

DA LÓGICA AOS CIRCUITOS DIGITAIS: UMA ANÁLISE INTEGRADA

Guilherme Silva Aguiar¹, Bruno Francisco Xavier²

Resumo: Este artigo de revisão discute a integração conceitual entre lógica proposicional, álgebra Booleana e circuitos digitais, destacando seu papel essencial na formação de estudantes na área de Tecnologia. Mediante análise bibliográfica integrativa e seleção sistemática de obras especializadas, apostilas didáticas, dissertações e estudos de caso em circuitos combinacionais, somadores e subtratores, investiga-se como a compreensão prévia de lógica formal facilita o aprendizado e a síntese dos circuitos digitais, especialmente na transição entre conteúdos matemáticos e tecnológicos em cursos de engenharia. A pesquisa inclui uma análise estruturada da conexão intrínseca entre conceitos da lógica formal (proposições, tabelas-verdade) e a síntese de circuitos digitais, mapeando convergências e lacunas acadêmicas. As análises confirmam que a articulação didática entre lógica e circuitos potencializa autonomia intelectual e aplicabilidade das competências de engenharia, evidenciando desafios recorrentes no ensino tradicional, como despreparo metodológico, ausência de interdisciplinaridade e limitações curriculares. Sugerem-se abordagens integradas e novas sequências didáticas para fortalecer a progressão lógica-algébrica-física na formação tecnológica. Conclui-se que uma abordagem pedagógica que desenvolva disciplinas de lógica previamente ou juntamente com circuitos digitais otimiza o aprendizado e fortalece a capacidade de abstração do estudante, indo além da simples memorização de regras da Álgebra Booleana.

Palavras-chave: lógica proposicional. álgebra Booleana. circuitos digitais. interdisciplinaridade. engenharia. formação tecnológica. síntese de circuitos

1. INTRODUÇÃO

O desenvolvimento tecnológico contemporâneo é sustentado por fundamentos lógicos e matemáticos que garantem desde a formulação de proposições até a implementação de sistemas digitais avançados. A literatura da área de sistemas digitais destaca que a álgebra booleana constitui um meio sistemático e conveniente para expressar e analisar o funcionamento de circuitos lógicos, evidenciando seu papel central na modelagem e no entendimento de sistemas digitais [1].

Estudos clássicos apontam que a álgebra booleana pode ser compreendida como a base matemática dos sistemas digitais, sendo indispensável para a análise e o projeto de circuitos lógicos [1]. Dessa forma, a integração entre lógica proposicional e álgebra booleana revela-se fundamental na formação em tecnologia, sobretudo nos cursos de engenharia, onde a compreensão desses conceitos impacta diretamente a capacidade de síntese e abstração do estudante.

A literatura especializada também evidencia que diversos postulados e conceitos algébricos estão diretamente presentes tanto na estrutura interna dos circuitos quanto em suas aplicações matemáticas. Nesse sentido, autores destacam que a álgebra de Boole fundamenta-se no sistema de numeração binário, o que permite a realização de operações lógicas e aritméticas utilizando apenas dois estados, característica que a torna especialmente adequada para representar circuitos eletrônicos digitais, portas lógicas, números e caracteres em sistemas computacionais [6]. Apesar dessa relevância, observa-se que, na prática pedagógica, esses conteúdos são frequentemente apresentados de forma fragmentada, sendo ensinados apenas como um conjunto de regras ou postulados a serem memorizados, sem a devida articulação conceitual com a lógica proposicional.

Diante desse cenário, este artigo busca responder ao seguinte problema de pesquisa: como a compreensão prévia da lógica formal impacta o processo de aprendizagem e a síntese de circuitos digitais? O objetivo consiste em analisar a literatura das áreas de lógica proposicional, álgebra booleana e circuitos digitais, demonstrando que os conceitos oriundos da matemática discreta são fundamentais para uma compreensão profunda, e não meramente operacional, dos sistemas digitais. Para tanto, adota-se como metodologia uma revisão bibliográfica, explorando os pilares da lógica formal, sua tradução algébrica e sua posterior materialização em circuitos digitais.

¹Discente do Curso Bacharelado em Ciência e Tecnologia, UFRSA, Mossoró – Rio Grande do Norte. E-mail: guilherme.aguiar@alunos.ufersa.edu.br.

²Docente do Curso de Engenharia de Computação, Departamento de Engenharias e Tecnologia, UFRSA, Pau dos Ferros – Rio Grande do Norte. E-mail: bruno.xavier@ufersa.edu.br.

2. DESENVOLVIMENTO

2.1. Método e Abordagem da Revisão

O método empregado é a revisão bibliográfica analítica. A abordagem deste artigo, conforme recomendado para trabalhos de revisão, não se limita a resumir as fontes, mas sim a "analisar e discutir trabalhos já publicados" para defender a tese central. A exposição ordenada do assunto segue um percurso dedutivo em três etapas, que espelham o processo de síntese de circuitos, indo do abstrato ao concreto:

A Lógica Formal: Análise do Cálculo Proposicional como sistema formal de raciocínio, abordando seus conectivos e tabelas-verdade, conforme descrito na literatura base.

A Álgebra de Tradução: Análise da Álgebra Booleana como a estrutura matemática que formaliza a lógica para aplicação em sistemas binários, incluindo seus postulados e teoremas fundamentais.

A Síntese Física: Análise das portas lógicas como implementações físicas dos operadores booleanos e sua utilização na construção de circuitos combinacionais básicos.

A discussão dos resultados será focada na interseção destas três etapas, argumentando que a lacuna pedagógica em conectar a primeira etapa (lógica) à terceira (síntese de circuitos) prejudica o aprendizado.

2.2 Lógica Proposicional

A lógica matemática é o alicerce fundamental do pensamento computacional. Ela pode ser definida como a ciência que estuda as formas do pensamento, utilizando regras e técnicas para determinar a validade de um argumento. Para os propósitos da Ciência da Computação, ela fornece uma linguagem formal, livre de ambiguidades, que permite descrever sistemas complexos.

A base dessa estrutura é a Lógica Proposicional (LP) que estuda como as sentenças declarativas podem ser combinadas e avaliadas. Todo o nosso argumento se inicia com o elemento mais básico deste cálculo: a proposição. Ela é uma sentença declarativa livre de ambiguidade e que tenha a propriedade de ser verdadeira ou falsa, mas não ambas.

As proposições podem ser combinadas por meio de conectivos para formar novas proposições. Os principais conectivos em qualquer curso básico de lógica são a negação ($\neg$), conjunção ($\wedge$), disjunção ($\vee$), implicação ($\rightarrow$) e bi-implicação ($\leftrightarrow$). Por exemplo, considere as proposições abaixo:

- p : Está chovendo
- q : Hoje é quarta

Podemos criar novas proposições adicionando conectivos:

- $\neg p$: Não está chovendo
- $q \wedge p$: Hoje é quarta e está chovendo
- $p \vee q$: Está chovendo ou hoje é quarta
- $p \rightarrow q$: Se está chovendo, então hoje é quarta

2.2.1 Sintaxe Básica

Se uma proposição p é indivisível (variável proposicional), isto é, não possui conectivos lógicos internamente, então ela é dita atômica. Além disso, p e seu complemento $\neg p$ são chamados de literais.

Definimos recursivamente o conjunto de fórmulas da lógica proposicional $Form$ como sendo o menor conjunto tal que:

1. Toda variável proposicional é uma fórmula.
2. Se p é fórmula, então $\neg p$ é fórmula.
3. Se p e q são fórmulas, então também são fórmulas: $p \wedge q$, $p \vee q$, $p \rightarrow q$, $p \leftrightarrow q$

2.2.2 Função de verdade e tabelas-verdade

Dado um conjunto de variáveis proposicionais, podemos atribuir a cada uma delas um valor em $\{V, F\}$. Ao aplicar essas atribuições a uma fórmula, obtemos um valor único, chamado valor de verdade da fórmula. Formalmente, cada fórmula proposicional com n variáveis determina uma função $f: \{V, F\}^n \rightarrow \{V, F\}$, obtida pela avaliação de seus conectivos. Essa função é chamada de função booleana e é representada operacionalmente por sua tabela-verdade.

A tabela-verdade é uma ferramenta um instrumento matemático, que permite determinar o valor lógico de proposições compostas. De acordo com [3, p. 3], tabelas-verdade são diagramas que explicitam a relação entre os valores-verdade de uma expressão composta em termos dos valores-verdade das sub-expressões e variáveis que a compõem.

Tabela 1. Tabela de conectivos básicos.

p	q	$p \wedge q$	$p \vee q$	$\neg p$	$\neg q$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

Fonte: Autoria Própria (2025).

Dessa forma, é necessário analisar todas as combinações possíveis de valores-verdade para cada proposição atômica. Portanto, se uma proposição é composta por n proposições indivisíveis, então haverá 2^n combinações possíveis.

Estas tabelas são a base para a análise de proposições compostas mais complexas. Para avaliar uma fórmula como, por exemplo, $p \vee q \rightarrow \neg p$ (1), constrói-se uma tabela-verdade com colunas intermediárias para cada sub-operação, permitindo encontrar o valor-verdade final da expressão inteira.

Tabela 2. Tabela de expressão (1).

p	q	$p \vee q$	$\neg p$	$p \vee q \rightarrow \neg p$
0	0	0	1	1
0	1	1	1	1
1	0	1	0	0
1	1	1	0	0

Fonte: Autoria Própria (2025).

A LP está presente no uso de estruturas de controle em linguagens de programação. O teste de uma condição em uma instrução if é uma proposição que deve ser avaliada como verdadeira ou falsa. Os operadores AND, OR, NOT (ou &&, ||, !) são a implementação direta dos conectivos lógicos. Além disso, as regras de validação de formulários ou entrada de dados são frequentemente expressas como fórmulas proposicionais.

A LP compõe uma estrutura algébrica mais abrangente chamada de álgebra de Boole composta por operadores booleanos e domínio em $\{0, 1\}$. O tipo bool em linguagens de programação vem de *boolean* que, por sua vez, vem da álgebra de Boole.

2.3 Álgebra Booleana

O filósofo e matemático britânico George Boole desempenhou um papel fundamental no desenvolvimento da lógica simbólica e algébrica. Ele desenvolveu a álgebra Booleana que é o fundamento teórico por trás do funcionamento dos circuitos digitais.

Boole iniciou suas publicações de destaque em 1841 focando na Teoria das Transformações Analíticas. Segundo [1], foi o seu trabalho intitulado “Uma investigação das leis do pensamento” que uma álgebra lógica, conhecida hoje em dia como álgebra Booleana, foi formulada.

Sua contribuição foi amplamente reconhecida em 1844, quando recebeu a medalha de ouro da Royal Society pelo artigo *On a General Method of Analysis*. Nesta obra, ele demonstrou a interconexão entre álgebra e lógica, propondo a união de métodos de cálculo com a álgebra. Em 1847, consolidando seu pensamento inovador, publicou *The Mathematical Analysis of Logic*, argumentando que a lógica deveria ser tratada como um ramo da matemática e não da filosofia.

Em 1938, o matemático americano Claude Shannon percebeu o paralelo entre lógica proposicional e lógica de circuitos, e compreendeu que a álgebra de Boole poderia ter um papel importante na sistematização desse novo ramo da eletrônica.

2.3.1 Definição de Álgebra Booleana

Uma álgebra Booleana é uma 6-tupla $(B, +, \cdot, \neg, 0, 1)$ onde

- B é um conjunto não vazio (universo dos elementos)
- $+$ e $\cdot$ são operações binárias chamadas soma e produto booleano, respectivamente
- $\neg$ é uma operação unária
- 0 e 1 são elementos distintos em B

As operações binárias devem ser comutativas e associativas. Além disso, $\cdot$ distribui sobre $+$ e vice-versa. Os elementos 0 e 1 são elementos neutros em relação a $+$ e $\cdot$, respectivamente. Além disso, $a \cdot \bar{a} = 0$ e $a + \bar{a} = 1$, para todo a em B .

É fácil constatar que a lógica proposicional que definimos anteriormente é uma álgebra de Boole. Basta definir 0 como F , 1 como V .

$$a + b \stackrel{\text{def}}{=} a \vee b$$

$$a \cdot b \stackrel{\text{def}}{=} a \wedge b$$

$$\bar{a} \stackrel{\text{def}}{=} \neg a$$

Dessa forma, todas as propriedades podem ser verificadas no quadro abaixo.

Quadro 1 - Propriedades booleanas

Axioma	Conjunção ($\wedge$)	Disjunção ($\vee$)
Associatividade	$a \wedge (b \wedge c) = (a \wedge b) \wedge c$	$a \vee (b \vee c) = (a \vee b) \vee c$
Comutatividade	$a \wedge b = b \wedge a$	$a \vee b = b \vee a$
Identidade/Neutro	$a \wedge V = a$	$a \vee F = a$
Distributividade	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
Complemento	$a \wedge \neg a = F$	$a \vee \neg a = V$

Fonte: Autoria Própria (2025)

Outro exemplo de álgebra booleana é a 6-tupla $(\mathcal{P}(A), \cap, \cup, \setminus, \emptyset, A)$ onde $\mathcal{P}(A)$ é o conjunto das partes de A , $\cap$ é a operação de interseção, $\cup$ é a união e $\setminus$ é o operador de complemento.

Como os circuitos digitais operam exclusivamente em um estado binário (tensão alta ou baixa), eles também se encaixam perfeitamente nos axiomas e operações da Álgebra Booleana.

2.3.2 Postulados e Teoremas Fundamentais

A Álgebra Booleana é formalizada por um conjunto de postulados que servem como base para derivar identidades e teoremas usados na simplificação e análise de funções lógicas. As propriedades comumente usadas na fundamentação algébrica booleana são fundamentais para o desenvolvimento de um racional aplicado para circuitos digitais.

Com as propriedades inerentes a esta estrutura algébrica (associatividade, comutatividade, identidade, distributividade e complemento) podemos inferir uma série de resultados importantes.

Quadro 2 - Teoremas e postulados

Identidade	Identidade
Leis de De Morgan	$\overline{a + b} = \bar{a} \cdot \bar{b}$ $\overline{a \cdot b} = \bar{a} + \bar{b}$
Leis idempotentes	$a + a = a$ $a \cdot a = a$
Leis da anulação	$a + 1 = 1$ $a \cdot 0 = 0$
Leis de absorção	$a + a \cdot b = a$ $a \cdot (a + b) = a$
Leis de involução	$\bar{\bar{a}} = a$

Fonte: Autoria Própria (2025)

Munido dessas leis, podemos provar equivalências lógicas e otimizar expressões antes da implementação física dos circuitos. Para maiores informações sobre outras propriedades, consulte [3].

2.3.3 Funções Booleanas

Uma variável a é dita booleana se seus únicos valores são 0 ou 1. Já uma função f é dita booleana se recebe como

argumento um tupla de variáveis booleanas e retorna um valor no conjunto $\{0, 1\}$. Por exemplo, $f(a, b) = a + (b + 1)$ é uma função booleana que retorna sempre 1, pois $a + (b + 1) = a + 1 = 1$, usando duas vezes a lei da anulação.

Um problema importante na Álgebra de Boole é saber como construir uma expressão booleana dado os valores de uma função booleana. Isso é possível porque podemos representar qualquer função booleana por uma soma de produtos de literais.

As formas canônicas constituem um modelo sistemático de representação de funções booleanas, permitindo traduzir a tabela-verdade em uma expressão algébrica precisa. Vamos abordar os mintermos (soma de produtos) e maxtermos (produtos de soma), que partem de deduções lógicas matemáticas, que na prática, esses, auxiliam durante o desenvolvimento de circuitos digitais.

Um mintermo corresponde aos átomos de uma álgebra de funções booleanas, sendo as unidades elementares que compõem essas funções. A partir do teorema da representação, toda função booleana pode ser expressa como uma disjunção (soma) de seus mintermos, o que garante uma forma sistemática de descrição algébrica das funções lógicas [11].

De maneira dual, um maxtermo é a soma (OR) de n literais, verdadeira para todas as combinações, exceto uma específica — correspondendo às saídas 0. O Produto de Maxtermos (POS) é, portanto, a conjunção (AND) de todos os maxtermos associados às linhas da tabela-verdade em que a função é falsa. Observa que essas formas são equivalentes, uma vez que descrevem completamente a função, diferindo apenas na forma de agrupar os resultados.

A seguir, temos um exemplo com duas variáveis A e B , ilustrando a correspondência entre as linhas da tabela-verdade e as formas canônicas.

A função considerada é $f(A,B) = 1$ para as combinações $(A,B) = (0,1)$ e $(1,0)$. (ainda vou formatar essa tabela)

Tabela 3. Representação por Mintermos e Maxtermos.

A	B	$f(A,B)$	Mintermo ativo
0	0	0	—
0	1	1	$A'B$
1	0	1	AB'
1	1	0	—

Fonte: Autoria Própria (2025).

Assim, $f(A,B) = A'B + AB'$, que corresponde exatamente à função XOR. O mesmo raciocínio pode ser aplicado para construir a forma POS, considerando os maxtermos associados às saídas 0: $f(A,B) = (A' + B')(A + B)$.

Na teoria de funções booleanas, um conjunto de operadores é dito funcionalmente completo quando permite representar qualquer função booleana a partir de combinações desses operadores. Em geral, parte-se do fato de que os conectivos básicos da lógica proposicional (conjunção, disjunção e negação) formam um conjunto completo, e que, a partir deles, é possível reescrever qualquer tabela-verdade como expressão algébrica. Em seções posteriores, serão apresentados exemplos de portas lógicas que também constituem conjuntos funcionalmente completos na implementação de circuitos digitais

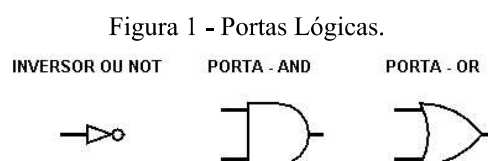
2.4 Circuitos Digitais

Em uma Álgebra Booleana, é bom frisar que as constantes são distintas entre si. É neste contexto binário que se fundamenta a computação digital. A propósito, os circuitos digitais são sistemas eletrônicos que manipulam e processam informações codificadas em apenas dois estados discretos ou níveis lógicos: 0 (baixo/falso) e 1 (alto/verdadeiro).

O principal componente de um circuito digital é a porta lógica. Essas portas formam os Circuitos Integrados (CIs), como microprocessadores e memórias. Abaixo, a representação pictórica das portas lógicas NOT, AND e OR.

2.4.1 Portas Lógicas

No mundo real, as portas lógicas são construídas usando componentes eletrônicos como transistores. Para a fase de projeto do circuito, usamos figuras representativas para abstrairmos o circuito propriamente dito. A Figura 1 mostra a representação gráfica das portas lógicas NOT, AND e OR.



Fonte: Autoria Própria (2025).

Note que a porta AND possui duas entradas e uma saída. A saída só será 1, se ambas as entradas estiverem em nível alto. Isso é deveras semelhante ao operador $\wedge$. O mesmo acontece entre as portas OR e NOT em relação à $\vee$ e $\neg$, respectivamente.

Cada porta recebe sinais de entrada (tipicamente 0V ou 5V, representando 0 ou 1), realiza uma operação lógica, e produz um sinal de saída. A correspondência entre operadores booleanos e portas lógicas é direta e fundamental:

- A porta NOT (inversor) implementa a negação booleana.
- A porta AND implementa a conjunção booleana (multiplicação lógica).
- A porta OR implementa a disjunção booleana (adição lógica).

Nesta altura, o leitor já deve ter percebido que os circuitos digitais compõem uma álgebra de Boole.

Nesse sentido, a álgebra dos circuitos insere-se no quadro geral da álgebra booleana como um sistema de dois elementos, sendo estruturalmente idêntica à álgebra das proposições, exceto por suas diferenças terminológicas e de aplicação prática [2].

A relação entre lógica proposicional e portas lógicas pode ser entendida como uma continuidade natural entre o pensamento abstrato e sua implementação física. Essa equivalência permite enxergar as portas lógicas não apenas como componentes eletrônicos, mas como realizações concretas de estruturas lógicas que antes existiam apenas no plano teórico. Assim, o funcionamento dos circuitos digitais passa a ser compreendido como uma materialização das mesmas regras que organizam o raciocínio proposicional.

No estudo dos circuitos digitais, podemos aumentar o nível de abstração usando equações algébricas. Em particular usamos os símbolos $+$, $\cdot$ e $\neg$ para representar a conjunção (AND), a disjunção (OR) e a negação (NOT), respectivamente, assim como fizemos com a lógica.

Quadro 3 - Relação de conectivos

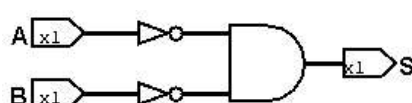
Lógica	Circuitos
$a \vee b$	$a + b$
$a \wedge b$	$a \cdot b$ ou AB
$\neg a$	$\bar{A}$ ou A'

Fonte: Autoria Própria (2025)

Desse modo, não se faz necessário desenhar as portas sempre que formos criar um circuito. Além disso, podemos tirar proveito das características da álgebra de Boole para simplificar os circuitos.

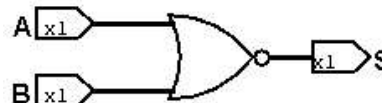
Podemos usar como exemplo a lei de De Morgan que diz: $\overline{A \cdot B} = \bar{A} + \bar{B}$. Em circuitos digitais, teremos os seguintes circuitos equivalentes:

Figura 2 - Expressão $\bar{A} \bar{B}$



Fonte: Autoria própria (2025)

Figura 3 - Expressão $\overline{A + B}$

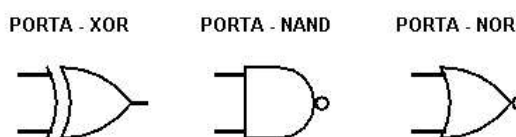


Fonte: Autoria própria (2025)

No circuito da figura 3, usamos menos componentes em comparação com a figura 2 e isso é importante em um mundo onde os recursos são escassos e caros, além da otimização de espaço em um microchip.

Como explicado na seção 2.3.3, todas as funções booleanas podem ser representadas usando outros operadores, nos circuitos também temos portas correspondentes.

Figura 4 - Portas Lógicas.



Fonte: Autoria Própria (2025).

-
- A porta XOR (exclusive OR) implementa a disjunção exclusiva.
 - A porta NAND (NOT AND) implementa a primeira lei de De Morgan
 - A porta NOR (NOT OR) implementa a segunda lei de De Morgan

Embora a porta XOR não seja funcionalmente completa, ela é particularmente importante em circuitos aritméticos.

2.5 Relação entre Lógica e Circuitos Digitais

Como mencionado, tanto a lógica proposicional quanto os circuitos digitais compõem uma álgebra de Boole. Isso significa que toda expressão booleana pode ser representada tanto como fórmula lógica como em hardware (portas digitais).

No campo da lógica, uma tautologia é uma proposição que é sempre verdadeira, em qualquer cenário. Já em uma contradição, o valor-verdade será sempre falso. No campo dos circuitos, a identificação de tautologias garante que o circuito funcionará sob quaisquer entradas, enquanto a detecção de contradições evita redundâncias ou falhas na formulação.

Enquanto isso, as equivalências lógicas permitem substituições seguras de expressões, favorecendo a simplificação de sistemas digitais. Essa é uma etapa crucial para o desenvolvimento de circuitos integrados, uma vez que permite reduzir o tamanho e custo de um circuito.

Existem dois resultados demasiadamente importantes na lógica proposicional chamados de completude e corretude. Para nosso campo de estudo, o primeiro implica que cada especificação funcional expressa em lógica proposicional pode ser implementada em hardware. Já a corretude, todo circuito digital tem uma expressão lógica equivalente. Isso permite a validação formal de circuitos com exatidão. Portanto, o poder expressivo dessas duas disciplinas é exatamente o mesmo.

A correção e a completude garantem que a lógica proposicional fornece uma fundamentação rigorosa e exata para o comportamento dos circuitos digitais: tudo o que pode ser especificado, provado ou otimizado em álgebra booleana pode ser reproduzido em portas lógicas. E tudo o que um circuito faz pode ser representado como fórmula bem definida.

Portanto, podemos (1) verificar se um circuito implementa corretamente uma especificação, (2) comparar dois circuitos e decidir se são equivalentes e (3) justificar matematicamente a corretude de um projeto.

Neste trabalho, são abordados circuitos combinacionais, os quais são compostos por uma rede de portas lógicas interconectadas, em que as saídas dependem exclusivamente das entradas presentes, sem qualquer referência a estados anteriores. Esses circuitos são caracterizados por comportamento instantâneo, não possuindo memória, e normalmente são derivados de funções booleanas [9].

A estrutura de um circuito combinacional é definida pelo grafo de conectividade das portas. As entradas primárias alimentam as portas de primeira camada, cujas saídas alimentam portas de segunda camada, e assim sucessivamente, até as saídas primárias. Não há loops de retroalimentação; o fluxo de dados é unidirecional das entradas para as saídas.

2.6 A Síntese de Circuitos Digitais

Parte da progressão conceitual envolvendo a construção de circuitos digitais, se dá pela materialização da Álgebra Booleana em estruturas eletrônicas reais. Esta etapa é onde a teoria lógica-algébrica se converte em implementação prática, permitindo a construção de dispositivos computacionais funcionais.

A progressão conceitual envolvendo a construção de circuitos digitais, se dá pela materialização da Álgebra Booleana em estruturas eletrônicas reais. Esta etapa é onde a teoria lógica-algébrica se converte em implementação prática, permitindo a construção de dispositivos computacionais.

Desse modo, o processo de construção de circuitos digitais segue uma sequência lógica bem definida: (1) especificação da função em tabela-verdade, (2) obtenção de expressão booleana, (3) simplificação e (4) implementação em circuito utilizando portas lógicas disponíveis.

A síntese de circuitos digitais pode ser realizada por meio de várias metodologias, cada uma adequada a diferentes contextos de especificação e complexidade. As abordagens mais comuns incluem a tabela-verdade, diagramas de tempo, expressões booleanas diretas, mapas de Karnaugh, métodos algébricos e descrições de hardware (VHDL/Verilog).

Este trabalho restringe-se a métodos e demonstrações fundamentados na álgebra booleana, onde o objetivo reside em traduzir especificações funcionais em implementações de circuitos digitais. Vale ressaltar que, embora nem sempre a síntese via álgebra booleana seja a mais prática computacionalmente, ela é crucial pedagogicamente para o entendimento lógico conceitual e auxilia na compreensão de todos os demais aspectos e técnicas empregadas em síntese e minimização de circuitos lógicos.

Para nosso objetivo, vamos abordar o desenvolvimento breve de circuitos combinacionais, ou seja, aqueles que as saídas dependem apenas das entradas a fim de discutir a importância da lógica neste processo.

2.6.1 A Tabela-Verdade como Ferramenta de Análise

Essa abordagem é o primeiro passo na síntese de circuitos combinacionais, facilitando a construção por meio de um guia inicial, onde se projeta resultados esperados a partir de um dado inicial, podendo esse ser a tabela verdade, permitindo validar a coerência entre a formulação teórica e sua implementação eletrônica.

No processo de projeto de circuitos digitais, a literatura indica que o primeiro passo consiste na descrição funcional do sistema desejado, a qual pode ser realizada por meio da construção de uma tabela-verdade [3]. Contudo, o próprio autor ressalta que, à medida que o número de variáveis aumenta, essa forma de representação torna-se impraticável, exigindo o uso de métodos alternativos de análise e simplificação [3].

Em várias situações, usamos outros métodos de simplificação a partir de resultados obtidos pela tabela-verdade, como a simplificação algébrica e simplificações a partir das formas canônicas (Soma de Mintermos e o Produto de Maxtermos), que podem ser trabalhadas por métodos como o Mapa de Karnaugh. Assim, ela constitui o elo entre o raciocínio lógico abstrato e o comportamento operacional dos circuitos digitais.

2.6.2 Formas Canônicas

As formas canônicas são ideais para auxiliar na sintetização de circuitos digitais, pois a partir delas, podemos concluir algumas abreviações nas portas e trabalhar métodos mais práticos de simplificação, como mapa de Karnaugh. A literatura destaca que uma das principais vantagens de expressar uma função booleana em forma canônica está na possibilidade de realizar sua representação numérica, o que facilita a análise, a sistematização e a aplicação de métodos de simplificação [8].

Entende-se que essa correspondência entre mintermos, maxtermos e tabelas-verdade permite uma transição direta entre a descrição funcional e a implementação em hardware, sendo a base para métodos de simplificação como os Mapas de Karnaugh.

Dessa forma, as formas canônicas não apenas garantem uma representação completa da função lógica, mas também servem de ponto de partida para técnicas de simplificação e otimização, conectando de modo direto a lógica proposicional simbólica à realidade física dos circuitos digitais.

2.7 A relevância da lógica formal na síntese de circuitos digitais

A transição do pensamento matemático abstrato para a engenharia de sistemas digitais exige uma ruptura epistemológica: o estudante deve deixar de enxergar a lógica apenas como um exercício de validação de argumentos (V ou F) para compreendê-la como uma ferramenta de modelagem física (0 ou 1). A literatura analisada indica que o conhecimento prévio de Lógica Proposicional e Matemática Discreta atua como um catalisador nesse processo.

Sem essa base, o aprendizado de Circuitos Digitais tende a se limitar à memorização de tabelas-verdade e desenhos de portas lógicas, sem a compreensão da causalidade subjacente. Conforme apontado por [3], a capacidade de abstrair elementos comuns de aplicações para chegar a definições teóricas é crucial. Quando o estudante domina previamente as leis de De Morgan, a distributividade e os conceitos de tautologia, a síntese de um circuito deixa de ser uma tentativa e erro de conexões para se tornar uma dedução lógica estruturada.

2.8 Estudo de Caso 1: O Meio-Somador (Half-Adder)

O meio-somador é o circuito combinacional fundamental para aritmética binária, responsável por somar dois bits isolados sem considerar um *carry* de entrada. Este é um exemplo pedagógico excelente pois é simples o suficiente para análise manual, mas suficientemente rico para demonstrar todos os princípios de síntese.

Passo 1: Especificação funcional;

Entradas: A (primeiro bit), B (segundo bit).

Saídas: S (soma), C0 (carry de saída)

Tabela 4. Tabela verdade do meio-somador.

A	B	S	C0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Fonte: Autoria Própria (2025).

Passo 2: Derivação das funções booleanas usando os mintermos

Para a saída S (soma), a função vale 1 quando A = 0, B = 1 ou quando A = 1, B = 0:

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

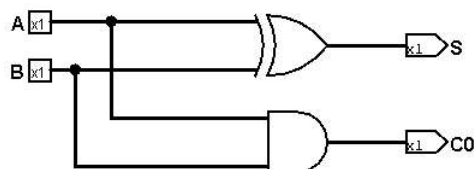
Para a saída C (*carry*), a função vale 1 apenas quando ambas entradas são 1:

$$C0 = AB$$

Passo 3: Síntese do circuito

O circuito do meio-somador pode ser implementado utilizando uma porta XOR para produzir *S* e uma porta AND para produzir *C0*. Esta implementação é ótima, pois realiza ambas as funções com o mínimo de portas possível.

Figura 5 - Meio somador.



Fonte: Autoria Própria (2025).

2.8.1 Significado pedagógico:

O meio-somador exemplifica de forma tangível como operações abstratas (adição binária) são implementadas através de lógica proposicional materializada em circuitos. A operação matemática $1 + 1 = 10$ (que em decimal é 2) é realizada pelo circuito através da aplicação de portas lógicas que seguem rigorosamente as regras da Álgebra Booleana derivadas da Lógica Proposicional. Esta continuidade conceitual (lógica $\rightarrow$ álgebra $\rightarrow$ circuito) é precisamente o que os estudantes devem compreender profundamente e que, com uma fortificação do conhecimento por parte da disciplina de matemática discreta, essa abordagem deixa de ser uma introdução teórica e sim uma revisão para implementar na prática do circuito, facilitando a implementação até nas disciplinas de laboratório.

2.9 Estudo de Caso 2: Somador-Completo (Full-Adder)

A composição do circuito *Full Adder* representa, em si, uma excelente oportunidade para a aplicação dos conhecimentos de álgebra booleana na síntese digital.

Passo 1: Especificação funcional

Entradas: A (primeiro bit), B (segundo bit), Cin (carry de entrada)

Saídas: S (soma), Cout (carry de saída)

Tabela 5. Tabela verdade do somador completo.

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fonte: Autoria Própria (2025).

Passo 2: Derivação das funções booleanas usando os mintermos

Para a saída S (soma), a função vale 1 em quatro casos:

- $A = 0, B = 0$ e $Cin = 1$
- $A = 0, B = 1$ e $Cin = 0$
- $A = 1, B = 0$ e $Cin = 0$
- $A = 1, B = 1$ e $Cin = 1$

$$S = \bar{A}\bar{B}Cin + \bar{A}B\bar{Cin} + A\bar{B}\bar{Cin} + ABCin$$

usando as propriedades booleanas:

$$S = (\bar{A}B + A\bar{B})\bar{Cin} + (AB + \bar{A}\bar{B})Cin = (A \oplus B)\bar{Cin} + (\overline{A \oplus B})Cin = (A \oplus B) \oplus Cin$$

para a saída Cout, a função vale 1 também em quatro casos:

- $A = 0, B = 1$ e $Cin = 1$
- $A = 1, B = 0$ e $Cin = 1$
- $A = 1, B = 1$ e $Cin = 0$
- $A = 1, B = 1$ e $Cin = 1$

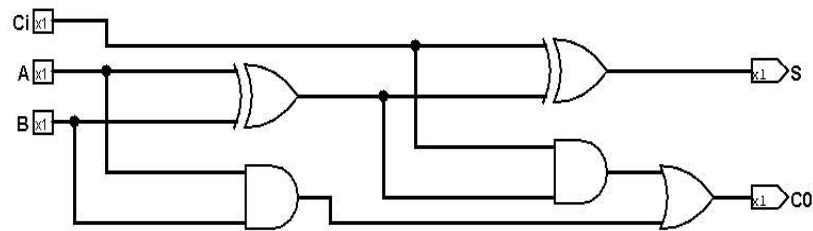
$$Cout = \bar{A}BCin + A\bar{B}Cin + AB\bar{Cin} + ABCin$$

usando as propriedades booleanas:

$$Cout = (\bar{A}B + A\bar{B})Cin + AB(Cin + \bar{Cin}) = (A \oplus B)Cin + AB$$

A figura 6 mostra o diagrama correspondente ao circuito de um somador completo.

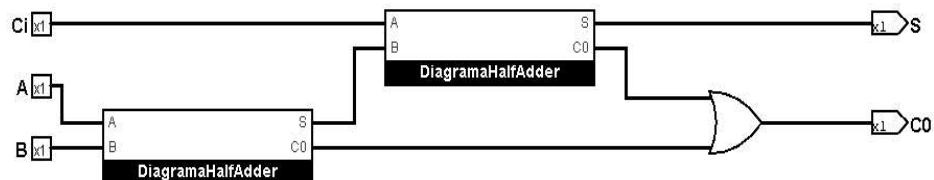
Figura 6 - Somador completo de 1 bit.



Fonte: Autoria Própria (2025).

Ou seja, o *Full Adder* pode ser implementado a partir de dois *Half-Adders* combinados e uma porta lógica OR e isso acontece porque a operação feita por um somador completo pode ser dividida naturalmente em duas partes: primeiro somando dois dos bits, e depois somando o resultado com o terceiro bit.

Figura 7 - Somador completo de um 1 bit (Diagrama de blocos).



Fonte: Autoria Própria (2025).

O mesmo raciocínio se revela essencial ao se trabalhar com outros módulos aritméticos. A manipulação de expressões para diferenciar, inverter sinais e agrupar variáveis parte sempre do conhecimento algébrico prévio, evidenciando que a progressão conceitual, da lógica ao projeto de circuitos, é fundamental para a compreensão e inovação em sistemas digitais.

3 RESULTADOS E DISCUSSÕES:

A análise bibliográfica das várias referências, especialmente [9], [3] e [1] são fundamentais e revelam uma progressão conceitual clara: a Lógica Proposicional serve como fundamento filosófico, a Álgebra Booleana como formalização matemática, e os Circuitos Digitais como materialização tecnológica.

A transição do pensamento matemático abstrato para a engenharia de sistemas digitais exige uma ruptura epistemológica: o estudante deve deixar de enxergar a lógica apenas como um exercício de validação de argumentos (V ou F) para compreendê-la como uma ferramenta de modelagem física (0 ou 1).

Sem essa base, o aprendizado de Circuitos Digitais tende a se limitar à memorização de tabelas-verdade e desenhos de portas lógicas, sem a compreensão da causalidade subjacente.

A análise bibliográfica conduzida revela uma progressão conceitual clara: a Lógica Proposicional serve como fundamento filosófico, a Álgebra Booleana como formalização matemática, e os Circuitos Digitais como materialização tecnológica. Esta estrutura não é meramente organizacional, mas reflete o próprio processo histórico de desenvolvimento da área.

A revisão evidencia a correspondência estrutural entre operadores lógicos e portas lógicas, conforme indicado nas referências analisadas. A álgebra dos circuitos se encaixa perfeitamente no quadro geral da álgebra booleana, sendo idêntica à álgebra das proposições quando considerada como sistema abstrato. Os conectivos lógicos (negação, conjunção e disjunção) encontram materialização física direta nas portas NOT, AND e OR, respectivamente. Esta correspondência não é coincidência, mas consequência necessária do fato de que ambos os sistemas operam sobre o mesmo domínio bivalente: verdadeiro/falso na lógica e 1/0 nos circuitos.

Os estudos de caso desenvolvidos ilustram como essa progressão se manifesta na prática. No meio-somador, a função de soma emerge diretamente da análise da tabela-verdade para adição binária, exemplificando como a lógica proposicional se traduz em expressão booleana e, finalmente, em implementação física. O somador completo revela complexidade adicional: a derivação algébrica do carry de saída requer domínio das leis de complemento, elemento neutro e absorção, todas derivadas dos axiomas fundamentais da álgebra booleana. Um estudante desconhecedor destas propriedades, teria dificuldade de enxergar o valor intrínseco que essas expressões representam e mesmo por mapa de Karnaugh, o entendimento seria incompleto.

A revisão identificou uma lacuna pedagógica recorrente: a desconexão entre o ensino elaborado de lógica matemática e a disciplina de circuitos digitais. Esta fragmentação prejudica diretamente o processo de aprendizagem. Quando o estudante não compreende *por que* as regras da álgebra booleana funcionam, ou seja, quando não percebe sua derivação da lógica proposicional, sua capacidade de aplicá-las criativamente fica comprometida.

A literatura sugere que o conhecimento prévio de lógica proposicional e matemática discreta atua como catalisador cognitivo: quando o estudante domina previamente as leis de De Morgan, a distributividade e os conceitos de tautologia, a síntese de um circuito deixa de ser tentativa e erro para se tornar dedução lógica estruturada.

A transição do pensamento matemático abstrato para a engenharia de sistemas digitais exige uma ruptura epistemológica: o estudante deve deixar de enxergar a lógica apenas como exercício de validação de argumentos para compreendê-la como ferramenta de modelagem física. A universalidade da álgebra booleana, do chaveamento elétrico aos processadores modernos, fundamenta-se precisamente na identidade com a lógica bivalente clássica.

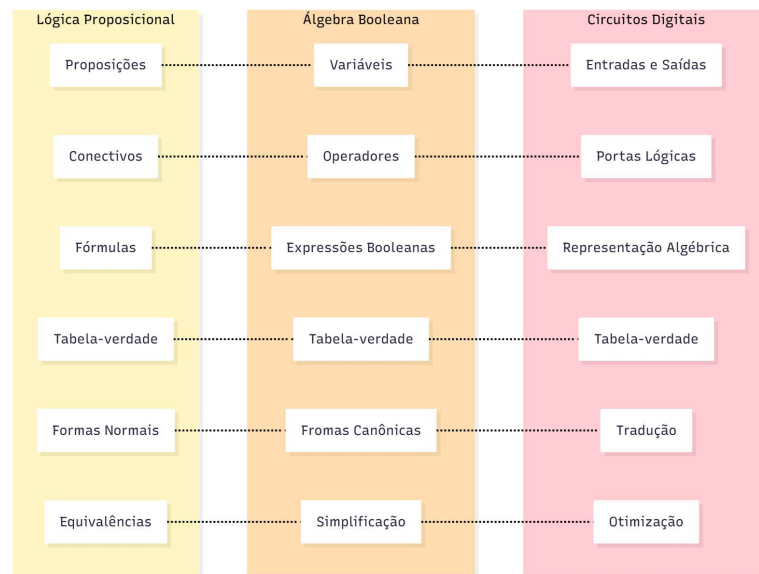
Os achados sugerem fortemente que uma abordagem pedagógica integrada, na qual disciplinas de lógica sejam desenvolvidas em conjunto ou como um conhecimento prévio para circuitos digitais, otimiza o aprendizado e fortalece a capacidade de abstração. A disciplina de matemática discreta, quando posicionada adequadamente no currículo, serve como ponte conceitual entre o raciocínio matemático formal e a aplicação tecnológica. Os conceitos de proposição, conectivos, tabelas-verdade, tautologias e equivalências lógicas são precisamente os pré-requisitos cognitivos para compreensão profunda dos circuitos digitais, mesmo que em uma cadeira como esta, seja demonstrado de forma introdutória, pedagogicamente não faz sentido não introduzir previamente ou em conjunto esses conceitos, visto que há disciplinas possíveis de cumprir esse papel.

Conclui-se que a articulação didática entre lógica e circuitos potencializa a autonomia intelectual e a aplicabilidade das competências de engenharia. Os desafios recorrentes no ensino tradicional, despreparo metodológico, ausência de interdisciplinaridade e limitações curriculares, podem ser mitigados através de abordagens integradas que fortaleçam a progressão lógica-algébrica-física na formação tecnológica. Uma abordagem pedagógica que desenvolve disciplinas de lógica juntamente com circuitos digitais permite ao estudante ir além da simples memorização de regras da álgebra booleana, alcançando compreensão genuína que o capacita para inovação e resolução criativa de problemas em sistemas digitais.

Para facilitar a compreensão da articulação entre lógica formal, álgebra booleana e a síntese de circuitos digitais, eixo central deste trabalho, foi elaborado um mapa conceitual de autoria própria que ilustra graficamente as três principais etapas da fundamentação teórica e metodológica. O diagrama destaca como os conceitos iniciais da lógica formal evoluem para a álgebra booleana, e posteriormente para a formalização matemática, evidenciando a correspondência direta desses pilares com os procedimentos práticos de representação simbólica, implementação física de portas lógicas e sintetização de circuitos digitais. Essa visualização reforça a importância da progressão lógica-algébrica-física como suporte para o desenvolvimento e compreensão dos sistemas digitais contemporâneos. Logo, entende-se que as constatações não apenas sustentam teoricamente a abordagem aqui adotada, como também corroboram, na literatura, a pertinência da integração entre lógica, álgebra booleana e circuitos no processo de

ensino-aprendizagem. A convergência entre as teorias observadas reforça a necessidade de conteúdos de lógica proposicional, álgebra booleana e circuitos digitais trabalhados de forma articulada e progressiva, garantindo que o estudante compreenda como cada nível conceitual se apoia no anterior. A articulação didática entre lógica e circuitos potencializa autonomia intelectual e aplicabilidade das competências de engenharia, mitigando desafios do ensino tradicional por meio de abordagens integradas.

Figura 8 – Mapa conceitual da progressão lógica-algébrica-física na síntese de circuitos digitais.



Fonte: Autoria Própria (2025).

3.1 TRABALHOS FUTUROS

Geralmente, disciplinas como Lógica ou Matemática Discreta não são pré-requisitos para Circuitos Digitais. Por esse motivo, muitos alunos cursam esta disciplina antes daquelas como forma de adiantar a conclusão do curso superior. Sabe-se que há uma maior dificuldade em compreender os conceitos e aplicá-los quando não se tem uma base sólida que fundamenta os Circuitos Digitais. Portanto, como trabalho futuro, pode-se comprovar essa observação direta por meio de uma pesquisa entre alunos que cursam e os que não cursaram disciplinas teóricas antes de Circuitos, a fim de se obter uma correlação.

Um trabalho futuro essencial também seria desenvolver e avaliar metodologias para mitigar a lacuna pedagógica entre Lógica e Circuitos. Outro caminho seria analisar e propor o posicionamento adequado de Matemática Discreta no currículo para que ela sirva efetivamente como catalisador cognitivo.

5 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] FLOYD, Thomas L. Sistemas digitais: fundamentos e aplicações. Tradução José Lucimar do Nascimento. Revisão técnica Antonio Pertence Júnior. 9. ed. Porto Alegre: Bookman, 2007. 888 p. ISBN 978-85-7780-107-7.
- [2] WHITESITT, J. Eldon. Boolean algebra and its applications. Reading: Addison-Wesley Publishing Company, 1961. 210 p. (Tradução pessoal).
- [3] HIRATA, Nina S. T. **Álgebra Booleana e Aplicações**. São Paulo: DCC/IME-USP, 2006. Notas de Aula .
- [4] BERTOLINI, Cristiano; CUNHA, Guilherme Bernardino da; FORTES, Patricia Rodrigues. **Lógica Matemática**. 1. ed. Santa Maria: UAB/NTE/UFSM, 2017.
- [5] SANTOS, Mauro Roberto Rosa dos. **O Uso da Lógica Booleana Matemática no Ensino Básico**. 2024. 91 p. Dissertação (Mestrado Profissional em Matemática) - Universidade Federal do Maranhão, São Luís, 2024.
- [6] TEIXEIRA, Hugo Tanzarella; TAVARES, Marley Fagundes; PEREIRA, Rodrigo Vinícius Mendonça. Sistemas digitais. Londrina: Editora e Distribuidora Educacional S.A., 2017.
- [7] TREBIEN, Heitor A. C. George Boole e sua importância para a inteligência artificial e a programação. 2022. Disponível em:
<<https://velip.com.br/blog/george-boole-e-sua-importancia-para-a-inteligencia-artificial-e-a-programacao/>>. Acesso em: 18 nov. 2025.
- [8] NOGUEIRA, Jurandyr S. **Eletrônica Digital Básica**. Salvador: EDUFBA, 2011 .
- [9] VEGA, Alexandre Santos de la. **Apostila de Teoria para Circuitos Digitais**. Niterói: UFF/TCE/TET, 2025. Apostila .
- [10] CARDOSO, Domingos Moreira; SZYMAŃSKI, Jerzy; ROSTAMI, Mohammad. Matemática discreta: combinatória, teoria dos grafos e algoritmos.
- [11] ALDEON ARAGÓN, Rolen; ROJAS TRAVEZAÑO, Dimar Darwin. Influencia del Álgebra Booleana en el aprendizaje de los circuitos electrónicos de los estudiantes del quinto grado de educación secundaria de la Institución Educativa “Augusto Salazar Bondy” - Ninacaca - 2017. 2021. Tesis (Licenciatura en Educación, Matemática – Física) – Universidad Nacional Daniel Alcides Carrión, Cerro de Pasco.
- [12] J. L. Gersting, *Fundamentos Matemáticos para a Ciência da Computação*, 7ª ed. Rio de Janeiro: LTC, 2017.
- [13] FLORES, Paulo. Utilização de diagramas de decisão binária (BDDs) para representar máquinas de estado (FSMs) e verificar fórmulas de lógica temporal (CTL). [S.l.]: INESC/IST, 1996. Relatório técnico.
- [14] MANO, M. Morris. Digital logic and computer design. 2. ed. Englewood Cliffs: Prentice-Hall, 1979.