

2 - Álgebra de Boole

2.1 - Função lógica

George Boole desenvolveu no século XIX a álgebra necessária à investigação das leis fundamentais das operações da mente humana ligadas ao raciocínio.

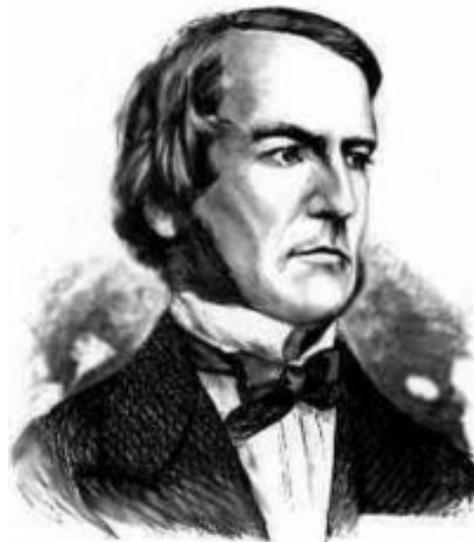


Fig. 2.1 - George Boole (1814-1864).

George Boole (1814-1864) foi um matemático e filósofo britânico, criador da Álgebra Booleana, base da actual aritmética computacional.

O objectivo da Álgebra de Boole passa pela definição de uma série de símbolos, com a finalidade de representar objectivos ou fenómenos que, encadeados convenientemente, dão lugar a expressões matemáticas mais complexas denominadas funções.

Enquanto que a álgebra tradicional opera com relações quantitativas, a álgebra de Boole opera com relações lógicas. Como exemplo podemos considerar os operadores “ + ” e “ x ”, e verificar que possuem significados diferentes conforme sejam utilizados na álgebra tradicional ou na Álgebra Booleana.

Operador	Álg. Tradicional	Álg. Booleana
“ + ”	Soma	“OR”
“ x ”	Multiplicação	“AND”

Tab. 2.1 – Operadores lógicos.

Enquanto que na álgebra tradicional as variáveis podem assumir qualquer valor, na álgebra booleana, as variáveis, aqui denominadas por variáveis binárias, apenas podem assumir um de dois valores binários. Estes valores binários não exprimem quantidades mas apenas, e só, estados do sistema.

A importância da lógica booleana no mundo digital é hoje indiscutível, sendo utilizada em áreas tão diversas como as memórias digitais; os circuitos discretos; ou os microprocessadores.

Nota: A multiplicação entre 2 variáveis a e b é frequentemente representada por um asterisco (“ * ”), ou por um ponto (“ . ”), ou simplesmente nada ente os dois, em vez de “ x ”. Ou seja:

$$a \times b, \quad a * b, \quad a \cdot b \quad \text{ou} \quad ab$$

Função lógica binária

A toda a variável binária, cujo valor depende de uma expressão algébrica, formada por outras variáveis binárias relacionadas por operadores lógicos, dá-se o nome de função lógica. Temos como exemplo a Eq. 2.1, onde S representa a variável binária dependente e a, b e c as variáveis binárias independentes.

$$S = f(a,b,c) = a \cdot b + b \cdot c, \tag{Eq. 2.1}$$

Outra forma de representar uma função lógica é através da utilização de um circuito eléctrico (Fig. 2.2). Nesta representação, as variáveis binárias são introduzidas como tratando-se de interruptores com duas posições (ON/OFF).

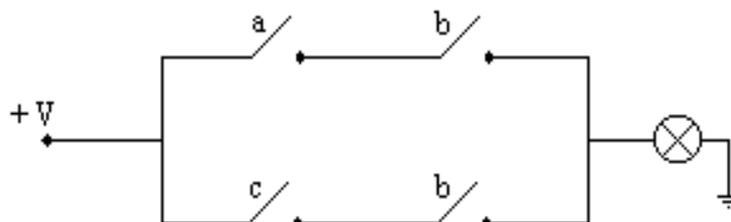


Fig. 2.2 - Circuito lógico.

Nestas topologias é normal fazer-se uso de dois tipos de interruptores (Fig. 2.3 e Fig. 2.4).

<i>Interruptor normalmente aberto</i>	no estado normal, ou seja, não actuado, o interruptor encontra-se aberto, não deixando por isso passar corrente.
<i>Interruptor normalmente fechado</i>	no estado normal, ou seja, não actuado, o interruptor encontra-se fechado, deixando por isso passar corrente.



Fig. 2.3 - Interruptor *a* do tipo normalmente aberto.

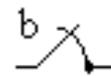


Fig. 2.4 - Interruptor *b* do tipo normalmente fechado.

Tabela de verdade

Para uma função $S = f(a, b, \dots)$ a tabela de verdade é um quadro formado por tantas colunas quantas as variáveis binárias independentes (a, b, \dots), tendo ainda uma última coluna correspondente à variável binária dependente S .

O número de linhas de uma tabela de verdade é dado por 2^N , onde N é o número de variáveis binárias independentes, garantimos assim todas as combinações possíveis das variáveis independentes.

A tabela de verdade representa o estado da variável binária dependente em função das variáveis binárias independentes. Para todas as combinações possíveis de estados das variáveis binárias independentes existe um estado da variável binária dependente.

Como exemplo podemos ver a tabela de verdade da Tab. 2.2 que representa a função lógica binária $S = f(a, b, c)$ dada pela Eq. 2.1. O número de variáveis binárias independentes é 3 (a, b e c), logo a tabela tem $2^3 = 8$ linhas.

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Tab. 2.2 – Tabela de verdade.

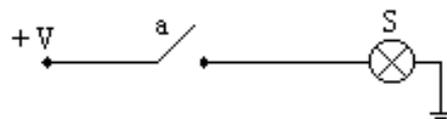
2.2 - Funções Booleanas básicas e suas tabelas de verdade

A seguir apresentamos as funções booleanas básicas e as respectivas tabelas de verdade.

Função igualdade

$$S(a) = a$$

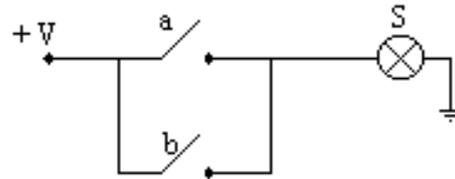
a	S
0	0
1	1



Função união [OR] (soma lógica)

$$S(a,b) = a + b$$

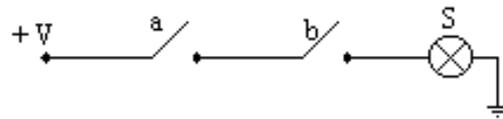
a	b	S
0	0	0
0	1	1
1	0	1
1	1	1



Função intersecção [AND] (multiplicação lógica)

$$S(a,b) = a \cdot b$$

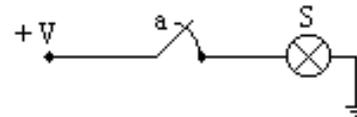
a	b	S
0	0	0
0	1	0
1	0	0
1	1	1



Função negação [NOT] (complementar lógico)

$$S(a) = \bar{a}$$

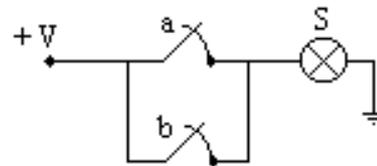
a	S
0	1
1	0



Função negação da intersecção [NAND]

$$S(a,b) = \overline{a \cdot b}$$

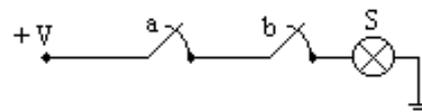
a	b	S
0	0	1
0	1	1
1	0	1
1	1	0



Função negação da reunião [NOR]

$$S = \overline{a + b}$$

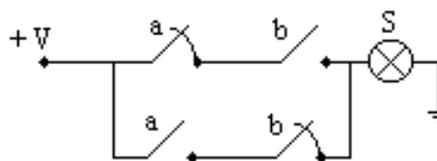
a	b	S
0	0	1
0	1	0
1	0	0
1	1	0



Função OR exclusivo [XOR]

$$S = a \cdot \bar{b} + \bar{a} \cdot b$$

a	b	S
0	0	0
0	1	1
1	0	1
1	1	0



2.3 - Postulados, propriedades e teoremas

A álgebra booleana, tal como a álgebra tradicional, segue um conjunto de postulados, propriedades, e teoremas que interessa conhecer e analisar.

Postulado 1 (Lei do elemento absorvente para a soma lógica)

A soma lógica de uma variável binária mais um 1 lógico equivale a 1 lógico, ou,

$$a + 1 = 1. \tag{Eq. 2.2}$$

Postulado 2 (Lei do elemento neutro para a soma lógica)

A soma lógica de uma variável binária mais um 0 lógico equivale ao valor da variável binária, ou,

$$a + 0 = a. \tag{Eq. 2.3}$$

Postulado 3 (Lei do elemento neutro para a multiplicação lógica)

O produto lógico de uma variável binária pôr um 1 lógico é igual ao valor da variável binária, ou,

$$a \cdot 1 = a. \tag{Eq. 2.4}$$

Postulado 4 (Lei do elemento absorvente para a multiplicação lógica)

O produto lógico de uma variável binária pôr um 0 lógico é igual a um 0 lógico, ou,

$$a \cdot 0 = 0. \tag{Eq. 2.5}$$

Postulado 5 (Lei da idempotência)

A soma lógica de duas variáveis binárias iguais equivale ao valor lógico dessa variável binária, ou,

$$a + a = a \quad \text{Eq. 2.6}$$

$$a \cdot a = a$$

Postulado 6 (Lei do elemento complementaridade para a soma lógica)

A soma lógica de uma variável binária mais a negação da mesma variável binária equivale a 1 lógico, ou,

$$a + \bar{a} = 1. \quad \text{Eq. 2.7}$$

Postulado 7 (Lei do elemento complementaridade para a multiplicação lógica)

O produto lógico de uma variável binária mais a negação da mesma variável binária equivale a 0 lógico, ou,

$$a \cdot \bar{a} = 0. \quad \text{Eq. 2.8}$$

Postulado 8 (Lei da involução)

Se uma variável binária é negada duas vezes esta não varia, ou,

$$\bar{\bar{a}} = a. \quad \text{Eq. 2.9}$$

Este postulado é válido para qualquer número par de negações.

Postulado 9

Se os dois membros de uma igualdade forem negados, esta não sofre qualquer alteração, ou seja,

$$S = a + b \quad \Rightarrow \quad \bar{S} = \overline{(a + b)} \quad \text{Eq. 2.10}$$

$$S = a \cdot b \quad \Rightarrow \quad \bar{S} = \overline{(a \cdot b)}$$

Propriedade comutativa

$$a + b = b + a \quad \text{Eq. 2.11}$$

$$a \cdot b = b \cdot a$$

Propriedade associativa

$$(a + b) + c = a + (b + c) \quad \text{Eq. 2.12}$$

$$(a \cdot b) \cdot c = a \cdot (b \cdot c)$$

Propriedade distributiva

$$a \cdot (b + c) = a \cdot b + a \cdot c \quad \text{Eq. 2.13}$$

$$(a \cdot b) + c = (a + c) \cdot (b + c)$$

Teorema 1 (Lei da absorção)

$$a + a \cdot b = a \quad \text{Eq. 2.14}$$

$$a \cdot (a + b) = a$$

Teorema 2

$$a + \bar{a} \cdot b = a + b \quad \text{Eq. 2.75}$$

$$b \cdot (a + \bar{b}) = a \cdot b$$

Teorema 3 (Leis de De Morgan ou Princípio da dualidade)

$$\overline{a + b} = \bar{a} \cdot \bar{b} \quad \text{Eq. 2.16}$$

$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

2.4 - Formas canónicas de uma função lógica

Uma função lógica binária $S = f(a, b, \dots)$ é dita estar representada na 1ª forma canónica se S estiver escrita como a soma de produtos, nos quais aparecem todas as variáveis binárias em cada um dos termos (parcelas ou MINTERMS) que constituem a expressão, em forma directa ou complementada (i.e., negação).

Por exemplo, a função lógica binária $S_1 = f(a, b, c)$ da Eq. 2.7 está expressa na 1ª forma canónica.

$$S_1 = a \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot \bar{b} \cdot \bar{c} \quad \text{Eq. 2.17}$$

Uma função lógica binária $S = f(a, b, \dots)$ é dita estar representada na 2ª forma canónica se S estiver escrita como produto de somas, nos quais aparecem todas as variáveis binárias em cada um dos termos (factores ou MAXTERMS) que constituem a expressão, em forma directa ou complementada (i.e., negação).

Por exemplo, a função lógica binária $S_2 = f(a, b, c)$ da Eq. 2.18 está expressa na 2ª forma canónica.

$$S_2 = (a + b + \bar{c}) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + b + \bar{c}) \quad \text{Eq. 2.18}$$

Para podermos obter a função lógica nas suas formas canónicas directamente da tabela de verdade temos que seguir as regras descritas abaixo:

- A 1ª forma canónica obtém-se somando todos os produtos lógicos (ou MINTERMS) formados das linhas da tabela de verdade que dão à função S o valor lógico de 1.
- A 2ª forma canónica obtém-se multiplicando todas as somas lógicas (ou MAXTERMS) formadas das linhas da tabela de verdade que dão à função S o valor lógico de 0.

Exemplo A: Considere a seguinte tabela de verdade:

a	b	c	S
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Como o valor lógico de S é 1 para as linhas 1, 6 e 7, a expressão de S na 1ª Forma Canónica fica sendo a soma de MINTERMS formados por estas linhas:

$$S_1 = \bar{a} \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}.$$

Por outro lado, como o valor lógico de S é 0 para as linhas 2, 3, 4, 5 e 8, a expressão de S na 2ª Forma Canónica fica sendo a multiplicação de MAXTERMS formados por estas linhas:

$$S_2 = (a + b + \bar{c}) \cdot (a + \bar{b} + c) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + b + c) \cdot (\bar{a} + \bar{b} + \bar{c}).$$

2.5 - Simplificação de funções lógicas

É muito importante simplificar as funções lógicas obtidas da tabela de verdade. Quanto mais simples for a expressão obtida, menor será o número de componentes a utilizar na sua implementação.

Uma forma de simplificar o número de termos de um função é através da álgebra de Boole, o que requer muita experiência. Existem, no entanto, outros métodos igualmente eficazes, são eles: o método gráfico ou de Karnaugh e o método numérico de Quine-McCluskey.

Simplificação pelo método algébrico

Não existem regras rígidas existindo por isso a necessidade de recorrer às regras já apresentadas: postulados; propriedades e teoremas da álgebra de Boole.

Exemplo B:

$$F = a + b \cdot \overline{(\bar{a} \cdot \bar{c})} = a + b \cdot (a + c) = a + b \cdot a + bc = a \cdot (b + 1) + bc = a + bc$$

Podemos transformar qualquer função lógica, seja qual for a forma em que está escrita, numa representação na forma canónica, tanto soma de produtos como produto de somas.

Na ausência de metodologias, bem definidas, temos que utilizar a álgebra de Boole de uma forma racional. Como exemplo de aplicação do método temos.

Exemplo C: Simplificar a seguinte expressão

$$F = a \cdot (b + c) = a \cdot b + a \cdot c$$

Para se passar para a **1ª forma canónica** observamos que no primeiro termo (MINITERM) falta a variável “ c ” e no segundo termo (MINITERM) falta a variável “ b ”.

Então completamos com $(c + \bar{c})$ no primeiro termo e com $(b + \bar{b})$ no segundo termo. Logo,

$$F = a \cdot b \cdot (c + \bar{c}) + a \cdot (b + \bar{b}) \cdot c$$

e, usando a propriedade distributiva (Eq. (2.13)) e desenvolvendo temos F na **1ª forma canónica**:

$$F = a \cdot b \cdot c + a \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot c$$

Para se passar para a **2ª forma canónica** observamos que no primeiro termo (MAXTERM) faltam as variáveis “ b ” e “ c ” e no segundo termo (MAXTERM) falta a variável “ a ”.

Então completamos com $(b \cdot \bar{b})$ e $(c \cdot \bar{c})$ no primeiro termo e com $(a \cdot \bar{a})$ no segundo termo. Logo,

$$F = (a + b \cdot \bar{b} + c \cdot \bar{c}) \cdot (a \cdot \bar{a} + b + c)$$

e, usando a propriedade distributiva (Eq. (2.13)), obtém-se:

$$F = [(a + b) \cdot (a + \bar{b}) + c \cdot \bar{c}] \cdot [(a + b + c) \cdot (\bar{a} + b + c)]$$

que desenvolvendo nos dá:

$$\begin{aligned} F &= [(a + b + c \cdot \bar{c}) \cdot (a + \bar{b} + c \cdot \bar{c})] \cdot [(a + b + c) \cdot (\bar{a} + b + c)] \\ &= (a + b + c) \cdot (a + b + \bar{c}) \cdot (a + \bar{b} + c) \cdot (a + \bar{b} + \bar{c}) \cdot (a + b + c) \cdot (\bar{a} + b + c) \end{aligned}$$

que ainda pode ser reduzida visto que o MAXTERM $(a + b + c)$ aparece 2 vezes e portanto pode ser eliminado. Logo, F na **2ª forma canónica**:

$$F = (a + b + c) \cdot (a + b + \bar{c}) \cdot (a + \bar{b} + c) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + b + c)$$

Entretanto, as formas canónicas não são, em geral, as mais simples expressões para uma função lógica.

Há outros métodos para simplificar funções lógicas que estejam escritas em uma das duas **formas canónicas** e um destes métodos, o método gráfico de Karnaugh, é o que veremos a seguir.

Método gráfico de Karnaugh

Este método é relativamente simples e eficaz até 4 variáveis. Pode ser usado com mais variáveis (5 ou 6) mas não é aconselhável.

Para utilizar o método é necessário utilizar a expressão numa das formas canónicas. Devemos, inicialmente, construir mapas com 2^N células, cada uma correspondente a um MINTERM.

F	0	1	b
0			
1			
a			

2 variáveis

F	0	0	1	1	b
	0	1	1	0	c
0					
1					
a					

3 variáveis

F	0	0	1	1	c
	0	1	1	0	d
0	0				
0	1				
1	1				
1	0				
a	b				

4 variáveis

O passo seguinte consiste no preenchimento destes mapas, com base na tabela de verdade ou da função lógica na 1ª ou 2ª forma canónica. Se de uma célula para a outra apenas variar uma das variáveis, então, podemos dizer que temos **adjacência lógica**.

Regras a respeitar para o preenchimento dos mapas:

- ❑ Coloca-se “ 1 ” em cada célula (quer seja a 1ª ou 2ª forma canónica) onde a função exista.
- ❑ Agrupam-se os 1's em blocos de 2, 4, 8 ou 16 células. Só se podem agrupar células adjacentes.
- ❑ Construir o menor número de blocos com o maior número de 1's possível. Uma mesma casa pode pertencer a mais do que um bloco.
- ❑ A cada grupo de 1's corresponde um termo (parcela) da expressão final.
- ❑ A variável é representada na forma negada quando o valor lógico dela for “ 0 ”, ou “ 1 ” quando o valor for directo (não negado).

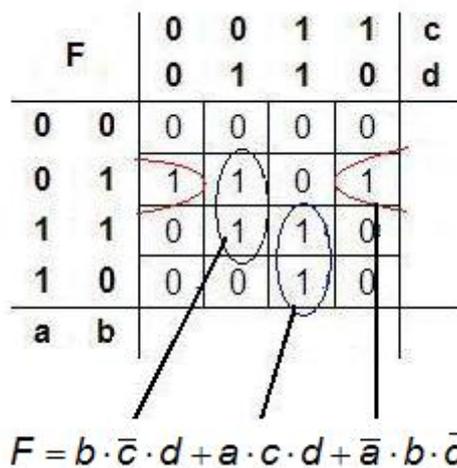
Exemplo D: Considere a função lógica $F = f(a, b, c, d)$ abaixo na 1ª forma canónica:

$$F = \bar{a} \cdot b \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot b \cdot \bar{c} \cdot d + \bar{a} \cdot b \cdot c \cdot \bar{d} + a \cdot \bar{b} \cdot c \cdot d + a \cdot b \cdot \bar{c} \cdot d + a \cdot b \cdot c \cdot d$$

cuja tabela da verdade é dada abaixo:

a	b	c	d	F
0	0	0	0	0
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	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Construindo-se o mapa abaixo (com $2^4 = 16$ células), temos:



E portanto, após agruparmos os 1's em blocos de 2, temos a expressão de F simplificada:

$$F = b \cdot \bar{c} \cdot d + a \cdot c \cdot d + \bar{a} \cdot b \cdot \bar{d}$$

que pode ser representada pelo seguinte circuito da Fig. 2.5.

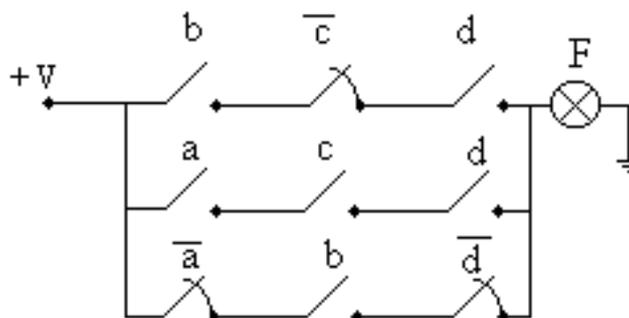


Fig. 2.5 – Sistema de controlo industrial típico.

Exemplo F: Considere a função lógica $F = f(a, b, c, d)$ abaixo na 1ª forma canónica:

$$F = a \cdot b \cdot c \cdot d + a \cdot b \cdot \bar{c} \cdot d + \bar{a} \cdot b \cdot \bar{c} \cdot d + a \cdot \bar{b} \cdot c \cdot \bar{d} + a \cdot b \cdot c \cdot \bar{d}$$

cuja tabela da verdade é dada abaixo:

a	b	c	d	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Construindo-se o mapa abaixo (com $2^4 = 16$ células), temos:

F		0	0	1	1	c
		0	1	1	0	d
0	0	0	0	0	0	
0	1	0	1	0	0	
1	1	0	1	1	1	
1	0	0	0	0	1	
a	b					

E portanto, após agruparmos os 1's em blocos de 2, temos a expressão de F simplificada:

$$F = b \cdot \bar{c} \cdot d + a \cdot b \cdot d + a \cdot c \cdot \bar{d}.$$

que pode ser realizada na forma de circuito como:

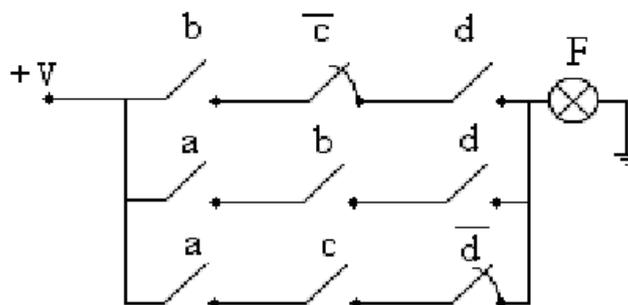


Fig. 2.6 – Sistema de controlo industrial típico.

Entretanto, se agruparmos os 1's em blocos diferentes:

F		0	0	1	1	c
		0	1	1	0	d
0	0	0	0	0	0	
0	1	0	1	0	0	
1	1	0	1	1	1	
1	0	0	0	0	1	
a	b					

teremos então outra expressão de F simplificada:

$$F = b \cdot \bar{c} \cdot d + a \cdot b \cdot c + a \cdot c \cdot \bar{d}$$

que pode ser realizada na forma de circuito como:

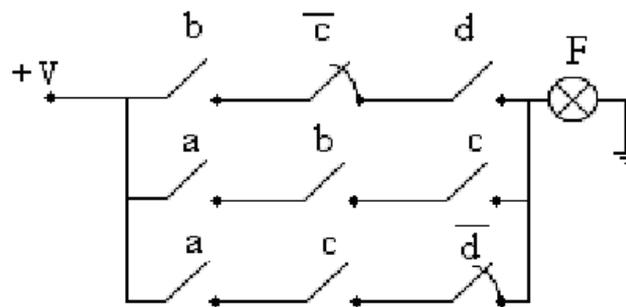


Fig. 2.7 – Sistema de controlo industrial típico.

- xxx -