

# Projeto AOC PIC10F202

André de Campos<sup>1</sup>, Leonardo Vinicius Salas Sell<sup>1</sup>, Kalyl Heings<sup>1</sup>

<sup>1</sup>Departamento de Ciências da Computação - Universidade do Estado de Santa Catarina (UDESC)  
R. Paulo Malschitzki, 200 - Zona Industrial Norte, Joinville - SC, 89219-710

{andre.campos0702,leonardo.sell,kalyl.henigs}@edu.udesc.br

## 1. Proposto do Projeto

O trabalho proposto pelo Professor Dr Yuri Kaszubowski Lopes é um projeto utilizando o microcontrolador PIC10F202, trabalhando com seu assembly e construindo os caminhos de dados necessários para executar as instruções dos programas em assembly PIC.

## 2. Questões

### 2.1. Quetão 1

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

Listing 1. Código PIC assembly

```
1 main:
2     MOVLW 10 ; W = 10
3     MOVWF 0x08 ; MEM[0x08] = W
4     MOVLW 0 ; W = 0t
5 loop:
6     DECFSZ 0x08, 1 ; MEM[0x08]--, pula a prox. inst. se res. 0
7     GOTO continue
8     break:
9     GOTO fim
10 continue:
11     ADDWF 0x08, 0 ; W = W + MEM[0x08]
12     GOTO loop
13 fim:
14     MOVWF 0x08 ; MEM[0x08] = W
15     END ; Pseudo inst. que termina o programa
```

## 2.2. Questão 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).

**Listing 2. Código C**

```
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 }
```

## 2.3. Questão 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.

## 2.4. Questão 4

Indique em uma tabela os sinais de controle para cada instrução utilizada nas Questões 1 e 2 conforme seu projeto na Questão 3. Se duas ou mais instruções utilizam os mesmos sinais agrupe elas na tabela.

## 3. Soluções

### 3.1. Questão 1

O programa abaixo é um somatório que vai de 10 até 0, somando um decremento de acc em w.

**Listing 3. Solução em C**

```
1 int w = 10, acc;
2 acc = w;
3 w = 0;
4 while ( acc > 0 ){
5     acc--;
6     w = w + acc;
7 }
8 acc = w;
```

### 3.2. Questão 2

O código em questão verifica se  $x$  e  $y == x+y$ , caso não, ele reduz 1 do valor de  $x$ .

#### Listing 4. Solução em PIC assembly

```
1  main:
2      MOVLW 4 ; W = 4
3      MOVWF 0x08 ; MEM[0x08] = W (neste caso variavel x)
4      MOVLW 7 ; W = 7
5      MOVWF 0x09 ; MEM[0x09] = W (neste caso variavel y)
6      MOVLW 0 ; W = 0
7  loop:
8      DECFSZ 0x08, 1 ; MEM[0x08]--, pula para prox. inst.(break) se res. 0
9      GOTO continue
10 break:
11     GOTO fim
12 continue:
13     MOVF 0x08, 0; W = x
14     ADDWF 0x09, 0 ; W = W + Y
15     MOVWF 0x10 ; MEM[0x10] = W (neste caso variavel z)
16     MOVF 0x08, 0; W = x
17     ANDWF 0x09, 0 ; W = W & Y
18     MOVWF 0x11 ; MEM[0x11] = W (neste caso variavel q)
19     MOVF 0x10, 0; W = z
20     SUBWF 0x11, 0; W = W-q
21     MOVWF 0x12; MEM[0x12] = W (neste caso z-q que sera usada a seguir)
22     INCF 0x12,1; MEM[0x12]++(+1 unidade para tirar no DECFSZ, com um if)
23     DECFSZ 0x12, 1;tirar 1 de 0x12 se for 0 temos que sair
24     GOTO loop
25     GOTO fim
26 fim:
27     MOVLW 0 ; W = 0
28     END ; Pseudo inst. que termina o programa
```

### 3.3. Questão 3

A Tabela 1 abaixo contém as instruções utilizadas na implementação da Questão 2. O texto foi retirado integralmente do *datasheet* do PIC10F202 [Inc 2014]

Table 1. Tabela de Instruções		
Operands	Operation	Description
ADDWF f, d	$(W) + (f) \rightarrow (dest)$	Add the contents of the W register and register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.
ANDWF f, d	$(W) .AND. (f) \rightarrow (dest)$	"The contents of the W register are AND'ed with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'."
DECFSZ f, d	$(f) - 1 \rightarrow d$ ; skip if result = 0	"The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '0', the next instruction, which is already fetched, is discarded and a NOP is executed instead making it a two-cycle instruction."
MOVWF f	$(W) \rightarrow (f)$	"Move data from the W register to register 'f'."
MOVLW k	$k \rightarrow (W)$	"The eight-bit literal 'k' is loaded into the W register. The "don't cares" will assembled as '0's'."
GOTO k	$k \rightarrow PC<8:0>$ STATUS<6:5> $\rightarrow$ PC<10:9>	"GOTO is an unconditional branch. The 9-bit immediate value is loaded into PC bits <8:0>. The upper bits of PC are loaded from STATUS<6:5>. GOTO is a two cycle instruction."



### 3.4. Questão 4

Os sinais de controle do diagrama da questão 4 estão destacados em azul. Os sinais são os seguintes:

**Goto** Ativado para que o multiplexador escolha o endereçamento vindo dos bits 0-8 no lugar do endereço de PC+1

**Literal** Ativado para que o multiplexador escolha o endereçamento vindo dos bits 0-7 no lugar de endereço do FileRegister

**Skip** Ativado para que quando o alu ter o bit zero o código pular pular uma operação, usado também no multiplexador para enviar um no lugar do conteúdo do Work Register

**d** 0 para Work Register e 1 para File Register

**MemOp** Ativado para que o conteúdo salvo em F venha do resultado de uma operação no lugar de um literal

Abaixo, esta a tabela com os sinais de controle:

	Goto	Literal	Skip	d	MemOp
ADDWF	0	0	0	x	1
ANDWF	-	-	-	-	-
DECFSZ	0	0	1	x	1
MOVWF	0	0	x	1	1
MOVLW	0	1	x	0	x
GOTO	1	x	0	x	x

**Table 2. Tabela com Sinais de Controle**

### 3.5. Considerações Finais

Professor, o André quase teve um derrame, dá dois pontos pra ele não morrer pfv (Hennings, K)

### References

Inc, M. T. (2004-2014). Pic10f200/202/204/206 datasheet. Disponível em: <https://ww1.microchip.com/downloads/en/DeviceDoc/40001239F.pdf>. Acesso em: 30/06/2023.