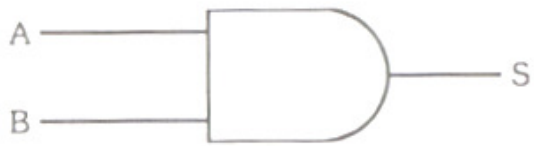
Anhanguera

FACULDADE ANHANGUERA DE JUNDIAÍ
Rua do Retiro, 3.000 • Retiro • Jundiaí (SP) • 13209-002 • (11) 3109-0600

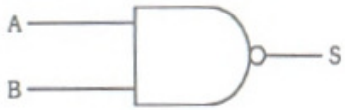
CIRCUITOS DIGITAIS
Funções e expressões lógicas

Prof. Rogério Moreira

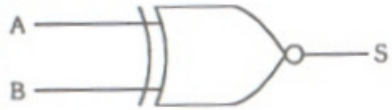
BLOCOS LÓGICOS BÁSICOS

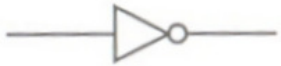
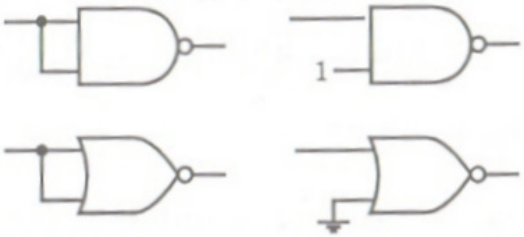
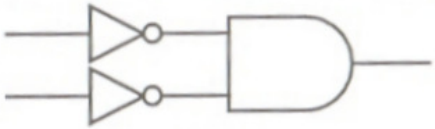
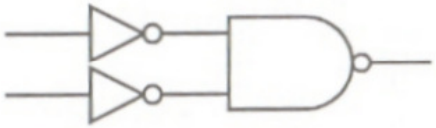
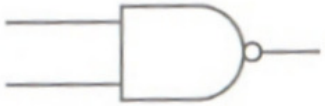
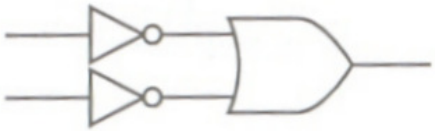
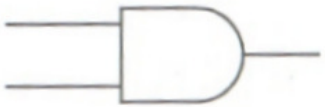
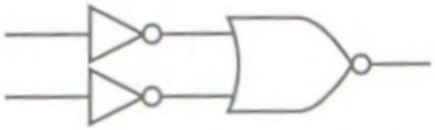
Porta	Símbolo Usual	Tabela da Verdade	Função Lógica	Expressão
E AND		A B S 0 0 0 0 1 0 1 0 0 1 1 1	**Função E:** assume 1 quando todas as variáveis forem 1 e 0 nos outros casos.	S = AB
OU OR		A B S 0 0 0 0 1 1 1 0 1 1 1 1	**Função OU:** assume 0 quando todas as variáveis forem 0 e 1 nos outros casos.	S = A + B

BLOCOS LÓGICOS BÁSICOS

Porta	Símbolo Usual	Tabela da Verdade	Função Lógica	Expressão
NÃO NOT INVERSOR		A S 0 1 1 0	**Função NÃO:** inverte a variável aplicada à sua entrada.	$S=\overline{A}$
NE NAND		A B S 0 0 1 0 1 1 1 0 1 1 1 0	**Função NE**: inverso da função E.	$S=\overline{AB}$
NOU NOR		A B S 0 0 1 0 1 0 1 0 0 1 1 0	**Função NOU:** inverso da função OU.	$S=\overline{A+B}$

BLOCOS LÓGICOS BÁSICOS

Porta	Símbolo Usual	Tabela da Verdade	Função Lógica	Expressão
OU EXCLUSIVO **EXCLUSIVE OR**		A B S 0 0 0 0 1 1 1 0 1 1 1 0	**Função OU** Exclusivo: assume 1 quando as variáveis assumirem valores diferentes entre si.	$S=\overline{A}.B+A.\overline{B}$ $S=A\oplus B$
NOU EXCLUSIVO **EXCLUSIVE NOR** **COINCIDÊNCIA**		A B S 0 0 1 0 1 0 1 0 0 1 1 1	**Função Coincidência:** assume 1 quando houver coincidência entre os valores das variáveis.	$S=\overline{A}.\overline{B}+A.B$ $S=A\odot B$

BLOCO LÓGICO	BLOCO EQUIVALENTE
	
	
	
	
	

AND – Logical AND

Description:

Performs the logical AND between the contents of register Rd and register Rr and places the result in the destination register Rd.

Operation:

(i) Rd ← Rd • Rr

	Syntax:	Operands:	Program Counter:
(i)	AND Rd,Rr	$0 \le d \le 31$, $0 \le r \le 31$	PC ← PC + 1

16-bit Opcode:

0010	00rd	dddd	rrrr

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
–	–	–	⇔	0	⇔	⇔	–

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
and   r2,r3     : Bitwise and r2 and r3, result in r2
ldi   r16,1     : Set bitmask 0000 0001 in r16
and   r2,r16    : Isolate bit 0 in r2
```

Words: 1 (2 bytes)
Cycles: 1

ANDI – Logical AND with Immediate

Description:

Performs the logical AND between the contents of register Rd and a constant and places the result in the destination register Rd.

Operation:

(i) Rd ← Rd • K

	Syntax:	Operands:	Program Counter:
(i)	ANDI Rd,K	$16 \le d \le 31$, $0 \le K \le 255$	PC ← PC + 1

16-bit Opcode:

0111	KKKK	dddd	KKKK

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
–	–	–	⇔	0	⇔	⇔	–

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
andi  r17,$0F   : Clear upper nibble of r17
andi  r18,$10   ; Isolate bit 4 in r18
andi  r19,$AA   : Clear odd bits of r19
```

Words: 1 (2 bytes)
Cycles: 1

EOR – Exclusive OR

Description:

Performs the logical EOR between the contents of register Rd and register Rr and places the result in the destination register Rd.

Operation:

(i) $Rd \leftarrow Rd \oplus Rr$

	Syntax:	Operands:	Program Counter:
(i)	EOR Rd,Rr	$0 \le d \le 31$, $0 \le r \le 31$	$PC \leftarrow PC + 1$

16-bit Opcode:

0010	01rd	dddd	rrrr

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
–	–	–	⇔	0	⇔	⇔	–

S: $N \oplus V$, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
eor     r4,r4       ; Clear r4
eor     r8,r22      ; Bitwise exclusive or between r8 and r22
```

Words: 1 (2 bytes)
Cycles: 1

OR – Logical OR

Description:

Performs the logical OR between the contents of register Rd and register Rr and places the result in the destination register Rd.

Operation:

(i) Rd ← Rd ∨ Rr

	Syntax:	Operands:	Program Counter:
(i)	OR Rd,Rr	$0 \le d \le 31$, $0 \le r \le 31$	PC ← PC + 1

16-bit Opcode:

0010	10rd	dddd	rrrr

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
–	–	–	⇔	0	⇔	⇔	–

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
        or      r15,r16     ; Do bitwise or between registers
        bst     r15,6       ; Store bit 6 of r15 in T Flag
        brts    ok          ; Branch if T Flag set
        ...
ok:     nop                 ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

ORI – Logical OR with Immediate

Description:

Performs the logical OR between the contents of register Rd and a constant and places the result in the destination register Rd.

Operation:

(i) Rd ← Rd ∨ K

	Syntax:	Operands:	Program Counter:
(i)	ORI Rd,K	$16 \le d \le 31$, $0 \le K \le 255$	PC ← PC + 1

16-bit Opcode:

0110	KKKK	dddd	KKKK

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
–	–	–	⇔	0	⇔	⇔	–

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \bullet \overline{R6} \bullet \overline{R5} \bullet \overline{R4} \bullet \overline{R3} \bullet \overline{R2} \bullet \overline{R1} \bullet \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
ori     r16,$F0   ; Set high nibble of r16
ori     r17,1     ; Set bit 0 of r17
```

Words: 1 (2 bytes)
Cycles: 1

Exercício

Desenhe o sinal na saída S do circuito da Figura 2.58.

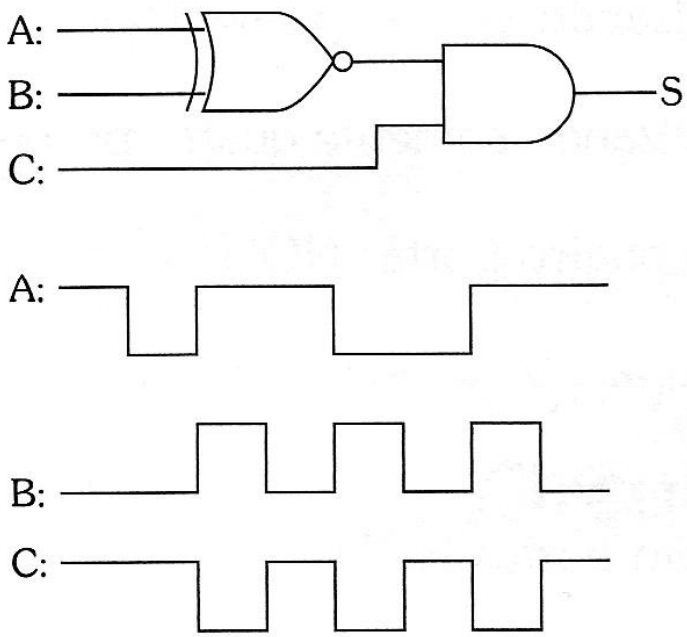
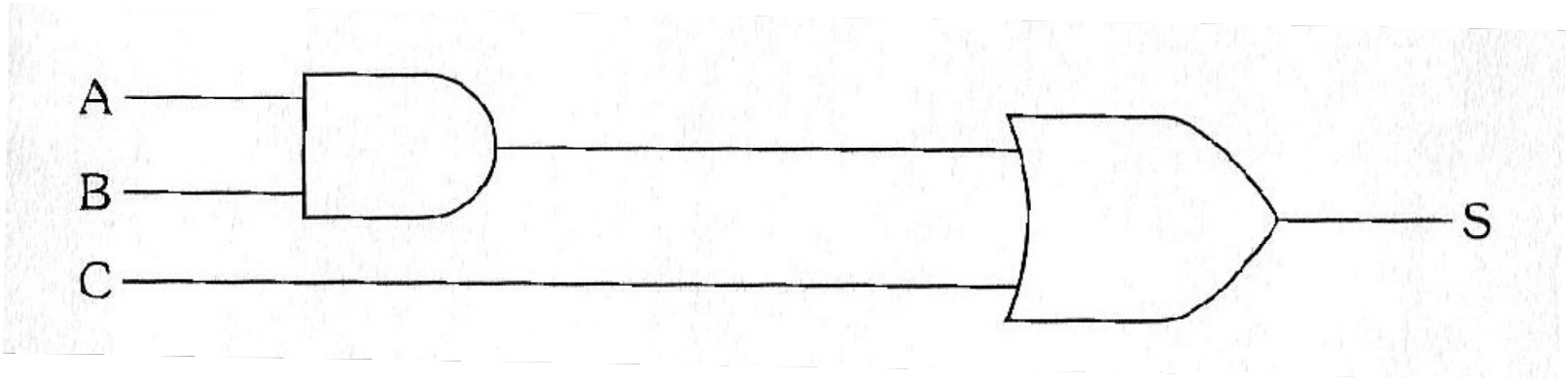
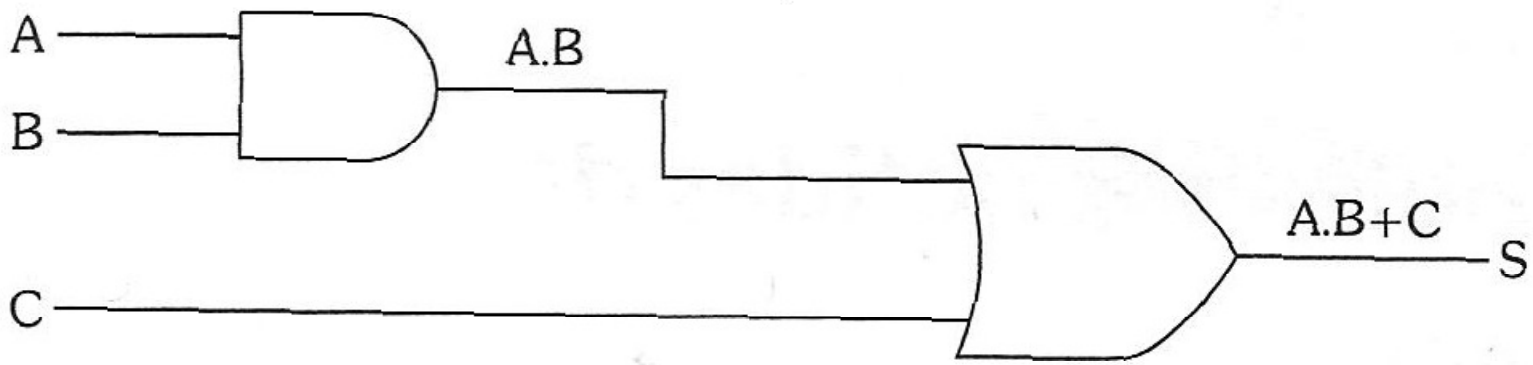


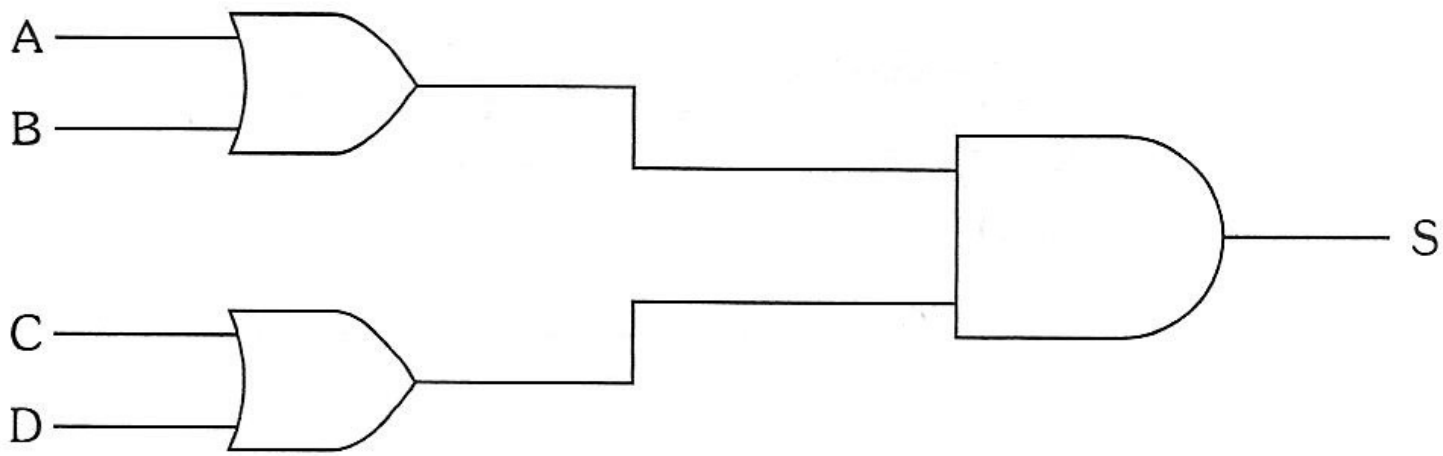
Figura 2.58

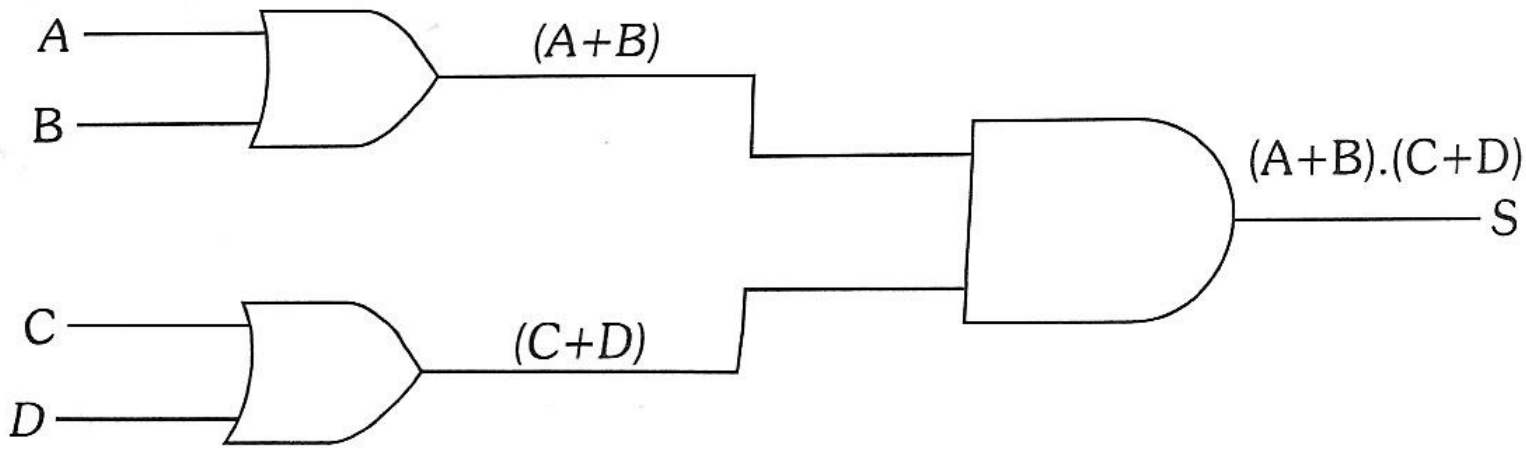
1. Expressões booleanas obtidas de circuitos lógicos
2. Circuitos lógicos obtidos de expressões booleanas
3. Tabelas verdade obtidas de expressões booleanas
4. Expressões booleanas obtidas de tabelas verdade

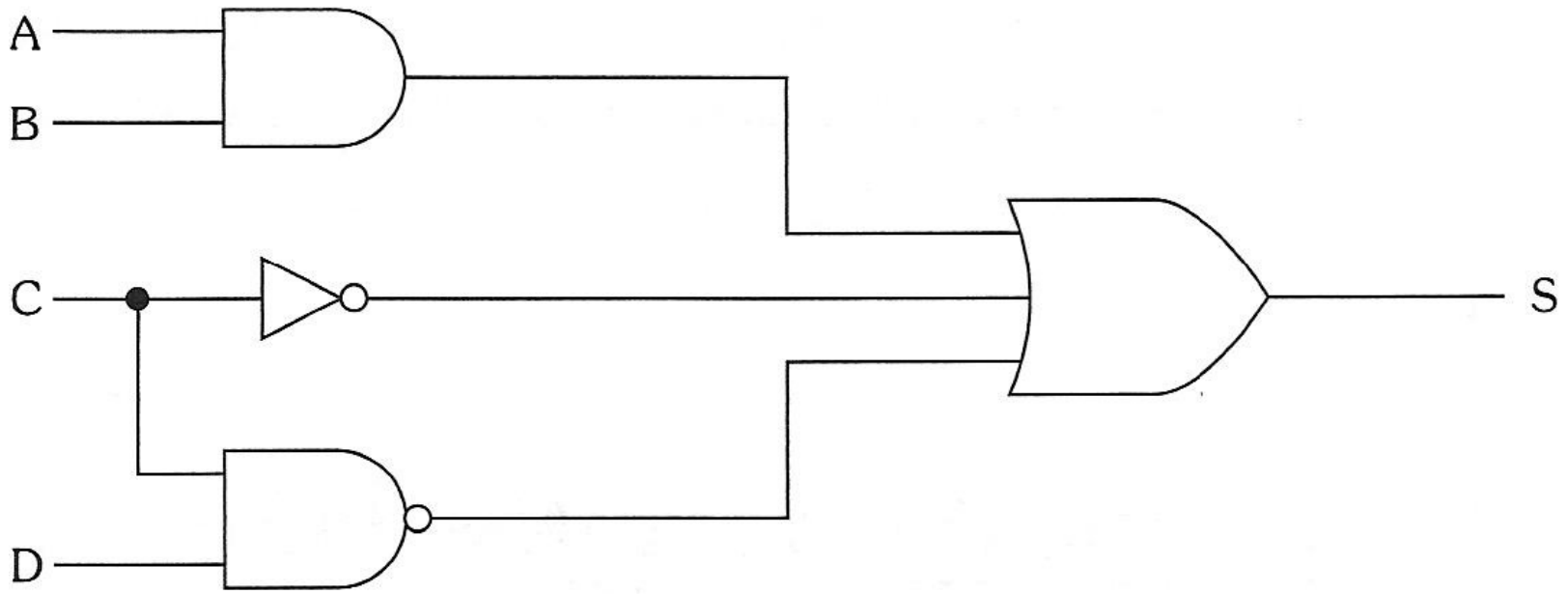
1. **Expressões booleanas obtidas de circuitos lógicos**
2. Circuitos lógicos obtidos de expressões booleanas
3. Tabelas verdade obtidas de expressões booleanas
4. Expressões booleanas obtidas de tabelas verdade

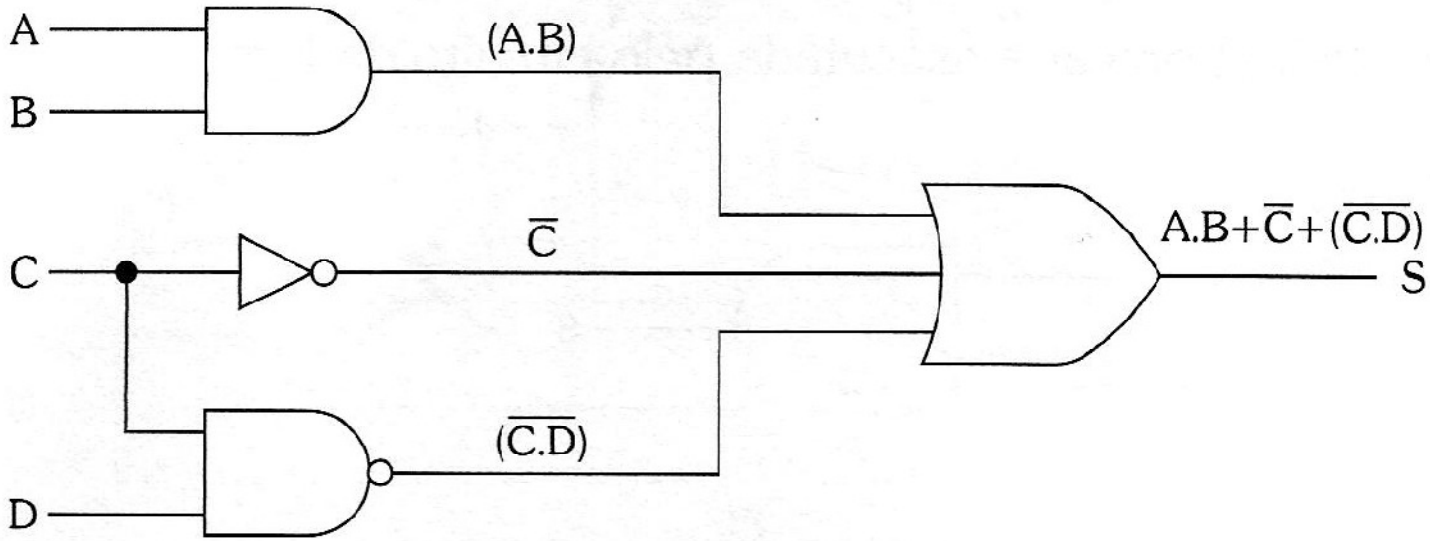

A
B
C
S

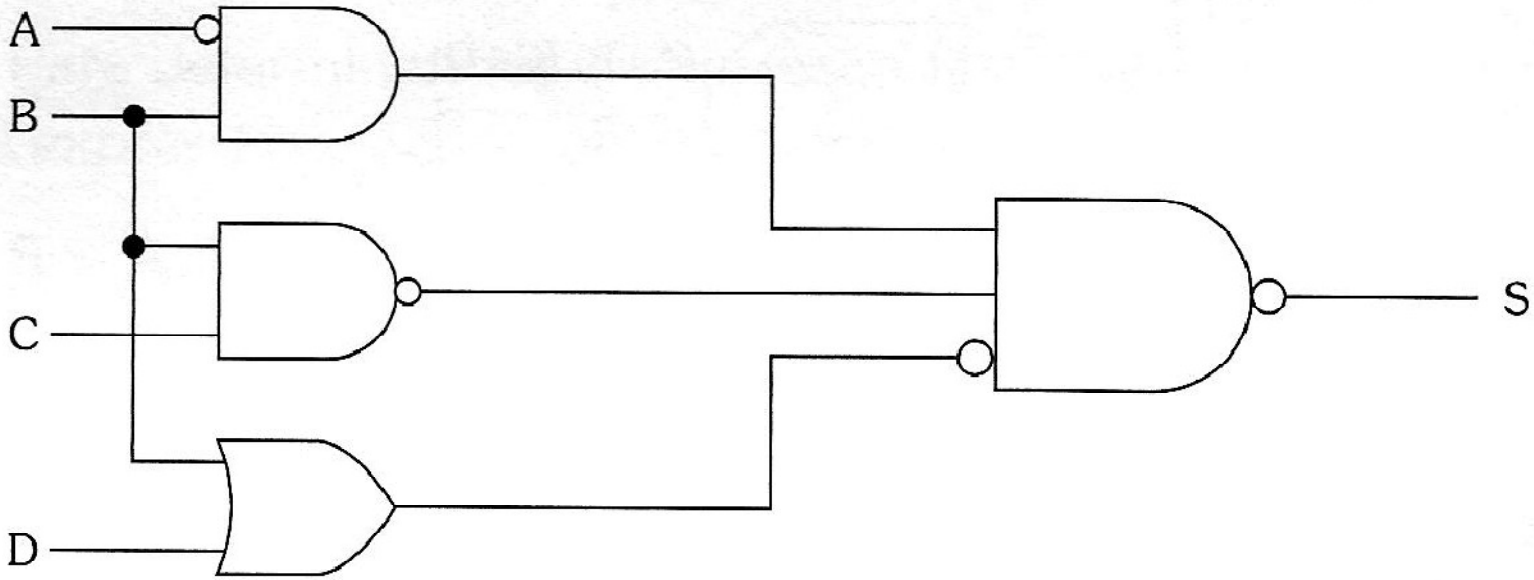

A
B
C
A.B
A.B+C
S

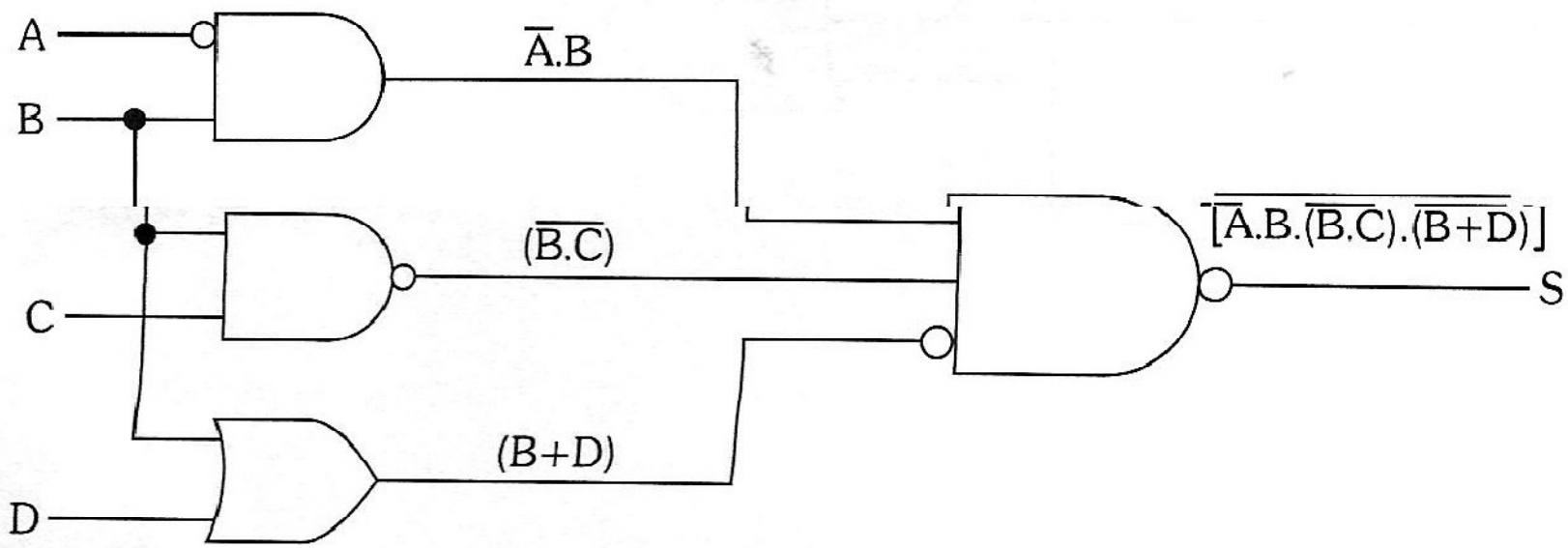

A
B
C
D
S


A
B
(A+B)
C
D
(C+D)
(A+B).(C+D)
S


A
B
C
D
S


A
B
(A.B)
C
$\overline{C}$
$A.B+\overline{C}+(\overline{C.D})$
S
$(\overline{C.D})$
D


A
B
C
D
S



$$\therefore S = \left[\overline{\overline{A}.B.(\overline{B.C}).(\overline{B+D})}\right]$$

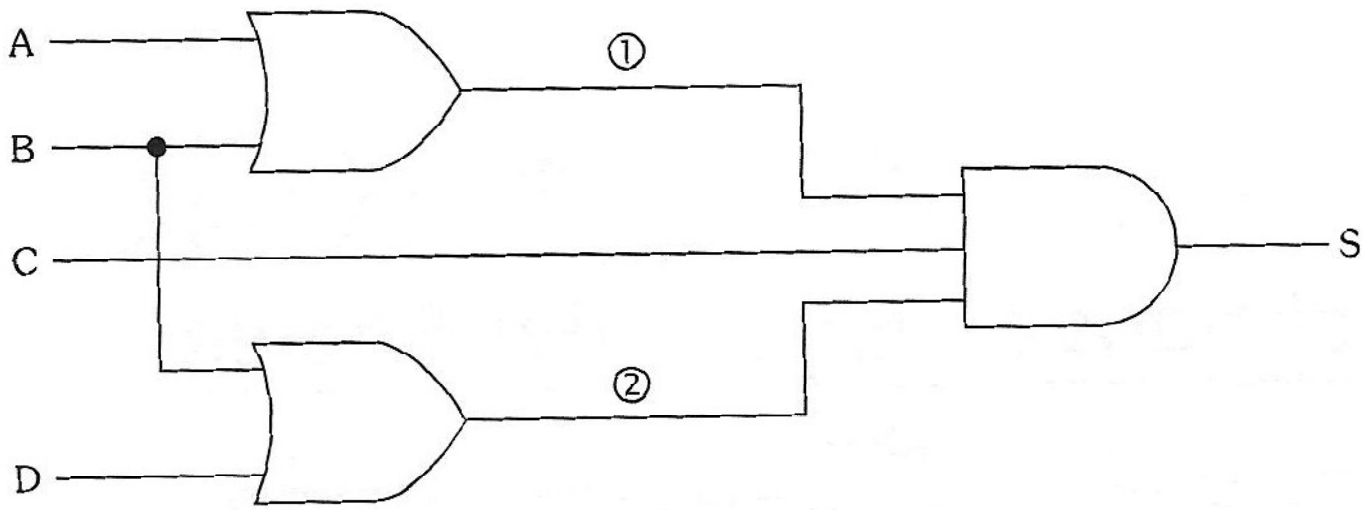
1. Expressões booleanas obtidas de circuitos lógicos

2. **Circuitos lógicos obtidos de expressões booleanas**

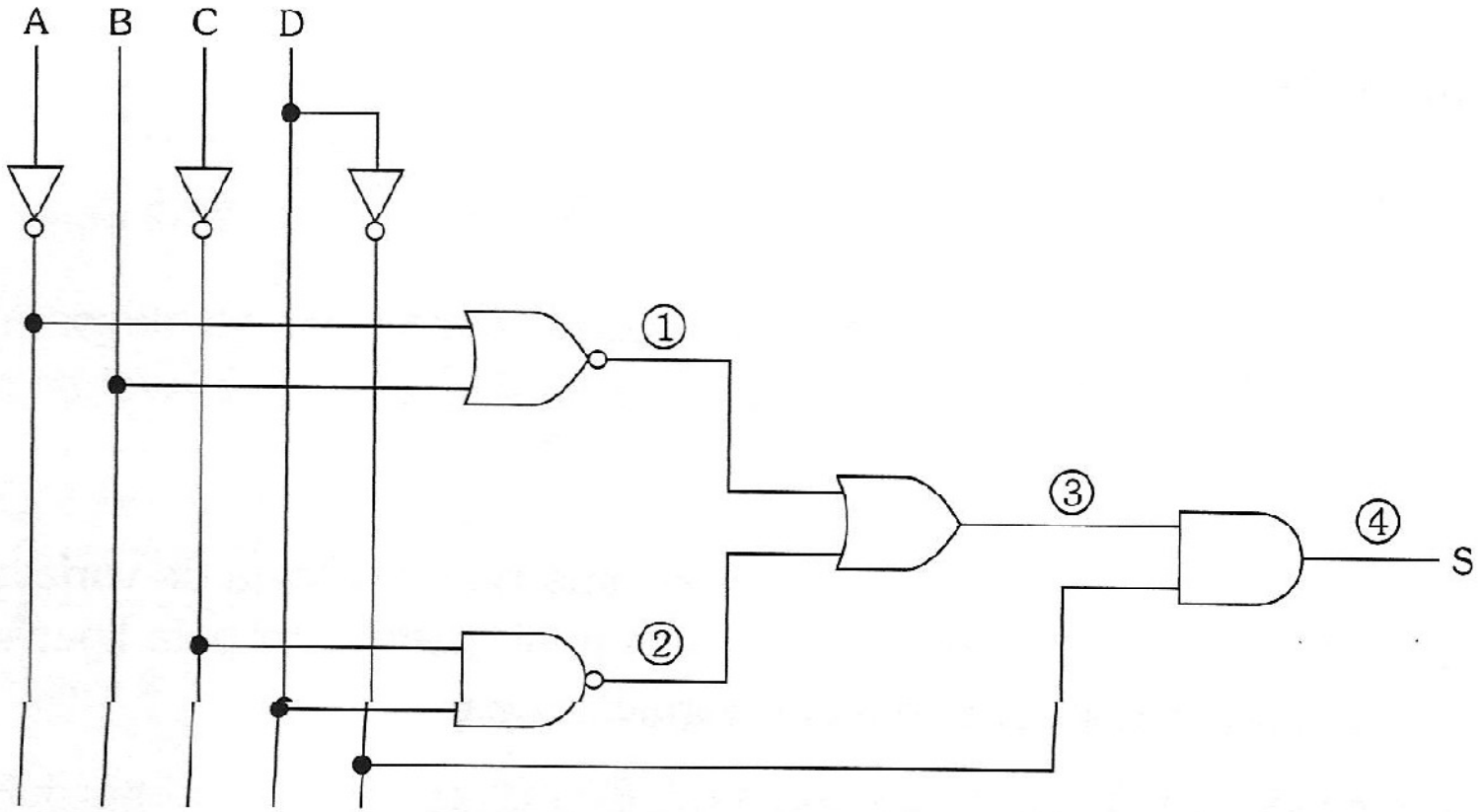
3. Tabelas verdade obtidas de expressões booleanas

4. Expressões booleanas obtidas de tabelas verdade

$$S = (A + B) . C . (B + D)$$


A
B
C
D
①
②
S

$$S = [(\overline{\overline{A} + B}) + (\overline{\overline{C} . D})] . \overline{D}$$


A
B
C
D
①
②
③
④
S

1. Expressões booleanas obtidas de circuitos lógicos
2. Circuitos lógicos obtidos de expressões booleanas
3. **Tabelas verdade obtidas de expressões booleanas**
4. Expressões booleanas obtidas de tabelas verdade

$$S = A \cdot \overline{B} \cdot C + A \cdot \overline{D} + \overline{A} \cdot B \cdot D.$$

A	B	C	D	1º membro $A . \overline{B} . C$	2º membro $A . \overline{D}$	3º membro $\overline{A} . B . D$	Resultado final S
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	1	1
0	1	1	0	0	0	0	0
0	1	1	1	0	0	1	1
1	0	0	0	0	1	0	1
1	0	0	1	0	0	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	0	0	1
1	1	0	0	0	1	0	1
1	1	0	1	0	0	0	0
1	1	1	0	0	1	0	1
1	1	1	1	0	0	0	0

1. Expressões booleanas obtidas de circuitos lógicos
2. Circuitos lógicos obtidos de expressões booleanas
3. Tabelas verdade obtidas de expressões booleanas
4. **Expressões booleanas obtidas de tabelas verdade**

#	A	B	S
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	1

Caso 00: $S = 1$ quando $A = 0$ e $B = 0$ ($\overline{A} = 1$ e $\overline{B} = 1$) $\Rightarrow \overline{A} \,.\, \overline{B}$

Caso 10: $S = 1$ quando $A = 1$ e $B = 0$ ($A = 1$ e $\overline{B} = 1$) $\Rightarrow A \,.\, \overline{B}$

Caso 11: $S = 1$ quando $A = 1$ e $B = 1 \Rightarrow A \,.\, B$

$$\therefore\ S = \overline{A} \,.\, \overline{B} + A \,.\, \overline{B} + A \,.\, B$$

#	A	B	C	S
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Para solucionar, extraímos os casos em que a expressão é verdadeira (S = 1): 000 ou 010 ou 110 ou 111.

$$\therefore S = \overline{A}.\overline{B}.\overline{C} + \overline{A}.B.\overline{C} + A.B.\overline{C} + A.B.C$$