CEFET-MG CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

PROGRAMA DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Disciplina: Laboratório de Sistemas microprocessados

PRÁTICA 2

Objetivo:

Apresentar a ULA – Unidade Lógica Aritmética
Conhecer o Registrador de Status

Exercícios

1) Faça um programa em assembly que soma um valor literal com o acumulador conforme equação: W=W+8. Explique, monitorando registradores de status e acumulador, como o processador realiza esta soma nas seguintes situações:

a) carregue o valor 10 no acumulador antes de realizar a soma.
b) carregue o valor 120 no acumulador antes de realizar a soma.
c) carregue o valor 250 no acumulador antes de realizar a soma.
d) Como os valores negativos são representados?

2) Faça um programa em assembly que realiza a seguinte operação matemática: VB = VB+VA. Considere VA e VB variáveis de 8 bits. Explique, monitorando registradores de status e memória, como a ULA realiza nas seguintes situações:
a) O resultado da soma é superior a 256.
b) O resultado da soma é superior a 127.
c) Realize a operação: VB = VB-VA

Dicas:

- **Utilize a diretiva DEFINE para criar as variáveis.**
- **Os flags N e OV somente possuem utilidade nas operações que envolvem operandos sinalizados (signed)**

3) Repita o exercício (2) considerando as variáveis VA e VB como 16 bits.

Dica:

- **O número de 16 bits deve ser criado em dois bytes da memória, exemplo: VAH e VAL, onde H é para high e L para low.**

4) Faça um programa em assembly que realiza a seguinte tarefa: mescla as variáveis A e B em uma variável C. O nibble menos significativo de C deve receber o nibble menos significativo de A e o nibble mais significativo de C deve receber o nibble mais significativo de B.

Dica: utilize a função AND como máscara para selecionar o nibble desejado e a função OR para mesclar as duas variáveis.

5) Resolva a seguinte equação VC = VB/4+VA*2

Dica: Utilize a função de rotação para multiplicar ou dividir o número por dois.

Anexo A

Instruções Aritméticas

Mnemônicos		Descrição	N ciclos	Flags Afetados
ADDLW	K	Add Literal and WREG	1	C, DC, Z, OV, N
ADDWF	f, d, a	Add WREG and f	1	C, DC, Z, OV, N
ADDWFC	f, d, a	Add WREG and Carry bit to f	1	C, DC, Z, OV, N
DECF	f, d, a	Decrement f	1	C, DC, Z, OV, N
INCF	f, d, a	Increment f	1	C, DC, Z, OV, N
NEGF	f, a	Negate f	1	C, DC, Z, OV, N
SUBFWB	f, d, a	Subtract f from WREG with Borrow	1	C, DC, Z, OV, N
SUBLW	K	Subtract WREG from Literal	1	C, DC, Z, OV, N
SUBWF	f, d, a	Subtract WREG from f	1	C, DC, Z, OV, N
SUBWFB	f, d, a	Subtract WREG from f with Borrow	1	C, DC, Z, OV, N

Instruções Lógicas

Mnemônicos		Descrição	N ciclos	Flags Afetados
ANDLW	K	AND Literal with WREG	1	Z, N
ANDWF	f, d, a	AND WREG with f	1	Z, N
COMF	f, d, a	Complement f	1	Z, N
IORLW	K	Inclusive OR Literal with WREG	1	Z, N
IORWF	f, d, a	Inclusive OR WREG with f	1	Z, N
RLCF	f, d, a	Rotate Left f through Carry	1	C, Z, N
RLNCF	f, d, a	Rotate Left f (No Carry)	1	Z, N
RRCF	f, d, a	Rotate Right f through Carry	1	C, Z, N
RRNCF	f, d, a	Rotate Right f (No Carry)	1	Z, N
XORLW	K	Exclusive OR Literal with WREG	1	Z, N
XORWF	f, d, a	Exclusive OR WREG with f	1	Z, N

Instruções de Movimentação de dados

Mnemônicos		Descrição	N ciclos	Flags Afetados
CLRF	f, a	Clear f	1	Z
LFSR	f, k	Move Literal (12-bit) 2nd word to FSR(f) 1st word	2	None
MOVF	f, d, a	Move f	1	Z, N
MOVFF	fs, fd	Move fs (source) to fd (destination)	2	None
MOVLB	K	Move Literal to BSR<3:0>	1	None
MOVLW	K	Move Literal to WREG	1	None
MOVWF	f, a	Move WREG to f	1	None