



Introdução à Computação

Jordana Sarmenghi Salamon

`jssalamon@inf.ufes.br`

jordanasalamon@gmail.com

<http://inf.ufes.br/~jssalamon>

Departamento de Informática

Universidade Federal do

Espírito Santo

- Noções de Circuitos Lógicos
 - Portas Lógicas
 - Equivalência de Circuitos
 - Circuitos Lógicos

Portas Lógicas



These blocks...



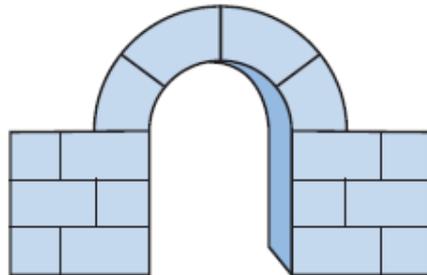
...are hard to work with.



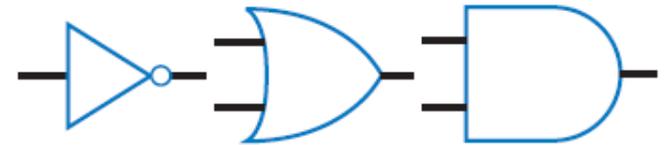
Transistors are hard to work with



The right building blocks...

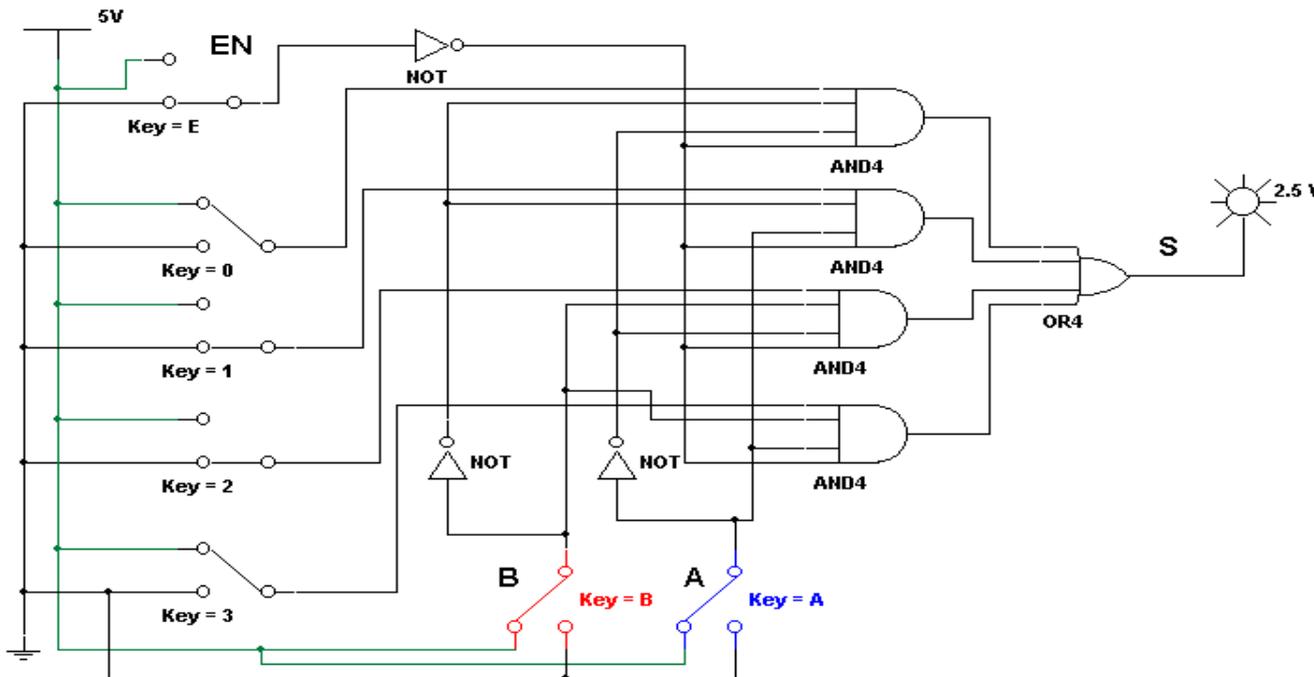


...enable greater designs.

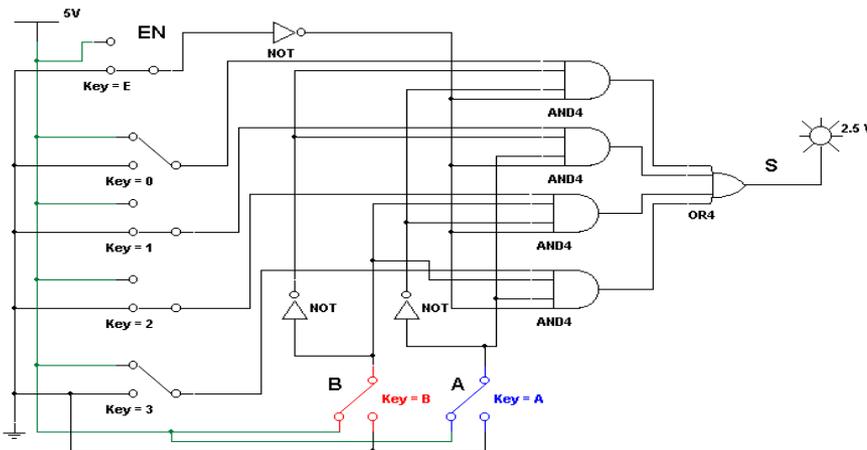


The logic gates that we'll soon introduce enable greater designs

- **Portas lógicas** ou circuitos lógicos, são dispositivos que operam um ou mais sinais lógicos de entrada para produzir uma e somente uma saída, dependente da função implementada no circuito.



- São geralmente usadas em **circuítos eletrônicos**, por causa das situações que os sinais deste tipo de circuito podem apresentar: presença de sinal, ou "1"; e ausência de sinal, ou "0". O comportamento das portas lógicas é conhecido pela tabela verdade que apresenta os estados lógicos das entradas e das saídas.



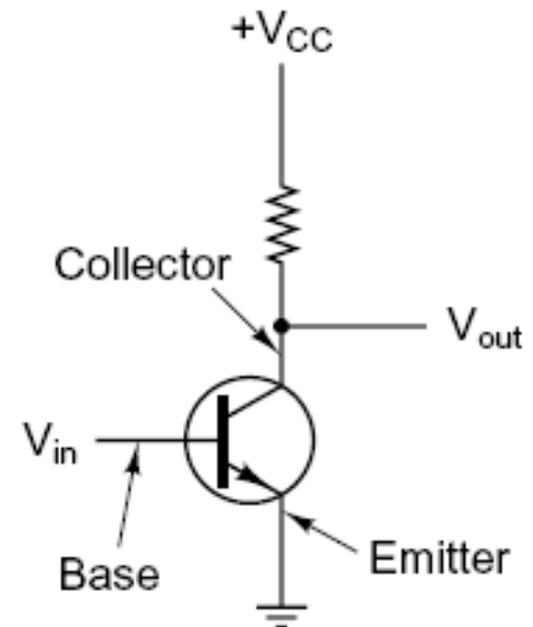
- Em 1854, o matemático britânico **George Boole** (1815 - 1864) apresentou um sistema matemático de análise lógica conhecido como álgebra de Boole.
- No início da era da eletrônica, todos os problemas eram resolvidos por **sistemas analógicos**, isto é, sistemas lineares.
- Apenas em 1938, o engenheiro americano Claude Shannon utilizou as teorias da álgebra de Boole para a solução de problemas de circuitos de telefonia com relés, praticamente introduzindo na área tecnológica o campo da **eletrônica digital**.
- Esse ramo da eletrônica emprega em seus sistemas um pequeno grupo de circuitos básicos padronizados conhecidos como **Portas Lógicas**.

- **Transistor**

- A lógica digital baseia-se no fato de que um transistor pode operar como uma chave binária cujo tempo de chaveamento é pequeno (nanosegundos).

- Componentes de um Transistor:

- Base;
- Coletor;
- Emissor.

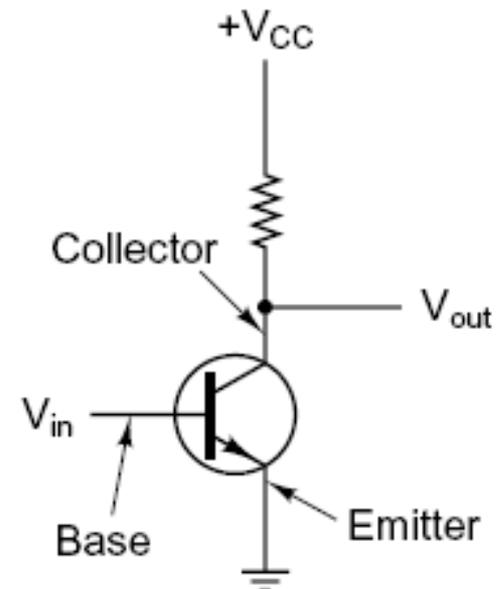


- **Transistor**

- Quando V_{in} estiver no nível lógico baixo, V_{out} estará no nível alto, e vice-versa.

- O circuito funciona logicamente como um **Inversor;**

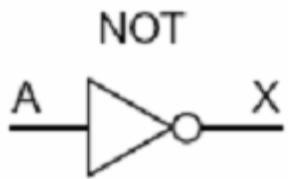
 - Porta NOT



- 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 1 e 0, 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.

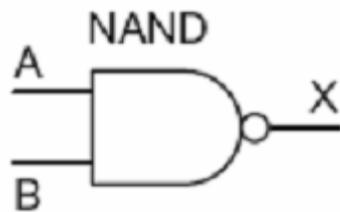
Porta Lógicas

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



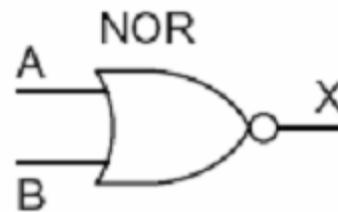
$$X = A'$$

A	X
0	1
1	0



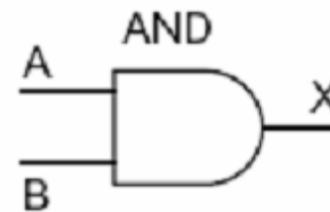
$$X = (AB)'$$

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



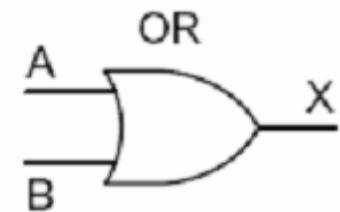
$$X = (A+B)'$$

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

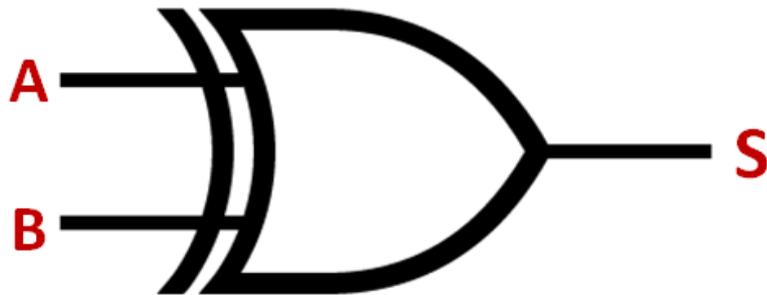


$$X = A+B$$

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

Porta Lógicas

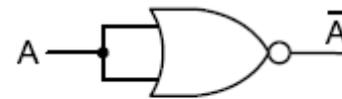
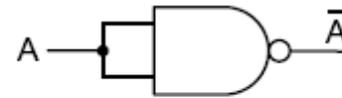
- As portas NAND e NOR precisam de dois transistores, enquanto as portas AND e OR precisam de três.
- Muitos computadores são baseados nas portas NAND e NOR, em vez das AND e OR.
- Geralmente, uma porta lógica pode conter mais do que duas entradas, exceto a inversora.

Equivalência de circuitos

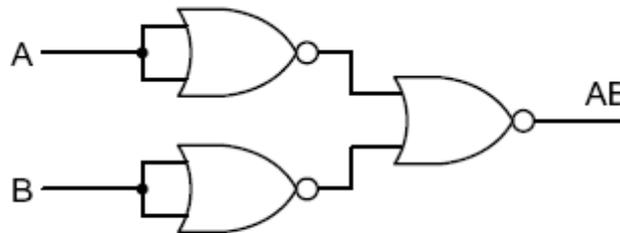
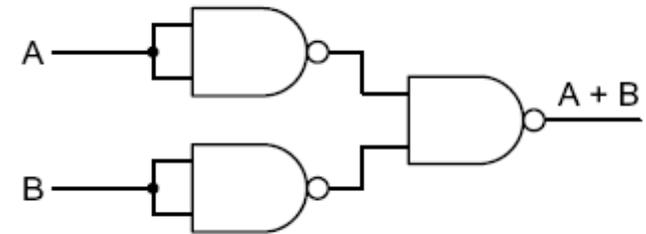
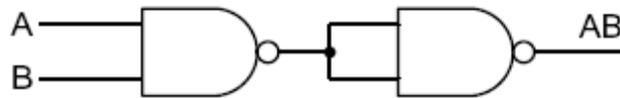
- Muitas vezes é conveniente que o circuito seja implementado por meio de um **único tipo de porta**.
- Assim, convertemos circuitos do tipo AND-OR-NOT resultantes de uma função em circuitos equivalentes que só usem portas **NAND ou NOR**.
- Para fazer isso, por exemplo, pode-se implementar as funções NOT, AND e OR usando uma dessas duas portas
- 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

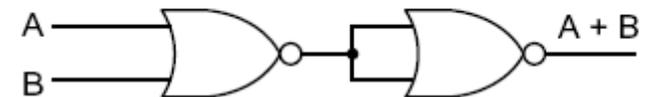
- Exemplo: construir portas NOT, AND e OR usando NAND ou NOR.



(a)



(b)



(c)

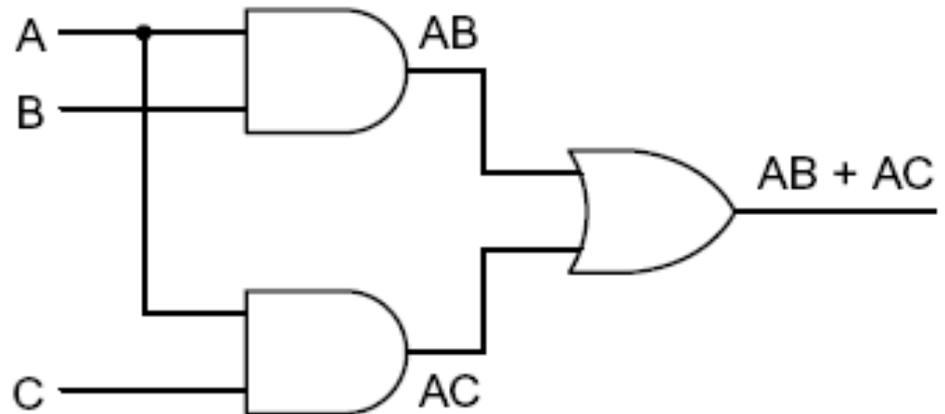
Equivalência de circuitos

- 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
- 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

- Exemplo:
- A expressão $AB + AC$ pode ser fatorada como $A(B + C)$ por aplicação da propriedade distributiva?

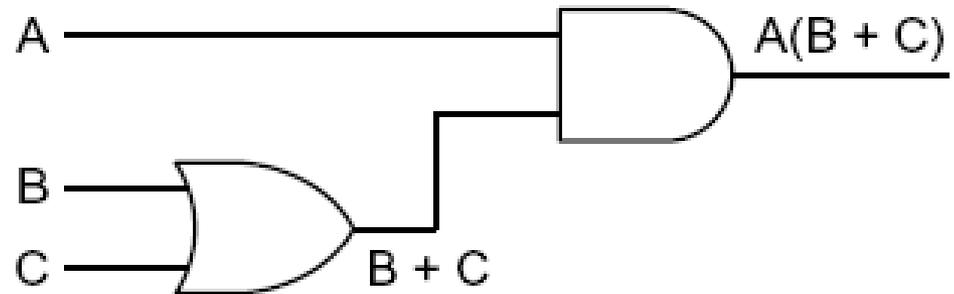
A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1



Equivalência de circuitos

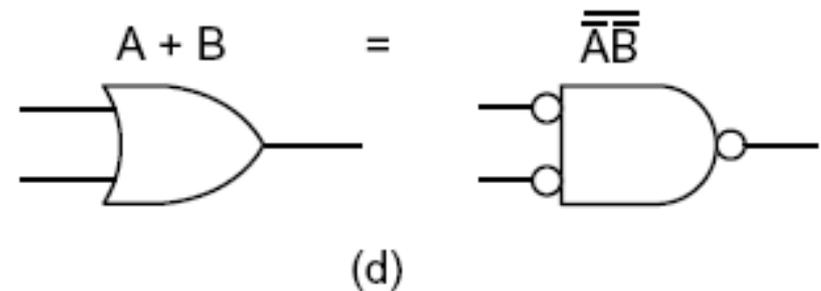
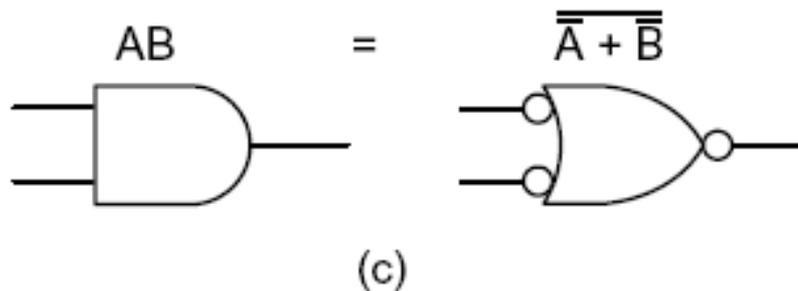
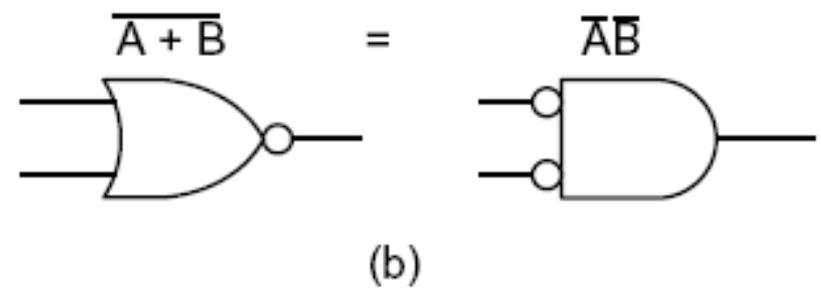
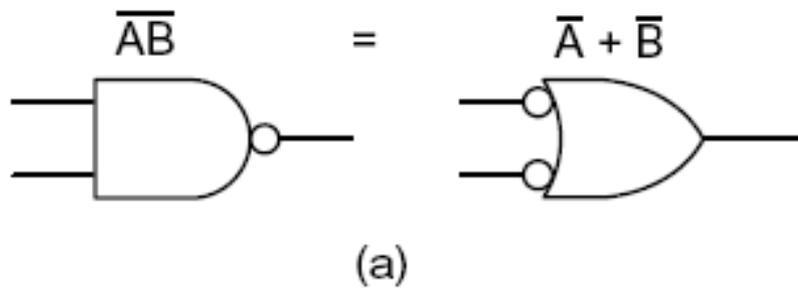
- Exemplo:
- A expressão $AB + AC$ pode ser fatorada como $A(B + C)$ por aplicação da propriedade distributiva?
- **Sim.**

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

- Abaixo, equivalência de circuitos em função da propriedade de De Morgan;

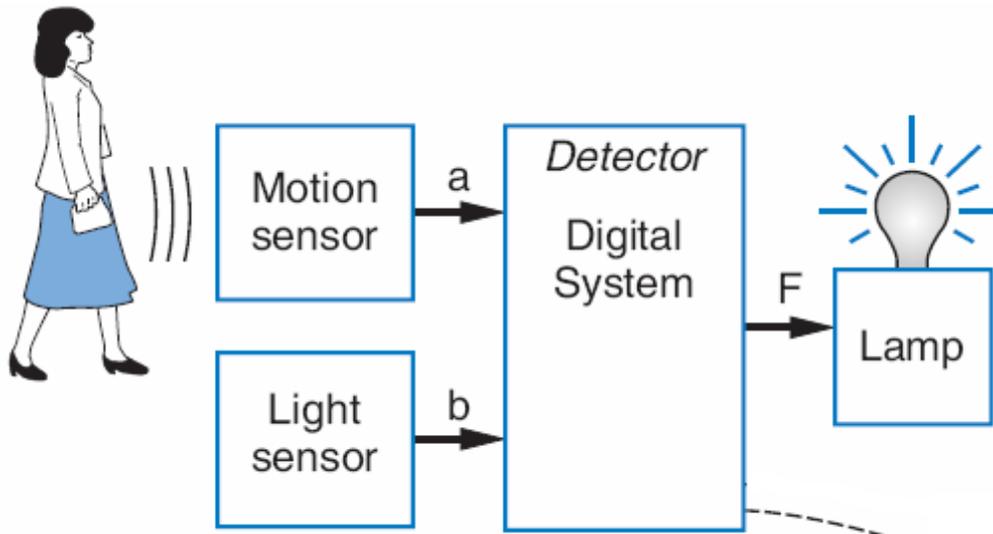


Nível da Lógica Digital

- 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.

Construindo circuitos com portas lógicas

- Exemplo:
- Como seria o circuito para acender uma lâmpada de um cômodo quando uma pessoa adentrasse o cômodo?
- Precisamos de formas de identificar se a pessoa está no cômodo e se a lâmpada se encontra apagada ou acesa;

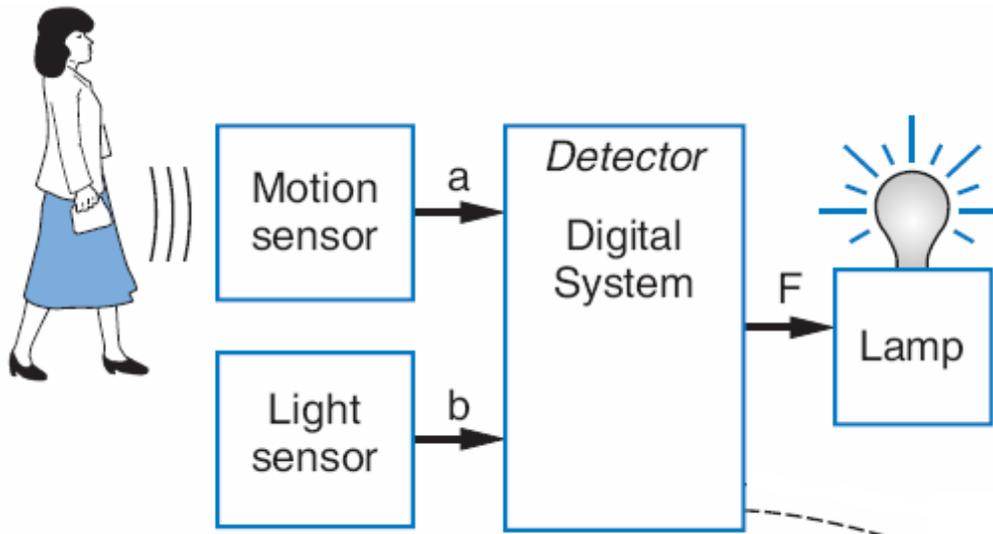


A lâmpada deverá ser acesa se:

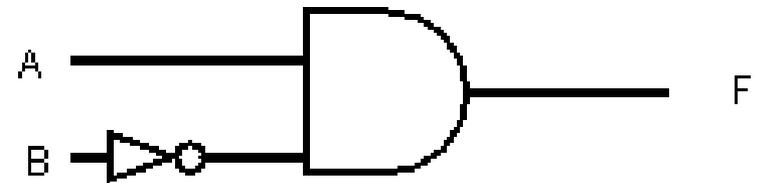
- O sensor detectar o movimento **E**
- O sensor detectar que a lâmpada **não está** acesa, ou seja, está apagada

Construindo circuitos com portas lógicas

- Exemplo:
- Lâmpada acesa: $b = 1$;
- Lâmpada apagada: $b = 0$;
- Sensor de movimento ativado: $a = 1$;



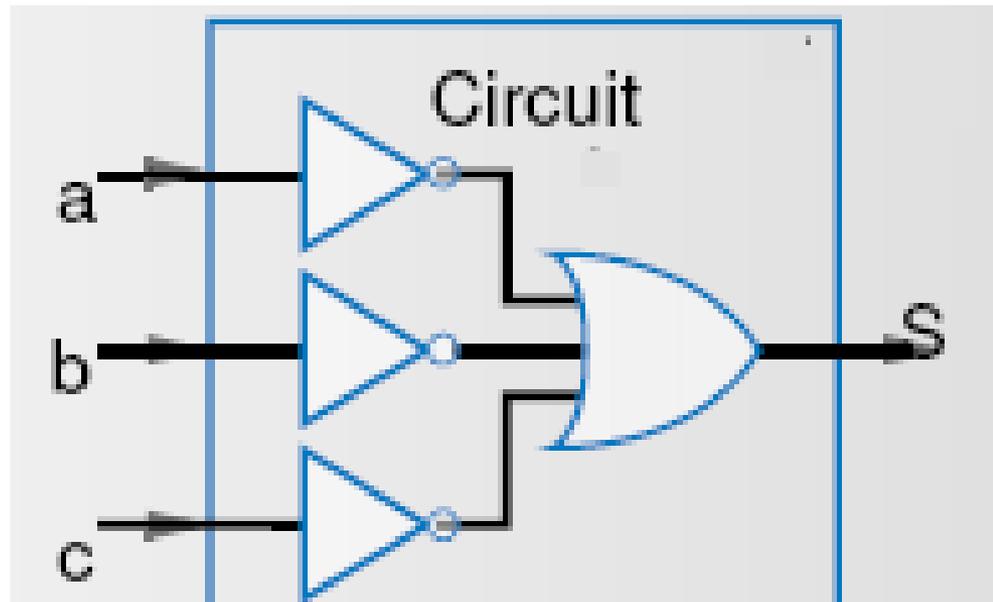
$$F = ab'$$



- Exercício:
- Construindo o circuito de sinalização para um banheiro de aeronave, temos:
 - 3 lavatórios: cada banheiro 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 banheiro disponível;
- Qual é a expressão booleana para a lâmpada acender?

Construindo circuitos com portas lógicas

- Exercício:
- Pelo menos uma das portas deve estar aberta, ou seja, não fechada
- Porta fechada = 1;
- Porta aberta = 0;
- $S = a' + b' + c'$



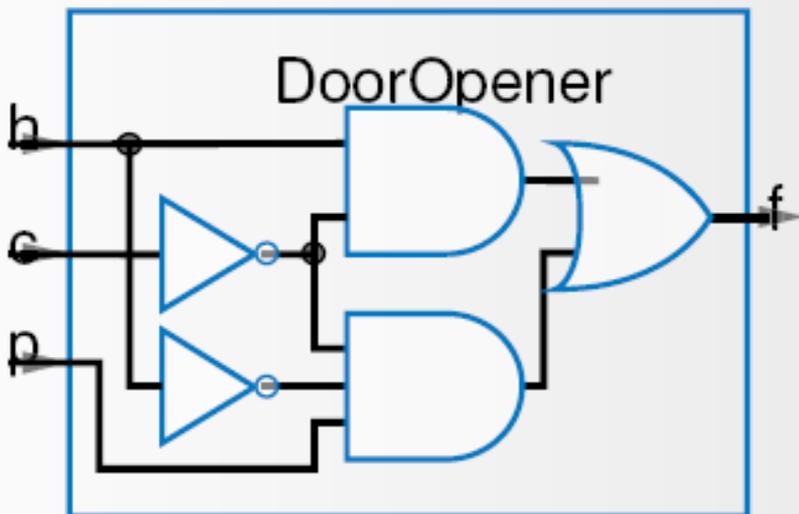
Construindo circuitos com portas lógicas



- Exercício:
- 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

Construindo circuitos com portas lógicas

- Exercício:
- Quando a porta deve ser aberta?
 - Quando houver uma chave para forçar a abertura e não o fechamento ou quando houver uma pessoa e nenhuma das duas chaves
- Equação: $f = hc' + h'pc'$



Construindo circuitos com portas lógicas



- Exercício:
- Equação: $f = hc' + h'pc'$
- Essa equação pode ser simplificada?

Construindo circuitos com portas lógicas

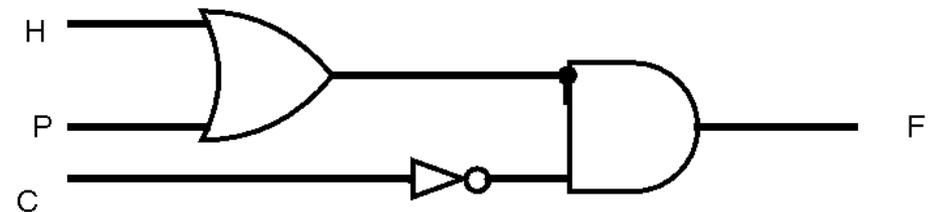
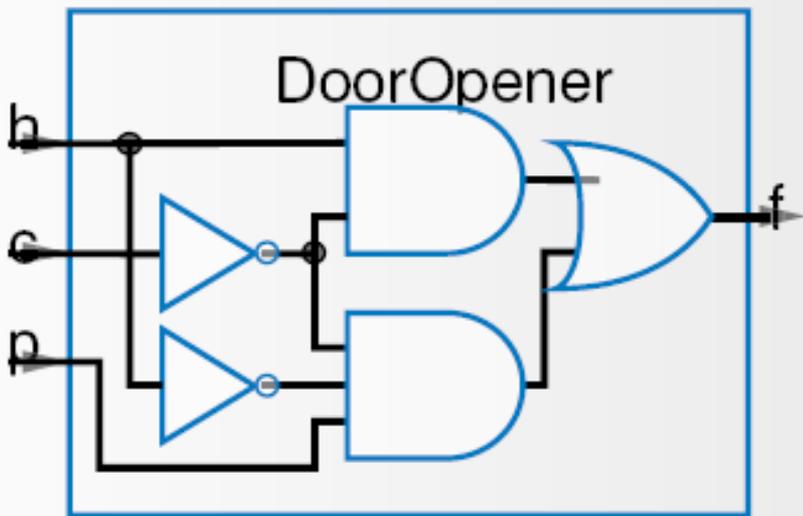
- Exercício:
- Equação: $f = hc' + h'pc'$
- Essa equação pode ser simplificada?
- **Sim.**
- $f = hc' + h'pc'$
- $f = c'(h + h'p)$ ← (distributiva)
- $f = c'[(h+h')(h+p)]$ ← (distributiva)
- $f = c'(h + p)$ ← (complemento)

Construindo circuitos com portas lógicas

- Exercício:

- $f = hc' + h'pc'$

$$f = c'(h + p)$$



Construindo circuitos com portas lógicas

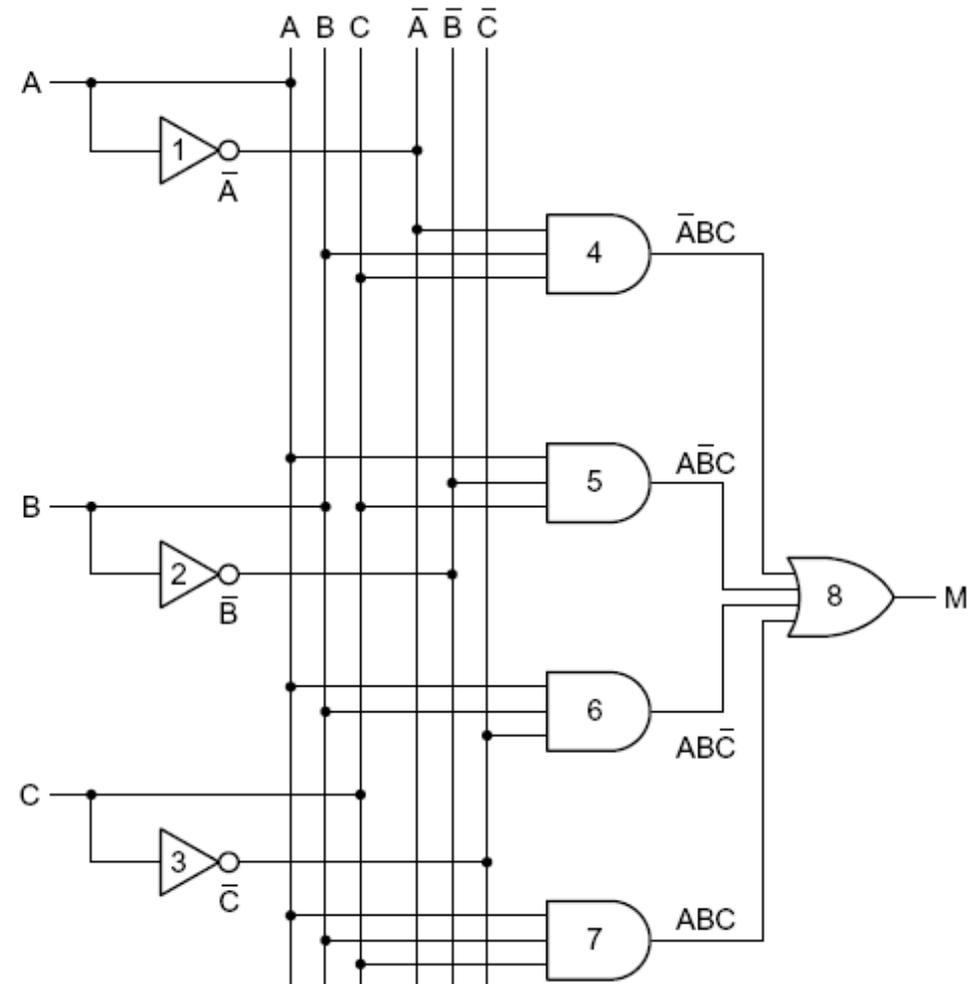


- Exercício: Função Maioria
- $M = f(A, B, C)$
- A saída será 0 se a maioria das variáveis de entrada for zero, e será 1 se a maioria das variáveis de entrada for 1.
- Resolver utilizando portas AND, OR e NOT.

Construindo circuitos com portas lógicas

- Função Maioria
- $M = A'BC + AB'C + ABC' + ABC$

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

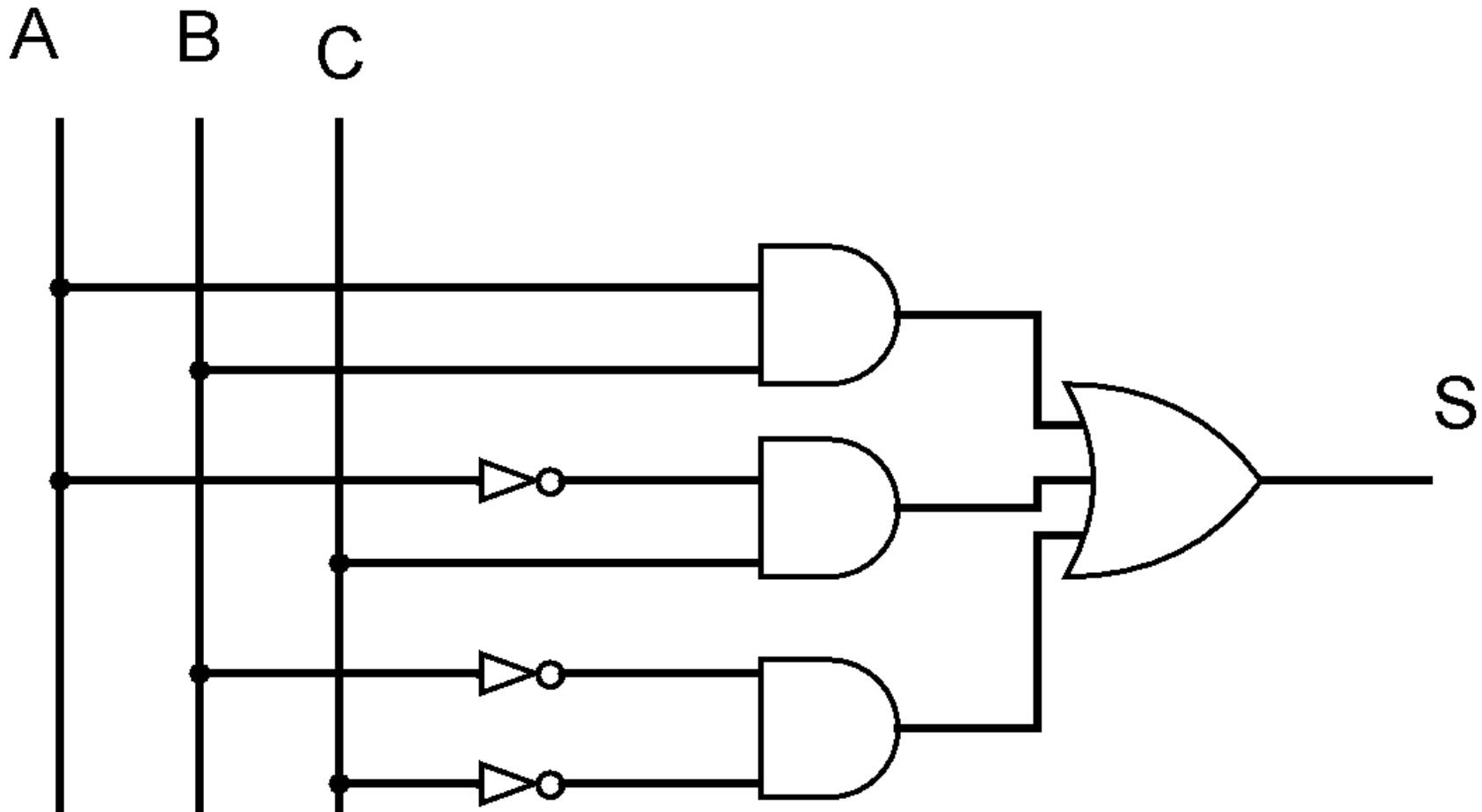


Equivalência de circuitos

- Exercício: construir o circuito abaixo usando somente portas NAND.
- **$S = AB + A'C + B'C'$**
- O primeiro passo é construir o circuito com as portas NOT, AND e OR
- O segundo passo para fazer o circuito com somente portas NAND ou NOR é trocar cada porta pela portas NAND/NOR equivalente
- O terceiro passo é remover as portas desnecessárias que repetem operações.

Equivalência de circuitos

- 1) Construindo o circuito com portas NOT, AND e OR



Equivalência de circuitos

- 2) Substituindo as portas pelas NAND equivalentes:

- **$S = AB + A'C + B'C'$**

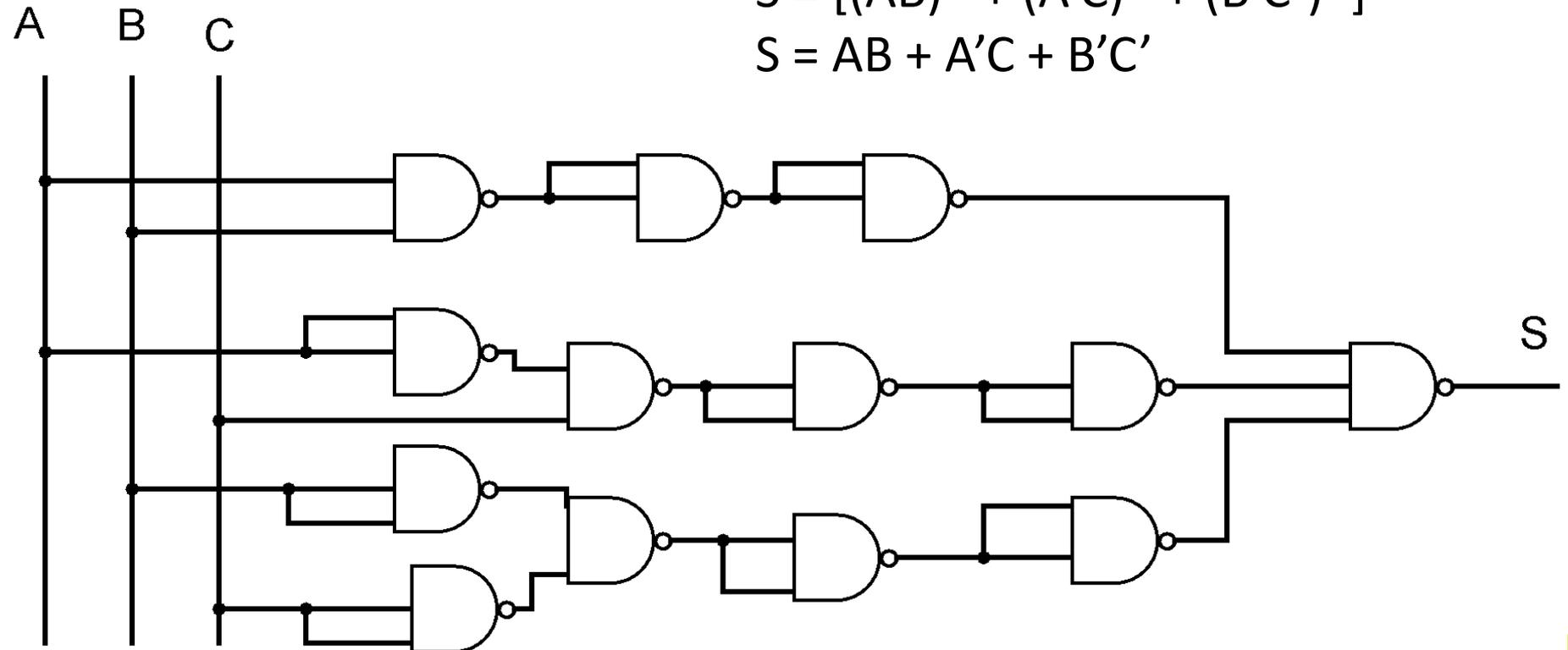
Conferindo a expressão do circuito

$$S = [(AB)''' \cdot (A'C)''' \cdot (B'C')''']'$$

$$S = [(AB)' \cdot (A'C)' \cdot (B'C')']'$$

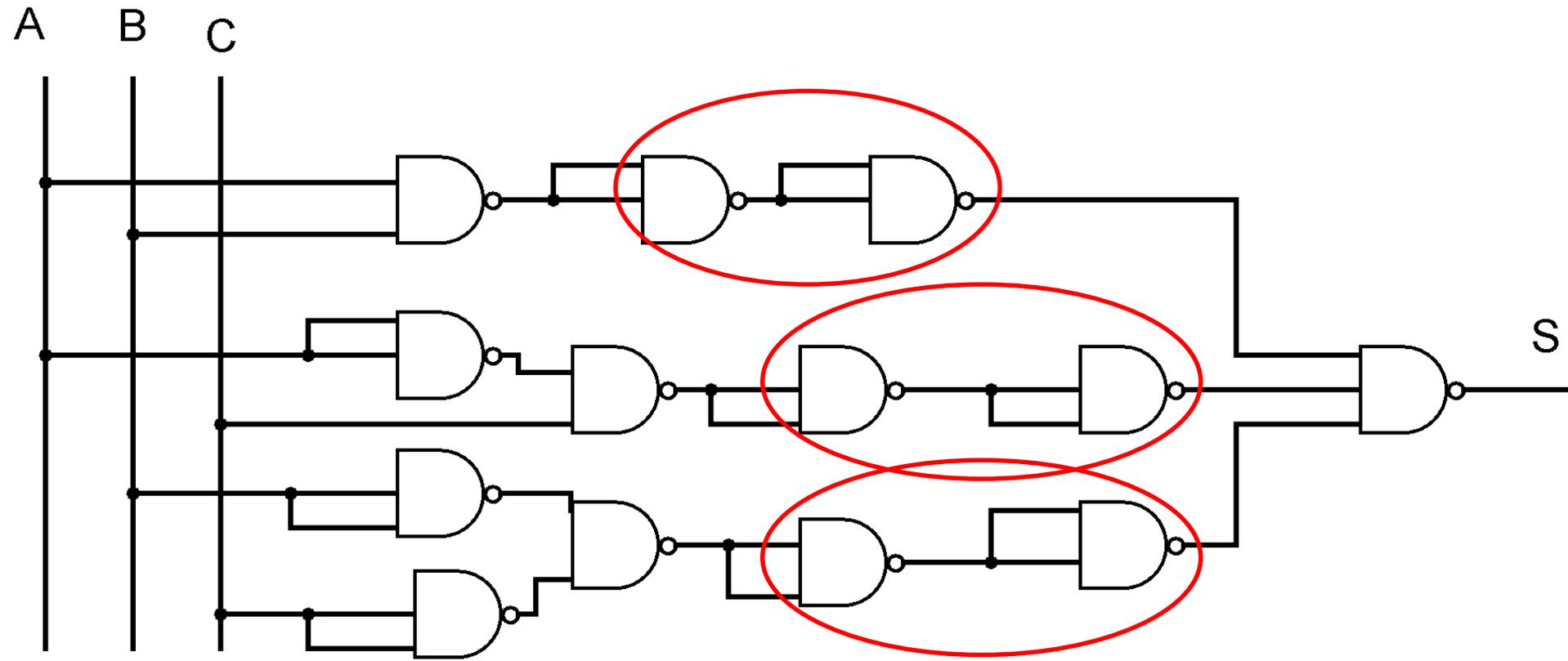
$$S = [(AB)'' + (A'C)'' + (B'C')'']$$

$$S = AB + A'C + B'C'$$



Equivalência de circuitos

- 3) Identificando as portas redundantes:
- **$S = AB + A'C + B'C'$**



Equivalência de circuitos

- 3) Removendo as portas que fazem operações redundantes:

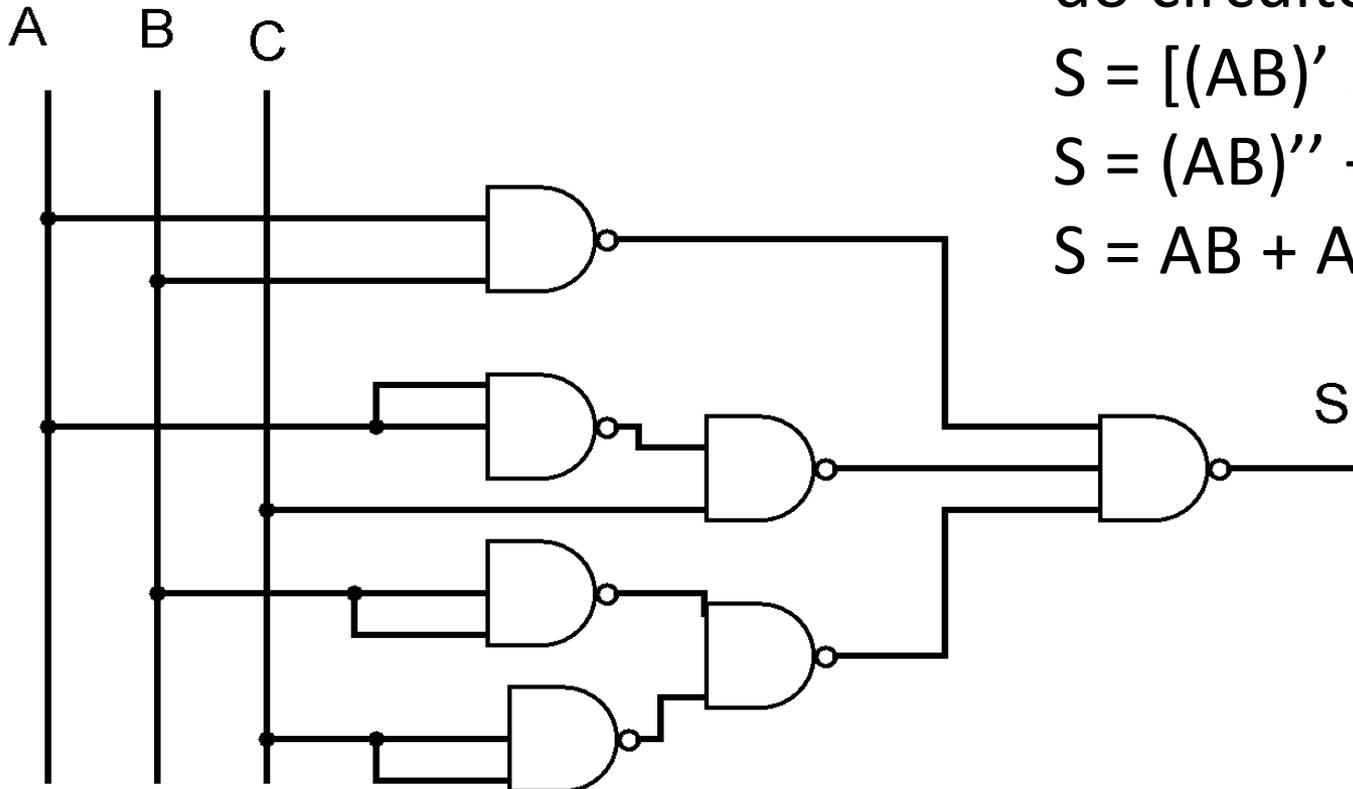
- **$S = AB + A'C + B'C'$**

Conferindo a expressão do circuito

$$S = [(AB)' \cdot (A'C)' \cdot (B'C')']'$$

$$S = (AB)'' + (A'C)'' + (B'C')''$$

$$S = AB + A'C + B'C'$$





Introdução à Computação

Jordana Sarmenghi Salamon

`jssalamon@inf.ufes.br`

jordanasalamon@gmail.com

<http://inf.ufes.br/~jssalamon>

Departamento de Informática

Universidade Federal do

Espírito Santo