

Projeto Final

Arquitetura e Organização de Computadores

Danton Joshua Silva da Conceição

Kauan Henrique Werlich

Letícia Capitani Trapp Sampaio

1. Descreva o trecho de programa em assembly do PIC10F202 abaixo e mostre um trecho de programa equivalente em C.

```
main:
    MOVLW 10      ; W = 10
    MOVWF 0x08    ; MEM[0x08] = W
    MOVLW 0       ; W = 0
loop:
    DECFSZ 0x08, 1 ; MEM[0x08]--, pula a próxima instrução se resultado é 0
    GOTO continue
break:
    GOTO fim
continue:
    ADDWF 0x08, 0  ; W = W + MEM[0x08]
    GOTO loop
fim:
    MOVWF 0x08    ; MEM[0x08] = W
    END           ; Pseudo instrução que termina o programa
```

Resolução:

Descrição passo a passo do código:

- O código acima realiza a operação em W: $9+8+7+6+5+4+3+2+1 = 45$, ou seja, o **somatório de 1 a 9**;
- É atribuído ao registrador W o valor literal 10 (inteiro);
- Em seguida é atribuído o valor de W no registrador 0x08;
- Depois é atribuído o valor literal 0 a W;
- Em seguida vamos para a próxima instrução (loop) que decrementa o valor contido em 0x08 em 1, enquanto o valor decrementado não for igual a 0 (caso seja, pula a próxima instrução) vamos para a instrução/label “continue” que soma a W o valor atual em 0x08, o argumento 0 nessa instrução diz que o valor deve ser salvo em W, em seguida voltamos para o loop;
- No break apenas vamos para o fim do programa;
- Na label “fim” o valor de W é movido para o registrador 0x08 e o programa encerra;
- Foi colocado print no código abaixo apenas para confirmar/fazer a prova real do que foi supracitado.

Implementação de código equivalente em C:

```
#include <stdio.h>
```

```

int main() {
int w = 10;
int e0x08 = w;
w = 0;
while (e0x08 != 0) {
    e0x08--;
    w+= e0x08;
}
printf ("w = %i",w);
return 0;
}

```

2. Traduza o trecho de programa em C abaixo para a linguagem assembly do PIC10F202. O resultado final de cada variável deve ser colocado em endereços de dados (não especiais) do *file register* (e não mantido em W).

```

1 int main() {
2     int x, y, z, q;
3     x = 4;
4     y = 7;
5     while(x) {
6         z = x + y;
7         q = x & y;
8         if (z == q) {
9             break;
10        }
11        x--;
12    }
13    return 0;
14 }

```

Resolução

Código traduzido para Assembly (PIC10F202):

main:

```

    movlw 4
    movwf 0x0A; x = 4
    movlw 7
    movwf 0x0B; y = 7

```

loop:

```

    movf 0x0A, 0; w = x
    movwf 0x0C; z = x
    movwf 0x0D; q = x
    movf 0x0B, 0; w = y
    addwf 0x0C, 1; z = x + y
    andwf 0x0D, 1; q = x & y
    movf 0x0C, 0; w = z
    subwf 0x0D, 0; w = z - q

```

```

movwf 0x0E, 1; varauxiliar = z - q
incf 0x0E, 1; varauxiliar++
decfsz 0x0E, 1; varauxiliar--, se a varauxiliar for igual à 0 vamos para o break
goto continue

```

break:

```
goto final
```

continue:

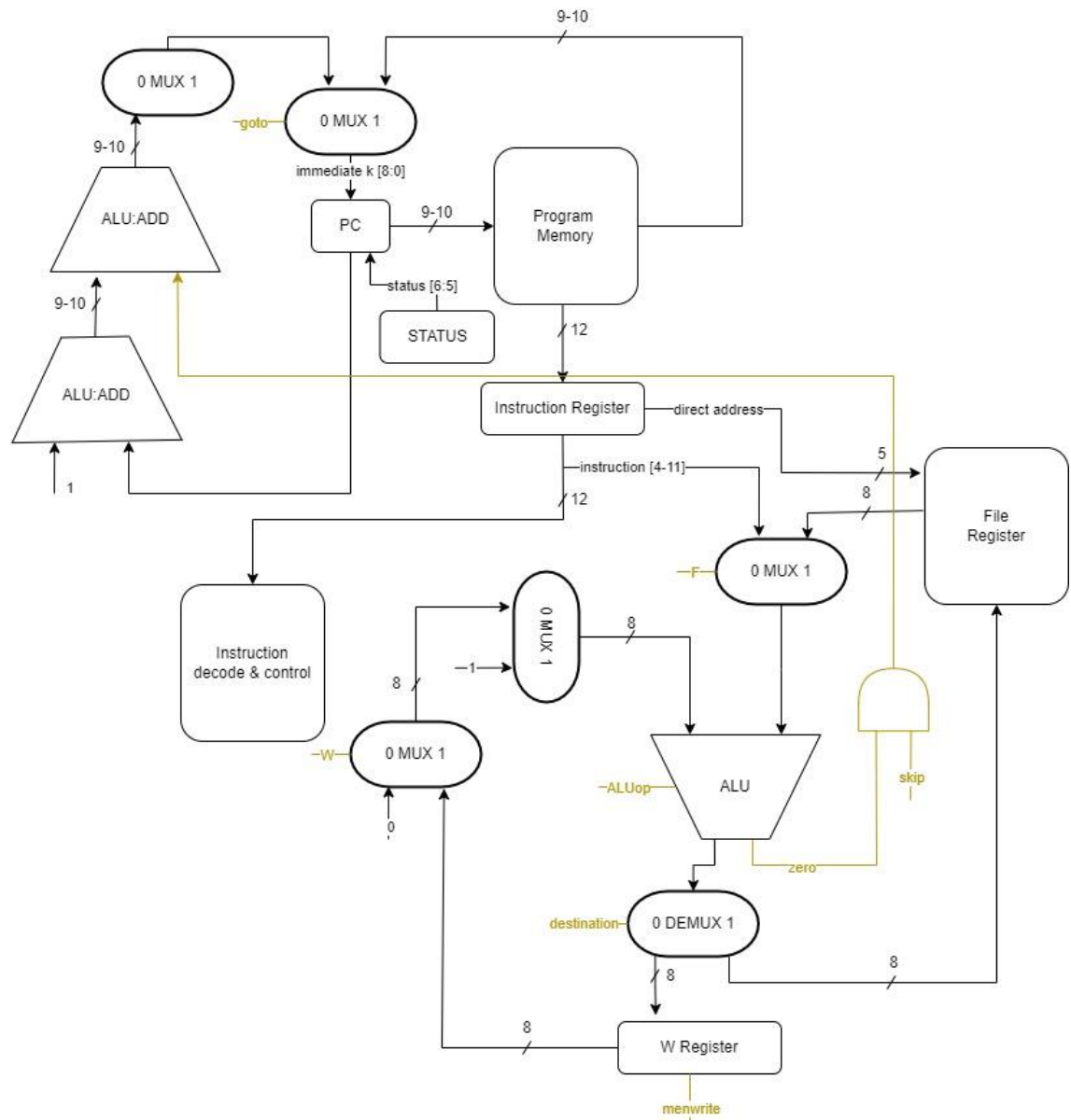
```
decf 0x0A
```

```
goto loop
```

final:

```
END
```

3. Faça o caminho de dados deste microcontrolador suficiente para executar os programas acima. Apresente a ALU, o registrador W, a memória de dados, os multiplexadores, os sinais de controle, o PC e demais componentes.



4. Indique os sinais de controle para cada instrução utilizada nas questões 1 e 2 conforme seu projeto na questão 3. Se duas instruções utilizam os mesmos sinais, agrupe elas em uma única resposta.

Control Signals									
	goto	skip	ALUop			F	W	destination	menwrite
movlw	0	0	and	or	xor	0	1	0	1
movwf	0	0	and	or	xor	0	0	1	1
Movf	0	0	and	or	xor	1	1	opcode	1
addwf	0	0	add			1	0	opcode	1
andwf	0	0	and			1	0	opcode	1
subwf	0	0	sub			1	0	opcode	1
incf	0	0	add			1	1	opcode	1
decfsz	0	1	sub			1	1	opcode	1
decf	0	0	sub			1	1	opcode	1
goto	1	0	add	or	xor	0	0	0	0