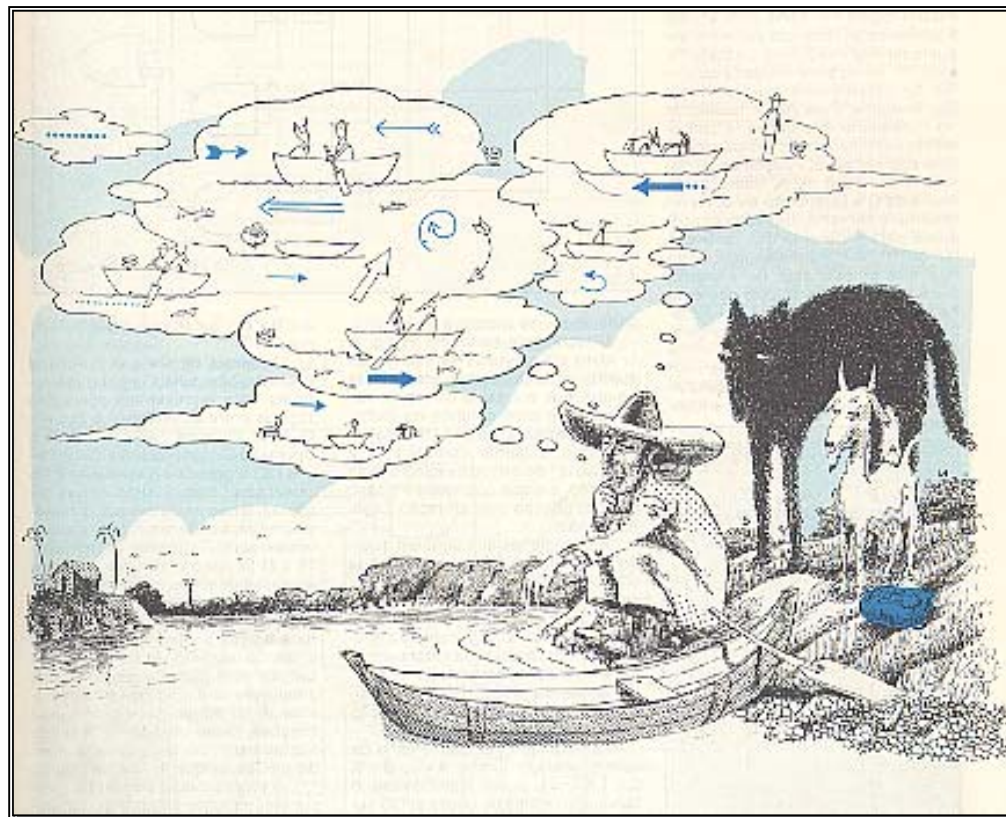


## PROJETO “O PROBLEMA DO FAZENDEIRO”

Aplicação de minitermos na execução de projetos  
Simplificação gráfica de expressões Booleanas por Mapa de Karnaugh (M.K.)



Um fazendeiro encontra-se na margem de um rio com seu barco e pretende transportar para outra margem um lobo, uma cabra e um pé de couve.

Sabe-se que ele pode transportar apenas um dos três por vez, um dos animais ou o pé de couve.

É claro que ele não pode levar a couve e deixar a cabra, indefesa, a mercê do lobo. Ou levar o lobo e deixar a cabra deliciar-se com o pé de couve.

*Como o fazendeiro resolverá o problema?*

Para padronizar o projeto utilize as variáveis:

A = lobo  
B = cabra  
C = couve  
D = fazendeiro

**DICA: definir as margens do rio com os níveis lógicos 0 e 1, por exemplo, margem esquerda do rio NL = 0 e a margem direita do rio, NL = 1**

Por exemplo: Se  $A = 1$ , significa que o lobo está na margem; se  $A = 0$ , significa que o lobo não está na margem, ou seja, está em transporte ou está na outra margem.

Assim, se  $A = 1$ ;  $B = 1$ ;  $C = 0$  e  $D = 0$ , significa que o lobo e a cabra estão na margem direita do rio, enquanto que, o fazendeiro e a couve estão se dirigindo para a outra margem.

Por padronização, admitiremos as seguintes condições:

Condição legal = NL 1  
Condição ilegal = NL 0

Podemos considerar condição legal todas as vezes que houver a presença do fazendeiro, ou então, lobo e couve em uma margem, no entanto, lobo e cabra em uma margem sem a presença do fazendeiro a condição será ilegal e assim por diante.

## **RESOLUÇÃO PARA O PROBLEMA DO FAZENDEIRO**

Iniciaremos com a elaboração da tabela da verdade que representa a situação proposta; toda a condição ilegal será representada na saída pelo nível lógico 0 enquanto que toda a condição legal será representada pelo nível lógico 1. Ao se ligar um led na saída, o mesmo deverá permanecer aceso para toda condição legal, logo, uma condição ilegal o apagará.

A (lobo)	B (cabra)	C (couve)	D (fazendeiro)	Saída
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0
1	1	1	1	1

Analisando a tabela da verdade verificamos que as linhas correspondentes a condição legal são: 0,2,4,5,7,8,10,11,13 e 15, ou seja, com NL = 1 na saída.

A partir daí podemos então escrever a função em minitermos, lembrando que, minitermos representa soma de produtos a partir de níveis lógicos "1" na saída da tabela da verdade.

Temos então:

$$f(ABCD) = \sum m(0,2,4,5,7,8,10,11,13,15)$$

ou

$$S = A'B'C'D' + A'B'CD' + A'BC'D' + A'BC'D + A'BCD + AB'C'D' + AB'CD' + AB'CD + ABC'D + ABCD$$

Simplificando graficamente pelo M.K.

	00	01	11	10
00	0 1	4 1	12	8 1
01	1	5 1	13 1	9
11	3	7 1	15 1	11 1
10	2 1	6	14	10 1

Obtemos então os seguintes agrupamentos: **opção 1**

$m(0,2,8,10)$   
 $m(5,7,13,15)$   
 $m(0,4)$   
 $m(15,11)$

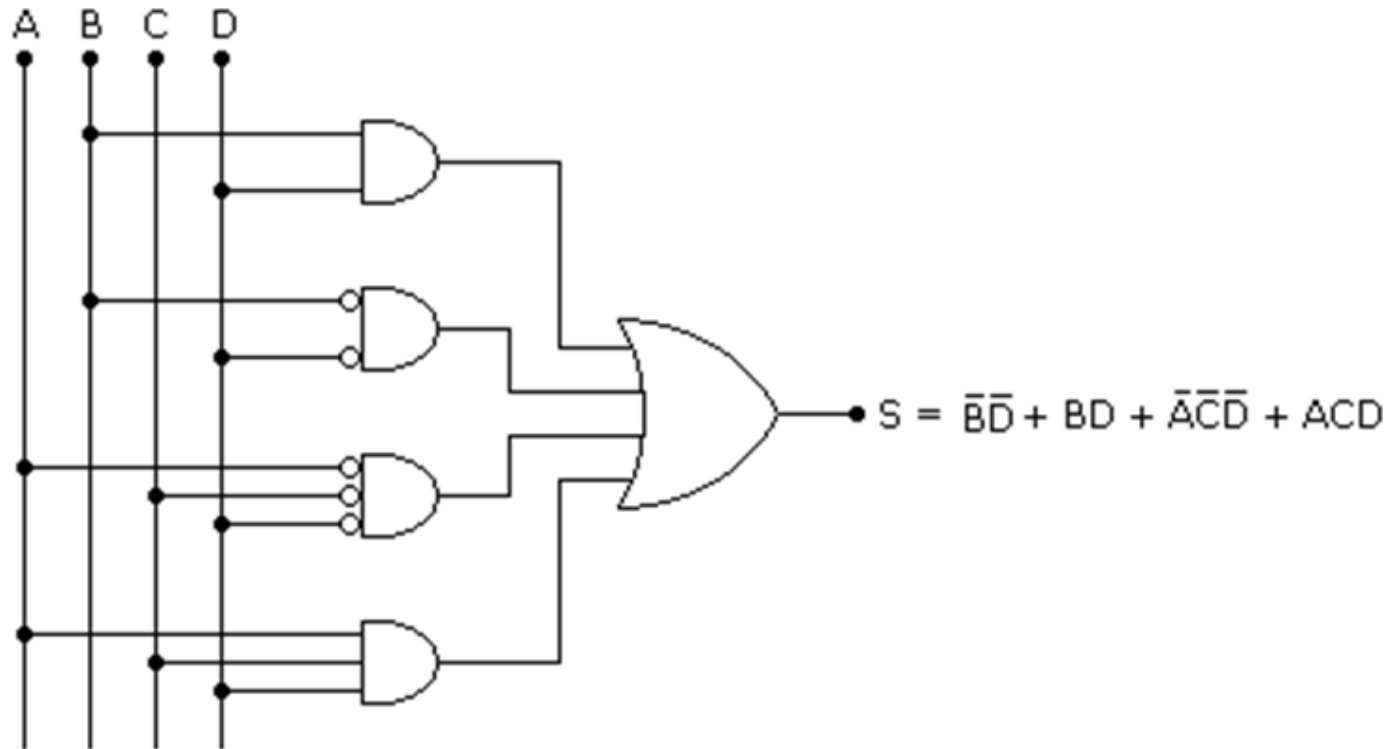
ou **opção 2**

$m(0,2,8,10)$   
 $m(5,7,13,15)$   
 $m(4,5)$   
 $m(11,10)$

Resolvendo pela **opção 1**, a expressão booleana do circuito é:  $S = BD + B'D' + A'C'D' + ACD$ , cujo circuito é mostrado a seguir:

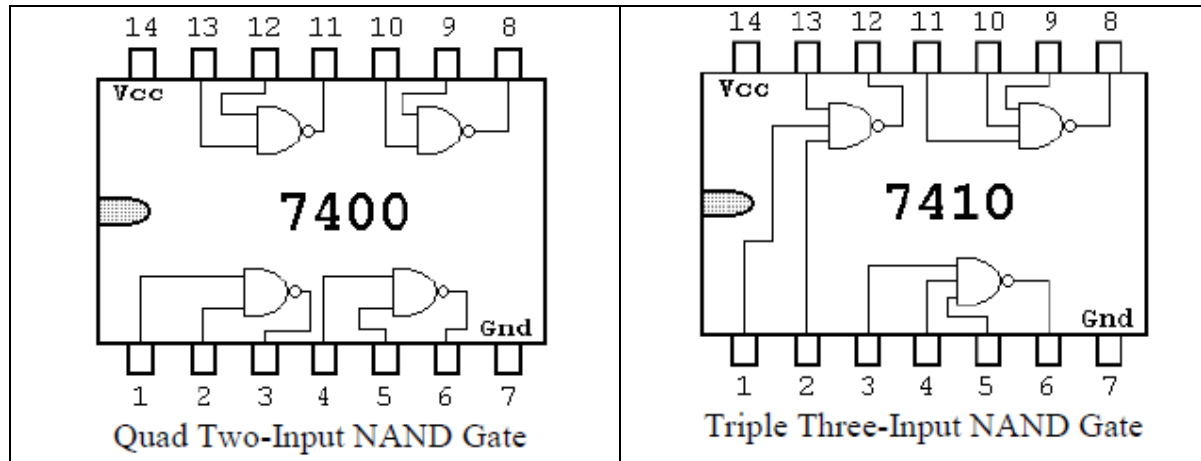
$$\begin{array}{l}
 \overline{A}B\overline{C}\overline{D} \\
 \overline{A}BC\overline{D} \\
 A\overline{B}\overline{C}\overline{D} \\
 ABC\overline{D} \\
 \hline
 \Rightarrow \overline{B}\overline{D}
 \end{array}
 \quad
 \begin{array}{l}
 \overline{A}B\overline{C}\overline{D} \\
 \overline{A}BC\overline{D} \\
 A\overline{B}\overline{C}\overline{D} \\
 ABC\overline{D} \\
 \hline
 \Rightarrow BD
 \end{array}
 \quad
 \begin{array}{l}
 \overline{A}\overline{B}\overline{C}\overline{D} \\
 \overline{A}B\overline{C}\overline{D} \\
 \hline
 \Rightarrow \overline{A}\overline{C}\overline{D} \\
 m(0,4)
 \end{array}$$

$m(0,2,8,10)$        $m(5,7,13,15)$        $m(15,11)$



Podemos verificar que se trata de um circuito com funções OR, AND e NOT.

No entanto, a proposta é utilizar apenas funções NAND, com 3 circuitos integrados comerciais 74LS00 e 1 circuito integrado comercial 74LS10, todos portas NAND.



Aplicando o teorema de "De Morgan" na expressão simplificada:

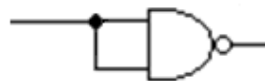
Aplicando De Morgan  

$$S = BD + \bar{B}\bar{D} + \bar{A}\bar{C}\bar{D} + ACD$$

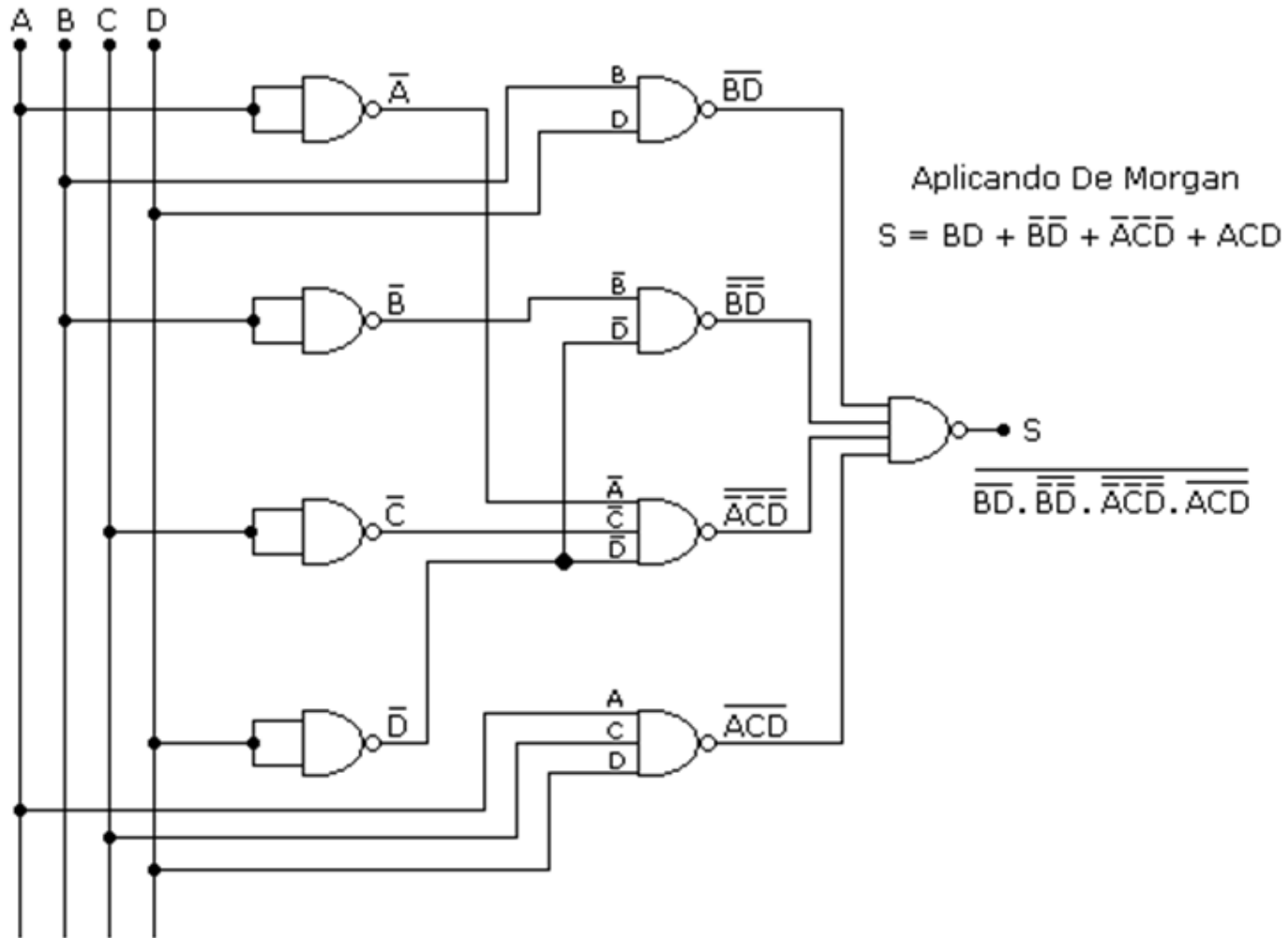


$$\overline{\bar{B}\bar{D} \cdot \bar{A}\bar{C}\bar{D} \cdot ACD}$$

Lembrando que ao se interligar as duas entradas em uma porta NAND ela se comportará como inversor ou porta NOT.



O circuito para simulação é mostrado abaixo, utilizando portas lógicas primárias:

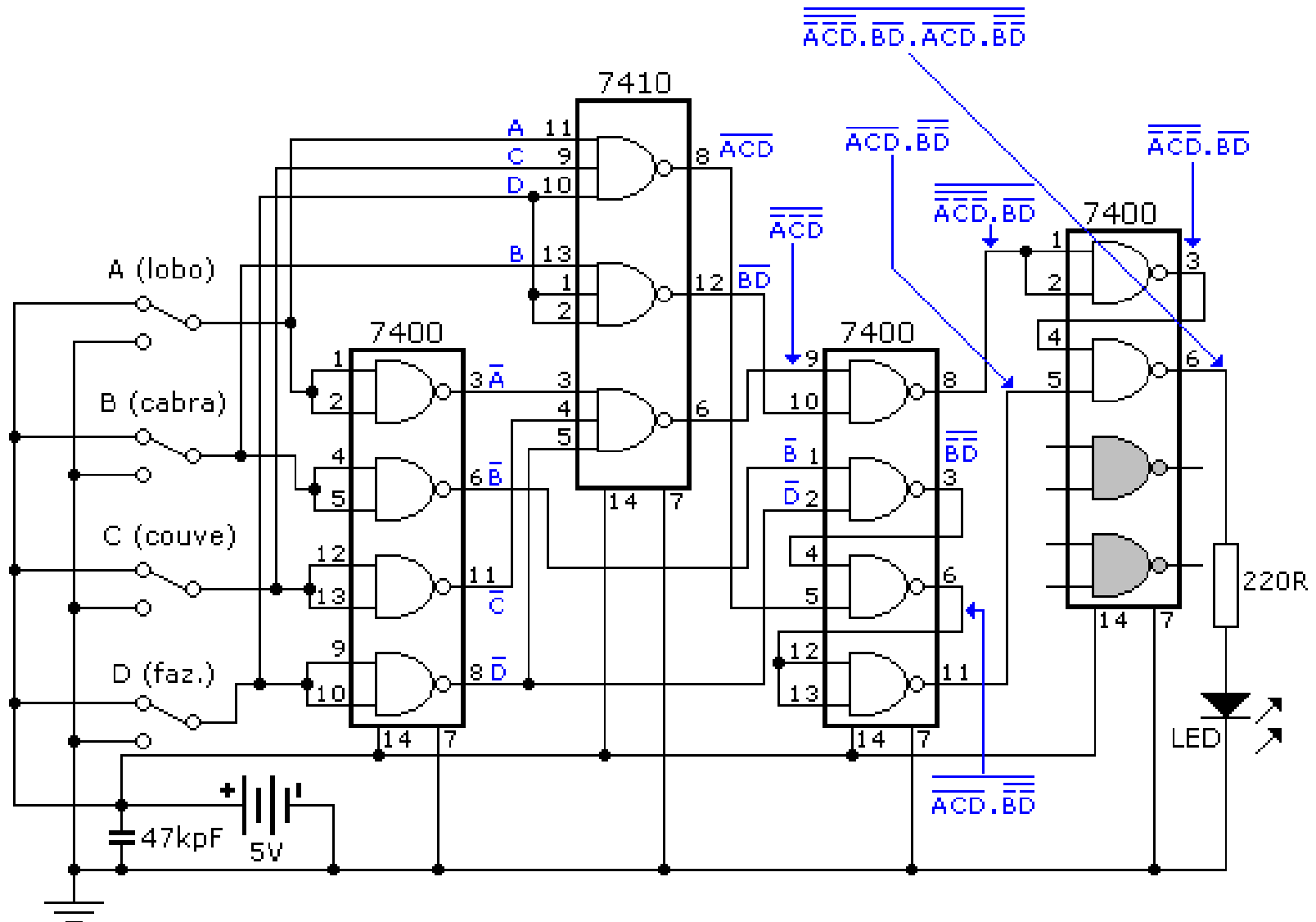


O próximo passo é converter o circuito acima, de forma a fazer com que ele opere apenas com portas NAND.

A figura a seguir mostra o esboço do circuito final, utilizando 3 CIs 7400 e 1 CI 7410.

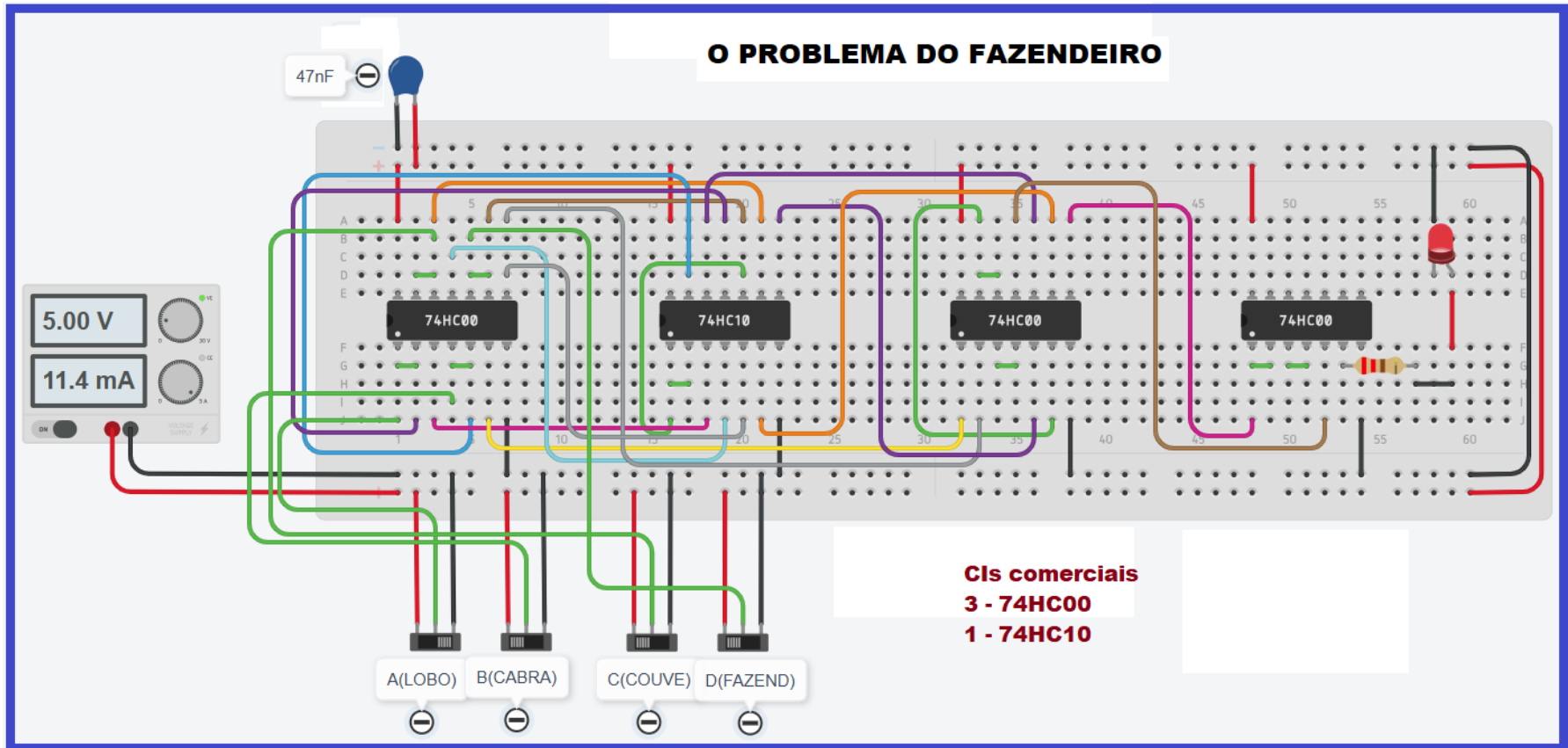




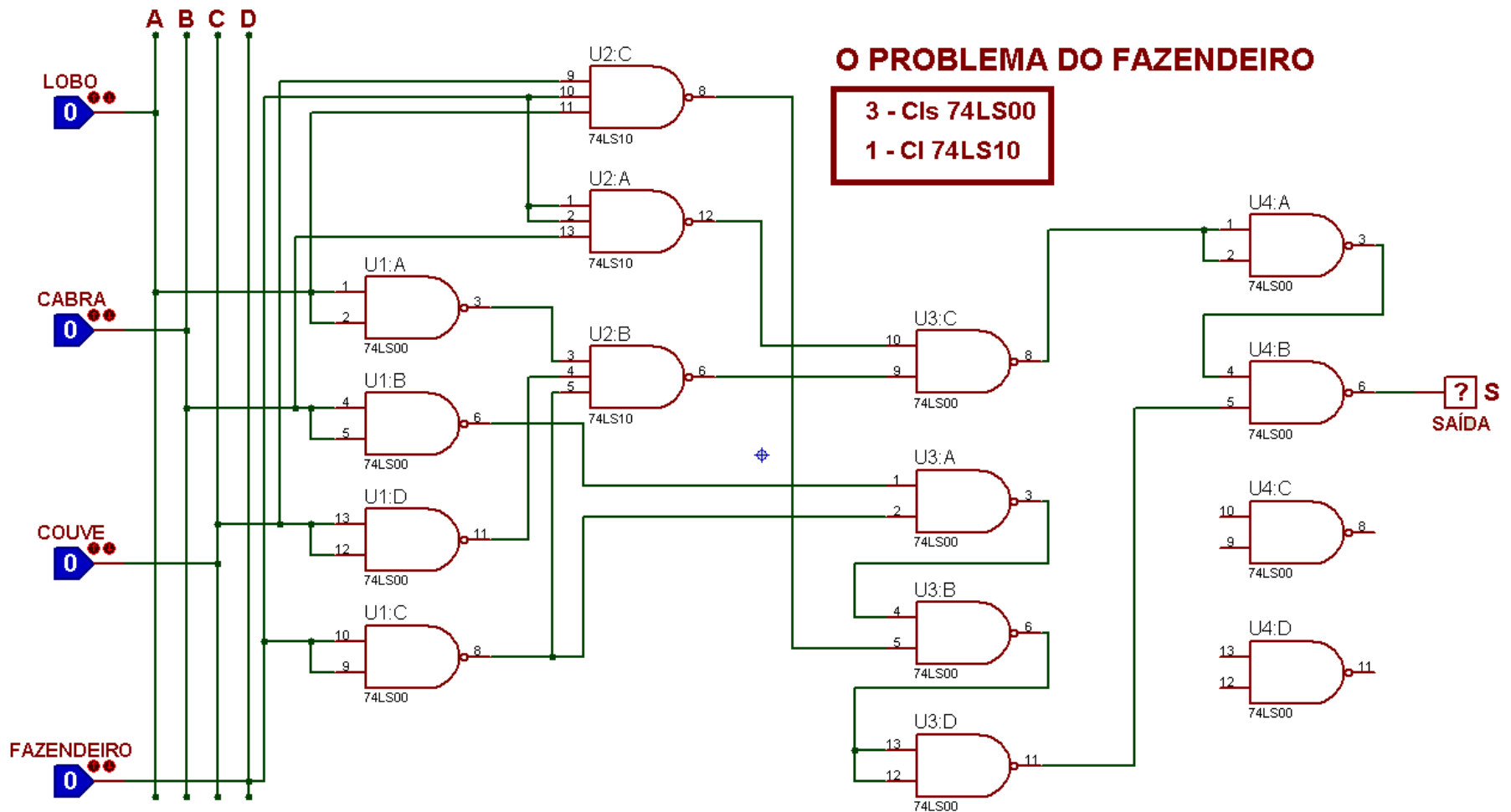


Veja a seguir o circuito montado para simulação na plataforma Tinkercad, utilizando apenas portas NAND, com CIs comerciais propostos:

**3 – CIs 74LS00**  
**1 – CI 74LS10**



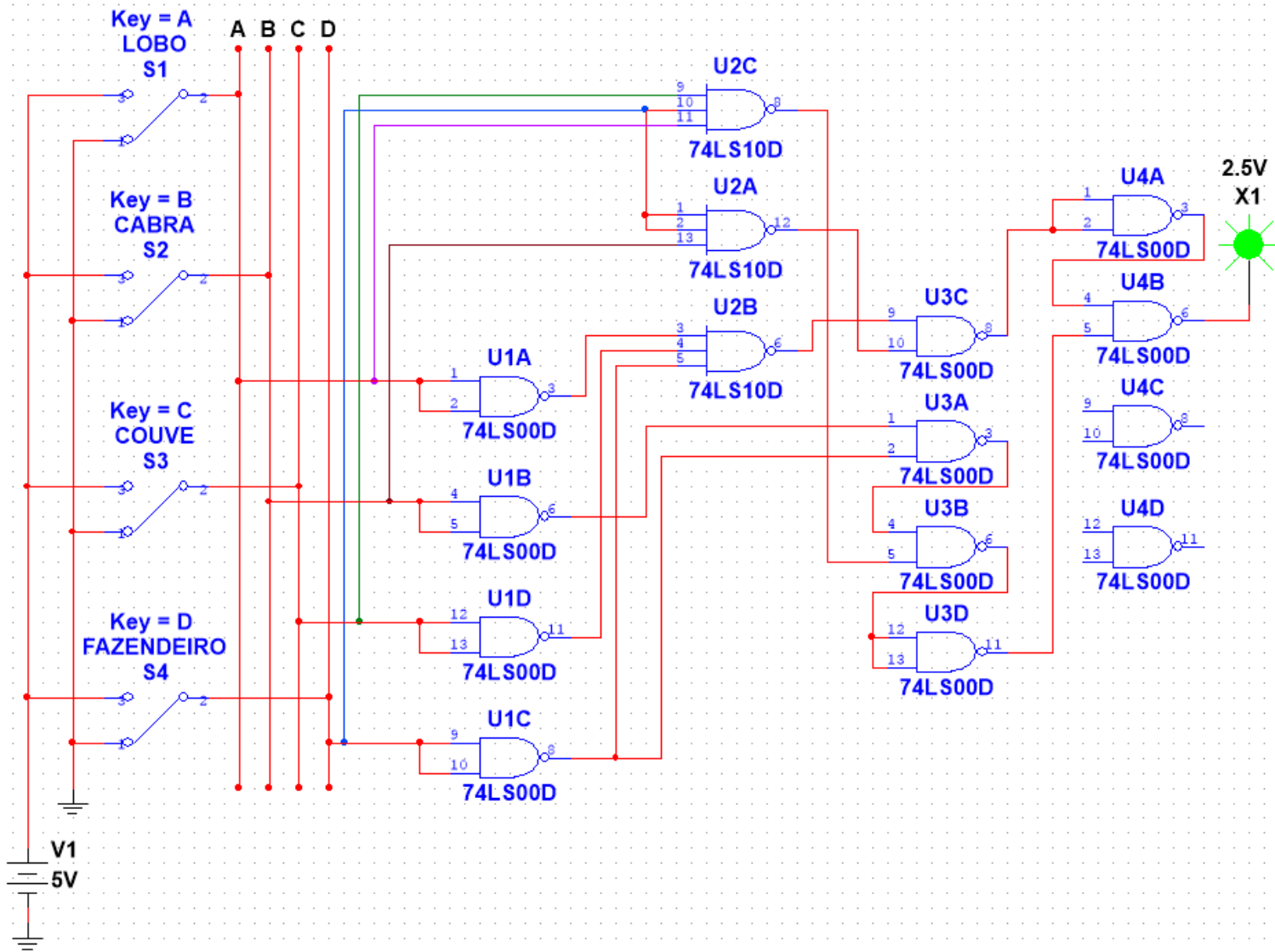
Veja a seguir o diagrama esquemático do circuito, desenvolvido no software Proteus-ISIS, que deu origem a montagem acima mostrada.



*diagrama esquemático desenvolvido no PROTEUS-ISIS*

A figura a seguir ilustra o mesmo circuito, porém montado no software Multisim:

**3 – CIs 74LS00**  
**1 – CI 74LS10**



*diagrama esquemático desenvolvido no MULTISIM*