

Capítulo 2–Projeto Lógico Combinacional B

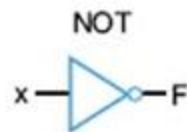
Profa. Eliete Caldeira



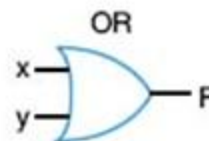
Álgebra booleana

▶ Operadores:

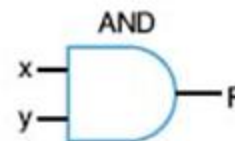
- NOT(a) é escrito como a' (lê-se alinha), também conhecido como complemento de a ou inverso de a
- a OR b é escrito como $a+b$ (lê-se a OU b)
- a AND b é escrito como $a.b$ (lê-se a E b). Pode-se usar ab desde que fique claro que a e b são variáveis separadas



x	F
0	1
1	0



x	y	F
0	0	0
0	1	1
1	0	1
1	1	1



x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

Álgebra booleana

► Precedência dos operadores:

TABLE 2.1 Boolean algebra precedence, highest precedence first.

Symbol	Name	Description
()	Parentheses	Evaluate expressions nested in parentheses first
'	NOT	Evaluate from left to right
*	AND	Evaluate from left to right
+	OR	Evaluate from left to right

Álgebra booleana

▶ Definições usando a equação

$$F(a,b,c) = a'bc + abc' + ab + c$$

- Variável – representa uma quantidade (0 ou 1).
 - Na equação as variáveis são a, b e c
- Literal – é a expressão de uma variável na forma normal ou complementada.
 - A equação tem 9 literais a', b, c, a, b, c', a, b, c
- Termo de produto – um termo de produto é um produto de literais.
 - Na equação a'bc, abc', ab e c são os termos de produto
- Soma de produtos – uma expressão com um OR de termos de produto.
 - A equação apresenta F na forma de soma de produtos

Álgebra booleana

- ▶ Propriedades da álgebra de Boole
 - Comutativa
 - $a+b = b+a$
 - $a.b = b.a$
 - Distributiva
 - $a.(b+c) = a.b + a.c$
 - $a+(b.c) = (a+b).(a+c)$
 - Elemento neutro aditivo e multiplicativo
 - $0+a = a+0 = a$
 - $1.a = a.1 = a$
 - Complemento
 - $a+a' = 1$
 - $a.a' = 0$

Álgebra booleana

- ▶ Propriedades da álgebra de Boole (cont.)
 - Princípio de dualidade – toda expressão algébrica dedutível dos postulados da álgebra de Boole permanece válida se:
 - As operações (+) e (.) forem intercambiadas entre si em toda a expressão **e**,
 - Os elementos de identidade 0 e 1 forem intercambiados entre si em toda a expressão
 - Elementos nulos
 - $a + 1 = 1$
 - $a \cdot 0 = 0$
 - Identidade
 - $0 + a = a + 0 = a$
 - $1 \cdot a = a \cdot 1 = a$

Álgebra booleana

- ▶ Propriedades da álgebra de Boole (cont.)
 - Lei da Idempotência
 - $a + a = a$
 - $a \cdot a = a$
 - Lei da Involução
 - $(a')' = a$
 - Lei da absorção
 - $a + ab = a$
 - $a \cdot (a + b) = a$
 - Simplificação
 - $a + a'b = a + b$
 - $a \cdot (a' + b) = a \cdot b$

Álgebra booleana

- ▶ Propriedades da álgebra de Boole (cont.)
 - Associativa
 - $(a+b)+c = a+(b+c)$
 - $(a.b).c = a.(b.c)$
 - Lei de DeMorgan
 - $(a+b)' = a'.b'$
 - $(a.b)' = a'+b'$
 - Lei de DeMorgan generalizada
 - $(a+b+\dots+j)' = a'.b'.\dots.j'$
 - $(a.b.\dots.j)' = a'+b'+\dots+j'$

Exercícios

- ▶ Use as propriedades da Álgebra de Boole para simplificar o circuito

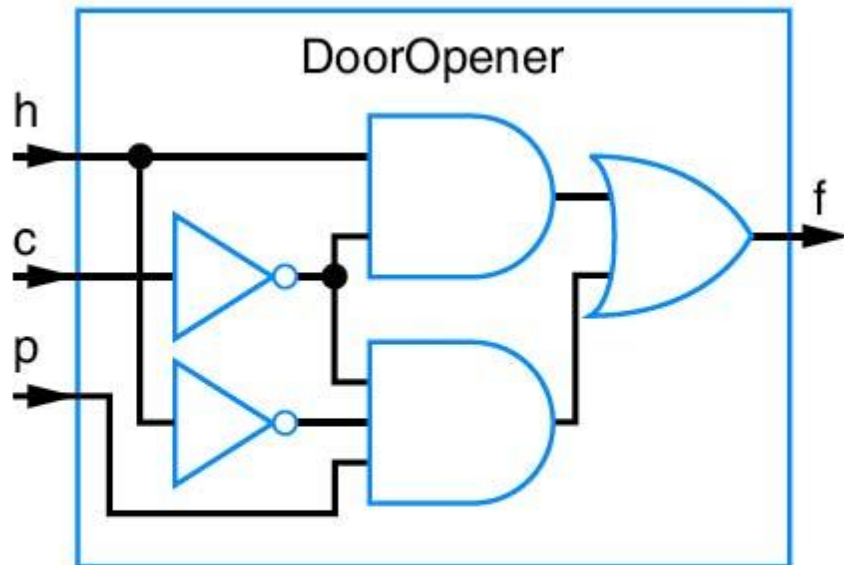
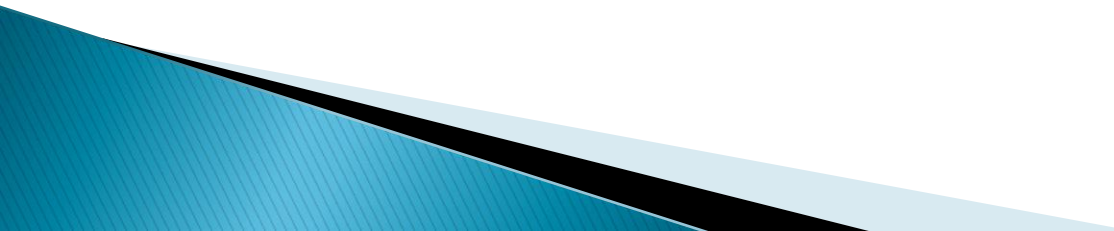


Figure 2.23 Initial door opener circuit.

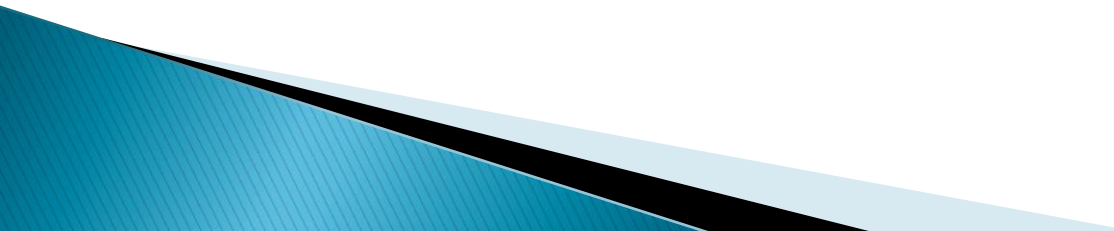
Exercícios

- ▶ Use as propriedades da Álgebra de Boole para mostrar que $f = c'hp + c'hp' + c'h'p$ é equivalente a $g = hc' + h'pc'$
- ▶ Determine se $F = a.(a+b)'$ e $G = a+b'$ são equivalentes usando as propriedades da álgebra de Boole
- ▶ Se $F = (ab' + c)$, determine a expressão para $G = F'$ em forma de soma de produtos

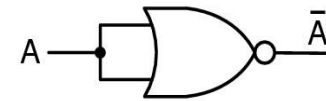
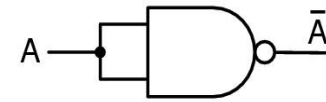
Conjunto universal de portas

- ▶ Um conjunto universal ou conjunto lógico completo é capaz de implementar qualquer função combinacional
 - ▶ Como qualquer expressão booleana pode ser implementada usando portas AND, OR e NOT, este é um conjunto universal
 - ▶ Um conjunto de portas é universal se é possível implementar AND, OR e NOT com as suas portas
- 

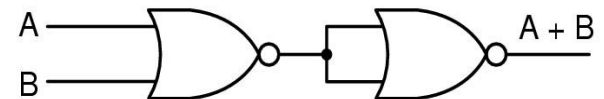
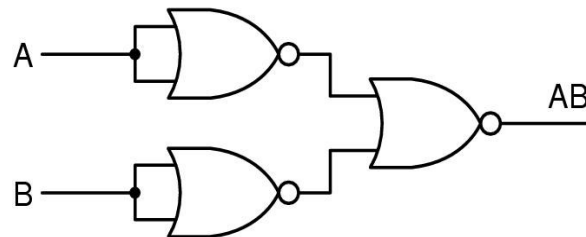
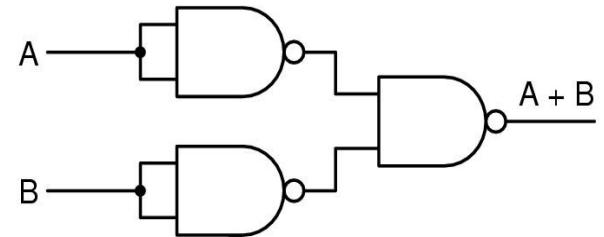
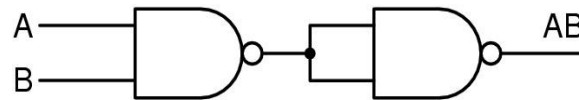
Conjunto universal de portas

- ▶ Conjuntos universais:
 - ▶ NAND
 - ▶ NOR
 - ▶ AND e NOT
 - ▶ OR e NOT
- 

Conjunto universal de portas



(a)

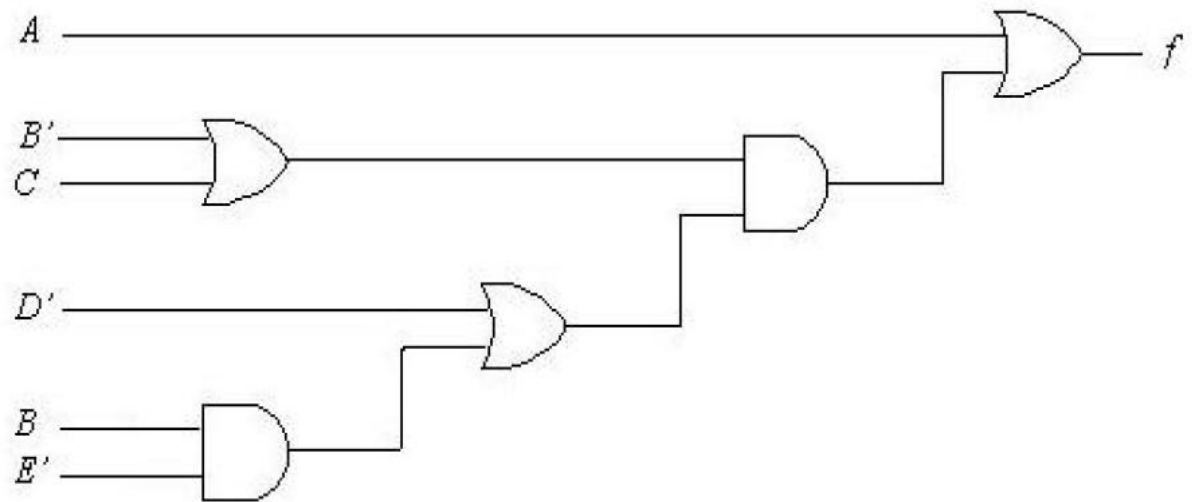


(b)

(c)

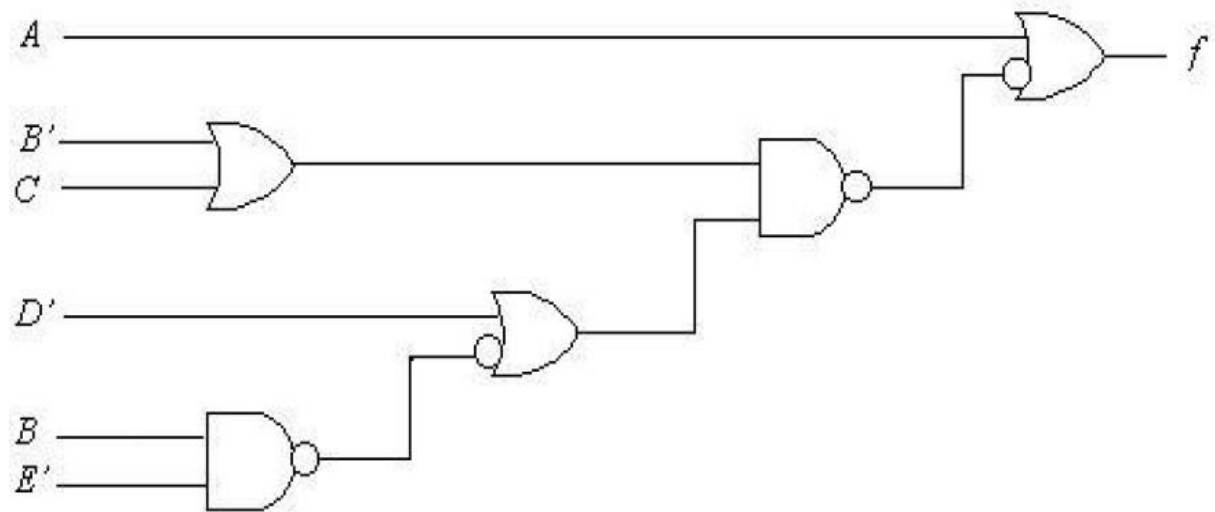
Representação de circuitos com NAND

- ▶ Implemente $f(A,B,C,D,E)=A+(B'+C).(D'+BE')$ usando apenas portas NAND de duas entradas
- ▶ Passo 1: obtenha o circuito lógico com portas OR e AND



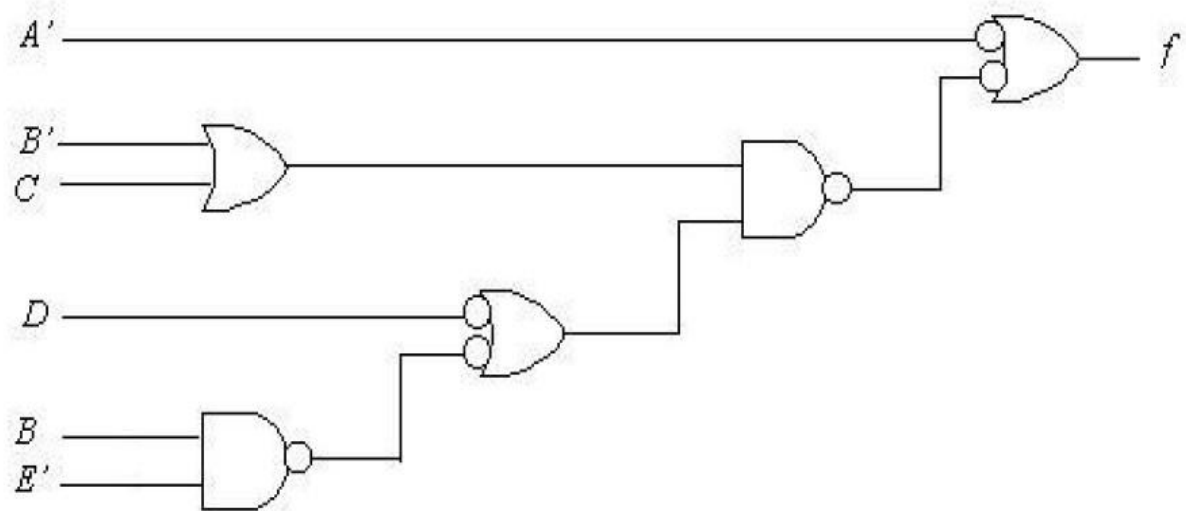
Representação de circuitos com NAND

- ▶ Passo 2: acrescenta negações nas saídas das portas AND e no final da mesma linha de conexão para não mudar a lógica



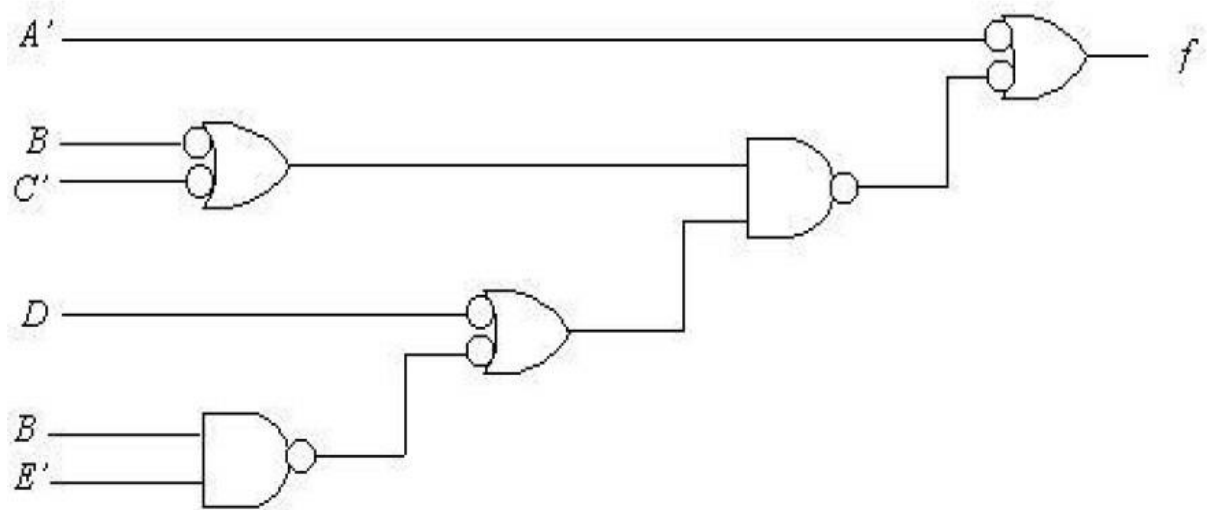
Representação de circuitos com NAND

- ▶ Passo3: acrescente negações nas demais entradas das portas OR que já receberam alguma negação



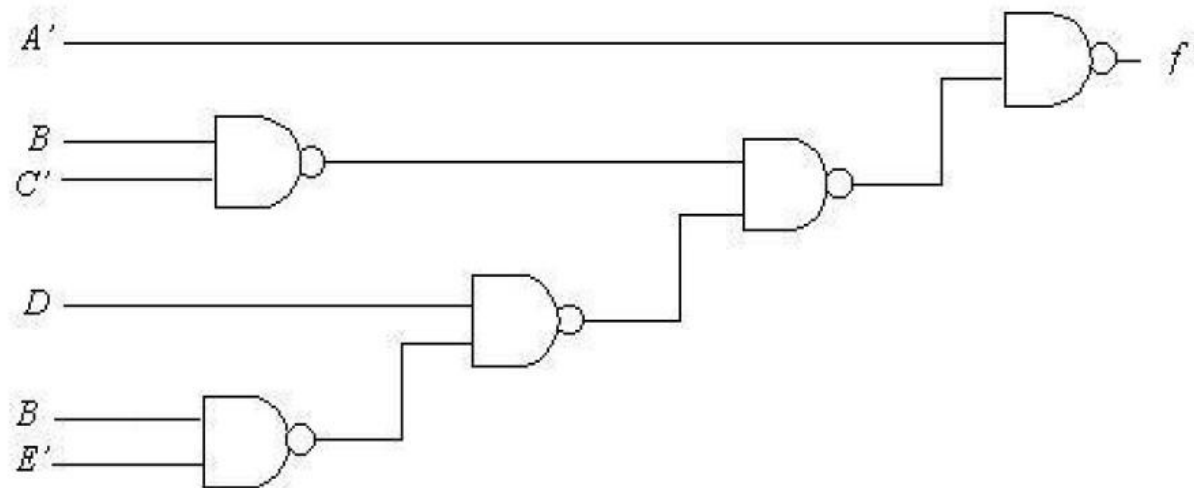
Representação de circuitos com NAND

- ▶ Passo 4: Negue as entradas das portas OR que ainda não foram alteradas



Representação de circuitos com NAND

- ▶ Passo 5: Converta as portas OR com entradas negadas por portas NAND

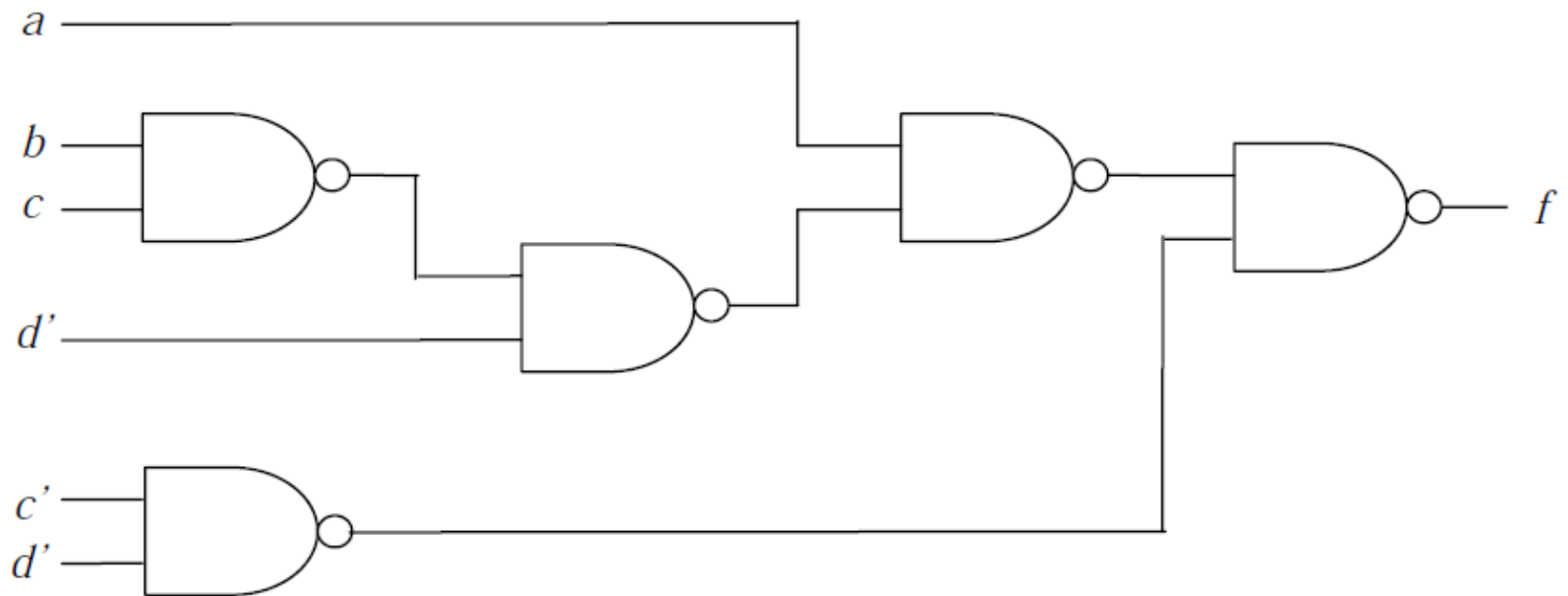


Exercício

- ▶ 1 – Implemente o circuito da função booleana f usando apenas portas NAND de 2 entradas
 $f(a, b, c, d) = abc + ad + c'd'$

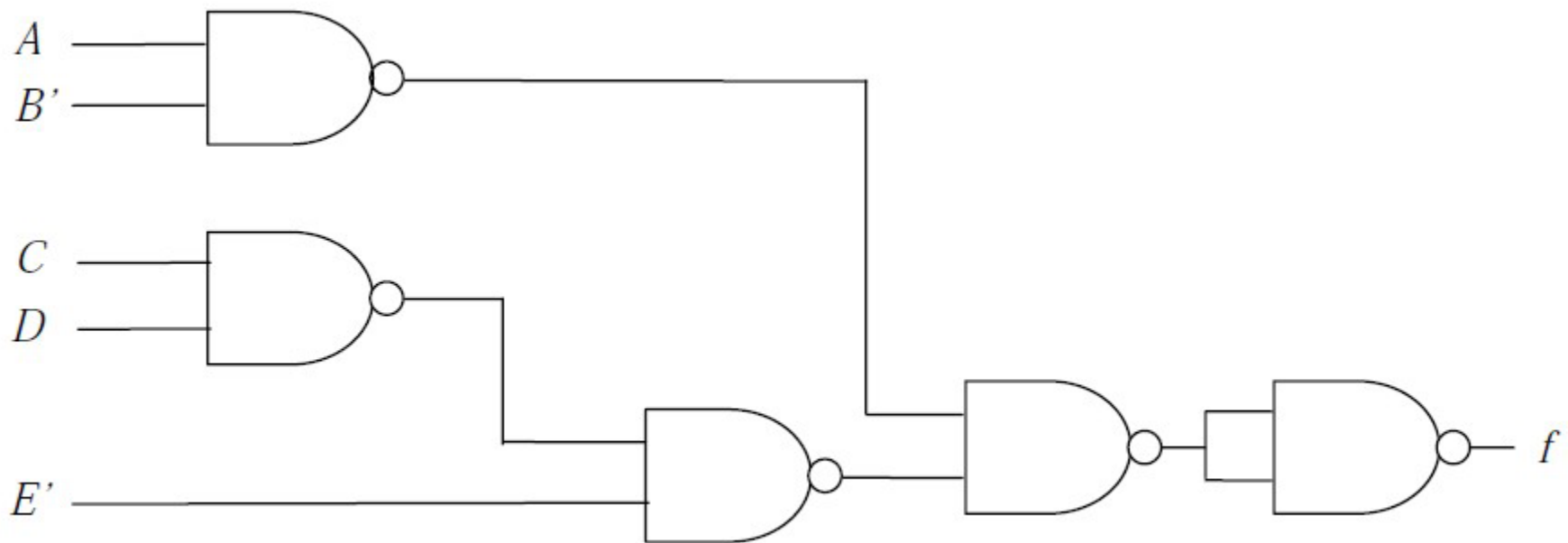
Exercício

- 1 – Implemente o circuito da função booleana f usando apenas portas NAND de 2 entradas
 $f(a, b, c, d) = abc + ad + c'd'$



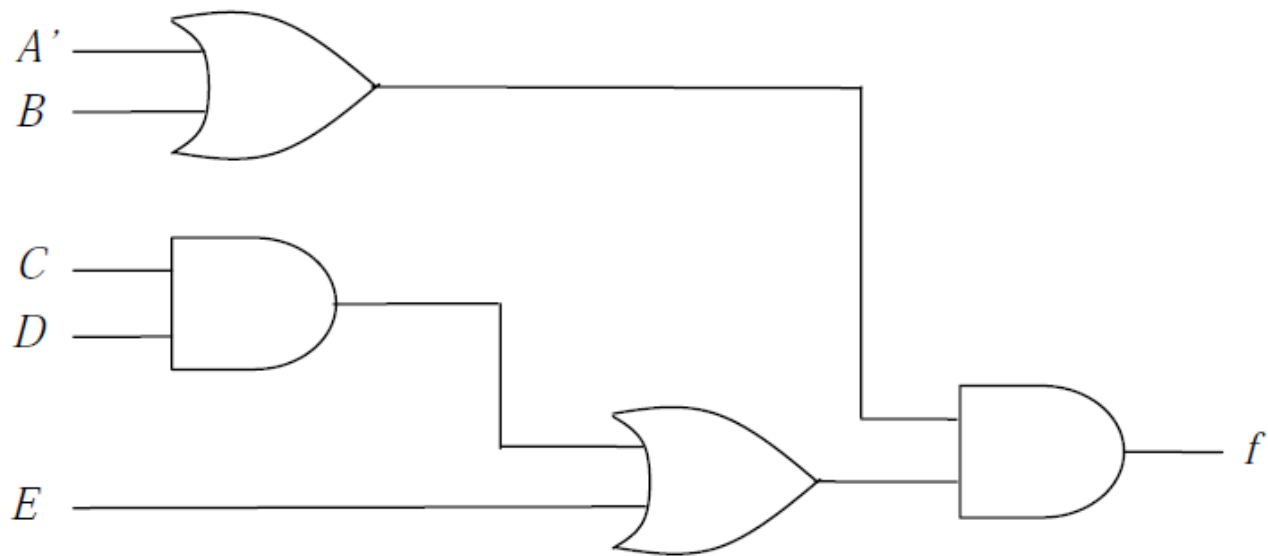
Exercício

- 2- Encontre a equação da função lógica representada pelo circuito abaixo



Exercício

- ▶ 2- Encontre a equação da função lógica representada pelo circuito abaixo
- ▶ Resposta: $f = (A' + B)(C.D + E)$

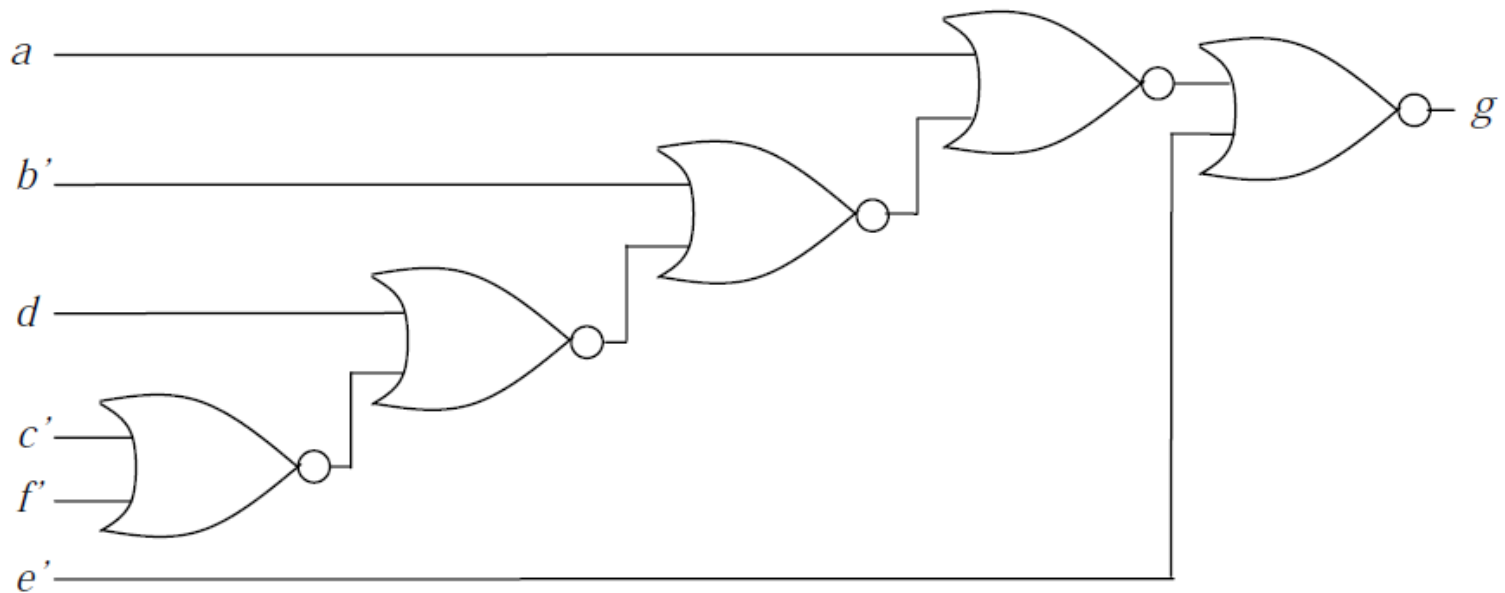


Representação de circuitos com NOR

- ▶ Implemente o circuito da função booleana g usando apenas portas NOR de 2 entradas
 $g(a, b, c, d, e, f) = ae + bde + bcef$

Exercício

- 3- Implemente o circuito da função booleana g usando apenas portas NOR de 2 entradas
 $g(a, b, c, d, e, f) = ae + bde + bcef$



Funções booleanas

- ▶ Uma função booleana F é um mapeamento de cada uma das combinações possíveis de valores das variáveis da função (as entradas) para 0 ou 1 (a saída)

Funções booleanas

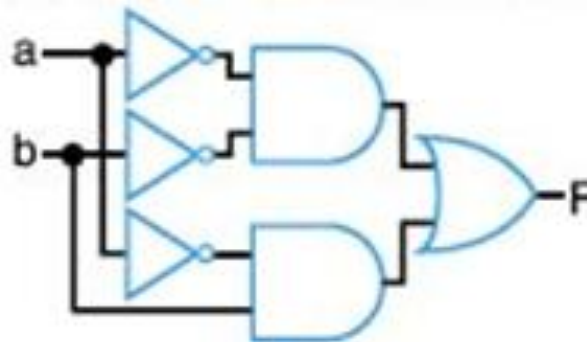
- ▶ Uma função booleana pode ser representada por meio de uma descrição:
 - A função f das variáveis a e b é 1 quando a é 0 e b é 0 ou quando a é 0 e b é 1.
 - A função f é 1 quando a é 0, independente do valor de b
- ▶ A função pode ser representada por uma destas expressões booleanas
 - $F = a'b' + a'b$
 - $F = a'$

Funções booleanas

- ▶ A função pode ser representada or meio de uma tabela verdade

a	b	F
0	0	1
0	1	1
1	0	0
1	1	0

- ▶ A função pode ser representada por um destes circuitos lógicos



Funções booleanas

- ▶ Tendo uma representação é possível passar para outra

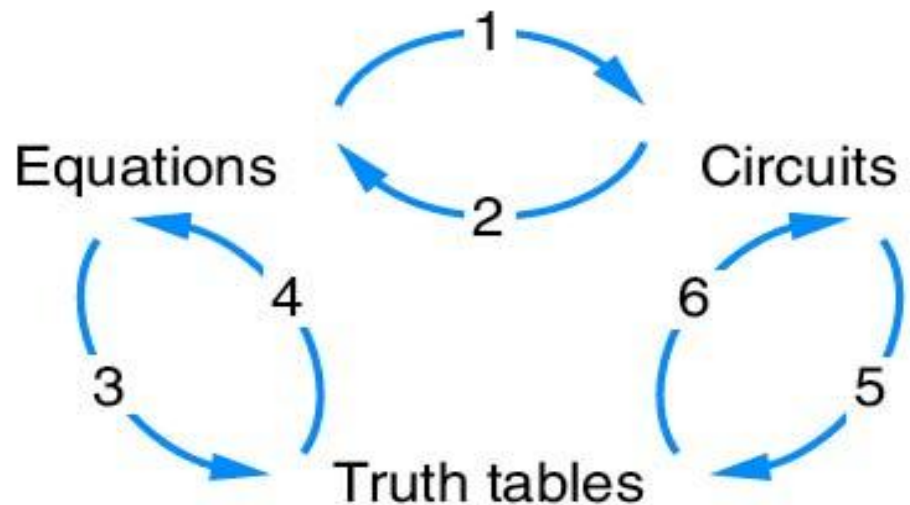
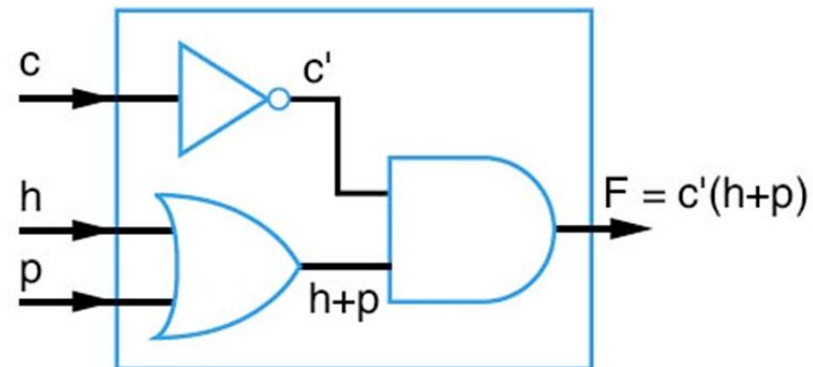


Figure 2.31 Possible conversions from one Boolean function representation to another.

Funções booleanas

- ▶ 1 – Equações em circuitos
 - Usando uma porta AND para cada operador (\cdot), uma porta OR para cada ($+$) e uma porta NOT para cada ($'$).
- ▶ 2 – Circuitos em equações
 - Começando com as entradas e, então, escrevendo a saída de cada porta como uma expressão que contém as entradas da porta. Isto é feito até alcançar a saída do circuito.



Funções booleanas

- ▶ 3– Equações em tabela-verdade
 - Montando uma tabela com as entradas em todas as combinações possíveis e calculando o valor da expressão para cada combinação.

Inputs				Output
a	b	$a' b'$	$a' b$	F
0	0	1	0	1
0	1	0	1	1
1	0	0	0	0
1	1	0	0	0

Figure 2.34 Truth table for $F(a,b)=a'b'+a'b$ with intermediate columns.

Funções booleanas

- ▶ 4– Tabela-verdade em equação
 - Criando um termo de produto para cada linha na tabela e aplicando um OR a todos os termos de produto.

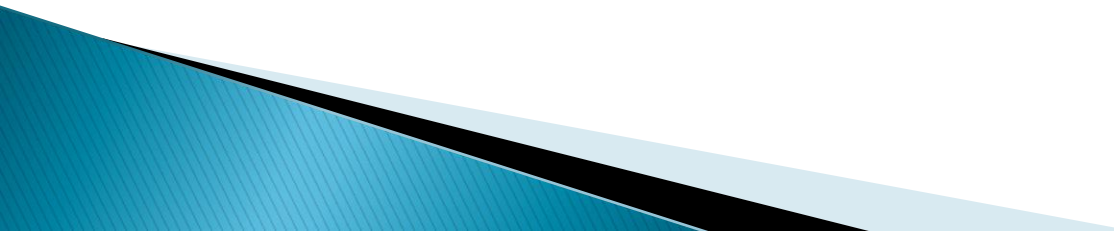
Inputs		Outputs	Term
a	b	F	F = sum of
0	0	1	$a' b'$
0	1	1	$a' b$
1	0	0	
1	1	0	

Figure 2.35 Converting a truth table to an equation.

Funções booleanas

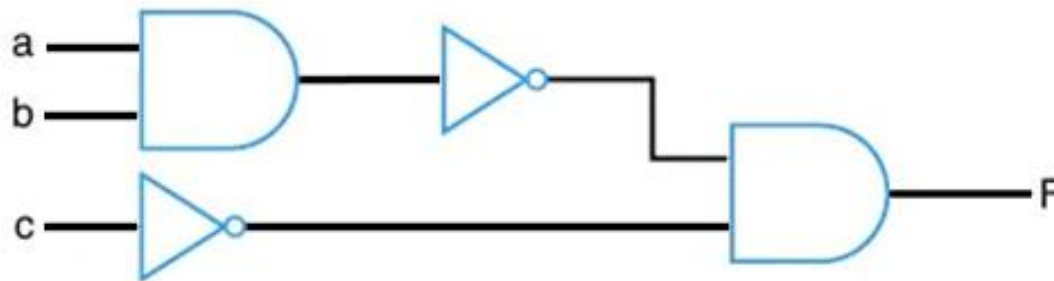
- ▶ 5– Circuito em tabela-verdade
 - Convertendo o circuito em uma equação e depois a equação em tabela verdade.
 - Avaliar no circuito o valor da saída para cada combinação das entradas é uma operação muito propensa a erros.
- ▶ 6– Tabela-verdade em circuito
 - Convertendo a tabela em equação e depois em circuito.

Exercício

- ▶ Descreva uma função que gera um bit de paridade par P para 3 bits de dados a , b e c . O bit de paridade deve ser 1 se o número de 1s em abc for ímpar e deve ser 0 caso contrário.
 - ▶ Qual é a representação mais fácil de usar para descrever a função booleana?
- 

Exercício

- ▶ Represente a função booleana do circuito da figura em uma tabela-verdade



- ▶ Neste caso é melhor escrever a expressão booleana antes e depois montar a tabela-verdade

Para ser continuado...