Faculdade de Computação e Engenharia Elétrica
Microprocessadores e Microcontroladores – Prof. Dr. Elton Alves
Experimento 8 – Projeto Assembly no PIC16F628A

- **Objetivo:**

- Utilizar registradores de uso geral, interrupções e temporização com timer.

- **Rodar o código 6 no Mplab para testar a criação de variáveis em registradores de uso geral**

```
; Microcontroladores e Microprocessadores
; Aula 03 - Registradores de Uso Geral
; Prof. Elton Alves
; Aciona LED1 ligado em RB1 e apaga LED2 ligado em RB3
;aguarda 500 milisegundos
;Aciona LED2 ligado em RB3 e apaga LED1 em RB1
;aguarda 500 milisegundos
;0 - botão acionado
;1 - botão desligado

;calculo de ciclos de Máquina
;ciclo de máquina = 1/(Freq. Cristal/4)= 1us


;list p=16f628A ; microcrontrolador utilizado
;---Arquivos incluidos no projeto---
#include <P16f628a.inc> ; inclui o arquivo do 16f628a (registradores)

;---FUSE bits---
;Cristal oscilador externo 4MHZ
;Sem watchdog time
;Com powe up time
   __config _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF

;---Paginação de Memória
#define  bank0   bcf STATUS, RP0 ;cria um mnemônico para o banco 0 de memória
#define  bank1   bsf STATUS, RP0 ; cria um mnemônico para o banco 1 de memória

;---Entradas---
 #define botao1  PORTB,RB0 ; botão 1 ligado em RB0
 #define botão2  PORTB,RB2; botão 2 ligado em RB2

;---Saídas---
 #define led1 PORTB,RB1 ; led1 ligado em RB1
 #define led2 PORTB,RB3 ;led2 ligado em RB3

;---Registradores de Uso Geral----
```

```
;Primeira forma de declaração
;tempo1 equ H'20'
;tempo2 equ H'21'

;Segunda forma de declaração
cblock H'20' ; inicio da memória do usuário
tempo1 ; registrador auxiliar para temporização 1
tempo2 ; registrador auxiliar para temporização 2
endc ; final de memória do usuário


;---Vetor de RESET---
    org H'000' ; origem no endereço 000h de memória
    goto inicio ; desvia do vetor de interrupção

;---Vetor de Interrupção----
    org H'0004'  ; todas as interruopções apontam para este endereço
    retfie ; retorna a interrupção

;---Programa Principal----
inicio
    CLRF PORTA ; Limpa PORTA
    CLRF PORTB ; Limpa PORTB

    bank1 ; seleciona o banco 1 de memória
    movlw H'FF' ; w=B'11111111
    movwf TRISA ; TRISA=H'FF' (todos bits são entradas)
    movlw H'F5' ; w=B'11110101'
    movwf TRISB ;TRISB=H'F5' (apenas RB1 e RB3 como saída)

    bank0 ; seleciona o banco 0 de memória (padrão RESET)
    movlw H'F5' ; w=B'11110101'
    movwf PORTB; (Leds iniciam desligados)

   ; goto $ ; segura o código - $ indica a posicial atual do código (loop infinito)


loop ; loop infinito

   bsf led1 ; liga LED1
   bcf led2 ; desliga LED2
   call delay500ms ;chama sub_rotina
   bcf led1 ; desliga LED1
   bsf led2 ; liga LED1
   call delay500ms ;chama sub-rotina

   goto loop ;volta para o label loop

   ;---Desenvolvimento das Sub-rotinas----
```

```
delay500ms
   movlw D'200' ; move o valor para W (constante)
   ;movwf H'20' ; inicialização da variavel tempo0 (posição de memória do registrador de uso geral)
   movwf tempo1 ; variavel tempo1
aux1
   movlw D'250'
   ;movwf H'21'
   movwf tempo2; variavel tempo2

aux2 ; gastar 1 ciclo de máquina (aproximar mais o tempo de 500ms)
   nop
   nop
   nop
   nop
   nop
   nop
   nop

   ;decfsz H'20' ; decrementa o tempo1 até que seja igual a 0(decremente em uma unidade e verifica se é 0)
   decfsz tempo2
   goto aux2 ; vai para label aux2
   ; 250 x 10 ciclos de máquina = 2500

   ;decfsz H'32'; decrementa o tempo0 até seja igual a 0
   decfsz tempo1
   goto aux1; vai para label aux1

   ; 3 ciclos de máquina
   ; 2500x200 = 500000

   return


    end
```

- **Rodar o código 7 no Mplab para testar as interrupções**

```
;Aula 04: Como entrar e sair de uma Interrupção (Salvamento de Contexto)
;Prof. Elton Alves

 ; Clock: 4MHz e Ciclo de Máquina  = 1us
 ;---Listagem do Processador Utilizado---
 list p=16F628A
 #include <P16f628a.inc>
 ;---Arquivos Inclusos no Projeto ---

 ;---FUSE Bits----
 ; Cristal de 4MHz
 ; Desabilitamos o Watch Dog Timer
```

```
; Habilitamos o Power Up Time
; Brown Out desabilitado
; Sem programação em baixa tensão, sem proteção de código

__config _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF

;---Paginação de Memória----
#define bank0 bcf STATUS, RP0
#define bank1 bsf STATUS, RP0

;---Registradores de Uso Geral----
cblock H'20'

W_TEMP ;  registrador para armazenar o conteúdo temporarío de w
STATUS_TEMP ; registrador para armazenar o conteudo temporário de STATUS
endc

;---Vetor de RESET
org  H'0000'
goto inicio

;---Vetor de Interrupção----
org H'0004' ; as interrupções apontam para esse endereço de memória

;---Salva Contexto----

MOVWF W_TEMP  ; copia o contéudo de w para W_TEMP
SWAPF STATUS,W  ; move o conteudo de STATUS com os nibles invertidos para w
bank0 ; seleciona o banco zero de memória
MOVWF STATUS_TEMP ; copia o conteúdo de STATUS com os nibles invertidos para
STATUS_TEMP
;-- Final de Salvamento de Contexto----

;----Recupera contexto (saída da interrupção)---
exit_ISR

SWAPF STATUS_TEMP,W ; copia em w o conteúdo de STATUS_TEMP com nibles
invertidos.
MOVWF STATUS        ; recuperando o conteúdo de STATUS
SWAPF W_TEMP,F      ;W_TEMP = W_TEMP com nibles invertidos
SWAPF W_TEMP,W      ; recupera conteúdo de w

retfie      ; retorna da interrupção

inicio
 bank1 ; seleciona o banco 0 de memória
 movlw H'00' ;w=00h (inicia o registrador em 0)
 movwf OPTION_REG ; configurar o OPTION_REG (Interrupções)
 movlw H'FE' ; w = FEh
 movwf TRISB ; configura entradas/saídas PORTB
```

```
 bank0      ; seleciona banco 0 de memória
 movlw H'07' ; w=7h
 movwf CMCON; CMCON = 7h desabilita os comparadores
 movlw H'00'; w=00h (inicia o registrador em 0)
 movwf INTCON ; configurar INTCON (Configura interrupções)

loop

 movlw H'01' ; move literal 01h para work
 xorwf PORTB, F ; inverte o estado de RB0
 goto H'0004' ; força o desvio para o vetor de interrupção
           ; e não deve ser utilizado na prática

              goto loop
end
```

- **Rodar o código 8 no Mplab para testar temporização TMR0**

```
 ; Temporização com TIMER0
; Interrupção com TIMER0

 ; Clock: 4MHz e Ciclo de Máquina  = 1us
 ;---Listagem do Processador Utilizado---
 list p=16F628A
 #include <P16f628a.inc>
 ;---Arquivos Inclusos no Projeto ---

 ;---FUSE Bits----
 ; Cristal de 4MHz
 ; Desabilitamos o Watch Dog Timer
 ; Habilitamos o Power Up Time
 ; Brown Out desabilitado
 ; Sem programação em baixa tensão, sem proteção de código

 __config _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF

 ;---Paginação de Memória----
 #define bank0 bcf STATUS, RP0
 #define bank1 bsf STATUS, RP0

 ;---Registradores de Uso Geral----
 cblock H'20'

 W_TEMP ;  registrador para armazenar o conteúdo temporarío de w
 STATUS_TEMP ; registrador para armazenar o conteudo temporário de STATUS
 endc

 ;---Vetor de RESET
 org  H'0000'
 goto inicio
```

```
;---Vetor de Interrupção----
org H'0004' ; as interrupções apontam para esse endereço de memória

;---Salva Contexto----

MOVWF W_TEMP  ; copia o contéudo de w para W_TEMP
SWAPF STATUS,W  ; move o conteudo de STATUS com os nibles invertidos para
w
bank0 ; seleciona o banco zero de memória
MOVWF STATUS_TEMP ; copia o conteúdo de STATUS com os nibles invertidos
para STATUS_TEMP
;-- Final de Salvamento de Contexto----

;--- Trata ISR
btfss INTCON, T0IF ; ocorreu um overflow no TIMER0
goto exit_ISR ; NÃO, desvia para saída da interrupção
movlw D'10'   ; move a literal 10d para w
bcf INTCON,T0IF ; SIM limpa a flag
comf PORTA ; complementando todo registrador PORTA

;----Recupera contexto (saída da interrupção)---
exit_ISR

SWAPF STATUS_TEMP,W ; copia em w o conteúdo de STATUS_TEMP com
nibles invertidos.
MOVWF STATUS        ; recuperando o conteúdo de STATUS
SWAPF W_TEMP,F      ;W_TEMP = W_TEMP com nibles invertidos
SWAPF W_TEMP,W      ; recupera conteúdo de w

retfie      ; retorna da interrupção

inicio
bank1 ; seleciona o banco 0 de memória
movlw H'80' ;w=80h (inicia o registrador em 0)
movwf OPTION_REG ; configurar o OPTION_REG (Interrupções)
              ; PULL ups internos desabilitzados
               ;Timer0 incrimenta com ciclo de máquima
               ;Prescaler 1:2 associado ao Timer0

movlw H'F3' ; w = F3h
movwf TRISA ; configura a saída dos LEDS

bank0      ; seleciona banco 0 de memória
movlw H'07' ; w=7h
movwf CMCON; CMCON = 7h desabilita os comparadores
movlw H'A0'; w=A0h (inicia o registrador em 0)
movwf INTCON ; configurar INTCON (Configura interrupções)
           ; habilita a interrupção global
          ; habilita a interrupção do TIMER0
```

```
movlw D'10'; move a literal 10 para w
movwf TMR0 ; inicializando timer0 com 10d (256-10 = 246)

goto $ ; aguarda a interrupção

end
```

ATIVIDADE AVALIATIVA:

- Desenvolva uma simulação que permita um LED ficar aceso enquanto o TIMER0 ficar rodando.
- Data da entrega: 28/07/2021
- O relatório deve ser enviado, juntando com .HEX e o arquivo de simulação.