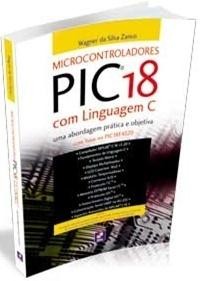
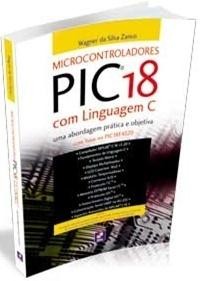
MICROCONTROLADORES

(Baseado no PIC18F4520)



## Professor: Wagner da Silva Zanco

**Arquivo 01** 1

## SOBRE ESTE MATERIAL

* Este material é para uso individual. Todos os direitos reservados. Proibida a distribuição total ou parcial deste material, por qualquer meio ou processo, sem a expressa autorização do autor. Essas proibições aplicam-se também ”s características gráficas da obra e ” sua editoração.
* A violação dos direitos autorais é punível como crime (art. 184 e parágrafos, do Código Penal, cf. Lei no 6.895, de 17.12.80) com pena de prisão e multa, conjuntamente com busca e apreensão e indenizações diversas (artigos 102, 103 parágrafo único, 104, 105, 106 e 107 itens 1, 2 e 3 da Lei no 9.610, de 19/06/98, Lei dos Direitos Autorais).
* Certos materiais contidos neste arquivo foram incluídos com a permissão da *Microchip Technology Incorporated*. Nenhuma distribuição, reimpressão ou reprodução do material citado pode ser feita sem o consentimento por escrito da Microchip Technology Inc.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 2



## EMENTA

#### **Ementa*:*** Botão e LED, Teclado matriz, *Displays* multiplexados, *Display* de cristal líquido, Conversão A/D, Modulação por largura de pulso PWM, Comunicação serial RS232 e RS485, Protocolos I2C e SPI.

**Material didático:** ZANCO, W. S. **Microcontroladores PIC18F4520 com Linguagem C -** *uma abordagem prática e objetiva*. 1ã edição. São Paulo: Érica, 2010.

**Material de apoio: site**: [http://www.wagnerzanco.com.br](http://www.wagnerzanco.com.br/)

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 3

**Objetivo geral:**

## OBJETIVOS

#### Desenvolver projetos de circuitos eletrônicos microcontrolados para fins profissionais, incluindo interface homem-máquina e comuni- cação de dados.

**Objetivos específicos:**

#### - Empregar as técnicas utilizadas para interfacear microcontrola- dores com dispositivos periféricos utilizados na construção de interface homem-máquina.

1. - Compreender os protocolos de comunicação mais utilizados na troca de informações entre microcontroladores e outros dispositivos.

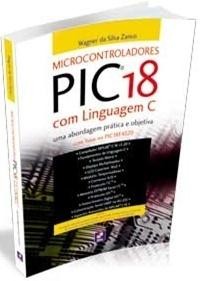
#### - Aplicar a Linguagem C no desenvolvimento de aplicações pro- fissionais para microcontroladores.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 4



## MICROCONTROLADORES

Baseado no PIC18F4520

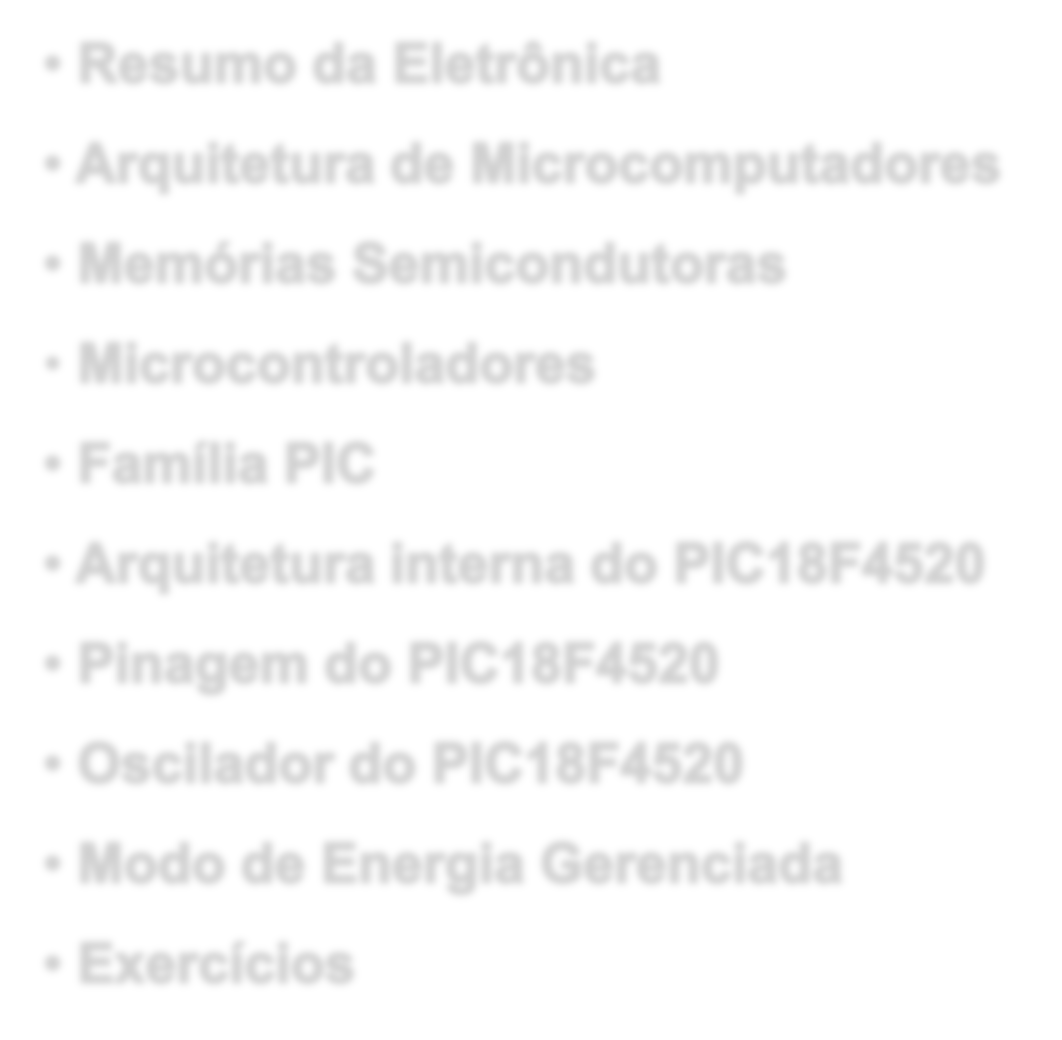


## ARQUIVO 1 - VISÃO GERAL

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 5



## SUMÁRIO ARQUIVO 1



•

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 6



## OBJETIVOS

* Ao final da aula o aluno deverá ser capaz de:

1. Diferenciar as diversas arquiteturas utilizadas no projeto de microcom- putadores.
2. Identificar os diversos blocos que compõem a arquitetura interna do

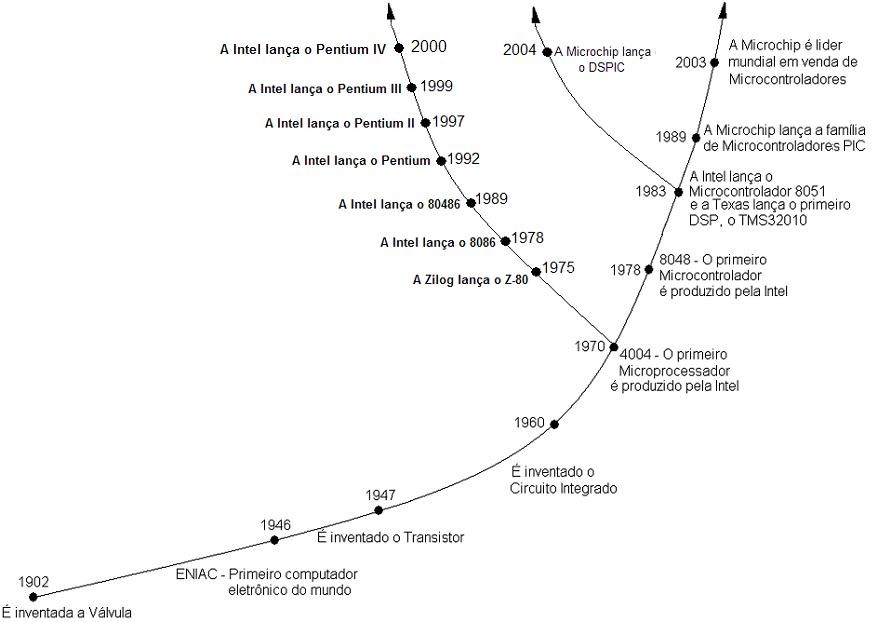
microcontrolador PIC18F4520.

1. Classificar os tipos de osciladores e configurar o oscilador do microcontrolador para uma dada aplicação.
2. Distinguir os diferentes modos de gerenciamento de energia do microcontrolador PIC18F4520.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 7



Disciplina: Microcontroladores



### Um Resumo da Evolução da Eletrônica

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco**

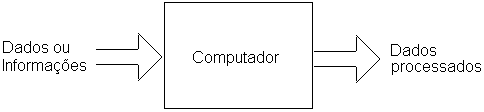
8



## FUNÇÃO BÁSICA DE UM COMPUTADOR

* O computador basicamente é um processador de informações.
* O usuário introduz as informações no computador por meio de um dispositivo de entrada como o teclado, mouse etc.
* O computador processa as informações e retorna o resultado para o usuário

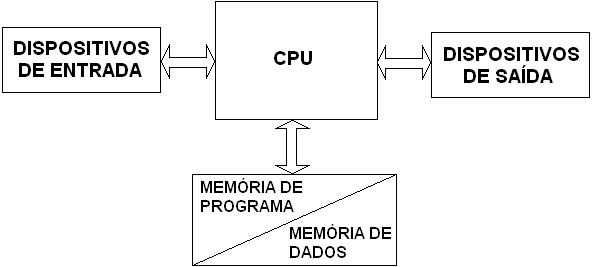
por meio de um dispositivo de saída como o vídeo ou impressora.



**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 9

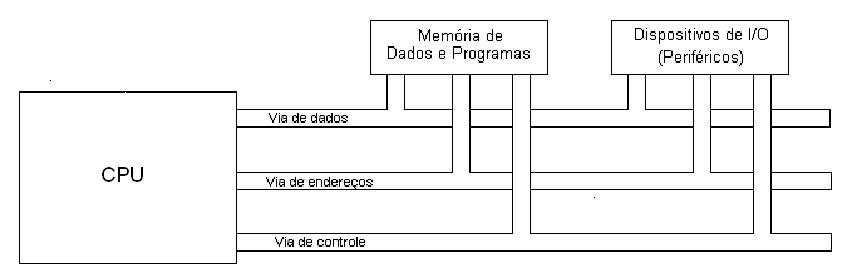


## MÓDULOS BÁSICOS DE UM MICROCOMPUTADOR



**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 10

## ARQUITETURA VON-NEUMANN



**CPU –** é um circuito integrado (CI) capaz de executar um conjunto de tarefas denominadas instruções. A CPU gerencia todo o sistema e executa os programas.

**Memória –** armazena os dados que serão processados e os programas que

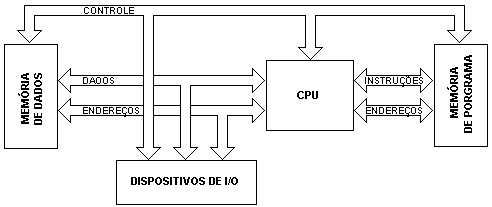
serão executados.

**Dispositivos de I/O –** são os responsáveis pela entrada e saída de dados do sistema.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 11



## ARQUITETURA HARVARD



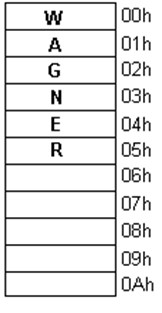
**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 12



## ORGANIZAÇÃO DA MEMÓRIA

#### Do ponto de vista lógico, a memória é composta por várias localidades podendo armazenar um conjunto de bits em cada uma. Cada localidade possui seu próprio endereço, sendo este um valor numérico representado no sistema hexadecimal.

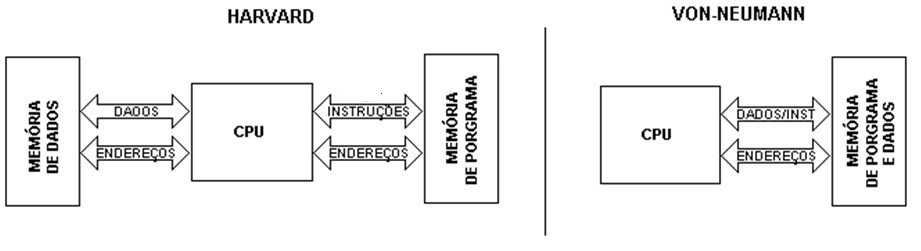
* + A unidade padrão de armazenamento de dados na memória é o byte.



**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 13



VON-NEUMANN *versus* HARVARD



#### **CPU CISC –** as CPUs utilizadas em computadores que seguem a arquitetura Von-Neumann são do tipo CISC, as quais possuem um set de instruções ampliado (muitas instruções).

**CPU RISC –** as CPUs utilizadas em computadores que seguem a arquitetura Harvard são do tipo RISC, as quais possuem um set de instruções reduzido (poucas instruções).

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 14



## GERENCIANDO O SISTEMA

**“Para gerenciar o sistema a CPU precisa se comunicar com a memória e com os dispositivos de I/O. Esta comunicação pode ser de leitura ou de escrita.”**

#### **Escrita** - quando a CPU envia uma informação para ser armazenada numa localidade de memória ou para um dispositivo de I/O. Por exemplo, quando a CPU envia uma informação para o vídeo, dizemos que ela escreveu no vídeo.

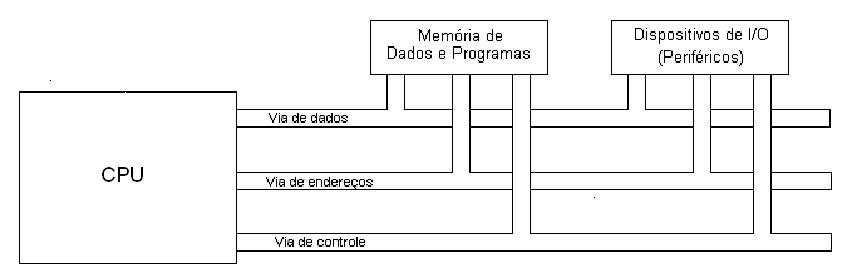
**Leitura** - quando a CPU busca uma informação na memória ou num dispositivo de I/O. Quando uma tecla é pressionada no teclado e aparece no vídeo, é porque a CPU efetuou uma leitura no teclado e escreveu o dado lido no vídeo.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 15



## GERENCIANDO O SISTEMA

**“A CPU reconhece cada um dos dispositivos de I/O, assim como cada localidade de memória pelo seu respectivo endereço”.**



**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 16



## CAPACIDADE DE PROCESSAMENTO DE UMA CPU

#### SET(conjunto) de instruções;

* Velocidade com que as instruções são executadas;

#### Multiprocessamento (execução de várias instruções simultâneas);

-Quantidade de memória que é capaz de endereçar;

#### Comprimento da via de dados

* Etc.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 17



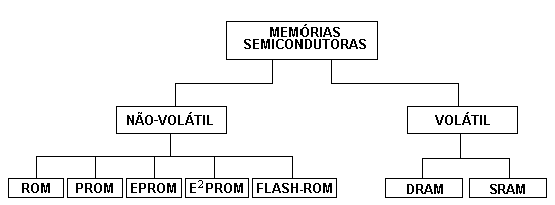
## MEMÓRIAS SEMICONDUTORAS

São memórias implementadas em circuitos integrados (chips de memória).

##### Estão divididas em dois grupos: Volátil e não-volátil.

**Volátil** – É a memória que perde os dados armazenados quando a energia é desligada.

**Não-volátil** – É a memória que não perde os dados armazenados quando a energia é desligada.



**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 18

**ROM**

* Somente leitura
* Vem programada de fábrica
* Não pode ser reprogramada

**PROM**



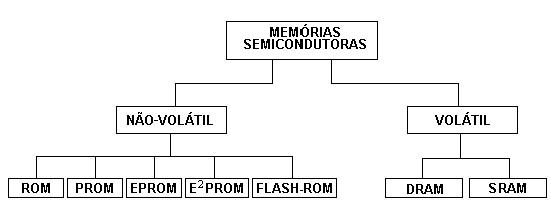
* Somente leitura
* É programada eletricamente pelo usuário
* Uma vez programada se transforma numa ROM
* Não pode ser reprogramada

**EPROM**

* Somente Leitura
* Programável Eletricamente pelo usuário

MEMÓRIA NÃO-VOLÁTIL

* Possui uma janela que quando exposta a raios ultravioletas seus dados são apagados, podendo ser reprogramada.



**E2PROM**

* Somente Leitura
* Programável Eletricamente pelo usuário
* É semelhante a uma EPROM só que seus dados são apagados por meio de uma tensão elétrica aplicada a um de seus pinos, podendo ser reprogramada da mesma forma que uma EPROM.

**FLASH-ROM**:



* Somente Leitura
* Programável Eletricamente pelo usuário
* Pode ser reprogramada por software.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 19

**DRAM**

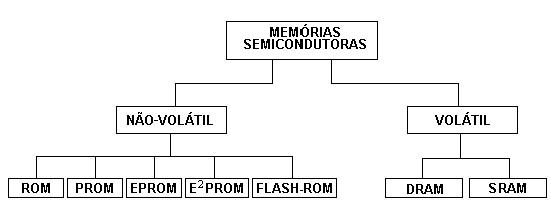
* Leitura e escrita
* Fabricada com capacitores
* Precisa de refresh

**SRAM**

* Leitura e escrita
* Fabricada com flip flop
* Não precisa de refresh
* É mais rápida do que DRAM

### MEMÓRIA NÃO-VOLÁTIL





**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 20



## MICROCONTROLADOR

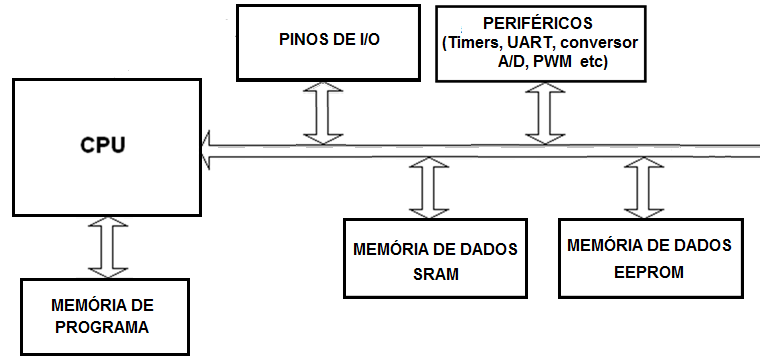
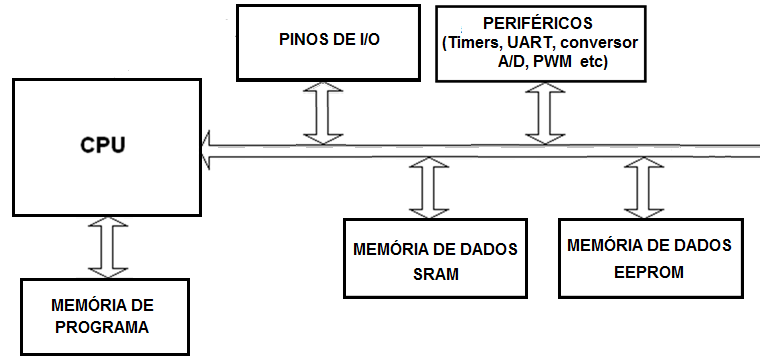
#### “É um computador em um único chip. O microcontrolador integra em um único invólucro CPU, memória e dispositivos de I/O. O primeiro microcontrolador, o 8048, foi desenvolvido pela empresa Intel no final da década de setenta.”

Ao contrário do microcontrolador, para se construir um computador com um microprocessador é necessário utilizar memória externa e dispositivos de I/O externos.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 21



## DIAGRAMA EM BLOCOS BÁSICO DE UM MICROCONTROLADOR COM ARQUITETURA HARVARD



**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 22



## ESTADO DA ARTE DOS MICROCONTROLADORES

**-** Família 8051 (8 bits)

* Família PIC (8, 16 e 32 bits)
* Família ARM (32 bits)
* Família MIPS (32 bits)

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 23



## FAMÍLIA PIC

### Desenvolvida pela empresa Microchip Technology Inc.

* PIC10 (8 bits) (menor microcontrolador do mundo)
* PIC12 (8 bits)
* PIC14 (8 bits)
* PIC16 (8 bits)
* PIC18 (8 bits)
* PIC24 (16 bits)
* PIC32 (32 bits)
* DSPIC (16 bits) (processador digital de sinais + microcontrolador)

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 24



## TIPOS DE MEMÓRIA DE PROGRAMA

**ROM TIPO MÁSCARA -** O chip já sai da fábrica com o programa gravado nele, não podendo ser regravado de forma nenhuma pelo usuário. O custo destes componentes é bem reduzido, mas só são viáveis se adquiridos em grande quantidade. Estes componentes são identificados pelo sufixo “CR”.

**OTP -** Estes dispositivos utilizam memória PROM para armazenar os programas. Eles vem de fábrica vazios, sendo a gravação do programa feita pelo usuário, não podendo o mesmo ser regravado. Isto impede a utilização destes componentes na fase de desenvolvimento e teste de programas. Os mesmos são identificados pelo sufixo “C”.



**EPROM -** Estes componentes possuem uma janela onde podemos, através da exposição a raios ultravioletas, apagar os programas gravados no chip. Embora seja trabalhoso efetuar o apagamento do programa, estes componentes podem ser utilizados na fase de testes e desenvolvimento de programas. O sufixo pode ser “JW” para dispositivos com encapsulamento do tipo DIP ou “CL” para dispositivos com encapsulamento do tipo PLCC.



**FLASH -** Identificados pelo sufixo “F”, estes componentes permitem ser apagados/regravados milhares de vezes através de sinais elétricos aplicados em alguns de seus pinos, o que pode ser feito automaticamente por um circuito gravador de Flash como o Picstart Plus, fabricado pela Microchip ou o JDM, cujo hardware é encontrado facilmente na internet. Estes componentes são a melhor opção para teste de programas e implementação final dos circuitos, visto que estão ficando cada dia mais baratos.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 25

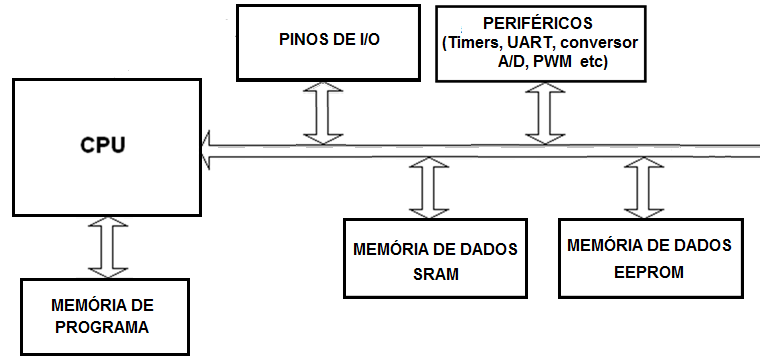
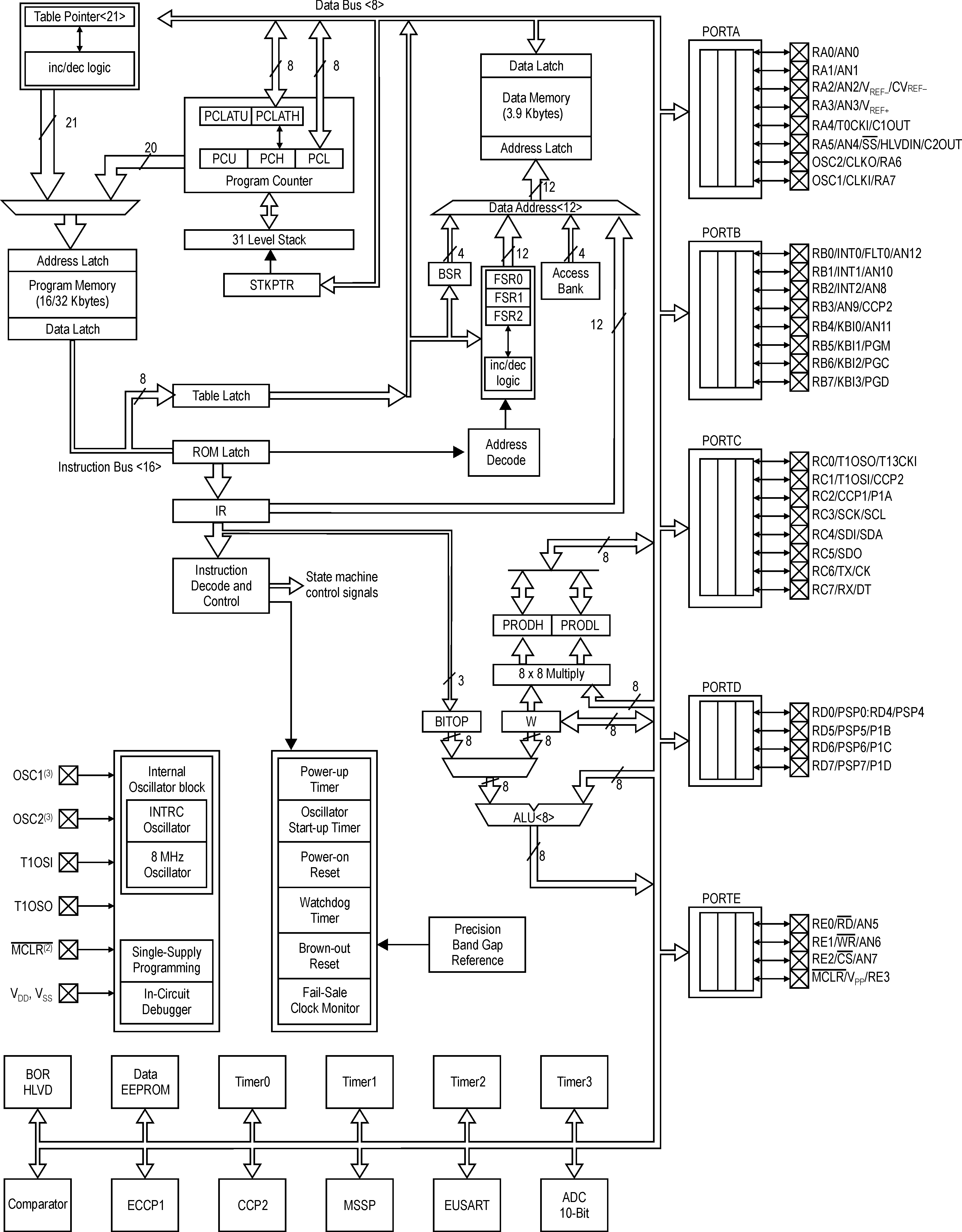


VISÃO GERAL DO PIC18

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Características** | **PIC18F2420** | **PIC18F2520** | **PIC18F4420** | **PIC18F4520** |
| **Frequência de operação** | **DC - 40MHz** | **DC - 40MHz** | **DC - 40MHz** | **DC - 40MHz** |
| **Memória de programa (Bytes)** | **16384** | **32768** | **16384** | **32768** |
| **Memória de programa (Instruções)** | **8192** | **16384** | **8192** | **16384** |
| **Memória de dados** | **768** | **1536** | **768** | **1536** |
| **Memória de dados EEPROM** | **256** | **256** | **256** | **256** |
| **Fontes de interrupção** | **19** | **19** | **20** | **20** |
| **Ports de I/O** | **Ports A, B, C, (E)** | **Ports A, B, C, (E)** | **Ports A, B, C, D, E** | **Ports A, B, C, D, E** |
| **Timers** | **4** | **4** | **4** | **4** |
| **Capture/Compare/PWM Modules** | **2** | **2** | **1** | **1** |
| **Enhanced Capture/Compare/PWM Modules** | **0** | **0** | **1** | **1** |
| **Comunicação serial** | **MSSP, Enhanced USART** | **MSSP, Enhanced USART** | **MSSP, Enhanced USART** | **MSSP, Enhanced USART** |
| **Comunicação paralela (PSP)** | **Não** | **Não** | **Sim** | **Sim** |
| **Conversor A/D de 10 bits** | **10 canais de entrada** | **10 canais de entrada** | **13 canais de entrada** | **13 canais de entrada** |
| **Resets (e Delays)** | **POR, BOR, instrução RESET, Stack Full, Stack Underflow (PWRT, OST),**  **(opcional), WDT** | **POR, BOR, instrução RESET, Stack Full, Stack Underflow (PWRT, OST),**  **(opcional), WDT** | **POR, BOR, instrução RESET, Stack Full, Stack Underflow (PWRT, OST),**  **(opcional), WDT** | **POR, BOR, instrução RESET, Stack Full, Stack Underflow (PWRT, OST),**  **(opcional), WDT** |
| **Detecção de programação em alta/baixa tensão** | **Sim** | **Sim** | **Sim** | **Sim** |
| **Brown-out Detect Programável** | **Sim** | **Sim** | **Sim** | **Sim** |
| **Set de instruções** | **75 instruções; 83 com set de instruções estendido**  **habilitado** | **75 instruções; 83 com set de instruções estendido**  **habilitado** | **75 instruções; 83 com set de instruções estendido**  **habilitado** | **75 instruções; 83 com set de instruções estendido**  **habilitado** |
| **Encapsulamentos** | **28 - Pin SPDIP**  **28 - Pin SOIC**  **28 - Pin QFN** | **28 - Pin SPDIP**  **28 - Pin SOIC**  **28 - Pin QFN** | **40 - Pin PDIP**  **44 - Pin QFN**  **44 - Pin TQFP** | **40 - Pin PDIP**  **44 - Pin QFN**  **44 - Pin TQFP** |

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 26





ARQUITETURA INTERNA DO PIC18F4520

##### Arquitetura Harvard

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 27



## PRINCIPAIS CARACTERÍSTICAS DO PIC18F4520

A partir da série PIC18, a arquitetura RISC foi otimizada para obter alta performance com a utilização de um compilador C.

* + Capacidade para executar até 10 milhões de instruções por segundo 10MIPS);
  + Até 40MHz de sinal de *clock*;
  + De 4MHz a 10MHz de sinal de *clock* com o PLL ativo;
  + Instruções com 16 bits de tamanho;
  + Níveis de prioridade no tratamento da interrupção;
  + Multiplicação por *hardware* entre operandos de 8 bits em um único ciclo de instrução;
  + Set de instruções estendido;
  + Modos de gerenciamento de energia.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 28



## PERIFÉRICOS DO PIC18F4520

* Capacidade de corrente de 20 mA pino;
* Três pinos de interrupção externa;
* Módulo temporizador/contadorTimer0 de 16 bits;
* Módulo temporizador/contadorTimer1 de 16 bits;
* Módulo temporizadorTimer2 de 8 bits;
* Módulo temporizador/contadorTimer3 de 16 bits;
* Treze canais de conversor A/D de 10 bits;
* Dois módulos *Capture/Compare/PWM*(CCP);
* Módulo de detecção de alta e baixa tensão;
* Módulo MSSP podendo operar nos modos I2CTM e SPITM;
* Módulo EUSART com endereçamento avançado com suporte para RS232, RS485 e LIN 1.2;
* Dois comparadores analógicos com entradas multiplexadas;
* Módulo PSP (*Parallel Slave Port*).

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 29



## 01-03PINAGEM DO PIC18F4520

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 30



## PINO MULTIPLEXADO

* + Um pino está multiplexado quando ele é capaz de desempenhar mais de uma função. É necessário, no entanto, que o pino seja configurado para executar uma das funções para as quais ele está apto. A configuração que define como o pino irá funcionar é normalmente feita por meio do programa. Algumas funções, entretanto, podem ser ativadas por meio dos bits de configuração na hora da gravação. Vejamos o caso a seguir.

|  |  |  |
| --- | --- | --- |
| **Pino 1** | **/MCLR/VPP/RE3** | |
| /MCLR | Função *Reset* | ST |
| VPP | Tensão de programação |  |
| RE3 | Pino digital | TTL |

* + Veja que o pino 1 do PIC18F4520 é multiplexado com tr„s funções diferentes. É importante observar que somente uma das funções pode esta ativa de cada vez. Quando uma função é ativada, automaticamente são desativadas as outras funções. Neste caso a função reset é ativada na hora da programação por meio dos bits de configuração. Estando a função /MCLR desativada a função digital do pino será ativada (RE3).

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 31



## PINO DIGITAIS (pinos de I/O)

* + - O PIC18F4520 possui 40 pinos.
    - Port A, Port B, Port C, Port D e Port E.
    - Todos os pinos são multiplexados.
    - Port A (RA7:RA0)
    - Port B (RB7:RB0)
    - Port C (RC7:RC0)
    - Port D (RD7:RD0)
    - Port E (RE3:RE0).
    - 36 pinos de I/O.
    - O pino RE3 só pode ser configurado como entrada digital.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 32



## CONFIGURAÇÃO DO OSCILADOR

* O PIC18F4520 possui dez formas diferentes de funcionamento do oscilador. Os bits de configuração FOSC2:FOSC0 (CONFIG1H<2:0>) são os responsáveis pela configuração do oscilador. As opções disponíveis para o oscilador são:

**LP**: cristal de baixa potência (até 200KHz).

**XT**: cristal/ressonador (até 4MHz).

**HS**: cristal/ressonador de alta frequência (acima de 4MHz).

**HSPLL**: cristal/ressonador de alta frequência com o PLL habilitado.

**RC**: RC externo com saída de *clock*. Essa opção fornece ao pino OSC2/CLKO/RA6 um

sinal digital com frequ„ncia quatro vezes menor que a do oscilador principal (Fosc/4).

**RCIO**: RC externo. Nessa opção o RA6 funciona como pino digital.

**INTIO1**: oscilador interno com Fosc/4 no pino RA6 e pino RA7 configurado como digital.

**INTIO2**: oscilador interno com RA6 e RA7 configurados como pinos digitais.

**EC**: oscilador externo com saída de *clock*. Essa opção fornece no pino OSC2/CLKO/RA6 um sinal digital com frequ„ncia quatro vezes menor que a do oscilador principal.

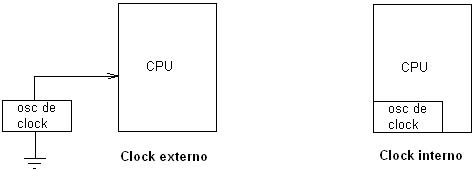
**ECIO**: oscilador externo. Nessa opção o pino RA6 funciona como digital.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 33



## OSCILADOR

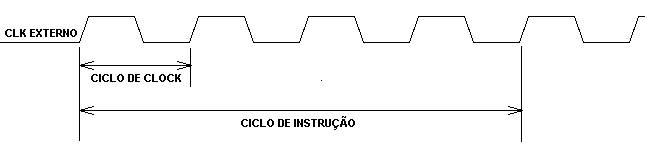
* + Qualquer instrução executada pela CPU utiliza como refer„ncia um sinal de *clock* que é gerado por um oscilador. O Oscilador pode ser externo ou interno.

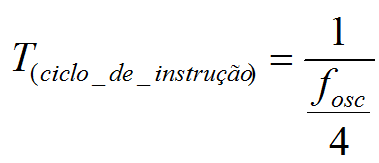


**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 34



## SINAL DE CLOCK

* O sinal de *clock* é uma onda quadrada que sincroniza a execução das instruções executadas pela CPU.
* Os Microcontroladores PIC gastam quatro ciclos de *clock* (ciclo de instrução) para executar uma instrução, salvo algumas exceções. O ciclo de instrução também é chamado de *clock* interno.



**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 35



## OSCILADOR CRISTAL/RESSONADOR

* Boa precisão do oscilador, como aquelas que envolvem o uso de temporizadores, por exemplo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de oscilador** | **Frequência do cristal** | **Valores típicos de**  **capacitores testados** | |
| **C1** | **C2** |
| LP | 32kHz | 30pF | 30pF |
| XT | 1MHz | 15pF | 15pF |
| 4MHz | 15pF | 15pF |
| HS | 4MHz | 15pF | 15pF |
| 10MHz | 15pF | 15pF |
| 20MHz | 15pF | 15pF |
| 25MHz | 0pF | 5pF |
| 25MHz | 15pF | 15pF |

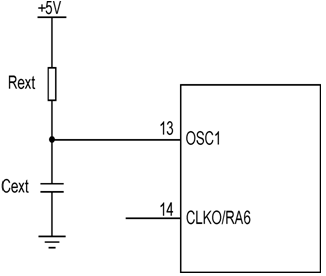
* Figura 1.4: Oscilador a cristal
* Figura 1.5: Ressonador de três pinos.
* A opção para os bits de configuração deve ser LP, XT, HS ou HSPLL e a frequ„ncia do oscilador será definida pelo cristal/ressonador.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 36



## OSCILADOR RC

* + Quando a precisão do *clock* não é essencial para uma determinada aplicação, o oscilador com RC externo pode ser uma boa opção. A frequ„ncia do sinal de *clock* é determinada pela tensão de alimentação, pelos valores de RC e pela variação de temperatura.



* + - Na configuração RC do oscilador, um sinal digital com Fosc/4 é fornecido no pino OSC2/CLK0/RA6. Na configuração RCIO é ativada a função RA6 no pino 14.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 37

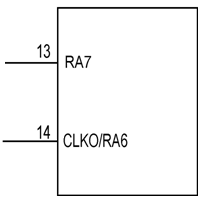
## OSCILADOR INTERNO

* + O PIC18F4520 possui dois osciladores internos que, se ativados, dispensam a utilização

de componentes externos.

* INTOSC - frequência de 8MHz e pode ser utilizado como oscilador de *clock*. Tem associado um recurso chamado *postscaler* que permite prover frequências na faixa de 31kHz - 4MHz. É habilitado quando é selecionada uma frequência de *clock* dentro da faixa de 125kHz a 8MHz.
* INTRC, provê uma frequência de 31kHz. É habilitado se ele for selecionado como origem do sinal de *clock*. Ele também é habilitado automaticamente quando um dos seguintes recursos é selecionado:



* + *Power-up Timer*



* + *Fail-Safe Clock Monitor*
  + *Watchdog Timer*
  + *Two-Speed Start-up*
  + O oscilador interno descarta a utilização de componentes externos. O pino 13 pode ser utilizado como digital e o pino 14 pode ser configurado como pino digital ou fornecer um sinal digital com a Fosc/4.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 38



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Registrador OSCTUNE** | | | | **Endereço F98h** | | | |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| R/W - 0 | R/W - 0 | - | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 | R/W - 0 |
| INTSRC | PLLEN | - | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 |

Bit 7: **INTSCR**: bit de seleção da origem da baixa frequ„ncia do oscilador interno

1 = 31,25jHz derivado do oscilador principal INTOSC (divisor por 256 habilitado) 0 = 31jHz derivado diretamente do oscilador INTRC

Bit 6: **PLLEN**: bit de seleção PLL para o oscilador INTOSC

1 = PLL habilitado para o oscilador principal INTOSC (4MHz e 8MHz somente) 0 = desabilita o PLL para o oscilador principal INTOSC

Bit 5: Não implementado: lido como 0

Bit 4:0: **TUN4:TUN0**: bits de seleção de frequ„ncia do *clock*

01111: frequ„ncia máxima

• •

• •

00000: frequ„ncia central. O módulo oscilador funciona na frequ„ncia calibrada. 11111:

• •

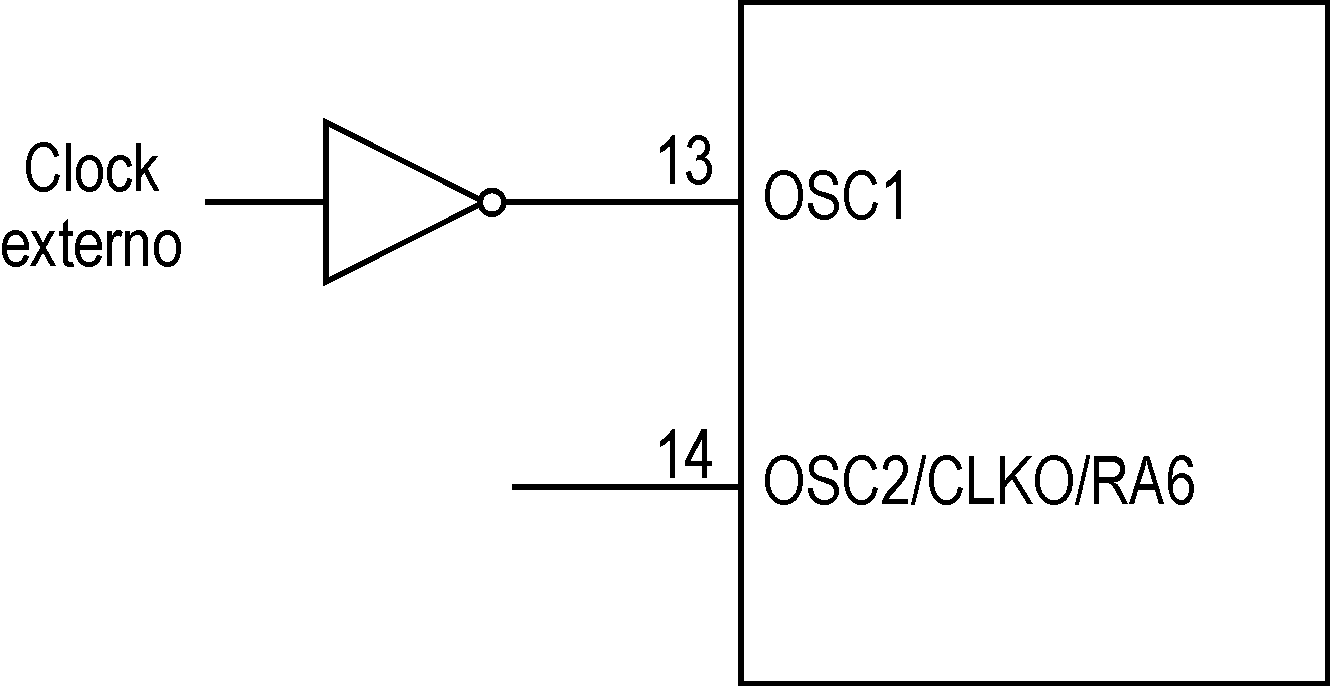
• •

10000: frequ„ncia mínima

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 39



## OSCILADOR EXTERNO

* + Seja para obter um alto nível de precisão do *clock* ou para sincronizar o microcontrolador com outros dispositivos presentes no sistema, o PIC18F4520 permite que um sinal de *clock* externo seja aplicado ao pino OSC1, como mostra a figura abaixo.
  + Na configuração EC do oscilador um sinal digital com Fosc/4 é fornecido no pino OSC2/CLKO/RA6. Na configuração ECIO é ativada a função RA6 no pino 14.
  + Um oscilador externo pode ser conectado ao pino OSC1 com uma das seguintes opções: EC, ECIO ou HS habilitada.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 40



## HSPLL

(*High Speed Phased Looked Loop*)

* O PLL (*Phase Looked Loop*) é um recurso utilizado em associação ” opção HS e pode ser utilizado para se obter uma frequ„ncia de *clock* quatro vezes maior do que aquela fornecida pelo cristal.
* Com um cristal de 10MHz pode-se obter uma frequ„ncia de *clock* de 40MHz com a habilitação do PLL. O PLL será útil quando o projetista desejar reduzir a emissão de EMI que é maior com a utilização de cristais de alta frequ„ncia.
* O PLL só pode ser habilitado com a opção HS e é ativado quando é selecionada a opção HSPLL do oscilador.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 41



## CHAVEAMENTO DO OSCILADOR

* + O PIC18F4520 possui tr„s fontes diferentes que podem gerar o sinal de *clock.* O sinal de *clock* pode vir do oscilador primário, do oscilador secundário ou de um dos osciladores internos.

**Oscilador primário:** inclui as opções cristal/ressonador externo, RC externo, oscilador externo e os osciladores internos.

**Oscilador secundário:** está associado ao módulo Timer1 e inclui a conexão de um cristal de baixa frequ„ncia entre os pinos T1OSO e T1OS1. Para que o oscilador secundário possa ser utilizado é necessário que o módulo Timer1 esteja habilitado. O cristal de baixa frequ„ncia conectado entre os pinos T1OSO e T1OS1 normalmente é de 32,768jHz e pode ser utilizado como base de tempo para um RTC (*Real Time Clock*).

**Osciladores internos:** além de fazerem parte do set de opções do oscilador primário, estão disponíveis como fontes de *clock* para o modo de energia gerenciado (*power-managed*). O oscilador interno INTRC ainda é utilizado como fonte de *clock* para vários recursos especiais, tais como WDT e *Fail-Safe Clock Monitor*.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 42



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Registrador OSCCON** | | | | **Endereço FD3h** | | | |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| R/W - 0 | R/W - 1 | R/W - 0 | R/W - 0 | R(1) | R - 0 | R/W - 0 | R/W - 0 |
| IDLEN | IRCF2 | IRCF1 | IRCF0 | OSTS | IOFS | SCS1 | SCS0 |

Bit 7: **IDELN**: bit de seleção do modo *Idle*

1 = dispositivo entra no modo *Idle* quando a instrução SLEEP é executada

0 = dispositivo entra no modo *Sleep* quando a instrução SLEEP é executada

Bit 6:4: **IRCF2:IRCF0**: bit de seleção da frequ„ncia do oscilador interno

111 = 8MHz

110 = 4MHz

101 = 2MHz

100 = 1MHz (saída frequ„ncia *default* do INTOSC no *Reset*) 011 = 500jHz

010 = 250jHz

001 = 125jHz

000 = 31jHz (de um ou outro INTOSC/256 ou INTRC)(2)

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 43



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Registrador OSCCON** | | | | **Endereço FD3h** | | | |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| R/W - 0 | R/W - 1 | R/W - 0 | R/W - 0 | R(1) | R - 0 | R/W - 0 | R/W - 0 |
| IDLEN | IRCF2 | IRCF1 | IRCF0 | OSTS | IOFS | SCS1 | SCS0 |

Bit 3: **OSTS**: bit de *status* de *Time-out* do *Oscillator Start-up Timer*

1 = ocorreu *Time-out* do *Oscillator Start-up Timer*; oscilador primário está sendo executado 0 = *Oscillator Start-up Timer* está sendo executado; oscilador primário não está pronto

Bit 2: **OOFS**: bit de sinalização de estabilidade da frequ„ncia do INTOSC

1 = frequ„ncia do INTOSC está estabilizada

0 = frequ„ncia do INTOSC não está estabilizada

Bit 1:0: **SCS1:SCS0**: bits de seleção do sistema de *clock*

1x: osciladores internos 01: oscilador secundário

00: oscilador principal

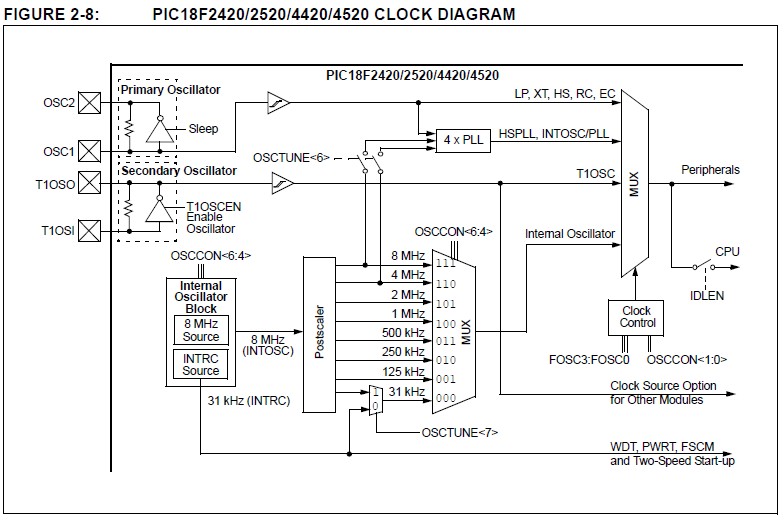
**Notas:** (1) - Nível lógico após o *Reset* depende do *status* do bit de configuração IESO.

(2) - Origem selecionada pelo bit OSCTUNE<7>.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 44



## OSCILADOR DO PIC18F4520



***Fonte: PIC18F2420/2520/4420/4520 Data Sheet DS39631D. Baixado de*** [***www.microchip.com***](http://www.microchip.com/)

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 45



MODOS DE ENERGIA GERENCIADA

# (power-managed)

* + O PIC18F4520 dispõe de um total de sete modos de funcionamento aplicados ” ger„ncia e conservação eficiente de energia. Os modos de energia gerenciada estão divididos em tr„s categorias. São elas:
  + *Run modes*
  + *Idle modes*
  + *Sleep mode*
  + Essas categorias definem que parte do microcontrolador receberá o sinal de

*clock* e, ”s vezes, com que velocidade.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 46



MODOS DE ENERGIA GERENCIADA

# (power-managed)

* + As categorias *Run* e *Idle modes* estão disponíveis para qualquer origem do sinal de *clock*, seja o oscilador principal, oscilador secundário ou um dos osciladores internos.
  + O *Sleep mode* (modo *Sleep*), quando ativado, desliga o oscilador.
  + Quando um determinado modo de gerenciamento é ativado, duas decisões precisam ser tomadas: se a CPU irá ou não receber o sinal de *clock* e qual a origem do sinal de *clock*.
  + O bit IDELN (OSCCON<7>) define se a CPU irá ou não receber o sinal de *clock*, enquanto os bits SCS1:SCS0(OSCCON<1:0>) indicam a origem do *clock*.

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 47



MODOS DE ENERGIA GERENCIADA

# (power-managed)

* + A Tabela abaixo resume como a CPU e os periféricos do microcontrolador são afetados pelos diversos modos de gerenciamento, derivados de uma das tr„s categorias.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Modo** | **OSCCON<7,1:0>** | | **Sinal de *clock*** | | **Disponibilidade do *clock***  **e origem do oscilador** |
| **IDLEN(1)** | **SCS1:SCS0** | **CPU** | **Periféricos** |
| Sleep | 0 | N/A | Off | Off | Nenhum - todos os *clocks* estão  desligados |
| PRI\_RUN | N/A | 00 | Recebe | Recebe | Primário - LP, XT, HS, HSPLL,  RC, EC e oscilador interno**(2)**  Este é o modo de execução normal |
| SEC\_RUN | N/A | 01 | Recebe | Recebe | Secundário - oscilador do Timer1 |
| RC\_RUN | N/A | 1x | Recebe | Recebe | Osciladores internos**(2)** |
| PRI\_IDLE | 1 | 00 | Off | Recebe | Primário - LP, XT, HS, HSPLL,  RC, EC |
| SEC\_IDLE | 1 | 01 | Off | Recebe | Secundário - oscilador do Timer1 |
| RC\_IDLE | 1 | 1x | Off | Recebe | Osciladores internos**(2)** |
| **Nota:** 1. O bit IDELN reflete o valor quando a instrução SLEEP é executada.  2. Inclui o *postscaler* do oscilador INTOSC e o oscilador INTRC. | | | | | |

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 48



## EXERCÍCIOS

* 1. Qual a frequ„ncia de *clock* máxima na qual pode operar o PIC18F4520?
  2. Quantos pinos de I/O possui o PIC18F4520 e como eles estão

divididos?

* 1. De quantas maneiras diferentes pode funcionar o oscilador do PIC18F4520? Quais são elas?
  2. Quando é indicada a utilização de um oscilador a cristal no PIC18F4520?
  3. O PIC18F4520 possui quantos osciladores internos? Quais são eles?
  4. O PIC18F4520 dispõe de tr„s fontes diferentes que podem gerar o sinal de *clock*. Como elas são classificadas?
  5. O PIC18F4520 dispõe de sete modos de funcionamento aplicados ” ger„ncia e ” conservação de energia. Quais são eles e como afetam a disponibilidade do *clock* para a CPU e os demais periféricos?

**Microcontroladores PIC18F4520 com Linguagem C Wagner da Silva Zanco** 49