



Laboratório de Pesquisa em Redes e Multimídia

# Projeto de Circuitos Lógicos

**Introdução ao Computador 2008/01**  
**Bernardo Gonçalves**



Universidade Federal do Espírito Santo  
Departamento de Informática

## Sumário

- Da Álgebra de Boole ao projeto de circuitos digitais;
- Portas lógicas;
- Equivalência de circuitos;
- Construindo circuitos com portas lógicas;
- Projeto de circuitos lógicos;

## Da Álgebra de Boole ao projeto de circuitos digitais

- Em 1937, Claude Shannon notou a similaridade entre álgebra booleana e circuitos de chaveamento de telefone;
- A aplicação da álgebra booleana a sistemas elétricos foi tema de sua tese de mestrado no MIT intitulada "*A Symbolic Analysis of Relay and Switching Circuits*";
- A tese basicamente dizia: "if we could someday invent a computing machine, the way to make it think would be to use binary code, by stringing together switches and applying Boole's logic system to the result."

## Da Álgebra de Boole ao projeto de circuitos digitais

- Essa tese, feita por ele aos 21 anos, é considerada a mais importante tese de mestrado do século XX.
- A idéia foi imediatamente colocada em prática no projeto de circuitos de chaveamento de telefones, e é ainda como computadores “pensam” (*Remembering Claude Shannon, 2002*);
- O trabalho de Shannon fundaria a ciência do projeto de circuitos digitais.

# Blocos básicos dos circuitos lógicos



These blocks...



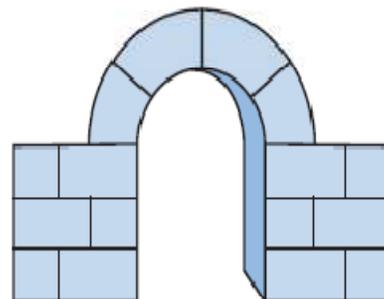
...are hard to work with.



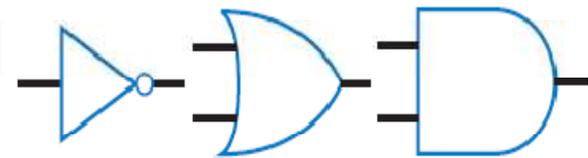
Transistors are hard to work with



The right building blocks...



...enable greater designs.

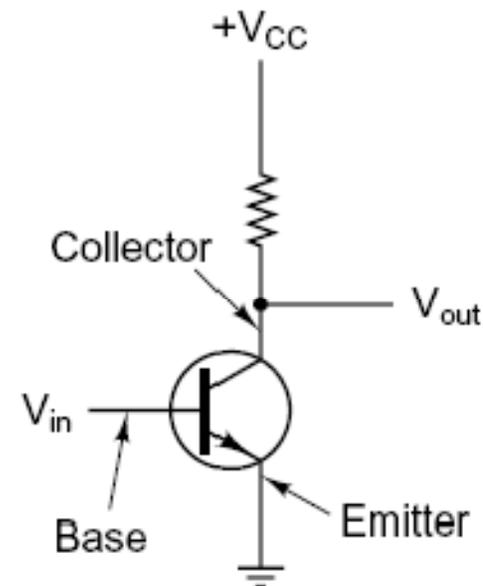


The logic gates that we'll soon introduce enable greater designs

## Portas Lógicas (1)

### ■ Transistor

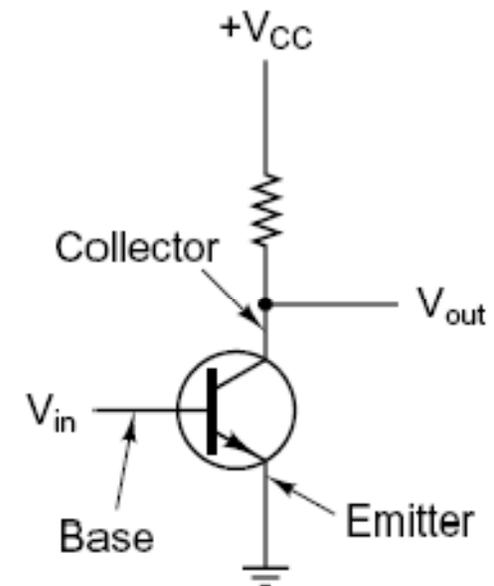
- A lógica digital baseia-se no fato de que um transistor pode operar como uma **chave binária** cujo tempo de comutação (chaveamento) é pequeno (nanosegundos).
- Componentes de um Transistor:
  - **Base;**
  - **Coletor;**
  - **Emissor.**



## Portas Lógicas (2)

### ■ Transistor

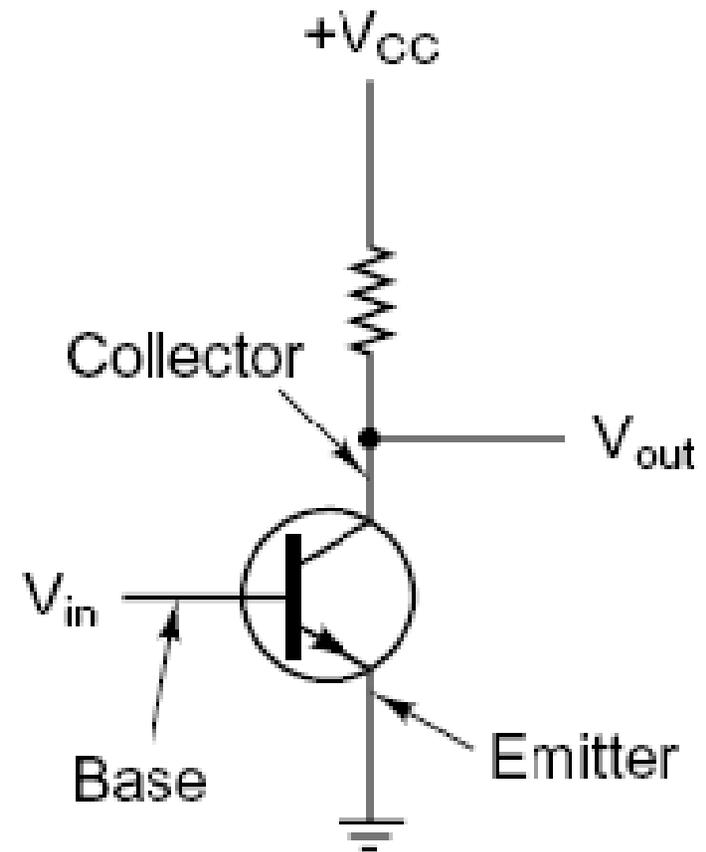
- Quando  $V_{in}$  estiver abaixo de um certo valor, o transistor abre;
  - $V_{out}$  assume um valor próximo a  $V_{cc}$  ( $V_{cc}$  é uma tensão regulada geralmente em +5 V);
- Quando  $V_{in}$  ultrapassa um certo valor, o transistor **comuta** e passa a agir como um fio sem resistência.
  - $V_{out}$  fica conectado logicamente à terra (0 volt);



## Portas Lógicas (3)

### ■ Transistor

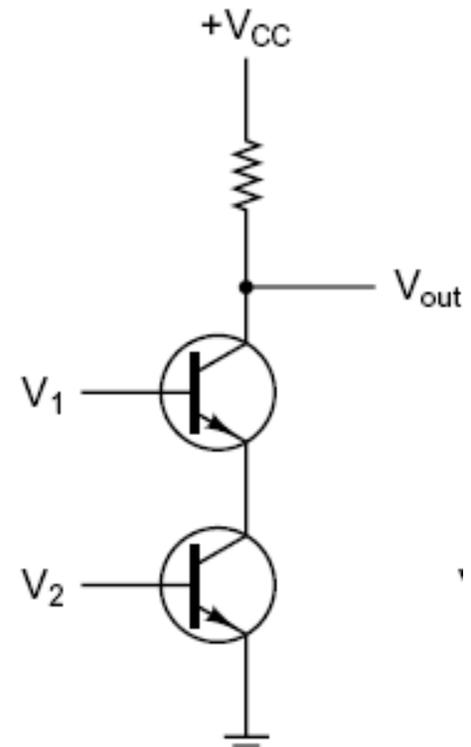
- Quando  $V_{in}$  estiver no nível lógico baixo,  $V_{out}$  estará no nível alto, e vice-versa.
- O circuito ao lado funciona logicamente como um **Inversor**;
  - Porta NOT



## Portas Lógicas (4)

### ■ Transistor

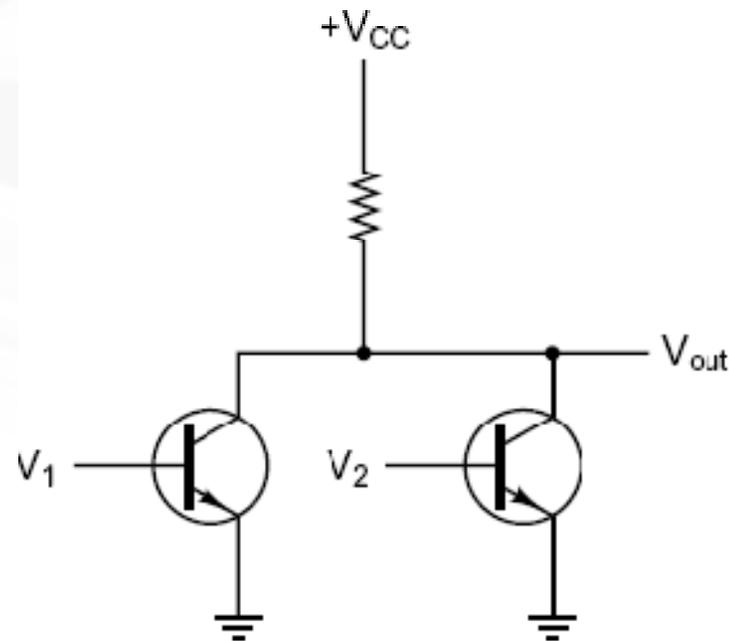
- Dois transistores ligados em série:
- Se  $V_1$  e  $V_2$  estiverem no nível lógico alto,  $V_{out}$  vai assumir nível lógico baixo.
- Se  $V_1$  ou  $V_2$  estiver no nível lógico baixo, o transistor correspondente estará aberto e a saída será alta.
- Qual a porta lógica correspondente?
  - (a)



## Portas Lógicas (5)

### ■ Transistor

- Dois transistores ligados em paralelo
- Qual a porta lógica correspondente?
  - (b)



## Portas Lógicas (6)

### ■ **Transistor**

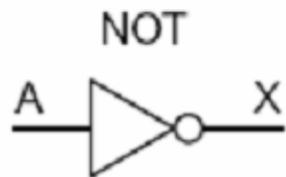
- Ao colocarmos um circuito inversor na saída de (a), o que obtemos?
- Se fizermos o mesmo na saída de (b)?

## Portas Lógicas (7)

- Um circuito lógico digital utilizado nos computadores atuais admite a presença de dois valores lógicos.
- Os valores lógicos são “materializados” através de sinais elétricos que representam **0** e **1**, ou **True** (verdadeiro) e **False** (falso). Em geral:
  - Sinal elétrico entre 0 e 1 volt pode representar o binário 0.
  - Sinal elétrico entre 2 e 5 volts pode representar o binário 1.
- **Portas Lógicas:** estruturas eletrônicas (componentes primitivos) capazes de calcular diversas funções utilizando esses sinais.
  - Formam a base de construção de inúmeros circuitos digitais e do *hardware* dos computadores.

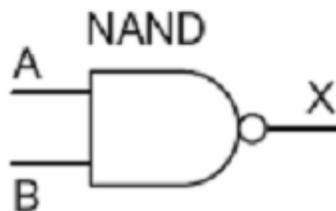
## Portas Lógicas (8)

- Principais portas lógicas
  - Podemos construir qualquer circuito lógico com apenas as portas AND, OR e NOT.
  - Ou apenas NAND, NOR e NOT.



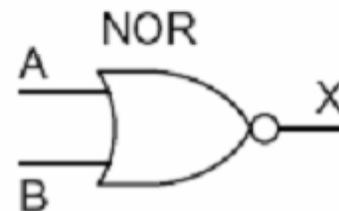
$$X = A'$$

A	X
0	1
1	0



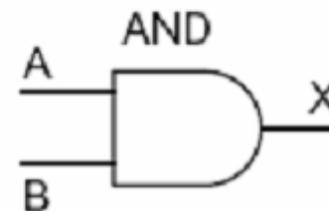
$$X = (AB)'$$

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



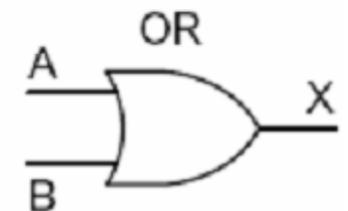
$$X = (A+B)'$$

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0



$$X = AB$$

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

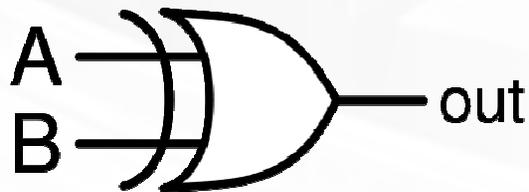


$$X = A+B$$

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

## Porta XOR

- $A'B + AB' = A \oplus B$



A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

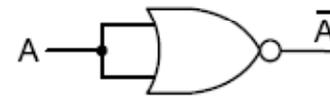
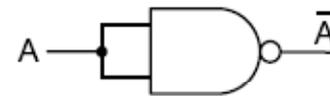
## Portas Lógicas (9)

- As portas NAND e NOR precisam de dois transistores, enquanto as portas AND e OR precisam de três.
  - Muitos computadores são baseados nas portas NAND e NOR, em vez das AND e OR.
- Na prática, existem outros tipos de implementações de portas lógicas, mas geralmente as portas NAND e NOR são mais simples que as AND e OR.
- Geralmente, uma porta lógica pode conter mais do que duas entradas, exceto a inversora.

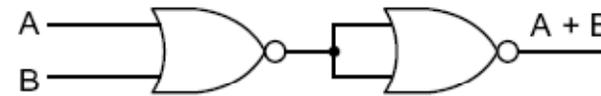
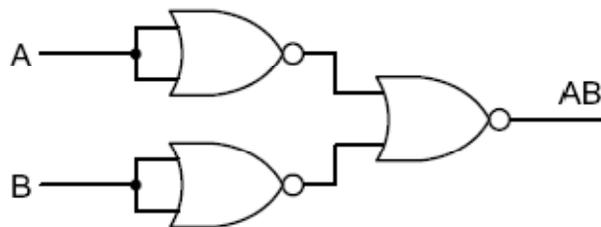
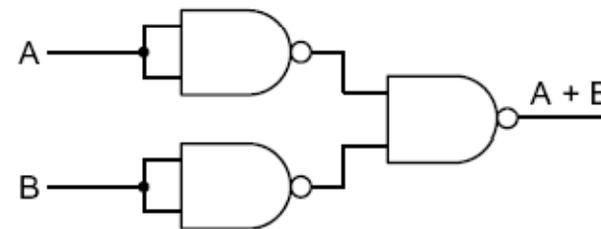
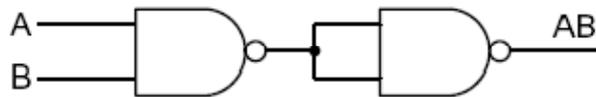
## Equivalência de circuitos (1)

- Muitas vezes é conveniente que o circuito seja implementado por meio de um único tipo de porta.
- Converter circuitos do tipo AND-OR-NOT resultantes de uma função em circuitos equivalentes que só usem portas NAND ou NOR.
- Para fazer isso, por exemplo, pode-se implementar as funções NOT, AND e OR usando uma dessas duas portas
  - Exercício: construir portas AND, OR e NOT usando NAND ou NOR.
- Em função disso, as portas NAND e NOR são conhecidas como **completas**, pois qualquer função booleana pode ser implementada com circuitos que só usem uma delas.

## Equivalência de circuitos (2)



(a)



(b)

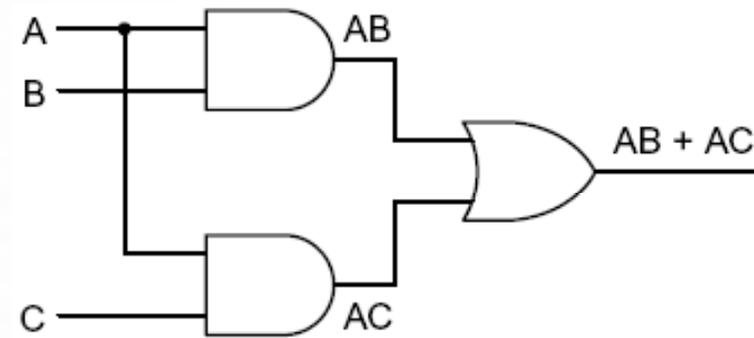
(c)

## Equivalência de Circuitos (3)

- Procura-se reduzir ao mínimo a quantidade de portas lógicas em circuitos integrados
  - Reduzir custos de componentes, espaço ocupado em placa de circuito impresso, consumo de energia, etc.
- **Equivalência de Circuitos**
  - Encontrar um outro circuito que calcule a mesma função calculada pelo original, usando menos portas lógicas ou portas mais simples de implementação (portas com duas entradas ao invés de quatro)
- Em geral, obtém-se em primeiro lugar uma função booleana para em seguida aplicar leis da álgebra de Boole para tentar encontrar uma equivalente mais simples

## Equivalência de Circuitos (4)

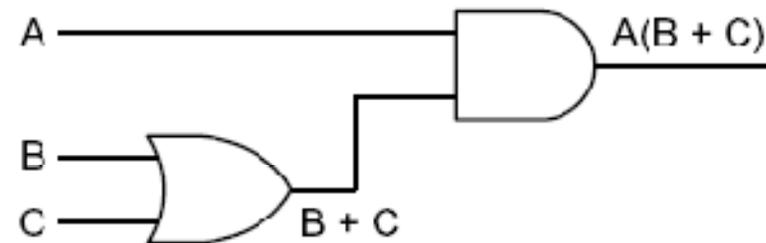
- $AB + AC$  pode ser fatorado como  $A(B + C)$  por aplicação da propriedade distributiva?



A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

## Equivalência de Circuitos (5)

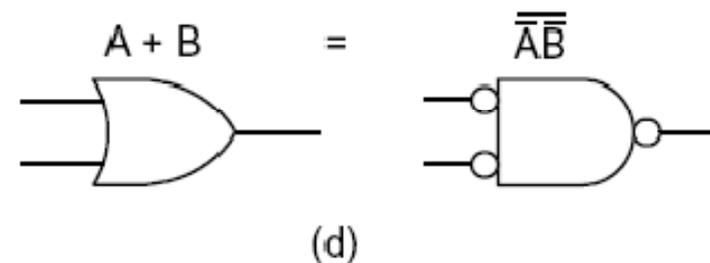
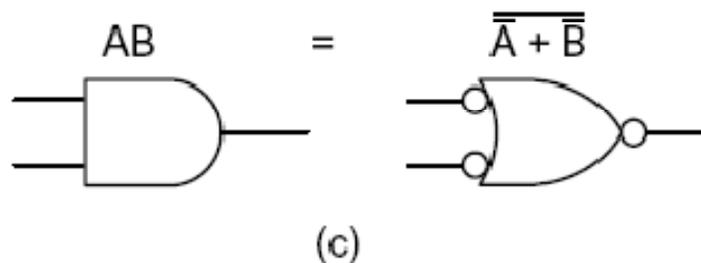
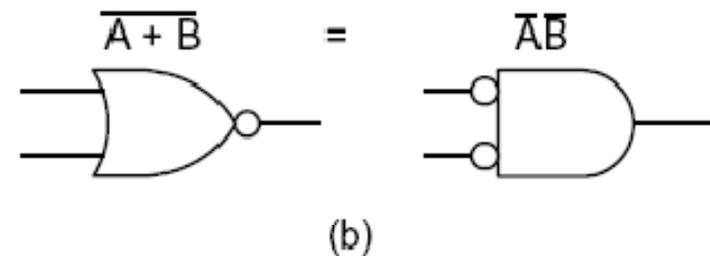
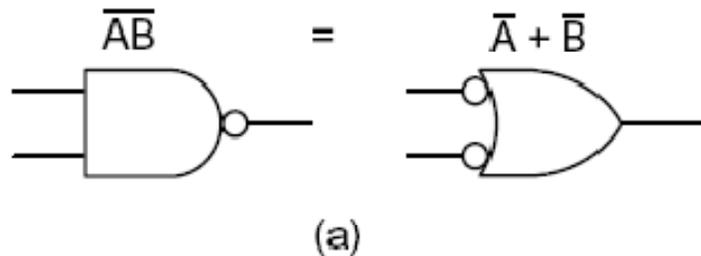
- Sim.



A	B	C	A	B + C	A(B + C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

## Equivalência de Circuitos (6)

- Por exemplo, considere a equivalência de circuitos a seguir em função da propriedade de De Morgan;



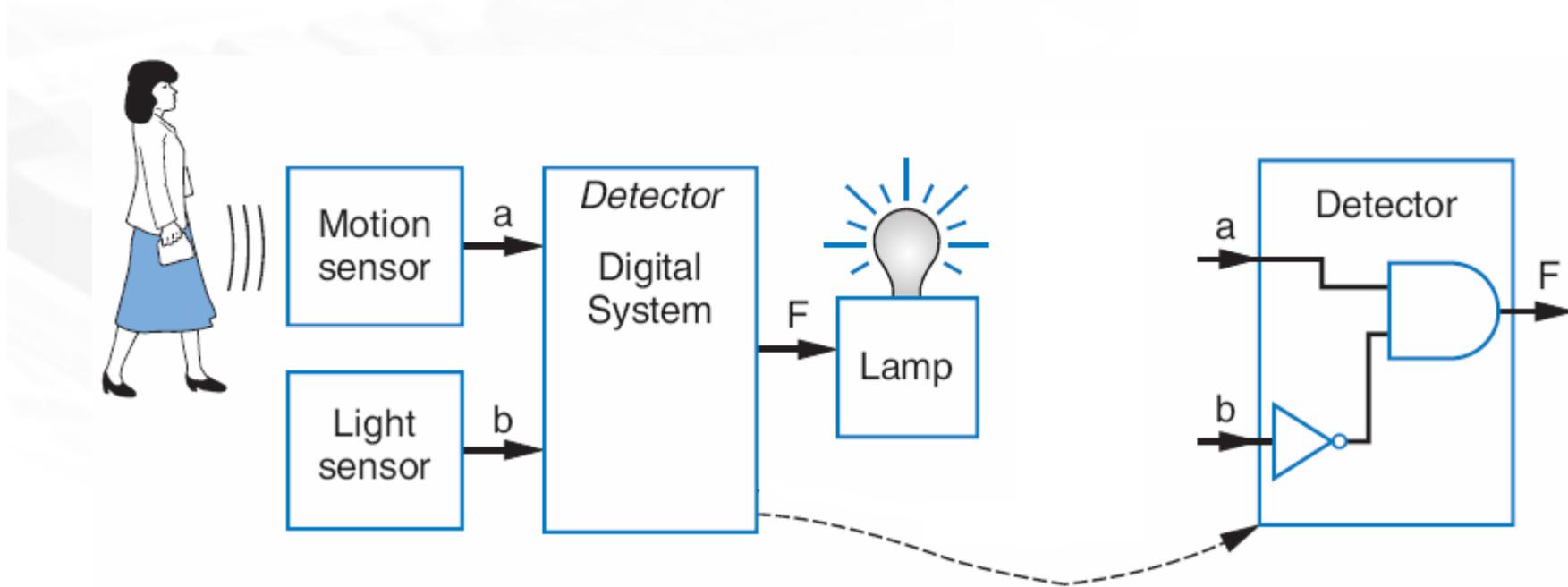
## Nível da Lógica Digital

- É composto pelo hardware da máquina
- Portas Lógicas são os objetos de interesse dos projetistas de computadores nesse nível
- As portas lógicas são os elementos primários de circuitos lógicos mais complexos;
- Qualquer circuito lógico pode ser descrito através de uma expressão booleana;
- Combinação de portas lógicas:
  - Funções aritméticas;
  - Memórias (registradores);
  - Processadores.

## Nível dos Dispositivos (Engenharia Elétrica)

- Situado abaixo do nível lógico digital
- Microeletrônica
- Características físicas
- Malha de transistores
- Tecnologias de fabricação de circuitos integrados
- (Naturalmente, poderia se indagar como funcionam os transistores por dentro: aí já pertence à física do estado sólido).

## Construindo circuitos com portas lógicas (1)



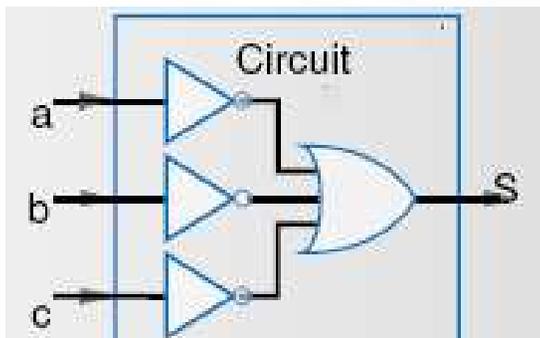
- Acende a lâmpada ( $F=1$ ) quando movimento é detectado ( $a=1$ ) e a luz está apagada ( $b=0$ );
- $F = a \text{ AND NOT}(b)$
- Acabamos de construir nosso primeiro circuito digital.

## Construindo circuitos com portas lógicas (2)

- Exemplo de sinalização para banheiro de aeronave;
  - 3 lavatórios: cada lavatório tem um sensor que possui valor lógico 1 se a porta estiver fechada (a, b, c);
  - A lâmpada (sinal S) acende se houver pelo menos um lavatório disponível;
- Qual é a expressão booleana para a lâmpada acender?

## Construindo circuitos com portas lógicas (2)

- Exemplo de sinalização para banheiro de aeronave;
  - 3 lavatórios: cada lavatório tem um sensor que possui valor lógico 1 se a porta estiver fechada (a, b, c);
  - A lâmpada (sinal S) acende se houver pelo menos um lavatório disponível;
- Qual é a expressão booleana para a lâmpada acender?
- $S = a' + b' + c'$

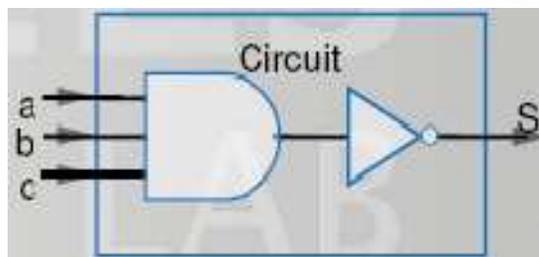


## Construindo circuitos com portas lógicas (2)

- Alternativa: agora pense considerando a situação em que o banheiro de aeronave estiver completamente ocupado;
- Qual será a expressão booleana correspondente?
- Lembrar:
  - 3 lavatórios: cada lavatório tem um sensor que possui valor lógico 1 se a porta estiver fechada (a, b, c);
  - A lâmpada (sinal S) acende se houver pelo menos um lavatório disponível;

## Construindo circuitos com portas lógicas (2)

- Alternativa: agora pense considerando a situação em que o banheiro de aeronave estiver completamente ocupado;
- Qual será a expressão booleana correspondente?
- Lembrar:
  - 3 lavatórios: cada lavatório tem um sensor que possui valor lógico 1 se a porta estiver fechada (a, b, c);
  - A lâmpada (sinal S) acende se houver pelo menos um lavatório disponível;
- $S' = (a' + b' + c')' = (a')' * (b')' * (c')' = abc$



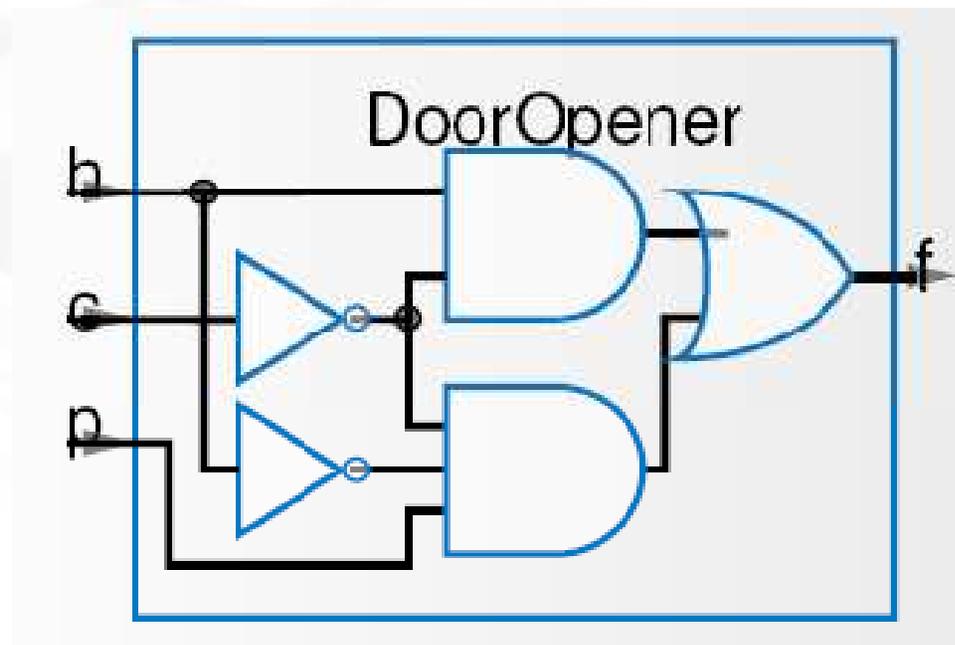
## Construindo circuitos com portas lógicas (3)

- Deseja-se um circuito para abertura automática de portas
- Saída:  $f=1$  abre a porta
- Entradas:
  - $p=1$ : pessoa detectada
  - $h=1$ : chave para forçar a abertura
  - $c=1$ : chave para forçar o fechamento
- A porta deve ser aberta quando  $h=1$  and  $c=0$ , ou  $h=0$  and  $p=1$  and  $c=0$

## Construindo circuitos com portas lógicas (3)

- Deseja-se um circuito para abertura automática de portas
- Saída:  $f=1$  abre a porta
- Entradas:
  - $p=1$ : pessoa detectada
  - $h=1$ : chave para forçar a abertura
  - $c=1$ : chave para forçar o fechamento
- A porta deve ser aberta quando  $h=1$  and  $c=0$ , ou  $h=0$  and  $p=1$  and  $c=0$
- Equação:  $f = hc' + h'pc'$

## Construindo circuitos com portas lógicas (3)



## Construindo circuitos com portas lógicas (3)

- Ache um chip barato que compute:
- $f = c'hp + c'hp' + c'h'p$
- Usando Álgebra Booleana:
  - $f = c'hp + c'hp' + c'h'p$
  - $f = c'h(p + p') + c'h'p$  (distributiva)
  - $f = c'h(1) + c'h'p$  (complemento)
  - $f = c'h + c'h'p$  (identidade) -> expressão encontrada anteriormente
  - $f = c'(h + h'p)$  (distributiva)
  - $f = c'[(h + h')(h + p)]$  (distributiva)
  - $f = c'(h + p)$  (complemento)

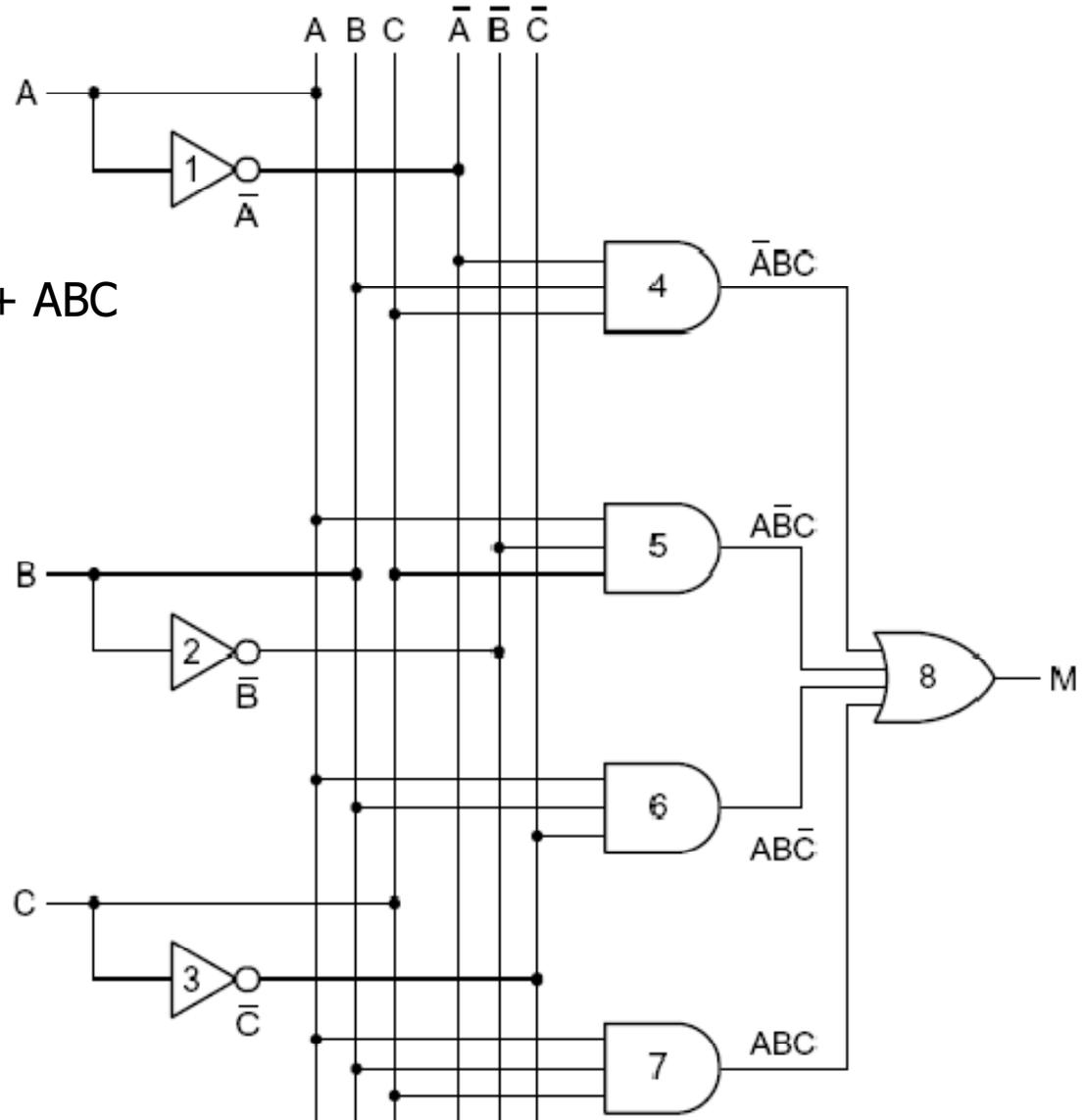
## Projeto de circuitos digitais

- Função Maioria
  - $M = f(A, B, C)$
  - A saída será 0 se a maioria das variáveis de entrada for zero, e será 1 se a maioria das variáveis de entrada for 1.
  - Resolver utilizando portas AND, OR e NOT
- Função Booleana x Implementação de circuitos digitais

■ Função Maioria

■  $M = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



## Projeto de circuitos digitais

- Implementar um circuito referente a uma função booleana, utilizando portas AND, OR e NOT:
  - Obtenha a tabela-verdade da função;
  - Utilize inversores para obter o complemento de cada uma das entradas da função;
  - Desenhe uma porta AND para cada termo com valor 1 na coluna de resultados;
  - Ligue as portas AND às entradas apropriadas;
  - Ligue a saída das portas AND a uma porta OR.