



Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – CCT
Departamento de Ciência da Computação

Disciplina: AOC

Semestre: 2022/1

Professor(a): Yuri Kaszubowski Lopes

Estudante: Arthur Jose Budal da Silva, Arthur Henrique Cavalcanti, Victor Eduardo Requia

PIC10F202

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

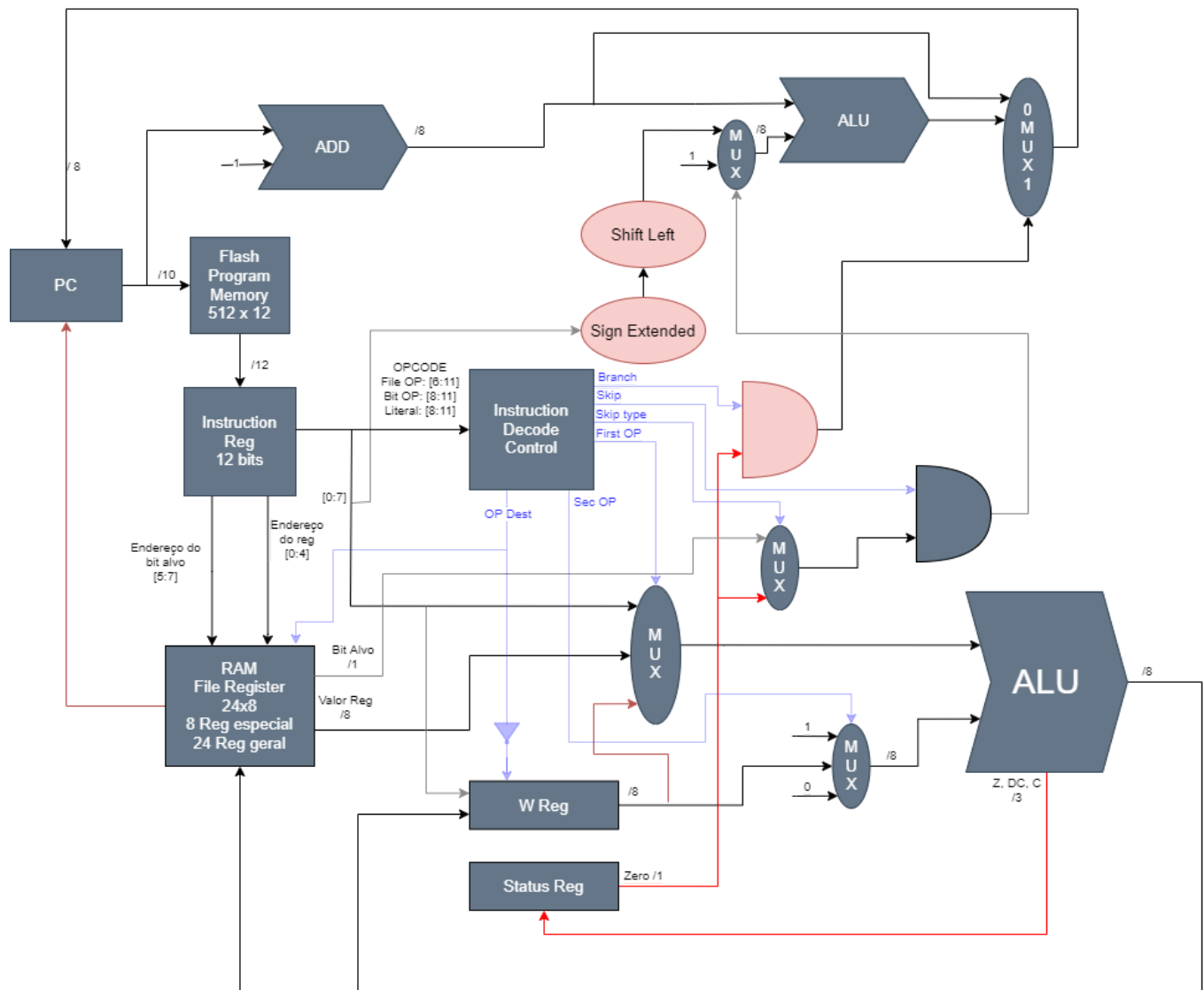
O programa realiza um somatório de 9 até 1, $9+8+7+6+5+4+3+2+1$

```
#include <stdio.h>
int main() {
    int w = 10, f = w;
    w = 0;
    while(f > 0) {
        f--;
        w += f;
    }
    f = 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 algum endereço de dados (não especial) do file register (e não mantido em W)

```
1. #include <pic10f202.inc>
2. PSECT Code,class=CODE,relloc=2,delta=2
3.
4.     main:
5.
6.     MOVLW 3      // W = 3
7.     MOVWF 0x08   // MEM[0x08] = W// 0x08 é x
8.     MOVLW 5      // W = 5
9.     MOVWF 0x09   // MEM[0x09] = W// 0x09 é y
10.    INCF 0x08    // MEM[0x08]++
11. loop:
12.    DECFSZ 0x08, 1 // MEM[0x08]--, skip if 0
13.    GOTO continue
14.
15. fim_loop:
16.    GOTO fim
17.
18. continue:
19.    //0x0A é z e 0x0B é q
20.
21.    // z = x + y
22.    MOVF 0x08, 0 // W = MEM[0x08] ou W = x
23.    ADDWF 0x09, 0 // W = W + MEM[0x09] ou W = x + y
24.    MOVWF 0x0A // MEM[0x18] = W ou z = W
25.
26.    // q = x & y
27.    MOVF 0x08, 0 // W = MEM[0x08] ou W = x
28.    ANDWF 0x09, 0 // W = W & MEM[0x10] ou W = x & y
29.    MOVWF 0x0B // MEM[0x20] = W ou q = W
30.
31.    // if(z == q) break
32.    MOVF 0x0B, 0 // W = MEM[0x0A] ou W = q
33.    SUBWF 0x0A, 0 // W = W - MEM[0x0B] ou W = z - q, se
    resultado for 0, status bit Z é 1
34.    BTFSC 0x03, 2 // Checa o status bit Z, se Z = 1, são
    iguais e dá o break, se não, dá o skip
35.    GOTO fim_loop
36.    GOTO loop
37. fim:
38.    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 instruções 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

Planilha 1: Sinais de Controle

OP / bit	0	1	10
Branch	PC é incrementado normalmente em 1 bit	PC é incrementado em 1 + o resultado da ALU de soma (que será definido pelos próximos sinais de controle) Uso: GOTO, DECFSZ, BTFSC	X
Skip	Se houver um salto (Branch = 1), o salto é determinado pelo literal enviado pela instrução Uso: GOTO	Se houver um salto (Branch = 1), o salto será de apenas uma instrução (SKIP) Uso: DECFSZ, BTFSC	X
Skip Type	Se houver um skip (Skip = 1), o skip é determinado pelo valor do bit do file register, determinado pela instrução Uso: BTFSC	Se houver um skip (Skip = 1), o skip é determinado pelo bit Z do registrador de status, ou seja, se o resultado da operação lógica ou aritmética for 0 Uso: DECFSZ	X
First OP	Primeiro operando que será enviado para a ALU é um literal. Uso: MOVLW	Primeiro operando que será enviado para a ALU é o valor armazenado em um file register. Uso: ADDWF, ANDWF, DECF, DECFSZ, INCF, MOVF, SUBWF	Primeiro operando que será enviado para a ALU é o valor armazenado em W Uso: MOVEWF
Sec OP	Segundo operando que será enviado para a ALU é o valor 1. Uso: DECF, DECFSZ, INCF	Segundo operando que será enviado para a ALU é o valor armazenado em W Uso: ADDWF, ANDWF, MOVEWF, SUBWF	Segundo operando que será enviado para a ALU é o valor 0 Uso: MOVF
Op Dest	Resultado da ALU é escrito no registrador W. Uso: ADDWF, ANDWF, DECF, DECFSZ, INCF, MOVF, SUBWF.	Resultado da ALU é escrito no file register. Uso: ADDWF, ANDWF, DECF, DECFSZ, INCF, MOVF, SUBWF.	X

Planilha 2: Operações

Operação / Sinal	Branch	Skip	Skip Type	First OP	Sec OP	OP dest	ALUOP
ADDWF	0	X	X	1	1	Definido por d	ADD
ANDWF	0	X	X	1	1	Definido por d	AND
DECF	0	X	X	1	0	Definido por d	SUB
DECFSZ	1	1	1	1	0	Definido por d	SUB
INCF	0	X	X	1	0	Definido por d	ADD
MOVF	0	X	X	1	10	Definido por d	ADD ou OR
MOVWF	0	X	X	10	1	1	AND
SUBWF	0	X	X	1	1	Definido por d	SUB
BTFSC	1	1	0	X	X	X	X
GOTO	1	0	X	X	X	X	X
MOVLW	0	X	X	0	10	0	ADD ou OR