Circuitos Combinacionais

Conceitos Gerais

1. Introdução

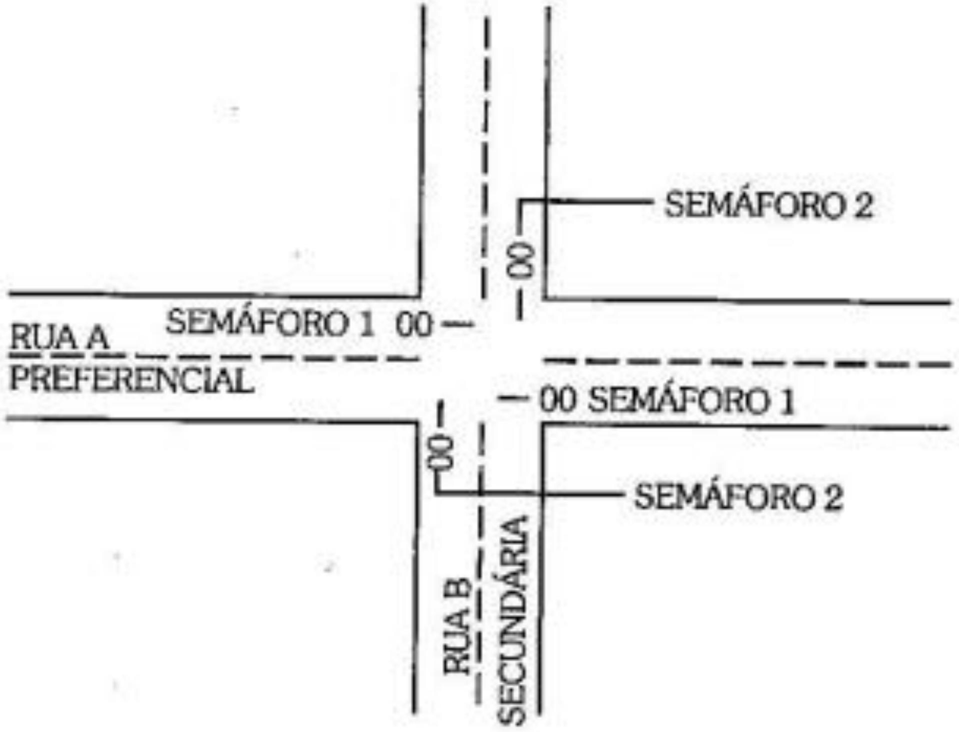
- Os **circuitos combinacionais** são importantes para que seja possível entender o funcionamento de alguns tipos de estruturas, como circuitos somadores, decodificadores, etc.

- Um circuito combinacional pode ser utilizado para solucionar problemas em que é necessário de uma resposta quando ocorrerem determinadas condições (situações) nas variáveis de entrada.

- Lembrando que é sempre importante garantir que o circuito combinacional tenha sua estrutura simplificada ao máximo possível, pois isto reduz seu custo e aumenta sua eficiência. A imagem abaixo mostra a representação de um circuito com suas variáveis de E/S:

2. Circuito com 2 Variáveis

- A imagem abaixo representa uma situação onde temos a rua A (preferencial) com um semáforo, e uma rua B que também possui um semáforo.



2. Circuito com 2 Variáveis

- Para gerencia a tráfego de carros, vamos elaborar um sistema de controle para o semáforo com as seguintes características:

1ª - Quando houver carros transitando somente na Rua B, o semáforo 2 deverá permanecer verde para que estas viaturas possam trafegar livremente.

2ª - Quando houver carros transitando somente na Rua A, o semáforo 1 deverá permanecer verde pelo mesmo motivo.

3ª - Quando houver carros transitando nas Ruas A e B, deveremos abrir o semáforo para a Rua A, pois é preferencial.

2. Circuito com 2 Variáveis

- Agora vamos estabelecer algumas convenções, para que posteriormente possa ser montada uma tabela verdade:

a) Existência de carro na Rua A: $A = 1$.

b) Não existência de carro na Rua A: $A = 0$ ou $\overline{A} = 1$.

c) Existência de carro na Rua B: $B = 1$.

d) Não existência de carro na Rua B: $B = 0$ ou $\overline{B} = 1$.

e) Verde do sinal 1 aceso: $V_1 = 1$.

f) Verde do sinal 2 aceso: $V_2 = 1$.

g) Quando $V_1 = 1 \rightarrow$ vermelho do semáforo 1 apagado: $V_{m1} = 0$,
verde do semáforo 2 apagado: $V_2 = 0$
e vermelho do semáforo 2 aceso: $V_{m2} = 1$.

h) Quando $V_2 = 1 \rightarrow V_1 = 0$, $V_{m2} = 0$ e $V_{m1} = 1$.

2. Circuito com 2 Variáveis

- Agora vamos montar a tabela verdade:

Situação	A	B	V_1	V_{m1}	V_2	V_{m2}
0	0	0				
1	0	1				
2	1	0				
3	1	1				

A situação 0 (A = 0 e B = 0) representa a ausência de veículos em ambas as ruas. Se não temos carros, tanto faz qual sinal permanece aberto. Vamos adotar, por exemplo, que o verde do sinal 2 parmaneça aceso. Neste caso, preenchemos a tabela da verdade da seguinte maneira:

Situação	A	B	V_1	V_{m1}	V_2	V_{m2}
0	0	0	0	1	1	0

($V_2 = 1 \rightarrow V_1 = 0$, $V_{m1} = 1$ e $V_{m2} = 0$)

2. Circuito com 2 Variáveis

- Continuando a montagem da tabela verdade:

A situação 1 ($A = 0$ e $B = 1$) representa a presença de veículo na Rua B e ausência de veículo na Rua A, logo, devemos acender o sinal verde para a Rua B ($V_2 = 1$). Temos, então:

Situação	A	B	V_1	V_{m1}	V_2	V_{m2}
1	0	1	0	1	1	0

($V_2 = 1 \rightarrow V_1 = 0$, $V_{m1} = 1$ e $V_{m2} = 0$)

A situação 2 ($A = 1$ e $B = 0$) representa a presença de veículo na Rua A e ausência de veículo na Rua B, logo, devemos acender o sinal verde para a Rua A ($V_1 = 1$). Temos, então:

Situação	A	B	V_1	V_{m1}	V_2	V_{m2}
2	1	0	1	0	0	1

($V_1 = 1 \rightarrow V_2 = 0$, $V_{m2} = 1$ e $V_{m1} = 0$)

2. Circuito com 2 Variáveis

- Continuando a montagem da tabela verdade:

A situação 3 (A = 1 e B = 1) representa a presença de veículos em ambas as ruas, logo, devemos acender o sinal verde para a Rua A, pois esta é preferencial. Temos, então:

Situação	A	B	V_1	V_{m1}	V_2	V_{m2}
3	1	1	1	0	0	1

($V_1 = 1 \rightarrow V_{m1} = 0$, $V_2 = 0$ e $V_{m2} = 1$)

A tabela totalmente preenchida é vista a seguir:

A	B	V_1	V_{m1}	V_2	V_{m2}
0	0	0	1	1	0
0	1	0	1	1	0
1	0	1	0	0	1
1	1	1	0	0	1

2. Circuito com 2 Variáveis

- Vamos transpor a tabela no mapa de Karnaugh:

Vamos transpor a tabela para diagramas de Veitch-Karnaugh e agrupar para obtermos as expressões simplificadas das saídas V_1, V_2, V_{m1} e V_{m2}:

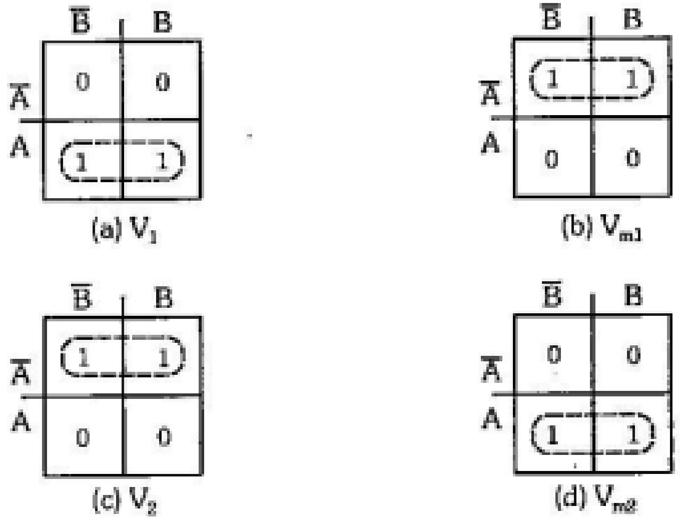


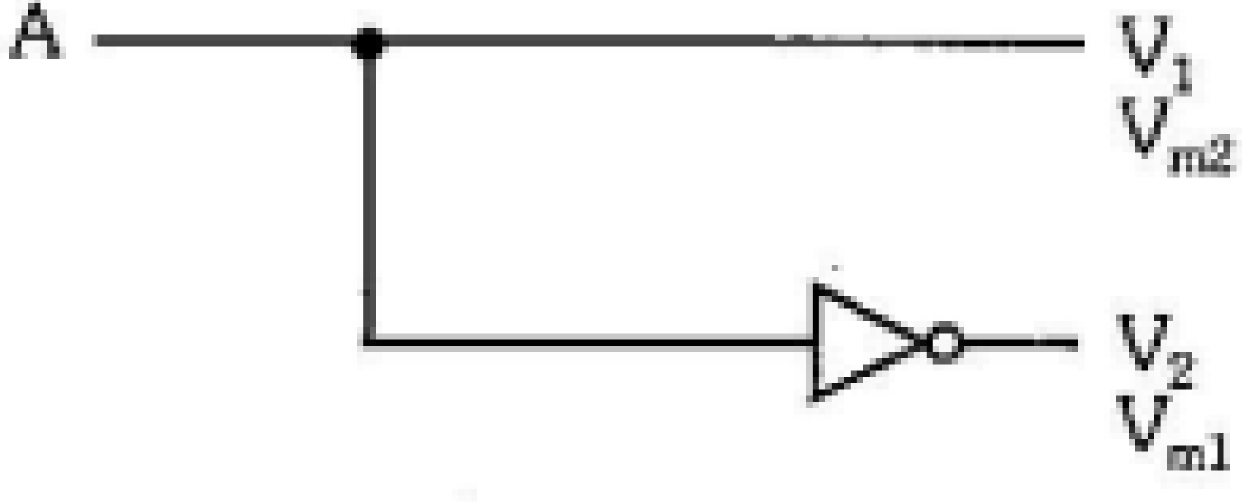
Figura 4.4

Pela tabela ou pelos diagramas, notamos que as expressões de V_1 e V_{m2} são idênticas, o mesmo ocorrendo com V_2 e V_{m1}. Assim sendo, as expressões simplificadas são:

$$V_1 = V_{m2} = A \quad \text{e} \quad V_2 = V_{m1} = \overline{A}$$

2. Circuito com 2 Variáveis

- A imagem abaixo mostra como podemos representar o circuito para atender os semáforos:



3. Circuito com 3 Variáveis

- Agora vamos analisar uma situação onde existem 3 variáveis:

Deseja-se utilizar um amplificador para ligar três aparelhos: um toca-fitas, um toca-discos e um rádio FM. Vamos elaborar um circuito lógico que nos permitirá ligar os aparelhos, obedecendo às seguintes prioridades:

1ª prioridade: Toca-discos

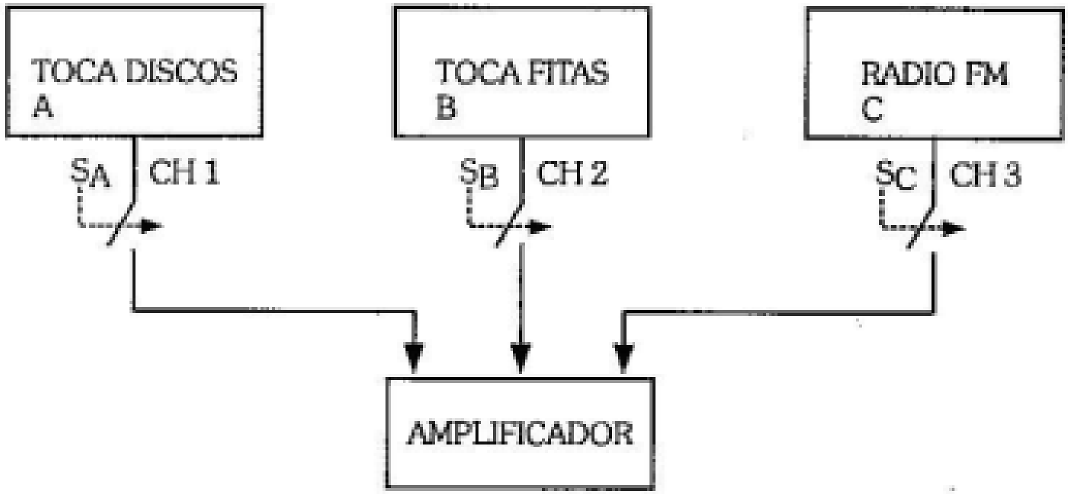
2ª prioridade: Toca-fitas

3ª prioridade: Rádio FM

Isto significa que quando não ligarmos nem o toca-discos, nem o toca-fitas, o rádio FM, se ligado, será conectado à entrada do amplificador. Se ligarmos o toca-fitas, automaticamente o circuito conectá-lo-á à entrada do amplificador, pois possui prioridade sobre o rádio FM. Se, então, ligarmos o toca-discos, este será conectado ao amplificador, pois representa a 1ª prioridade. A partir disto, podemos montar o diagrama de blocos com as respectivas ligações:

3. Circuito com 3 Variáveis

- A imagem mostra um esboço de como o projeto deve ficar:



Neste projeto, o circuito lógico receberá as informações das variáveis de entrada A, B e C, representando os aparelhos, e através das saídas S_A, S_B e S_C comutará as chaves CH1, CH2 e CH3 para fazer a conexão conforme a situação requerida.

3. Circuito com 3 Variáveis

- O primeiro passo é estabelecer as convenções e estruturar a tabela verdade:

Convenções Utilizadas:

⇨ Variáveis de entrada (A, B e C): aparelho desligado = 0 e ligado = 1.

⇨ Saídas (S_A, S_B e S_C): S = 0 → chave aberta e S = 1 → chave fechada.

Tabela da Verdade

Situação	A	B	C	S_A	S_B	S_C
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			

3. Circuito com 3 Variáveis

- Agora vamos analisar cada uma das oito situações possíveis:

			S_A	S_B	S_C
Caso 0 -	Os 3 estão desligados, logo, condição irrelevante, pois não importa qual chave dever ser ligada.	⇨	X	X	X
Caso 1 -	Está ligado apenas o FM, logo somente S_C assume valor 1.	⇨	0	0	1
Caso 2 -	Está ligado apenas o toca-fitas, logo somente S_B assume valor 1.	⇨	0	1	0
Caso 3 -	Estão ligados o FM e o toca-fitas. O toca-fitas tem prioridade sobre o FM, logo somente S_B assume valor 1.	⇨	0	1	0
Caso 4 -	Está ligado apenas o toca-discos, logo somente o S_A assume o valor 1.	⇨	1	0	0
Caso 5 -	Estão ligados o toca-discos e o FM. O toca-discos é a 1ª prioridade, logo somente S_A assume valor 1.	⇨	1	0	0
Caso 6 -	Análogo ao caso 5.	⇨	1	0	0
Caso 7 -	Análogo aos casos 5 e 6.	⇨	1	0	0

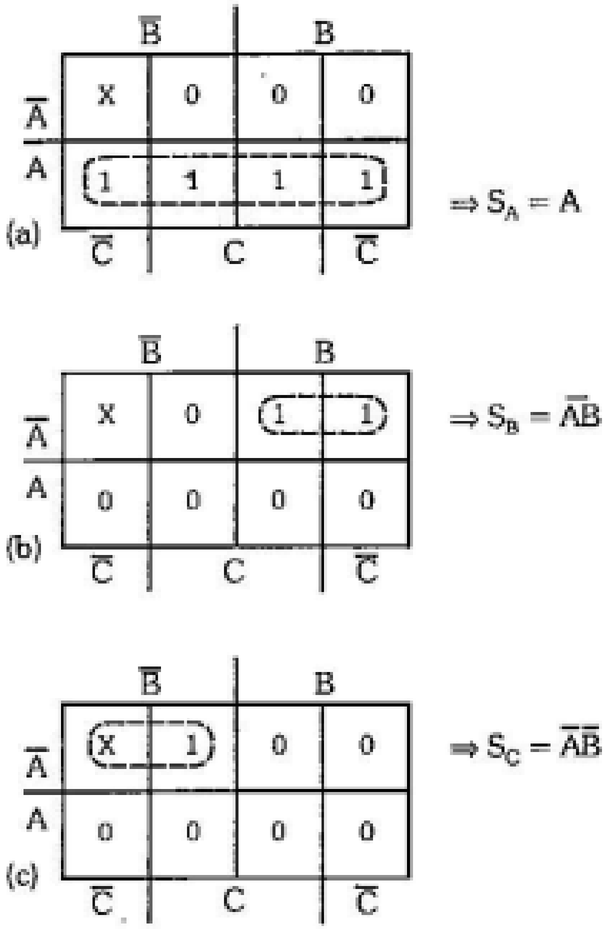
3. Circuito com 3 Variáveis

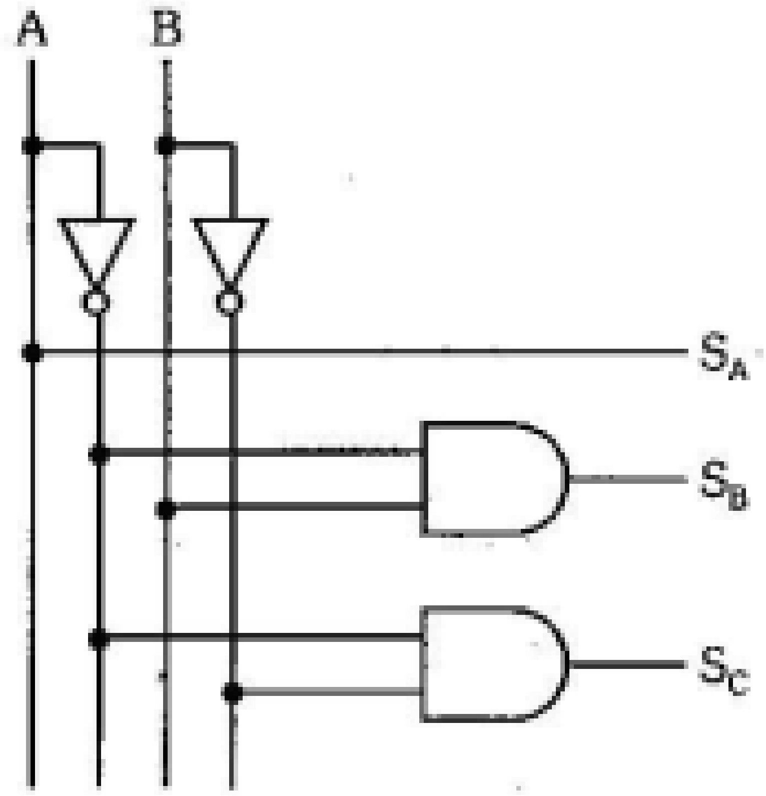
- A tabela verdade ficou da seguinte forma:

Situação	A	B	C	S_A	S_B	S_C
0	0	0	0	X	X	X
1	0	0	1	0	0	1
2	0	1	0	0	1	0
3	0	1	1	0	1	0
4	1	0	0	1	0	0
5	1	0	1	1	0	0
6	1	1	0	1	0	0
7	1	1	1	1	0	0

3. Circuito com 3 Variáveis

- Montando o mapa de Karnaugh e o circuito:





4. Circuito com 4 Variáveis

- Como exemplos, vamos supor que uma empresa tem um sistema de comunicação interna. E a prioridade de chamada no intercomunicador é mostrada na imagem abaixo:

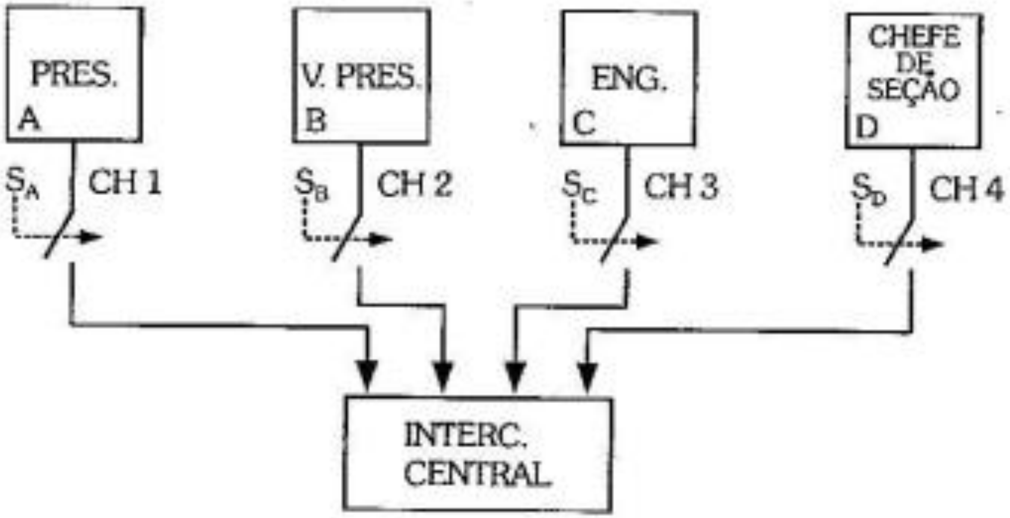
Presidente: 1ª prioridade

Vice-presidente: 2ª prioridade

Engenharia: 3ª prioridade

Chefe de seção: 4ª prioridade

Esquematicamente, temos:



4. Circuito com 4 Variáveis

- Vamos estabelecer as variáveis:

Variáveis de entrada:

- ⇨ intercomunicador do presidente: A
- ⇨ intercomunicador do vice-presidente: B
- ⇨ intercomunicador da engenharia: C
- ⇨ intercomunicador do chefe de seção: D

Convenções utilizadas:

- ⇨ presença de chamada: 1
- ⇨ ausência de chamada: 0

Saídas: S_A, S_B, S_C e S_D

Convenções utilizadas:

- ⇨ efetivação de chamada: 1
- ⇨ não efetivação de chamada: 0

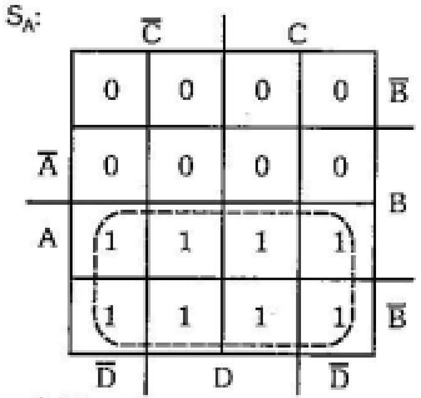
4. Circuito com 4 Variáveis

- Analisando as situações e montando a tabela verdade:

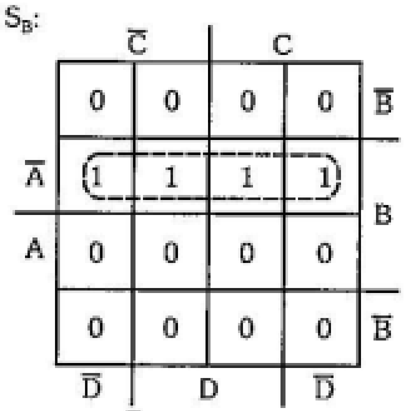
A	B	C	D	S_A	S_B	S_C	S_D	
0	0	0	0	0	0	0	0	→ não efetua chamada.
0	0	0	1	0	0	0	1	→ efetua chamada do chefe de seção.
0	0	1	0	0	0	1	0	→ efetua chamada da engenharia.
0	0	1	1	0	0	1	0	→ efetua chamada da engenharia, pois é prioritária.
0	1	0	0	0	1	0	0	→ efetua chamada do vice-presidente.
0	1	0	1	0	1	0	0	
0	1	1	0	0	1	0	0	} → efetua chamada do vice-presidente, pois é prioritário.
0	1	1	1	0	1	0	0	
1	0	0	0	1	0	0	0	→ efetua chamada do presidente.
1	0	0	1	1	0	0	0	
1	0	1	0	1	0	0	0	
1	0	1	1	1	0	0	0	
1	1	0	0	1	0	0	0	} → efetua chamada do presidente, pois é a 1ª prioridade.
1	1	0	1	1	0	0	0	
1	1	1	0	1	0	0	0	
1	1	1	1	1	0	0	0	

4. Circuito com 4 Variáveis

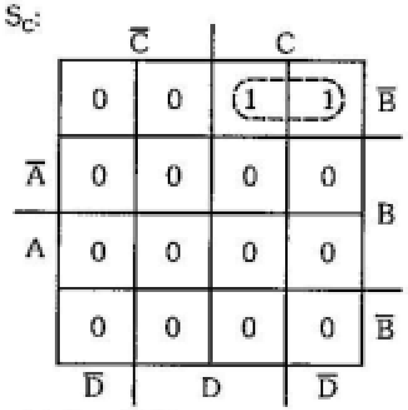
- Montando o mapa:



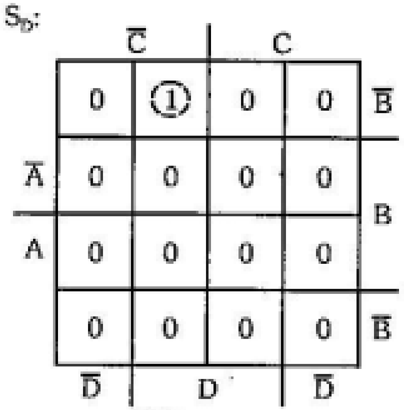
(a) $S_A = A$



(b) $S_B = \bar{A}B$



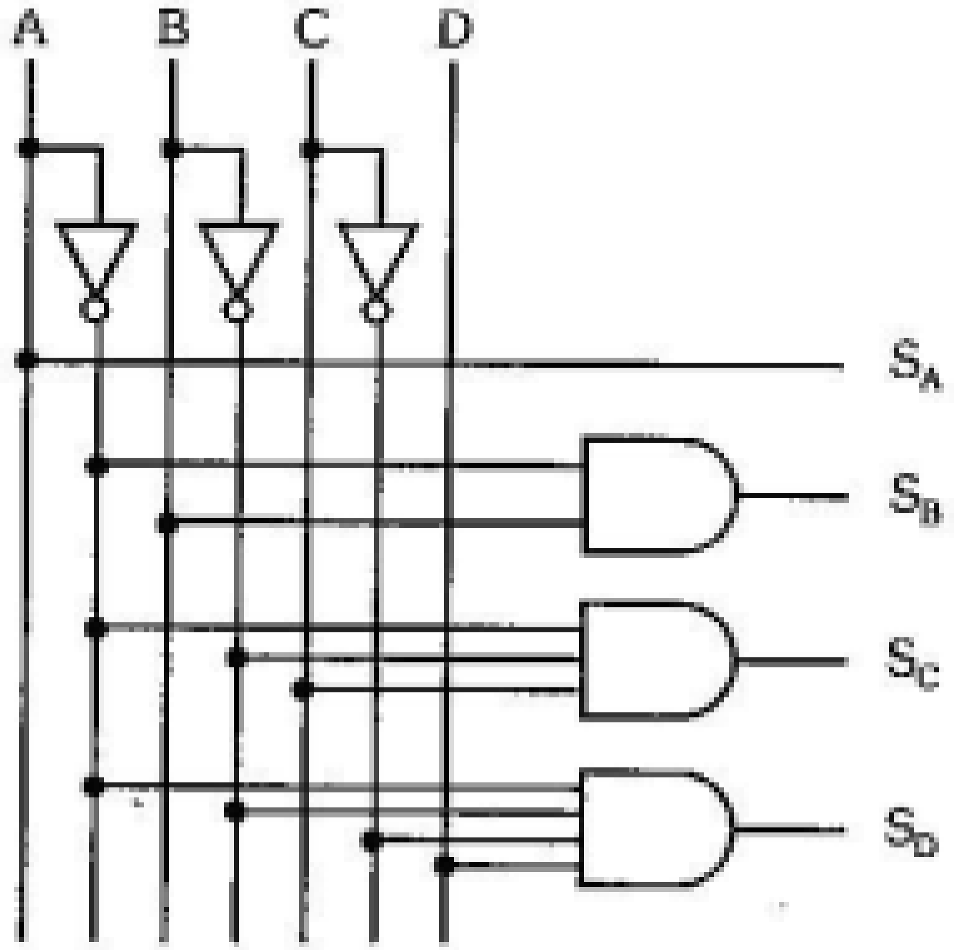
(c) $S_C = \bar{A}\bar{B}C$



(d) $S_D = \bar{A}\bar{B}\bar{C}D$

4. Circuito com 4 Variáveis

- Montando o circuito:

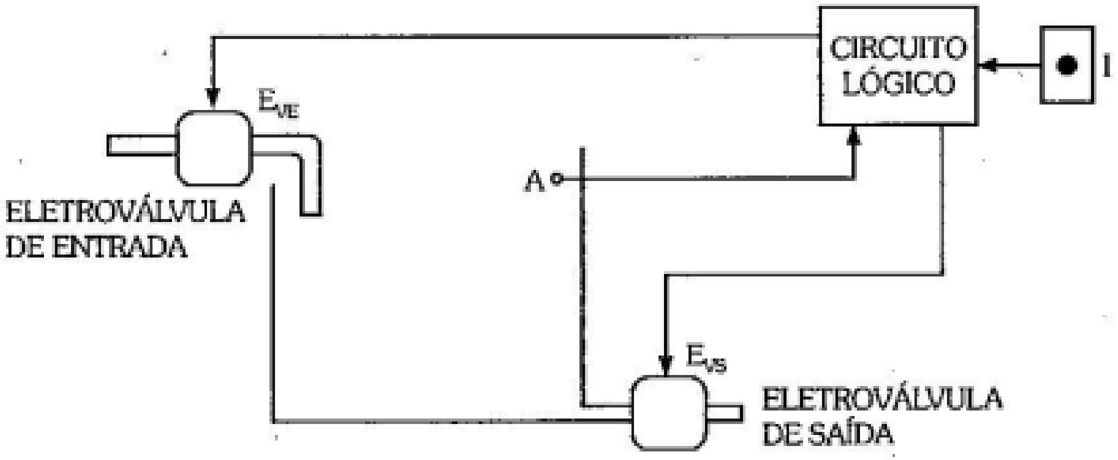


4. Circuito com 4 Variáveis

- Exercício:

1 - Elabore um circuito lógico para encher ou esvaziar um tanque industrial por meio de duas eletroválvulas, sendo uma para a entrada do líquido e outra para o escoamento de saída. O circuito lógico, através da informação de um sensor de nível máximo no tanque e de um botão interruptor de duas posições, deve atuar nas eletroválvulas para encher o tanque totalmente (botão ativado) ou, ainda, esvaziá-lo totalmente (botão desativado).

Para solucionar, vamos traçar o esquema de ligação, determinar e convencionar as variáveis de entrada e saída do circuito lógico. Este esquema é visto na figura 4.12.



4. Circuito com 4 Variáveis

- Resposta:

Variáveis de entrada: sensor de líquido A e botão interruptor I.

Variáveis de saída: eletroválvulas E_{VE} e E_{VS}.

Convenções:

Sensor A:

- ⇨ presença de água = nível 1
- ⇨ ausência de água = nível 0

Interruptor I:

- ⇨ ativado = nível 1
- ⇨ desativado = nível 0

Eletroválvulas E_{VE} e E_{VS}:

- ⇨ ligada = nível 1
- ⇨ desligada = nível 0

4. Circuito com 4 Variáveis

- Resposta: Analisando todos os casos possíveis

Caso: $I = 0$ e $A = 0$ ⇨ O caso representa o botão desativado e a ausência de líquido no sensor. O circuito não deve ligar a eletroválvula de entrada ($E_{VE} = 0$), mas deve ligar a eletroválvula de saída ($E_{VS} = 1$), para o total escoamento do líquido remanescente abaixo do nível do sensor.

Caso: $I = 0$ e $A = 1$ ⇨ O caso representa o botão desativado para esvaziamento do tanque e a presença de líquido no sensor. O circuito deve ligar apenas a eletroválvula de saída ($E_{VE} = 0$ e $E_{VS} = 1$)

Caso: $I = 1$ e $A = 0$ ⇨ Representa o botão ativado para encher o tanque, não havendo presença de líquido no sensor. O circuito deve ligar apenas a eletroválvula de entrada ($E_{VE} = 1$ e $E_{VS} = 0$)

Caso: $I = 1$ e $A = 1$ ⇨ Representa o tanque cheio e o botão ativado. Nenhuma das eletroválvulas devem ser ligadas ($E_{VE} = 0$ e $E_{VS} = 0$).

4. Circuito com 4 Variáveis

- Resposta: Tabela verdade e mapa

I	A	E_{VE}	E_{VS}
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

Para simplificar a saída E_{VE}, não necessitamos do diagrama de Veitch-Karnaugh, pois teríamos apenas um termo isolado, sendo de qualquer maneira, a expressão simplificada: $E_{VE} = I\overline{A}$.

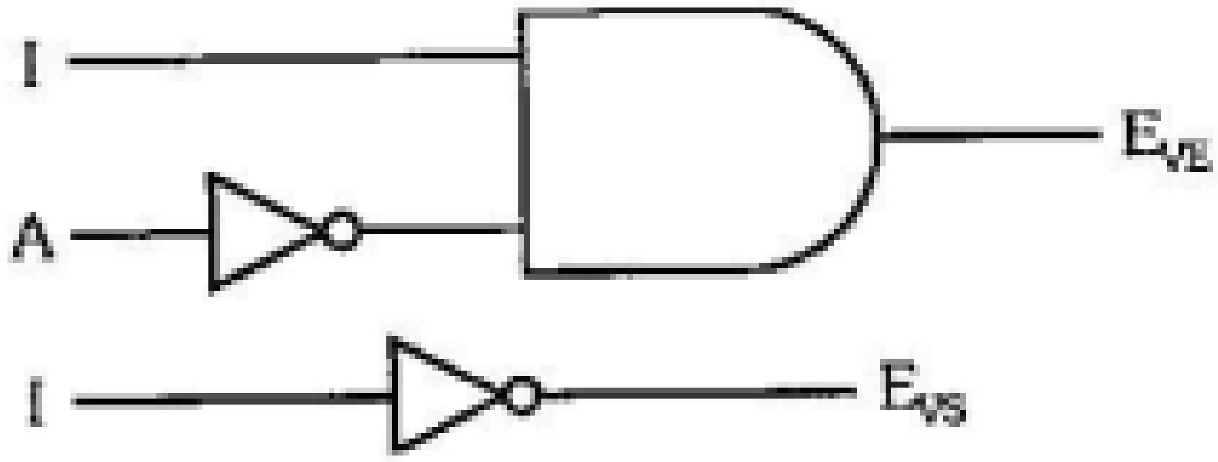
O mapa para a simplificação da saída E_{VS} é visto na figura 4.13.

	$\overline{A}$	A
$\overline{I}$	1	1
I	0	0

$\Rightarrow E_{vs} = \overline{I}$

4. Circuito com 4 Variáveis

- Resposta: Montando o circuito



5. Circuitos Combinacionais para Computação

- Os exemplos anteriores tiveram como objetivo mostrar circuitos para automação de problemas gerais.

- Vamos analisar agora como elaborar circuitos combinacionais de uso mais específicos para computação, como codificadores, somadores, subtratores, etc.

- Geralmente, este tipo de circuitos são usados dentro de circuitos maiores, como microcontroladores, processadores, etc.

6. Padrões de Codificação

- Dentro da arquitetura computacional existem diversos tipos de padrões de codificação. Para cada situação existe o padrão mais adequado.

- Os padrões mais conhecidos são:

1. BCD 8421
2. Excesso de 3
3. Gray

7. Codigo BCD 8421

- Este é um dos padrões mais conhecidos de codificação.

Decimal	BCD 8421			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Vamos iniciar explicando que no nome deste código, a sigla BCD representa as iniciais de Binary Coded Decimal, que significa uma codificação do sistema decimal em binário. Os termos seguintes (8421) significam os valores dos algarismos num dado número binário, que conforme estudado no capítulo 1, representam respectivamente: 2^3, 2^2, 2^1 e 2^0.

O número de bits de um código é o número de dígitos binários que este possui. Notamos, então, que o código BCD 8421 é um código de 4 bits e, ainda, que é válido de 0_{10} a 9_{10}.

8. Código Excesso 3

- É a transformação do número decimal no binário correspondente, somando-se 3 unidades. Por exemplo, zero em binário (0000) é 0011 no Excesso 3. Ele é usado em alguns circuitos aritméticos.

Decimal	Excesso 3			
	A	B	C	D
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

9. Código Gray

- Sua principal característica é que de um número para o outro apenas um bit irá variar:

Decimal	Gray			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1

10. Código 2 entre 5

- Neste código cada número possui sempre 2 bits iguais a 1, dentre 5 bits.

Decimal	2 entre 5				
	A	B	C	D	E
0	0	0	0	1	1
1	0	0	1	0	1
2	0	0	1	1	0
3	0	1	0	0	1
4	0	1	0	1	0
5	0	1	1	0	0
6	1	0	0	0	1
7	1	0	0	1	0
8	1	0	1	0	0
9	1	1	0	0	0

11. Código Johnson

- Este padrão de código é muito usado em contadores:

Decimal	Johnson				
	A	B	C	D	E
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	1	1
4	0	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	0
7	1	1	1	0	0
8	1	1	0	0	0
9	1	0	0	0	0

12. Codificadores e Decodificadores

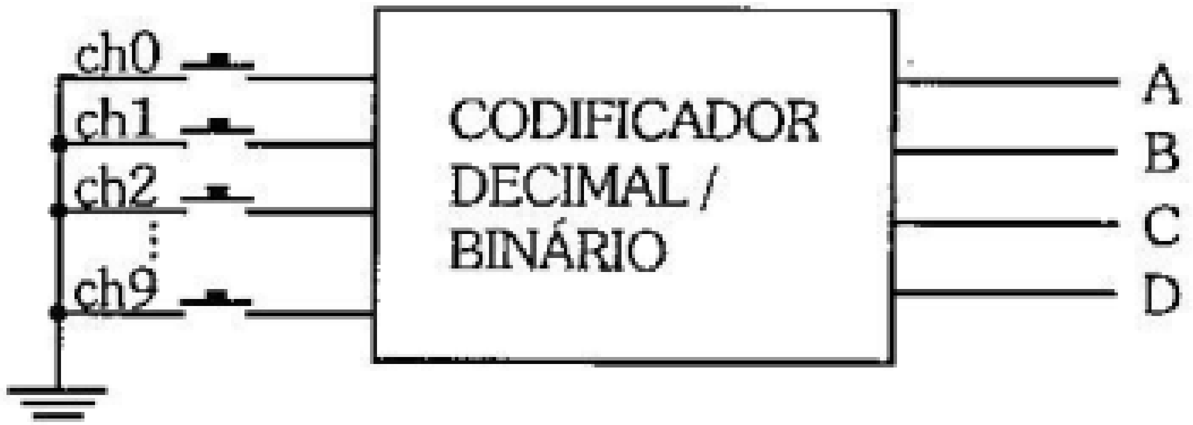
- **Codificador** é o circuito combinacional que torna possível passar um código conhecido para um desconhecido.

- Um exemplo seria uma calculadora que recebe um número decimal (conhecido) e codifica para um número binário (desconhecido).

- Já o **decodificador** é o processo contrário, que transforma um valor desconhecido para conhecido.

13. Codificador Decimal/Binário

- Vamos elaborar um codificador para transformar um código decimal em binário (padrão BCD8421). A entrada do código decimal será feito por meio de chaves numeradas de 0 até 9. A saída será por 4 fios para retornar o binário correspondente.



13. Codificador Decimal/Binário

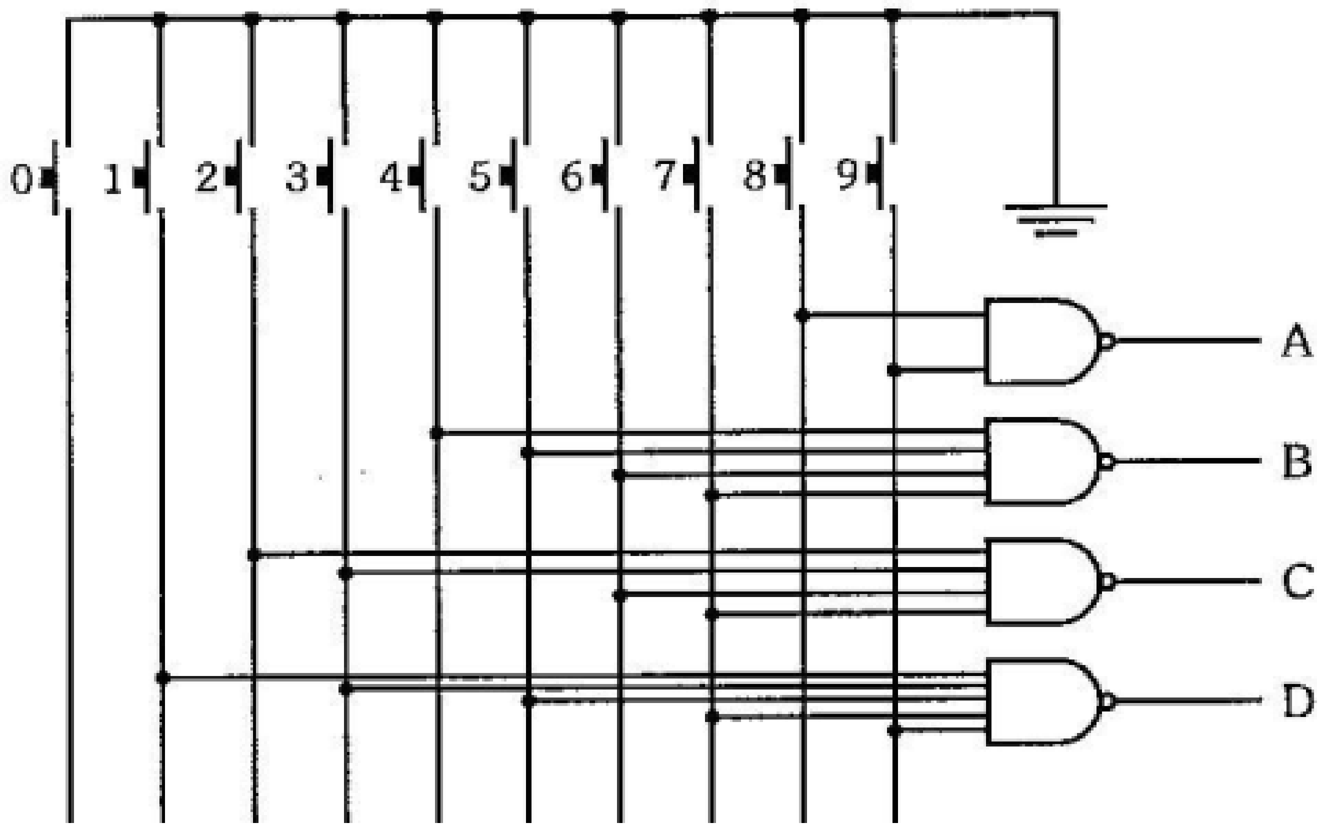
- O primeiro passo é construir a tabela verdade do codificador que relaciona decimal com seu respectivo binário:

Chave	A	B	C	D
Ch0	0	0	0	0
Ch1	0	0	0	1
Ch2	0	0	1	0
Ch3	0	0	1	1
Ch4	0	1	0	0
Ch5	0	1	0	1
Ch6	0	1	1	0
Ch7	0	1	1	1
Ch8	1	0	0	0
Ch9	1	0	0	1

Através da tabela, concluímos que a saída A valerá 1 quando Ch8 ou Ch9 for acionada. A saída B quando Ch4, Ch5, Ch6 ou Ch7 for acionada. A saída C quando Ch2, Ch3, Ch6 ou Ch7 for acionada. A saída D quando Ch1, Ch3, Ch5, Ch7 ou Ch9 for acionada.

13. Codificador Decimal/Binário

- Para montar o circuito usaremos NAND em cada saída, pois fornece valor 1 quando qualquer das portas tiver entrada 0:



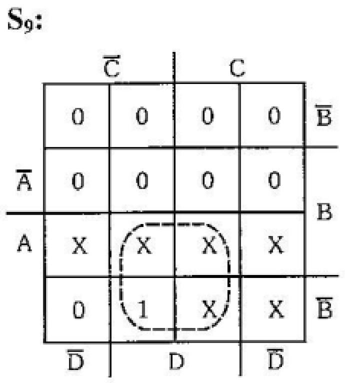
14. Decodificador Binário/Decimal

- Agora o objetivo é montar um circuito que receba um binário e converta para decimal. O primeiro passo é montar a tabela verdade:

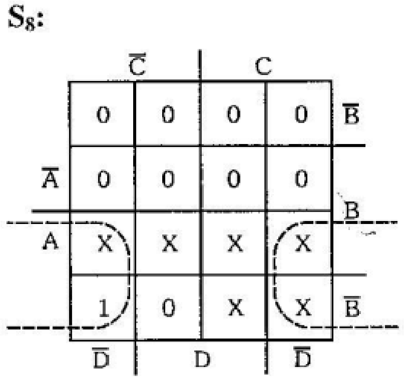
BCD 8421				Código 9876543210									
A	B	C	D	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0

14. Decodificador Binário/Decimal

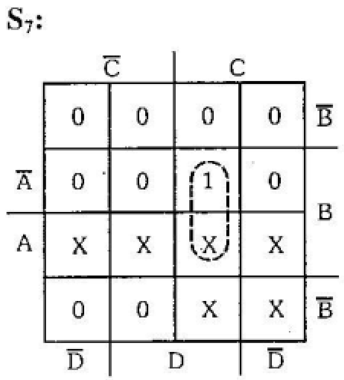
- Agora montaremos um mapa de Karnaugh para cada saída. O código BCD8421 não possui números maiores que 9, logo no mapa esses valores excedentes podem ser marcados com x:



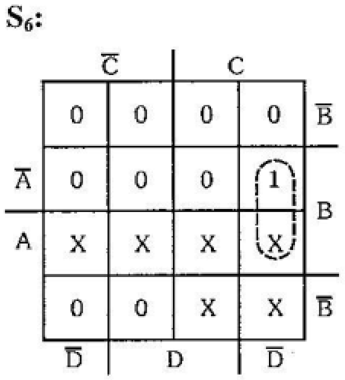
(a) $S_9 = AD$



(b) $S_8 = A\overline{D}$



(c) $S_7 = BCD$

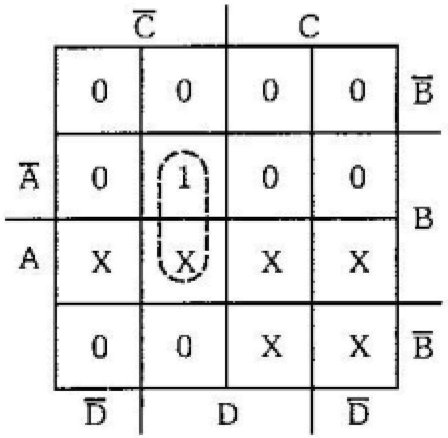


(d) $S_6 = BC\overline{D}$

14. Decodificador Binário/Decimal

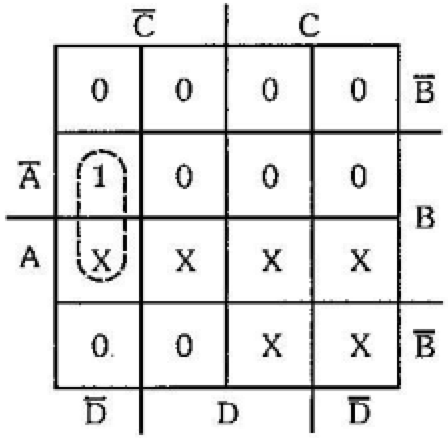
- Montando o mapa de Karnaugh para demais saídas.

S_5:



(e) $S_5 = B\overline{C}D$

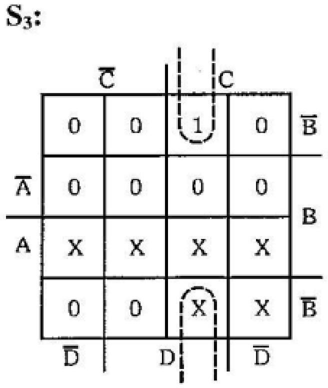
S_4:



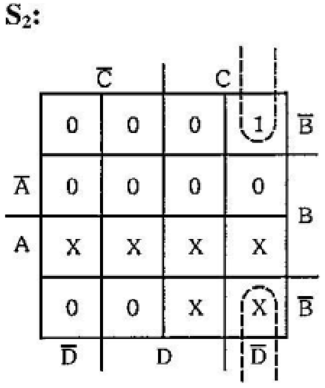
(f) $S_4 = B\overline{C}\,\overline{D}$

14. Decodificador Binário/Decimal

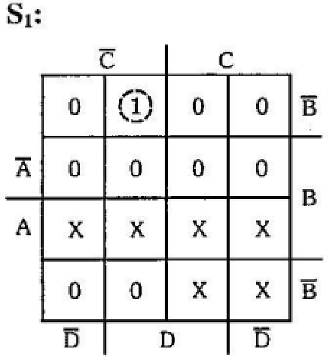
- Montando o mapa de Karnaugh para demais saídas.



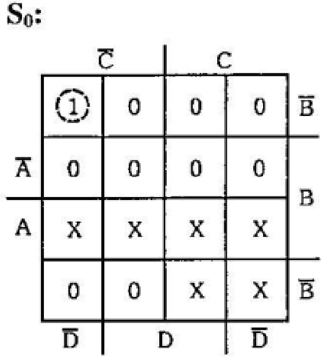
(g) $S_3 = \overline{B}CD$



(h) $S_2 = \overline{B}C\overline{D}$



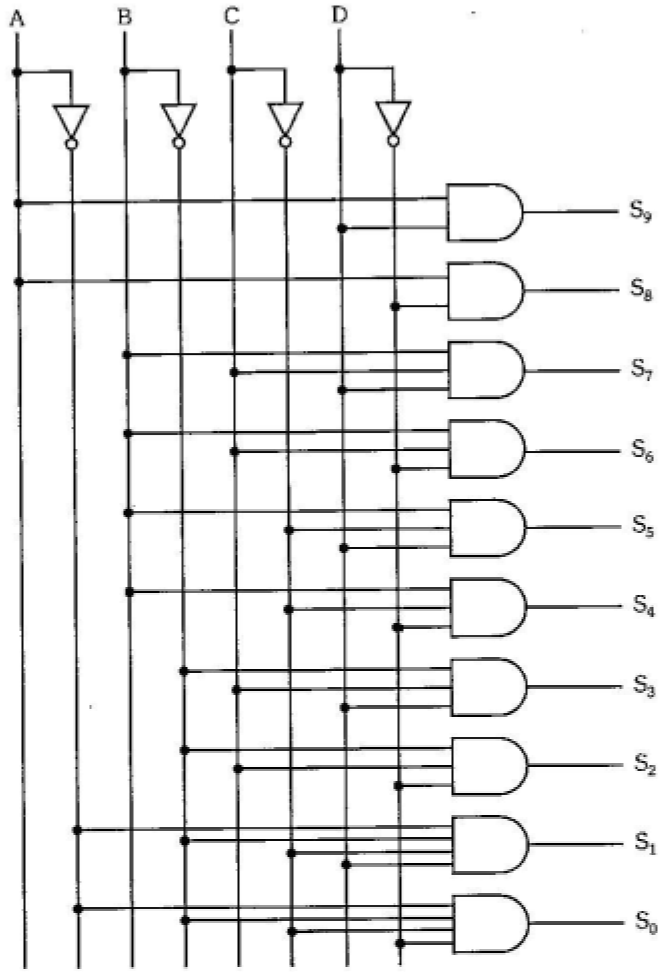
(i) $S_1 = \overline{A}\,\overline{B}\,\overline{C}D$



(j) $S_0 = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$

14. Decodificador Binário/Decimal

- A partir das expressões, montamos o circuito:



15. Projeto de Decodificadores

- Seguindo esta linha de raciocínio, é possível construir decodificadores de qualquer padrão de código para qualquer outro. Como exemplo, vamos montar um decodificador de BCD8421 para excesso de 3. O primeiro passo é montar a tabela:

BCD 8421				Excesso 3			
A	B	C	D	S_3	S_2	S_1	S_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

15. Projeto de Decodificadores

- Montamos o mapa para construção do circuito:

S_3:

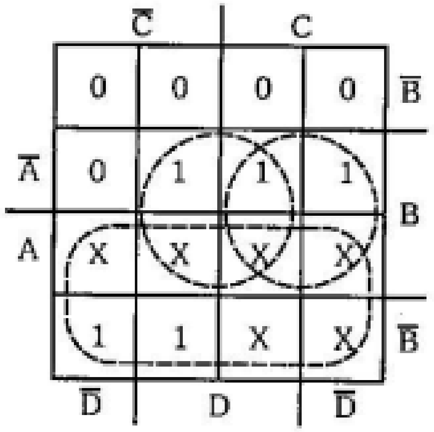


Figura 5.8

Agrupamentos: 1 oitava A e

2 quadras BD e BC,

$\therefore S_3 = A + BD + BC$

S_2:

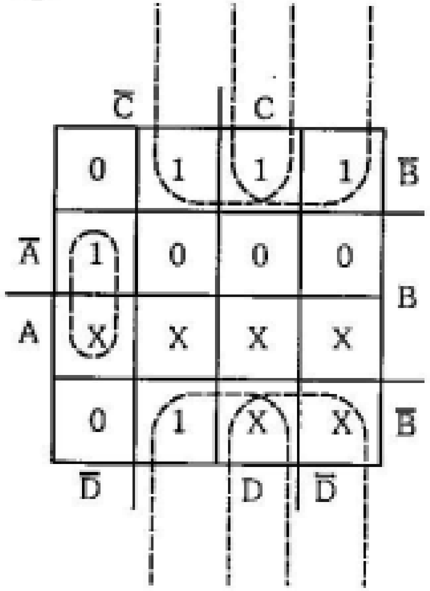


Figura 5.9

Agrupamentos: 2 quadras $\overline{B}D$, $\overline{B}C$

e 1 par $B\overline{CD}$,

$\therefore S_2 = \overline{B}D + \overline{B}C + B\overline{CD}$

15. Projeto de Decodificadores

- Montamos o mapa para construção do circuito:

S_1:

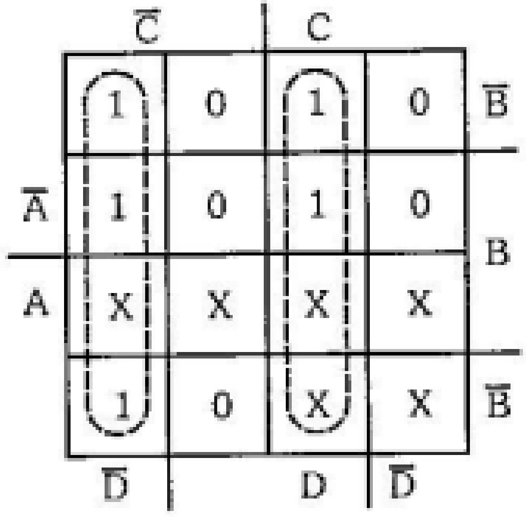


Figura 5.10

Agrupamentos: 2 quadras $\overline{CD}$ e CD.

$\therefore S_1 = \overline{CD} + CD$ ou $S_1 = C \odot D$

S_0:

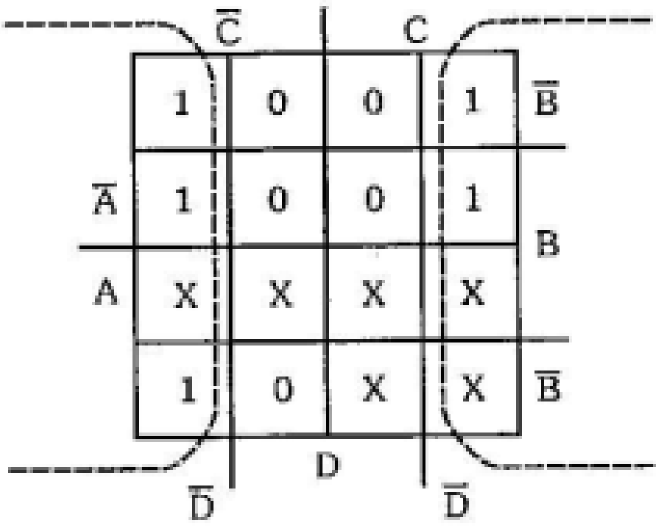


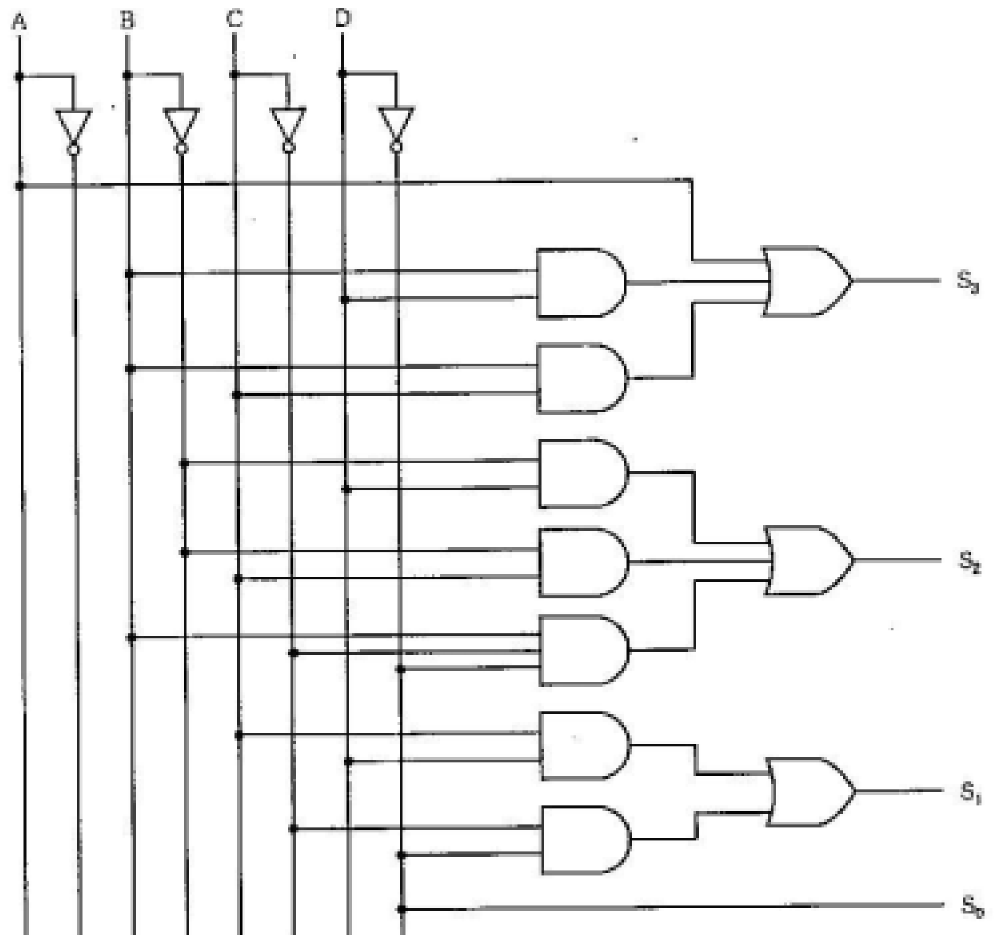
Figura 5.11

Agrupamento: 1 oitava $\overline{D}$.

$\therefore S_0 = \overline{D}$

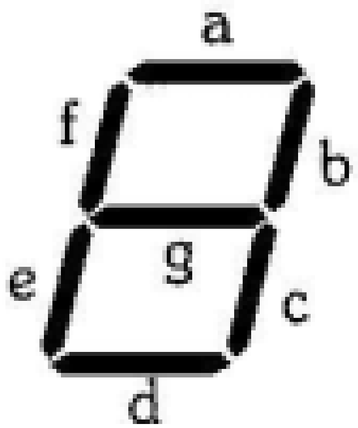
15. Projeto de Decodificadores

- Montamos o circuito:



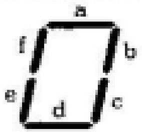
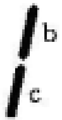
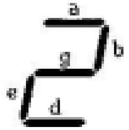
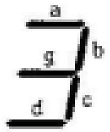
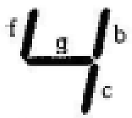
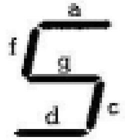
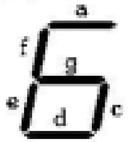
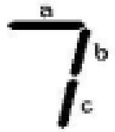
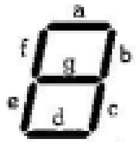
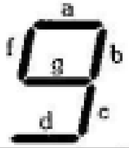
16. Decodificador para Display de 7 Segmentos

- O display de 7 segmentos possibilita representar números decimais de 0 até 9 por meio de um led. O objetivo é montar um circuito onde o usuário entra com um número binário, e este é convertido para decimal e representado no display. O primeiro passo é montar a tabela:



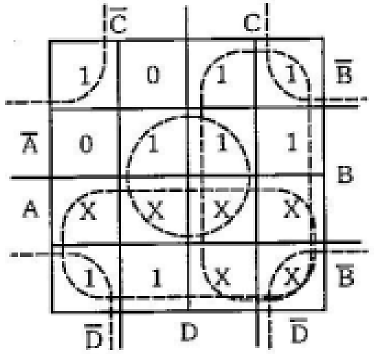
16. Decodificador para Display de 7 Segmentos

- O primeiro passo é montar a tabela:

Characteres	Display	BCD 8421				Código para 7 Segmentos						
		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	1	0	1	1

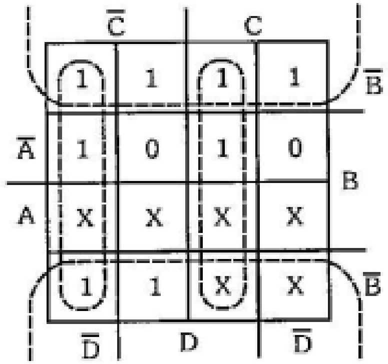
16. Decodificador para Display de 7 Segmentos

- Montamos o mapa de Karnaugh da tabela:



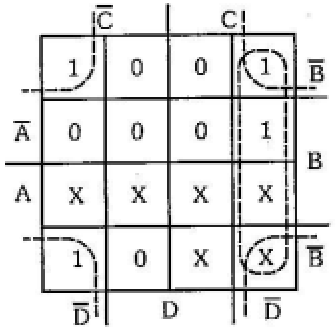
(a) $a = A + C + BD + \overline{B}\,\overline{D}$

ou $a = A + C + B \odot D$

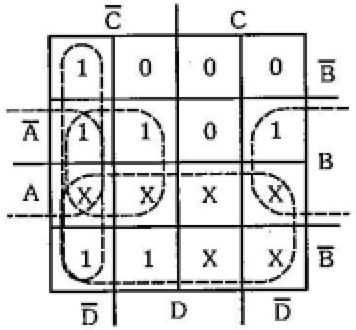


(b) $b = \overline{B} + \overline{C}\,\overline{D} + CD$

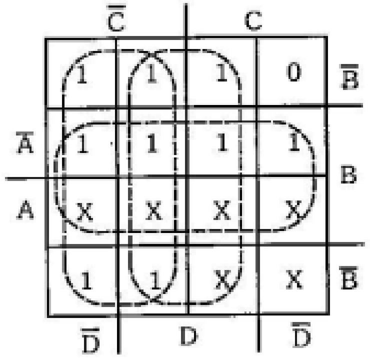
ou $b = \overline{B} + C \odot D$



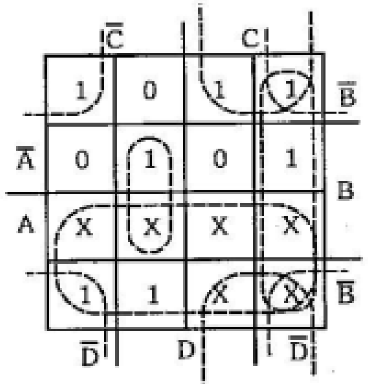
(e) $e = \overline{B}\,\overline{D} + C\overline{D}$



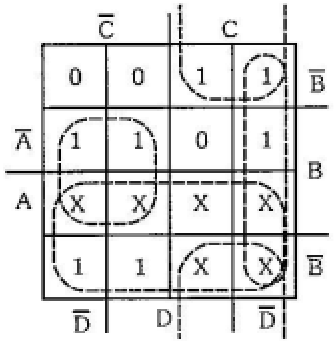
(f) $f = A + \overline{C}\,\overline{D} + B\overline{C} + B\overline{D}$



(c) $c = B + \overline{C} + D$



(d) $d = A + \overline{B}\,\overline{D} + \overline{B}C + C\overline{D} + B\overline{C}D$

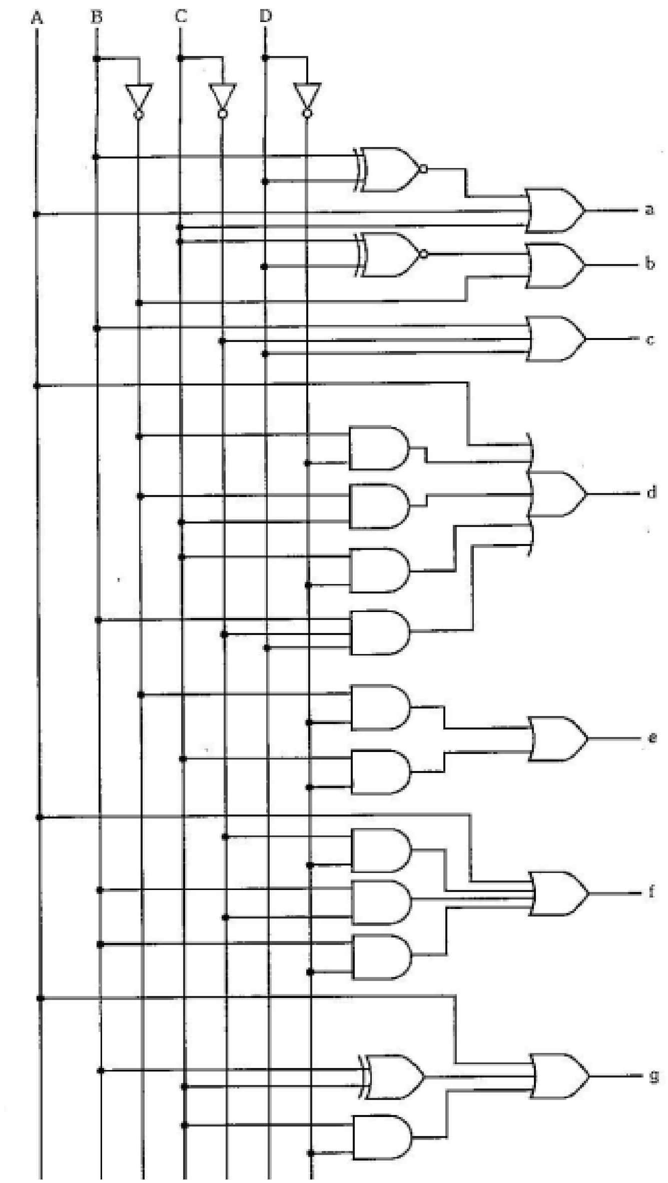


(g) $g = A + B\overline{C} + \overline{B}C + C\overline{D}$

ou $g = A + B \oplus C + C\overline{D}$

16. Decodificador para Display de 7 Segmentos

- Montamos o circuito:



17. Circuitos Aritméticos

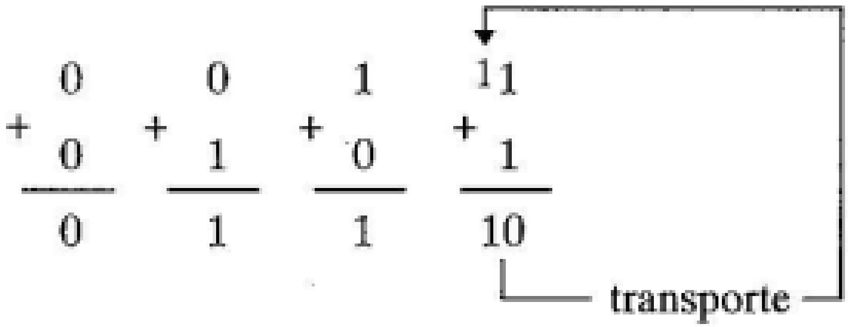
- Dentro do conjunto de circuitos combinacionais aplicados a computação, temos os circuitos aritméticos. Eles são utilizados, principalmente para construir a ULA dos processadores.

- Estes circuitos também podem ser encontrados em circuitos integrados comerciais, onde em conjunto com um microcontrolador podem fazer cálculos específicos.

- Temos circuitos somadores, subtratores, contadores, entre outros.

18. Circuito Meio Somador

- Antes de entender o funcionamento do circuito, vamos relembrar como funciona a soma binária, para em seguida analisar a tabela verdade da soma de 2 números:



A	B	S	Ts
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Ts → transporte de saída

(0 + 0 = 0 → Ts = 0)

(0 + 1 = 1 → Ts = 0)

(1 + 0 = 1 → Ts = 0)

(1 + 1 = 0 → Ts = 1)

18. Circuito Meio Somador

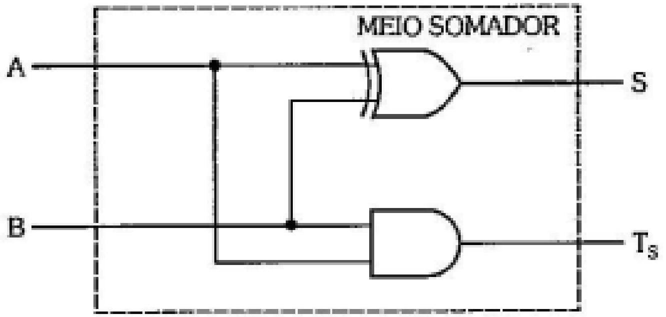
- A imagem mostra como podemos montar o circuito meio somador utilizando a porta XOR:

Representando cada número por 1 bit, podemos, então, montar um circuito que possui como entradas A e B, e como saída, a soma dos algarismos (S) e o respectivo transporte de saída (Ts). As expressões características do circuito, extraídas da tabela, são:

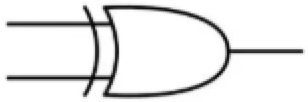
$$S = A \oplus B$$

$$Ts = AB$$

O circuito a partir destas expressões é visto na figura 5.31.



XOR

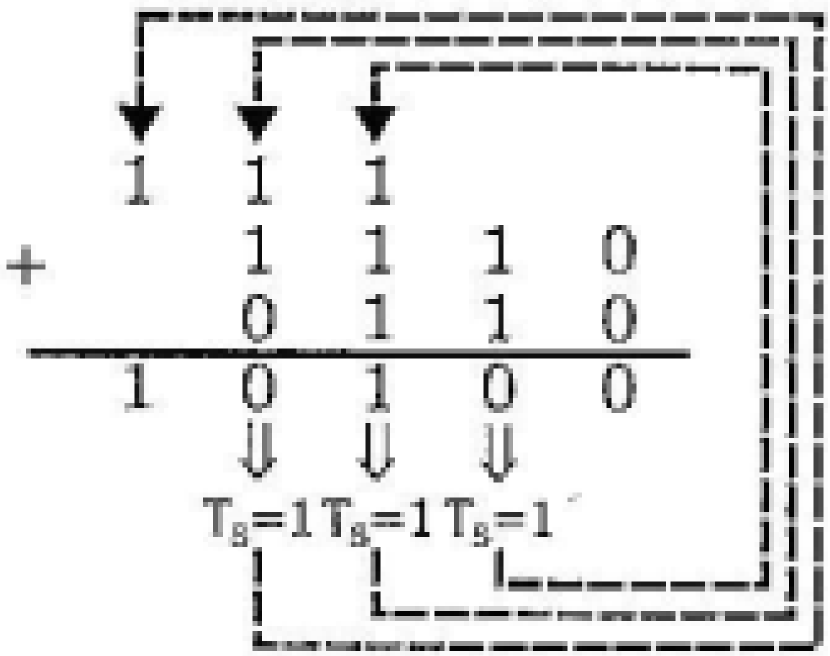


Gate	Notation	Truth table		
		INPUT		OUTPUT
		A	B	A XOR B
XOR	$A \oplus B$	0	0	0
		0	1	1
		1	0	1
		1	1	0

A representação em bloco deste circuito

19. Somador Completo

- O Meio Somador possibilita soma de apenas números binários com 1 algarismo. Para soma de mais algarismos, é preciso considerar o transporte da coluna anterior caso ele exista, como mostra o exemplo:



19. Somador Completo

- Para resolver este problema, vamos usar o somador completo. A imagem abaixo mostra a tabela verdade de um somador completo. A sigla TE representa "Transporte de Entrada" e TS representa "Transporte de Saída".

O Somador Completo é um circuito para efetuar a soma completa de uma coluna, considerando o transporte de entrada. Vamos, agora, montar a tabela da verdade deste circuito:

A	B	T_E	S	T_S
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$T_E \rightarrow$ transporte de entrada

$(0 + 0 + 0 = 0 \rightarrow Ts = 0)$

$(0 + 0 + 1 = 1 \rightarrow Ts = 0)$

$(0 + 1 + 0 = 1 \rightarrow Ts = 0)$

$(0 + 1 + 1 = 0 \rightarrow Ts = 1)$

$(1 + 0 + 0 = 1 \rightarrow Ts = 0)$

$(1 + 0 + 1 = 0 \rightarrow Ts = 1)$

$(1 + 1 + 0 = 0 \rightarrow Ts = 1)$

$(1 + 1 + 1 = 1 \rightarrow Ts = 1)$

19. Somador Completo

- Representando a tabela no mapa:

S:

	$\overline{B}$		B	
$\overline{A}$	0	1	0	1
A	1	0	1	0
	$\overline{T_E}$	T_E		$\overline{T_E}$

Figura 5.33

Conforme já estudado, podemos escrever:

$S = A \oplus B \oplus T_E$

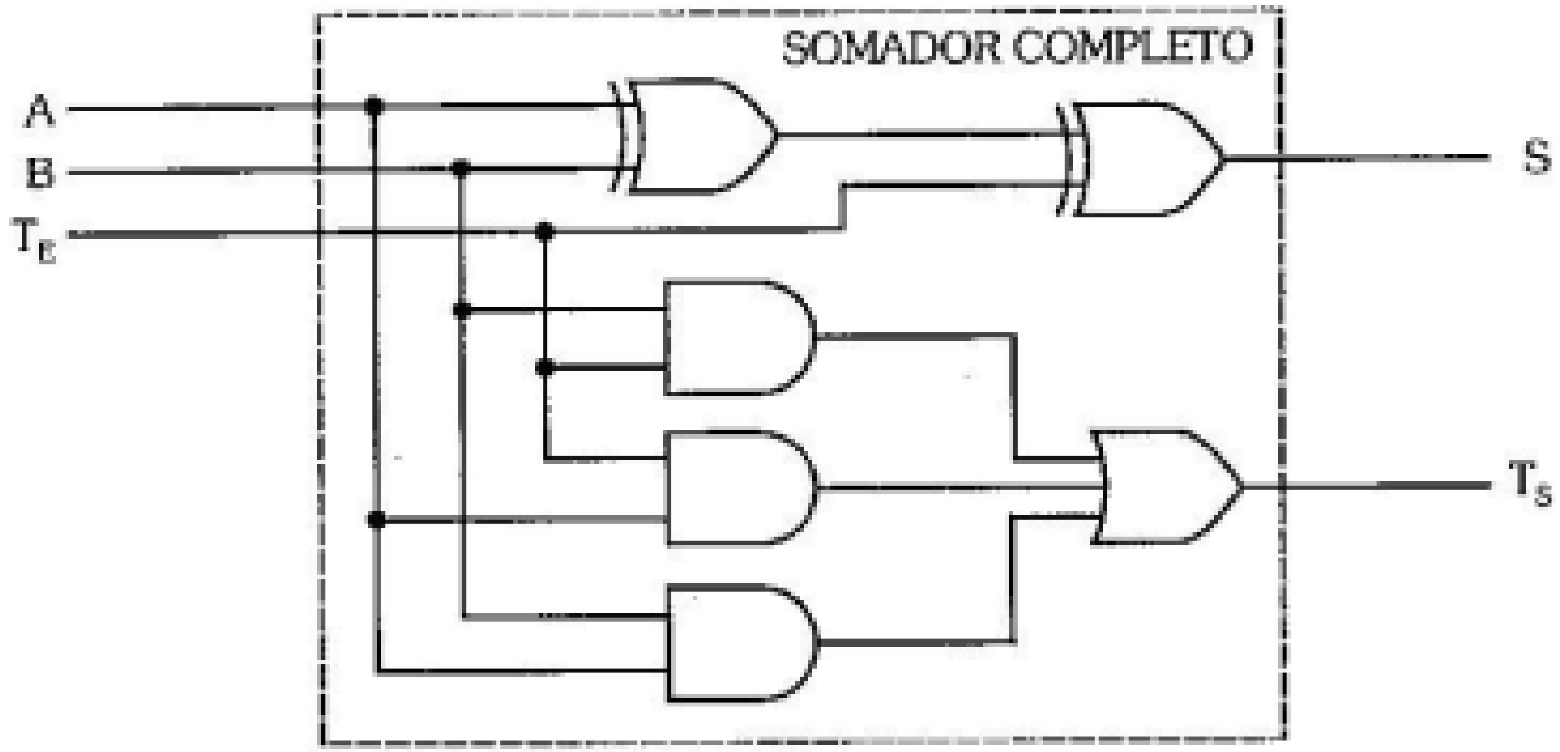
T_S:

	$\overline{B}$		B	
$\overline{A}$	0	0	1	0
A	0	1	1	1
	$\overline{T_E}$	T_E		$\overline{T_E}$

$T_S = BT_E + AT_E + AB$

19. Somador Completo

- Representando o circuito:



19. Somador Completo

- Formas de representar o circuito Somador Completo:

O circuito Somador Completo é também conhecido como **Full Adder**, sendo a entrada de transporte denominada **carry in**, ambos os termos derivados do inglês.

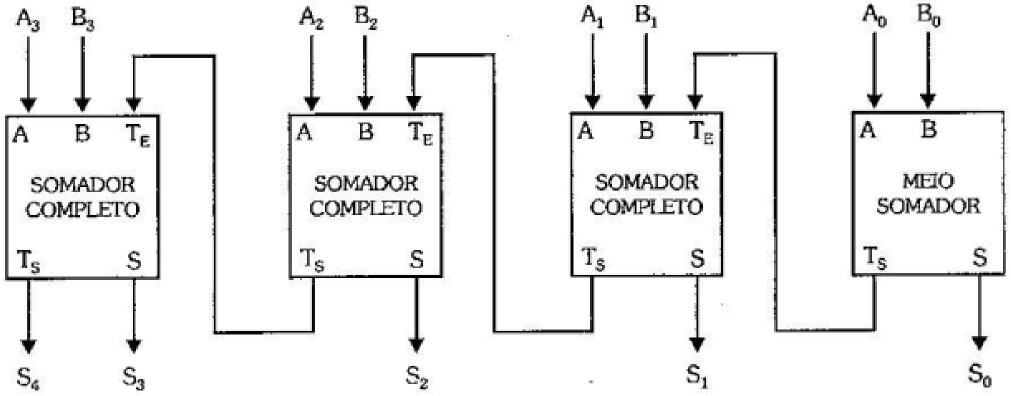
19. Somador Completo

- Para formar um circuito somador completo (também conhecido como Full Adder), podemos conectar meio somadores e somadores completo, de acordo com o número de bits necessários:

Vamos, para exemplo de aplicação, montar um sistema em blocos que efetua a soma de 2 números de 4 bits, conforme o esquema a seguir:

$$\begin{array}{cccccc} & A_3 & A_2 & A_1 & A_0 \\ + & B_3 & B_2 & B_1 & B_0 \\ \hline S_4 & S_3 & S_2 & S_1 & S_0 \end{array}$$

Para efetuar a soma dos bits A_0 e B_0 dos números (1ª coluna), vamos utilizar um Meio Somador, pois não existe transporte de entrada, mas para as outras colunas utilizaremos Somadores Completos, pois necessitaremos considerar os transportes provenientes das colunas anteriores. O sistema montado é visto na figura 5.37.



20. Meio Subtrator

- Antes de iniciarmos a análise de um circuito subtrator, vamos relembrar como funciona a subtração binária:

0 - 0 = 0

0 - 1 = 1 e transporta 1 ("empresta" 1)

1 - 0 = 1

1 - 1 = 0

Vamos montar a tabela da verdade de uma subtração de 2 números binários de 1 algarismo:

A	B	S	Ts	
0	0	0	0	(0 - 0 = 0 → Ts = 0)
0	1	1	1	(0 - 1 = 1 → Ts = 1)
1	0	1	0	(1 - 0 = 1 → Ts = 0)
1	1	0	0	(1 - 1 = 0 → Ts = 0)

20. Meio Subtrator

- O meio circuito subtrator é similar ao somador, apenas com um inversor a mais:

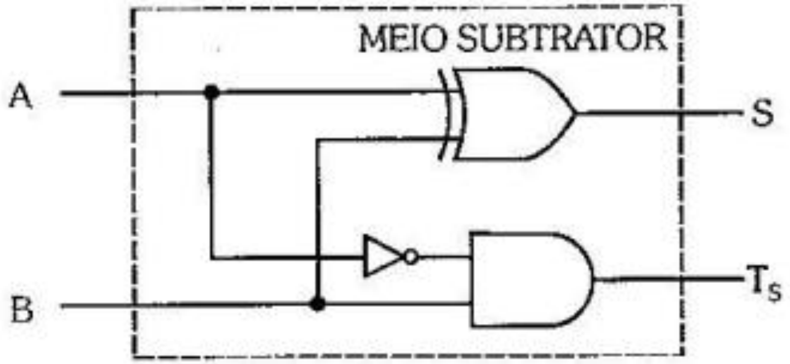
Representando cada número por 1 bit, podemos montar um circuito com as entradas A e B, e como saída, a subtração (S) e o transporte de saída (Ts).

As expressões características do circuito, extraídas da tabela, são:

$S = A \oplus B$

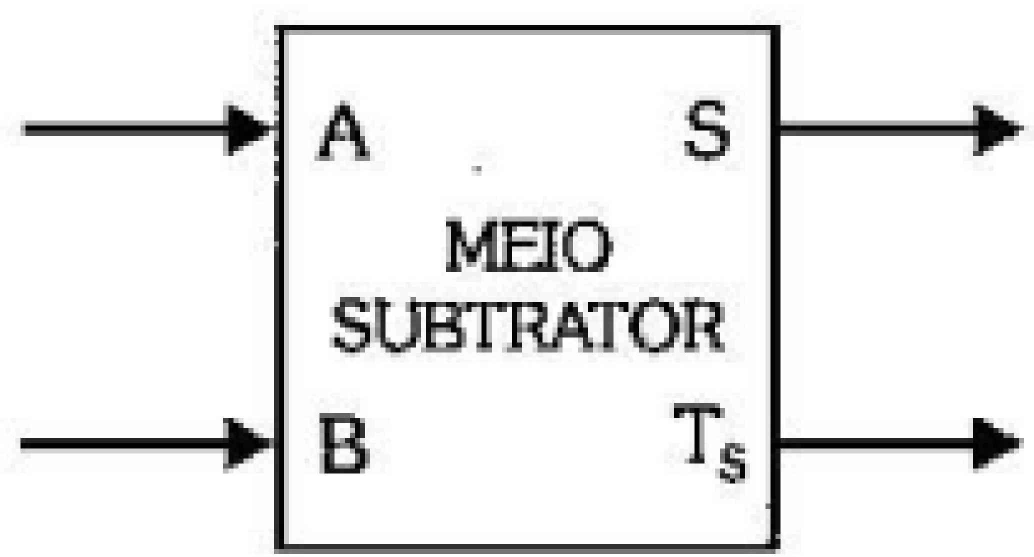
$Ts = \overline{A}B$

O circuito a partir destas, é visto na figura 5.44.



20. Meio Subtrator

- Podemos representar o meio circuito subtrator conforme a figura abaixo:



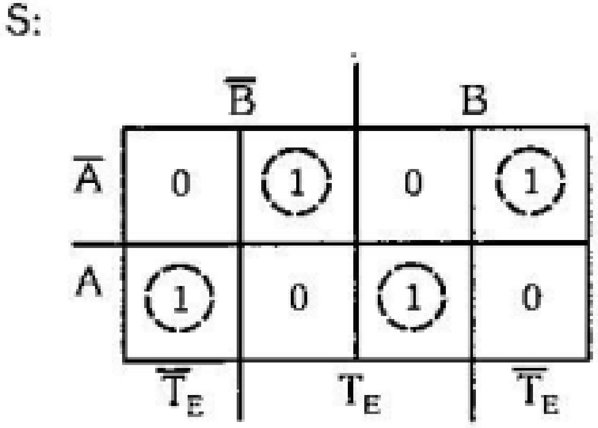
21. Subtrator Completo

- O meio subtrator possui limitações de operar com operandos maiores que 1 bit, pois não trata o transporte de entrada e de saída. Por isso é necessário um circuito completo.

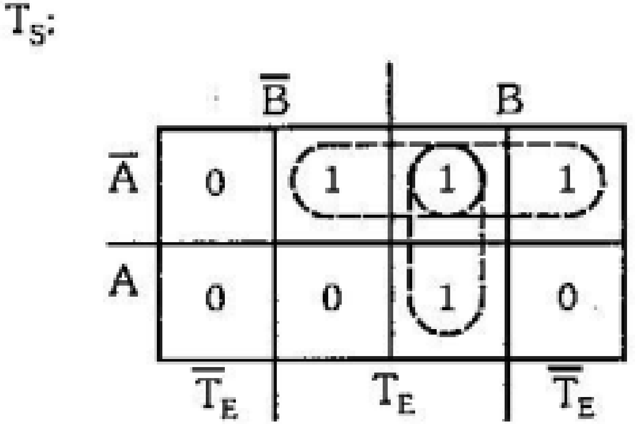
A	B	T_E	S	T_S
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

21. Subtrator Completo

- A partir da tabela, podemos montar o mapa:



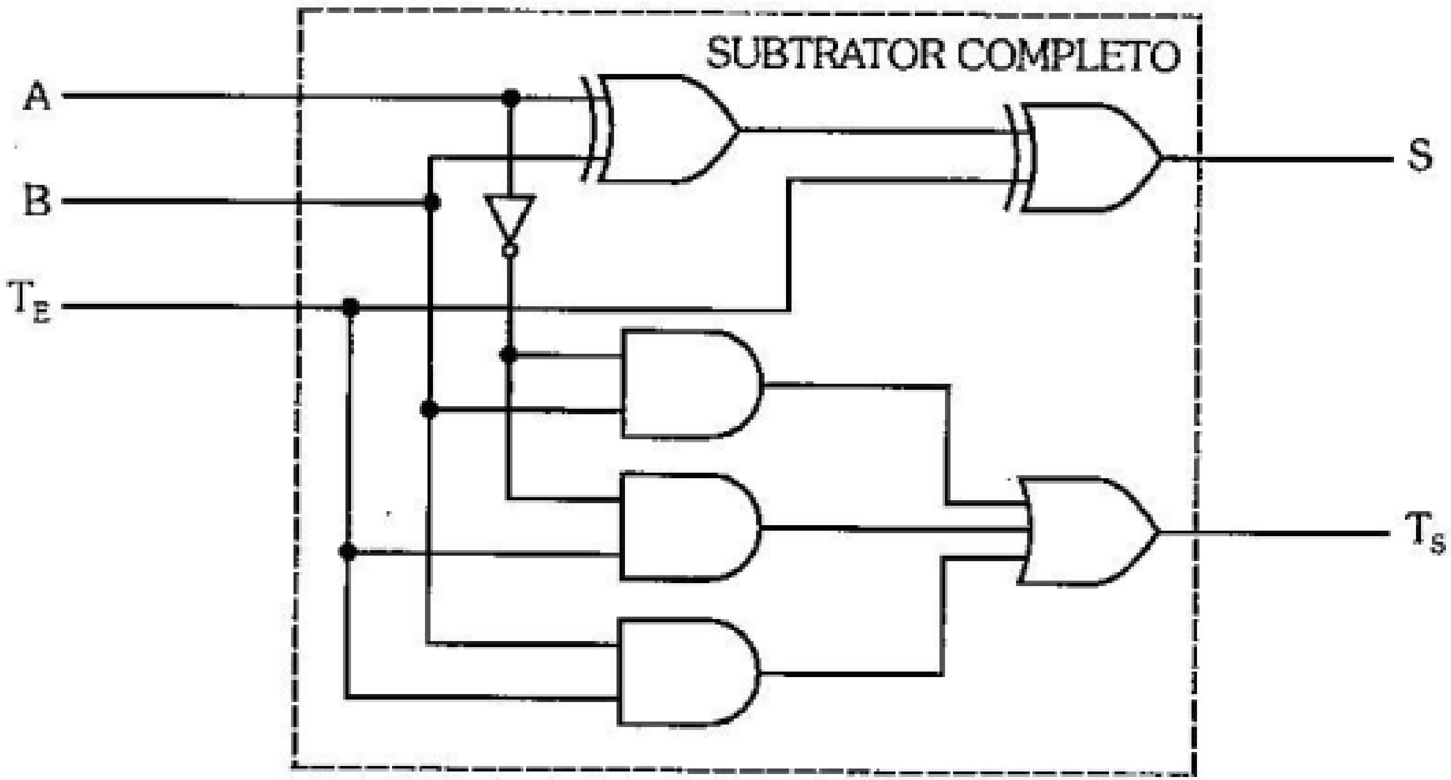
(a) $S = A \oplus B \oplus T_E$



(b) $Ts = \overline{A}B + \overline{A}T_E + BT_E$

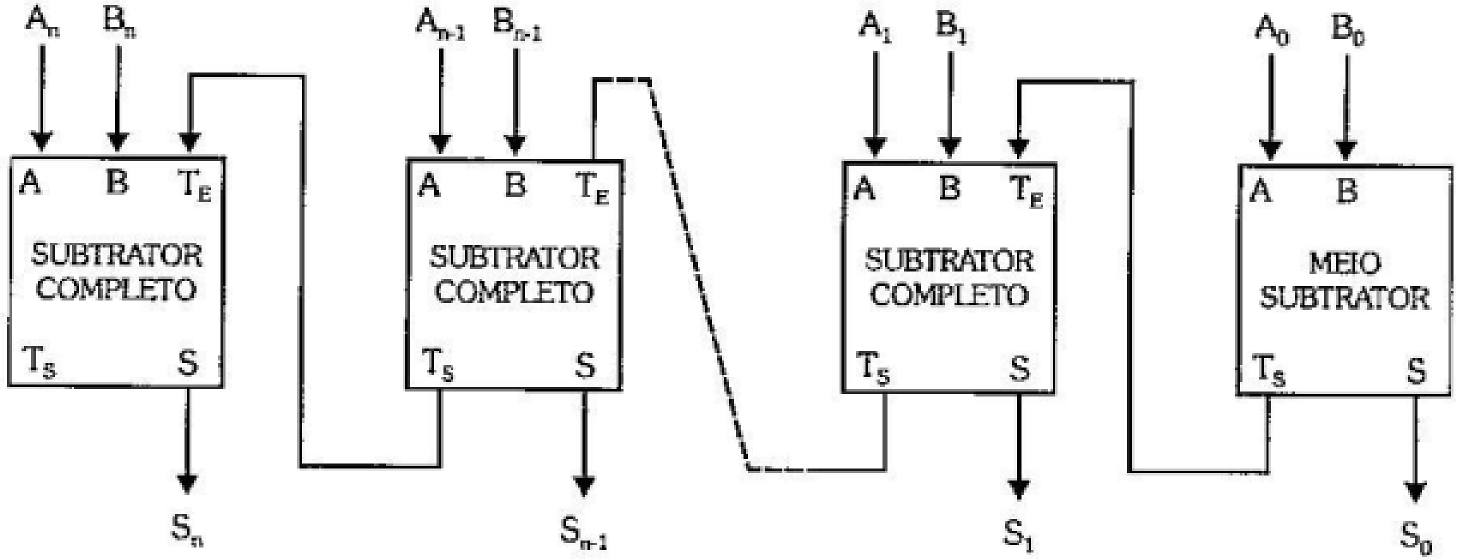
21. Subtrator Completo

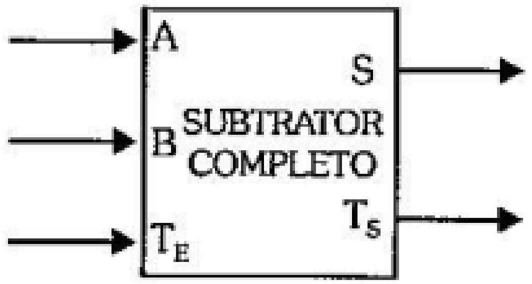
- Com as expressões simplificadas no mapa, podemos gerar o circuito:



21. Subtrator Completo

- Podemos representar o subtrator conforme a figura abaixo. Também é possível acoplar subtratores para operações com mais bits:





22. Somador/Subtrator Completo

- Podemos esquematizar um circuito que efetue as duas operações (soma e subtração). Para isso, vamos introduzir um flag de controle que chamaremos de M, como mostra a tabela:

M	A	B	T_E	S	T_S
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	1

Soma Completa (M = 0)

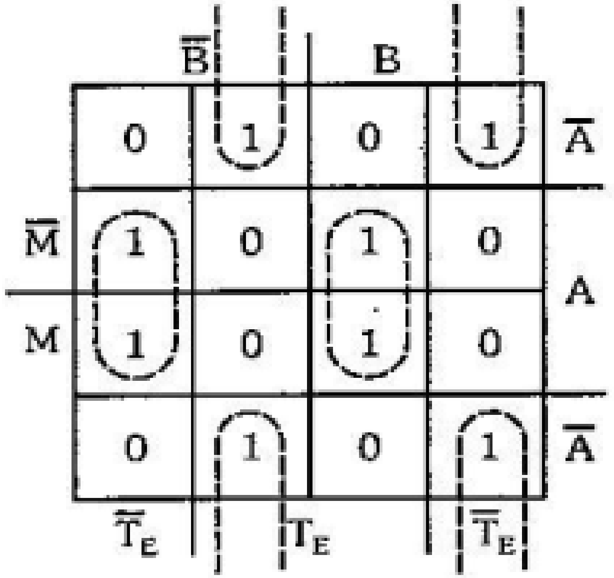
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	1	1

Subtração Completa (M = 1)

22. Somador/Subtrator Completo

- Simplificando as expressões da saída (S) e transporte de saída (Ts) no mapa:

S:



Do diagrama, obtemos:

$$S = A\overline{B}\,\overline{T}_E + \overline{A}\,\overline{B}T_E + ABT_E + \overline{A}B\overline{T}_E$$

Fatorando a expressão, temos:

$$S = \overline{A}(\overline{B}T_E + B\overline{T}_E) + A(\overline{B}\,\overline{T}_E + BT_E)$$

$$S = \overline{A}(B \oplus T_E) + A(B \odot T_E)$$

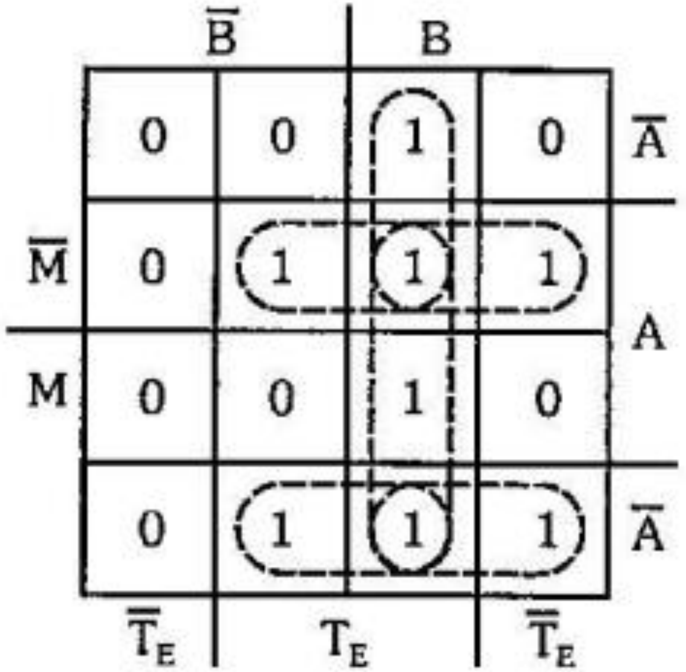
$$S = \overline{A}(B \oplus T_E) + A(\overline{B \oplus T_E})$$

$$\therefore S = A \oplus B \oplus T_E$$

22. Somador/Subtrator Completo

- Simplificando as expressões da saída (S) e transporte de saída (Ts) no mapa:

Ts:



Do diagrama, obtemos: $Ts = BT_E + \overline{M}AB + \overline{M}AT_E + M\overline{A}B + M\overline{A}T_E$

Fatorando a expressão, temos:

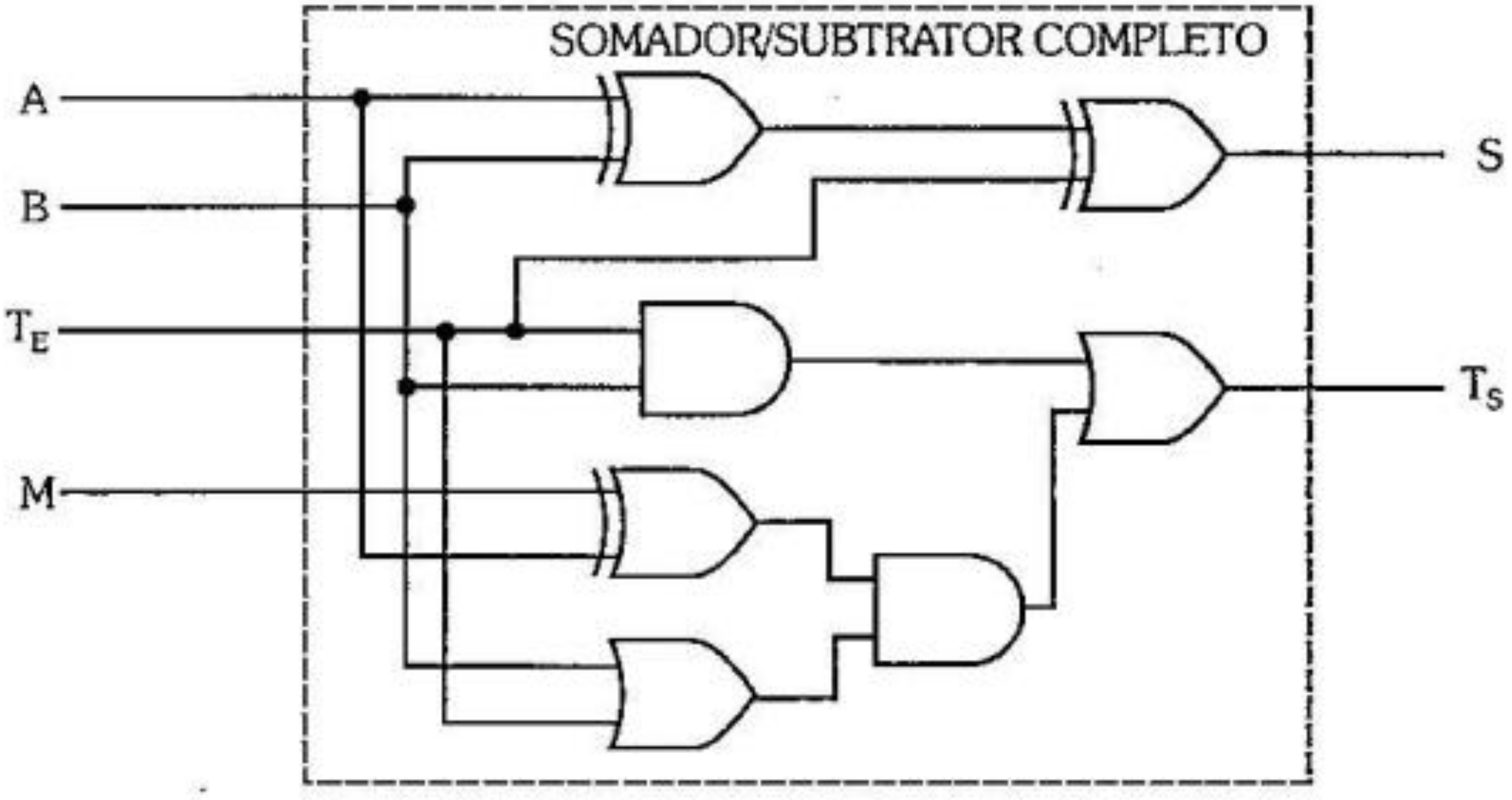
$$Ts = BT_E + B(\overline{M}A + M\overline{A}) + T_E(M\overline{A} + \overline{M}A)$$

$$Ts = BT_E + B(M \oplus A) + T_E(M \oplus A)$$

$$Ts = BT_E + (M \oplus A)(B + T_E)$$

22. Somador/Subtrator Completo

- Montando o circuito:



23. Exercícios

- Exercício 1: Se M=0, a saída (S) será igual a entrada (A), mas se M=1, a saída (S) será o complemento de 1.

- Lembrando: O cálculo do **complemento de 1** de um número binário é muito simples: basta **inverter todos os bits** do número original.

Desenvolva um circuito com uma entrada de controle M, para fornecer à saída o complemento de 1 de um número binário de 1 bit. (M = 0 = > Saída = número de entrada e M = 1 = > Saída = complemento de 1).

23. Exercícios

- Solução:

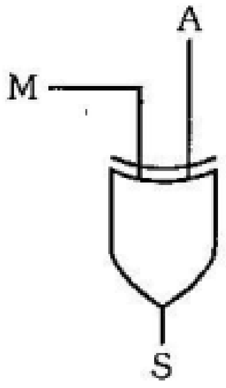
Para solucionar, vamos levantar a tabela da verdade, considerando a variável de controle M.

M	A	S	
0	0	0	Saída = número de entrada
0	1	1	
1	0	1	Saída = complemento de 1
1	1	0	

23. Exercícios

- Solução:

A partir da tabela, obtemos a expressão: $S = \overline{M}A + M\overline{A}$ ou $S = M \oplus A$, sendo o circuito derivado, visto na figura 5.60.



23. Exercícios

- Exercício 2: Esquematize, em blocos, um sistema subtrator para 2 números com 2 bits:

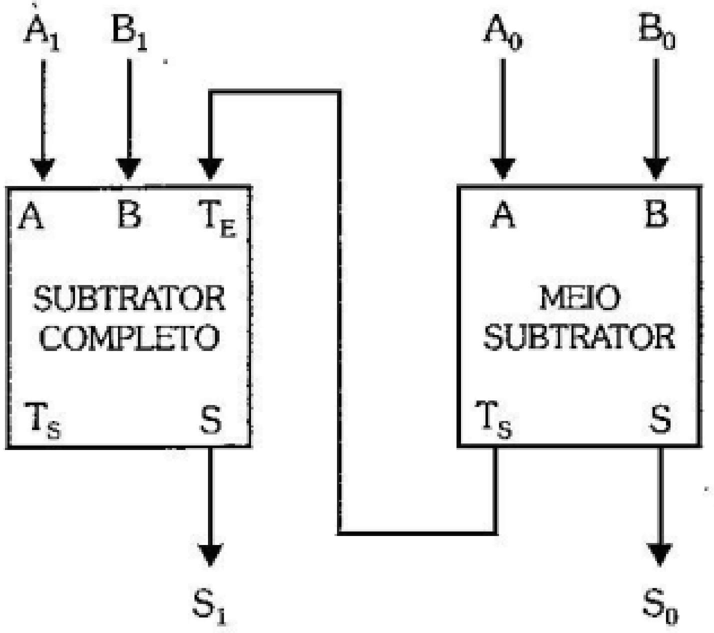
O sistema proposto irá realizar a subtração do número A_1A_0 com o número B_1B_0. Assim sendo, temos:

$$\begin{array}{rcc} & A_1 & A_0 \\ - & B_1 & B_0 \\ \hline & S_1 & S_0 \end{array}$$

23. Exercícios

- Resposta:

Para a 1ª coluna da operação, vamos utilizar um Meio Subtrator, pois não há transporte de entrada. Para a 2ª coluna, porém, utilizamos um Subtrator Completo, pois este possui entrada para o bit proveniente da coluna anterior. O circuito, assim esquematizado, é visto na figura 5.61.



24. Mercado Atual de Chips e Semicondutores

- A ASML, empresa sediada na Holanda, fabrica equipamentos de litografia (a etapa de “impressão” dos circuitos nos wafers de silício).

- Em particular, a ASML é a **única empresa no mundo** que fabrica as máquinas de litografia de **ultravioleta extremo (EUV)**, que são usadas para fabricar os chips mais avançados.

- Essas máquinas EUV são extremamente complexas, têm dezenas de milhares de partes e são essenciais para nós produzirmos processadores muito sofisticados.

24. Mercado Atual de Chips e Semicondutores

- **ASML** não fabrica os chips finais de processadores (como CPUs ou GPUs). O que ela produz são as **máquinas** usadas por fabricantes de chips (como TSMC, Intel, Samsung) para imprimir os circuitos.

- **Nem todas as máquinas de litografia são só da ASML**: para tecnologias menos avançadas (por exemplo, litografia DUV — deep ultraviolet), existem outras empresas que fazem equipamentos.

- **Custo e complexidade enormes**: desenvolver uma máquina EUV exige décadas de P&D, parcerias e investimentos bilionários.

- **Exportação controlada**: por serem tão estratégicas, há controvérsias políticas e restrições para a venda dessas máquinas para certos países (por exemplo, controles de exportação).

25. Soberania na Produção de Processadores

- Nenhum país é completamente soberano na produção de processadores. Mas alguns países dominam boa parte do processo.

1. Taiwan

- É um dos líderes mais importantes na fabricação de semicondutores avançados. A TSMC (Taiwan Semiconductor Manufacturing Company) está sediada lá e é a maior fundição (foundry) independente do mundo.
- A TSMC fabrica chips com nós de tecnologia muito avançados, usados por empresas como Apple, NVIDIA, etc.

25. Soberania na Produção de Processadores

2. **Coreia do Sul**

- Também muito forte na fabricação: a **Samsung** é um dos gigantes de semicondutores (memória, lógicos, etc.).
- Tem capacidade de fundição e outros tipos de produção de semicondutores.

3. Estados Unidos

- Os EUA dominam bastante no **design de chips**: empresas como Intel, AMD, NVIDIA, Qualcomm, etc. são americanas.
- Também há fábricas: embora a participação de fabricação nos EUA tenha diminuído historicamente, ainda há plantas importantes.
- Além disso, empresas americanas dominam parte das ferramentas (ferramentas de lithografia, equipamento) necessárias para produzir chips.

25. Soberania na Produção de Processadores

4. **China**

- A China tem investido muito para aumentar sua capacidade de produção de semicondutores.
- Existe a **SMIC**, por exemplo, que é uma foundry chinesa.
- Mas é importante notar que, em tecnologias ultra-avançadas (como nós de processo muito pequenos), a China ainda tem limitações e depende de algumas importações.

5. União Europeia / Europa

- A participação da Europa na produção de semicondutores não é tão dominante quanto Ásia, mas há capacidades estratégicas.
- Países como Alemanha, França, Holanda têm empresas importantes no ecossistema de semicondutores (design, equipamentos, alguns processos de fabricação)

25. Soberania na Produção de Processadores

- Engenheiros chineses tentaram desmontar e copiar uma das máquinas de litografia da ASML usadas na fabricação de chips, mas acabaram quebrando o equipamento e chamando a própria empresa holandesa para consertar.

26. E o Brasil?

- Para fabricar um processador moderno, você precisa de **toda uma cadeia industrial**:

1. fábricas de wafers
2. fornecedores de gases e químicos de alta pureza
3. máquinas de litografia (extremamente complexas)
4. empresas de design de chips
5. empresas de embalagem e teste
6. mão de obra altamente especializada
7. Nenhum destes setores existe no Brasil em escala.

E montar essa cadeia custa **dezenas de bilhões de dólares**.

26. E o Brasil?

- Uma fábrica moderna (para processadores de ponta, como 5 nm ou 3 nm) custa: **20 a 30 bilhões de dólares**

- Precisa de mais de 10 mil engenheiros especializados

- Cadeias logísticas globais extremamente rigorosas

- Para comparar: isso é **quase todo o orçamento anual do Ministério da Ciência e Tecnologia do Brasil**. Mesmo países ricos têm dificuldade de bancar isso sozinhos.

26. E o Brasil?

- O Brasil teria que comprar máquinas como:
- **EUV** da ASML (Holanda) — única empresa do mundo que faz as máquinas mais avançadas
- Equipamentos de deposição, corrosão (etch), ion implantation etc. (EUA, Japão)
- Há ainda **restrições políticas e comerciais** severas: EUA e Europa controlam rigorosamente para quem vendem certos equipamentos.

26. E o Brasil?

A política industrial brasileira para semicondutores:

- começou forte nos anos 2000
- teve incentivos fiscais (Lei de Informática)
- tentou montar fábricas (como a CEITEC, em Porto Alegre)
- mas faltou continuidade, investimento, escala e visão de longo prazo

A CEITEC, por exemplo:

- produziu chips **simples**, como RFID;
- mas não chips de computação;
- e acabou sendo encerrada/desmobilizada em 2020–2022 por falta de viabilidade econômica.

26. E o Brasil?

O mundo inteiro se especializou — e o Brasil ficou na ponta errada

A indústria global se tornou **altamente especializada**:

- Taiwan lidera manufatura avançada
- EUA lideram design
- Japão domina materiais e máquinas
- Coreia do Sul domina memória
- Holanda domina as máquinas EUV
- China domina encapsulamento e volume

O Brasil ficou voltado para:

- agronegócio
- mineração
- indústria básica
- poucos setores de alta tecnologia

26. E o Brasil?

Mesmo países grandes e ricos têm tido dificuldade:

- **Alemanha** e **França** estão tentando atrair fábricas, mas dependem da TSMC
- **China** gasta centenas de bilhões e ainda não domina nós de 3 nm
- **Índia** tenta e falha desde os anos 80

Se esses países têm dificuldades, imagine o Brasil com menos recursos e infraestrutura.

27. Crise dos Semicondutores

O que foi a crise dos semicondutores?

Entre 2020 e 2023, o mundo **não tinha chips suficientes** para atender à demanda da indústria. Isso causou:

- fábricas de carros paradas,
- atrasos em eletrônicos,
- produtos mais caros,
- estoques zerados.

27. Crise dos Semicondutores

1) A pandemia desorganizou toda a cadeia de suprimentos: Com lockdowns, fábricas na Ásia pararam, navios acumulados, portos travados.

2) A demanda por eletrônicos explodiu: Com home office e aulas online, aumentou a compra equipamentos eletrônicos.

3) Indústria automotiva errou previsões: No início da pandemia, montadoras acharam que as vendas iam cair e **cancelaram pedidos de chips**. Meses depois, as vendas dispararam mas os fabricantes de chips já tinham direcionado a capacidade para eletrônicos.

4)Pouquíssimas empresas conseguem fabricar chips avançados:

- TSMC (Taiwan)
- Samsung (Coreia)
- Intel (EUA)

27. Crise dos Semicondutores

5) Tensões geopolíticas

- Disputa EUA × China
- Restrições de exportação
- Taiwan (TSMC) no centro global da fabricação

6) Falta de componentes menores (não só processadores)

A crise não afetou apenas chips avançados: faltaram **microcontroladores simples**, memórias, controladores de energia — usados em eletrodomésticos e carros.

27. Crise dos Semicondutores

Problemas gerados pela crise:

- Carros ficaram sem **chips para central multimídia, ABS, airbag**.
- PlayStation 5 e placas de vídeo sumiram.
- Notebooks ficaram mais caros e demorados para chegar às lojas.
- Indústrias inteiras pararam por **falta de um único componente** de centavos.

A partir de **2023**, a situação foi se normalizando:

- indústrias ajustaram pedidos,
- novas fábricas iniciaram operação,
- a demanda de eletrônicos caiu após o pico da pandemia.

  Mas alguns setores **ainda sofrem**, especialmente automotivo e industrial.