

PROGRAMA DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Disciplina: Laboratório de Sistemas microprocessados

PRÁTICA 1

Objetivo:

- Introduzir principais funcionalidades do ambiente de desenvolvimento
- Entender a memória de programa, memória dados e principais registradores
- Entender os modos de endereçamento

Atividades

1) Faça apenas um programa em assembly que realiza as seguintes instruções:

- Carrega o valor literal 0x5A no acumulador
- Move este valor para a primeira posição da memória de acesso
- Move o conteúdo do endereço 0x000 para o endereço 0x3AA.
- Move o valor do acumulador para o endereço 0x5AA da memória RAM.

2) Faça a depuração do programa passo a passo monitorando a memória de dados e de programa a cada instrução. Utilizando as observações ao depurar o programa, descreva com suas palavras o seu entendimento sobre os pontos discutidos em sala: endereçamento direto, endereçamento literal, memória de dados, memória de programa e set de instruções. Utilize as telas do programa para exemplificar.

Anexo A

<div><div>MOVLW</div><div>Move Literal to W</div></div> <div>Syntax:MOVLW k</div> <div>Operands:$0 \leq k \leq 255$</div> <div>Operation:$k \rightarrow W$</div> <div>Status Affected:None</div> <div>Encoding:<table><tr><td>0000</td><td>1110</td><td>kkkk</td><td>kkkk</td></tr></table></div> <div>Description:The eight-bit literal 'k' is loaded into W.</div>	0000	1110	kkkk	kkkk	<div><div>MOVLB</div><div>Move Literal to Low Nibble in BSR</div></div> <div>Syntax:MOVLB k</div> <div>Operands:$0 \leq k \leq 255$</div> <div>Operation:$k \rightarrow \text{BSR}$</div> <div>Status Affected:None</div> <div>Encoding:<table><tr><td>0000</td><td>0001</td><td>kkkk</td><td>kkkk</td></tr></table></div> <div>Description:The eight-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0' regardless of the value of $k_7:k_4$.</div>	0000	0001	kkkk	kkkk
0000	1110	kkkk	kkkk						
0000	0001	kkkk	kkkk						
<div><div>MOVWF</div><div>Move W to f</div></div> <div>Syntax:MOVWF f {,a}</div> <div>Operands:$0 \leq f \leq 255$ $a \in [0,1]$</div> <div>Operation:(W) \rightarrow f</div> <div>Status Affected:None</div> <div>Encoding:<table><tr><td>0110</td><td>111a</td><td>ffff</td><td>ffff</td></tr></table></div> <div>Description:Move data from W to register 'f'.</div>	0110	111a	ffff	ffff	<div><div>MOVF</div><div>Move f</div></div> <div>Syntax:MOVF f {,d {,a}}</div> <div>Operands:$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$</div> <div>Operation:f \rightarrow dest</div> <div>Status Affected:N, Z</div> <div>Encoding:<table><tr><td>0101</td><td>00da</td><td>ffff</td><td>ffff</td></tr></table></div> <div>Description:The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).</div>	0101	00da	ffff	ffff
0110	111a	ffff	ffff						
0101	00da	ffff	ffff						
<div><div>MOVFF</div><div>Move f to f</div></div> <div>Syntax:MOVFF f_s,f_d</div> <div>Operands:$0 \leq f_s \leq 4095$ $0 \leq f_d \leq 4095$</div> <div>Operation:(f_s) \rightarrow f_d</div> <div>Status Affected:None</div> <div>Encoding:<table><tr><td>1100</td><td>ffff</td><td>ffff</td><td>ffff_s</td></tr><tr><td>1111</td><td>ffff</td><td>ffff</td><td>ffff_d</td></tr></table></div> <div>Description:The contents of source register 'f_s' are moved to destination register 'f_d'.</div>		1100	ffff	ffff	ffff _s	1111	ffff	ffff	ffff _d
1100	ffff	ffff	ffff _s						
1111	ffff	ffff	ffff _d						

a = 0 -> Access Bank

a = 1 -> BSR

d = 0 -> Destino Acumulador

d = 1 -> Destino Registrador

Template

```
#include <p18f4550.inc>
```

```
config CPUDIV = OSC1_PLL2 ;
config FOSC = HS ;
config WDT = OFF ;
config PBADEN = OFF ;
config LVP = OFF ;
config MCLRE = ON ;
config PWRT = ON ;
```

```
org 0x0000
bra start
```

```
org 0x0020
```

start:

```
;codigo
bra start
```

end