



# **ELTD03z**

## **Microcontroladores/Microprocessadores**

### **Teoria\_04a1\_2b**

**Prof. Enio R. Ribeiro**

**Universidade Federal de Itajubá - UNIFEI**

## T4.6a) Adição com carry

Microcontrolador => registros = 1 byte => soma fatores de 1 byte => resultado: 2 bytes => somar com carry.

```

1
11101100
+ 10010001
-----
01111101
    
```

```

1
11101100
+ 10010001
-----
01111101
    
```

```

1
EC
+ 91
----
7D
    
```

```

1
00 EC
+ 00 91
-----
01 7D
    
```

Microcontrolador => registros = 1 byte => soma fatores de 1 e 2 (de 2) bytes => resultado: 3 bytes => somar com carry (soma múltipla precisão).

```

      ?      1
      01100010 11101100
+           10010001
-----
00000000 01100011 01111101
    
```

```

      ?      1
      00 62 EC
+ 00 00 91
-----
00 63 7D
    
```

## T4.6a) Adição com carry

Microcontrolador (STM32F103C8) => registros = 4 bytes (word) => soma fatores de 4 bytes => resultado 5 bytes => somar com carry.

1	1	1	1
E17C0235	00 E17C0235	0000 E17C0235	00000000 E17C0235
+ 9A124581	+ 00 9A124581	+ 0000 9A124581	+ 00000000 9A124581
-----	-----	-----	-----
7B8E47B6	01 7B8E47B6	0001 7B8E47B6	00000001 7B8E47B6

Microcontrolador (STM32F103C8) => registros = 4 bytes => soma fatores de 1 e 2 words => resultado 3 words => somar com carry (soma múltipla precisão).

1	?	1
F3A791D2 E17C0235	00 F3A791D2 E17C0235	
+ 9A124581	+ 9A124581	
-----	-----	
F3A791D3 7B8E47B6	00 F3A791D3 7B8E47B6	

## T4.6a) Adição com carry

### Instruction: **ADD**

The ADD instruction adds values of registers and writes the result to the destination register.

#### Syntax

ADD{S} Rd, Rn, Operand2 ; (operand2) = Rm

- '*Rd*' is the destination register
- '*Rn*' is the source register
- '*Operand2*' is a register for the second operand
- '*S*' is an optional suffix. If S is specified, the condition code flags are updated on the result of the operation

• Se usar o sufixo "S", o "Operand2" é, obrigatoriamente, um registro.

### Instruction: **ADC**

The ADC instruction adds with carry values of registers and writes the result to the destination register.

#### Syntax

ADC{S} Rd, Rn, Operand2 ; (operand2) = Rm

- '*Rd*' is the destination register
- '*Rn*' is the source register
- '*Operand2*' is a register for the second operand
- '*S*' is an optional suffix. If S is specified, the condition code flags are updated on the result of the operation

## T4.6b) Adição com carry

**Meta:** - Fazer programas para somar com carry (e ou múltipla precisão).

Ex.: 4.6.1 - Faça um programa para somar duas constantes: g e h. As constantes g e h podem representar valores entre:  $(0 \leq g, h \leq 255)$ . Guarde o resultado em local apropriado. Faça as designações e alocações necessárias.

Ex.: 4.6.2 - Faça um programa para somar duas constantes: g e h. As constantes g e h podem representar valores entre:  $(0 \leq g, h \leq 65535)$ . Guarde o resultado em local apropriado. Faça as designações e alocações necessárias.

Ex.: 4.6.3 - Faça um programa para somar duas constantes: g e h. As constantes g e h podem representar valores entre:  $(0 \leq g, h \leq 2^{32} - 1)$ . Guarde o resultado em local apropriado. Faça as designações e alocações necessárias.

Ex.: 4.6.4 - Faça um programa para somar duas constantes: g e h. As constantes g e h podem representar valores entre:  $(0 \leq g \leq 2^{32} - 1)$  e  $(0 \leq h \leq 2^{48} - 1)$ . Guarde o resultado em local apropriado. Faça as designações e alocações necessárias.

Ex.: 4.6.5 - Faça um programa para somar duas constantes: g e h. As constantes g e h podem representar valores entre:  $(0 \leq g \leq 2^{32} - 1)$  e  $(0 \leq h \leq 2^{64} - 1)$ . Guarde o resultado em local apropriado. Faça as designações e alocações necessárias.