

# Microprocessadores e Aplicações

## *Acetatos de apoio às aulas teóricas*

Ana Cristina Lopes  
Dep. Engenharia Electrotécnica  
<http://orion.ipt.pt> [anacris@ipt.pt](mailto:anacris@ipt.pt)

1. Introdução;
2. Relógio/ Ciclo de Instrução;
3. Pipelining;
4. Pinagem;
5. Gerador de Relógio;
6. Reset;
7. Organização da Memória;

- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- **Introdução**

- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

- Os MCs da família PIC 18FXX8 pertencem a uma classe de microcontroladores de 8 bits;
- Integram diversos periféricos;
- Seguem uma arquitectura de Harvard, em que a memória de dados se encontra separada da memória de programa;
- A separação da memória de dados da de programa possibilita a representação de instruções em mais de 8 bits;
- A família de MCs PIC 18FXX8 utiliza uma palavra por instrução;
- A arquitectura de Harvard, quando comparada com a de Von Newman, compreende a existência de um menor número de instruções;

- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- **Introdução**

- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

- Os microcontroladores com arquitectura de Harvard são também designados por "microcontroladores RISC (*Reduced Instruction Set Computer*)"
- Os microcontroladores com arquitectura de Von Neumann são também designados por "microcontroladores CISC (*Complex Instruction Set Computer*)"
- Os MCs da família PIC18FXX8 seguem uma arquitectura do tipo RISC e por isso possuem um número reduzido de instruções: 77 instruções;
- Regra geral as instruções são executadas num ciclo de máquina, excepto as instruções de salto e de ramificação;

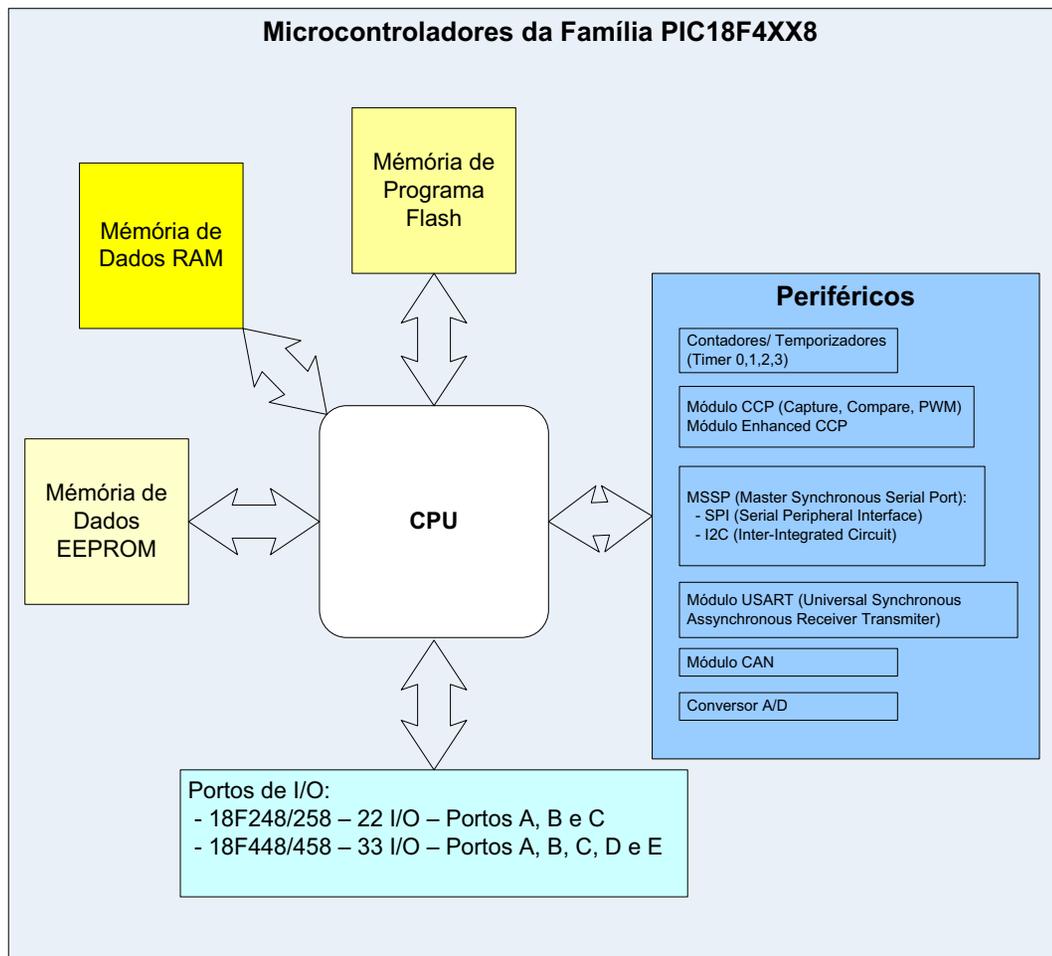
- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

● **Introdução**

- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

A Figura seguinte mostra um esquema genérico dos MCs da família PIC18FXX58.



● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

● **Introdução**

- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

A Tabela seguinte mostra as características dos MCs da família PIC18FXX.

Microcontrolador	PIC18F248	PIC18F258	PIC18F448	PIC18F458
Mem. de Programa flash (bytes)	16K	32K	16K	32K
Mem. de Programa # Single-Word Instructions	8192	16384	8192	16384
Mem. Dados SRAM (bytes)	768	1536	768	1536
Mem. Dados EEPROM (bytes)	256	256	256	256
I/O	22	22	33	33
10-bit A/D	5	5	8	8
Comparadores	-	-	2	2
CCP/ECCP	1/0	1/0	1/1	1/1
MSSP - SPI	Sim	Sim	Sim	Sim
MSSP - I2C	Sim	Sim	Sim	Sim
USART	Sim	Sim	Sim	Sim
Timers 8/16 bits	1/3	1/3	1/3	1/3

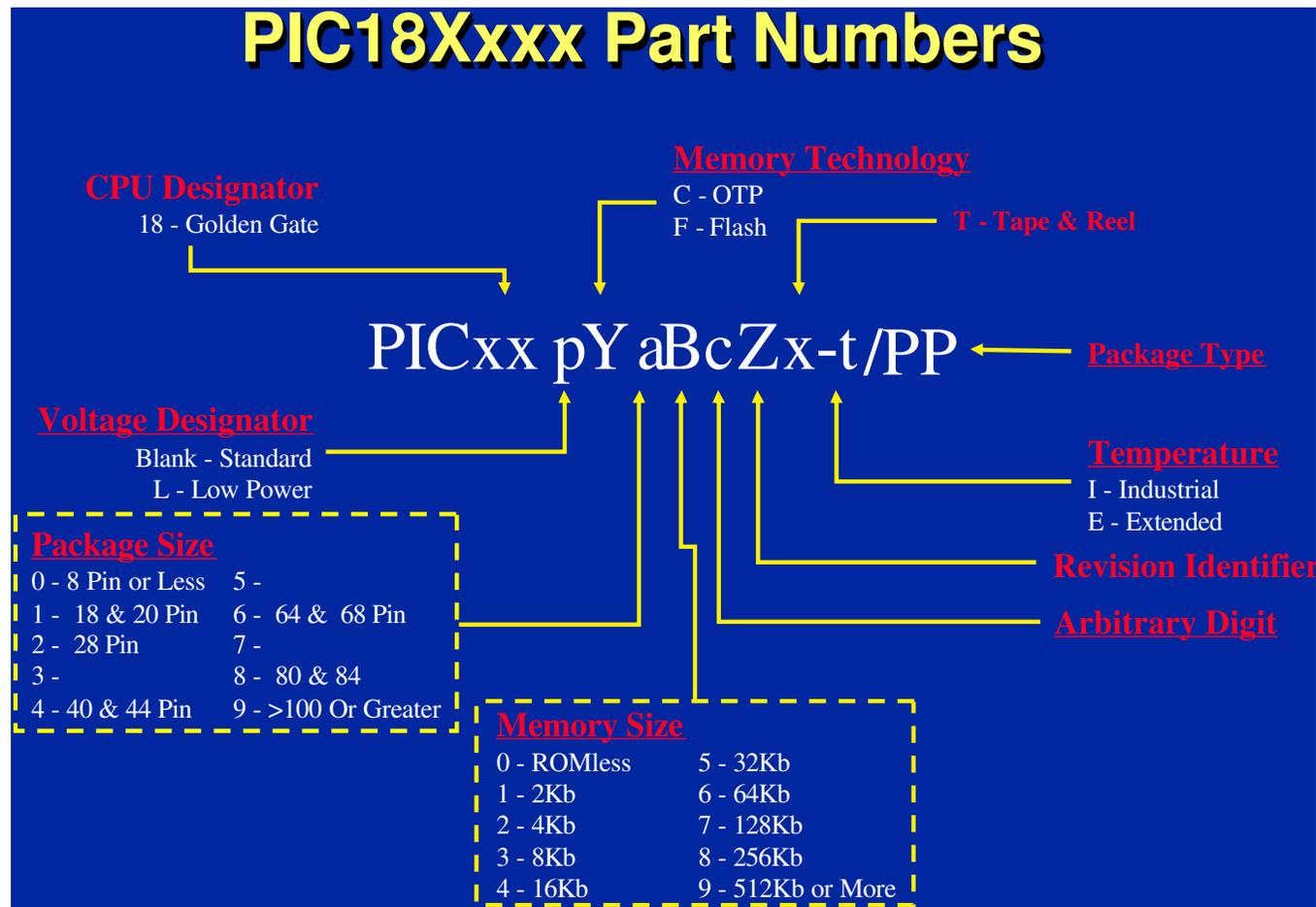
- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

● **Introdução**

- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

A Figura seguinte mostra o significado das referências dos microcontroladores PIC.



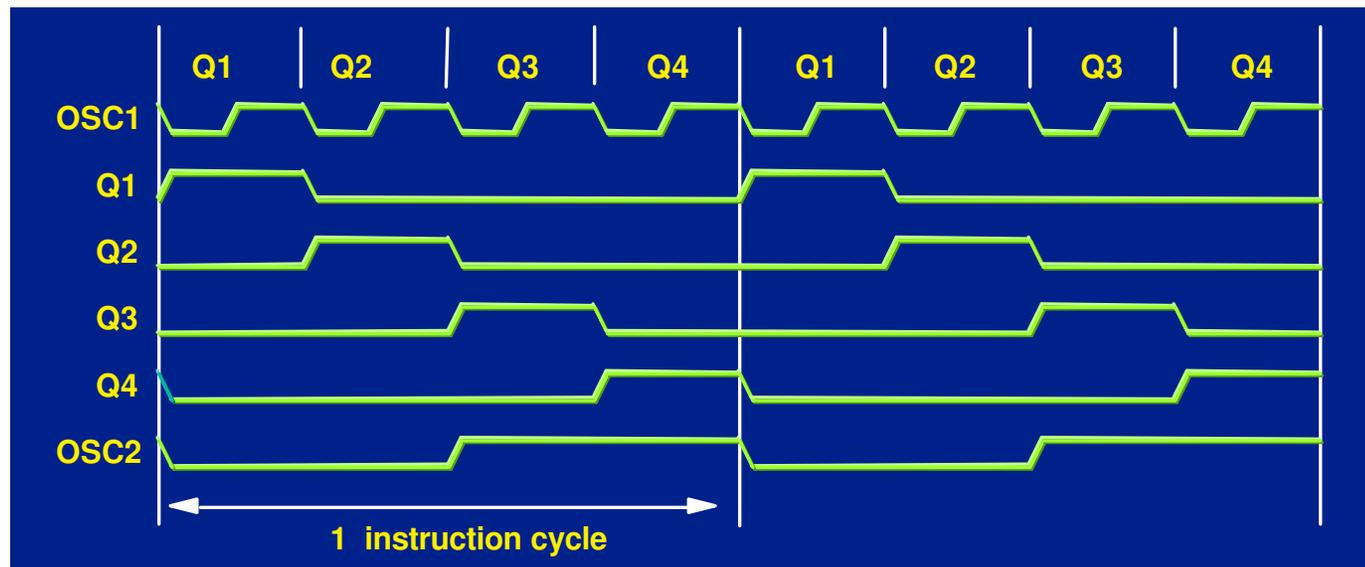
# Relógio/Ciclo de Instrução

- Introdução aos MCs PIC18FXX8

A Figura seguinte mostra o ciclo de instrução de um PIC18F458.

## Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória



- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

- Neste caso o sinal de relógio encontra-se dividido em 4 fases: Q1, Q2, Q3 e Q4;
- Estas 4 fases perfazem o chamado ciclo de instrução, ou ciclo de máquina;
- Para o caso do PIC18F458, o relógio pode no máximo ser de 40 MHz;
- Tal significa que um ciclo de instrução, ou ciclo de máquina, pode no mínimo demorar 100ns.

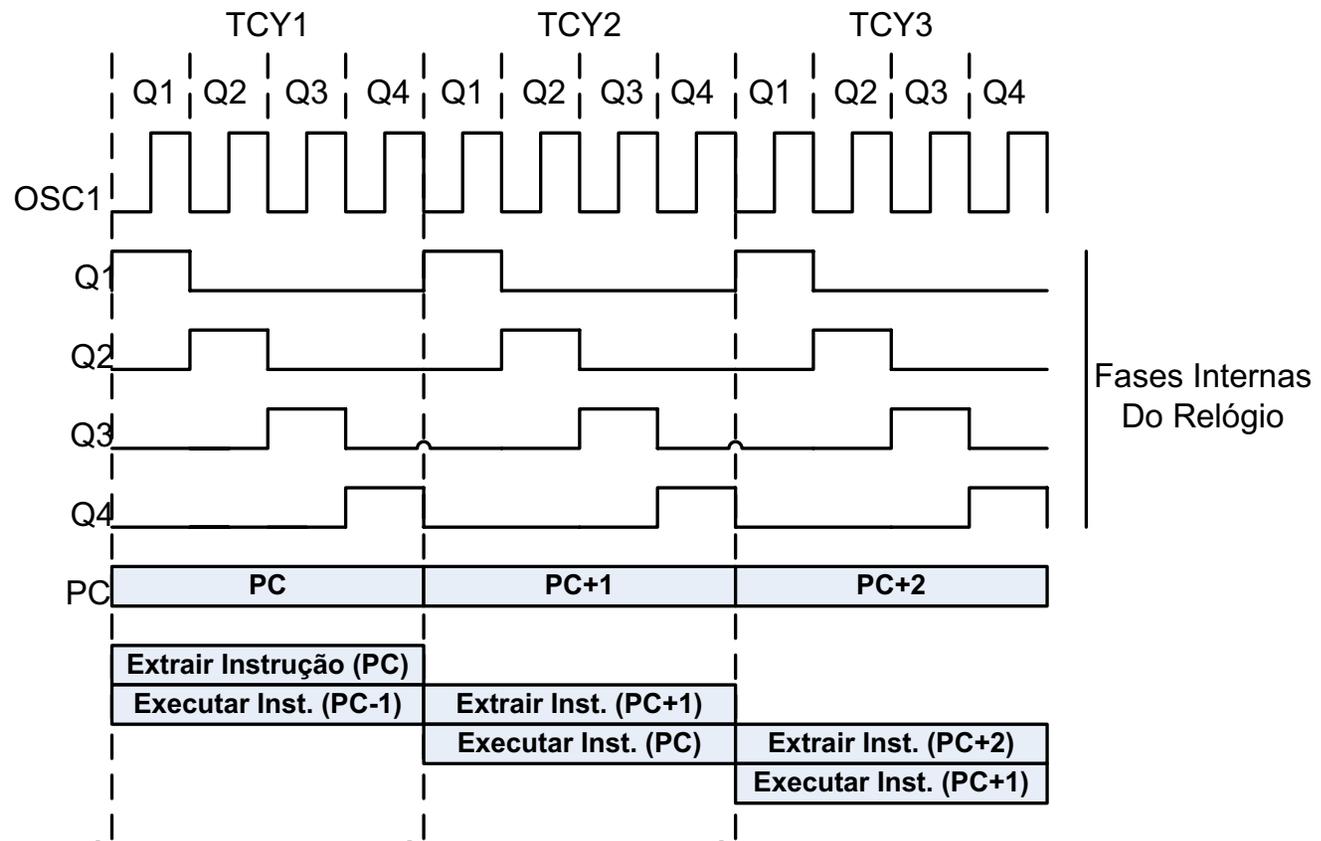
# Pipelining

- Introdução aos MCs PIC18FXX8

## Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

A Figura seguinte mostra a relação entre o ciclo de instrução (ou máquina) e a extracção e execução de instruções.



- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

- O circuito interno do microcontrolador divide o sinal de relógio em quatro fases, Q1, Q2, Q3 e Q4 que não se sobrepõem;
- A execução de uma instrução, é antecidida pela extracção da instrução que está na linha seguinte;
- O código da instrução é extraído da memória de programa em Q1 e é escrito no registo de instrução em Q4;
- A descodificação e execução dessa mesma instrução, faz-se entre as fases Q1 e Q4 seguintes;
- O contador de programa (Program Counter ou PC) guarda o endereço da próxima instrução a ser executada.

- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

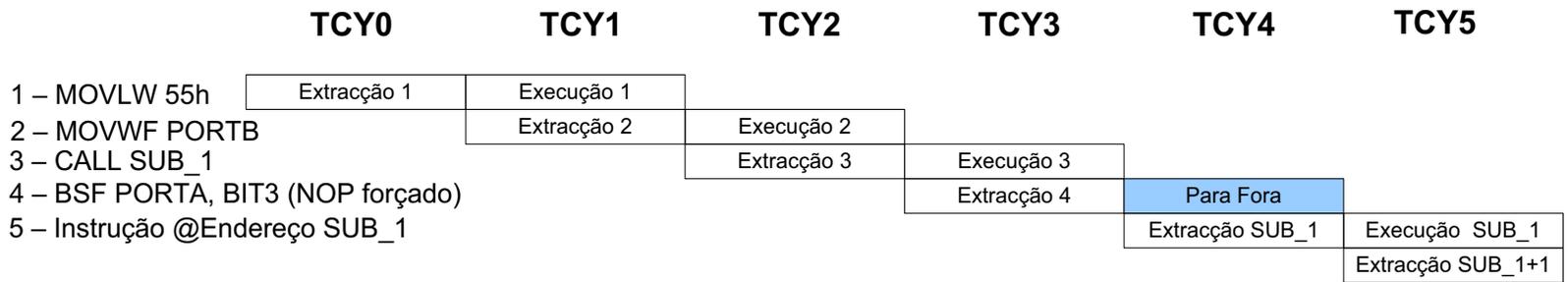
- A extracção do código de uma instrução da memória de programa, é feita num ciclo de instrução, enquanto que a sua descodificação e execução, são feitos no ciclo de instrução seguinte.
- Contudo, devido à sobreposição - pipelining (o microcontrolador ao mesmo tempo que executa uma instrução extrai simultaneamente da memória o código da instrução seguinte), podemos considerar que, para efeitos práticos, cada instrução demora um ciclo de instrução a ser executada.
- No entanto, se a instrução provocar uma mudança no conteúdo do contador de programa (PC), ou seja, se o PC não tiver que apontar para o endereço seguinte na memória de programa, mas sim para outro (como no caso de saltos ou de chamadas de subrotinas), então deverá considerar-se que a execução desta instrução demora dois ciclos.
- Isto acontece, porque a instrução vai ter que ser processada de novo, mas, desta vez, a partir do endereço correcto.
- O ciclo de chamada começa na fase Q1, escrevendo a instrução no registo de instrução (Instruction Register - IR). A descodificação e execução continua nas fases Q2, Q3 e Q4 do clock.

● Introdução aos MCs PIC18FXX8

A Figura seguinte mostra a o fluxograma das instruções no pipelining.

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória



Todas as intruções são de ciclo único, excepto as relativas a ramificações de programa, as quais demoram dois ciclos.

- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

- **TCY0:** é lido da memória o código da instrução MOVLW 55h;
- **TCY1:** é executada a instrução MOVLW 55h e é lida da memória a instrução MOVWF PORTB.
- **TCY2:** é executada a instrução MOVWF PORTB e lida a instrução CALL SUB1.
- **TCY3:** é executada a chamada (call) de um subrotina CALL SUB1 e é lida a instrução BSF PORTA,BIT3. Como esta instrução não é a que nos interessa, ou seja, não é a primeira instrução da subrotina SUB1, cuja execução é o que vem a seguir, a leitura de uma instrução tem que ser feita de novo (este é um exemplo de uma instrução a precisar de mais que um ciclo).
- **TCY4:** este ciclo de instrução é totalmente usado para ler a primeira instrução do subrotina no endereço SUB1.
- **TCY5:** é executada a primeira instrução do subrotina SUB1 e lida a instrução seguinte.



O PIC18FXX8 trabalha com 8 modos de osciladores, os quais são programáveis por três bits de configuração, designadamente: FOSC2, FOSC1 e FOSC0 (bits 0,1 e 2 do registo CONFIG1H). Os modos do oscilador encontram-se listados de seguida:

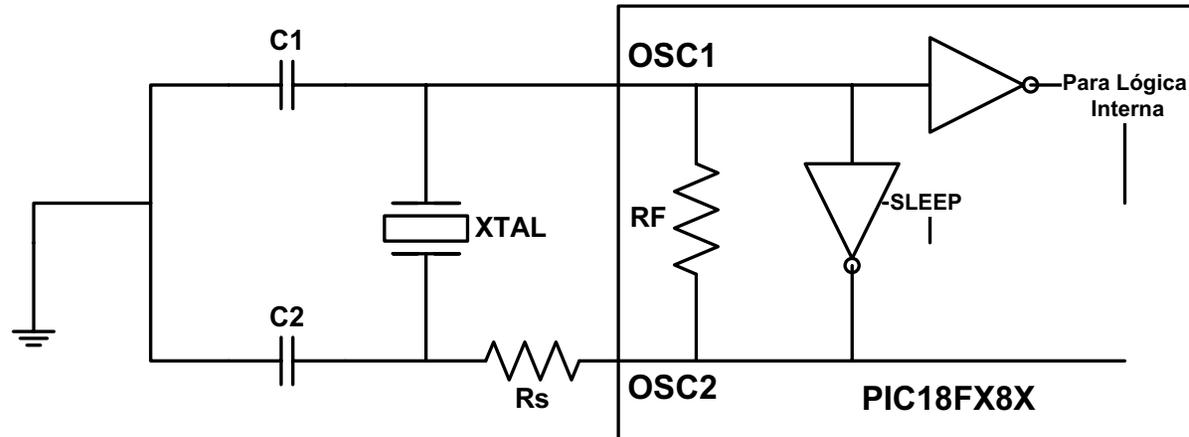
1. LP - Cristal de baixa potência - *Low-Power Crystal* (200KHZ no máximo);
2. XT - Cristal *Crystal/ Resonator* (4MHZ no máximo);
3. HS - Cristal rápido *High-Speed Crystal/ Resonator*(40MHZ no máximo);
4. HS4 - Cristal rápido com PLL (*Phase Locked Loop*) - *High-Speed Crystal/ Resonator with PLL* (10MHZ vezes 4 no máximo);
5. RC - Circuito RC externo (4MHZ no máximo);
6. RCIO - Circuito RC com pino de I/O (40MHZ no máximo);
7. EC - Relógio Externo (40MHZ no máximo);
8. ECIO - Relógio externo com pino de I/O (200KHZ no máximo);
9. INTOSC - Adicionalmente existe ainda o oscilador RC interno (30/500 kHz, 1/4/8 MHz).

● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

Nos modos 1 a 4 utiliza-se um cristal, o qual é ligado aos pinos OSC1 e OSC2, de forma a estabelecer a oscilação. A figura seguinte mostra como se processam as ligações do cristal aos pinos do microcontrolador.



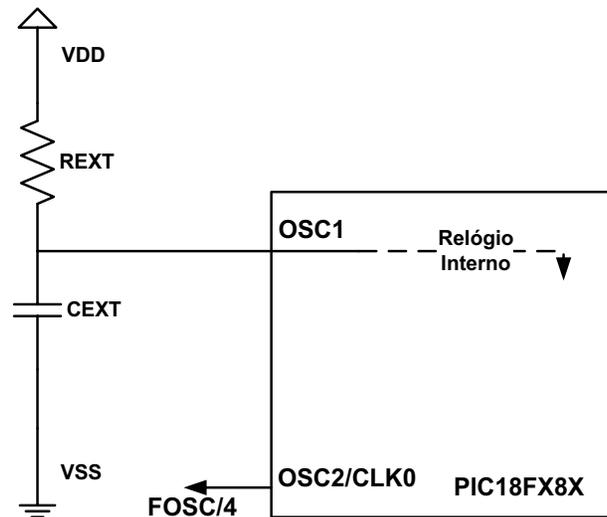
Consultar manual (disponível na página) para obter valores dos componentes

- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

Para aplicações insensíveis à temporização, a utilização de um circuito RC ou RCIO para gerar o sinal de relógio apresentam-se como uma solução simples e de baixo custo. A frequência do sinal gerado pelo circuito RC é função da tensão aplicada, de REXT, CEXT e da temperatura. Note-se que se obtém a frequência a dividir por 4 no pino OSC2. Caso este sinal não sirva para a aplicação em questão é preferível usar o modo RCIO, em que o pino OSC2 passa a ser um pino geral de I/O (bit 6 do porto A - RA6). A Figura seguinte mostra as ligações necessárias.



Valores Recomendados:  $3\text{KOHM} \leq \text{REXT} \leq 100\text{kOHM}$   
 $\text{CEXT} > 20\text{pf}$

# Relógio Externo

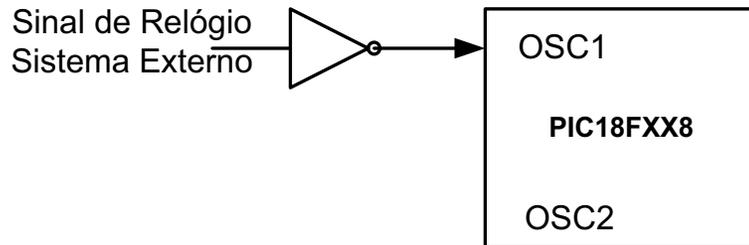
- Introdução aos MCs PIC18FXX8

## Introdução aos MCs PIC18FXX8

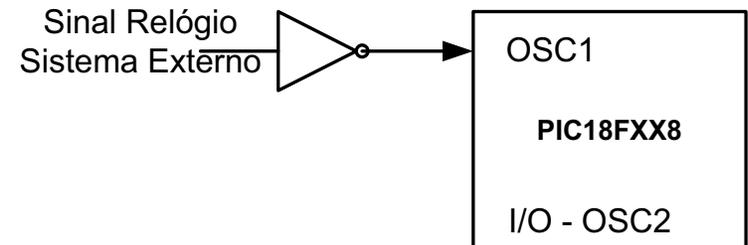
- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

Os modos EC e ECIO requerem um sistema externo de relógio que deve ser ligado ao pino OSC1. Nestes modos não é requerido o tempo de arranque após um Power-ON-Reset ou quando se retorna de um Modo Sleep. Tal como no caso anterior, caso o sinal de relógio a dividir por quatro não seja necessário para a aplicação, o pino OSC2 pode ser utilizado como pino de I/O. A Figura seguinte mostra as ligações necessárias.

EC - Modo Relógio Externo



EC - Modo Relógio Externo com pino de I/O



● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

- O PIC18FXX8 integra um circuito opcional PLL, para utilizadores que pretendam multiplicar a frequência de entrada por 4.
- O circuito PLL só fica activo quando os bits respeitantes ao modo do oscilador são programados para o modo HS, ou seja FOSC2 FOSC1 FOSC0 = 1 0 0 (registo CONFIG1H), estes registos de configuração serão falados em detalhe posteriormente (características especiais da CPU).
- Note-se que o modo do oscilador é seleccionado por software.

● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

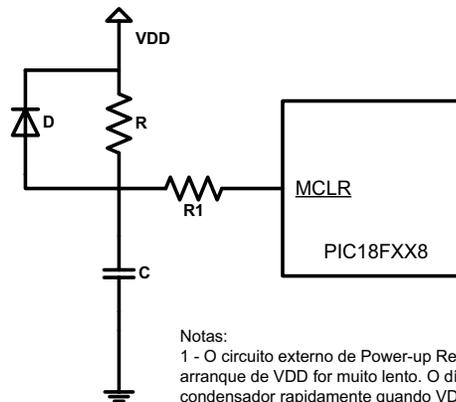
- O reset é utilizado para colocar o microcontrolador num estado conhecido;
- Na prática isto significa que às vezes o microcontrolador pode comportar-se de um modo inadequado em determinadas condições indesejáveis.
- De modo a que o seu funcionamento normal seja restabelecido, é preciso fazer o reset do microcontrolador, isto significa que alguns dos seus registos vão conter valores iniciais pré-definidos, correspondentes a uma posição inicial.
- O reset também pode ser utilizado quando ocorre uma interrupção por parte de um outro dispositivo, ou quando se pretende que o microcontrolador esteja pronto para executar um programa.

● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

- O PIC18FXX8 integra um filtro (pino  $\overline{MCLR}$ ) que detecta e ignora pequenos impulsos;
- Note-se que o o reset provocado pelo Watchdog timer não coloca o pino  $\overline{MCLR}$  num nível baixo;
- Ao contrário do que se passa com outras famílias de PICs, a Microchip recomenda que o pino  $\overline{MCLR}$  não seja directamente ligado a Vdd, devendo utilizar-se um circuito semelhante ao apresentado na Figura seguinte.



Notas:

1 - O circuito externo de Power-up Reset só é necessário se o arranque de VDD for muito lento. O diodo D ajuda a descarregar o condensador rapidamente quando VDD vai a zero.

2 - Recomenda-se que  $R < 40\text{KOHM}$ .

3 - R1 deve variar entre 1 e 100OHM (serve para limitar o fluxo de corrente que é fornecido ao pino MCLR).

● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

O microcontrolador PIC18FXX8, admite várias formas de reset, designadamente:

- Reset quando se liga a alimentação (Power-On-Reset);
- Reset durante o funcionamento normal, quando se põe a nível lógico baixo o pino  $\overline{MCLR}$  do microcontrolador;
- Reset durante o modo Sleep;
- Reset quando o temporizador do Watchdog transborda (passa para 0 depois de atingir o valor máximo);
- Brown-Out Reset Programável (PBOR);
- Instrução RESET;
- Reset pilha cheia (*Stack Full Reset*);
- Stack underflow Reset.

- Introdução aos MCs PIC18FXX8

#### Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

O Power-On-Reset, ocorre sempre que é ligada a alimentação do microcontrolador e serve para trazer os registos para um estado inicial (consultar manual Tab3-2).

O reset durante o funcionamento normal resulta da aplicação de um valor lógico baixo ao pino  $\overline{MCLR}$  durante o funcionamento normal do microcontrolador. É muito usado durante o desenvolvimento de um programa.

Durante um reset, os locais de memória da RAM (registos) não são alterados. Ou seja, os conteúdos destes registos, são desconhecidos durante o restabelecimento da alimentação, mas mantêm-se inalterados durante qualquer outro reset.

Ao contrário dos registos normais, os SFR (Tab3-3 do manual) (registos com funções especiais) são reiniciados com um valor inicial pré-definido.

Um dos mais importantes efeitos de um reset, é introduzir no contador de programa (PC), o valor zero (0000), o que faz com que o programa comece a ser executado a partir da primeira instrução.

- Introdução aos MCs PIC18FXX8

## Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- **Reset**
- Organização da Memória

O bit de configuração, BOREN, pode desactivar (se estiver a zero), ou activar (se estiver a 1), o circuito Brown-Out Reset.

Se VDD cair abaixo de determinado parâmetro durante um tempo superior a um determinado valor, o Brown-Out Reset origina o reset do microcontrolador;

O microcontrolador permanecerá no estado de Brown-Out Reset até que VDD suba acima do parâmetro pré-estabelecido;

Assim que VDD atinja um valor acima do parâmetro pré-estabelecido, o Power-up Timer será invocado e manterá o microcontrolador em estado de RESET durante um tempo (atraso) pré-especificado;

Caso o VDD caia durante o tempo de atraso mencionado, o microcontrolador volta ao estado de Brown-Out Reset.

O PIC18FXX8 tem dois blocos de memória separados, um para dados e o outro para o programa:

- A memória EEPROM e os registos de uso genérico (GPR) e registos SFR na memória RAM constituem o bloco para dados;
- A memória Enhanced FLASH constitui o bloco de programa.

## Memória de programa:

- É implementada com tecnologia flash;
- Permite ser programada diversas vezes antes do microcontrolador estar instalado num dispositivo, mesmo após a instalação é possível alterar o programa e os parâmetros nele contidos;
- O PIC18F458 tem um PC de 21 bits que tem a capacidade de endereçar até 2 MBytes de espaço de memória de programa.
- Os endereços 0000h, 0008h e 0018h estão, respectivamente, reservados para o reset e para os vectores de interrupções.
- O PIC18F458 pode armazenar até 16K de instruções simples (de 16 bits).

# Organização da Memória

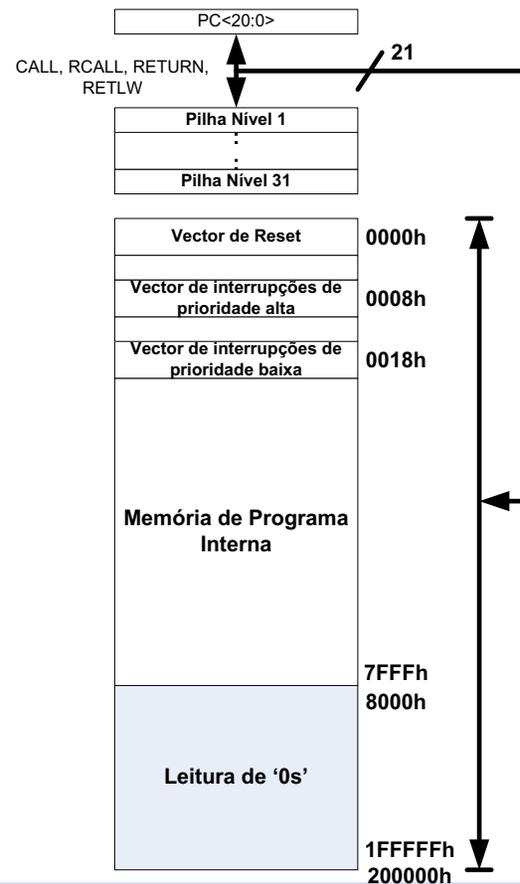
● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset

● Organização da Memória

O acesso a um local de memória entre a memória física implementada e os 2MBytes de memória internos, resultam na leitura de 0s (instrução NOP). A Figura seguinte ilustra a estrutura da memória de programa do PIC18F458:



- Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

**Memória de dados:** A memória de dados do PIC18F458 compreende uma memória EEPROM e uma memória RAM:

- A memória EEPROM consiste em 256 bytes, cujos conteúdos não se perdem durante uma falha na alimentação, e pode ser acedida indirectamente através dos registos EEADR e EEDATA.
- A memória de dados propriamente dita é implementada em RAM estática (SRAM);
- Cada registo da memória de dados possui um endereço de 12 bits, o que permite endereçar até 4096 bytes de memória de dados;
- O mapa da memória de dados encontra-se dividido em em 16 bancos que contém 256 bytes cada;

● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset

● Organização da Memória

## Memória de dados - continuação:

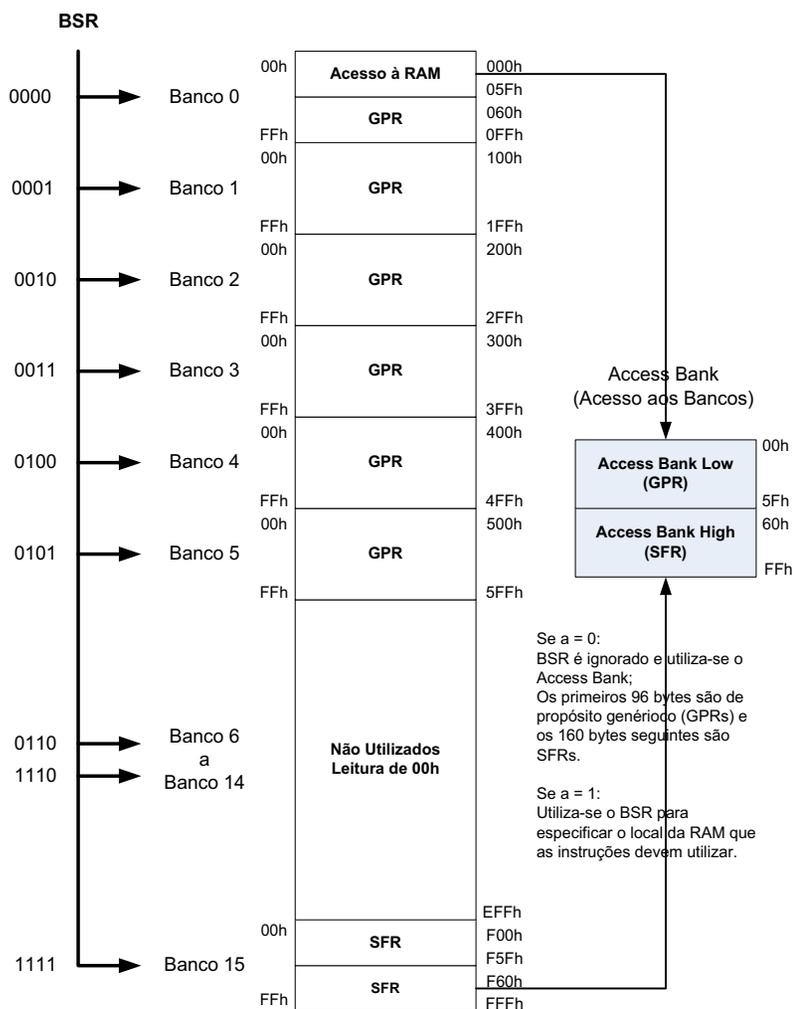
- Os 4 bits menos significativos do registo BSR (Bank Select Register) servem para seleccionar o banco de dados que vais ser acedido (os 4 bits mais significativos não se utilizam);
- A memória de dados contém ainda SFRs e GPR (*General Purpose Registers*);
- Os SFRs são utilizados para controlar e monitorizar o estado das funções periféricas do Microcontrolador (os endereços dos SFR começam no último byte do do Banco 15 (FFFh));
- Os GPRs são utilizados para armazenamento de dados e cálculos intermédios (os endereços dos GPRs começam no primeiro byte do banco 0).

- Introdução aos MCs PIC18FXX8

A Figura seguinte ilustra a estrutura da memória de dados do PIC18F458:

## Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória



● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

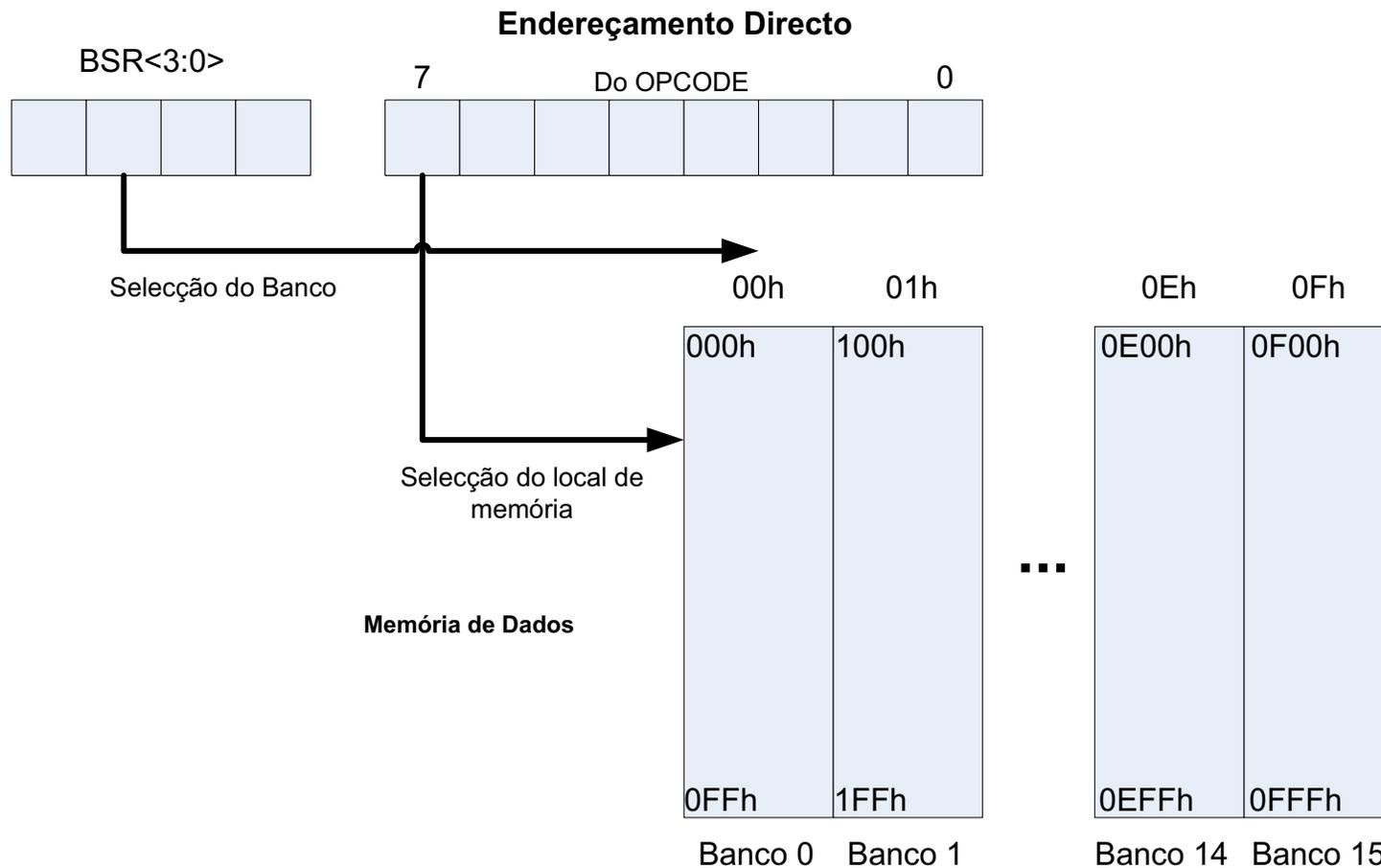
## Endereçamento Directo:

- A memória de dados encontra-se dividida em 16 bancos de dados;
- No endereçamento directo, o registo BSR deve ser configurado de modo a seleccionar o banco pretendido;
- Relativamente ao registo BSR, apenas os 4 bits menos significativos são utilizados para realizar a selecção do Banco de dados, os bits mais significativos não se utilizam;
- A instrução MOVLB serve para realizar a selecção do banco;
- Se o banco seleccionado não estiver implementado, qualquer leitura devolve apenas zeros e qualquer escrita será ignorada;
- Os bits do registo Status serão afectados pela execução da instrução;
- A instrução MOVFF ignora o BSR porque o endereço de 12 bits já se encontra embebido na própria instrução.

A Figura seguinte ilustra como se processa o endereçamento directo:

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset

● Organização da Memória



## Endereçamento Indirecto:

- Existem três registos de 12 bits associados ao endereçamento indirecto (responsáveis por endereçar 4096 bytes); para armazenar os 12 bits são necessários três registos de 8 bits, designadamente:
  - ◆ FSR0 - FSR0H:FSR0L;
  - ◆ FSR1 - FSR1H:FSR1L;
  - ◆ FSR2 - FSR2H:FSR2L;
- Adicionalmente existem registos INDF0, INDF1 e INDF2 que não estão implementados fisicamente;
- A leitura ou escrita destes últimos activa o endereçamento indirecto, em que o valor contido no FSR correspondente corresponde ao endereço do local de memória que se pretende aceder de forma indirecta;
- Ou seja: caso se escreva um valor em INDF0, esse valor será escrito no endereço apontado por FSR0H:FSR0L;
- A leitura indirecta de um registo INDF (através de um FSR), resultará numa leitura de zeros e a escrita indirecta de um registo INDF é equivalente a uma instrução NOP e o registo de estado não é afectado.

● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset

● Organização da Memória

## Endereçamento Indirecto:

- Cada registo FSR tem um registo INDF associado mais quatro endereços de registos adicionais;
- A realização de uma operação a qualquer um destes cinco registos determina como o FSR vai ser modificado durante o endereçamento indirecto
- Quando o acesso a dados é realizado a partir de um dos cinco locais  $INF_n$ , o endereço seleccionado configura o FSR da seguinte forma:
  - ◆  $INDF_n$  - Não altera  $FSR_n$  após o endereçamento indirecto;
  - ◆  $POSTDEC_n$  - Auto-decrementa  $FSR_n$  após endereçamento indirecto;
  - ◆  $POSTINC_n$  - Auto-incrementa  $FSR_n$  após endereçamento indirecto;
  - ◆  $PREINC_n$  - Auto-incrementa  $FSR_n$  antes do endereçamento indirecto;
  - ◆  $PLUSW_n$  - Utiliza o valor do WREG como offset de  $FSR_n$ ; Não modifica os conteúdos de ambos os registos após o endereçamento indirecto;

● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset

● Organização da Memória

## Endereçamento Indirecto:

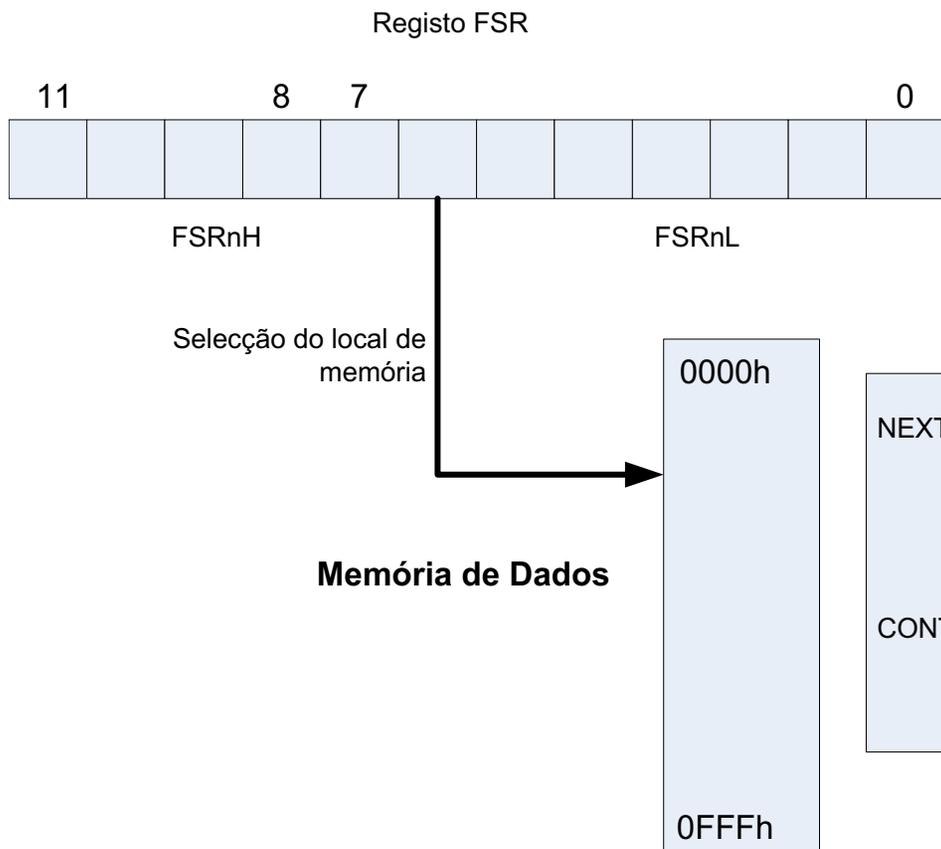
- Ao usar o auto-incrementa/decrementa, o efeito causado no FSR não afecta registo status, por exemplo se uma operação destas resultar em  $FSR = 0$ , o bit Z não será afectado;
- O incremento e decremento do FSR afecta os 12 bits. Quando  $FSRnL$  transborda após um incremento o  $FSRnH$  será incrementado automaticamente;
- Dadas estas características os registos FSR podem ser utilizados como ponteiros para pilhas e na indexação de tabelas;

A Figura seguinte mostra como se processa o endereçamento indirecto e mostra um exemplo de endereçamento indirecto.

A Figura seguinte ilustra como se processa o endereçamento indirecto:

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

## Endereçamento Indirecto



Exemplo de Endereçamento Indirecto:  
Como limpar a RAM (Banco 1) utilizando o endereçamento indirecto?

```

NEXT    LFSR    FSR0, 100h ;
        CLRF   POSTINC0 ; Limpa INDF e
                        ; incrementa SFR
        BTFSS FSR0H, 1   ; O Banco 1 está
                        ; limpo?
        BRA   NEXT      ; Não, salta para NEXT

CONTINUE                                ; Sim, continua
.
.
.
    
```

● Introdução aos MCs PIC18FXX8

Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset

● Organização da Memória

## Pilha:

- O PIC18F458 tem uma pilha (stack) de 21 bits e 31 níveis de profundidade, o que corresponde a 31 locais de memória com 21 bits de largura.
- O ponteiro para a pilha tem 5 bits e fica com o valor 00000b após um reset;
- O seu papel básico é guardar o valor do contador de programa quando ocorre um salto do programa principal para o endereço de uma subrotina a ser executado.
- Depois de ter executado a subrotina, para que o microcontrolador possa continuar com o programa principal a partir do ponto em que o deixou, ele tem que ir buscar à pilha esse endereço e carregá-lo no contador de programa.

- Introdução aos MCs PIC18FXX8

## Introdução aos MCs PIC18FXX8

- Introdução
- Relógio/Ciclo Instrução
- Pipelining
- Pinagem
- Gerador de Relógio
- Reset
- Organização da Memória

## Pilha - Continuação:

- Quando nos movemos de um programa para uma subrotina, o conteúdo do contador de programa é colocado no interior da pilha - operação PUSH (um exemplo disto é a instrução CALL).
- Quando são executadas instruções tais como RETURN, RETLW ou RETFIE no fim de uma subrotina, o contador de programa é retirado da pilha - operação POP, de modo a que o programa possa continuar a partir do ponto em que a sequência foi interrompida.

A Figura seguinte mostra o registo Stack Pointer - STKPTR:

