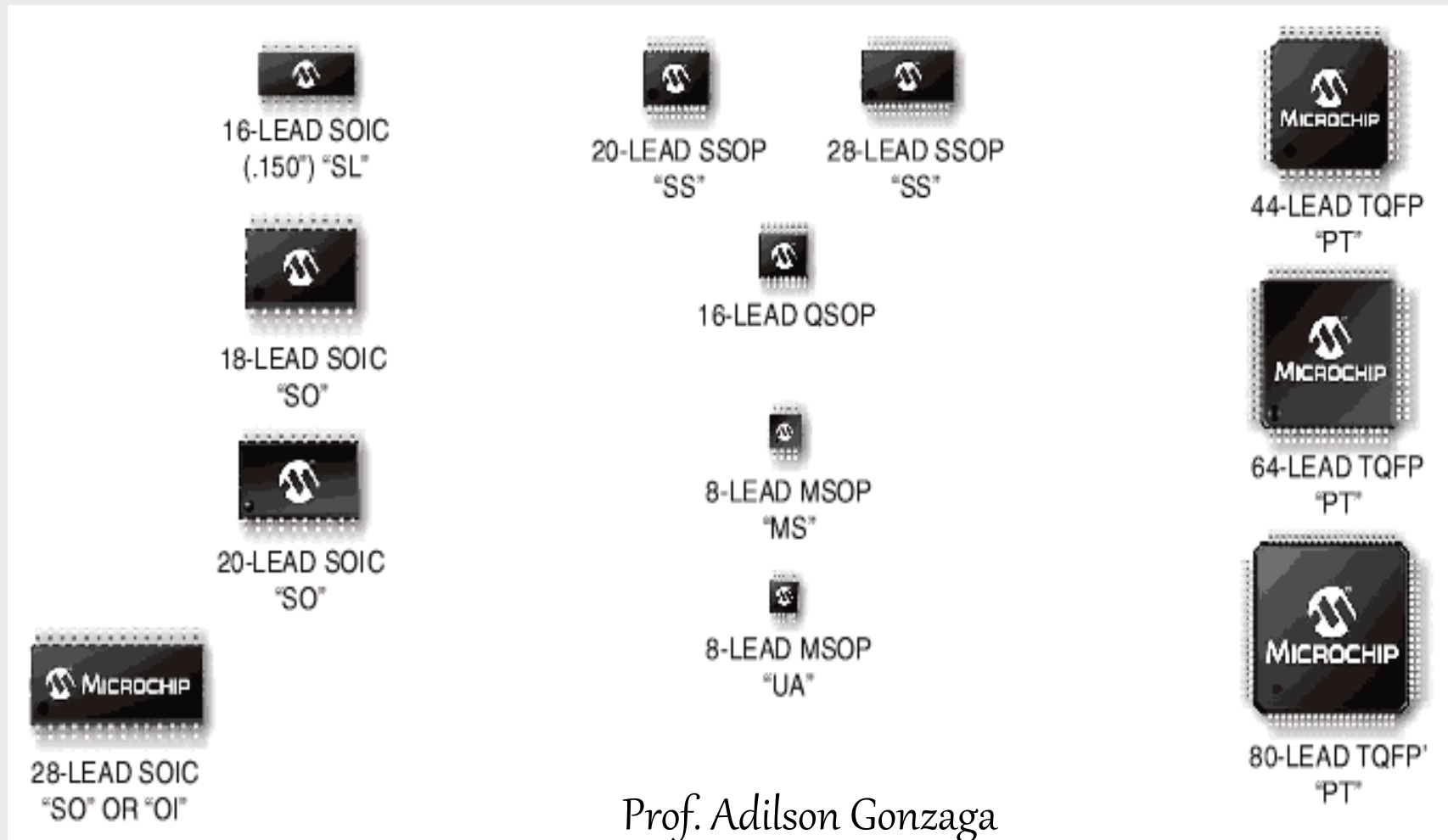


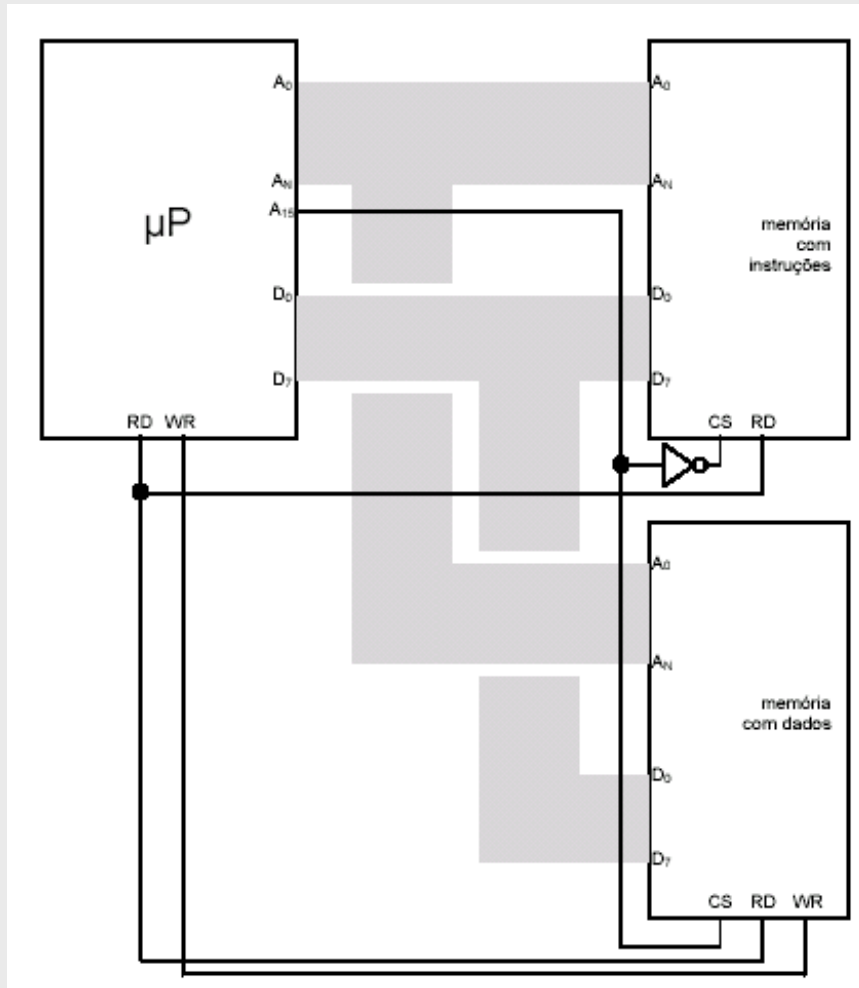
# Microcontroladores PIC



Prof. Adilson Gonzaga

# Arquitetura de Microcomputadores

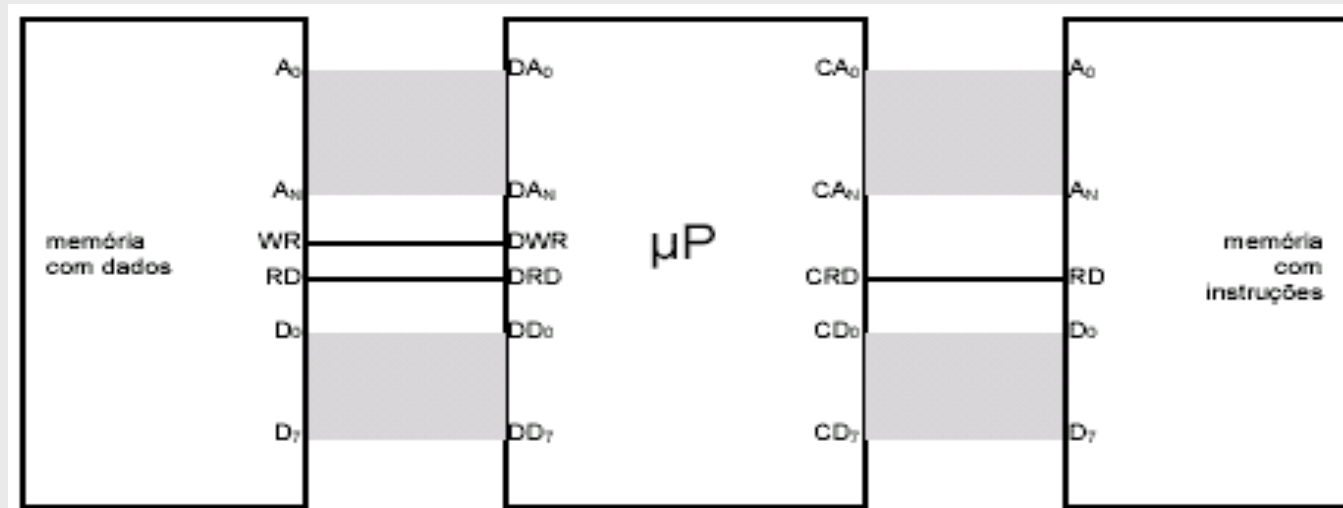
## Arquitetura von Neumann:



- Há apenas um barramento (duto) de dados e um de endereço.
- As Instruções e os dados ocupam o mesmo endereçamento de memória.
- As Instruções e os dados trafegam pelo mesmo duto (Duto de Dados).

# Arquitetura de Microcomputadores

## Arquitetura Harvard:



- **Existem dois barramentos de dados independentes e dois barramentos de endereços independentes**
- **Um barramento serve para leitura de instruções, outro para leitura e escrita de dados.**
- **Instruções e dados estão localizados em endereçamentos independentes**

# Arquitetura de Microcomputadores

## Exemplos:

### Arquitetura von Neumann:

- **Microprocessadores mais tradicionais**  
Intel --> 8080,8085,8088,.....  
Motorola --> MC6800, MC68000

### Arquitetura Harvard:

- **Microcontroladores Microchip --> PIC**
  
- **DSP (SHARC) --> Texas, Analog Devices, Motorola**

# Arquitetura de Microcomputadores

- **Os Microcontroladores da família MCS-51, possuem uma arquitetura mista.**
- **Apesar da Memória de dados ser separada da memória de Instruções, o duto que transporta os dados e as instruções é o mesmo.**

- **A Arquitetura Harvard é em geral mais veloz mas exige mais de um barramento.**
- **A Arquitetura von Neumann, não pode realizar acesso simultâneo às memórias.**

# Arquitetura de Microcomputadores

## Arquitetura CISC x Arquitetura RISC

**CISC** - Complex Instruction Set Computer

**4004 - 46 instruções**  
**8080 - 78 instruções**  
**8085 - 150 instruções**  
**Z80 - mais de 500 instruções**  
**8086/8088 - mais de 700 instruções**  
**80386 - mais de 1500 instruções**

**RISC** - Reduced Instruction Set Computer

**PIC (RISC)--  
35 instruções**

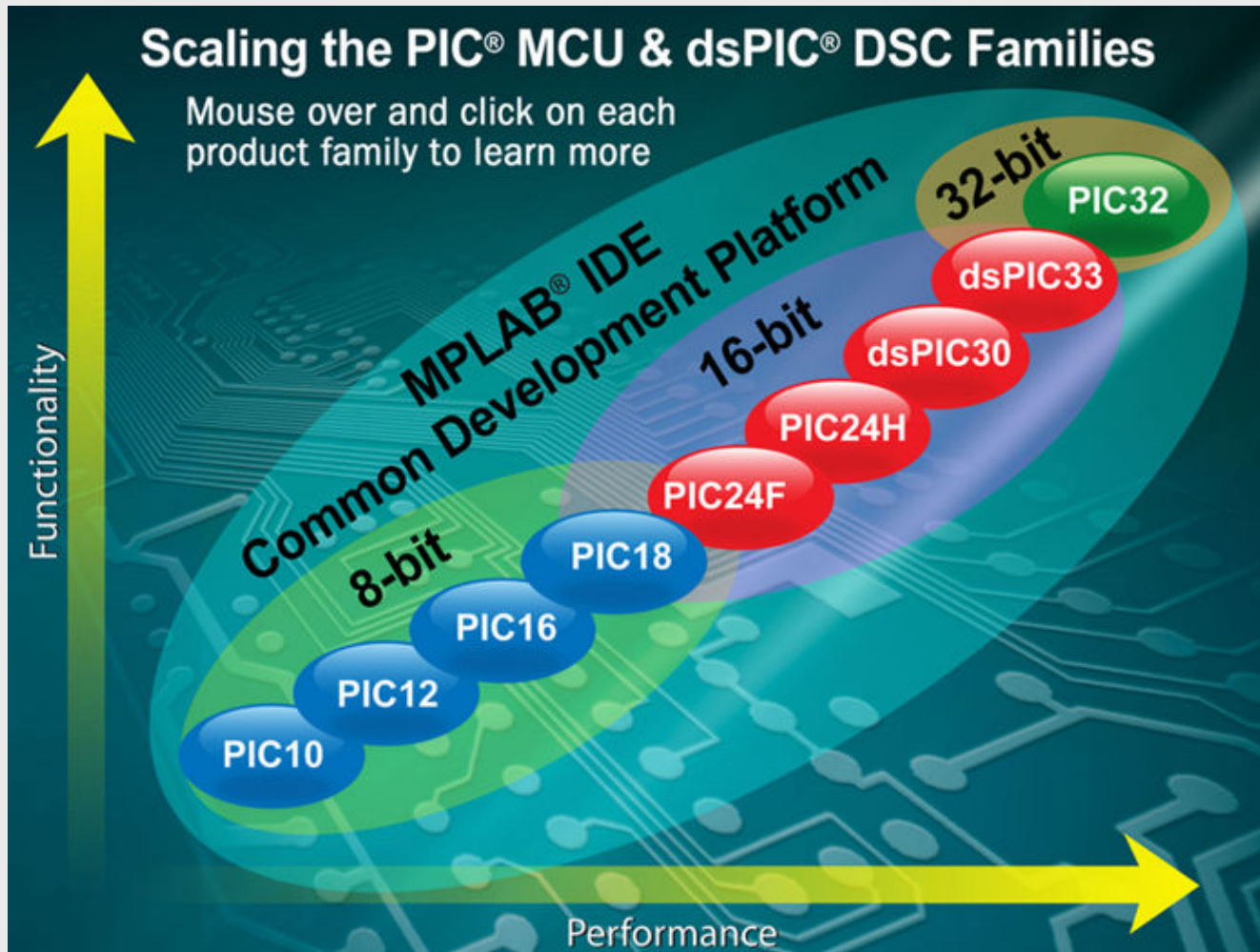
# Arquitetura de Microcomputadores

- **Aumento no número de instruções = aumento na complexidade dos circuitos do Microprocessador**

- **Arquitetura RISC --> menos de 250 instruções e alta velocidade**

	<b>Ciclo de Projeto</b>	<b>Volume de testes para certificação</b>	<b>Enfoque</b>	<b>Tamanho dos Programas</b>
<b>CISC</b>	Longo	Grande	Otimização de circuitos	Menor
<b>RISC</b>	Menor	Menor	Elevação do clock Barramentos maiores	Maior

# Microcontroladores PIC

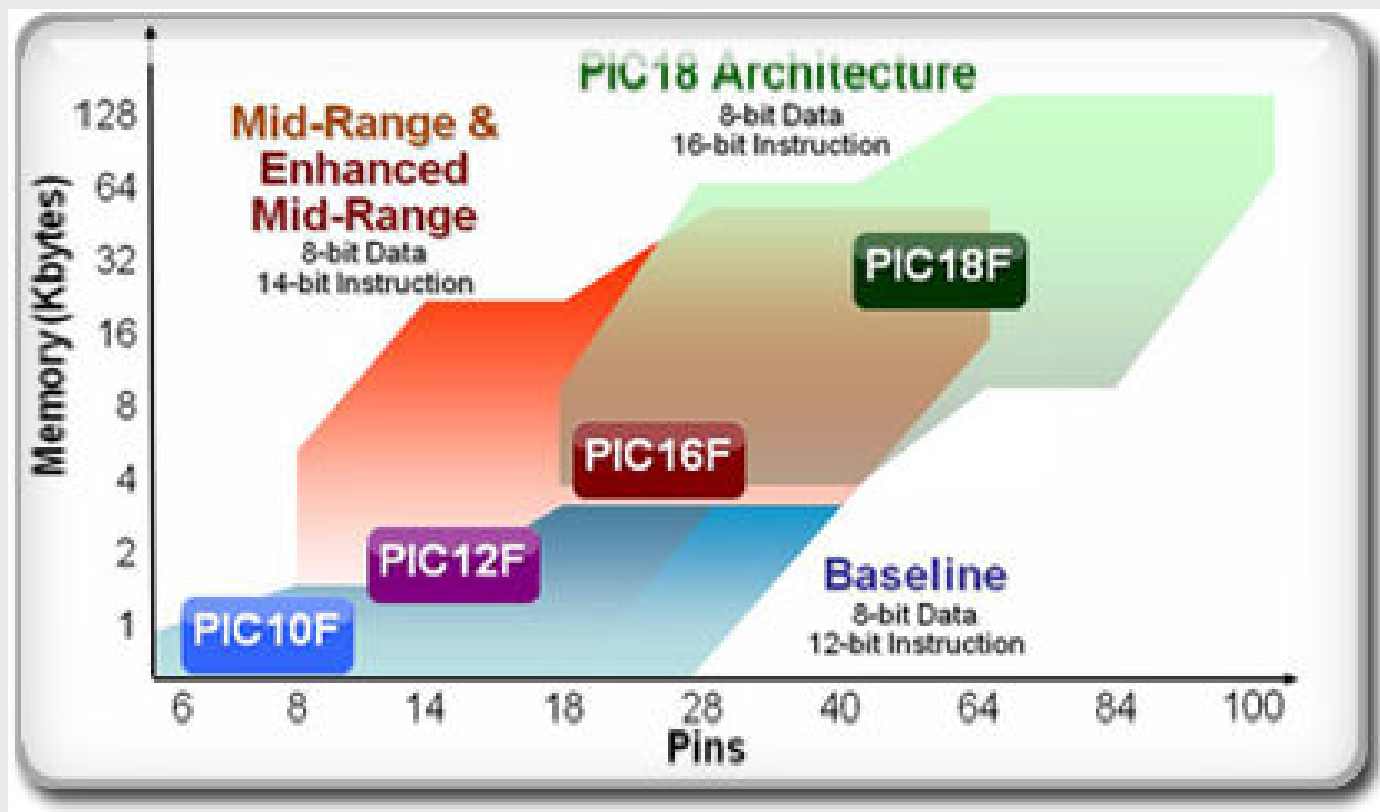




# Microcontroladores de 8 Bits

- Originalmente projetados para servirem como **P**eripheral **I**nterface **C**ontroller para um microprocessador de 16-Bits.
- Os Microcontroladores PIC de 8 Bits são agrupados de acordo com o comprimento da palavra de instrução em:
  1. **Base-Line:** 12-bit
  2. **Mid-Range & Enhanced Mid-Range:** 14-bit
  3. **PIC18 Architecture:** 16-bit

# Memória dos PIC Mid-Range



# Microcontroladores PIC de 8 Bits

	Baseline	Mid-Range	Enhanced Mid-Range	PIC18 Architecture
Pin count	6 - 32	8 - 64	8 - 64	18 - 100
Instructions	33 (12-bit)	35 (14-bit)	49 (14-bit)	75 - 83 (16-bit)
Interrupt Priority	None	Single (Multi-Source)	Single (Multi-source) (Hardware Context Save)	Multiple (Multi-source) (Hardware Context Save)
Performance	5 MIPS	5 MIPS	8 MIPS	10-16 MIPS
Program Memory	Up to 3KB	Up to 14KB	Up to 28KB	Up to 128KB
Data Memory	Up to 138bytes	Up to 368bytes	Up to 1KB	Up to 4KB
	Small, cost-effective solutions Disposable electronics Add intelligence to existing mechanical functions	Excellent for cost effective mixed signal interface Low-to-high levels of peripheral integration applications	Improved Performance & Code Density Faster Interrupt response Increased Memory & Peripheral integration	High performance involved applications Advanced communication Hardware single cycle Multiplier

# PIC Mid-Range

## Oscilador

Existem 8 maneiras diferentes de se implementar o Clock dos microcontroladores PIC.

- LP Low Frequency (Power) Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC External Resistor/Capacitor (same as EXTRC with CLKOUT)
- EXTRC External Resistor/Capacitor
- EXTRC External Resistor/Capacitor with CLKOUT
- INTRC Internal 4 MHz Resistor/Capacitor
- INTRC Internal 4 MHz Resistor/Capacitor with CLKOUT

# PIC Mid-Range

## Oscilador

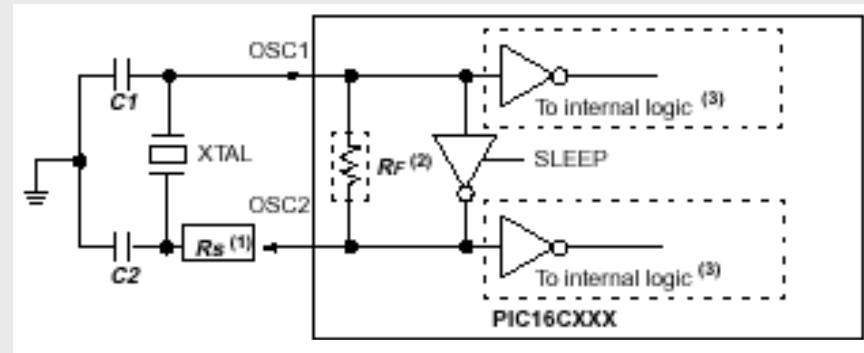
Configuration bits FOSC2:FOSC0	OSC Mode	OSC Feedback Inverter Gain	Comment
1 1 1	EXTRC with CLKOUT	—	Inexpensive solution for device oscillation. Most variation in timebase. CLKOUT is enabled on pin. Device's default mode.
1 1 0	EXTRC	—	Inexpensive solution for device oscillation. Most variation in timebase. CLKOUT is disabled (use as I/O) on pin.
1 0 1	INTRC with CLKOUT	—	Least expensive solution for device oscillation. 4 MHz oscillator, which can be tuned. CLKOUT is enabled on pin.
1 0 0	INTRC	—	Least expensive solution for device oscillation. 4 MHz oscillator, which can be tuned. CLKOUT is disabled (use as I/O) on pin.
0 1 1	—	—	Reserved
0 1 0	HS	High Gain	High frequency application. Oscillator circuit's mode consumes the most current of the three crystal modes.
0 0 1	XT	Medium Gain	Standard crystal/resonator frequency. Oscillator circuit's mode consumes the middle current of the three crystal modes.
0 0 0	LP	Low Gain	Low power/frequency applications. Oscillator circuit's mode consumes the least current of the three crystal modes.

O usuário deve programar os três Bits de configuração (FOSC2, FOSC1, FOSC0) para selecionar um entre os 8 modos.

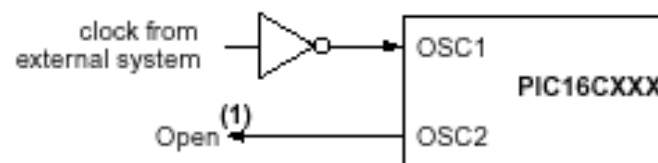
# PIC Mid-Range

## Oscilador - Cristal

Modos LP,  
XT, HS



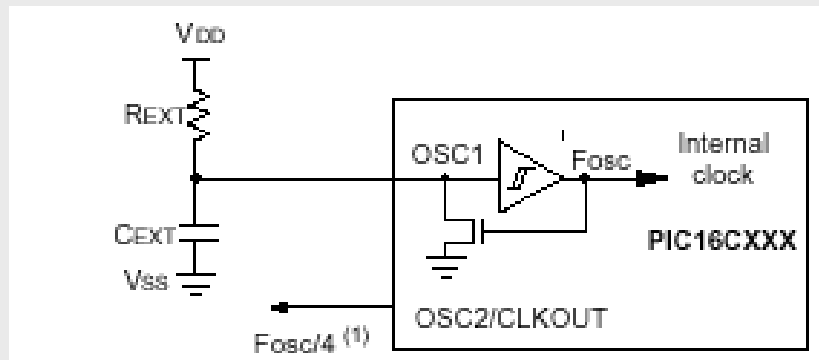
- A frequência de operação é dividida por 4.
- Utilização de um oscilador externo.



Note 1: A resistor to ground may be used to reduce system noise.  
This may increase system current.

# PIC Mid-Range

## Oscilador - RC

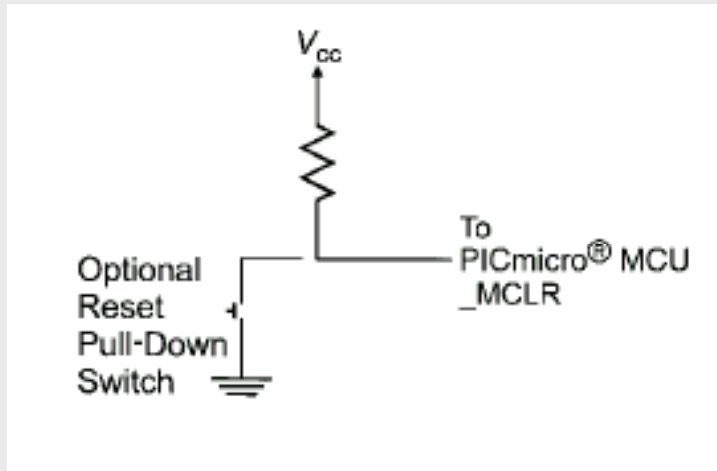


$$3K < R_{EXT} < 100 K$$

$$C_{EXT} \approx 20 \text{ pF}$$

- Escala de freqüências:  
RC -- de 0 a 4 MHz  
LP -- de 5 KHz a 200 KHz  
XT -- de 100 KHz a 4 MHz  
HS -- de 4 MHz a 20 MHz

# Reset



- Resistor de pull-up (1 a 10K) conectado ao pino  $\_MCLR$  do PIC.
- O pino de entrada é CMOS e não drena corrente significativa do resistor.

**PWRTE** - Este bit de configuração (mid-range) insere 72ms de atraso antes que a primeira instrução seja buscada e executada.



# Reset

Condition	Program Counter	STATUS Register
Power-on Reset	000h	0001 1xxx
MCLR Reset during normal operation	000h	000u uuuu
MCLR Reset during SLEEP	000h	0001 0uuu
WDT Reset (during normal operation)	000h	0000 1uuu
WDT Wake-up	PC + 1	uuu0 0uuu
Interrupt wake-up from SLEEP	PC + 1 <sup>(1)</sup>	uuu1 0uuu

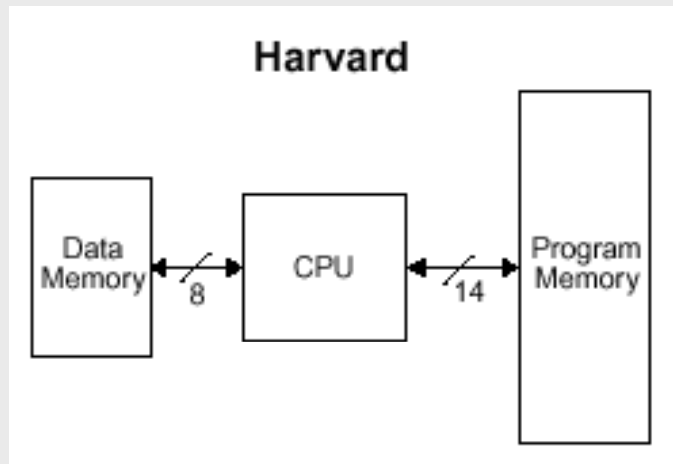
Legend: u = unchanged, x = unknown.

Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

- Power-On Reset (POR) - gerado no chip quando a tensão de alimentação está entre 1,2 e 1,7 volts.

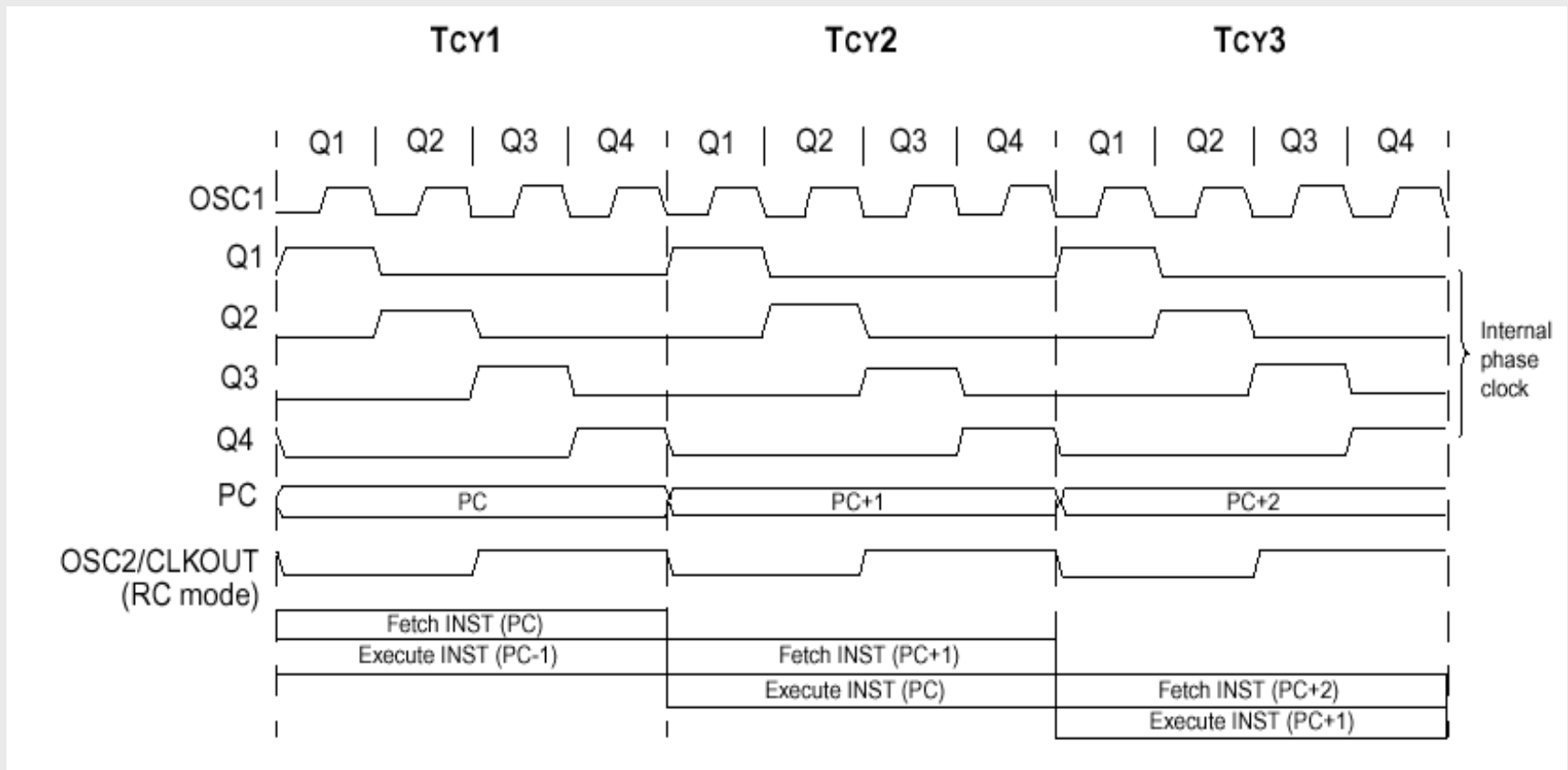
# Arquitetura dos PIC Mid-range

## Características:



- Arquitetura Harvard
- Instruções de Palavra Longa (14 bits)
- Instruções de Palavra Simples (cada instrução só ocupa os 14 bits)
- Instruções de Ciclo Simples (um ciclo de máquina)
- Pipelining de Instruções
- Conjunto de Instruções Reduzido
- Arquitetura de Arquivo de Registradores (SFR mapeado na memória de dados)
- Instruções Ortogonais (as instruções usam qualquer modo de endereçamento)

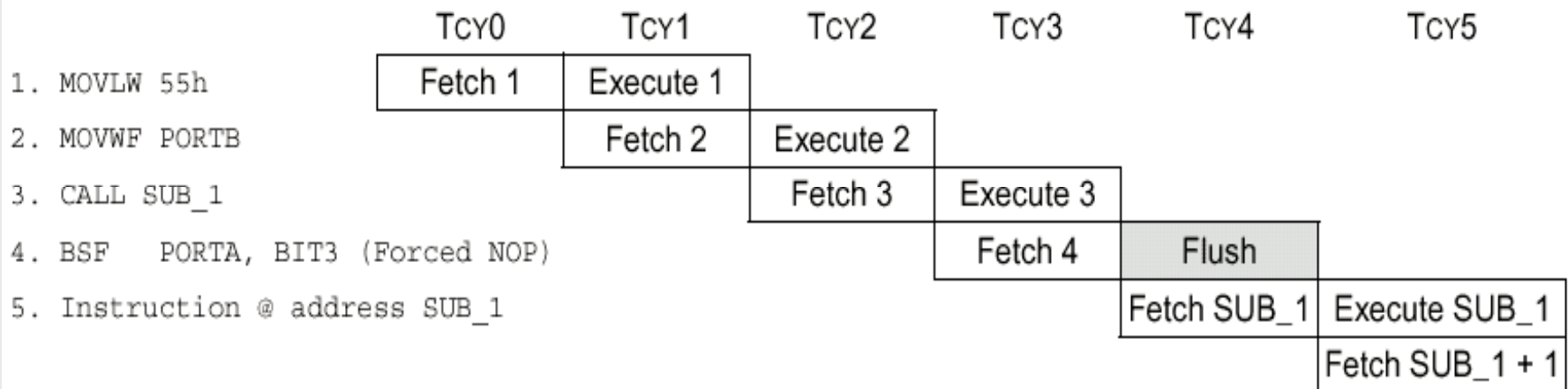
# Ciclo de Instrução



# Pipelining das Instruções

Todas as instruções são de Ciclo Simples.

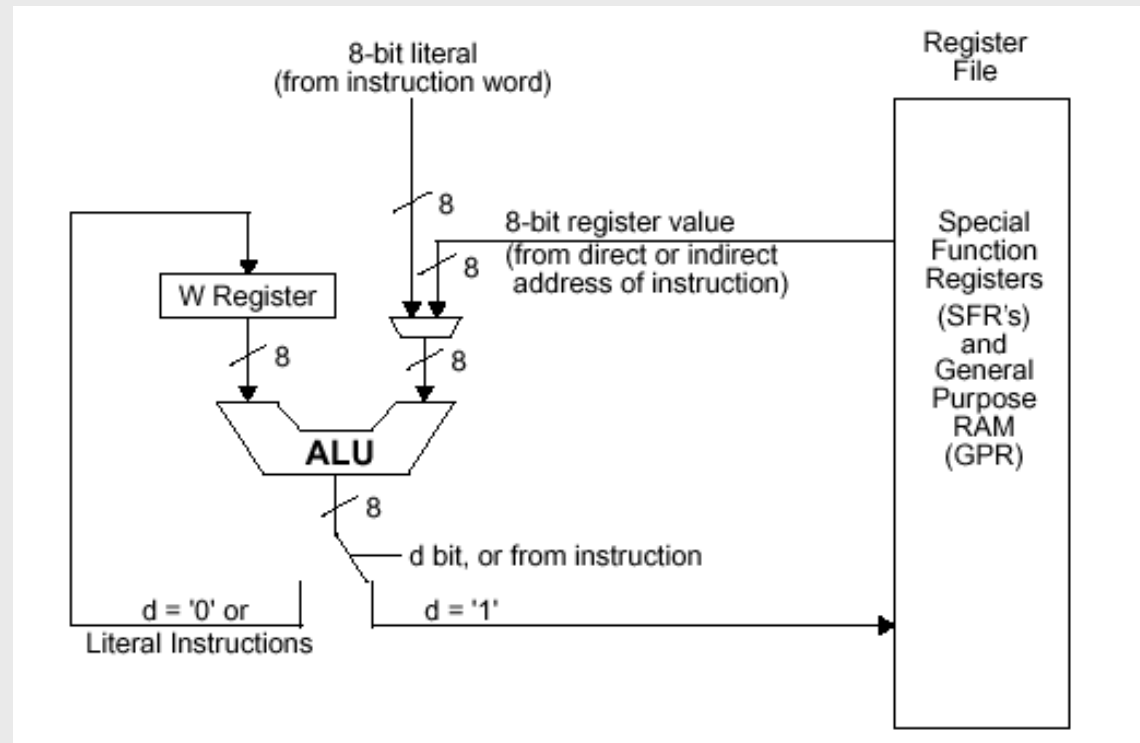
Exemplo:



A instrução de salto (CALL SUB\_1) faz com que a próxima instrução (BSF PORTA,BIT3) não seja executada.

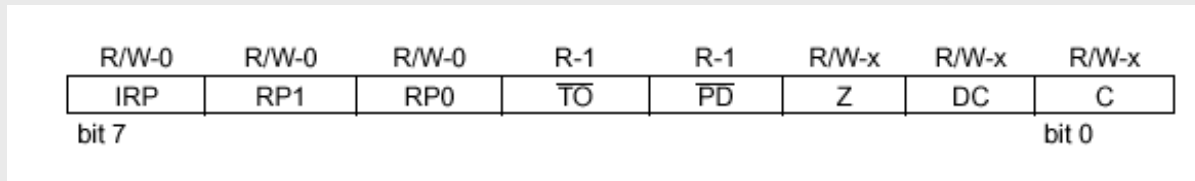
# Unidade Lógico Aritmética

Realiza operações em 8 Bits.



- Adição
- Subtração (complemento de dois)
- Shift
- Operações Lógicas

# Registrador de Status



C - Carry/Borrow

DC - Digit Carry/Borrow

Z - Zero

PD - Power Down

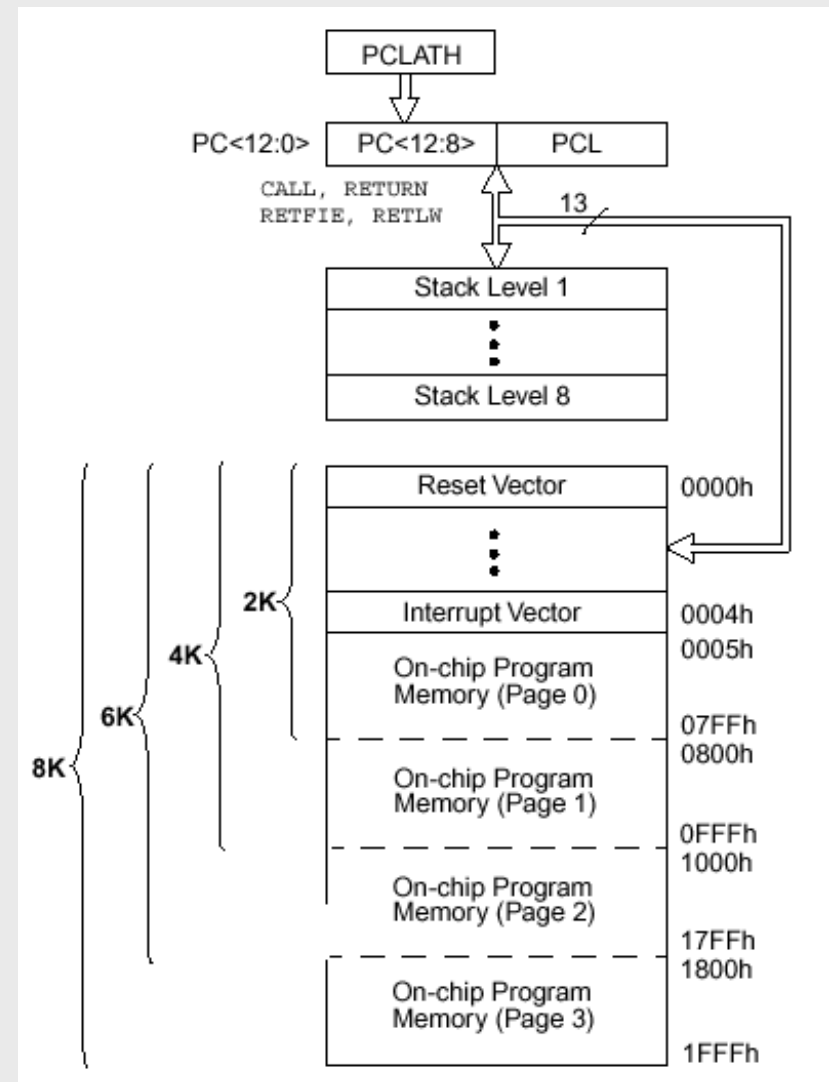
TO - Time out

RP1:RP0 - Seletor de Banco de Registradores  
(endereçamento Direto)

IRP - Seletor de Banco de Registradores (endereçamento Indireto)

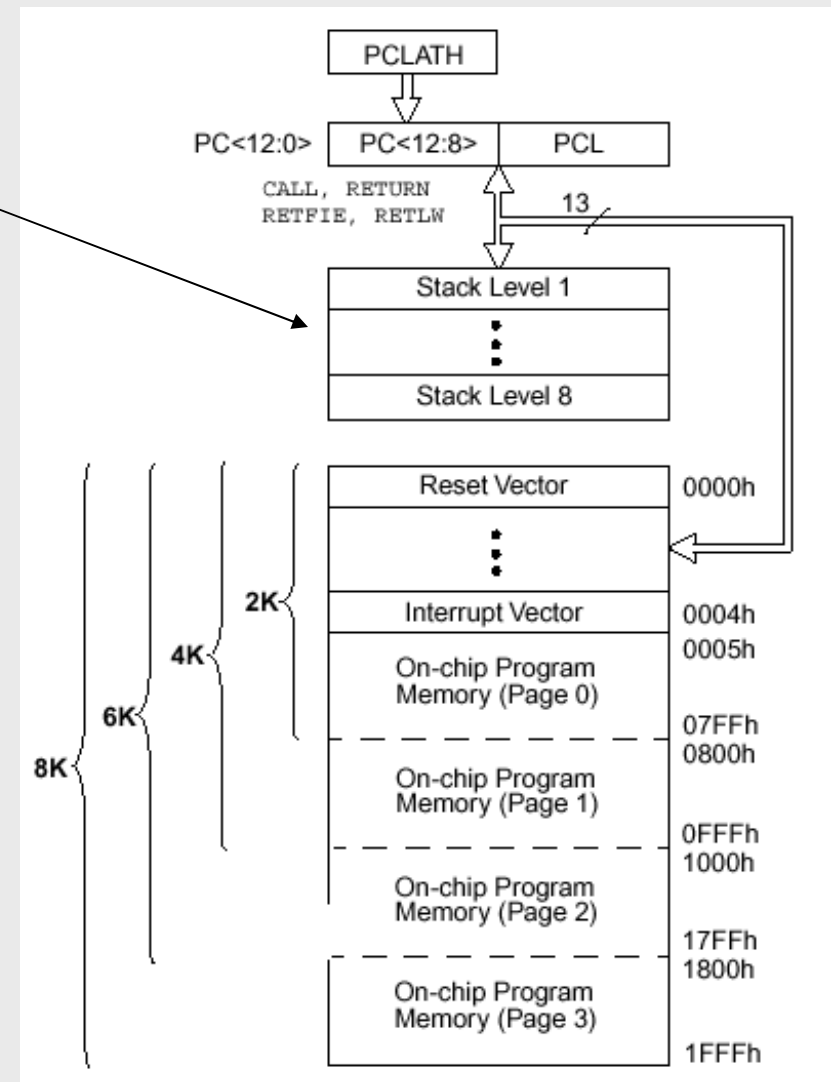
# Memória de Programa

- PC de 13 Bits endereça até 8k x 14 Bits.  
A parte mais significativa, não é acessada diretamente, mas somente através de PCLATH.
- O espaço de programa é dividido em 4 páginas de 2K
  - Instrução de palavra Simples ==> cada endereço contém uma instrução.



# Pilha

- A Pilha suporta até 8 chamadas de sub-rotinas ou interrupções.
- Após uma instrução CALL ou uma Interrupção, o PC é armazenado na Pilha.
- As instruções RETURN, RETFIN e RETLW, recuperam a Pilha.
- A Pilha não ocupa espaço de memória.



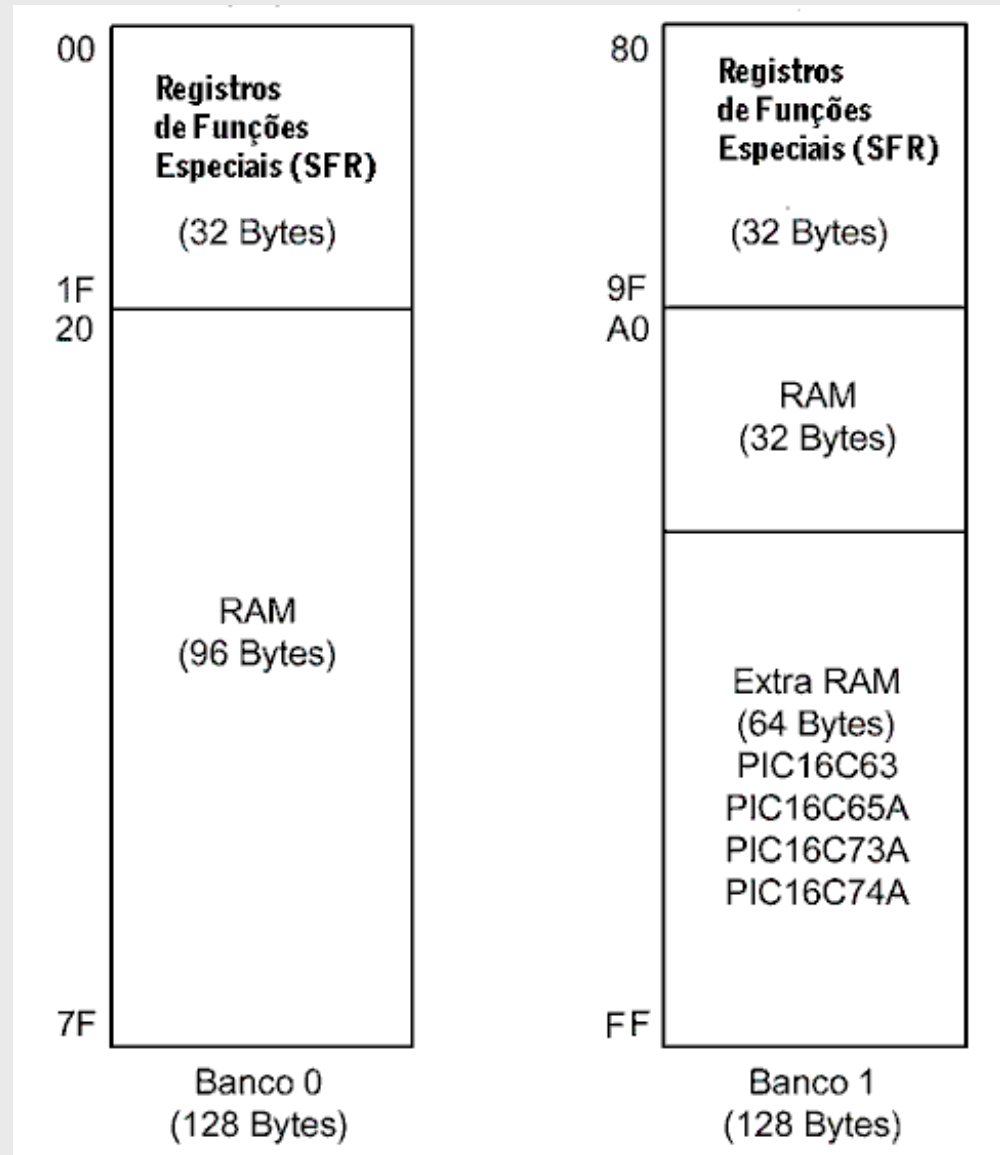


# Memória de Dados (RAM)

- A Memória de Dados é formada por:
  - SFR - Special Function Registers (Controle de operação do dispositivo)
  - GPR - General Purpose Registers (Armazenamento de dados)
- A Memória de Dados é dividida em bancos. Os bits de controle de banco são os bits 5,6 e 7 da palavra de STATUS.
- Para mover valores de um Registrador para outro, passa-se através do Registrador W, logo, são necessários dois ciclos de instruções.
- Toda a Memória de Dados pode ser acessada tanto Direta como Indiretamente.

# Memória de Dados (RAM)

- Os SFR 's realizam o controle de I/O, porta serial, timers, conversor A/D, etc...
- Todos os Registradores e posições de memória RAM são de 8 Bits.



# RAM - Mapa do Arquivo de Registradores

File Address		File Address		File Address		File Address	
<b>INDF</b>	00h	<b>INDF</b>	80h	<b>INDF</b>	100h	<b>INDF</b>	180h
<b>TMR0</b>	01h	<b>OPTION_REG</b>	81h	<b>TMR0</b>	101h	<b>OPTION_REG</b>	181h
<b>PCL</b>	02h	<b>PCL</b>	82h	<b>PCL</b>	102h	<b>PCL</b>	182h
<b>STATUS</b>	03h	<b>STATUS</b>	83h	<b>STATUS</b>	103h	<b>STATUS</b>	183h
<b>FSR</b>	04h	<b>FSR</b>	84h	<b>FSR</b>	104h	<b>FSR</b>	184h
<b>PORTA</b>	05h	<b>TRISA</b>	85h		105h		185h
<b>PORTB</b>	06h	<b>TRISB</b>	86h	<b>PORTB</b>	106h	<b>TRISB</b>	186h
<b>PORTC</b>	07h	<b>TRISC</b>	87h	<b>PORTF</b>	107h	<b>TRISF</b>	187h
<b>PORTD</b>	08h	<b>TRISD</b>	88h	<b>PORTG</b>	108h	<b>TRISG</b>	188h
<b>PORTE</b>	09h	<b>TRISE</b>	89h		109h		189h
<b>PCLATH</b>	0Ah	<b>PCLATH</b>	8Ah	<b>PCLATH</b>	10Ah	<b>PCLATH</b>	18Ah
<b>INTCON</b>	0Bh	<b>INTCON</b>	8Bh	<b>INTCON</b>	10Bh	<b>INTCON</b>	18Bh
<b>PIR1</b>	0Ch	<b>PIE1</b>	8Ch		10Ch		18Ch
<b>PIR2</b>	0Dh	<b>PIE2</b>	8Dh		10Dh		18Dh
<b>TMR1L</b>	0Eh	<b>PCON</b>	8Eh		10Eh		18Eh
<b>TMR1H</b>	0Fh	<b>OSCCAL</b>	8Fh		10Fh		18Fh
<b>T1CON</b>	10h		90h		110h		190h
<b>TMR2</b>	11h		91h		111h		191h
<b>T2CON</b>	12h	<b>PR2</b>	92h		112h		192h
<b>SSPBUF</b>	13h	<b>SSPADD</b>	93h		113h		193h
<b>SSPCON</b>	14h	<b>SSPATAT</b>	94h		114h		194h
<b>CCPR1L</b>	15h		95h		115h		195h
<b>CCPR1H</b>	16h		96h		116h		196h
<b>CCP1CON</b>	17h		97h		117h		197h
<b>RCSTA</b>	18h	<b>TXSTA</b>	98h		118h		198h
<b>TXREG</b>	19h	<b>SPBRG</b>	99h		119h		199h
<b>RCREG</b>	1Ah		9Ah		11Ah		19Ah
<b>CCPR2L</b>	1Bh		9Bh		11Bh		19Bh
<b>CCPR2H</b>	1Ch		9Ch		11Ch		19Ch
<b>CCP2CON</b>	1Dh		9Dh		11Dh		19Dh
<b>ADRES</b>	1Eh		9Eh		11Eh		19Eh
<b>ADCON0</b>	1Fh	<b>ADCON1</b>	9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose Registers <sup>(2)</sup>		General Purpose Registers <sup>(3)</sup>	EFh	General Purpose Registers <sup>(3)</sup>	16Fh	General Purpose Registers <sup>(3)</sup>	1EFh
		Mapped in Bank0	F0h	Mapped in Bank0	170h	Mapped in Bank0	1F0h
		70h - 7Fh <sup>(4)</sup>	FFh	70h - 7Fh <sup>(4)</sup>	17Fh	70h - 7Fh <sup>(4)</sup>	1FFh
Bank0		Bank1		Bank2 <sup>(5)</sup>		Bank3 <sup>(5)</sup>	

# RAM - Mapa do Arquivo de Registradores

	File Address		File Address
<b>INDF</b>	00h	<b>INDF</b>	80h
<b>TMR0</b>	01h	<b>OPTION_REG</b>	81h
<b>PCL</b>	02h	<b>PCL</b>	82h
<b>STATUS</b>	03h	<b>STATUS</b>	83h
<b>FSR</b>	04h	<b>FSR</b>	84h
<b>PORTA</b>	05h	<b>TRISA</b>	85h
<b>PORTB</b>	06h	<b>TRISB</b>	86h
	07h	<b>PCON</b>	87h
<b>ADCON0 / EEDATA (2)</b>	08h	<b>ADCON1 / EECON1 (2)</b>	88h
<b>ADRES / EEADR (2)</b>	09h	<b>ADRES / EECON2 (2)</b>	89h
<b>PCLATH</b>	0Ah	<b>PCLATH</b>	8Ah
<b>INTCON</b>	0Bh	<b>INTCON</b>	8Bh
	0Ch		8Ch
General Purpose Registers (3)		General Purpose Registers (4)	
	7Fh		FFh
Bank0		Bank1	

- O que está em **Negrito**, existe em todos os dispositivos da família PIC.

(2) Estes registradores podem não estar implementados da mesma forma nos diversos dispositivos.

(3) Nem todas as posições da RAM estão implementadas. Quando não existem, são lidas como Zero.

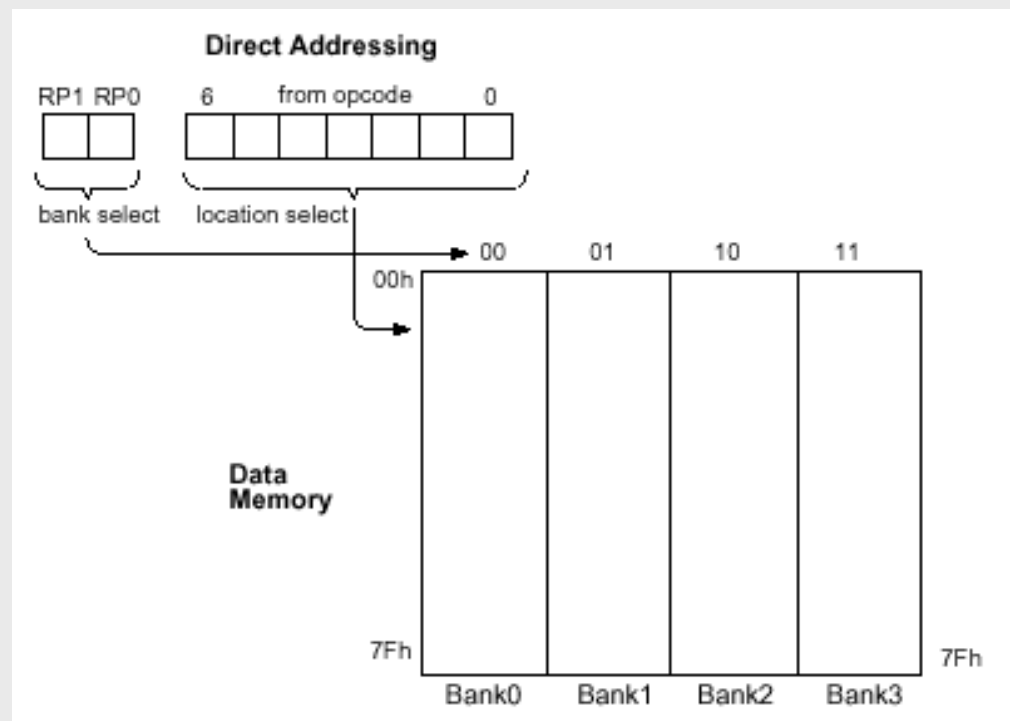
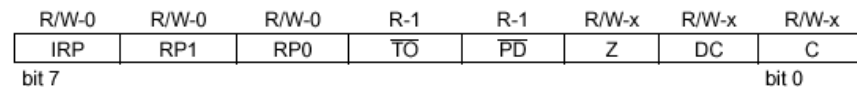
(4) Acessos a posições não implementadas no Banco 1, acessa as mesmas posições no Banco 0.

# Endereçamento Direto (RAM)

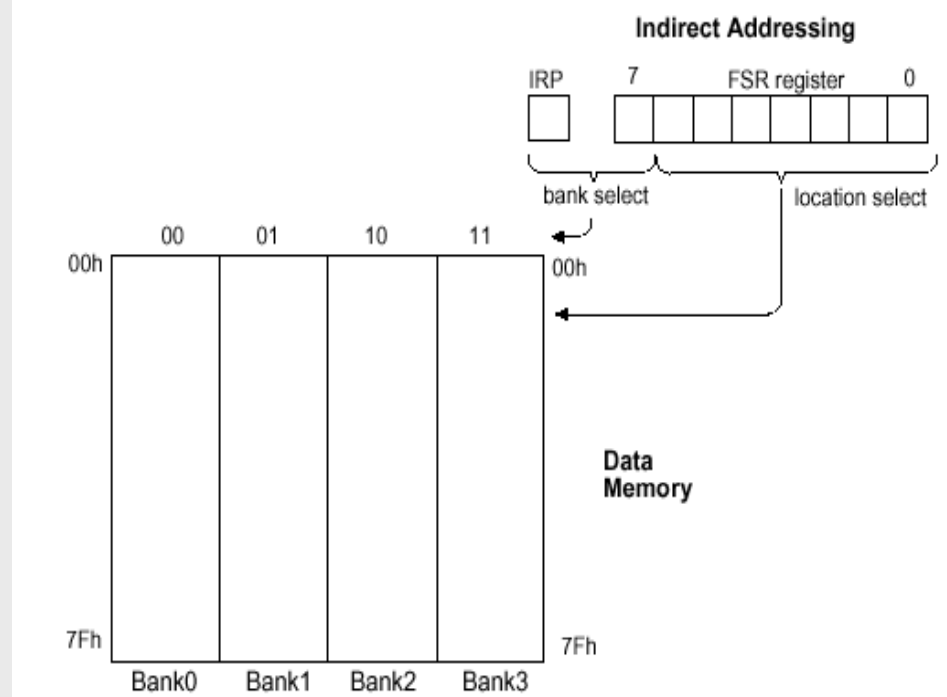
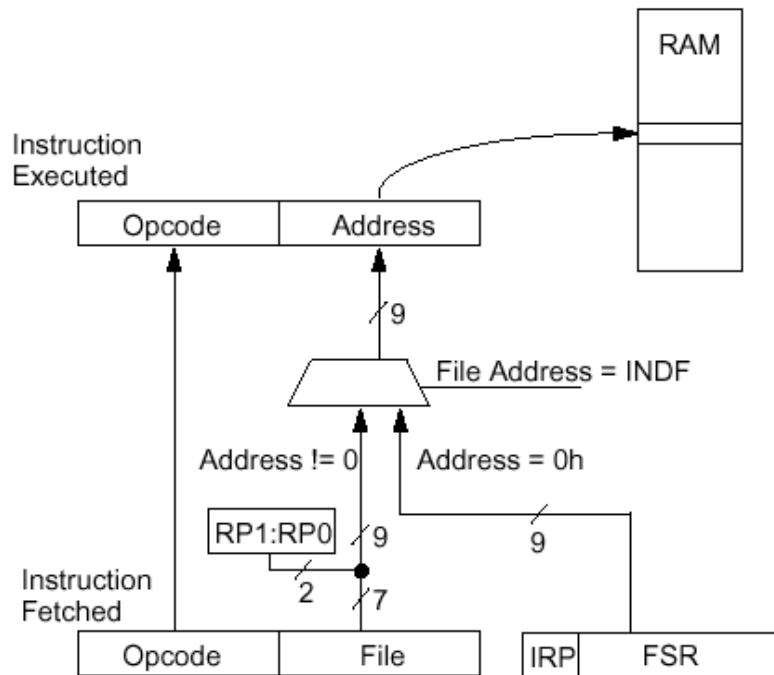
## Registrador de STATUS

Accessed Bank	Direct (RP1:RP0)	Indirect (IRP)
0	0 0	0
1	0 1	
2	1 0	1
3	1 1	

- O endereço do dado é obtido através dos bits de Seleção de página no Registrador de Status.



# Endereçamento Indireto (RAM)



- A operação realizada sobre o INDF será na verdade realizada sobre o registrador apontado pelo FSR.

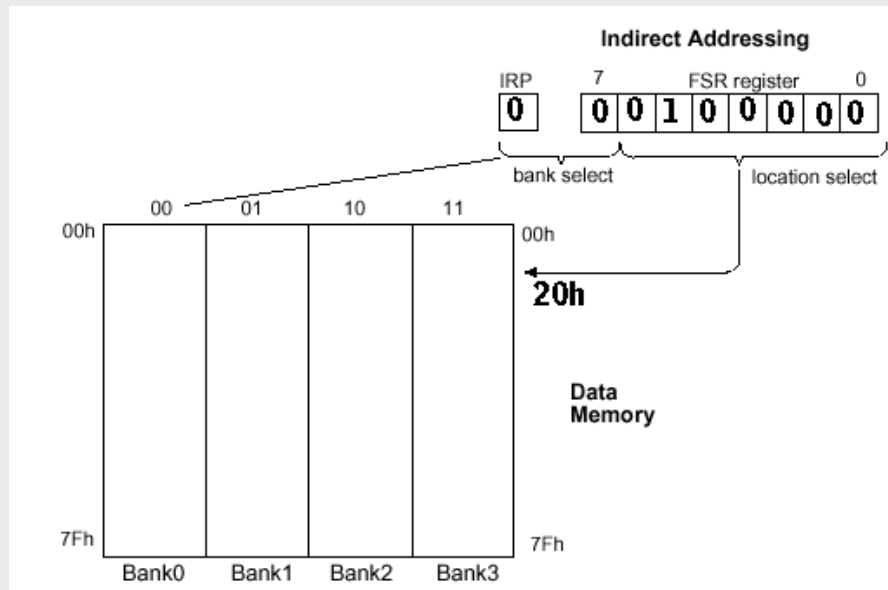
# Endereçamento Indireto (RAM)

**Exemplo:** Preencher da posição 20 à 2F da RAM com zeros

```
BCF     STATUS, IRP ; Zera o Bit IRP da palavra de Status
MOVLW  0x20        ; Carrega o registrador W com 20 h
MOVWF  FSR         ; Move o conteúdo de W para FSR
NEXT   CLRf        ; Zera o conteúdo da posição apontada por FSR
        INCF       FSR, F ; Incrementa o conteúdo de FSR
        BTFSS     FSR, 4 ; Salta a próxima instrução se o Bit 4 de FSR for 1
        GOTO      NEXT ; Volta para NEXT
```

CONTINUE

:



# Memória de Dados (EEPROM)

- Pode ser lida ou escrita durante operação normal do dispositivo.
- É Indiretamente endereçável através de registradores SFR.
  - **EECON1** (registrador de controle de escrita/leitura na EEPROM)
  - **EECON2** (registrador de controle de escrita/leitura na EEPROM)
  - **EEDATA** (armazena os 8 Bits de dados que serão lidos ou gravados na EEPROM)
  - **EEADR** (armazena os 8 Bits de endereço onde serão lidos ou gravados os dados na EEPROM)

Data EEPROM Size <sup>(1)</sup>	Address Range
64	0h - 3Fh
128	0h - 7Fh
256	0h - FFh



# Interrupção

Quatro fontes de Interrupção:

1. Interrupção Externa através do pino RB0/INT
2. Interrupção por transbordo do Timer 0 - TMR0
3. Interrupção quando se troca alguma entrada em RB4:RB7
4. Interrupção quando se completa uma escrita de dados na EEPROM

# Interrupção

- Habilitação/desabilitação das interrupções é feita através do bit de controle: **INTCON.GIE**
- As interrupções são desabilitadas com o Reset
- A instrução de Retorno de Interrupção - RETFIE - habilita as interrupções
- Quando se atende à uma interrupção, desabilita-se automaticamente todas as interrupções e o programa salta para o endereço 0004 da memória de programa.

# Watch Dog Timer (WDT)

- É um temporizador com um circuito oscilador RC interno.
- Se o Micro está em **modo normal**, um transbordo no WDT causa um reset.
- Se o Micro está no **modo SLEEP**, um transbordo no WDT acorda o Micro.
- Pode-se desativá-lo permanentemente.
- O período nominal de transbordo é de 18 ms.

# Power-down (Modo SLEEP)

- É um modo de **baixo consumo** caracterizado pelo congelamento do oscilador externo.
- Para entrar nesse modo, basta executar a instrução SLEEP.
- O WDT é resetado mas não congelado.

- **Modos de se despertar o Micro:**
  1. Reset no pino \_MCLR
  2. Transbordo de WDT
  3. Interrupção (RB0/INT ; RB4:RB7 ; escrita na EEPROM)

# Bits de Configuração

- O espaço de memória reservado para testes e configuração (2000 a 3FFF) só é acessível durante a programação
- A palavra de Configuração está localizada na posição 2007

R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	$\overline{\text{PWRTE}}$	WDTE	FOSC1	FOSC0
bit13										bit0			

bit 13:4 **CP**: Code Protection bit  
1 = Code protection off  
0 = All memory is code protected

bit 3  **$\overline{\text{PWRTE}}$** : Power-up Timer Enable bit  
1 = Power-up timer is disabled  
0 = Power-up timer is enabled

bit 2 **WDTE**: Watchdog Timer Enable bit  
1 = WDT enabled  
0 = WDT disabled

bit 1:0 **FOSC1:FOSC0**: Oscillator Selection bits  
11 = RC oscillator  
10 = HS oscillator  
01 = XT oscillator  
00 = LP oscillator

R = Readable bit
P = Programmable bit
- n = Value at POR reset
u = unchanged

# Bits de Configuração

- Através do programa realiza-se a configuração:
  - Exemplo: usando o PIC 16F84.

```
INCLUDE "P16F84A.INC"  
  
__CONFIG _WDT_OFF & _CP_OFF & _XT_OSC & _PWRTE_ON
```

- Watch Dog (WDT) desabilitado
- Proteção do código (CP) desabilitada
- Oscilador de Cristal (XT) selecionado
- Temporizador de Power-up (PWRTE) desabilitado