

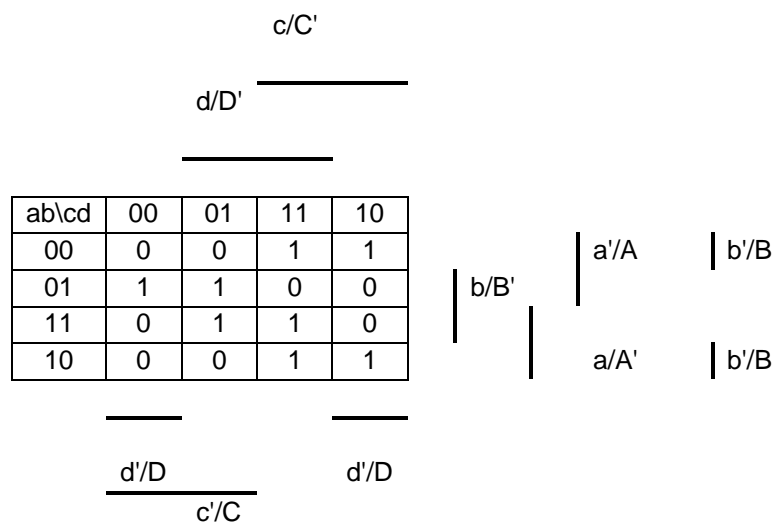
ARQ1 - Recuperação 01

Tema: Sistemas de Numeração e circuitos combinatórios

Exercícios:

01.) Dado o mapa de Veitch-Karnaugh:

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



	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
MAXTERMS																

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 por mapa de Veitch-Karnaugh
- simplificação de MAXTERMS por 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 a, input b, input c );  
  wire w1, w2, w3, w4;  
  not NOT_1 (w1, b);  
  not NOT_1 (w2, c);  
  and AND_1 (w3, a, w2);  
  and AND_2 (w4, a, w1, c);  
  or OR_1 (s, w3, w4);  
endmodule // s = f (a,b,c)
```

03. Implementar no Logisim as expressões abaixo para obter as tabelas verdade:

- a.) **nand (nand(nor(a,a), b), nand(nor(a,a), b))**
- b.) **mux (mux(b,b',c), mux(b',b,c'), a)**

04.) Dado o valor negativo, em complemento de 2, $357_{(8)}$, com um byte de representação

- a.) encontrar o positivo na base 2
- b.) representar em um byte 4 vezes o valor do binário equivalente
DICA: Desprezar os bits que ultrapassarem a representação. Completar com zeros.
- c.) representar em um byte o valor do binário equivalente dividido por 8
DICA: Desprezar os bits que ultrapassarem a representação. Completar com zeros.

05.) Calcular a diferença, em binário, entre o hexadecimal C4 e o quaternário 1232 em um byte.