# Sistemas Digitais

## Operações lógicas e funções básicas

Prof. Manoel Eusebio de Lima
Centro de Informática
Universidade Federal de Pernambuco

## Operações lógicas e funções básicas

- Funções binárias
  - $f(X) = f(x_1, x_2, \ldots\ldots, x_r)$, $x_i \, \varepsilon \, \{0, 1\}$
    f(X) é uma correspondência única que associa um valor que pode ser 0 ou 1 para cada um dos $2^r$ valores possíveis que o vetor $[x_1, \ldots\ldots, x_r]$ pode tomar.
- Representação de uma função escalar
  - tabela verdade

| $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

**Quantas combinações diferentes podem definidas com r = 3 ?**
**$2^3 = 8$**

- **Quando r é muito grande este tipo de representação torna-se impraticável**

## Operações lógicas

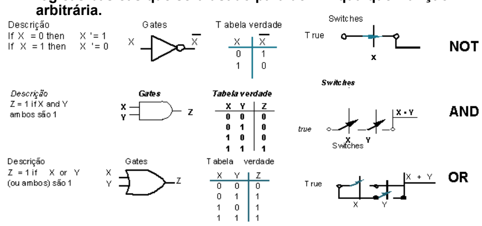
- Operadores lógicos são funções escalares usados para representarem o comportamento de um circuito lógico.
- Tipos de operadores
  - **Operadores Unários**
    - **Têm uma única variável**

| x | $f_1(x) = 0$ | $f_2(x) = x$ | $f_3(x) = \bar{x}$ | $f_4(x) = 1$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

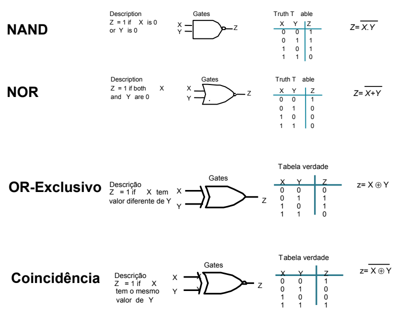
- **As funções $f_1(x) = 0$ e $f_4(x) = 1$ tomam valores constantes independentes de x**
- **A função $f_2(x) = x$ é chamada função identidade**
- **A função $f_3(x) = \bar{x}$ é chamada complementar, negação ou inversa. Esta função pode ser ainda representada por $x'$ e $\neg$ x**

## Operações lógicas

- **Operações binárias**
  - **Operações binárias são funções representadas por duas variáveis representadas na forma $f(x_1, x_2) = x_1$ <operador> $x_2$**
  - **Seis destas funções são muito usadas e têm nomes especiais. Junto com a função NOT formam o conjunto de operações lógicas básicas que será usado para definir qualquer função arbitrária.**

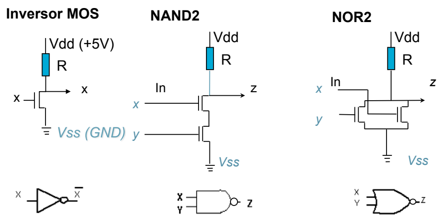



## Operações lógicas




## Operações lógicas

Tecnologia MOS (Metal Óxido Silício)




Vdd – positive supply
Vss - negative supply (GND)

## Funções compostas

- Se $f(x_1, x_2, \ldots\ldots, x_r)$, $g(x_1, x_2, \ldots\ldots, x_r)$ e $h(x_1, x_2, \ldots\ldots, x_r)$ são três funções, então h $[f(x_1, x_2, \ldots\ldots, x_r), g(x_1, x_2, \ldots\ldots, x_r)]$ também é uma função.
  - Exemplo:

$$f(x_1, x_2) = x_1 \oplus x_2$$
$$g(x_2, x_3) = \overline{x_2 + x_3}$$
$$h(f, g) = f + g \text{ (composta)}$$

↓ = nor
↑ = nand
∧ = and
∨ = or

então podemos definir

$$h(x_1, x_2, x_3) = x_1 \oplus x_2 + (\overline{x_2 + x_3})$$

- Avaliação de funções compostas (exemplo)

$f(x_1, x_2, x_3) = (\overline{x_1 \cdot x_2}) \oplus (\overline{x_2 + x_3})$ *com* $y_1 = \overline{x_1 \cdot x_2}$ e $y_2 = \overline{x_2 + x_3}$

*Logo* $f(x_1, x_2, x_3) = y_1 \oplus y_2$

*Para* $[x_1, x_2, x_3] = [1\ 1\ 0]$, $y_1 = \overline{1 \cdot 1} = 0$ e $y_2 = \overline{1 + 0} = 0$, *temos*

*portanto* $\boxed{f(x_1, x_2, x_3) = 0 \oplus 0 = 0}$

## Funções compostas

- Quando usamos operadores NAND, NOR ou OU-Exclusivo é melhor usar parênteses para separar operações lógicas.
- Quando se usa operadores AND, OR e NOT os parênteses podem ser eliminados desde que se use as seguintes regras de precedência:
  - Expressões sem parêntese
    - **Aplicar todas as instâncias da operação NOT às variáveis da esquerda para a direita.**
    - **Aplicar as instâncias da operação AND e em seguida as da operação OR sempre da esquerda para a direita.**
  - **Expressões com parênteses**
    - **Neste caso é feito uma avaliação dentro dos parênteses e o valor resultante de cada parêntese entra como uma variável nas futuras avaliações.**
  - **O operador NOT aplicado sobre uma expressão tem o efeito de colocar parênteses na expressão.**

## Uso de tabela verdade para avaliação de funções compostas

- $f(x_1,x_2,x_3)=(x_1 \wedge \overline{x}_2 \oplus x_3) \odot (x_1 \vee x_3)$

| | 1º | 2º | 3º | 4º | 5º |
|---|---|---|---|---|---|
| $x_1\ x_2\ x_3$ | $\overline{x}_2$ | $x_1 \wedge \overline{x}_2$ | $x_1 \wedge \overline{x}_2 \oplus x_3$ | $x_1 \vee x_3$ | $f(x_1, x_2, x_3)$ |
| 0 0 0 | 1 | 0 | 0 | 0 | 1 |
| 0 0 1 | 1 | 0 | 1 | 1 | 1 |
| 0 1 0 | 0 | 0 | 0 | 0 | 1 |
| 0 1 1 | 0 | 0 | 1 | 1 | 1 |
| 1 0 0 | 1 | 1 | 1 | 1 | 1 |
| 1 0 1 | 1 | 1 | 0 | 1 | 0 |
| 1 1 0 | 0 | 0 | 0 | 1 | 0 |
| 1 1 1 | 0 | 0 | 1 | 1 | 1 |

## Equivalência lógica de funções

- Duas funções $f(x_1, x_2, \ldots\ldots, x_r)$ e $g(x_1, x_2, \ldots\ldots, x_r)$ definidas para os mesmos argumentos são equivalentes logicamente se e somente se $f(x_1, x_2, \ldots\ldots, x_r) = g(x_1, x_2, \ldots\ldots, x_r)$ para todos as possíveis combinações da r-tupla $x_1, x_2, \ldots\ldots, x_r$
  - Exemplo: $f(x_1, x_2, x_3) = (x_1 \oplus x_2)(x_2 \vee x_3)$
    $g(x_1, x_2, x_3) = \overline{x}_1 x_2 \overline{x}_3 \vee \overline{x}_1 x_2 x_3 \vee x_1 \overline{x}_2 x_3$

| $x_1\ x_2\ x_3$ | $x_2 \oplus x_3$ | $x_2 \vee x_3$ | $f(x_1, x_2, x_3)$ | $\overline{x}_1 x_2 \overline{x}_3$ | $\overline{x}_1 x_2 x_3$ | $x_1 \overline{x}_2 x_3$ | $g(x_1, x_2, x_3)$ |
|---|---|---|---|---|---|---|---|
| 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 0 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 1 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 1 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 0 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 0 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 1 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 1 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

≡

## Forma canônica de funções escalares

- Nós podemos mostrar que qualquer tabela verdade pode ser representada por uma função escalar que envolve apenas operações de AND, OR e NOT.
- Suponha uma r-tupla $[x_1, x_2, \ldots\ldots, x_r]$
  - **Termo de produto**- é uma função que é definida como o AND de um conjunto de termos, onde cada termo é uma variável $x_j$ ou o seu complemento $\overline{x}_i$. Nenhuma variável pode aparecer mais que uma vez em um termo produto.
- Exemplo
  - $m_4 = x_1x_2x_3$
  - $m_5 = x_1\overline{x}_2x_3$
  - $m_7 = x_1\overline{x}_2\overline{x}_3$

## Forma canônica de funções escalares

**Minitermo** (ou minitermo –1 ) é um termo produto que possui um valor '1' para apenas uma das $2^r$ possíveis combinações da r-tupla $[x_1, x_2, \ldots\ldots, x_r]$ e zero para os outros valores.

| $x_1\ x_2\ x_3$ | $m_0=\overline{x1}\ \overline{x_2}\ \overline{x_3}$ | $m_1=\overline{x_1}\ \overline{x_2}\ x_3$ | $m_2=\overline{x_1}\ x_2\ \overline{x_3}$ | .................... | $m_6=x_1\ x_2\ \overline{x_3}$ | $m_7=x_1\ x_2\ x_3$ |
|---|---|---|---|---|---|---|
| 0 0 0 | 1 | 0 | 0 | | 0 | 0 |
| 0 0 1 | 0 | 1 | 0 | | 0 | 0 |
| 0 1 0 | 0 | 0 | 1 | | 0 | 0 |
| 0 1 1 | 0 | 0 | 0 | | 0 | 0 |
| 1 0 0 | 0 | 0 | 0 | | 0 | 0 |
| 1 0 1 | 0 | 0 | 0 | | 0 | 0 |
| 1 1 0 | 0 | 0 | 0 | | 1 | 0 |
| 1 1 1 | 0 | 0 | 0 | | 0 | 1 |

– **Observe que:**
- **•$m_4 = x_1\overline{x}_2\overline{x}_3$ porque $4_{10} = 100_2$**
- **•$m_5 = x_1\overline{x}_2x_3$ porque $5_{10} = 101_2$**

**.................................................**

**Qualquer função booleana pode ser expressa como uma soma (OR) destes minitermos**

- **Forma canônica soma de produtos ou forma normal disjuntiva**
  **$f(x_1, x_2, x_3) = m_0 + m_1 + m_4$ ou $\Sigma (m_0, m_1, m_4)$**

**Exemplo:**

| | $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 1 |
| $m_1$ | 0 | 0 | 1 | 1 |
| $m_2$ | 0 | 1 | 0 | 0 |
| $m_3$ | 0 | 1 | 1 | 0 |
| $m_4$ | 1 | 0 | 0 | 1 |
| $m_5$ | 1 | 0 | 1 | 0 |
| $m_6$ | 1 | 1 | 0 | 0 |
| $m_7$ | 1 | 1 | 1 | 0 |

**$f(x_1, x_2, x_3) = m_0 + m_1 + m_4$ ou $\Sigma (m_0, m_1, m_4)$**

# Forma canônica de funções escalares

- **Termo de soma** -é uma função que é definida como o OR de um conjunto de termos, onde cada termo é uma variável $x_j$ ou o seus complemento $\overline{x_i}$ . Nenhuma variável pode aparecer mais que uma vez no termo soma.

- **Exemplo:**
  - **$M_4 = \overline{x}_1 + x_2 + x_3$ porque $4_{10} = 100_2$**
  - **$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$ porque $5_{10} = 101_2$**
  - **$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$ porque $7_{10} = 111_2$**

# Forma canônica de funções escalares

**Maxtermo (ou Maxtermo – 0)** é um termo soma que possui um valor '0' para apenas uma das $2^r$ possíveis combinações da r-tupla $[x_1, x_2, \ldots\ldots, x_r]$ e '1' para todos os outros valores.

| $x_1\ x_2\ x_3$ | $M_0 = x_1 + x_2 + x_3$ | $M_1 = x_1 + x_2 + \overline{x}_3$ | $M_2 = x_1 + \overline{x}_2 + x_3$ | ................... | $M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$ |
|---|---|---|---|---|---|
| 0 0 0 | 0 | 1 | 1 | . | 1 |
| 0 0 1 | 1 | 0 | 1 | . | 1 |
| 0 1 0 | 1 | 1 | 0 | . | 1 |
| 0 1 1 | 1 | 1 | 1 | . | 1 |
| 1 0 0 | 1 | 1 | 1 | . | 1 |
| 1 0 1 | 1 | 1 | 1 | . | 1 |
| 1 1 0 | 1 | 1 | 1 | . | 1 |
| 1 1 1 | 1 | 1 | 1 | . | 0 |

Observe que:

- **$M_4 = \overline{x}_1 + x_2 + x_3$ porque $4_{10} = 100_2$**
- **$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$ porque $5_{10} = 101_2$**

Qualquer função booleana pode ser expressa como um produto (AND) destes Maxtermos

- **Forma canônica produto de somas (ORs) ou forma normal conjuntiva**
  **$f(x_1, x_2, x_3) = M_2.M_3.M_5.M_6.M_7$ ou $\Pi (M_2, M_3, M_5, M_6, M_7)$**

**Exemplo:**

| | $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|---|
| $M_0$ | 0 | 0 | 0 | 1 |
| $M_1$ | 0 | 0 | 1 | 1 |
| $M_2$ | 0 | 1 | 0 | 0 |
| $M_3$ | 0 | 1 | 1 | 0 |
| $M_4$ | 1 | 0 | 0 | 1 |
| $M_5$ | 1 | 0 | 1 | 0 |
| $M_6$ | 1 | 1 | 0 | 0 |
| $M_7$ | 1 | 1 | 1 | 0 |

**$f(x_1, x_2, x_3) = M_2.M_3.M_5.M_6.M_7$ ou $\Pi (M_2, M_3, M_5, M_6, M_7)$**

# Representação canônica de expressões booleanas

- Após o conceito de minitermos e maxitermos podemos usar estas funções como um conjunto de blocos básicos que representam qualquer função descrita por uma tabela verdade.

  - **tabela verdade**

| **termo** | **$x_1$** | **$x_2$** | **$x_3$** | **$f(x_1, x_2, x_3)$** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 |

**$f(x_1, x_2, x_3) = m_1 + m_3 + m_4 + m_7$**

**$f(x_1, x_2, x_3) = M_0.M_2.M_5.M_6$**

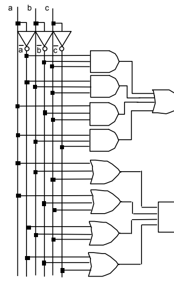
# Representação canônica de expressões booleanas

- **Forma canônica soma de produtos ou forma normal disjuntiva**
  **$f(x_1, x_2, x_3) = m_1 + m_3 + m_4 + m_7$ ou $\Sigma (m_1, m_3, m_4, m_7)$**
- **Forma canônica produto de somas ou forma normal conjuntiva**
  **$f(x_1, x_2, x_3) = M_0.M_2.M_5.M_6$ ou $\Pi (M_0, M_2, M_5, M_6)$**
- **Qualquer função de r variáveis pode ser representada em termos de NAND (NOT, AND) e NOR (OR, NOT)**
- **Para converter de uma forma canônica para outra troque o símbolo $\Sigma$ por $\Pi$ e liste os números que foram excluídos da forma original.**

# Exemplos de circuitos lógicos

- Implementar a função Z abaixo

| a | b | c | z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |




*Minitermos*

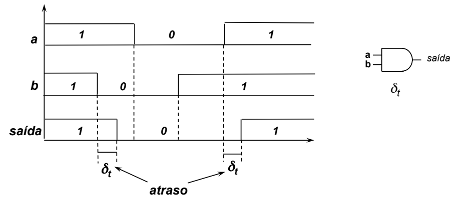
$z_1(a,b,c)= m_1+m_3+m_5+ m_6$

*Maxtermos*

$z_2(a,b,c)= M_0.M_2.M_4.M_7$

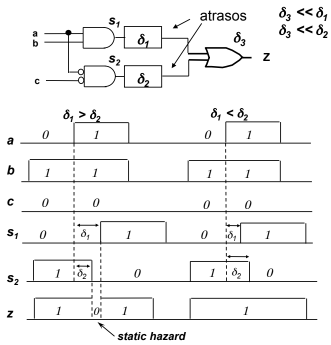
## *Comportamento dinâmico de circuitos digitais*

- Diagrama de tempo




• *A velocidade de operação da rede é determinada pelo tempo total que a porta lógica leva para alcançar um estado permanente*

## *Características elétricas de circuitos digitais*




# Exemplo:

*Desenvolver um circuito de alarme de um automóvel com as seguintes características funcionais:*
*O alarme/advertência deve ser acionado quando a ignição estiver acionada (carro ligado) e pelos menos uma das porta estiver aberta. Implementar o circuito resultante utilizando as duas formas formais conjuntivas (minitermos e maxtermos)*

*Obs: considere que o carro possui apenas duas portas.*