

# Operadores binários (*Bitwise operations*)

Prof. Dr. Edson J. R. Justino

Prof. Dr. Alcides Calsavara

---

# OPERADORES

## I. Lógicos

1. AND: &
2. OR: |
3. XOR (exclusive or): ^
4. COMPLEMENTO DE UM (NOT): ~

## II. Deslocamento de bits

1. deslocamento à direita (right shift): >>
2. deslocamento à esquerda (left shift): <<

# Operadores Lógicos em Binário

(1) AND: &

int	a = 0x023F;	00000010	00111111	↓
int	b = 0xF3AC ;	11110011	10101100	
int c;				
	c = a & b	00000010	00101100	(0x022C)

## Exercícios

- 1) 0x123A & 0xEA
- 2) 0x9C13 & 0x3279
- 3) 0xB500 & 0xAC4F
- 4) 0xFFFF & 0x0100

# Operadores Lógicos em Binário

(2) OR: |

int	a = 0x023F;	00000010	00111111	↓
int	b = 0xF3AC ;	11110011	10101100	
int c;				
	c = a   b ;	11110011	10111111	(0xF3BF)

## Exercícios

- 1) 0x368 & 0xC
- 2) 0xBA42 & 0x89FF
- 3) 0x3579 & 0xF01B
- 4) 0x75CD & 0x0100

# Operadores Lógicos em Binário

(2) XOR: ^


int	a = 0x023F;	00000010	00111111	↓
int	b = 0xF3AC ;	11110011	10101100	
int c;				
	c = a ^ b ;	11110001	10010011	(0xF193)

## Exercícios

- 1) 0x368 & 0xC
- 2) 0xBA42 & 0x89FF
- 3) 0x3579 & 0xF01B
- 4) 0x75CD & 0x0100

# Operadores Lógicos em Binário

(4) Complemento de um: ~

int	a = 0x023F;	00000010 00111111	
int	b = 0xF3AC ;	11110011 10101100	
int c;			
	c = ~a ;	11111101 11000000 (0xFDC0)	
	c = ~b ;	00001100 01010011 (0x0C53)	

## Exercícios

- 1) ~0xC
- 2) ~0x89FF
- 3) ~0x3579
- 4) ~ 0x0100

# OPERADORES DE DESLOCAMENTO

---

# OPERADORES DE DESLOCAMENTO DE BITS

>> (Direita) - Desloca os bits para a direita com inclusão de zeros a esquerda e eliminando os elementos da direita.

Ex.:

```
int a = 0xF66F;    11010110    01101111
Int b;
                    ──────────>
    b = a >> 4;    00001101    01100110 (0x0D66)
    a = a >> 14;   00000000    00000011 (0x0003)
```

Exercícios:

1) C = 0x45FC      C = C >> 1;      C = C >> 8      C = C >> 16



# OPERADORES DE DESLOCAMENTO DE BITS

**<< (Esquerda)** - Desloca os bits para a esquerda com inclusão de zeros a direita e eliminando os elementos da esquerda

Ex.:

int a = 0xF66F;      11010110      01101111

Int b;

b = a << 2;      01011001 — 10111100 (0x59BC)

a = a << 8;      01101111      00000000 (0x6F00)

Exercícios:

1) C = 0x45FC

C = C << 1;

C = C << 8

C = C << 16

# OPERADORES DE DESLOCAMENTO DE BITS

- O deslocamento permite a multiplicação e a divisão do número por potência de 2.
  - $a = a * 16$                        $a = a << 4$                        $(16 = 2^{**}4)$
  - $a = a / 4$                                $a = a >> 2$                        $(4 = 2^{**}2)$

BITWISE OPERATORS

EXEMPLOS

Função para exibir os bits de um char (um byte)

```
void showbitschar(char ch)
{
    for (int i = (sizeof(char) * 8) - 1; i >= 0; i--)
    {
        putchar(ch & (1u << i) ? '1' : '0');
    }
    printf("\n");
}
```

```
int main()
{
    for (char ch = 'A'; ch < 'z'; ch++)
    {
        printf("%c\t%d\t", ch, ch);
        showbitschar(ch);
    }

    return 0;
}
```

Função para exibir os bits de um int (quatro bytes)

```
void showbitsint( int x )
{
    int i=0;
    for (i = (sizeof(int) * 8) - 1; i >= 0; i--)
    {
        putchar(x & (1u << i) ? '1' : '0');
    }
    printf("\n");
}
```

```
int main()
{
    for (int i = 0; i < 32; i++)
    {
        int n = pow(2, i);
        showbitsint(n);
    }

    return 0;
}
```

```
void show(int x)
{
    printf("%20d: ", x); showbitsint(x);
    putchar('\n');
}
```

bitwise\_2.c

#include <limits.h>

```
show(0);
show(1);
show(2);
show(-1);
show(-2);
```

```
puts("MAIOR INTEIRO:");
show(INT_MAX);
show(pow(2, 31) - 1);
```

```
puts("MENOR INTEIRO:");
show(INT_MIN);
show(-pow(2, 31));
```



## Representação de valores inteiros usando 4 bits

4 bits → 16 combinações → 16 valores

	bit de sinal				valor decimal
1	0	1	1	1	7
2	0	1	1	0	6
3	0	1	0	1	5
4	0	1	0	0	4
5	0	0	1	1	3
6	0	0	1	0	2
7	0	0	0	1	1
8	0	0	0	0	0
9	1	1	1	1	-1
10	1	1	1	0	-2
11	1	1	0	1	-3
12	1	1	0	0	-4
13	1	0	1	1	-5
14	1	0	1	0	-6
15	1	0	0	1	-7
16	1	0	0	0	-8

~

+1

Números Negativos:

Notação Complemento de 2

```
int x = 7;  
  
int y = ~x + 1;  
  
printf("%d\n", y);
```

bitwise\_3.c

## EXERCÍCIO PROPOSTO

Desenvolver um programa em linguagem C que simule um dispositivo de monitoramento de sinais vitais em uma UTI.

O programa deverá ler seguidamente um byte de um dispositivo de entrada (um micro-controlador simulado via teclado).

Especificação de cada bit do byte recebido do micro-controlador:

- 0 – indica que o dado no barramento de entrada é válido (0 válido / 1 inválido)
- 1 e 2 – indica o dispositivo de leitura (4 dispositivos)
- 3 – indica a classe do dispositivo (classe A ou B)
- 4 a 7 – A escala de valores ( 16 valores)

O programa deverá apresentar os valores de entrada em uma tela seguidamente.