Interface Controller* (controlador de interfaces periféricas ou simplesmente controlador de periféricos) - linha de microcontroladores fabricada exclusivamente pela Microchip Technology

Dentre as suas principais características, estão:

- CPU RISC.
- Arquitetura Harvard.
- Estrutura *pipeline*.
- Formato de instruções ortogonal (qualquer elemento constituinte de sua arquitetura pode ser usado como fonte ou destino de dados).
- Arquitetura orientada a registros.
- Instruções de tamanho fixo (a maioria executável em um único ciclo).

Caso esteja em dúvida do que seja CPU RISC, arquitetura Harvard ou estrutura *pipeline*, retome as Aulas 1 e 2 para rever esses conceitos.

A Microchip divide seus microcontroladores em famílias, de acordo com o número de bits de suas instruções de programa. Assim sendo:

- A *Low Range* representa a família dos PIC com instruções de 12 bits (PIC 12X5xx, 16X5x e 16C505).
- A *Mid Range* representa a família dos PIC com instruções de 14 bits (PIC 12C6x, 16X5xx, 16X6xx, 16X7xxx, 16X8xx, 16X9xx).
- A *High Range* representa a família dos PIC com instruções de 16 bits (PIC 17Cxxx e 18Cxxx).

Em todas as famílias citadas, o processador opera com 8 bits de dados mudando apenas o comprimento da palavra de programa. Devido à mudança no comprimento das instruções, podemos observar valores pouco comuns nas especificações dos *datasheets* quanto à capacidade de memória de programa dos microcontroladores PIC. Por exemplo: para o PIC 12C509 é especificado 1.536 bytes de memória de programa. Tal informação equivale, no entanto, a 1.024 palavras de programa, já que suas instruções são de 12 bits. A conta é a seguinte:  $1.536 \times 8 / 12 = 1.024$ .

## Estudo de microcontroladores PIC - pt.2

Recentemente, a Microchip apresentou a sua mais nova linha de dispositivos, formando a quarta família de dispositivos - chamada de DsPIC - com largura de dados de 16 bits (em vez dos 8 bits das outras famílias) e largura de memória de programa de 24 bits, além de implementar diversas facilidades que os tornam dispositivos intermediários aos DSP. De fato, eles são chamados de DSC (Controladores Digitais de Sinais) numa analogia aos DSP.

O aumento no tamanho da palavra de programa possibilita um aumento no número de instruções: os PIC de 12 bits possuem 33 instruções, os de 14 bits, 35 instruções e os de 16 bits, até 77 instruções.

A arquitetura de construção dos PICs é otimizada para a execução de um pequeno conjunto de instruções, mas com velocidade muito alta. Podem-se obter velocidades de até 5 MIPS (Milhões de Instruções por Segundo) com um PIC da série 16 trabalhando a 20 MHz.

Outra característica importante da arquitetura PIC reside na semelhança e compatibilidade entre os diversos *chips*. Isso facilita a migração de uma MCU (unidade microcontroladora) para outra, pois os princípios básicos e a grande parte dos registradores não diferem entre si.

As unidades básicas presentes na arquitetura PIC (séries 12, 14, 16 e 18) são:

- Um registrador de trabalho (denominado de W).
- Um registrador de *flags* – sinalizadores ou bandeiras (denominado de STATUS).
- Um registrador de controle de interrupções (denominado de INTCON).
- Registradores de configuração de periféricos (denominados de OPTION\_REGx, onde x representa o número 1, 2, ..., dependendo da quantidade de registradores de configuração existentes).
- Registradores de acesso a dados de E/S (denominados de PORTx, onde x representa as letras A, B, C, D ou E, dependendo da quantidade de registradores de acesso a dados de E/S existentes). E/S é a abreviatura usada para designar dados de entrada ou de saída da porta.

# Atividade 01

1. Entre no site da Microchip <<http://www.microchip.com/>>, em produtos -> Microcontrollers and Microprocessors para ter uma visão ampla desses microcontroladores. Estando na página, clique nos links 8 bit, 16 bit e 32 bit e faça anotações das características apresentadas pelo fabricante para cada um destes grupos de microcontroladores.

## Estudo de microcontroladores PIC - pt.3

Todos os PIC são nomeados por **PICxyyzzzV - rr t/cc**, onde:

- xx determina a família (12, 14, 16, 17 ou 18).
- yy - determina o tipo de tecnologia (C para CMOS EPROM, HC para *High Speed*, CE para CMOS EEPROM, CR para CMOS ROM, HV para High Voltage, F para *Flash*, LF para *Low Power Flash* e LCR para *Low Power ROM*).
- zzz - determina o membro da família como, por exemplo: 508, 509, 54, 83, 84, 628, 877, dentre outros.
- V determina a versão do *chip* (branco para a primeira versão, A para a primeira revisão, B para a segunda revisão e C para a terceira revisão).
- rr determina a frequência máxima de operação.
- t determina a faixa de temperatura de operação (branco para faixa de 0 a 70° C, I para faixa de -40° a 85° C e E para a faixa de -40° a 125° C).
- cc determina o tipo de encapsulamento usado pelo fabricante.

O termo encapsulamento é usado para definir a proteção que envolve um circuito integrado. Sua principal função é garantir a conexão segura entre os circuitos e as placas físicas. Essa proteção é composta por pinos de metal, ligados ao *chip* e que fixam um envoltório retangular feito de metal, plástico ou de outros materiais.

No site da própria Microchip, no item *Design Support -> Quality -> Packaging Specifications*, você terá informações de todos os tipos de encapsulamentos usados nos *chips*.

O grande diferencial desses dispositivos é o seu preço, viabilizando o desenvolvimento de projetos e a implementação de sistemas digitais muito baratos. Por exemplo, o 16F84A e o 16F628 custam, aproximadamente, R\$ 8,00 e o 16F877A custa algo em torno de R\$ 18,00.

A linha PIC está disponível em uma ampla variedade de modelos (de 8 a 68 pinos) para melhor adaptar-se às exigências de projetos específicos.

Cada modelo pode apresentar recursos diferenciados, mas um *chip* padrão PIC apresenta normalmente:

- Oscilador interno (que pode ser usado como fonte de *clock* própria em projetos com PIC).
- Várias entradas de interrupção de programas (permitem que eventos externos ou internos, associados aos periféricos, possam ser atendidos).
- detector de *brown out* (permite que uma falha ou uma queda na alimentação possa ser detectada e tratada sem prejuízo no funcionamento correto do sistema).
- EEPROM interna para armazenamento de dados (na qual podem ser armazenadas informações que não podem ser perdidas, caso haja o desligamento do sistema, ou seja, informações não voláteis – que não são perdidas quando o circuito é desligado).

- vários *timers* (temporizadores), os quais podem ser usados para contar tempo e assim permitir a sincronização, em instantes precisos, de tomadas de decisões.
- um *Watchdog* (cão de guarda).
- [Focus Product Selector Guide](#)

## Estudo de microcontroladores PIC - pt.4

O termo cão de guarda pode lhe parecer estranho, mas, esse recurso permite que o microcontrolador detecte alguma falha operacional ou erro de execução e se recupere desta falha ou deste erro, ou seja, o *Watchdog* está sempre vigilante como um verdadeiro cão de guarda. O uso desse dispositivo será tratado ainda em vários momentos do curso, principalmente, na programação.

Um ponto positivo para se trabalhar com o PIC é a grande quantidade de ferramentas existentes no mercado para suporte ao desenvolvimento de hardware e de software. Dentre essas ferramentas, destacam-se:

**1. O MPLAB® IDE - ambiente completo, em plataforma Windows, Linux e MAC OS, para desenvolvimento e gerenciamento de software para toda linha de microcontroladores PIC. Software desenvolvido e disponibilizado pela própria Microchip. Incorpora: editor de texto, montador (assembler), compiladores C, simuladores, emuladores, depurador de programa e suporte para gravação de memória EEPROM. Sua última versão pode ser baixada do site da Microchip <<http://www.microchip.com>>.**

**2. PROTEUS - ambiente gráfico para projeto, desenho e simulação de circuitos digitais com microprocessadores, desenvolvido e comercializado pela Labcenter Electronics Ltd. Em nosso curso será disponibilizada uma versão do Proteus, mas, se desejar uma versão demo, pode ser baixada através do link:**

<[http://www.labcenter.com/download/prodemo\\_autodl\\_general.cfm](http://www.labcenter.com/download/prodemo_autodl_general.cfm)>.

Essas duas ferramentas de projeto serão, exaustivamente, estudadas e usadas nas aplicações práticas que iremos desenvolver no decorrer do curso.



## Atividade 02

1. Entre em sites de outros fabricantes como Texas Instruments <<http://www.ti.com/>> e Freescale <<http://www.freescale.com>> e compare características e preços dos seus microcontroladores com os do PIC. Monte uma tabela com informações que considere mais importantes e que possam ajudar um projetista na escolha de uma família de microcontroladores a ser usada em seus projetos.

## Análise dos microcontroladores PIC 16F84, 16F628 e 16F877

As características gerais destes três microcontroladores estão sintetizadas na **Tabela 1**. O significado de algumas das características apresentadas nessa tabela, embora muito intuitivo, carece de alguns esclarecimentos e é o que faremos a seguir.

<b>Linha</b>	<b>Função</b>	<b>16F84</b>	<b>16F628</b>	<b>16F877</b>
1	CPU RISC de alta performance	x	x	x
2	Número de instruções	35	35	35
3	Velocidade máxima de operação	20 MHz	20 MHz	20 MHz
4	Número de ciclos de <i>clock</i> por instrução	4	4	4
5	Ciclo de instrução	200 ns	200 ns	200 ns
6	Tamanho da memória de programa <i>Flash</i>	1k x 14bits	2k x 14bits	8k x 14bits
7	Tamanho da memória de dados RAM	68 bytes	224 bytes	368 bytes
8	Tamanho da memória de dados EEPROM	64 bytes	128 bytes	256 bytes
9	Tamanho da memória pilha	8 x 14bits	8 x 14bits	8 x 14bits
10	Tamanho do bus de instruções	14 bits	14 bits	14 bits
11	Tamanho do bus de dados	8bits	8bits	8bits
12	<i>Power on Reset</i> (POR)	x	x	x
13	<i>Brown out detect</i> (BOD)	-	-	x
14	<i>Power up timer</i> (PWRT)	x	x	x
15	<i>Oscillator Start up Timer</i> (OST)	x	x	x
16	<i>Watchdog timer</i> (WDT)	x	x	x

17	Timer de 8 bits ( <i>Timer 0/Timer 2</i> )	1	2	2
18	Timer de 16 bits ( <i>Timer 1</i> )	-	1	1
19	Comunicação serial SPI	-	-	x
20	Comunicação serial I2C	-	-	x
21	USART/SCI	-	x	x
22	Conversores analógicos de 10 bits	-	-	x
23	Módulo CCP (Capture/Compare/PWM)	-	x	x
24	Comparador analógico	-	2	2
25	ICSP (Programação Serial <i>In-Circuit</i> )	x	x	x
26	Número de pinos	18	18	40
27	Modos de endereçamento	3	3	3
28	Fontes de interrupção	4	10	14
29	Operação em modo <i>SLEEP</i>	x	x	x
30	Gravação com código de proteção	x	x	x
31	Quantidade de portas de E/S	13	16	33
32	Corrente de saída	25 mA	25 mA	25 mA
33	Ciclos de apagamento/escrita na memória de programa	1.000	1.000	1.000

34	Ciclos de apagamento ou escrita na EEPROM de dados	1.000.000	1.000.000	1.000.000
35	Capacidade de retenção de dados na EEPROM	40 anos	40 anos	40 anos
36	Oscilador RC interno	x	x	x

**Tabela 1** – Características gerais dos microprocessadores 16F84, 16F628 e 16F877

As linhas 3, 4 e 5 da tabela especificam que a velocidade máxima de *clock* é de 20 MHz, que o número de ciclos de *clock* por instrução é 4 e que o tempo de um ciclo de instrução, para um *clock* de 20 MHz, é de 200 ns. A conta para se encontrar o tempo de 200 ns para um ciclo de instrução é  $1/(20.000.000/4)$

Observe que:

- 1. O ciclo de instrução fornecido de 200 ns é para um sinal de relógio de 20 MHz. Para se obter o tempo de um ciclo de instrução (ou tempo de execução de uma instrução) para outras frequências de *clock*,  $f_{ck}$ , basta dividir a frequência do *clock* pelo número de ciclos de *clock* gastos na execução de uma instrução e encontrar o seu inverso.**  
 $1/(f_{ck}/4)$
- 2. Se a frequência de *clock* cair para 4 MHz (que é a usada com frequência em alguns kits de desenvolvimento), o tempo de execução de uma instrução aumenta para 1  $\mu$ s. Faça as contas e comprove este valor.**

## Atividade 03

- 1. Calcule o tempo de execução de uma instrução caso fossem usados *clocks* com frequências de 10 MHz e de 1 MHz.**

## Análise dos microcontroladores PIC 16F84, 16F628 e 16F877 - pt.2

Nas linhas 6, 7, 8 e 9 são especificados alguns tipos e tamanho de memórias presentes nos microcontroladores.

Na linha 6, é especificado o tamanho da memória de programa (do tipo *Flash*). Esta memória é usada, como o nome mesmo diz, para armazenamento de programa. Por ser uma memória do tipo *Flash*, é não volátil, ou seja, após um programa ser armazenado ele permanecerá gravado, mesmo sem alimentação. Quando o sistema for religado, ele estará lá pronto para ser executado.

Observe, na linha 6 da tabela, que dentre os microcontroladores analisados, a maior quantidade de memória de programa é do 16F877 com 8 K x 14 bits.

Se na linha 10 é dito que o tamanho da instrução é de 14 bits, significa que a capacidade de armazenamento de bytes de programa nesta memória será de 14 Kbytes. Concorde? Faça as contas.

### Atividade 04

#### **1. Quais as capacidades de armazenamento das memórias de programa do 16F84 e do 16F628?**

## Análise dos microcontroladores PIC 16F84, 16F628 e 16F877 - pt.3

Nas linhas 7 e 8 são citados os tamanhos de duas memórias para armazenamento de dados, uma do tipo RAM (volátil) e uma do tipo EEPROM (não volátil). Ambas são muito pequenas, mas, normalmente o suficiente nas aplicações para as quais se prestam os microcontroladores. A RAM é usada para armazenar

variáveis de programa temporárias e a EEPROM é usada para armazenar variáveis de programa requeridas já na partida do programa ou tabelas permanentes, como uma tabela com valores de seno de ângulos, por exemplo.

Na linha 9 é especificado o tamanho da memória pilha, de apenas 8 bytes para os três microcontroladores. Essa memória só é usada internamente pelos microcontroladores para controle de fluxo de programa, guardando endereços de retorno de sub-rotinas ou de interrupções de programa. Como é muito pequena (apenas 8 bytes), evite durante a programação chamar uma função, que chama outra função, que chama outra função e, assim, sucessivamente. Muitas chamadas de funções podem provocar o estouro de pilha e a perda de rumo do programa.

Veja na linha 11 da tabela que, para os três microcontroladores, a largura do barramento de dados é de 8 bits (lembra que na Aula 2 foi dito que uma das características importantes de um microprocessador é a largura do seu barramento de dados? Se não, recapitule o conteúdo). Isso os classificam como sendo de 8 bits e só poderem fazer operações de transferência, operações lógicas ou operações aritméticas com tipos de dados de largura de byte.

Da linha 12 à 16 são citadas as características dos módulos de:

- POR (*Power on Reset*) – que força ou garante que o microcontrolador seja “resetado”, ou seja, reinicializado ao ser ligado.
- BOD (*Brown out detect*) – que força também um *reset* no microcontrolador ao perceber uma falha ou queda na alimentação, garantindo assim a sua operacionalidade apenas com níveis aceitáveis de tensão.
- PWRT (*Power up timer*) – que temporiza, ao ligar, a subida do nível da tensão de alimentação mantendo o microcontrolador em *reset* por um tempo de aproximadamente 66 ms, após o POR.
- OST (*Oscillator Start up Timer*) – que temporiza a partida do oscilador externo, garantindo que o microcontrolador também seja mantido em *reset* enquanto ocorre a estabilização do sinal de *clock* gerado pelo oscilador externo.

- WDT (*Watchdog* ou cão de guarda) – que força um *reset* no microcontrolador sempre que ocorrer um “estouro” de contagem (ou seja, passa de 255 a 0).

## Análise dos microcontroladores PIC 16F84, 16F628 e 16F877 - pt.4

A associação do PWRT e do OST é usada algumas vezes como forma de garantir um tempo mínimo de *reset* ao microcontrolador, equivalente ao tempo de todos os dispositivos envolvidos no projeto estarem funcionando adequadamente.

Na realidade, todos os módulos citados (POR, BOD, PWRT, OST e WDT) são usados para garantir situação ótima ao ligar e para garantir que, na ocorrência de quedas de alimentação ou de falhas na programação, sempre ocorra um pulso de *reset* (ou de inicialização) no microcontrolador. Isso nos dá maior confiança em deixar um sistema a microcontrolador no campo e ter a certeza de que, mesmo sem a presença humana, o sistema poderá se autocorrigir de uma falha ou falta. Essa autocorreção se dá pela reinicialização do sistema sem a ação direta de um operador. Afinal, quando um sistema é colocado em locais longínquos e de difícil acesso, como a floresta Amazônica, ou pouco acessíveis como, por exemplo, o dorso de uma onça pintada, nem sempre pode haver a presença humana para dar um CTRL+ALT+DEL, equivalente a quando queremos, reinicializar o Windows após uma falha.

Na medida em que adquirir maturidade no assunto, você poderá assimilar melhor as colocações feitas anteriormente. Entenda pelo menos uma coisa: o fabricante desses microcontroladores pensou em tudo para garantir que um sistema projetado com esses dispositivos não venha a deixar de funcionar, caso alguma anomalia temporária altere sua operação normal.

Nas linhas 17 e 18 são especificados três temporizadores, os quais podem ser usados em aplicações gerais que exijam contagem de tempo. Dois de 8 bits (o *Timer 0* e o *Timer 2* que contam até 255) e outro de 16 bits (o *Timer 1* que conta até 0FFFFh). No 16F84 só existe um *Timer* de 8 bits que é o *Timer 0*.

Da linha 19 à 21 são citados os três padrões de comunicação serial de dados disponíveis nestes *chips*:

- SPI (Serial Peripheral Interface) – padrão serial programável.
- I2C (Inter Integrated Communication) – padrão de comunicação serial muito usado entre componentes como, por exemplo, entre o microcontrolador e uma unidade de memória serial externa.
- USART (Universal Synchronous Assynchronous Receiver Transmitter) – padrão de comunicação serial síncrono ou assíncrono entre um transmissor e um receptor.

## Análise dos microcontroladores PIC 16F84, 16F628 e 16F877 - pt.5

O padrão USART foi muito usado nas quatro últimas décadas, mas está perdendo fôlego para o padrão USB (Universal Serial Bus – Barramento Serial Universal). Um módulo de comunicação serial USB está presente em alguns microcontroladores da família 18F. É possível encontrar também alguns *chips* da família 18F com outras interfaces de comunicação como CAN e Ethernet.

Você já ouviu falar sobre os protocolos de comunicação USB, CAN e Ethernet? O Ethernet e o USB talvez sim, devido ao uso do primeiro na internet e do segundo na ligação de periféricos como impressora e mouse a um PC – Computador Pessoal. De qualquer forma, busque informações sobre eles. Será de muito bom proveito ter essas informações.

Nas linhas 22, 23 e 24 são citados módulos para conversão analógica / digital, para capturar, comparar e gerar PWM e para comparação. Esses módulos funcionais são muito importantes para quem trabalha, por exemplo, projetando sistemas de controle de ambiente que precisam receber sinais analógicos e convertê-los para digital ou que precisa, por exemplo, gerar sinais para controle de velocidade de motores que precisa gerar sinais PWM (sinais modulados por largura de pulso. Sinais que podem atuar mais ou menos de acordo com a sua largura de pulso em nível alto. Não queira saber mais por enquanto, este conhecimento lhe será transmitido quando estiver cursando engenharia elétrica, que tal?).



Na linha 25, o termo Programação Serial *In-Circuit*, ou ICSP, significa que você pode programar, apagar e reprogramar o microcontrolador sem ter que retirá-lo do circuito ou do kit de desenvolvimento. Esta técnica será utilizada durante nossas aulas práticas.

Na linha 28 faz-se referência à grande quantidade de fontes de interrupção. No 16F877 são 14 fontes de interrupção, enquanto, no 16F84 são apenas 4. Mais adiante, trataremos sobre interrupção de programa e como utilizar cada uma dessas fontes geradoras de pedidos de interrupção. Normalmente, cada fonte de interrupção está relacionada a um recurso do microcontrolador. Daí, já dá para perceber o quanto o 16F877 é mais “poderoso” do que o 16F84.

## Análise dos microcontroladores PIC 16F84, 16F628 e 16F877 - pt.6

Na linha 29 é dito que todos os três processadores podem ser colocados para dormir. Ótimo! Com isso, quando não estiverem executando alguma tarefa, colocá-los para dormir ajuda na redução do consumo. Para circuitos em campo isso é primordial, não acha?

Outra coisa boa: na linha 30 é dito que você pode gravar seu programa no microcontrolador usando um código de segurança. Feito isso, não tem perigo de algum pirata roubar seu código. Só use o código de segurança num produto final e nunca enquanto estiver desenvolvendo ou testando o seu software. Se esquecer o código de segurança, adeus! Ninguém mais terá acesso ao programa fonte.

Na linha 31 são especificadas as quantidades de portas de E/S (Entrada/Saída) presentes em cada um dos microcontroladores em análise. Você percebe que o 16F877 apresenta uma quantidade de portas de Entrada/Saída bem maior que os outros dois microcontroladores (18 a mais que o 16F628 e, 20 a mais que o 16F84). Isso lhe dá maior potencial de trabalho quando muitas entradas devem ser monitoradas ou quando muitas saídas devem ser ativadas.

Na linha 32 é fornecido o valor da corrente de saída das portas (no caso, 25 mA para todas as saídas dos três *chips*). Esta corrente por si só já é suficiente, sem circuitos adicionais para alimentar diretamente circuitos externos, tais como os de

acendimento de LED (que precisam de correntes da ordem de 10 mA).

Das linhas 33 a 35, existem informações de tempo de operação e de vida dos componentes. Não dá para acreditar que não sejam verdadeiras. Aliás, não dá para esperar para ver se são verdadeiras, não é mesmo? Veja a informação da linha 35, a qual especifica que o tempo de retenção de dados na memória EEPROM é de 40 anos.

Na linha 36, para encerrar, é dito que todos estes microcontroladores possuem um oscilador RC interno, capaz de gerar um sinal de *clock* de 4 MHz. Isso possibilita criarmos circuitos com esses microcontroladores sem a necessidade de outro componente externo, nem mesmo um cristal ou gerador de *clock* externo, já que possuem o seu próprio oscilador. Um oscilador RC, como o que foi visto na disciplina de Eletricidade, se constitui em um circuito extremamente simples, tendo em sua malha de oscilação apenas um capacitor (de capacitância C) e um resistor (de resistência R).

## Atividade 05

- 1. Quando você teve as informações sobre o *fan-out* dos integrados das famílias TTL na disciplina de Sistemas Digitais, quais os níveis das correntes de saída informados para eles? Caso não tenha esta informação, busque na internet. Em seguida, compare estes valores com a informação de corrente fornecida na linha 32 da tabela. Quantos componentes TTL podem ser ativados diretamente de uma porta de saída de qualquer destes microcontroladores?**

## Análise dos microcontroladores PIC 16F84, 16F628 e 16F877 - pt.7

Nas duas próximas figuras, estão representadas em diagramas de blocos as arquiteturas internas dos três microcontroladores cujas características gerais foram apresentadas na Tabela 1.

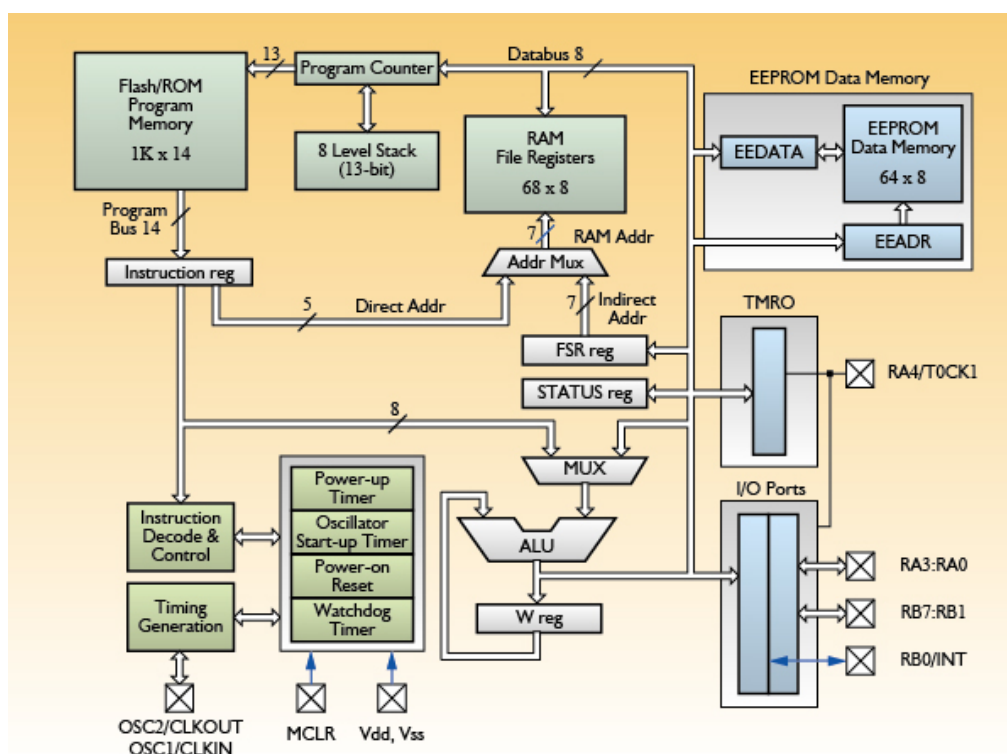
Na Figura 1 é mostrada a arquitetura interna do 16F84 como representada no datasheet do fabricante.

Por questões didáticas, na Figura 2 foram omitidos os barramentos internos e estão representados apenas os blocos funcionais constituintes do 16F84, do 16F628 e do 16F877, sendo que os blocos comuns aos três microcontroladores estão representados na cor branca; os blocos comuns apenas aos 16F628 e 16F877 estão representados na cor amarela e os blocos que existem apenas no 16F877 estão representados na cor azul.

Sempre que se projeta um circuito digital a partir de componentes padrões integrados, o projetista deve ter em mãos as especificações completas dos dispositivos a serem utilizados, fornecidas em seus respectivos datasheets.

Como informação relevante para o projeto está a designação dos pinos, necessária para que sejam feitas corretamente as ligações do circuito ao *chip*. Na Figura 3 é mostrada a pinagem do dispositivo 16F877. Caso necessite para projeto a pinagem de outros dispositivos PIC ou DsPIC, basta acessar seu respectivo datasheet no site da Microchip.

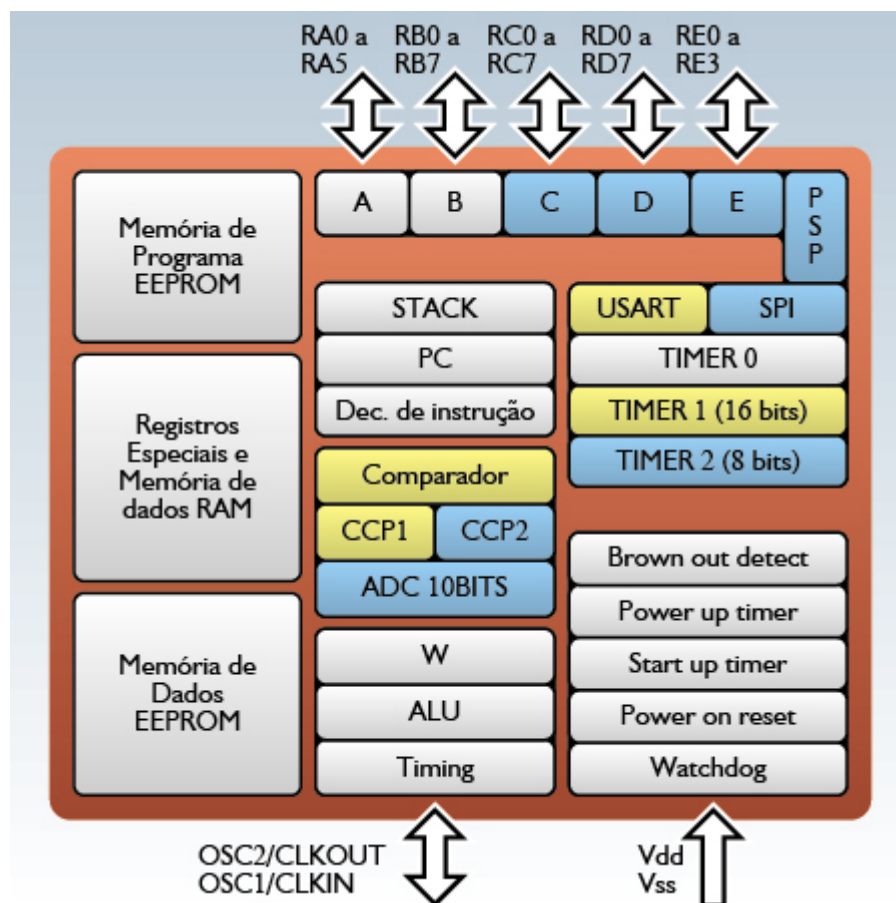
**Figura 01** - Arquitetura interna do 16F84.



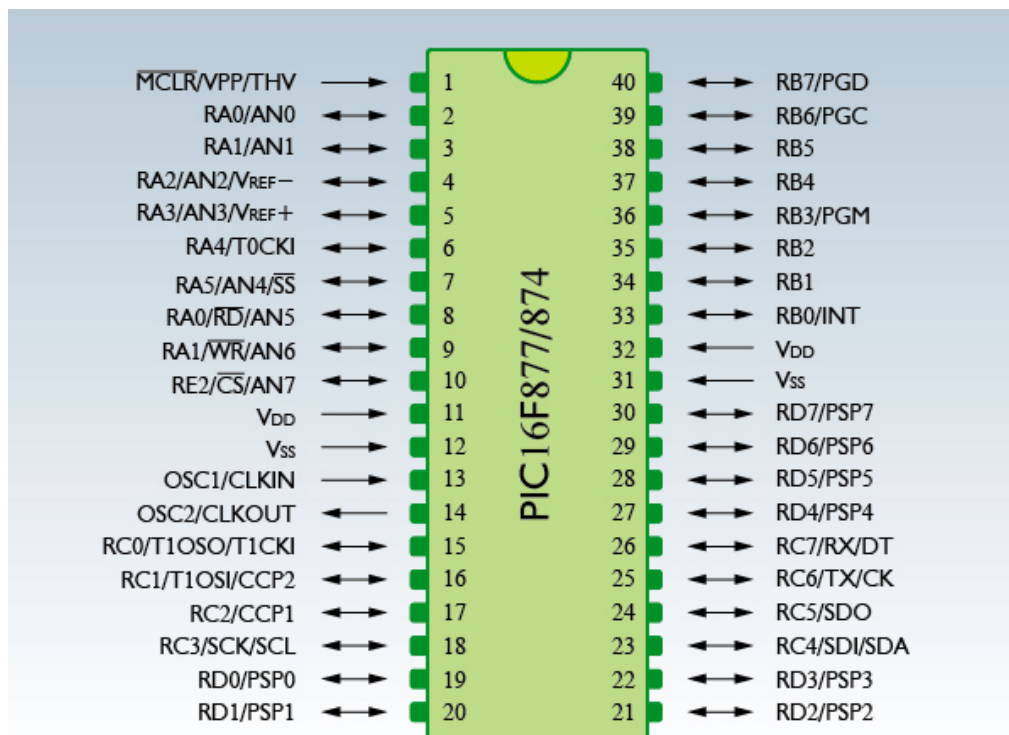
Observe na Figura 3 que alguns pinos apresentam-se com mais de uma função. A escolha correta da função desejada para cada pino é feita por software, através de comandos de configuração.

A escolha de uma função para um determinado pino poderá inviabilizar o uso de alguma funcionalidade do microcontrolador. Por exemplo, a escolha dos pinos 25 (RC6/TX/CK) e 26 (RC7/RX/DT) como entradas ou saídas digitais comuns através das portas RC6 e RC7 poderá inviabilizar o uso do módulo de comunicação USART para uma comunicação assíncrona, já que, é através deles que os dados são, respectivamente, transmitidos e recebidos; ou síncrona, já que é através deles também que os dados e o *clock* estão, respectivamente, presentes.

**Figura 02** - Diagrama com os blocos funcionais dos 16F84, 16F628 e 16F877.



**Figura 03** - Designação dos pinos do microcontrolador 16F877.



## Organização da memória de programa

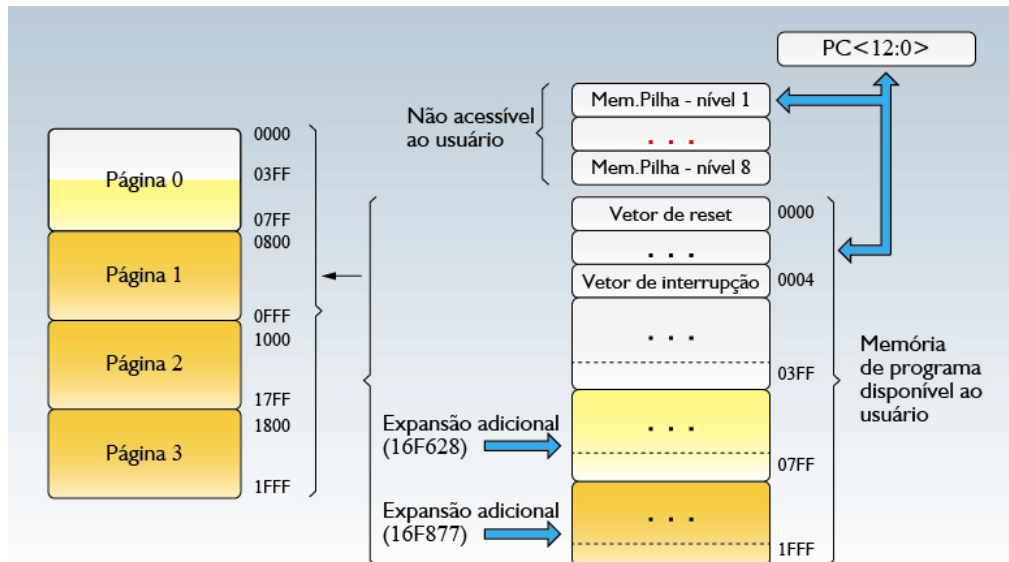
Bem, agora que você já aprendeu e analisou alguns detalhes sobre os microcontroladores PIC, vamos estudar sua memória de programa, cuja organização é mostrada na Figura 4.

Relacionados à área da memória de programa do PIC estão, além da memória de programa em si, apenas um contador de programa (o registrador PC) e uma memória pilha.

A memória pilha não é acessível ao usuário e serve exclusivamente para armazenar até 8 endereços de retorno de desvios provocados por chamadas de sub-rotinas ou por interrupções de programa. O que denomino chamada de sub-rotina pode ser associado a uma chamada de função em linguagens de mais alto nível, ok? Uma interrupção de programa também pode ser entendida como uma chamada de uma função, só que sua ação é em resposta a uma intervenção externa, normalmente em resposta a um evento que ocorreu em uma unidade periférica do PIC.

O nome de memória pilha se dá a uma analogia ao empilhamento de pratos, o primeiro que for empilhado será o último a ser retirado. Lembra-se deste conceito? Ele deve ter sido tratado em alguma disciplina em que você estudou as estruturas de dados.

**Figura 04** - Organização da memória de programa dos microcontroladores 16Fxxx.



No momento da chamada de uma sub-rotina (o que é feito por uma instrução chamada CALL) ou quando ocorre o atendimento a um pedido de interrupção, o valor presente no PC é salvo na pilha e só será restaurado na execução de uma instrução de retorno, através de instruções do tipo RETURN ou RETLW (se tiver sido ativada por uma instrução CALL) ou por uma instrução RETFIE (se tiver sido ativada em resposta a um pedido de interrupção). Quanto às instruções citadas, elas estão em conformidade com a linguagem Assembly. Nas próximas aulas, trataremos dessas instruções e de mais algumas quando fizermos a análise do conjunto de instruções dos PIC da família 16F.

## Organização da memória de programa - pt.2

Após terem sido guardados oito valores na pilha, na nona vez que for guardado um valor, o primeiro armazenado será perdido. A décima vez que for escrito um valor, o segundo será perdido e, assim por diante.

Os PIC16FXX têm um contador de programa (PC) de 13 bits, o que lhes permitem endereçar 8 KBytes x 14 espaços de memória de programa. No entanto, para o PIC16F84A apenas o primeiro 1 KByte x 14 (0000h - 07FFh) endereços são fisicamente implementados e para o 16F628 apenas os primeiros 2 KBytes x 14 (0000h - 07FFh) endereços são fisicamente implementados.

O PC pode ser entendido como um registrador que sempre aponta para o endereço de memória da próxima instrução que será executada pelo PIC.

Para o 16F877, a memória de programa, como pode ser visto na Figura 4, está dividida em quatro páginas de 2 KBytes x 14 endereços cada uma, nomeadas de Página 0 (endereços 0000 a 07FFh), Página 1 (endereços 0800h a 0FFFh), Página 2 (endereços 1000h a 17FFh) e Página 3 (endereços 1800h a 1FFFh). A Página 0 corresponde a toda memória de programa do 16F628.

Quando se desenvolvem programas para o 16F877, deve-se ter um cuidado especial com as instruções GOTO e CALL (desvio de programa e chamada de sub-rotina, respectivamente).

O programa poderá não ser adequadamente executado se a posição chamada não estiver dentro da mesma página. Quando formos programar, teremos a oportunidade de esclarecer como resolver esse problema.

A Figura 4 chama atenção para os endereços 0000h e 0004h. O endereço 0000h da memória de programa está reservado para o vetor de *reset* e o endereço 0004h está reservado para o vetor de interrupção de programa. O que significa que:

- numa situação de *reset*, o PC assume o valor 0000h e que a partir deste endereço (0000h) deve estar posicionada a primeira instrução do programa de partida do seu aplicativo;
- na ocorrência de um pedido de interrupção e do seu atendimento, o PC assume o valor 0004h e que a partir deste endereço (0004h) deve estar posicionada uma rotina para tratamento de interrupções.

Como esses endereços estão muito próximos, normalmente, no endereço 0000h é colocada uma instrução de desvio para o início efetivo do programa de partida do aplicado em execução (normalmente, instruções que permitem fazer a inicialização do PIC e da aplicação desejada). Esta técnica será mostrada durante as práticas de programação dos PIC.

## Atividade 06

- 1. Pelo mapa de memória de programa dos microprocessadores apresentados, considerando a largura de 14 bits usada para armazenamento de uma instrução, quantas instruções cabem em cada página? Qual o maior tamanho de programa que pode ser escrito para o 16F628? E para o 16F877?**



# Conclusão

Para não estendermos demais o tratamento de informações sobre os microcontroladores PIC, vamos encerrar esta aula por aqui. Na próxima aula, retomaremos o assunto partindo do estudo da organização da memória de dados dos PIC da família 16F. Adiantando, a memória de dados do PIC se constitui de uma memória volátil, do tipo RAM e uma memória não volátil, do tipo EEPROM.

Na memória de dados do tipo RAM estão fixados os FSR (registradores de funções especiais), definidos pelo fabricante para programação das unidades periféricas presentes nos PIC, os GPR (registradores de propósito geral) definidos pelo usuário e, como não poderiam deixar de existir, as áreas para armazenamento temporário de dados, tipo as variáveis de programa.

# Resumo

Nesta aula, você iniciou o estudo dos microcontroladores PIC, conhecendo em detalhes as características gerais da família 16F. Você também conheceu a organização e suas arquiteturas internas, com as devidas referências aos seus principais blocos funcionais construtivos. E por fim, foi vista a organização de suas memórias de programa.

## Autoavaliação

1. Quais as principais características dos microcontroladores PIC?
2. Que unidades básicas estão presentes numa arquitetura PIC e que são comuns a todos os microcontroladores PIC?
3. O que entende por detector de *brown out*? Para que serve o *Watchdog*?
4. Quais os tipos de memórias presentes nos microcontroladores PIC e como é a organização da memória de programa do 16F84?
5. Para que serve a memória pilha presente na organização da memória de programa?

## Referências

PEREIRA, Fábio. **Microcontroladores PIC: técnicas Avançadas**. São Paulo: Érica, 2002.

SOUZA, David José de. **Desbravando o PIC**. São Paulo: Editora Érica, 2000.