LISTA - CD

Resolvido LISTA 1 - 1 à 16, 18, 19
LISTA Complementar - 1 a 5

1-

0V = 00
1V = 01
2V = 10
3V = 11

Sequência ⇒ 0V 0V 1V 2V 3V 2V 1V
Sinal ⇒ 00 00 01 10 11 10 01

2- numero decimal 128

a) binário ⇒ 1 0 0 0 0 0 0 0 ⇒ 10000000
128 64 32 16 8 4 2 1

b) Hexadecimal ⇒ 80

16^2	16^1	16^0
0	8	0

128

c) base 3 ⇒ 11202

3^5	3^4	3^3	3^2	3^1	3^0
0	1	1	2	0	2

81 27 18 0 2 = 128

d) base 5 ⇒ 1003

5^4	5^3	5^2	5^1	5^0
0	1	0	0	3

125 3 = 128

e) base 15 ⇒ 88

15^2	15^1	15^0
0	8	8

120 8 = 128

3-

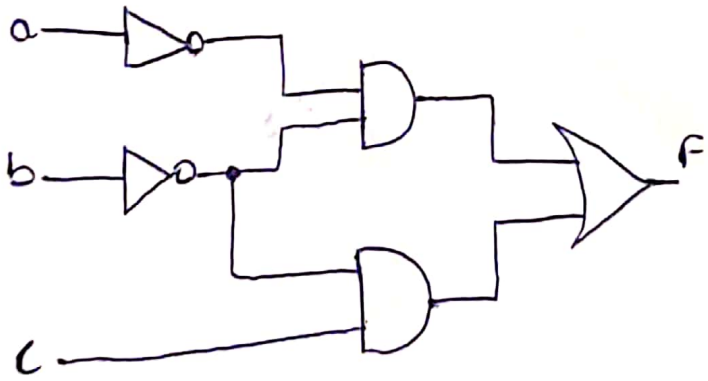
	Tarefa		
	A	B	C
Microprocessador	50ms	20ms	20ms
Circuito digital	1ms	4ms	1ms

• Tem que ser decodificado pelo menos 40 transações por segundo, ou seja terá uma decodificação a cada 0,025 segundos, isso 25ms
• Particionar as tarefas entre microprocessador e circuito digital
• Cada tarefa requer a mesma quantidade de I/O

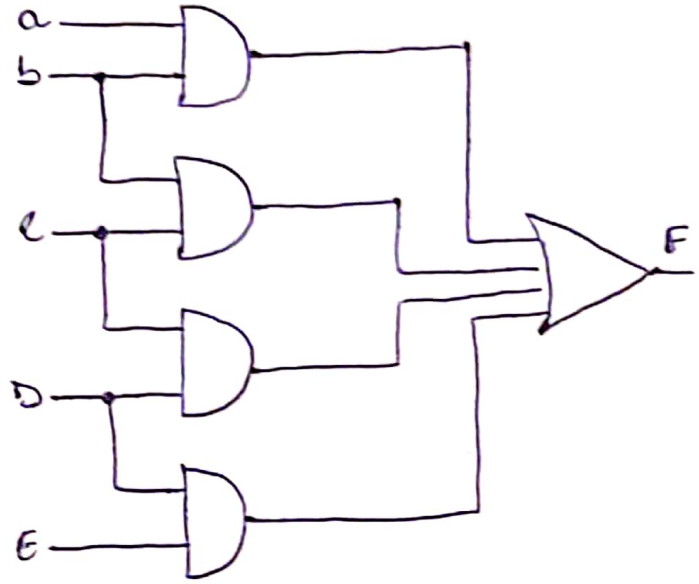
Dessa forma se tem duas soluções possíveis:

1? A e B como circuitos digitais, e C no microprocessador
2? A e C como circuitos digitais, e B no microprocessador

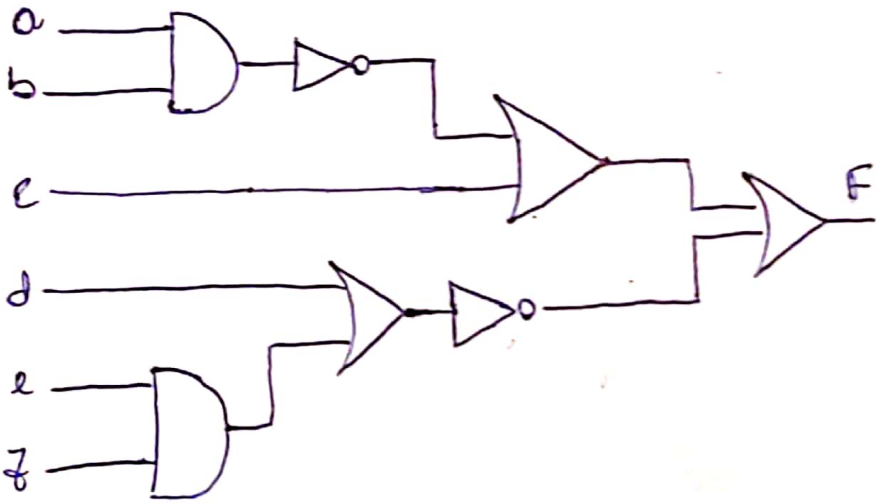
4- a) $F = a'b' + b'c$



b) $F = ab + bc + cd + de$



c) $F = ((ab)' + c) + (d + ef)'$



5. $F = abd' + acd$. Usar a Lei de Morgan

$F' = (abd' + acd)'$

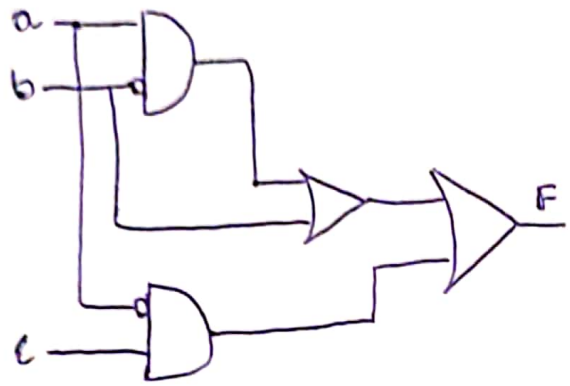
$F' = (abd')'(acd)' \Rightarrow F' = (a' + b' + d'')(a' + c' + d)' \Rightarrow$

$\Rightarrow F' = (a' + b' + d)(a' + c' + d') \Rightarrow F' = a'(a' + b' + d) + c'(a' + b' + d) + d'(a' + b' + d)$

$\Rightarrow F' = a' + a'b' + a'd + c'a' + c'b' + c'd + d'a' + d'b' + 0$ (Prop. complemento)

$\Rightarrow F' = a' + c'b + b'B'$ (Lei da absorção)

6.



$F = (ab' + b) + a'c$

Inputs					Out. Puts
a	b	c	ab' + b	a'c	F
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	0	1

7 - Converter em tabelas verdades

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

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

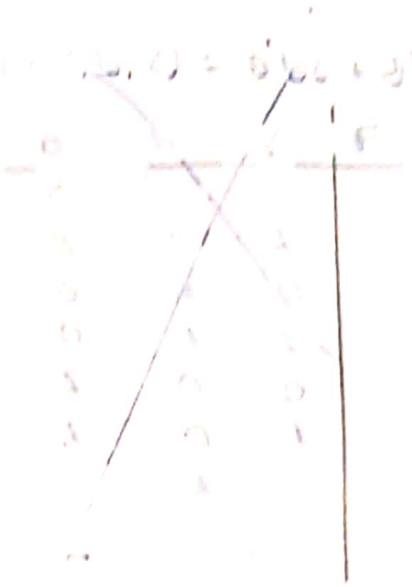
b) $F(a,b,c) = (ab)' + ac' + bc$

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

7.

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

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



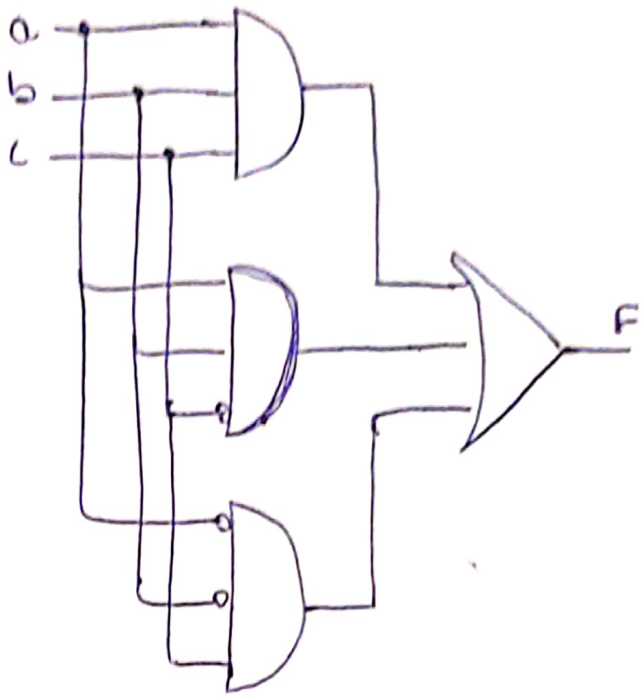
d) $F(a,b,c,d) = a'bc + d'$

a	b	c	d	F
0	0	0	1	1
0	0	0	0	1
0	0	1	1	1
0	0	1	0	0
0	1	0	1	1
0	1	0	0	1
0	1	1	1	1
0	1	1	0	1
1	0	0	1	0
1	0	0	0	1
1	0	1	1	0
1	0	1	0	1
1	1	0	1	0
1	1	0	0	1
1	1	1	1	0
1	1	1	0	1

8-

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

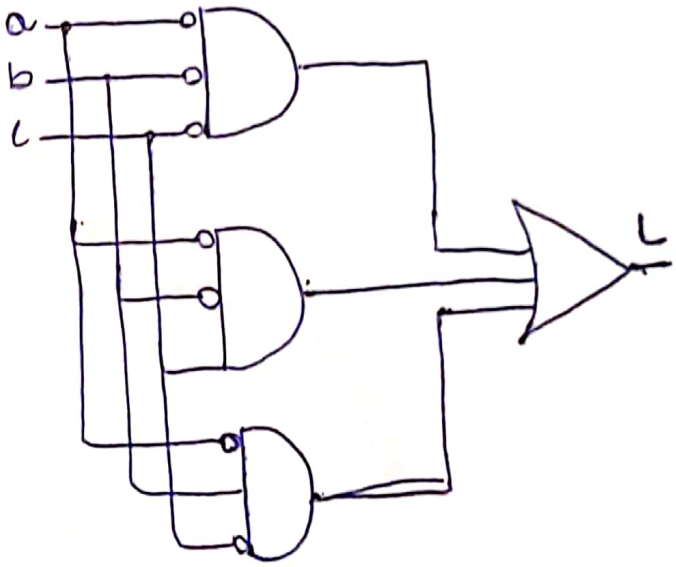
$F = a'b'c + ab c' + abc$



9-

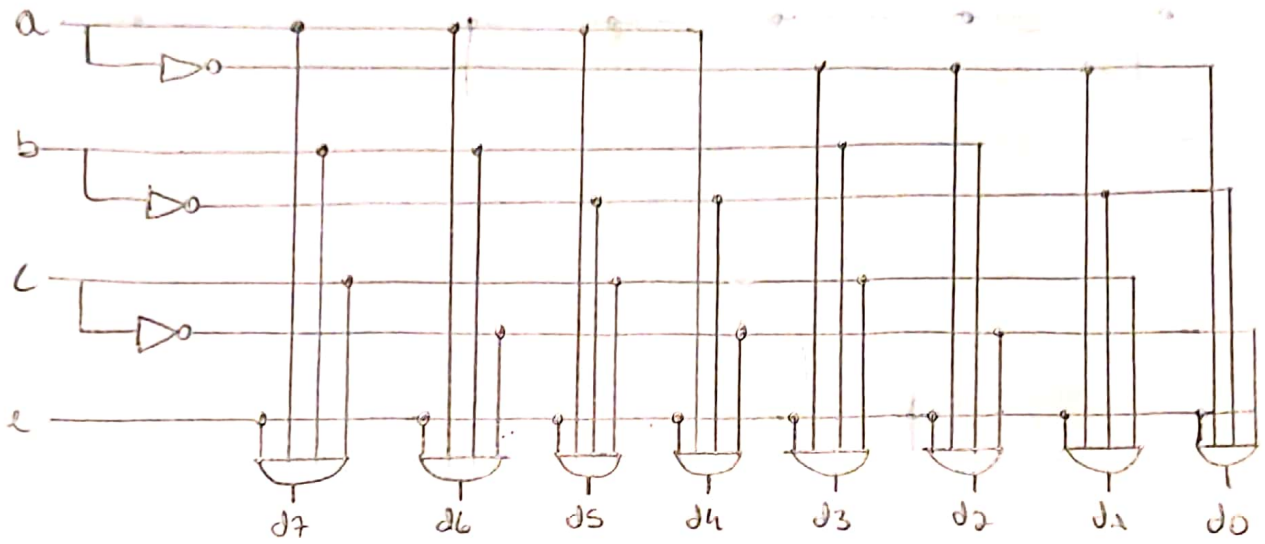
Inputs			outputs
a	b	c	L
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$L = a'b'c' + a'b'c + a'bc'$

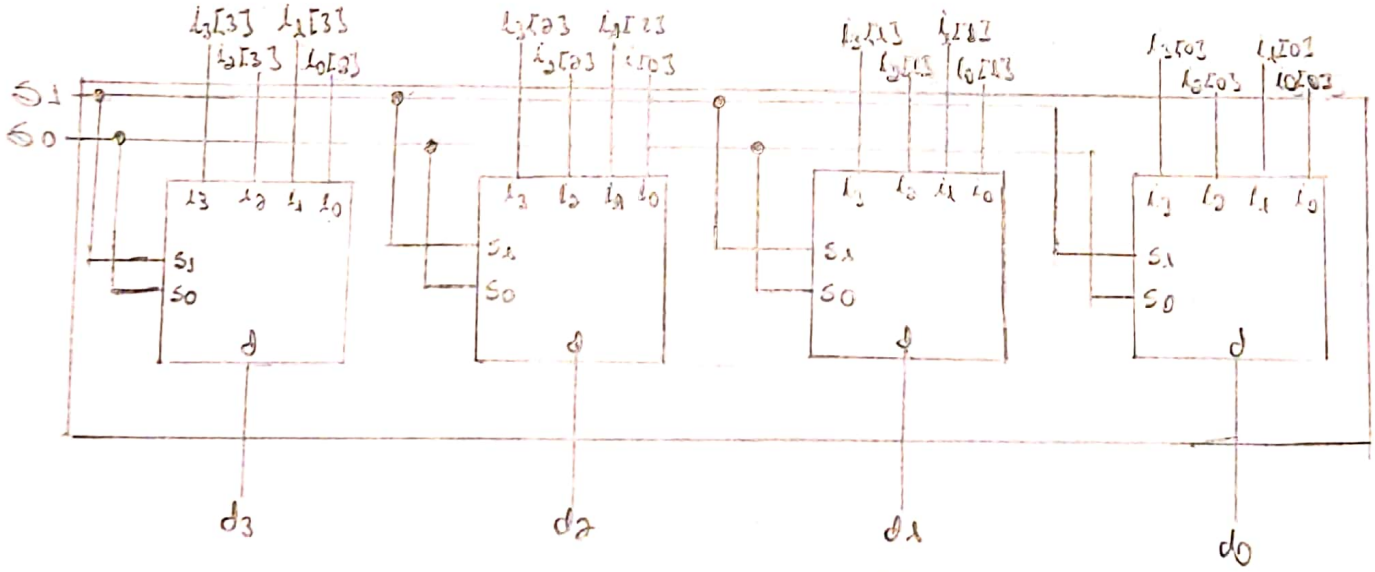


10- Decodificador 3x8 com Enable

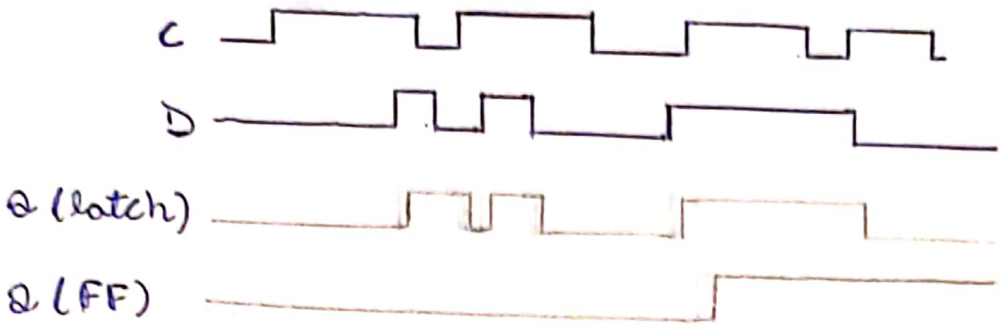
Slide 57



11-

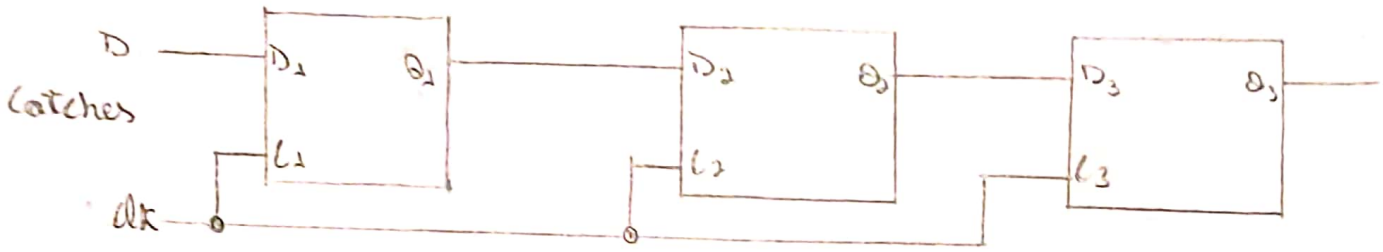


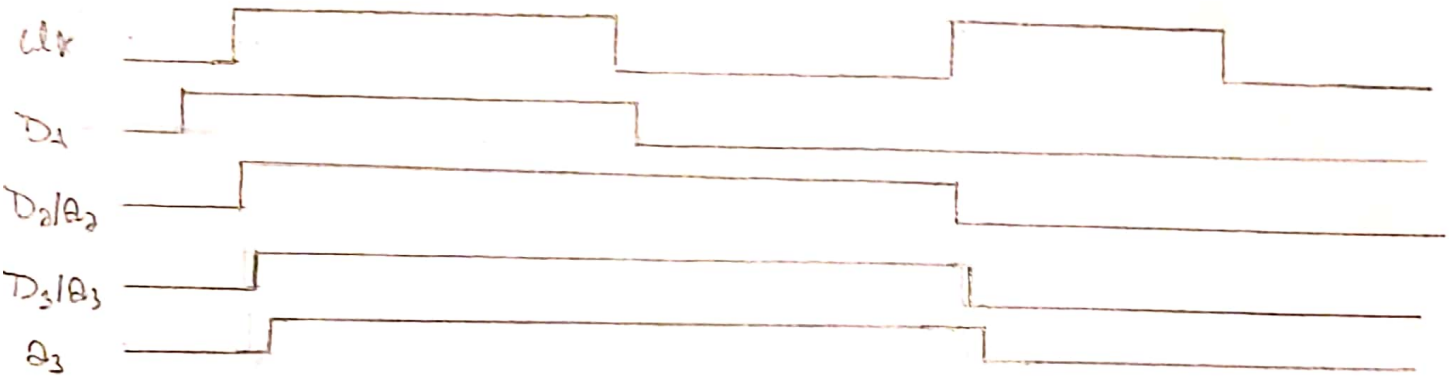
12-

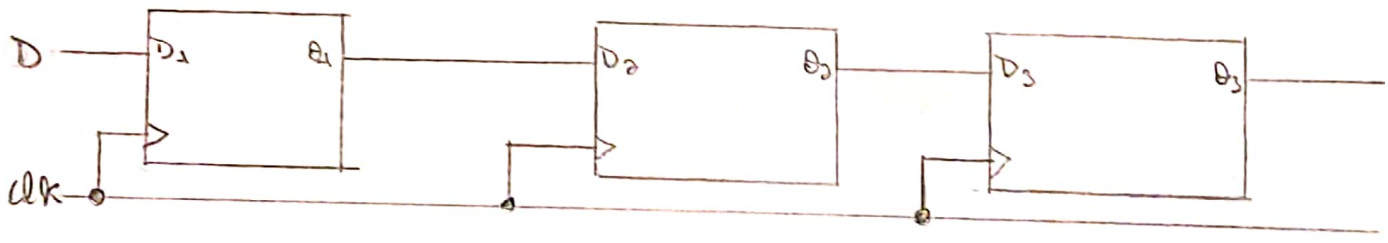


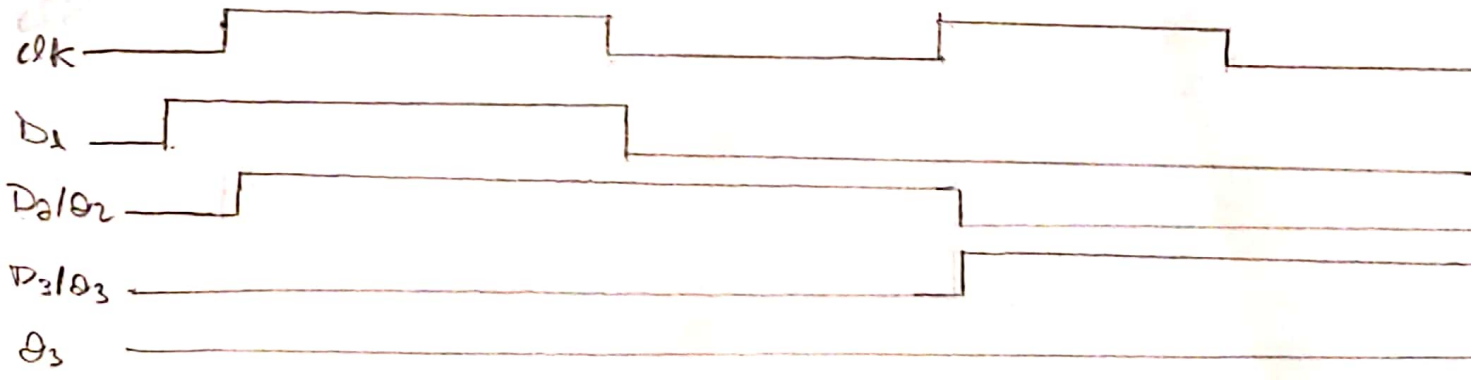
- latch tem o estado alterado durante o clock ativo, já o flip-flop só tem seu estado alterado apenas nos bordos do clock. ambos contendo um atraso.

13-

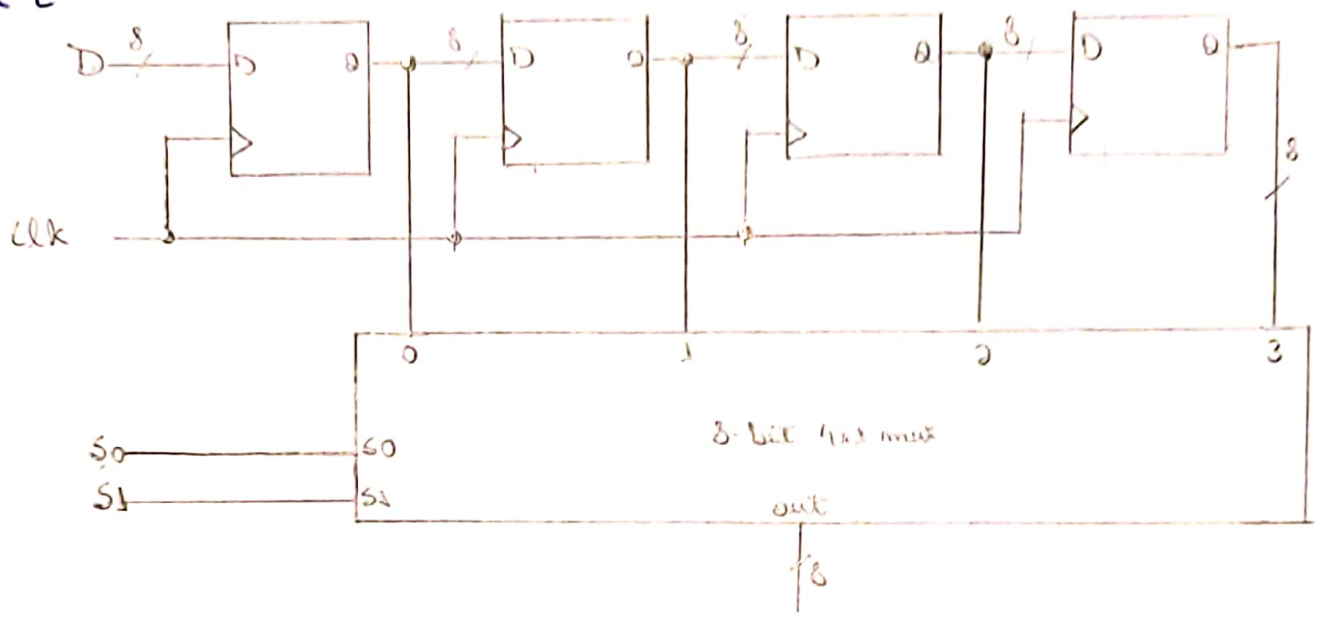




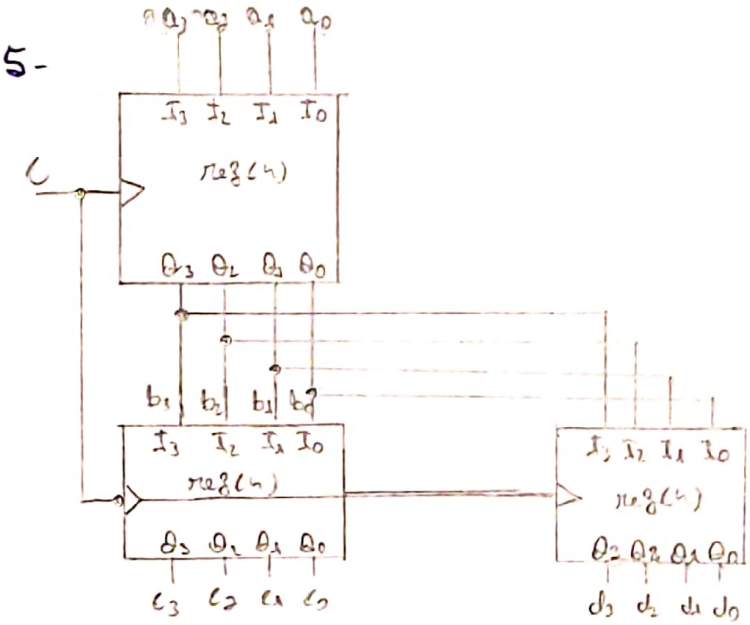


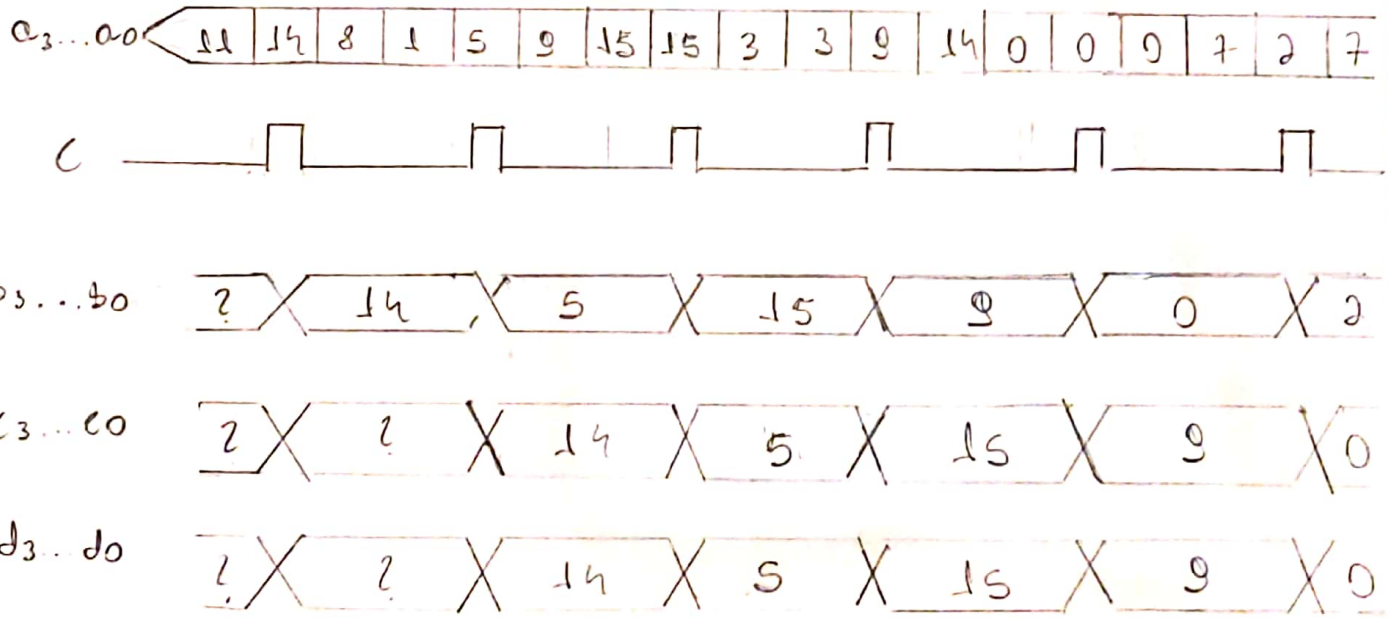


14-

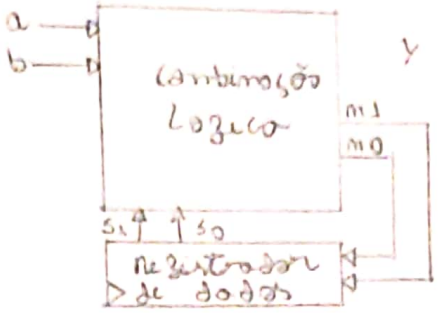


15-

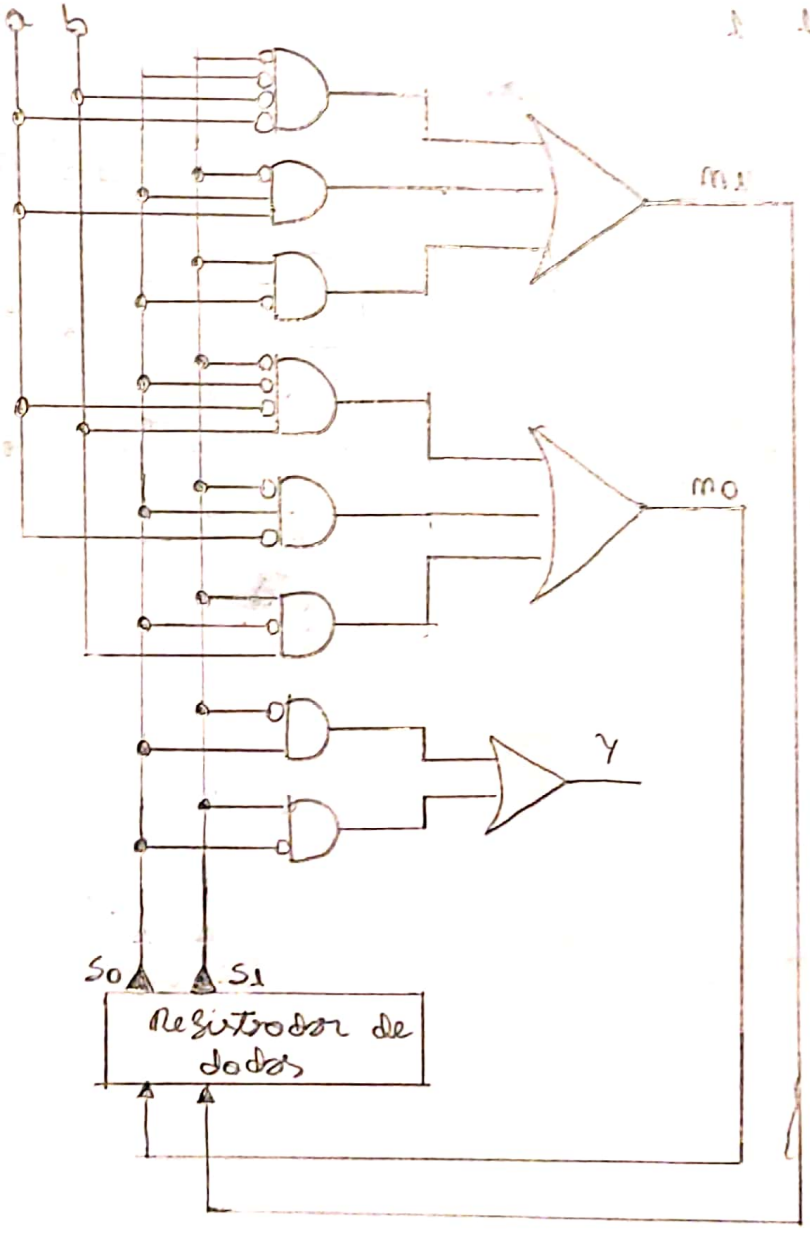




16 -



Inputs				outputs		
S_1	S_0	a	b	m_1	m_0	Y
0	0	0	0	1	0	0
0	0	0	1	0	1	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	1	0	1
1	0	0	1	1	1	1
1	0	1	0	1	0	1
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	0	1	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0



$m_1 = S_1'S_0'a'b' + S_1'S_0a + S_1S_0'$

$m_0 = S_1'S_0'a'b + S_1'S_0a' + S_1S_0'b$

$Y = S_1'S_0 + S_1S_0'$

18-

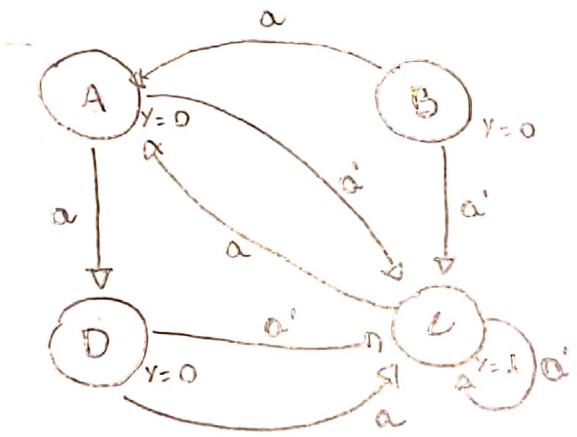
Equações: $m_1 = (S_1'S_0a + S_1S_0'a)'$

$m_0 = S_1'S_0'a$

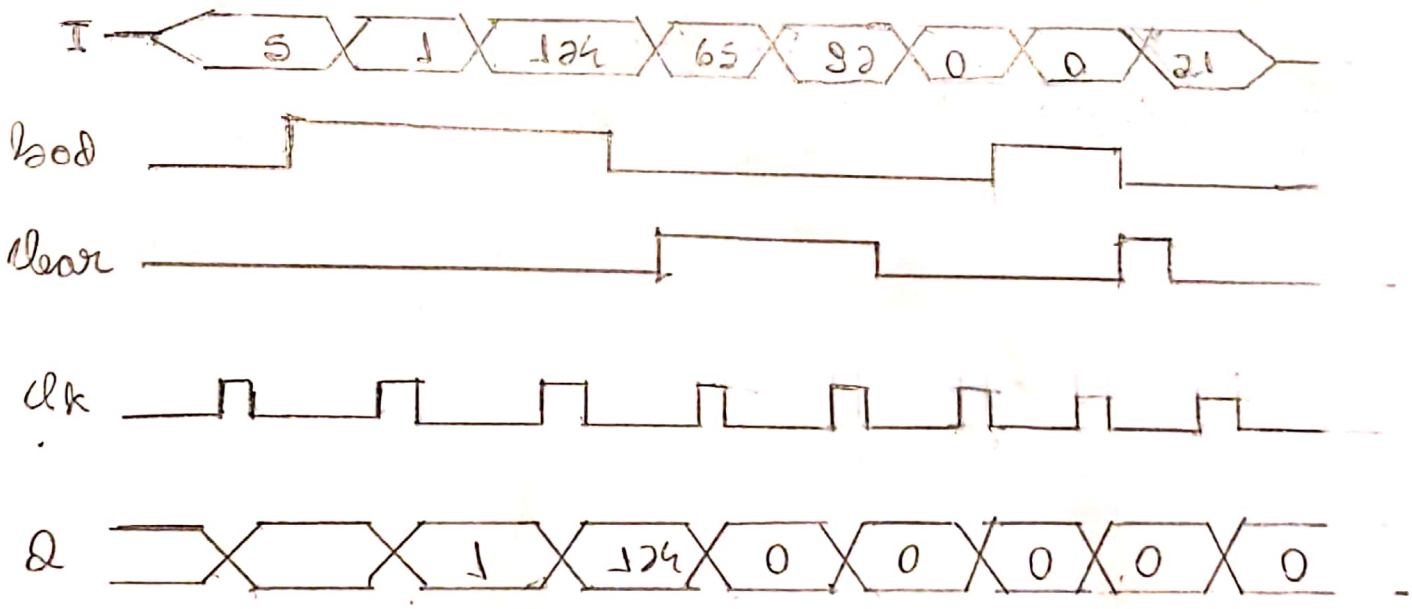
$Y = S_1S_0'$

Inputs			outputs		
S_1	S_0	a	m_1	m_0	Y
0	0	0	1	0	0
0	0	1	1	1	0
0	1	0	1	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	1	0	0

Codificações de estado: A = 00
B = 01
C = 10
D = 11



19-



Lista Extra

1-

00000000000 => 00

00000000001 => 01

11111111111 => 10

12 bits sem comprestão => 11

00000000000 00000000001 10000000000 11111111111

=> 00 01 11 10000000000 10

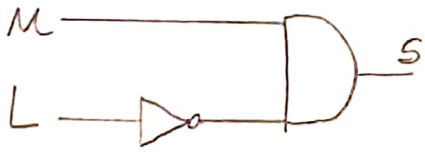
2- Os transitores são do tipo PMOS, ou seja, só conduz quando o valor lógico é 0. Dessa forma para que o transitor superior e inferior conduzam, x e y tem que assumir o valor lógico 0. Assim o circuito só conduz quando x e y possuem valor lógico 0.

3-

- M indica um movimento
- L indica se há luz do dia
- A sirene deve receber 1 para disparar o som do alarme.

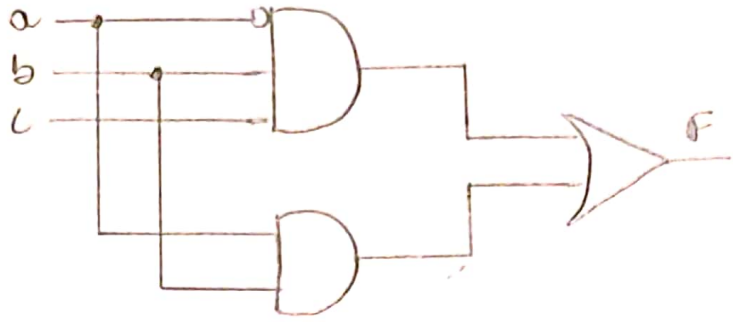
A sirene dispara quando houver movimento e não luz

$S = ML'$

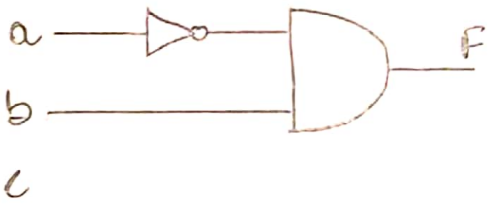


4-

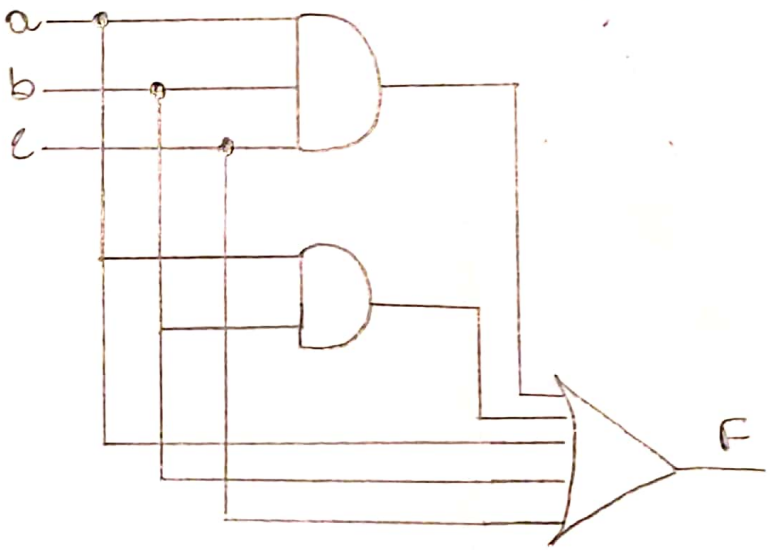
a) $F(a,b,c) = a'bc + ab$



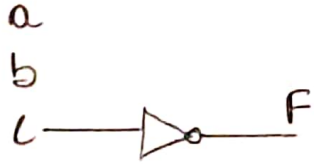
b) $F(a,b,c) = a'b$



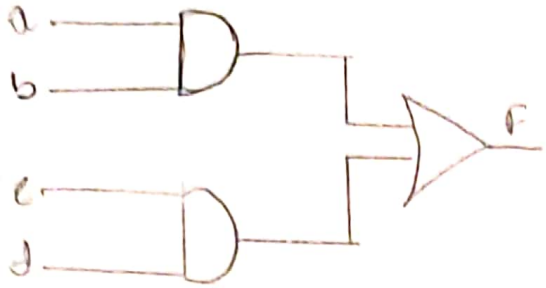
c) $F(a,b,c) = abc + ab + a + b + c$

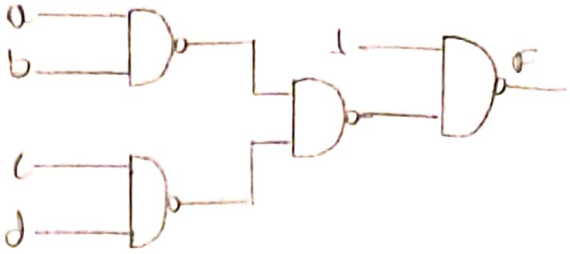


d) $F(a,b,c) = c'$



5.





$F_1(a,b,c,d) = ab + cd$ $\qquad$ $F_2(a,b,c,d) = (1((ab)'(cd)')')'$

- Simplificando $F_2(a,b,c,d) = (1((ab)'(cd)')')'$

$1' + ((AB)'(cd)')')'$ Teorema de Morgan

$1' + (AB)'(cd)'$ Lei da involução

$1' + (A'+B')(c'+D')$ Teorema de Morgan

$0 + (A'+B')(c'+D') \Rightarrow (A'+B')(c'+D')$ Lei do Identidade

$(A'c' + A'D' + B'c' + B'D')$ Destribuindo

$F_1(a,b,c,d) = ab + cd$; $F_2(a,b,c,d) = (1((ab)'(cd)')')'$

- Os dois circuitos não são equivalentes

Inputs				Output
a	b	c	d	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Inputs				Output
a	b	c	d	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
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	0