

# Introdução à Ciência da Computação: operações sobre dados

## Parte 1: Lógicas, Máscaras e Deslocamentos

Prof. Danilo Medeiros Eler  
danilo.eler@unesp.br

# Conteúdo

- Operações Sobre Dados
  - Lógicas
  - Máscaras
  - Deslocamentos
  - Aritmética

# Operações Lógicas

- São aquelas que aplicam a mesma operação básica sobre os bits de um padrão
- As operações podem ser efetuadas
  - No nível dos bits e
  - No nível dos padrões

# Operações Lógicas

- NOT
- AND
- OR
- XOR

# Operações Lógicas – NOT

- A operação NOT inverte o valor do bit
  - $\text{NOT}(0) = 1$
  - $\text{NOT}(1) = 0$

| X | NOT(X) |
|---|--------|
| 0 |        |
| 1 |        |

# Operações Lógicas – NOT

- A operação NOT inverte o valor do bit
  - $\text{NOT}(0) = 1$
  - $\text{NOT}(1) = 0$

| X | NOT(X) |
|---|--------|
| 0 | 1      |
| 1 | 0      |

# Operações Lógicas – AND

- A operação AND resulta em um bit de valor 1 somente se os dois bits da operação forem 1

# Operações Lógicas – AND

- A operação AND resulta em um bit de valor 1 somente se os dois bits da operação forem 1

| A | B | AND(A,B) |
|---|---|----------|
| 0 | 0 |          |
| 0 | 1 |          |
| 1 | 0 |          |
| 1 | 1 |          |



# Operações Lógicas – AND

- A operação AND resulta em um bit de valor 1 somente se os dois bits da operação forem 1

| A | B | AND(A,B) |
|---|---|----------|
| 0 | 0 | 0        |
| 0 | 1 | 0        |
| 1 | 0 | 0        |
| 1 | 1 | 1        |

# Operações Lógicas – OR

- A operação OR resulta em um bit de valor 1 sempre que pelo menos um dos bits da operação for 1

| A | B | OR(A,B) |
|---|---|---------|
| 0 | 0 |         |
| 0 | 1 |         |
| 1 | 0 |         |
| 1 | 1 |         |

# Operações Lógicas – OR

- A operação OR resulta em um bit de valor 1 sempre que pelo menos um dos bits da operação for 1

| A | B | OR(A,B) |
|---|---|---------|
| 0 | 0 | 0       |
| 0 | 1 | 1       |
| 1 | 0 | 1       |
| 1 | 1 | 1       |

# Operações Lógicas – XOR

- A operação XOR (ou exclusivo) resulta em um bit de valor 1 sempre os bits da operação tiverem valores diferentes

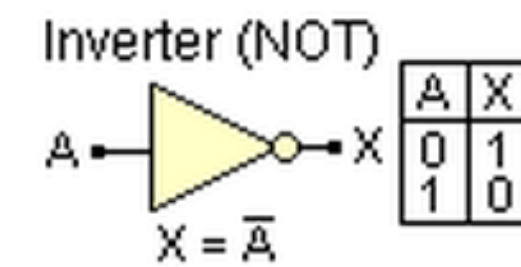
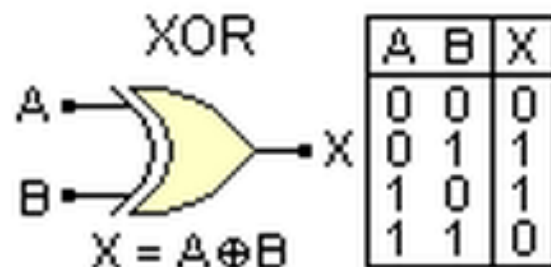
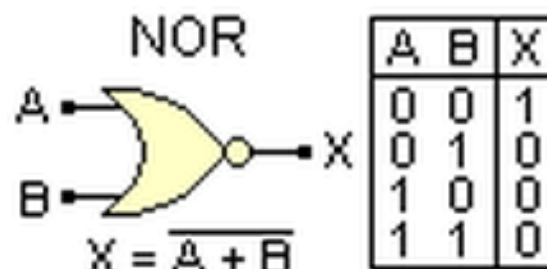
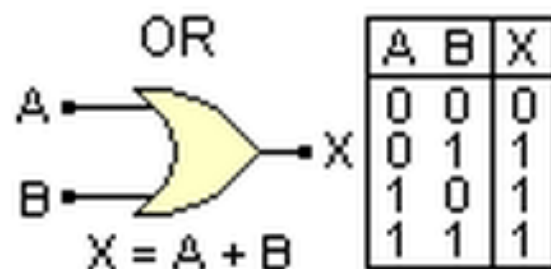
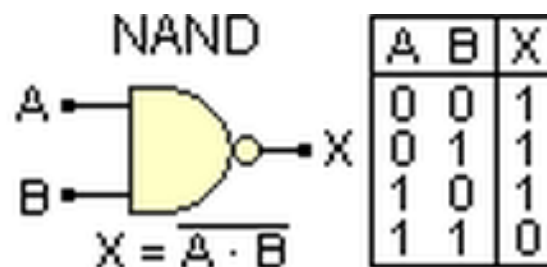
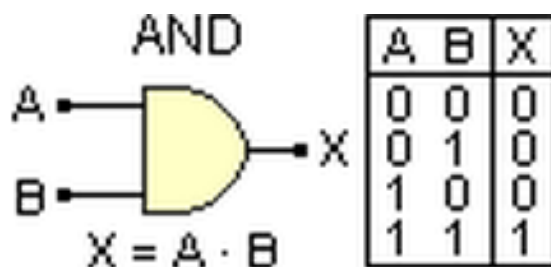
| A | B | XOR(A,B) |
|---|---|----------|
| 0 | 0 |          |
| 0 | 1 |          |
| 1 | 0 |          |
| 1 | 1 |          |

# Operações Lógicas – XOR

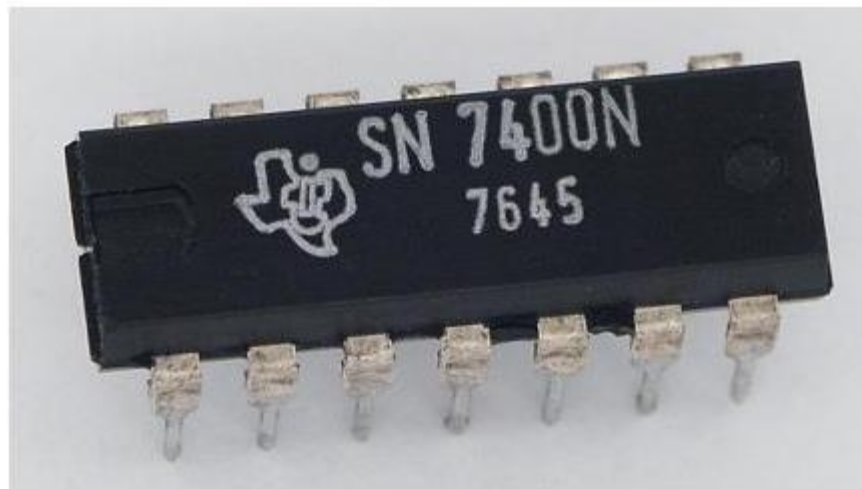
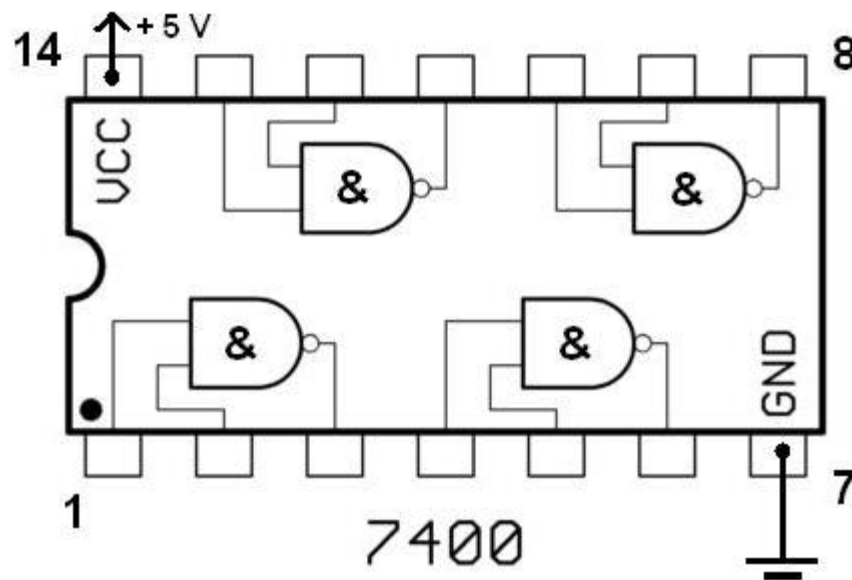
- A operação XOR (ou exclusivo) resulta em um bit de valor 1 sempre os bits da operação tiverem valores diferentes

| A | B | XOR(A,B) |
|---|---|----------|
| 0 | 0 | 0        |
| 0 | 1 | 1        |
| 1 | 0 | 1        |
| 1 | 1 | 0        |

# Portas Lógicas



# Portas Lógicas



# Operações no Nível dos Padrões

n bits



NOT



n bits

Entrada 1 0 0 1 1 0 0 0

Saída



# Operações no Nível dos Padrões

n bits



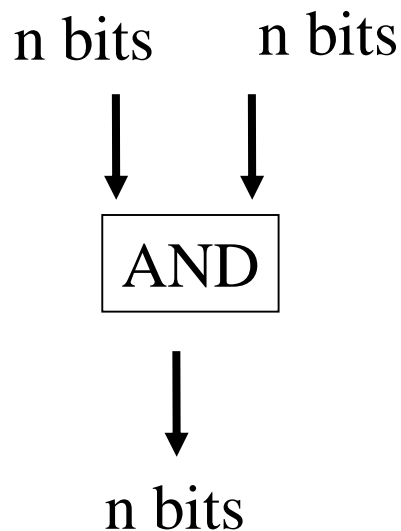
NOT



n bits

|         |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|
| Entrada | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Saída   | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

# Operações no Nível dos Padrões

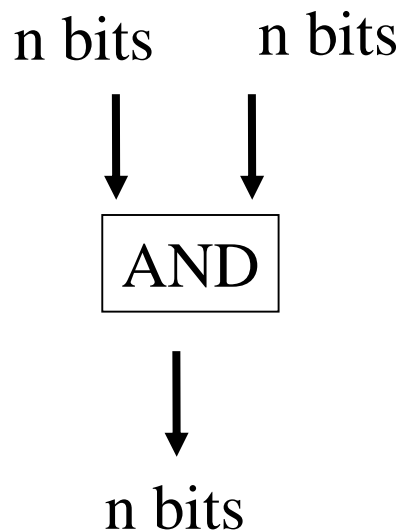


Entrada1 1 0 0 1 1 0 0 0

Entrada2 0 0 1 0 1 0 1 0

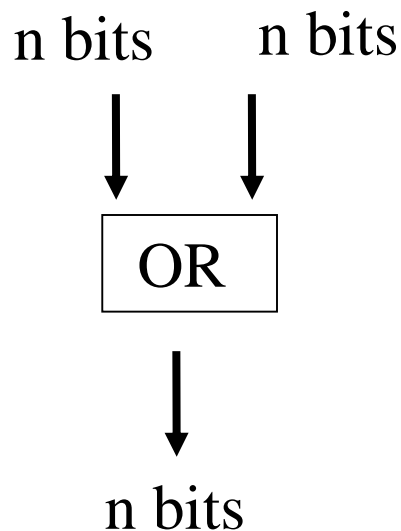
Saída

# Operações no Nível dos Padrões



|          |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|
| Entrada1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Entrada2 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| <hr/>    |   |   |   |   |   |   |   |   |
| Saída    | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

# Operações no Nível dos Padrões

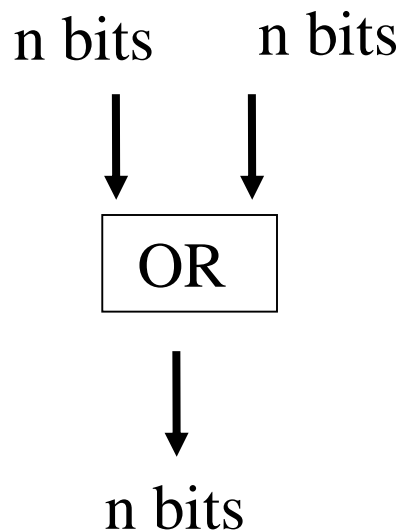


Entrada1 1 0 0 1 1 0 0 0

Entrada2 0 0 1 0 1 0 1 0

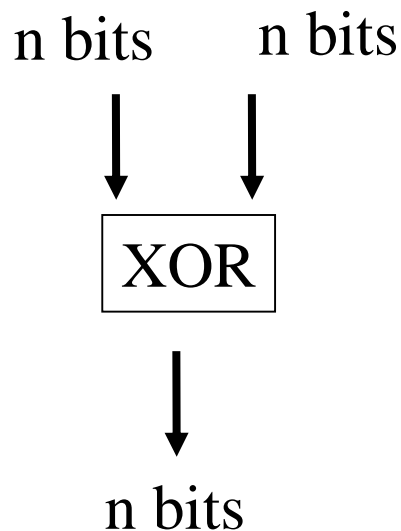
Saída

# Operações no Nível dos Padrões



|          |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|
| Entrada1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Entrada2 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| <hr/>    |   |   |   |   |   |   |   |   |
| Saída    | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |

# Operações no Nível dos Padrões



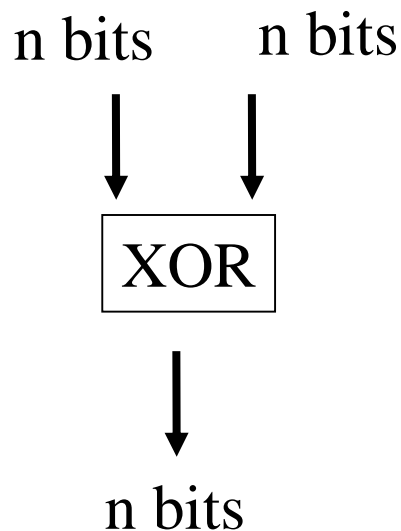
Entrada1 1 0 0 1 1 0 0 0

Entrada2 0 0 1 0 1 0 1 0

---

Saída

# Operações no Nível dos Padrões



Entrada1 1 0 0 1 1 0 0 0

Entrada2 0 0 1 0 1 0 1 0

---

Saída 1 0 1 1 0 0 1 0

# Máscaras



# Limpando bits específicos

- O operador AND pode ser utilizado para **limpar** bits específicos
  - Forçar para 0
- Nesse caso, o segundo padrão binário é chamado de **máscara**
  - Ex.: 00000111
    - Essa máscara resulta em um padrão binário que considera somente os três bits da direita

# Limpando bits específicos

- Exemplo de aplicação da máscara 00000111 com a operação AND

|          |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|
| Entrada1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| Máscara  | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| <hr/>    |   |   |   |   |   |   |   |   |
| Saída    |   |   |   |   |   |   |   |   |

# Limpando bits específicos

- Exemplo de aplicação da máscara 00000111 com a operação AND

|          |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|
| Entrada1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| Máscara  | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Saída    | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

- Os cinco bits da esquerda foram desconsiderados (zerados)

# Marcando bits específicos

- O operador OR pode ser utilizado para **marcar** bits específicos
  - Isto é, forçar para 1
- Mais uma vez podemos utilizar um segundo padrão binário como **máscara**
  - Ex.: 11111000
    - Essa máscara forçará que cinco os bits mais à esquerda sejam 1 e os três bits da direita são copiados

# Marcando bits específicos

- Exemplo de aplicação da máscara 11111000 com o operador OR

Entrada 1    1 0 1 0 0 1 1 0

Máscara    1 1 1 1 1 0 0 0

---

Saída

# Marcando bits específicos

- Exemplo de aplicação da máscara 11111000 com o operador OR

|          |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|
| Entrada1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| Máscara  | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Saída    | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

- Os cinco bits mais à esquerda são forçados a armazenar o valor 1 e os três bits da direita são copiados

# Invertendo bits específicos

- Uma das aplicações do **XOR** é a **inversão** de bits específicos
  - Ou seja, complemento
- Podemos utilizar um segundo padrão binário como **máscara**
  - Ex.: 11111000
    - Essa máscara fará com que os cinco bits mais à esquerda tenham os seus valores invertidos, enquanto que os três bits mais à direita serão copiados

# Invertendo bits específicos

- Exemplo de aplicação da máscara 11111000 com o operador XOR

|          |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|
| Entrada1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| Máscara  | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| <hr/>    |   |   |   |   |   |   |   |   |
| Saída    |   |   |   |   |   |   |   |   |



# Invertendo bits específicos

- Exemplo de aplicação da máscara 11111000 com o operador XOR

|          |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|
| Entrada1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| Máscara  | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Saída    | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |

- Os cinco bits da esquerda tiveram seus valores invertidos e os três bits da direita foram apenas copiados

# Operações de Deslocamento

# Operações de Deslocamento

- Movem bits em um padrão binário, modificando suas posições
- Podem mover bits para a direita ou esquerda
- Deslocamento à esquerda
  - Desloca os bits à esquerda e adiciona um 0

|   |                 |                     |
|---|-----------------|---------------------|
| ← | 1 0 0 1 1 0 0 0 | Original            |
|   | 0 0 1 1 0 0 0 0 | Após o deslocamento |

# Operações de Deslocamento

- Movem bits em um padrão binário, modificando suas posições
- Podem mover bits para a direita ou esquerda
- Deslocamento à direita
  - Desloca os bits à direita e adiciona um 0

|   |                 |                     |
|---|-----------------|---------------------|
| → | 1 0 0 1 1 0 0 0 | Original            |
|   | 0 1 0 0 1 1 0 0 | Após o deslocamento |

# Operações de Deslocamento

- A operação de deslocamento à esquerda pode fazer a multiplicação por dois

← 0 0 0 0 0 0 1 1    Original    = 3 em decimal

# Operações de Deslocamento

- A operação de deslocamento à esquerda pode fazer a multiplicação por dois

← 0 0 0 0 0 0 1 1    Original        = 3 em decimal  
     0 0 0 0 0 1 1 0    Deslocamento = 6 em decimal

# Operações de Deslocamento

- A operação de deslocamento à esquerda pode fazer a multiplicação por dois

|   |                 |                              |                |
|---|-----------------|------------------------------|----------------|
| ← | 0 0 0 0 0 0 1 1 | Original                     | = 3 em decimal |
|   | 0 0 0 0 0 1 1 0 | Deslocamento = 6 em decimal  |                |
|   | 0 0 0 0 1 1 0 0 | Deslocamento = 12 em decimal |                |

# Operações de Deslocamento

- A operação de deslocamento à esquerda pode fazer a multiplicação por dois

|   |                 |                              |                |
|---|-----------------|------------------------------|----------------|
| ← | 0 0 0 0 0 0 1 1 | Original                     | = 3 em decimal |
|   | 0 0 0 0 0 1 1 0 | Deslocamento = 6 em decimal  |                |
|   | 0 0 0 0 1 1 0 0 | Deslocamento = 12 em decimal |                |
|   | 0 0 0 1 1 0 0 0 | Deslocamento = 24 em decimal |                |



# Operações de Deslocamento

- A operação de deslocamento à direita pode fazer a divisão por dois

→ 0 0 0 1 1 0 0 0      Original      = 24 em decimal

# Operações de Deslocamento

- A operação de deslocamento à direita pode fazer a divisão por dois

→ 0 0 0 1 1 0 0 0      Original      = 24 em decimal  
0 0 0 0 1 1 0 0      Deslocamento = 12 em decimal

# Operações de Deslocamento

- A operação de deslocamento à direita pode fazer a divisão por dois

|   |                 |              |                 |
|---|-----------------|--------------|-----------------|
| → | 0 0 0 1 1 0 0 0 | Original     | = 24 em decimal |
|   | 0 0 0 0 1 1 0 0 | Deslocamento | = 12 em decimal |
|   | 0 0 0 0 0 1 1 0 | Deslocamento | = 6 em decimal  |

# Operações de Deslocamento

- A operação de deslocamento à direita pode fazer a divisão por dois

|   |                 |              |                 |
|---|-----------------|--------------|-----------------|
| → | 0 0 0 1 1 0 0 0 | Original     | = 24 em decimal |
|   | 0 0 0 0 1 1 0 0 | Deslocamento | = 12 em decimal |
|   | 0 0 0 0 0 1 1 0 | Deslocamento | = 6 em decimal  |
|   | 0 0 0 0 0 0 1 1 | Deslocamento | = 3 em decimal  |

# Operações de Deslocamento

- Deslocamento circular à direita

→ 1 0 0 1 1 0 0 1    Original  
    1 1 0 0 1 1 0 0    Após o deslocamento

- Deslocamento circular à esquerda

← 1 0 0 1 1 0 0 0    Original  
    0 0 1 1 0 0 0 1    Após o deslocamento

# Combinação de Operações

# Combinação de operações

- Podemos combinar operações lógicas, máscaras e deslocamentos para manipular os padrões binários
- Por exemplo, podemos deslocar um bit ou um conjunto de bits até uma posição específica do padrão e então aplicar uma máscara

# Combinação de operações

- Exemplo para obter o valor do terceiro e quarto bits de um padrão binário

1 0 0 1 **1** 0 0 0      Original



# Combinação de operações

- Exemplo para obter o valor do terceiro e quarto bits de um padrão binário

1 0 0 1 **1** 0 0 0      Original

0 1 0 0 1 **1** 0 0      Um deslocamento para a direita

# Combinação de operações

- Exemplo para obter o valor do terceiro e quarto bits de um padrão binário

|                        |                                   |
|------------------------|-----------------------------------|
| 1 0 0 1 <b>1</b> 0 0 0 | Original                          |
| 0 1 0 0 1 <b>1</b> 0 0 | Um deslocamento para a direita    |
| 0 0 1 0 0 1 <b>1</b> 0 | Dois deslocamentos para a direita |

# Combinação de operações

- Exemplo para obter o valor do terceiro e quarto bits de um padrão binário

|                 |                                   |
|-----------------|-----------------------------------|
| 1 0 0 1 1 0 0 0 | Original                          |
| 0 1 0 0 1 1 0 0 | Um deslocamento para a direita    |
| 0 0 1 0 0 1 1 0 | Dois deslocamentos para a direita |
| 0 0 0 0 0 0 1 1 | Máscara e Operação AND            |

# Combinação de operações

- Exemplo para obter o valor do terceiro e quarto bits de um padrão binário

|                 |                                   |
|-----------------|-----------------------------------|
| 1 0 0 1 1 0 0 0 | Original                          |
| 0 1 0 0 1 1 0 0 | Um deslocamento para a direita    |
| 0 0 1 0 0 1 1 0 | Dois deslocamentos para a direita |
| 0 0 0 0 0 0 1 1 | Máscara e Operação AND            |
| 0 0 0 0 0 0 1 0 | Resultado                         |

# Combinação de operações

- Essa estratégia pode ser utilizada para recuperar as intensidades de cor em padrões binários que armazenem pixels de imagens
- Deslocamentos e máscaras seriam utilizados para se obter os valores das intensidades de vermelho (R), verde (V) e azul (A)
  - Exemplo com 6 bits:

011110 → 01 11 10

# Combinação de operações

011110  $\rightarrow$  01 11 10

0 1 1 1 1 0    Original

# Combinação de operações

011110 → 01 11 10

0 1 1 1 1 0    Original

0 0 0 0 1 1    Máscara Blue e Operação AND

# Combinação de operações

011110 → 01 11 10

0 1 1 1 1 0    Original

0 0 0 0 1 1    Máscara Blue e Operação AND

0 0 0 0 1 0    Resultado

Valor da intensidade de azul é 2



# Combinação de operações

011110  $\rightarrow$  01 11 10

0 1 1 1 1 0    Original

# Combinação de operações

011110 → 01 11 10

0 1 1 1 1 0    Original

0 0 1 1 0 0    Máscara Green e Operação AND

# Combinação de operações

011110 → 01 11 10

0 1 1 1 1 0    Original

0 0 1 1 0 0    Máscara Green e Operação AND

0 0 1 1 0 0    Resultado da Máscara Green

# Combinação de operações

011110 → 01 11 10

0 1 1 1 1 0    Original

0 0 1 1 0 0    Máscara Green e Operação AND

0 0 1 1 0 0    Resultado da Máscara Green

0 0 0 1 1 0    Um deslocamento à direita

# Combinação de operações

011110 → 01 11 10

|             |                              |
|-------------|------------------------------|
| 0 1 1 1 1 0 | Original                     |
| 0 0 1 1 0 0 | Máscara Green e Operação AND |
| 0 0 1 1 0 0 | Resultado da Máscara Green   |
| 0 0 0 1 1 0 | Um deslocamento à direita    |
| 0 0 0 0 1 1 | Dois deslocamentos à direita |

# Combinação de operações

011110 → 01 11 10

0 1 1 1 1 0    Original

0 0 1 1 0 0    Máscara Green e Operação AND

0 0 1 1 0 0    Resultado da Máscara Green

0 0 0 1 1 0    Um deslocamento à direita

0 0 0 0 1 1    Dois deslocamentos à direita

0 0 0 0 1 1    Resultado

Valor da intensidade de verde é 3

# Combinação de operações

011110  $\rightarrow$  01 11 10

0 1 1 1 1 0    Original

# Combinação de operações

011110 → 01 11 10

0 1 1 1 1 0    Original

1 1 0 0 0 0    Máscara Red e Operação AND



# Combinação de operações

011110 → 01 11 10

0 1 1 1 1 0    Original

1 1 0 0 0 0    Máscara Red e Operação AND

0 1 0 0 0 0    Resultado da Máscara Red

# Combinação de operações

011110 → 01 11 10

|             |                                     |
|-------------|-------------------------------------|
| 0 1 1 1 1 0 | Original                            |
| 1 1 0 0 0 0 | Máscara Red e Operação AND          |
| 0 1 0 0 0 0 | Resultado da Máscara Red            |
| 1 0 0 0 0 0 | Um deslocamento circular à esquerda |

# Combinação de operações

011110 → 01 11 10

|             |  |
|-------------|--|
| 0 1 1 1 1 0 | Original                                 |
| 1 1 0 0 0 0 | Máscara Red e Operação AND               |
| 0 1 0 0 0 0 | Resultado da Máscara Red                 |
| 1 0 0 0 0 0 | Um deslocamento circular à esquerda      |
| 0 0 0 0 0 1 | Dois deslocamentos circulares à esquerda |

# Combinação de operações

011110 → 01 11 10

0 1 1 1 1 0    Original

1 1 0 0 0 0    Máscara Red e Operação AND

0 1 0 0 0 0    Resultado da Máscara Red

1 0 0 0 0 0    Um deslocamento circular à esquerda

0 0 0 0 0 1    Dois deslocamentos circulares à esquerda

0 0 0 0 0 1    Resultado

Valor da intensidade de Vermelho é 1

# Bibliografia

## BIBLIOGRAFIA BÁSICA

1. BROOKSHEAR, J. G. **Ciência da computação**: uma visão abrangente. 5ª ed., Bookman Editora, 2000. 499p.
2. FOROUZAN, B. A., MOSHARRAF, F. **Fundamentos da Ciência da Computação**. 2ª ed., São Paulo: Cengage Learning, 2011. 560p.

## BIBLIOGRAFIA COMPLEMENTAR

1. BROOKSHEAR, J. G. **Ciência da computação**: uma visão abrangente. 5ª ed., Bookman Editora, 2000. 499p.
2. CORMEN, T.H., Leiserson, C.E., Rivest R.L., Stein, C. **Algoritmos**: teoria e Prática. Rio de Janeiro: Editora Campus, 2002. 916p.
3. PLAUGER, P. L. **A Biblioteca Standard C**. Rio de Janeiro: Editora Campus, 1994. 614p.
4. PRATA, S. **C primer plus**, 4ª ed. SAMS Publishing, 2002. 931p.