

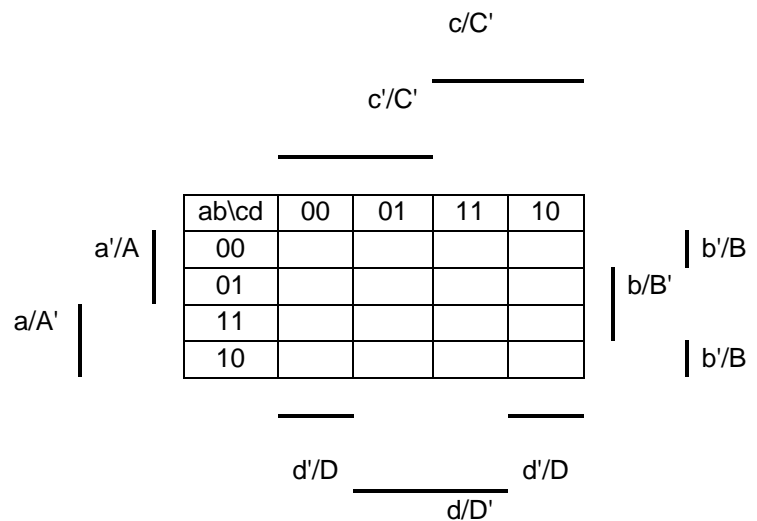
ARQ1 - Preparação 01

Tema: Sistemas de Numeração e circuitos combinatórios

Exercícios:

01.) Dada tabela-verdade e o mapa de Veitch-Karnaugh abaixo:

n	m	M	f(a,b,c,d)
0	a'b'c'd'	A+B+C+D	0
1	a'b'c'd	A+B+C+D'	0
2	a'b'c d'	A+B+C'+D	1
3	a'b'c d	A+B+C'+D'	0
4	a'b c'd'	A+B'+C+D	0
5	a'b c'd	A+B'+C+D'	1
6	a'b c d'	A+B'+C'+D	1
7	a'b c d	A+B'+C'+D'	1
8	a b'c'd'	A'+B+C+D	0
9	a b'c'd	A'+B+C+D'	0
A	a b'c d'	A'+B+C'+D	0
B	a b'c d	A'+B+C'+D'	0
C	a b c'd'	A'+B'+C+D	1
D	a b c'd	A'+B'+C+D'	1
E	a b c d'	A'+B'+C'+D	0
F	a b c d	A'+B'+C'+D'	1



Marcar os termos:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
mintermos																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MAXTERMOS																

Determinar e implementar os circuitos equivalentes em Verilog e no Logisim:

- expressão canônica para SoP(a,b,c,d)
- expressão canônica para PoS(A,B,C,D)
- simplificação de mintermos pelo mapa de Veitch-Karnaugh
- simplificação de MAXTERMOS pelo mapa de Veitch-Karnaugh
- expressão SoP equivalente com portas NAND (usar dupla negação)
- expressão PoS equivalente com portas NOR (usar dupla negação)

- 02.) Implementar e testar a função lógica descrita em Verilog e no Logisim para obter a tabela-verdade:

```
module f ( output s, input x, input y );  
  wire w1, w2, w3, w4, w5;  
  not NOT_1 (w1, x);  
  not NOT_2 (w2, y);  
  and OR__1 (w3, y, w2);  
  or OR__2 (w4, w1, x);  
  not NOT_3 (w5, w4);  
  and AND_1 (s, w3, w5);  
endmodule // s = f (x,y)
```

- 03.) Implementar no Verilog e no Logisim a expressão abaixo para obter a tabela verdade:

DICA: Usar MUX(a,b,chave). Se chave=0, (a); senão, (b).

MUX (MUX(a,b,c), MUX(NOT(b),NOT(a),c), NOT(c))

- 04.) Dado o valor negativo, já em complemento de 2, $2310_{(4)}$, com 01 byte de representação, encontrar a soma binária desse valor com $67_{(8)}$. Completar com zeros, se necessário.

- 05.) Implementar no Verilog e no Logisim a expressão abaixo para obter a expressão por mintermos equivalente:

NAND (XNOR (a, b), NOR (NOT(a),NOT (b)))