Introdução à Eletrônica Digital

Prof. Carlos Fernando Teodósio Soares

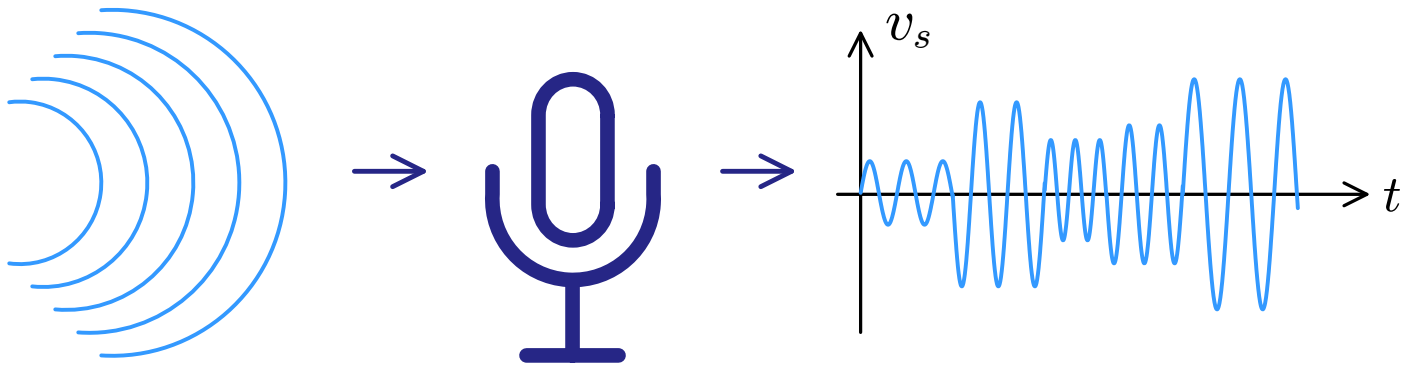
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA POLITÉCNICA
Departamento de Engenharia Eletrônica e de Computação

Agenda da Aula - Capítulo 01

- **Conceitos Introdutórios**
  - **Sinais e Circuitos Digitais**
  - Sistema de Numeração Binário
  - Álgebra Booleana
- Portas Lógicas Digitais
- Circuitos Lógicos Combinacionais
- Células de Memória Digitais

Analógico × Digital

- **Sinais Analógicos** podem assumir qualquer valor real ao longo do tempo. Em Eletrônica, os sinais analógicos são constituídos por uma tensão ou uma corrente elétrica, cujo valor a cada instante de tempo representa alguma grandeza medida (temperatura, pressão, velocidade, posição, etc.) por algum elemento transdutor. Assim, a forma de onda do sinal acaba sendo análoga à variação no tempo da grandeza medida.

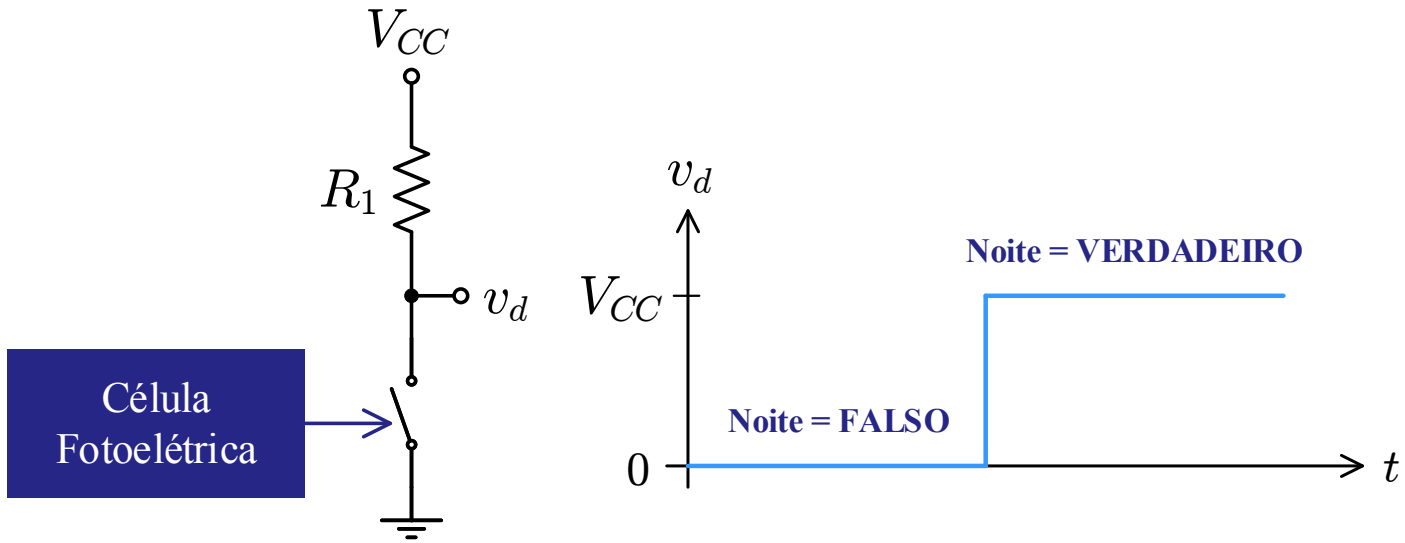


Exemplo – Sinal de Áudio

O sinal de áudio captado por um microfone (transdutor) é análogo à variação no tempo da pressão do ar causada pelas ondas sonoras incidentes.

Analógico × Digital

- **Sinais Digitais** podem assumir somente valores discretos. Em geral, as grandezas digitais assumem apenas valores binários (0 ou 1), aos quais podem ser atribuídos significados lógicos como FALSO ou VERDADEIRO.

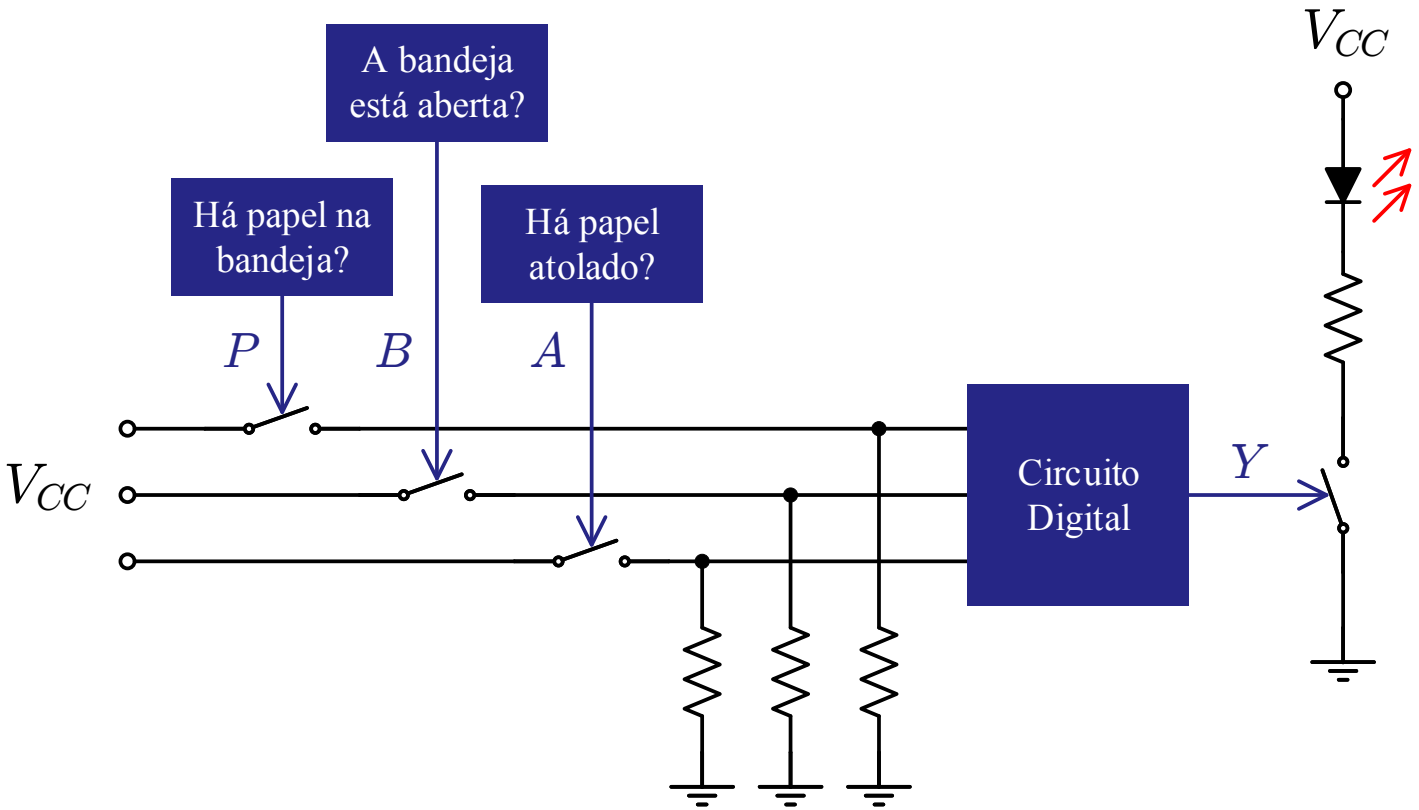


Exemplo – Controle de Iluminação Urbana

O acendimento de uma lâmpada de iluminação urbana é um exemplo de sinal digital. Durante o dia (Noite = FALSO), a célula fotoelétrica é sensibilizada pela luz do Sol e, então, realiza o fechamento da chave, produzindo $v_d = 0$. Já durante a noite (Noite = VERDADEIRO), a chave fica aberta, produzindo $v_d = 1$.

Circuitos Digitais

- **Circuitos Digitais** são circuitos eletrônicos que processam sinais digitais aplicados à sua entrada e produzem uma ou mais saídas digitais como resposta. A função lógica que relaciona as respostas do circuito com todas as possíveis combinações das entradas pode ser expressa na forma de uma Tabela Verdade.
- **EXEMPLO:** Circuito de alerta de uma impressora



Circuitos Digitais

- **Circuitos Digitais** são circuitos eletrônicos que processam sinais digitais aplicados à sua entrada e produzem uma ou mais saídas digitais como resposta. A função lógica que relaciona as respostas do circuito com todas as possíveis combinações das entradas pode ser expressa na forma de uma Tabela Verdade.
- **EXEMPLO:** Circuito de alerta de uma impressora

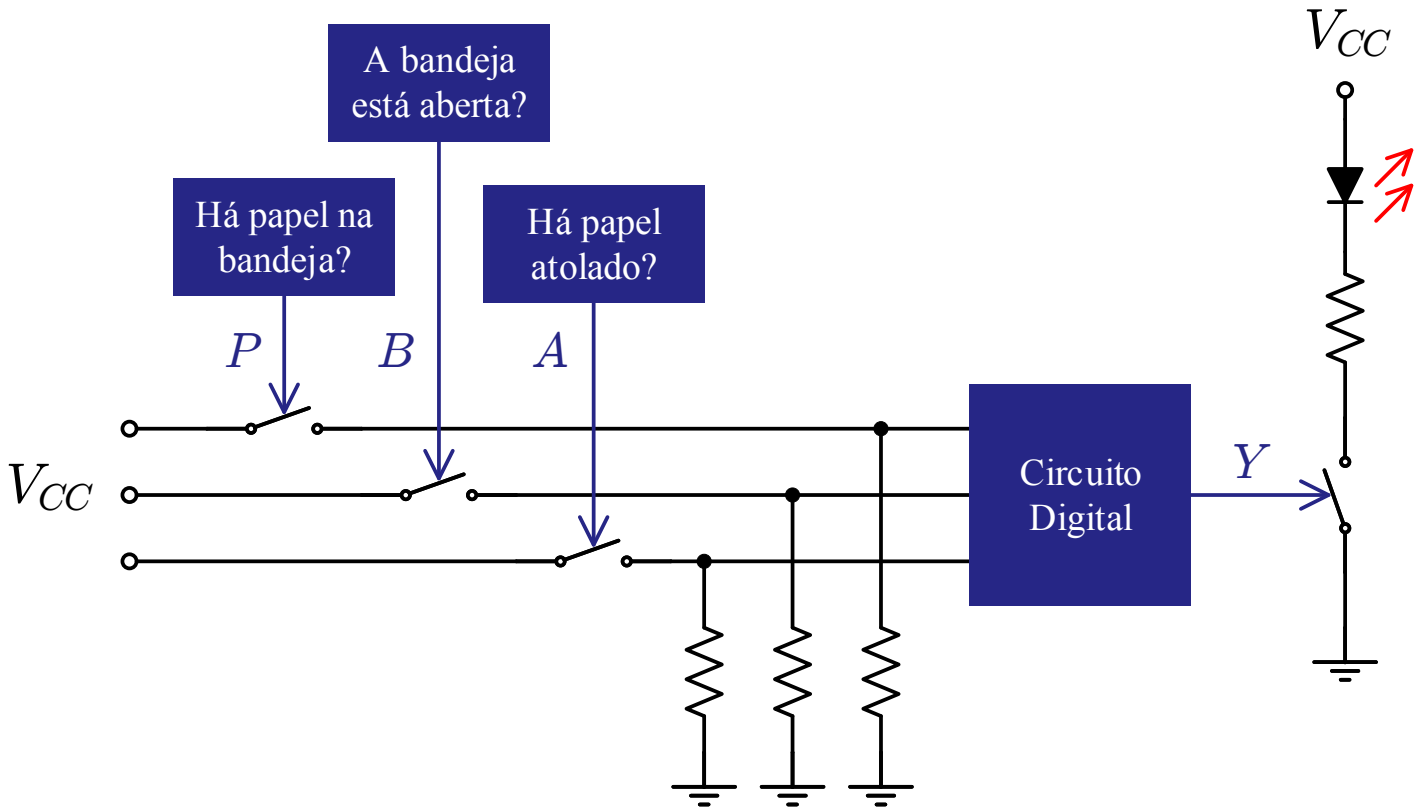


Tabela Verdade

P	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Agenda da Aula - Capítulo 02

- **Conceitos Introdutórios**
  - Sinais e Circuitos Digitais
  - **Sistema de Numeração Binário**
  - Álgebra Booleana
- Portas Lógicas Digitais
- Circuitos Lógicos Combinacionais
- Células de Memória Digitais

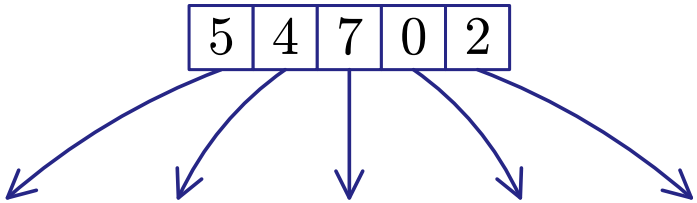
Sistema de Numeração Binário

- O sistema de numeração decimal é universalmente adotado por nós seres humanos para a representação de números. Nesse sistema, os números são representados através de dez símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9), onde a posição de cada símbolo no número possui um determinado peso.

$$5 \quad 4 \quad 7 \quad 0 \quad 2$$

Sistema de Numeração Binário

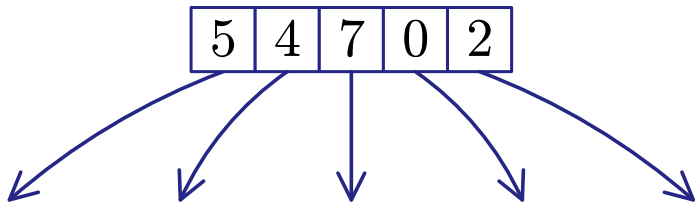
- O sistema de numeração decimal é universalmente adotado por nós seres humanos para a representação de números. Nesse sistema, os números são representados através de dez símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9), onde a posição de cada símbolo no número possui um determinado peso.



$$5 \cdot 10^4 + 4 \cdot 10^3 + 7 \cdot 10^2 + 0 \cdot 10^1 + 2 \cdot 10^0$$

Sistema de Numeração Binário

- O sistema de numeração decimal é universalmente adotado por nós seres humanos para a representação de números. Nesse sistema, os números são representados através de dez símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9), onde a posição de cada símbolo no número possui um determinado peso.

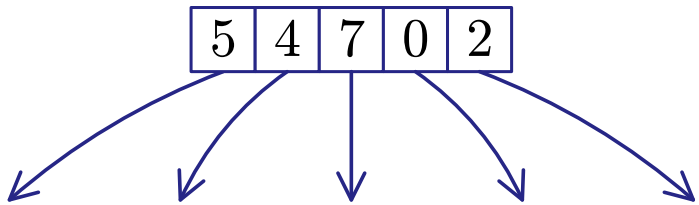


$$5 \cdot 10^4 + 4 \cdot 10^3 + 7 \cdot 10^2 + 0 \cdot 10^1 + 2 \cdot 10^0$$

- Como os circuitos digitais operam com sinais que somente podem assumir dois valores possíveis, os sistemas digitais adotam o sistema de numeração binário.

Sistema de Numeração Binário

- O sistema de numeração decimal é universalmente adotado por nós seres humanos para a representação de números. Nesse sistema, os números são representados através de dez símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9), onde a posição de cada símbolo no número possui um determinado peso.



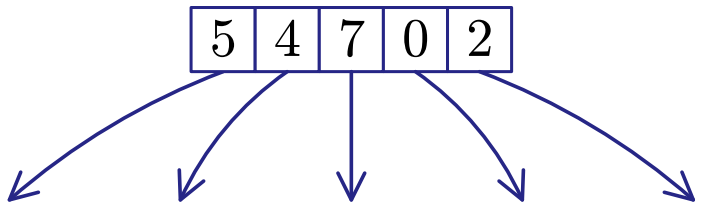
$$5 \cdot 10^4 + 4 \cdot 10^3 + 7 \cdot 10^2 + 0 \cdot 10^1 + 2 \cdot 10^0$$

- Como os circuitos digitais operam com sinais que somente podem assumir dois valores possíveis, os sistemas digitais adotam o sistema de numeração binário.
- No sistema de numeração binário, os números são representados através de apenas dois símbolos (0 e 1), onde a posição de cada símbolo no número também possui um determinado peso.

$$1 \quad 0 \quad 0 \quad 1 \quad 1$$

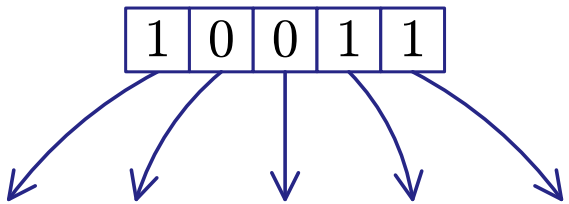
Sistema de Numeração Binário

- O sistema de numeração decimal é universalmente adotado por nós seres humanos para a representação de números. Nesse sistema, os números são representados através de dez símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9), onde a posição de cada símbolo no número possui um determinado peso.



$$5 \cdot 10^4 + 4 \cdot 10^3 + 7 \cdot 10^2 + 0 \cdot 10^1 + 2 \cdot 10^0$$

- Como os circuitos digitais operam com sinais que somente podem assumir dois valores possíveis, os sistemas digitais adotam o sistema de numeração binário.
- No sistema de numeração binário, os números são representados através de apenas dois símbolos (0 e 1), onde a posição de cada símbolo no número também possui um determinado peso.



$$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (19)_{10}$$

Sistema de Numeração Binário

- Todo número pode ser representado utilizando a representação decimal ou a representação binária, embora a representação binária seja a mais conveniente para sistemas digitais.
- Quando digitamos um número decimal no teclado do computador ou de uma calculadora, esse número é convertido para a representação binária antes de ser processado pelos circuitos digitais.
- Algo semelhante acontece quando uma tensão elétrica analógica é aplicada a um Conversor A/D, que converte o valor da tensão em Volts em um número no formato binário que depende da resolução do conversor.
- Inversamente, quando um computador ou calculadora precisa apresentar na tela um resultado numérico ao seu usuário, o número precisa ser convertido do formato binário para o decimal.
- Portanto, as conversões entre as representações decimal e binária são bastante comuns quando estamos lidando com sistemas digitais.

Representação Decimal	Representação Binária
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:

Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:

$$1 \qquad 1 \qquad 0 \qquad 0 \qquad 1$$

Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:

$$\begin{array}{ccccc} 1 & 1 & 0 & 0 & 1 \\ \downarrow & & & & \\ 1 \times 2 = 2 & & & & \end{array}$$

Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:

$$
\begin{array}{ccccccccc}
1 & & 1 & & 0 & & 0 & & 1 \\
\downarrow & & \downarrow & & & & & & \\
1 & \times & 2 & = & 2 & & & & \\
 & & & + & 1 & & & & \\
 & & & & \overline{\quad 3 \quad} & & & &
\end{array}
$$

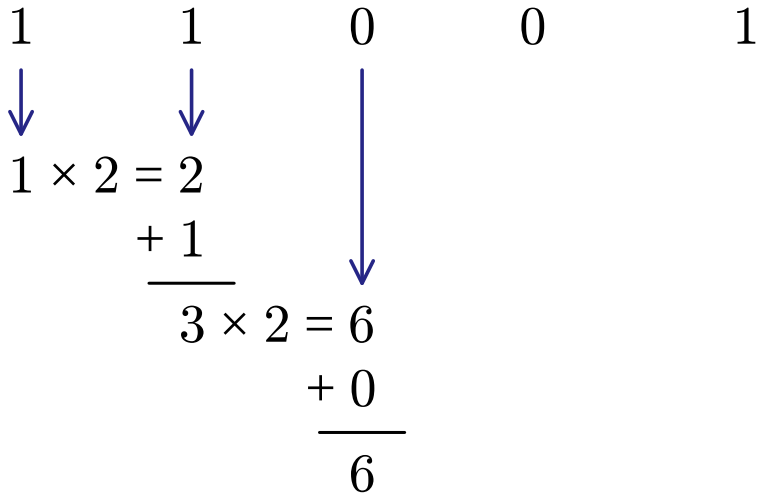
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:

$$
\begin{array}{ccccc}
1 & 1 & 0 & 0 & 1 \\
\downarrow & \downarrow & & & \\
1 \times 2 = 2 & & & & \\
+\ 1 & & & & \\
\hline
3 \times 2 = 6 & & & &
\end{array}
$$

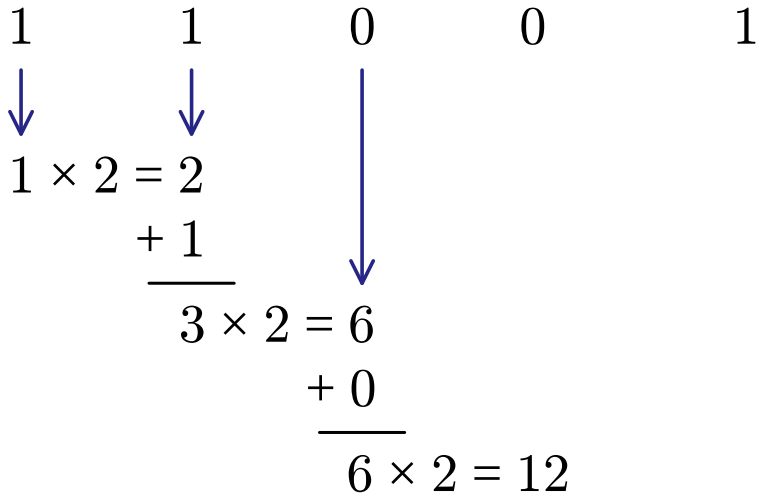
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:



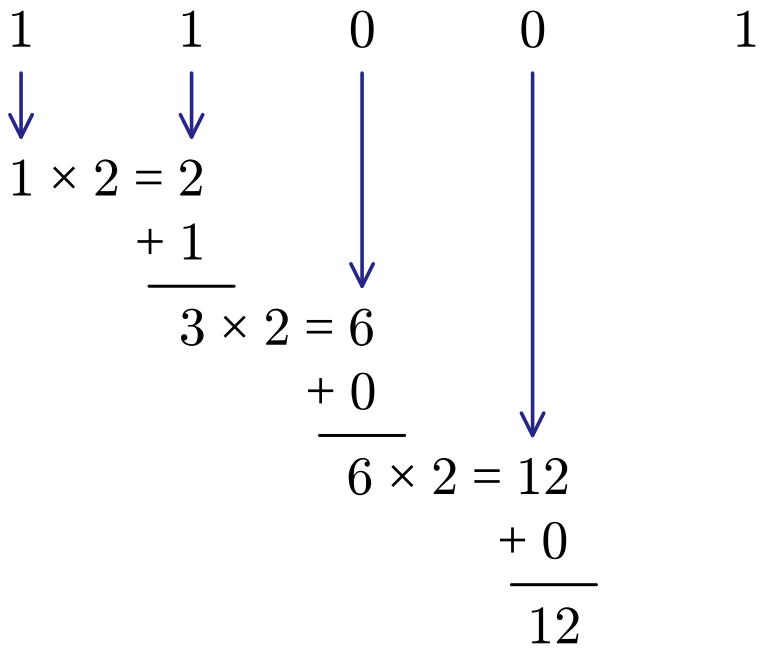
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:



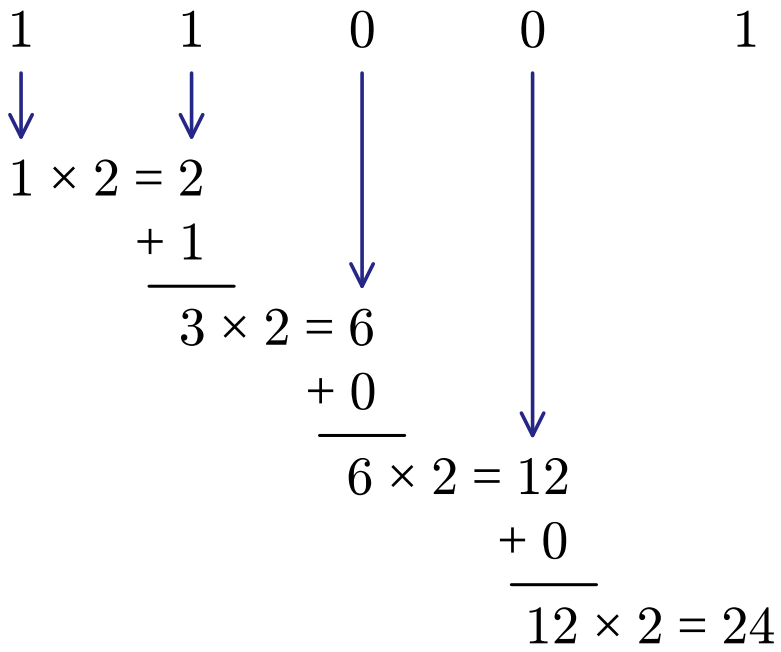
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:



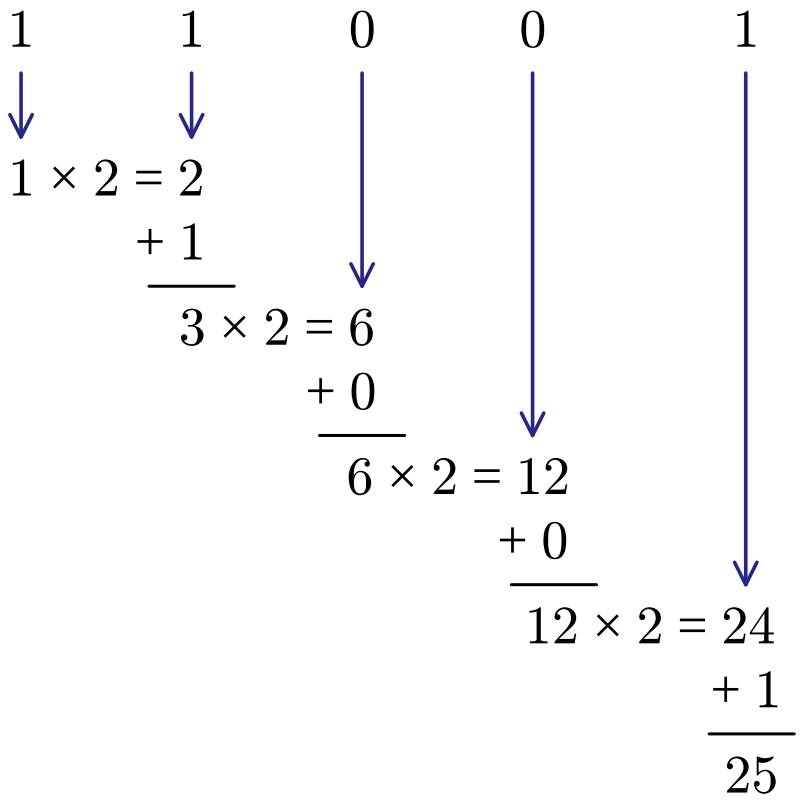
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:



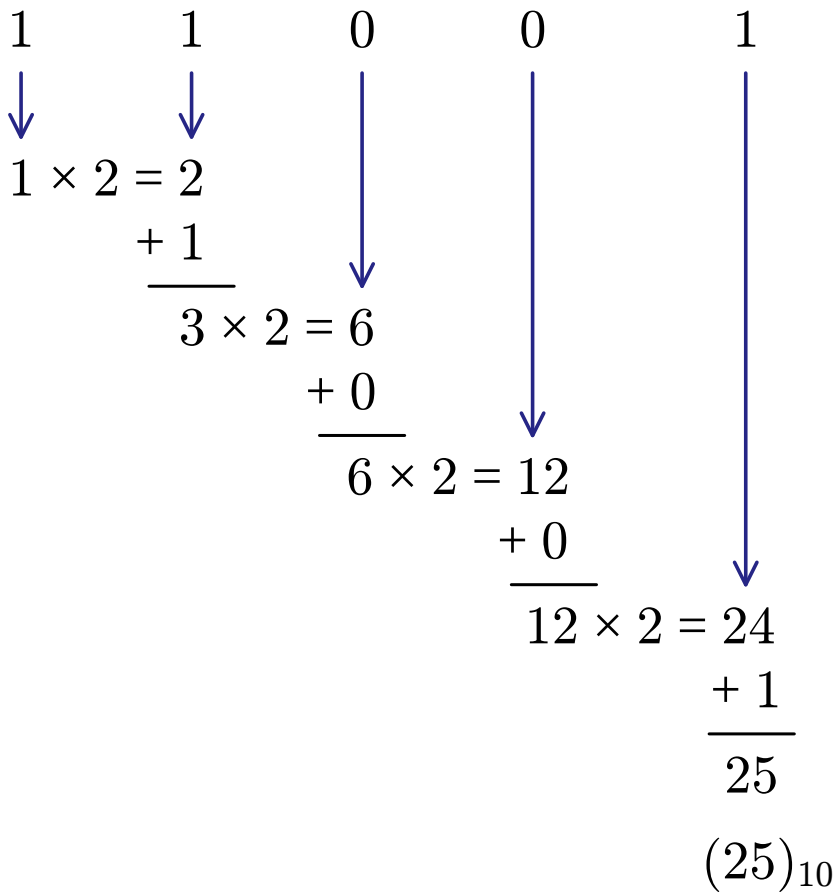
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:



Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato binário para o formato decimal pode ser realizada através de uma sequência de multiplicações por dois:
- **EXEMPLO**: Expressar o número $(11001)_2$ no formato decimal:



Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato decimal para o formato binário pode ser realizada através de uma sequência de divisões inteiras por dois.
- **EXEMPLO**: Expressar o número $(25)_{10}$ no formato binário:

Conversão entre os Formatos Decimal e Binário

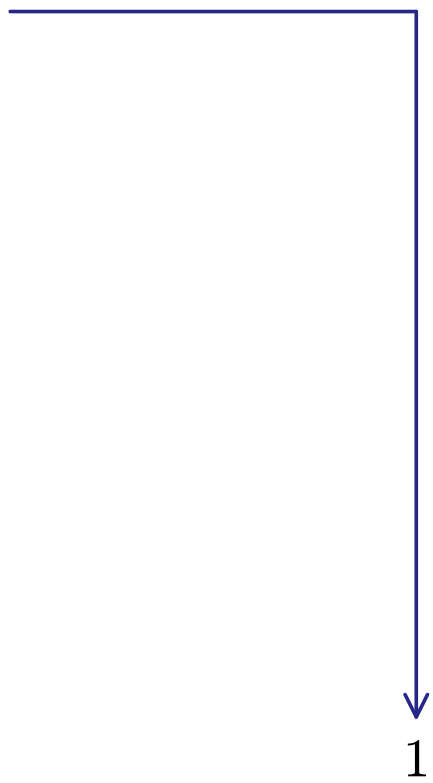
- A conversão de um número expresso em formato decimal para o formato binário pode ser realizada através de uma sequência de divisões inteiras por dois.
- **EXEMPLO**: Expressar o número $(25)_{10}$ no formato binário:

$$25 \div 2 = 12 + (\text{Resto } 1)$$

Conversão entre os Formatos Decimal e Binário

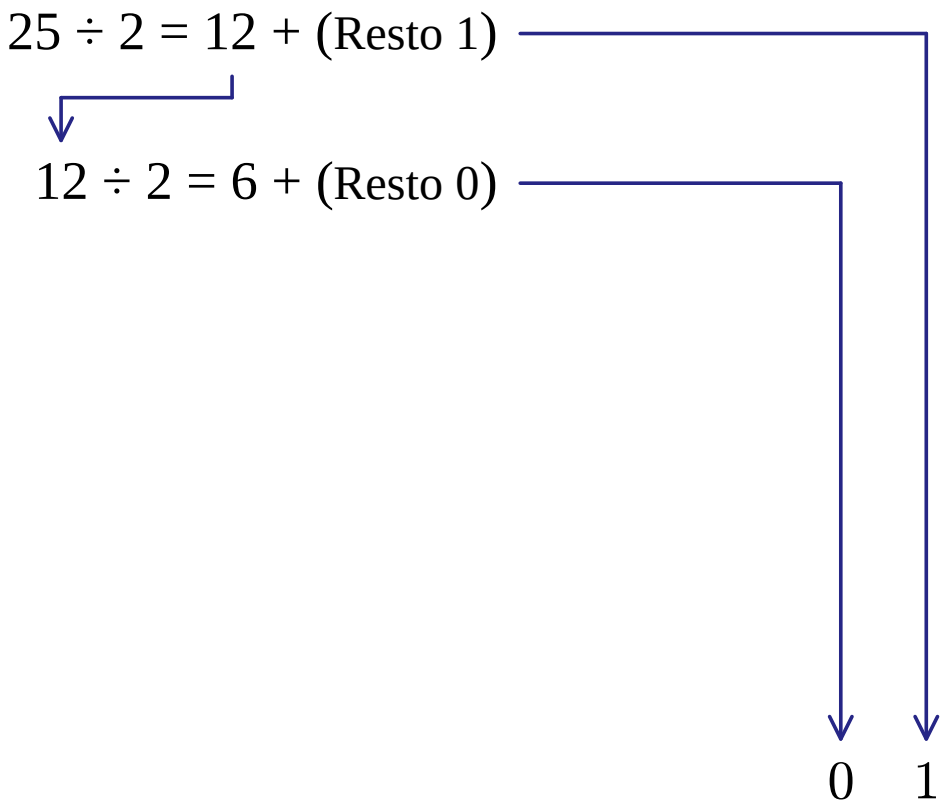
- A conversão de um número expresso em formato decimal para o formato binário pode ser realizada através de uma sequência de divisões inteiras por dois.
- **EXEMPLO**: Expressar o número $(25)_{10}$ no formato binário:

$25 \div 2 = 12 + (\text{Resto } 1)$



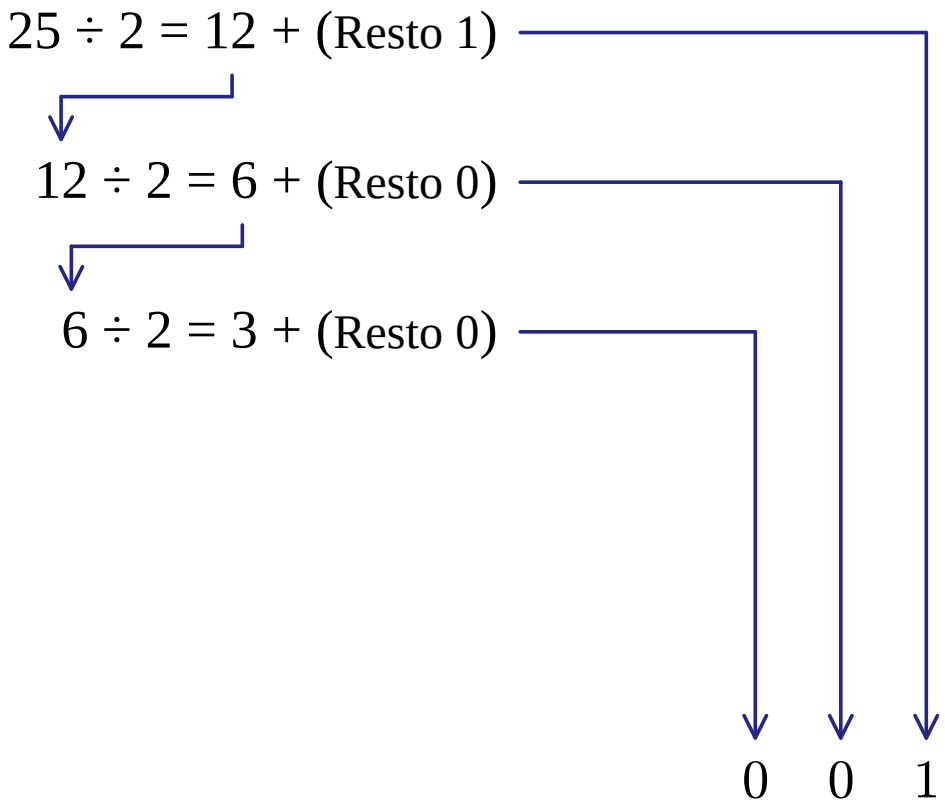
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato decimal para o formato binário pode ser realizada através de uma sequência de divisões inteiras por dois.
- **EXEMPLO**: Expressar o número $(25)_{10}$ no formato binário:



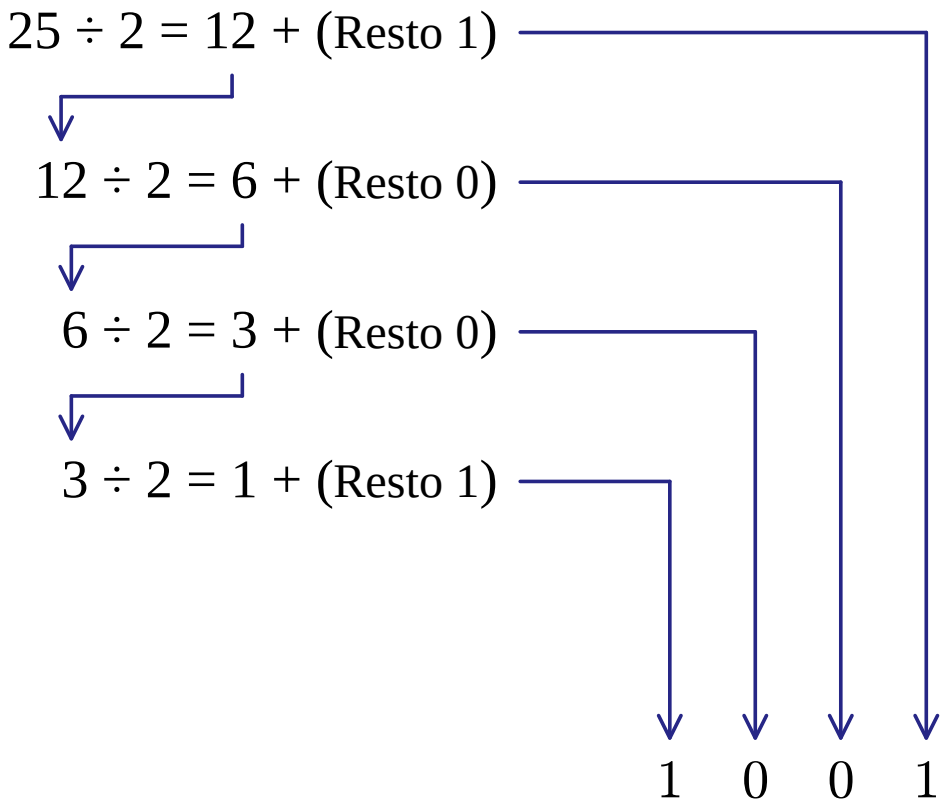
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato decimal para o formato binário pode ser realizada através de uma sequência de divisões inteiras por dois.
- **EXEMPLO**: Expressar o número $(25)_{10}$ no formato binário:



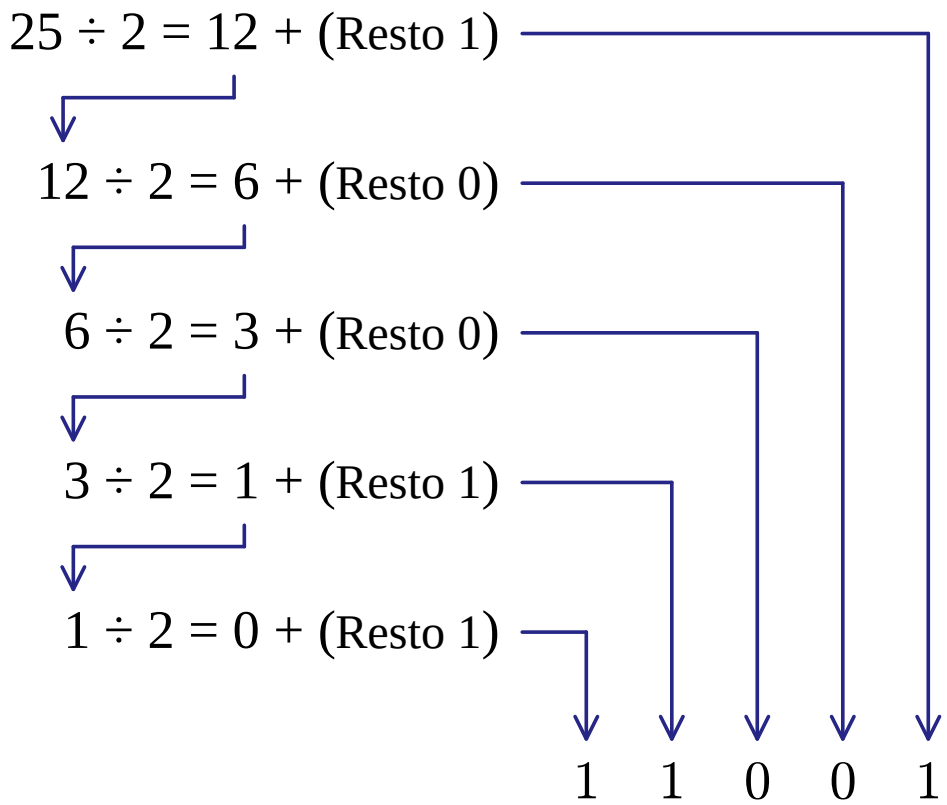
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato decimal para o formato binário pode ser realizada através de uma sequência de divisões inteiras por dois.
- **EXEMPLO**: Expressar o número $(25)_{10}$ no formato binário:



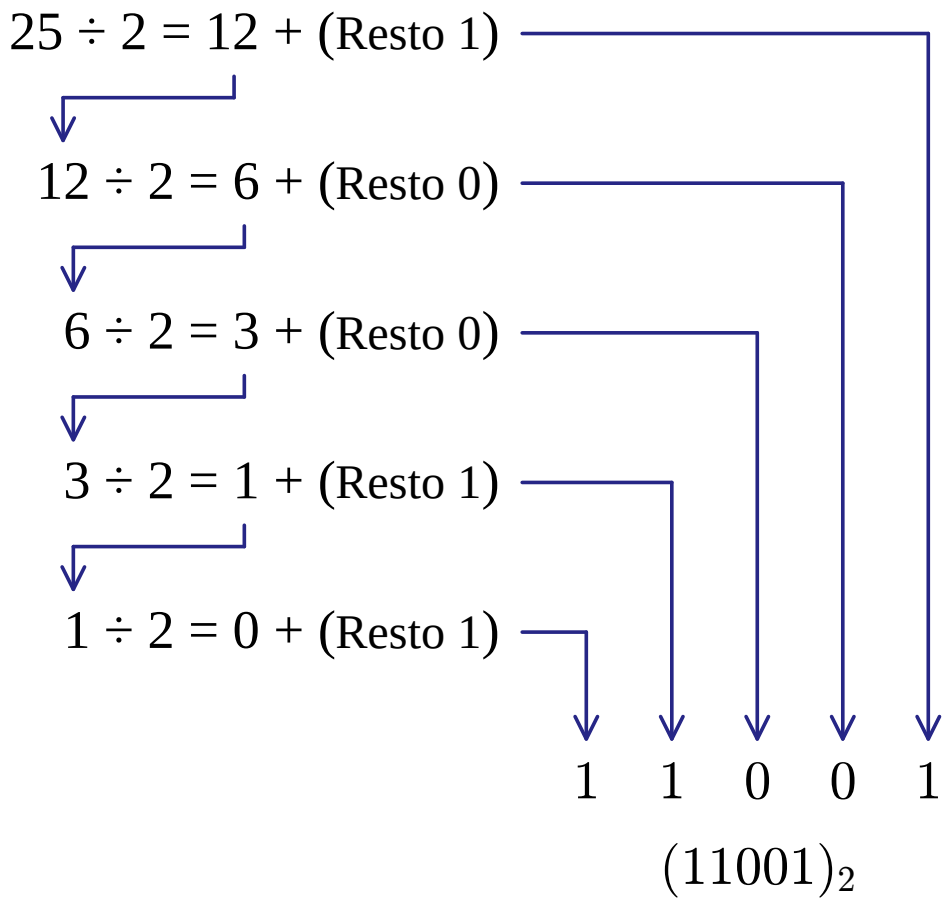
Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato decimal para o formato binário pode ser realizada através de uma sequência de divisões inteiras por dois.
- **EXEMPLO**: Expressar o número $(25)_{10}$ no formato binário:



Conversão entre os Formatos Decimal e Binário

- A conversão de um número expresso em formato decimal para o formato binário pode ser realizada através de uma sequência de divisões inteiras por dois.
- **EXEMPLO**: Expressar o número $(25)_{10}$ no formato binário:

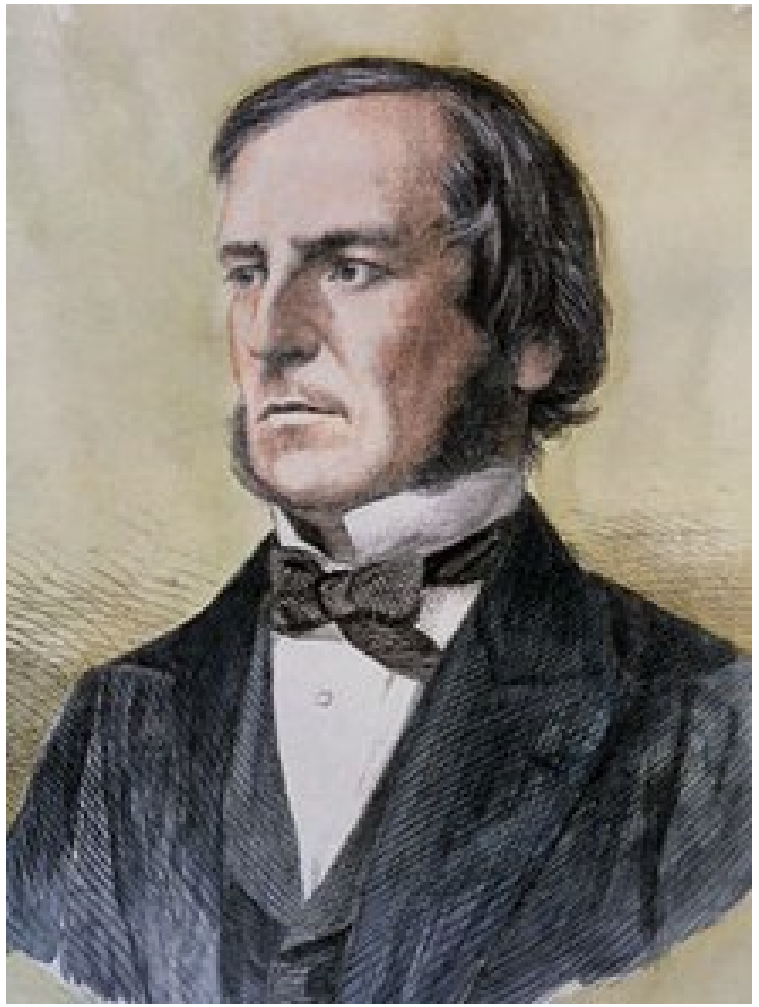


Agenda da Aula - Capítulo 03

- **Conceitos Introdutórios**
  - Sinais e Circuitos Digitais
  - Sistema de Numeração Binário
  - **Álgebra Booleana**
- Portas Lógicas Digitais
- Circuitos Lógicos Combinacionais
- Células de Memória Digitais

Álgebra Booleana

- Como os sinais digitais somente podem assumir dois valores (0 ou 1), a álgebra usual não pode ser utilizada para tratar matematicamente os circuitos digitais.
- Em 1854, o matemático George Boole escreveu o livro *An Investigation of the Laws of Thought*, no qual ele descreveu os meios pelos quais nós fazemos decisões lógicas baseadas em circunstâncias **verdadeiras** ou **falsas**.
- Os métodos descritos por Boole fundamentaram a chamada **Lógica Booleana**, e o sistema de símbolos e operadores matemáticos usados para fazer decisões lógicas são chamados de **Álgebra Booleana**.
- Como a Álgebra Booleana lida com variáveis lógicas que podem assumir apenas os valores **Verdadeiro** ou **Falso**, ela é perfeitamente adequada para tratar matematicamente as decisões lógicas dos Circuitos Digitais.
- Por essa razão, os Circuitos Digitais são frequentemente referidos como **Circuitos Lógicos**.



George Boole (1815-1864)

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$$Y = \overline{A}$$

A	Y
0	1
1	0

AND

$$Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$$Y = \overline{A}$$

A	Y
0	1
1	0

AND

$$Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

EXEMPLO: Circuito de alerta de uma impressora

P	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$$Y = \overline{A}$$

A	Y
0	1
1	0

AND

$$Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

EXEMPLO: Circuito de alerta de uma impressora

P	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Função Lógica

A função lógica que o circuito digital do exemplo deverá executar de modo a tomar a decisão correta quanto ao acendimento do LED é dada por:

$$Y = \overline{P} + B + A$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$Y = \overline{A}$

A	Y
0	1
1	0

AND

$Y = A \cdot B$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$Y = A + B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- As operações lógicas OR e AND também apresentam as mesmas propriedades das operações fundamentais, como a adição e a multiplicação:

$$A + B = B + A \quad \text{e} \quad A \cdot B = B \cdot A \qquad (\textbf{Comutativa})$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$Y = \overline{A}$

A	Y
0	1
1	0

AND

$Y = A \cdot B$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$Y = A + B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- As operações lógicas OR e AND também apresentam as mesmas propriedades das operações fundamentais, como a adição e a multiplicação:

$$A + B = B + A \quad \text{e} \quad A \cdot B = B \cdot A \qquad (\textbf{Comutativa})$$

$$(A + B) + C = A + (B + C) \quad \text{e} \quad (A \cdot B) \cdot C = A \cdot (B \cdot C) \qquad (\textbf{Associativa})$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$Y = \overline{A}$

A	Y
0	1
1	0

AND

$Y = A \cdot B$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$Y = A + B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- As operações lógicas OR e AND também apresentam as mesmas propriedades das operações fundamentais, como a adição e a multiplicação:

$$A + B = B + A \quad \text{e} \quad A \cdot B = B \cdot A \qquad (\textbf{Comutativa})$$

$$(A + B) + C = A + (B + C) \quad \text{e} \quad (A \cdot B) \cdot C = A \cdot (B \cdot C) \qquad (\textbf{Associativa})$$

$$A \cdot (B + C) = A \cdot B + A \cdot C \qquad (\textbf{Distributiva})$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$$Y = \overline{A}$$

A	Y
0	1
1	0

AND

$$Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- As operações lógicas OR e AND também apresentam as mesmas propriedades das operações fundamentais, como a adição e a multiplicação:

$$A + B = B + A \quad \text{e} \quad A \cdot B = B \cdot A \qquad (\textbf{Comutativa})$$

$$(A + B) + C = A + (B + C) \quad \text{e} \quad (A \cdot B) \cdot C = A \cdot (B \cdot C) \qquad (\textbf{Associativa})$$

$$A \cdot (B + C) = A \cdot B + A \cdot C \qquad (\textbf{Distributiva})$$

$$A + 0 = A \quad \text{e} \quad A \cdot 1 = A \qquad (\textbf{Elemento Neutro})$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$$Y = \overline{A}$$

A	Y
0	1
1	0

AND

$$Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Entretanto, as operações lógicas OR e AND apresentam propriedades adicionais que não são observadas nas operações fundamentais adição e multiplicação:

$$A + 1 = 1 \quad \text{e} \quad A \cdot 0 = 0$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$$Y = \overline{A}$$

A	Y
0	1
1	0

AND

$$Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Entretanto, as operações lógicas OR e AND apresentam propriedades adicionais que não são observadas nas operações fundamentais adição e multiplicação:

$$A + 1 = 1 \quad \text{e} \quad A \cdot 0 = 0$$

$$A + A = A \quad \text{e} \quad A \cdot A = A$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$Y = \overline{A}$

A	Y
0	1
1	0

AND

$Y = A \cdot B$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$Y = A + B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Entretanto, as operações lógicas OR e AND apresentam propriedades adicionais que não são observadas nas operações fundamentais adição e multiplicação:

$$A + 1 = 1 \quad \text{e} \quad A \cdot 0 = 0$$

$$A + A = A \quad \text{e} \quad A \cdot A = A$$

$$A + \overline{A} = 1 \quad \text{e} \quad A \cdot \overline{A} = 0 \qquad (\textbf{Lei do Complemento})$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$Y = \overline{A}$

A	Y
0	1
1	0

AND

$Y = A \cdot B$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$Y = A + B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Entretanto, as operações lógicas OR e AND apresentam propriedades adicionais que não são observadas nas operações fundamentais adição e multiplicação:

$$A + 1 = 1 \quad \text{e} \quad A \cdot 0 = 0$$

$$A + A = A \quad \text{e} \quad A \cdot A = A$$

$$A + \overline{A} = 1 \quad \text{e} \quad A \cdot \overline{A} = 0 \qquad (\textbf{Lei do Complemento})$$

$$\overline{\overline{A}} = A \qquad (\textbf{Lei da Dupla Negação})$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$$Y = \overline{A}$$

A	Y
0	1
1	0

AND

$$Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Finalmente, uma das principais propriedades da Álgebra Booleana é dada pelo Teorema de De Morgan:

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

Álgebra Booleana

Operadores Lógicos

De acordo com a teoria da Álgebra Booleana, qualquer função lógica pode ser descrita matematicamente usando apenas três operadores básicos: **NOT**, **AND** e **OR**. Portanto, essas são as operações lógicas básicas utilizadas na síntese dos Circuitos Lógicos Digitais.

NOT

$$Y = \overline{A}$$

A	Y
0	1
1	0

AND

$$Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- Finalmente, uma das principais propriedades da Álgebra Booleana é dada pelo Teorema de De Morgan:

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

Álgebra Booleana e Funções Lógicas

EXEMPLO: Qual seria a expressão lógica Booleana que realiza a função lógica descrita na Tabela Verdade abaixo:

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Álgebra Booleana e Funções Lógicas

EXEMPLO: Qual seria a expressão lógica Booleana que realiza a função lógica descrita na Tabela Verdade abaixo:

A	B	C	Y	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\rightarrow \overline{A}\cdot B\cdot\overline{C}$
0	1	1	1	$\rightarrow \overline{A}\cdot B\cdot C$
1	0	0	0	
1	0	1	1	$\rightarrow A\cdot\overline{B}\cdot C$
1	1	0	0	
1	1	1	1	$\rightarrow A\cdot B\cdot C$

Álgebra Booleana e Funções Lógicas

EXEMPLO: Qual seria a expressão lógica Booleana que realiza a função lógica descrita na Tabela Verdade abaixo:

A	B	C	Y	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\rightarrow \overline{A}\cdot B\cdot\overline{C}$
0	1	1	1	$\rightarrow \overline{A}\cdot B\cdot C$
1	0	0	0	
1	0	1	1	$\rightarrow A\cdot\overline{B}\cdot C$
1	1	0	0	
1	1	1	1	$\rightarrow A\cdot B\cdot C$

- A função lógica pode ser expressa na forma de uma **Soma de Produtos**:

$$Y = \overline{A}\cdot B\cdot\overline{C} + \overline{A}\cdot B\cdot C + A\cdot\overline{B}\cdot C + A\cdot B\cdot C$$

Álgebra Booleana e Funções Lógicas

EXEMPLO: Qual seria a expressão lógica Booleana que realiza a função lógica descrita na Tabela Verdade abaixo:

A	B	C	Y	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\rightarrow \overline{A}\cdot B\cdot\overline{C}$
0	1	1	1	$\rightarrow \overline{A}\cdot B\cdot C$
1	0	0	0	
1	0	1	1	$\rightarrow A\cdot\overline{B}\cdot C$
1	1	0	0	
1	1	1	1	$\rightarrow A\cdot B\cdot C$

- A função lógica pode ser expressa na forma de uma **Soma de Produtos**:

$$Y = \overline{A}\cdot B\cdot\overline{C} + \overline{A}\cdot B\cdot C + A\cdot\overline{B}\cdot C + A\cdot B\cdot C$$

- A propriedade associativa também é válida na Álgebra Booleana:

$$Y = \overline{A}\cdot B\left(\overline{C} + C\right) + A\cdot C\left(\overline{B} + B\right)$$

Álgebra Booleana e Funções Lógicas

EXEMPLO: Qual seria a expressão lógica Booleana que realiza a função lógica descrita na Tabela Verdade abaixo:

A	B	C	Y	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\rightarrow \overline{A}\cdot B\cdot\overline{C}$
0	1	1	1	$\rightarrow \overline{A}\cdot B\cdot C$
1	0	0	0	
1	0	1	1	$\rightarrow A\cdot\overline{B}\cdot C$
1	1	0	0	
1	1	1	1	$\rightarrow A\cdot B\cdot C$

- A função lógica pode ser expressa na forma de uma **Soma de Produtos**:

$$Y = \overline{A}\cdot B\cdot\overline{C} + \overline{A}\cdot B\cdot C + A\cdot\overline{B}\cdot C + A\cdot B\cdot C$$

- A propriedade associativa também é válida na Álgebra Booleana:

$$Y = \overline{A}\cdot B\left(\overline{C} + C\right) + A\cdot C\left(\overline{B} + B\right)$$

- Como $\overline{B} + B = 1$ e $\overline{C} + C = 1$ para quaisquer valores das variáveis lógicas B e C, então, podemos escrever que:

$$Y = \overline{A}\cdot B + A\cdot C$$

Mapa de Karnaugh

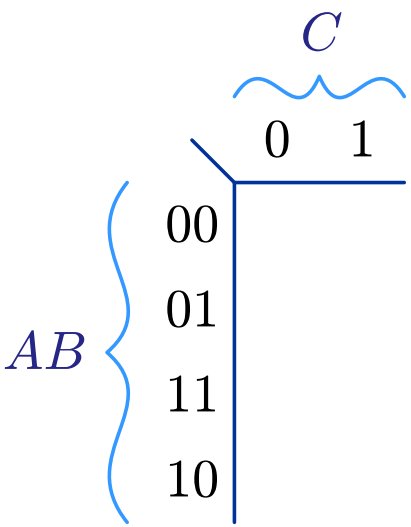
- O **Mapa de Karnaugh** (K-Map) é um método gráfico que permite obter diretamente a função lógica booleana simplificada a partir de uma Tabela Verdade.
- A grande vantagem do Mapa de Karnaugh está no fato de que a tabela verdade é organizada de forma que todas as combinações possíveis dos sinais de entrada estão dispostas de modo que as combinações adjacentes difiram entre si por apenas um único dígito binário (*bit*).

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Mapa de Karnaugh

- O **Mapa de Karnaugh** (K-Map) é um método gráfico que permite obter diretamente a função lógica booleana simplificada a partir de uma Tabela Verdade.
- A grande vantagem do Mapa de Karnaugh está no fato de que a tabela verdade é organizada de forma que todas as combinações possíveis dos sinais de entrada estão dispostas de modo que as combinações adjacentes difiram entre si por apenas um único dígito binário (*bit*).

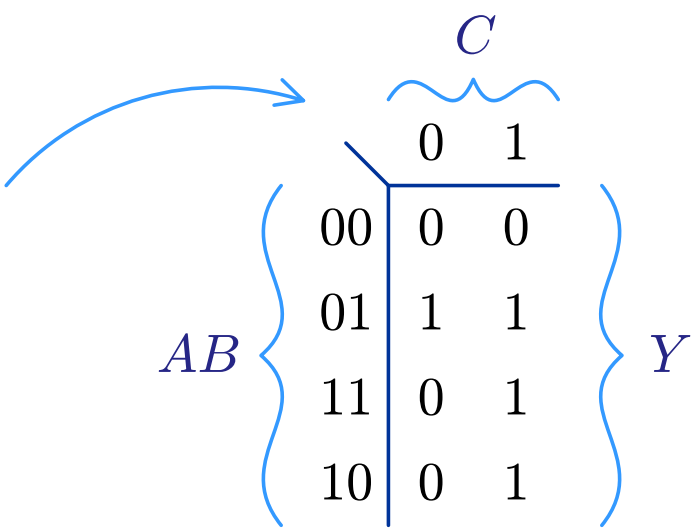
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Mapa de Karnaugh

- O **Mapa de Karnaugh** (K-Map) é um método gráfico que permite obter diretamente a função lógica booleana simplificada a partir de uma Tabela Verdade.
- A grande vantagem do Mapa de Karnaugh está no fato de que a tabela verdade é organizada de forma que todas as combinações possíveis dos sinais de entrada estão dispostas de modo que as combinações adjacentes difiram entre si por apenas um único dígito binário (*bit*).

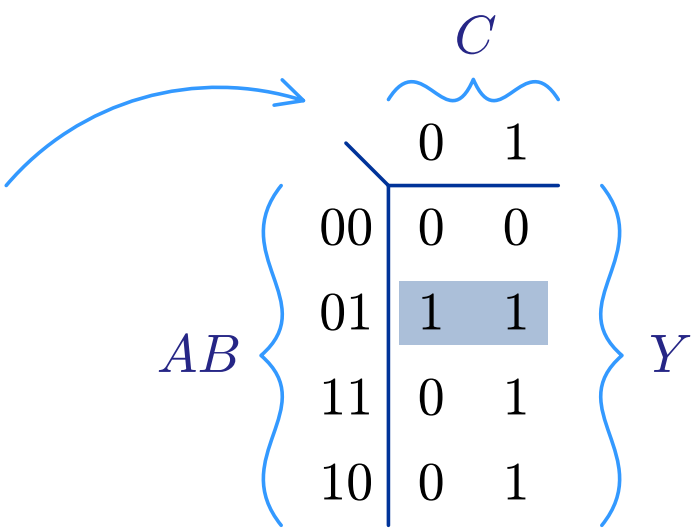
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Mapa de Karnaugh

- O **Mapa de Karnaugh** (K-Map) é um método gráfico que permite obter diretamente a função lógica booleana simplificada a partir de uma Tabela Verdade.
- A grande vantagem do Mapa de Karnaugh está no fato de que a tabela verdade é organizada de forma que todas as combinações possíveis dos sinais de entrada estão dispostas de modo que as combinações adjacentes difiram entre si por apenas um único dígito binário (*bit*).

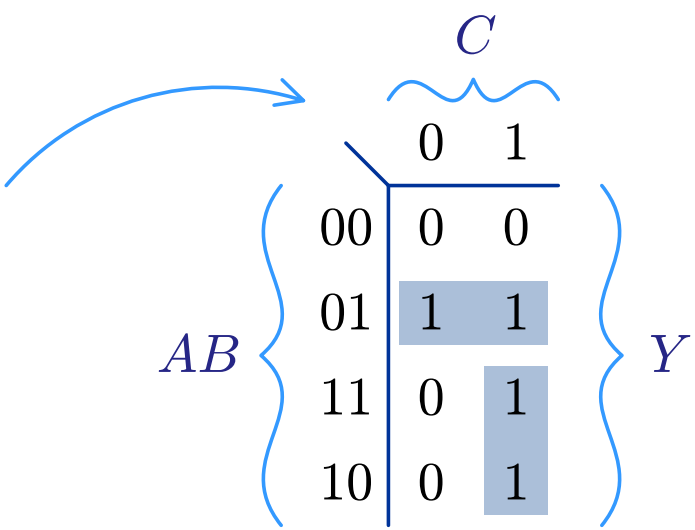
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Mapa de Karnaugh

- O **Mapa de Karnaugh** (K-Map) é um método gráfico que permite obter diretamente a função lógica booleana simplificada a partir de uma Tabela Verdade.
- A grande vantagem do Mapa de Karnaugh está no fato de que a tabela verdade é organizada de forma que todas as combinações possíveis dos sinais de entrada estão dispostas de modo que as combinações adjacentes difiram entre si por apenas um único dígito binário (*bit*).

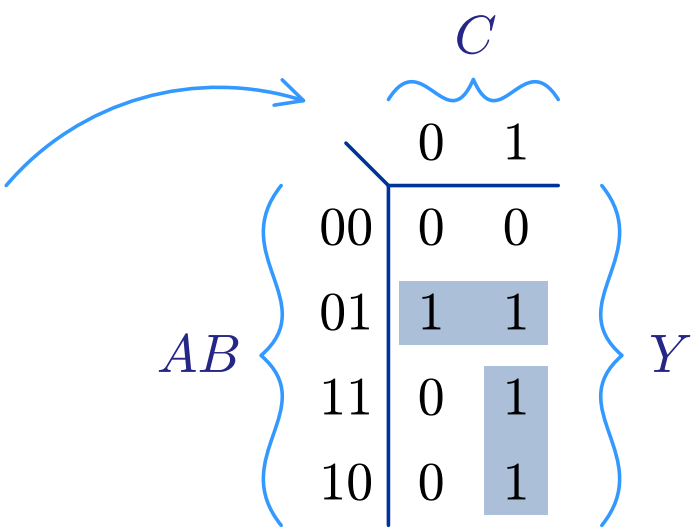
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Mapa de Karnaugh

- O **Mapa de Karnaugh** (K-Map) é um método gráfico que permite obter diretamente a função lógica booleana simplificada a partir de uma Tabela Verdade.
- A grande vantagem do Mapa de Karnaugh está no fato de que a tabela verdade é organizada de forma que todas as combinações possíveis dos sinais de entrada estão dispostas de modo que as combinações adjacentes difiram entre si por apenas um único dígito binário (*bit*).

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



- Função lógica booleana realizada pela Tabela Verdade:

$$Y = \overline{A} \cdot B + A \cdot C$$

Mapa de Karnaugh

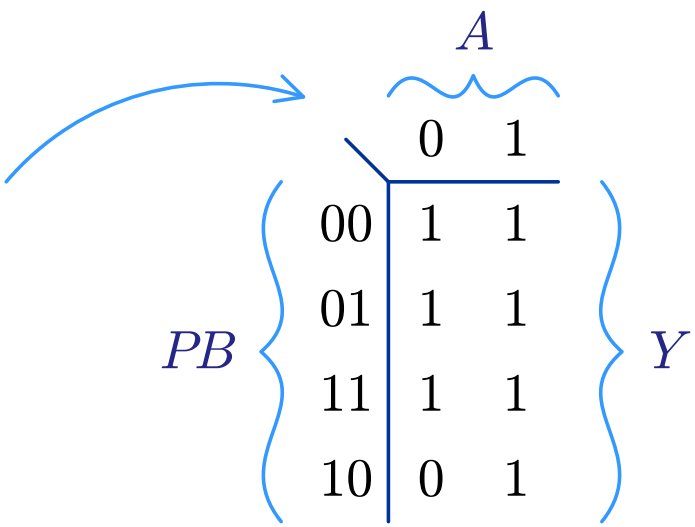
- O Mapa de Karnaugh também pode ser usado para obter a função lógica booleana que realiza a Tabela Verdade do sistema de alerta da impressora.
- Neste exemplo, vemos que a expressão lógica resultante será mais simples quanto maior for a quantidade de saídas '1' que conseguimos agrupar. Além disso, observamos que é possível agrupar as mesmas saídas mais de uma vez.

P	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado para obter a função lógica booleana que realiza a Tabela Verdade do sistema de alerta da impressora.
- Neste exemplo, vemos que a expressão lógica resultante será mais simples quanto maior for a quantidade de saídas '1' que conseguimos agrupar. Além disso, observamos que é possível agrupar as mesmas saídas mais de uma vez.

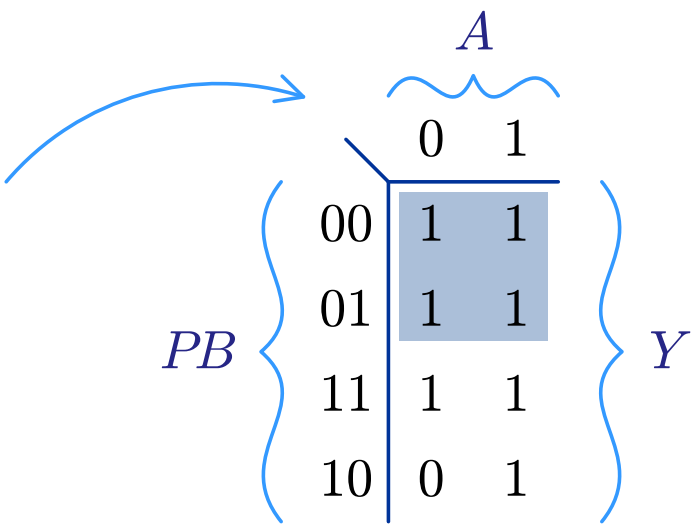
P	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado para obter a função lógica booleana que realiza a Tabela Verdade do sistema de alerta da impressora.
- Neste exemplo, vemos que a expressão lógica resultante será mais simples quanto maior for a quantidade de saídas '1' que conseguimos agrupar. Além disso, observamos que é possível agrupar as mesmas saídas mais de uma vez.

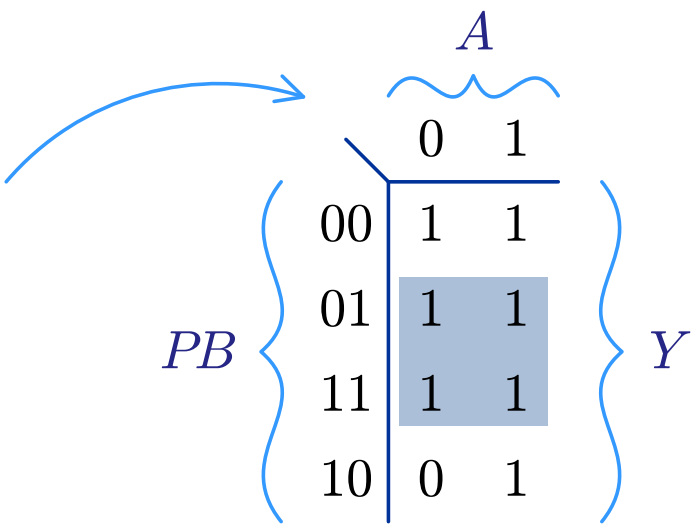
P	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado para obter a função lógica booleana que realiza a Tabela Verdade do sistema de alerta da impressora.
- Neste exemplo, vemos que a expressão lógica resultante será mais simples quanto maior for a quantidade de saídas '1' que conseguimos agrupar. Além disso, observamos que é possível agrupar as mesmas saídas mais de uma vez.

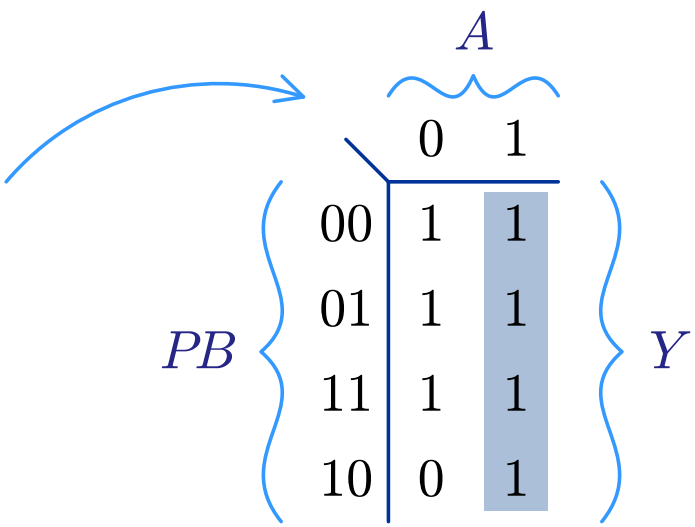
P	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado para obter a função lógica booleana que realiza a Tabela Verdade do sistema de alerta da impressora.
- Neste exemplo, vemos que a expressão lógica resultante será mais simples quanto maior for a quantidade de saídas '1' que conseguimos agrupar. Além disso, observamos que é possível agrupar as mesmas saídas mais de uma vez.

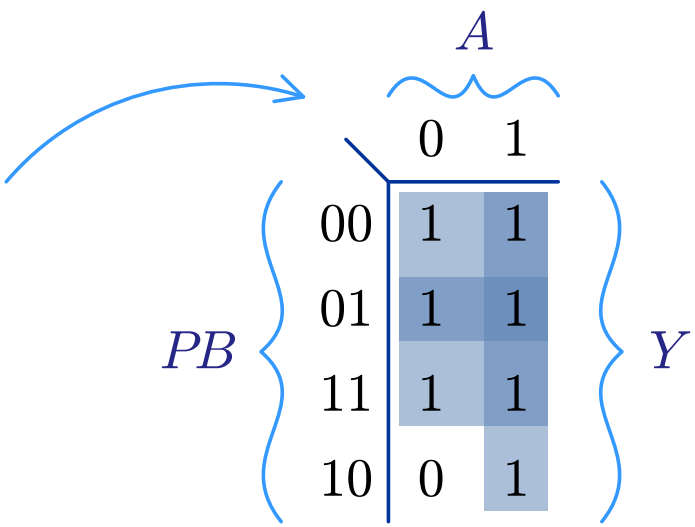
P	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado para obter a função lógica booleana que realiza a Tabela Verdade do sistema de alerta da impressora.
- Neste exemplo, vemos que a expressão lógica resultante será mais simples quanto maior for a quantidade de saídas '1' que conseguimos agrupar. Além disso, observamos que é possível agrupar as mesmas saídas mais de uma vez.

P	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



- Função lógica booleana realizada pela Tabela Verdade:

$$Y = \overline{P} + B + A$$

Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado no caso de Tabelas Verdade com quatro sinais de entrada.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0

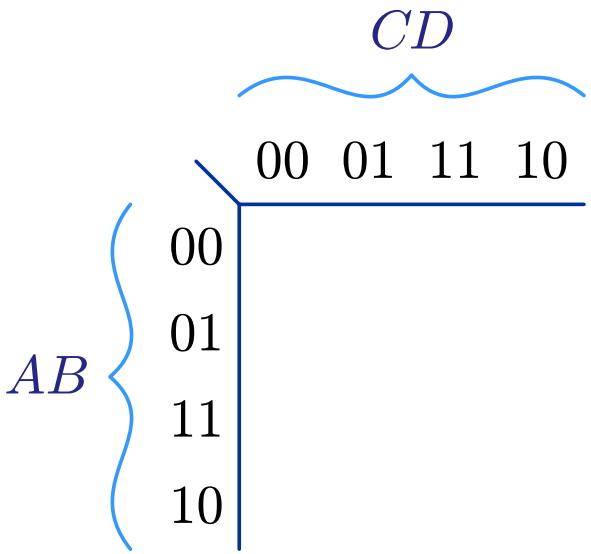
A	B	C	D	Y
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado no caso de Tabelas Verdade com quatro sinais de entrada.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0

A	B	C	D	Y
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

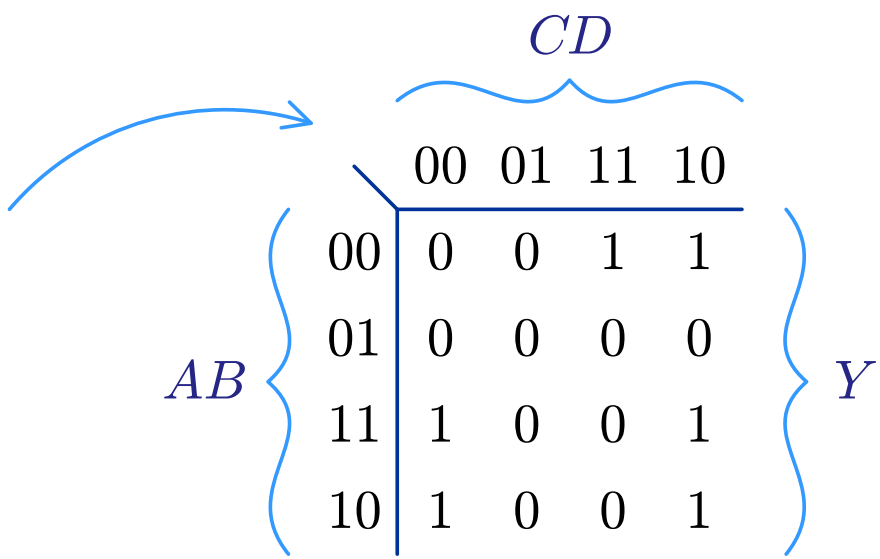


Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado no caso de Tabelas Verdade com quatro sinais de entrada.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0

A	B	C	D	Y
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

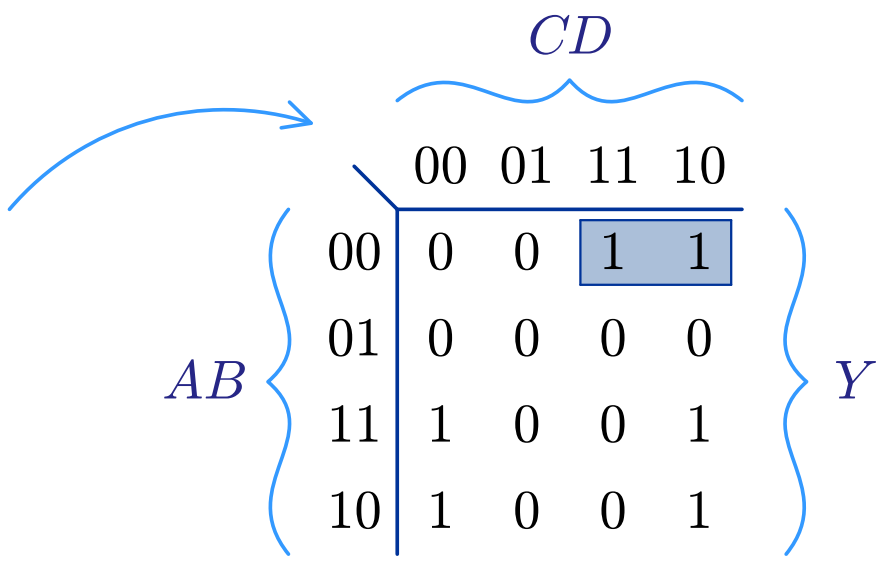


Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado no caso de Tabelas Verdade com quatro sinais de entrada.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0

A	B	C	D	Y
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

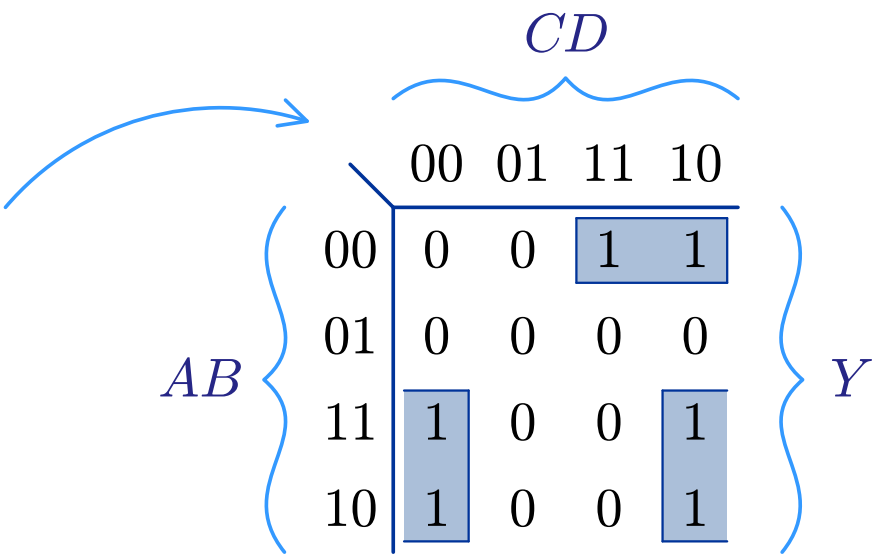


Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado no caso de Tabelas Verdade com quatro sinais de entrada.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0

A	B	C	D	Y
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

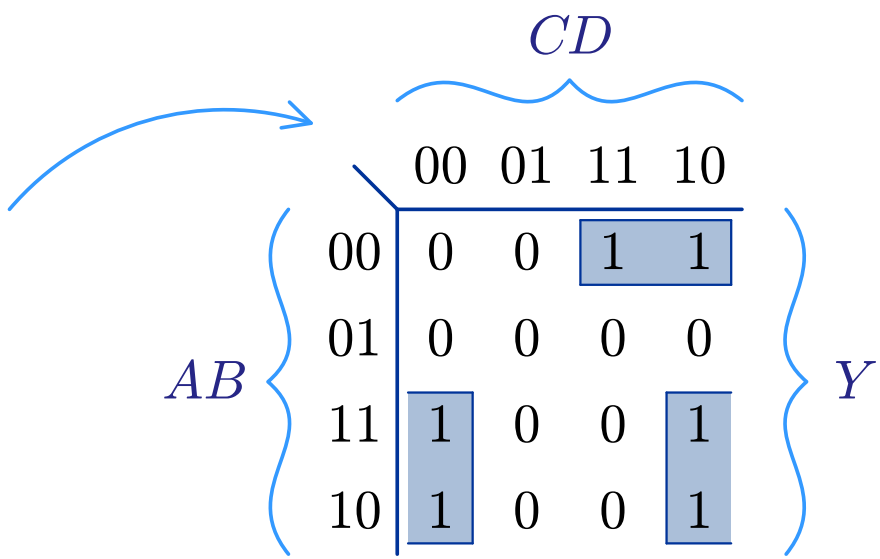


Mapa de Karnaugh

- O Mapa de Karnaugh também pode ser usado no caso de Tabelas Verdade com quatro sinais de entrada.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0

A	B	C	D	Y
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0



- Função lógica booleana realizada pela Tabela Verdade:

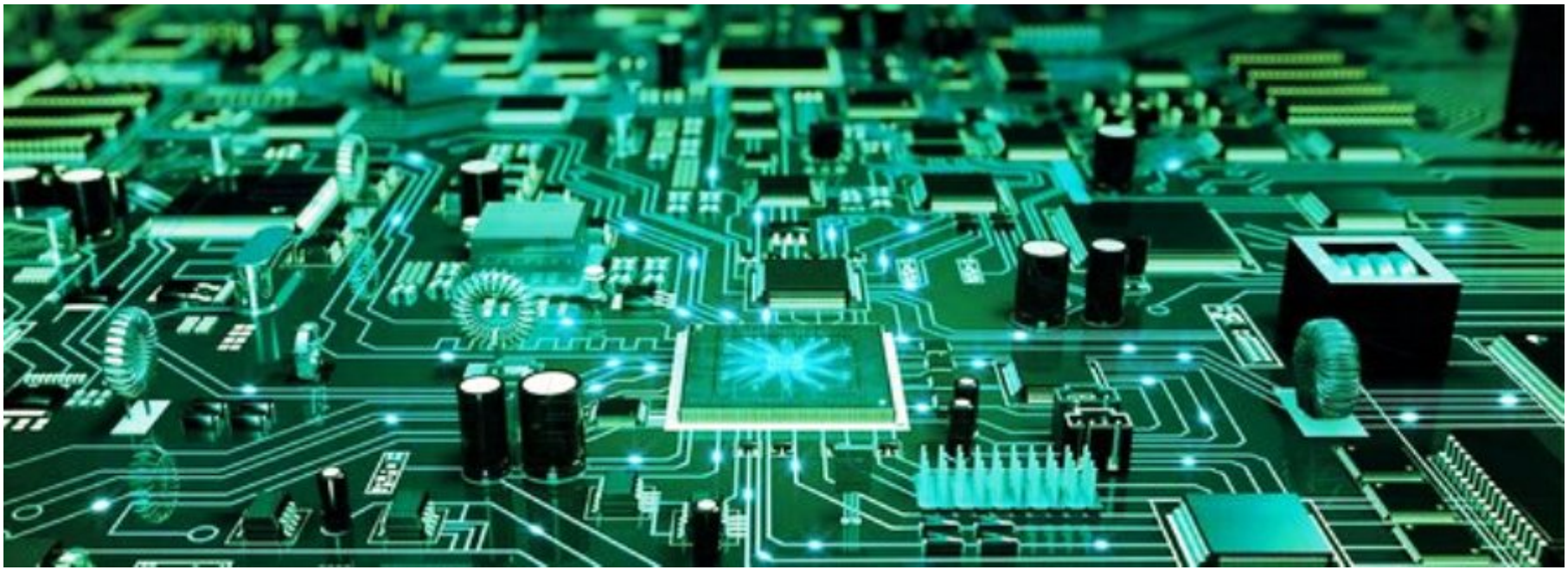
$$Y = \overline{A}\,\overline{B}\,C + A\,\overline{D}$$

Agenda da Aula - Capítulo 03

- Conceitos Introdutórios
  - Sinais e Circuitos Digitais
  - Sistema de Numeração Binário
  - Álgebra Booleana
- **Portas Lógicas Digitais**
- Circuitos Lógicos Combinacionais
- Células de Memória Digitais

Portas Lógicas Digitais

Os Circuitos Digitais são construídos a partir de blocos básicos de circuito chamados **Portas Lógicas** (*Logic Gates*), cuja função é realizar eletricamente as operações lógicas da Álgebra Booleana (NOT, AND e OR). Combinando essas portas lógicas, é possível construir circuitos digitais capazes de realizar qualquer função lógica.

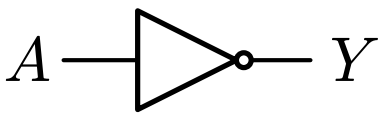


Porta Lógica Inversora (NOT)

- A Porta Lógica inversora realiza eletricamente a operação Booleana básica **NOT**. Neste circuito, uma tensão elétrica próxima de zero (Terra) é interpretada como o valor lógico **FALSO** ou '0', e uma tensão próxima da alimentação do circuito (V_{CC}) é interpretada como **VERDADEIRO** ou '1'.

Porta Lógica NOT

$$Y = \overline{A}$$



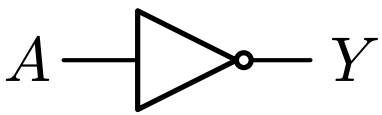
A	Y
0	1
1	0

Porta Lógica Inversora (NOT)

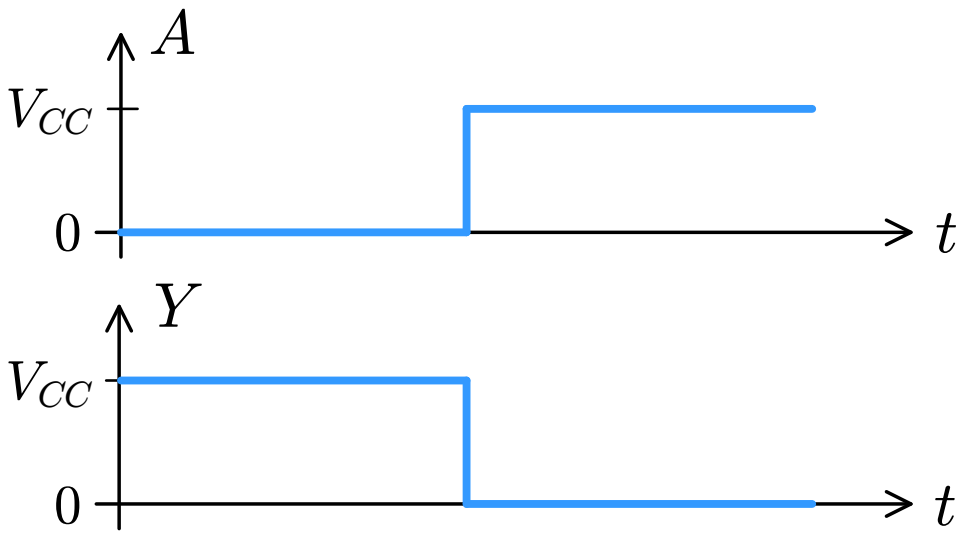
- A Porta Lógica inversora realiza eletricamente a operação Booleana básica **NOT**. Neste circuito, uma tensão elétrica próxima de zero (Terra) é interpretada como o valor lógico **FALSO** ou '0', e uma tensão próxima da alimentação do circuito (V_{CC}) é interpretada como **VERDADEIRO** ou '1'.

Porta Lógica NOT

$$Y = \overline{A}$$



A	Y
0	1
1	0

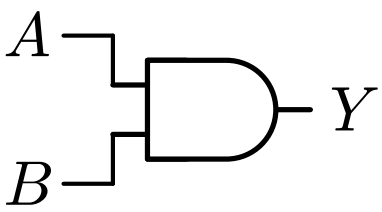


Porta Lógica AND

- Esta Porta Lógica realiza eletricamente a operação Booleana básica **AND**, onde uma tensão elétrica próxima de zero (Terra) é interpretada como o valor lógico **FALSO** ou '0', e uma tensão próxima da alimentação do circuito (V_{CC}) é interpretada como **VERDADEIRO** ou '1'.

Porta Lógica AND

$$Y = A \cdot B$$



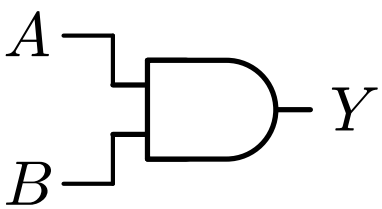
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Porta Lógica AND

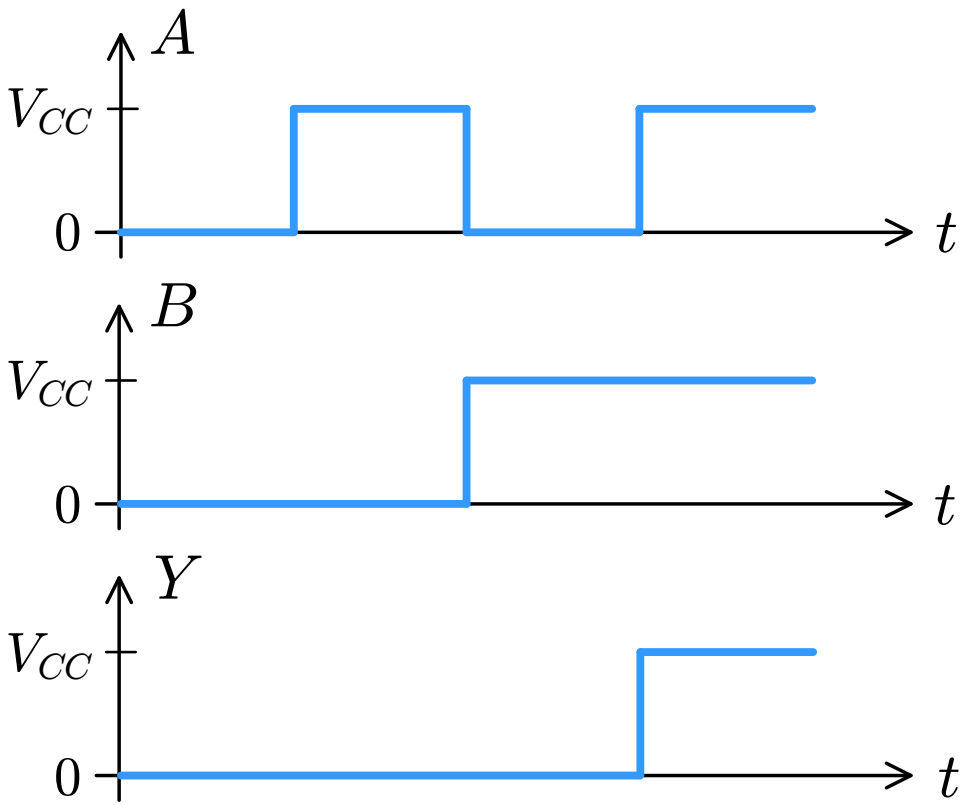
- Esta Porta Lógica realiza eletricamente a operação Booleana básica **AND**, onde uma tensão elétrica próxima de zero (Terra) é interpretada como o valor lógico **FALSO** ou '0', e uma tensão próxima da alimentação do circuito (V_{CC}) é interpretada como **VERDADEIRO** ou '1'.

Porta Lógica AND

$$Y = A \cdot B$$



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

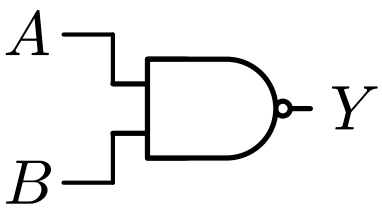


Porta Lógica NAND

- Para facilitar a construção de Circuitos Lógicos, também existe a Porta Lógica **NAND**, que combina as operações lógicas básicas AND e NOT.

Porta Lógica NAND

$$Y = \overline{A \cdot B}$$



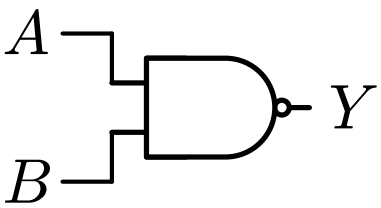
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Porta Lógica NAND

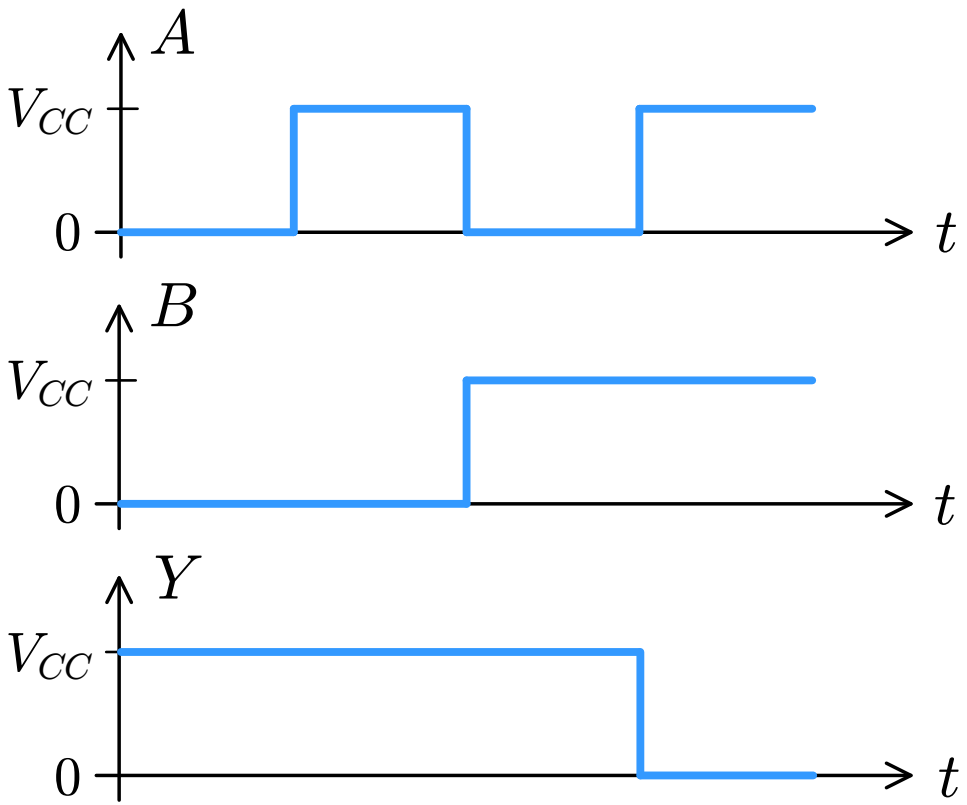
- Para facilitar a construção de Circuitos Lógicos, também existe a Porta Lógica **NAND**, que combina as operações lógicas básicas AND e NOT.

Porta Lógica NAND

$$Y = \overline{A \cdot B}$$



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

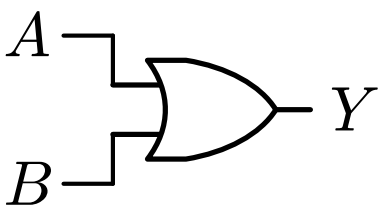


Função Lógica OR

- Esta Porta Lógica realiza eletricamente a operação Booleana básica **OR**, onde uma tensão elétrica próxima de zero (Terra) é interpretada como o valor lógico **FALSO** ou '0', e uma tensão próxima da alimentação do circuito (V_{CC}) é interpretada como **VERDADEIRO** ou '1'.

Porta Lógica OR

$$Y = A+B$$



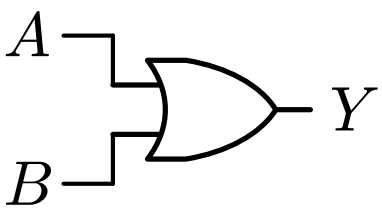
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Função Lógica OR

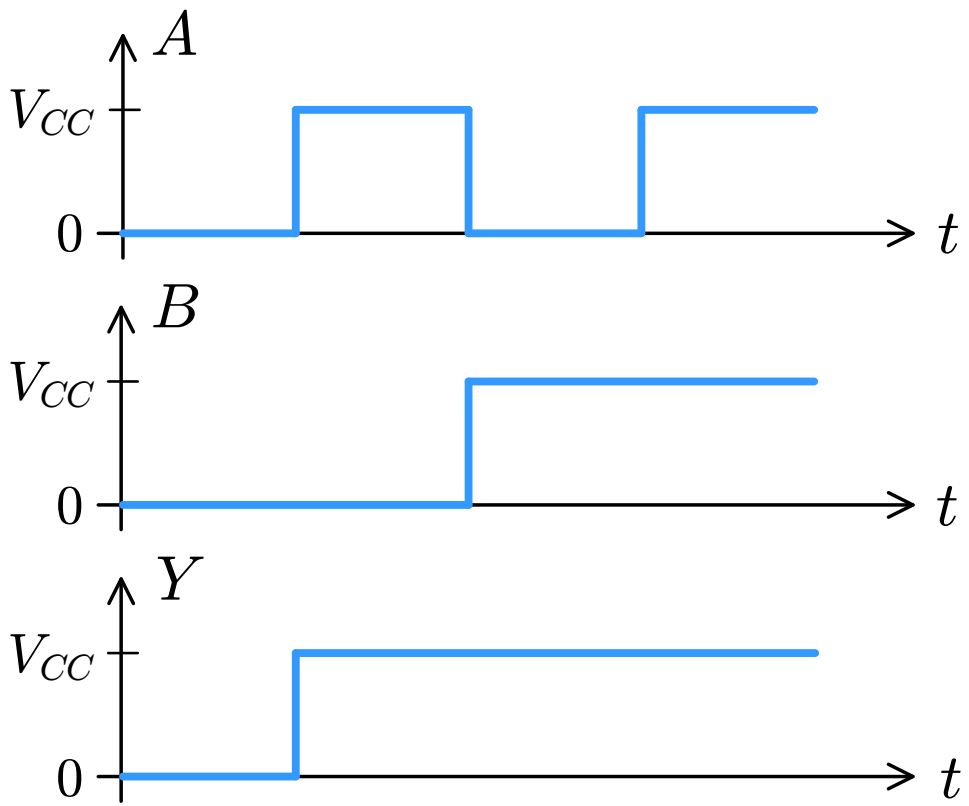
- Esta Porta Lógica realiza eletricamente a operação Booleana básica **OR**, onde uma tensão elétrica próxima de zero (Terra) é interpretada como o valor lógico **FALSO** ou '0', e uma tensão próxima da alimentação do circuito (V_{CC}) é interpretada como **VERDADEIRO** ou '1'.

Porta Lógica OR

$$Y = A+B$$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

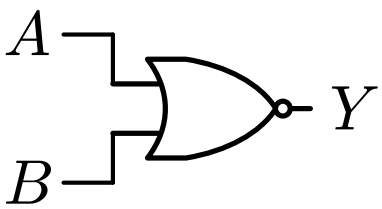


Função Lógica NOR

- Para facilitar a construção de Circuitos Lógicos, também existe a Porta Lógica **NOR**, que combina as operações lógicas básicas OR e NOT.

Porta Lógica NOR

$$Y = \overline{A+B}$$



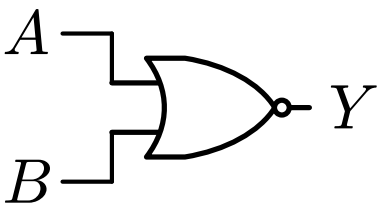
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Função Lógica NOR

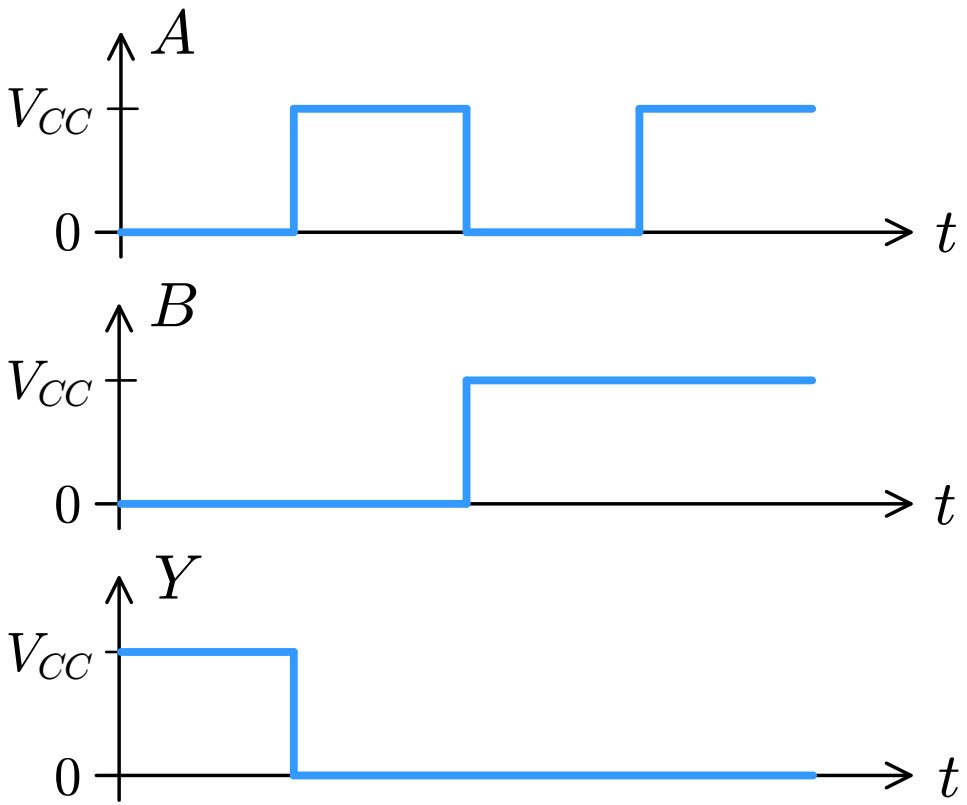
- Para facilitar a construção de Circuitos Lógicos, também existe a Porta Lógica **NOR**, que combina as operações lógicas básicas OR e NOT.

Porta Lógica NOR

$$Y = \overline{A+B}$$



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

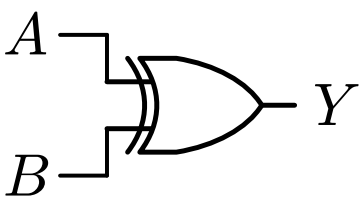


Porta Lógica XOR

- Uma função lógica frequentemente usada em circuitos lógicos é o **OU Exclusivo** (*Exclusive OR* - XOR). A Porta Lógica **XOR** realiza eletricamente essa função.

Porta Lógica XOR

$$Y = A \oplus B$$



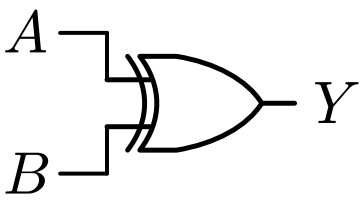
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Porta Lógica XOR

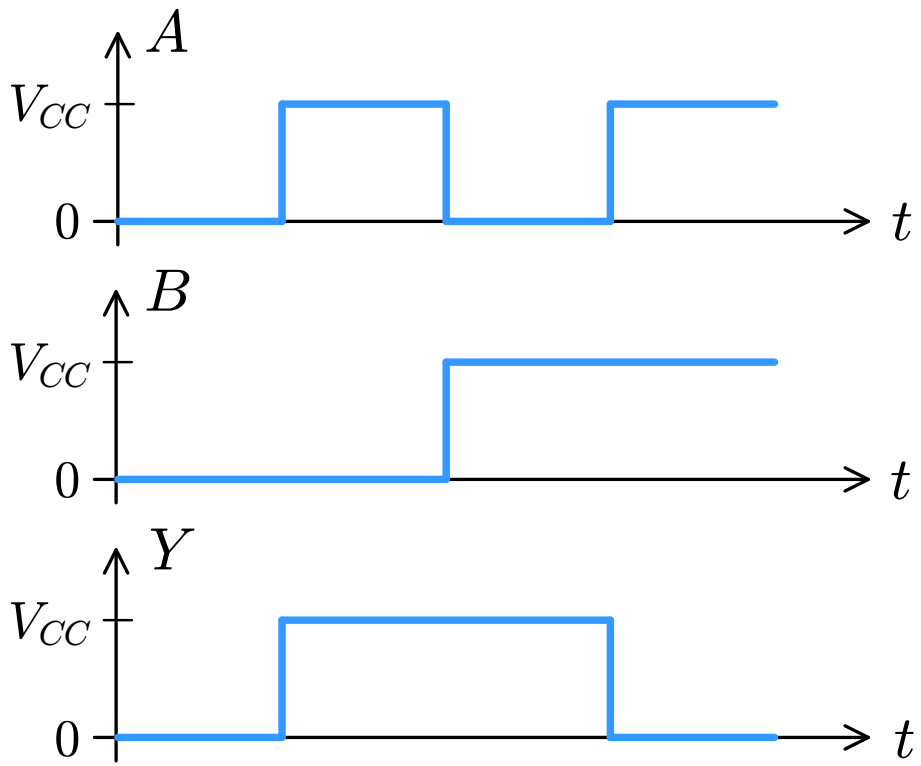
- Uma função lógica frequentemente usada em circuitos lógicos é o **OU Exclusivo** (*Exclusive OR* - XOR). A Porta Lógica **XOR** realiza eletricamente essa função.

Porta Lógica XOR

$$Y = A \oplus B$$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

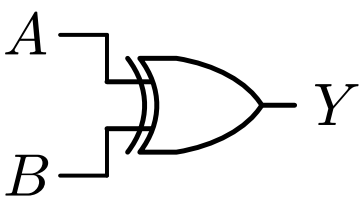


Porta Lógica XOR

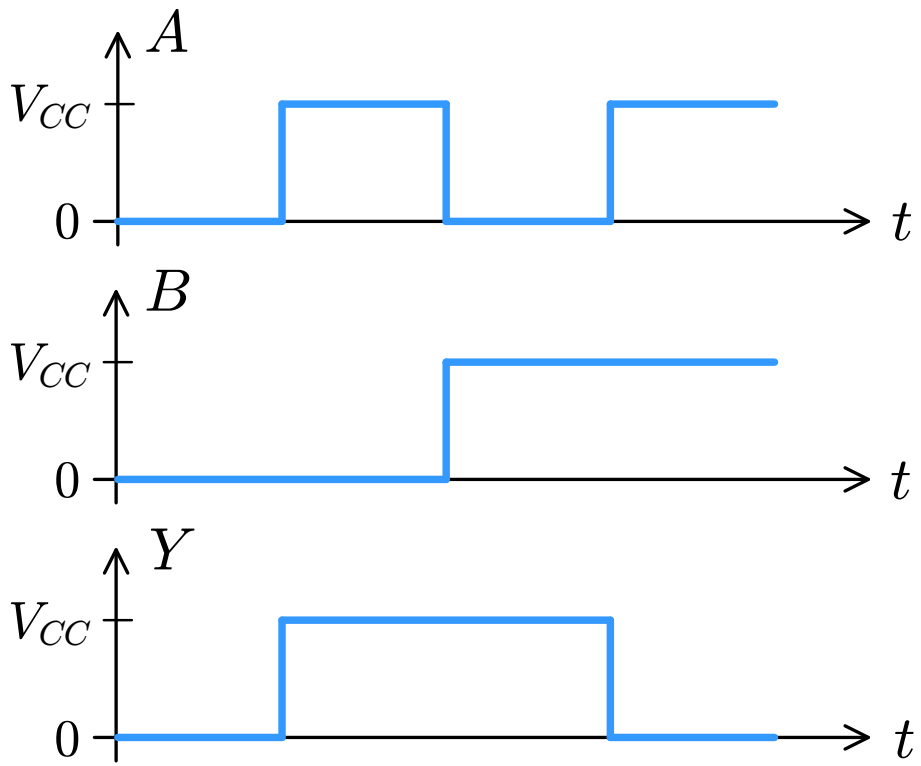
- Uma função lógica frequentemente usada em circuitos lógicos é o **OU Exclusivo** (*Exclusive OR* - XOR). A Porta Lógica **XOR** realiza eletricamente essa função.

Porta Lógica XOR

$$Y = A \oplus B$$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



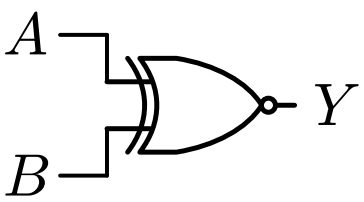
OBSERVAÇÃO: A função XOR pode ser expressa em função das operações lógicas básicas da Álgebra Booleana (NOT, AND e OR) da seguinte forma: $A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$.

Porta Lógica XNOR

- Da mesma forma que as portas lógicas NAND e NOR, a porta lógica **XNOR** (*Exclusive NOR*) é formada pela combinação das operações XOR e NOT.

Porta Lógica XNOR

$$Y = \overline{A \oplus B}$$



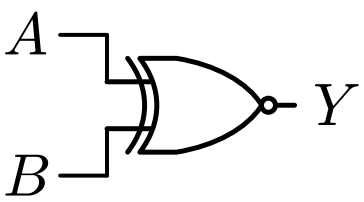
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Porta Lógica XNOR

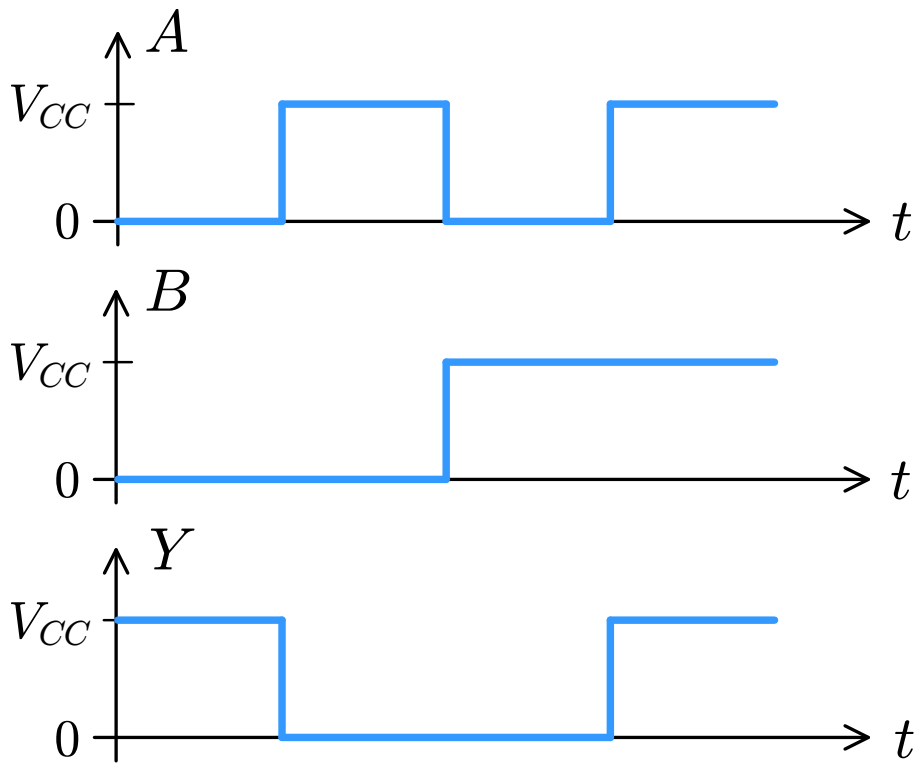
- Da mesma forma que as portas lógicas NAND e NOR, a porta lógica **XNOR** (*Exclusive NOR*) é formada pela combinação das operações XOR e NOT.

Porta Lógica XNOR

$$Y = \overline{A \oplus B}$$



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

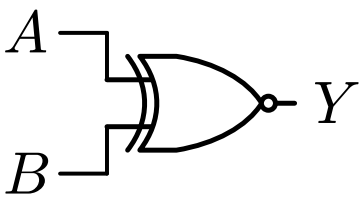


Porta Lógica XNOR

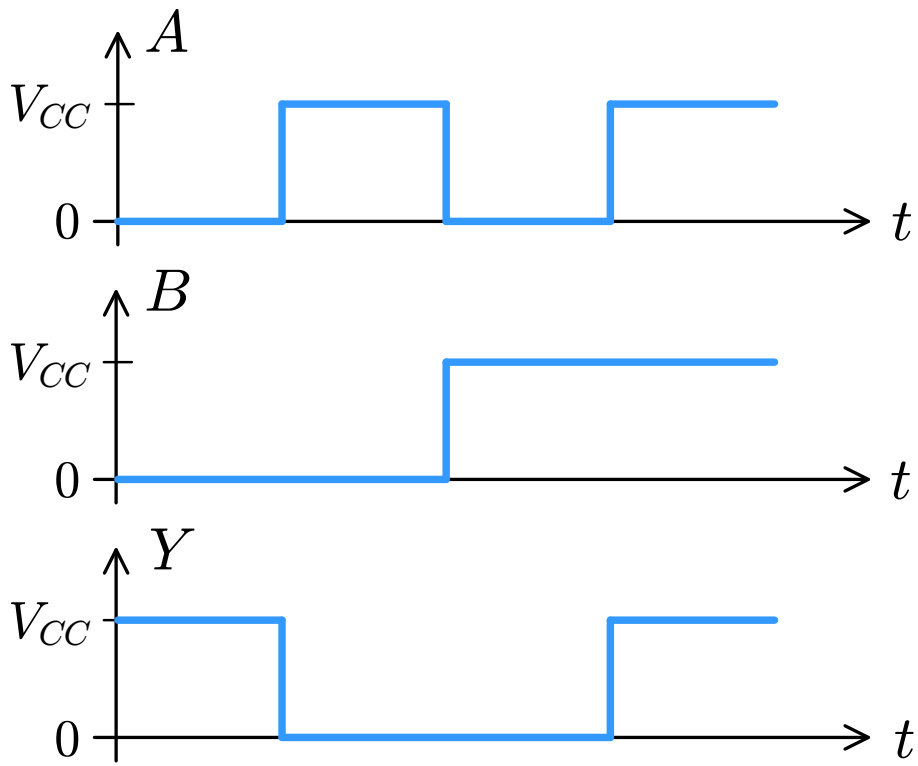
- Da mesma forma que as portas lógicas NAND e NOR, a porta lógica **XNOR** (*Exclusive NOR*) é formada pela combinação das operações XOR e NOT.

Porta Lógica XNOR

$$Y = \overline{A \oplus B}$$



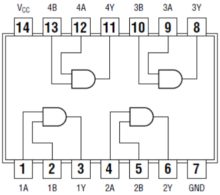
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

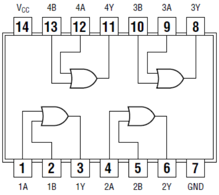


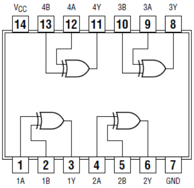
OBSERVAÇÃO: A função XNOR pode ser expressa em função das operações lógicas básicas da Álgebra Booleana (NOT, AND e OR) da seguinte forma: $\overline{A \oplus B} = A \cdot B + \overline{A} \cdot \overline{B}$.

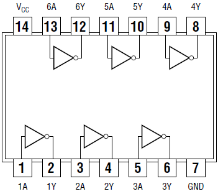
Circuitos Integrados Digitais

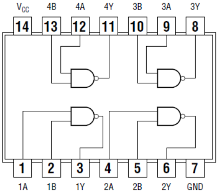
As Portas Lógicas estão disponíveis comercialmente na forma de Circuitos Integrados Digitais. Atualmente, circuitos integrados dedicados podem conter milhões ou até bilhões de portas lógicas, como é o caso dos microprocessadores digitais.

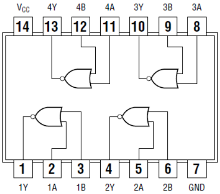


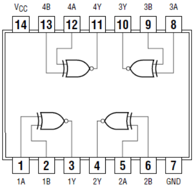


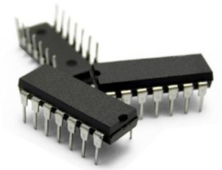












Agenda da Aula - Capítulo 04

- Conceitos Introdutórios
  - Sinais e Circuitos Digitais
  - Sistema de Numeração Binário
  - Álgebra Booleana
- Portas Lógicas Digitais
- **Circuitos Lógicos Combinacionais**
- Células de Memória Digitais

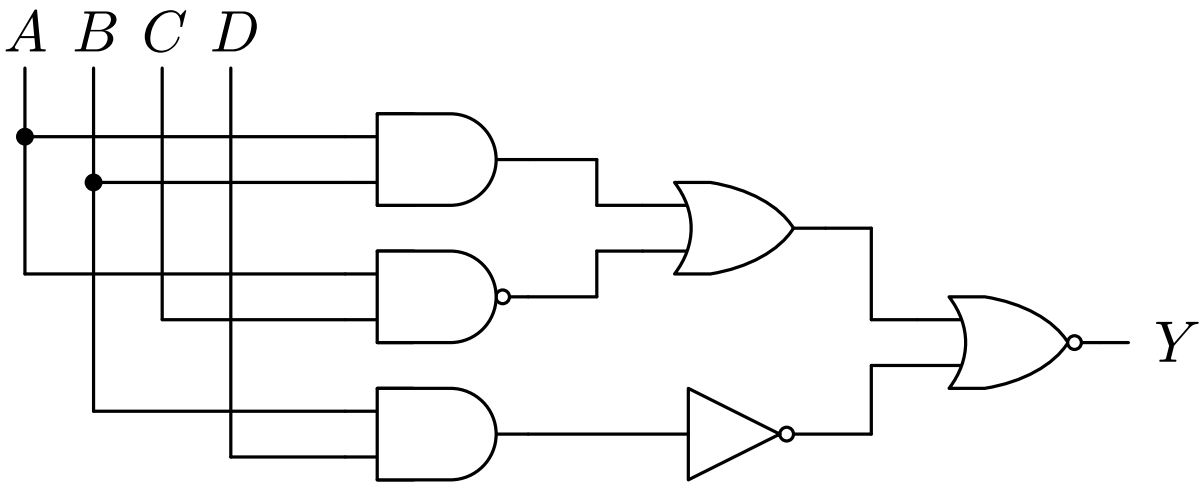
Circuitos Lógicos Combinacionais

Combinando as Portas Lógicas adequadamente, pode-se realizar eletricamente qualquer função lógica digital. Nesses circuitos lógicos, uma resposta digital é produzida para cada combinação possível dos sinais de entrada. Por essa razão, tais circuitos são denominados **Circuitos Lógicos Combinacionais**.

Circuitos Lógicos Combinacionais

Combinando as Portas Lógicas adequadamente, pode-se realizar eletricamente qualquer função lógica digital. Nesses circuitos lógicos, uma resposta digital é produzida para cada combinação possível dos sinais de entrada. Por essa razão, tais circuitos são denominados **Circuitos Lógicos Combinacionais**.

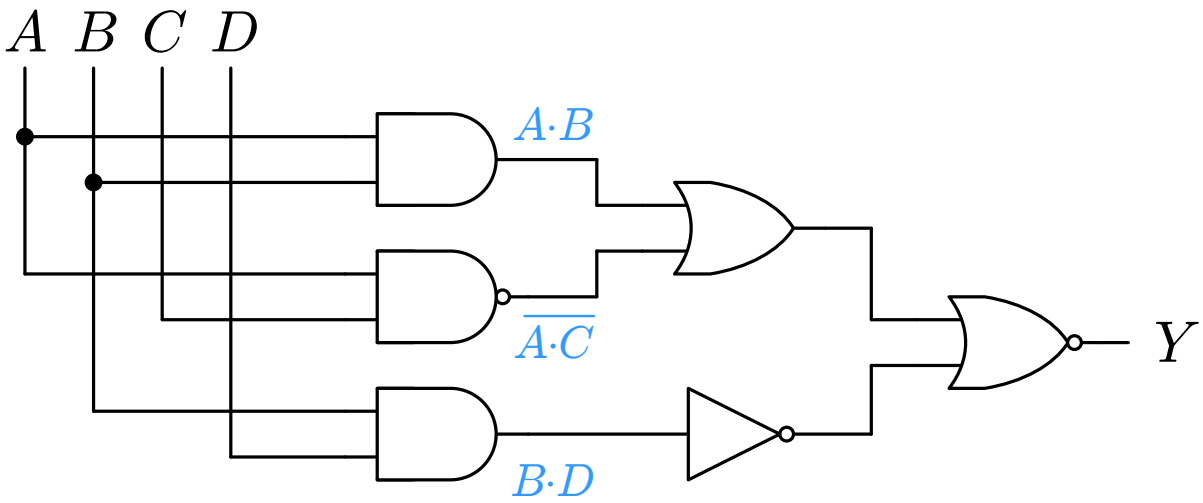
EXEMPLO: Qual a função lógica realizada pelo circuito abaixo?



Circuitos Lógicos Combinacionais

Combinando as Portas Lógicas adequadamente, pode-se realizar eletricamente qualquer função lógica digital. Nesses circuitos lógicos, uma resposta digital é produzida para cada combinação possível dos sinais de entrada. Por essa razão, tais circuitos são denominados **Circuitos Lógicos Combinacionais**.

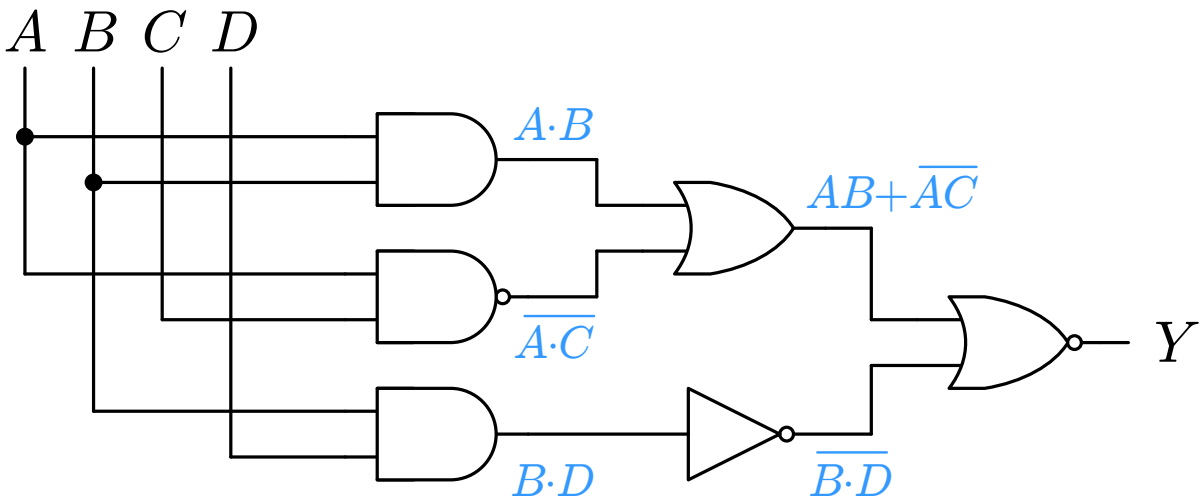
EXEMPLO: Qual a função lógica realizada pelo circuito abaixo?



Circuitos Lógicos Combinacionais

Combinando as Portas Lógicas adequadamente, pode-se realizar eletricamente qualquer função lógica digital. Nesses circuitos lógicos, uma resposta digital é produzida para cada combinação possível dos sinais de entrada. Por essa razão, tais circuitos são denominados **Circuitos Lógicos Combinacionais**.

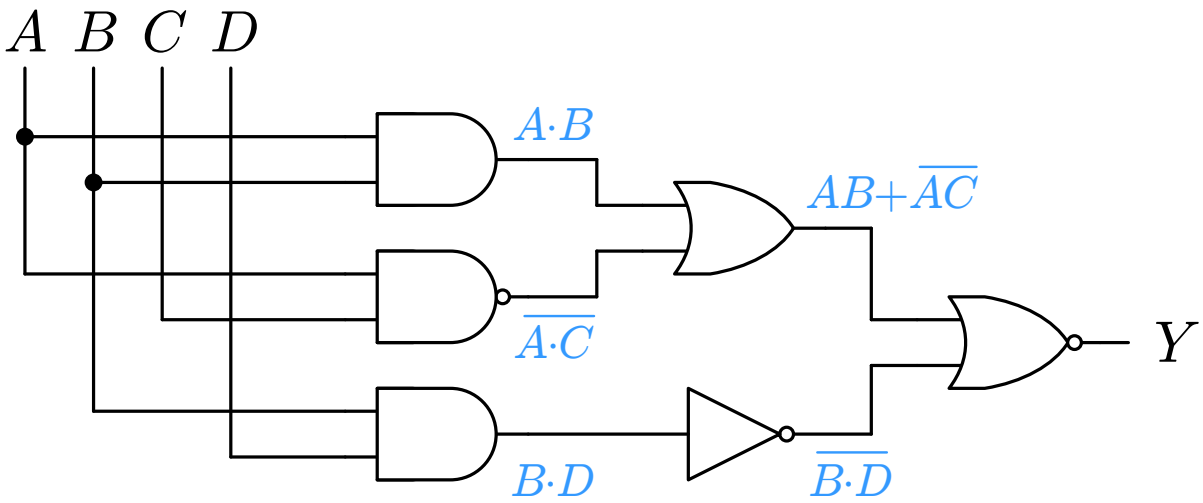
EXEMPLO: Qual a função lógica realizada pelo circuito abaixo?



Circuitos Lógicos Combinacionais

Combinando as Portas Lógicas adequadamente, pode-se realizar eletricamente qualquer função lógica digital. Nesses circuitos lógicos, uma resposta digital é produzida para cada combinação possível dos sinais de entrada. Por essa razão, tais circuitos são denominados **Circuitos Lógicos Combinacionais**.

EXEMPLO: Qual a função lógica realizada pelo circuito abaixo?

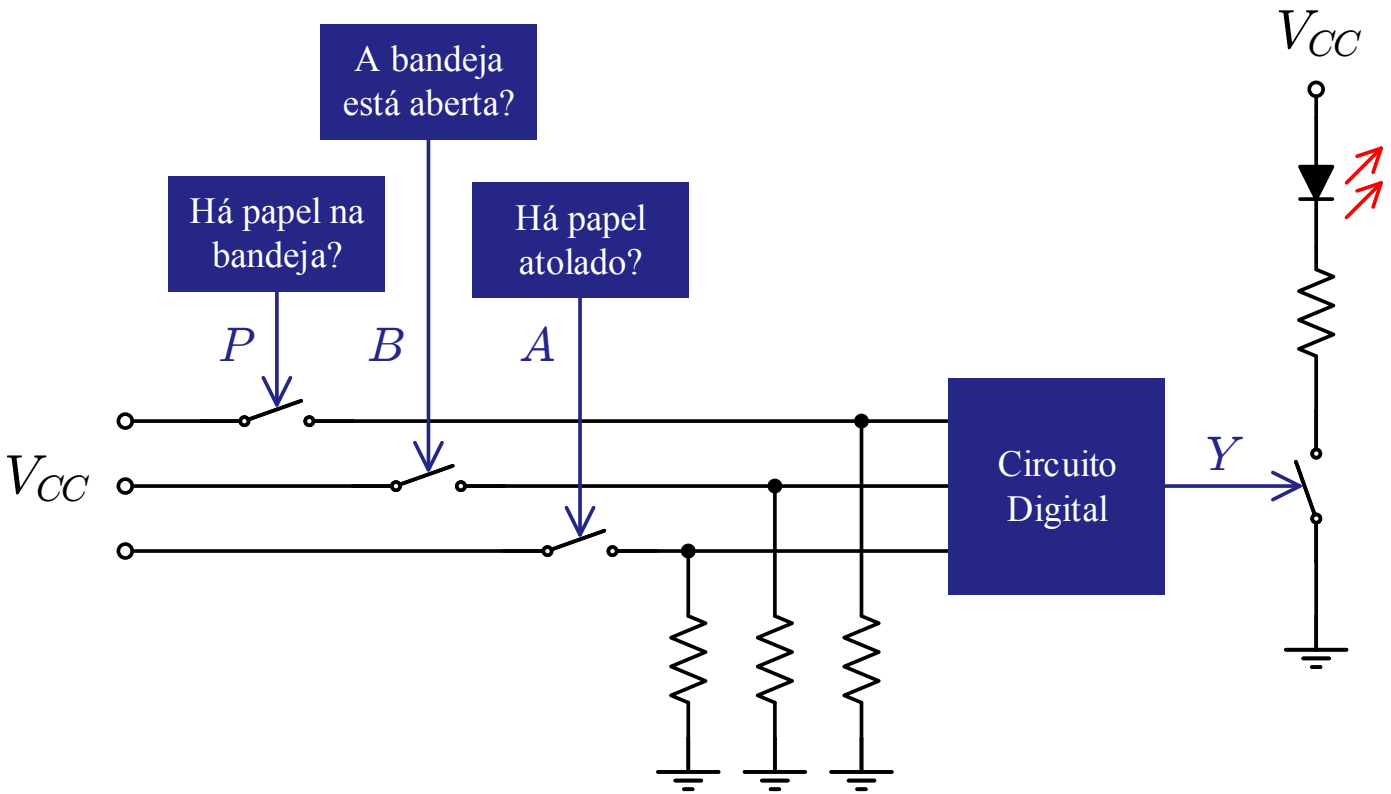


Solução

$$Y = \overline{(A\,B + \overline{A\,C}) + \overline{B\,D}}$$

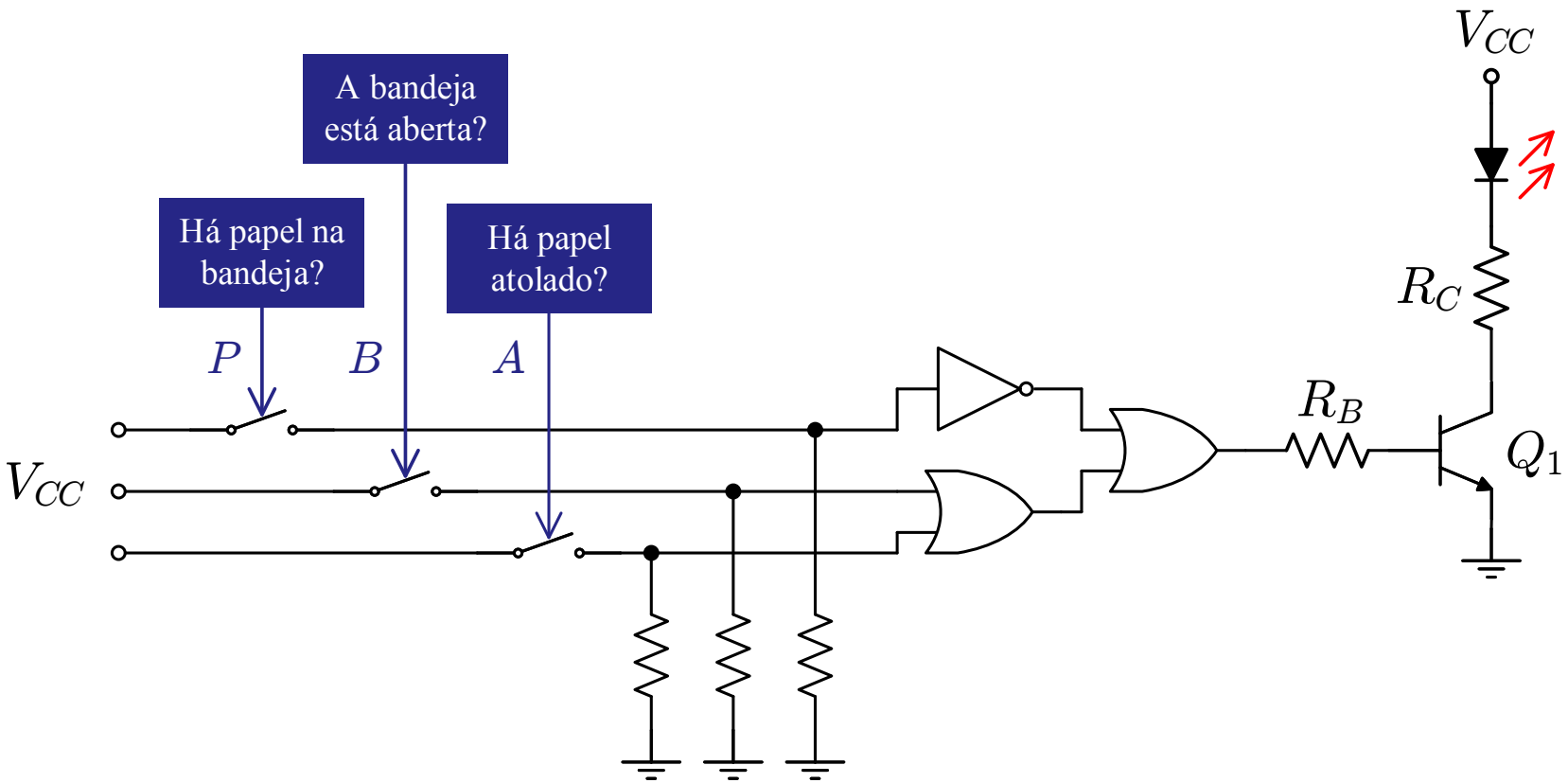
Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Como ficaria o circuito digital que realiza a função lógica $Y = \overline{P} + B + A$ para o sistema de alerta da impressora?



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Como ficaria o circuito digital que realiza a função lógica $Y = \overline{P} + B + A$ para o sistema de alerta da impressora?



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Obtenha o Circuito Lógico Combinacional que é capaz de realizar a função lógica descrita pela Tabela Verdade abaixo.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Obtenha o Circuito Lógico Combinacional que é capaz de realizar a função lógica descrita pela Tabela Verdade abaixo.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

		BC			
		00	01	11	10
A	0	0	0	0	1
	1	1	1	0	1

Y

Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Obtenha o Circuito Lógico Combinacional que é capaz de realizar a função lógica descrita pela Tabela Verdade abaixo.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

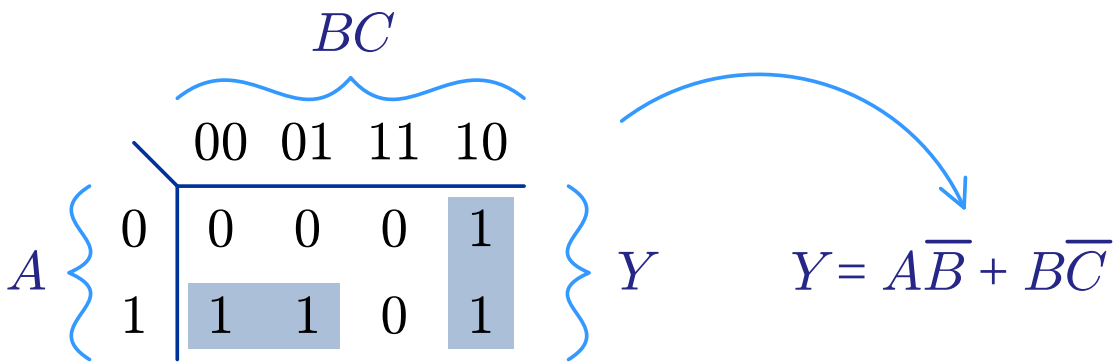
		BC			
		00	01	11	10
A	0	0	0	0	1
	1	1	1	0	1

Y

Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Obtenha o Circuito Lógico Combinacional que é capaz de realizar a função lógica descrita pela Tabela Verdade abaixo.

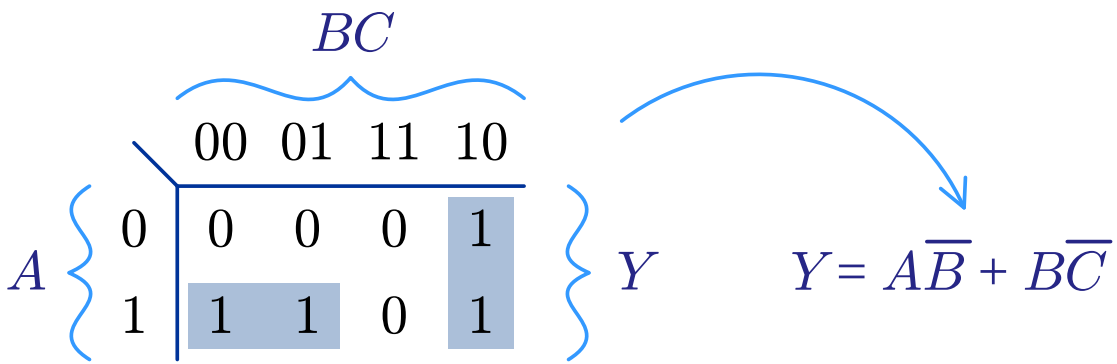
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

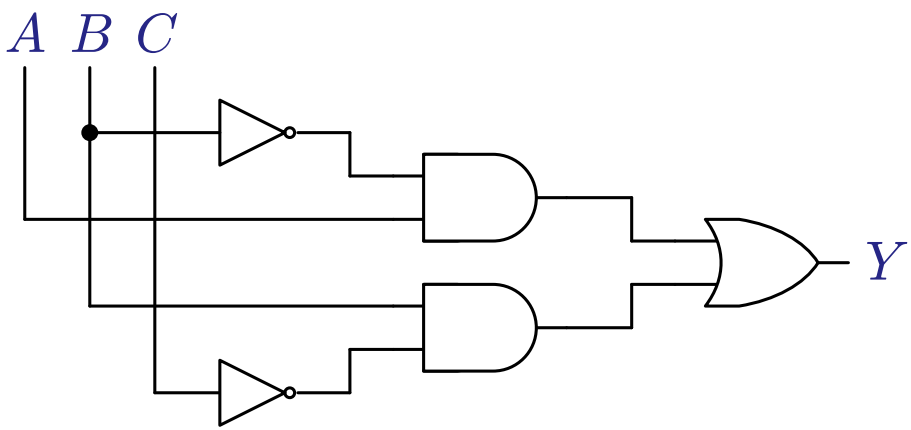


Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Obtenha o Circuito Lógico Combinacional que é capaz de realizar a função lógica descrita pela Tabela Verdade abaixo.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

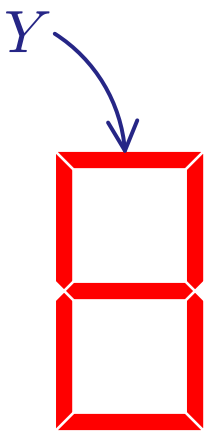




Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

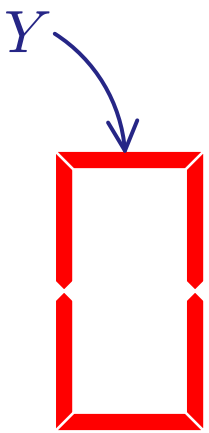
A_3	A_2	A_1	A_0	Y
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

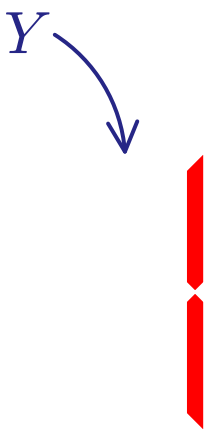
A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

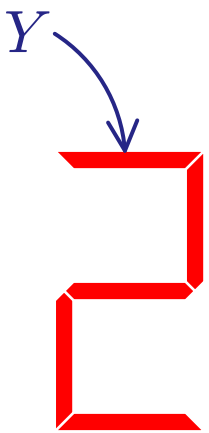
A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

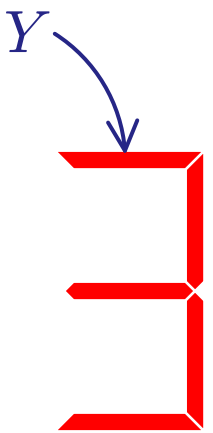
A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

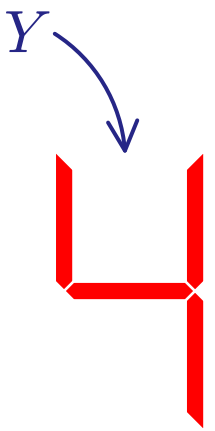
A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

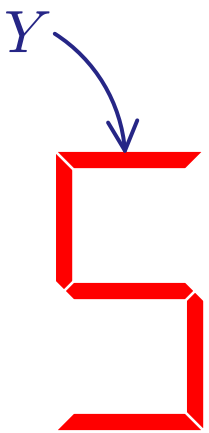
A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

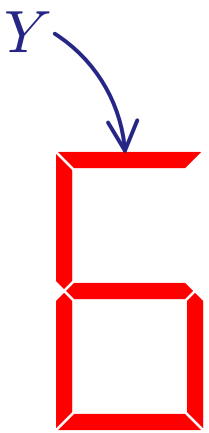
A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

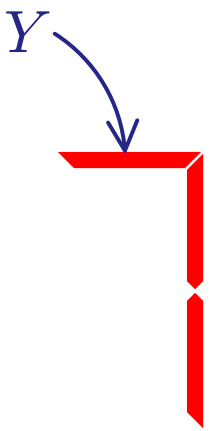
A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	
1	0	0	0	
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

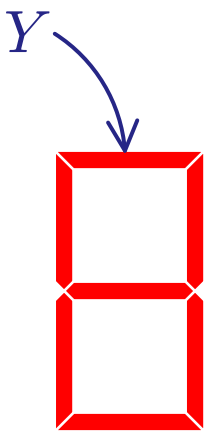
A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

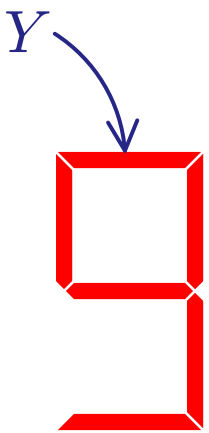
A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1



Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1

Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

A_3A_2 \ A_1A_0	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Y

Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

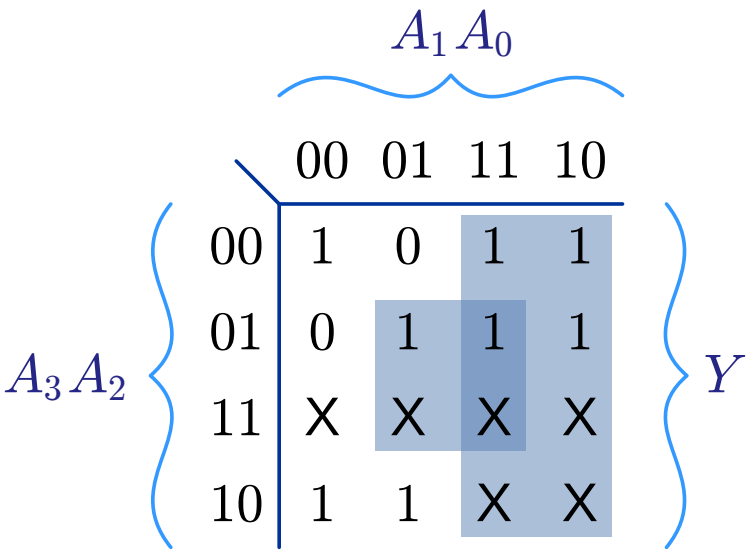
A_3A_2 \ A_1A_0	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Y

$$Y = A_2A_0$$

Circuitos Lógicos Combinacionais - Exercício

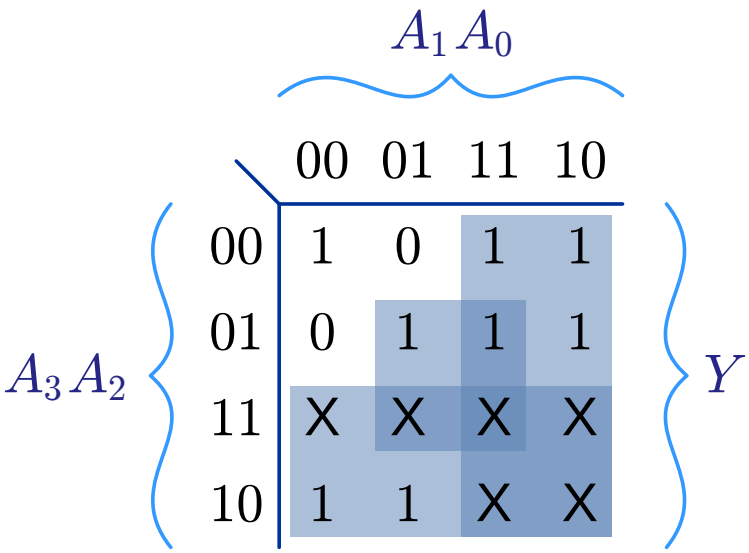
EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.



$$Y = A_2A_0 + A_1$$

Circuitos Lógicos Combinacionais - Exercício

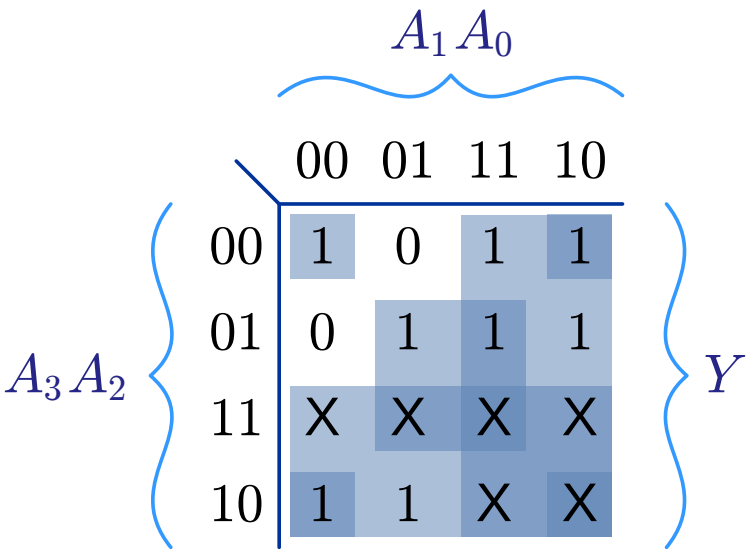
EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.



$$Y = A_2 A_0 + A_1 + A_3$$

Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.



$$Y = A_2A_0 + A_1 + A_3 + \overline{A_2}\,\overline{A_0}$$

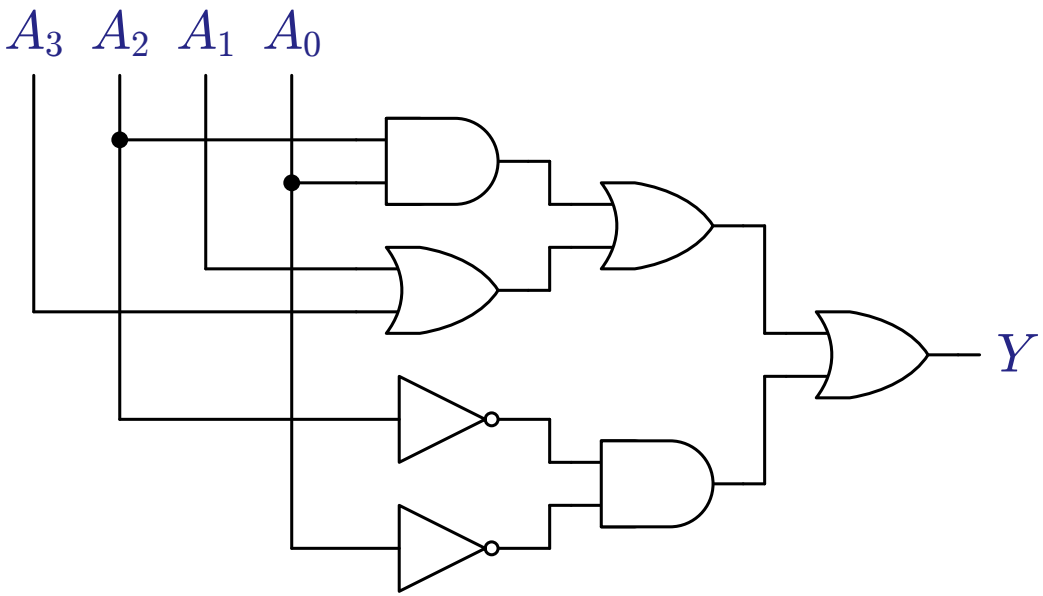
Circuitos Lógicos Combinacionais - Exercício

EXEMPLO: Sintetize o circuito lógico digital que comanda o acendimento do segmento superior do display de sete segmentos, onde o LED do segmento em questão deverá estar aceso quando $Y = 1$ e apagado quando $Y = 0$.

A_3A_2 \ A_1A_0	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Y

$$Y = A_2A_0 + A_1 + A_3 + \overline{A_2}\,\overline{A_0}$$



Agenda da Aula - Capítulo 05

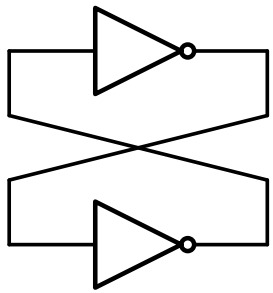
- Conceitos Introdutórios
  - Sinais e Circuitos Digitais
  - Sistema de Numeração Binário
  - Álgebra Booleana
- Portas Lógicas Digitais
- Circuitos Lógicos Combinacionais
- **Células de Memória Digitais**

Células de Memória Digitais

- Nos Circuitos Lógicos Combinacionais, os níveis lógicos obtidos na saída dependem exclusivamente dos níveis lógicos aplicados à entrada no mesmo instante de tempo. Ou seja, a resposta de um circuito combinacional não depende dos sinais aplicados à entrada em instantes de tempo anteriores.
- Para que a saída de um circuito digital dependa das entrada aplicadas em instantes de tempo anteriores, é necessário que o circuito seja dotado de memória. Os sistemas digitais mais complexos, como controladores e microprocessadores, são constituídos pela combinação de circuitos combinacionais com elementos de memória.

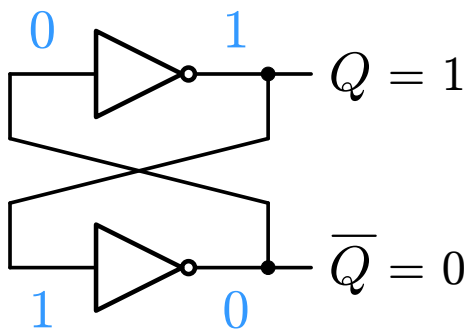
Células de Memória Digitais

- Nos Circuitos Lógicos Combinacionais, os níveis lógicos obtidos na saída dependem exclusivamente dos níveis lógicos aplicados à entrada no mesmo instante de tempo. Ou seja, a resposta de um circuito combinacional não depende dos sinais aplicados à entrada em instantes de tempo anteriores.
- Para que a saída de um circuito digital dependa das entrada aplicadas em instantes de tempo anteriores, é necessário que o circuito seja dotado de memória. Os sistemas digitais mais complexos, como controladores e microprocessadores, são constituídos pela combinação de circuitos combinacionais com elementos de memória.
- As células de memória digitais são baseadas em circuitos que possuem dois estados estáveis (biestáveis), conhecidos como *Latches*. Uma forma de construir um circuito biestável consiste em duas portas inversoras:



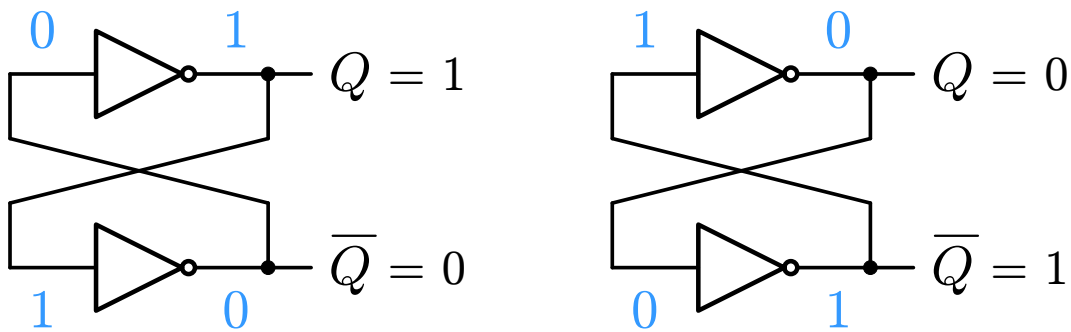
Células de Memória Digitais

- Nos Circuitos Lógicos Combinacionais, os níveis lógicos obtidos na saída dependem exclusivamente dos níveis lógicos aplicados à entrada no mesmo instante de tempo. Ou seja, a resposta de um circuito combinacional não depende dos sinais aplicados à entrada em instantes de tempo anteriores.
- Para que a saída de um circuito digital dependa das entrada aplicadas em instantes de tempo anteriores, é necessário que o circuito seja dotado de memória. Os sistemas digitais mais complexos, como controladores e microprocessadores, são constituídos pela combinação de circuitos combinacionais com elementos de memória.
- As células de memória digitais são baseadas em circuitos que possuem dois estados estáveis (biestáveis), conhecidos como *Latches*. Uma forma de construir um circuito biestável consiste em duas portas inversoras:



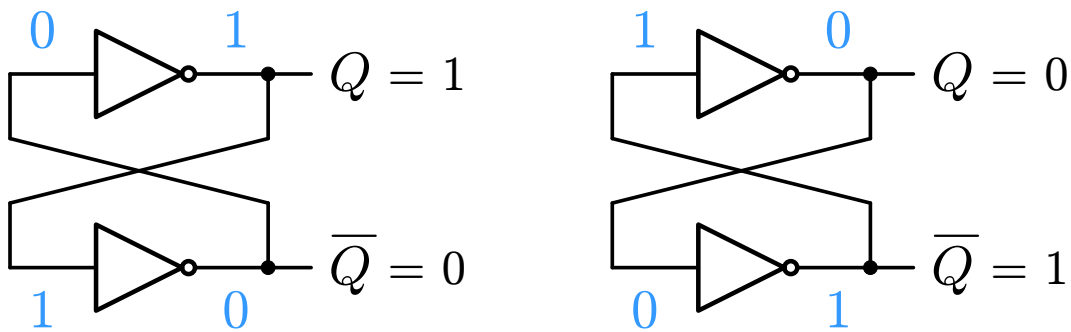
Células de Memória Digitais

- Nos Circuitos Lógicos Combinacionais, os níveis lógicos obtidos na saída dependem exclusivamente dos níveis lógicos aplicados à entrada no mesmo instante de tempo. Ou seja, a resposta de um circuito combinacional não depende dos sinais aplicados à entrada em instantes de tempo anteriores.
- Para que a saída de um circuito digital dependa das entrada aplicadas em instantes de tempo anteriores, é necessário que o circuito seja dotado de memória. Os sistemas digitais mais complexos, como controladores e microprocessadores, são constituídos pela combinação de circuitos combinacionais com elementos de memória.
- As células de memória digitais são baseadas em circuitos que possuem dois estados estáveis (biestáveis), conhecidos como *Latches*. Uma forma de construir um circuito biestável consiste em duas portas inversoras:



Células de Memória Digitais

- Nos Circuitos Lógicos Combinacionais, os níveis lógicos obtidos na saída dependem exclusivamente dos níveis lógicos aplicados à entrada no mesmo instante de tempo. Ou seja, a resposta de um circuito combinacional não depende dos sinais aplicados à entrada em instantes de tempo anteriores.
- Para que a saída de um circuito digital dependa das entrada aplicadas em instantes de tempo anteriores, é necessário que o circuito seja dotado de memória. Os sistemas digitais mais complexos, como controladores e microprocessadores, são constituídos pela combinação de circuitos combinacionais com elementos de memória.
- As células de memória digitais são baseadas em circuitos que possuem dois estados estáveis (biestáveis), conhecidos como *Latches*. Uma forma de construir um circuito biestável consiste em duas portas inversoras:



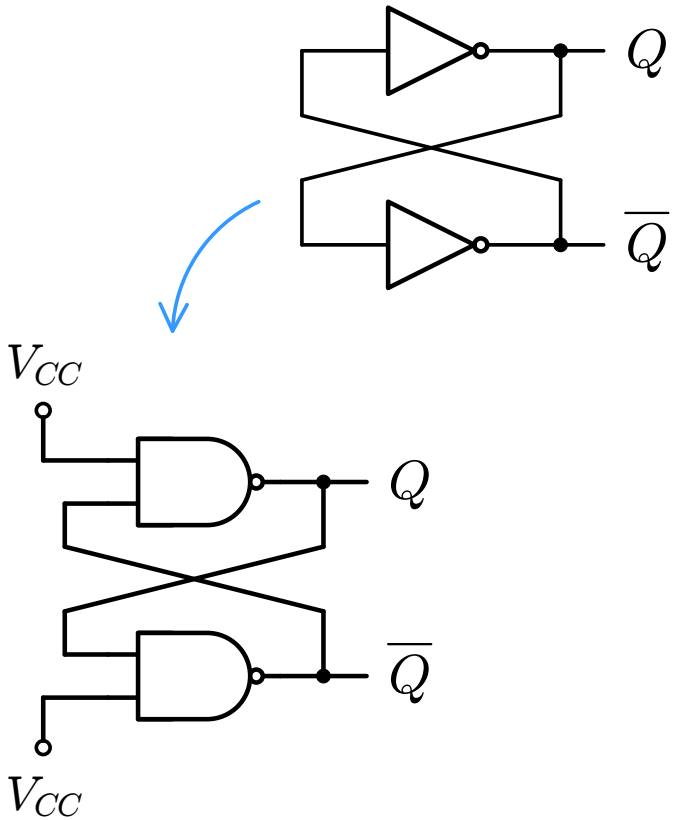
- Entretanto, o circuito biestável acima não possui nenhuma entrada que permita alterar o estado memorizado pelo circuito. Isso pode ser resolvido substituindo-se as portas inversoras por portas NAND ou NOR.

Latch-SR com Portas NAND

- A função lógica NOT pode ser realizada a partir de uma porta NAND, onde uma de suas entradas está sempre conectada a um nível lógico alto. Portanto, o circuito *Latch* biestável pode ser construído da seguinte forma:

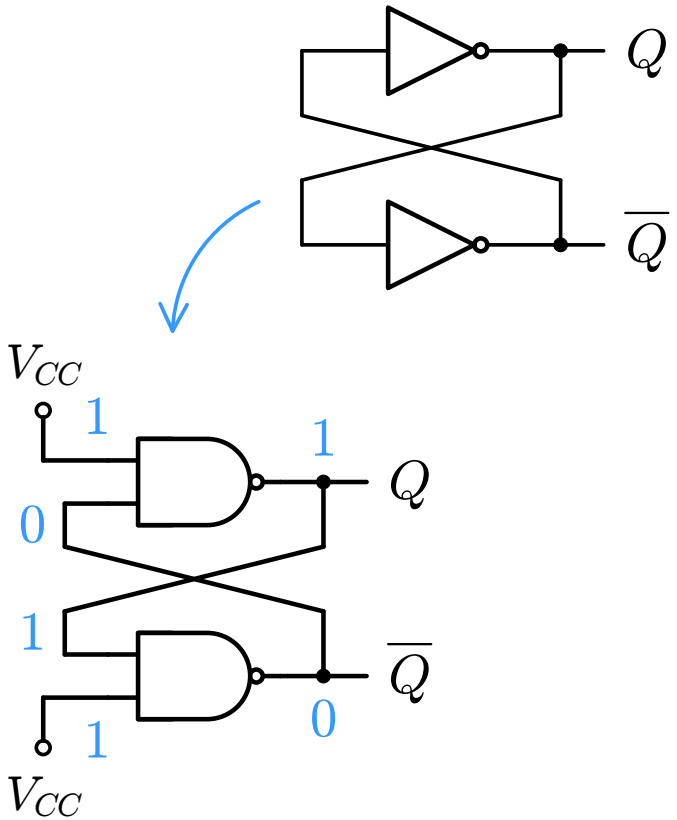
Latch-SR com Portas NAND

- A função lógica NOT pode ser realizada a partir de uma porta NAND, onde uma de suas entradas está sempre conectada a um nível lógico alto. Portanto, o circuito *Latch* biestável pode ser construído da seguinte forma:



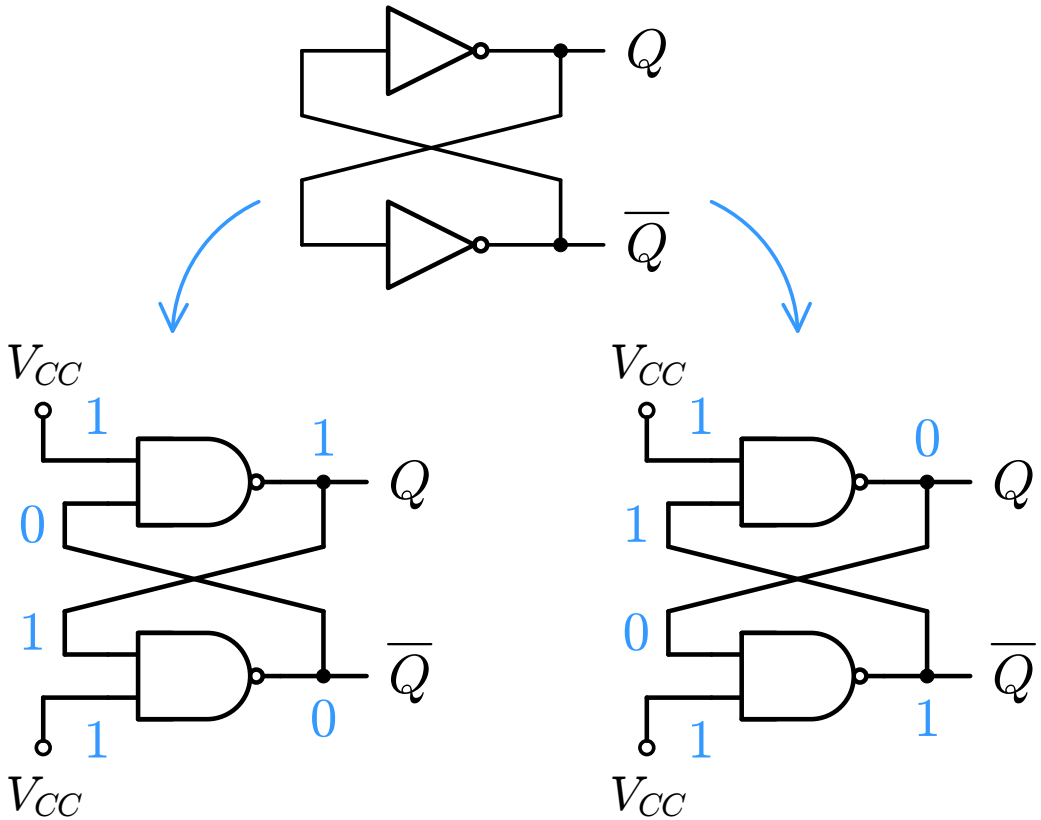
Latch-SR com Portas NAND

- A função lógica NOT pode ser realizada a partir de uma porta NAND, onde uma de suas entradas está sempre conectada a um nível lógico alto. Portanto, o circuito *Latch* biestável pode ser construído da seguinte forma:



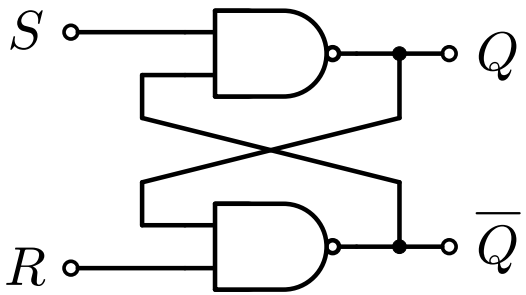
Latch-SR com Portas NAND

- A função lógica NOT pode ser realizada a partir de uma porta NAND, onde uma de suas entradas está sempre conectada a um nível lógico alto. Portanto, o circuito *Latch* biestável pode ser construído da seguinte forma:



Latch-SR com Portas NAND

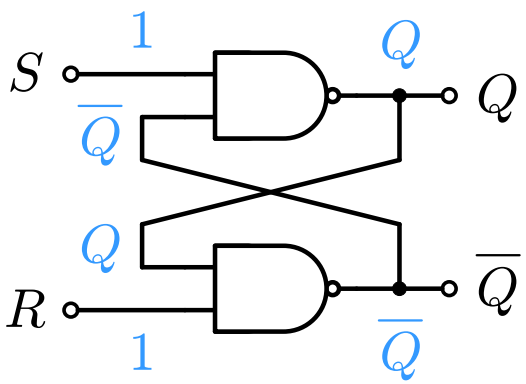
- Usando as entradas das portas NAND que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:



S	R	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Latch-SR com Portas NAND

- Usando as entradas das portas NAND que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:

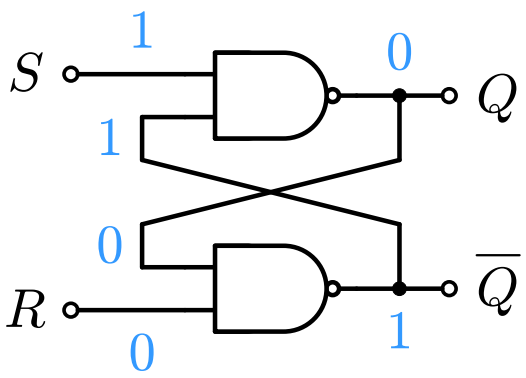


S	R	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1	Q	$\overline{Q}$

→ Memoriza o estado atual

Latch-SR com Portas NAND

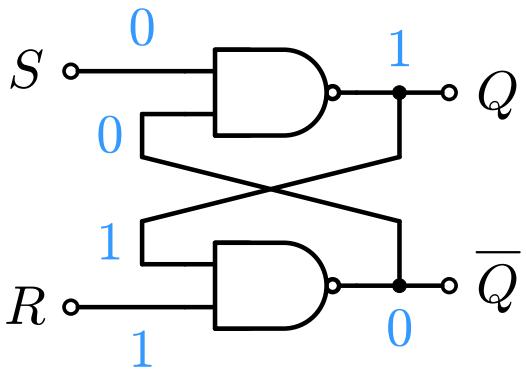
- Usando as entradas das portas NAND que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:



S	R	Q	$\overline{Q}$	
0	0			
0	1			
1	0	0	1	$\longrightarrow$ RESET
1	1	Q	$\overline{Q}$	

Latch-SR com Portas NAND

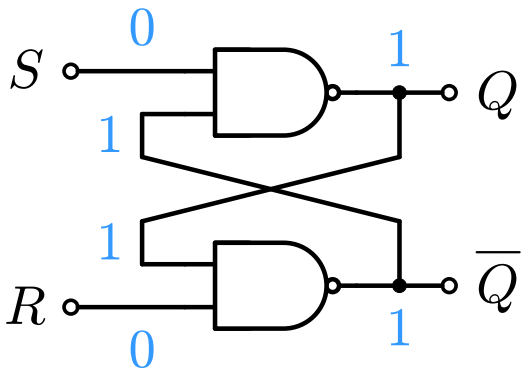
- Usando as entradas das portas NAND que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:



S	R	Q	$\overline{Q}$	
0	0			
0	1	1	0	$\longrightarrow$ SET
1	0	0	1	
1	1	Q	$\overline{Q}$	

Latch-SR com Portas NAND

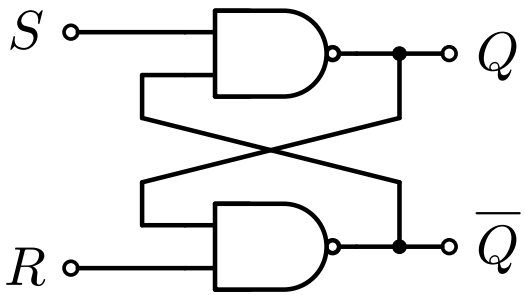
- Usando as entradas das portas NAND que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:



S	R	Q	$\overline{Q}$	
0	0	1	1	→ Estado Inválido
0	1	1	0	
1	0	0	1	
1	1	Q	$\overline{Q}$	

Latch-SR com Portas NAND

- Usando as entradas das portas NAND que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:

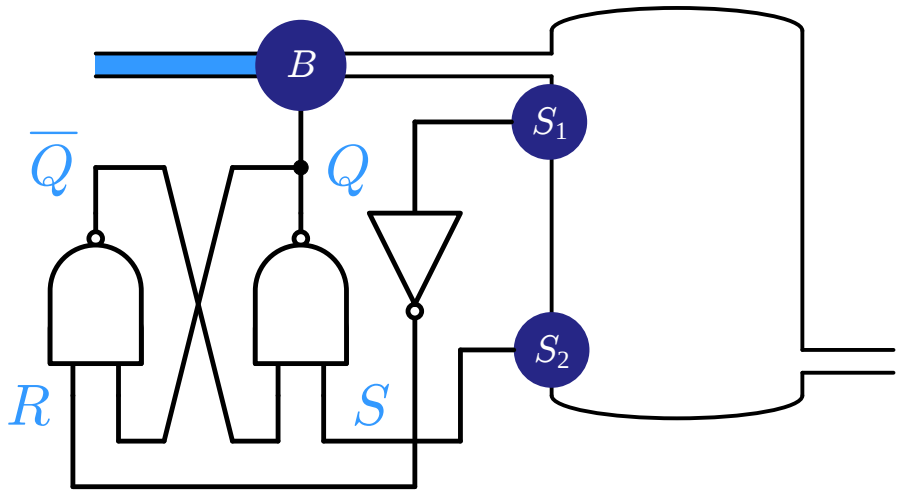


S	R	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\overline{Q}$

- No *Latch*-SR construído com portas NAND, os sinais de controle S (SET) e R (RESET) são ativados em nível lógico baixo. Portanto, durante a operação desse circuito como célula de memória, não devemos colocar ambas as entradas ativas, ou seja, não devemos fazer $S = R = 0$.

Latch-SR com Portas NAND

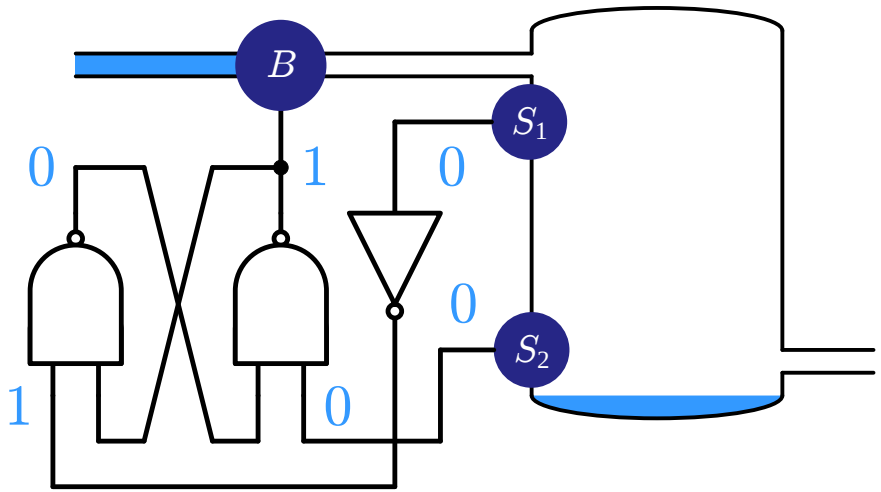
EXEMPLO: No esquema abaixo está ilustrado o sistema de controle do acionamento de uma bomba hidráulica responsável por encher um reservatório de água em uma planta industrial. Nesse reservatório, estão posicionados dois sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário. A partir dos sinais entregues por esses dois sensores, podemos empregar um *Latch*-SR para controlar o acionamento da bomba hidráulica.



S	R	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\overline{Q}$

Latch-SR com Portas NAND

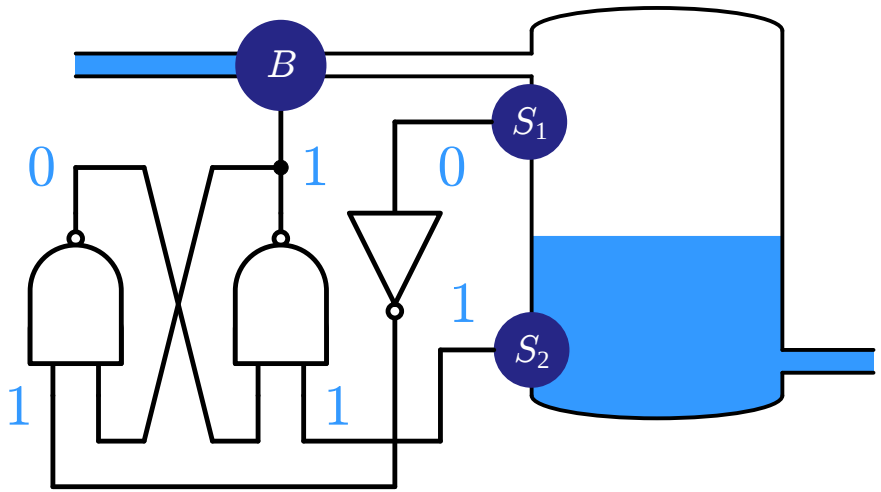
EXEMPLO: No esquema abaixo está ilustrado o sistema de controle do acionamento de uma bomba hidráulica responsável por encher um reservatório de água em uma planta industrial. Nesse reservatório, estão posicionados dois sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário. A partir dos sinais entregues por esses dois sensores, podemos empregar um *Latch*-SR para controlar o acionamento da bomba hidráulica.



S	R	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\overline{Q}$

Latch-SR com Portas NAND

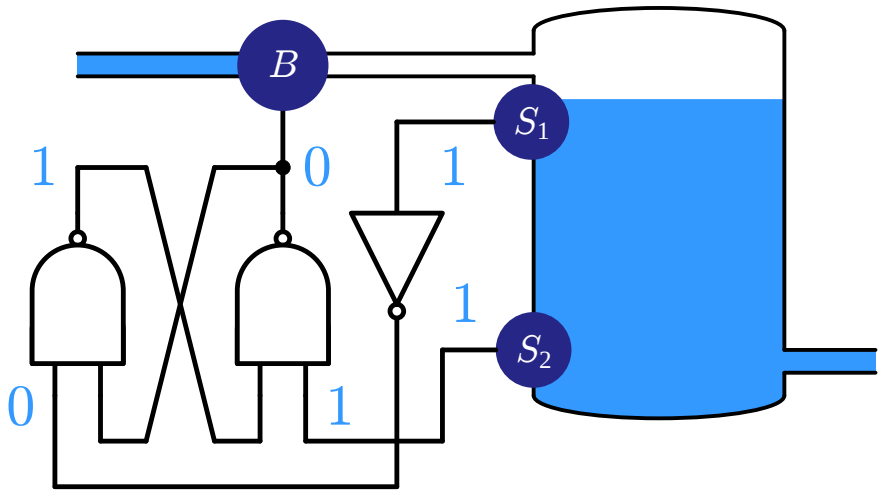
EXEMPLO: No esquema abaixo está ilustrado o sistema de controle do acionamento de uma bomba hidráulica responsável por encher um reservatório de água em uma planta industrial. Nesse reservatório, estão posicionados dois sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário. A partir dos sinais entregues por esses dois sensores, podemos empregar um *Latch*-SR para controlar o acionamento da bomba hidráulica.



S	R	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\overline{Q}$

Latch-SR com Portas NAND

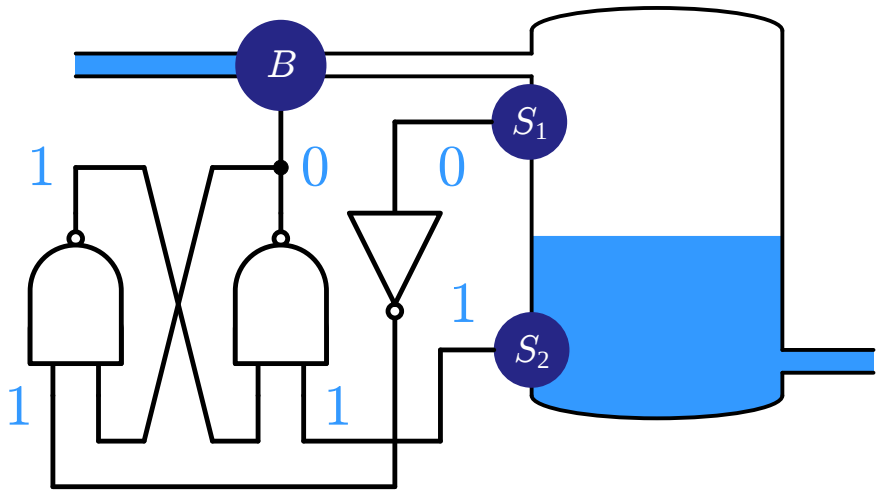
EXEMPLO: No esquema abaixo está ilustrado o sistema de controle do acionamento de uma bomba hidráulica responsável por encher um reservatório de água em uma planta industrial. Nesse reservatório, estão posicionados dois sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário. A partir dos sinais entregues por esses dois sensores, podemos empregar um *Latch*-SR para controlar o acionamento da bomba hidráulica.



S	R	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\overline{Q}$

Latch-SR com Portas NAND

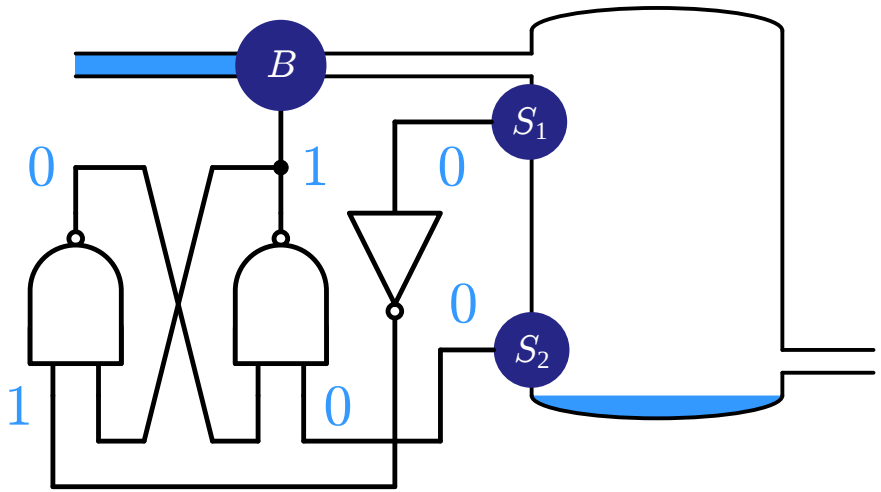
EXEMPLO: No esquema abaixo está ilustrado o sistema de controle do acionamento de uma bomba hidráulica responsável por encher um reservatório de água em uma planta industrial. Nesse reservatório, estão posicionados dois sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário. A partir dos sinais entregues por esses dois sensores, podemos empregar um *Latch*-SR para controlar o acionamento da bomba hidráulica.



S	R	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\overline{Q}$

Latch-SR com Portas NAND

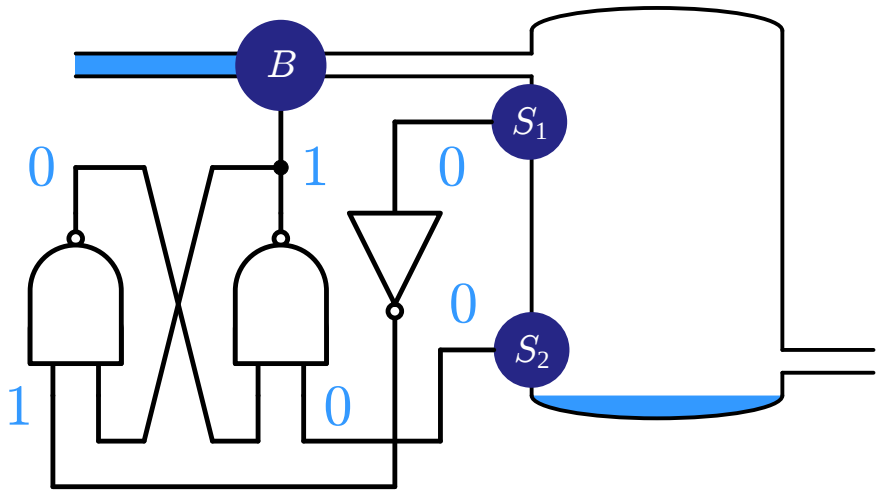
EXEMPLO: No esquema abaixo está ilustrado o sistema de controle do acionamento de uma bomba hidráulica responsável por encher um reservatório de água em uma planta industrial. Nesse reservatório, estão posicionados dois sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário. A partir dos sinais entregues por esses dois sensores, podemos empregar um *Latch*-SR para controlar o acionamento da bomba hidráulica.



S	R	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\overline{Q}$

Latch-SR com Portas NAND

EXEMPLO: No esquema abaixo está ilustrado o sistema de controle do acionamento de uma bomba hidráulica responsável por encher um reservatório de água em uma planta industrial. Nesse reservatório, estão posicionados dois sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário. A partir dos sinais entregues por esses dois sensores, podemos empregar um *Latch*-SR para controlar o acionamento da bomba hidráulica.



S	R	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\overline{Q}$

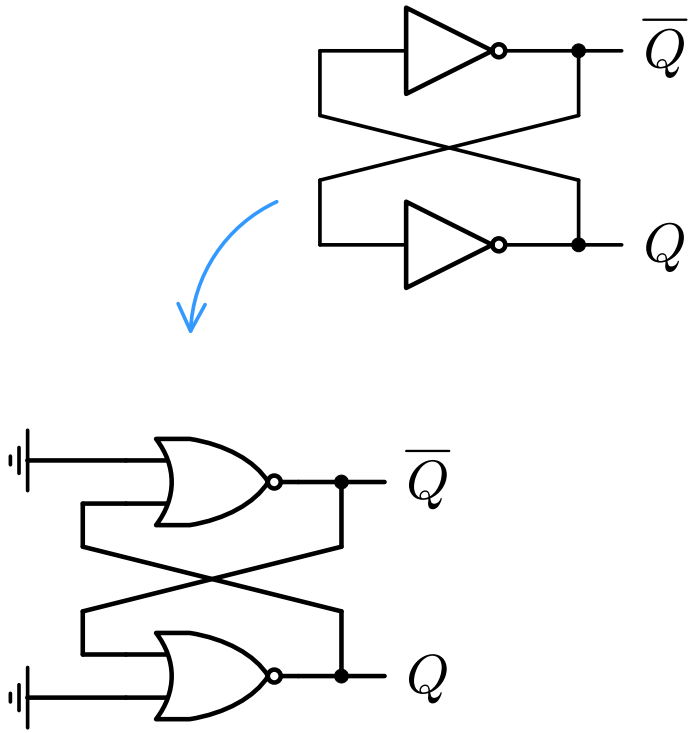
OBSERVAÇÃO: Note que, nesta situação prática, a combinação de entradas inválida ($S = R = 0$) nunca vai acontecer durante a operação do sistema.

Latch-SR com Portas NOR

- A função lógica NOT também pode ser realizada a partir de uma porta NOR, onde uma de suas entradas está sempre conectada a um nível lógico baixo. Portanto, o circuito *Latch* biestável pode ser construído da seguinte forma:

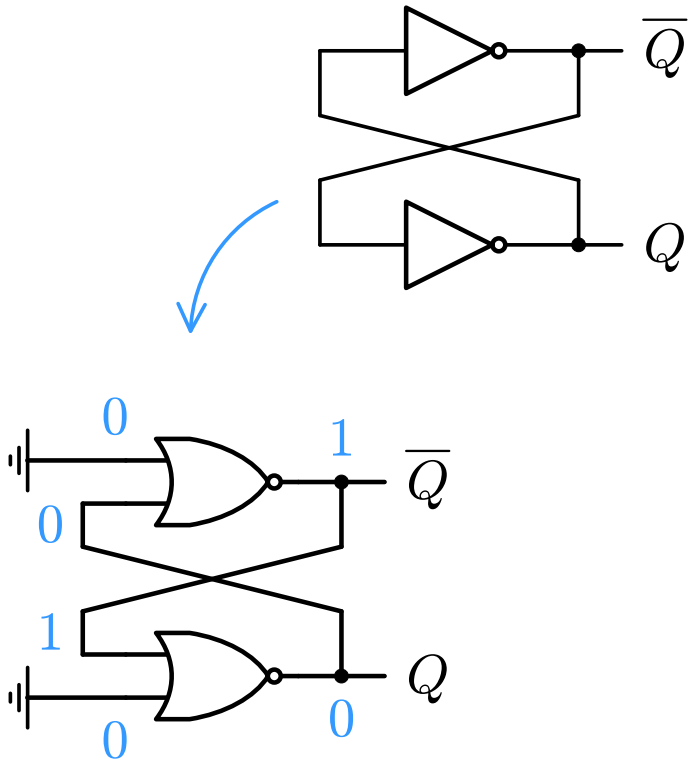
Latch-SR com Portas NOR

- A função lógica NOT também pode ser realizada a partir de uma porta NOR, onde uma de suas entradas está sempre conectada a um nível lógico baixo. Portanto, o circuito *Latch* biestável pode ser construído da seguinte forma:



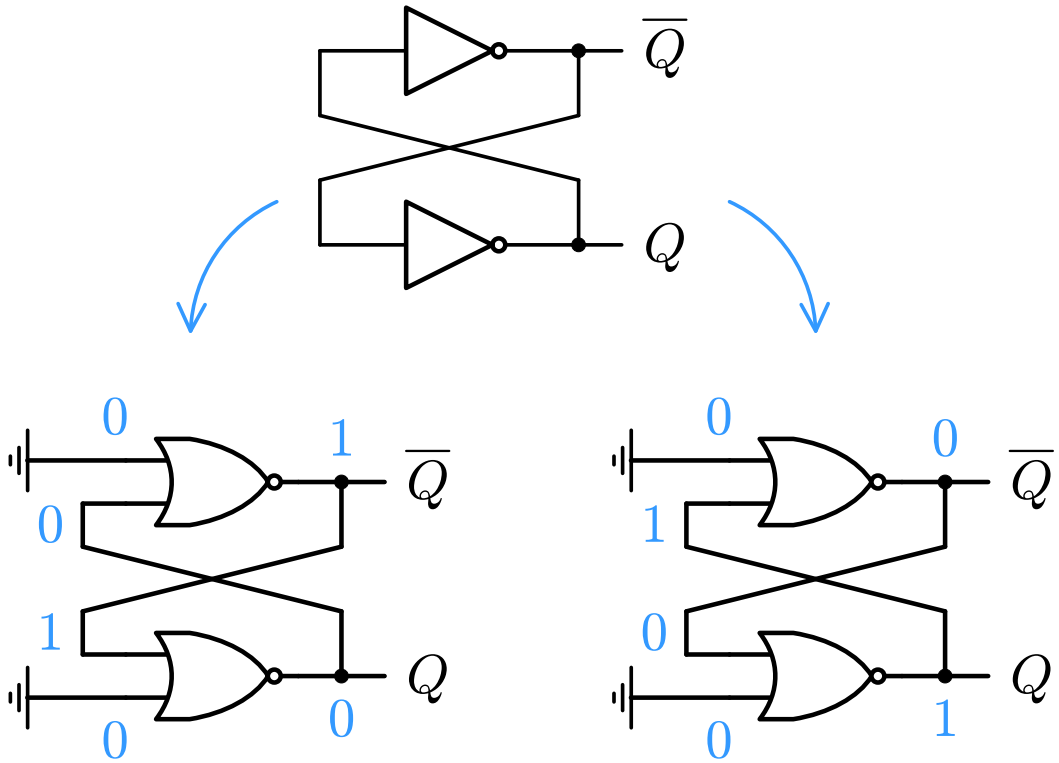
Latch-SR com Portas NOR

- A função lógica NOT também pode ser realizada a partir de uma porta NOR, onde uma de suas entradas está sempre conectada a um nível lógico baixo. Portanto, o circuito *Latch* biestável pode ser construído da seguinte forma:



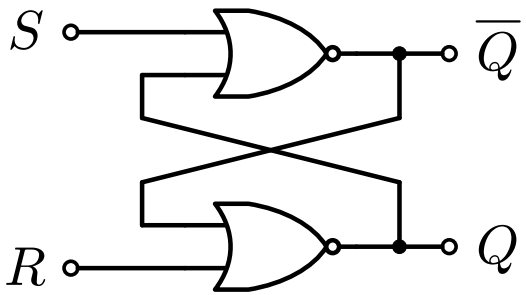
Latch-SR com Portas NOR

- A função lógica NOT também pode ser realizada a partir de uma porta NOR, onde uma de suas entradas está sempre conectada a um nível lógico baixo. Portanto, o circuito *Latch* biestável pode ser construído da seguinte forma:



Latch-SR com Portas NOR

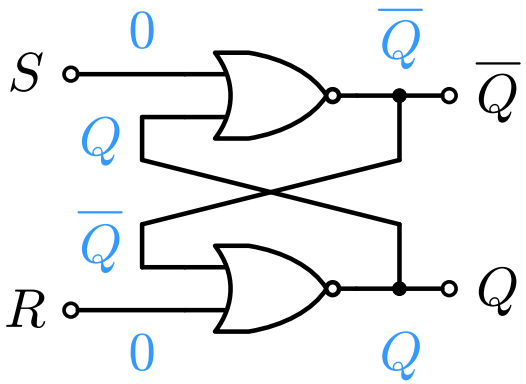
- Usando as entradas das portas NOR que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:



S	R	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Latch-SR com Portas NOR

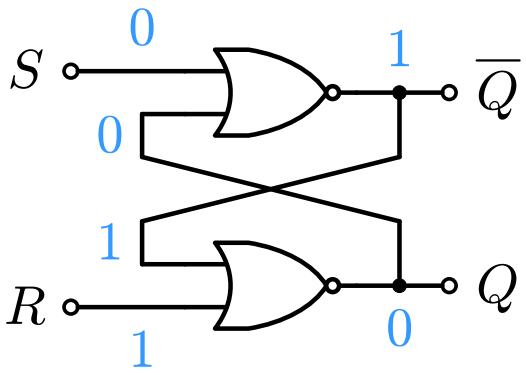
- Usando as entradas das portas NOR que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:



S	R	Q	$\overline{Q}$	
0	0	Q	$\overline{Q}$	→ Memoriza o estado atual
0	1			
1	0			
1	1			

Latch-SR com Portas NOR

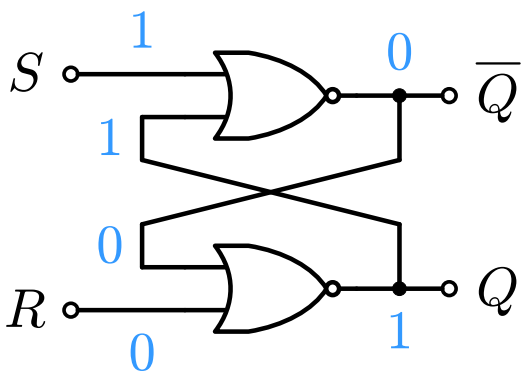
- Usando as entradas das portas NOR que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:



S	R	Q	$\overline{Q}$	
0	0	Q	$\overline{Q}$	
0	1	0	1	$\longrightarrow$ RESET
1	0			
1	1			

Latch-SR com Portas NOR

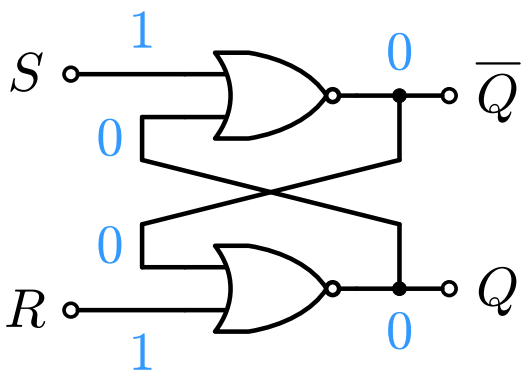
- Usando as entradas das portas NOR que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:



S	R	Q	$\overline{Q}$	
0	0	Q	$\overline{Q}$	
0	1	0	1	
1	0	1	0	$\longrightarrow$ SET
1	1			

Latch-SR com Portas NOR

- Usando as entradas das portas NOR que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:

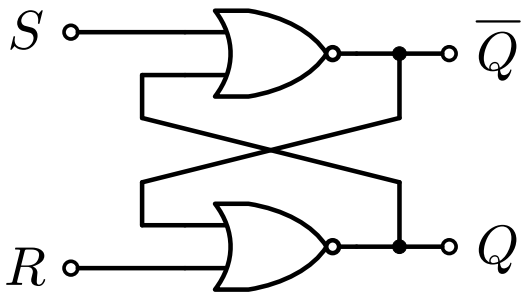


S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

→ Estado Inválido

Latch-SR com Portas NOR

- Usando as entradas das portas NOR que constituem o *Latch*-SR, é possível alterar o estado do circuito através de operações denominadas **SET** e **RESET**:

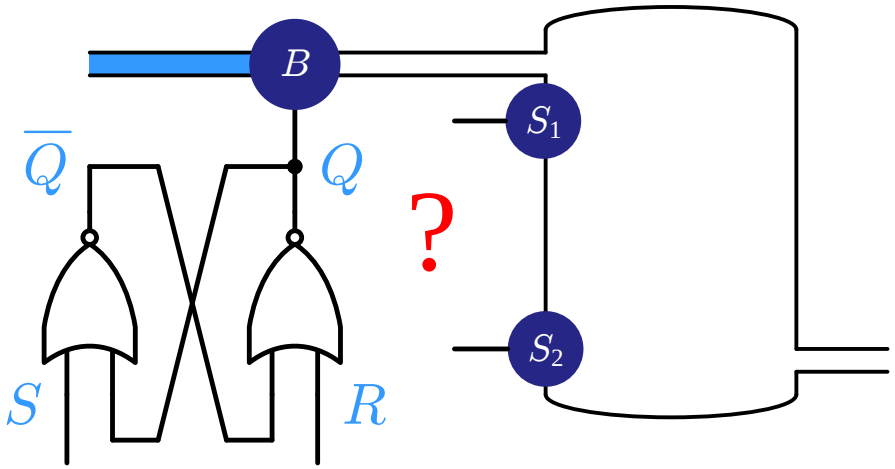


S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

- No *Latch*-SR construído com portas NOR, os sinais de controle S (SET) e R (RESET) são ativados em nível lógico alto. Portanto, durante a operação desse circuito como célula de memória, não devemos colocar ambas as entradas ativas, ou seja, não devemos fazer $S = R = 1$.

Latch-SR com Portas NOR

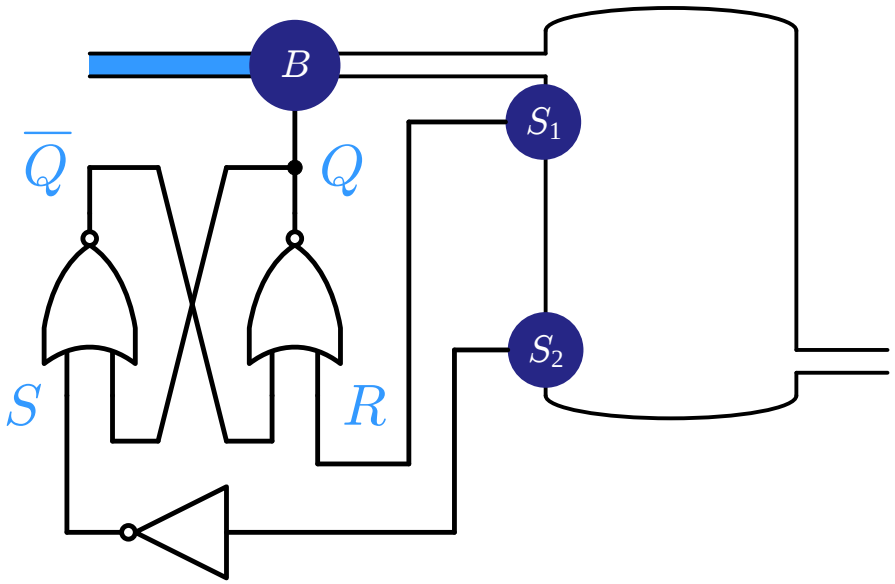
EXEMPLO: Considerando o mesmo sistema de controle do acionamento de uma bomba hidráulica apresentado anteriormente, como ficaria o circuito de controle digital utilizando um *Latch*-SR com portas NOR? Lembrando que os sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário.



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

Latch-SR com Portas NOR

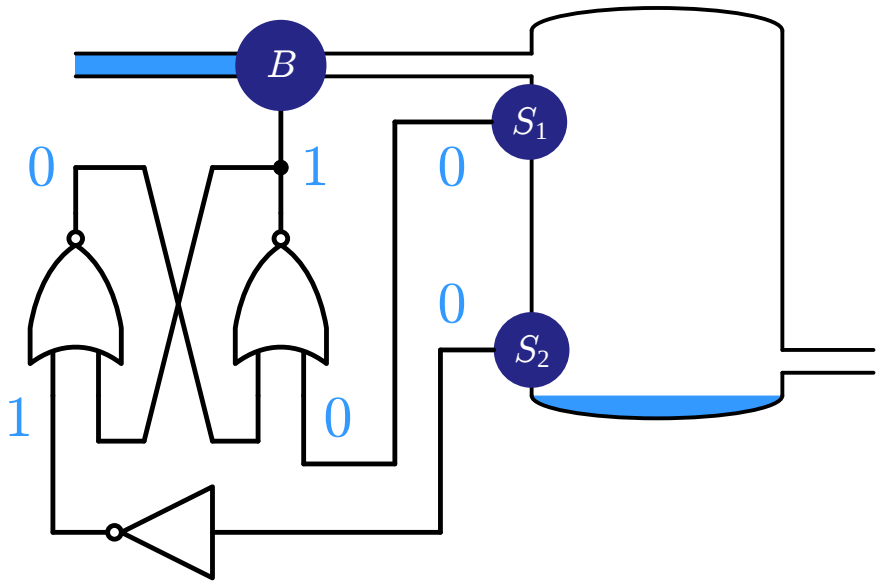
EXEMPLO: Considerando o mesmo sistema de controle do acionamento de uma bomba hidráulica apresentado anteriormente, como ficaria o circuito de controle digital utilizando um *Latch*-SR com portas NOR? Lembrando que os sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário.



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

Latch-SR com Portas NOR

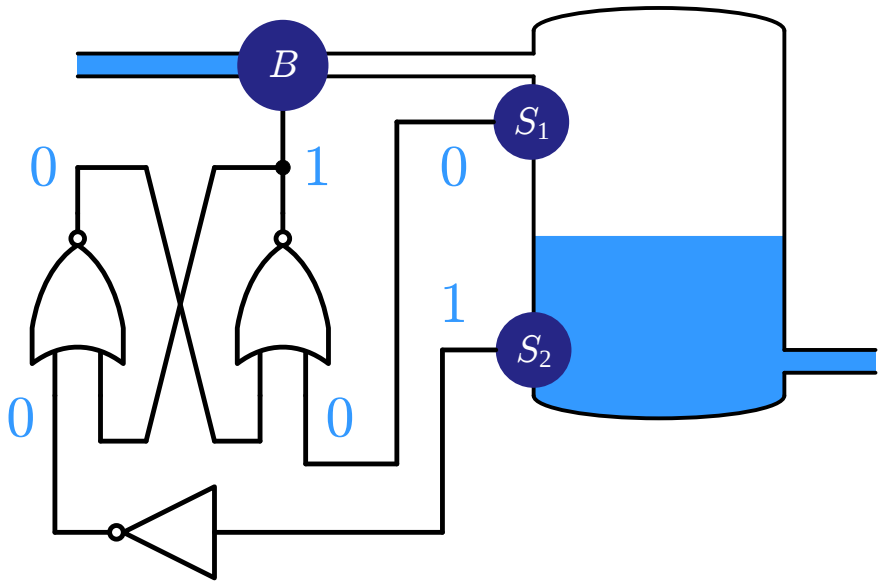
EXEMPLO: Considerando o mesmo sistema de controle do acionamento de uma bomba hidráulica apresentado anteriormente, como ficaria o circuito de controle digital utilizando um *Latch*-SR com portas NOR? Lembrando que os sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário.



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

Latch-SR com Portas NOR

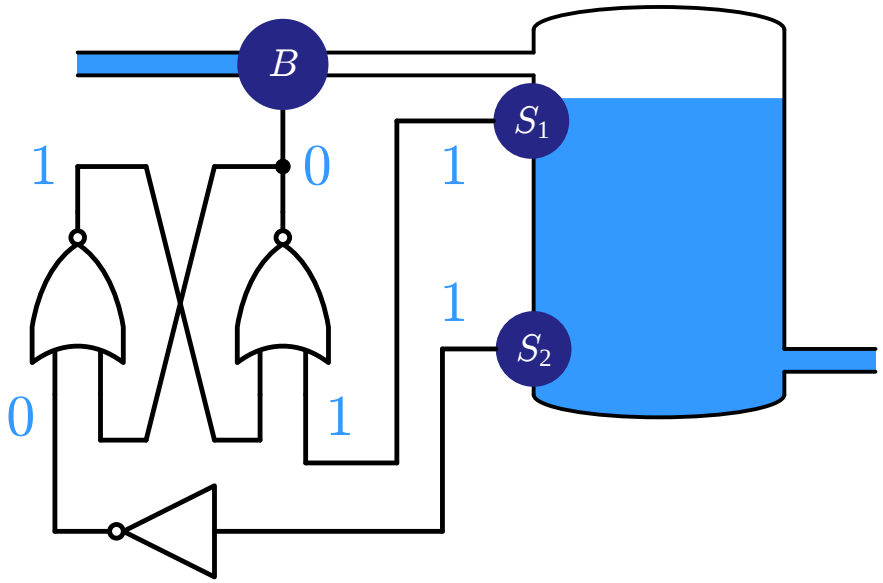
EXEMPLO: Considerando o mesmo sistema de controle do acionamento de uma bomba hidráulica apresentado anteriormente, como ficaria o circuito de controle digital utilizando um *Latch*-SR com portas NOR? Lembrando que os sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário.



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

Latch-SR com Portas NOR

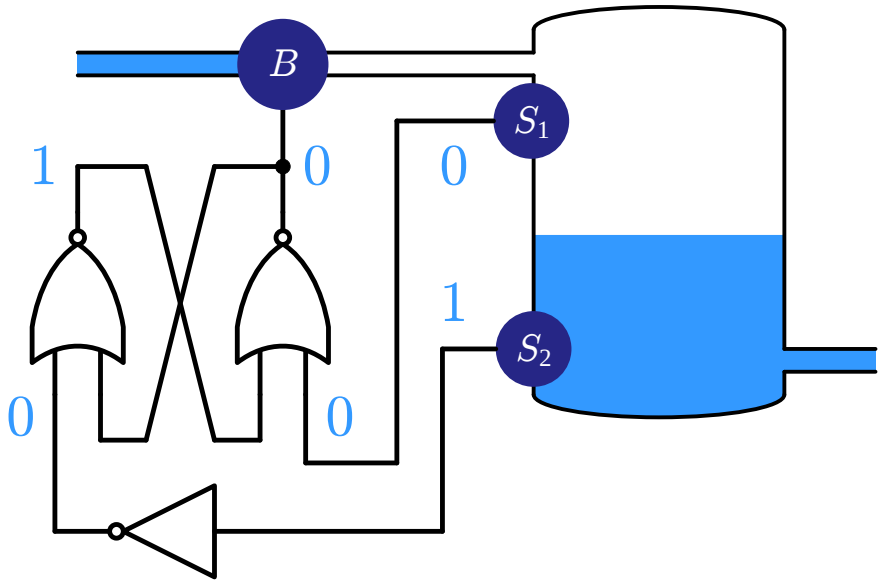
EXEMPLO: Considerando o mesmo sistema de controle do acionamento de uma bomba hidráulica apresentado anteriormente, como ficaria o circuito de controle digital utilizando um *Latch*-SR com portas NOR? Lembrando que os sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário.



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

Latch-SR com Portas NOR

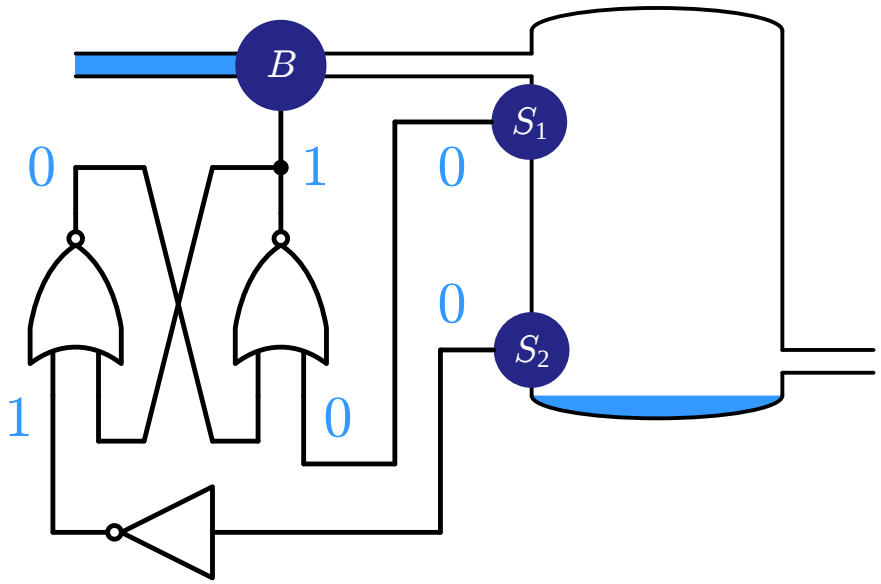
EXEMPLO: Considerando o mesmo sistema de controle do acionamento de uma bomba hidráulica apresentado anteriormente, como ficaria o circuito de controle digital utilizando um *Latch*-SR com portas NOR? Lembrando que os sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário.



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

Latch-SR com Portas NOR

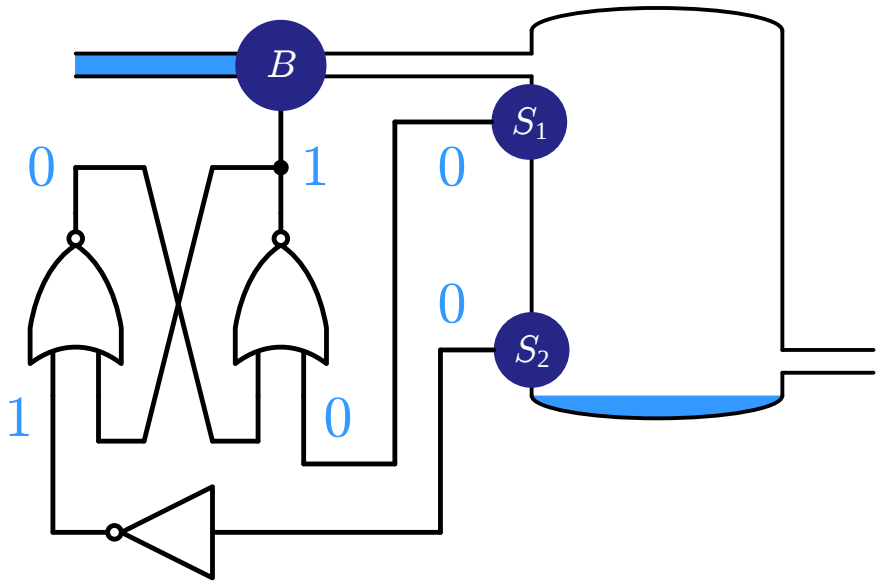
EXEMPLO: Considerando o mesmo sistema de controle do acionamento de uma bomba hidráulica apresentado anteriormente, como ficaria o circuito de controle digital utilizando um *Latch*-SR com portas NOR? Lembrando que os sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário.



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

Latch-SR com Portas NOR

EXEMPLO: Considerando o mesmo sistema de controle do acionamento de uma bomba hidráulica apresentado anteriormente, como ficaria o circuito de controle digital utilizando um *Latch*-SR com portas NOR? Lembrando que os sensores S_1 e S_2 que entregam um sinal em nível lógico alto quando eles estão em contato direto com a água e um sinal de nível lógico baixo caso contrário.



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

OBSERVAÇÃO: Novamente, percebemos que a combinação de entradas inválida ($S = R = 1$) nunca vai acontecer durante a operação do sistema.