

CIRCUITOS DIGITAIS

EXERCÍCIOS

CIRCUITOS COMBINACIONAIS

Prof. Marcelo Grandi Mandelli

`mgmandelli@unb.br`

Projeto de Circuitos Combinacionais

1º Passo – Defina as entradas e as saídas

2º Passo – Defina valores para os estados das estradas e saídas

3º Passo – Obtenha a(s) função(ões) booleana(s)

4º Passo – Simplifique a equação se necessário

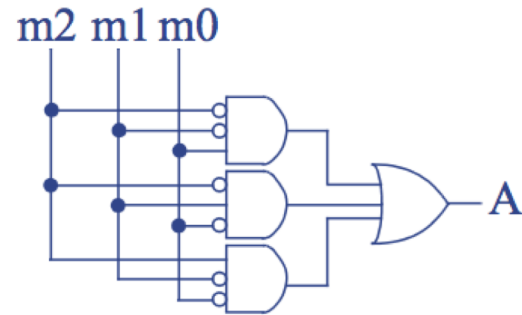
5º Passo – Implemente um circuito baseado em portas

Alarme de Segurança 1

Crie um circuito para o museu do Exercício 2.53 que detecta se o guarda está fazendo a ronda no museu de maneira apropriada. Isso pode ser detectado quando há *exatamente* um sensor de movimento em 1. (Se nenhum sensor estiver em 1, o guarda deve estar sentado ou dormindo.)

Alarme de Segurança 1

Inputs			Outputs
m2	m1	m0	A
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



$$A = m2'm1'm0 + m2'm1m0' + m2m1'm0'$$

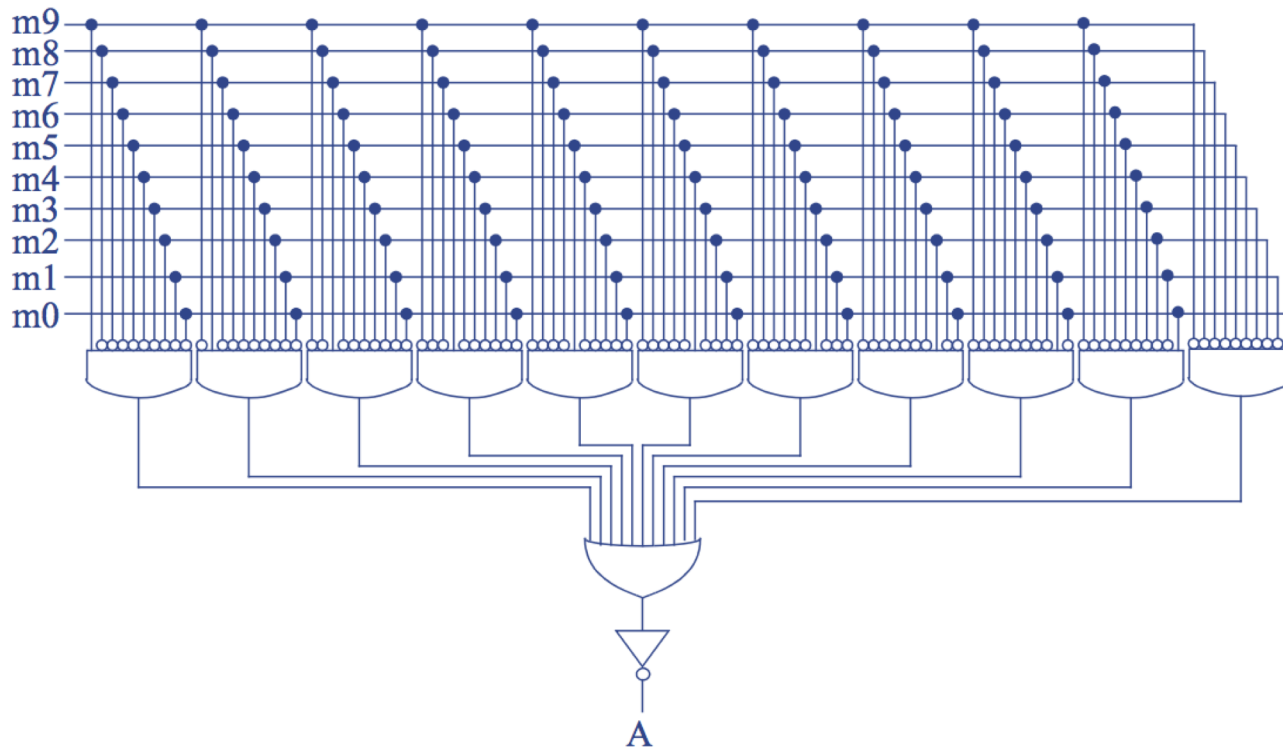
Alarme de Segurança 2

Considere a função do alarme de segurança para museu do Exercício 2.53, mas para um museu com 10 salões. Uma tabela-verdade não é um bom ponto de partida (linhas demais), nem uma equação que descreva quando o alarme deve soar (equações demais). No entanto, a inversa (NOT) da função de alarme pode ser obtida rapidamente na forma de uma equação. Projete o circuito para um sistema de segurança de 10 salões, projetando o inverso da função e, então, simplesmente acrescentando um inversor antes da saída da função.

Alarme de Segurança 2

$A' =$

$$\begin{aligned} & m_9 m_8' m_7' m_6' m_5' m_4' m_3' m_2' m_1' m_0' + m_9' m_8 m_7' m_6' m_5' m_4' m_3' m_2' m_1' m_0' + \\ & m_9' m_8' m_7 m_6' m_5' m_4' m_3' m_2' m_1' m_0' + m_9' m_8' m_7' m_6 m_5' m_4' m_3' m_2' m_1' m_0' + \\ & m_9' m_8' m_7' m_6' m_5 m_4' m_3' m_2' m_1' m_0' + m_9' m_8' m_7' m_6' m_5' m_4 m_3' m_2' m_1' m_0' + \\ & m_9' m_8' m_7' m_6' m_5' m_4' m_3 m_2' m_1' m_0' + m_9' m_8' m_7' m_6' m_5' m_4' m_3' m_2 m_1' m_0' + \\ & m_9' m_8' m_7' m_6' m_5' m_4' m_3' m_2' m_1 m_0' + m_9' m_8' m_7' m_6' m_5' m_4' m_3' m_2' m_1' m_0 + \\ & m_9' m_8' m_7' m_6' m_5' m_4' m_3' m_2' m_1' m_0' \end{aligned}$$

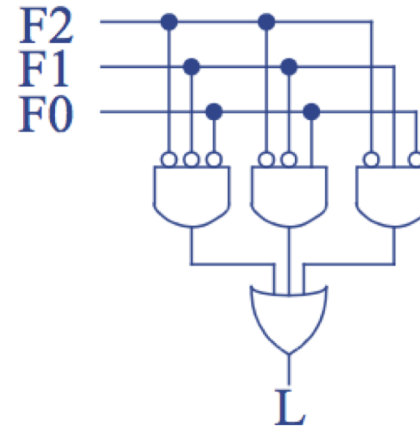


Sensores Automobilísticos 1

Um carro tem um detector de nível baixo de combustível que fornece o nível corrente de combustível na forma de um número binário de três bits, com 000 significando vazio e 111 significando cheio. Crie um circuito que acende a luz indicadora “Pouco combustível” (fazendo uma saída L ir para 1) quando o nível de combustível cai para abaixo do nível 3.

Sensores Automobilísticos 1

Inputs			Outputs
F2	F1	F0	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



$$L = F2'F1'F0' + F2'F1'F0 + F2'F1F0'$$

Sensores Automobilísticos 2

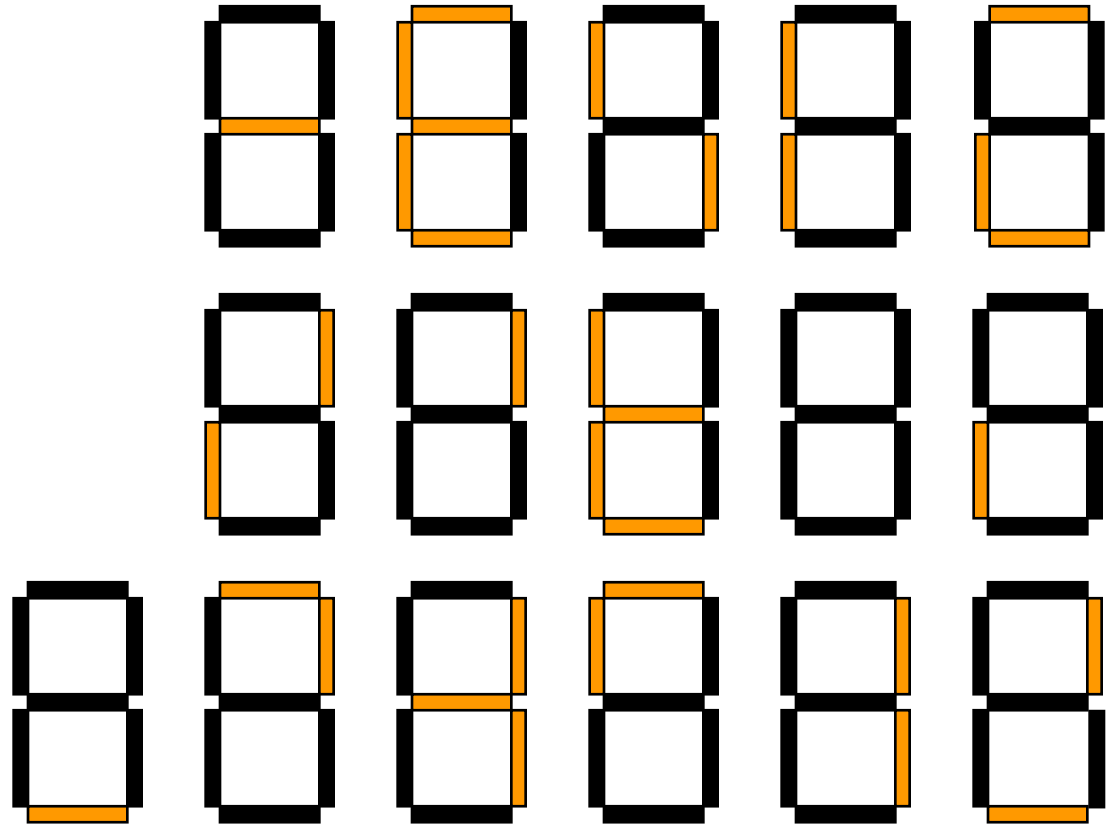
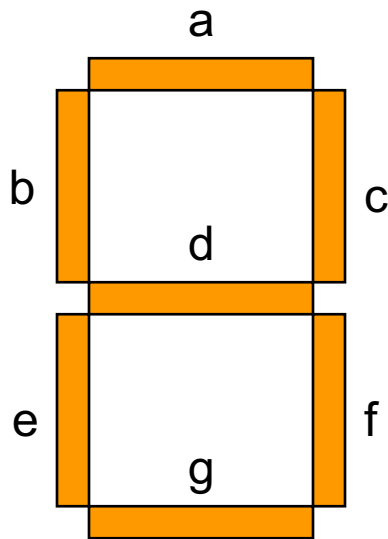
Um carro tem um sensor para pressão baixa de pneu que fornece a pressão atual do pneu na forma de um número binário de cinco bits. Crie um circuito que acende a luz indicadora “Pressão de pneu baixa” (fazendo uma saída L ir para 1) quando a pressão cai para abaixo do nível 16. Sugestão: pode ser mais fácil criar um circuito que detecta a função inversa. Você poderá então acrescentar simplesmente um inversor à saída do circuito.

Sensores Automobilísticos 2

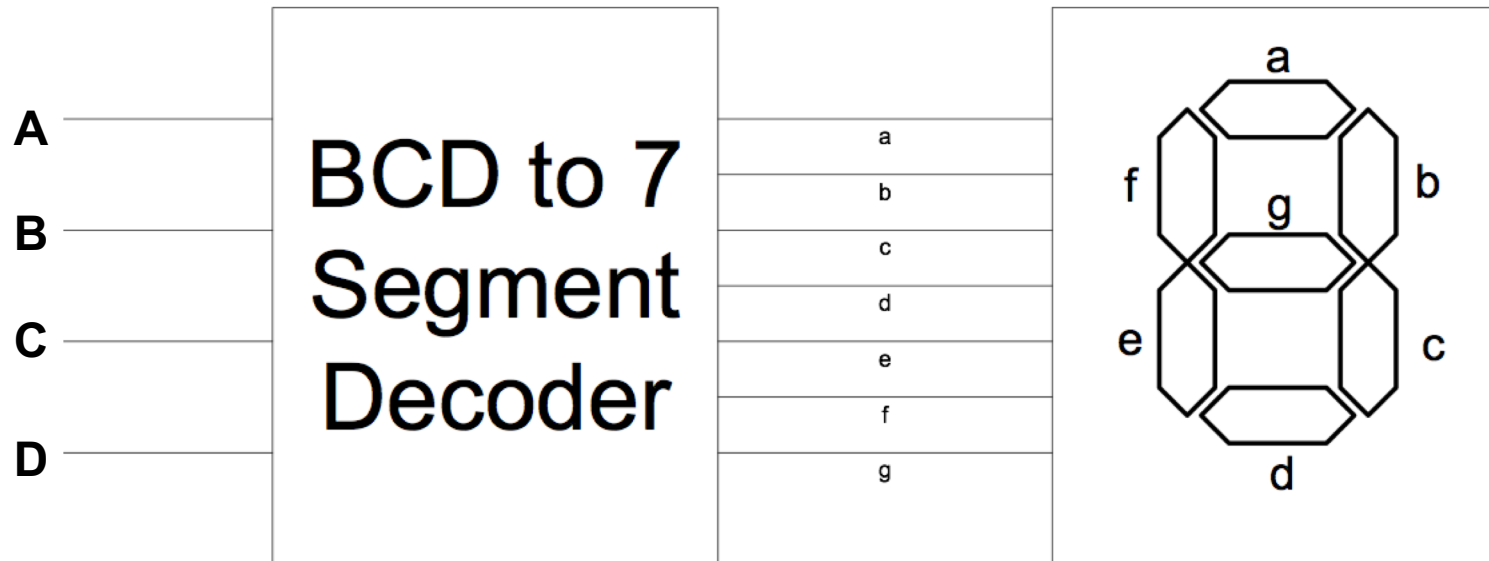
Um carro tem um sensor para pressão baixa de pneu que fornece a pressão atual do pneu na forma de um número binário de cinco bits. Crie um circuito que acende a luz indicadora “Pressão de pneu baixa” (fazendo uma saída L ir para 1) quando a pressão cai para abaixo do nível 16. Sugestão: pode ser mais fácil criar um circuito que detecta a função inversa. Você poderá então acrescentar simplesmente um inversor à saída do circuito.



Display de 7 segmentos



Decodificador BCD/display de 7 seg.



Números possíveis e sua representação em display de 7 segmentos

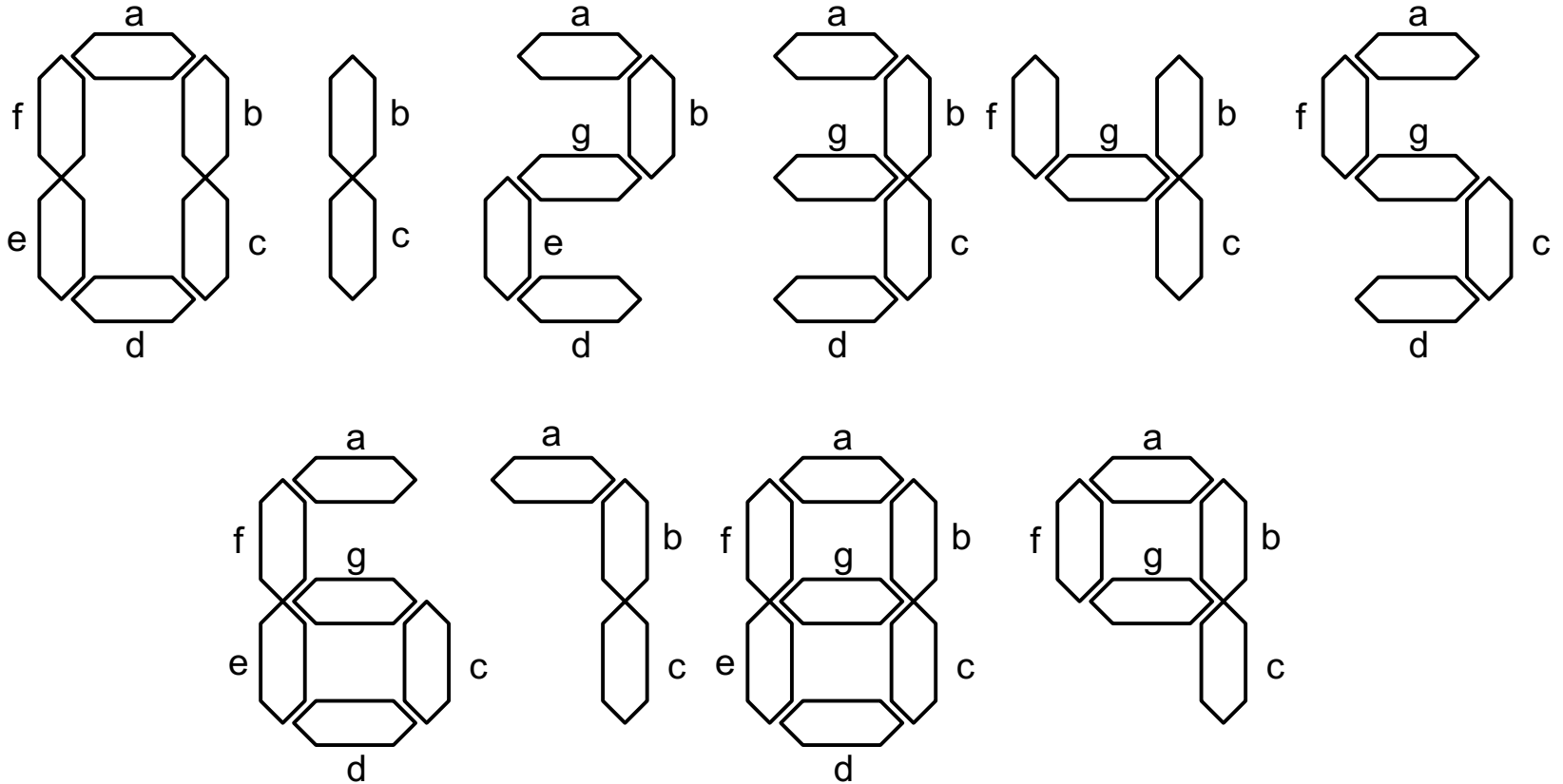
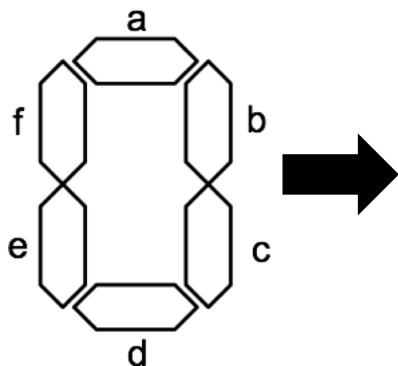


Tabela Verdade



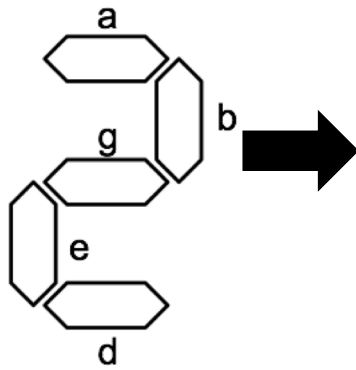
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							

Tabela Verdade



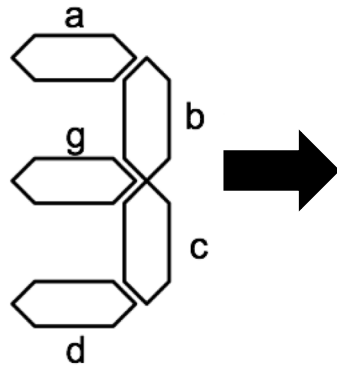
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							

Tabela Verdade



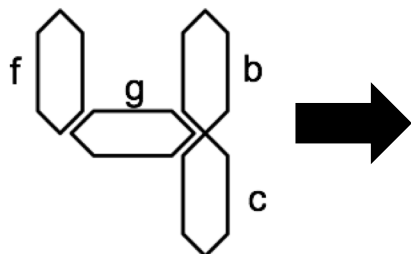
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							

Tabela Verdade



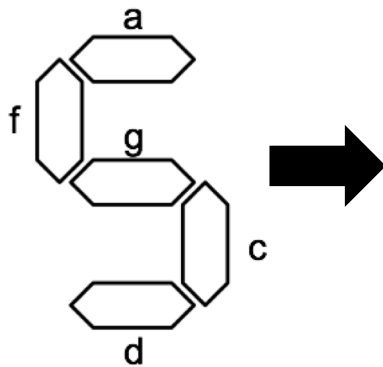
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							

Tabela Verdade



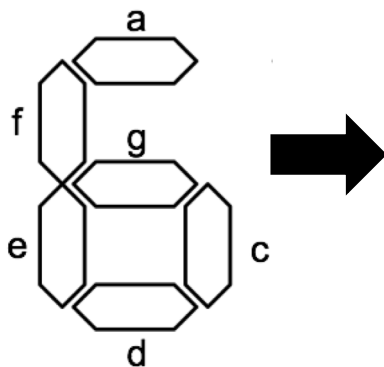
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							

Tabela Verdade



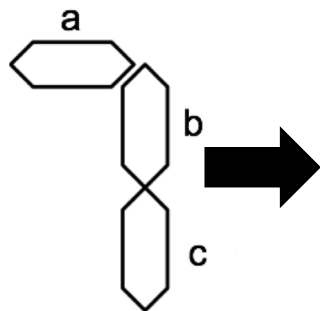
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							

Tabela Verdade



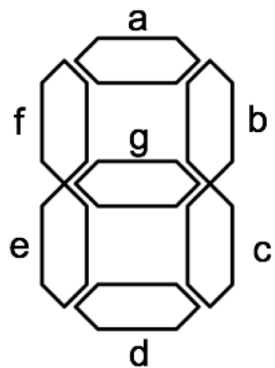
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1							
1	0	0	0							
1	0	0	1							

Tabela Verdade



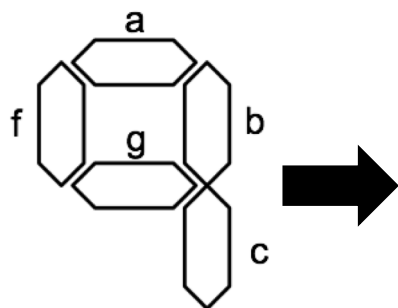
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0							
1	0	0	1							

Tabela Verdade



A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1							

Tabela Verdade



A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1

Tabela Verdade

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

Tabela Verdade

A	B	C	D	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

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

$$a = (A+B+C+D')(B'+C+D)$$

Tabela Verdade

A	B	C	D	b
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

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

$$b = (B' + C + D')(B' + C' + D)$$

Tabela Verdade

A	B	C	D	c
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

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

$$c = (B + C' + D)$$

Tabela Verdade

A	B	C	D	d
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
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	1
1	0	0	1	0
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

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

$$d = (B'+C+D)(B'+C'+D') (B+C+D')$$

Tabela Verdade

A	B	C	D	e
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

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

$$e = (B' + C)(D')$$

Tabela Verdade

A	B	C	D	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
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	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

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

$$f = (A+B+D')(C'+D')(B+C')$$

Tabela Verdade

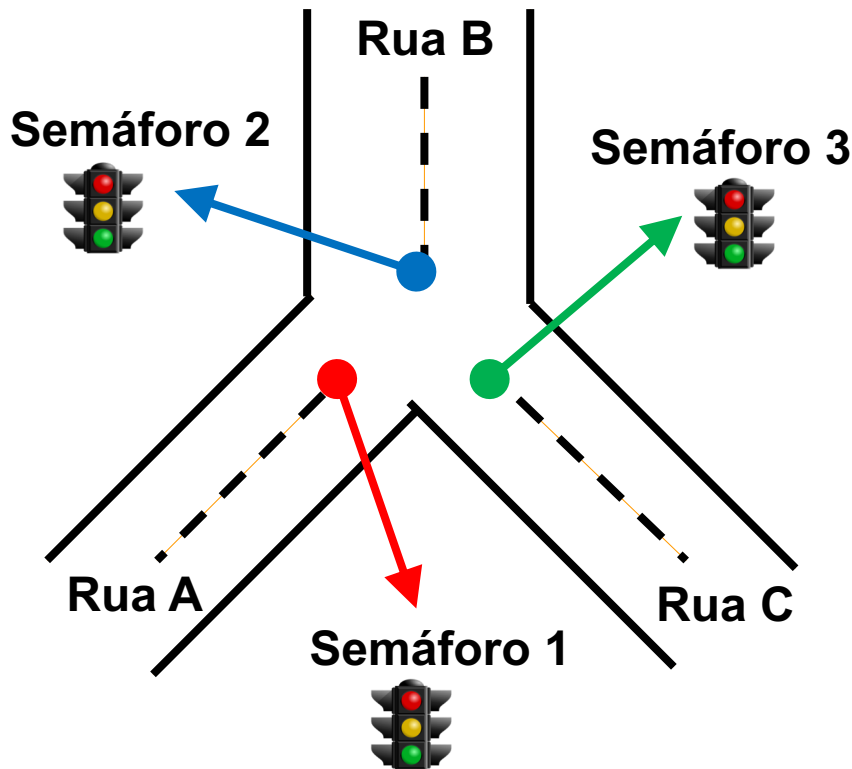
A	B	C	D	g
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

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

$$g = (A+B+C)(B'+C'+D')$$

Semáforos

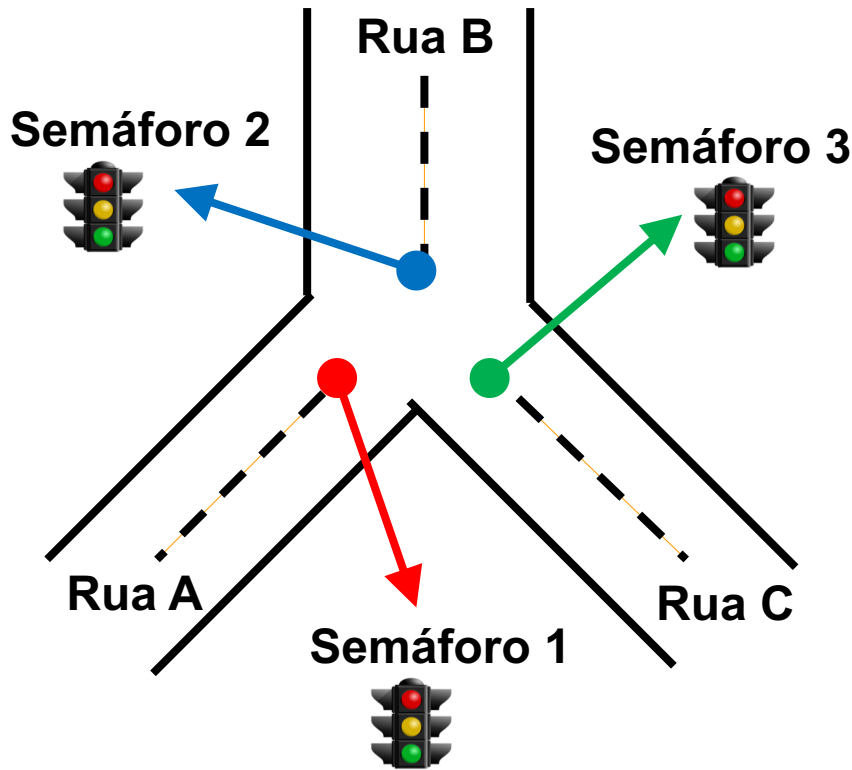
- A figura abaixo mostra o entroncamento das ruas A, B e C. Neste cruzamento, queremos instalar um conjunto de semáforos para as seguintes funções:



- Os semáforos somente tem sinais verde e vermelho
- Quando o semáforo 1 abrir para a Rua A, automaticamente os semáforos 2 e 3 devem fechar, para possibilitar ao motorista ambas as conversões.
- Analogamente, quando o semáforo 2 abrir, devem fechar os semáforos 1 e 3.
- Pelo mesmo motivo, quando o semáforo 3 abrir, devem fechar os semáforos 1 e 2.

Semáforos

□ Devemos seguir também, as seguintes prioridades:



O motorista que está na rua A tem prioridade em relação ao motorista que está na rua B.

O motorista que está na rua B tem prioridade em relação ao motorista que está na rua C.

O motorista que está na rua C tem prioridade em relação ao motorista que está na rua A.

Quando houver carros nas três ruas, a rua A é preferencial.

Quando não houver nenhum carro nas ruas, devemos abrir o sinal para a rua A.

Semáforos

- Obtenha as expressões e os circuitos dos sinais verdes e vermelhos, dos semáforos 1, 2 e 3.

