

# Adições e Multiplicações em Binário

Yuri Kaszubowski Lopes  
Éverlin Fighera Costa Marques

UDESC

Anotações

## Revisão

Converta entre as bases

Binário	Octal	Decimal	Hexadecimal
1011011 <sub>2</sub>			
	74 <sub>8</sub>		
		37 <sub>10</sub>	
			A4 <sub>16</sub>

Anotações

## Adições em binário

- O processo é o mesmo que na base decimal
- Considere a soma de dois valores binários de 8 bits cada

111 11

0011 1001

+ 0001 1011

0101 0100

- Os "vai um" são chamados de bits de carry

Anotações

Adições em binário: Exemplos

1 1 1 1  
0 1 1 1 0 0 0 1  
+ 1 1 1 1 1 0 0 0  
-----  
1 0 1 1 0 1 0 0 1

- No exemplo os operandos possuem 8 bits, e o resultado possui 9.
  - Nono bit gerado devido ao carry final
  - No papel isto não é um problema
  - Para a máquina isso pode ser um problema sério
    - Overflow: resultado muito grande para a quantidade de dígitos disponíveis
    - Se armazenamos os valores na memória em duas variáveis de 8 bits cada (ex.: um char em C) , e realizamos a conta de forma que o resultado seja armazenado em outra variável de 8 bits

```
1 char v1, v2, resultado;  
2 ...  
3 resultado = v1 + v2;
```

- O bit extra vai ser desconsiderado por não caber na região de memória
  - O resultado será incorreto

Anotações

---

---

---

---

---

---

---

---

Overflow

```
1 #include<stdio.h>  
2 #include<stdlib.h>  
3  
4 int main() {  
5     unsigned char v1 = 0b01110001;  
6     unsigned char v2 = 0b11111000;  
7     unsigned char resultado_char;  
8     unsigned short resultado_short;  
9  
10    resultado_char = v1+v2;  
11    resultado_short = v1 + v2;  
12  
13    printf("%u %u\n", resultado_char, resultado_short);  
14  
15    return 0;  
16 }
```

Saída:

105 361

Anotações

---

---

---

---

---

---

---

---

Multiplicações

- Exemplo em decimal

1 2 3<sub>10</sub> Multiplicando  
× 3 2 1<sub>10</sub> Multiplicador  
-----  
1 2 3  
2 4 6 Desloca uma casa para a esquerda  
3 6 9 Desloca mais uma casa para a esquerda  
-----  
1  
3 9 4 8 3 Produto

- Produto (muito) maior que Multiplicando e Multiplicador
- Se multiplicarmos dois números com *m* e *n* casas (dígitos) teremos um resultado que pode ocupar até *m + n* casas (dígitos)

Anotações

---

---

---

---

---

---

---

---

Deslocamento para a esquerda

- Qual o motivo?

$22_{10} \times 23_{10}$   
 $22 \times 23$   
 $22 \times (3 + 20)$   
 $22 \times (3 + (2 \times 10))$   
 $22 \times 3 + 22 \times 2 \times 10$   
 $22 \times 3 + 22 \times 2 \times 10$

$$\begin{array}{r} 22_{10} \\ \times 23_{10} \\ \hline 66 \\ 440 \\ \hline \end{array}$$

Anotações

---

---

---

---

---

---

---

Em binário

- O raciocínio em binário (ou em qualquer outra base) é o mesmo
  - Realizar as multiplicações, deslocar os bits, e depois somar os resultados intermediários
- Esse algoritmo é o usado em nossas máquinas
- O hardware implementa esse algoritmo
- Em hardwares mais sofisticados
  - As multiplicações dos bits individuais é feita em paralelo para economizar tempo
  - Mas o algoritmo continua sendo o mesmo

Anotações

---

---

---

---

---

---

---

Exemplo em binário

$$\begin{array}{r} 11101_2 \\ \times 11110_2 \\ \hline 00000 \\ 11101 \\ 11101 \\ 11101 \\ 11101 \\ \hline 1101100110_2 \end{array}$$

Anotações

---

---

---

---

---

---

---

## Observações

- Alguns processadores, principalmente para sistemas embarcados (e.g. PIC16F628a), não possuem instruções de multiplicação via hardware implementadas
  - ▶ Se você precisar multiplicar, precisa criar um programa (função) que executa o algoritmo que discutimos
- Realizar subtrações e divisões em binário seguem raciocínios semelhantes
  - ▶ Implementamos os mesmos métodos utilizados na base 10

### Anotações

---

---

---

---

---

---

---

## Exemplos/Exercícios

- 1 Realize as seguintes adições, considerando que os valores ocupam 8 bits. Marque as operações que levam a overflows
  - 1 00001000<sub>2</sub> + 00000001<sub>2</sub>
  - 2 01100100<sub>2</sub> + 01111001<sub>2</sub>
  - 3 11111111<sub>2</sub> + 00000001<sub>2</sub>
  - 4 11111111<sub>2</sub> + 11111111<sub>2</sub>
  - 5 156<sub>8</sub> + 271<sub>8</sub>
- 2 Realize as seguintes multiplicações:
  - 1 1111<sub>2</sub> × 1100<sub>2</sub>
  - 2 11100<sub>2</sub> × 111101<sub>2</sub>
  - 3 123<sub>8</sub> × 121<sub>8</sub>
  - 4 F11<sub>16</sub> × 1A1<sub>16</sub>

### Anotações

---

---

---

---

---

---

---

## Referências

- TOCCI, R.J.; WIDMER, N.S. **Sistemas digitais: princípios e aplicações**. 11a ed, Prentice-Hall, 2011.
- RUGGIERO, M.; LOPES, V. da R. **Cálculo numérico: aspectos teóricos e computacionais**. Makron Books do Brasil, 1996.
- NULL, L.; LOBUR, J. **Princípios Básicos de Arquitetura e Organização de Computadores**. 2014. Bookman, 2009. ISBN 9788577807666.

### Anotações

---

---

---

---

---

---

---