FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

Aula 14
Interrupções

Microcontroladores PIC18 – Programação em C

MICROCHIP

Prof. Ítalo Jáder Loiola Batista
Universidade de Fortaleza - UNIFOR
Centro de Ciências Tecnológicas - CCT
E-mail: italoloiola@unifor.br

Jan / 2011

Roteiro

- Interrupções
  - Introdução;
  - Registradores;
  - Fontes de interrupções;
  - Interrupção Externa;
  - Código-fonte

Interrupções

- A interrupção é um evento de hardware que provoca uma interrupção no programa;
- Desvia o programa para uma localidade específica da memória de programa para que o evento seja tratado;
- Em seguida, o programa retorna a execução do ponto em que foi interrompido;
- O PIC18F4520 possui 20 fontes de interrupção, sendo 18 diferentes;

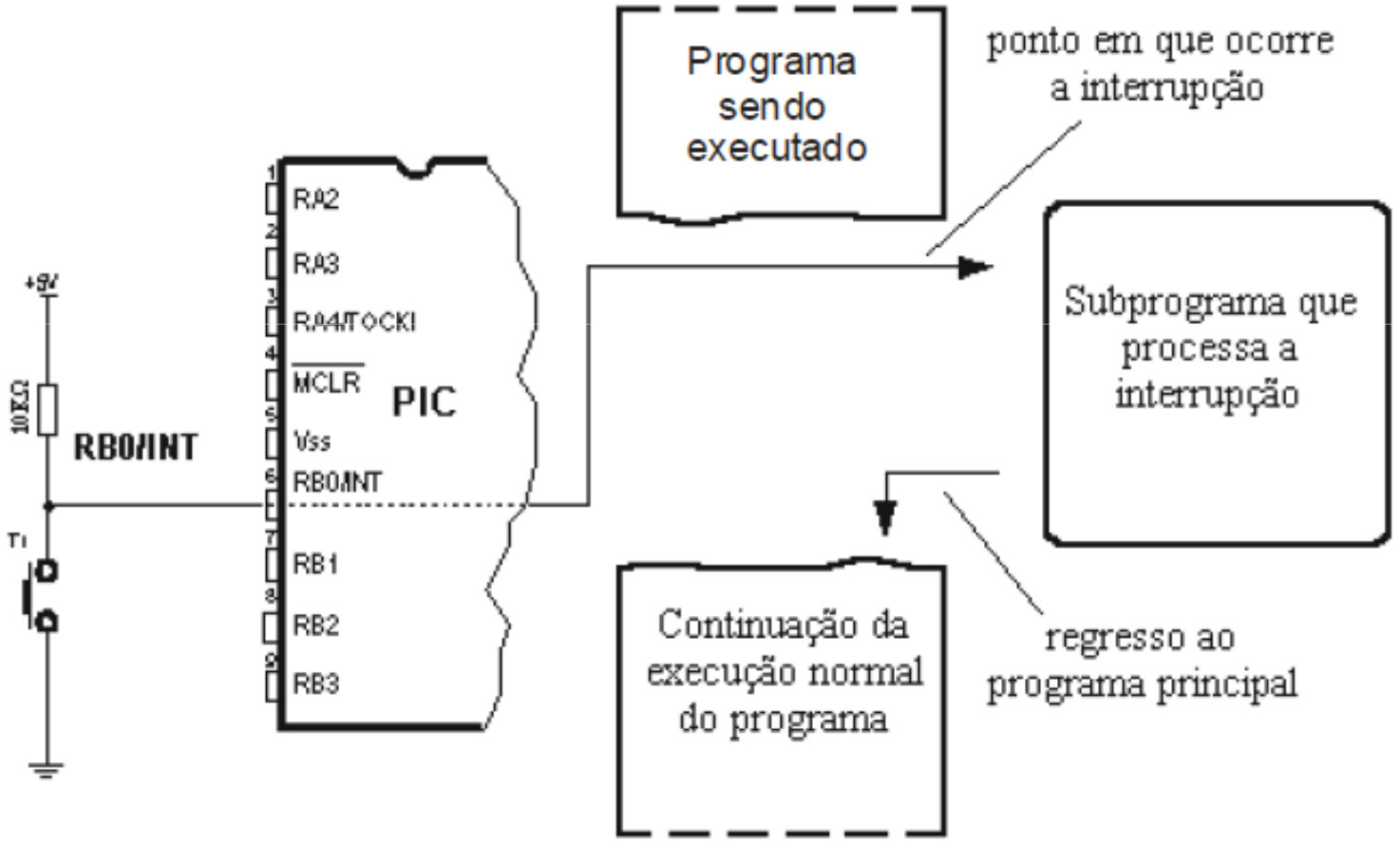
Interrupções

- O PIC18F4520 trabalha com dois níveis de prioridade no serviço de tratamento de interrupção (ISR):
  - *High-priority* (alta prioridade);
    - O programa é desviado para o endereço 0008h da memória de programa quando ocorrer o evento responsável pela interrupção.
  - *Low-priority* (baixa prioridade);
    - O programa é desviado para o endereço 0018h da memória de programa quando ocorrer o evento responsável pela interrupção.

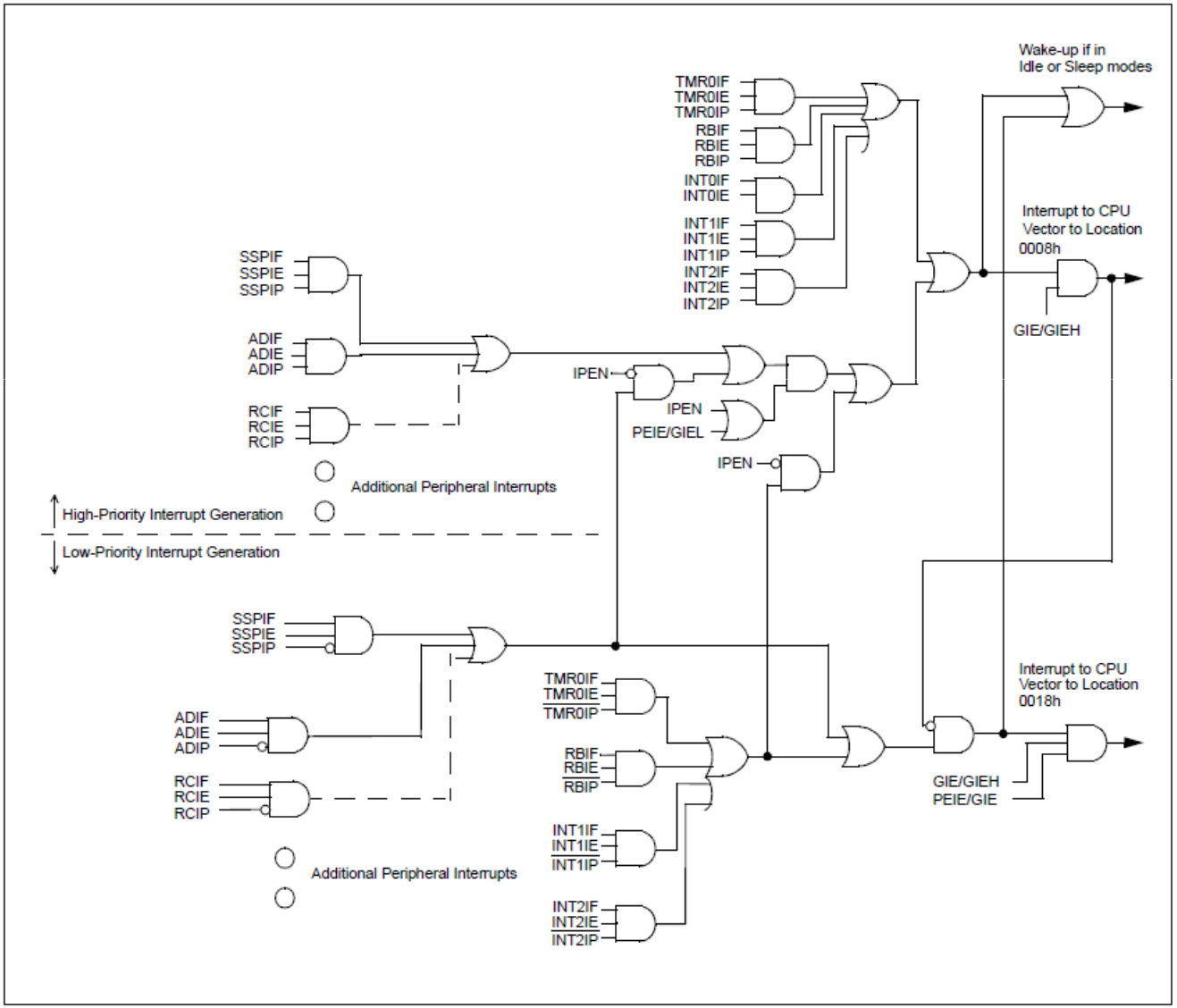
Interrupções

- De uma forma geral, três bits estão envolvidos com o recurso de interrupção.
  - ***Flag bit***
    - Bit de Sinalização;
    - Cada interrupção possui um bit de sinalização que é setado quando o evento associado ocorre;
    - O bit de sinalização deve ser apagado por software dentro da função de interrupção para que o programa não retorne à rotina de interrupção após ela ter sido tratada;
  - ***Enable bit***
    - Bit de habilitação da interrupção;
  - ***Priority bit***
    - Bit que define a prioridade no tratamento da interrupção;

Interrupções



Estrutura de habilitação das interrupções



Registradores FSR de Interrupção

- Existem dez registradores no PIC18F4520 envolvidos com o recurso da interrupção:
  - RCON
  - INTCON
  - INTCON2
  - INTCON2
  - PIR1, PIR2
  - PIE1, PIE2
  - IPR1, IPR2

Registradores FSR de Interrupção

- A seguir as informações e suas respectivas descrições, encontradas na descrição dos registradores que são utilizados pela interrupção:
  - **R**: bit de leitura
  - **W**: bit de escrita
  - **S**: só pode ser setado
  - **U**: não implementado, lido como 0
  - **-n**: nível lógico assumido no POR
  - **-x**: valor desconhecido no (POR)
  - **-q**: depende da condição

Registradores – INTCON

- Os registros do INTCON podem ser lidos e escritos e que contêm vários bits de habilitação, prioridade e sinalização;

Registradores - INTCON

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
bit 7							bit 0

Quando IPEN(RCON<7>) é habilitado:

GIEH: habilita as interrupções de alta prioridade.

GIEL: habilita as interrupções de baixa prioridade.

Quando IPEN(RCON<7>) é apagado:

GIE: liga a chave geral de interrupção.

PEIE: liga a chave que habilita a interrupção dos perifféricos.

bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high-priority interrupts
0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1:
1 = Enables all low-priority peripheral interrupts
0 = Disables all low-priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit(1)
1 = At least one of the RB<7:4> pins changed state (must be cleared in software)
0 = None of the RB<7:4> pins have changed state

Bits de habilitação

Bits de sinalização

Registradores - INTCON

- Se os dois níveis de prioridade de interrupção estiverem habilitados na ocorrência de um evento que pode dar origem a uma interrupção, um dos bits GIEH e GIEL será apagado para evitar futuras interrupções;
- No entanto se uma interrupção de baixa prioridade estiver sendo tratada e ocorrer uma interrupção de alta prioridade, o tratamento da 1ª será interrompido para que a 2ª seja tratada;
- Quando a interrupção de alta prioridade terminar de ser tratada, a interrupção de baixa prioridade volta a ser tratada;

Registradores – INTCON2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
bit 7							bit 0

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **Unimplemented:** Read as '0'

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **Unimplemented:** Read as '0'

bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

Registradores – INTCON3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
bit 7							bit 0

bit 7 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **Unimplemented:** Read as '0'

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

bit 2 **Unimplemented:** Read as '0'

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

Registradores – PIR

- Os registradores PIR possuem os *flag bits (bits de sinalização)* individuais das interrupções dos periféricos;
- Registradores PIR:
  - PIR1;
  - PIR2;

Registradores – PIR1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **PSPIF:** Parallel Slave Port Read/Write Interrupt Flag bit(1)
1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

bit 5 **RCIF:** EUSART Receive Interrupt Flag bit
1 = The EUSART receive buffer, RCREG, is full (cleared when RCREG is read)
0 = The EUSART receive buffer is empty

bit 4 **TXIF:** EUSART Transmit Interrupt Flag bit
1 = The EUSART transmit buffer, TXREG, is empty (cleared when TXREG is written)
0 = The EUSART transmit buffer is full

bit 3 **SSPIF:** Master Synchronous Serial Port Interrupt Flag bit
1 = The transmission/reception is complete (must be cleared in software)
0 = Waiting to transmit/receive

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode.

bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

Registradores – PIR2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	CMIF	—	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF
bit 7							bit 0

bit 7 **OSCFIF:** Oscillator Fail Interrupt Flag bit
1 = Device oscillator failed, clock input has changed to INTOSC (must be cleared in software)
0 = Device clock operating

bit 6 **CMIF:** Comparator Interrupt Flag bit
1 = Comparator input has changed (must be cleared in software)
0 = Comparator input has not changed

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** Data EEPROM/Flash Write Operation Interrupt Flag bit
1 = The write operation is complete (must be cleared in software)
0 = The write operation is not complete or has not been started

bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
1 = A bus collision occurred (must be cleared in software)
0 = No bus collision occurred

bit 2 **HLVDIF:** High/Low-Voltage Detect Interrupt Flag bit
1 = A high/low-voltage condition occurred (direction determined by VDIRMAG bit, HLVDCON<7>)
0 = A high/low-voltage condition has not occurred

bit 1 **TMR3IF:** TMR3 Overflow Interrupt Flag bit
1 = TMR3 register overflowed (must be cleared in software)
0 = TMR3 register did not overflow

bit 0 **CCP2IF:** CCP2 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode.

Registradores – PIE

- Os registradores PIR possuem os *priority bits* (*bits* de prioridade) individuais das interrupções dos periféricos;
- Registradores PIR:
  - PIE1;
  - PIE2;

Registradores – PIE1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

bit 7 **PSPIE:** Parallel Slave Port Read/Write Interrupt Enable bit(1)
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt

bit 5 **RCIE:** EUSART Receive Interrupt Enable bit
1 = Enables the EUSART receive interrupt
0 = Disables the EUSART receive interrupt

bit 4 **TXIE:** EUSART Transmit Interrupt Enable bit
1 = Enables the EUSART transmit interrupt
0 = Disables the EUSART transmit interrupt

bit 3 **SSPIE:** Master Synchronous Serial Port Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Registradores – PIE2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIE	CMIE	—	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE
bit 7							bit 0

bit 7 **OSCFIE:** Oscillator Fail Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **CMIE:** Comparator Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIE:** Data EEPROM/Flash Write Operation Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3 **BCLIE:** Bus Collision Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **HLVDIE:** High/Low-Voltage Detect Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **TMR3IE:** TMR3 Overflow Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit
1 = Enabled
0 = Disabled

Registradores – IPR

- Os registradores PIR possuem os *Enable bit (bits* de habilitação) individuais das interrupções dos periféricos;
- Registradores PIR:
  - PIE1;
  - PIE2;

Registradores – IPR1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
bit 7							bit 0

bit 7 **PSPIP:** Parallel Slave Port Read/Write Interrupt Priority bit(1)
1 = High priority
0 = Low priority

bit 6 **ADIP:** A/D Converter Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **RCIP:** EUSART Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **TXIP:** EUSART Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **SSPIP:** Master Synchronous Serial Port Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CCP1IP:** CCP1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR2IP:** TMR2 to PR2 Match Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR1IP:** TMR1 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

Registradores – IPR2

R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
OSCFIP	CMIP	—	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP
bit 7							bit 0

bit 7 **OSCFIP:** Oscillator Fail Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **CMIP:** Comparator Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIP:** Data EEPROM/Flash Write Operation Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **BCLIP:** Bus Collision Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **HLVDIP:** High/Low-Voltage Detect Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **CCP2IP:** CCP2 Interrupt Priority bit
1 = High priority
0 = Low priority

Registradores – RCON

- O Registrador RCON possuem os *flag bits (bits de sinalização)* que são usados para determinar as causas de Reset ou Wake-up (Modo Sleep);
- O RCON possui também o bit que habilita as prioridades das interrupções;

Registradores – RCON

R/W-0	R/W-1(1)	U-0	R/W-1	R-1	R-1	R/W-0(1)	R/W-0
IPEN	SBOREN	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

Define os níveis de prioridade no tratamento das interrupção

bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6 **SBOREN:** Software BOR Enable bit(1)
For details of bit operation, see Register 4-1.

bit 5 **Unimplemented:** Read as '0'

bit 4 **$\overline{RI}$:** RESET Instruction Flag bit
For details of bit operation, see Register 4-1.

bit 3 **$\overline{TO}$:** Watchdog Timer Time-out Flag bit
For details of bit operation, see Register 4-1.

bit 2 **$\overline{PD}$:** Power-Down Detection Flag bit
For details of bit operation, see Register 4-1.

bit 1 **$\overline{POR}$:** Power-on Reset Status bit(1)
For details of bit operation, see Register 4-1.

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
For details of bit operation, see Register 4-1.

Funções de tratamento de interrupção

- Duas diretivas são responsáveis pelas funções de tratamento de interrupções:
  - **#pragma** interrupt
    - Destinada á rotina de tratamento da interrupção de alta prioridade;
  - **#pragma** interruptlow
    - Destinada á rotina de tratamento da interrupção de baixa prioridade;

Interrupção Externa

- Para mostrar na prática como o recurso de interrupção pode ser utilizado em uma determinada aplicação, vamos ver como funciona a interrupção externa;
- O PIC184520 possui 3 interrupções externas:
  - RB2/INT2, RB1/INT1 e RB0/INT0;
  - A INT0 sempre possui alta prioridade;
  - Enquanto as outras podem ser configuras como de alta ou baixa prioridade;

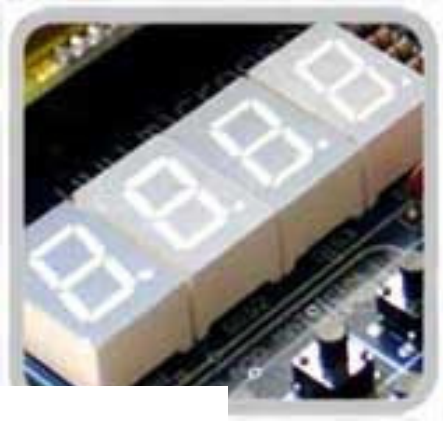
Interrupção Externa

- Prioridade
  - Os bits responsáveis pelas prioridades são INT2IP (INTCON3<7>) e INT1IP (INTCON3<6>);
- Borda
  - Os bits INTEDGx (INTCON2<6:4>) configuram a borda do sinal aplicado nos pinos que vai gerar a interrupção;
    - Se a borda de subida ou descida;

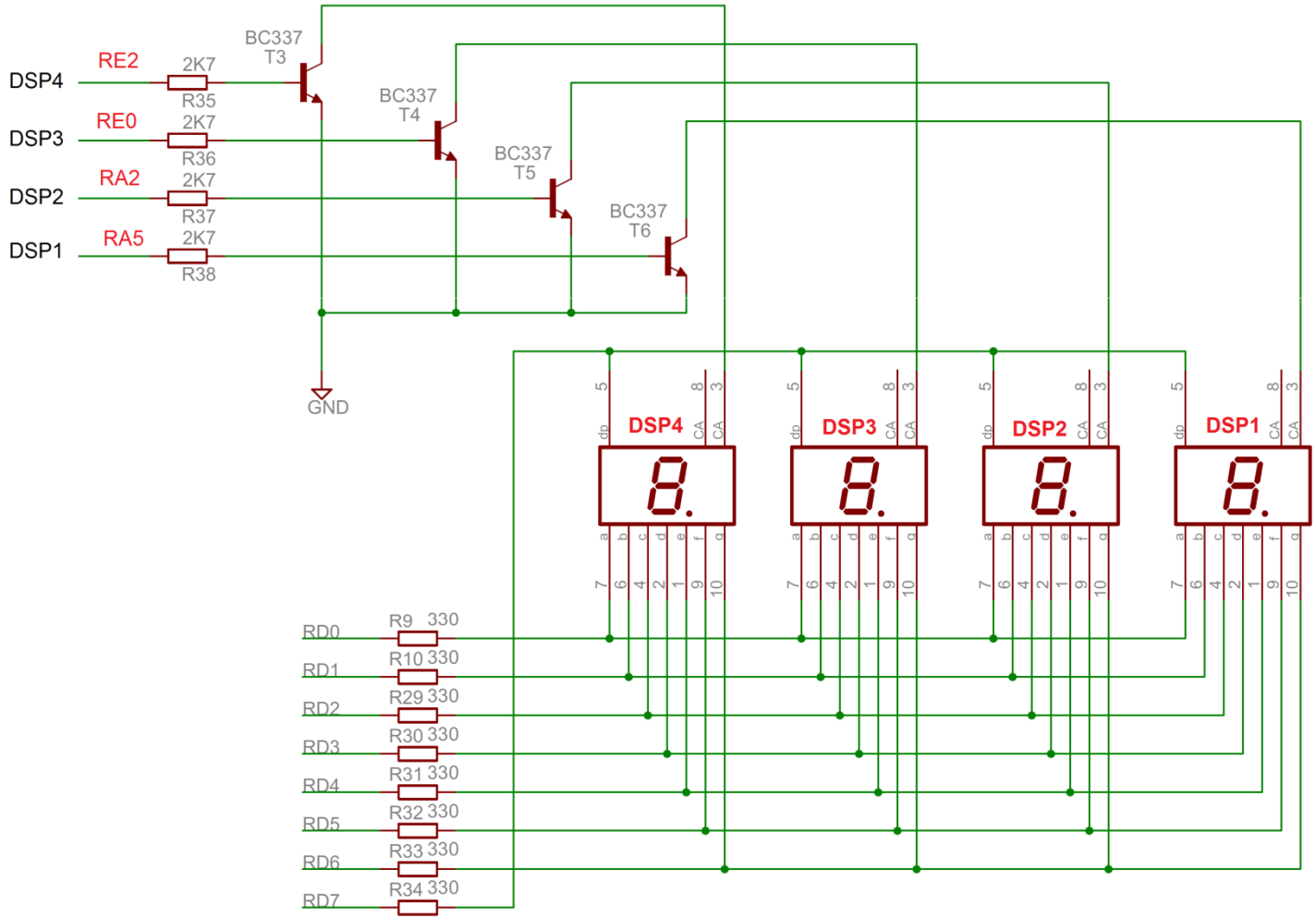
Interrupção Externa

- Habilitação
  - Cada uma das interrupções externas, assim como as demais pode ser habilitada individualmente por meio das chaves individuais:
    - INT2IE (INTCON3<4>), INT1IE (INTCON3<3>) e INT0IE (INTCON<4>);
  - Para habilitar a chave individual, deve-se também setar o bit associado àquela interrupção;
  - A chave geral (o bit GIE) também precisa estar setado para que o uC possa tratar a interrupção externa;

Display de Sete Segmentos



- Displays Multiplexados;



Interrupção Externa / Código-fonte

- DSP_7Seg_x4.h

  Arquivo cabeçalho com as definições dos pinos nos quais serão conectados os pinos do display.

- DSP_7Seg_x4.c

  Arquivo que compõe a biblioteca que contém a função que fará a atualização do display;

- Main_33.c

  Arquivo principal responsável por realizar a aplicação de um contador de 0 a 9.999 utilizando uma interface de vídeo com quatro displays multiplexados que é incrementado por um pulso no pino RB0/INT0;

DSP_7Seg_x4.h

Interrupção Externa / Código-fonte

Esse identificador impede que a definição a seguir seja duplicada se o arquivo cabeçalho foi incluído em outro arquivo-fonte associado ao projeto.

```
8  #ifndef __DSP_7SEGx4_H
9  #define __DSP_7SEGx4_H
10 //*********************************************************************
11 #include <p18cxxx.h>      //diretiva de compilação
12 //*********************************************************************
13 //definições
14 #define DSP_1    PORTAbits.RA5
15 #define DSP_2    PORTAbits.RA2
16 #define DSP_3    PORTEbits.RE0
17 #define DSP_4    PORTEbits.RE2
18 #define L_DADOS TRISD
19 #define DIR_A1    TRISAbits.TRISA5
20 #define DIR_A2    TRISAbits.TRISA2
21 #define DIR_A3    TRISEbits.TRISE0
22 #define DIR_A4    TRISEbits.TRISE2
23 //*********************************************************************
24 void Aciona_DPS_7_seg (unsigned char Dsp4, unsigned char Dsp3, unsigned char Dsp2, unsigned char Dsp1);
25 #endif
```

DSP_7Seg_x4.c

Display Multiplexado / Código-fonte

```
8   #include <p18cxxx.h>          //diretiva de compilação
9   #include "DSP_7Seg_x4.h"       //diretiva de compilação
10  //*********************************************************************
11  void Aciona_DPS_7_seg (unsigned char Dsp4, unsigned char Dsp3, unsigned char Dsp2, unsigned c
12  {
13  static unsigned char Atual_Dsp = 1;          //declaração  de variável local static inicializa
14  const char tabela[] = {
15                      0x3F,    // número 0
16                      0x06,    // número 1
17                      0x5B,    // número 2
18                      0x4F,    // número 3
19                      0x66,    // número 4
20                      0x6D,    // número 5
21                      0x7C,    // número 6
22                      0x07,    // número 7
23                      0x7F,    // número 8
24                      0x67,    // número 9
25                      0x00    //apaga display
26                     };
27  //*********
28  //configuração dos pinos
29     L_DADOS = 0x00;          //configura pinos das linhas de dados como saída
30     ADCON1 = 0x0F;          //configura Port A e Port E como pinos digitais
31     DIR_A1 = 0;             //configura linha de endereço A1 como saída
32     DIR_A2 = 0;             //configura linha de endereço A2 como saída
33     DIR_A3 = 0;             //configura linha de endereço A3 como saída
34     DIR_A4 = 0;             //configura linha de endereço A4 como saída
35  //*********************************************************************
```

Embora a função Aciona_DSP_7_seg() receba os valores de cada display, apenas um display é atualizado em cada chamada.

DSP_7Seg_x4.c

Display Multiplexado / Código-fonte

```
36  //atualiza display
37  if(Atual_Dsp==1)            //atualizar display 1
38      {
39            PORTD = tabela[Dsp1];    //atualiza display 1
40            DSP_1 = 1;               //ativa linha A1
41            DSP_2 = 0;               //desativa linha A2
42            DSP_3 = 0;               //desativa linha A3
43            DSP_4 = 0;               //desativa linha A4
44         Atual_Dsp = 2;           //aponta endereço para o próximo display
45      }
46      else    if(Atual_Dsp==2)    //atualizar display 2
47      {
48            PORTD = tabela[Dsp2];    //atualiza display 2
49            DSP_1 = 0;               //desativa linha A1
50         DSP_2 = 1;               //ativa linha A2
51          DSP_3 = 0;               //desativa linha A3
52          DSP_4 = 0;               //desativa linha A4
53         Atual_Dsp = 3;           //aponta endereço para o próximo display
54      }
```

Main_31.c

Display Multiplexado / Código-fonte

```
55      else   if(Atual_Dsp==3)   //atualizar display 3
56     {
57        PORTD = tabela[Dsp3];   //atualiza display 3
58        DSP_1 = 0;             //desativa linha A1
59        DSP_2 = 0;             //desativa linha A2
60        DSP_3 = 1;             //ativa linha A3
61        DSP_4 = 0;             //desativa linha A4
62        Atual_Dsp = 4;          //aponta endereço para o próximo display
63     }
64      else   if(Atual_Dsp==4)   //atualizar display 4
65     {
66        PORTD = tabela[Dsp4];   //atualiza display 4
67        DSP_1 = 0;             //desativa linha A1
68        DSP_2 = 0;             //desativa linha A2
69        DSP_3 = 0;             //desativa linha A3
70        DSP_4 = 1;             //ativa linha A4
71        Atual_Dsp = 1;          //aponta endereço para o próximo display
72     }
73  //*************************************************************************
74  }
```

Main_33.c

Display Multiplexado / Código-fonte

```
8  #include <p18f4520.h>                          //diretiva de compilação
9  #include <delays.h>                            //diretiva de compilação
10 #include "DSP_7Seg_x4.h"                       //diretiva de compilação
11 //*****************************************************************
12 //protótipos de funções
13 void Inic_Regs (void);
14 void high_isr (void);
15 //*****************************************************************
16 //variáveis globais
17 volatile unsigned char Dsp1=0;                  //declaração  de variável local ir
18 volatile unsigned char Dsp2=0;                  //declaração  de variável local ir
19 volatile unsigned char Dsp3=0;                  //declaração  de variável local ir
20 volatile unsigned char Dsp4=0;                  //declaração  de variável local ir
21 //*****************************************************************/
22 #pragma code high_vector=0x08                   //vetor de interrupção de alta pric
23 void interrupt_at_high_vector(void)
24 {
25     _asm GOTO high_isr _endasm                  //desvia programa para rotina de tr
26 }
27 #pragma code
```

Observe que as variáveis Dsp1, Dsp2, Dsp3, Dsp4 foram declaradas como volatile.

Istoé feito por recomendação do fabricante do MPLAB C18 porque elas são manipuladas dentro e fora da rotina de tratamento de interrupção ;

Main_33.c

Display Multiplexado / Código-fonte

```
29  //Rotina de tratamento de interrupção
30  #pragma interrupt high_isr
31  void high_isr (void)
32  {
33    if(!INTCONbits.INT0IF);                    //interrupção externa?
34    else
35    {
36       INTCONbits.INT0IF = 0;                  //sim, limpa bit de sinalização
37       Dsp1+=1;                               //incrementa unidade
38       if(Dsp1==10)                           //unidade estourou?
39       {
40          Dsp1=0;                             //sim, zera unidade
41          Dsp2+=1;                            //incrementa dezena
42       }
43       if(Dsp2==10)                           //dezena estourou?
44       {
45          Dsp2=0;                             //sim,zera dezena
46          Dsp3+=1;                            //incrementa centena
47       }
48       if(Dsp3==10)                           //centena estourou?
49       {
50          Dsp3=0;                             //sim, zera centena
51          Dsp4+=1;                            //incrementa unidade de milhar
52       }
53       if(Dsp4==10)                           //unidade de milhar estourou?
54       {
55          Dsp4=Dsp3=Dsp2=Dsp1=0;              //sim, zera contador
56       }
57    }
58  }
```

Verificação efetuada por um teste no bit de sinalização da interrupção externa, o bitINT0IF (INTCON<1>)

Confirmada a origem da interrupção, o bitINT0IF é então apagado

Main_33.c

Display Multiplexado / Código-fonte

```
60  void main(void)                                  //função main
61  {
62  //***********************************
63     Inic_Regs ();                                 //configurar SFRs
64     while(1)                                     //loop infinito
65     {
66        Aciona_DPS_7_seg (Dsp4, Dsp3, Dsp2, Dsp1);  //chamada à função: atualizar d
67        Delay1KTCYx(8);                            //delay de 4ms
68     }
69  }
70  /*********************************************************************
71  Esta funcao inicializa os registradores SFRs.*/
72   void Inic_Regs (void)
73  {
74     TRISA = 0x00;                                //PORTA saída
75     TRISB = 0x01;                                //RB0 como entrada e demais pinnos do
76     TRISC = 0x00;                                //PORTC saída
77     TRISD = 0x00;                                //PORTD saída
78     TRISE = 0x00;                                //PORTE saída
79     ADCON1 = 0x0F;                                //configura pinos dos PORTA e PORTE c
80     PORTA = 0;                                   //limpa PORTA
81     PORTB = 0;                                   //limpa PORTB
82     PORTC = 0;                                   //limpa PORTC
83     PORTD = 0x00;                                //apaga displays
84     PORTE = 0;                                   //limpa PORTE
85  //***************************************
86  //habilita interrupção externa
87     INTCONbits.GIE = 1;                           //liga chave geral de interrupção
88     INTCONbits.INT0IE = 1;                        //liga chave individual de interrup
89     INTCON2bits.INTEDG0 = 0;                     //interrupção externa 0 ocorrerá na
90  //*********************************************************************
91  }
```

Configurações para Interrupção Externa

Próxima Aula

Aula 15

Timers