Sistemas Numéricos

- Nosso sistema -> decimal ou de base 10

  Inventado por matemáticos hindus 2000 anos atrás e posteriormente adotado pelos árabes que o introduziram aos europeus. Também denominado sistema arábico por que utiliza símbolos arábicos para representar os dez dígitos (dedo em Latim) que a base suporta.

  A representação de qualquer número na base 10 é posicional, isto é, cada digito que o compõe assume um "peso" dependendo da posição que ocupa na representação.
  Ex:

$$794 = 7 \times 10^2 + 9 \times 10^1 + 4 \times 10^0$$

- Propriedades dos sistemas numéricos:

  - o número de dígitos usados no sistema é sempre igual a base.
  - o maior dígito é igual ao valor da base -1.
  - o valor que cada dígito assume na notação de um número é ponderado e é igual ao seu valor absoluto multiplicado pela base elevada `a posição relativa do dígito menos 1.

Sistemas Numéricos

- Exemplos:

a) $426_7 = (4 \times 7^2) + (2 \times 7^1) + (6 \times 7^0) = 216_{10}$

b) $7777_8 = (7 \times 8^3) + (7 \times 8^2) + (7 \times 8^1) + (7 \times 8^0) = 4095_{10}$

c) $1101_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 13_{10}$

CONVERSÃO ENTRE BASES:

- Dado um número N expresso em uma base s achar sua representação na base r.

Dois métodos:
divisões sucessivas e desenvolvimento polinomial

a) Divisões sucessivas:

$$
\begin{array}{llll}
N & \underline{|\,r} & & \\
B_0 & N_1 & \underline{|\,r} & \\
 & B_1 & N_2 & \\
 & & \vdots & \\
 & & N_{m-1} & \underline{|\,r} \\
 & & B_{m-1} & N_m \quad \underline{|\,r} \\
 & & & B_m \quad\; 0
\end{array}
\qquad B_m = N_m
$$

Sistemas Numéricos

- Se s > r os restos B obtidos já são os dígitos procurados, ou seja, $N_r = B_m\, B_{m-1}\, B_{m-2} --- B_0$.
- Se s < r os restos B devem ser primeiramente convertidos para a base r.
- Demonstração:

$$N_s = N_1 \,.\, r + B_0$$

$$N_1 = N_2 \,.\, r + B_1$$

$$\vdots$$

$$N_m = 0 \,.\, r + B_m$$

$$N_s = (N_2 \,.\, r + B_1)\, r + B_0 = N_2 \,.\, r^2 + B_1 \,.\, r + B_0$$

$$N_s = (N_3 \,.\, r + B_2)\, r^2 + B_1 \,.\, r + B_0$$

$$N_s = B_m \,.\, r^m + B_{m-1} \,.\, r^{m-1} + \_\_\_ + B_1 \,.\, r + B_0 = N_r$$

Sistemas Numéricos

- Exemplos:

a) $s = 10,\ N_s = 69,\ r = 2,\ N_r = ?$

```
69 |2
 1  34 |2
     0  17 |2
         1  8 |2
            0  4 |2
               0  2 |2
                  0  1 |2
                     1  0
```

$$N_r = 1000101_2$$

b) $s = 2,\ N_2 = 1000101,\ r = 10,\ N_r = ?$

$$r = 10_{10} = 1010_2$$

```
 1000101 |1010
-1010     110 |1010
  1110          0
 -1010
  (1001)    (110)
```

$$N_r = (110)_2\ (1001)_2 = 69_{10}$$

b) Desenvolvimento polinomial:

Dado $N_s = A_n \_\_\_ A_1 A_0$, N_r é obtido avaliando a expressão $N_r = A_n s^n + \_\_ + A_1 s + A_0$ no sistema de base r.

- Se $s < r$ a expressão é avaliada diretamente.
- Se $s > r$ é necessário primeiramente converter a base s e os dígitos A para a base r.

Exemplos:

a) $s = 2$, $N_s = 1110101$, $r = 10$, $N_r = ?$

$$N_r = 1x2^6 + 1x2^5 + 1x2^4 + 1x2^2 + 1 = 117$$

b) $s = 10$, $N_s = 117$, $r = 2$, $N_r = ?$

$$N_r = 1 \times (10_2)^2 + 1 \times (10_2)^1 + 7_2$$

$$N_r = 1010^2 + 1010 + 111$$

$$N_r = 1100100 + 10001 = 1110101$$

c) $s = 7$, $N_s = 65$, $r = 2$, $N_r = ?$

Sistemas Numéricos

- Mudança de base de números reais:

$N_R = N_I + N_F$, onde

$N_I = A_n r^n + \_\_ + A_1 r^1 + A_0$

$N_F = A_{-1} r^{-1} + A_{-2} r^{-2} + A_{-3} r^{-3} + \_\_$

- A parte fracionária de um número em uma dada base corresponde sempre \`a parte fracionária de sua representação em uma outra base.

- Cálculo dos digitos A_{-1}, A_{-2}, A_{-3},—

Multiplicando N_F por r temos:

$r N_F = A_{-1} + A_{-2} r^{-1} + A_{-3} r^{-2} + \_\_$, onde

A_{-1} é a parte inteira de $r N_F$.

Subtraindo A_{-1} de $r N_F$ e multiplicando novamente por r a expressão resultante temos:

$r (r N_F - A_{-1}) = A_{-2} + A_{-3} r^{-1} + \_\_$

O processo continua até alcançar o número de de digitos desejado na parte fracionária.

Sistemas Numéricos

- **Exemplo:**

Converter $69{,}71_{10}$ para a base 2.

$N_1 = 1000101$ de um exemplo anterior.

$2 \times (0{,}71) = 1{,}42 \rightarrow A_{-1} = 1$
$2 \times (0{,}42) = 0{,}84 \rightarrow A_{-2} = 0$
$2 \times (0{,}84) = 1{,}68 \rightarrow A_{-3} = 1$
$2 \times (0{,}68) = 1{,}36 \rightarrow A_{-4} = 1$
$2 \times (0{,}36) = 0{,}72 \rightarrow A_{-5} = 0$
$2 \times (0{,}72) = 1{,}44 \rightarrow A_{-6} = 1$
$\vdots$

Resultado: $69{,}71_{10} = 1000101{,}101101_2$--

Conferindo:

$$1000101{,}101101 = 2^6 + 2^2 + 2^0 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} \ldots$$

$$= 64 + 4 + 1 + 0{,}5 + 0{,}125 + 0{,}0625 +$$

$$+ 0{,}015625 = 69{,}703125$$

Sistemas Numéricos

- Relação entre as bases 2, 8 e 16:

Além da base 2, as bases 8, e principalmente a 16, são muito importantes em computação pois, por serem múltiplas da base 2, permitem uma notação compacta de valores binários.

A conversão de um número em binário para o sistema octal ou hexadecimal é imediata bastando para tal partilhar convenientemente o número binário já que

$8 = 2^3$ e $16 = 2^4$.

Exemplo:

$$62{,}71_{10} = \underbrace{111}_{(7}\underbrace{110}_{6},\underbrace{101}_{5}\underbrace{101}_{5)_8}{}_2$$

$$62{,}71_{10} = \underbrace{0011}_{(3}\underbrace{1110}_{E},\underbrace{1011}_{B}\underbrace{0100}_{4)_{16}}$$

Sistema Decimal

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Sistema Hexadecimal

- ADIÇÃO EM DIVERSAS BASES

- BASE 10

Ex:

$$
\begin{array}{rrr}
 & (1) & \\
 & 6 & 7 \\
+ & 8 & 4 \\
\hline
1 & 5 & 1
\end{array}
\qquad\qquad
\begin{array}{rrr}
(1) & (1) & \\
 & 6 & 7 \\
+ & 8 & 4 \\
\hline
1 & (15) & (11) \\
- & 10 & 10 \\
\hline
1 & 5 & 1
\end{array}
$$

- BASE 7

Ex:

$$
\begin{array}{rrr}
(1) & (1) & \\
 & 2 & 5_7 \\
+ & 5 & 4_7 \\
\hline
1 & (8) & (9) \\
- & 7 & 7 \\
\hline
(1 & 1 & 2)_7
\end{array}
$$

- BASE 8 (OCTAL)

Ex:

$$
\begin{array}{rrr}
 & 7 & 7_8 \\
+ & 7 & 7_8 \\
\hline
\end{array}
$$

- Base 16

Ex:

$$
\begin{array}{rrrr}
 & (1) & (1) & \\
 & 6 & B & E_{16} \\
+ & 5 & 6 & 9_{16} \\
\hline
 & 12 & (18) & (23) \\
- & & 16 & 16 \\
\hline
 & (C & 2 & 7)_{16}
\end{array}
$$

$$
\begin{array}{rrrrr}
 & (1) & & (1) & \\
 & F & & 1 & A_{16} \\
+ & E & & 0 & 9_{16} \\
\hline
 & 1 & (29) & 2 & (19) \\
 & & 16 & & 16 \\
\hline
 & (1 & D & 2 & 3)_{16}
\end{array}
$$

$$
\begin{array}{rrrr}
 & 2 & F & F_{16} \\
+ & 1 & F & E_{16} \\
\hline
\end{array}
$$

- BASE 2 (BINÁRIO)

Ex:

$$
\begin{array}{rcccccc}
 & (1) & (1) & (1) & (1) & (1) & \\
 & & 1 & 0 & 1 & 1 & 1_2 \\
+ & & 1 & 1 & 0 & 1 & 1_2 \\
\hline
 & 1 & (11) & (10) & (10) & (11) & (10) \\
- & & 10 & 10 & 10 & 10 & 10 \\
\hline
 & (1 & 1 & 0 & 0 & 1 & 0)_2
\end{array}
$$

$$
\begin{array}{rcccccc}
 & & & (1) & & & \\
 & & 1 & 0 & 1 & 0 & 0_2 \\
+ & & 1 & 1 & 1 & 0 & 1_2 \\
\hline
 & (1 & 1 & 0 & 0 & 0 & 1)_2
\end{array}
$$

$$
\begin{array}{rccccc}
 & 1 & 1 & 0 & 0 & 1_2 \\
 & 1 & 0 & 1 & 1 & 1_2 \\
+ & 1 & 0 & 0 & 1 & 0_2 \\
\hline
\end{array}
$$

- SUBTRAÇÃO EM DIVERSAS BASES

- BASE 10

Ex:

$$\begin{array}{rr} \overset{2}{\cancel{3}} & \overset{(10)}{+1} \\ -\ 2 & 4 \\ \hline 0 & 7 \end{array} \qquad (1)$$

$$\begin{array}{rr} 5 & 2 \\ -\ 3 & \overset{+1}{7} \\ \hline 1 & 5 \end{array}$$

- BASE 8

Ex:

$$\begin{array}{rr} 7 & 5_8 \\ -\ 3 & 4_8 \\ \hline (4 & 1)_8 \end{array}$$

$$\begin{array}{rr} \overset{4}{\cancel{5}} & \overset{(10)_8 =}{+3_8} \\ -\ 3 & 6_8 \\ \hline (1 & 5)_8 \end{array} \qquad (1)$$

8+3=1

$$\begin{array}{rrr} 6 & 0 & 1_8 \\ -\ 4 & 2 & 5_8 \\ \hline (1 & 5 & 4)_8 \end{array}$$

$$\begin{array}{rrr} 5 & 0 & 2_8 \\ -\ 3 & 6 & 7_8 \\ \hline \end{array}$$

- BASE 16

Ex:

$$\begin{array}{cccl} \cancel{6}^{5} & 3 & A_{16} & \quad (16+3=19) \\ -\ 2 & D & 5_{16} & \\ \hline (3 & 6 & 5)_{16} & \end{array}$$

$$\begin{array}{ccc} 2 & 0 & 0_{16} \\ -\ 1^{+1} & E^{+1} & A_{16} \\ \hline (0 & 1 & 6)_{16} \end{array}$$

$$\begin{array}{ccc} 3 & D^{13} & C^{12}_{16} \\ -\ 1^{+1} & F^{+1}\ (16) & E_{16}\ (14) \\ \hline 1 & (13) & (14) \\ \downarrow & \downarrow & \downarrow \\ (1 & D & E)_{16} \end{array}$$

$$\begin{array}{ccc} B & 0 & A_{16} \\ -\ 2 & C & C_{16} \\ \hline \end{array}$$

- BASE 2

Ex:

$$\begin{array}{cccccc} 1 & 1 & 0 & 0 & 1 & 0_2 \\ -\ 1 & 0^{+1} & 1^{+1} & 1 & 0^{+1} & 1_2 \\ \hline (0 & 0 & 0 & 1 & 0 & 1)_2 \end{array}$$

$$\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0_2 \\ -\ 1 & 0 & 1 & 1 & 1 & 1 & 0_2 \\ \hline \end{array}$$

- REPRESENTAÇÃO EM "COMPLEMENTO À BASE"

- O COMPLEMENTO DE UM NÚMERO N EM UM DADO BASE B É IGUAL À DIFERENÇA ENTRE O NÚMERO E A PRÓXIMA POTÊNCIA DA BASE.

Ex: BASE 10 → COMPLEMENTO A 10.

NÚMERO 734 → PRÓXIMA POTÊNCIA → $10^3 = 1000$

$$\begin{array}{r} 1000 \\ -\ 734 \\ \hline 266 \end{array}$$

266 É O COMPLEMENTO A 10 DE 734

Ex: BASE 2 → COMPLEMENTO A 2

NÚMERO 1011 → PRÓXIMA POTÊNCIA → $2^4 = 16 = 10000$

$$\begin{array}{r} 10000 \\ -\ 1011 \\ \hline 101 \end{array}$$

101 É O COMPLEMENTO A 2 DE 1011

Ex: Base 2: complemento a 2 de 101101

- Cálculo do complemento à base em qualquer base é tedioso por causa dos "vem-um".
- Uma alternativa mais confortável é calcular o "complemento à base menos 1" e depois somar 1 ao resultado para obter o complemento à base.

Ex: Base 10 → "complemento a 9" de 734

$$\begin{array}{r} 999 \\ -\ 734 \\ \hline 265 \end{array}$$ (complemento a 9 de 734)

$$\begin{array}{r} 265 \\ +\ 1 \\ \hline 266 \end{array}$$ (" a 10 de 734)

Ex: Base 2 → "complemento a 1" de 1011

$$\begin{array}{r} 1111 \\ -\ 1011 \\ \hline 100 \end{array}$$ (complemento a 1 de 1011)

Ex: Base 2 → complemento a 1 de 1101011

- Observação importante:

  Para obter o complemento a 1 de um número basta inverter o número bit a bit.

- Operações de subtração em qualquer base podem ser feitas usando complemento à base

Ex: Na base 10 → 913 − 734

$$\begin{array}{r} 913 \\ -\,734 \\ \hline 179 \end{array} \qquad \begin{array}{r} 999 \\ -\,734 \\ \hline 265 \\ +\,1 \\ \hline 266 \end{array} \qquad \begin{array}{r} 913 \\ +\,266 \\ \hline 1)\,179 \end{array}$$

Ex: Na base 2 → 11001 − 10011

Complemento a 2 do subtraendo 10011

$$\begin{array}{rl} 01100 & \text{(comp. a 1)} \\ +\quad 1 & \\ \hline 01101 & \text{(comp. a 2)} \end{array} \qquad \begin{array}{rl} 11001 & \\ +\,01101 & \\ \hline 1)\,00110 & \text{(resultado)} \end{array}$$

Ex: Base 2 → 110110 - 101101

- Regra prática para obtenção do complemento a 2

Para se obter diretamente o complemento a 2 de um número basta percorrer o número da direita para a esquerda repetindo-se os dígito zeros até encontrar o primeiro dígito 1 (um), o qual deve ser mantido. Daí para a frente todos os dígitos devem ser invertidos.

Ex: 1010 ⟹ 0110

1001100 ⟹ 0110100

1100110 ⟹

- A vantagem de se representar números negativos em notação complemento a 2 internamente ao computador é que todas as operações de subtração ficam transformadas em simples somas.

- REPRESENTAÇÃO DE NÚMEROS COM SINAIS

- SUPONDO QUE OS NÚMEROS TENHAM UMA LARGURA INTERNA DE 8 BITS (1 BYTE), OU SEJA ELES SÃO OPERADOS E ARMAZENADOS EM 8 BITS.

7							Ø
MSB							LSB

BIT DE SINAL $\begin{cases} \text{Ø – Nº POSITIVO} \\ \text{1 – Nº NEGATIVO (EM COMP. A:} \end{cases}$

- EM 8 BITS É POSSÍVEL REPRESENTAR 256 DIFERENTES NÚMEROS: DE $\underbrace{00000000}_{0}$ A $\underbrace{1111111}_{255}$ JÁ QUE $2^8 = 256$

- COM O BIT MAIS SIGNIFICATIVO REPRESENTANDO O SINAL, A GAMA DE NÚMEROS POSSÍVEIS DE SEREM REPRESENTADOS PERMANECE A MESMA SÓ QUE AGORA METADE NEGATIVA E METADE POSITIVA.

+127 ⟶ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

−128 ⟶ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- Operações aritméticas com números com sinal

- Números positivos são submetidos na forma "rea

Ex:

7							0	
0	0	0	1	0	1	1	0	⟹ +22

↑ sinal positivo

- Números negativos na forma complemento a 2.

Ex:

7							0	
1	1	1	0	1	0	1	0	⟹ −22

↑ sinal negativo

- A partir daí todas as operações de soma e subtração envolvendo números com bits de sinal produzem diretamente resultados consistentes, ou seja, positivo em forma "real" e negativo em complemento a 2.

- REGRAS PARA ADIÇÃO E SUBTRAÇÃO DE NÚMEROS BINÁRIOS COM SINAL:

  - A SOMA DE DOIS NÚMEROS COM MESMO SINAL PRODUZ UM RESULTADO COM O SINAL COMUM.

  - O RESULTADO DA SOMA DE DOIS NÚMEROS DE SINAIS DIFERENTES É A DIFERENÇA ENTRE SEUS MÓDULOS E RECEBE O SINAL DO MAIOR.

  - PARA SUBTRAIR, MUDAR O SINAL DO SUBTRAENDO E SOMÁ-LO AO MINUENDO VERIFICADAS AS REGRAS ACIMA.

• EXEMPLOS: NÚMEROS 4 E 9.

+4	00000100
(+) +9	00001001
+13	00001101 (POSITIVO)

−4	11111100
(+) −9	11110111
−13	1) 11110011 (NEGATIVO)

−4	11111100
(+) +9	00001001
+5	1) 00000101 (POSITIVO)

+4	00000100
(+) −9	11110111
−5	11111011 (NEGATIVO)

	+9	00001001	00001001
(-)	+4	00000100 →	11111100
	+5		1)00000101

	+4	00000100	00000100
(-)	+9	00001001 →	11110111
	-5		11111011

	+4	00000100	00000100
(-)	-9	11110111 →	00001001
	+13		00001101

	-4	11111100	11111100
(-)	+9	00001001 →	11110111
	-13		1)11110011

- OVERFLOW EM OPERAÇÕES ARITMÉTICAS

  - O MAIOR NÚMERO POSITIVO QUE PODE SER ARMAZENADO EM UM REGISTRO DE 8 BITS É +127 (01111111).
  - NAS MESMAS CONDIÇÕES O MENOR NEGATIVO É −128 (10000000).

  - SE O RESULTADO DE ALGUMA OPERAÇÃO ARITMÉTIC EXCEDER UM DOS NÚMEROS ACIMA É DITO QUE UMA CONDIÇÃO DE OVERFLOW OCORREU. ESTA, SE NÃO DETECTADA, ACARRETA ERRO.

  - A DETECÇÃO DE OVERFLOW É SIMPLES E CONSISTE EM
    1. HÁ UM "VAI-UM" PROPAGADO PARA O BIT DE SINA SEM "VAI-UM" SAINDO DO BIT DE SINAL.
    2. HÁ UM "VAI-UM" PROPAGADO PELO BIT DE SINAL SEM ESTE TER RECEBIDO "VAI-UM".

- EXEMPLOS DE OPERAÇÕES ARITMÉTICAS EM 8 BITS

1. -73
   $(+)\ \underline{+18}$

2. $+27$
   $(+)\ \underline{-43}$

3. -83
   $(+)\ \underline{-49}$

$$\begin{array}{r} 1\,1\,1\,1 \\ 10101101 \\ +\ 11001111 \\ \hline 101111100 \end{array}$$

Erro de Overflow!

↳ Ultrapassa 8 bits

4. -37
   $(-)\ \underline{+22}$

ÁLGEBRA DE BOOLE

- CRIADA EM 1854 POR GEORGE BOOLE COM O INTUITO DE FORMALIZAR MATEMATICAMENTE O PENSAMENTO LÓGIC
É A ÁLGEBRA DO SISTEMA BINÁRIO.

- VARIÁVEL LÓGICA OU BOOLEANA
UMA VARIÁVEL LÓGICA SÓ PODE ASSUMIR DOIS ESTADOS: VERDADEIRO OU FALSO; LIGADO OU DESLIGADO; ACESO OU APAGADO; SIM OU NÃO; FECHADO OU ABERTO; "1" OU "0".

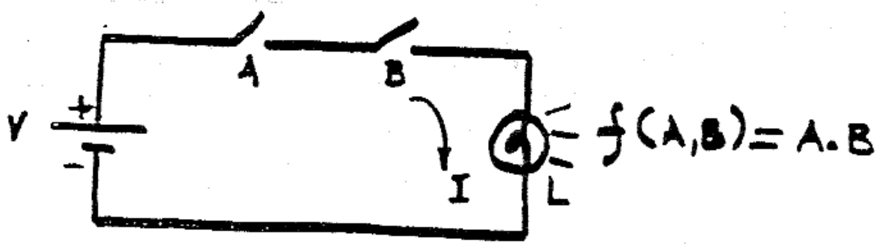
- OPERAÇÕES LÓGICAS
EXISTEM SOMENTE 3 OPERAÇÕES LÓGICAS BÁSICAS:
1. PRODUTO LÓGICO → FUNÇÃO AND (•)
2. SOMA LÓGICA → FUNÇÃO OR (+)
3. NEGAÇÃO → FUNÇÃO NOT (‾) OU (')

- TODAS AS OPERAÇÕES INTERNAS A UM COMPUTADOR PODEM SER DESCRITAS COMO COMBINAÇÕES DESTAS 3 OPERAÇÕES BÁSICAS.

• Função AND

$$f(A,B,C,\ldots,N) = A \cdot B \cdot C \cdot \ldots \cdot N$$

$f(A,B,\ldots N) = 1$ se e somente se $A = B = C = \ldots = N = 1$



Símbolo lógico:

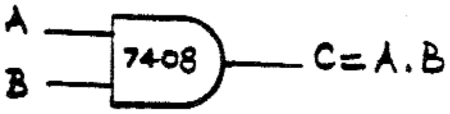


Tabela verdade:

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

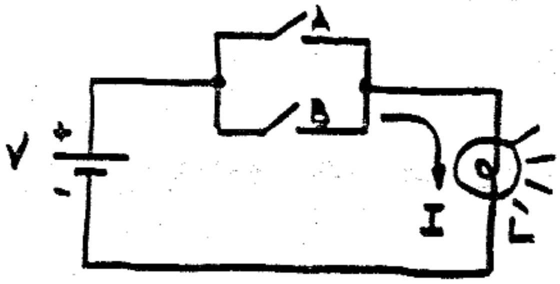
(3 variáveis)

A	B	C	D
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

Função OR

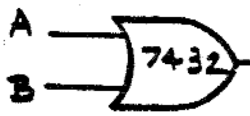
$$f(A,B,\ldots,N) = A+B+\ldots+N$$

$$f(A,B,\ldots,N) = 1 \text{ se } A=1 \text{ ou } B=1 \text{ ou } \ldots \text{ ou } N=1$$



$$f(A,B) = A+B$$

Símbolo lógico:



$C = A+B$

Tabela verdade:

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

(3 variáveis)

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

- FUNÇÃO NOT (INVERSOR)

$$f(A) = \overline{A} \implies \begin{cases} f(A) = 0 \text{ se } A = 1 \\ f(A) = 1 \text{ se } A = 0 \end{cases}$$

SÍMBOLO LÓGICO:

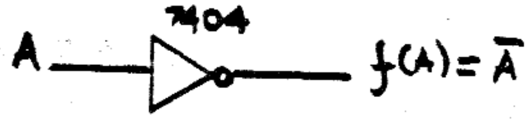


TABELA VERDADE:

A	f(A)
0	1
1	0

- OUTRAS FUNÇÕES LÓGICAS:

- FUNÇÃO NAND

$$f(A, B, \ldots, N) = \overline{A \cdot B \cdot C \cdots N}$$

$f(A, B, \ldots, N) = 0$ SE E SOMENTE $A = B = \cdots = N = 1$

SÍMBOLO LÓGICO:

TABELA VERDADE:

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

- FUNÇÃO NOR

$$f(A,B,\dots,N) = \overline{A+B+\dots+N}$$

$f(A,B,\dots,N) = 1$ SE E SOMENTE SE $A=B=\dots=N=0$

SÍMBOLO LÓGICO:

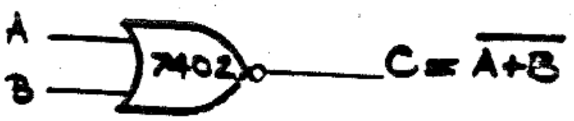


TABELA VERDADE:

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

• FUNÇÃO OU-EXCLUSIVO

$$f(A,B) = A \oplus B \quad \begin{cases} f(A,B) = 1 \text{ SE } A \neq B \\ f(A,B) = 0 \text{ SE } A = B \end{cases}$$

SÍMBOLO LÓGICO :

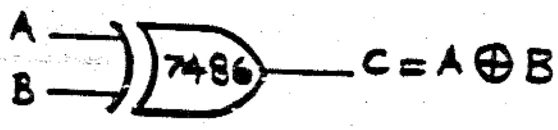


TABELA VERDADE:

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

POR INSPEÇÃO NA TABELA VERDADE:

$$C = 1 \text{ se } \underbrace{A=0 \text{ E } B=1}_{\bar{A}.B} \ \underbrace{\text{OU}}_{+} \text{ SE } \underbrace{A=1 \text{ E } B=0}_{A.\bar{B}}$$

$$C = \bar{A}B + A\bar{B}$$

RELAÇÕES DA ÁLGEBRA BOOLEANA:

POSTULADOS:

1a. $A=1$ (SE $A\neq 0$) 1b. $A=0$ (SE $A\neq 1$)

2a. $0.0=0$ 2b. $0+0=0$

3a. $1.1=1$ 3b. $1+1=1$

4a. $1.0=0$ 4b. $1+0=1$

5a. $\overline{1}=0$ 5b. $\overline{0}=1$

TEOREMAS:

6a. $A.0=0$ 6b. $A+0=A$

7a. $A.1=A$ 7b. $A+1=1$

8a. $A.A=A$ 8b. $A+A=A$

9a. $A.\overline{A}=0$ 9b. $A+\overline{A}=1$

10a. $\overline{\overline{A}}=A$ 10b. $A=\overline{\overline{A}}$

- PROPRIEDADES ALGÉBRICAS:

COMUTATIVA:

11a. $AB = BA$ 11b. $A + B = B + A$

ASSOCIATIVA:

12a. $A(BC) = AB(C)$ 12b. $A + (B + C) = (A + B) + C$

DISTRIBUTIVA:

13a. $A(B + C) = AB + AC$ 13b. $A + BC = (A + B)(A + C)$

- TEOREMA DA ABSORÇÃO:

14a. $A(A + B) = A$ 14b. $A + AB = A$

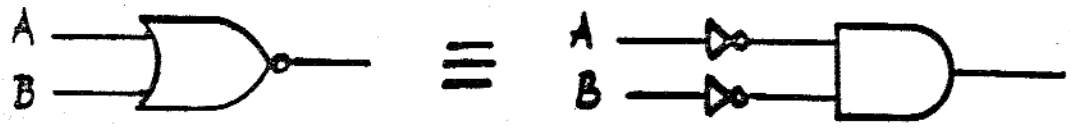
15a. $A(\overline{A} + B) = AB$ 15b. $A + \overline{A}B = A + B$

- TEOREMAS DE MORGAN:

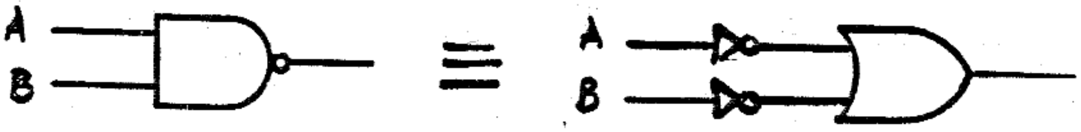
16a. $\overline{A.B.C} = \overline{A} + \overline{B} + \overline{C}$ 16b. $\overline{A+B+C} = \overline{A}.\overline{B}.\overline{C}$

• Consequências imediatas da Lei de Morgan:

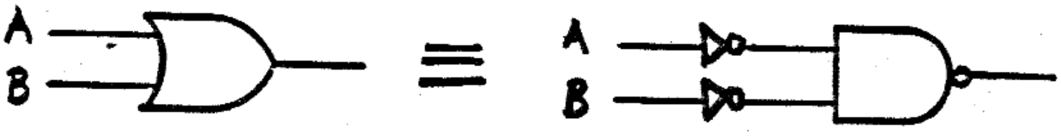
1. $\overline{A+B} = \overline{A} \cdot \overline{B}$



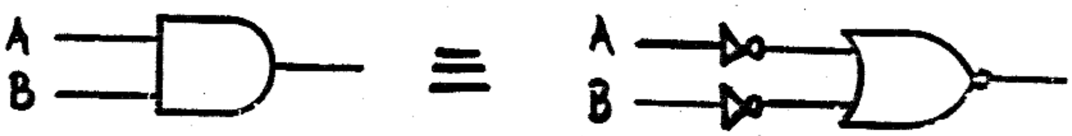
2. $\overline{A \cdot B} = \overline{A} + \overline{B}$



3. $\overline{A+B} = \overline{A} \cdot \overline{B} \Rightarrow \overline{\overline{A+B}} = \overline{\overline{A} \cdot \overline{B}} \Rightarrow A+B = \overline{\overline{A} \cdot \overline{B}}$



4. $\overline{A \cdot B} = \overline{A} + \overline{B} \Rightarrow \overline{\overline{A \cdot B}} = \overline{\overline{A} + \overline{B}} \Rightarrow A \cdot B = \overline{\overline{A} + \overline{B}}$



- Exercícios:

1. Mostrar que $A + BC = (A+B)(A+C)$

$$(A+B).(A+C) = AA + AC + AB + BC = A + AC + AB + BC$$
$$= A(1 + C + B) + BC = A + BC$$

2. Mostrar que $A + \bar{A}B = A + B$

$$A + \bar{A}B = \underbrace{A + AB}_{A} + \bar{A}B = A + (A + \bar{A})B = A + B$$

3. Mostrar que $\overline{A \oplus B} = \bar{A} \oplus B$

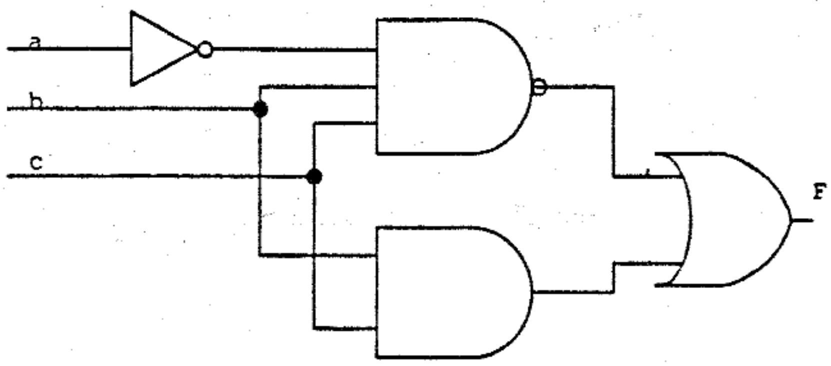
$$\overline{A \oplus B} = \overline{A\bar{B} + \bar{A}B} \overset{M}{=} (\bar{A} + B).(A + \bar{B}) = \cancelto{0}{\bar{A}A} + \bar{A}\bar{B} + AB + \cancel{B\bar{B}}$$
$$= \bar{A}\bar{B} + AB = \bar{A} \oplus B = A \oplus \bar{B}$$

4. Mostrar que $AB + \bar{A}C + BC = AB + \bar{A}C$

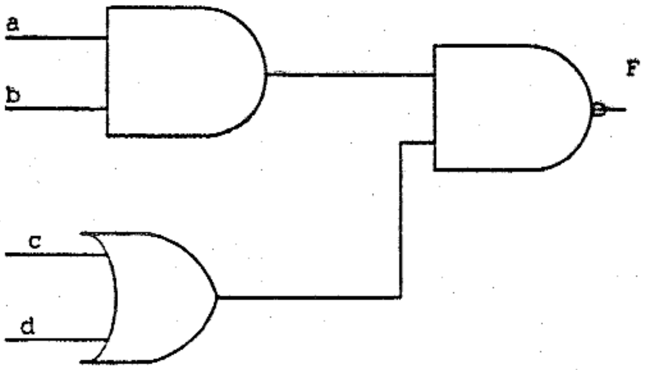
$$AB + \bar{A}C + BC = AB + \bar{A}C + BC\overbrace{(A + \bar{A})} = AB + \bar{A}C + BCA + BC\bar{A}$$
$$= A(\underbrace{B + BC}_{B}) + \bar{A}(\underbrace{C + BC}_{C}) = AB + \bar{A}C$$

$1^{\underline{o}}$: Desenhe o circuito correspondente a equação abaixo:

$$F = \overline{\overline{a}bc} + bc$$



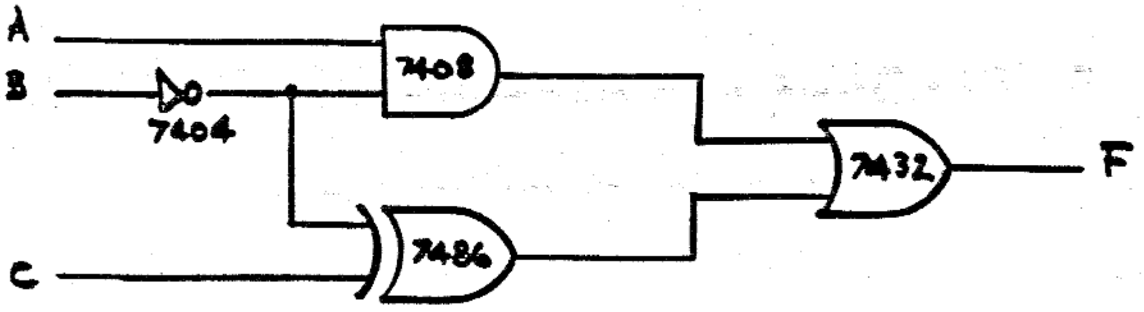
$2^{\underline{o}}$: Escreva a equação correspondente ao desenho abaixo:



$$F = \overline{(ab).(c+d)}$$

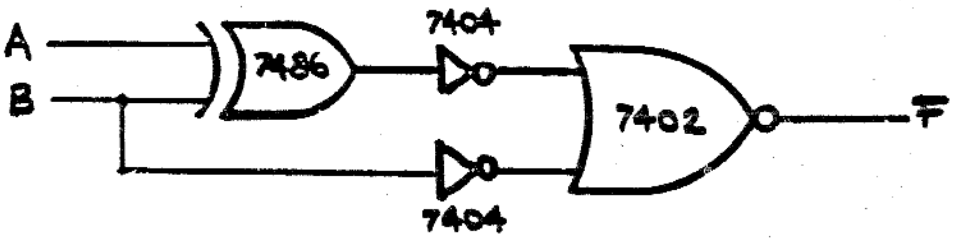
• EXERCÍCIOS :

1. DETERMINE A EXPRESSÃO LÓGICA IMPLEMENTADA PELO CIRCUITO ABAIXO:



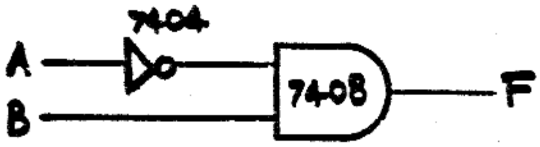
$$F = A\bar{B} + (\bar{B} \oplus C) = A\bar{B} + \bar{B}\bar{C} + BC$$

2. O MESMO PARA O CIRCUITO ABAIXO:



$$F = \overline{\overline{(A \oplus B)} + \bar{B}} = \overline{\bar{A}\bar{B} + AB + \bar{B}} \overset{M}{=} (A+B).(\bar{A}+\bar{B}).B =$$

$$= (A\bar{A} + A\bar{B} + \bar{A}B + B\bar{B}).B = \bar{A}B$$



SIMPLIFIQUE AS FUNÇÕES BOOLEANAS ABAIXO DE MODO QUE ELAS TERMINEM COM O MENOR NÚMERO POSSÍVEL DE LITERAIS:

a) $x + \bar{x}y$

b) $\bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$

c) $(x+y)(x+\bar{y})$

d) $zx + z\bar{x}y$

a) $x + \bar{x}y = (x+\bar{x})(x+y) =$

$= 1 \cdot (x+y) = x+y$

b) $\bar{x}yz + x\bar{y}z + xy\bar{z} + xyz =$

$= \bar{x}yz + xyz + x\bar{y}z + xyz + xy\bar{z} + xyz =$

$= yz(x+\bar{x}) + xz(\bar{y}+y) + xy(z+\bar{z}) =$

$= yz + xz + xy$

c) $(x+y)(x+\bar{y}) = x + y\bar{y} = x + 0 = x$

d) $zx + z\bar{x}y = z(x+\bar{x}y) = z(x+\bar{x})(x+y) =$

$= z(x+y)$

AB12.2

- MINTERMOS E MAXTERMOS:

CONSIDERE A TABELA VERDADE ABAIXO QUE EXPRESSA A FUNÇÃO VOTADOR MAJORITÁRIO:

	A	B	C	F	$\overline{F}$
0 →	0	0	0	0	1
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0

POR INSPEÇÃO NA TABELA VERDADE PODE-SE ESCREVER:

$$F(A,B,C) = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

F É CONSTITUÍDA DE UMA SOMA DE PRODUTOS CADA UM ENVOLVENDO AS 3 VARIÁVEIS DE QUE F DEPENDE.

CADA UM DESTES PRODUTOS É DENOMINADO MINTERMO.

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

$$F = m_3 + m_5 + m_6 + m_7 = \Sigma\ 3,5,6,7$$

(NOTAÇÃO COMPACTA)

A DESCRIÇÃO DE UMA FUNÇÃO POR MEIO DE SOMA DE PRODUTOS (MINTERMOS) REPRESENTA A FUNÇÃO IMPLEMENTADA NOS PONTOS EM QUE ELA É "1".

APLICANDO A LEI DE MORGAN A EQUAÇÃO ACIMA TEMOS:

$$\bar{F} = \overline{\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC} = \overline{(\bar{A}BC)}.\overline{(A\bar{B}C)}.\overline{(AB\bar{C})}.\overline{(ABC)}$$

$$\bar{F} = (A+\bar{B}+\bar{C}).(\bar{A}+B+\bar{C}).(\bar{A}+\bar{B}+C).(\bar{A}+\bar{B}+\bar{C})$$

$\bar{F}$ ACHA-SE REPRESENTADO POR UM PRODUTO DE SOMAS ONDE CADA SOMA CONTÉM AS 3 VARIÁVEIS A QUE $\bar{F}$ SE REFERE. CADA UMA DESTAS SOMAS É DENOMINADA MAXTERMO.

A DESCRIÇÃO DE UMA FUNÇÃO BOOLEANA POR MEIO DA NOTAÇÃO PRODUTO DE SOMAS (MAXTERMOS) CORRESPO: À IMPLEMENTAÇÃO DA FUNÇÃO NOS PONTOS ONDE ELA É "(

$$\overline{F} = (A+\overline{B}+\overline{C}).(\overline{A}+B+\overline{C}).(\overline{A}+\overline{B}+C).(\overline{A}+\overline{B}+\overline{C})$$

$$\overline{F} = M_3 \quad . \quad M_5 \quad . \quad M_6 \quad . \quad M_7$$

$$\overline{F} = \Pi\ 3,5,6,7$$

PARA ACHAR F REPRESENTADA POR MAXTERMOS APLICAMC MORGAN À EXPRESSÃO DE $\overline{F}$ DESCRITA POR MINTERMOS:

$$\overline{F} = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} = m_0 + m_1 + m_2 + m_4$$

$$\overline{\overline{F}} = \overline{\overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C}} = (\overline{\overline{A}\overline{B}\overline{C}}).(\overline{\overline{A}\overline{B}C}).(\overline{\overline{A}B\overline{C}}).(\overline{A\overline{B}}$$

$$F = (A+B+C).(A+B+\overline{C}).(A+\overline{B}+C).(\overline{A}+B+C)$$

$$F = M_0 \quad . \quad M_1 \quad . \quad M_2 \quad . \quad M_4$$

$$F = \Pi\ 0,1,2,4$$

Mintermos e Maxtermos para uma função de 3 variáveis

Linha	x	y	z	Mintermo	Maxtermo	
0	0	0	0	m_0 $\bar{x}\bar{y}\bar{z}$	$x + y + z$	M_0
1	0	0	1	m_1 $\bar{x}\bar{y}z$	$x + y + \bar{z}$	M_1
2	0	1	0	m_2 $\bar{x}y\bar{z}$	$x + \bar{y} + z$	M_2
3	0	1	1	m_3 $\bar{x}yz$	$x + \bar{y} + \bar{z}$	M_3
4	1	0	0	m_4 $x\bar{y}\bar{z}$	$\bar{x} + y + z$	M_4
5	1	0	1	m_5 $x\bar{y}z$	$\bar{x} + y + \bar{z}$	M_5
6	1	1	0	m_6 $xy\bar{z}$	$\bar{x} + \bar{y} + z$	M_6
7	1	1	1	m_7 xyz	$\bar{x} + \bar{y} + \bar{z}$	M_7

A.E. I.S. 1

- Simplificação de expressões booleanas:

Dois métodos $\begin{cases} \text{a. Manipulação algébrica} \\ \text{b. Mapas de Karnaugh} \end{cases}$

a. Manipulação algébrica – Exercícios:

a.1. Simplificar a função votador majoritário

$F(A,B,C) = \Sigma\, 3,5,6,7$

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

$F = \bar{A}B\underbrace{C + A\bar{B}C + A}B(\cancel{\bar{C} + C}^{1})$

$F = B(A + \bar{A}C) + A\bar{B}C$

$F = B(A + C) + A\bar{B}C$

$F = AB + BC + A\bar{B}C$

$F = AB + C(B + \bar{B}A)$

$F = AB + C(B + A) \Longrightarrow F = AB + AC + BC$

a.2. Simplificar $F = A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$

$$F = A\bar{B}\,\cancelto{1}{(\bar{C}+C)} + AB\,\cancelto{1}{(\bar{C}+C)} = A\bar{B} + AB = A\,\cancelto{1}{(\bar{B}+B)} = A$$

a.3. Simplificar $F(A,B,C) = \sum 3,7$

$$F = \bar{A}BC + ABC = BC(\bar{A}+A) = BC$$

a.4. Simplificar $F(A,B,C) = \sum 0,1,7$

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$$

$$F = \bar{A}\bar{B}(\bar{C}+C) + ABC = \bar{A}\bar{B} + ABC$$

a.5. Simplificar $F(A,B,C) = \sum 0,2,4,6$

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C}$$

$$F = \bar{A}\bar{C}(\bar{B}+B) + A\bar{C}(\bar{B}+B)$$

$$F = \bar{A}\bar{C} + A\bar{C} = \bar{C}(\bar{A}+A) \Longrightarrow F = \bar{C}$$

EXEMPLOS

a) REPRESENTE A FUNÇÃO $F(A,B,C) = \sum 1, 3, 5, 7$ POR MEIO DE PRODUTOS DE MAXTERMOS:

$$F = \prod 0, 2, 4, 6 = M_0 \cdot M_2 \cdot M_4 \cdot M_6$$

$$F = (A+B+C)(A+\bar{B}+C)(\bar{A}+B+C)(\bar{A}+\bar{B}+C)$$

b) REPRESENTE A FUNÇÃO F POR MEIO DE SOMAS DE MINTERMOS E PRODUTOS DE MAXTERMOS:

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$$F = \sum 1, 2, 4 = m_1 + m_2 + m_4$$

$$F = \prod 0, 3, 5, 6, 7 =$$
$$= M_0 \cdot M_3 \cdot M_5 \cdot M_6 \cdot M_7$$

$$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

$$F = (A+B+C)(A+\bar{B}+\bar{C})(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$$

ABIJ.3

6. MAPAS DE KARNAUGH

CONSIDERE A FUNÇÃO ABAIXO DESCRITA POR UA TABELA DA VERDADE:

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

DA TABELA VERDADE ACIMA PODEMOS TIRAR QUE:

$$X = \sum 2, 3, 4, 5$$

REPARE QUE OS MINTERMOS 2 E 3, 4 E 5 SÃO TAIS QUE PERMITEM OBTER UM TERMO COM UMA VARIÁVEL A MENOS ATRAVÉS DE SIMPLIFICAÇÕES DO TIPO $(A+\bar{A})$. TAIS MINTERMOS SÃO DENOMINADOS ADJACENT

ABIS

- UM MAPA DE KARNAUGH É UMA REPRESENTAÇÃ GRÁFICA DA TABELA VERDADE ONDE CADA QUADRADO REPRESENTA UM MINTERMO DE TAL MANEIRA QUE QUADRADOS ADJACENTES CONTÉM MINTERMOS ADJACENTES.

- PARA A FUNÇÃO ANTERIOR DE 3 VARIÁVEIS O MAPA DE KARNAUGH FICA:

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

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

B (columns 11, 10); A (row 1); C (columns 01, 11)

SIMPLIFICANDO PELO MAPA TEMOS:

$$X = \bar{A}B + A\bar{B} = A \oplus B$$

b. Mapas de Karnaugh:

Considere a função votador majoritário de 4 votantes descrito pela tabela verdade abaixo:

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	X
0	1	0	0	0
0	1	0	1	X
0	1	1	0	X
0	1	1	1	1
1	0	0	0	0
1	0	0	1	X
1	0	1	0	X
1	0	1	1	1
1	1	0	0	X
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

ex 3:

- Os valores X representam pontos onde a função F não é definida. São denominados estados opcionais (DON'T CARE). Os estados DON'T CARE podem ser utilizados em simplificações de funções como será visto mais adiante.

Da tabela verdade anterior podemos tirar que:

$F = \sum 7, 11, 13, 14, 15$

$F = \bar{A}BCD + A\bar{B}CD + AB\bar{C}D + ABC\bar{D} + ABCD$

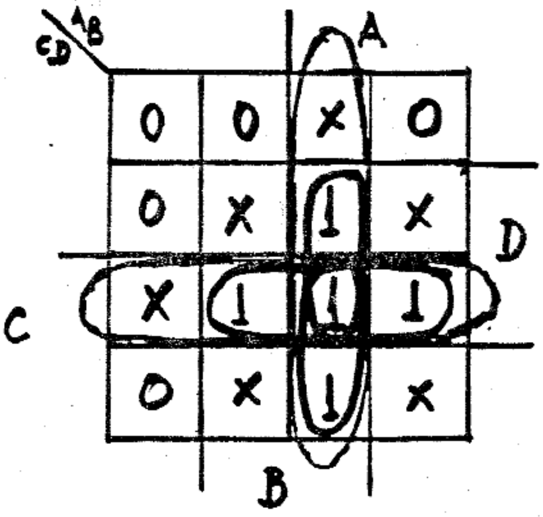
- Repare que os mintermos 7 e 15, 11 e 15, 13 e 15, 14 e 15 são tais que permitem obter um termo com uma variável a menos por meio de simplificações do tipo $(x+\bar{x})$. Tais mintermos são denominados adjacentes. Desta forma a combinação de dois estados adjacentes resulta sempre na eliminação da variável que os difer

- Um mapa de Karnaugh é uma representação gráfica da tabela verdade onde cada quadrado representa um mintermo de tal maneira que quadrados adjacentes contém mintermos adjacent

Para a função votador majoritário de 4 votanti o mapa de Karnaugh fica:

CD \ AB	00	01	11	10
00	0	0	X	0
01	0	X	1	X
11	X	1	1	1
10	0	X	1	X

CD \ AB	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}



Simplificando pelo mapa tem-se que:

$F = ABD + ACD + ABC + BCD$

$F = AB + CD$

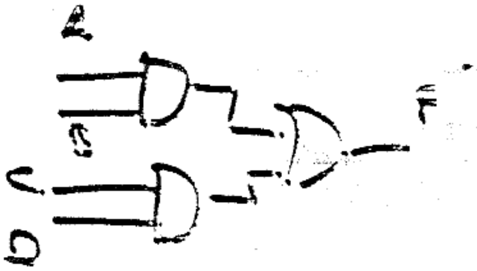
$F = \bar{A}BCD + A\bar{B}CD + AB\bar{C}D + ABC\bar{D} + ABCD$

PARA A FUNÇÃO CONSIDERADA O MAPA DE KARNAUGH FICA:

AB \ CD	00	01	11	10
00	0	0	X	0
01	0	X	1	X
11	X	1	1	1
10	0	X	1	X

AB \ CD	00	01	11	10
00			X	
01		X	1	X
11	X	1	1	1
10		X	1	X

C B A D



$F = AB + CD$

MAPAS DE 4 VARIÁVEIS

SIMPLIFIQUE A FUNÇÃO BOOLEANA ABAIXO!

$$F(A,B,C,D) = \sum 0,1,2,3,4,5,6,7,9,11,13,15$$

F

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

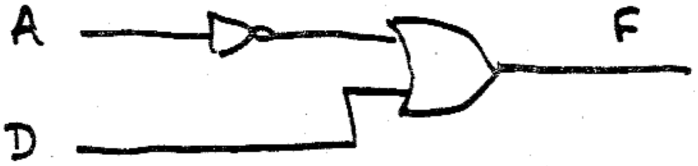
C, B, A, D

F

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11		1	1	
10		1	1	

C, B, A, D

$$F = \bar{A} + D$$



- Regras para simplificação com o Mapa de Karnaugh:

1. Começar pelos quadrados isolados, isto é que não tenham adjacentes. Estes representam mintermos que não podem ser simplificados mas que fazem parte da funç

2. Combinar grupos de 2 quadrados adjacentes que correspondem à simplificação de uma variável.

3. Combinar grupos de 4 quadrados ⟹ 2 variáveis a menos

4. Combinar grupos de 8 quadrados ⟹ 3 variáveis a menos

- Observações:

a) Estados opcionais (X-DON'T CARES) podem ser incluídos na combinações caso conduzam a uma melhor simplificaçõe

b) Combinações onde <u>todos</u> quadrados participantes já tenham sido utilizados em combinações prévias não geram simplificações adicionais.

MAPAS DE KARNAUGH – EXERCÍCIOS:

6.1. $F(A,B,C) = \sum 3,4,6,7$

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

C \ AB	00	01	11	10
0	0	0	1	1
1	0	1	1	0

$$F = A\bar{C} + BC$$

6.2. $F(A,B,C) = \sum 0, X_2, X_3, 4, 5, 6$

$$F = \bar{A}\bar{B}\bar{C} + \underset{\uparrow}{\overset{x}{\bar{A}B\bar{C}}} + \underset{\uparrow}{\overset{x}{\bar{A}BC}} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}$$

C \ AB				
	1	X	1	1
C	0	X	0	1

$$F = A\bar{B} + \bar{C}$$

• Mapas de Karnaugh - Exercícios:

6.1 $F(A,B,C) = \Sigma\, 3, 4, 6, 7$

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

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

$F = A\bar{C} + BC$

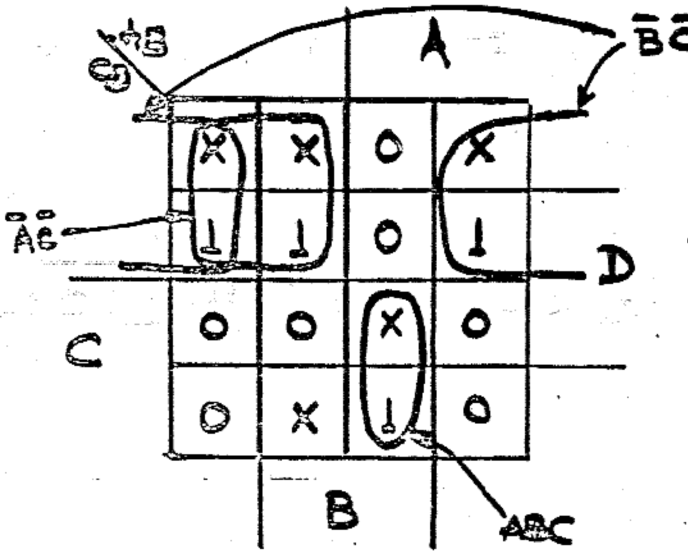
6.2 $F(A,B,C) = \Sigma\, 0, X_2, X_3, 4, 5, 6$

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

$F = \bar{C} + A\bar{B}$

6.3. $F(A,B,C,D) = \sum x_0, 1, x_4, 5, x_6, x_8, 9, 14, x_{15}$

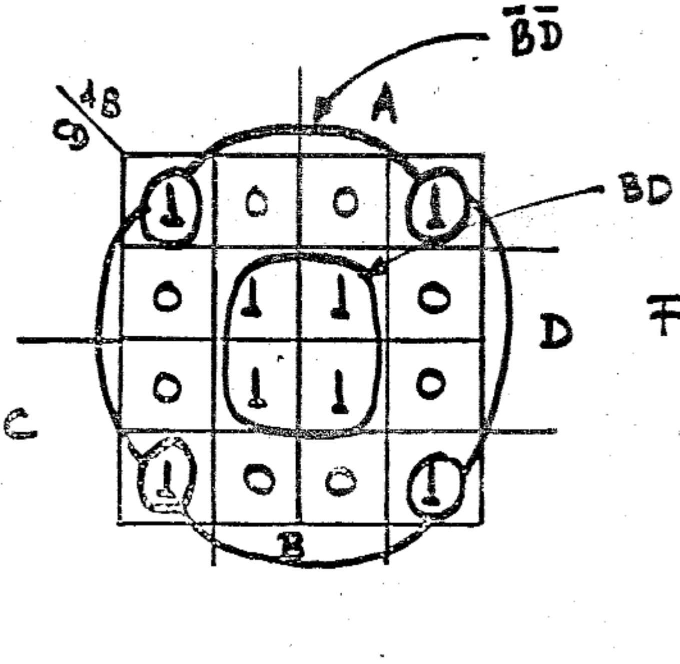
$$F = \underset{x_0}{\bar{A}\bar{B}\bar{C}\bar{D}} + \bar{A}\bar{B}\bar{C}D + \underset{x_4}{\bar{A}B\bar{C}\bar{D}} + \bar{A}B\bar{C}D + \underset{x_6}{\bar{A}BC\bar{D}} + \underset{x_8}{A\bar{B}\bar{C}\bar{D}} + A\bar{B}\bar{C}D + ABC\bar{D} + \ldots$$



$F = \bar{A}\bar{C} + \bar{B}\bar{C} + ABC$

6.4. $F(A,B,C,D) = \sum 0, 2, 5, 7, 8, 10, 13, 15$

$$F = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}D + \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + AB\bar{C}D + ABC\ldots$$

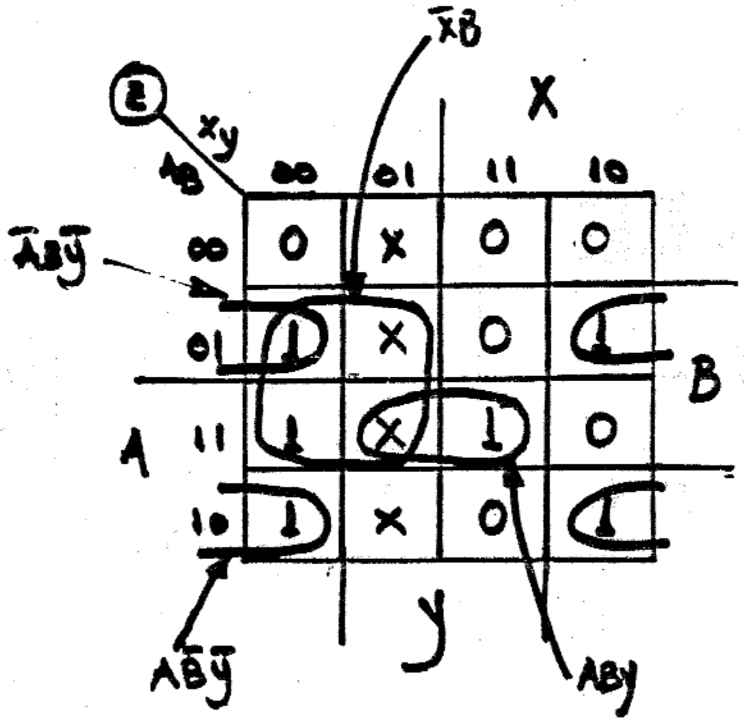


$F = BD + \bar{B}\bar{D}$

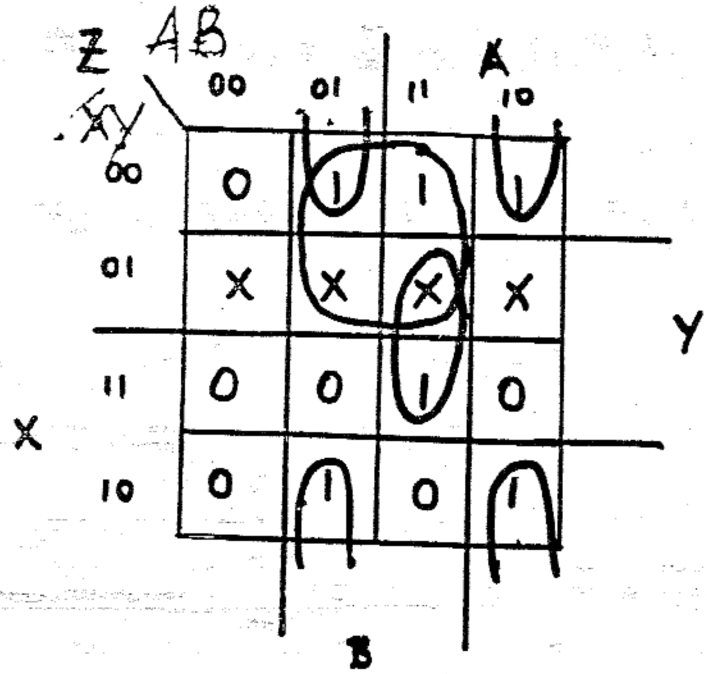
5. Construir um circuito que realize a seguinte função:

$$\begin{cases} x=0,\ y=0 \implies Z = A+B \\ x=0,\ y=1 \implies \text{Não ocorre} \\ x=1,\ y=0 \implies Z = A \oplus B \\ x=1,\ y=1 \implies Z = A.B \end{cases}$$

X	y	A	B	Z
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	X
0	1	0	1	X
0	1	1	0	X
0	1	1	1	X
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



$$Z = \bar{X}B + ABy + \bar{A}B\bar{y} + A\bar{B}\bar{y}$$



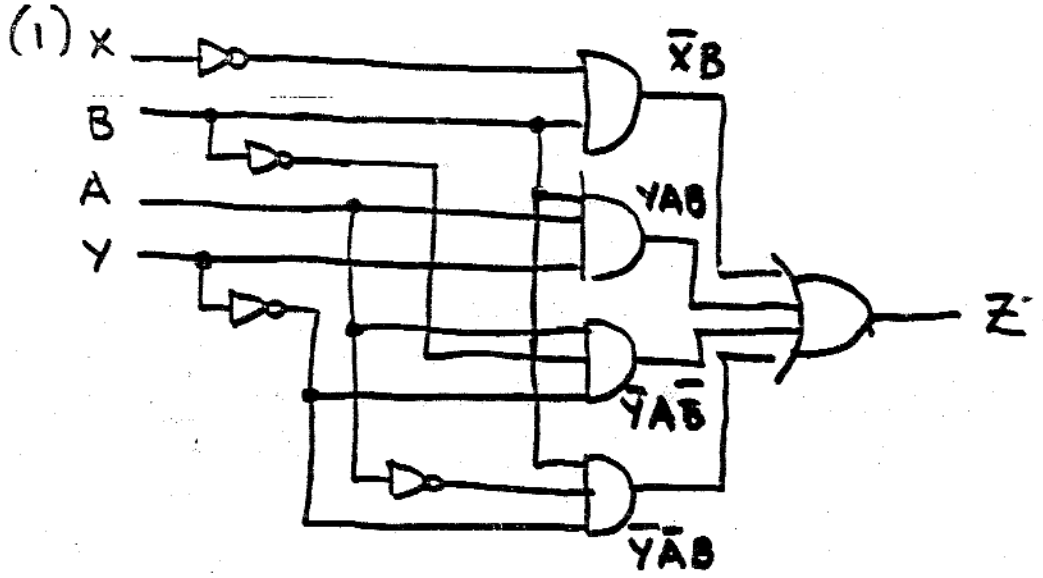
$F = \bar{X}B + YAB + \bar{Y}A\bar{B} +$

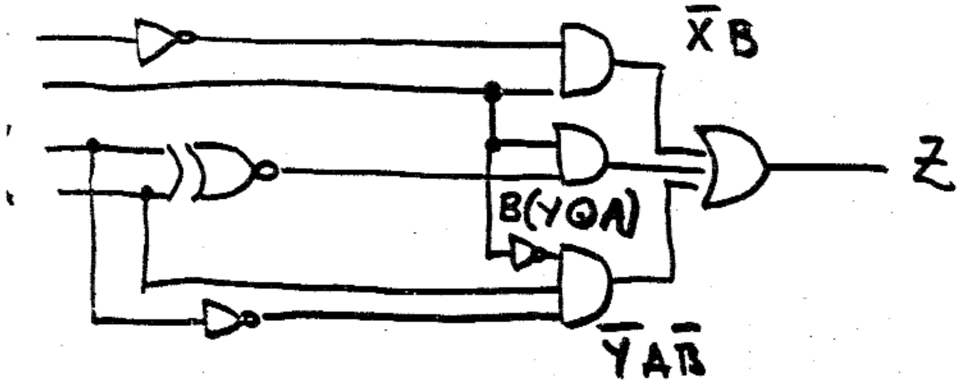
$+ \bar{Y}\bar{A}B$ (1)

$F = \bar{X}B + B(YA + \bar{Y}\bar{A}) +$

$+ \bar{Y}A\bar{B} = \bar{X}B + B(Y \odot A) +$

$+ \bar{Y}A\bar{B}$ (2)

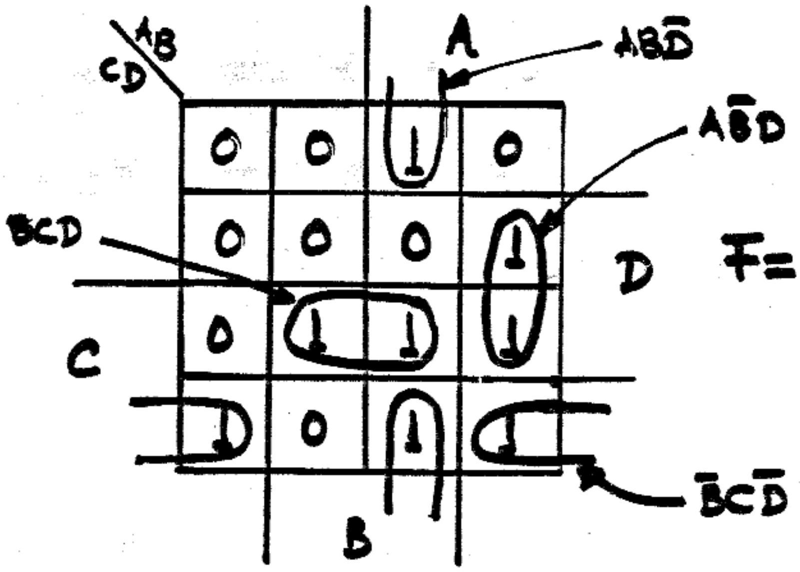




A828

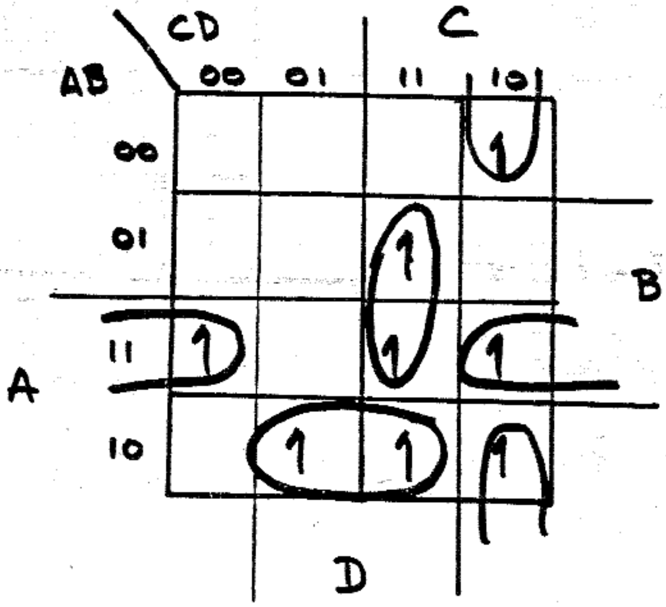
6.6. $F(A,B,C,D) = \sum 2,7,9,10,11,12,14,15$

$F = \bar{A}\bar{B}C\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD + AB\bar{C}\bar{D} + ABC\bar{D} + ABC$

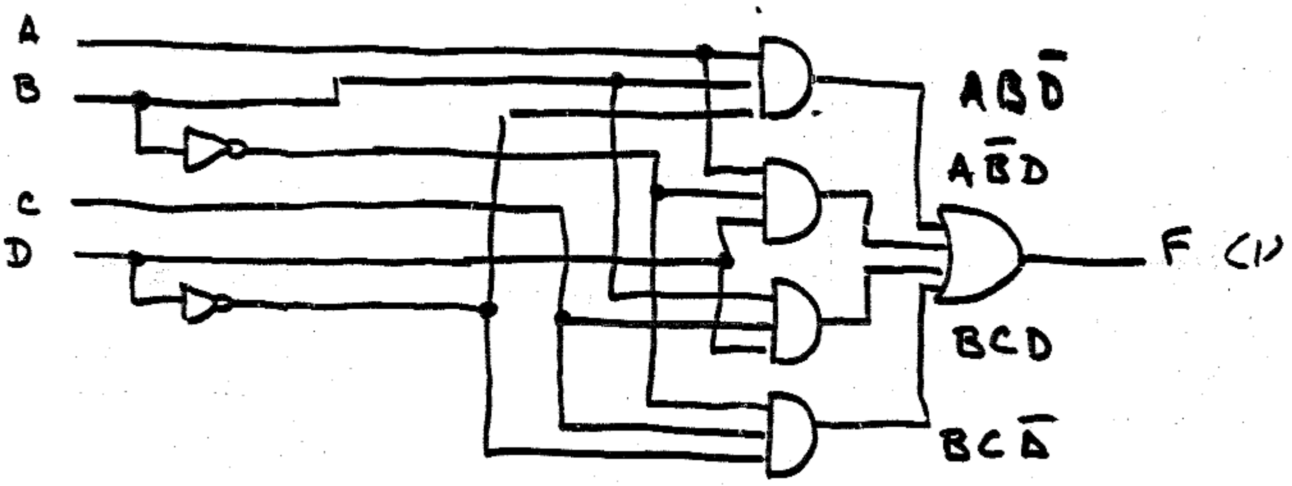


$F = AB\bar{D} + A\bar{B}D + BCD + \bar{B}C\bar{D}$

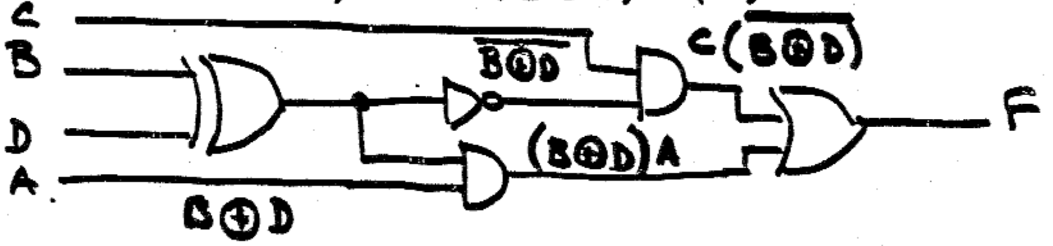
6.6 $F(A,B,C,D) = \sum 2,7,9,10,11,12,14,15$



$$F = AB\bar{D} + A\bar{B}D + BCD + \bar{B}C\bar{D} \qquad (1)$$



$$F = AB\bar{D} + A\bar{B}D + BCD + \bar{B}C\bar{D} = A(B\bar{D} + \bar{B}D) + C(\bar{B}\bar{D} + BD) =$$
$$= A(B \oplus D) + C\,\overline{(B \oplus D)} \qquad (2)$$



// ! Original

Método de Minização de Quine-McCluskey

Passo 1: A função deve estar representada por soma de mintermos. Caso não esteja expandir até que todos os mintermos contenham todas as variáveis.

Exemplo:

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

$F(a, b, c) = ab(c + \bar{c}) + bc(a + \bar{a}) = abc + ab\bar{c} + \bar{a}bc$

$F(a, b, c) = \Sigma 3, 6, 7$

Método de Minização de Quine-McCluskey

Passo 2: Os mintermos devem ser separados em grupos por índices, onde o índice é dado pelo número de váriaveis iguais a 1.

Exemplo:

$F(A, B, C, D) = \Sigma 1, 4, 10, 11, 12, 14, 15$

Índice	Mintermo
0	...
1	1,4
2	10,12
3	11,14
4	15

Coluna 1
1
4
10
12
11
14
15

Método de Minização de Quine-McCluskey

Passo 3:

A representação decimal de cada mintermo deve ser comparada com todos os mintermos do próximo grupo de índice maior. Somente devem ser considerados no próximo grupo mintermos de número mais alto. Dois mintermos devem ser combinados se seus números diferem de 2^{N_1}. Os mintermos usados devem ser marcados. Ao final os mintermos não marcados são considerados essenciais.

Exemplo: Podemos combinar o mintermo 4 do grupo 1 e mintermo 12 do grupo 2

Método de Minização de Quine-McCluskey

Passo 4:

Os termos simplificados durante o passo anterior devem ser colocados em uma próxima coluna da seguinte forma

$$I(2^N)$$

onde I é o mintermo menor e 2^N é a diferença entre os dois mintermos que foram combinados. Somente o menor mintermo precisa ser colocado na nova coluna.

Ex. Mintermos 4 e 12 combinados passam para a próxima coluna como 4(8).

Método de Minização de Quine-McCluskey

Passo 5:

Os termos de cada grupo da coluna gerada são comparados com os elementos do próximo grupo que tem as mesmas potências de 2 entre parênteses. Se os números I diferem de 2^{N_i} então eles podem ser combinados e marcados. O novo termo tem a forma $I(2^{N_1}, 2^{N_2}, \ldots)$, onde I é o número do min-termo e $2^{N_1}, 2^{N_2}, \ldots$ são as potências de 2. Estes novos termos formam a nova coluna. O processo deve continuar até que novos grupos não possam mais ser formados.

Exemplo:

Coluna 1	Coluna 2	Coluna 3
1	4(8)	10(1,4)
4√	10(1)√	
10√	10(4)√	
12√	12(2)	
11√	11(4)√	
14√	14(1)√	
15√		

Resposta: 1, 4(8), 12(2), 10(1,4)

Método de Minização de Quine-McCluskey

Resultado da simplificação após o passo 4.

Coluna 1	Coluna 2
1	4(8)
4√	10(1)
10√	10(4)
12√	12(2)
11√	11(4)
14√	14(1)
15√	

Método de Minização de Quine-McCluskey

Passo 6:

Conversão para soma de mintermos.

1. Escreva o número em forma de equação;
2. Corte os literais correspondentes as potências entre entre parênteses.

Exemplo:

Resposta: 1, 4(8), 12(2), 10(1,4)

$$
\begin{array}{|lcccc|}
 & & 8\,4\,2\,1 & & \\
1 & \rightarrow & \bar{A}\bar{B}\bar{C}D & \rightarrow & \bar{A}\bar{B}\bar{C}D \\
4(8) & \rightarrow & \cancel{\bar{A}}B\bar{C}\bar{D} & \rightarrow & B\bar{C}\bar{D} \\
12(2) & \rightarrow & AB\cancel{\bar{C}}\bar{D} & \rightarrow & AB\bar{D} \\
10(1,4) & \rightarrow & A\cancel{\bar{B}}C\cancel{\bar{D}} & \rightarrow & AC
\end{array}
$$

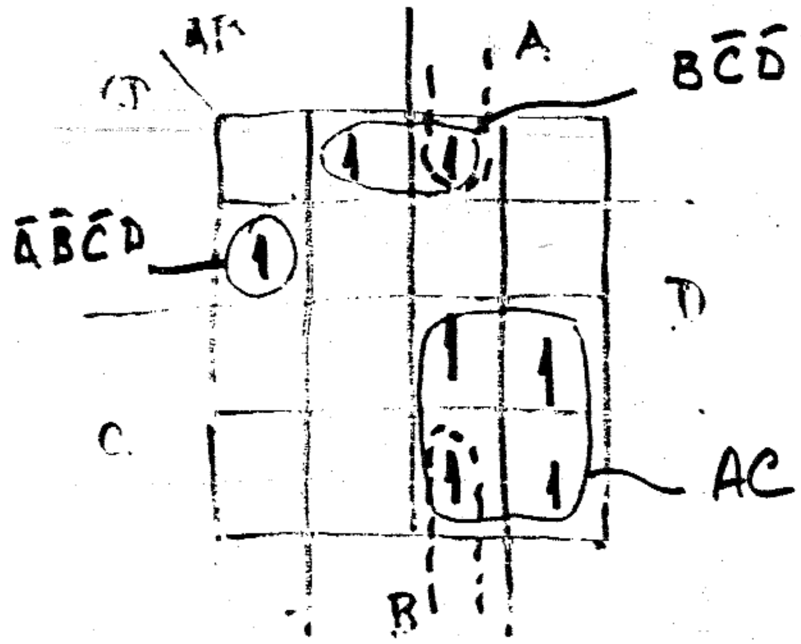
Resposta Final:

$$F(A,B,C,D) = \bar{A}\bar{B}\bar{C}D + B\bar{C}\bar{D} + AB\bar{D} + AC$$

AB3C

SIMPLIFICAÇÃO P/ KARNAUGH DO EXEMPL

$$F(A,B,C,D) = \Sigma\ 1,4,10,11,12,14,15$$



$$F = AC + B\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D$$

- $\underline{AB\bar{D}}$ É UM ~~MINTERMO~~ IMPLICANTE PRIMO REDUNDANTE!
- O MÉTODO ~~KARNAUGH~~ QUINE ENCONTRA TODOS OS ~~TERMOS~~ IMP-PRIMOS POSSÍVEIS SEM CONSIDERAR OS QUE SÃO ESSENCIAIS.

AB3

SELEÇÃO DOS MINTERMOS

CONSTRUIR A TABELA ABAIXO

	1	4	10	11	12	14	15
1	X						
4(8)		X			X		
12(2)					X	X	
10(1,4)			X	X		X	X
	1	1	1	1	2	2	1

- COLOCAR X NOS MINTERMOS COBERTOS PELO RESULTADO;
- SOMAR X NAS COLUNAS;
- SELECIONAR TERMOS ESSENCIAIS PROCURANDO COLUNAS COM UM X;

  RESP.: 1 ; 4(8); 10(1,4)

- RETIRAR DA TABELA MINTERMOS COBERTOS PELOS ESSENCIAIS

  SE TODOS RETIRADOS FIM

ABJ8 CONTINUA

→ SELEÇÃO DOS MINTERMOS 7º (CONT.)

- REPETIR OS PASSOS ANTERIORES CASO SOBRAREM MINTERMOS DA FUNÇÃO

OBS: CASO A FUNÇÃO TENHA DON'T CARE, INCLUA-OS NA LISTA DE MINTERMOS & ACHE AS POSSÍVEIS SOLUÇÕES. NA CONSTRUÇÃO DA TABELA FINAL IGNORE-OS.

AB3c

Método de Minimização de Quine-McCluskey

Ex. Simplifique a função $F(a, b, c, d, e) = \Sigma$ 0, 1, 2, 3, 10, 16, 17, 18, 19, 28, 29

$$F = \underbrace{\bar A\bar B\bar C\bar D\bar E}_{0 \to 0} + \underbrace{\bar A\bar B\bar C\bar D E}_{1 \to 1} + \underbrace{\bar A\bar B\bar C D\bar E}_{2 \to 1} + \underbrace{\bar A\bar B\bar C D E}_{3 \to 2} + \underbrace{\bar A B\bar C D\bar E}_{10 \to 2} + \underbrace{A\bar B\bar C\bar D\bar E}_{16 \to 1}$$
$$+ \underbrace{A\bar B\bar C\bar D E}_{17 \to 2} + \underbrace{A\bar B\bar C D\bar E}_{18 \to 2} + \underbrace{A\bar B\bar C D E}_{19 \to 3} + \underbrace{ABC\bar D\bar E}_{28 \to 3} + \underbrace{ABC\bar D E}_{29 \to 4}$$

Coluna 1	Coluna 2	Coluna 3	Coluna 4
0√	0(1)√	0(1,2)√	0(1,2,16)
1√	0(2)√	0(1,16)√	
2√	0(16)√	0(2,16)√	
16√	1(2)√	1(2,16)√	
3√	1(16)√	2(1,16)√	
10√	2(1)√	16(1,2)√	
17√	2(8)		
18√	2(16)√		
19√	16(1)√		
28√	16(2)√		
29√	3(16)√		
	17(2)√		
	18(1)√		
	28(1)		

Resultado = 0(1,2,16), 2(8), 28(1)

Método de Minimização de Quine-McCluskey

$$
\begin{array}{lll}
0(1,2,16) & = \bar{b}\bar{c} & \cancel{\bar{A}}\bar{B}\bar{C}\cancel{\bar{D}}\cancel{\bar{E}} \\
2(8) & = \bar{a}\bar{c}d\bar{e} & \bar{A}\cancel{\bar{B}}\bar{C}D\bar{E} \\
28(1) & = abc\bar{d} & ABC\bar{D}\cancel{\bar{E}}
\end{array}
$$

Função Simplificada

$$F(a,b,c,d,e) = \bar{b}\bar{c} + \bar{a}\bar{c}d\bar{e} + abc\bar{d}$$

	0	1	2	3	10	16	17	18	19	28	29
0(1,2,16)	X	X	X	X		X	X	X	X		
2(8)			X		X						
28(1)										X	X

SIMPLIFIQUE USANDO O MÉTODO QUINE-McCLUSKEY.

$$F(w,x,y,z) = \sum 2, 6, 7, 9, 13, 15$$

1	2 ✓	2(4)
2	6 ✓	6(1)
	9 ✓	9(4)
3	7 ✓	7(8)
	13 ✓	13(2)
4	15 ✓	

	2 ✓	6 ✓	7	9 ✓	13 ✓	15
→ 2(4)	X	X				
6(1)		X	X			
→ 9(4)				X	X	
7(8)			X			X
13(2)					X	X
	1	2	2	1	2	2

2(4) ⇒ RETIRAR 2, 6

9(4) ⇒ RETIRAR 9, 13

FICAMOS COM A TABELA ABAIXO:

	7	15
6(1)	X	
7(8)	X	X
13(2)		X
	2	2

A SOLUÇÃO 7(8) É A MAIS BARATA.

RESP.: 2(4), 9(4), 7(8)

$$2(4) = \overset{8}{\overline{w}}\,\cancel{\overset{4}{x}}\,\overset{2}{y}\,\overset{1}{\overline{z}} = \overline{w}\,y\,\overline{z}$$

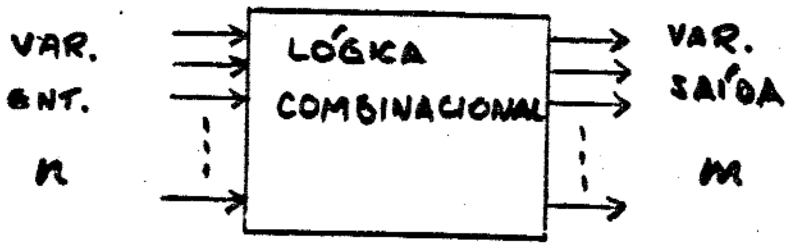
$$9(4) = w\,\cancel{x}\,\overline{y}\,z = w\,\overline{y}\,z$$

$$7(8) = \cancel{\overline{w}}\,x\,y\,z = x\,y\,z$$

$$F(w,x,y,z) = \underline{\overline{w}\,y\,\overline{z} + x\,y\,z + w\,\overline{y}\,z}$$

LÓGICA COMBINACIONAL

UM CIRCUITO COMBINACIONAL CONSISTE DE VARIÁVEIS DE ENTRADA, PORTAS LÓGICAS E VARIÁVEIS DE SAÍDA.



PROJETO:

1) DEFINA O PROBLEMA.

2) DETERMINE O NÚMERO DE VARIÁVEIS DE ENTRADA E SAÍDA.

3) DE NOMES AS VARIÁVEIS.

4) ESCREVA A TABELA VERDADE QUE DESCREVE O CIRCUITO.

5) OBTENHA UMA FUNÇÃO BOOLEANA SIMPLIFICADA PARA CADA UMA DAS VARIÁVEIS DE SAÍDA.

6) DESENHE O CIRCUITO LÓGICO.

C.1 SOMADORES

TABUADA: $0+0=0$ $\quad 1+0=1$

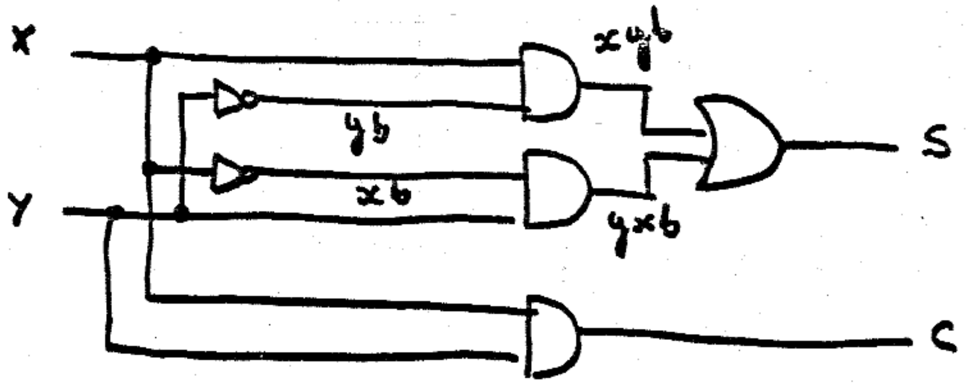
$0+1=1$ $\quad 1+1=10$

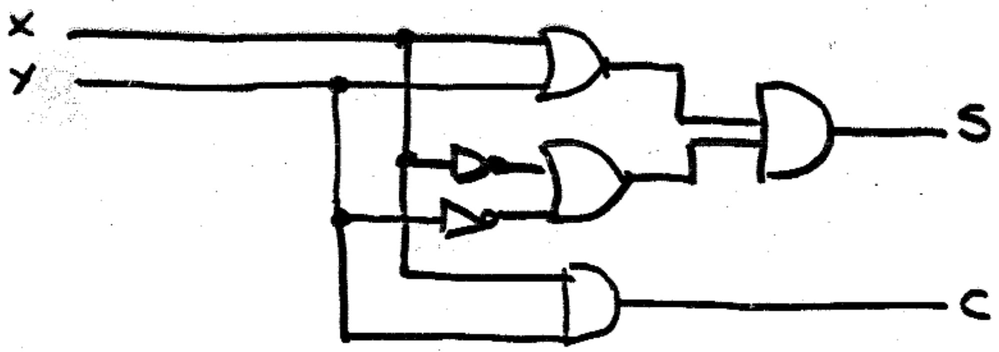
C.1.1 CIRCUITO MEIO SOMADOR: SOMA DOIS BITS

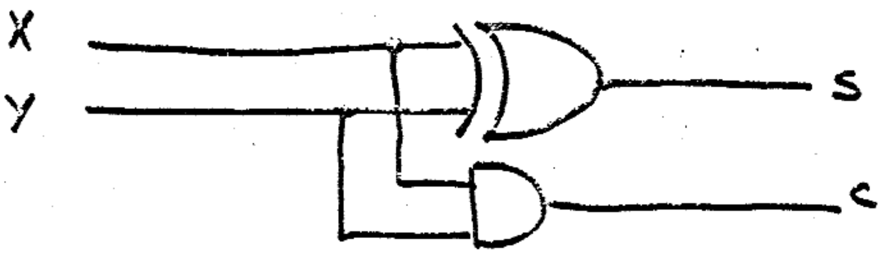
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$S = x\bar{y} + \bar{x}y = (x+y)(\bar{x}+\bar{y}) = X \oplus Y$

$C = xy$







LCZ

LC1.2 SOMADOR COMPLETO

EXECUTA A SOMA DE TRÊS BITS. DOIS (X,Y) REPRESENTAM OS DÍGITOS A SEREM SOMADOS E O TERCEIRO CORRESPONDE AO VAI UM DOS BITS MENOS SIGNIFICATIVOS.

ENT.			SAÍDAS	
X	Y	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

C_o

X \ YC_i	00	01	11	10
	0	0	1	0
X 1	0	1	1	1

Y (colunas 11, 10); C_i (colunas 01, 11)

S

X \ YC_i	00	01	11	10
	0	1	0	1
X	1	0	1	0

Y (colunas 11, 10); C_i (colunas 01, 11)

$$C_o = XY + XC_i + YC_i$$

$$S = X\overline{Y}\overline{C_i} + \overline{X}\,\overline{Y}C_i + XYC_i + \overline{X}Y\overline{C_i}$$

LC3

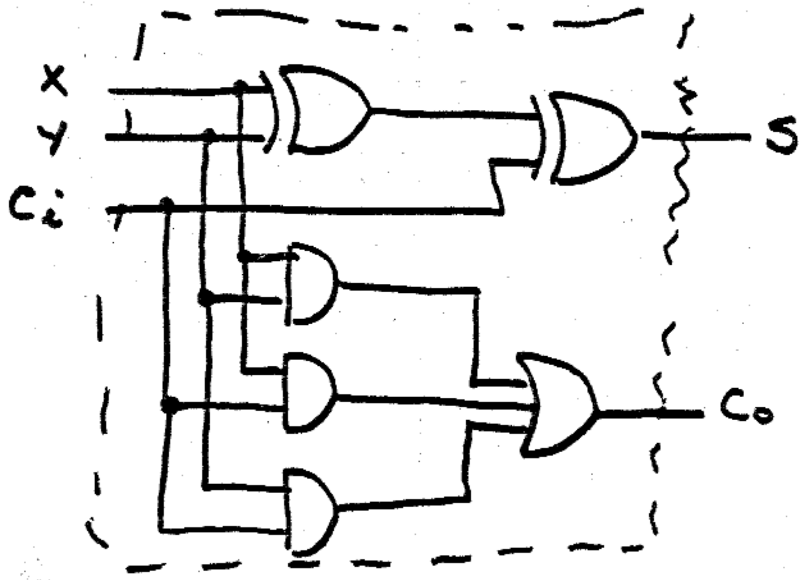
$$\boxed{C_o = XY + XC_i + YC_i}$$

$$S = X\overline{Y}\,\overline{C_i} + \overline{X}\,\overline{Y}C_i + XYC_i + \overline{X}Y\overline{C_i} =$$

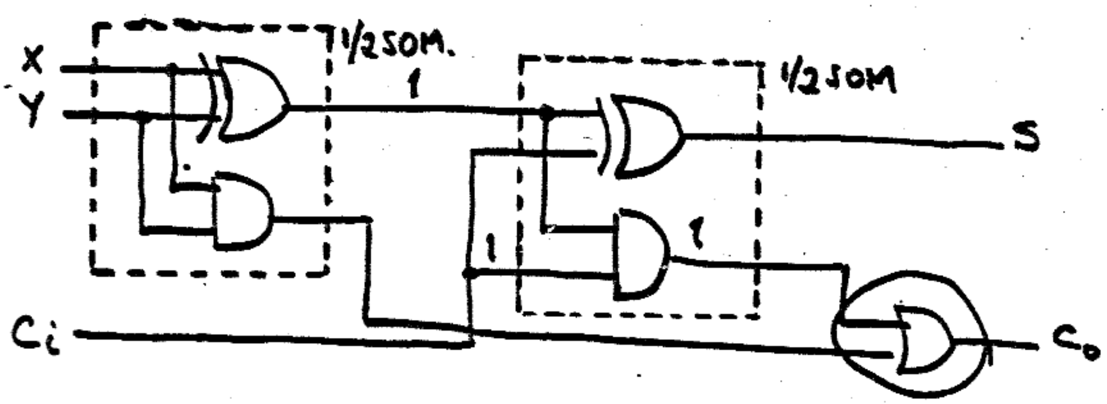
$$= \overline{C_i}(X\overline{Y} + \overline{X}Y) + C_i(\overline{X}\,\overline{Y} + XY) =$$

$$= \overline{C_i}(X \oplus Y) + C_i\overline{(X \oplus Y)} =$$

$$\boxed{S = C_i \oplus (X \oplus Y)}$$



SOMADOR COMPLETO COM 2 1/2 SOMADORES.

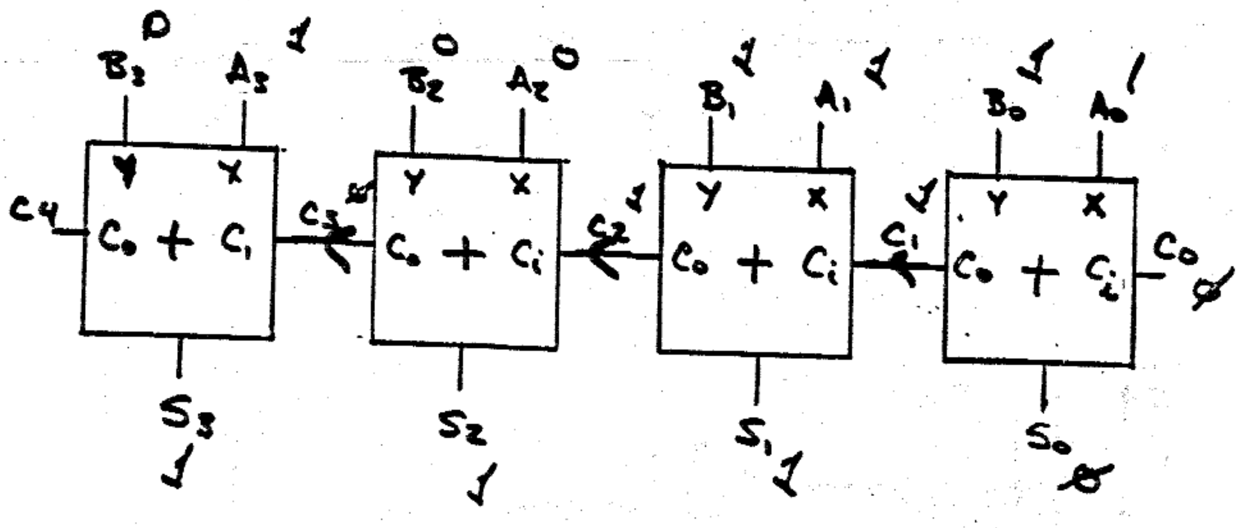


LC4

SOMADOR BINÁRIO PARALELO (DE 4 BITS)

Ex:

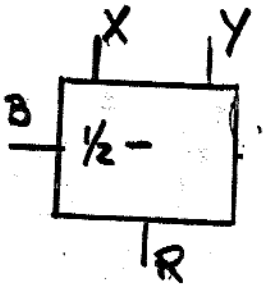
$$\begin{array}{lcccc} & & 1 & 1 & \\ A = & 1 & 0 & 1 & 1 \\ B = & 0 & 0 & 1 & 1 \\ \hline & 1 & 1 & 1 & 0 \end{array} \qquad \begin{array}{rcccc} & A_3 & A_2 & A_1 & A_0 \\ + & B_3 & B_2 & B_1 & B_0 \\ \hline & S_3 & S_2 & S_1 & S_0 \end{array}$$



LCS

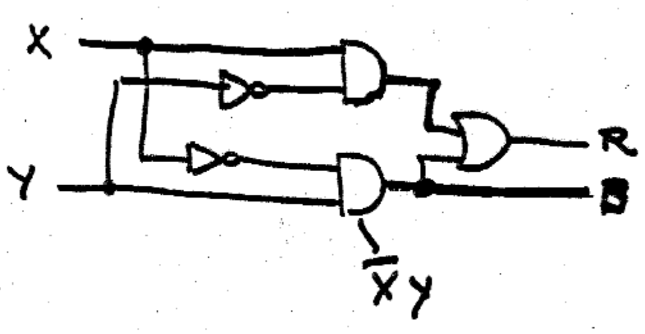
LC.2 SUBTRATORES

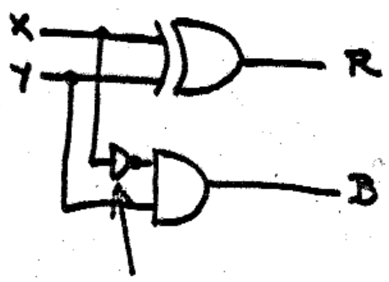
LC.2.1 MEIO - SUBTRATOR



X	Y	R	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$B = \bar{X}Y$ $R = X\bar{Y} + \bar{X}Y = X \oplus Y$





LC22 SUBTRATOR COMPLETO

X	Y	Bi	Bo	R
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Bo (Y, Bi)

X \ Y Bi	00	01	11	10
0	0	1	1	1
1	0	0	1	0

R (Y, Bi)

X \ Y Bi	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$B_o = \overline{X}Y + \overline{X}B_i + YB_i$

$R = X\overline{Y}\overline{B_i} + \overline{X}\,\overline{Y}B_i + XYB_i + \overline{X}Y\overline{B_i}$

$R = X \oplus Y \oplus B_i$

VAMOS COMPARAR AS EXPRESSÕES DO SOMADOR COMPLETO E DO SUBTRATOR COMPLETO

SOMADOR

$S = C_i \oplus (X \oplus Y)$

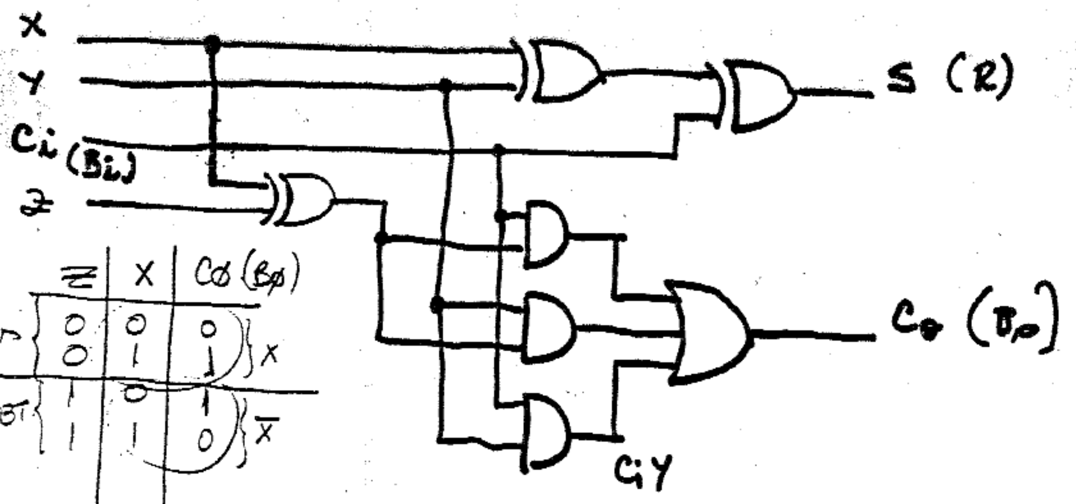
$C_o = C_i X + C_i Y + XY$

SUBTRATOR

$R = B_i \oplus (X \oplus Y)$

$B_o = B_i \overline{X} + B_i Y + \overline{X} Y$

DAS EXPRESSÕES ACIMA VEMOS QUE SE INVERTERMOS X NA EXPRESSÃO DE C_o, O SOMADOR VIRA SUBTRATOR.



	Z	X	C_o (B_o)	
SOMA	0	0	0	X
	0	1	1	
SUBT	1	0	1	$\overline{X}$
	1	1	0	

SE Z=1 ENTÃO X PASSA A SER INVERTIDO (SUBTRAÇÃO)

$Z=0 \Rightarrow$ SOMA $Z=1 \Rightarrow$ SUBTRAÇÃO

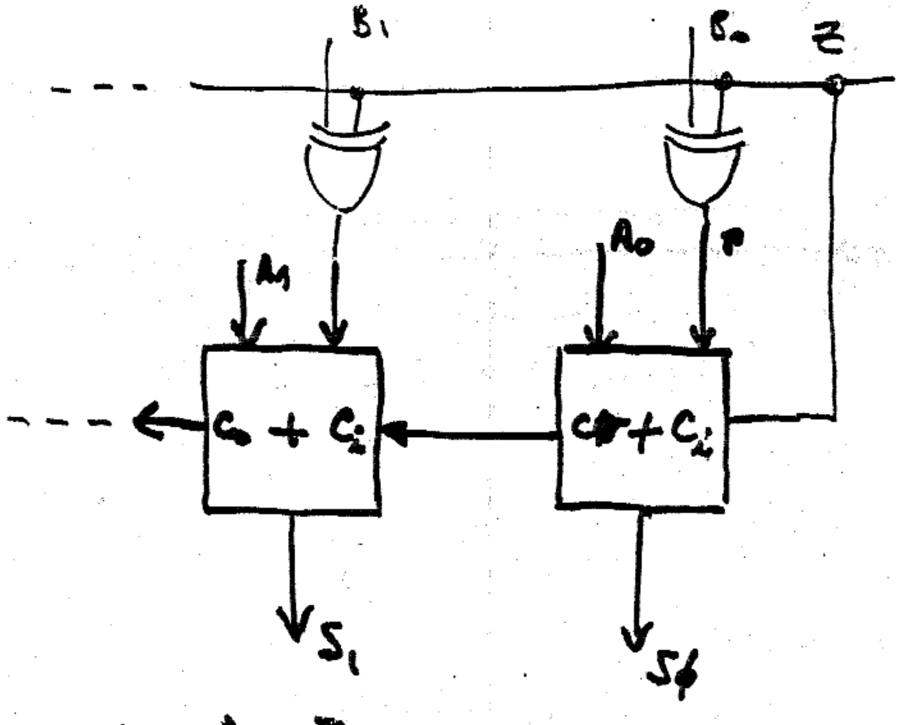
SOLUÇÃO ALTERNATIVA

UMA SUBTRAÇÃO $A - B$ É

EQUIVALENTE A $A + \overline{B} + 1$

$$\begin{array}{r} 10 \\ -\,01 \\ \hline 01 \end{array} \qquad \begin{array}{rl} 10 & — A \\ 10 & — \overline{B} \\ \hline 1\,00 & \\ +1 & +1 \\ \hline 01 & \end{array}$$

PORTANTO PODEMOS ADOTAR A SEGUINTE SOLUÇÃO, USANDO UM SOMADOR COMPLETO



$$\begin{cases} Z=0 \Rightarrow A + B \\ Z=1 \Rightarrow A + \overline{B} + 1 \Rightarrow A - B \end{cases}$$

LC.3 CONVERSÃO DE CÓDIGO

BCD ⟶ EXCESSO 3

BCD				Excesso 3			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

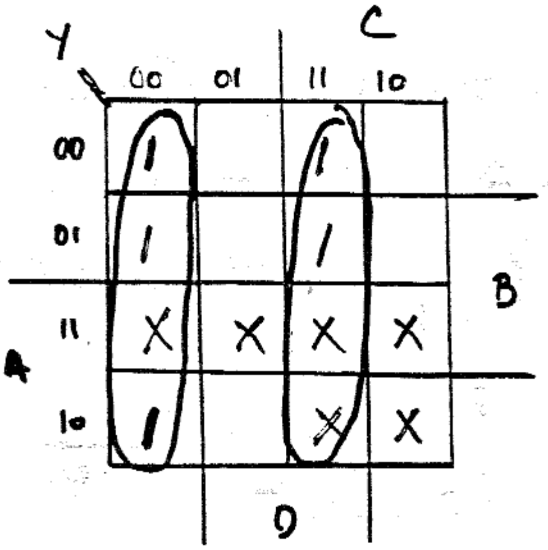
W (C, B, A, D)

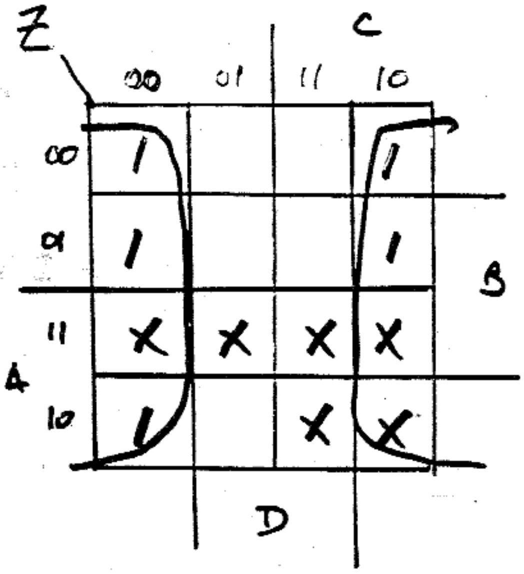
W	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

X (C, B, A, D)

X	00	01	11	10
00		1	1	1
01	1			
11	X	X	X	X
10		1	X	X

LC10



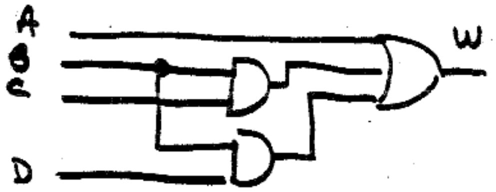


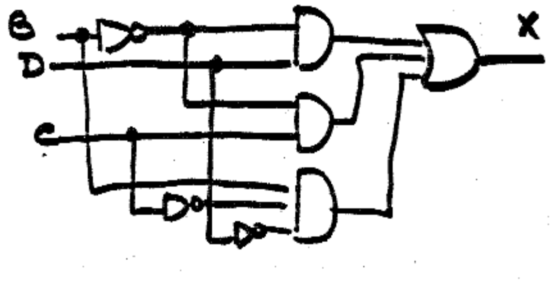
$$W = A + BC + BD = A + B(C+D)$$

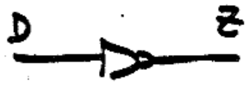
$$X = \bar{B}D + \bar{B}C + B\bar{C}\bar{D}$$

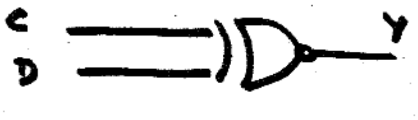
$$Y = \bar{C}\bar{D} + CD = C \odot D$$

$$Z = \bar{D}$$

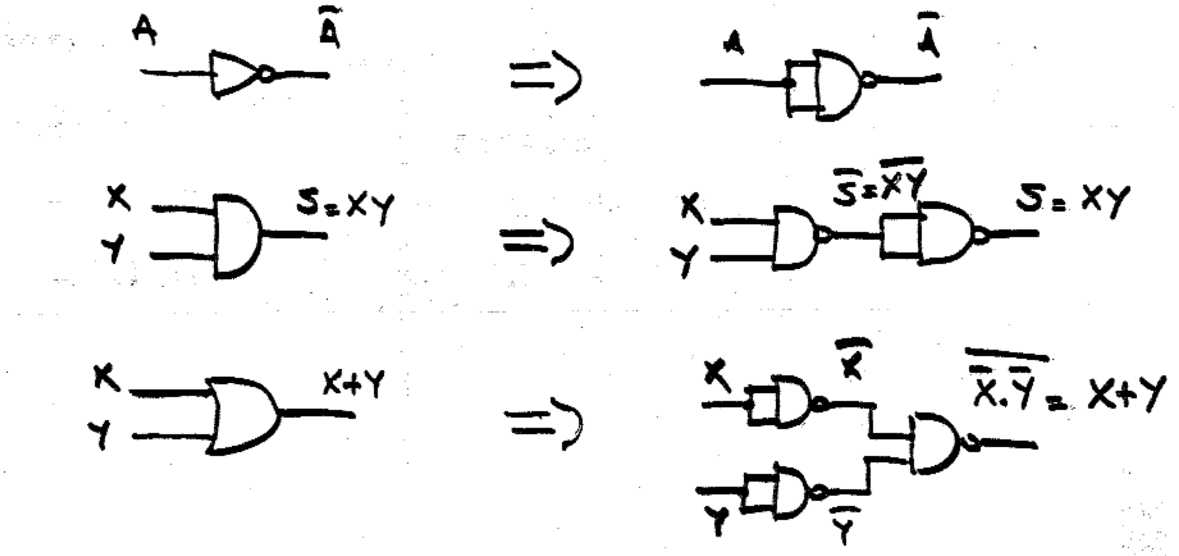








LC.4 CIRCUITOS BASEADOS EM NAND

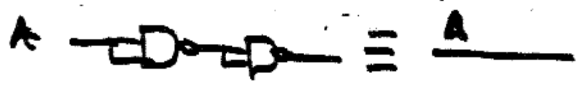


LC.4.1 MÉTODO P/ OBTER CIRCUITOS C/ NANDS.

1) DESENHE O CIRCUITO COM AS PORTAS USUAIS (AND, OR E INVERSOR).

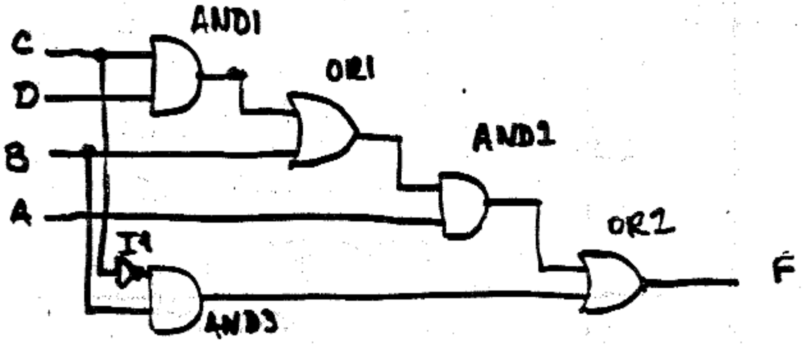
2) SUBITITUA AS PORTAS AND, OR E INVERSOR POR SUAS EQUIVALENTES USANDO NAND

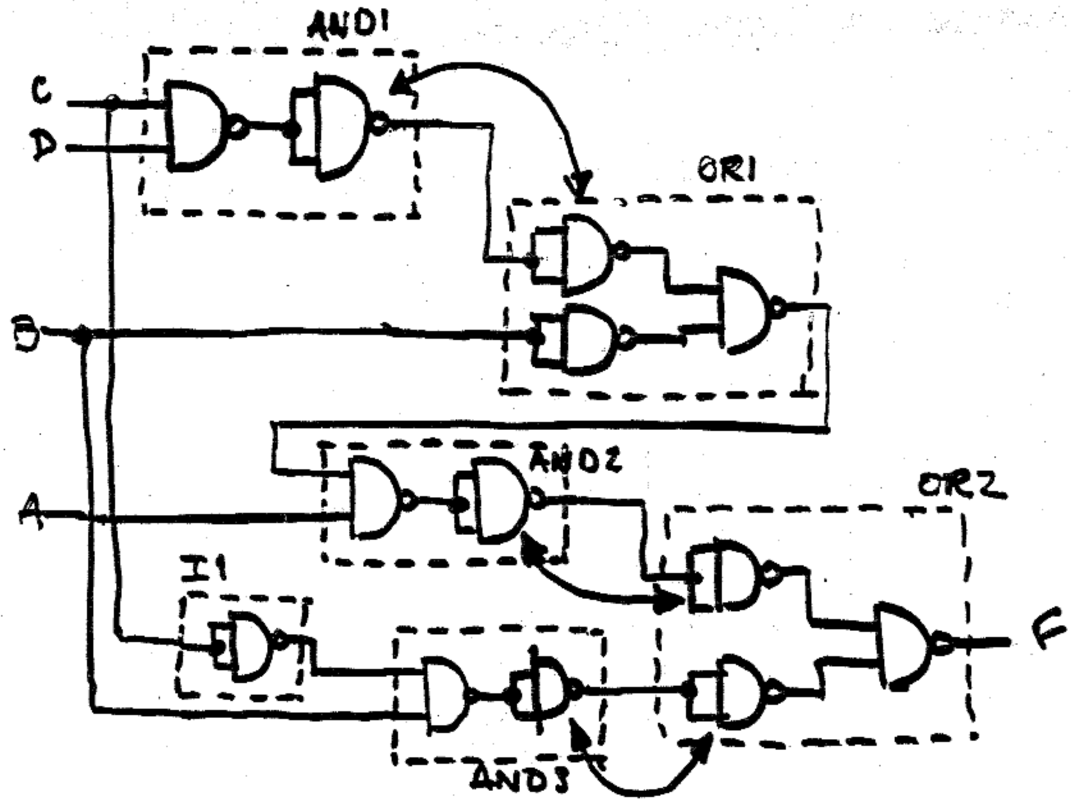
3) REMOVA DO CIRCUITO TODOS PARES DE INVERSORES EM CASCATA.



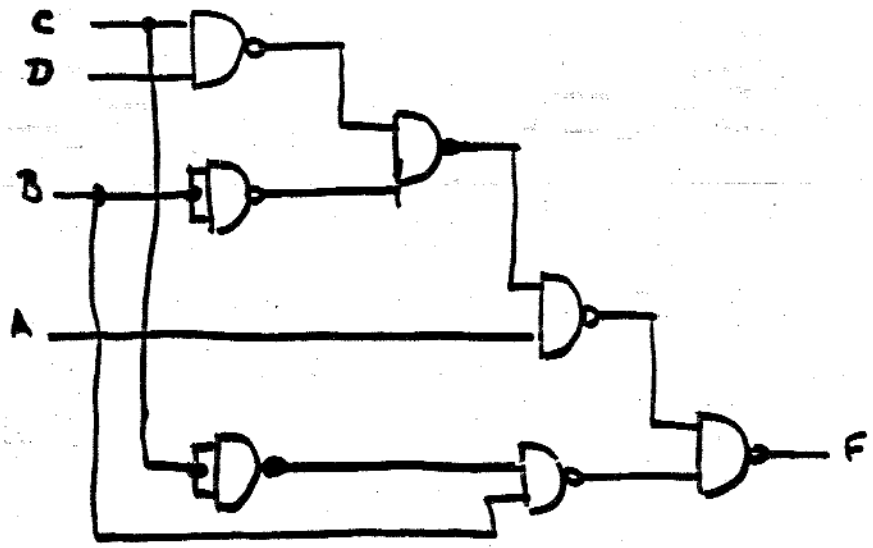
EXEMPLO: DESENHE A FUNÇÃO ABAIXO USANDO PORTAS COMUNS E EM SEGUIDA SOMENTE NAND'S.

$$F = A(B+CD) + B\overline{C}$$



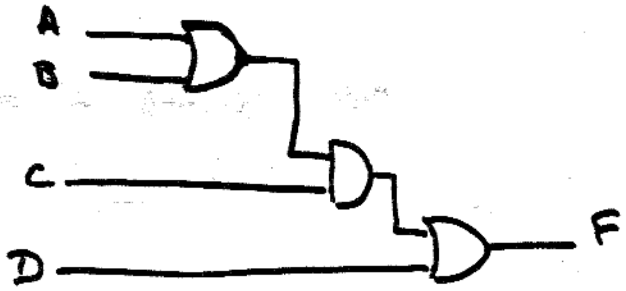


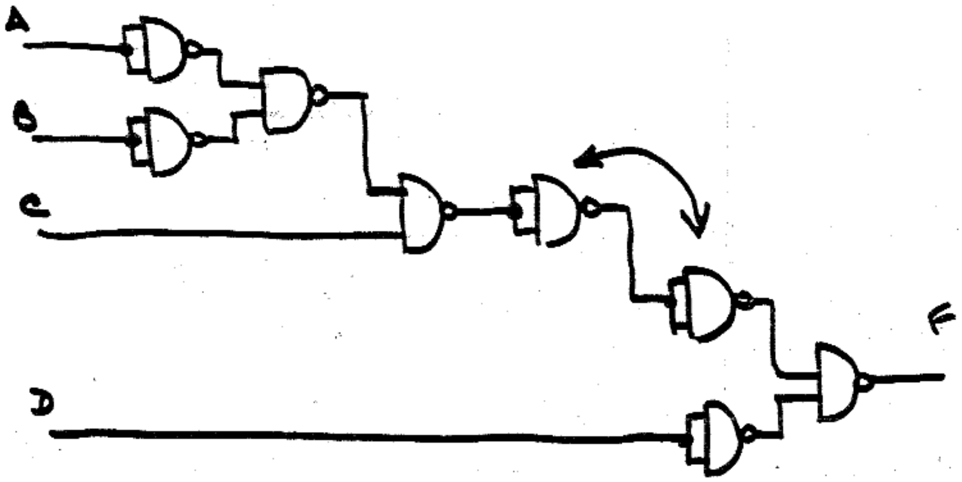
APÓS A SIMPLIFICAÇÃO DOS NANDS EM PARES.

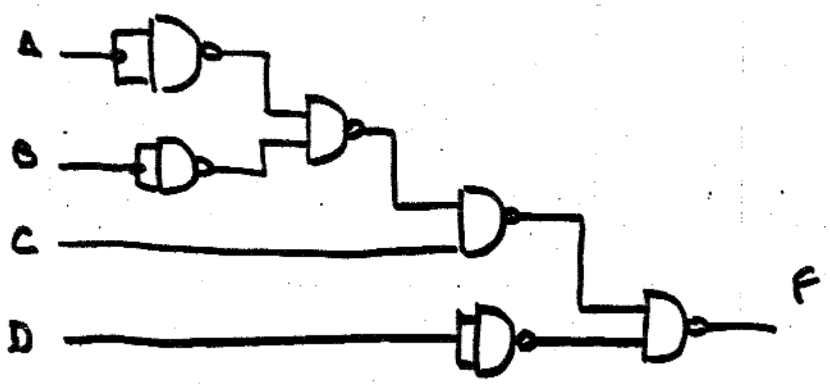


Exercício: Desenhe a função abaixo usando NANDs.

$F = D + C(A+B)$

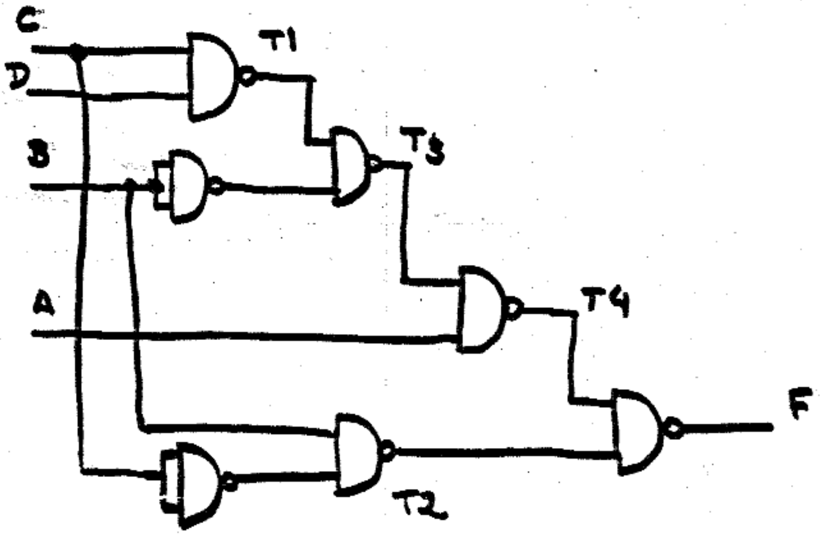






LCIS

LC.4.2 OBTENÇÃO DA FUNÇÃO BOOLEANA A PARTIR DO CIRCUITO



$$T_1 = \overline{(CD)} = \bar{C} + \bar{D}$$

$$T_2 = \overline{(B\bar{C})} = \bar{B} + C$$

$$T_3 = \overline{\bar{B} . T_1} = \overline{\bar{B}.(\bar{C}+\bar{D})} = \overline{\bar{B}\bar{C} + \bar{B}\bar{D}} = \overline{\bar{B}\bar{C}} . \overline{\bar{B}\bar{D}} =$$

$$= (B+C)(B+D) = B + CD$$

$$T_4 = \overline{AT_3} = \overline{A(B+CD)}$$

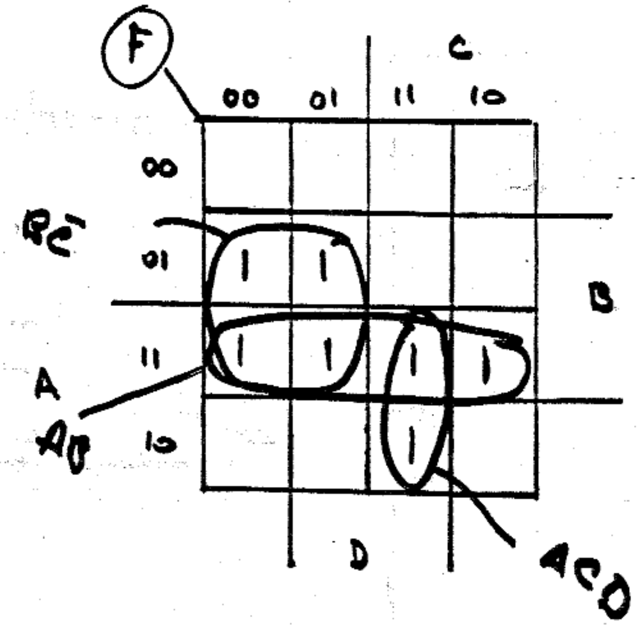
$$F = \overline{T_2 . T_4} = \overline{\overline{(B\bar{C})} . \overline{(A(B+CD))}} =$$

$$= (B\bar{C}) + A(B+CD)$$

LC.4.3

GERAÇÃO DA TABELA VERDADE

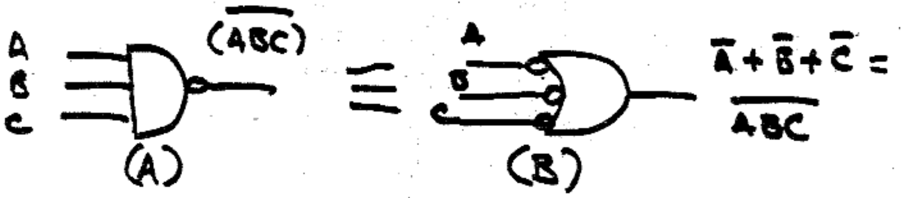
A	B	C	D	T_1	T_2	T_3	T_4	F
0	0	0	0	1	1	0	1	0
0	0	0	1	1	1	0	1	0
0	0	1	0	1	1	0	1	0
0	0	1	1	0	1	1	1	0
0	1	0	0	1	0	1	1	1
0	1	0	1	1	0	1	1	1
0	1	1	0	1	1	1	1	0
0	1	1	1	0	1	1	1	0
1	0	0	0	1	1	0	1	0
1	0	0	1	1	1	0	1	0
1	0	1	0	1	1	0	1	0
1	0	1	1	0	1	1	0	1
1	1	0	0	1	0	1	0	1
1	1	0	1	1	0	1	0	1
1	1	1	0	1	1	1	0	1
1	1	1	1	0	1	1	0	1



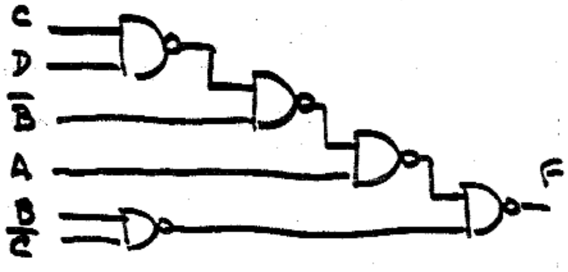
$F = AB + B\bar{C} + ACD$

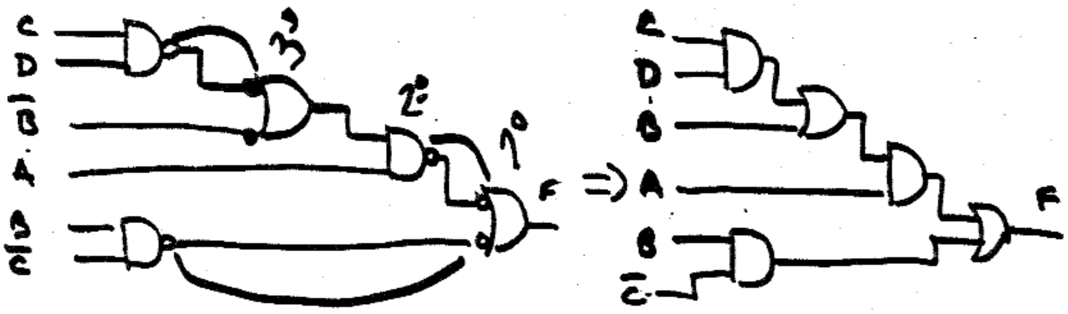
LC44 REPRESENTAÇÃO AND-OR-INVERSOR A PARTIR DA NAND-NAND.

LEMBRAR QUE

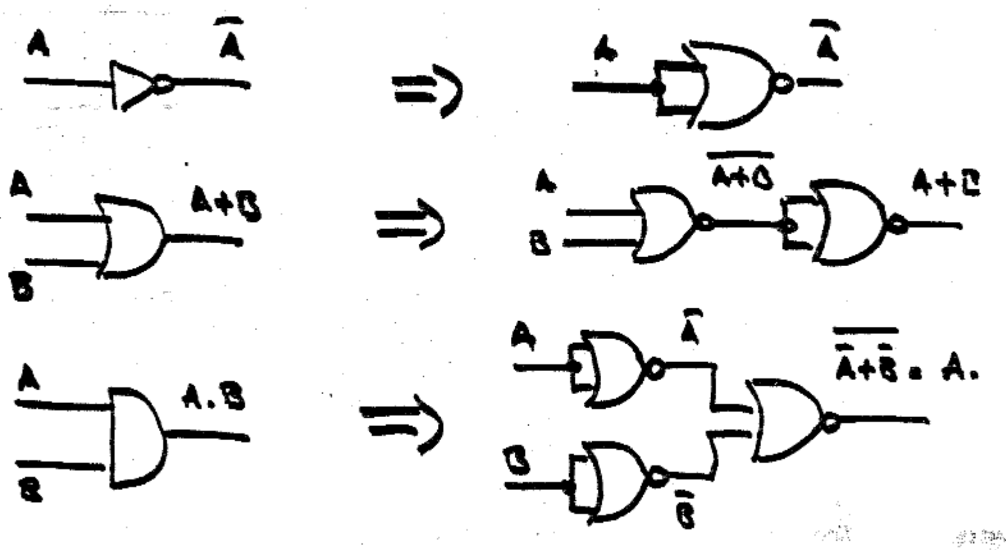


PARA RECONVERTER DE NAND-NAND PARA AND-OR-INVERSOR TROQUE DO SÍMBOLO (A) PARA (B) EM NÍVEIS ALTERNADOS DE PORTAS A PARTIR DA SAÍDA.





LC.5 Circuitos Baseados em Nor



LC.5.1 Método p/ obter circuitos c/ NORs

1) Desenhe o circuito com as portas usuais (AND, OR e inversor).

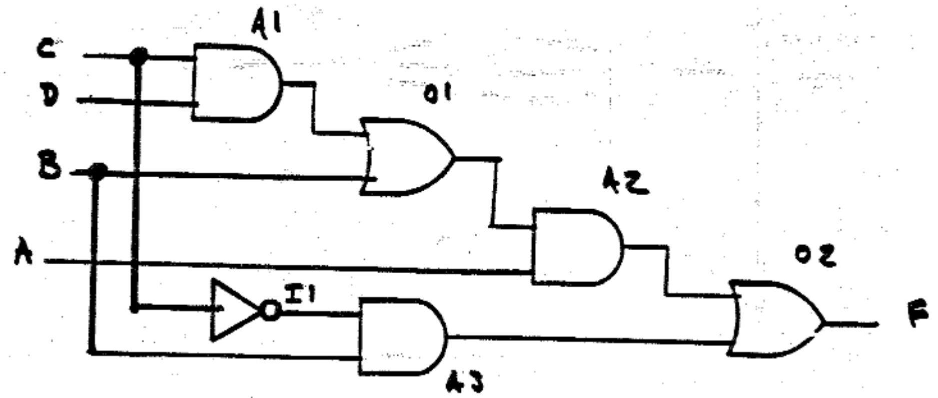
2) Substitua as portas AND, OR e inve[rsor] por suas equivalentes usando NOR.

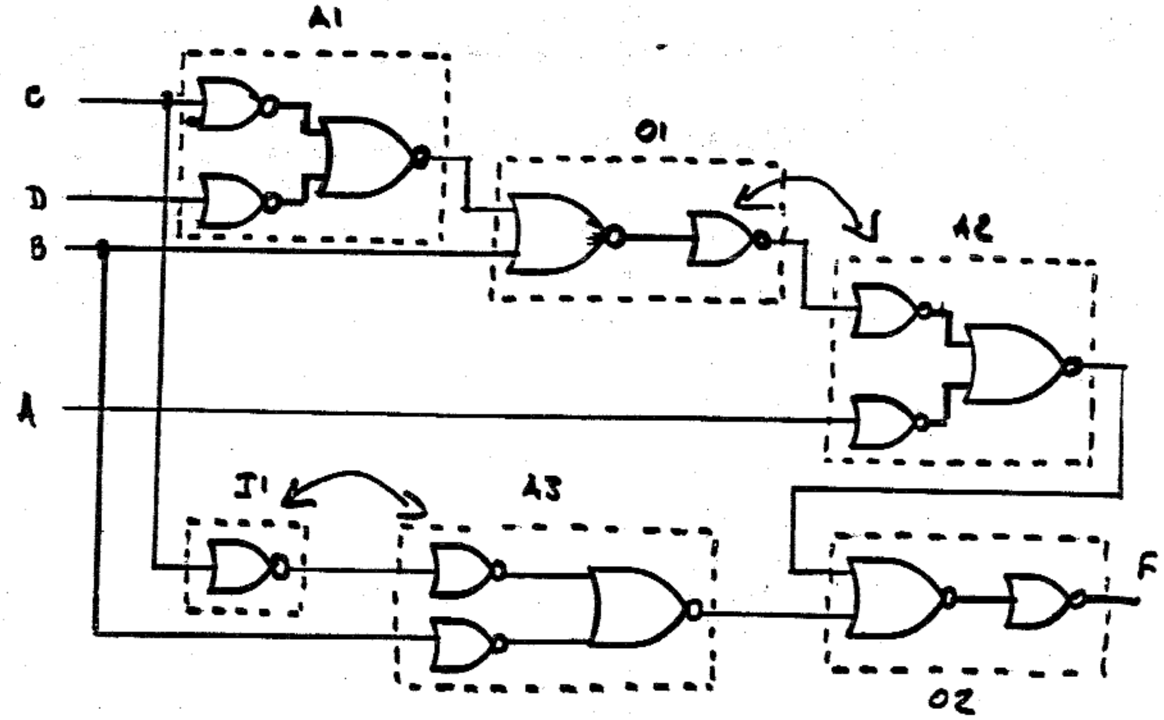
3) Remova do circuito todos os pares de inversores em cascata.

LC

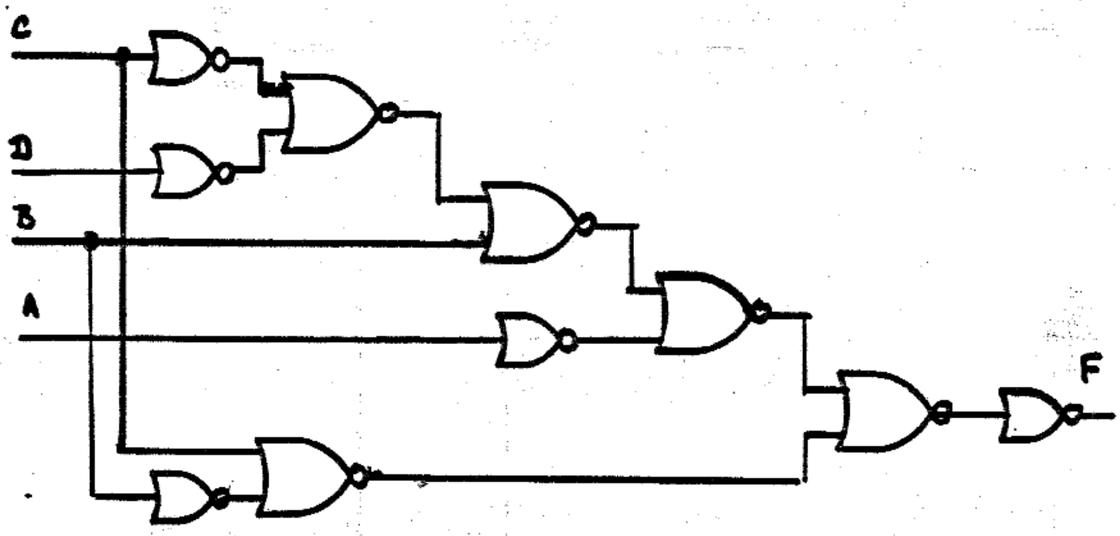
EX. DESENHE A FUNÇÃO ABAIXO USANDO PORTAS COMUNS E EM SEGUIDA SOMENTE NOR's

$$F = A(B + CD) + B\overline{C}$$



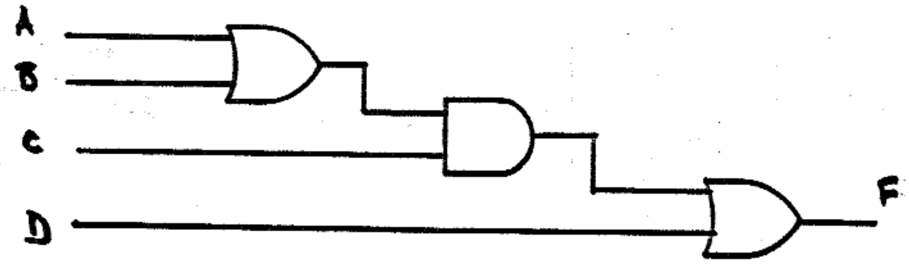


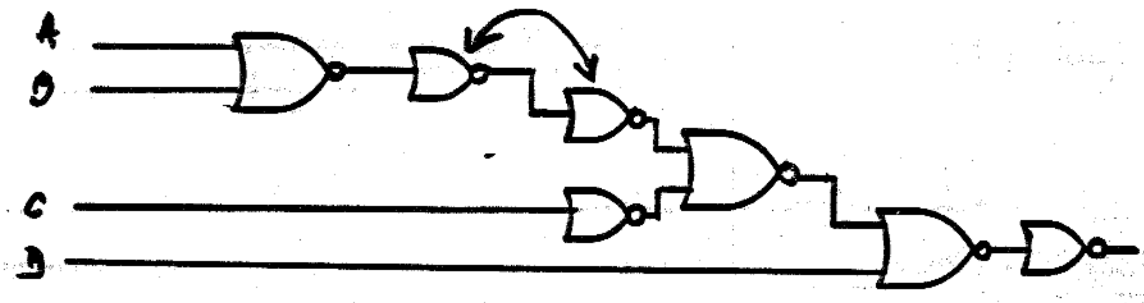
Após a simplificação dos OR's em pares temos:

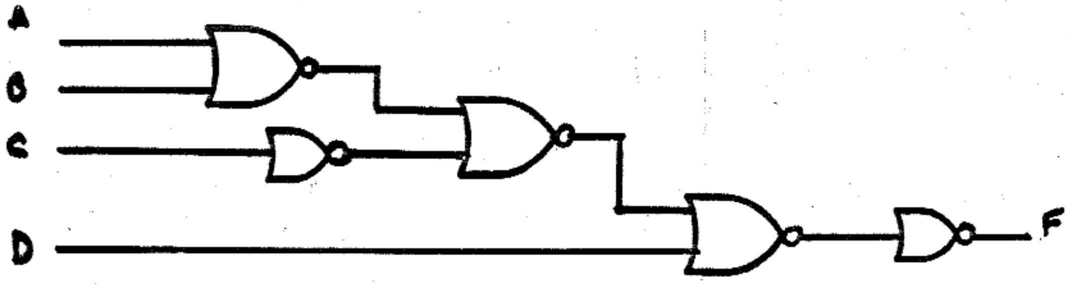


Exercício: Desenhe a função abaixo usando Nor's.

$F = D + C(A+B)$

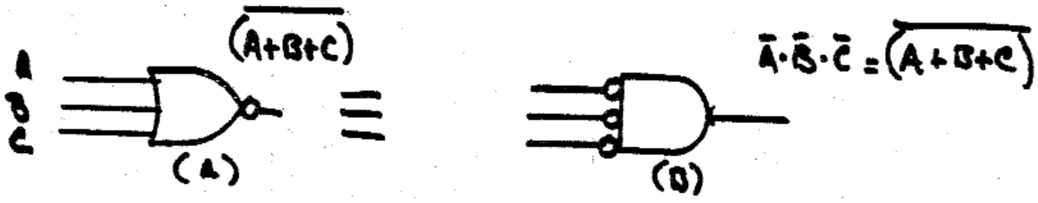




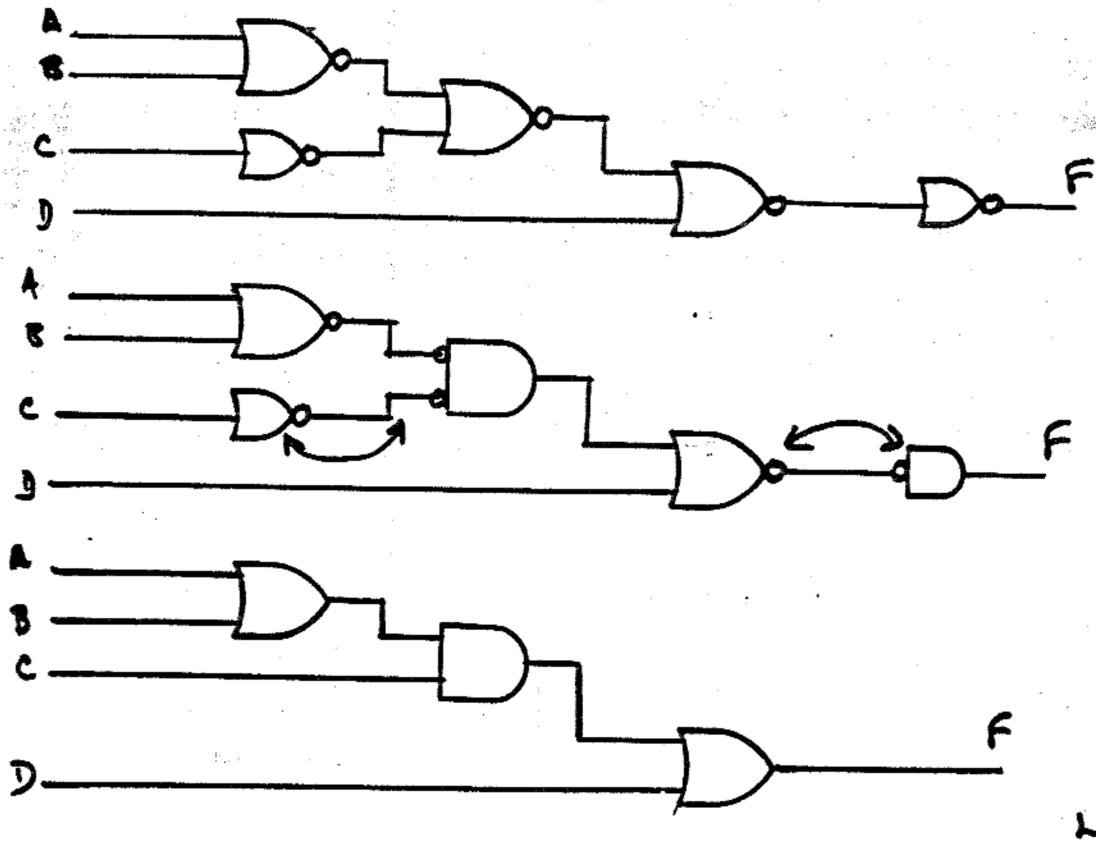


LC 5.1 Representação A-O-I a partir da NOR-NOR.

Lembrar que



Para reconverter de ~~NAND~~ NOR-NOR para A-O-I troque do símbolo (A) para o símbolo (B) em níveis alternados de portas a partir da saída.



LC6 - Funções OR-Exclusivo e NOR-Exclusivo

2-Variáveis

$$X \oplus Y = \bar{X}Y + X\bar{Y} = \overline{X \odot Y} = \overline{(XY+\bar{X}\bar{Y})}$$

$$X \odot Y = XY + \bar{X}\bar{Y} = \overline{X \oplus Y} = \overline{(X\bar{Y}+\bar{X}Y)}$$

$$A \oplus B \oplus C = (A \oplus B) \oplus C = A \oplus (B \oplus C)$$

$$A \odot B \odot C = (A \odot B) \odot C = A \odot (B \odot C)$$

3-Variáveis

$$(A \oplus B) \oplus C = (\bar{A}B + A\bar{B}) \oplus C =$$

$$= \overline{(\bar{A}B + A\bar{B})} \cdot C + (\bar{A}B + A\bar{B}) \cdot \bar{C} =$$

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

$$= (A+\bar{B})(\bar{A}+B)C + \bar{A}B\bar{C} + A\bar{B}\bar{C} =$$

$$= (AB + \bar{A}\bar{B})C + \bar{A}B\bar{C} + A\bar{B}\bar{C} = ABC + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

$$= \Sigma\, 1, 2, 4, 7$$

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

$$= A \oplus B \oplus C \equiv A \odot B \odot C$$

$$A\odot B\odot C = (\bar{A}\bar{B} + AB)\odot C =$$

$$= \overline{(\bar{A}\bar{B} + AB)}\cdot\bar{C} + (\bar{A}\bar{B} + AB)\cdot C =$$

$$= (\overline{\bar{A}\bar{B}}\cdot\overline{AB})\cdot\bar{C} + \bar{A}\bar{B}C + ABC$$

$$= (A+B)(\bar{A}+\bar{B})\cdot\bar{C} + \bar{A}\bar{B}C + ABC =$$

$$= A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + ABC =$$

$$\boxed{= \sum 1, 2, 4, 7} \qquad = \sum 001, 010, 100, 111$$

OR-EXCLUSIVO - Mintermos com números binários tendo nº ímpar de 1's.

NOR-EXCLUSIVO - Mintermos com números binários tendo nº par de 0's.

Quando o número de variáveis é ímpar as funções são equivalentes

$$A\oplus B\oplus C = A\odot B\odot C$$

Quando o número de variáveis é par as funções são complementares.

$$A\oplus B\oplus C\oplus D = \overline{A\odot B\odot C\odot D}$$

AB \ CD	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

OR-EXCLUSIVO

Nº IMPAR 1's

$A \oplus B \oplus C \oplus D$

AB \ CD	00	01	11	10
00	1		1	
01		1		1
11	1		1	
10		1		1

(EQUIVALÊNCIA)

NOR-EXCLUSIVO

Nº PAR 0's.

$A \odot B \odot C \odot D$

É INTERESSANTE SABER QUE PARA FUNÇÕES C/ NÚMERO ÍMPAR DE VARIÁVEIS TEMOS P/ EXEMPLO:

$$\overline{A \oplus B \oplus C} = A \oplus B \odot C = A \odot B \oplus C$$

$$\overline{A \odot B \odot C} = \overline{A \oplus B \oplus C}$$

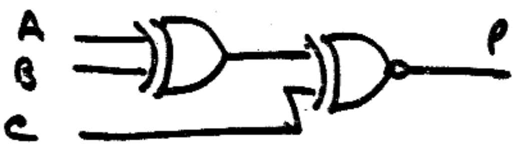
EXEMPLO DE USO DE OR/NOR EXCLUSIVO:

GERADOR DE PARIDADE:

X	Y	Z	P (ÍMPAR)
0	0	0	1
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	0

$P=1$ QUANDO HÁ UM NÚMERO PAR DE UNS NA ENTRADA. NA FUNÇÃO OR-EXCLUSIVO O Nº DE UNS É ÍMPAR, LOGO P É O COMPLEMENTO DO OR-EXCLUSIVO DE 3 VARIÁVEIS.

$$P = \overline{A \oplus B \oplus C} = A \oplus B \odot C$$



1 ANO

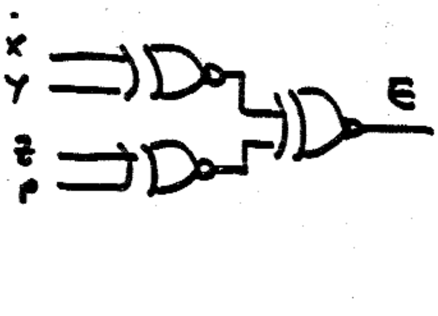
Verificador de Paridades

Dados os 3 bits (x, y, z) e a Paridade P o circuito deve indicar se houve erro na transmissão.

X	Y	Z	P	E
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

A função vale 1 nos mintermos com nº par de zeros, logo

$$E = X \odot Y \odot Z \odot P$$



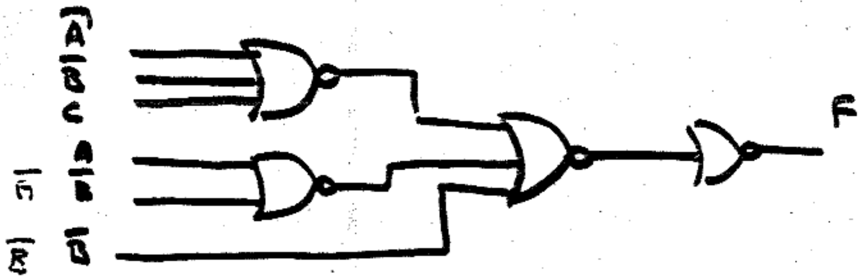
Exercícios:

1) Projete um circuito que detecte se um número de 4 bits é divisível por 2.

2) Represente a função ou-exclusivo usando somente:

a) Portas NAND (NE)
b) Portas NOR (NOU)

3) Determine a função booleana para a saída F do circuito abaixo. Obtenha um circuito equivalente usando menos portas NOR.



4) Projete um circuito que analise um nº binário de 4 bits e informe quantos bits 1's ele possui.

5) Projete um conversor de excesso-3 para código BCD baseado em somadores de 4 bits.

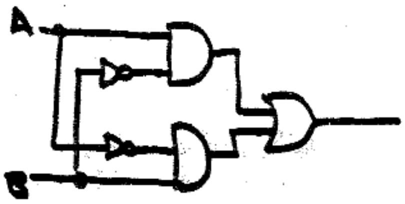
LC 30

SOLUÇÕES:

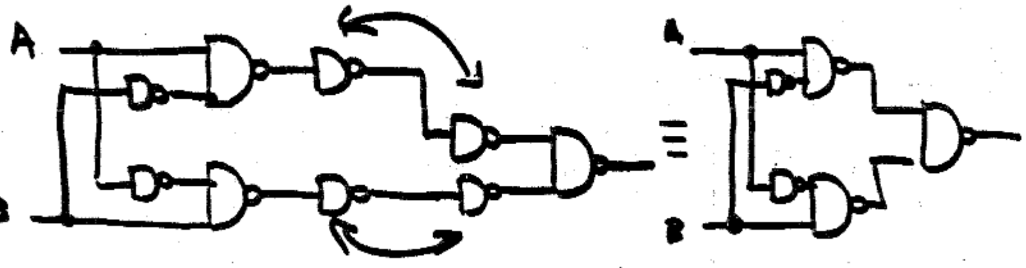
1) TODO Nº PAR É DIVISÍVEL POR 2 E TERMINA EM ZERO.

No. = $A_3 A_2 A_1 A_0$

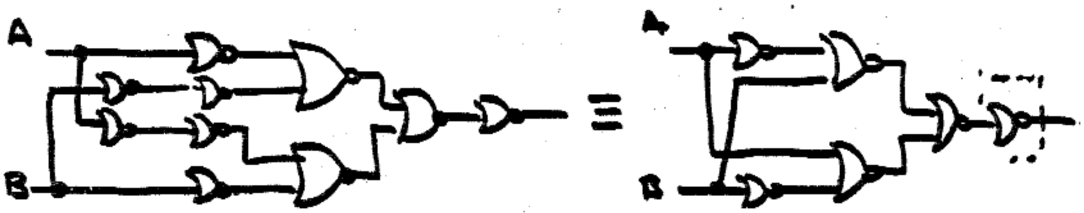
2) OR-EXCLUSIVO – $F = A\bar{B} + \bar{A}B$



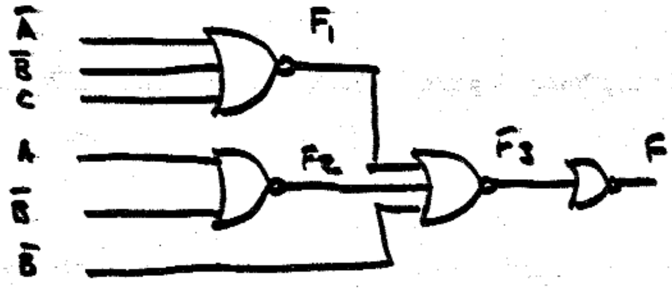
a) NAND



b) NOR



3)



$F_2 = \overline{A+\bar{B}} = \bar{A} \cdot B$

$F_1 = \overline{\bar{A}+\bar{B}+C} = A \cdot B \cdot \bar{C}$

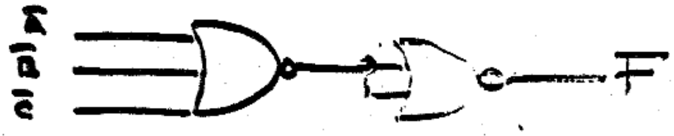
$F_3 = \overline{F_1+F_2+\bar{B}}$

$F = \overline{F_3} = F_1+F_2+\bar{B}$

$F = AB\bar{C} + \bar{A}B + \bar{B} =$

$= B(A\bar{C}+\bar{A}) + \bar{B} = B(\cancel{A+\bar{A}})(\bar{C}+\bar{A}) + \bar{B} = B(\bar{C}+\bar{A})+\bar{B}$

$= (\cancel{B+\bar{B}})(\bar{B}+\bar{C}+\bar{A}) = \bar{B}+\bar{C}+\bar{A}$

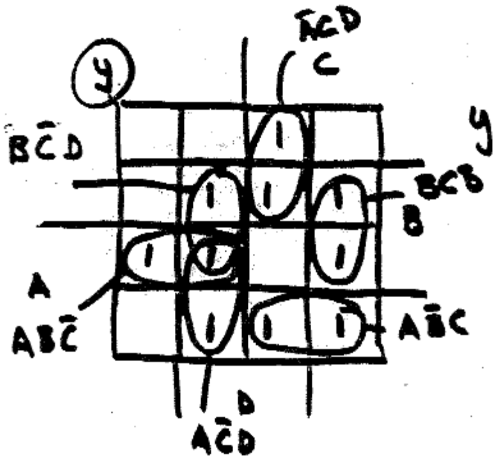


4)

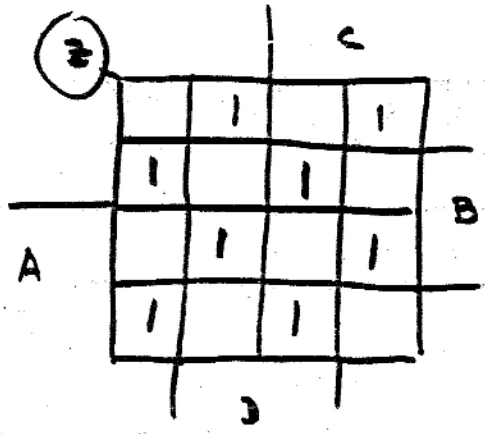
A	B	C	D	x	y	z
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	1

A	B	C	D	x	y	z
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	1	1
1	1	0	0	0	1	0
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	1	0	0

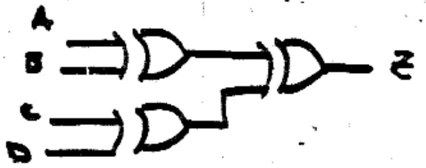
$x = ABCD$

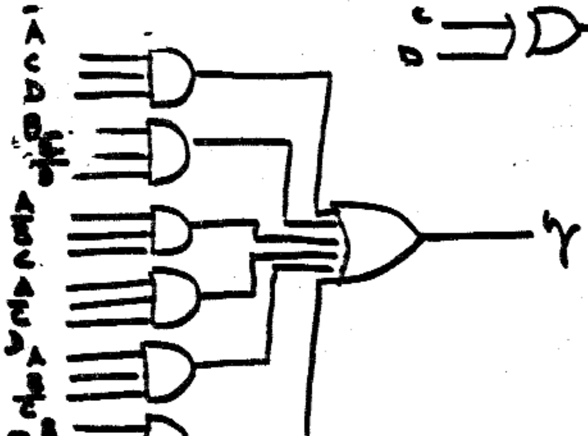


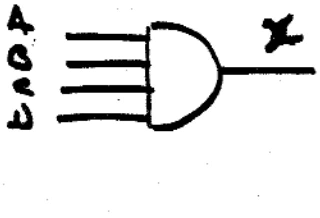
$y = \bar{A}CD + BC\bar{D} + A\bar{B}C + A\bar{C}D + A\bar{B}\bar{C} + B\bar{C}D.$



$Z = A \oplus B \oplus C \oplus D$

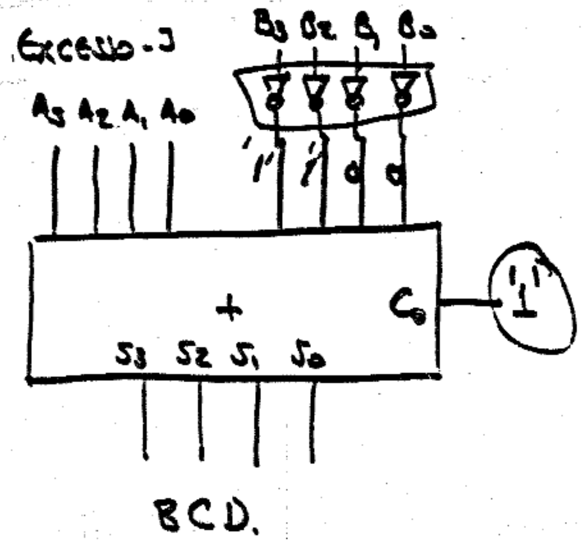






5) $A-B = A+\overline{B}+1$

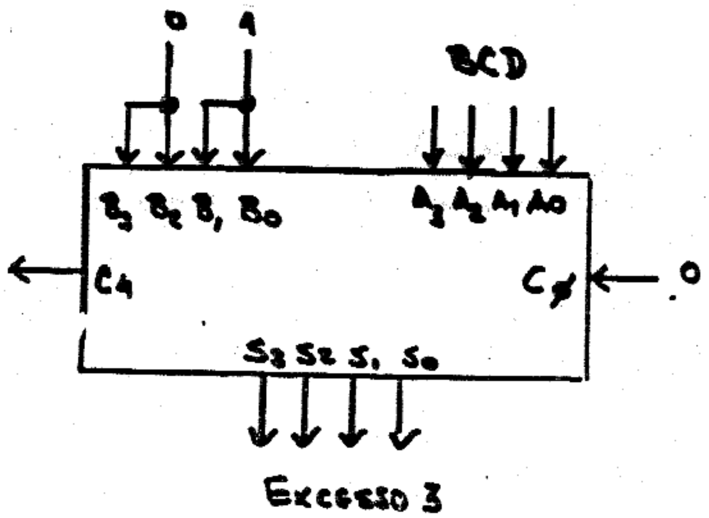
PARA CONVERTER DE EXCESSO-3 PARA BCD DEVEMOS SUBTRAIR 3 DO CÓDIGO ORIGINAL.



LÓGICA COMBINACIONAL COM INTEGRADOS MSI E LSI.

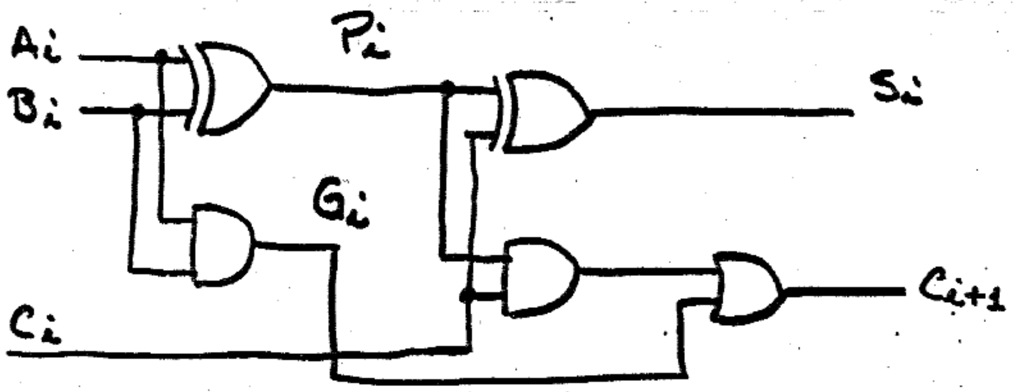
MSI.1 SOMADORES PARALELOS

EXEMPLO: DADO UM SOMADOR DE 4 BITS, TRANSFORMÁ-LO EM UM CONVERSOR BCD - EXCESSO 3.



ACELERANDO OS SOMADORES

- O CAMINHO DE GERAÇÃO DO VAI UM É O FATOR DE ATRASO NA SOMA.



EM CADA SOMADOR A SAÍDA C_{i+1} ESTÁ 2 PORTAS DEPOIS DA ENTRADA C_i. EM UM SOMADOR DE 4 BITS HÁ 8 PORTAS ENTRE C_4 E C_0.

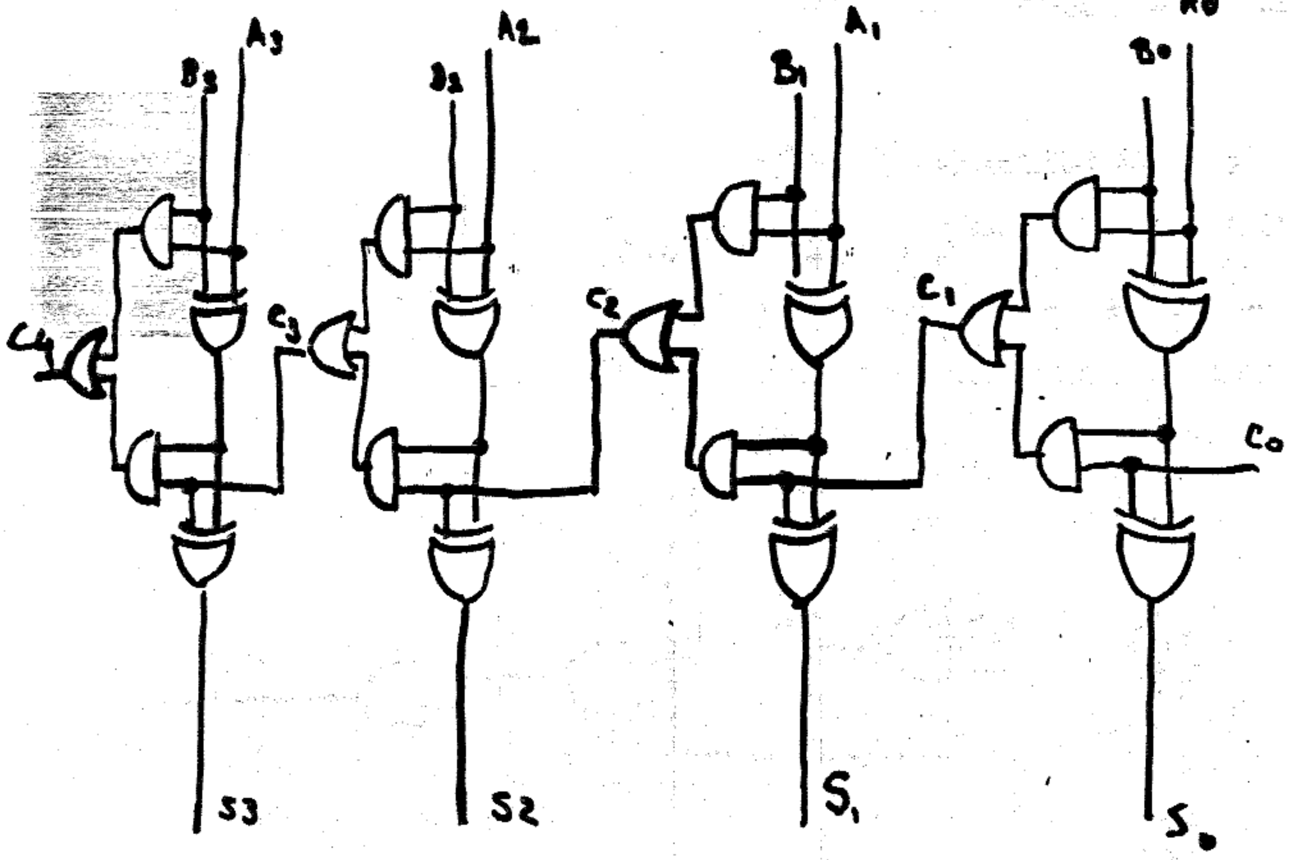
MÉTODO "CARRY LOOK AHEAD"

- $P_i = A_i \oplus B_i \Rightarrow S_i = P_i \oplus C_i$

  $G_i = A_i \cdot B_i \Rightarrow C_{i+1} = G_i + P_i C_i$

_MSI2

Circuito Somador Comum



$C_0 \rightarrow C_4$ 8 portas

$C_0 \rightarrow S_3$ 7 portas

$A_0, B_0 \rightarrow C_4$ 8 portas

$A_0, B_0 \rightarrow S_3$ 7 portas

FÓRMULA PARA GERAÇÃO DO CARRY

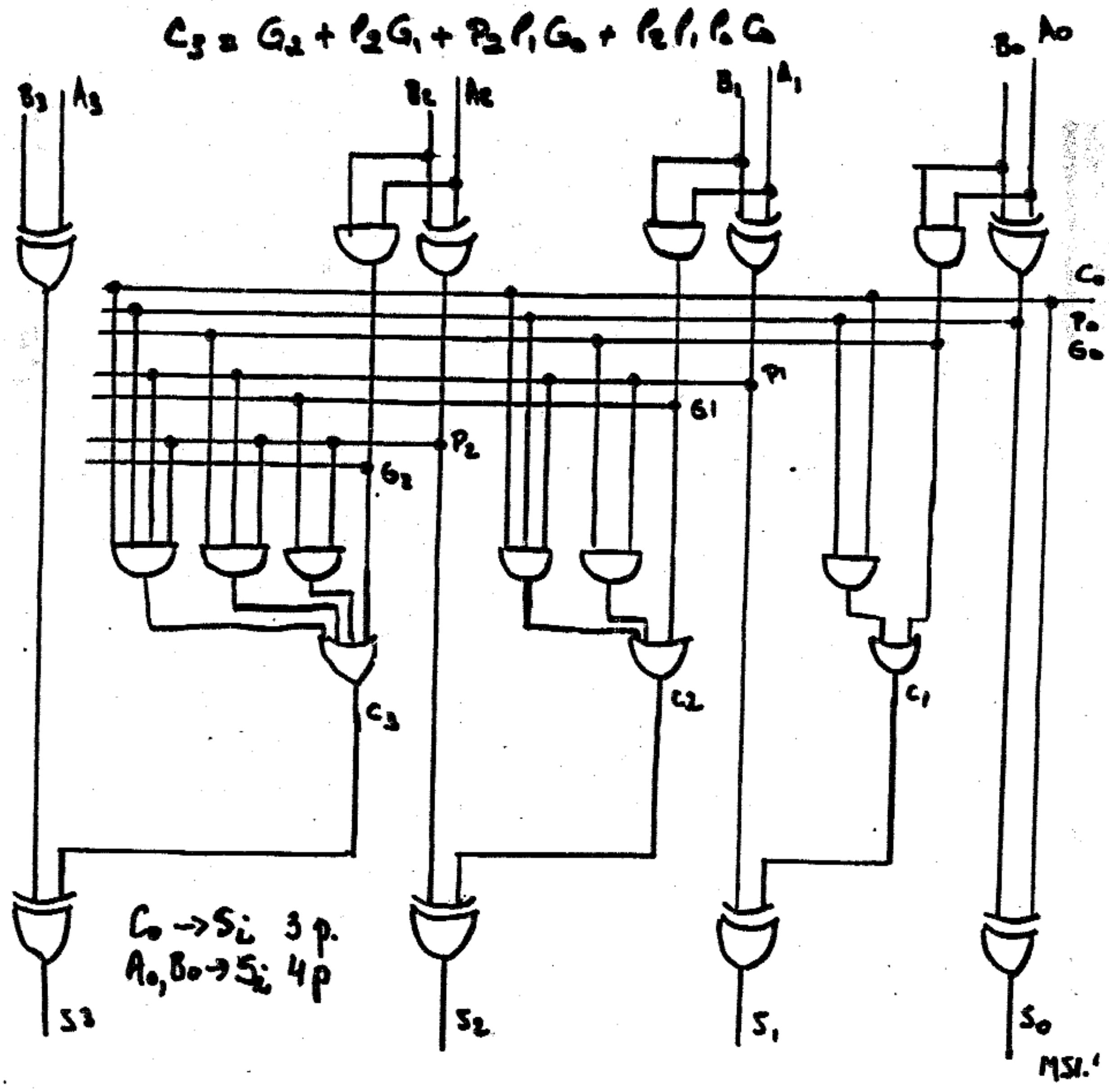
C_0

$C_1 = G_0 + P_0 C_0$

$C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C$

$C_3 = G_2 + P_2 C_2 = G_2 + P_2(G_1 + P_1 G_0 + P_1 P_0 C_0) =$

$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$



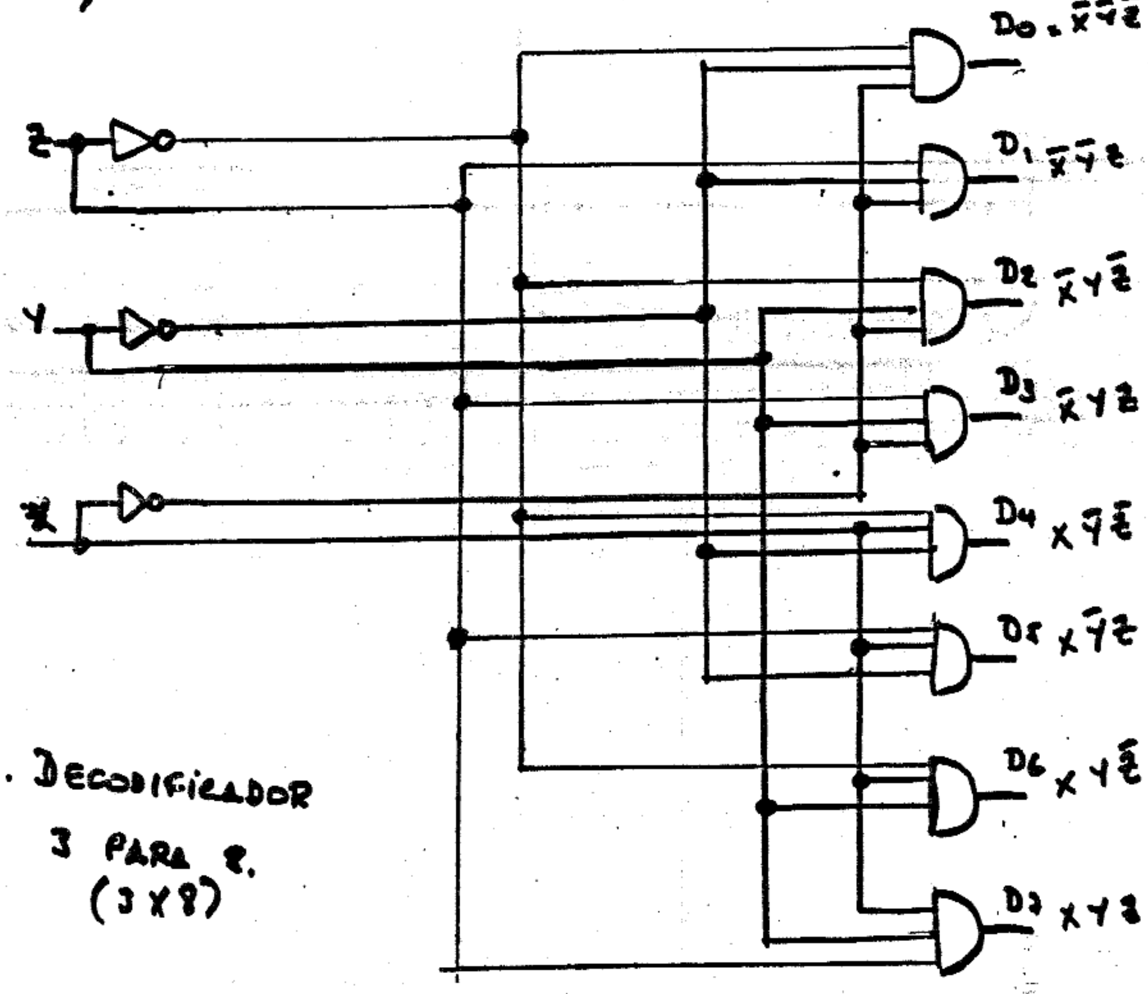
$C_0 \to S_i$ 3 p.
$A_0, B_0 \to S_i$ 4 p

MSI.'

MSI.2 Decodificadores

Decodificadores convertem números binári(os) com N bits para até 2^N saídas.

Usados quando é necessário a partir de um número binário escolher uma entre 2^N opções.



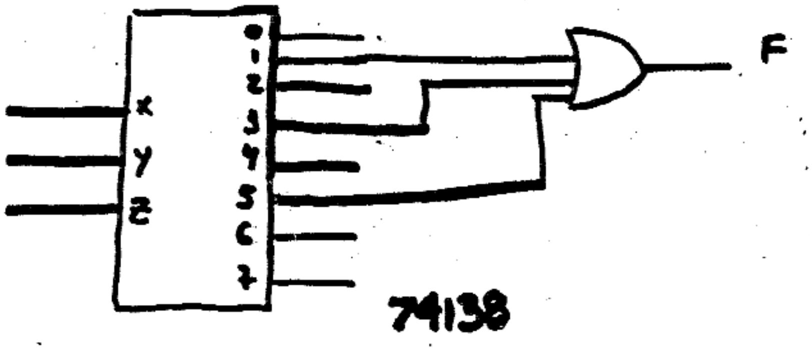
Ex. Decodificador 3 para 8. (3 x 8)

Tabela Verdade do Decodificador (3x8)

X	Y	Z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Observe que o decodificador gera todos os mintermos da função. Deste modo é possível usá-lo para implementar funções booleanas.

Ex. Função $F(x,y,z) = \sum 1,3,5.$

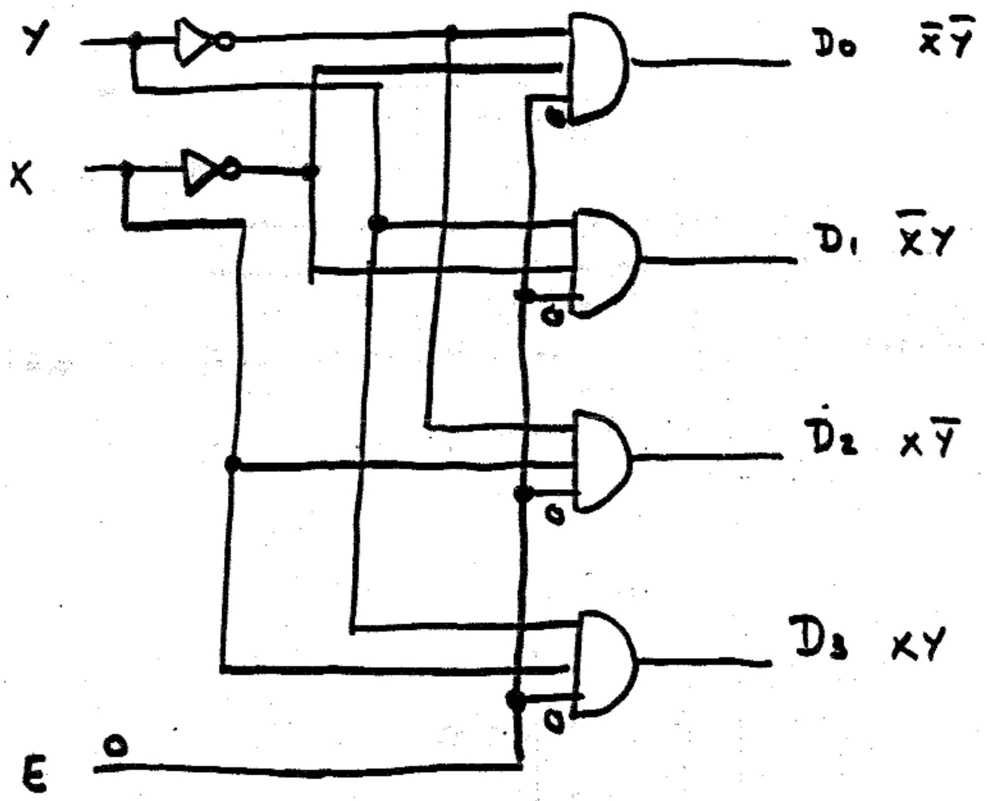


74138

MSI.6

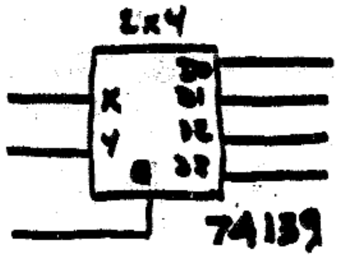
DECODIFICADOR C/ LIBERAÇÃO

E	X	Y	D_0	D_1	D_2	D_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

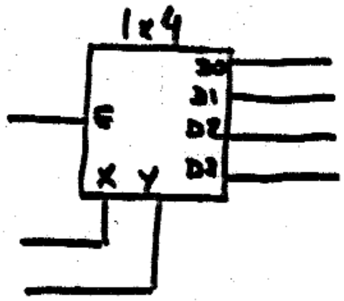


O DECODIFICADOR C/ LIBERAÇÃO PODE SER CONSIDERADO UM DEMULTIPLEXADOR, ISTO É UM CIRCUITO QUE RECEBE UM SINAL $\underline{\underline{E}}$ E O TRANSMITE PARA UMA DE 2^N SAÍDAS.

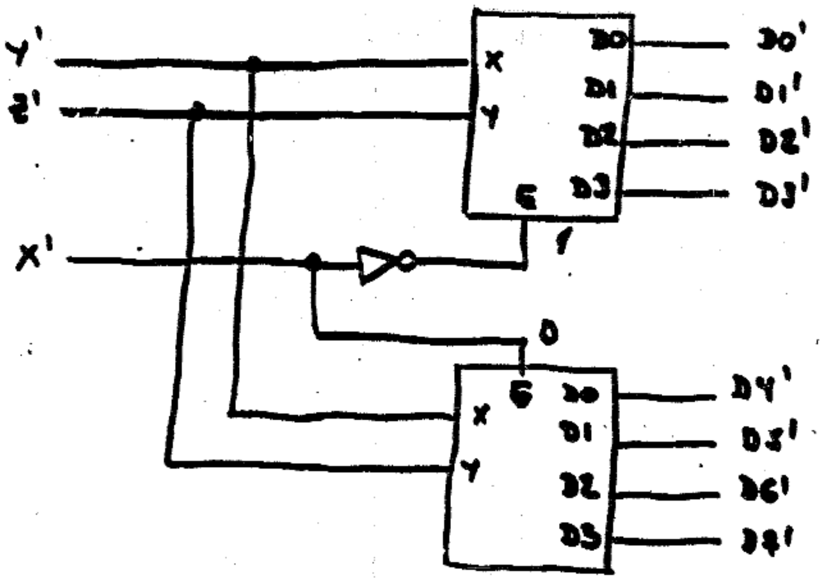
. SÍMBOLO DO DECODIFICADOR C/ LIBERAÇÃO



. SÍMBOLO DO DEMULTIPLEXADOR

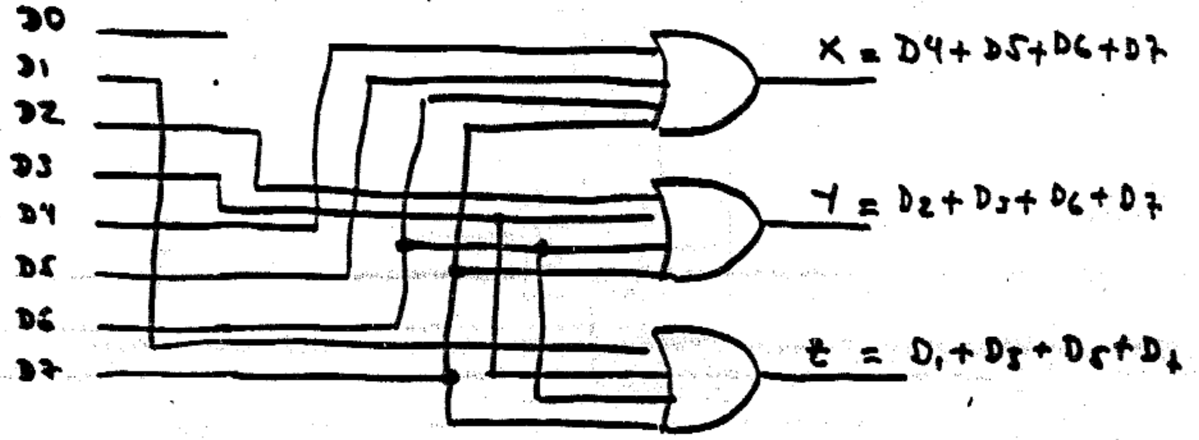


Exercício: Use decodificadores 2x4 c/ liberação para implementar decodificador 3x8.



CODIFICADORES.

- PROJETE UM CIRCUITO QUE INFORME EM 3 BITS QUAL DAS 8 LINHAS DE ENTRADA FOI ATIVADA. ASSUMA QUE SOMENTE UMA DAS LINHAS PODE ESTAR EM UM A CADA INSTANTE.

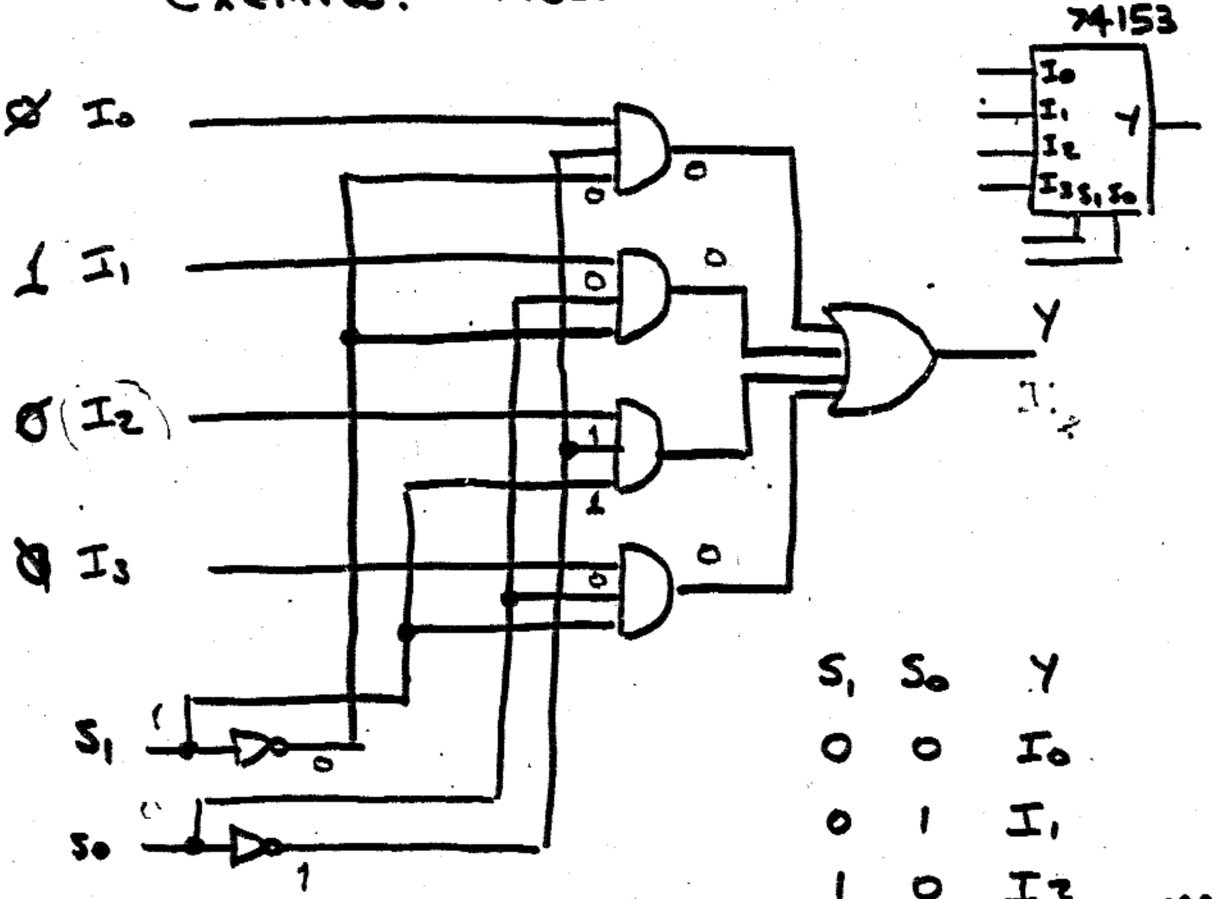


MSI.4 MULTIPLEXADORES.

MULTIPLEXADOR É UM CIRCUITO COMBINACIONAL USADO PARA SELECIONAR UMA ENTRE VÁRIAS LINHAS DE DADOS. A ESCOLHA DA LINHA DE DADO É FEITA POR UM CONJUNTO DE LINHAS DE SELEÇÃO.

PARA ESCOLHER ENTRE 2^N ENTRADAS USAMOS N LINHAS DE SELEÇÃO.

EXEMPLO: MULTIPLEXADOR 4x1.



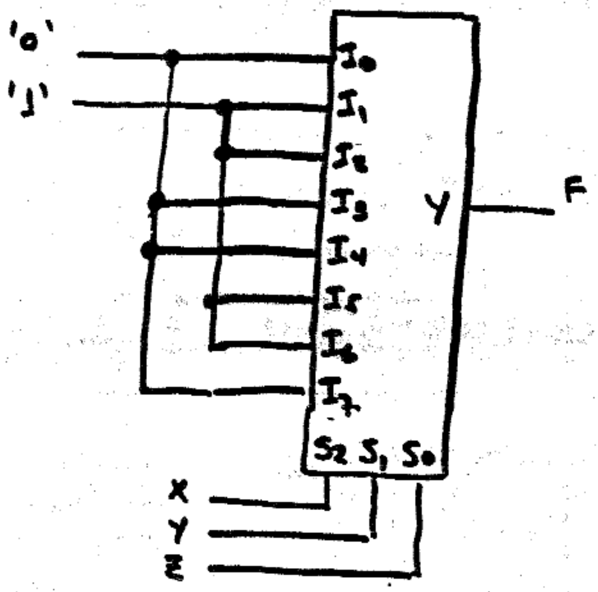
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

MSI.9.1

Implementação de Funções Booleanas com Multiplexadores.

a) Solução Trivial

Implementar $F(x,y,z) = \sum 1,2,5,6$

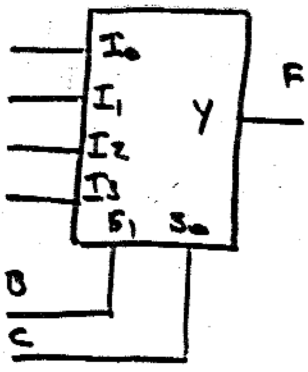


Solução Econômica

- Função de N+1 variáveis use $2^N \times 1$ Multiplexador.
- Conecte N variáveis às linhas de seleção do multiplexador.
- A variável restante será usada nas entradas.

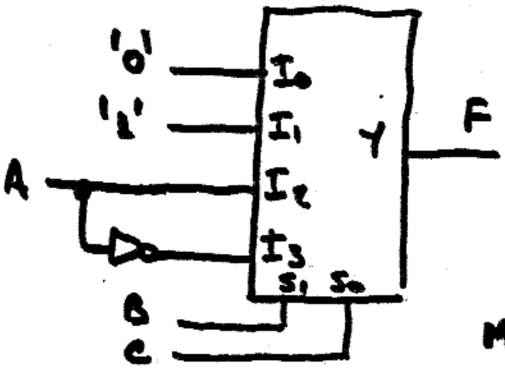
Exemplo. Função $F(A,B,C) = \sum 1,3,5,6$

3 variáveis $\Rightarrow$ Mux $2^2 \times 1$ $\Rightarrow$ Mux 4×1



A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

	I_0	I_1	I_2	I_3
A'	0	1	2	3
A	4	5	6	7
	0	1	A	$\bar{A}$



REGRAS P/OBTER AS ENTRADAS DO MUX

- SE 2 MINTERMOS EM UMA COLUNA NÃO ESTÃO EM CÍRCULOS, COLOQUE '0' NA ENTRADA CORRESPONDENTE.
- SE 2 MINTERMOS EM UMA COLUNA ESTÃO EM CÍRCULOS, COLOQUE '1' NA ENTRADA CORRESPONDENTE.
- SE SOMENTE O MINTERMO SUPERIOR ESTÁ EM CÍRCULO, COLOQUE '$\bar{A}$' NA ENTRADA CORRESPONDENTE
- SE SOMENTE O MINTERMO INFERIOR ESTÁ EM CÍRCULO, COLOQUE 'A' NA ENTRADA CORRESPONDENTE

EXEMPLO. IMPLEMENTE A FUNÇÃO $F(A,B,C,D) = \sum 0, 1, 5, 6, 9, 13, 15$, USANDO MULTIPLEXADORES.

3º CASO. $A < B$

SE

$(A_3 < B_3)$ OU

$(A_3 = B_3)$ E $(A_2 < B_2)$ OU

$(A_3 = B_3)$ E $(A_2 = B_2)$ E $(A_1 < B_1)$ OU

$(A_3 = B_3)$ E $(A_2 = B_2)$ E $(A_1 = B_1)$ E $(A_0 < B_0)$

$$A<B = \overline{A_3} B_3 + x_3 \overline{A_2} B_2 + x_3 x_2 \overline{A_1} B_1 + x_3 x_2 x_1 \overline{A_0} B_0$$

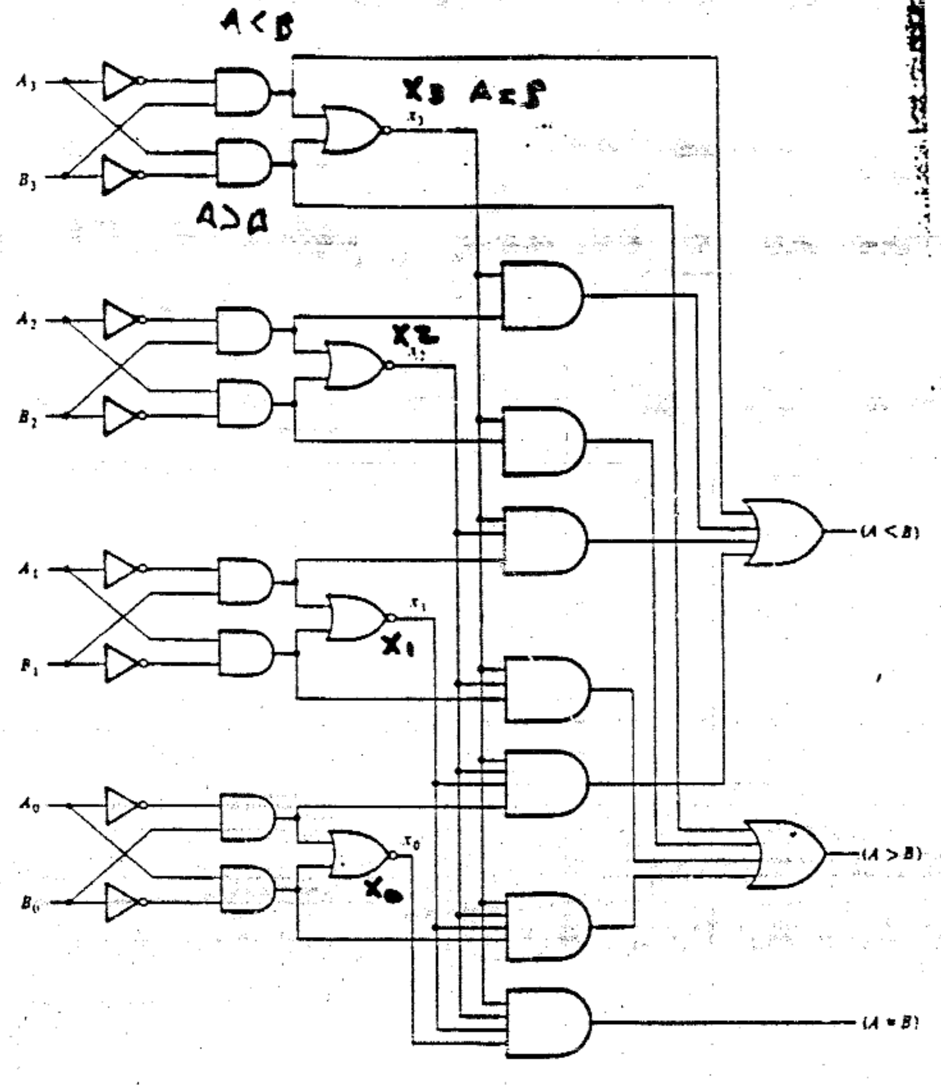
PARA O CIRCUITO $A > B$ NECESSITAMOS DE $A_i \overline{B_i}$ E PARA $A < B$ DE $\overline{A_i} B_i$.

PARA $A = B$ TEMOS

$$x_i = A_i B_i + \overline{A_i}\,\overline{B_i} = \overline{A_i \oplus B_i} = \overline{A_i \overline{B_i} + \overline{A_i} B_i}$$

DESTE MODO O CIRCUITO PODE SER CONSTRUÍDO A PARTIR DE

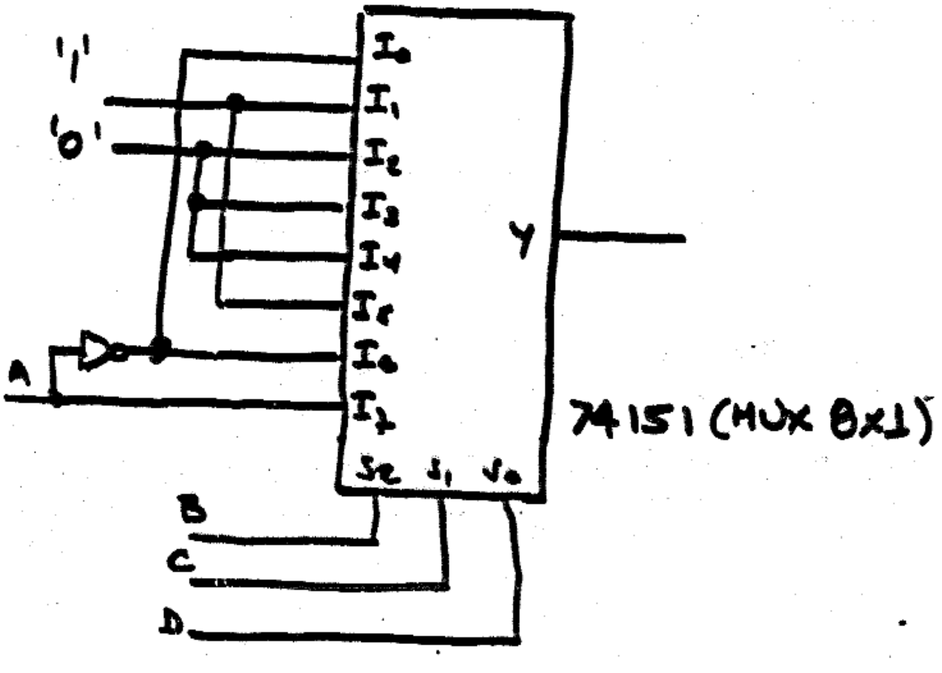
$A_i \overline{B_i}$ E $\overline{A_i} B_i$



CIRCUITO COMPARADOR

SOLUÇÃO:

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	$\bar{A}$	1	0	0	0	1	$\bar{A}$	A



MSI.9

Comparador de Magnitude

Circuito que compara dois números binários A e B, e determina se $A > B$, $A = B$ ou $A <$

$A = A_3 A_2 A_1 A_0$; $B = B_3 B_2 B_1 B_0$

1º Caso $A = B$

$A = B$ se $A_3 = B_3$ e $A_2 = B_2$ e $A_1 = B_1$ e $A_0 = B_0$

$A_i = B_i \Rightarrow x_i = A_i B_i + \overline{A_i}\,\overline{B_i} =$

$(A = B) = x_3 \cdot x_2 \cdot x_1 \cdot x_0$

2º Caso $A > B$ se

$(A_3 > B_3)$ ou

$(A_3 = B_3)$ e $(A_2 > B_2)$ ou

$(A_3 = B_3)$ e $(A_2 = B_2)$ e $(A_1 > B_1)$ ou

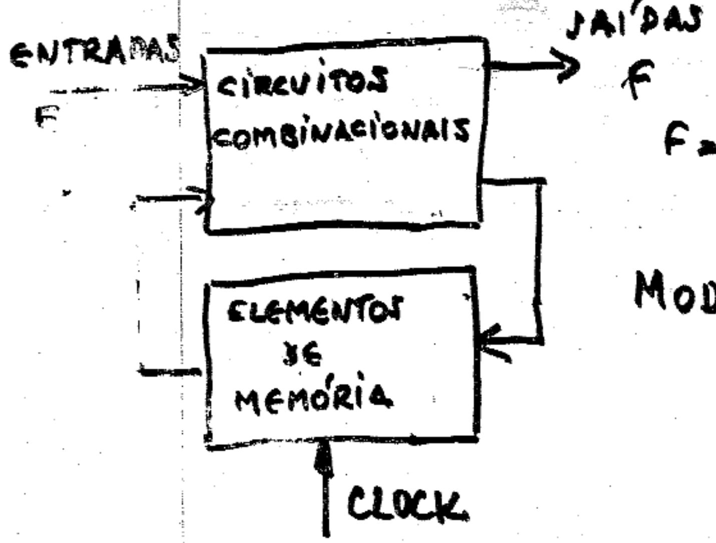
$(A_3 = B_3)$ e $(A_2 = B_2)$ e $(A_1 = B_1)$ e $(A_0 > B_0)$

$A_3 \overline{B_3} + x_3 A_2 \overline{B_2} + x_3 x_2 A_1 \overline{B_1} + x_3 x_2 x_1 A_0 \overline{B_0} = A > B$

Circuitos Sequenciais

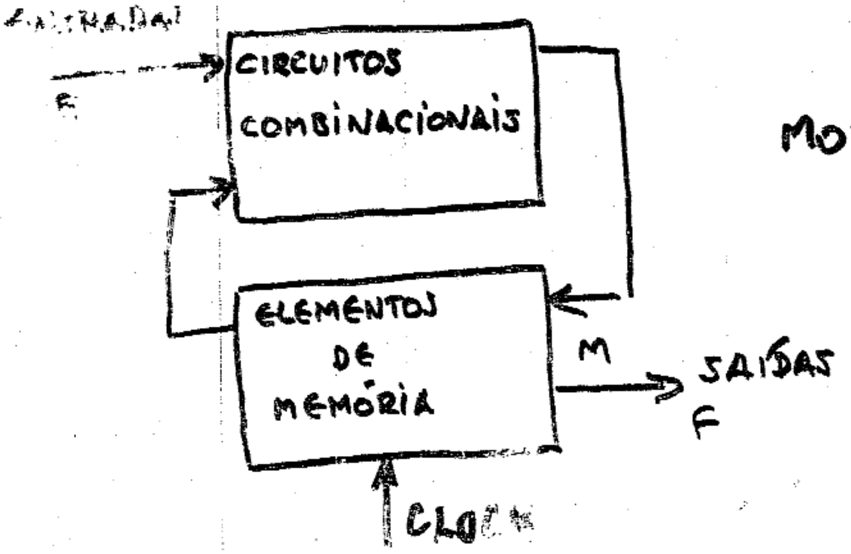
CS.1 Introdução

Circuitos combinacionais com elementos de memória.



$F = f(E, M)$

MODELO MEALY



MODELO MOORE

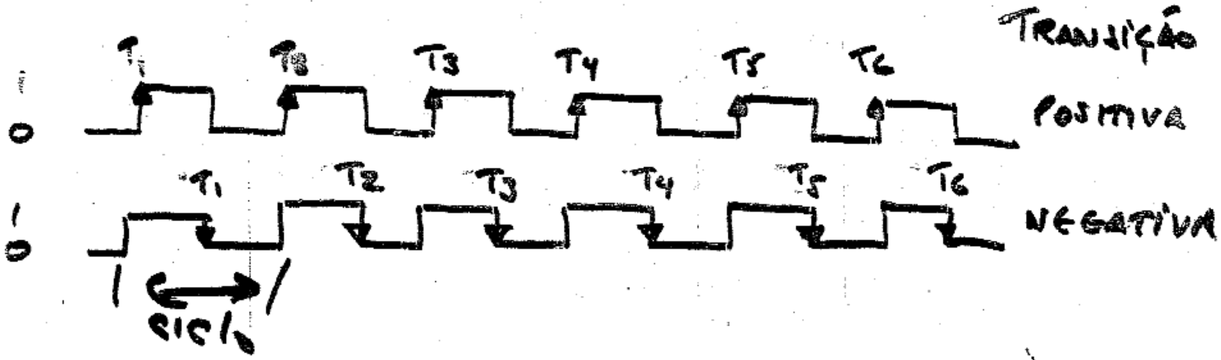
$F = f(M)$

/ CS.

CIRCUITOS SEQUENCIAIS <u>SÍNCRONOS</u>: SÃO CIRCUITOS CUJO COMPORTAMENTO PODE SER DEFINIDO A PARTIR DO CONHECIMENTO DO VALOR DE SEUS SINAIS EM DETERMINADOS INSTANTES DE TEMPO.

CIRUITOS SEQUENCIAIS <u>ASSÍNCRONOS</u>: O COMPORTAMENTO DO CIRCUITO ASSÍNCRONO DEPENDE DA ORDEM EM QUE OS SINAIS DE ENTRADA MUDAM, E PODE SER INFLUENCIADO A QUALQUER MOMENTO.

OS INSTANTES DE TEMPO EM QUE OS CIRCUITOS SÍNCRONOS MUDAM SÃO MARCADOS POR UM GERADOR DE TEMPOS (RELÓGIO, "CLOCK GENERATOR").

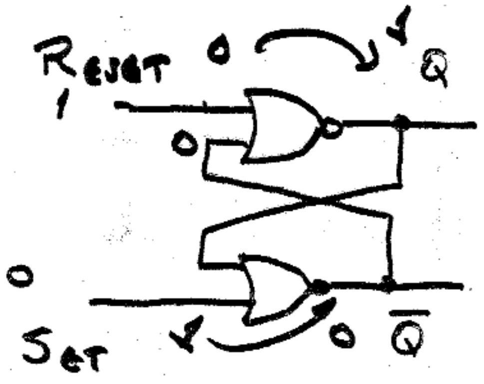


5.2 FLIP-FLOPS

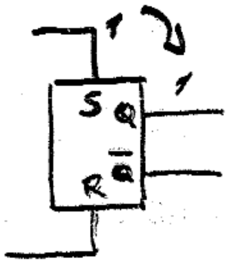
Os elementos de memória usados nos circuitos sequenciais são chamados flip-flops. Estes circuitos mantém um estado binário (1 ou Ø) enquanto não receber comando para mudar.

Existem vários tipos de flip-flops.

– Flip-Flop Set-Reset.



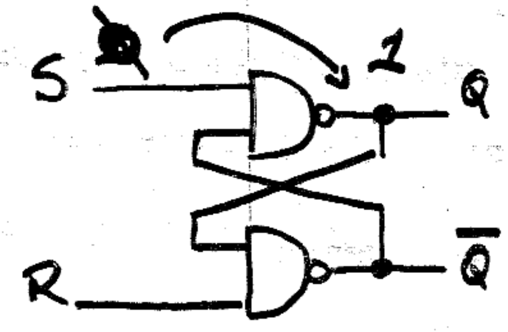
(NOR)



S	R	Q	$\overline{Q}$	
1	0	1	0	
0	0	1	0	(após S=1, R=0)
0	1	0	1	
0	0	0	1	(após S=0, R=1)
1	1	0	0	

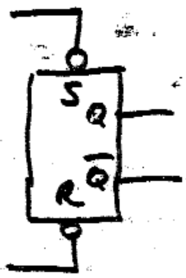
S	R	Q_{t+1}	$\overline{Q}_{t+1}$
0	0	Q_t	$\overline{Q}_t$
0	1	0	1
1	0	1	0
1	1	0	0

Set - Reset (NAND)



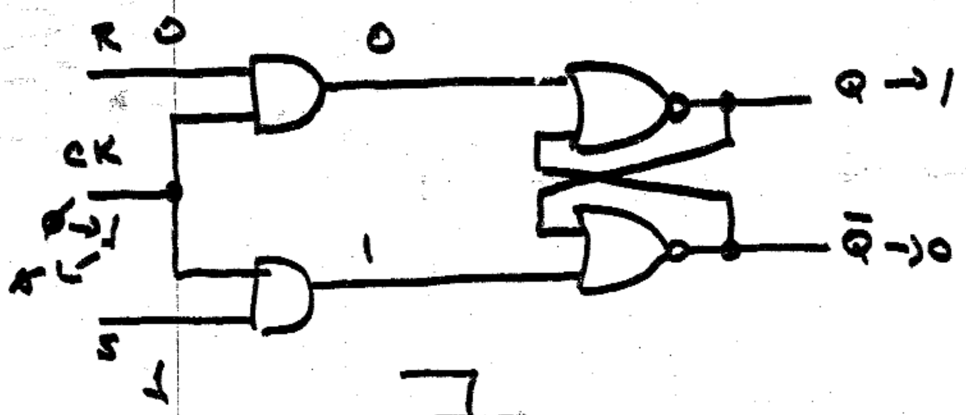
S	R	Q	$\overline{Q}$	
1	0	0	1	
1	1	0	1	- Após S=1 R=0
0	1	1	0	
1	1	1	0	- Após S=0 R=1
0	0	1	1	

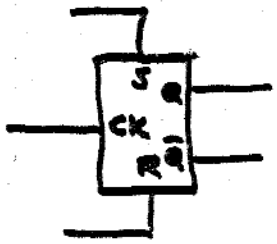
S	R	Q_{t+1}	$\overline{Q}_{t+1}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q_t	$\overline{Q_t}$



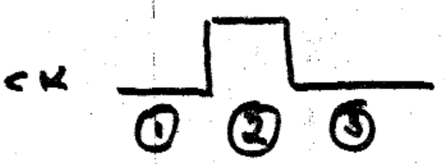
C54

FLIP-FLOP SR COM "CLOCK"





- FLIP-FLOP MUDA DE ESTADO QUANDO CK=1. CASO CK=0 AS SAÍDAS SÃO INSENSÍVEIS A MUDANÇAS EM S, R.



1. ESTADO ATUAL.
2. FLIP-FLOP MUDA DE ESTADO.
3. PRÓXIMO ESTADO (Q_{T+1}).

TABELA FF SR "CLOCKED"

Q_t	S	R	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	INDETERMINADO.
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	INDETERMINADO.

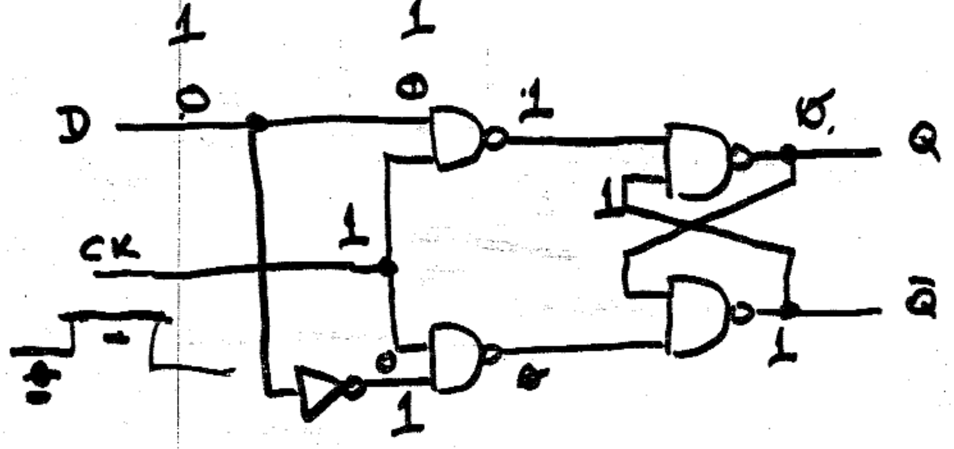
Q_{t+1}

Q_t \ SR	00	01	11	10
0			X	1
1	1		X	1

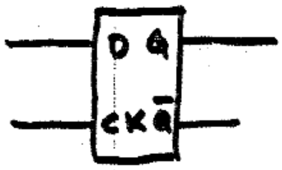
$$Q_{t+1} = S + Q_t \overline{R}$$

QUANDO S=R=1 E O SINAL DE CLOCK PASSA DE 1 PARA ϕ, O PRÓXIMO ESTADO DO FF NÃO PODE SER CALCULADO.

FLIP-FLOP TIPO D.



7474



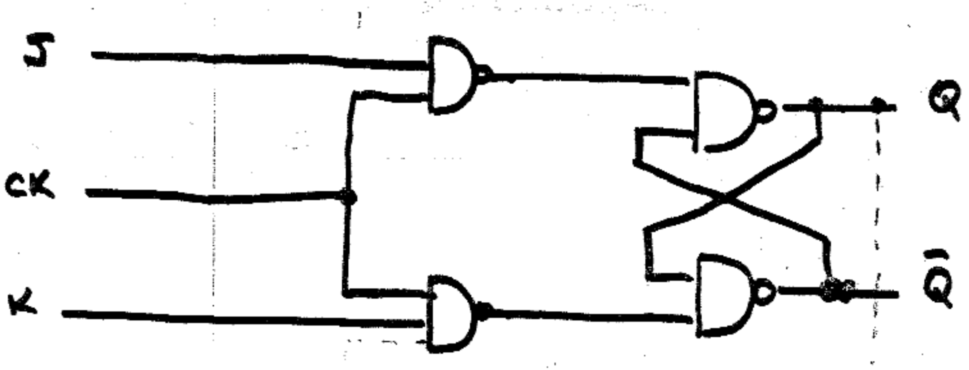
Q_t	D	Q_{t+1}
0	0	0
0	1	1
1	0	0
1	1	1

$Q_{t+1} = D$

APÓS O PULSO DE CLOCK A SAÍDA Q FICA COM O VALOR DA ENTRADA D.

FLIP-FLOP J-K

- MODIFICAÇÃO DO FF SR PARA ACABAR COM ESTADO INDETERMINADO.



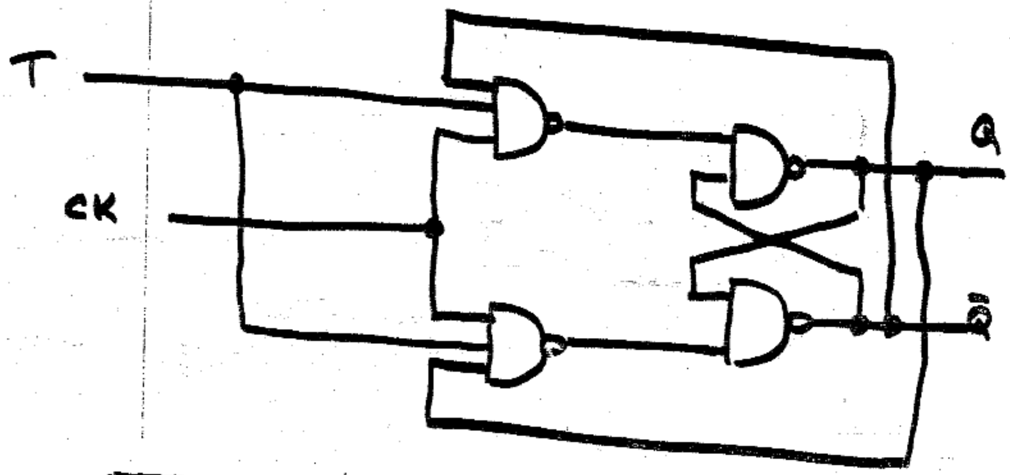
Q_t	J	K	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

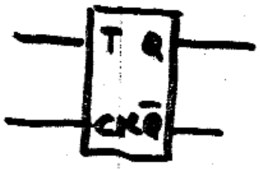
TAB SIMPLIFICAD

J	K	Q_t
0	0	Q_t
0	1	0
1	0	1
1	1	$\overline{Q_t}$

$$Q_{t+1} = J\overline{Q}_t + \overline{K}Q_t$$

FLIP-FLOP TIPO T (TOGGLE)





Q_t	T	Q_{T+1}
0	0	0
0	1	1
1	0	1
1	1	0

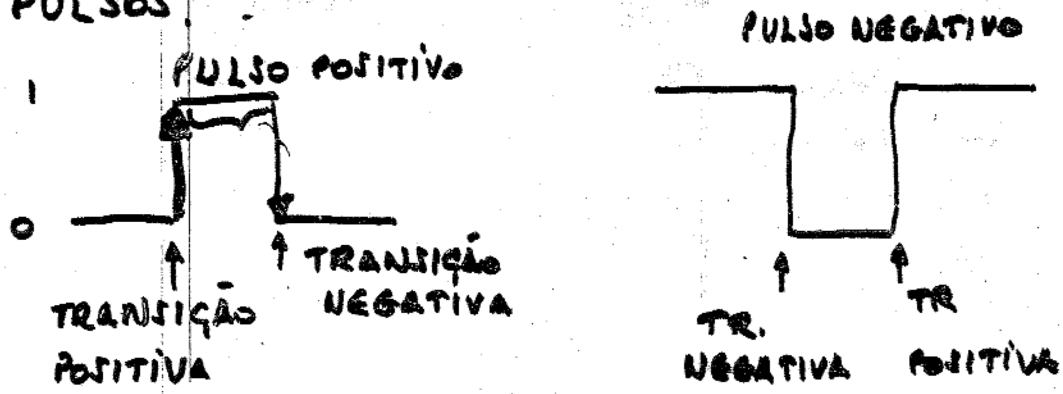
$$Q_{t+1} = \overline{Q_t}\,T + Q_t\,\overline{T}$$

– GATILHAMENTO DOS FLIP-FLOPS –

O ESTADO DOS FLIP-FLOPS É MODIFICADO POR MUDANÇAS NOS SINAIS DE ENTRADA.

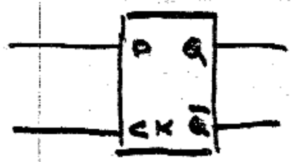
FLIP-FLOPS ASSÍNCRONOS COMO OS S-R MUDAM DE ESTADO CONTROLADOS POR ALTERAÇÕES NO NÍVEL DOS SINAIS.

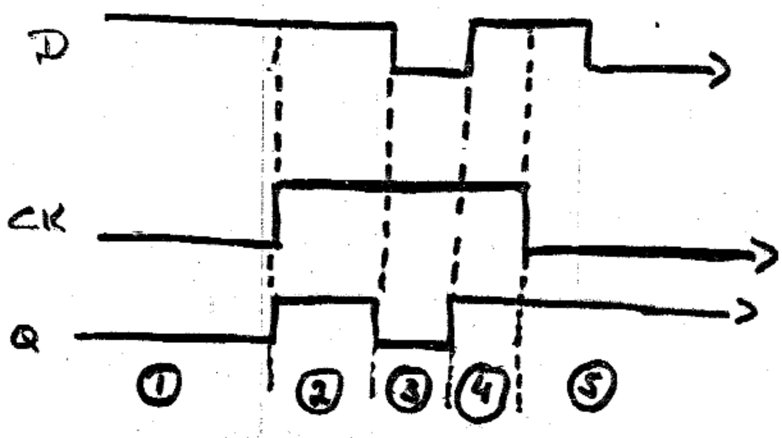
FLIP-FLOPS COM CLOCKS MUDAM POR PULSOS.



OS FLIP-FLOPS "POR NÍVEL" RESPONDEM AOS VALORES NA ENTRADA DE DADOS ENQUANTO O PULSO NA ENTRADA CLOCK ESTIVER EM 1. CONSIDERE FF. TIPO D.

OS FLIP-FLOPS SENSÍVEIS A NÍVEL SÃO COMUMENTE DENOMINADOS DE "LATCHES".

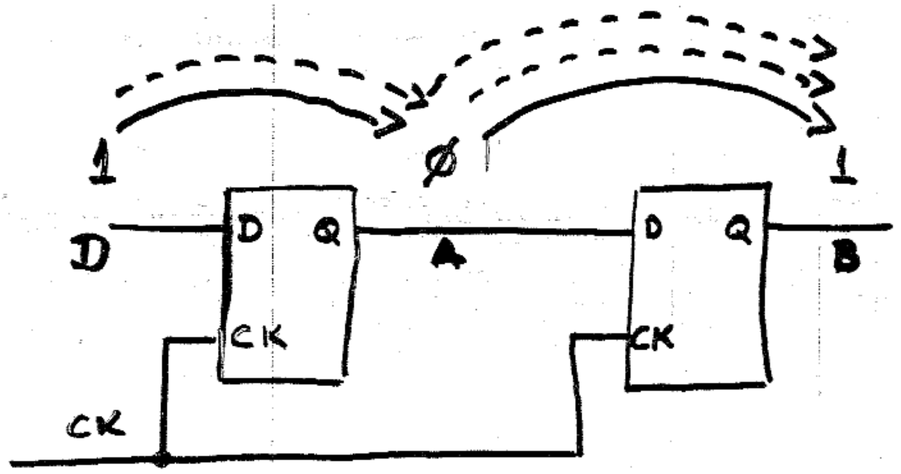




1. FF NÃO MUDA (CK = 0).
2. FF COPIA ENTRADA D EM 1
3. FF " " " " 0
4. FF " " " " 1
5. FF MANTÉM VALOR 1 APÓS CK = 0

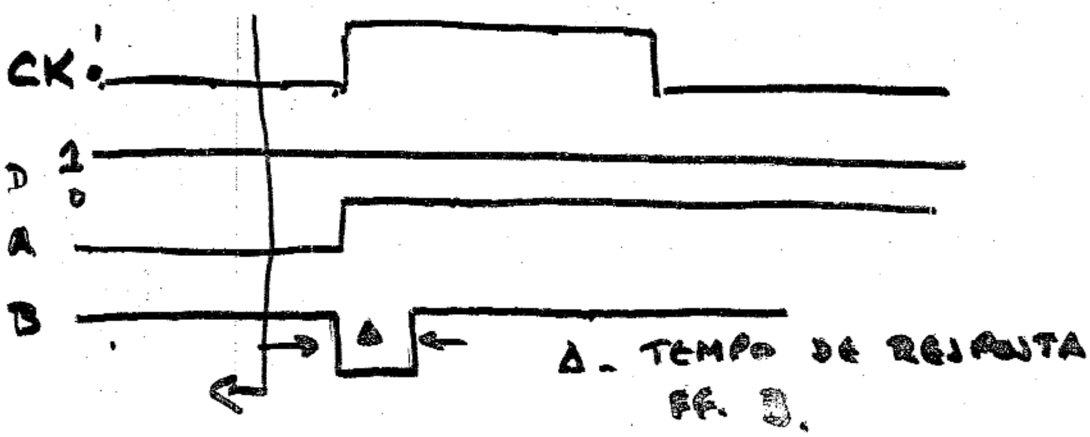
PROBLEMA:

NÃO É POSSÍVEL TRANSFERIR O CONTEÚDO DE UM FF PARA OUTRO.



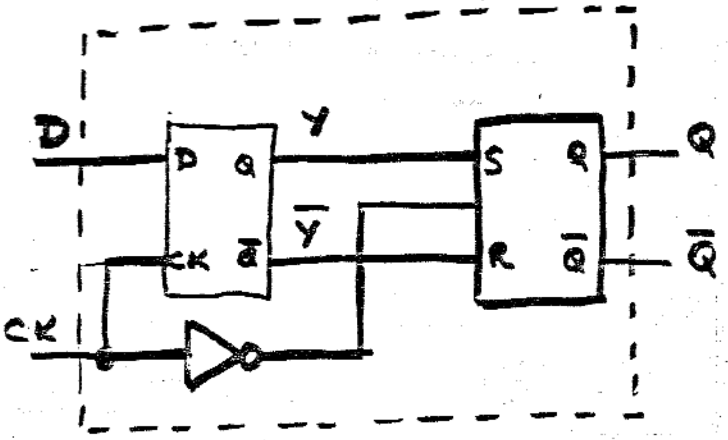
1. O QUE QUEREMOS ⟶

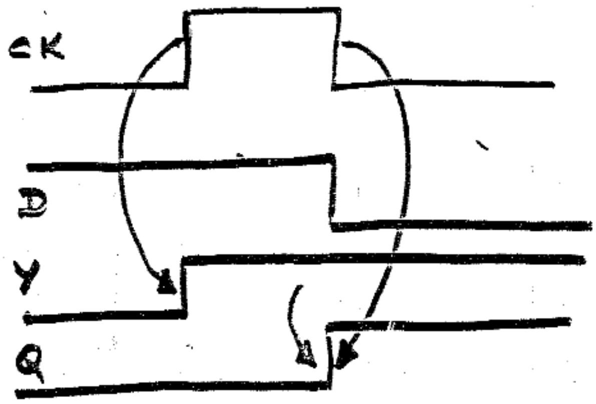
2. O QUE ACONTECE QUANDO O CK FICA EM '1' POR UM TEMPO.



SOLUÇÕES:

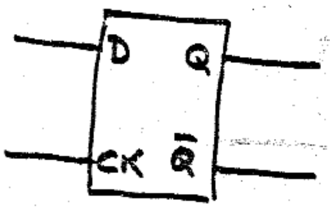
A) MASTER - SLAVE (MESTRE - ESCRAVO).

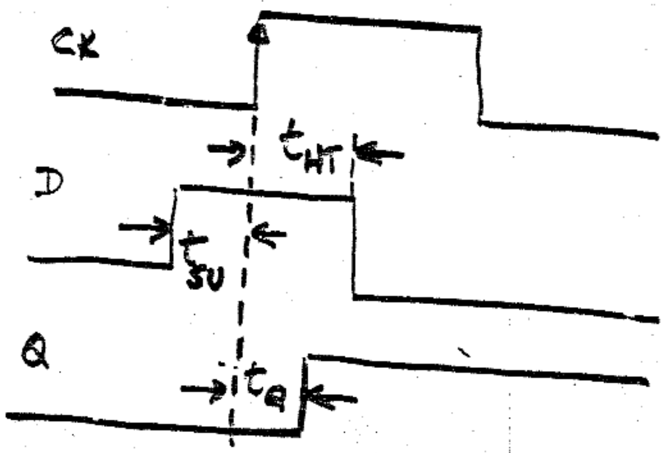




CS.13

B) FF POR TRANSIÇÃO (POSITIVA OU NEGATIVA)





T_Q - ATRASO NA SAÍDA

T_{SU} - (SET UP TIME) - TEMPO QUE A ENTRADA DEVE SER MANTIDA CONSTANTE ANTES DA TRANSIÇÃO DO CLOCK

T_{HT} (HOLD TIME) - TEMPO QUE A ENTRADA DEVE SER MANTIDA CONSTANTE APÓS A TRANSIÇÃO DE CLOCK.

CS.14