

”

**E-fólio A** | Folha de resolução para E-fólio

**UNIDADE CURRICULAR:** Arquitetura de Computadores

**CÓDIGO:** 21010

**DOCENTE:** Gracinda Carvalho; José Coelho; Carlos Sousa; Rui Manuel Silva

**A preencher pelo estudante**

**NOME:** Andreia Isabel Teófilo Agostinho Romão

**N.º DE ESTUDANTE:** 1702430

**CURSO:** [2105] Licenciatura em Engenharia Informática

## TRABALHO / RESOLUÇÃO:

### Questão 1)

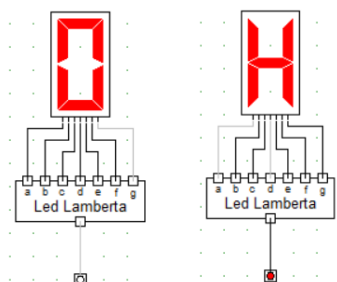
Como só temos, como valores de entrada, 0 ou 1, só precisamos de um bit.

Então para a criação do circuito sequencial que cumprisse os requisitos mencionados no enunciado, considerei só uma entrada (A), e 7 saídas (a, b, c, e, f e g). Ao elaborar a tabela de verdade, cheguei às seguintes funções lógicas correspondentes a cada segmento:

Entrada	Saídas						
A	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	1	1	1
Resultado	$\bar{A}$	1	1	$\bar{A}$	1	1	A

Teoricamente, na macro, o segmento c deveria de funcionar com o segmento ligado entre o “tag device” e a porta “not”, pois funcionou corretamente sem ser na macro, mas forcei com a porta “and” e assim garanto que o valor de entrada por uma porta “and” é lido antes da saída. E todos os segmentos que tem o valor “1” estão ligados ao Vcc.

Deixo a imagem do circuito funcional com a macro “Led Lamberta”, em anexo o logograma da macro.



### Questão 2)

Como a ideia era poder controlar quando se poderia alterar o valor do led/casa, de forma síncrona, e também poder guardar em memória o valor da casa, para quando o load fosse ativado haver alteração do led, decidi criar um registo, de duas entradas,  $Ld$  e  $P$ , e uma saída,  $Q$ , com o auxílio de um flip-flop SR (por ter memória) e por ser síncrono, sendo que foi usado um registo para cada casa. Atendendo ao comportamento dos flip-flops SR (fig1), criei o circuito atendendo à tabela da fig. 2, de onde se pode verificar que  $S = Ld * P$  e  $R = Ld * \bar{P}$  e que com  $\bar{Ld}$  vai manter o estado atual, tal como solicitado.

Fig 1

S	R	Q(n)
0	0	Q(n-1)
0	1	0
1	0	1
1	1	X

Fig 2

Ld	P	S	R	Q(n)
0	0	0	0	Q(n-1)
0	1	0	0	Q(n-1)
1	0	0	1	0
1	1	1	0	1

Mostra-se então o circuito funcional, fig 3, sem a macro, fig 4 com a macro “Registo de 1 Bit com Flip-Flop SR” com as entradas  $Ld$  e  $P$  e uma saída, para fazer a ligação à macro “Led\_Lamberta”:

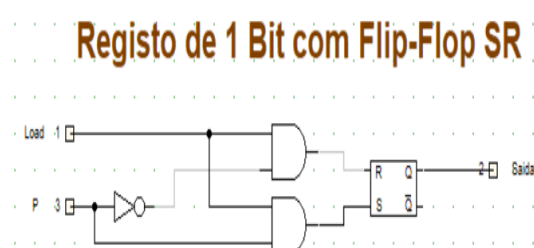


Fig 3

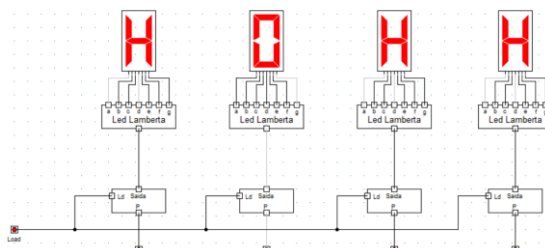
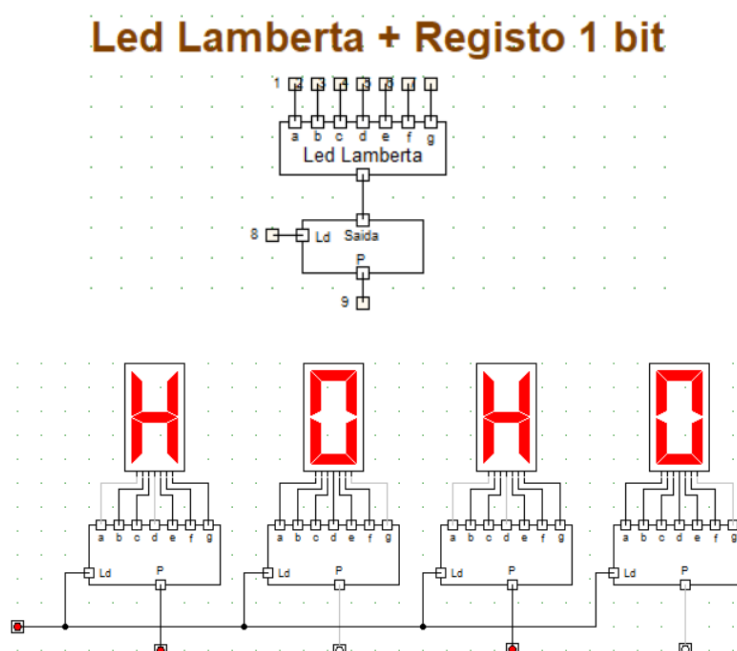


Fig4

Após a criação de todo o circuito conforme pedido, achei interessante agrupar em uma só macro, “Led Lamberta + Registro 1 bit”, as macros “Registro de 1 Bit com Flip-Flop SR” e “Led Lamberta” pois assim simplificava um pouco quando fosse necessário adicionar mais casas. Mostra-se então o circuito funcional e a respectiva macro:



### Questão 3)

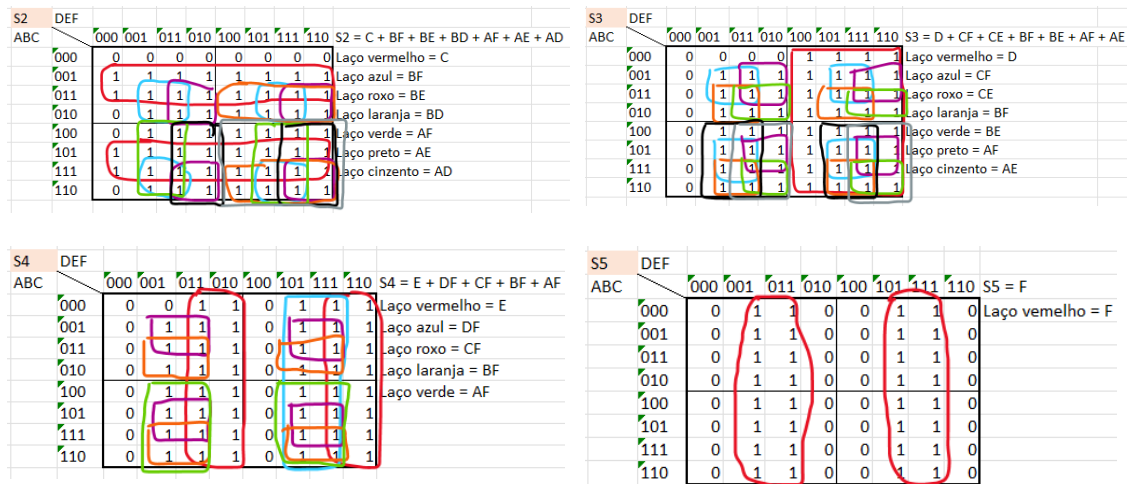
Para esta questão decidi partir o problema em duas partes para melhor implementação, a parte da seleção das casas e consequente ativação dos Led's e a parte da inversão das casas selecionadas. Comecei por resolver a parte da seleção das casas. A minha abordagem foi elaborar uma tabela de verdade com todas as possibilidades de casas selecionadas para saber quais os led's a ativar, e com base na tabela e respectivos mapas de Karaugh. Como a tabela é muito grande, foi colocada nos anexos, coloquei também os mapas para melhor entendimento.

S0	DEF	000	001	011	010	100	101	111	110
ABC									
000		0	0	0	0	0	0	0	0
001		0	0	0	0	0	0	0	0
011		0	0	0	0	0	0	0	0
010		0	0	0	0	0	0	0	0
100		1	1	1	1	1	1	1	1
101		1	1	1	1	1	1	1	1
111		1	1	1	1	1	1	1	1
110		1	1	1	1	1	1	1	1

S0 = A  
Laço vermelho = A

S1	DEF	000	001	011	010	100	101	111	110
ABC									
000		0	0	0	0	0	0	0	0
001		0	0	0	0	0	0	0	0
011		1	1	1	1	1	1	1	1
010		1	1	1	1	1	1	1	1
100		0	1	1	1	1	1	1	1
101		1	1	1	1	1	1	1	1
111		1	1	1	1	1	1	1	1
110		1	1	1	1	1	1	1	1

S1 = B + AF + AE + AD + AC  
Laço vermelho = B  
Laço azul = AF  
Laço roxo = AE  
Laço laranja = AD  
Laço verde = AC



Com base nestes mapas criei uma macro “*Seleção Tabuleiro*” onde, para tornar mais legível, criei sub-macros para os segmentos das casas/display’s selecionados, com exceção do primeiro e o último pela sua simplicidade, identificados com “*DisplayX\_segment*”, consoante a que casa se refere.

Para a parte da inversão, como não preciso de manter o estado no FF, este apenas existe para tornar a jogada síncrona com clock. Decidi usar um flip-flop tipo D, pois este armazena na saída Q o valor inserido na entrada D. Como o FF usado é ativo no flanco ascendente, a inversão só é efetivada quando o botão de clock é inativado e depois ativado.

Para conseguir saber qual o valor que está no momento no display, uma vez que o botão desse display pode estar inativo, decidi alterar a macro criada na alínea anterior, “*Led Lamberta + Registo 1 bit*”, e acrescentar uma saída para saber qual esse valor.

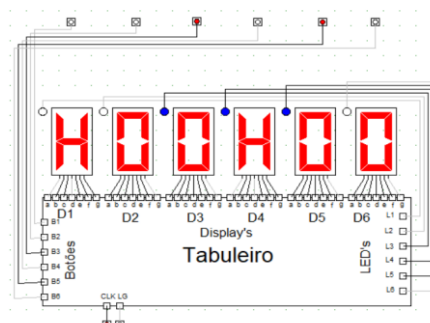
Fig 3

Led(A)	Display(B)	Q(n)
0	0	0
0	1	1
1	0	1
1	1	0

Então com o valor do Display e do Led através da tabela de verdade da Fig3, chegamos à lógica do FF:  $Q = A \oplus B$

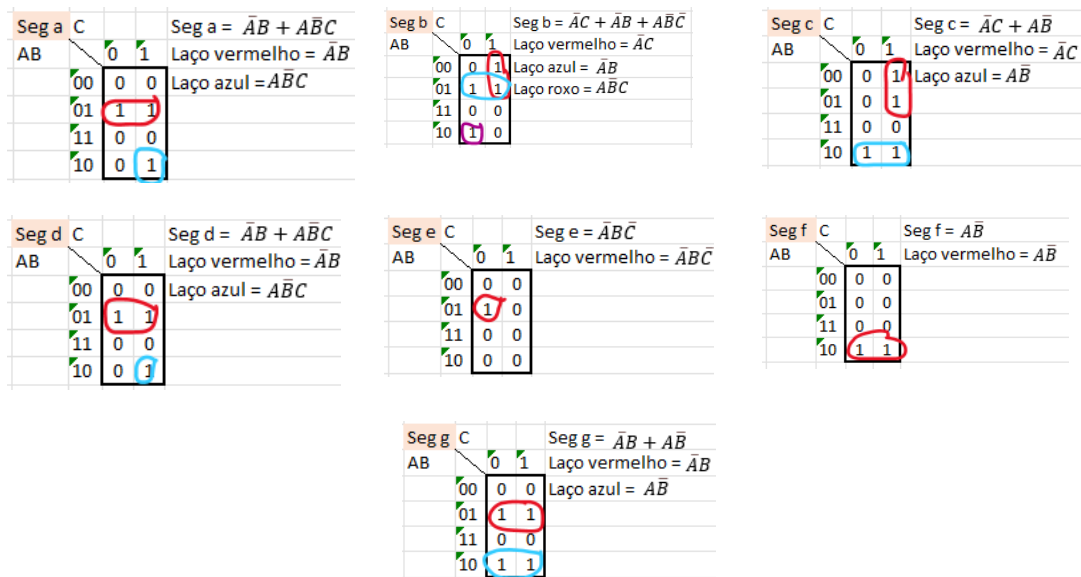
Para poder juntar à lógica da inversão também a possibilidade de se poder alterar o display e a seleção da casa aquando do set do tabuleiro, usei uma porta “AND” a receber o input do botão e do *load* onde liguei posteriormente a uma porta “OR”, porta essa também a receber a saída do FF, criando a macro “*Segmento inverte Tabuleiro*”.

Foi criado então uma macro com a junção da macro “*Segmento inverte Tabuleiro*”, uma para cada casa e a macro “*Seleção Tabuleiro*”. Para uma maior simplificação foi criada outra marco, “*Tabuleiro*” e agrega todas as macros criadas até então para permitir a criação do tabuleiro de jogo, todos os logigramas usados na criação da macro “*Tabuleiro*” foram colocados em anexo. Como são várias macros mostra-se unicamente o referido tabuleiro:



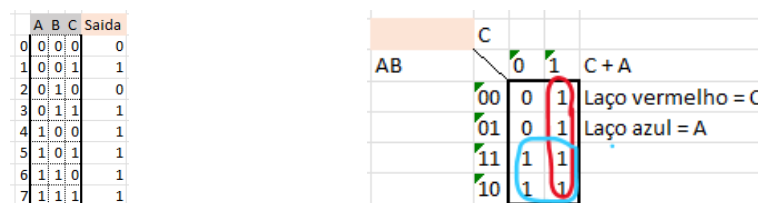
#### Questão 4)

O display do contador foi criado com o auxílio da tabela de verdade, em anexo, e respetivos mapas de Karnaugh.



Para o contador, implementei o circuito da AF6g, porém, como era pretendido que fosse um contador decrescente, ao invés do clock do 2 FF e seguintes, receber a saída Q do FF anterior, foi ligado o complementar e como é pretendido reiniciar o contador, o botão reset foi ligado aos “preset” dos FF’s, tudo isto implementado na macro “*Contador Decrescente 3 Bits*”.

Para a verificação se as casas seleccionadas eram inferiores ou iguais ao contador/jogada, criei um circuito “*Somador*” mas em cascata, macro em anexo, reaproveitando o circuito da AF5a. Está em cascata pois foi considerado que seria a soma de várias parcelas, sendo cada LED uma parcela, estando o carry\_out de um FF ligado ao carry\_in do FF seguinte. Ao qual depois foi ligado a um circuito “*Comparador interativo 3Bits*”, macro em anexo, reutilizado da AF6b, porém ao invés de 4bits, fiz só com 3 bits, os necessários para este programa. Então para poder validar quando o contador/jogada fosse maior ou igual que as casas seleccionadas, implementei as ligações de saída do comparador consoante a tabela de verdade, em anexo, e respetivo mapa de Karnaugh.



Para verificar se existia algum “H” nas casas seleccionadas, liguei uma porta “and” a receber o led e a saída 0/H, isto para todos os leds/ saída 0/H, e tudo ligado a uma porta “or”, por limitações do DW tive de desdobrar em 3 portas “or”, a ideia é: com a porta “and” se ambos forem 1, a saída é 1, e como basta um dos conjuntos (LED+H) para a jogada ser válida, ligar tudo á porta “or”.

Não fiz tabela para este esquema porque era direto e iria ser penoso fazer com tantas entradas.

Para validar se a jogada é valida, ou seja, se na seleção tem pelo menos um “H” e se o contador é maior ou igual que as casas seleccionadas, consoante a tabela de verdade e respetivo mapa de Karnaugh, onde A (Clock), B(H), C(Comparação) ficamos com o circuito da jogada válida:

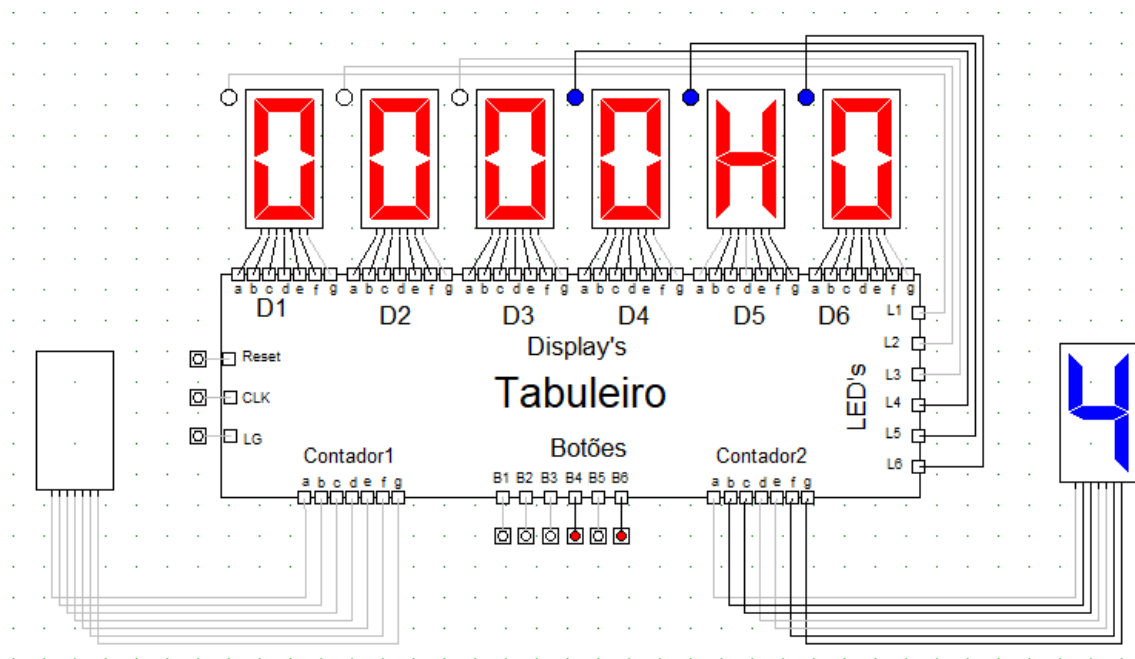
A	B	C	Saída
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

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

Laço vermelho = ABC

Todas estas macros e ligações criadas nesta alinea foram implementadas na macro do tabuleiro.

Ficamos assim com o aspeto final do “*tabuleiro*”, onde o botão “reset” reinicia o contador, o “CLK” é para fazer a jogada, e “LG”, se ativo, permite alterar entre 0/H, e apagado seleciona as casas ativas.



## Melhorias)

→ Quando jogador faz jogada inválida:

- **Presente:**  
Displays do contador dependem apenas do ímpar (bit de menor peso), o que faz alternar, porém essa alternância não tem implementado a validação da jogada.
- **Futuro:**  
Deveria também depender da jogada válida, onde com o recurso à tabela infra, podemos verificar que o resultado seria  $\bar{A}\bar{B} + AB$ , logo um “XNOR”. Porém não estou a conseguir implementar por causa do clock, pois interage antes do tempo.

JV	Clock	
A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

→ Quando contador a 1 + jogada válida:

- **Presente:**  
Contador continua sempre a decrescer até ambos os contadores chegarem ao 0 (apagados).
- **Futuro:**  
Inibir o clock e continuar a alternar o display, desde que jogada válida.

## ANEXOS

Macro "Led Lamberta":

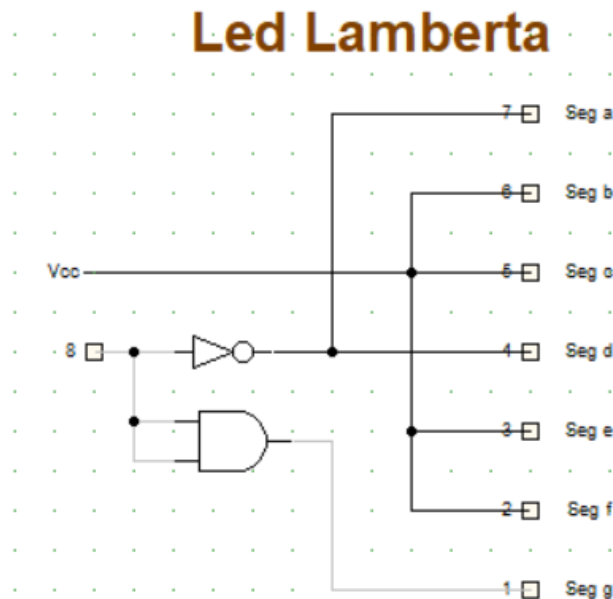


Tabela de verdade com todas as possibilidades de casas seleccionadas para saber quais os led's a ativar e mapas de Karnaugh mencionados na Q2:

S0	DEF								
ABC		000	001	011	010	100	101	111	110
000		0	0	0	0	0	0	0	0
001		0	0	0	0	0	0	0	0
011		0	0	0	0	0	0	0	0
010		0	0	0	0	0	0	0	0
100		1	1	1	1	1	1	1	1
101		1	1	1	1	1	1	1	1
111		1	1	1	1	1	1	1	1
110		1	1	1	1	1	1	1	1

S0 = A  
Laço vermelho = A

S1	DEF								
ABC		000	001	011	010	100	101	111	110
000		0	0	0	0	0	0	0	0
001		0	0	0	0	0	0	0	0
011		1	1	1	1	1	1	1	1
010		1	1	1	1	1	1	1	1
100		0	1	1	1	1	1	1	1
101		1	1	1	1	1	1	1	1
111		1	1	1	1	1	1	1	1
110		1	1	1	1	1	1	1	1

S1 = B + AF + AE + AD + AC  
Laço vermelho = B  
Laço azul = AF  
Laço roxo = AE  
Laço laranja = AD  
Laço verde = AC

S2	DEF									
ABC		000	001	011	010	100	101	111	110	S2 = C + BF + BE + BD + AF + AE + AD
000		0	0	0	0	0	0	0	0	Linha vermelho = C
001		1	1	1	1	1	1	1	1	Linha azul = BF
011		1	1	1	1	1	1	1	1	Linha roxo = BE
010		0	1	1	1	1	1	1	1	Linha laranja = BD
100		0	1	1	1	1	1	1	1	Linha verde = AF
101		1	1	1	1	1	1	1	1	Linha preto = AE
111		1	1	1	1	1	1	1	1	Linha cinzento = AD
110		0	1	1	1	1	1	1	1	

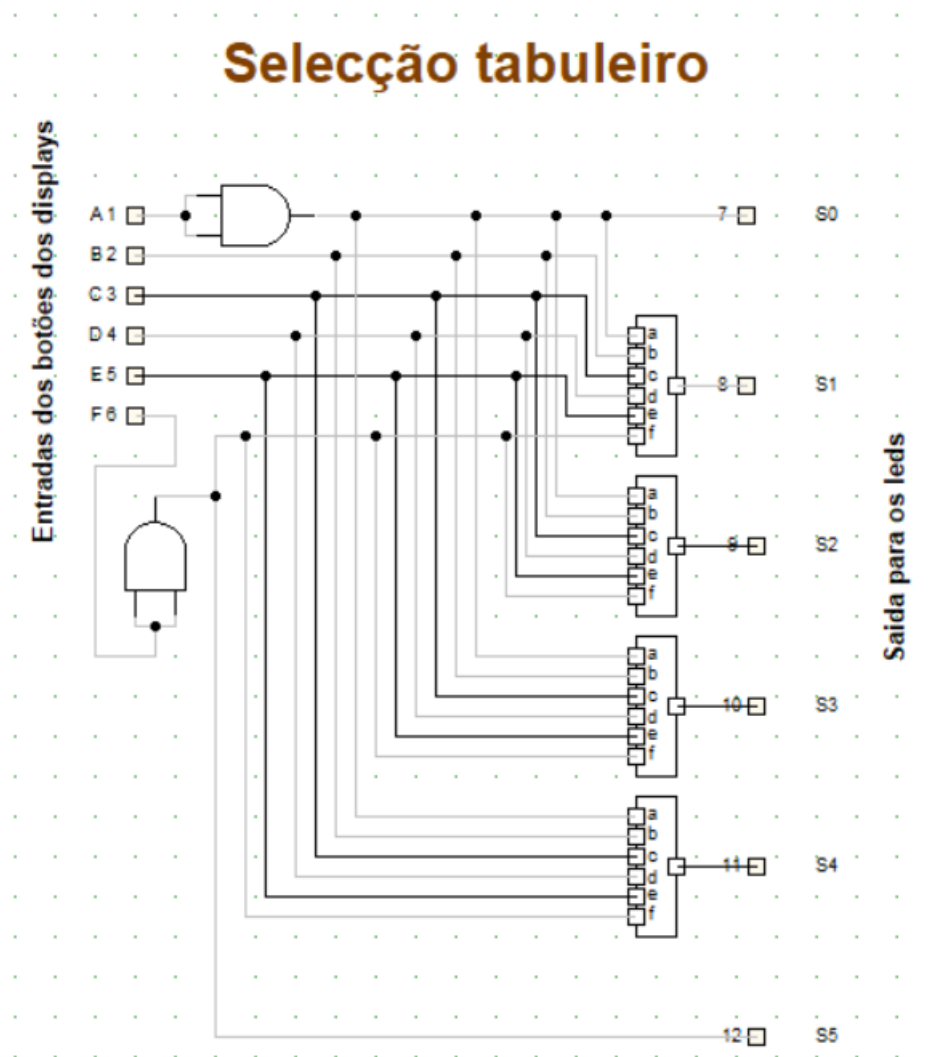
S3	DEF									
ABC		000	001	011	010	100	101	111	110	S3 = D + CF + CE + BF + BE + AF + AE
000		0	0	0	0	1	1	1	1	Laço vermelho = D
001		0	1	1	1	1	1	1	1	Laço azul = CF
011		0	1	1	1	1	1	1	1	Laço roxo = CE
010		0	1	1	1	1	1	1	1	Laço laranja = BF
100		0	1	1	1	1	1	1	1	Laço verde = BE
101		0	1	1	1	1	1	1	1	Laço preto = AF
111		0	1	1	1	1	1	1	1	Laço cinzento = AE
110		0	1	1	1	1	1	1	1	

S4	DEF									
ABC		000	001	011	010	100	101	111	110	S4 = E + DF + CF + BF + AF
000	0	0	1	1	0	1	1	1	1	Laço vermelho = E
001	0	1	1	1	0	1	1	1	1	Laço azul = DF
011	0	1	1	1	0	1	1	1	1	Laço roxo = CF
010	0	1	1	1	0	1	1	1	1	Laço laranja = BF
100	0	1	1	1	0	1	1	1	1	Laço verde = AF
101	0	1	1	1	0	1	1	1	1	
111	0	1	1	1	0	1	1	1	1	
110	0	1	1	1	0	1	1	1	1	

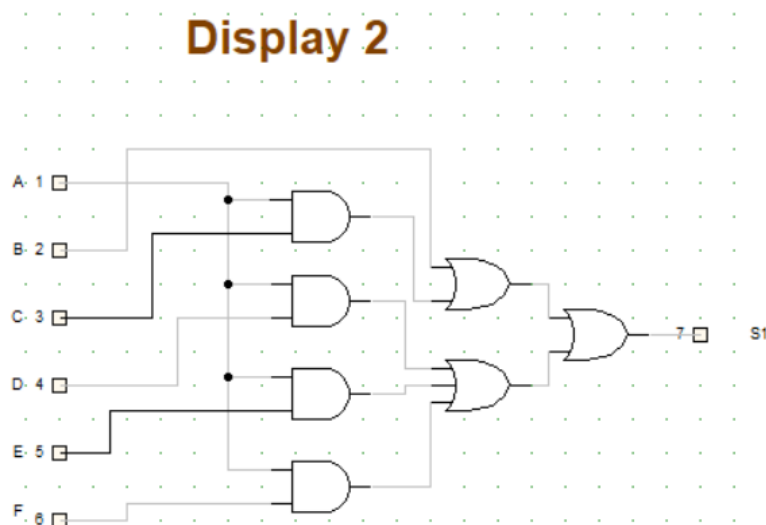
S5	DEF	000	001	011	010	100	101	111	110	
ABC										S5 = F
	000	0	1	1	0	0	1	1	0	Laço vermelho = F
	001	0	1	1	0	0	1	1	0	
	011	0	1	1	0	0	1	1	0	
	010	0	1	1	0	0	1	1	0	
	100	0	1	1	0	0	1	1	0	
	101	0	1	1	0	0	1	1	0	
	111	0	1	1	0	0	1	1	0	
	110	0	1	1	0	0	1	1	0	



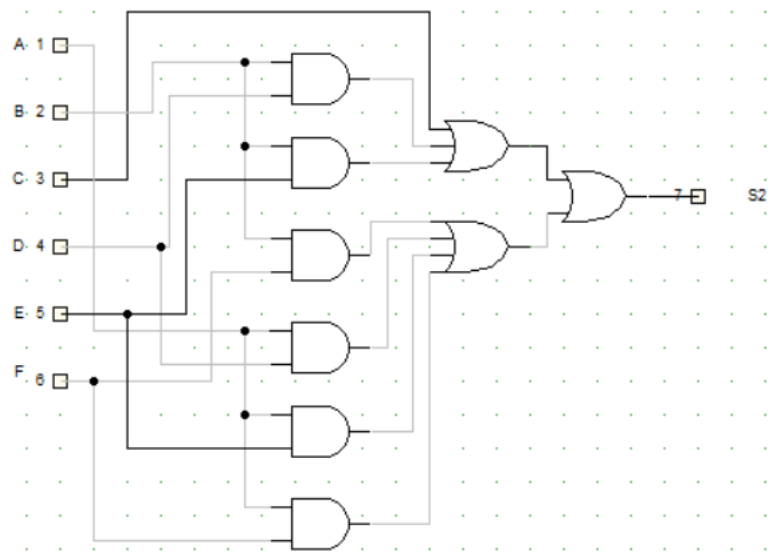
	A	B	C	D	E	F	S0	S1	S2	S3	S4	S5
0	0	0	0	0			0	0	0	0	0	0
1	0	0	0	0			0	0	0	0	0	1
2	0	0	0	0			0	0	0	0	1	0
3	0	0	0	0			0	0	0	0	1	1
4	0	0	0	1			0	0	0	1	0	0
5	0	0	0	1			0	0	0	1	1	1
6	0	0	0	1			0	0	0	1	1	0
7	0	0	0	1			0	0	0	1	1	1
8	0	0	1	0			0	0	1	0	0	0
9	0	0	1	0			0	0	1	1	1	1
10	0	0	1	0			0	0	1	1	1	0
11	0	0	1	0			0	0	1	1	1	1
12	0	0	1	1			0	0	1	1	0	0
13	0	0	1	1			0	0	1	1	1	1
14	0	0	1	1			0	0	1	1	1	0
15	0	0	1	1			0	0	1	1	1	1
16	0	1	0	0			0	1	0	0	0	0
17	0	1	0	0			0	1	1	1	1	1
18	0	1	0	0			0	1	1	1	1	0
19	0	1	0	0			0	1	1	1	1	1
20	0	1	0	1			0	1	1	1	0	0
21	0	1	0	1			0	1	1	1	1	1
22	0	1	0	1			0	1	1	1	1	0
23	0	1	0	1			0	1	1	1	1	1
24	0	1	1	0			0	1	1	0	0	0
25	0	1	1	0			0	1	1	1	1	1
26	0	1	1	0			0	1	1	1	1	0
27	0	1	1	0			0	1	1	1	1	1
28	0	1	1	1			0	1	1	1	0	0
29	0	1	1	1			0	1	1	1	1	1
30	0	1	1	1			0	1	1	1	1	0
31	0	1	1	1			0	1	1	1	1	1
32	1	0	0	0			1	0	0	0	0	0
33	1	0	0	0			1	1	1	1	1	1
34	1	0	0	0			1	1	1	1	1	0
35	1	0	0	0			1	1	1	1	1	1
36	1	0	0	1			1	1	1	1	0	0
37	1	0	0	1			1	1	1	1	1	1
38	1	0	0	1			1	1	1	1	1	0
39	1	0	0	1			1	1	1	1	1	1
40	1	0	1	0			1	1	1	0	0	0
41	1	0	1	0			1	1	1	1	1	1
42	1	0	1	0			1	1	1	1	1	0
43	1	0	1	0			1	1	1	1	1	1
44	1	0	1	1			1	1	1	1	0	0
45	1	0	1	1			1	1	1	1	1	1
46	1	0	1	1			1	1	1	1	1	0
47	1	0	1	1			1	1	1	1	1	1
48	1	1	0	0			1	1	0	0	0	0
49	1	1	0	0			1	1	1	1	1	1
50	1	1	0	0			1	1	1	1	1	0
51	1	1	0	0			1	1	1	1	1	1
52	1	1	0	1			1	1	1	1	0	0
53	1	1	0	1			1	1	1	1	1	1
54	1	1	0	1			1	1	1	1	1	0
55	1	1	0	1			1	1	1	1	1	1
56	1	1	1	0			1	1	1	0	0	0
57	1	1	1	0			1	1	1	1	1	1
58	1	1	1	0			1	1	1	1	1	0
59	1	1	1	0			1	1	1	1	1	1
60	1	1	1	1			1	1	1	1	0	0
61	1	1	1	1			1	1	1	1	1	1
62	1	1	1	1			1	1	1	1	1	0
63	1	1	1	1			1	1	1	1	1	1



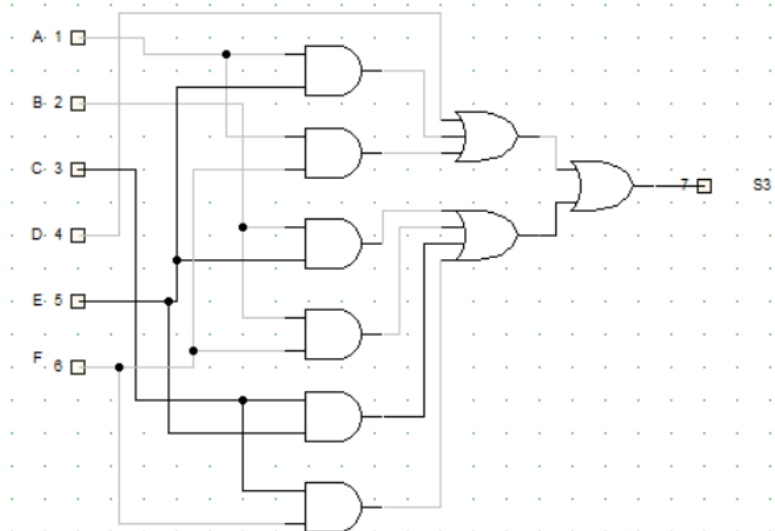
Macros dos displays identificados com "DisplayX\_segment"



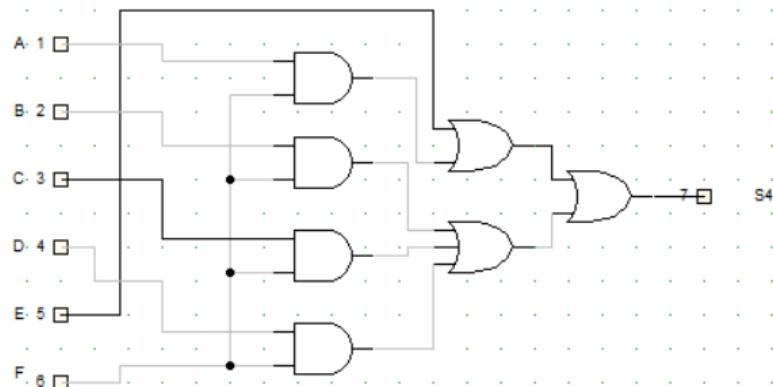
### Display 3



### Display 4

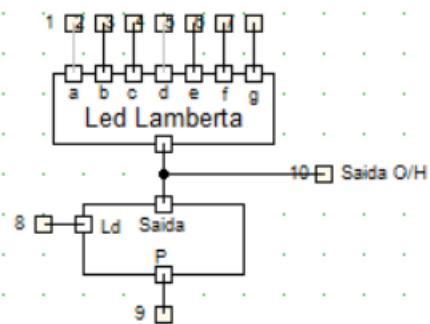


## Display 5



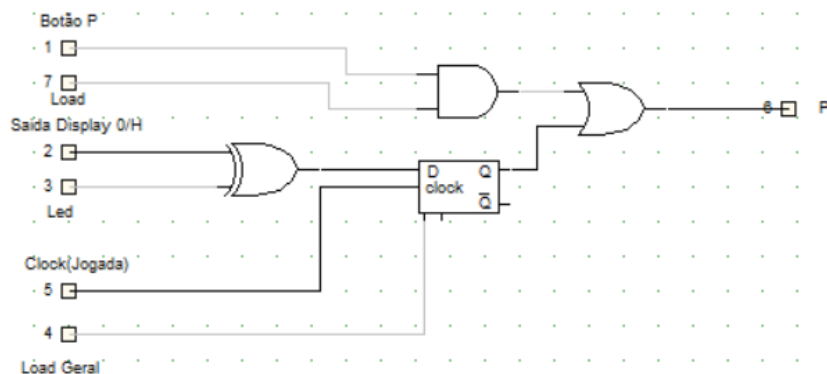
Macro “Led Lamberta + Registro 1 bit” com a inclusão de Saida O/H:

## Led Lamberta + Registro 1 bit

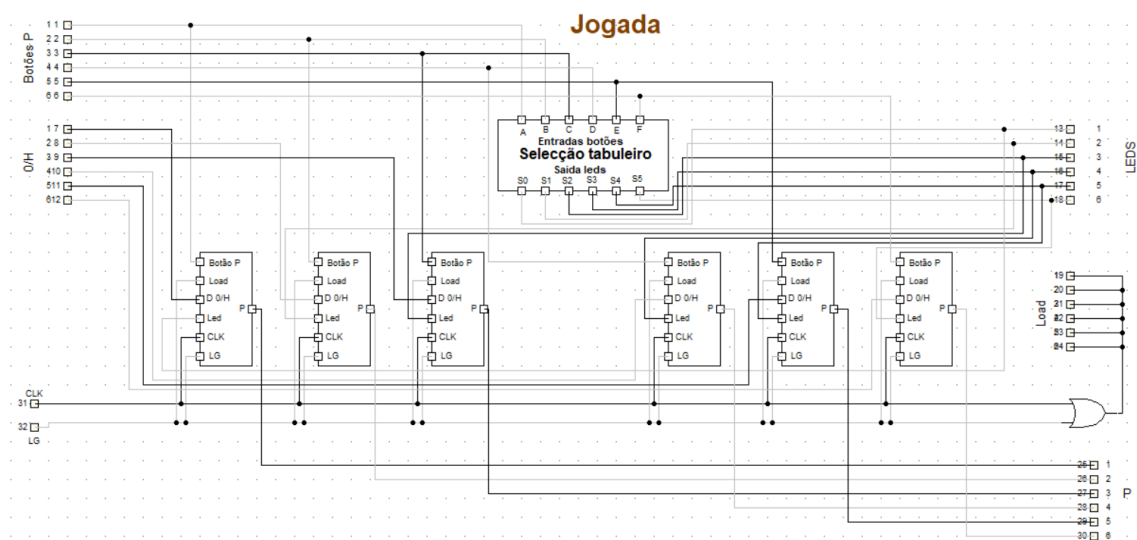


Macro “Segmento inverte Tabuleiro”:

## Segmento inverte Tabuleiro



Macro “Jogada”:



Macro “Tabuleiro”:

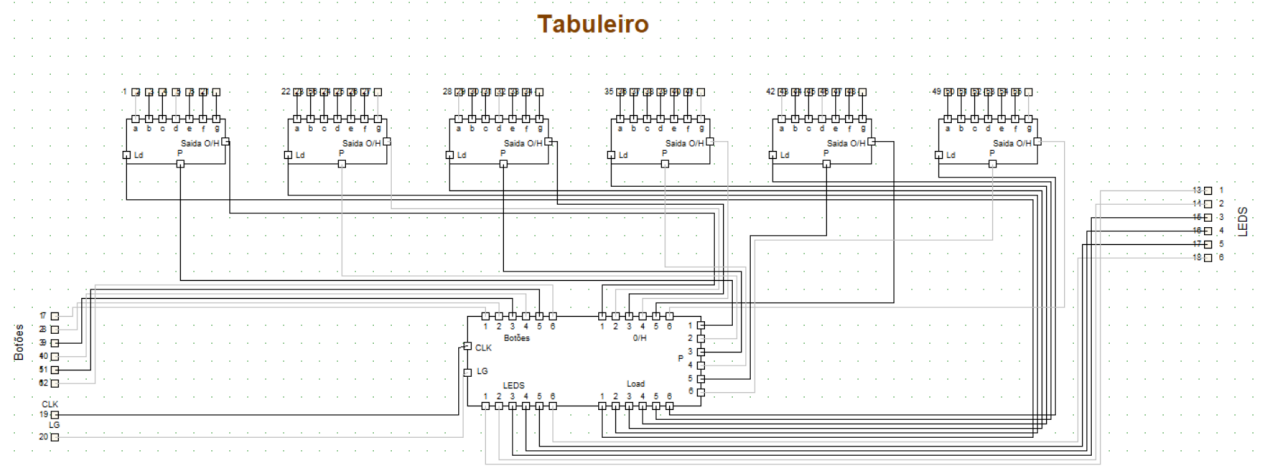


Tabela de verdade do Display do contador e mapas de Karnaugh mencionados na Q4:

	A	B	C	a	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	0	Jogada Inválida
1	0	0	1	0	1	1	0	0	0	0	1
2	0	1	0	1	1	0	1	1	0	1	2
3	0	1	1	1	1	1	1	0	0	1	3
4	1	0	0	1	1	0	0	1	1	1	4
5	1	0	1	1	0	1	1	0	1	1	5
6	1	1	0	0	0	0	0	0	0	0	Apagado
7	1	1	1	0	0	0	0	0	0	0	Apagado

Seg a	C		
AB	0	1	
00	0	0	
01	1	1	
11	0	0	
10	0	1	

Seg a =  $\bar{A}B + A\bar{B}C$   
 Laço vermelho =  $\bar{A}B$   
 Laço azul =  $A\bar{B}C$

Seg b	C		
AB	0	1	
00	0	1	
01	1	1	
11	0	0	
10	1	0	

Seg b =  $\bar{A}C + \bar{A}B + A\bar{B}C$   
 Laço vermelho =  $\bar{A}C$   
 Laço azul =  $\bar{A}B$   
 Laço roxo =  $A\bar{B}C$

Seg c		C			Seg c = $\bar{A}C + A\bar{B}$
AB		0	1		Laço vermelho = $\bar{A}C$
	00	0	1		Laço azul = $A\bar{B}$
	01	0	1		
	11	0	0		
	10	1	1		

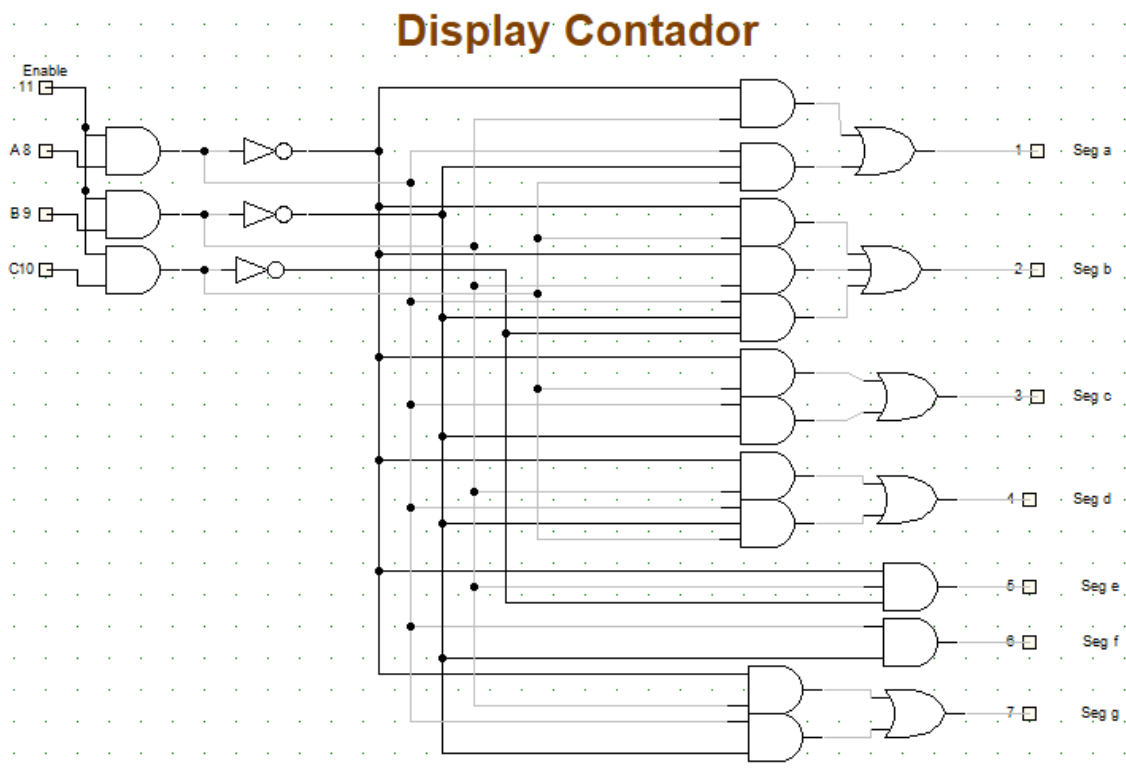
Seg d		C			Seg d = $\bar{A}B + A\bar{B}C$
AB		0	1		Laço vermelho = $\bar{A}B$
	00	0	0		Laço azul = $A\bar{B}C$
	01	1	1		
	11	0	0		
	10	0	1		

Seg e		C			Seg e = $\bar{A}B\bar{C}$
AB		0	1		Laço vermelho = $\bar{A}B\bar{C}$
	00	0	0		
	01	1	0		
	11	0	0		
	10	0	0		

Seg f		C			Seg f = $A\bar{B}$
AB		0	1		Laço vermelho = $A\bar{B}$
	00	0	0		
	01	0	0		
	11	0	0		
	10	1	1		

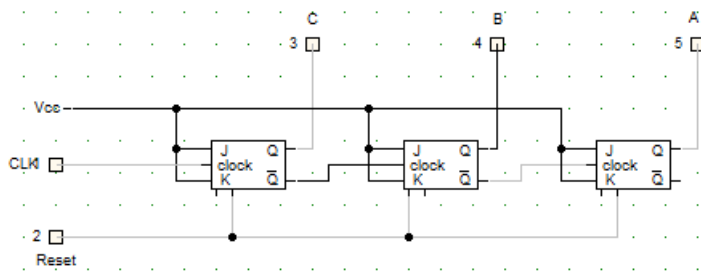
Seg g		C			Seg g = $\bar{A}B + A\bar{B}$
AB		0	1		Laço vermelho = $\bar{A}B$
	00	0	0		Laço azul = $A\bar{B}$
	01	1	1		
	11	0	0		
	10	1	1		

Macro "Display Contador":



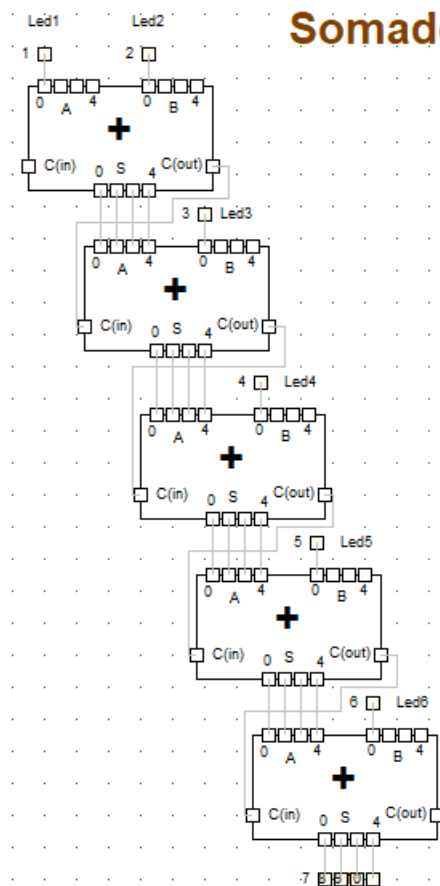
Macro “Contador Decrescente 3 Bits “.

## Contador Decrescente 3 Bits (101)

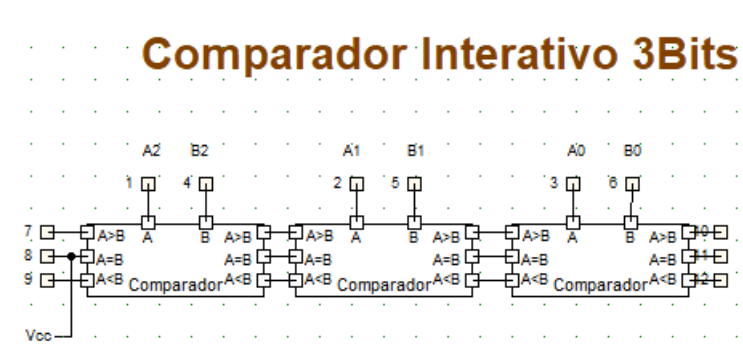


Macro “Somador Completo em cascata”

## Somador Completo em cascata



Macro “Comparador interativo 3Bits”



Macro final do “Tabuleiro”

