



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
CURSO INTERDISCIPLINAR DE CIÊNCIA E TECNOLOGIA

MIGUEL FERNANDES DE MELO NETO

**ANÁLISE DAS VANTAGENS DA SIMPLIFICAÇÃO DE CIRCUITOS
COMBINACIONAIS PELA MINIMIZAÇÃO DE EXPRESSÕES BOOLEANAS: UMA
SIMULAÇÃO COM MULTISIM.**

CARAÚBAS

2025

MIGUEL FERNANDES DE MELO NETO

**ANÁLISE DAS VANTAGENS DA SIMPLIFICAÇÃO DE CIRCUITOS
COMBINACIONAIS PELA MINIMIZAÇÃO DE EXPRESSÕES BOOLEANAS: UMA
SIMULAÇÃO COM MULTISIM.**

Trabalho de conclusão de curso apresentado a
Universidade Federal Rural do Semi-Árido
como requisito para obtenção do título de
Bacharel em Ciência e Tecnologia.

Orientador: Prof.^a Dra. Kelyane Barboza de
Abreu

CARAÚBAS

2025

©Todos os direitos estão reservados à Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996, e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tornar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata, exceto as pesquisas que estejam vinculadas ao processo de patenteamento. Esta investigação será base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) seja devidamente citado e mencionado os seus créditos bibliográficos.

N469 Neto, Miguel Fernandes de Melo.
 ANÁLISE DAS VANTAGENS DA SIMPLIFICAÇÃO DE
 CIRCUITOS COMBINACIONAIS PELA MINIMIZAÇÃO DE
 EXPRESSÕES BOOLEANAS: UMA SIMULAÇÃO COM MULTISIM.
 / Miguel Fernandes de Melo Neto. - 2025.
 52 f. : il.

 Orientadora: Kelyane Barboza de Abreu.
 Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência e
Tecnologia, 2025.

 1. Álgebra Booleana. 2. circuitos
combinacionais. 3. simplificação. 4. Multisim. I.
Abreu, Kelyane Barboza de, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo(a) autor(a).

 Biblioteca Campus Caraúbas
Bibliotecária: Dalvanira Brito Rodrigues
CRB: 15/700

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

MIGUEL FERNANDES DE MELO NETO

**ANÁLISE DAS VANTAGENS DA SIMPLIFICAÇÃO DE CIRCUITOS
COMBINACIONAIS PELA MINIMIZAÇÃO DE EXPRESSÕES BOOLEANAS: UMA
SIMULAÇÃO COM MULTISIM.**

Trabalho de conclusão de curso apresentado a
Universidade Federal Rural do Semi-Árido
como requisito para obtenção do título de
Bacharel em Ciência e Tecnologia.

Defendida em: 14/03/2025.

BANCA EXAMINADORA

Prof.^a Dra. Kelyane Barboza de Abreu (UFERSA)
Presidente

Prof. Dr. Antonival Lopes do Nascimento Filho (UFERSA)
Membro Examinador

Prof. Dr. Fernando Neres de Oliveira (UFERSA)
Membro Examinador

DEDICATÓRIA

Dedico este trabalho aos meus pais, Genivaldo e Maria Auxiliadora, e aos meus avós, Ricardina e, em especial, ao meu avô Antonio Costa, que já não está entre nós, mas sua memória e legado seguem me inspirando.

RESUMO

Este trabalho tem como objetivo analisar as vantagens da simplificação de circuitos combinacionais por meio da minimização de expressões booleanas, utilizando o software de simulação Multisim. A Álgebra Booleana desempenha um papel fundamental na engenharia elétrica e na computação, permitindo a otimização de circuitos digitais para reduzir o número de portas lógicas, melhorar o desempenho e diminuir o consumo de energia. Para validar essa abordagem, foram realizados experimentos simulados em circuitos de controle de ar-condicionado, iluminação, bombas d'água e alarmes automotivos. Os resultados demonstraram que a aplicação de técnicas como o Mapa de Karnaugh e os teoremas de De Morgan proporcionam circuitos mais eficientes, confiáveis e econômicos. Assim, este estudo reforça a importância do domínio da Álgebra Booleana na formação de engenheiros e profissionais da área tecnológica.

Palavras-chave: Álgebra Booleana; circuitos combinacionais; simplificação; Multisim.

ABSTRACT

This work aims to analyze the advantages of simplifying combinational circuits through Boolean expression minimization, using the Multisim simulation software. Boolean Algebra plays a fundamental role in electrical engineering and computing, enabling the optimization of digital circuits to reduce the number of logic gates, improve performance, and decrease energy consumption. To validate this approach, simulated experiments were conducted on air conditioning control, lighting, water pump, and automotive alarm circuits. The results demonstrated that applying techniques such as Karnaugh Maps and De Morgan's theorems leads to more efficient, reliable, and cost-effective circuits. Thus, this study reinforces the importance of mastering Boolean Algebra in the education of engineers and technology professionals.

Keywords: Boolean Algebra; combinational circuits; simplification; Multisim.

LISTA DE FIGURAS

Figura 1 - Tabela verdade com duas entradas	4
Figura 2 - (a) Mapas de Karnaugh e tabelas-verdade para duas variáveis	12
Figura 3 - (b) Mapas de Karnaugh e tabelas-verdade para três variáveis.....	13
Figura 4 - (c) Mapas de Karnaugh e tabelas-verdade para quatro variáveis.	14
Figura 5 - Exemplo de agrupamento de pares de 1s adjacentes.	15
Figura 6 - Exemplo de agrupamento de pares de 1s adjacentes.	15
Figura 7 - Exemplo de agrupamento de quatro de 1s adjacentes.	16
Figura 8 - Exemplo de agrupamento de quatro de 1s adjacentes.	16
Figura 9 - Exemplos de agrupamentos de oito 1s (octetos).....	17
Figura 10 - Exemplos de agrupamentos de oito 1s (octetos).....	17
Figura 11 - Exemplos de agrupamentos de oito 1s (octetos).....	18
Figura 12 - Exemplos de vários agrupamentos.	18
Figura 13 - (a) Tabela-verdade que define a operação OR; (b) símbolo de uma porta OR de duas entradas.....	21
Figura 14 - (a) Tabela-verdade para a operação AND; (b) símbolo da porta AND	22
Figura 15 - (a) Tabela-verdade; (b) símbolo para o INVERSOR (circuito NOT).	22
Figura 16 - (a) Símbolo da porta NOR; (b) circuito equivalente; (c) tabela-verdade.	23
Figura 17 - (a) Símbolo da porta NAND; (b) circuito equivalente; (c) tabela-verdade.	23
Figura 18 - Diagramas de níveis de tensão.....	25
Figura 19 - Encapsulamento comum de CIs.....	26
Figura 20 - Data sheet da porta OR do CI 7432.	27
Figura 21 - Como obter o circuito a partir da expressão booleana.....	28
Figura 22 - Circuito de controle do ar condicionado simulado no Multisim.	32
Figura 23 - Circuito de controle do ar condicionado simplificado pelo mapa K.	33
Figura 24 - Simulação do circuito controle do ar condicionado simplificado, pelo Multisim.	33
Figura 25 - Circuito de controle de iluminação simulado no Multisim.	35
Figura 26 - Circuito de controle de iluminação simplificando pelo mapa k.	36
Figura 27 - Simulação do circuito de controle de iluminação simplificado, pelo Multisim.	36
Figura 28 - Circuito de controle para bombas d'água simulado no Multisim.	39
Figura 29 - Circuito de controle para bombas d'água B1 simplificando pelo mapa k.	40
Figura 30 - Circuito de controle para bombas d'água B2 simplificando pelo mapa k.	40

Figura 31 - Simulação do circuito de controle de bombas d'água simplificado, pelo Multisim.	41
Figura 32 - Circuito de alarme de automóvel simulado no Multisim.....	43
Figura 33 - Circuito de alarme de automóvel simplificando pelo mapa k.	43
Figura 34 - Simulação do circuito de alarme de automóvel simplificado, pelo Multisim.	44
Figura 35 - Leitura do atraso de propagação do circuito simplificado de controle de iluminação feita no osciloscópio.	48

LISTA DE TABELAS

Tabela 1 - Identidade do postulado da complementação.....	5
Tabela 2 - Identidades do postulado da adição.....	6
Tabela 3 - Identidades do postulado da multiplicação.	7
Tabela 4 - Tabela verdade: propriedade distributiva.	8
Tabela 5 - Tabela-verdade: primeiro teorema de De Morgan	9
Tabela 6 - Tabela-verdade: segundo teorema de De Morgan.....	10
Tabela 7 - Nomenclaturas dos níveis lógicos.	20
Tabela 8 - Tabela-verdade do controle do ar condicionado.	31
Tabela 9 - Tabela-verdade do circuito de controle de iluminação.	34
Tabela 10 - Tabela-verdade do circuito de controle para bombas d'água.....	38
Tabela 11 - Tabela-verdade do circuito de alarme de automóvel.....	42
Tabela 12 - Comparações dos circuitos nas suas formas não simplificadas e simplificadas. ...	46
Tabela 13 - Tempo de atraso de propagação dos circuitos não simplificados e simplificados.	48

LISTA DE ABREVIATURAS E SIGLAS

IEEE	<i>Institute of Electrical and Electronics Engineers</i>
TTL	<i>Transistor-Transistor Logic</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
V_{IL}	<i>low-level input voltage</i> : valor de tensão máxima
V_{OL}	<i>low-level output voltage</i> : valor de tensão máxima
V_{IH}	<i>high-level input voltage</i> : valor de tensão mínima
V_{OH}	<i>high-level output voltage</i> : valor de tensão mínima
I_{IL}	<i>low-level input current</i> : valor de corrente máxima
I_{OL}	<i>low-level output current</i> : valor de corrente máxima
I_{IH}	<i>high-level input current</i> : valor de corrente de entrada máxima
I_{OH}	<i>high-level output current</i> : valor de corrente de saída máxima
CI	Circuito Integrado
IoT	Internet das Coisas

SUMÁRIO

1. INTRODUÇÃO.....	1
2. ÁLGEBRA BOOLEANA.....	4
2.1 Postulados	5
2.1.1 Postulado da Complementação	5
2.1.2 Postulado da Adição	6
2.1.3 Postulado da Multiplicação	7
2.2 Propriedades	7
2.2.1 Propriedade Comutativa.....	8
2.2.2 Propriedade Associativa.....	8
2.2.3 Propriedade Distributiva	8
2.3 Teoremas de De Morgan	8
2.3.1 Primeiro Teorema de De Morgan	9
2.3.2 Segundo Teorema de De Morgan	9
2.4 Identidades Auxiliares	10
2.5 Mapa de Karnaugh.....	11
2.5.1 Agrupamento de Quadros	14
2.5.1.1 Agrupamento de Dois Quadros (pares ou dupla).....	14
2.5.1.2 Agrupamento de quatro quadros (quartetos).....	15
2.5.1.3 Agrupamento de oito quadros (octetos).....	17
2.6 A Importância da Álgebra Booleana na Formação em Engenharia Elétrica	18
3. CIRCUITOS LÓGICOS DIGITAIS.....	20
3.1 Função e Porta Lógica OU ou OR	21
3.2 Função e Porta Lógica E ou AND	21
3.3 Função e Porta Lógica NÃO ou NOT	22
3.4 Portas NOR e Portas NAND.....	22
3.4.1 Porta NOR (NÃO-OU)	22
3.4.2 Porta NAND (NÃO-E)	23
3.5 Famílias de Circuitos Lógicos e Tecnologia de Circuitos Integrados (CIs)	24
3.6 Encapsulamento de CIs.....	26
3.7 Especificações Técnicas (<i>data sheets</i>).....	26
3.8 Implementando Circuitos a partir de Expressões Booleanas.....	27
3.9 Simplificação de circuitos lógicos	28

3.10 Software de Simulação <i>Multisim</i>	29
4. PESQUISA EXPERIMENTAL	30
5. RESULTADOS E DISCUSSÕES	45
5.1 Redução do número de portas lógicas	45
5.2 Os atrasos de propagação	47
5.3 A eficiência geral dos circuitos	49
6. CONSIDERAÇÕES FINAIS	51
7. REFERÊNCIAS	52

1. INTRODUÇÃO

Em 1854, o matemático britânico George Boole introduziu um sistema inovador de análise lógica em sua obra *An Investigation of the Laws of Thought*, que ficou conhecido como Álgebra de Boole. Esse sistema marcou o início da lógica moderna, essa álgebra utiliza um conjunto finito de valores binários, tradicionalmente representados por 0 e 1, para expressar operações lógicas fundamentais como OR (OU), AND (E) e NOT (NÃO). Hoje, essas operações estão no cerne do funcionamento dos circuitos digitais, responsáveis por processar e transmitir informações em praticamente todos os dispositivos eletrônicos modernos. A aplicação da álgebra booleana tornou-se crucial no desenvolvimento de circuitos combinacionais, sistemas de controle e até mesmo na otimização de código e estruturas lógicas de softwares.

A simplificação de funções booleanas, um processo essencial dentro da engenharia elétrica e da ciência da computação, visa reduzir a complexidade das expressões que descrevem os circuitos, mantendo sua funcionalidade. Essa redução não só diminui o número de portas lógicas necessárias, como também contribui para aumentar a eficiência energética, melhorar o desempenho e reduzir custos de implementação. Métodos como o mapa de Karnaugh e os teoremas de De Morgan são amplamente usados para realizar essas simplificações, sendo ferramentas essenciais tanto em contextos acadêmicos quanto na indústria.

Dada a crescente demanda por dispositivos eletrônicos mais rápidos e compactos, a otimização de circuitos digitais tornou-se uma prioridade no campo da engenharia elétrica e da tecnologia da informação. A necessidade de minimizar o consumo de energia e maximizar o desempenho sem comprometer a funcionalidade tem impulsionado a busca por técnicas de simplificação eficientes. Neste sentido, o uso de softwares de simulação, como o *Multisim*, é uma ferramenta poderosa para validar teorias e observar, na prática, os benefícios de técnicas de minimização de expressões booleanas.

Em um cenário onde a tecnologia avança rapidamente e a demanda por dispositivos mais rápidos e energeticamente eficientes cresce continuamente, a álgebra de Boole torna-se indispensável para o entendimento de estruturas de controle de fluxo, lógica de decisão e otimização de código no desenvolvimento de circuitos. Em sistemas embarcados, onde os

recursos são limitados e a eficiência é crítica, a aplicação correta de princípios booleanos é primordial para garantir a funcionalidade e o desempenho do sistema.

Este trabalho tem como objetivo demonstrar as vantagens da simplificação de circuitos combinacionais utilizando a minimização de expressões booleanas por meio de simulações práticas realizadas no *Multisim*. Serão analisadas questões como a redução do número de portas lógicas, o impacto no consumo de energia, a diminuição de custos de implementação e o aumento do desempenho dos circuitos.

Além disso, será discutida a relevância de uma formação sólida em álgebra booleana para engenheiros, especialmente no contexto de desenvolvimento de tecnologias modernas. O referencial teórico abordará as bases matemáticas e lógicas por trás dessas simplificações, e a pesquisa experimental validará os resultados teóricos, utilizando circuitos simulados para comparar a eficiência dos métodos aplicados.

Inicialmente, é realizada uma revisão bibliográfica que fornecerá o embasamento teórico necessário para fundamentar a análise e os experimentos. Esta fase tem como principais objetivos:

- I. Explorar os fundamentos teóricos dos circuitos combinacionais: Identificar os principais tipos de circuitos, suas aplicações e como são projetados a partir de expressões booleanas.
- II. Apresentar métodos de simplificação de expressões booleanas: Concentrar nos teoremas e métodos usados para simplificar expressões booleanas, como o Mapa de Karnaugh e os teoremas da álgebra de Boole, que são amplamente utilizados para reduzir circuitos lógicos.
- III. Estudar as vantagens teóricas da simplificação: Revisar a literatura que discute os benefícios da simplificação de circuitos, como redução do consumo de energia, economia de componentes, maior confiabilidade e menor atraso de propagação.
- IV. Analisar estudos e pesquisas anteriores: Revisar artigos científicos, dissertações e teses relacionados à implementação e simplificação de circuitos combinacionais e os impactos resultantes em termos de desempenho e custos.

Posteriormente, uma Pesquisa Experimental é realizada. Nesta fase, os objetivos serão simular diferentes circuitos combinacionais no *Multisim*, tanto na forma original quanto simplificada. As simplificações serão feitas utilizando os métodos de minimização

mencionados. Os dados obtidos a partir das simulações serão compilados e organizados em tabelas comparativas. Serão analisados os seguintes aspectos: Avaliar se a simplificação foi eficaz na redução do número de portas lógicas e conexões, comparar o tempo de propagação do sinal nos circuitos simplificados e originais, verificando se houve melhora no tempo de resposta, considerar como a redução no número de conexões e componentes pode melhorar a confiabilidade do circuito, reduzindo possíveis falhas, estimar se a simplificação resulta em menor consumo de energia, com base na redução de operações lógicas e componentes.

Por fim, os dados coletados na pesquisa fornecerão uma análise detalhada das vantagens práticas da simplificação de circuitos combinacionais pela minimização de expressões booleanas, tanto do ponto de vista teórico quanto prático, utilizando simulações como ferramenta para validar os resultados. A expectativa é que as conclusões possam contribuir para o entendimento dos benefícios da simplificação em diferentes cenários, desde projetos acadêmicos até implementações industriais. Além de reforçar a necessidade de uma educação consistente em álgebra booleana para preparar engenheiros capazes de atender às demandas tecnológicas modernas com eficiência e inovações.

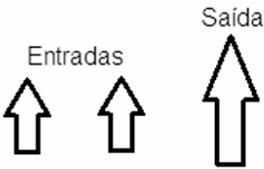
2. ÁLGEBRA BOOLEANA

Em 1854, o matemático britânico George Boole introduziu um sistema matemático para a análise lógica em sua obra intitulada *An Investigation of the Laws of Thought*. Esse sistema ficou conhecido como Álgebra de Boole e marcou o início da lógica moderna, consolidando Boole como uma figura fundamental nesse campo. (COSTA; SILVA; GARCIA, 2022, p.1)

Ao contrário da álgebra convencional dos números reais, onde as variáveis podem assumir valores em todo o intervalo $(-\infty; +\infty)$, as variáveis Booleanas têm um conjunto limitado de possíveis valores. Especificamente, na álgebra Booleana binária, cada variável pode assumir um entre dois valores distintos, que podem ser representados como [F, V] (falso ou verdadeiro), [H, L] (alto ou baixo), ou ainda [0,1]. Como o número de valores possíveis para cada variável é limitado (e pequeno), o número de estados possíveis para uma função Booleana também será finito. Isso nos permite descrever completamente as funções Booleanas por meio de tabelas. Por essa razão, uma tabela que detalha uma função Booleana é chamada de tabela verdade, e nela são listadas todas as combinações de valores que as variáveis de entrada podem assumir, juntamente com os correspondentes valores da função (saídas). (GÜNTZEL; NASCIMENTO, 2001, p.1)

A tabela-verdade é uma ferramenta essencial na álgebra Booleana. Na Figura 1, temos um exemplo de um conjunto lógico com duas entradas (A e B), representadas nas duas primeiras colunas. A última coluna, X, mostra a saída lógica resultante, que varia de acordo com a operação lógica aplicada e os valores finitos das entradas.

Figura 1 - Tabela verdade com duas entradas



A	B	X
0 ou F	0 ou F	?
0 ou F	1 ou V	?
1 ou V	0 ou F	?
1 ou V	1 ou V	?

Expressões matemáticas formadas por variáveis booleanas são classificadas de expressões booleanas. A álgebra booleana é baseada em três postulados fundamentais: o postulado da complementação, o postulado da adição e o postulado da multiplicação. Esses postulados correspondem, respectivamente, às três operações lógicas básicas: NOT (NÃO), OR (OU) e AND (E).

2.1 Postulados

Se faz necessário um estudo dos postulados da complementação, da adição e da multiplicação da álgebra de Boole e as respectivas identidades resultantes.

2.1.1 Postulado da Complementação

Este postulado expõe as regras de complementação na álgebra de Boole. Chamaremos de $\bar{A}$ o complemento de A :

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

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

A Tabela 1 mostra que por meio do postulado da complementação, podemos estabelecer a seguinte identidade:

Tabela 1 - Identidade do postulado da complementação.

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

Fonte: O autor, 2025.

Esta barra sobre a letra que representa a variável significa a inversão desta e se lê, para a variável A, “A barrado”, e no caso das duas barras, se lê, “A barrado/barrado”. Esse símbolo denota a função da porta lógica NOT (NÃO).

2.1.2 Postulado da Adição

Este postulado define as regras para a adição na álgebra de Boole. O sinal de adição, é lido como (OU). A porta lógica que executa o postulado da adição é a OR. Então:

$$1^{\circ}) \mathbf{0 + 0 = 0}$$

$$2^{\circ}) \mathbf{0 + 1 = 1}$$

$$3^{\circ}) \mathbf{1 + 0 = 1}$$

$$4^{\circ}) \mathbf{1 + 1 = 1}$$

Com base no postulado da adição, é possível estabelecer algumas identidades algébricas adicionais, dispostas na Tabela 2.

Tabela 2 - Identidades do postulado da adição.

Identidade	Possibilidades	Observação
$\mathbf{A + 0 = A}$	$\mathbf{A = 0 \rightarrow 0 + 0 = 0}$ $\mathbf{A = 1 \rightarrow 1 + 0 = 1}$	Observamos que o resultado será sempre igual à variável A.
$\mathbf{A + 1 = 1}$	$\mathbf{A = 0 \rightarrow 0 + 1 = 1}$ $\mathbf{A = 1 \rightarrow 1 + 1 = 1}$	Percebemos que, ao somar 1 a uma variável, o resultado será sempre 1.
$\mathbf{A + A = A}$	$\mathbf{A = 0 \rightarrow 0 + 0 = 0}$ $\mathbf{A = 1 \rightarrow 1 + 1 = 1}$	Notamos que, se somarmos a mesma variável, o resultado será ela mesma.
$\mathbf{A + \bar{A} = 1}$	$\mathbf{A = 0 \rightarrow \bar{A} = 1 \rightarrow 0 + 1 = 1}$ $\mathbf{A = 1 \rightarrow \bar{A} = 0 \rightarrow 1 + 0 = 1}$	Observamos que, sempre que somarmos uma variável ao seu complemento, o resultado será 1.

Fonte: O autor, 2025.

2.1.3 Postulado da Multiplicação

Este postulado determina as regras da multiplicação booleana, o sinal de multiplicação é lido como E. A porta lógica que executa o postulado da multiplicação é o AND (E). Então:

$$1^{\circ}) 0 \cdot 0 = 0$$

$$2^{\circ}) 0 \cdot 1 = 0$$

$$3^{\circ}) 1 \cdot 0 = 0$$

$$4^{\circ}) 1 \cdot 1 = 1$$

E, com base no postulado da multiplicação, podem-se estabelecer outras identidades, que são as seguintes: ver Tabela 3.

Tabela 3 - Identidades do postulado da multiplicação.

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.
$A \cdot 1 = A$	$A = 0 \rightarrow 0 \cdot 1 = 0$ $A = 1 \rightarrow 1 \cdot 1 = 1$	Conclui-se que o resultado dessas expressões numéricas é sempre igual a A.
$A \cdot A = A$	$A = 0 \rightarrow 0 \cdot 0 = 0$ $A = 1 \rightarrow 1 \cdot 1 = 1$	Percebemos que os resultados serão sempre iguais a A.
$A \cdot \bar{A} = 0$	$A = 0 \rightarrow \bar{A} = 1 \rightarrow 0 \cdot 1 = 0$ $A = 1 \rightarrow \bar{A} = 0 \rightarrow 1 \cdot 0 = 0$	Notamos que o resultado dessas expressões numéricas é sempre igual a 0.

Fonte: O autor, 2025.

2.2 Propriedades

A seguir, serão apresentadas as principais propriedades algébricas, fundamentais para o manuseio e simplificação de expressões Booleanas. Essas propriedades, que serão aplicadas na álgebra de Boole, são as mesmas encontradas na matemática tradicional, e incluem:

2.2.1 Propriedade Comutativa

Esta propriedade é válida tanto na adição como na multiplicação:

Adição: $A + B = B + A$

Multiplicação: $A \cdot B = B \cdot A$

2.2.2 Propriedade Associativa

Assim como na anterior, temos a propriedade associativa válida na adição e na multiplicação:

Adição: $A + (B + C) = (A + B) + C = A + B + C$

Multiplicação: $A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C$

2.2.3 Propriedade Distributiva

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

A propriedade distributiva pode ser facilmente confirmada usando uma tabela-verdade, desde que sejam consideradas todas as possibilidades, conforme ilustrado na Tabela 4 a seguir:

Tabela 4 - Tabela verdade: propriedade distributiva.

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

Fonte: O autor, 2025.

2.3 Teoremas de De Morgan

Os teoremas têm o nome de Augustus De Morgan, um matemático e lógico britânico do século XIX, que formalizou e popularizou essas leis no contexto da lógica formal. Na

prática, os teoremas de De Morgan são amplamente utilizados na simplificação de expressões booleanas e no desenvolvimento de circuitos digitais.

2.3.1 Primeiro Teorema de De Morgan

O complemento do produto é igual à soma dos complementos:

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

A prova deste teorema é representada na tabela-verdade a seguir. Ver Tabela 5.

Tabela 5 - Tabela-verdade: primeiro teorema de De Morgan

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

Fonte: O autor, 2025.

Nota-se a igualdade dos valores lógicos nas duas últimas colunas para todas as possibilidades de A e B. Portanto, o primeiro teorema de De Morgan é válido. Esse teorema pode ser estendido para mais de duas variáveis:

$$\overline{(A_1 \cdot A_2 \cdot \dots \cdot A_n)} = \bar{A}_1 + \bar{A}_2 + \dots + \bar{A}_n$$

2.3.2 Segundo Teorema de De Morgan

O complemento da soma é igual ao produto dos complementos.

Este teorema é uma ampliação do primeiro:

$$\overline{(A + B)} = \bar{A} \cdot \bar{B} \leftarrow 1^\circ \text{ teorema}$$

$$\bar{A} \cdot \bar{B} = \overline{(A + B)} \leftarrow 2^\circ \text{ teorema}$$

A prova desta igualdade é comprovada na seguinte tabela-verdade. Ver Tabela 6.

Tabela 6 - Tabela-verdade: segundo teorema de De Morgan.

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

Fonte: O autor, 2025.

Nota-se, novamente, a igualdade de ambas as colunas para todas as possibilidades. Assim como o anterior, o teorema pode ser estendido para mais de duas variáveis:

$$\overline{(A_1 + A_2 + \dots + A_n)} = \bar{A}_1 \cdot \bar{A}_2 \cdot \dots \cdot \bar{A}_n$$

2.4 Identidades Auxiliares

Há três outras identidades úteis para simplificação de expressões booleanas.

$$A + A \cdot B = A \quad (1).$$

Para provar essa identidade (1) utiliza-se a propriedade distributiva. E evidencia A no 1º termo.

$$A(1 + B) = A$$

Do postulado da soma $1 + B = 1$; então, podemos escrever:

$$A \cdot 1 = A \therefore A + A \cdot B = A$$

$$(A + B) \cdot (A + C) = A + B \cdot C \quad (2).$$

Para provar essa identidade (2), temos:

$$(A + B) \cdot (A + C)$$

$$= A \cdot A + A \cdot C + B \cdot A + B \cdot C \rightarrow \text{Propriedade distributiva}$$

$$= A + A \cdot C + A \cdot B + B \cdot C \rightarrow \text{Identidade } A \cdot A = A$$

$$\begin{aligned}
&= \mathbf{A} \cdot (\mathbf{1} + \mathbf{B} + \mathbf{C}) + \mathbf{B} \cdot \mathbf{C} \rightarrow \text{Propriedade distributiva} \\
&= \mathbf{A} \cdot \mathbf{1} + \mathbf{B} \cdot \mathbf{C} \rightarrow \text{Identities } \mathbf{1} + \mathbf{X} = \mathbf{1} \text{ e } \mathbf{A} \cdot \mathbf{1} = \mathbf{A} \\
&= \mathbf{A} + \mathbf{B} \cdot \mathbf{C}
\end{aligned}$$

Então, conclui-se que:

$$(\mathbf{A} + \mathbf{B}) \cdot (\mathbf{A} + \mathbf{C}) = \mathbf{A} + \mathbf{B} \cdot \mathbf{C}$$

$$\mathbf{A} + \bar{\mathbf{A}} \cdot \mathbf{B} = \mathbf{A} + \mathbf{B} \quad (3).$$

Para provar essa identidade (3), temos:

$$\begin{aligned}
\mathbf{A} + \bar{\mathbf{A}} \cdot \mathbf{B} &= \overline{(\overline{\mathbf{A} + \bar{\mathbf{A}} \cdot \mathbf{B}})} \rightarrow \text{Identidade } \bar{\bar{\mathbf{X}}} = \mathbf{X} \\
&= \overline{\bar{\mathbf{A}} \cdot (\bar{\mathbf{A}} \cdot \mathbf{B})} \rightarrow 2^\circ \text{ teorema de De Morgan: } \bar{\mathbf{A}} \cdot \bar{\mathbf{B}} = \overline{(\mathbf{A} + \mathbf{B})} \\
&= \overline{\bar{\mathbf{A}} \cdot (\mathbf{A} + \bar{\mathbf{B}})} \rightarrow 1^\circ \text{ teorema de De Morgan aplicados nos parênteses: } \overline{(\mathbf{A} \cdot \mathbf{B})} = \bar{\mathbf{A}} + \bar{\mathbf{B}} \\
&= \overline{(\bar{\mathbf{A}} \cdot \mathbf{A} + \bar{\mathbf{A}} \cdot \bar{\mathbf{B}})} \rightarrow \text{Propriedade distributiva e identidade } \bar{\mathbf{A}} \cdot \mathbf{A} = \mathbf{0} \\
&= \overline{(\bar{\mathbf{A}} \cdot \bar{\mathbf{B}})} \rightarrow 1^\circ \text{ teorema de De Morgan: } \overline{(\mathbf{A} \cdot \mathbf{B})} = \bar{\mathbf{A}} + \bar{\mathbf{B}} \\
&= \mathbf{A} + \mathbf{B}
\end{aligned}$$

Então, concluímos que:

$$\mathbf{A} + \bar{\mathbf{A}} \cdot \mathbf{B} = \mathbf{A} + \mathbf{B}$$

2.5 Mapa de Karnaugh

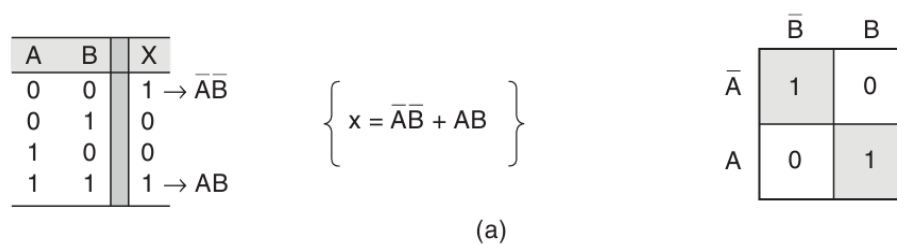
Maurice Karnaugh nasceu em Nova Iorque em 1924. Ele estudou matemática, ciência da computação e engenharia de telecomunicações, obtendo seu doutorado nesta última área. Entre 1952 e 1966, trabalhou na Bell Labs, uma empresa pioneira no desenvolvimento do transistor de silício. Durante esse período, Karnaugh criou o método conhecido como mapa de Karnaugh (mapa K). Desde 1975, ele era membro do *Institute of Electrical and Electronics Engineers* (IEEE) em reconhecimento por suas contribuições no uso de métodos numéricos em telecomunicações.

O mapa de Karnaugh ou também como é conhecido de mapa k é um método gráfico utilizado para simplificar expressões booleanas ou converter uma tabela-verdade em um circuito lógico correspondente de forma simples e sistemática. Embora o mapa de Karnaugh possa ser aplicado a problemas com qualquer número de variáveis de entrada, sua utilidade prática é mais evidente para até cinco ou seis variáveis. Assim como a tabela verdade, o mapa K é preenchido com 0 e 1.

O mapa K, igualmente a tabela-verdade, é utilizado para demonstrar a relação entre as entradas lógicas e a saída desejada. As Figuras 2, 3 e 4 apresenta três exemplos de mapas K, com duas, três e quatro variáveis, juntamente com as tabelas-verdade correspondentes. Esses exemplos ilustram os seguintes pontos importantes:

1. A tabela-verdade indica o valor da saída X para cada combinações de entrada, enquanto o mapa K organiza essas informações em um formato visual diferente. Cada linha da tabela-verdade corresponde a um quadrado no mapa K. Por exemplo, na Figura 2(a), a condição $A = 0, B = 0$ na tabela-verdade equivale ao quadrado $\bar{A}\bar{B}$ no mapa K. Como a tabela-verdade mostra $X = 1$ para esse caso, o 1 é inserido no quadrado $\bar{A}\bar{B}$ no mapa K. Da mesma forma, a condição $A = 1, B = 1$ na tabela-verdade corresponde ao quadrado AB no mapa K, e como $X = 1$, nesse caso, o 1 também é colocado nesse quadrado. Todos os demais quadrados são preenchidos com 0s.

Figura 2 - (a) Mapas de Karnaugh e tabelas-verdade para duas variáveis



Fonte: TOCCI; WIDMER; MOSS, 2011.

2. Após preencher um mapa K com 0s e 1s, a expressão na forma de soma-de-produtos para a saída X pode ser determinada realizando a operação OR entre os quadrados que contêm 1. No mapa de três variáveis da Figura 3(b), os quadrados $\bar{A}\bar{B}\bar{C}$, $\bar{A}\bar{B}C$, $\bar{A}B\bar{C}$, e $AB\bar{C}$ contém 1, resultando na expressão $X = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C}$.

Figura 3 - (b) Mapas de Karnaugh e tabelas-verdade para três variáveis.

A	B	C	X
0	0	0	1 → $\bar{A}\bar{B}\bar{C}$
0	0	1	1 → $\bar{A}\bar{B}C$
0	1	0	1 → $\bar{A}B\bar{C}$
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1 → $AB\bar{C}$
1	1	1	0

$$\left\{ X = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C} \right\}$$

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

(b)

Fonte: TOCCI; WIDMER; MOSS, 2011.

- Os quadrados do mapa K são organizados para que os adjacentes na horizontal e na vertical se diferenciam por apenas uma variável. No mapa de quatro variáveis ilustrado na Figura 4(c), por exemplo, o quadrado $\bar{A}\bar{B}\bar{C}\bar{D}$ é adjacente ao quadrado à direita $\bar{A}\bar{B}\bar{C}D$ (diferem apenas na variável D) e ao quadrado abaixo $\bar{A}\bar{B}C\bar{D}$ (diferem na variável B). Além disso, os quadrados na linha superior são considerados adjacentes aos correspondentes na linha inferior, e o mesmo vale para as colunas mais à esquerda e à direita.
- Para garantir que os quadrados adjacentes, tanto na vertical quanto na horizontal, diferenciam-se por apenas uma variável, as denominações devem seguir a ordem específica mostrada na Figura 4(c): de cima para baixo, as variáveis são organizadas como $\bar{A}\bar{B}$, $\bar{A}B$, AB , e $A\bar{B}$. O mesmo princípio se aplica na direção horizontal, da esquerda para a direita, com as variáveis organizadas como $\bar{C}\bar{D}$, $\bar{C}D$, CD , e $C\bar{D}$.

Figura 4 - (c) Mapas de Karnaugh e tabelas-verdade para quatro variáveis.

A	B	C	D	X
0	0	0	0	0
0	0	0	1	1 → $\bar{A}\bar{B}\bar{C}D$
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1 → $\bar{A}B\bar{C}D$
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1 → $AB\bar{C}D$
1	1	1	0	0
1	1	1	1	1 → $ABCD$

$$\left\{ \begin{array}{l} X = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D \\ + AB\bar{C}D + ABCD \end{array} \right\}$$

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

(c)

Fonte: TOCCI; WIDMER; MOSS, 2011.

2.5.1 Agrupamento de Quadros

A expressão para a saída X pode ser simplificada unindo corretamente os blocos do mapa de Karnaugh que possuem 1. Esse processo de união dos 1s é chamado de agrupamento.

2.5.1.1 Agrupamento de Dois Quadros (pares ou dupla)

A Figura 5 mostra o mapa de Karnaugh correspondente a uma tabela-verdade de três variáveis. Nesse mapa, há um par de 1s verticalmente adjacentes; o primeiro representa $\bar{A}\bar{B}\bar{C}$ e o segundo, $AB\bar{C}$. Note que, nesses dois termos, a variável A aparece tanto na forma normal quanto na forma complementada (invertida), enquanto B e C permanecem inalterados. Esses dois termos podem ser agrupados, o que resulta na eliminação da variável A, já que ela aparece nos dois termos de formas complementadas e não complementadas. Isso pode ser facilmente comprovado, como será mostrado a seguir:

Figura 5 - Exemplo de agrupamento de pares de 1s adjacentes.

	$\bar{C}$	C	
$\bar{A}\bar{B}$	0	0	
$\bar{A}B$	1	0	$X = \bar{A}B\bar{C} + AB\bar{C}$ $= B\bar{C}$
AB	1	0	
$A\bar{B}$	0	0	

Fonte: TOCCI; WIDMER; MOSS, 2011.

O mesmo princípio é válido para qualquer par de 1s adjacentes vertical ou horizontalmente. Ao agrupar um par de 1s adjacentes em um mapa de Karnaugh, a variável que aparece tanto na forma complementada quanto na não complementada é eliminada.

Outro exemplo é apresentado na Figura 6. Nesse mapa de K, as linhas superior e inferior dos blocos são consideradas adjacentes. Dessa forma, os dois 1s presentes no mapa podem ser agrupados, resultando em $\bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} = \bar{B}\bar{C}$.

Figura 6 - Exemplo de agrupamento de pares de 1s adjacentes.

	$\bar{C}$	C	
$\bar{A}\bar{B}$	1	0	$X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} = \bar{B}\bar{C}$
$\bar{A}B$	0	0	
AB	0	0	
$A\bar{B}$	1	0	

Fonte: TOCCI; WIDMER; MOSS, 2011.

2.5.1.2 Agrupamento de quatro quadros (quartetos)

Um mapa de Karnaugh pode conter um grupo de quatro 1s que são adjacentes entre si, chamado de quarteto. A Figura 7 apresenta um exemplo de quartetos. Os quatro 1s estão

adjacentes verticalmente. Quando um quarteto é agrupado, o termo resultante conterá apenas as variáveis que não alteram a forma considerando todos os quadros 1s do quarteto.

Figura 7 - Exemplo de agrupamento de quatro de 1s adjacentes.

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

$X = C$

Fonte: TOCCI; WIDMER; MOSS, 2011.

No mapa K da Figura 8, os quadros são considerados adjacentes entre si. Da mesma forma, assim como os 1s na Figura 6, pois, como mencionado anteriormente, as linhas superior e inferior são adjacentes, assim como as colunas mais à esquerda e mais à direita.

Figura 8 - Exemplo de agrupamento de quatro de 1s adjacentes.

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

$X = \bar{B}\bar{D}$

Fonte: TOCCI; WIDMER; MOSS, 2011.

2.5.1.3 Agrupamento de oito quadros (octetos)

Um grupo de oito 1s adjacentes entre si é chamado de octeto. Quando um octeto é agrupado em um mapa de quatro variáveis, três são eliminadas, porque apenas uma variável permanece inalterada. Por exemplo, ao analisar o agrupamento dos oito quadros 1s na Figura 9, observa-se que apenas a variável B mantém-se inalterada em todos os oito quadros, enquanto as outras variáveis aparecem tanto na forma complementada quanto na não complementada. Portanto, para este mapa, $X = B$.

Figura 9 - Exemplos de agrupamentos de oito 1s (octetos).

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

$X = B$

Fonte: TOCCI; WIDMER; MOSS, 2011.

Podemos verificar outros tipos de agrupamentos octetos de 1s adjacentes entre si nas Figuras 10 e 11.

Figura 10 - Exemplos de agrupamentos de oito 1s (octetos).

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

$X = \bar{B}$

Fonte: TOCCI; WIDMER; MOSS, 2011.

Figura 11 - Exemplos de agrupamentos de oito 1s (octetos).

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

$X = \bar{D}$

Fonte: TOCCI; WIDMER; MOSS, 2011.

Importante ressaltar que é possível que haja mais de um modelo de agrupamento em um mapa de Karnaugh. E 1s que não sejam adjacentes com nenhum outro quadrado, estes são denominados de solos ou isolados. Enfatizando outro ponto significativo, é que, para aumentar um determinado agrupamento pode-se utilizar de 1s que já estão inclusos em outros agrupamentos. A Figura 12 ilustra um mapa K das mencionadas possibilidades.

Figura 12 - Exemplos de vários agrupamentos.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0 ₁	0 ₂	0 ₃	1 ₄
$\bar{A}B$	0 ₅	1 ₆	1 ₇	0 ₈
AB	0 ₉	1 ₁₀	1 ₁₁	0 ₁₂
$A\bar{B}$	0 ₁₃	0 ₁₄	1 ₁₅	0 ₁₆

$$X = \underbrace{\bar{A}\bar{B}\bar{C}\bar{D}}_{\text{grupo 4}} + \underbrace{ACD}_{\text{grupo 6, 7, 10, 11}} + \underbrace{BD}_{\text{grupo 6, 7, 10, 11}}$$

Fonte: TOCCI; WIDMER; MOSS, 2011.

2.6 A Importância da Álgebra Booleana na Formação em Engenharia Elétrica

Após a apresentação das definições, propriedades e postulados que fundamentam a Álgebra Booleana, é essencial destacar a importância de uma formação sólida nesse campo. Mais do que um conjunto de ferramentas matemáticas, a Álgebra Booleana constitui a base teórica indispensável para o desenvolvimento de sistemas digitais, que estão no coração de diversas áreas da engenharia elétrica.

De acordo com Vieira (2000), a lógica binária de Boole se adapta perfeitamente à descrição de circuitos elétricos em computadores: os circuitos estão ligados ou desligados, assim como uma proposição pode ser verdadeira ou falsa. Um interruptor opera de maneira semelhante a uma porta lógica, controlando o fluxo de corrente para o próximo estágio. Atualmente, todos os computadores utilizam a álgebra de Boole implementada em microchips, os quais contêm milhares de interruptores organizados em portas lógicas, permitindo a realização de operações e a produção de resultados por meio da linguagem binária.

A falta de uma compreensão aprofundada da Álgebra Booleana pode ser um obstáculo significativo para a concepção e análise de circuitos digitais. Conceitos como simplificação de expressões lógicas, otimização de circuitos e compreensão de tabelas-verdade são habilidades fundamentais que permitem o desenvolvimento de soluções mais eficientes, tanto do ponto de vista do desempenho quanto do custo.

Além disso, uma sólida formação em Álgebra Booleana é crucial para áreas específicas como:

- **Projeto de Sistemas Digitais:** A construção de circuitos combinacionais e sequenciais exige o domínio de conceitos booleanos para garantir funcionalidade e eficiência.
- **Automação e Controle:** Sistemas de automação frequentemente dependem de lógica digital para a tomada de decisões e controle de processos industriais.
- **Processamento de Sinais e Comunicação:** A lógica digital fundamenta operações como codificação, decodificação e compressão de dados.

Em um contexto onde a eletrônica digital está presente em praticamente todos os dispositivos modernos, desde smartphones até sistemas industriais complexos, a formação em Álgebra Booleana não é apenas um diferencial, mas uma necessidade.

Portanto, compreender os conceitos apresentados ao longo deste capítulo não apenas amplia a base teórica, mas também habilita o estudante a enfrentar desafios práticos que exigem criatividade, precisão e eficiência no desenvolvimento de soluções tecnológicas.

3. CIRCUITOS LÓGICOS DIGITAIS

Um dos temas fundamentais da eletrônica digital é o estudo dos circuitos combinacionais. Através desse estudo, podemos entender o funcionamento de circuitos como somadores, subtratores, circuitos de prioridade, codificadores, decodificadores, entre outros, que são amplamente utilizados na construção de sistemas digitais.

Um circuito lógico combinacional é um tipo de circuito digital cuja saída depende exclusivamente dos valores atuais das entradas. Ao contrário dos circuitos sequenciais, que têm memória e cujas saídas dependem tanto das entradas atuais quanto de estados anteriores, os circuitos combinacionais não armazenam informações. Isso significa que cada combinação específica de entradas gera um único e definido conjunto de saídas.

Podemos utilizar um circuito lógico combinacional para solucionar problemas em que necessitamos de uma resposta, quando acontecerem determinadas situações, representadas pelas variáveis de entrada. Para construir esses circuitos, necessitamos de suas expressões características, que são obtidas das tabelas-verdade. (IDOETA; CAPUANO, 2019, p.146).

Portanto, os circuitos digitais realizam operações algébricas utilizando portas lógicas, que são formadas por diodos, transistores e resistores. Essas portas lógicas implementam operações booleanas como "OU", "E" e "NÃO", de modo que a saída do circuito reflita o resultado da operação correspondente. Assim, as portas lógicas seguem os mesmos princípios e definições das operações algébricas já mencionadas.

É importante esclarecer o que são os níveis lógicos, são os estados binários que representam as informações em circuitos digitais. Geralmente, esses níveis são indicados como "0" (baixo ou falso) e "1" (alto ou verdadeiro). Esses estados são fundamentais para o funcionamento de sistemas digitais, pois são utilizados para transmitir, processar e armazenar dados. Na lógica digital também são utilizados outros termos alternativos para designar esses níveis lógicos, como mostrado na Tabela 7 a seguir.

Tabela 7 - Nomenclaturas dos níveis lógicos.

Nível Lógico 0	Nível Lógico 1
Baixo	Alto
Falso	Verdadeiro
Não	Sim

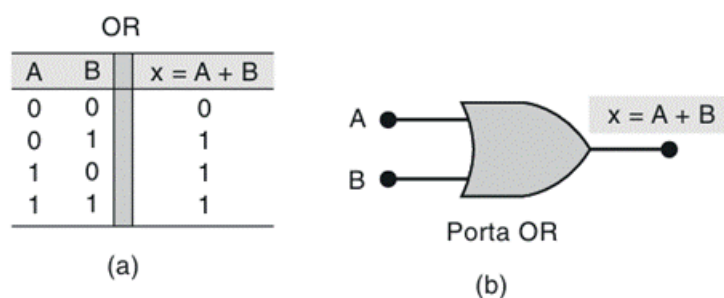
Aberto	Fechado
Desligado	Ligado

Fonte: TOCCI; WIDMER; MOSS, 2011.

3.1 Função e Porta Lógica OU ou OR

A operação OR retorna verdadeiro (1) se pelo menos uma das entradas for verdadeira (1). O resultado será falso (0) apenas se todas as entradas forem falsas (0). É usada em situações onde a ocorrência de qualquer uma das condições é suficiente para satisfazer a expressão. A Figura 13(a) mostra a tabela-verdade para uma porta OR de duas entradas, e a Figura 13(b) mostra o símbolo lógico.

Figura 13 - (a) Tabela-verdade que define a operação OR; (b) símbolo de uma porta OR de duas entradas.

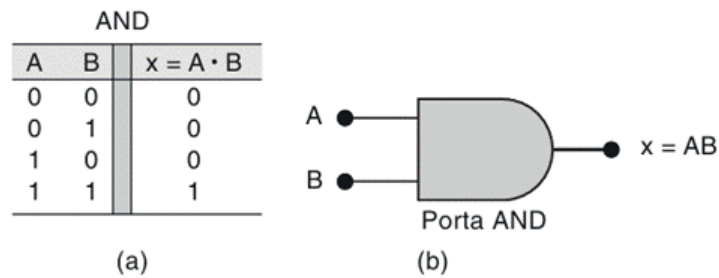


Fonte: TOCCI; WIDMER; MOSS, 2011.

3.2 Função e Porta Lógica E ou AND

A operação AND retorna verdadeiro (1) apenas se todas as entradas forem verdadeiras (1). Se qualquer uma das entradas for falsa (0), o resultado será falso (0). Essa operação é frequentemente usada em circuitos e expressões onde uma condição deve ser atendida em conjunto com outra. A tabela-verdade na Figura 14(a) mostra o que acontece quando duas entradas lógicas, A e B, são combinadas usando uma operação AND para gerar a saída X, e a Figura 14(b) mostra o símbolo da porta AND.

Figura 14 - (a) Tabela-verdade para a operação AND; (b) símbolo da porta AND

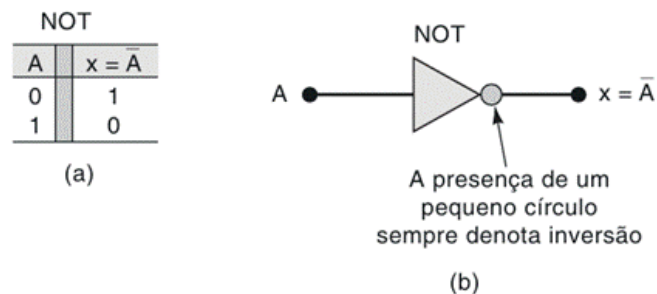


Fonte: TOCCI; WIDMER; MOSS, 2011.

3.3 Função e Porta Lógica NÃO ou NOT

A operação NOT é uma operação unitária, o que significa que ela opera sobre apenas uma variável. Ela inverte o valor lógico da entrada: se a entrada for 1, o resultado será 0, e vice-versa. A operação NOT também é conhecida como inversão ou complemento. A Figura 15(a) demonstra a tabela-verdade da porta inversora e a Figura 15(b) mostra o símbolo para o circuito NOT.

Figura 15 - (a) Tabela-verdade; (b) símbolo para o INVERSOR (circuito NOT).



Fonte: TOCCI; WIDMER; MOSS, 2011.

3.4 Portas NOR e Portas NAND

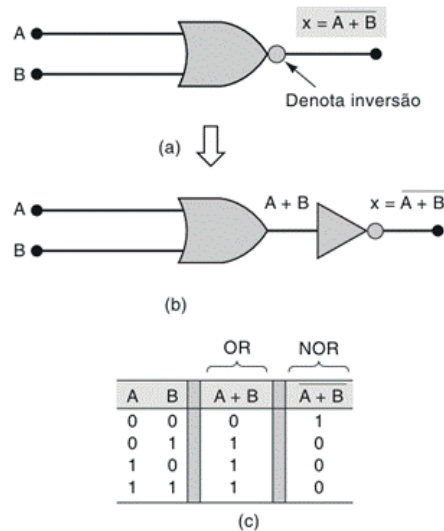
Existem outros dois tipos de portas lógicas, NOR e NAND, utilizadas amplamente em circuitos digitais. Essas portas operam basicamente combinando funções básicas, AND, OR e NOT.

3.4.1 Porta NOR (NÃO-OU)

Igualmente ao da porta OR, o que difere é apenas o círculo na saída que representa a operação de inversão. Portanto, a operação da porta NOR é igual a porta OR seguida de um inversor. Os circuitos na Figura 16(a) e (b) são equivalentes, e a expressão de saída para a

porta NOR é $X = \overline{A + B}$. A tabela-verdade da Figura 16(c) mostra que a saída da porta NOR é exatamente o inverso da saída da porta OR para todas as condições possíveis de entrada.

Figura 16 - (a) Símbolo da porta NOR; (b) circuito equivalente; (c) tabela-verdade.

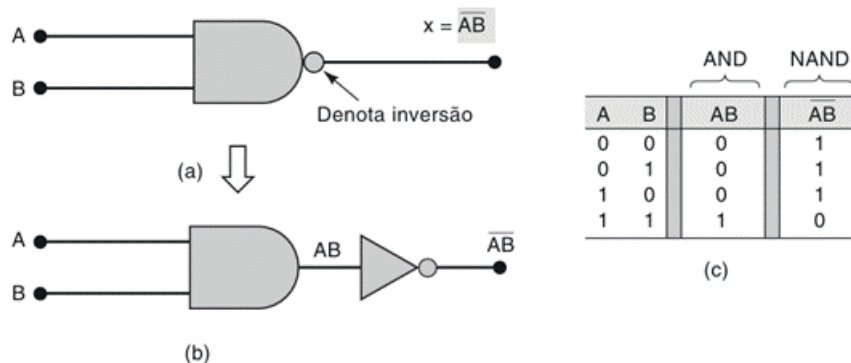


Fonte: TOCCI; WIDMER; MOSS, 2011.

3.4.2 Porta NAND (NÃO-E)

É o mesmo que o da porta AND, exceto pelo pequeno círculo na saída que indica a operação de inversão. Então, a operação da porta NAND é semelhante à da porta AND seguida de um inversor. Os circuitos na Figura 17(a) e (b) são equivalentes, e a expressão de saída para a porta NAND é $X = \overline{A \cdot B}$. A tabela-verdade da Figura 17(c) mostra que a saída da porta NAND é igual ao inverso da porta AND para todas as condições possíveis de entrada.

Figura 17 - (a) Símbolo da porta NAND; (b) circuito equivalente; (c) tabela-verdade.



Fonte: TOCCI; WIDMER; MOSS, 2011.

3.5 Famílias de Circuitos Lógicos e Tecnologia de Circuitos Integrados (CIs)

Até o momento, foi utilizado os blocos lógicos sem nos preocuparmos com suas estruturas internas. Famílias de circuitos lógicos referem-se aos diferentes tipos de estruturas internas empregadas na criação de blocos em circuitos integrados. Cada uma dessas famílias faz uso de componentes específicos, o que determina suas características de funcionamento e desempenho prático.

Na eletrônica digital atual, as famílias ainda em uso são a TTL (*Transistor-Transistor Logic*) e a CMOS (*Complementary Metal Oxide Semiconductor*). Estas famílias, embora hoje restritas a projetos menores, originam-se de várias outras famílias de circuitos lógicos, muitas das quais já estão obsoletas.

Os circuitos TTL são fabricados em duas séries comerciais: a 74XXX e a 54XXX. A série 54XXX, também chamada de série militar ou profissional, possui maior tolerância nas especificações de alimentação e temperatura, garantindo desempenho confiável mesmo em condições extremas. Já a família CMOS inclui circuitos integrados nas séries 4000A, 4000B e 54/74C, sendo esta última semelhante à TTL em relação à pinagem e às funções dos blocos disponíveis.

Definimos na Tabela 7 os níveis lógicos 1 e 0. Na realidade esses níveis são faixas de tensão. O nível 0 não corresponde exatamente a 0 volts, mas sim a uma tensão pequena dentro de um limite máximo. Da mesma forma, o nível 1 representa uma faixa de tensão acima de um valor mínimo e abaixo de um valor máximo, em vez de um valor fixo. Dependendo da tecnologia usada na construção do circuito, cada família ou versão derivada terá faixas de operação específicas para esses níveis, com especificações distintas para entrada e saída dos blocos.

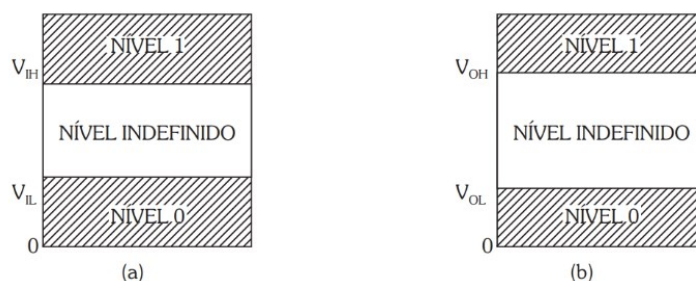
Outro parâmetro importante é a corrente. Quando aplicamos um nível lógico 1 à entrada de um bloco, ele consome uma corrente. O mesmo ocorre ao conectar a saída de um bloco em nível 1 à entrada de outro, ocasionando uma drenagem de corrente, embora limitada. De modo semelhante, se aplicarmos um nível 0 (potencial de terra) à entrada de um bloco, haverá uma corrente originada a partir do circuito do bloco em direção ao terminal. A saída em nível 0 também absorverá corrente proveniente da entrada do bloco seguinte ao qual está conectada.

Os principais fabricantes de circuitos integrados utilizam uma terminologia padrão em seus manuais para definir esses parâmetros. Elas são:

- V_{IL} (*low-level input voltage*): valor de tensão (máxima), que garante o nível 0 na entrada.
- V_{OL} (*low-level output voltage*): valor de tensão (máxima), que garante o nível 0 na saída.
- V_{IH} (*high-level input voltage*): valor de tensão (mínima), que garante o nível 1 na entrada.
- V_{OH} (*high-level output voltage*): valor de tensão (mínima), que garante o nível 1 na saída.
- I_{IL} (*low-level input current*): valor de corrente (máxima), no terminal de entrada (no sentido do bloco para o terminal), quando é aplicado o nível 0.
- I_{OL} (*low-level output current*): valor de corrente (máxima), que a saída pode receber quando em nível 0.
- I_{IH} (*high-level input current*): valor de corrente de entrada (máxima), quando é aplicado nível 1.
- I_{OH} (*high-level output current*): valor de corrente de saída (máxima), quando em nível 1.

A Figura 18 apresenta os diagramas relativos aos níveis de tensão definidos, tanto para a entrada (a) como para a saída (b), de um mesmo bloco lógico.

Figura 18 - Diagramas de níveis de tensão.



Fonte: IDOETA; CAPUANO, 2019.

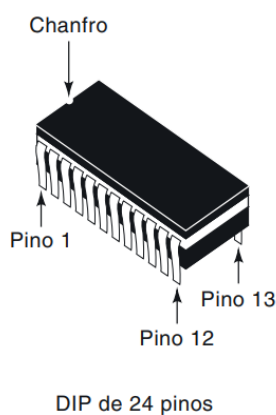
Observa-se que, na faixa entre o valor máximo do nível 0 (V_{IL} e V_{OL}) e o valor mínimo do nível 1 (V_{IH} e V_{OH}), o nível lógico fica indefinido. Para garantir uma

compatibilidade segura entre entrada e saída dentro da mesma família, isso garante que o CI não venha entender errado a informação a ser transmitida, é essencial que V_{OL} seja inferior a V_{IL} e que V_{OH} seja superior a V_{IH} .

3.6 Encapsulamento de CIs

Há uma variedade de tipos de encapsulamentos que se diferenciam pelo tamanho físico, pelas condições ambientais e de consumo de energia sob as quais o dispositivo opera de forma confiável, além da forma de montagem do CI na placa de circuito impresso. A Figura 19 apresenta uma representatividade de encapsulamento de CIs.

Figura 19 - Encapsulamento comum de CIs.



Fonte: TOCCI; WIDMER; MOSS, 2011.

3.7 Especificações Técnicas (*data sheets*)

O *data sheet* (ou folha de dados) é um documento técnico fornecido pelos fabricantes de componentes eletrônicos e circuitos integrados. Ele contém todas as informações essenciais sobre um componente, incluindo suas características elétricas, especificações de desempenho, parâmetros de operação, e detalhes de encapsulamento. A Figura 20 ilustra um exemplo de *data sheet*.

Figura 20 - Data sheet da porta OR do CI 7432.

Electrical Characteristics						
over recommended operating free air temperature range (unless otherwise noted)						
Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}, I_I = -18 \text{ mA}$			-1.5	V
V_{OH}	HIGH Level Output Voltage	$V_{CC} = \text{Min}, I_{OH} = \text{Max}$ $V_{IH} = \text{Min}$	2.7	3.4		V
V_{OL}	LOW Level Output Voltage	$V_{CC} = \text{Min}, I_{OL} = \text{Max}$ $V_{IL} = \text{Max}$		0.35	0.5	V
		$I_{OL} = 4 \text{ mA}, V_{CC} = \text{Min}$		0.25	0.4	
I_I	Input Current @ Max Input Voltage	$V_{CC} = \text{Max}, V_I = 7 \text{ V}$			0.1	mA
I_{IH}	HIGH Level Input Current	$V_{CC} = \text{Max}, V_I = 2.7 \text{ V}$			20	μA
I_{IL}	LOW Level Input Current	$V_{CC} = \text{Max}, V_I = 0.4 \text{ V}$			-0.36	mA
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}$ (Note 3)	-20		-100	mA
I_{CCH}	Supply Current with Outputs HIGH	$V_{CC} = \text{Max}$		3.1	6.2	mA
I_{CCL}	Supply Current with Outputs LOW	$V_{CC} = \text{Max}$		4.9	9.8	mA

Note 2: All typicals are at $V_{CC} = 5 \text{ V}, T_A = 25^\circ\text{C}$.

Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Fonte: FAIRCHILD, 2000.

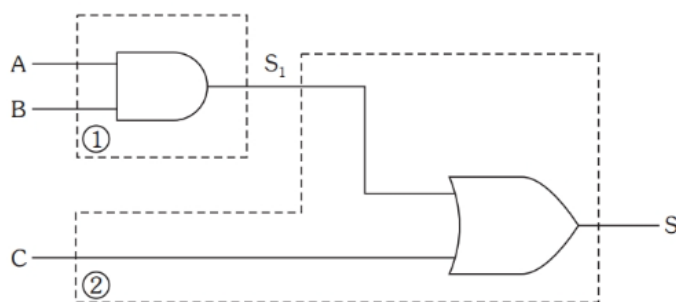
3.8 Implementando Circuitos a partir de Expressões Booleanas

Todo circuito lógico executa uma expressão booleana e, por mais complexo que seja, é construído pela interligação de portas lógicas básicas. É possível determinar a expressão booleana correspondente a qualquer circuito lógico, ou a partir da expressão booleana projetar este circuito combinacional. Para demonstrar o procedimento, vamos obter o circuito lógico da expressão booleana $S = A \cdot B + C$, ver Figura 21.

Quando a operação de um circuito é definida por uma expressão booleana, podemos desenhar o diagrama do circuito lógico diretamente a partir da expressão. Por exemplo, se precisarmos de um circuito definido por $x = A \cdot B \cdot C$, saberemos imediatamente que precisamos de uma porta AND de três entradas. Se precisarmos de um circuito definido por $x = A + B$ poderemos usar uma porta OR de duas entradas. O mesmo raciocínio pode ser estendido para circuitos mais complexos. (TOCCI; WIDMER; MOSS, 2011, p.62).

Para facilitar o entendimento, será dividido o circuito em duas partes:

Figura 21 - Como obter o circuito a partir da expressão booleana.



Fonte: IDOETA; CAPUANO, 2019.

Na saída S_1 , temos $A \cdot B$, sendo esse bloco uma porta AND, sua expressão de saída será $S_1 = A \cdot B$. Como S_1 é em uma das entradas da porta OR pertencente à segunda parte do circuito e na outra entrada está a variável C , a expressão de saída será $S = S_1 + C$

Para comprovar a expressão dada, basta substituímos S_1 na expressão anterior, obtendo, então:

$$S = A \cdot B + C$$

que é a expressão que o circuito executa.

3.9 Simplificação de circuitos lógicos

Após obtermos a expressão de um circuito lógico, podemos simplificá-la de forma a reduzir o número de termos ou variáveis presentes em um ou mais de seus termos. A expressão simplificada resultante pode então ser utilizada para implementar um circuito equivalente ao original, porém com um menor número de portas lógicas e conexões.

Dispondo de dois circuitos que possuem a mesma lógica, o circuito simplificado é preferível, pois utiliza menos portas lógicas, sendo menor e mais econômico que o original. Além disso, a confiabilidade é aumentada devido ao menor número de conexões, que são potenciais pontos de falha nos circuitos. Outra vantagem estratégica da simplificação de circuitos lógicos está relacionada à velocidade de operação. Se circuitos lógicos forem configurados de forma que as mudanças nas entradas precisem passar por muitas portas para gerar a saída, eles serão mais lentos em comparação com circuitos que têm menos camadas de portas.

Para efetuar essas simplificações, existem, basicamente, dois processos já mencionados anteriormente. O primeiro deles é a simplificação pelos teoremas da álgebra de

Boole. O segundo, a utilização dos mapas de Karnaugh. Nesse trabalho será usado somente o mapa de Karnaugh.

3.10 Software de Simulação *Multisim*

O *Multisim* é um software de simulação de circuitos eletrônicos, desenvolvido pela *National Instruments*, amplamente utilizado em engenharia elétrica, eletrônica e em áreas relacionadas ao design de sistemas digitais e analógicos.

O *Multisim* é uma ferramenta essencial para engenheiros e estudantes de eletrônica, pois acelera o processo de design e desenvolvimento, reduzindo custos e melhorando a precisão dos projetos. Ao permitir a simulação detalhada de circuitos, o software ajuda a evitar erros de design e a garantir que os circuitos funcionem conforme o esperado antes de serem implementados fisicamente.

4. PESQUISA EXPERIMENTAL

Nesta etapa do trabalho, o objetivo será simular diferentes circuitos combinacionais no *Multisim*, tanto na forma original quanto simplificada. As simplificações serão feitas utilizando os métodos de minimização mencionados. Os dados obtidos a partir das simulações serão compilados e organizados em tabelas comparativas. Serão analisados os seguintes aspectos: Avaliar se a simplificação foi eficaz na redução do número de portas lógicas e conexões, comparar o tempo de propagação do sinal nos circuitos simplificados e originais, verificando se houve melhora no tempo de resposta, considerar como a redução no número de conexões e componentes pode melhorar a confiabilidade do circuito, reduzindo possíveis falhas, estimar se a simplificação resulta em menor consumo de energia, com base na redução de operações lógicas e componentes.

A automação de sistemas é uma área essencial da engenharia que busca melhorar a eficiência, segurança e comodidade em diversas aplicações do dia a dia. Circuitos eletrônicos combinacionais são amplamente utilizados para implementar sistemas que tomam decisões lógicas com base em condições pré-estabelecidas. Neste contexto, o trabalho destaca quatro sistemas de automação que ilustram diferentes aplicações práticas: um circuito de controle de ar condicionado, um sistema automatizado de iluminação, um controle de bombas d'água para tanques e um circuito de alarme automotivo. Cada um deles utiliza sensores para monitorar variáveis específicas e tomar decisões inteligentes, garantindo segurança operacional e conforto aos usuários. Esses exemplos mostram como a tecnologia pode ser integrada ao cotidiano para solucionar problemas de forma inovadora e confiável.

Controle do ar condicionado

Nos dias atuais, a busca por conforto, eficiência energética e praticidade tem impulsionado o desenvolvimento de tecnologias automatizadas para o gerenciamento de ambientes residenciais e comerciais. Dentre essas inovações, destaca-se o sistema de automação para acionamento de ar condicionado, projetado para proporcionar um clima de forma inteligente e adaptativa.

Esse tipo de sistema utiliza sensores para analisar variáveis do ambiente e tomar decisões em tempo real, garantindo que o uso do ar condicionado seja eficiente, alinhado às necessidades do momento e econômico em termos de consumo de energia. Além de aumentar

o conforto dos usuários, esses sistemas refletem a integração da tecnologia ao cotidiano, oferecendo soluções que unem conveniência e sustentabilidade.

Descrição do Sistema:

O sistema de automatizar o acionamento de um ar condicionado utiliza como variáveis:

- **Horário: H** - O ar condicionado liga automaticamente em horários específicos, como à noite ou durante o dia. Verdadeiro (1) se o horário for um dos específicos para ligar o ar condicionado. Falso (0) se o horário não for adequado.
- **Nível de iluminação: L** - O ar condicionado liga quando a iluminação no quarto é baixa, como durante a noite ou quando as luzes estão apagadas. Verdadeiro (1) se a iluminação estiver baixa (indicando noite ou luzes apagadas). Falso (0) se a iluminação estiver alta.
- **Deteção de movimento: M** - O ar condicionado liga quando o movimento é detectado dentro do quarto, como quando alguém entra ou se movimenta. Verdadeiro (1) se houver movimento no quarto. Falso (0) se não houver movimentação.
- **Ajuste de conforto(temperatura): C** - O ar condicionado liga quando a sensação térmica, ajustada por sensores de temperatura e umidade, indica que o ambiente não está confortável. Verdadeiro (1) se a sensação térmica indicar que o ambiente não está confortável. Falso (0) se temperatura está confortável.
- **Ar condicionado: A** - Verdadeiro (1) ar ligado. Falso (0) ar desligado.

Independentemente das outras condições, se não houver detecção de movimento, o ar condicionado não deve ligar. A Tabela 8 ilustra a tabela-verdade do circuito de forma não simplificada e a Figura 22 apresenta o circuito simulado no software *Multisim*.

Tabela 8 - Tabela-verdade do controle do ar condicionado.

M	H	C	L	A
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	0
0	1	1	0	0
0	1	1	1	0

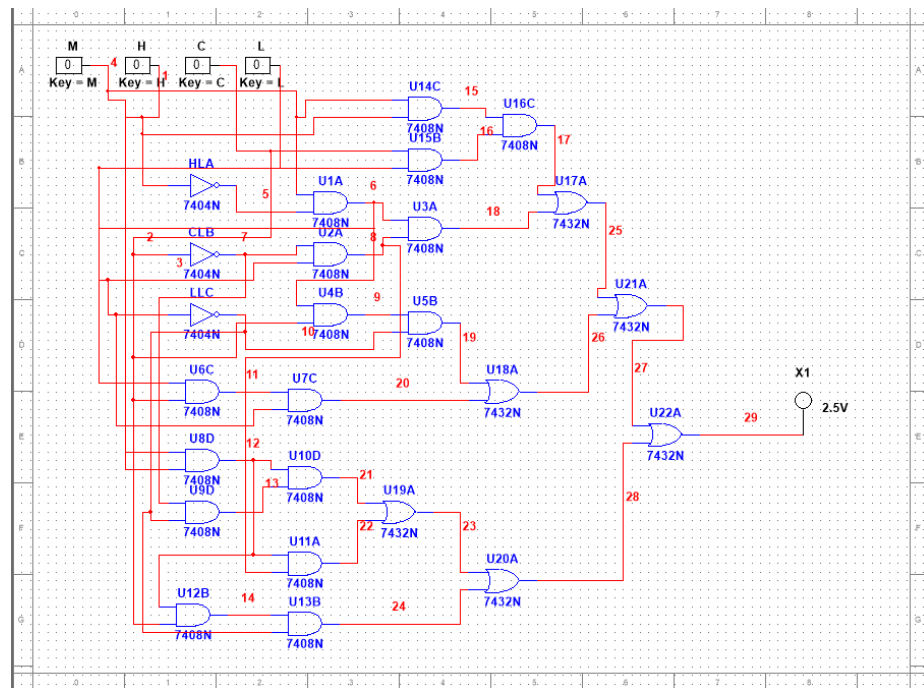
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Fonte: O autor, 2025.

Assim, temos a seguinte expressão booleana:

$$A = M\bar{H}\bar{C}\bar{L} + M\bar{H}\bar{C}L + M\bar{H}CL + MH\bar{C}\bar{L} + MH\bar{C}L + MHCL + MHCL$$

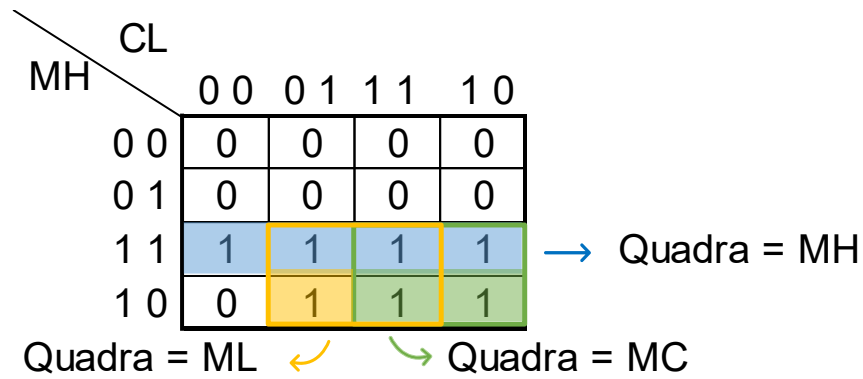
Figura 22 - Circuito de controle do ar condicionado simulado no *Multisim*.



Fonte: O autor, *Multisim*, 2025.

A seguir, apresenta-se a simplificação do circuito, destacando as etapas realizadas para otimizar sua estrutura e funcionalidade. A Figura 23 exemplifica o processo de minimização pelo método de mapa de Karnaugh, a Figura 24 demonstra o novo circuito simulado no *Multisim*.

Figura 23 - Circuito de controle do ar condicionado simplificado pelo mapa K.

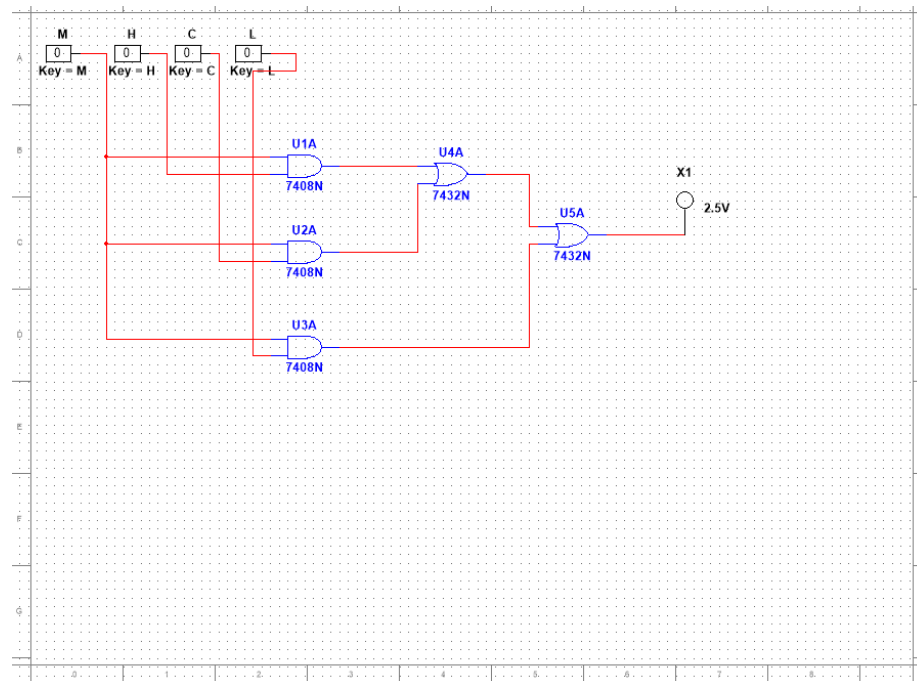


Fonte: O autor, 2025.

Dada a simplificação, temos a nova expressão booleana:

$$A = MH + MC + ML$$

Figura 24 - Simulação do circuito controle do ar condicionado simplificado, pelo *Multisim*.



Fonte: O autor, *Multisim*, 2025.

Controle de Iluminação

Os sistemas inteligentes de automação estão transformando o gerenciamento de recursos nos ambientes domésticos e corporativos. O sistema de iluminação automatizado exemplifica essa evolução, oferecendo uma solução que combina praticidade, eficiência energética. Esse tipo de tecnologia é especialmente útil em residências, escritórios e espaços públicos, onde o controle eficiente da iluminação contribui para a redução do consumo de energia e melhora a experiência dos usuários.

Ao integrar sensores de luminosidade, sensores de presença e a opção de controle manual, o sistema proporciona flexibilidade, ajustando-se automaticamente às condições do ambiente ou às preferências do usuário. Dessa forma, promove um uso inteligente dos recursos e o bem-estar.

Descrição do Sistema:

A iluminação será ativada se o ambiente estiver escuro e houver presença ou se for acionada manualmente.

- **Sensor de Luminosidade (L):** Detecta a intensidade da luz ambiente. 1 (Verdadeiro) ambiente está escuro, acender luzes. 0 (Falso) ambiente iluminado, apagar luzes.
- **Sensor de Presença (P):** Detecta movimento no ambiente. 1 (Verdadeiro) presença detectada, acender luzes. 0 (Falso) – Sem presença, apagar luzes.
- **Interruptor Manual (M):** Permite controle manual das luzes. 1 (Verdadeiro) acender luzes manualmente.
- **Iluminação (I):** 1 (Verdadeiro) luzes acessas. 0 (Falso) luzes apagadas.

A Tabela 9 mostra a tabela-verdade do circuito de forma não simplificada e a Figura 25 apresenta o circuito simulado no software *Multisim*.

Tabela 9 - Tabela-verdade do circuito de controle de iluminação.

M	P	L	I
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1

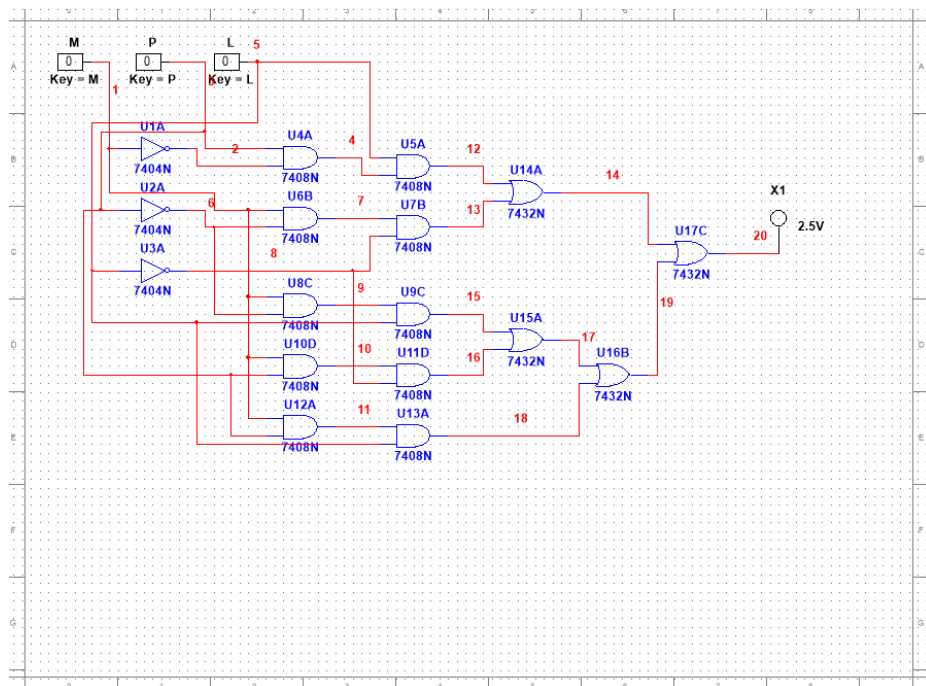
1	0	1	1
1	1	0	1
1	1	1	1

Fonte: O autor, 2025.

A partir da tabela obtivemos a expressão booleana correspondente:

$$I = \overline{M}PL + M\overline{P}L + M\overline{P}\overline{L} + M\overline{P}\overline{L} + MPL$$

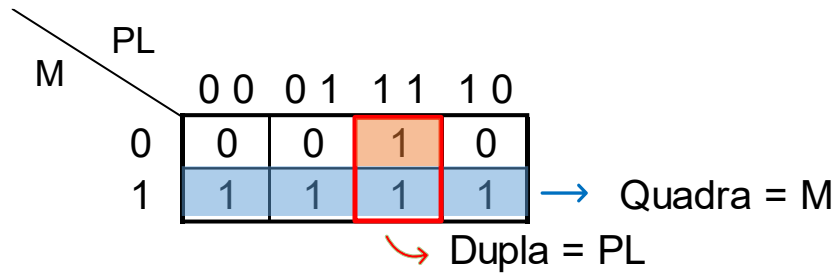
Figura 25 - Circuito de controle de iluminação simulado no *Multisim*.



Fonte: O autor, *Multisim*, 2025.

A seguir, é apresentada a simplificação do circuito combinacional, evidenciando as etapas utilizadas para otimizar sua configuração e desempenho. A Figura 26 elucida o processo de simplificação pelo método de mapa de Karnaugh, a Figura 27 mostra o novo circuito simulado no *Multisim*.

Figura 26 - Circuito de controle de iluminação simplificando pelo mapa k.

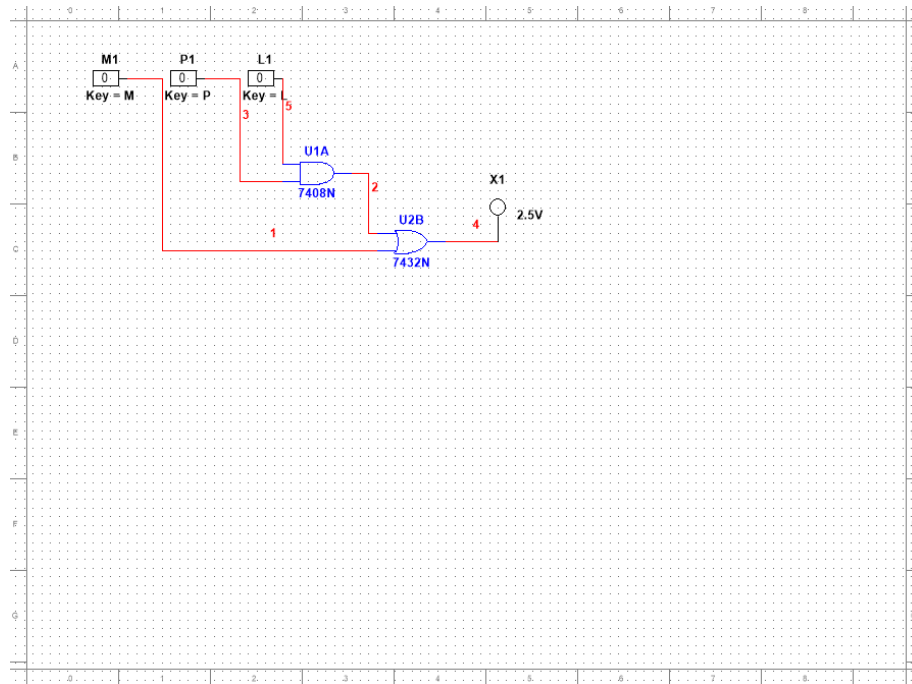


Fonte: O autor, 2025.

Após simplificar temos a seguinte expressão booleana:

$$I = M + PL$$

Figura 27 - Simulação do circuito de controle de iluminação simplificado, pelo *Multisim*.



Fonte: O autor, *Multisim*, 2025.

Controle para bombas d'água [13]

A gestão eficiente de recursos hídricos é uma necessidade em diversas aplicações, desde sistemas agrícolas até o abastecimento doméstico ou industrial. Um controle automatizado de bombas d'água é uma solução tecnológica que otimiza o uso da água, garantindo o abastecimento adequado dos reservatórios e prevenindo desperdícios ou falhas operacionais. Esse tipo de sistema é especialmente importante em situações onde o fluxo de água entre diferentes tanques precisa ser controlado com precisão para atender à demanda e evitar danos ao equipamento.

O sistema implementa um controle lógico baseado em sensores de nível, que monitoram constantemente os níveis de água nos tanques inferior e superior. Ele assegura que as bombas sejam acionadas ou desativadas de acordo com critérios bem definidos, promovendo um funcionamento seguro e eficiente. Além disso, o sistema possui uma camada adicional de segurança que desliga as bombas imediatamente se uma condição de operação impossível for detectada, prevenindo danos e riscos ao equipamento ou ao ambiente.

Descrição do Sistema:

A bomba d'água B1 leva água de um riacho até o tanque inferior, e a bomba B2 leva água do tanque inferior para o superior. A bomba B1 deve ligar com o objetivo de manter a água sempre próxima do nível máximo (S2), desligando ao atingir S2. A bomba B2 funciona da mesma forma, baseada nos níveis S3 e S4, mas não poderá funcionar caso o nível do tanque inferior esteja abaixo de S1. Se qualquer combinação que os sensores enviarem for impossível de ocorrer na prática, as duas bombas devem ser imediatamente desligadas, independentemente de qualquer outra situação.

Sensores do tanque inferior:

- **S₁**= Sensor de nível baixo do tanque inferior.
 - S₁ = 1 se há água suficiente no nível mínimo do tanque inferior.
 - S₁ = 0 se tanque inferior está abaixo do nível baixo (não deve bombear).
- **S₂** = Sensor de nível alto do tanque inferior.
 - S₂ =1 se a água do tanque inferior está acima do nível alto (nível adequado para bombear).
 - S₂ =0 se a água tanque inferior está abaixo do nível alto.

Sensores do tanque superior:

- **S₃** = Sensor de nível baixo do tanque superior.
 - S₃ = 0 se a água na caixa está abaixo do nível baixo (precisa ligar a bomba).
 - S₃ = 1 se a água está acima do nível baixo.
- **S₄** = Sensor de nível alto do tanque superior.
 - S₄ = 0 se a água na caixa está abaixo do nível alto.
 - S₄ = 1 se a água na caixa atingiu o nível alto (desliga a bomba).

Bomba 1, bomba 2:

- **B₁** = Estado da bomba.
 - B₁ = 1 se a bomba deve estar ligada.
 - B₁ = 0 se a bomba deve estar desligada.
- **B₂** = Estado da bomba.
 - B₂ = 1 se a bomba deve estar ligada.
 - B₂ = 0 se a bomba deve estar desligada.

A Tabela 10 mostra a tabela-verdade do circuito de forma não simplificada e a Figura 28 apresenta o circuito simulado no software *Multisim*.

Tabela 10 - Tabela-verdade do circuito de controle para bombas d'água.

S ₁	S ₂	S ₃	S ₄	B ₁	B ₂
0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	1	0	1	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	0	0
1	1	1	0	0	1
1	1	1	1	0	0

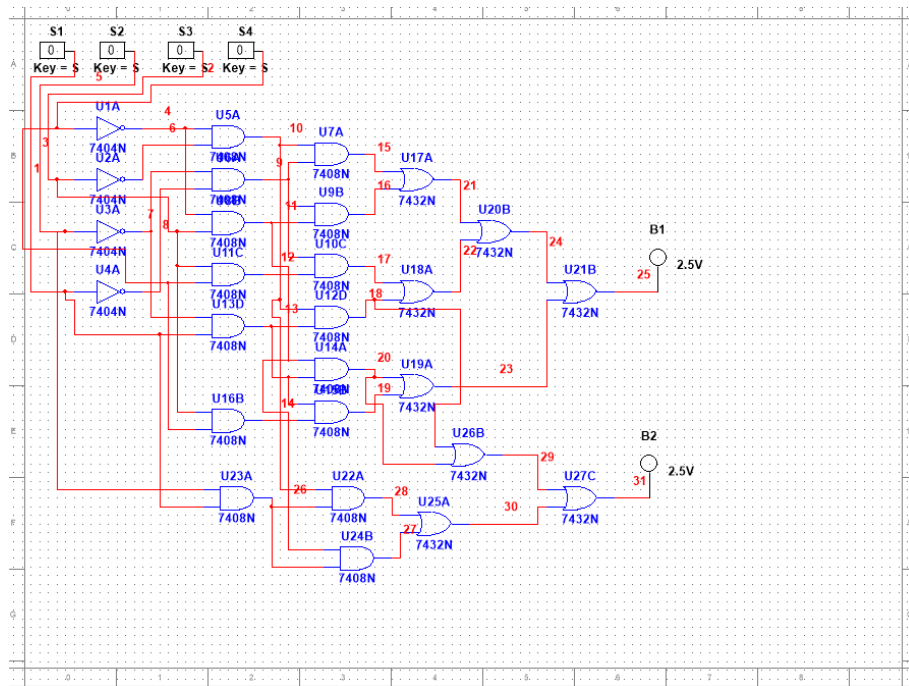
Fonte: O autor, 2025.

As expressões booleanas correspondentes para B₁ e B₂ desse circuito de acordo com a tabela-verdade são:

$$B_1 = \bar{S}_1 \bar{S}_2 \bar{S}_3 \bar{S}_4 + \bar{S}_1 \bar{S}_2 S_3 \bar{S}_4 + \bar{S}_1 \bar{S}_2 S_3 S_4 + S_1 \bar{S}_2 \bar{S}_3 \bar{S}_4 + S_1 \bar{S}_2 S_3 \bar{S}_4 + S_1 \bar{S}_2 S_3 S_4$$

$$B_2 = S_1 \bar{S}_2 \bar{S}_3 \bar{S}_4 + S_1 \bar{S}_2 S_3 \bar{S}_4 + S_1 S_2 \bar{S}_3 \bar{S}_4 + S_1 S_2 S_3 \bar{S}_4$$

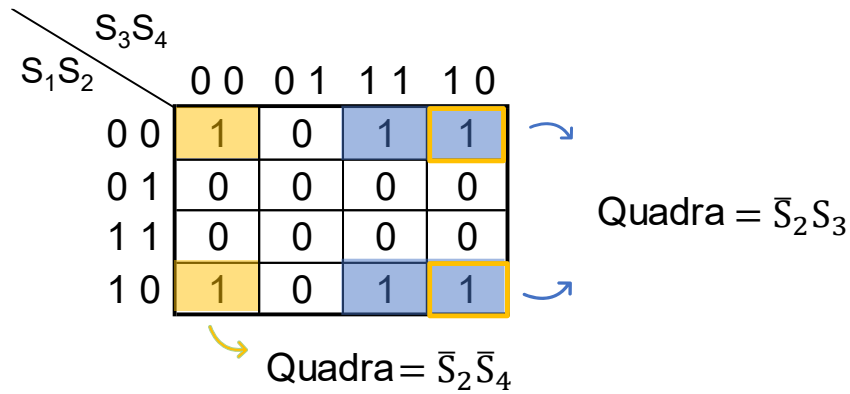
Figura 28 - Circuito de controle para bombas d'água simulado no *Multisim*.



Fonte: O autor, *Multisim*, 2025.

Segue abaixo a simplificação do circuito combinacional, demonstrando o processo aplicado para aprimorar sua estrutura e eficiência. A Figura 29 e 30 ilustra o processo de minimização utilizando o método do mapa de Karnaugh para B₁ e B₂ respectivamente, enquanto a Figura 31 apresenta o circuito otimizado simulado no *Multisim*.

Figura 29 - Circuito de controle para bombas d'água B1 simplificando pelo mapa k.

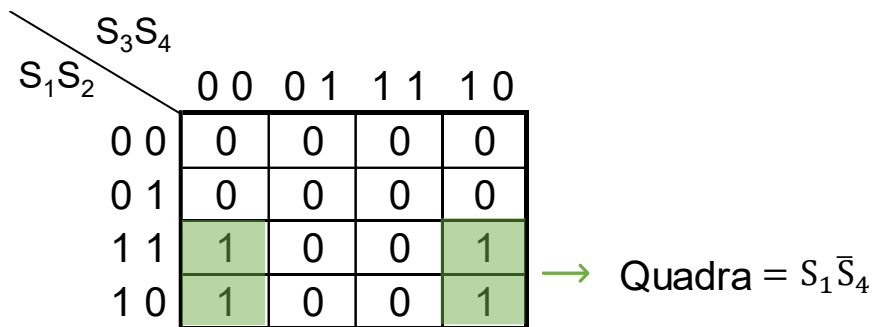


Fonte: O autor, 2025.

Com a simplificação, obtém-se a nova expressão booleana para B_1 :

$$B_1 = \bar{S}_2S_3 + \bar{S}_2\bar{S}_4$$

Figura 30 - Circuito de controle para bombas d'água B2 simplificando pelo mapa k.

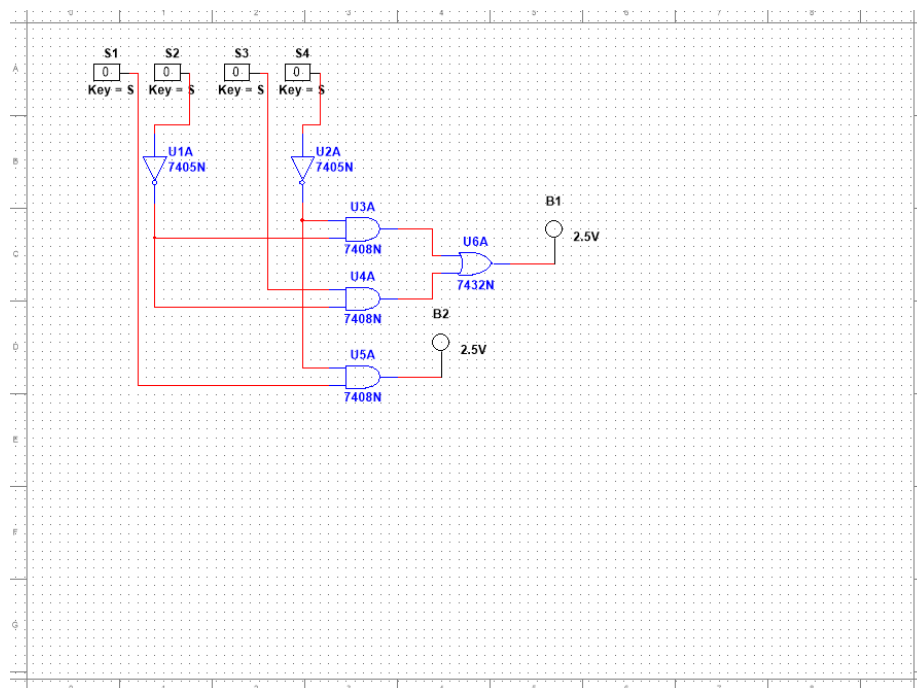


Fonte: O autor, 2025.

Com a seguinte simplificação, obtém-se a nova expressão booleana para B_2 :

$$B_2 = S_1\bar{S}_4$$

Figura 31 - Simulação do circuito de controle de bombas d'água simplificado, pelo *Multisim*.



Fonte: O autor, *Multisim*, 2025.

Circuito de alarme de automóvel [13]

A segurança automotiva é uma prioridade que tem levado ao desenvolvimento de tecnologias cada vez mais avançadas para prevenir acidentes e alertar os motoristas sobre condições inadequadas ou perigosas. Um circuito de alarme integrado ao sistema eletrônico do veículo desempenha um papel crucial ao alertar sobre situações que requerem atenção imediata.

O sistema é um exemplo de aplicação prática de um circuito combinacional para gerenciar alarmes de forma eficiente. Ele utiliza sensores para monitorar o estado da porta do motorista, do motor, dos faróis e do cinto de segurança. Com base nos dados coletados, o circuito avalia uma série de condições e, quando necessário, ativa um alarme para alertar o motorista sobre situações potencialmente perigosas ou não conformes.

Descrição do Sistema:

Um circuito de alarme de automóvel possui quatro sensores eletrônicos utilizados para indicar o estado da porta do motorista, do motor, dos faróis e do uso de cinto de segurança.

Projete o circuito combinacional mínimo que ative um alarme de acordo com as seguintes condições:

- Os faróis estão acesos e o motor está desligado; ou
- A porta do motorista está aberta e o motor está ligado; ou
- A porta do motorista está fechada, o motor está ligado e o passageiro não estiver usando o cinto de segurança.

- **P** - Porta P = 0 para porta fechada; P = 1 para porta aberta;
- **M** - Motor M = 0 para motor desligado; M = 1 para motor ligado;
- **F** - Faróis F = 0 para faróis apagados; F = 1 para faróis acesos
- **C** - Cinto de segurança C = 0 sem o cinto de segurança; C = 1 com o cinto de segurança.
- **A** - Alarme A = 0 alarme desativado; A = 1 alarme ativado.

Tabela 11 ilustra a tabela-verdade do circuito de forma não simplificada e a Figura 32 apresenta o circuito simulado no software *Multisim*.

Tabela 11 - Tabela-verdade do circuito de alarme de automóvel.

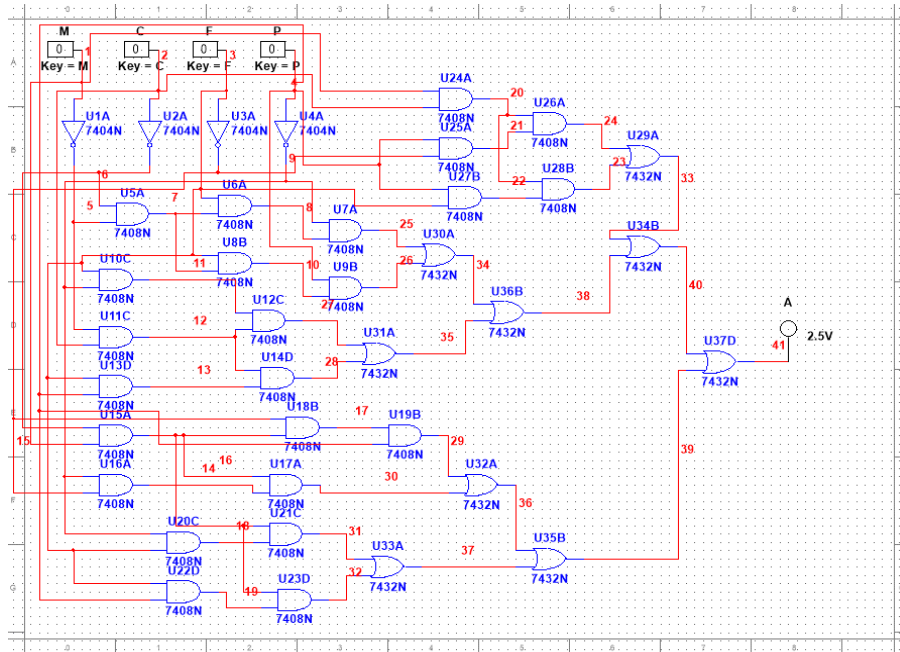
M	C	F	P	A
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Fonte: O autor, 2025.

A partir da tabela a expressão booleana correspondente é:

$$A = \bar{M}\bar{C}\bar{F}\bar{P} + \bar{M}\bar{C}F\bar{P} + \bar{M}C\bar{F}\bar{P} + \bar{M}C\bar{F}P + \bar{M}\bar{C}\bar{F}P + \bar{M}\bar{C}FP + M\bar{C}\bar{F}\bar{P} + M\bar{C}\bar{F}P + MC\bar{F}\bar{P} + MCF\bar{P}$$

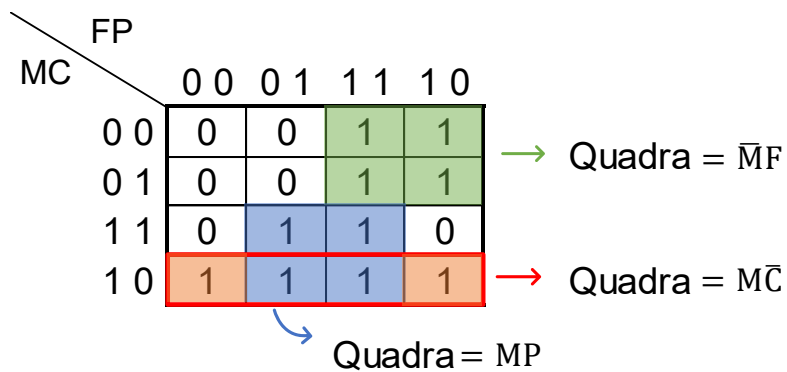
Figura 32 - Circuito de alarme de automóvel simulado no *Multisim*.



Fonte: O autor, *Multisim*, 2025.

A seguir, apresenta-se a simplificação do circuito, evidenciando o método empregado para melhorar sua organização e funcionalidade. A Figura 33 ilustra o processo de simplificação utilizando o mapa de Karnaugh, enquanto a Figura 34 apresenta o circuito otimizado simulado no *Multisim*.

Figura 33 - Circuito de alarme de automóvel simplificando pelo mapa k.

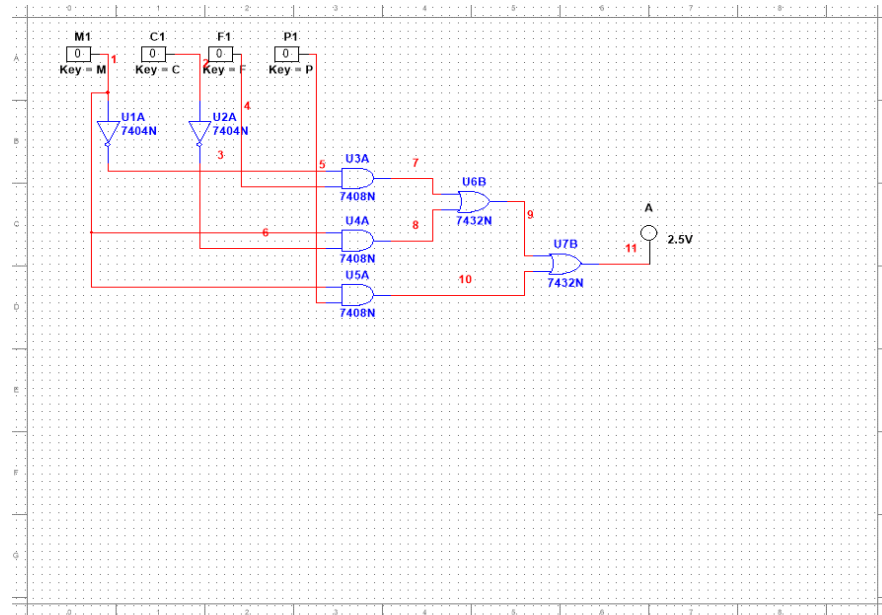


Fonte: O autor, 2025.

Com a seguinte simplificação, a nova expressão booleana pra o alarme é:

$$A = \bar{M}F + MP + M\bar{C}$$

Figura 34 - Simulação do circuito de alarme de automóvel simplificado, pelo *Multisim*.



Fonte: O autor, *Multisim*, 2025.

5. RESULTADOS E DISCUSSÕES

Neste capítulo, são apresentados os resultados referentes ao processo de simplificação dos circuitos e discussões desse procedimento realizado

A fim de validar e analisar as vantagens decorrentes da simplificação dos circuitos, é necessário realizar uma avaliação de diferentes aspectos, como a quantidade de portas lógicas utilizadas, os atrasos de propagação e a eficiência geral de cada projeto

Neste contexto, os circuitos propostos – controle de ar condicionado, sistema de controle de iluminação, controle de bombas d'água e alarme automotivo – serão analisados comparativamente em suas versões não simplificada e simplificada. Por meio dessas análises, será possível determinar:

- **A redução do número de portas lógicas:** Avaliando como a simplificação impacta na economia de recursos físicos e no consumo de energia.
- **Os atrasos de propagação:** Identificando o tempo necessário para que os sinais de entrada atinjam as saídas e como a redução de portas lógicas impacta no desempenho.
- **A eficiência geral dos circuitos:** Demonstrando as melhorias práticas obtidas com a simplificação.

Essa abordagem visa evidenciar os benefícios diretos e indiretos proporcionados pela otimização dos circuitos, destacando como a simplificação contribui para a viabilidade de implementação e a funcionalidade dos sistemas em aplicações reais. A análise servirá como base para entender a importância do projeto eficiente e racional de circuitos combinacionais no desenvolvimento de sistemas digitais.

5.1 Redução do número de portas lógicas

A primeira etapa do estudo consistiu em analisar as quantidades de portas lógicas utilizadas em cada circuito, tanto na sua forma não simplificada quanto após a simplificação. Essa análise é fundamental para compreender a complexidade estrutural inicial dos circuitos e os benefícios obtidos com a simplificação. Ao contabilizar o número de portas lógicas, foi possível identificar o impacto direto na eficiência, no consumo de energia e na viabilidade de implementação dos sistemas analisados, a comparação está ilustrada na Tabela 12.

Tabela 12 - Comparações dos circuitos nas suas formas não simplificadas e simplificadas.

CIRCUITO	SIMPLIFICAÇÃO	NOT	AND	OR	TOTAL
Ar Condicionado	Não Simplificado	3	16	5	24
	Simplificado	0	3	2	5
Iluminação	Não Simplificado	3	10	4	17
	Simplificado	0	1	1	2
Bombas D'água B1	Não Simplificado	4	18	5	27
	Simplificado	3	2	1	6
Bombas D'água B2	Não Simplificado	4	12	3	19
	Simplificado	1	1	0	2
Alarme de Automóvel	Não Simplificado	4	24	9	37
	Simplificado	2	3	2	7

Fonte: O autor, 2025.

As simplificações dos circuitos lógicos resultaram em reduções significativas no número de portas utilizadas. No sistema do quarto, o circuito não simplificado usava 24 portas, enquanto o simplificado reduziu esse número para 5, essa redução representa uma economia de 79% no número total de portas lógicas. No controle de iluminação, houve uma economia de 88%, passando de 17 para apenas 2 portas. No controle das bombas, a bomba B1 teve uma redução de 78%, de 27 para 6 portas, enquanto a bomba B2 passou de 19 para 2 portas, economizando 89%. Já o circuito de alarme automotivo reduziu de 37 para 7 portas, representando uma diminuição de 81%. Essas otimizações demonstram a eficiência da simplificação na redução da complexidade e do consumo de recursos eletrônicos.

Essas reduções trazem impactos diretos na eficiência do sistema. Com menos portas, o consumo de energia elétrica do circuito é consideravelmente menor, já que cada porta lógica consome uma quantidade de energia para operar. Em sistemas que funcionam continuamente, essa economia de energia se torna cumulativa, gerando benefícios de longo prazo.

O design fica mais enxuto e diminui a área ocupada no hardware, permitindo dispositivos mais compactos, econômicos e diminuindo os custos de manutenção, uma vez que o design mais simples requer menos material e é mais fácil de inspecionar ou reparar em caso de falhas. Assim, a simplificação no projeto destaca a importância da simplificação nos circuitos e os ganhos em eficiência e robustez.

Portanto, a análise comparativa entre os circuitos não simplificados e os simplificados dos sistemas destaca como a simplificação pode reduzir custos, melhorar a eficiência energética e estrutural, e aumentar a confiabilidade do sistema.

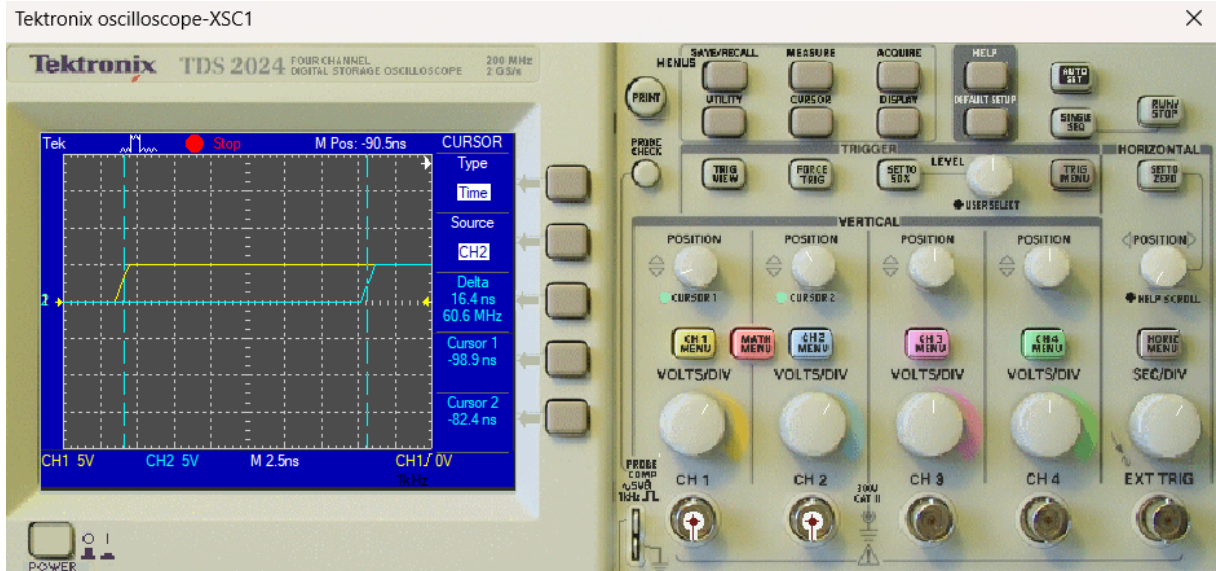
5.2 Os atrasos de propagação

A segunda etapa do estudo consistiu em avaliar os atrasos de propagação nos circuitos, tanto na versão não simplificada quanto na simplificada. Essa análise é para compreender como a redução no número de portas lógicas impacta o tempo de processamento dos sinais dentro do sistema.

Para esta fase do experimento, foi utilizado um osciloscópio no simulador *Multisim*. A ferramenta permitiu medir a diferença de tempo entre o sinal de entrada e a resposta na saída do circuito, identificando o tempo necessário para que a transição ocorra. A Figura 35 ilustra a leitura feita no circuito simplificado de controle de iluminação. O processo, foi repetido para todos os circuitos e os tempos estão apresentado na Tabela 13, foi feito de acordo com os seguintes passos:

1. Foi conectada uma das entradas no canal 1(CH 1) do osciloscópio e configurada as demais entradas para que o sinal de saída do circuito conectado no canal 2(CH 2) seja nível lógico alto quando a entrada conectada no canal 1 for 1.
2. Para exibir os canais, pressione os botões MENU CH 1(onda amarela) e, em seguida, MENU CH 2(onda azul).
3. Pressione AUTOSSET para acionar um display estável.
4. Ajustado os controles horizontal e vertical para otimizar o display.
5. Pressione o botão CURSOR para verificar o menu Cursor.
6. Pressione o botão da caixa de menu superior (Typer) para selecionar a opção Tempo.
7. Pressione o botão da caixa de menu Soucer para selecionar a opção CH1, e após o CH2.
8. Use o botão CURSOR 1 para posicionar o cursor na borda ativa do sinal de seleção do microcomponente.
9. Use o botão CURSOR 2 para posicionar o segundo cursor na transição da saída de dados.
10. Leia o atraso de propagação na leitura Delta do menu de cursor.

Figura 35 - Leitura do atraso de propagação do circuito simplificado de controle de iluminação feita no osciloscópio.



Fonte: O autor, *Multisim*, 2025.

Tabela 13 - Tempo de atraso de propagação dos circuitos não simplificados e simplificados.

CIRCUITO	SIMPLIFICAÇÃO	TEMPO	UNIDADE
Ar Condicionado	Não Simplificado	101	ns
	Simplificado	43,4	
Iluminação	Não Simplificado	83,5	
	Simplificado	16,4	
Bombas D'água B1	Não Simplificado	86,4	
	Simplificado	43,1	
Bombas D'água B2	Não Simplificado	86	
	Simplificado	29	
Alarme de Automóvel	Não Simplificado	100	ns
	Simplificado	58,6	

Fonte: O autor, 2025.

As simplificações dos circuitos resultaram em uma diminuição significativa nos atrasos de propagação, melhorando o tempo de resposta dos sistemas, impactando diretamente no consumo de recursos e no desempenho. A redução do número de portas lógicas não apenas simplificou a estrutura dos circuitos, tornando-os mais compactos e econômicos, mas também permitindo respostas mais rápidas e precisas. Essas melhorias são essenciais para sistemas

automatizados, onde a velocidade do processamento é fator crítico para o funcionamento adequado.

Além disso, circuitos com menor latência tornam-se mais viáveis para implementação em dispositivos como sistemas de automação residencial, que se torna um controle mais dinâmico ajustando-se quase instantaneamente às mudanças no ambiente, no controle das bombas d'água, que garante uma resposta mais eficiente na regulação dos níveis dos reservatórios, reduzindo o que poderiam comprometer o fluxo adequado de água, e automotiva que melhora a prontidão do sistema. Dessa forma, a simplificação dos circuitos analisados demonstrou ser uma estratégia vantajosa, proporcionando mais eficiência para aplicações diversas.

5.3 A eficiência geral dos circuitos

A otimização dos circuitos digitais não é apenas reduzir o número de portas lógicas, mas também atrasar a propagação do sinal ao mínimo possível. A redução levou a uma diminuição na complexidade da fiação, à capacidade de construir circuitos mais compactos e à redução do tempo de sinal de comutação. Menos processamento interno para as aplicações resulta em uma redução no consumo de energia e um aumento na estabilidade do circuito, especialmente em sistemas embarcados e dispositivos com recursos limitados.

Outro ponto é a escalabilidade dos circuitos simplificados. Com menos componentes e menos espaço físico para a demanda, esses sistemas podem facilmente ser incorporados a dispositivos portáteis, sistemas de IoT e empresas que realizam tarefas automatizadas. A confiabilidade sobe a taxas significativas, uma vez que os circuitos menos complexos têm taxa de erros lógicos, ruído elétrico e interferência eletromagnética, menores. Essa redução de falhas é essencial para aplicações como alarmes e controle de processos automatizados, onde a precisão e a resposta imediata são indispensáveis.

Por último, a simplificação também influencia de maneira direta a flexibilidade dos projetos, que podem sofrer futuras expansões ou adaptações com muito menos esforços. A redução do número de componentes também facilita a produção em larga escala e diminui os custos de manutenção e substituição de peças, tornando a implementação dos projetos mais acessível e sustentável. Portanto, os benefícios obtidos por meio da otimização dos circuitos

não estão restritos apenas ao desempenho imediato, mas prestam benefícios no longo prazo em que sistemas mais bem fundamentados, econômicos e eficazes serão garantidos.

6. CONSIDERAÇÕES FINAIS

As simplificações e análises dos circuitos evidenciam como a otimização dos sistemas lógicos influencia diretamente a eficiência, o desempenho e a viabilidade de sua implementação, como redução nos custos de produção e manutenção simplificada. A diminuição do número de portas lógicas, aliada à redução dos atrasos de propagação, resulta em circuitos mais ágeis, compactos e com menor consumo energético. Esses fatores destacam a importância de otimizar circuitos desde as fases iniciais do projeto, possibilitando soluções mais eficazes e adaptáveis a diferentes aplicações.

Entretanto, para que essas melhorias sejam alcançadas, é indispensável um conhecimento em Álgebra Booleana e em técnicas de simplificação de expressões lógicas. Com isso, engenheiros e projetistas conseguem desenvolver sistemas mais inteligentes, acessíveis e sustentáveis. Dessa forma, a simplificação lógica se consolida como um elemento essencial no desenvolvimento de circuitos digitais, impulsionando inovação e eficiência na criação de sistemas eletrônicos.

Neste trabalho, apresentamos algumas aplicações práticas desses conceitos que foram realizados por meio da análise e comparação entre circuitos em suas formas simplificada e não simplificada. Os resultados confirmaram que a utilização da Álgebra Booleana e de técnicas de minimização de circuitos lógicos proporciona melhorias significativas. Assim, comprovou-se que a simplificação lógica não é apenas uma ferramenta teórica, mas uma estratégia essencial para otimizar projetos eletrônicos, tornando-os mais eficazes e adequados às necessidades tecnológicas atuais.

7. REFERÊNCIAS

- [1] TOCCI, Ronald Joseph; WIDMER, Neal Scott; MOSS, Gregory L. **Sistemas digitais: princípios e aplicações**. 11. ed. São Paulo: Pearson Prentice Hall, 2011. 817 p. (ISBN 978-85-7605-922-6). Revisão técnica Renato Giacomini; tradução Jorge Ritter. Disponível em: <http://plataforma.bvirtual.com.br/Leitor/Publicacao/2621>. Acesso em: 15 out. 2024.
- [2] CAPUANO, Francisco Gabriel; IDOETA, Ivan Valeije. **Elementos de Eletrônica Digital**. 42. ed. São Paulo: Érica, 2019. 440 p. (ISBN 978-85-365-3039-0). Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788536530390/pageid/58>. Acesso em: 15 out. 2024.
- [3] COSTA, Breno R. de S.; SILVA, Paulo C. L.; GARCIA, Antonio R. G. **A UTILIZAÇÃO DA ÁLGEBRA BOOLEANA NA SIMPLIFICAÇÃO DE CIRCUITOS LÓGICOS**. 2022. 15 f. TCC (Graduação) - Curso de Bacharelado em Ciência e Tecnologia, Departamento de Ciências Naturais Matemática e Estatística, Universidade Federal Rural do Semi-Árido, Mossoró, 2022. Disponível em: <https://repositorio.ufersa.edu.br/items/78e42344-9360-4845-b57f-c6d24e5d0b36>. Acesso em: 15 out. 2024.
- [4] SOUSA, Joilson Sena de. **ÁLGEBRA BOOLEANA: histórico e aplicações**. 2018. 36 f. TCC (Graduação) - Curso de Licenciatura Integrada em Matemática e Física, Universidade Federal do Oeste do Pará, Santarém, 2018. Disponível em: https://scholar.google.com.br/scholar?hl=pt-BR&as_sdt=0%2C5&scioq=Introdu%C3%A7%C3%A3o+aos+Sistemas+Digitais&q=%C3%81LGEBRA+BOOLEANA%3A+hist%C3%B3rico+e+aplica%C3%A7%C3%B5es&btnG=. Acesso em: 15 out. 2024.
- [5] SANTOS, Vilton Ricardo; OLIVEIRA, Cassius Gomes. **A ÁLGEBRA BOOLENA PRESENTE NOS CIRCUITOS LÓGICOS**. Caderno de Graduação - Ciências Exatas e Tecnológicas - UNIT - SERGIPE, [S. l.], v. 3, n. 3, p. 63, 2016. Disponível em: <https://periodicosgrupotiradentes.emnuvens.com.br/cadernoexatas/article/view/3471>. Acesso em: 15 out. 2024.
- [6] GÜNTZEL, José Luís; NASCIMENTO, Francisco Assis do. **Introdução aos sistemas digitais**. Disponível em www.ufpel.edu.br/~guntzel/isd/isd.html, (versão 2001/1), 2001. Acesso em: 15 out. 2024.
- [7] DIAS, Carlos Magno Corrêa. **Álgebra booleana e lógica digital uma aplicação da lógica matemática**. Revista Acadêmica: ciências agrárias e ambientais, 1994. Disponível em: https://repositorio.utfpr.edu.br/jspui/bitstream/1/424/1/REV.%20ACAD._Dias%2C%20Carlos%20Magno%20Corr%C3%AAa_1994.pdf. Acesso em: 20 out. 2024.
- [8] HIRATA, Nina ST. **Notas de aula de MAC0329–Álgebra Booleana e Aplicações**. São Paulo: USP, 2006. Disponível em: <http://www.vision.ime.usp.br/~nina/cursos/mac0329-03/p1.pdf>. Acesso em: 20 out. 2024.
- [9] SERRANO, Ariana; NASCIMENTO, Vitor. **PSI 3212 - Laboratório de Circuitos Elétricos: Utilização do Multisim**. São Paulo: Escola Politécnica da Universidade de São Paulo, 2020. Disponível em: <https://scholar.google.com.br/citations?hl=pt-BR&user=Ljmu-QUAAAAJ>. Acesso em: 30 out. 2024.

- [10] FAIRCHILD. **7432 Datasheet (PDF) - Fairchild Semiconductor**. 2000. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/view/51073/FAIRCHILD/7432.html>. Acesso em: 30 out. 2024.
- [11] VIEIRA, Fábila Magali Santos. Álgebra booleana. **Educação & Tecnologia**, v. 5, n. 1, 2000. Disponível em: <https://www.periodicos.cefetmg.br/index.php/revista-et/article/view/341>. Acesso em: 20 jan. 2025
- [12] MATEUS, Gustavo Pereira. **CARACTERIZAÇÃO AUTOMÁTICA DE ATRASOS DE PORTAS LÓGICAS EM CIRCUITOS COMBINACIONAIS CMOS**. 2006. 56 f. TCC (Graduação) - Curso de Bacharelado em Ciência da Computação, Departamento de Informática, Universidade Federal de Pelotas, Pelotas, 2006. Disponível em: https://scholar.google.com.br/citations?view_op=list_works&hl=pt-BR&hl=pt-BR&user=tvaMoI4AAAAJ. Acesso em: 20 jan. 2025
- [13] BASTOS, Sandro Rodrigo G. **Sistemas Digitais I**. Universidade Santa Cecília, 2009. Disponível em: <https://pt.scribd.com/document/21872794/circuito-logico>. Acesso em: 15 out. 2024.