

A UTILIZAÇÃO DA ÁLGEBRA BOOLEANA NA SIMPLIFICAÇÃO DE CIRCUITOS LÓGICOS.

Breno R. de S. Costa¹, Paulo C. L. da Silva², Antonio R. G. Garcia³

Resumo: Os dispositivos eletrônicos que nos rodeiam são constituídos por inúmeros circuitos eletrônicos, que operam de acordo com a álgebra booleana. Ela, por sua vez, é composta por postulados, identidades e propriedades que podem ser utilizados para a simplificação destes circuitos e, além deste, há a simplificação através do mapa de Karnaugh, que é um método de mapeamento visual. Através da expressão de um circuito lógico combinacional realizou-se uma comparação entre esses métodos de simplificação. Com isso, foi percebido que a simplificação traz benefícios para o setor da eletrônica, independentemente do método utilizado, e que o mapa Karnaugh é um método mais prático, simples e rápido quando comparado com a álgebra booleana.

Palavras-chave: Álgebra de Boole. Mapa de Karnaugh. Eletrônica digital.

1. INTRODUÇÃO

Em 1854, um matemático britânico chamado George Boole apresentou, através de sua obra intitulada como *An Investigation of the Laws of Thought* [1], um sistema matemático de análise lógica que ficou conhecido como Álgebra de Boole. Ele é considerado uma figura importante da lógica principalmente por ter iniciado a lógica moderna [2]. A álgebra de Boole utiliza-se de símbolos e operadores para descrever o modo como se toma decisões a partir de condições verdadeiras e falsas, podendo ser representado pelo sistema binário, que ficou conhecido como lógica booleana. Boole estabeleceu regras que corresponderiam ao processamento de informação na mente humana. Para Boole, o raciocínio era composto de operações: associação de coisas simples para a formação de algo mais complexo ou desmembramento de algo complexo em coisas simples. As operações realizadas pelo pensamento seriam equivalentes aos conceitos relacionados pelas operações lógicas OR (OU), NOT (NÃO) e AND (E), descrevendo o relacionamento das entradas (circunstâncias) e das saídas (decisões) dos circuitos lógicos [3]. Hoje, mecanismos de buscas populares em todo mundo utilizam-se da ferramenta chamada busca booleana que, através da combinação de informações presentes nos documentos por meio dos operadores lógicos, tornam a busca mais detalhada e/ou restrita [4].

Essa teoria não possuía nenhuma utilidade prática até que em 1938, quase um século depois, o matemático e engenheiro eletricitista estadunidense Claude Elwood Shannon percebeu a semelhança entre lógica booleana e a lógica dos circuitos e publicou um trabalho intitulado como *Symbolic Analysis of Relay and Switching* [5] tomando como base as teorias da álgebra de Boole para solucionar problemas de circuitos. Isso transformou radicalmente a abordagem lógica de circuitos operados por relés e foram estendidas a componentes eletrônicos como válvulas termoiônicas, aos transistores e até a um componente mais moderno conhecido como circuito integrado [6]. Os circuitos digitais executam funções booleanas por meio de portas lógicas e muitas dessas funções admitem simplificação que podem ser feitas basicamente por dois métodos: o primeiro é a aplicação de postulados, identidades e propriedades da álgebra de Boole e o segundo é o método do mapa de Karnaugh, desenvolvido por Maurice Karnaugh, que permite a simplificação de funções booleanas de forma mais simples [7]. A depender do número de variáveis, o primeiro método se torna mais tedioso e cansativo ficando mais suscetível a erros e não garantindo o menor circuito e uma ferramenta relativamente simples como mapa de Karnaugh pode ser bem mais útil [8, 9]. A partir da álgebra de Boole também foi desenvolvida uma ferramenta pedagógica chamada Jogos de Boole, visando o desenvolvimento da capacidade de raciocínio lógico através de histórias construídas a partir de estruturas lógico-matemáticas, na forma de enigmas ou problemas resolvidos com jogos de cartas [10].

A álgebra de Boole está presente em projetos de sistemas de segurança residencial, empresarial e automobilístico, caixas eletrônicos de banco, sistema de iluminação pública, portas automatizadas associadas a sensores, dentre outros, todos por meio da utilização de portas lógicas [11, 12]. Com o advento de novas tecnologias e a produção de aparelhos ultrafinos como televisões, smartphones e afins foi observada a necessidade de obter otimização, sendo possível através da simplificação, em projetos de circuitos digitais por meio da minimização da quantidade de portas lógicas que, além da diminuição de espaço ocupado, há redução do consumo de energia, bem como a aplicação da metodologia *troubleshooting* [3]. Essa metodologia estabelece procedimentos que permitem a identificação e solução de falhas seguindo um passo a passo ou um fluxograma não necessitando

um conhecimento técnico específico [13, 14]. Dessa forma, objetiva-se apresentar os conceitos que envolvem a álgebra de Boole, explicar os métodos de simplificação e compará-los realizando a simplificação de uma função lógica.

2. A ÁLGEBRA BOOLEANA

O filho de um sapateiro, George Boole, nasceu em Lincoln, Inglaterra, no ano de 1815. Autodidata, traduziu poemas do latim para o inglês com apenas catorze anos, sua base de matemática foi dada pelo seu pai. Aos dezesseis anos começou a dar aula em escolas primárias para ajudar na renda da família, aos vinte anos abriu uma escola para lecionar matemática. Em 1849 se tornou professor de matemática da *Queen's College* (atualmente *University College Cork*) na Irlanda. Produziu os mais variados trabalhos matemáticos, mas ficou conhecido pelo seu trabalho de 1854 [1] que fundamentou o que ficaria conhecida como álgebra booleana.

Na álgebra booleana as constantes e variáveis admitem apenas dois valores distintos, 0 ou 1. Eles não representam números de forma efetiva, mas sim um estado conhecido como nível lógico. Na lógica digital também são utilizados outros termos alternativos para designar esses níveis lógicos, como mostrado na tabela a seguir.

Tabela 1. Nomenclaturas dos níveis lógicos. (Adaptado de Tocci)

Nível Lógico 0	Nível Lógico 1
Baixo	Alto
Falso	Verdadeiro
Não	Sim
Desligado	Ligado
Aberto	Fechado

Sentenças matemáticas compostas por variáveis booleanas são chamadas de expressões booleanas. A álgebra booleana possui três postulados, são eles: postulado da complementação, postulado da adição e postulado da multiplicação. Esses postulados fundamentam, respectivamente, as três operações lógicas básicas: NOT (NÃO), OR (OU) e AND (E). Para explorar esse conteúdo será utilizado como base o trabalho de Idoeta e Capuano [15].

Uma ferramenta considerável que auxiliará no entendimento deste conteúdo é a chamada Tabela Verdade, que é um mapa onde coloca-se todas as situações possíveis no que tange os níveis lógicos da(s) variáveis de entrada e os seus respectivos resultados de saída, representado pela letra S. Um exemplo de tabela verdade pode ser visto na figura a seguir.

Entradas		Saída
A	B	S
0	0	1
0	1	0
1	0	1
1	1	0

Figura 1. Exemplo de tabela verdade. (Adaptado de Tocci)

2.1. Funções lógicas

Antes, é necessário esclarecer que a eletrônica digital se divide basicamente em dois campos de estudo: lógica combinacional e lógica sequencial. Na lógica combinacional, a saída depende apenas dos valores atuais de entrada e, por isso, constituem uma classe básica. Já na lógica sequencial, a saída não só depende dos valores de entrada, mas também de informações armazenadas anteriormente em uma memória. Será explorada, a seguir, a lógica combinacional, pois constitui uma parte fundamental deste trabalho.

2.1.1. Função NOT

A função NOT é a que inverte ou complementa o estado da variável de entrada: se o nível lógico de entrada estiver 0, o nível lógico de saída S será 1, e se o nível lógico de entrada estiver em 1, o nível lógico de saída S será 0. A função algébrica que representa essa operação é dada por:

$$S = \bar{A} \quad (1)$$

A tabela verdade da função NOT (NÃO) pode ser vista a seguir:

Tabela 2. Tabela verdade:
função NOT. (Autoria Própria)

A	$S = \bar{A}$
0	1
1	0

A porta NOT é um circuito que executa a função NOT e é representado como na figura:

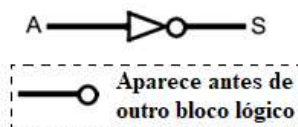


Figura 2. Representação porta NOT. (Autoria Própria)

2.1.2. Função OR

A função OR terá uma saída de nível lógico 0 se, e somente se, todas as variáveis de entradas estiverem em nível lógico 0 e uma saída com nível lógico 1 se pelo menos uma das variáveis de entrada comportar o nível lógico 1. A representação dessa operação, para duas variáveis de entrada, é dada pela seguinte função lógica:

$$S = A + B \quad (2)$$

onde a operação é lida como “ $S = A$ ou B ” e sua tabela verdade pode ser vista na tabela a seguir.

Tabela 3. Tabela verdade:
função OR. (Autoria Própria)

A	B	$S = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

A porta OR executa a função que recebe o mesmo nome. Ela é representada, na prática, pela seguinte figura:



Figura 3. Representação porta OR. (Autoria Própria)

2.1.3. Função AND

Já a função AND executa a multiplicação do nível lógico de duas ou mais variáveis de entrada. Neste caso o nível lógico de saída só será 1 se, e somente se, todas as variáveis de entrada comportarem o nível lógico 1, caso contrário o nível lógico de saída será 0. A representação algébrica para esta operação, com duas variáveis, é dada por:

$$S = A \cdot B \quad (3)$$

que é lida como “ $S = A$ e B ”. A tabela verdade para esta função pode ser vista a seguir.

Tabela 4. Tabela verdade:
função AND. (Autoria Própria)

A	B	$S = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

A porta AND é representada pelo símbolo da figura a seguir:



Figura 4. Representação porta AND. (Autoria Própria)

2.1.4. Função NOR

Essa função é composta pelas funções OR e NOT, isto é, a função OR invertida. Sua representação pode ser vista a seguir:

$$S = \overline{(A + B)} \quad (4)$$

A tabela verdade da função NOR pode ser visto na tabela:

Tabela 5. Tabela verdade:
função NOR. (Autoria Própria)

A	B	$S = \overline{(A + B)}$
0	0	1
0	1	0
1	0	0
1	1	0

A representação simbólica da função é vista da figura.

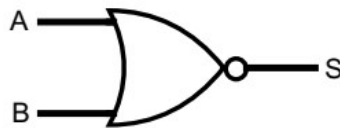


Figura 5. Representação porta NOR. (Autoria Própria)

2.1.5. Função NAND

A função NAND nada mais é que a composição das funções AND e NOT, ou seja, é a inversão da função AND. Ela é representada algebricamente pela equação:

$$S = \overline{(A \cdot B)} \quad (5)$$

A tabela verdade desta função pode ser visto a seguir.

Tabela 6. Tabela verdade:
função NAND. (Autoria Própria)

A	B	$S = \overline{(A \cdot B)}$
0	0	1
0	1	1
1	0	1
1	1	0

A função NAND é representada graficamente como visto na figura.

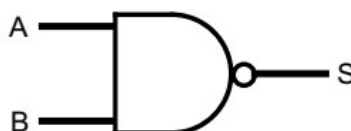


Figura 6. Representação porta NAND. (Autoria Própria)

2.2. Postulados da álgebra de Boole

A combinação das portas lógicas supracitas formam circuitos lógicos. Como já citado, os circuitos lógicos admitem, muitas vezes, simplificações. Para isto é necessário um estudo dos postulados, identidades e propriedades da álgebra de Boole. Vale destacar que é na álgebra de Boole estão todos os fundamentos da eletrônica digital.

2.2.1. Postulado da Complementação

Este postulado expõe as regras de complementação na álgebra Booleana:

1º) Se $A = 0 \rightarrow \bar{A} = 1$

2º) Se $A = 1 \rightarrow \bar{A} = 0$

Através deste postulado, pode-se estabelecer uma identidade:

Tabela 7. Identidade do postulado da complementação. (Autoria Própria)

Identidade	Possibilidades	Observação
$\bar{\bar{A}} = A$	$A = 0 \rightarrow \bar{A} = 1 \rightarrow \bar{\bar{A}} = 0$ $A = 1 \rightarrow \bar{A} = 0 \rightarrow \bar{\bar{A}} = 1$	Nota-se que, ao efetuar a operação em uma variável anteriormente invertida, o resultado sempre será igual ao valor inicial da variável.

Se $A = 1$, então tem-se que $\bar{A} = 0$ e, por conseguinte, se $\bar{A} = 0 \rightarrow \bar{\bar{A}} = 1$. O mesmo acontece se $A = 0$, logo $\bar{A} = 1$ e, por consequência, se $\bar{A} = 1 \rightarrow \bar{\bar{A}} = 0$.

Chama-se de $\bar{A}$ o complemento de A. Esta barra sobre a letra que representa a variável significa a inversão desta e se lê, para a variável A, “A barrado/barra” ou “Não A”. Esse postulado reflete a função lógica NOT (NÃO).

2.2.2. Postulado de Adição

Postulado que dita as regras de adição no âmbito da Álgebra Booleana.

1º) $0 + 0 = 0$

2º) $0 + 1 = 1$

3º) $1 + 0 = 1$

4º) $1 + 1 = 1$

Pode-se, através do postulado de adição, estabelecer mais algumas identidades algébricas, são elas:

Tabela 8. Identidades do postulado da adição. (Autoria Própria)

Identidade	Possibilidades	Observação
$A + 0 = A$	$A = 0 \rightarrow 0 + 0 = 0$ $A = 1 \rightarrow 1 + 0 = 1$	Nota-se que, independentemente do valor da variável, o resultado sempre será igual a variável A.
$A + 1 = 1$	$A = 0 \rightarrow 0 + 1 = 1$ $A = 1 \rightarrow 1 + 1 = 1$	Nota-se que, independentemente do valor da variável, o resultado sempre será igual a 1.
$A + A = A$	$A = 0 \rightarrow 0 + 0 = 0$ $A = 1 \rightarrow 1 + 1 = 1$	Nota-se que, independentemente do valor da variável, o resultado sempre será igual a variável A.
$A + \bar{A} = 1$	$A = 0 \rightarrow \bar{A} = 1 \rightarrow 0 + 1 = 1$ $A = 1 \rightarrow \bar{A} = 0 \rightarrow 1 + 0 = 1$	Nota-se que, independentemente do valor da variável, o resultado sempre será igual a 1.

A porta lógica que executa este postulado é o da função OR (OU).

2.2.3. Postulado da Multiplicação

Já este postulado refere-se as regras da multiplicação na álgebra de Boole:

1º) $0 \cdot 0 = 0$

2º) $0 \cdot 1 = 0$

3º) $1 \cdot 0 = 0$

4º) $1 \cdot 1 = 1$

E, através do postulado da multiplicação, pode-se estabelecer mais identidades, são elas:

Tabela 9. Identidades do postulado da multiplicação. (Autoria Própria)

Identidade	Possibilidades	Observação
$A \cdot 0 = 0$	$A = 0 \rightarrow 0 \cdot 0 = 0$ $A = 1 \rightarrow 1 \cdot 0 = 0$	Nota-se que, independentemente do valor da variável, o resultado sempre será igual a 0.
$A \cdot 1 = A$	$A = 0 \rightarrow 0 \cdot 1 = 0$ $A = 1 \rightarrow 1 \cdot 1 = 1$	Nota-se que, independentemente do valor da variável, o resultado sempre será igual a variável A.
$A \cdot A = A$	$A = 0 \rightarrow 0 \cdot 0 = 0$ $A = 1 \rightarrow 1 \cdot 1 = 1$	Nota-se que, independentemente do valor da variável, o resultado sempre será igual a variável A.
$A \cdot \bar{A} = 0$	$A = 0 \rightarrow \bar{A} = 1 \rightarrow 0 \cdot 1 = 0$ $A = 1 \rightarrow \bar{A} = 0 \rightarrow 1 \cdot 0 = 0$	Nota-se que, independentemente do valor da variável, o resultado sempre será igual a 0.

O postulado da multiplicação é executado pela função lógica AND (E).

2.3. Propriedades

Adiante, será mostrado as principais propriedades algébricas, elas são importantes para a realização do manuseio e da simplificação de expressões booleanas. Essas propriedades que serão utilizadas na álgebra de Boole, são as mesmas da matemática tradicional, são elas:

- a) Propriedade Comutativa $\rightarrow$ Adição: $A + B = B + A$
Multiplicação: $A \cdot B = B \cdot A$
- b) Propriedade Associativa $\rightarrow$ Adição: $A + (B + C) = (A + B) + C = A + B + C$
Multiplicação: $A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C$
- c) Propriedade Distributiva $\rightarrow A \cdot (B + C) = A \cdot B + A \cdot C$

A propriedade distributiva pode ser facilmente verificada utilizando uma tabela verdade, desde que seja analisada todas as possibilidades, como visto na tabela a seguir:

Tabela 10. Tabela verdade: propriedade distributiva. (Autoria Própria)

A	B	C	$S = A \cdot (B + C)$	$S = A \cdot B + A \cdot C$
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	1
1	1	0	1	1
1	1	1	1	1

2.4. Teoremas de De Morgan

O matemático e lógico britânico Augustus De Morgan, desenvolveu teoremas muito úteis na simplificação das expressões booleanas que são bastantes empregados na prática, inclusive no desenvolvimento de circuitos digitais. Com esses teoremas é possível converter operações lógicas OR em operações lógicas AND e vice versa.

2.4.1. Primeiro Teorema

O complemento do produto é igual à soma dos complementos:

$$(\overline{A \cdot B}) = \bar{A} + \bar{B} \quad (6)$$

A prova deste teorema pode ser vista na tabela verdade a seguir.

Tabela 11. Tabela verdade: primeiro teorema de De Morgan. (Autoria Própria)

A	B	$S = (\overline{A \cdot B})$	$S = \overline{A} + \overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Nota-se a igualdade em ambas as saídas S para todas as possibilidades. O teorema pode ser estendido para mais de duas variáveis.

2.4.2. Segundo Teorema

O complemento da soma é igual ao produto dos complementos:

$$(\overline{A + B}) = \overline{A} \cdot \overline{B} \quad (7)$$

A prova desta igualdade pode ser vista na seguinte tabela verdade.

Tabela 12. Tabela verdade: segundo teorema de De Morgan. (Autoria Própria)

A	B	$S = \overline{A} \cdot \overline{B}$	$S = \overline{(A + B)}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Nota-se, novamente, que o nível lógico das saídas é igual em todas as possibilidades. Esta também pode ser estendido para mais de duas variáveis.

2.5. Identidades Auxiliares

Há mais três identidades muito úteis para a realização de simplificação de expressões booleanas. Estas identidades e suas provas podem ser vistas na seguinte tabela.

Tabela 13. Identidades auxiliares e suas provas. (Autoria Própria)

Identidade	Prova
$A + A \cdot B = A$	$A + A \cdot B$ $= A(1 + B) \rightarrow$ evidenciar o A no primeiro termo $= A \cdot 1 \rightarrow$ identidade $1 + B = 1$ $\therefore A + AB = A$
$(A + B) \cdot (A + C) = A + B \cdot C$	$(A + B) \cdot (A + C)$ $= A \cdot A + A \cdot C + A \cdot B + B \cdot C \rightarrow$ propriedade distributiva $= A + A \cdot C + A \cdot B + B \cdot C \rightarrow$ identidade $A \cdot A = A$ $= A \cdot (1 + B + C) + B \cdot C \rightarrow$ Propriedade distributiva $= A \cdot 1 + B \cdot C \rightarrow$ identidades $1 + X = 1$ e $A \cdot 1 = A$ $\therefore (A + B) \cdot (A + C) = A + B \cdot C$
$A + \overline{A}B = A + B$	$A + \overline{A} \cdot B = \overline{\overline{A + \overline{A} \cdot B}} \rightarrow$ identidade $\overline{\overline{X}} = X$ $= \overline{[\overline{A} \cdot (\overline{A \cdot B})]} \rightarrow$ 2º teorema de De Morgan $= \overline{[\overline{A} \cdot (A + \overline{B})]} \rightarrow$ 1º teorema de De Morgan apenas no parêntese $= \overline{[\overline{A} \cdot A + \overline{A} \cdot \overline{B}]} \rightarrow$ propriedade distributiva $= \overline{(\overline{A} \cdot \overline{B})} \rightarrow$ identidade $\overline{A} \cdot A = 0$ $= (A + B) \rightarrow$ 1º teorema de De Morgan $\therefore A + \overline{A}B = A + B$

3. MAPA DE KARNAUGH

Maurice Karnaugh nasceu em Nova Iorque no ano de 1924, estudou matemática, ciência da computação e engenharia de telecomunicações obtendo o seu título de doutor nesta área. Trabalhou na *Bell Labs*, empresa que desenvolveu o transistor de silício, de 1952 a 1966 e lá desenvolveu o método conhecido como mapa de Karnaugh que será trabalhado nesta seção. Desde 1975 é membro do *Institute of Electrical and Electronics Engineers* (IEEE) por contribuir acerca do uso de métodos numéricos em telecomunicações.

O método do mapa de Karnaugh é utilizado para simplificar uma equação lógica ou para transformar uma tabela verdade num circuito lógico correspondente, de uma forma mais simples. O mapa de Karnaugh, assim como a tabela verdade, é preenchido com 0 e 1 podendo ser utilizado para qualquer número de variáveis de entrada, mas sua utilidade prática se limita a cinco ou seis variáveis. Recomenda-se, a partir de cinco variáveis, a utilização de softwares para solucioná-los [3]. Serão trabalhados os mapas para duas, três e quatro variáveis e eles podem ser vistos (vazios), respectivamente, na figura a seguir.

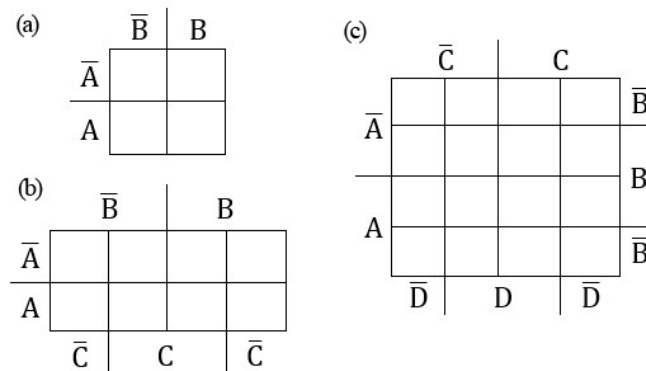


Figura 7. Mapa de Karnaugh: (a) duas variáveis, (b) três variáveis, (c) quatro variáveis. (Autoria Própria)

3.1. Para duas variáveis

Com duas variáveis existem quatro possibilidades, estas são distribuídos no diagrama na forma como é visto na figura a seguir.

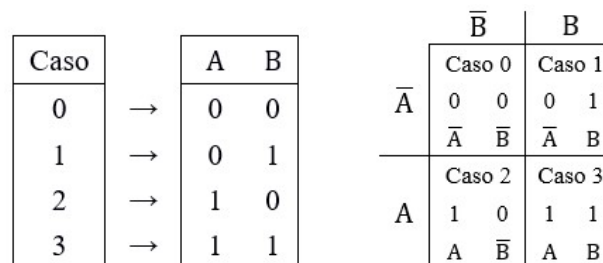


Figura 8. Estrutura do mapa de Karnaugh: duas variáveis. (Autoria Própria)

Após entender o posicionamento dos valores da tabela verdade no mapa de Karnaugh, pode-se introduzir a simplificação através deste método. O objetivo é agrupar as regiões onde a saída S é igual a 1, de modo que seja o menor número possível de agrupamentos e as regiões não agrupadas serão consideradas isoladamente. Esses agrupamentos formarão um produto (função AND) e esses agrupamentos serão somados (função OR) formando uma função lógica combinacional. Vale ressaltar, que no mapa, os lados extremos se comunicam podendo ser agrupados, como será visto adiante. Para um mapa de duas variáveis, os agrupamentos possíveis são:

- Quadra: conjunto de quatro regiões onde S é igual a 1, é o agrupamento máximo em um mapa de Karnaugh de duas variáveis, todos os casos possuem o S igual a 1. A simplificação obtida será $S = 1$. A ilustração dessa situação pode ser vista na figura a seguir:

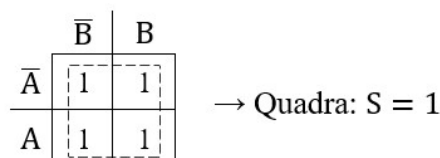


Figura 9. Mapa de Karnaugh duas variáveis: quadra. (Idoeta e Capuano)

- b) Pares: conjunto de duas regiões onde S é igual a 1, possui lado em comum, ou seja, vizinhos. Exemplo de pares agrupados na vertical e horizontal, duas entre 4 possibilidades, são vistos na figura adiante:

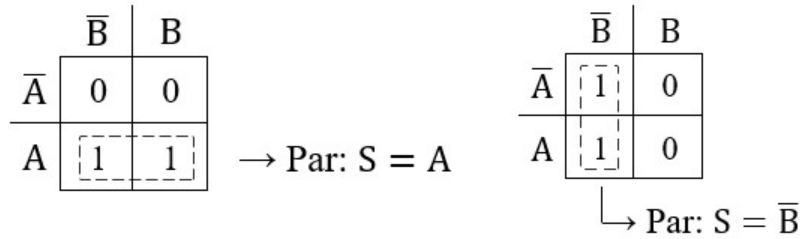


Figura 10. Mapa de Karnaugh duas variáveis: pares. (Adaptado Idoeta e Capuano)

- c) Isolados: regiões, sem vizinhança, onde S é igual a 1 e não há possibilidade dos agrupamentos supracitados. Essa situação pode ser vista na figura:

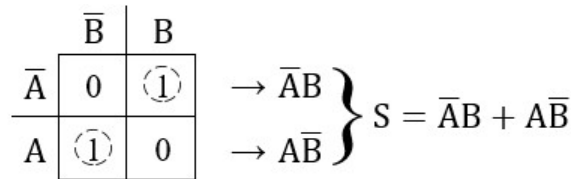


Figura 11. Mapa de Karnaugh duas variáveis: isolado. (Adaptado de Idoeta e Capuano)

3.2. Para três variáveis

Com três variáveis existem oito possibilidades, estas são distribuídos no diagrama da seguinte forma como mostra a figura.

Caso	A	B	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

	$\bar{B}$	B		
$\bar{A}$	Caso 0	Caso 1	Caso 3	Caso 2
	0 0 0	0 0 1	0 1 1	0 1 0
	$\bar{A} \ \bar{B} \ \bar{C}$	$\bar{A} \ \bar{B} \ C$	$\bar{A} \ B \ C$	$\bar{A} \ B \ \bar{C}$
A	Caso 4	Caso 5	Caso 7	Caso 6
	1 0 0	1 0 1	1 1 1	1 1 0
	A $\bar{B} \ \bar{C}$	A $\bar{B} \ C$	A B C	A B $\bar{C}$
	$\bar{C}$	C		$\bar{C}$

Figura 12. Estrutura do mapa de Karnaugh: três variáveis. (Idoeta e Capuano)

Para três variáveis os agrupamentos possíveis são:

- a) Oitava: agrupamento máximo para o mapa com três variáveis, aqui todos valem 1. Esse agrupamento pode ser visto na figura:

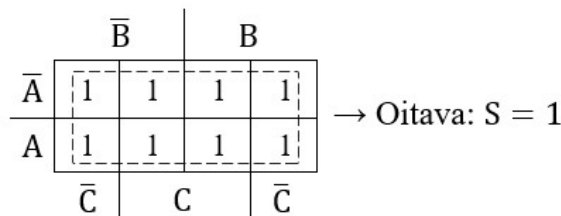


Figura 13. Mapa de Karnaugh três variáveis: oitava. (Idoeta e Capuano)

- b) Quadras: como acontece no duas variáveis, este é o agrupamento de quatro regiões, onde S é igual a 1. Algumas quadras possíveis no mapa de três variáveis podem ser vistas na figura a seguir:

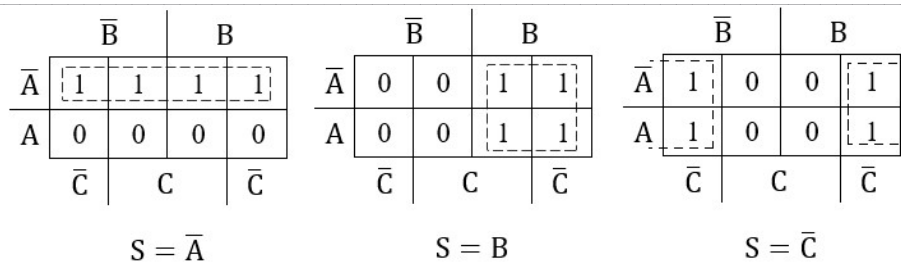


Figura 14. Mapa de Karnaugh três variáveis: quadra. (Adaptado de Idoeta e Capuano)

- c) Pares: Este também funciona como no mapa de duas variáveis, agrupamento de duas regiões onde S é igual a 1. Dois pares, entre dose pares possíveis no mapa de três variáveis, podem ser vistos na figura:

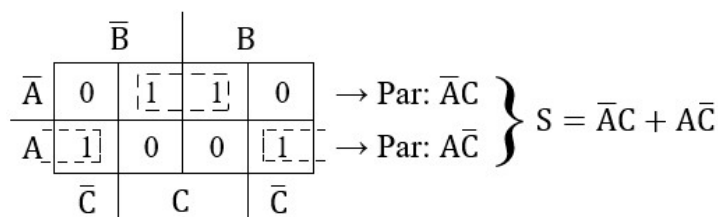


Figura 15. Mapa de Karnaugh três variáveis: pares. (Autoria Própria)

- d) Isolados: Mais um que funciona da mesma forma que no mapa de duas variáveis, são as regiões que S é igual a 1, mas não possuem vizinhança. Um exemplo no mapa de três variáveis pode ser visto na figura adiante:

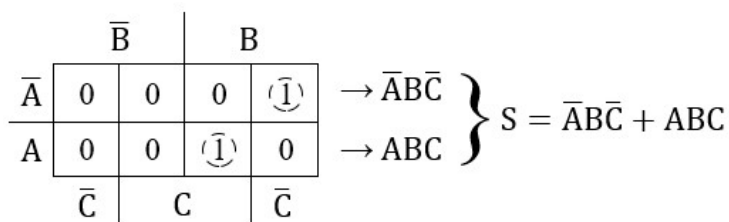


Figura 16. Mapa de Karnaugh três variáveis: isolado. (Adaptado de Idoeta e Capuano)

3.3. Para quatro variáveis

Com quatro variáveis existem dezesseis possibilidades, estas são distribuídos no diagrama na forma como é visto na figura a seguir.

Caso	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
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

	$\bar{C}$				C												
$\bar{A}$	Caso 0 0 0 0 0 $\bar{A} \bar{B} \bar{C} \bar{D}$				Caso 1 0 0 0 1 $\bar{A} \bar{B} \bar{C} D$				Caso 3 0 0 1 1 $\bar{A} \bar{B} C D$				Caso 2 0 0 1 0 $\bar{A} \bar{B} C \bar{D}$				$\bar{B}$
	Caso 4 0 1 0 0 $\bar{A} B \bar{C} \bar{D}$				Caso 5 0 1 0 1 $\bar{A} B \bar{C} D$				Caso 7 0 1 1 1 $\bar{A} B C D$				Caso 6 0 1 1 0 $\bar{A} B C \bar{D}$				
A	Caso 12 1 1 0 0 $A B \bar{C} \bar{D}$				Caso 13 1 1 0 1 $A B \bar{C} D$				Caso 15 1 1 1 1 $A B C D$				Caso 14 1 1 1 0 $A B C \bar{D}$				B
	Caso 8 1 0 0 0 $A \bar{B} \bar{C} \bar{D}$				Caso 9 1 0 0 1 $A \bar{B} \bar{C} D$				Caso 11 1 0 1 1 $A \bar{B} C D$				Caso 10 1 0 1 0 $A \bar{B} C \bar{D}$				$\bar{B}$
	$\bar{D}$				D				$\bar{D}$								

Figura 17. Estrutura do mapa de Karnaugh: quatro variáveis. (Idoeta e Capuano)

A simplificação no mapa para quatro variáveis segue o mesmo processo do mapa de três variáveis, nesse caso o principal agrupamento será o oitavado. Um mapa de quatro variáveis com oitava, quadra e par, respectivamente, pode ser visto na figura.

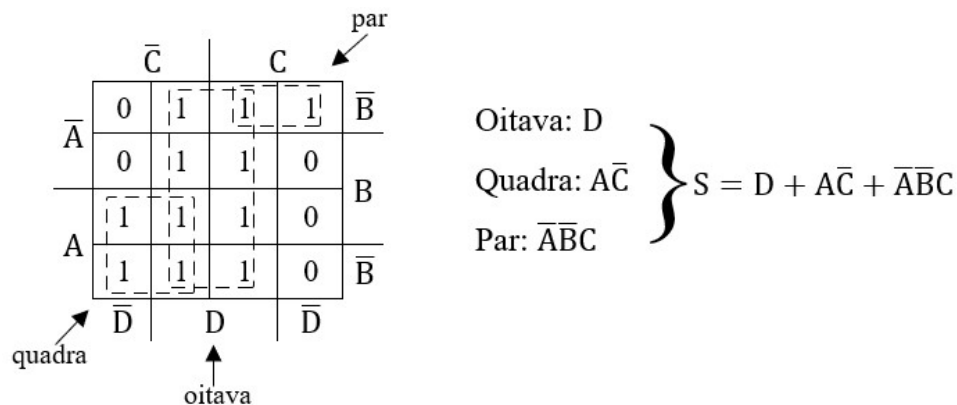


Figura 18. Mapa de Karnaugh quatro variáveis: oitava, quadra e par. (Adaptado de Idoeta e Capuano)

4. METODOLOGIA

Objetiva-se demonstrar a simplificação de uma função lógica através dos métodos tradicionais da álgebra de Boole utilizando seus postulados, identidades e propriedades e, também, realizar a simplificação pelo método do mapa de Karnaugh. Por fim, os resultados obtidos por cada método serão comparados levando em consideração a facilidade para a realização da simplificação e consequentemente verificar que o método do mapa de Karnaugh é mais rápido como visto em [7, 15], além de provar os benefícios da simplificação dentro da eletrônica digital. O desenvolvimento desse trabalho contará com o auxílio do software livre chamado Logisim, ele possibilita a simulação de circuitos lógicos através de uma interface gráfica [16]. A função lógica S a ser trabalhada será a seguinte:

$$S = \overline{[(B + C\bar{D} + \bar{D} + AC)(A + \bar{B} + \bar{C}) + \bar{B}(C + \bar{A}BC + AC)](A + B)} \quad (8)$$

A tabela verdade desta função lógica, obtida através do software Logisim, pode ser vista na tabela a seguir

Tabela 14. Tabela verdade:
função S. (Autoria própria)

A	B	C	D	S
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	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Na eletrônica digital, a simulação do circuito combinacional que executa essa função lógica, através do software Logisim, pode ser observado na seguinte imagem:

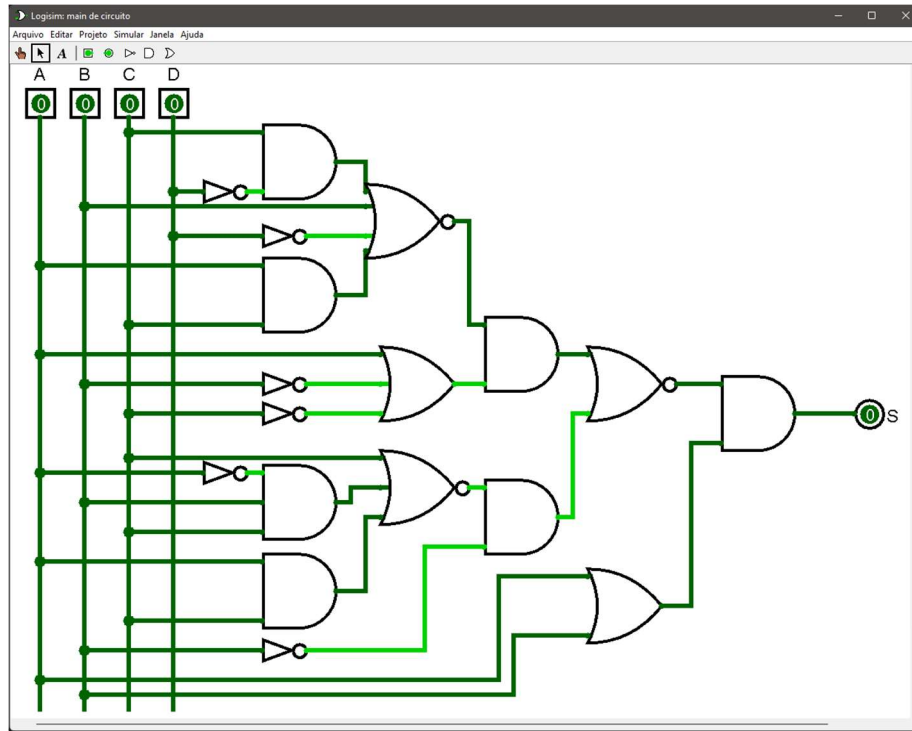


Figura 19. Circuito que executa a função lógica S. (Autoria Própria)

5. ANALISE E RESULTADOS

5.1. Simplificação através da álgebra de Boole

Dada a função lógica (8), aplicam-se o primeiro e o segundo teorema de De Morgan:

$$\overline{[(B + C\bar{D} + \bar{D} + AC) + (A + \bar{B} + \bar{C}) \cdot \bar{B} + (C + \bar{A}BC + AC)](A + B)} \quad (9)$$

Agora utiliza-se da propriedade associativa para reorganizar e aplica-se a identidade da complementação $\bar{\bar{A}} = A$, ficando com:

$$[(B + C\bar{D} + \bar{D} + AC + \overline{A + \bar{B} + \bar{C}})(B + C + \bar{A}BC + AC)](A + B) \quad (10)$$

Novamente, aplica-se o segundo teorema de De Morgan no termo destacado, obtendo:

$$[(B + C\bar{D} + \bar{D} + AC + \overline{\bar{A}\bar{B}\bar{C}})(B + C + \bar{A}BC + AC)](A + B) \quad (11)$$

Utiliza-se no termo destacado, mais uma vez, a identidade da complementação:

$$[(B + C\bar{D} + \bar{D} + AC + \bar{A}BC)(B + C + \bar{A}BC + AC)](A + B) \quad (12)$$

Neste momento é possível utilizar a identidade auxiliar $A + AB = A$ nos termos destacados:

$$\left(\frac{B + \bar{A}BC}{B} + \frac{C\bar{D} + \bar{D} + AC}{\bar{D}} \right) \left(\frac{B + \bar{A}BC}{B} + \frac{C + AC}{C} \right) (A + B) \quad (13)$$

Ficando apenas com:

$$(B + \bar{D} + AC) \underbrace{(B + C)}_{(A + B)} (A + B) \quad (14)$$

Agora pode-se utilizar mais uma identidade auxiliar $(A + B)(A + C) = A + BC$, ficando com:

$$(B + \bar{D} + AC)(B + AC) \quad (15)$$

Utiliza-se, novamente, a propriedade distributiva:

$$BB + B\bar{D} + ABC + ABC + AC\bar{D} + AACD \quad (16)$$

Neste momento pode-se aplicar duas identidades: $A + A = A$ e $AA = A$. Obtém-se:

$$\underline{B + B\bar{D}} + ABC + \underline{AC\bar{D} + AC} \quad (17)$$

A partir da identidade auxiliar $A + AB = A$, aplicada nos termos destacados, tem-se:

$$B + \underline{ABC + AC} \quad (18)$$

E é possível utilizá-la novamente, nos termos destacados, obtendo a seguinte expressão lógica simplificada que chamaremos de S_2 :

$$S_2 = B + AC \quad (19)$$

5.2. Simplificação através do mapa de Karnaugh

Com o conhecimento do método de simplificação do mapa de Karnaugh mostrado em 2.9 é possível gerar o um desses mapas para a função S a partir da sua tabela verdade presente na Tabela 14. A função lógica S possui quatro variáveis de entrada, então será um mapa com dezesseis casos e ele pode ser visto na figura.

		$\bar{C}$		C		
	$\bar{A}$	0	0	0	0	$\bar{B}$
		1	1	1	1	
		1	1	1	1	B
	A	0	0	1	1	$\bar{B}$
		$\bar{D}$		D	$\bar{D}$	

Figura 20. Mapa de Karnaugh da expressão S. (Autoria Própria)

Também com os conhecimentos da seção 2.9 é facilmente percebido que há a possibilidades de duas formas de agrupamento, das quatro possíveis para um mapa de quatro variáveis, são elas: oitava e quadra. O agrupamento pode ser visto na figura.

	$\bar{C}$		C		
	0	0	0	0	$\bar{B}$
$\bar{A}$	1	1	1	1	B
	1	1	1	1	
A	0	0	1	1	$\bar{B}$
	$\bar{D}$	D	$\bar{D}$		

Oitava: B
Quadra: AC

Figura 21. Simplificação através do Mapa de Karnaugh: expressão S. (Autoria Propria)

Agora é possível somar as expressões dos agrupamentos e então obter a função lógica simplificada S_3 :

$$S_3 = B + AC \quad (20)$$

5.3. Resultados

Primeiramente, é possível perceber que, pelos dois métodos de simplificação, as expressões simplificadas (19) e (20) são iguais. Ao fazer uma tabela verdade para a função lógica simplificada obtida e comparar os níveis lógico de saídas das duas funções, percebe-se que elas são equivalentes. Isso pode ser observado na tabela verdade comparativa a seguir.

Tabela 15. Tabela verdade comparativa. (Autoria Própria)

A	B	C	D	S	$S_2 = S_3$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

E a simulação do circuito que executa a função lógica reduzida, através do software Logisim, é visto na imagem a seguir

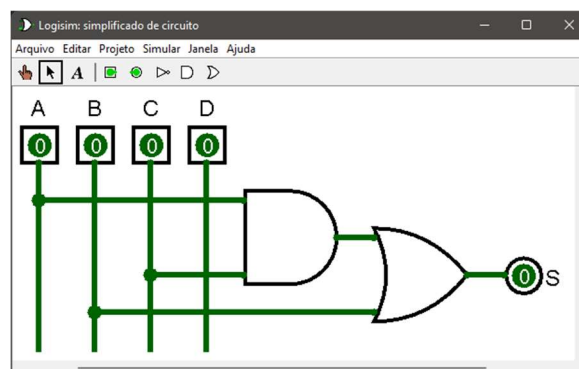


Figura 22. Circuito que executa a função lógica S_2 e S_3 . (Autoria Própria)

O circuito combinacional não simplificado possui um total de dezoito portas lógicas: seis portas NOT, sete portas AND, duas portas OR e três portas NOR. O circuito combinacional simplificado possui apenas duas portas: uma porta AND e uma porta OR.

6. CONCLUSÕES

No artigo, foi proposto uma comparação entre os métodos de simplificação para expressões booleanas e, por consequência, apresentar alguma vantagem da sua realização no âmbito da eletrônica digital. Se fez necessário, antes de qualquer coisa, a apresentação da álgebra de Boole, as funções e portas lógicas que derivam dela, como utilizá-la para reduzir expressões lógicas além da apresentação do método de simplificação utilizando o mapa de Karnaugh.

O método do mapa de Karnaugh, por ser uma técnica de mapeamento visual, muitas das vezes pode ser feita de cabeça com um pouco de prática, a depender do número de variáveis. Além disso, o mapa de Karnaugh mostra muito mais da expressão por ser a representação em matriz para a tabela verdade. Já a utilização da álgebra booleana requer todo o conhecimento dos seus postulados e identidades, caso contrário o caminho até a expressão simplificada final poderá ser bem mais longo e, em algumas situações, nem chegando na melhor simplificação possível.

No que tange a simplificação de circuitos lógicos, demonstrou-se que deve ser considerada independentemente do método a ser utilizado. O resultado obtido foi o mesmo para ambos os métodos e, nesse caso, houve a redução de dezoito portas lógicas para duas portas lógicas mantendo-se a saída equivalente, sendo considerado bastante satisfatório para a eletrônica digital por ser espacialmente mais compacto e tendo o consumo de energia reduzido, visto que cada porta lógica deve ser alimentada.

Já em relação a utilização dos métodos de simplificação, foi possível perceber que o mapa de Karnaugh obteve vantagem na simplificação da expressão lógica proposta. No quesito agilidade o método de simplificação tradicional da álgebra de Boole necessitou um total de dez etapas sendo que, em algumas destas, foram necessários o uso de mais de um postulado, identidade ou propriedade, já o método do mapa de Karnaugh, necessitou apenas três etapas: produção do mapa a partir da tabela verdade, realização de agrupamentos e, por fim, a soma desses agrupamentos formando a expressão final.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BOOLE, G. *An Investigation of the Laws of Thought*. London: Walton and Maberly, 1854.
- [2] DUARTE, A. B. A lógica de classes de Boole e a lei index. *Perspectiva Filosófica*, v. 47, n. 2, p. 150-178, 2020.
- [3] TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. *Sistemas digitais: princípios e aplicações* / Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss; revisão técnica Renato Giacomini; tradução Jorge Ritter. – 11. ed. – São Paulo: Pearson Prentice Hall, 2011.
- [4] SAKS, F. do C. *Busca booleana: teoria e prática*. 2005. 61p. Trabalho de Conclusão de Curso (Graduação em Biblioteconomia) - Universidade Federal do Paraná, Curitiba, 2005.
- [5] SHANNON, C. E. *A symbolic analysis of relay and switching circuits*. *American Institute of Electrical Engineers Transactions*, v. 57, p. 713-723, dez. 1938.
- [6] NOGUEIRA, Jurandyr Santos. *Eletrônica digital básica* / Jurandyr Santos Nogueira. - Salvador: EDUFBA, 2011. 170 p.
- [7] CAJUEIRO, J. P. C. *Mapa de Karnaugh*. 2009. 10p
- [8] SNEHA, H.L. *The Karnaugh map boolean algebraic simplification technique*. All about circuits, [s.l.], 24 de jun. de 2016. Disponível em: <<https://www.allaboutcircuits.com/technical-articles/karnaugh-map-boolean-algebraic-simplification-technique/>>. Acesso em: 24 de abr. de 2022.
- [9] SANTOS, R. B. *Simplificação usando mapa de Karnaugh*. Disponível em: <<https://www.passeidireto.com/arquivo/64123780/simplificacao-usando-mapa-de-karnaugh>>. Acesso em: 30 de abr. de 2022.
- [10] MELLO, P. M.; MELLO, D. A. *Jogos Boole: a maneira divertida de ficar inteligente*. Disponível em: <<https://jogosboole.com.br/>>. Acesso em: 30 de abr. de 2022.
- [11] SANTOS, V. R.; OLIVEIRA, C. G. A álgebra booleana presente nos circuitos lógicos. *Caderno de Graduação Ciências Exatas e Tecnológicas Unit, Aracaju*, v. 3, n. 3, p. 63-72, out. 2016.
- [12] OLIVEIRA, D. A.; CAVALCANTE, A. L. B. *Sistema de alarme residencial comercial e/ou automobilístico*. 7p. UPIS Faculdades Integradas, Brasília, DF, 2006.
- [13] SANTOS NETO, A. J. *Troubleshooting em cromatografia*. In: 2º Congresso Analítica Latin America. São Paulo, 2011.
- [14] PASSAMANI, Thiago. *O que é troubleshooting?* [s.l.], 2010. Disponível em: <<http://www.thiagopassamani.com.br/dicas/o-que-e-troubleshooting.html>>. Acesso em: 20 abr. 2022.
- [15] IDOETA, Ivan Valeije; CAPUANO, Francisco G. *Elementos de eletrônica digital*. 40ª ed. São Paulo: Érica, 2008. 544 p.
- [16] LOGISIM. v. 2.7.1. Boston: Carl Burch, 2011. Disponível em: <<http://www.cburch.com/logisim/index.html>>. Acesso em: 25 abr. 2022.