

**CIRCUITOS DIGITAIS COMBINACIONAIS NA
LÓGICA DE MÚLTIPLOS VALORES**

HERBERT LUQUE PERALTA

CAMPO GRANDE - MS

2008

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA

Circuitos Digitais Combinacionais na Lógica de Múltiplos
Valores

Herbert Luque Peralta

Orientador: *Prof. Milton Ernesto Romero Romero - Dr.*

Dissertação apresentada ao Departamento de Engenharia Elétrica da Universidade Federal de Mato Grosso do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

UFMS - Campo Grande
Dezembro / 2008

Circuitos Digitais Combinacionais na Lógica de Múltiplos Valores

Herbert Luque Peralta

‘Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em Energia Elétrica e Linha de Pesquisa Inteligência Artificial em que foi realizado o trabalho, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Mato Grosso do Sul.’

Milton Ernesto Romero Romero - Dr.
Prof. DEL/UFMS - Orientador

Luciana Cambraia Leite - Dr.
Coordenadora do Programa de Pós-Graduação
em Engenharia Elétrica

Banca Examinadora:

Milton Ernesto Romero Romero - Dr.
Prof. DEL/UFMS - Presidente

Ricardo Ribeiro dos Santos - Dr.
Prof. CCET/UCDB

Evandro Mazina Martins - Dr.
Prof. DEL/UFMS

*A DEUS,
pelas oportunidades e
dádivas em minha vida.
A minha família.
Com carinho especial,
aos meus pais Felipe e Regina
e aos meus irmãos Hubert e Sonia.*

Agradecimentos

A Deus, por acompanhar meus passos e colocar as pessoas certas em meu caminho.

Ao Prof. Dr. Milton Ernesto Romero Romero, pelo entusiasmo na condução desta pesquisa. Sou-lhe grato não só pela transmissão do conhecimento científico, como também pela oportunidade de tê-lo como exemplo profissional íntegro.

Ao Prof. Dr. Evandro Mazina Martins, pela participação fundamental neste trabalho, que transmitiu conhecimentos tão importantes durante o mestrado especialmente na área de MVL.

Aos professores da banca examinadora especialmente ao Prof. Dr. Ricardo Ribeiro dos Santos pelas importantes contribuições que melhoraram a elaboração deste trabalho.

Aos professores do Mestrado em Engenharia Elétrica que contribuíram na minha formação especialmente ao Prof. Dr. João Onofre Pinto e a Profa. Dra. Luciana Cambraia Leite, pela colaboração muito valiosa. E aos funcionários da pós-graduação, especialmente a Marcira.

Ao Prof. Dr(c). Marco Alvarez, responsável pelo meu ingresso na pesquisa científica, pelo exemplo de simplicidade. Sinto-me orgulhoso de contar com sua amizade.

Aos meus pais, Felipe e Regina, aos meus irmãos, Hubert e Sonia, e aos meus tios Jaime, Raúl, Walter e Rolando.

Também não poderia deixar de agradecer a vários amigos, tanto da vida quanto da universidade: Profa. Dra. Kathya Silvia Collazos Linares, Prof. Dr. Jorge Luis Roel Ortiz, M.Sc. Wellington Rocha Araújo, Meliton Apaza Tito, M.Sc. Graciela Lecireth Meza Lovon, M.Sc. José Edison Cabral Junior, M.Sc. Lindinei Santiago Santana, Edvaldo Francisco Freitas Lima, Rafael Nishimura, Márcio Lorenzoni Portella, Edgard Jose Dos Santos Arinos, Jose Elias Montalvan Barbaran, Marcelo Maldonado Corrêa e a todos meus companheiros de mestrado que caminharam juntos nessa jornada.

Finalmente, agradeço ao CNPq pelo apoio financeiro¹.

¹O presente trabalho foi realizado com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq - Brasil.

*"É verdade que não podemos encontrar a pedra filosofal,
mas é bom que ela seja procurada. Procurando-a,
encontramos muitos segredos que não procurávamos".*

(Bernard le Bovier de Fontenelle, 1657- 1757)

Resumo da Dissertação apresentada à UFMS como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

CIRCUITOS DIGITAIS COMBINACIONAIS NA LÓGICA DE MÚLTIPLOS VALORES

Herbert Luque Peralta
Dezembro / 2008

Orientador: Prof. Milton Ernesto Romero Romero, Ph.D.

Área de Concentração: Energia Elétrica (Inteligência Artificial).

Palavras-chave: Álgebra de Múltiplos Valores, Síntese de Circuitos MVL, Simplificação de Funções MVL.

Número de Páginas: 73.

Os circuitos digitais combinacionais são projetados na lógica de dois valores conhecida como Álgebra de Chaveamento, e dependendo da complexidade apresentam limitações, sendo algumas delas o consumo de energia e o tamanho dos circuitos gerado pelo número de trilhas, uma das alternativas para solucionar este problema é o uso da Lógica Multinível também conhecida como Lógica de Múltiplos Valores (MVL - *Multi-valued logic*). Na atualidade, existem diferentes Álgebras para Múltiplos Valores (MV - *Multi-Valued*), algumas delas apresentam completeza funcional, mas na maioria a síntese e a simplificação de circuitos lógicos são complexos, gerando isto longas funções MVL.

O propósito desta pesquisa é propor uma Álgebra de MV com funcionalidade completa, que permita realizar a análise, a síntese e a simplificação de funções MVL dos circuitos digitais combinacionais. A Álgebra de MV proposta apresenta dois conjuntos universais de operadores lógicos, e permite a síntese de funções MVL na forma de Soma de Operações Produto Estendido (SOPE) e Produto de Operações Soma Estendida (POSE) de maneira análoga as formas SOP e POS da álgebra de Chaveamento. Nesta pesquisa também propõe-se métodos de simplificação de funções MVL e a extensão dos Mapas de Karnaugh, Método Quine McCluskey e Algoritmos Petrick para MVL.

Para mostrar a funcionalidade da álgebra proposta projeta-se alguns circuitos lógicos com simulações em VHDL. Finalmente, são expostas algumas considerações sobre as idéias apresentadas nos capítulos componentes desta dissertação, além de serem apresentadas as conclusões provenientes desta pesquisa e alguns trabalhos futuros propostos a partir desta dissertação.

Abstract of Dissertation presented to UFMS as a partial fulfillment of the requirements for the degree of
Master in Electrical Engineering.

COMBINATIONAL DIGITAL CIRCUIT IN MULTI-VALUED LOGIC

Herbert Luque Peralta

December / 2008

Advisor: Milton Ernesto Romero Romero, Ph.D.

Area of Concentration: Electrical Energy (Artificial Intelligence).

Keywords: Multivalued Algebra, Synthesis of Multivalued Circuit, Simplification of Multivalued Functions.

Number of Pages: 73.

The combinational digital circuits are designed in binary logic known as Switching Algebra, and depending of the complexity have limitations, some of them are consumption of energy and the size of the circuits generated by the number of interconnections, one alternative to solve this problem is the use of Multilevel Logic also known as Multi-valued Logic (MVL). Currently, there are different Multi-Valued Algebras, some of them are completeness, but in most of them the synthesis and simplification of logic circuits are complex, generating long Multi-valued functions.

The purpose of this research is to propose a Multi-valued Algebra with complete functionality, allowing conducting the analysis, synthesis and the simplification of Multi-valued functions of combinational digital circuits. The Multi-valued Algebra proposal presents two universal sets, logical operators, and enables the synthesis of Multi-valued circuit in the form of Sum of Product Extended operations (SOPE - *Soma de Operações Produto Estendido*) and Product of Sum Extended operations (POSE - *Produto de Operações Soma Estendida*) in a similar way of SOP and POS of Switching Algebra. This research also proposes some methods of simplification of MVL functions and extension of Karnaugh maps, Quine McCluskey Method and Petrick's algorithms to MVL.

To show the functionality of the proposed algebra, some logic circuits has been designed with simulations in VHDL. Finally, some considerations about the ideas presented in chapters components of this paper are exposed, in addition some future works from this paper are presented.

Sumário

1	Introdução	1
1.1	Objetivo	4
1.2	Organização da Dissertação	4
2	Álgebra de MV	5
2.1	Introdução	5
2.2	Literal MVL	5
2.3	Operadores	6
2.3.1	Operador Sucessor	6
2.3.2	Operador Máximo	7
2.3.3	Operador Produto Estendido	7
2.3.4	Operador Mínimo	8
2.3.5	Operador Soma Estendida	9
2.3.6	Precedência de Operadores	10
2.4	Axiomas	11
2.4.1	Identidade	11
2.4.2	Elemento Nulo	11
2.4.3	Idempotência	11
2.4.4	Comutatividade	11
2.4.5	Associatividade	11
2.4.6	Complemento	12
2.4.7	Redução	12
2.4.8	Unicidade	12
2.4.9	Involução	12
2.5	Conjuntos Funcionalmente Completos	12
2.6	Dualidade	13
2.7	Teoremas	13
2.7.1	Expansão	13
2.8	Funções MVL	15
2.8.1	Mintermos e Maxtermos	15
2.8.2	Formas Algébricas de Funções MVL	16
2.8.3	Operador Sucessor em Funções MVL	17
2.9	Lei de De Morgan Estendida	19
2.10	Portas Lógicas MVL	19
2.10.1	Simulações das Portas Lógicas MVL	19

3	Circuitos Digitais Combinacionais MVL	21
3.1	Análise de Circuitos Digitais Combinacionais MVL	21
3.1.1	Análise com a Tabela Verdade	21
3.1.2	Análise com a Função MVL	22
3.1.3	Análise com o Diagrama de Tempo	23
3.2	Síntese de Circuitos Digitais Combinacionais MVL	23
3.2.1	Síntese de Circuitos Lógicos na Forma SOPE	23
3.2.2	Síntese de Circuitos Lógicos na Forma POSE	26
4	Simplificação de Funções MVL	31
4.1	Simplificação Algébrica MVL	31
4.2	Minimização por Aplicação do Operador Sucessor	33
4.3	Mapas de Karnaugh Estendido	35
4.4	Quine McCluskey Estendido	40
4.5	Algoritmo Petrick Estendido	47
4.6	Métodos de Minimização na Forma POSE e SOPE	50
4.6.1	Minimização Simétrica	50
4.6.2	Método Janela	53
5	Aplicações MVL	54
5.1	Decodificador	54
5.1.1	Síntese do Decodificador MVL	55
5.1.2	Simulação do Decodificador MVL	56
5.1.3	Síntese do Decodificador Booleano	57
5.1.4	Comparação de Resultados	57
5.2	Somador Completo	58
5.2.1	Síntese do Somador Completo MVL	58
5.2.2	Simulação do Somador Completo MVL	60
5.2.3	Síntese do Somador Completo Booleano	62
5.2.4	Comparação de Resultados	62
5.3	Multiplexer	63
5.3.1	Síntese do Multiplexer MVL	63
5.3.2	Simulação do Multiplexer MVL	64
5.3.3	Síntese do Multiplexer Booleano	65
5.3.4	Comparação de Resultados	65
5.4	Discussão das Comparações	66
6	Considerações Finais	68
6.1	Conclusões	68
6.2	Trabalhos Futuros	69
	Referências	73

Lista de Figuras

2.1	Representação das portas lógicas MVL.	20
2.2	Simulação dos operadores MVL de $\mathbf{N} = 4$	20
3.1	Análise de circuitos por tabelas verdade.	22
3.2	Análise de circuitos por funções MVL.	22
3.3	Análise de circuitos por diagramas de tempo.	23
3.4	Circuito lógico do meio somador MVL de $\mathbf{N} = 4$	26
3.5	Simulação do meio somador MVL de $\mathbf{N} = 4$	27
3.6	Circuito lógico do comparador de duas entradas e uma saída de $\mathbf{N} = 4$	30
3.7	Simulação do comparador de duas entradas e uma saída de $\mathbf{N} = 4$	30
4.1	Separação da Tabela 4.8 em sub-tabelas para cada subfunção.	38
4.2	Agrupamento dos mintermos em regiões retangulares.	39
4.3	Tabela de implicantes primos de $F_1(A, B, C)$	45
4.4	Tabela de implicantes primos de $F_2(A, B, C)$	45
4.5	Tabela de implicantes primos de $F_3(A, B, C)$	47
4.6	Tabela de implicantes primos de $F_2(A, B, C)$	49
4.7	Circuito minimizado do meio somador de $\mathbf{N} = 4$	52
4.8	Simulação do meio somador MVL de $\mathbf{N} = 4$	52
5.1	Decodificadores: MVL de $\mathbf{N} = 4$ e seu equivalente na Álgebra de Chaveamento.	55
5.2	Circuito lógico do decodificador MVL de $\mathbf{N} = 4$	56
5.3	Simulação do decodificador MVL de $\mathbf{N} = 4$	56
5.4	Circuito lógico do decodificador Booleano equivalente ao circuito MVL mostrado na Figura 5.2.	58
5.5	Somador completo: MVL de $\mathbf{N} = 4$, e seu equivalente na Álgebra de Chaveamento.	59
5.6	Circuito lógico do somador completo MVL de $\mathbf{N} = 4$ baseado no meio somador.	60
5.7	Circuito lógico do somador completo MVL de $\mathbf{N} = 4$	61
5.8	Simulação do somador MVL de $\mathbf{N} = 4$	61
5.9	Circuito lógico do somador completo Booleano.	62
5.10	Multiplexer: MVL de $\mathbf{N} = 4$ e seu equivalente na Álgebra de Chaveamento.	63
5.11	Circuito lógico do multiplexer MVL de $\mathbf{N} = 4$	64
5.12	Simulação do multiplexer MVL de $\mathbf{N} = 4$	65
5.13	Circuito lógico do multiplexer Booleano equivalente ao circuito MVL mostrado na Figura 5.11.	66

5.14 Quadro resumo das comparações. 67

Lista de Tabelas

2.1	Tabela verdade do operador MVL <i>Sucessor</i> de $\mathbf{N} = 4$	6
2.2	Tabela verdade do operador MVL <i>Sucessor</i> de $\mathbf{N} = 2$	6
2.3	Tabela verdade do operador MVL <i>Máximo</i> de $\mathbf{N} = 4$	7
2.4	Tabela verdade do operador MVL <i>Máximo</i> de $\mathbf{N} = 2$	7
2.5	Tabela verdade dos operadores MVL <i>Produto Estendido</i> de $\mathbf{N} = 4$	8
2.6	Tabela verdade do operador MVL <i>Produto Estendido</i> de $\mathbf{N} = 2$	8
2.7	Tabela verdade do operador MVL <i>Mínimo</i> de $\mathbf{N} = 4$	9
2.8	Tabela verdade do operador MVL <i>Mínimo</i> de $\mathbf{N} = 2$	9
2.9	Tabela verdade dos operadores MVL <i>Soma Estendida</i> de $\mathbf{N} = 4$	9
2.10	Tabela verdade do operador MVL <i>Soma Estendida</i> de $\mathbf{N} = 2$	10
2.11	Exemplo do operador <i>Sucessor</i> em funções MVL.	17
3.1	Exemplo de análise de circuitos usando a tabela verdade.	22
3.2	Exemplo de síntese na forma SOPE.	24
3.3	Tabela verdade do meio-somador de $\mathbf{N} = 4$	25
3.4	Exemplo de síntese na forma POSE.	27
3.5	Comparador de duas entradas e uma saída de $\mathbf{N} = 4$	29
4.1	Exemplo 1 de minimização por aplicação do operador <i>Sucessor</i>	33
4.2	Três aplicações do operador <i>Sucessor</i> em $F(A, B)$ da Tabela 4.1.	34
4.3	Exemplo 2 de minimização por aplicação do operador <i>Sucessor</i>	34
4.4	Três aplicações do operador <i>Sucessor</i> em $F(A, B)$ da Tabela 4.3.	35
4.5	Mapa Karnaugh Estendido para 2 literais de $\mathbf{N} = 4$	35
4.6	Mapa Karnaugh Estendido para 3 literais de $\mathbf{N} = 4$	36
4.7	Simplificação da função lógica da Tabela 4.2 utilizando mapas de Karnaugh Estendido.	37
4.8	Exemplo de simplificação utilizando mapas de Karnaugh Estendido, $\mathbf{N} = 4$	38
4.9	Exemplo do método Quine McCluskey Estendido.	42
4.10	Exemplo do método Quine McCluskey Estendido: Separação da Tabela 4.9 em sub-tabelas.	42
4.11	Procedimento do método Quine McCluskey Estendido para $F_1(A, B, C)$ da Tabela 4.10.	43
4.12	Passo 5 do método Quine McCluskey Estendido.	44
4.13	Passo 6 do método Quine McCluskey Estendido.	44

4.14	Procedimento do método Quine McCluskey Estendido para $F_2(A, B, C)$ da Tabela 4.10.	46
4.15	Procedimento do método Quine McCluskey Estendido para $F_3(A, B, C)$ da Tabela 4.10.	47
4.16	Exemplo do método Petrick Estendido.	48
4.17	Exemplo de minimização simétrica.	51
4.18	Tabela do meio somador MVL de $\mathbf{N} = 4$	51
4.19	Tabela Simplificada do meio somador MVL de $\mathbf{N} = 4$	51
4.20	Exemplo 1 de simplificação pelo método janela de $\mathbf{N} = 4$	53
4.21	Exemplo 2 de simplificação pelo método janela de $\mathbf{N} = 4$	53
5.1	Tabela verdade dos decodificadores MVL de $\mathbf{N} = 4$ e seu equivalente na Álgebra de Chaveamento.	55
5.2	Comparação dos decodificadores MVL de $\mathbf{N} = 4$ e seu equivalente na Álgebra de Chaveamento.	58
5.3	Tabela verdade do somador completo MVL de $\mathbf{N} = 4$, e seu equivalente na Álgebra de Chaveamento.	59
5.4	Comparação dos somadores MVL de $\mathbf{N} = 4$ e seu equivalente na Álgebra de Chaveamento.	63
5.5	Tabela verdade dos multiplexer MVL de $\mathbf{N} = 4$, e seu equivalente na Álgebra de Chaveamento.	64
5.6	Comparação dos multiplexer MVL de $\mathbf{N} = 4$ e seu equivalente na Álgebra de Chaveamento.	66

Introdução

Os circuitos digitais combinacionais são sintetizados na lógica de dois valores, conhecida como Álgebra de Chaveamento [1], e dependo da complexidade apresentam limitações, sendo algumas delas o consumo de energia e o tamanho dos circuitos gerado pelo número de trilhas. Uma das alternativas de solução ao problema é o uso da Lógica de Múltiplos Valores (MVL *Multi-valued logic*) também conhecida como Lógica Multi-nível (*Multi-level Logic*) ou Lógica de Muitos Valores (*Many-valued logic*).

As pesquisas em MVL começaram no ano de 1917 com Łukasiewicz [2], quem desenvolveu o cálculo proposicional de três valores [3], seguido por Post (1920) [4], Bernstein (1924), Webb (1935) e outros [5]. Na atualidade existem diferentes MVLs, baseadas nas Álgebras de Łukasiewicz [2], Post [4, 6] e Kleene [7].

Os estudos em MVL para projetar circuitos digitais tiveram início nos anos 50 com o objetivo de minimizar trilhas e conexões em sistemas digitais para poupar a área de silício, mas as lógicas conhecidas nessa data eram complexas e contrariamente ao esperado os circuitos MVL eram maiores. Isto levou a necessidade de uma MVL apropriada com procedimentos simples na síntese e com métodos de simplificação. Desde então começaram as pesquisas nas áreas de:

- a. Álgebras de Múltiplos Valores.
- b. Síntese de funções MVL.
- c. Circuitos e semicondutores de múltiplos valores.

Álgebras de Múltiplos Valores

Atualmente existem diversas pesquisas na área da Álgebra de MV como as de: Mou Hu [8] que apresenta um álgebra para o modelamento de circuitos MOS VLSI ao nível de chaveamento, sendo uma nova ferramenta para a análise e a síntese de circuitos MOS VLSI. Tatsuki [9] propõe o método de controle de carga chamado MVL CMOS do tipo estático e uma álgebra baseada nos operadores MAX e MIN, mostrando que os circuitos para conversão lógica de valores são realizados usando complicadas configurações dos circuitos. Noboru [10] propõe a Álgebra Kleene-Stone posto que a Álgebra de Kleene tem correspondência com os conjuntos Fuzzy ou lógica Fuzzy sendo sistemas ambíguos e a Álgebra de Stone tem conexões de modalidade, conseguindo satisfazer a monotonicidade para a relação de ordem parcial. Corina Reischer [11] apresenta a generalização da Álgebra de Post inspirada nas lattices distributivas com uma generalização do operador negação. Zhang [12], estabelece um sistema algébrico equivalente ao sistema de cálculo proposicional, o sistema que propõe é chamado de meia álgebra, pelo fato de ser uma derivação da Álgebra Booleana e manter similaridade. Hurst [13] apresenta um álgebra utilizando os operadores adição e máximo. Atul [14] propõe um conjunto de operadores literal, ciclo, complemento de literal, complemento de ciclo, mínimo e tsum, conseguindo sintetizar circuitos MVL. Outras metodologias propõem diferentes propriedades e operadores com o objetivo principal de melhorar os tempos de retardo dos sistemas e reduzir o número de conexões [15], [16].

Síntese de funções MVL

As pesquisas na área de síntese recebem consideração especial nos anos de 1986 com os trabalhos de Allan e Givone e de Hurst que mostraram a simplificação de funções ternárias, sendo os métodos de simplificação MVL mais complexos quanto aos conhecidos da Álgebra de Chaveamento [5]. Assim, além da síntese começa a pesquisa na simplificação de funções MVL, algumas delas são apresentadas a seguir: Mahmood e George Epstein [17] apresentam os Mapas de Karnaugh para três valores, utilizando os operadores disjunção, conjunção, mínimo e máximo, assim como a soma de operações produto, sendo útil até quatro variáveis. Tsutomu [18] apresenta um algoritmo de simplificação de funções ESOP (*Exclusive-Or Sum-of-Products*), o algoritmo que apresenta é baseado numa melhora iterativa com sete regras usadas para trocar um par de produtos por outro. Dueck [19] apresenta o algoritmo heurístico de minimização baseado na aplicação da operação coseno, permitindo a rápida detecção dos implicantes primos essencial e pseudo-essencial. Konrad [20] apresenta um método para obter funções de uma variável com baixo custo de realização, baseado no incremento do custo, se mostrando a síntese de circuitos no modo de corrente CMOS. Em [21] apresenta-se a função de síntese usando MOS no modo corrente usando transistores para a realização de funções unárias, propondo um método de decomposição em funções unárias.

Circuitos e semicondutores de múltiplos valores

Os primeiros trabalhos na área de semicondutores de múltiplos valores foram os trabalhos de Henle (1955), Muhldorf (1958), Hallworth e Heath (1962). É claro que as pesquisas naquela data foram irrelevantes comparadas as tecnologias atuais. No período de 1968 a indústria de microeletrônica fez tentativas para desenvolver eficientes circuitos lógicos de MV utilizando conceitos baseados em: Tecnologia Bipolar ECL, tecnologia I²L, tecnologia nMAS e a tecnologia CMOS [5]. Xunwei Wu [22] estende a teoria de função de transmissão dos circuitos ternários CMOS aos quaternários CMOS e obteve baixa dissipação e baixa saída de impedância, sendo uma técnica apropriada para circuitos VLSI. Takahiro Hanyu et al [23] apresenta o projeto de um processador paralelo de tempo real com técnicas de inteligência artificial, conseguindo demonstrar que na lógica quaternária o número de células de memória, interconexão de células e transistores podem ser consideravelmente reduzidos em comparação ao equivalente implementado na Álgebra de Chaveamento. Michitaka Kameyama et al [24] propõem a redução de circuitos aritméticos baseado na MVL bidirecional na tecnologia de modo corrente MOS e consegue construir um somador múltiplo projetado na tecnologia CMOS com μm . Em [25] são pesquisadas as estruturas PLA para a implementação em MVL usando tecnologia CCD, conseguindo uma estrutura completamente programável no nível usuário. Em [26] a possibilidade de empregar a tecnologia *Switched-capacitor* (SD) para MVL é pesquisada e discutida as vantagens, também é projetado um multiplicador com a tecnologia proposta. Em [27] é apresentado um circuito VLSI de alta densidade orientado à associação de células de memória em tempo real para operações numéricas e não numéricas, demonstrando que o número de transistores, células e interconexões entre células em r-valued CAM são reduzidas a mais de $1/\log_2 r$ em comparação ao correspondente binário. Em [24] é proposto um módulo composto por um somador, um multiplicador parcial e um gerador de dígito cociente implementado em MVL no modo de corrente bidirecional, as simulações mostraram que a velocidade do módulo proposto como um multiplicador é comparável com a rapidez do binário. Em [28] utilizou-se os operadores mínimo, máximo, inversor, literal e ciclo com os quais foi projetado um somador completo em representação digital de base 8. Em [16] implementou-se um codificador e um decodificador usando MOSFETs de alta qualidade e baixo consumo de energia e apresenta os operadores, inversor unário e conversor de nível. Em [29] apresentou-se a decomposição de funções MVL numa álgebra com os operadores complemento, máximo, mínimo, ciclo, e entrada comutativa. Em [30] procurou-se a otimização das funções lógicas de múltiplos valores, além de propor a aplicação em software embarcados. Kinvi [31] encaixa-se na concepção de uma nova arquitetura com o objetivo de validar os circuitos ternários construídos.

Nesta pesquisa propõe-se uma álgebra que atua num domínio \mathbf{D} e é composta de cinco operadores; *Sucessor*, *Máximo*, *Produto Estendido*, *Mínimo* e *Soma Estendida*, os três primeiros

formam um conjunto universal de operadores¹ [33], [34]. E os dois últimos operadores com o operador *Sucessor* fazem parte de um segundo conjunto universal de portas lógicas, sendo o primeiro conjunto de operadores o dual do segundo conjunto de operadores. O primeiro conjunto de operadores permite a representação de funções na forma padrão *Soma de Operações Produto Estendido* (SOPE) baseada em mintermos da função e o segundo conjunto na forma padrão *Produto de Operações Soma Estendida* (POSE) baseada em maxtermos da função.

A partir das formas SOPE e POSE é possível simplificar a função lógica utilizando propriedades da álgebra ou métodos estendidos da álgebra de chaveamento tais como mapas de Karnaugh, Quine McCluskey, Algoritmos Petrick e outros. Para demonstrar a correta funcionalidade da álgebra proposta foram realizadas simulações dos operadores e dos circuitos em VHDL e também foram projetados alguns circuitos lógicos digitais.

1.1 Objetivo

O objetivo desta pesquisa é propor uma álgebra que atue num domínio D de dois ou mais valores (Múltiplos valores) de forma que os conceitos e métodos de simplificação estabelecidos na Álgebra de Chaveamento possam ser utilizados no máximo possível na construção de circuitos lógicos combinacionais digitais.

1.2 Organização da Dissertação

Este trabalho está organizado em seis capítulos da seguinte maneira:

O Capítulo 1 introduz o trabalho proposto, apresenta os trabalhos prévios, o objetivo da pesquisa, e a organização da dissertação.

O Capítulo 2 propõe uma Álgebra de MV, e define os operadores, axiomas, teoremas e funções, assim como as formas padrão de representação e alguns conceitos relacionados à álgebra proposta.

O Capítulo 3 aborda a análise e a síntese de circuitos digitais combinacionais utilizando a álgebra proposta mostrando alguns exemplos.

O Capítulo 4 trata o método algébrico para a simplificação de funções MVL e apresenta o método de minimização por aplicação do operador *Sucessor* e estende-se os métodos de mapas de Karnaugh, Quine McCluskey e Petrick da Álgebra de Chaveamento para a álgebra proposta.

O Capítulo 5 mostra algumas aplicações utilizando a álgebra proposta e também apresenta uma comparação com os circuitos equivalentes da Álgebra de Chaveamento.

Finalmente, o Capítulo 6 discute os resultados obtidos, apresenta as conclusões deste trabalho e propõe alguns trabalhos futuros.

¹Um conjunto de operadores é chamado Universal se qualquer função lógica pode ser expressa utilizando exclusivamente os operadores do conjunto [32].

Álgebra de MV

2.1 Introdução

Neste capítulo são apresentados os fundamentos matemáticos da álgebra proposta desenvolvida como uma extensão dos operadores conhecidos na Álgebra de Chaveamento [1] e de Post [4], formada por cinco operadores: *Sucessor*, *Máximo*, *Produto Estendido*, *Mínimo* e *Soma Estendida*. Os operadores *Sucessor*, *Máximo* e *Mínimo* são conhecidos e os outros dois operadores são uma extensão dos operadores AND e OR da Álgebra de Chaveamento [35].

Na Seção 2.2, é definido o literal no domínio MVL. Na Seção 2.3, são formalizados os operadores que compõem esta álgebra. Na Seção 2.4, são apresentados os axiomas atuando num domínio \mathbf{D} . Na Seção 2.5, são mostrados os conjuntos de operadores com funcionalidade completa. Na Seção 2.6, é apresentada a dualidade MVL e mostradas as propriedades do segundo conjunto de operadores pelo princípio de dualidade. Na Seção 2.7, são mencionados os teoremas mais importantes desta álgebra. Na Seção 2.8, é definida a função MVL e são apresentadas algumas características importantes. Na Seção 2.9, é apresentada a extensão da lei de De Morgan. Finalmente, na Seção 2.10 são mostradas as representações das portas lógicas MVL.

2.2 Literal MVL

Um literal MVL é uma variável ou alguma forma do operador *Sucessor* na variável que pode assumir um valor que pertence ao conjunto ordenado $\mathbf{D} = \{0, 1, 2, \dots, \mathbf{L}\}$ onde \mathbf{D} é o domínio, \mathbf{L} é o elemento superior de \mathbf{D} igual a $\mathbf{N} - 1$ e \mathbf{N} é a representação digital escolhida da MVL.

2.3 Operadores

2.3.1 Operador Sucessor

Seja A , elemento de \mathbf{D} na lógica de \mathbf{N} valores, o *Sucessor* de A é denotado como A^1 , e o resultado desta operação é: $(A + 1) \bmod \mathbf{N}$, e $\text{Suc}(\text{Suc}(A))$ é denotado como A^{11} ou A^2 .

O operador *Sucessor* foi apresentado por Post [4], como operador *clockwise cycle* (deslocador direito) sendo classificado como operador de apenas um argumento.

A seguir são apresentadas as tabelas verdade para o operador *Sucessor* de $\mathbf{N} = 4$ e 2 nas Tabelas 2.1 e 2.2, respectivamente.

Tabela 2.1: Tabela verdade do operador MVL *Sucessor* de $\mathbf{N} = 4$.

A	A^1	A^2	A^3
0	1	2	3
1	2	3	0
2	3	0	1
3	0	1	2

Tabela 2.2: Tabela verdade do operador MVL *Sucessor* de $\mathbf{N} = 2$.

A	A^1
0	1
1	0

Exemplo 1: Considere $\mathbf{N} = 4$.

$$x = 3^2$$

$$x = (3 + 2) \bmod 4$$

$$x = 5 \bmod 4$$

$$x = 1$$

Exemplo 2: Considere $\mathbf{N} = 2$.

$$x = 1^1$$

$$x = (1 + 1) \bmod 2$$

$$x = 2 \bmod 2$$

$$x = 0$$

2.3.2 Operador Máximo

Sejam A e B elementos de \mathbf{D} , o *Máximo* de A e B é denotado como $A + B$, e o resultado desta operação é: A se, somente se, A for maior ou igual que B , caso contrário é B .

O operador *Máximo* foi apresentado por Łukasiewicz no Journal Ruch Filozoficzny [3].

A seguir são apresentadas as tabelas verdade para o operador *Máximo* de $\mathbf{N} = 4$ e 2 nas Tabelas 2.3 e 2.4, respectivamente.

Tabela 2.3: Tabela verdade do operador MVL *Máximo* de $\mathbf{N} = 4$.

$A \setminus B$	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

Tabela 2.4: Tabela verdade do operador MVL *Máximo* de $\mathbf{N} = 2$.

A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Exemplo 1: Considere $\mathbf{N} = 4$.

$$x = 3 + 1$$

$$x = 3$$

Exemplo 2: Considere $\mathbf{N} = 2$.

$$x = 0 + 1$$

$$x = 1$$

2.3.3 Operador Produto Estendido

Sejam A e B elementos de \mathbf{D} , o *Produto Estendido* de A e B é denotado como $A \star B$, e o resultado desta operação é: i se, somente se, $A = B = i$, caso contrário é igual a 0.

O operador *Produto Estendido* faz parte da metodologia proposta e foi apresentado em [36].

A seguir são apresentadas as tabelas verdade para o operador *Produto Estendido* de $\mathbf{N} = 4$ e 2 nas Tabelas 2.5 e 2.6, respectivamente.

Tabela 2.5: Tabela verdade dos operadores MVL *Produto Estendido* de $\mathbf{N} = 4$.

		$A \star^1 B$				$A \star^2 B$				$A \star^3 B$							
$A \setminus B$		0	1	2	3	$A \setminus B$		0	1	2	3	$A \setminus B$		0	1	2	3
0		0	0	0	0	0		0	0	0	0	0		0	0	0	0
1		0	1	0	0	1		0	0	0	0	1		0	0	0	0
2		0	0	0	0	2		0	0	2	0	2		0	0	0	0
3		0	0	0	0	3		0	0	0	0	3		0	0	0	3

Tabela 2.6: Tabela verdade do operador MVL *Produto Estendido* de $\mathbf{N} = 2$.

A	B	$A \star^1 B$
0	0	0
0	1	0
1	0	0
1	1	1

Exemplo 1: Considere $\mathbf{N} = 4$.

$$x = 1^2 \star^3 2^1$$

$$x = 3 \star^3 3$$

$$x = 3$$

Exemplo 2: Considere $\mathbf{N} = 2$.

$$x = 1^1 \star^1 1$$

$$x = 0 \star^1 1$$

$$x = 0$$

2.3.4 Operador Mínimo

Sejam A e B elementos de \mathbf{D} , o *Mínimo* de A e B é denotado como: $A \cdot B$, e o resultado desta operação é: A se, somente se; A for menor ou igual que B , caso contrário é B .

O operador *Mínimo* foi apresentado por Łukasiewicz no Journal Ruch Filozoficzny [3].

A seguir são apresentadas as tabelas verdade para o operador *Mínimo* de $\mathbf{N} = 4$ e 2 nas Tabelas 2.7 e 2.8, respectivamente.

Exemplo 1: Considere $\mathbf{N} = 4$.

$$x = 2 \cdot 1$$

$$x = 1$$

Tabela 2.10: Tabela verdade do operador MVL *Soma Estendida* de $\mathbf{N} = 2$.

A	B	$A +^0 B$
0	0	0
0	1	1
1	0	1
1	1	1

Exemplo 1: Considere $\mathbf{N} = 4$.

$$x = 3^2 +^3 1^2$$

$$x = 1 +^3 3$$

$$x = 3$$

Exemplo 2: Considere $\mathbf{N} = 2$.

$$x = 1^1 +^0 1$$

$$x = 0 +^0 1$$

$$x = 1$$

2.3.6 Precedência de Operadores

A ordem de avaliação das operações nas expressões MVL é a seguinte: primeiro são avaliadas as operações *Sucessor*, em seguida as operações *Soma Estendida* que têm a mesma precedência com as operações *Produto Estendido* e, finalmente, as operações *Mínimo* que têm a mesma precedência que as operações *Máximo*.

Quando as operações são adjacentes e têm a mesma precedência, elas são associadas da esquerda para a direita.

Exemplo 1: Considere $\mathbf{N} = 4$.

$$x = 2^3 \star^3 2^1 + 2 \star^2 3^3$$

$$x = 1 \star^3 3 + 2 \star^2 2$$

$$x = 0 + 2$$

$$x = 2$$

Exemplo 2: Considere $\mathbf{N} = 2$.

$$x = 1 \star^1 1 + 1^1 \star^1 1 \star^1 0^1$$

$$x = 1 \star^1 1 + 0 \star^1 1 \star^1 1$$

$$x = 1 \star^1 1 + 0 \star^1 1$$

$$x = 1 + 0$$

$$x = 1$$

2.4 Axiomas

A Álgebra de MV proposta é um sistema algébrico fechado e contido num conjunto D de dois ou mais elementos. Com quatro operações de dois argumentos $A \cdot B$, $A + B$, $A +^i B$, $A \star^i B$ e uma operação de um argumento A^1 ; onde A , B e o resultado de cada operação pertencem ao conjunto D . Isto é o postulado de **fechamento**. A seguir apresentam-se outros axiomas que satisfazem a álgebra proposta.

2.4.1 Identidade

- $A^0 = A$
- $A + 0 = A$

2.4.2 Elemento Nulo

- $A + L = L$
- $A \star^i 0 = 0$

2.4.3 Idempotência

- $A + A = A$

2.4.4 Comutatividade

- $A + B = B + A$
- $A \star^i B = B \star^i A$

2.4.5 Associatividade

- $A + (B + C) = (A + B) + C$
- $A \star^i (B \star^i C) = (A \star^i B) \star^i C$

2.4.6 Complemento

- $A^0 + A^1 + A^2 + \dots + A^L = \mathbf{L}$
- $A^0 \star^i A^1 \star^i A^2 \star^i \dots \star^i A^L = 0$

2.4.7 Redução

- $(A^p \star^i B^0) + (A^p \star^i B^1) + \dots + (A^p \star^i B^L) = A^p \star^i i$

2.4.8 Unicidade

- $i \star^i i = i$

2.4.9 Involução

- $A^N = A$

2.5 Conjuntos Funcionalmente Completos

A Álgebra de MV proposta forma dois conjuntos de operadores com funcionalidade completa [33], [34], [38]. A seguir são apresentados os conjuntos de funcionalidade completa.

Primeiro conjunto de operadores com funcionalidade completa:

- *Sucessor*,
- *Máximo*,
- *Produto Estendido*.

Segundo conjunto de operadores com funcionalidade completa:

- *Sucessor*,
- *Mínimo*,
- *Soma Estendida*.

2.6 Dualidade

De maneira similar à Álgebra de Chaveamento [1], onde se estabelece que o problema dual é formado pela troca do 1 por 0 e vice-versa, e a troca do AND por OR e vice-versa, na Álgebra de MV é formada pela troca de operações *Soma Estendida* $+^i$ pelo *Produto Estendido* \star^i e vice-versa, troca de operadores *Máximo* por operadores *Mínimo* e vice-versa, e finalmente pela troca de 0 por \mathbf{L} e vice-versa.

Na Seção 2.4, foram apresentados os axiomas para o primeiro conjunto de operadores, a seguir apresentam-se os axiomas para o segundo conjunto de operadores obtidos pelo princípio de dualidade.

Propriedades	Dual
$A + 0 = A$	$A \cdot \mathbf{L} = A$
$A + \mathbf{L} = \mathbf{L}$	$A \cdot 0 = 0$
$A \star^i 0 = 0$	$A +^i \mathbf{L} = \mathbf{L}$
$A + A = A$	$A \cdot A = A$
$A + B = B + A$	$A \cdot B = B \cdot A$
$A \star^i B = B \star^i A$	$A +^i B = B +^i A$
$A + (B + C) = (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
$A \star^i (B \star^i C) = (A \star^i B) \star^i C$	$A +^i (B +^i C) = (A +^i B) +^i C$
$A^0 + A^1 + A^2 + \dots + A^{\mathbf{L}} = \mathbf{L}$	$A^{\mathbf{L}} \cdot A^1 \cdot A^2 \cdot \dots \cdot A^0 = 0$
$A^0 \star^i A^1 \star^i A^2 \star^i \dots \star^i A^{\mathbf{L}} = 0$	$A^{\mathbf{L}} +^i A^1 +^i A^2 +^i \dots +^i A^0 = \mathbf{L}$
$(A^p \star^i B^0) + (A^p \star^i B^1) + \dots + (A^p \star^i B^{\mathbf{L}}) = A^p \star^i i$	$(A^p +^i B^{\mathbf{L}}) \cdot (A^p +^i B^1) \cdot \dots \cdot (A^p +^i B^0) = A^p +^i i$
$i \star^i i = i$	$i +^i i = i$

2.7 Teoremas

2.7.1 Expansão

- $(A^p \star^i B^q) = (A^{p-i} \star^0 B^{q-i}) + (A^{p-i+1} \star^1 B^{q-i+1}) + \dots + (A^p \star^i B^q)$

Prova:

A operação *Produto Estendido* da forma $(A^p \star^i B^q)$ tem dois possíveis resultados: o primeiro é igual a i se $A^p = B^q = i$, caso contrário, é igual a zero.

Por indução completa:

1. Quando $A^p = B^q = i$:

$$\underbrace{(A^p \star^i B^q)}_i = \underbrace{(A^{p-i} \star^0 B^{q-i})}_0 + \underbrace{(A^{p-i+1} \star^1 B^{q-i+1})}_1 + \dots + \underbrace{(A^p \star^i B^q)}_i$$

$$i = 0 + 1 + \dots + i$$

$$i = i$$

2. Quando em A^p, B^q , i pelo menos um deles é diferente:

$$\underbrace{(A^p \star^i B^q)}_0 = \underbrace{(A^{p-i} \star^0 B^{q-i})}_0 + \underbrace{(A^{p-i+1} \star^1 B^{q-i+1})}_0 + \dots + \underbrace{(A^p \star^i B^q)}_0$$

$$0 = 0 + 0 + \dots + 0$$

$$0 = 0$$

- $(A^{p+i} B^q) = (A^{p+i} B^q) \cdot (A^{p^1+i^1} B^{q^1}) \cdot \dots \cdot (A^{(p-i)^L+i^L} B^{(q-i)^L})$

Prova:

A operação *Soma Estendida* da forma $(A^{p+i} B^q)$ tem dois possíveis resultados: o primeiro é igual a i se $A^p = B^q = i$, caso contrário é igual a L .

Por indução completa:

1. Quando $A^p = B^q = i$:

$$\underbrace{(A^{p+i} B^q)}_i = \underbrace{(A^{p+i} B^q)}_i \cdot \underbrace{(A^{p^1+i^1} B^{q^1})}_{i+1} \cdot \dots \cdot \underbrace{(A^{(p-i)^L+i^L} B^{(q-i)^L})}_L$$

$$i = i \cdot (i + 1) \cdot \dots \cdot L$$

$$i = i$$

2. Quando em A^p, B^q , i pelo menos um deles é diferente:

$$\underbrace{(A^{p+i} B^q)}_L = \underbrace{(A^{p+i} B^q)}_L \cdot \underbrace{(A^{p^1+i^1} B^{q^1})}_L \cdot \dots \cdot \underbrace{(A^{(p-i)^L+i^L} B^{(q-i)^L})}_L$$

$$L = L \cdot L \cdot \dots \cdot L$$

$$L = L$$

2.8 Funções MVL

Uma função MVL, $F(A, B, C, \dots, S)$ está composta de L sub funções, sendo elas: $(F_1(A, B, C, \dots, S))$ indicando o valor da saída igual a 1, $(F_2(A, B, C, \dots, S))$ indicando o valor da saída igual a 2, e assim sucessivamente até $(F_L(A, B, C, \dots, S))$ indicando o valor da saída igual a L .

As funções MVL podem ser representadas de diferentes formas, sendo as mais conhecidas as tabelas verdade e os diagramas de tempo. Tabelas verdade representam funções na forma tabular, enquanto os diagramas de tempo apresentam os valores num período de tempo. Da mesma forma que na Álgebra de Chaveamento [1], nesta álgebra existem as formas standard, conhecidas como "Formas Padrão".

2.8.1 Mintermos e Maxtermos

Mintermos

Para uma função MVL de S literais, se um termo *Produto Estendido* contém S literais e uma única forma do operador *Produto Estendido*, onde cada um dos literais aparece exatamente uma única vez em alguma forma do operador *Sucessor*, então o termo *Produto Estendido* é chamado de mintermo.

Denotado também como $m_{A,B,\dots,S}^i$, onde 'm' identifica o mintermo, e 'i' identifica o tipo de operador *Produto Estendido*.

Assim, para os mintermos $m_{A,B,\dots,S}^i$ e $m_{A,B,\dots,S}^j$, é considerado $m_{A,B,\dots,S}^i$ o mintermo de ordem inferior e $m_{A,B,\dots,S}^j$ o mintermo de ordem superior se, e somente se $i < j$. Caso contrário $m_{A,B,\dots,S}^i$ é considerado mintermo de ordem superior e $m_{A,B,\dots,S}^j$ o mintermo de ordem inferior.

Exemplos para $N = 4$, de 3 literais: $A^1 \star^1 B^3 \star^1 C^2 = m_{0,2,3}^1$, $A^3 \star^2 B^{\star^2} C^1 = m_{3,2,1}^2$, $A^{\star^3} B^1 \star^3 C = m_{3,2,3}^3$.

Maxtermos

Para uma função MVL de S literais, se um termo *Soma Estendida* contém S literais e uma única forma do operador *Soma Estendida*, onde cada um dos literais aparece exatamente uma única vez em alguma forma do operador *Sucessor*, então o termo *Soma Estendida* é chamado de maxtermo.

Denotado também como $M_{A,B,\dots,S}^i$ onde 'M' identifica o maxtermo, e 'i' identifica o tipo de operador *Soma Estendida*.

Assim, para os maxtermos $M_{A,B,\dots,S}^i$ e $M_{A,B,\dots,S}^j$, é considerado $M_{A,B,\dots,S}^i$ o maxtermo de ordem inferior e $M_{A,B,\dots,S}^j$ o maxtermo de ordem superior se, e somente se $i < j$. Caso contrário $M_{A,B,\dots,S}^i$ é considerado maxtermo de ordem superior e $M_{A,B,\dots,S}^j$ o maxtermo de ordem inferior.

Exemplos para $N = 4$, de 3 literais: $A^3 +^0 B^2 +^0 C^1 = M_{1,2,3}^0$, $A^2 +^1 B^1 +^1 C = M_{3,0,1}^1$, $A +^2 B^3 +^2 C^2 = M_{2,3,0}^2$.

2.8.2 Formas Algébricas de Funções MVL

As formas algébricas das funções MVL mantêm similaridade com as formas algébricas das funções de chaveamento. Porém, assim como nas funções de chaveamento existem as formas SOP e POS (Soma de Produtos e Produto de Somas) [35], nas funções MVL são definidas as formas SOPE e POSE (Soma de Operações Produtos Estendidos e Produto de Operações Somas Estendidas).

Formas SOPE e POSE

Funções MVL na forma SOPE são construídas por operações *Produto Estendido* ligadas por operações *Máximo*, onde cada operação *Produto Estendido* está formada por literais em alguma forma do operador *Sucessor*. Exemplo SOPE de uma função de 2 literais, sendo $\mathbf{N} = 4$:

$$F(A, B) = A^2 \star^1 A^2 + A^0 \star^2 B^1 + A^3 \star^3 B^0 \quad (2.1)$$

Funções MVL na forma POSE são construídas por operações *Soma Estendida* ligadas por operações *Mínimo*, onde cada operação *Soma Estendida* esta formada por literais em alguma forma do operador *Sucessor*. Exemplo POSE de uma função de 3 literais, sendo $\mathbf{N} = 4$:

$$F(A, B, C) = A^0 +^0 B^2 +^0 C^2 \cdot A^0 +^1 B^1 \cdot A^3 +^2 B^1 +^2 C^0 \quad (2.2)$$

Formas Padrão MVL

As formas padrão das funções MVL são formas SOPE e POSE com uma característica especial. Como foi visto nos parágrafos anteriores, as funções MVL podem ser representadas de formas diferentes mas que são expressões MVL equivalentes. Mas as formas padrão SOPE e POSE são únicas para cada função MVL.

A característica especial da forma padrão SOPE é ser formada unicamente por mintermos, onde os mintermos iguais a zero são considerados insignificantes porém não são mencionados. E a característica especial da forma padrão POSE é ser formada unicamente por maxtermos, onde os maxtermos iguais a \mathbf{L} são considerados insignificantes porém não são mencionados.

Exemplo 1: Forma padrão SOPE da Equação 2.1.

$$F(A, B) = m_{3,1}^1 + m_{3,0}^1 + m_{3,3}^1 + m_{3,2}^1 + m_{2,1}^2 + m_{0,3}^3$$

$$F(A, B) = A^2 \star^1 B^0 + A^2 \star^1 B^1 + A^2 \star^1 B^2 + A^2 \star^1 B^3 + A^0 \star^2 B^1 + A^3 \star^3 B^0 \quad (2.3)$$

Exemplo 2: Forma padrão POSE da Equação 2.2.

$$F(A, B, C) = M_{0,2,2}^0 \cdot M_{1,0,1}^1 \cdot M_{1,0,0}^1 \cdot M_{1,0,3}^1 \cdot M_{1,0,2}^1 \cdot M_{3,1,2}^2$$

$$\begin{aligned} F(A, B, C) = & A^0 +^0 B^2 +^0 C^2 \cdot A^0 +^1 B^1 +^1 C^0 \cdot A^0 +^1 B^1 +^1 C^1 \cdot \\ & A^0 +^1 B^1 +^1 C^2 \cdot A^0 +^1 B^1 +^1 C^3 \cdot A^3 +^2 B^1 +^2 C^0 \end{aligned} \quad (2.4)$$

2.8.3 Operador Sucessor em Funções MVL

O *Sucessor* de uma função MVL é igual ao *Sucessor* de cada um dos valores das células que correspondem a função na tabela. Um exemplo do operador *Sucessor* numa função MVL é mostrado na Tabela 2.11.

Tabela 2.11: Exemplo do operador *Sucessor* em funções MVL.

$F(A, B)$					$F^1(A, B) = (F(A, B))^1$					$F^2(A, B) = (F^1(A, B))^1$				
$A \setminus B$	0	1	2	3	$A \setminus B$	0	1	2	3	$A \setminus B$	0	1	2	3
0	0	0	0	0	0	1	1	1	1	0	2	2	2	2
1	0	0	0	0	1	1	1	1	1	1	2	2	2	2
2	0	0	0	3	2	1	1	1	0	2	2	2	2	1
3	0	0	1	0	3	1	1	2	1	3	2	2	3	2

O procedimento algébrico da operação *Sucessor* em funções MVL é o seguinte:

Passo 1: Expressar a função $F(A, B)$ em \mathbf{N}^S mintermos onde S é o número de literais. Isto é adicionar na função MVL os mintermos insignificantes.

$$F(A, B) = [A^p \star^v B^q + \dots + A^r \star^w B^s]^t$$

$$\begin{aligned} F(A, B) = & [A^0 \star^0 B^0 + A^0 \star^0 B^1 + \dots + A^0 \star^0 B^L + \\ & A^1 \star^0 B^0 + A^1 \star^0 B^1 + \dots + A^1 \star^0 B^L + \dots + \\ & A^L \star^0 B^0 + A^L \star^0 B^1 + \dots + A^L \star^0 B^L + \\ & A^p \star^v B^q + \dots + A^r \star^w B^s]^t \end{aligned}$$

Logo utiliza-se o teorema de expansão e a função fica com \mathbf{N}^S mintermos.

Passo 2: Distribuir o operador *Sucessor* da função nos literais e operadores *Produto Estendido*.

Passo 3: Avaliar as operações *Sucessor* distribuídas.

Exemplo 1: Seja $F(A, B)$ da Tabela 2.11 obter $F^2(A, B)$ utilizando o procedimento algébrico considerando $\mathbf{N} = 4$.

$$F^2(A, B) = [A^2 \star^1 B^3 + A^1 \star^3 B]^2$$

Passo 1: Expressar a função $F(A, B)$ em $\mathbf{N}^S = 4^2 = 16$ mintermos.

$$\begin{aligned} F^2(A, B) = & [A \star^0 B + A \star^0 B^1 + A \star^0 B^2 + A \star^0 B^3 + \\ & A^1 \star^0 B + A^1 \star^0 B^1 + A^1 \star^0 B^2 + A^1 \star^0 B^3 + \\ & A^2 \star^0 B + A^2 \star^0 B^1 + A^2 \star^0 B^2 + A^2 \star^0 B^3 + \\ & A^3 \star^0 B + A^3 \star^0 B^1 + A^3 \star^1 B^2 + A^3 \star^0 B^3 + \\ & A^2 \star^1 B^3 + A^1 \star^3 B]^2 \end{aligned}$$

$$\begin{aligned} F^2(A, B) = & [A \star^0 B + A \star^0 B^1 + A \star^0 B^2 + A \star^0 B^3 + \\ & A^1 \star^0 B + A^1 \star^0 B^1 + A^1 \star^0 B^3 + A^2 \star^0 B + \\ & A^2 \star^0 B^2 + A^2 \star^0 B^3 + A^3 \star^0 B + A^3 \star^0 B^1 + \\ & A^3 \star^1 B^2 + A^3 \star^0 B^3 + A^2 \star^1 B^3 + A^1 \star^3 B]^2 \end{aligned}$$

Pelo teorema de expansão, veja que o número de mintermos obtidos é igual a 16.

Passo 2: Distribuir o operador *Sucessor* da função nos literais e operadores *Produto Estendido*.

$$\begin{aligned} F^2(A, B) = & A^2 \star^{02} B^2 + A^2 \star^{02} B^{12} + A^2 \star^{02} B^{22} + A^2 \star^{02} B^{32} + \\ & A^{12} \star^{02} B^2 + A^{12} \star^{02} B^{12} + A^{12} \star^{02} B^{32} + A^{22} \star^{02} B^2 + \\ & A^{22} \star^{02} B^{22} + A^{22} \star^{02} B^{32} + A^{32} \star^{02} B^2 + A^{32} \star^{02} B^{12} + \\ & A^{32} \star^{12} B^{22} + A^{32} \star^{02} B^{32} + A^{22} \star^{12} B^{32} + A^{12} \star^{32} B^2 \end{aligned}$$

Passo 3: Avaliar as sucessões distribuídas.

$$\begin{aligned} F^2(A, B) = & A^2 \star^2 B^2 + A^2 \star^2 B^3 + A^2 \star^2 B + A^2 \star^2 B^1 + \\ & A^3 \star^2 B^2 + A^3 \star^2 B^3 + A^3 \star^2 B^1 + A \star^2 B^2 + \\ & A \star^2 B + A \star^2 B^1 + A^1 \star^2 B^2 + A^1 \star^2 B^3 + \\ & A^1 \star^3 B + A^1 \star^2 B^1 + A \star^3 B^1 + A^3 \star^1 B^2 \end{aligned}$$

2.9 Lei de De Morgan Estendida

A lei de De Morgan da Álgebra de Chaveamento [35] é formada por dois passos: sendo que o primeiro passo consiste em trocar os operandos pelo operador dual e o segundo passo consiste no ingresso do operador complemento nos literais. De maneira similar na álgebra MVL é formada por dois passos: o primeiro consiste em escrever a expressão baseando-se no operador dual, e o segundo passo consiste em operar o *Sucessor* da função MVL como foi mostrado na Seção 2.8.3.

Exemplo 1: Considere $\mathbf{N} = 2$.

$$F(A, B) = [A \star^1 B^1]^1$$

Passo 1: Aplicar o princípio de dualidade.

$$F(A, B) = [A^1 +^0 B]^1$$

Passo 2: Operar o *Sucessor* da função MVL.

$$F(A, B) = A^1 +^0 B$$

Logo:

$$F(A, B) = [A \star^1 B^1]^1 = A^1 +^0 B$$

2.10 Portas Lógicas MVL

As portas lógicas são dispositivos eletrônicos que têm a capacidade de realizar operações lógicas com variáveis lógicas MVL e as representações das portas lógicas são mostradas na Figura 2.1.

2.10.1 Simulações das Portas Lógicas MVL

As simulações que mostram o correto funcionamento das portas lógicas mostradas na Figura 2.1 foram implementadas em VHDL sendo os sinais de entrada e de saída elementos de \mathbf{D} . Na simulação apresentada na Figura 2.2, pode-se observar que na primeira coluna "*Cont*" identifica-se a simulação como **MVL**, na segunda coluna "*Signal*" mostra-se os nomes das entradas e das saídas, a terceira coluna, "*Value*" representa os valores das saídas e entradas aos 6580ns de simulação e a última coluna corresponde aos valores MVL de cada variável. Nesta simulação foi considerada a representação numérica quaternária $\mathbf{N} = 4$, sendo a entrada do operador *Sucessor* "a" e para todo o resto dos operadores as entradas são "a" e "b". Entende-se "sext" como *Soma Estendida* e "pext" como *Produto Estendido*. Finalmente no eixo horizontal mostra-se o tempo de simulação.

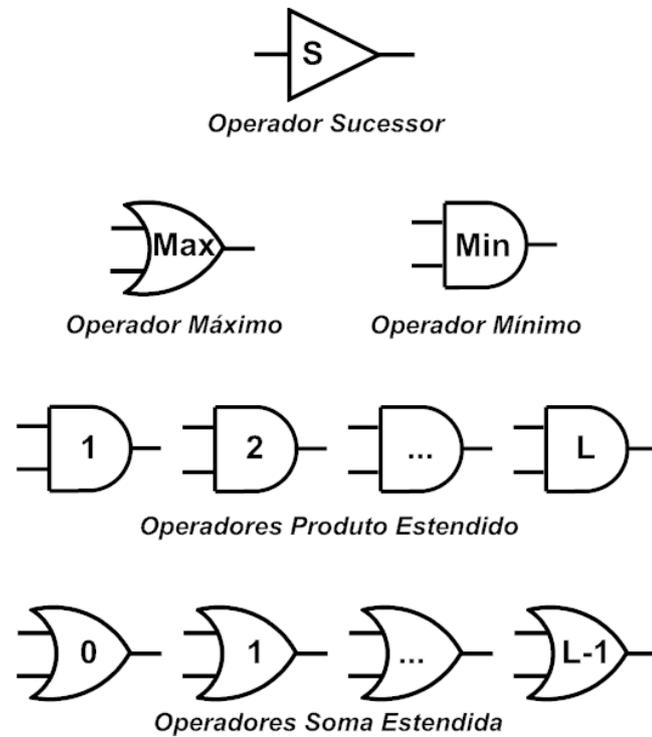


Figura 2.1: Representação das portas lógicas MVL.

Cont	Signal	Value	6580ns																							
			0ns	2000ns	4000ns	6000ns	8000ns	10000ns	12000ns	14000ns	16000ns															
mv1	a	2	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3								
mv1	b	1	0				1				2				3											
mv1	sucessor	3	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0								
mv1	minimo	1	0				1				0				1				2				3			
mv1	maximo	2	0	1	2	3	1	2	3	2	3	2	3	2	3	2	3	2	3							
mv1	sext0	3	0																							
mv1	sext1	3	3				1				3				3											
mv1	sext2	3	3				3				2				3											
mv1	pext1	0	0				1				0				0											
mv1	pext2	0	0				0				2				0											
mv1	pext3	0	0																							

Figura 2.2: Simulação dos operadores MVL de $N = 4$.

Circuitos Digitais Combinacionais

MVL

Os circuitos digitais combinacionais também chamados circuitos digitais sem memória, são aqueles onde a saída depende somente da entrada [32]. Assim a operação de tais circuitos pode ser descrita completamente em uma tabela verdade [39].

Adicionalmente, na álgebra proposta pode-se realizar a análise e a síntese dos circuitos com os dois conjuntos de operadores totalmente funcionais apresentados na Seção 2.5.

Na Seção 3.1 é mostrada a análise de circuitos digitais combinacionais MVL usando tabelas verdade, funções MVL e diagramas de tempo e na Seção 3.2 é realizada a síntese de circuitos MVL nas formas SOPE e POSE.

3.1 Análise de Circuitos Digitais Combinacionais MVL

A análise do circuito é o processo de descrição exata da sua operação. Na Álgebra de MV pode-se descrever o circuito mediante uma tabela verdade, uma função MVL ou por diagramas de tempo.

3.1.1 Análise com a Tabela Verdade

Seja um circuito combinacional no qual deseja-se conhecer a operação do circuito, o primeiro passo é criar uma tabela verdade com o número de colunas igual ao número de entradas mais

um, onde a última coluna corresponde ao valor de saída da função. A tabela deverá conter todas as possíveis combinações das entradas. O próximo passo é determinar a função de saída para cada uma das possíveis combinações na entrada.

Exemplo 1: Na Figura 3.1, mostra-se um circuito e as duas primeiras combinações para preencher a tabela verdade mostrada na Tabela 3.1.

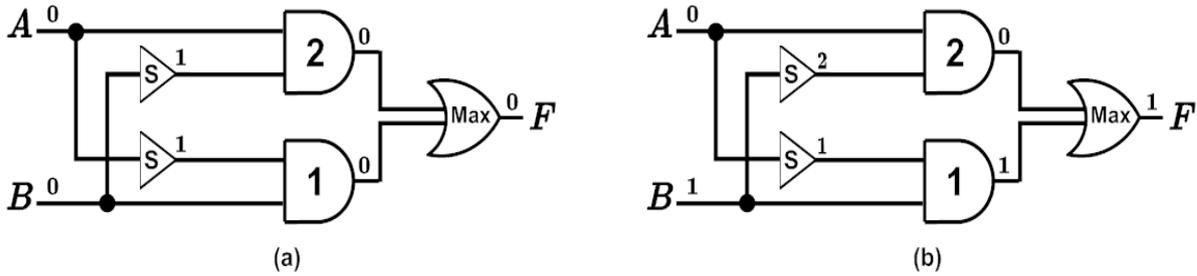


Figura 3.1: Análise de circuitos por tabelas verdade.

Tabela 3.1: Exemplo de análise de circuitos usando a tabela verdade.

A	B	$F(A, B)$
0	0	0
0	1	1
⋮	⋮	⋮
3	3	

3.1.2 Análise com a Função MVL

Seja um circuito combinacional no qual se deseja conhecer a operação do circuito expressa numa função MVL, o procedimento é escrever as funções MVL na saída de cada porta lógica. A função que estiver na saída da última porta é a que representa o circuito.

Exemplo 1: Na Figura 3.2 mostra-se a análise de um circuito lógico utilizando funções MVL.

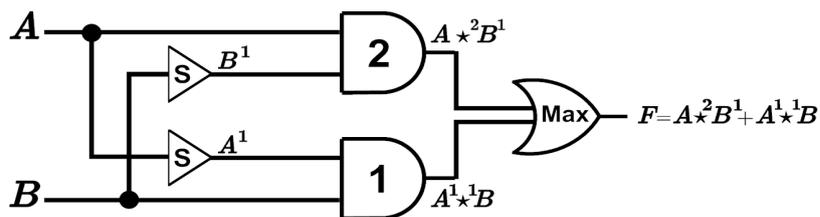


Figura 3.2: Análise de circuitos por funções MVL.

3.1.3 Análise com o Diagrama de Tempo

Seja um circuito combinacional no qual se deseja conhecer a operação do circuito expressada numa função MVL, o que se faz é aplicar a seqüência de valores para as entradas do circuito num período de tempo e observar a relação da entrada com a correspondente seqüência da saída.

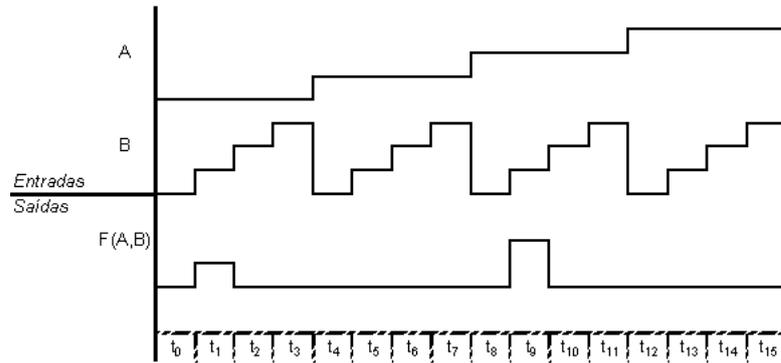


Figura 3.3: Análise de circuitos por diagramas de tempo.

3.2 Síntese de Circuitos Digitais Combinacionais MVL

A síntese de circuitos combinacionais é o procedimento inverso da análise de circuitos combinacionais. A síntese inicia-se com a descrição da operação do circuito, da descrição deriva a tabela verdade da qual se obtém a função lógica que descreve exatamente o circuito. Depois de obter a função lógica pode-se construir o diagrama do circuito.

Uma tabela verdade pode ser sintetizada de duas formas a primeira é na forma padrão SOPE (Mintermos) e a segunda é na forma padrão POSE (Maxtermos).

3.2.1 Síntese de Circuitos Lógicos na Forma SOPE

A síntese de circuitos na forma SOPE consiste em expressar uma tabela verdade numa função da forma SOPE. Assim para sintetizar a função $F(A, B, C, \dots, S)$ com A, B, C, \dots, S literais, deve-se obter as seguintes subfunções; subfunção com saída igual a 1 ($F_1(A, B, C, \dots, S)$), subfunção com saída igual a 2 ($F_2(A, B, C, \dots, S)$), subfunção com saída igual a 3 ($F_3(A, B, C, \dots, S)$), assim até a subfunção com saída igual a L ($F_L(A, B, C, \dots, S)$). Logo, sobre todas as subfunções é aplicado o operador *Máximo*, como é mostrado na Equação 3.1.

$$F(A, B, \dots, S) = F_1(A, B, \dots, S) + F_2(A, B, \dots, S) + \dots + F_L(A, B, \dots, S) \quad (3.1)$$

A seguir mostra-se a função para a síntese da Tabela 3.2 de $N = 4$, para ilustrar o procedimento de síntese na forma SOPE.

Tabela 3.2: Exemplo de síntese na forma SOPE.

$A \setminus B$	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	3	0
3	0	1	1	2

Como a representação digital é quaternária ($N = 4$), então é preciso $F_1(A, B)$, $F_2(A, B)$ e $F_3(A, B)$ para a síntese.

A função a ser sintetizada com saída igual a 1, tem dois mintermos: $F(A = 3, B = 1) = 1$ e $F(A = 3, B = 2) = 1$. Para a saída ser igual a 1 é preciso utilizar o operador *Produto Estendido* (\star^1). Como mostrado na Seção 2.3 para que a operação *Produto Estendido* (\star^1) seja igual a 1, é preciso que os operandos também sejam iguais a 1.

Logo a função que sintetiza $F_1(A, B) = 1$ da Tabela 3.2 com dois mintermos é mostrada na Equação 3.2.

$$F_1(A, B) = A^2 \star^1 B^0 + A^2 \star^1 B^3 \quad (3.2)$$

Veja, se $A = 3 \Rightarrow A^2 = 1$ e se $B = 1 \Rightarrow B^0 = 1$, assim para o primeiro mintermo resulta $A^2 \star^1 B^0 = 1$. Para o segundo mintermo se $A = 3 \Rightarrow A^2 = 1$ e se $B = 2 \Rightarrow B^3 = 1$, logo, $A^2 \star^1 B^3 = 1$. Para todos os outros valores dos operandos A e B têm-se $F_1(A, B) = 0$.

A síntese das funções $F_2(A, B)$ e $F_3(A, B)$ efetuam-se de maneira similar.

Para sintetizar $F_2(A, B)$ utiliza-se o operador *Produto Estendido* (\star^2). A função que sintetiza $F_2(A, B) = 2$ da Tabela 3.2 é mostrada na Equação 3.3.

$$F_2(A, B) = A^3 \star^2 B^3 \quad (3.3)$$

Para sintetizar $F_3(A, B)$ utiliza-se o operador *Produto Estendido* (\star^3). A função que sintetiza $F_3(A, B) = 3$ da Tabela 3.2 é mostrada na Equação 3.4.

$$F_3(A, B) = A^1 \star^3 B^1 \quad (3.4)$$

De acordo com a metodologia aplica-se o operador *Máximo* a $F_1(A, B)$, $F_2(A, B)$ e $F_3(A, B)$ resultando a Equação 3.5 das Equações 3.2, 3.3 e 3.4, que representam a função de síntese da Tabela 3.2.

$$F(A, B) = F_1(A, B) + F_2(A, B) + F_3(A, B)$$

$$F(A, B) = A^2 \star^1 B^0 + A^2 \star^1 B^3 + A^3 \star^2 B^3 + A^1 \star^3 B^1 \quad (3.5)$$

Exemplo 1: Projeto de um circuito meio-somador para duas entradas A, B , onde as entradas podem ser 0, 1, 2 e 3. Encontre a soma das duas entradas.

Depois de avaliar a descrição, constrói-se a tabela verdade para o meio-somador com todas as características mencionadas. A base digital de representação escolhida é a quaternária $\mathbf{N} = 4$, e a saída da função lógica é a soma de A e B . A Tabela 3.3 mostra a descrição do circuito.

Tabela 3.3: Tabela verdade do meio-somador de $\mathbf{N} = 4$.

A	B	C_{out}		Q_1	
0	0	0	$A \star^0 B$	0	$A \star^0 B$
0	1	0	$A \star^0 B^3$	1	$A^1 \star^1 B$
0	2	0	$A \star^0 B^2$	2	$A^2 \star^2 B$
0	3	0	$A \star^0 B^1$	3	$A^3 \star^3 B$
1	0	0	$A^3 \star^0 B$	1	$A \star^1 B^1$
1	1	0	$A^3 \star^0 B^3$	2	$A^1 \star^2 B^1$
1	2	0	$A^3 \star^0 B^2$	3	$A^2 \star^3 B^1$
1	3	1	$A \star^1 B^2$	0	$A^3 \star^0 B^1$
2	0	0	$A^2 \star^0 B$	2	$A \star^2 B^2$
2	1	0	$A^2 \star^0 B^3$	3	$A^1 \star^3 B^2$
2	2	1	$A^3 \star^1 B^3$	0	$A^2 \star^0 B^2$
2	3	1	$A^3 \star^1 B^2$	1	$A^2 \star^1 B^1$
3	0	0	$A^1 \star^0 B$	3	$A \star^3 B^3$
3	1	1	$A^2 \star^1 B$	0	$A^1 \star^0 B^3$
3	2	1	$A^2 \star^1 B^3$	1	$A^2 \star^1 B^3$
3	3	1	$A^2 \star^1 B^2$	2	$A^3 \star^2 B^3$

Para obter as funções que representem a Tabela 3.3, utiliza-se a forma padrão de representação SOPE (Minterms). As Equações 3.6 e 3.7 representam as funções lógicas da Tabela 3.3 na forma SOPE.

$$Q_1 = A^1 \star^1 B + A^2 \star^2 B + A^3 \star^3 B + A \star^1 B^1 + A^1 \star^2 B^1 + A^2 \star^3 B^1 + A \star^2 B^2 + A^1 \star^3 B^2 + A^2 \star^1 B^1 + A \star^3 B^3 + A^2 \star^1 B^3 + A^3 \star^2 B^3 \quad (3.6)$$

$$C_{out} = A \star^1 B^2 + A^3 \star^1 B^3 + A^3 \star^1 B^2 + A^2 \star^1 B + A^2 \star^1 B^3 + A^2 \star^1 B^2 \quad (3.7)$$

A simulação que mostra o correto funcionamento do circuito lógico do meio somador mostrado na Figura 3.4 têm sido implementado em VHDL a partir das Equações 3.6 e 3.7, sendo os dois sinais de entrada e os dois sinais de saída elementos de \mathbf{D} . Na simulação apresentada na Figura 3.5 pode-se observar que na primeira coluna "Cont" identifica a simulação como \mathbf{MVL} , na segunda coluna "Signal" mostra-se os nomes dos sinais sendo eles entradas e saídas, já a terceira coluna "Value" representa os valores das entradas e das saídas aos 7580ns de simulação e a última coluna corresponde aos valores MVL de cada sinal. Para a simulação do circuito digital foi considerada a representação digital quaternária $\mathbf{N} = 4$, sendo as entradas do meio somador "a" e "b" e as saídas "cout" e "soma" que correspondem a A, B, C_{out} e Q_1 das Equações 3.6 e 3.7. Finalmente no eixo horizontal mostra-se o tempo de simulação.

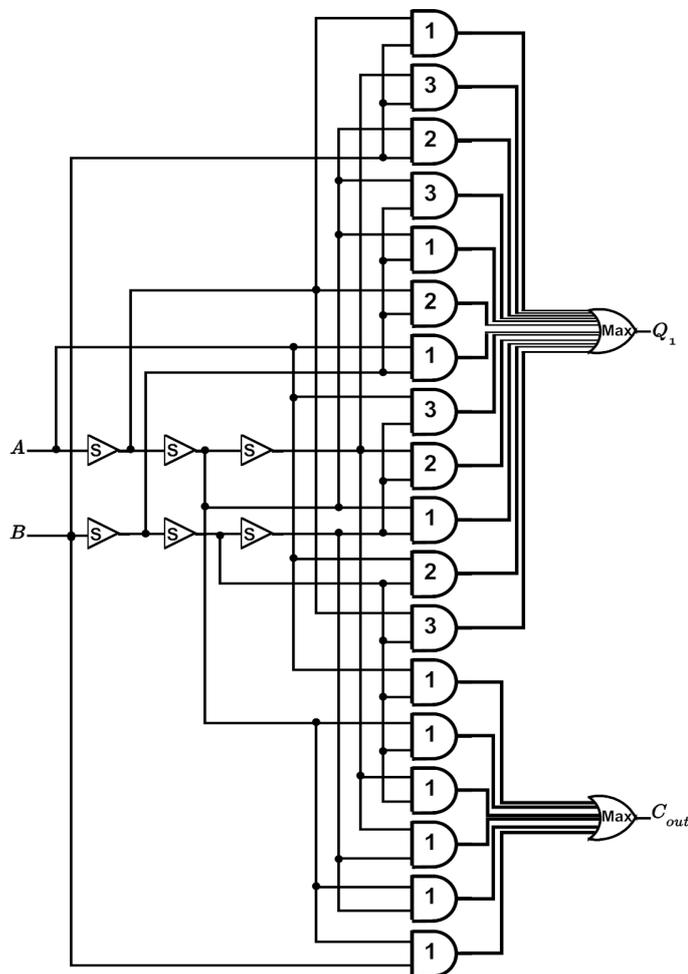


Figura 3.4: Circuito lógico do meio somador MVL de $\mathbf{N} = 4$.

3.2.2 Síntese de Circuitos Lógicos na Forma POSE

A síntese de circuitos na forma POSE consiste em expressar uma tabela verdade numa função da forma POSE. Assim para sintetizar a função $F(A, B, C, \dots, S)$ com A, B, C, \dots, S literais, deve-se obter as seguintes subfunções; subfunção com saída igual a 0 ($F_0(A, B, C, \dots, S)$),

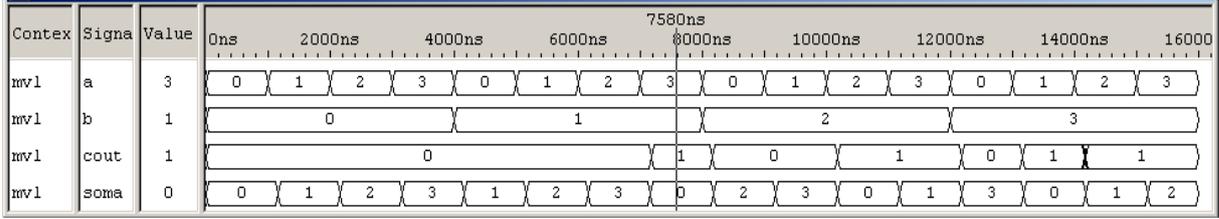


Figura 3.5: Simulação do meio somador MVL de $N = 4$.

subfunção com saída igual a 1 ($F_1(A, B, C, \dots, S)$), subfunção com saída igual a 2 ($F_2(A, B, C, \dots, S)$), assim até a subfunção com saída igual a $L-1$ ($F_{L-1}(A, B, C, \dots, S)$) logo sobre todas as subfunções é aplicado o operador *Mínimo*, como mostra-se na Equação 3.8.

$$F(A, B, C, \dots, S) = F_0(A, B, C, \dots, S) \cdot F_1(A, B, C, \dots, S) \cdot \dots \cdot F_{L-1}(A, B, C, \dots, S) \quad (3.8)$$

A seguir mostra-se a função para a síntese da Tabela 3.4 com $N = 4$, para ilustrar o procedimento de síntese na forma POSE.

Tabela 3.4: Exemplo de síntese na forma POSE.

$A \setminus B$	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	3	0
3	0	1	1	2

Como a representação digital é quaternária ($N = 4$), então é preciso $F_0(A, B)$, $F_1(A, B)$ e $F_2(A, B)$ para a síntese.

A função a ser sintetizada com saída igual a 0, tem doze maxtermos: $F(A = 0, B = 0) = 0$, $F(A = 0, B = 1) = 0$ até $F(A = 3, B = 0) = 0$. Para a saída ser igual a 0 é preciso utilizar o operador *Soma Estendida* ($+^0$). Como mostrado na Seção 2.3, para que a operação *Soma Estendida* ($+^0$) seja igual a 0, é preciso que os operandos também sejam iguais a 0.

Logo, a função que sintetiza $F_0(A, B) = 0$ da Tabela 3.4 com doze maxtermos é mostrada na Equação 3.9 e a Equação 3.10 mostra uma forma simplificada.

$$\begin{aligned}
 F_0(A, B) = & (A +^0 B) \cdot (A +^0 B^3) \cdot (A +^0 B^2) \cdot (A +^0 B^1) \cdot (A^3 +^0 B) \cdot \\
 & (A^3 +^0 B^3) \cdot (A^3 +^0 B^2) \cdot (A^3 +^0 B^1) \cdot (A^2 +^0 B) \cdot \\
 & (A^2 +^0 B^3) \cdot (A^2 +^0 B^1) \cdot (A^1 +^0 B)
 \end{aligned} \quad (3.9)$$

$$F_0(A, B) = (A +^0 A) \cdot (A^3 +^0 A^3) \cdot (B +^0 B) \cdot (A^2 +^0 B^3) \cdot (A^2 +^0 B^1) \quad (3.10)$$

Veja, se $A = 0 \Rightarrow A^0 = 0$ e se $B = 0 \Rightarrow B^0 = 0$, assim para o primeiro maxtermo resulta $A^0 +^0 B^0 = 0$. Para o segundo maxtermo se $A = 0 \Rightarrow A^0 = 0$ e se $B = 1 \Rightarrow B^3 = 0$, logo, $A^0 +^0 B^3 = 0$. Para todos os outros maxtermos com saída igual a zero se faz o mesmo.

A síntese das funções $F_1(A, B)$ e $F_2(A, B)$ efetuam-se de maneira similar.

Para sintetizar $F_1(A, B)$ utiliza-se o operador *Soma Estendida* ($+^1$). A função que sintetiza $F_1(A, B) = 1$ da Tabela 3.4 é mostrada na Equação 3.11.

$$F_1(A, B) = (A^2 +^1 B^0) \cdot (A^2 +^1 B^3) \quad (3.11)$$

Para sintetizar $F_2(A, B)$ utiliza-se o operador *Soma Estendida* ($+^2$). A função que sintetiza $F_2(A, B) = 2$ da Tabela 3.4 é mostrada na Equação 3.12.

$$F_2(A, B) = A^3 +^2 B^3 \quad (3.12)$$

De acordo com a metodologia aplica-se o operador *Mínimo* a $F_0(A, B)$, $F_1(A, B)$ e $F_2(A, B)$ resultando a Equação 3.13 das Equações 3.10, 3.11 e 3.12, que representam a função de síntese da Tabela 3.4.

$$F(A, B) = F_0(A, B) \cdot F_1(A, B) \cdot F_2(A, B)$$

$$F(A, B) = (A +^0 A) \cdot (A^3 +^0 A^3) \cdot (B +^0 B) \cdot (A^2 +^0 B^3) \cdot (A^2 +^0 B^1) \cdot (A^2 +^1 B^0) \cdot (A^2 +^1 B^3) \cdot (A^3 +^2 B^3) \quad (3.13)$$

Exemplo 1: Projeto de um circuito comparador que para duas entradas A , B , onde as entradas podem ser 0, 1, 2 e 3. Encontre qual das duas entradas é maior.

Depois de avaliar a descrição, constrói-se a tabela verdade para o comparador com todas as características mencionadas. A base digital de representação escolhida é a quaternária $\mathbf{N} = 4$, e a saída da função lógica quando A é maior será igual a 1, se B for maior será igual a 2 e fossem iguais será 0. A Tabela 3.5 mostra a descrição do circuito.

Para obter a função que represente a Tabela 3.5, temos duas possibilidades, sendo a primeira representar na forma padrão SOPE (Mintermos) e a segunda na forma padrão POSE (Maxtermos). Para este exemplo se representará das duas formas. A Equação 3.14 apresenta a função lógica da forma SOPE da Tabela 3.5, enquanto a Equação 3.15 apresenta a função equivalente na forma POSE.

Tabela 3.5: Comparador de duas entradas e uma saída de $\mathbf{N} = 4$.

$A \setminus B$	0	1	2	3
0	0	2	2	2
1	1	0	2	2
2	1	1	0	2
3	1	1	1	0

$$F(A, B) = A^2 \star^2 B^1 + A^2 \star^2 B + A^2 \star^2 B^3 + A \star^1 B^1 + A^1 \star^2 B + A^1 \star^2 B^3 + A^3 \star^1 B^1 + A^3 \star^1 B + A \star^2 B^3 + A^2 \star^1 B^1 + A^2 \star^1 B + A^2 \star^1 B^3 \quad (3.14)$$

$$F(A, B) = (A +^0 B) \cdot (A^2 +^2 B^1) \cdot (A^2 +^2 B) \cdot (A^2 +^2 B^3) \cdot (A +^1 B^1) \cdot (A^3 +^0 B^3) \cdot (A^1 +^2 B) \cdot (A^1 +^2 B^3) \cdot (A^3 +^1 B^1) \cdot (A^3 +^1 B) \cdot (A^2 +^0 B^2) \cdot (A +^2 B^3) \cdot (A^2 +^1 B^1) \cdot (A^2 +^1 B) \cdot (A^2 +^1 B^3) \cdot (A^1 +^0 B^1) \quad (3.15)$$

A simulação que mostra o correto funcionamento do circuito lógico do comparador mostrado na Figura 3.6 foi implementado em VHDL a partir da Equação 3.14 sendo os dois sinais de entrada e o sinal de saída elementos de \mathbf{D} . Na simulação apresentada na Figura 3.7, pode-se observar que a primeira coluna "*Cont*" identifica a simulação como \mathbf{MVL} , na segunda coluna "*Signal*" mostra-se os nomes dos sinais sendo eles entradas e saídas, a terceira coluna "*Value*" representa os valores das entradas e a saída aos 4760ns de simulação e a última coluna corresponde aos valores MVL de cada sinal. Para a simulação do circuito digital foi considerada a representação digital quaternária $\mathbf{N} = 4$, sendo as entradas do comparador "a" e "b" que correspondem a A e B da Equação 3.14 e a "saida" que é $F(A, B)$. Finalmente no eixo horizontal mostra-se o tempo de simulação.

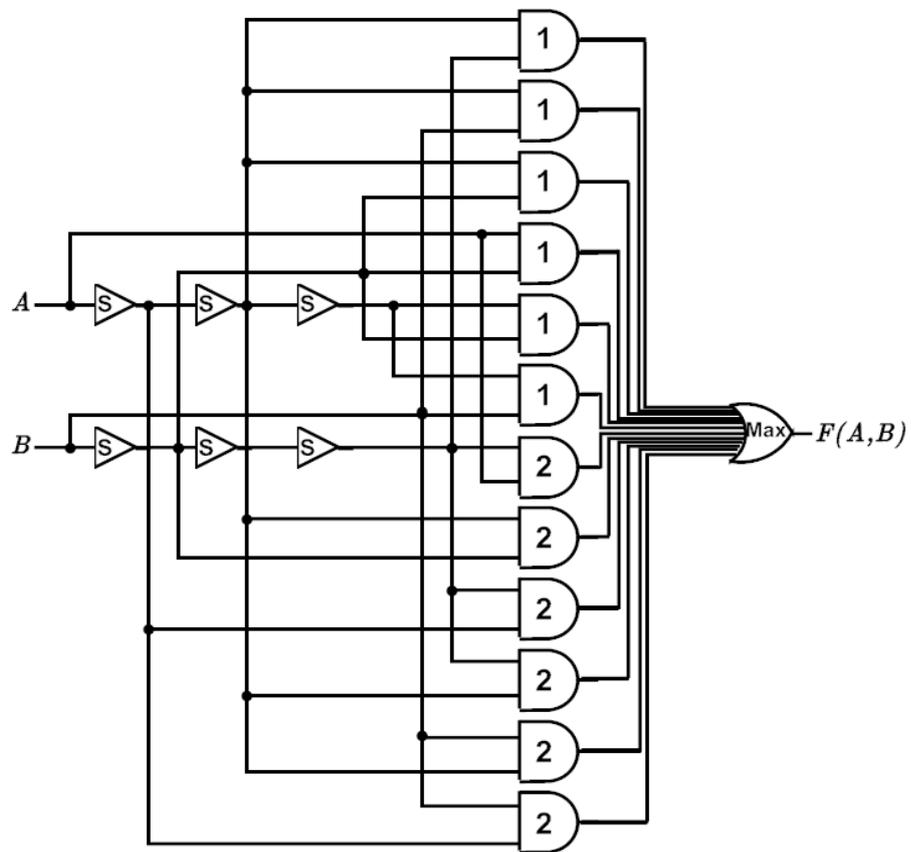


Figura 3.6: Circuito lógico do comparador de duas entradas e uma saída de $N = 4$.

Cont	Signal	Value	4760ns																
			0ns	2000ns	4000ns	6000ns	8000ns	10000ns	12000ns	14000ns	16000ns								
mv1	a	0	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	
mv1	b	1	0				1				2				3				
mv1	saida	2	0	1	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0

Figura 3.7: Simulação do comparador de duas entradas e uma saída de $N = 4$.

Simplificação de Funções MVL

No projeto de circuitos digitais é importante a simplificação de funções lógicas com o objetivo de otimizar o circuito em termos do número de portas lógicas, velocidade e consumo de energia. Assim, simplificar é reduzir o número de termos ou literais de uma função MVL. O tamanho de um circuito é proporcional ao tamanho ou complexidade da função algébrica, razão pela qual neste capítulo estende-se; o mapa de Karnaugh [35], método Quine McCluskey [40] e Algoritmo Petrick [32] da Álgebra de Chaveamento [1] para a Álgebra de MV. Também são apresentados outros métodos para a minimização de funções MVL.

Na Seção 4.1 é apresentada a simplificação algébrica MVL. Na Seção 4.2 é mostrada a minimização de funções MVL usando o operador *Sucessor*. Nas Seções 4.3, 4.4 e 4.5 são estendidos: os mapas de Karnaugh, o método Quine McCluskey e o Algoritmo Petrick, respectivamente para MVL. Finalmente, na Seção 4.6 utiliza-se os dois conjuntos de operadores para minimizar funções.

4.1 Simplificação Algébrica MVL

A simplificação algébrica de funções MVL está baseada diretamente na utilização dos axiomas e teoremas apresentados com o objetivo de reduzir o número de termos, dita manipulação algébrica também é chamada como minimização de funções lógicas, porém a utilização de diferentes teoremas e axiomas levarão a funções algébricas distintas mas equivalentes, mas a simplificação algébrica não garante que a função achada seja a função mínima equivalente.

A seguir mostra-se alguns exemplos.

Exemplo 1: Seja a Equação 4.1, procure a equação mínima equivalente. Considere $\mathbf{N} = 4$.

$$F(A, B) = A^1 \star^1 B^1 + A^1 \star^1 B^3 + A \star^1 B^1 + A \star^1 B + A^2 \star^3 B^1 + A \star^1 B^2 + A^3 \star^1 B^1 + A^1 \star^3 B^2 + A^3 \star^1 B^3 + A^3 \star^1 B^2 + A^2 \star^1 B + A^2 \star^1 B^3 \quad (4.1)$$

$$F(A, B) = A \star^1 B + A \star^1 B^1 + A \star^1 B^2 + A^1 \star^1 B^1 + A^1 \star^1 B^3 + A^2 \star^1 B + A^2 \star^1 B^3 + A^3 \star^1 B^1 + A^3 \star^1 B^2 + A^3 \star^1 B^3 + A^1 \star^3 B^2 + A^2 \star^3 B^1$$

(Axioma de Comutatividade)

Utiliza-se o axioma de comutatividade para ordenar os termos de acordo ao tipo de operador *Produto Estendido* com o objetivo de identificar os termos que possam ser simplificados.

$$F(A, B) = A \star^1 B + A \star^1 B^1 + A \star^1 B^2 + A^1 \star^1 B^1 + A^1 \star^1 B^3 + A^2 \star^1 B + A^2 \star^1 B^3 + A^3 \star^1 B^1 + A^3 \star^1 B^2 + A^3 \star^1 B^3 + \underline{A^1 \star^3 B^2 + A^3 \star^1 B} + \underline{A^2 \star^3 B^1 + A \star^1 B^3} \text{ (Teorema de Expansão)}$$

Gera-se pelo teorema de expansão termos úteis para a simplificação utilizando o Teorema de Expansão.

$$F(A, B) = A \star^1 B + A \star^1 B^1 + A \star^1 B^2 + A \star^1 B^3 + A^1 \star^1 B^1 + A^1 \star^1 B^3 + A^2 \star^1 B + A^2 \star^1 B^3 + A^3 \star^1 B + A^3 \star^1 B^1 + A^3 \star^1 B^2 + A^3 \star^1 B^3 + A^1 \star^3 B^2 + A^2 \star^3 B^1 \text{ (Axioma de Comutatividade)}$$

Os termos gerados pelo teorema de expansão são ordenados utilizando o axioma da comutatividade para serem simplificados pelo axioma de redução.

$$F(A, B) = \underline{A \star^1 A} + A^1 \star^1 B^1 + A^1 \star^1 B^3 + A^2 \star^1 B + A^2 \star^1 B^3 + \underline{A^3 \star^1 A^3} + A^1 \star^3 B^2 + A^2 \star^3 B^1 \text{ (Axioma de Redução)}$$

4.2 Minimização por Aplicação do Operador Sucessor

O método de minimização por aplicação do operador *Sucessor*, consiste em obter o menor número de mintermos pela aplicação do operador *Sucessor* numa função MVL. Porém este método não garante a função mínima equivalente.

O procedimento para a minimização de funções pela aplicação do operador *Sucessor* é:

1. Identificar o valor do mintermo com maior número de repetições na saída da função. Se existir dois mintermos com o mesmo número de repetições considerar o mintermo de ordem inferior;
2. Obter x , onde x é igual a \mathbf{N} menos o valor obtido no passo 1;
3. Aplicar x vezes o operador *Sucessor* a cada valor de saída da função;
4. Sintetizar a nova função $F^x(A, B)$;
5. A função simplificada será igual à função sintetizada com $\mathbf{N} - x$ aplicações do operador *Sucessor* ($F^{x\mathbf{N}-x}(A, B)$).

Exemplo 1: Minimizar a função lógica do comparador apresentado na Seção 3.2.1, utilizando o método proposto, sendo $\mathbf{N} = 4$.

Tabela 4.1: Exemplo 1 de minimização por aplicação do operador *Sucessor*.

		$F(A, B)$			
$A \setminus B$		0	1	2	3
0		0	2	2	2
1		1	0	2	2
2		1	1	0	2
3		1	1	1	0

1. A Tabela 4.1, apresenta seis mintermos iguais a 1 e seis mintermos iguais a 2. Então, por ter o mesmo número de repetições considera-se o menor, neste caso 1;
2. Logo, $x = \mathbf{N} - 1$, então $x = 3$;
3. Ao aplicar 3 vezes o operador *Sucessor* a todos os valores da função $F(A, B)$, obteve-se a Tabela 4.2 igual a $F^3(A, B)$;
4. Síntese de $F^3(A, B) = A^3 \star^3 B^3 + A^1 \star^1 B + A^1 \star^1 B^3 + A^1 \star^1 B^2 + A^2 \star^3 B^2 + A \star^1 B^3 + A \star^1 B^2 + A^1 \star^3 B^1 + A^3 \star^1 B^2 + A \star^3 B$;

Tabela 4.2: Três aplicações do operador *Sucessor* em $F(A, B)$ da Tabela 4.1.

		$F^3(A, B)$			
$A \setminus B$		0	1	2	3
0		3	1	1	1
1		0	3	1	1
2		0	0	3	1
3		0	0	0	3

5. $F(A, B) = (F^{3^{4-3}}(A, B))$

$$F(A, B) = [A^3 \star^3 B^3 + A^1 \star^1 B + A^1 \star^1 B^3 + A^1 \star^1 B^2 + A^2 \star^3 B^2 + A \star^1 B^3 + A \star^1 B^2 + A^1 \star^3 B^1 + A^3 \star^1 B^2 + A \star^3 B]^1 \quad (4.2)$$

Logo a função minimizada do comparador MVL mostrada na Equação 4.2 fica com dois termos a menos que a Equação 3.14.

Exemplo 2: Minimizar a Tabela 4.3, utilizando o método proposto, sendo $N = 4$.

Tabela 4.3: Exemplo 2 de minimização por aplicação do operador *Sucessor*.

		$F(A, B)$			
$A \setminus B$		0	1	2	3
0		3	3	3	3
1		3	3	3	3
2		3	0	1	2
3		3	3	3	3

1. A Tabela 4.3, apresenta um mintermo igual a 1, um mintermo igual a 2 e treze mintermos iguais a 3, logo o mintermo com maior número de repetições é igual a 3;
2. Logo, $x = N - 3$ então $x = 1$;
3. Ao aplicar 1 vez o operador *Sucessor* a todos os valores da saída de $F(A, B)$, obtém-se a Tabela 4.4 igual a $F^1(A, B)$;
4. Síntese de $F^1(A, B) = A^3 \star^1 B + A \star^2 B + A^1 \star^3 B$;
5. $F(A, B) = (F^1(A, B))^{4-1}$
Logo a função minimizada fica com 3 termos como mostra-se na Equação 4.3.

$$F(A, B) = [A^3 \star^1 B + A \star^2 B + A^1 \star^3 B]^3 \quad (4.3)$$

Tabela 4.4: Três aplicações do operador *Sucessor* em $F(A, B)$ da Tabela 4.3.

		$F^1(A, B)$			
$A \setminus B$	0	1	2	3	
0	0	0	0	0	
1	0	0	0	0	
2	0	1	2	3	
3	0	0	0	0	

4.3 Mapas de Karnaugh Estendido

O mapa de Karnaugh também é conhecido como diagramas de Veitch (mapa-K, K-map ou KV map) [35], e é uma ferramenta gráfica para a simplificação de funções lógicas Booleanas. Este método foi inventado no ano de 1952 por Edward W. Veitch e tem a particularidade que de célula a célula horizontalmente ou verticalmente mas não diagonalmente muda somente uma variável deixando em evidência os mintermos ou maxtermos que podem ser combinados, sendo útil para funções com menos de seis literais [41].

Inspirados no Mapa de Karnaugh da Álgebra de Chaveamento nesta seção é proposta uma extensão para MVL, utilizando-se de maneira similar os conceitos de agrupamento de células adjacentes¹.

Um mapa-KE é uma figura geométrica que contém uma célula correspondente para cada linha da tabela verdade. Assim, o mapa-KE tem tantas células quanto número de linhas tem a tabela verdade. Na Tabela 4.5 mostra-se o mapa-KE para dois literais de $N = 4$, onde $m_{0,0}$ corresponde à primeira combinação de literais da tabela verdade e $m_{3,3}$ é correspondente à última combinação dos literais da tabela verdade.

Tabela 4.5: Mapa Karnaugh Estendido para 2 literais de $N = 4$.

$A \setminus B$	0	1	2	3
0	$m_{0,0}$	$m_{0,1}$	$m_{0,2}$	$m_{0,3}$
1	$m_{1,0}$	$m_{1,1}$	$m_{1,2}$	$m_{1,3}$
2	$m_{2,0}$	$m_{2,1}$	$m_{2,2}$	$m_{2,3}$
3	$m_{3,0}$	$m_{3,1}$	$m_{3,2}$	$m_{3,3}$

Na Tabela 4.6 é mostrado o mapa-KE para três literais de $N = 4$, onde $m_{0,0,0}$ é correspondente à primeira combinação dos literais da tabela verdade e $m_{3,3,3}$ é correspondente à última combinação de literais da tabela verdade.

¹Células Adjacentes. Duas células (quadrículos) são chamados adjacentes se, e somente se eles são vizinhos horizontalmente ou verticalmente mas não diagonalmente.

Tabela 4.6: Mapa Karnaugh Estendido para 3 literais de $N = 4$.

$A = 0$					$A = 1$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	$m_{0,0,0}$	$m_{0,0,1}$	$m_{0,0,2}$	$m_{0,0,3}$	0	$m_{1,0,0}$	$m_{1,0,1}$	$m_{1,0,2}$	$m_{1,0,3}$
1	$m_{0,1,0}$	$m_{0,1,1}$	$m_{0,1,2}$	$m_{0,1,3}$	1	$m_{1,1,0}$	$m_{1,1,1}$	$m_{1,1,2}$	$m_{1,1,3}$
2	$m_{0,2,0}$	$m_{0,2,1}$	$m_{0,2,2}$	$m_{0,2,3}$	2	$m_{1,2,0}$	$m_{1,2,1}$	$m_{1,2,2}$	$m_{1,2,3}$
3	$m_{0,3,0}$	$m_{0,3,1}$	$m_{0,3,2}$	$m_{0,3,3}$	3	$m_{1,3,0}$	$m_{1,3,1}$	$m_{1,3,2}$	$m_{1,3,3}$

$A = 2$					$A = 3$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	$m_{2,0,0}$	$m_{2,0,0}$	$m_{2,0,2}$	$m_{2,0,3}$	0	$m_{3,0,0}$	$m_{3,0,1}$	$m_{3,0,2}$	$m_{3,0,3}$
1	$m_{2,1,0}$	$m_{2,1,0}$	$m_{2,1,2}$	$m_{2,1,3}$	1	$m_{3,1,0}$	$m_{3,1,1}$	$m_{3,1,2}$	$m_{3,1,3}$
2	$m_{2,2,0}$	$m_{2,2,0}$	$m_{2,2,2}$	$m_{2,2,3}$	2	$m_{3,2,0}$	$m_{3,2,1}$	$m_{3,2,2}$	$m_{3,2,3}$
3	$m_{2,3,0}$	$m_{2,3,0}$	$m_{2,3,2}$	$m_{2,3,3}$	3	$m_{3,3,0}$	$m_{3,3,1}$	$m_{3,3,2}$	$m_{3,3,3}$

O método Mapa de Karnaugh Estendido (Mapa-KE) para simplificação de funções MVL, consiste no agrupamento de regiões retangulares de N^T células (mintermos) onde T pode ser igual a $\{0, 1, 2, \dots\}$ tais regiões retangulares devem ter a largura igual a N^T . Sendo o objetivo formar o menor número de grupos possível, mas que cada mintermo seja parte de pelo menos um grupo.

A seguir apresenta-se alguns conceitos que são importantes no método do Mapa-KE e outros métodos de simplificação de funções MVL.

Implicante:

É todo termo Produto Estendido que pode representar um ou mais mintermos de uma função MVL.

Implicante Primo:

É um implicante que não forma parte de nenhum outro implicante na função MVL.

Implicante Essencial:

É um implicante primo que cobre pelo menos um mintermo que não forma parte de nenhum outro implicante primo.

O procedimento de simplificação por Mapas de Karnaugh Estendido para funções MVL na forma SOPE é o seguinte:

1. Separar em diferentes tabelas para cada subfunção $F_1(A, B, \dots, S)$, $F_2(A, B, \dots, S)$, ..., $F_L(A, B, \dots, S)$ considerando os mintermos de ordem inferior como 0 e os de ordem

superior como funções incompletas também conhecidas como *don't care*²(marcados com x).

2. Identificar todos os implicantes primos e os implicantes essenciais, procurando cobrir todos os mintermos com o menor número de regiões retangulares.
3. Gerar a subfunção MVL para cada tabela.
4. Aplicar o operador *Máximo* sobre todas as subfunções.

Exemplo 1: Simplificar a função lógica da Tabela 4.2, utilizando mapas-KE.

Tabela 4.7: Simplificação da função lógica da Tabela 4.2 utilizando mapas de Karnaugh Estendido.

$F_1(A, B)$				
$A \setminus B$	0	1	2	3
0	x	1	1	1
1	0	x	1	1
2	0	0	x	1
3	0	0	0	x

$F_3(A, B)$				
$A \setminus B$	0	1	2	3
0	3	0	0	0
1	0	3	0	0
2	0	0	3	0
3	0	0	0	3

Passo 1: Separação em diferentes tabelas uma para cada subfunção como mostrado na Tabela 4.7.

Passo 2: Identificar os implicantes primos e essenciais.

Para $F_1(A, B)$: $A \star^1 B^3$, $A^1 \star^1 A^1$ (formado pelos mintermos: $m_{0,0}^1, m_{0,1}^1, m_{0,2}^1$ e $m_{0,3}^1$) e $B^2 \star^1 B^2$ (formado pelos mintermos: $m_{0,3}^1, m_{1,3}^1, m_{2,3}^1$ e $m_{3,3}^1$).

Para $F_3(A, B)$: $A^3 \star^3 B^3$, $A^2 \star^3 B^2$, $A^1 \star^3 B^1$ e $A \star^3 B$.

Passo 3: Subfunções MVL.

$$F_1(A, B) = A^1 \star^1 A^1 + B^2 \star^1 B^2 + A \star^1 B^3$$

$$F_3(A, B) = A^3 \star^3 B^3 + A^2 \star^3 B^2 + A^1 \star^3 B^1 + A \star^3 B$$

Passo 4: Ao aplicar o operador *Máximo* sobre todas as subfunções obteve-se a Equação 4.4.

$$F(A, B) = A^1 \star^1 A^1 + B^2 \star^1 B^2 + A^3 \star^3 B^3 + A^2 \star^3 B^2 + A \star^1 B^3 + A^1 \star^3 B^1 + A \star^3 B \quad (4.4)$$

Exemplo 2: Realizar a simplificação da Tabela 4.8 utilizando mapas-KE.

²Funções incompletas. Em Algumas funções MVL não é possível especificar as saídas para determinadas combinações de entradas. Isto significa que tais combinações de entradas são irrelevantes para a saída.

Tabela 4.8: Exemplo de simplificação utilizando mapas de Karnaugh Estendido, $N = 4$.

	$A = 0$				$A = 1$					
	$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	0	1	2	1	2	0	1	2	1	2
1	0	0	0	0	0	1	0	0	0	0
2	3	3	3	3	3	2	1	2	1	2
3	0	0	0	0	0	3	0	0	0	0

	$A = 2$				$A = 3$					
	$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	0	1	2	1	2	0	1	2	1	2
1	2	2	2	2	2	1	0	0	0	0
2	0	2	0	2	2	2	0	2	0	2
3	2	2	2	2	2	3	3	3	3	3

Passo 1: Separação em diferentes tabelas uma para cada subfunção como mostrado na Figura 4.1.

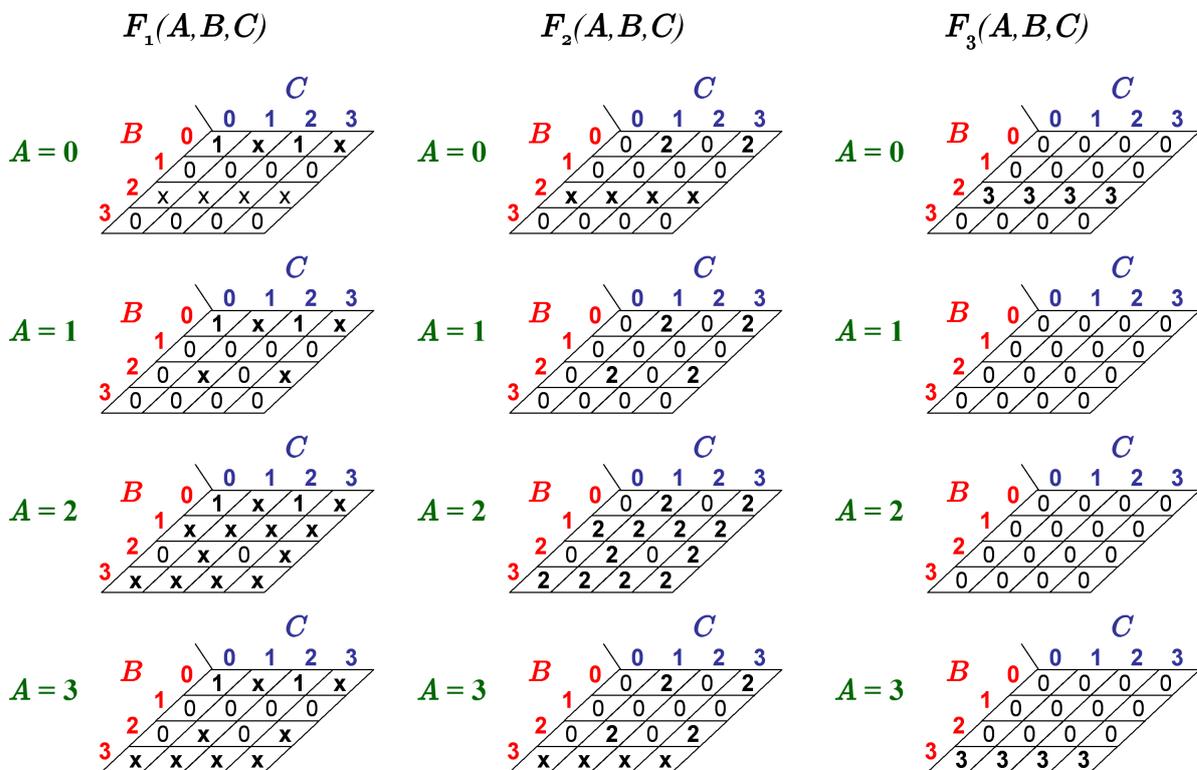


Figura 4.1: Separação da Tabela 4.8 em sub-tabelas para cada subfunção.

Passo 2: Identificar os implicantes primos e essenciais, como é mostrado na Figura 4.2, onde cada cor corresponde a um implicante primo ou essencial.

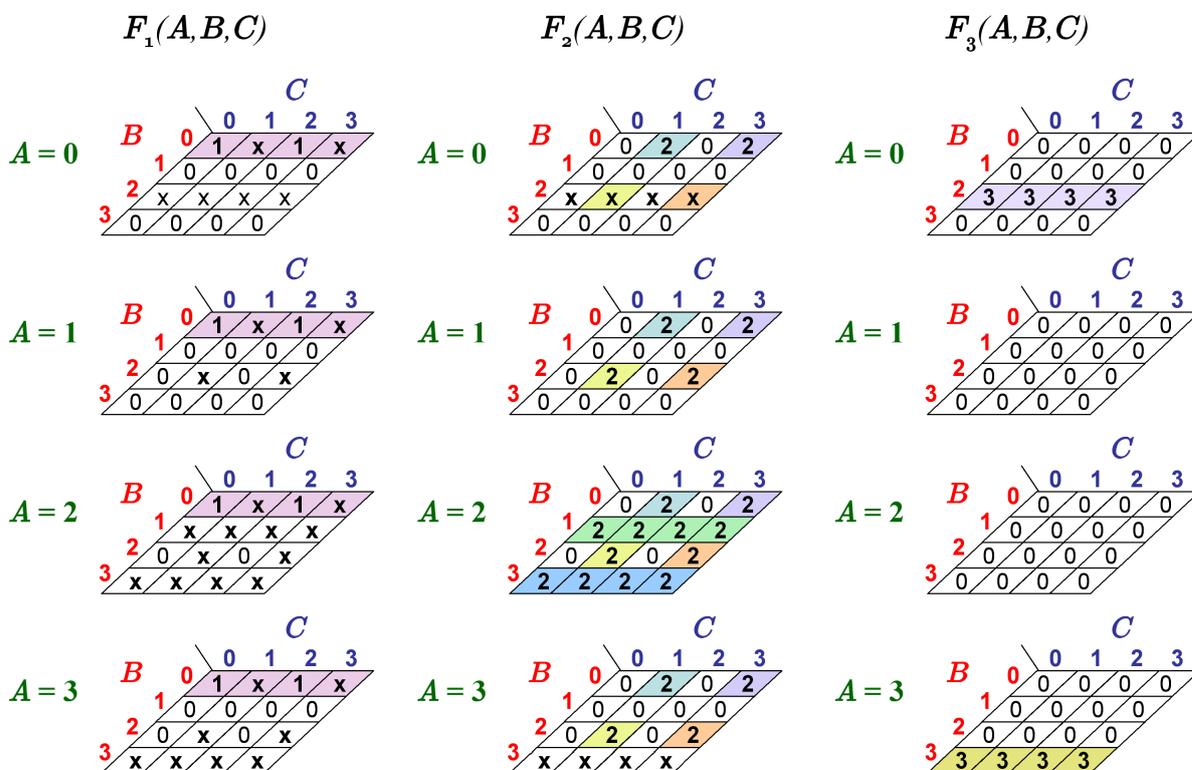


Figura 4.2: Agrupamento dos mintermos em regiões retangulares.

Para a subfunção $F_1(A, B, C)$:

Grupo 1, formado pelos mintermos: $m_{0,0,0}^1, m_{0,0,1}^1, m_{0,0,2}^1, m_{0,0,3}^1, m_{1,0,0}^1, m_{1,0,1}^1, m_{1,0,2}^1, m_{1,0,3}^1, m_{2,0,0}^1, m_{2,0,1}^1, m_{2,0,2}^1, m_{2,0,3}^1, m_{3,0,0}^1, m_{3,0,1}^1, m_{3,0,2}^1$ e $m_{3,0,3}^1 = B^1 \star^1 B^1$.

Para a subfunção $F_2(A, B, C)$:

Grupo 1, formado pelos mintermos: $m_{0,0,1}^2, m_{1,0,1}^2, m_{2,0,1}^2$ e $m_{3,0,1}^2 = B^2 \star^2 C^1$.

Grupo 2, formado pelos mintermos: $m_{0,0,3}^2, m_{1,0,3}^2, m_{2,0,3}^2$ e $m_{3,0,3}^2 = B^2 \star^2 C^3$.

Grupo 3, formado pelos mintermos: $m_{0,2,1}^x, m_{1,2,1}^2, m_{2,2,1}^2$ e $m_{3,2,1}^2 = B \star^2 C^1$.

Grupo 4, formado pelos mintermos: $m_{0,2,3}^x, m_{1,2,3}^2, m_{2,2,3}^2$ e $m_{3,2,3}^2 = B \star^2 C^3$.

Grupo 5, formado pelos mintermos: $m_{2,1,0}^2, m_{2,1,1}^2, m_{2,1,2}^2$ e $m_{2,1,3}^2 = A \star^2 B^1$.

Grupo 6, formado pelos mintermos: $m_{2,3,0}^2, m_{2,3,1}^2, m_{2,3,2}^2$ e $m_{2,3,3}^2 = A \star^2 B^3$.

Para a subfunção $F_3(A, B, C)$:

Grupo 1, formado pelos mintermos: $m_{0,2,0}^3, m_{0,2,1}^3, m_{0,2,2}^3$ e $m_{0,2,3}^3 = A^3 \star^3 B^1$.

Grupo 2, formado pelos mintermos: $m_{3,3,0}^3, m_{3,3,1}^3, m_{3,3,2}^3$ e $m_{3,3,3}^3 = A \star^3 B$.

Passo 3: Subfunções MVL.

Logo as subfunções de síntese para $F_1(A, B, C)$, $F_2(A, B, C)$ e $F_3(A, B, C)$ são mostradas nas Equações: 4.5, 4.6 e 4.7 respectivamente.

$$F_1(A, B, C) = B^1 \star^1 B^1 \quad (4.5)$$

$$F_2(A, B, C) = B^2 \star^2 C^1 + B^2 \star^2 C^3 + B \star^2 C^1 + B \star^2 C^3 + A \star^2 B^1 + A \star^2 B^3 \quad (4.6)$$

$$F_3(A, B, C) = A^3 \star^3 B^1 + A \star^3 B \quad (4.7)$$

Passo 4: Ao aplicar o operador *Máximo* sobre as equações: 4.5, 4.6 e 4.7 como é mostrado na Equação 4.8. Obteve-se a Equação 4.9.

$$F(A, B, C) = F_1(A, B, C) + F_2(A, B, C) + F_3(A, B, C) \quad (4.8)$$

$$F(A, B, C) = B^1 \star^1 B^1 + B^2 \star^2 C^1 + B^2 \star^2 C^3 + B \star^2 C^1 + B \star^2 C^3 + A \star^2 B^1 + A \star^2 B^3 + A^3 \star^3 B^1 + A \star^3 B \quad (4.9)$$

4.4 Quine McCluskey Estendido

Os mapas de Karnaugh Estendido apresentados na Seção 4.3 são utilizados para funções com menos de quatro literais. Se o número de literais for maior torna-se complicado identificar os implicantes primos e essenciais (regiões retangulares). Já o método de Quine McCluskey Estendido (QM-E) é um procedimento tabular e pode ser implementado em um programa de computador.

O método de Quine McCluskey Estendido, assim como o Mapa-KE (Seção 4.3), baseia-se no axioma de redução. Assim o método QM-E procura para cada mintermo os outros $\mathbf{N-1}$

mintermos faltantes para aplicar o axioma de redução de forma recursiva. O método QM-E têm os seguintes passos:

1. Separar em diferentes tabelas, uma para cada subfunção $F_1(A, B, \dots, S)$, $F_2(A, B, \dots, S)$, ..., $F_L(A, B, \dots, S)$ considerando os mintermos inferiores como 0 e os superiores como funções incompletas (marcados com x).
2. Listar os mintermos da subfunção na representação de números MVL.
3. Ordenar os mintermos da lista pela quantidade de literais diferentes zero.
4. Agrupar os mintermos pela quantidade de literais diferentes de zero em forma crescente. Assim o primeiro grupo estará formado por aqueles que tem o maior número de zeros.
5. Para cada mintermo do grupo \mathbf{g} ($\mathbf{g} = 1$ até a quantidade de literais), procurar no grupo $\mathbf{g} + 1$ os $\mathbf{N}-1$ mintermos que faltam para aplicar o axioma de redução. No caso de achar \mathbf{N} mintermos onde pode-se aplicar o axioma de redução, gera-se um novo termo e o literal que muda nos \mathbf{N} mintermos é substituído por '-', logo os \mathbf{N} mintermos são marcados.
6. Repetir o passo 5 para todos os grupos de mintermos na lista. O resultado obtido é a nova lista de termos.
7. Repetir os passos 4, 5 e 6 na nova lista de mintermos. Os termos que não formaram parte da aplicação do axioma de redução (não marcados) são os implicantes primos.
8. Fazer a tabela de implicantes primos e procurar o menor subconjunto que represente a função MVL.

Na tabela de implicantes primos escreve-se nas linhas os implicantes primos, e nas colunas os mintermos correspondentes.
9. Os passos 2 ao 8 são executados para cada uma das tabelas geradas (Correspondentes a cada subfunção).
10. Aplica-se o operador *Máximo* nas subfunções minimizadas resultantes F_1, F_2, \dots, F_L .

Exemplo 1: Minimizar a Tabela 4.9 de $\mathbf{N} = 4$, pelo método de Quine McCluskey Estendido.

Passo 1: Separação da Tabela 4.9 em \mathbf{L} sub-tabelas, como mostra-se na Tabela 4.10.

Passo 2: Listar os mintermos da subfunção $F_1(A, B, C)$ como é mostrado na Tabela 4.11.(a).

Passo 3: Ordenar os mintermos da subfunção $F_1(A, B, C)$ pela quantidade de zeros como é mostrado na Tabela 4.11.(b).

Passo 4: Agrupar os mintermos pela quantidade de zeros como é mostrado na Tabela 4.11.(c).

Tabela 4.9: Exemplo do método Quine McCluskey Estendido.

$A = 0$					$A = 1$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	1	2	1	2	0	1	2	1	2
1	0	0	0	0	1	0	0	0	0
2	3	3	3	3	2	1	2	1	2
3	0	0	0	0	3	0	0	0	0

$A = 2$					$A = 3$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	1	2	1	2	0	1	2	1	2
1	2	2	2	2	1	0	0	0	0
2	0	2	0	2	2	0	2	0	2
3	2	2	2	2	3	3	3	3	3

Tabela 4.10: Exemplo do método Quine McCluskey Estendido: Separação da Tabela 4.9 em sub-tabelas.

$F_1(A, B, C)$					$F_2(A, B, C)$					$F_3(A, B, C)$				
$A = 0$					$A = 0$					$A = 0$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	1	x	1	x	0	0	2	0	2	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
2	x	x	x	x	2	x	x	x	x	2	3	3	3	3
3	0	0	0	0	3	0	0	0	0	3	0	0	0	0

$A = 1$					$A = 1$					$A = 1$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	1	x	1	x	0	0	2	0	2	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
2	1	x	1	x	2	0	2	0	2	2	0	0	0	0
3	0	0	0	0	3	0	0	0	0	3	0	0	0	0

$A = 2$					$A = 2$					$A = 2$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	1	x	1	x	0	0	2	0	2	0	0	0	0	0
1	x	x	x	x	1	2	2	2	2	1	0	0	0	0
2	0	x	0	x	2	0	2	0	2	2	0	0	0	0
3	x	x	x	x	3	2	2	2	2	3	0	0	0	0

$A = 3$					$A = 3$					$A = 3$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	1	x	1	x	0	0	2	0	2	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
2	0	x	0	x	2	0	2	0	2	2	0	0	0	0
3	x	x	x	x	3	x	x	x	x	3	3	3	3	3

Tabela 4.11: Procedimento do método Quine McCluskey Estendido para $F_1(A, B, C)$ da Tabela 4.10.

(a)			(b)			(c)		
$m_{0,0,0}$	000	1	$m_{0,0,0}$	000	1	$m_{0,0,0}$	000	1
$m_{0,0,1}$	001	x	$m_{0,0,1}$	001	x	$m_{0,0,1}$	001	x
$m_{0,0,2}$	002	1	$m_{0,0,2}$	002	1	$m_{0,0,2}$	002	1
$m_{0,0,3}$	003	x	$m_{0,0,3}$	003	x	$m_{0,0,3}$	003	x
$m_{0,2,0}$	020	x	$m_{0,2,0}$	020	x	$m_{0,2,0}$	020	x
$m_{0,2,1}$	021	x	$m_{1,0,0}$	100	x	$m_{1,0,0}$	100	x
$m_{0,2,2}$	022	x	$m_{2,0,0}$	200	x	$m_{2,0,0}$	200	x
$m_{0,2,3}$	023	x	$m_{3,0,0}$	300	1	$m_{3,0,0}$	300	1
$m_{1,0,0}$	100	1	$m_{0,2,1}$	021	1	$m_{0,2,1}$	021	1
$m_{1,0,1}$	101	x	$m_{0,2,2}$	022	1	$m_{0,2,2}$	022	1
$m_{1,0,2}$	102	1	$m_{0,2,3}$	023	x	$m_{0,2,3}$	023	x
$m_{1,0,3}$	103	x	$m_{1,0,1}$	101	x	$m_{1,0,1}$	101	x
$m_{1,2,1}$	121	x	$m_{1,0,2}$	102	1	$m_{1,0,2}$	102	1
$m_{1,2,3}$	123	x	$m_{1,0,3}$	103	x	$m_{1,0,3}$	103	x
$m_{2,0,0}$	200	1	$m_{2,0,1}$	201	x	$m_{2,0,1}$	201	x
$m_{2,0,1}$	201	x	$m_{2,0,2}$	202	1	$m_{2,0,2}$	202	1
$m_{2,0,2}$	202	1	$m_{2,0,3}$	203	x	$m_{2,0,3}$	203	x
$m_{2,0,3}$	203	x	$m_{2,1,0}$	210	x	$m_{2,1,0}$	210	x
$m_{2,1,0}$	210	x	$m_{2,3,0}$	230	x	$m_{2,3,0}$	230	x
$m_{2,1,1}$	211	x	$m_{3,0,1}$	301	x	$m_{3,0,1}$	301	x
$m_{2,1,2}$	212	x	$m_{3,0,2}$	302	1	$m_{3,0,2}$	302	1
$m_{2,1,3}$	213	x	$m_{3,0,3}$	303	x	$m_{3,0,3}$	303	x
$m_{2,2,1}$	221	x	$m_{3,3,0}$	330	x	$m_{3,3,0}$	330	x
$m_{2,2,3}$	223	x	$m_{1,2,1}$	121	x	$m_{1,2,1}$	121	x
$m_{2,3,0}$	230	x	$m_{1,2,3}$	123	x	$m_{1,2,3}$	123	x
$m_{2,3,1}$	231	x	$m_{2,1,1}$	211	x	$m_{2,1,1}$	211	x
$m_{2,3,2}$	232	x	$m_{2,1,2}$	212	x	$m_{2,1,2}$	212	x
$m_{2,3,3}$	233	x	$m_{2,1,3}$	213	x	$m_{2,1,3}$	213	x
$m_{3,0,0}$	300	1	$m_{2,2,1}$	221	x	$m_{2,2,1}$	221	x
$m_{3,0,1}$	301	x	$m_{2,2,3}$	223	x	$m_{2,2,3}$	223	x
$m_{3,0,2}$	302	1	$m_{2,3,1}$	231	x	$m_{2,3,1}$	231	x
$m_{3,0,3}$	303	x	$m_{2,3,2}$	232	x	$m_{2,3,2}$	232	x
$m_{3,2,1}$	321	x	$m_{2,3,3}$	233	x	$m_{2,3,3}$	233	x
$m_{3,2,3}$	323	x	$m_{3,2,1}$	321	x	$m_{3,2,1}$	321	x
$m_{3,3,0}$	330	x	$m_{3,2,3}$	323	x	$m_{3,2,3}$	323	x
$m_{3,3,1}$	331	x	$m_{3,3,1}$	331	x	$m_{3,3,1}$	331	x
$m_{3,3,2}$	332	x	$m_{3,3,2}$	332	x	$m_{3,3,2}$	332	x
$m_{3,3,3}$	333	x	$m_{3,3,3}$	333	x	$m_{3,3,3}$	333	x

Passo 5: Para cada mintermo do grupo $g = 1$ procura-se mintermos no grupo $g = 2$ nos quais seja possível aplicar o axioma de redução. Por exemplo, para o mintermo da linha 1 $m_{0,0,0}$ que pertence ao grupo 1, procuramos no grupo 2 os mintermos: $m_{0,0,1}$, $m_{0,0,2}$, $m_{0,0,3}$ ou $m_{0,1,0}$, $m_{0,2,0}$, $m_{0,3,0}$ ou $m_{1,0,0}$, $m_{2,0,0}$, $m_{3,0,0}$. Veja que só nos três casos é possível aplicar o axioma de redução.

Passo 6: Repete-se o passo 5 para os grupos 2, 3 e 4 como é mostrado na Tabela 4.13.

Passo 7: Repete-se os passos 4, 5 e 6 para a lista obtida nas Tabelas 4.12.(b) e 4.13.(b), logo obteve-se o seguinte:

Tabela 4.12: Passo 5 do método Quine McCluskey Estendido.

Linha	Grupo(g)	(a)	(b)
✓ 1	1	m _{0,0,0} 0 0 0 1	m _{0,0,0} , m _{0,0,1} , m _{0,0,2} , m _{0,0,3} 0 0 - 1
✓ 2	2	m _{0,0,1} 0 0 1 x	m _{0,0,0} , m _{1,0,0} , m _{2,0,0} , m _{3,0,0} - 0 0 1
✓ 3	2	m _{0,0,2} 0 0 2 1	
✓ 4	2	m _{0,0,3} 0 0 3 x	
5	2	m _{0,2,0} 0 2 0 x	
✓ 6	2	m _{1,0,0} 1 0 0 x	
✓ 7	2	m _{2,0,0} 2 0 0 x	
✓ 8	2	m _{3,0,0} 3 0 0 1	

Tabela 4.13: Passo 6 do método Quine McCluskey Estendido.

Linha	Grupo(g)	(a)	(b)
✓ 2	2	m _{0,0,1} 0 0 1 x	m _{0,0,1} , m _{1,0,1} , m _{2,0,1} , m _{3,0,1} - 0 1 x
✓ 3	2	m _{0,0,2} 0 0 2 1	m _{0,0,2} , m _{1,0,2} , m _{2,0,2} , m _{3,0,2} - 0 2 1
✓ 4	2	m _{0,0,3} 0 0 3 x	m _{0,0,3} , m _{1,0,3} , m _{2,0,3} , m _{3,0,3} - 0 3 x
✓ 5	2	m _{0,2,0} 0 2 0 x	m _{0,2,0} , m _{0,2,1} , m _{0,2,2} , m _{0,2,3} 0 2 - x
✓ 6	2	m _{1,0,0} 1 0 0 x	m _{1,0,0} , m _{1,0,1} , m _{1,0,2} , m _{1,0,3} 1 0 - 1
✓ 7	2	m _{2,0,0} 2 0 0 x	m _{2,0,0} , m _{2,0,1} , m _{2,0,2} , m _{2,0,3} 2 0 - 1
✓ 8	2	m _{3,0,0} 3 0 0 1	m _{3,0,0} , m _{3,0,1} , m _{3,0,2} , m _{3,0,3} 3 0 - 1
✓ 9	3	m _{0,2,1} 0 2 1 1	m _{0,2,1} , m _{1,2,1} , m _{2,2,1} , m _{3,2,1} - 2 1 x
✓ 10	3	m _{0,2,2} 0 2 2 1	m _{0,2,2} , m _{1,2,2} , m _{2,2,2} , m _{3,2,2} - 2 2 x
✓ 11	3	m _{0,2,3} 0 2 3 x	m _{0,2,3} , m _{1,2,3} , m _{2,2,3} , m _{3,2,3} - 2 3 x
✓ 12	3	m _{1,0,1} 1 0 1 x	m _{2,0,1} , m _{2,1,1} , m _{2,2,1} , m _{2,3,1} 2 - 1 x
✓ 13	3	m _{1,0,2} 1 0 2 1	m _{2,0,2} , m _{2,1,2} , m _{2,2,2} , m _{2,3,2} 2 - 2 x
✓ 14	3	m _{1,0,3} 1 0 3 x	m _{2,0,3} , m _{2,1,3} , m _{2,2,3} , m _{2,3,3} 2 - 3 x
✓ 15	3	m _{2,0,1} 2 0 1 x	m _{2,1,0} , m _{2,1,1} , m _{2,1,2} , m _{2,1,3} 2 1 - x
✓ 16	3	m _{2,0,2} 2 0 2 1	m _{2,3,0} , m _{2,3,1} , m _{2,3,2} , m _{2,3,3} 2 3 - x
✓ 17	3	m _{2,0,3} 2 0 3 x	m _{3,3,0} , m _{3,3,1} , m _{3,3,2} , m _{3,3,3} 3 3 - x
✓ 18	3	m _{2,1,0} 2 1 0 x	
✓ 19	3	m _{2,3,0} 2 3 0 x	
✓ 20	3	m _{3,0,1} 3 0 1 x	
✓ 21	3	m _{3,0,2} 3 0 2 1	
✓ 22	3	m _{3,0,3} 3 0 3 x	
✓ 23	3	m _{3,3,0} 3 3 0 x	
✓ 24	4	m _{1,2,1} 1 2 1 x	
✓ 25	4	m _{1,2,3} 1 2 3 x	
✓ 26	4	m _{2,1,1} 2 1 1 x	
✓ 27	4	m _{2,1,2} 2 1 2 x	
✓ 28	4	m _{2,1,3} 2 1 3 x	
✓ 29	4	m _{2,2,1} 2 2 1 x	
✓ 30	4	m _{2,2,3} 2 2 3 x	
✓ 31	4	m _{2,3,1} 2 3 1 x	
✓ 32	4	m _{2,3,2} 2 3 2 x	
✓ 33	4	m _{2,3,3} 2 3 3 x	
34	4	m _{3,2,1} 3 2 1 x	
35	4	m _{3,2,3} 3 2 3 x	
36	4	m _{3,3,1} 3 3 1 x	
37	4	m _{3,3,2} 3 3 2 x	
38	4	m _{3,3,3} 3 3 3 x	

Passo 8: Construção da tabela de implicantes primos obtida a partir dos mintermos encontrados no Passo 7, como é mostrado na Figura 4.3.

$m_{0,0,0}, m_{0,0,1}, m_{0,0,2}, m_{0,0,3},$
 $m_{1,0,0}, m_{1,0,1}, m_{1,0,2}, m_{1,0,3},$
 $m_{2,0,0}, m_{2,0,1}, m_{2,0,2}, m_{2,0,3},$
 $m_{3,0,0}, m_{3,0,1}, m_{3,0,2}, m_{3,0,3} \quad - 0 - 1$

	$m_{0,0,0}^1$	$m_{0,0,1}^1$	$m_{0,0,2}^1$	$m_{0,0,3}^1$	$m_{1,0,0}^1$	$m_{1,0,1}^1$	$m_{1,0,2}^1$	$m_{1,0,3}^1$	$m_{2,0,0}^1$	$m_{2,0,1}^1$	$m_{2,0,2}^1$	$m_{2,0,3}^1$	$m_{3,0,0}^1$	$m_{3,0,1}^1$	$m_{3,0,2}^1$	$m_{3,0,3}^1$
IP1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Figura 4.3: Tabela de implicantes primos de $F_1(A, B, C)$.

A Equação 4.10 é obtida da Tabela 4.11.

$$F_1(A, B, C) = B^1 \star^1 B^1 \tag{4.10}$$

Passo 9: Repete-se os passos 2 ao 8 para as subfunções $F_2(A, B, C)$ e $F_3(A, B, C)$.

Na Tabela 4.14 é mostrado o procedimento para obter os implicantes primos da sub-tabela $F_2(A, B, C)$ mostrada na Tabela 4.10.

	$m_{0,0,1}^2$	$m_{0,0,3}^2$	$m_{0,2,1}^2$	$m_{0,2,3}^2$	$m_{1,0,1}^2$	$m_{1,0,3}^2$	$m_{1,2,1}^2$	$m_{1,2,3}^2$	$m_{2,0,1}^2$	$m_{2,0,3}^2$	$m_{2,1,0}^2$	$m_{2,1,1}^2$	$m_{2,1,2}^2$	$m_{2,1,3}^2$	$m_{2,2,1}^2$	$m_{2,2,3}^2$	$m_{2,3,0}^2$	$m_{2,3,1}^2$	$m_{2,3,2}^2$	$m_{2,3,3}^2$	$m_{3,0,1}^2$	$m_{3,0,3}^2$	$m_{3,2,1}^2$	$m_{3,2,3}^2$	
IP1	x				x				x																
IP2	x				x				x																
IP3		x				x								x									x		
IP4			x				x								x										x
IP5								x							x										
IP6									x																
IP7										x	x	x	x												
IP8																	x	x	x	x					

Figura 4.4: Tabela de implicantes primos de $F_2(A, B, C)$.

A Equação 4.11 é obtida da Tabela 4.14.

$$F_2(A, B, C) = B^2 \star^2 C^1 + B^2 \star^2 C^3 + B \star^2 C^1 + B \star^2 C^3 + A \star^2 B^1 + A \star^2 B^3 \tag{4.11}$$

Na Tabela 4.15 é mostrado o procedimento para obter os implicantes primos da sub-tabela $F_3(A, B, C)$ mostrada na Tabela 4.10.

Tabela 4.14: Procedimento do método Quine McCluskey Estendido para $F_2(A, B, C)$ da Tabela 4.10.

(a)		(b)		(c)	
$m_{0,0,1}$	0 0 1 2	$m_{0,0,1}$	0 0 1 2	$m_{0,0,1}, m_{1,0,1}, m_{2,0,1}, m_{3,0,1}$	- 0 1 2
$m_{0,0,3}$	0 0 3 2	$m_{0,0,3}$	0 0 3 2	$m_{0,0,3}, m_{1,0,3}, m_{2,0,3}, m_{3,0,3}$	- 0 3 2
$m_{0,2,0}$	0 2 0 x	$m_{0,2,0}$	0 2 0 x	$m_{0,2,0}, m_{0,2,1}, m_{0,2,2}, m_{0,2,3}$	0 2 - x
$m_{0,2,1}$	0 2 1 x	$m_{0,2,1}$	0 2 1 x	$m_{0,2,1}, m_{1,2,1}, m_{2,2,1}, m_{3,2,1}$	- 2 1 2
$m_{0,2,2}$	0 2 2 x	$m_{0,2,2}$	0 2 2 x	$m_{0,2,3}, m_{1,2,3}, m_{2,2,3}, m_{3,2,3}$	- 2 3 2
$m_{0,2,3}$	0 2 3 x	$m_{0,2,3}$	0 2 3 x	$m_{2,0,1}, m_{2,1,1}, m_{2,2,1}, m_{2,3,1}$	2 - 1 2
$m_{1,0,1}$	1 0 1 2	$m_{1,0,1}$	1 0 1 2	$m_{2,0,3}, m_{2,1,3}, m_{2,2,3}, m_{2,3,3}$	2 - 3 2
$m_{1,0,3}$	1 0 3 2	$m_{1,0,3}$	1 0 3 2	$m_{2,1,0}, m_{2,1,1}, m_{2,1,2}, m_{2,1,3}$	2 1 - 2
$m_{1,2,1}$	1 2 1 2	$m_{2,0,1}$	2 0 1 2	$m_{2,3,0}, m_{2,3,1}, m_{2,3,2}, m_{2,3,3}$	2 3 - 2
$m_{1,2,3}$	1 2 3 2	$m_{2,0,3}$	2 0 3 2	$m_{3,3,0}, m_{3,3,1}, m_{3,3,2}, m_{3,3,3}$	3 3 - x
$m_{2,0,1}$	2 0 1 2	$m_{2,1,0}$	2 1 0 2		
$m_{2,0,3}$	2 0 3 2	$m_{2,3,0}$	2 3 0 2		
$m_{2,1,0}$	2 1 0 2	$m_{3,0,1}$	3 0 1 2		
$m_{2,1,1}$	2 1 1 2	$m_{3,0,3}$	3 0 3 2		
$m_{2,1,2}$	2 1 2 2	$m_{3,3,0}$	3 3 0 x		
$m_{2,1,3}$	2 1 3 2	$m_{1,2,1}$	1 2 1 2		
$m_{2,2,1}$	2 2 1 2	$m_{1,2,3}$	1 2 3 2		
$m_{2,2,3}$	2 2 3 2	$m_{2,1,1}$	2 1 1 2		
$m_{2,3,0}$	2 3 0 2	$m_{2,1,2}$	2 1 2 2		
$m_{2,3,1}$	2 3 1 2	$m_{2,1,3}$	2 1 3 2		
$m_{2,3,2}$	2 3 2 2	$m_{2,2,1}$	2 2 1 2		
$m_{2,3,3}$	2 3 3 2	$m_{2,2,3}$	2 2 3 2		
$m_{3,0,1}$	3 0 1 2	$m_{2,3,1}$	2 3 1 2		
$m_{3,0,3}$	3 0 3 2	$m_{2,3,2}$	2 3 2 2		
$m_{3,2,1}$	3 2 1 2	$m_{2,3,3}$	2 3 3 2		
$m_{3,2,3}$	3 2 3 2	$m_{3,2,1}$	3 2 1 2		
$m_{3,3,0}$	3 3 0 x	$m_{3,2,3}$	3 2 3 2		
$m_{3,3,1}$	3 3 1 x	$m_{3,3,1}$	3 3 1 x		
$m_{3,3,2}$	3 3 2 x	$m_{3,3,2}$	3 3 2 x		
$m_{3,3,3}$	3 3 3 x	$m_{3,3,3}$	3 3 3 x		

$m_{0,0,1}, m_{1,0,1}, m_{2,0,1}, m_{3,0,1}$	- 0 1 2
$m_{0,0,3}, m_{1,0,3}, m_{2,0,3}, m_{3,0,3}$	- 0 3 2
$m_{0,2,1}, m_{1,2,1}, m_{2,2,1}, m_{3,2,1}$	- 2 1 2
$m_{0,2,3}, m_{1,2,3}, m_{2,2,3}, m_{3,2,3}$	- 2 3 2
$m_{2,0,1}, m_{2,1,1}, m_{2,2,1}, m_{2,3,1}$	2 - 1 2
$m_{2,0,3}, m_{2,1,3}, m_{2,2,3}, m_{2,3,3}$	2 - 3 2
$m_{2,1,0}, m_{2,1,1}, m_{2,1,2}, m_{2,1,3}$	2 1 - 2
$m_{2,3,0}, m_{2,3,1}, m_{2,3,2}, m_{2,3,3}$	2 3 - 2

A Equação 4.12 é obtida da Tabela 4.15.

$$F_3(A, B, C) = A^3 \star^3 B^1 + A \star^3 B \tag{4.12}$$

Tabela 4.15: Procedimento do método Quine McCluskey Estendido para $F_3(A, B, C)$ da Tabela 4.10.

	(a)		(b)		(c)
$m_{0,2,0}$	0 2 0 3	$m_{0,2,0}$	0 2 0 3	$m_{0,2,0}, m_{0,2,1}, m_{0,2,2}, m_{0,2,3}$	0 2 - 3
$m_{0,2,1}$	0 2 1 3	$m_{0,2,1}$	0 2 1 3	$m_{3,3,0}, m_{3,3,1}, m_{3,3,2}, m_{3,3,3}$	3 3 - 3
$m_{0,2,2}$	0 2 2 3	$m_{0,2,2}$	0 2 2 3		
$m_{0,2,3}$	0 2 3 3	$m_{0,2,3}$	0 2 3 3		
$m_{3,3,0}$	3 3 0 3	$m_{3,3,0}$	3 3 0 3		
$m_{3,3,1}$	3 3 1 3	$m_{3,3,1}$	3 3 1 3		
$m_{3,3,2}$	3 3 2 3	$m_{3,3,2}$	3 3 2 3		
$m_{3,3,3}$	3 3 3 3	$m_{3,3,3}$	3 3 3 3		

	$m_{0,2,0}^3$	$m_{0,2,1}^3$	$m_{0,2,2}^3$	$m_{0,2,3}^3$	$m_{3,3,0}^3$	$m_{3,3,1}^3$	$m_{3,3,2}^3$	$m_{3,3,3}^3$
IP1	✗	✗	✗	✗				
IP2					✗	✗	✗	✗

Figura 4.5: Tabela de implicantes primos de $F_3(A, B, C)$.

Passo 10: Aplica-se o operador *Máximo* as Equações 4.10, 4.11 e 4.12.

A função minimizada pelo método QM-E da Tabela 4.9 é mostrada na Equação 4.13.

$$\begin{aligned}
 F(A, B, C) = & B^1 \star^1 B^1 + B^2 \star^2 C^1 + B^2 \star^2 C^3 + B \star^2 C^1 + B \star^2 C^3 + \\
 & A \star^2 B^1 + A \star^2 B^3 + A^3 \star^3 B^1 + A \star^3 B
 \end{aligned}
 \tag{4.13}$$

4.5 Algoritmo Petrick Estendido

Como mostrado na Seção 4.4, para obter a função minimizada é preciso cobrir todos os implicantes na tabela de implicantes primos, sendo necessário para isso, uma heurística a qual não garante uma solução ótima. Na Álgebra de Chaveamento o método que garante a função ótima e com menor custo computacional é o algoritmo Petrick [35], o qual gera todas as possíveis combinações de uma função. Na álgebra MVL aqui proposta o algoritmo Petrick tem somente uma modificação, sendo esta no Passo 1, que é trocar o algoritmo de Quine McCluskey pelo algoritmo Quine McCluskey Estendido na busca dos implicantes primos da função MVL, como foi apresentado na Seção 4.4, nos Passos 2 ao 5 utiliza-se os conceitos da Álgebra de Chaveamento.

Algoritmo Petrick Estendido:

1. Usar o Algoritmo do método Quine McCluskey Estendido para encontrar os implicantes primos da função.

2. Criar as tabelas de implicants primos. Identificar e remover todos os implicants essenciais como no método Quine McCluskey.
3. Escrever expressões na forma de produto de operações soma representando todos os implicants primos.
4. Converter a expressão de produto de operações soma à forma soma de operações produto usando as leis da distributividade e simplificar a expressão usando involução e absorção para remover os termos redundantes até obter a soma de operações produto.
5. Selecionar o produto com menor número de implicants primos.
6. Repetir o mesmo procedimento para todas as tabelas de implicants primos de uma função MVL.

Exemplo 1: Minimizar a Tabela 4.16 de $N = 4$, pelo método de Petrick Estendido.

Tabela 4.16: Exemplo do método Petrick Estendido.

	$A = 0$				$A = 1$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	0	2	0	2	0	0	2	0	2
1	0	0	0	0	1	0	0	0	0
2	0	2	0	2	2	0	2	0	2
3	0	0	0	0	3	0	0	0	0
	$A = 2$				$A = 3$				
$B \setminus C$	0	1	2	3	$B \setminus C$	0	1	2	3
0	0	2	0	2	0	0	2	0	2
1	2	2	2	2	1	0	0	0	0
2	0	2	0	2	2	0	2	0	2
3	2	2	2	2	3	0	0	0	0

Passo 1: Utilizar o método de Quine McCluskey Estendido para encontrar os implicants primos.

Passo 2: Construir a tabela de implicants primos como é mostrado na Figura 4.6.

Passo 3: Escrever as expressões na forma de produto de operações soma. Logo, continua-se neste passo e nos próximos da mesma forma que na Álgebra de Chaveamento.

$$\begin{aligned}
 F_2(A, B, C) = & (IP_1)(IP_2)(IP_3)(IP_4)(IP_1)(IP_2)(IP_3)(IP_4)(IP_1 + IP_5)(IP_2 + IP_6) \\
 & (IP_7)(IP_5 + IP_7)(IP_7)(IP_6 + IP_7)(IP_3 + IP_5)(IP_4 + IP_6)(IP_8) \\
 & (IP_5 + IP_8)(IP_8)(IP_6 + IP_8)(IP_1)(IP_2)(IP_3)(IP_4) \quad (4.14)
 \end{aligned}$$

Passo 4: Converter o produto de operações soma, em soma de operações soma utilizando as propriedades da Álgebra de Chaveamento como mostra-se a seguir.

	$m^2_{0,0,1}$	$m^2_{0,0,3}$	$m^2_{0,2,1}$	$m^2_{0,2,3}$	$m^2_{1,0,1}$	$m^2_{1,0,3}$	$m^2_{1,2,1}$	$m^2_{1,2,3}$	$m^2_{2,0,1}$	$m^2_{2,0,3}$	$m^2_{2,1,0}$	$m^2_{2,1,1}$	$m^2_{2,1,2}$	$m^2_{2,1,3}$	$m^2_{2,2,1}$	$m^2_{2,2,3}$	$m^2_{2,3,0}$	$m^2_{2,3,1}$	$m^2_{2,3,2}$	$m^2_{2,3,3}$	$m^2_{3,0,1}$	$m^2_{3,0,3}$	$m^2_{3,2,1}$	$m^2_{3,2,3}$
IP1	X				X				X													X		
IP2		X				X				X													X	
IP3			X				X								X									X
IP4				X				X								X								X
IP5									X			X			X			X						
IP6										X				X		X						X		
IP7											X	X	X	X										
IP8																	X	X	X	X				

Figura 4.6: Tabela de implicantes primos de $F_2(A, B, C)$.

$$\begin{aligned}
 F_2(A, B, C) &= (IP_1)(IP_2)(IP_3)(IP_4)(IP_1)(IP_2)(IP_3)(IP_4)(IP_1 + IP_5)(IP_2 + IP_6) \\
 &\quad (IP_7)(IP_5 + IP_7)(IP_7)(IP_6 + IP_7)(IP_3 + IP_5)(IP_4 + IP_6)(IP_8) \\
 &\quad (IP_5 + IP_8)(IP_8)(IP_6 + IP_8)(IP_1)(IP_2)(IP_3)(IP_4) \tag{4.15}
 \end{aligned}$$

$$\begin{aligned}
 F_2(A, B, C) &= (IP_1)(IP_2)(IP_3)(IP_4)(IP_1 + IP_5)(IP_2 + IP_6)(IP_7)(IP_5 + IP_7) \\
 &\quad (IP_6 + IP_7)(IP_3 + IP_5)(IP_4 + IP_6)(IP_8)(IP_5 + IP_8) \\
 &\quad (IP_6 + IP_8)(\text{Axioma booleano de idempotência}) \tag{4.16}
 \end{aligned}$$

$$\begin{aligned}
 F_2(A, B, C) &= (IP_1 IP_2 IP_3 IP_4 IP_7 IP_8)(IP_5 + IP_1)(IP_5 + IP_7)(IP_5 + IP_3) \\
 &\quad (IP_5 + IP_8)(IP_6 + IP_2)(IP_6 + IP_7)(IP_6 + IP_4)(IP_6 + IP_8) \\
 &\quad (\text{Axioma booleano de comutatividade}) \tag{4.17}
 \end{aligned}$$

$$\begin{aligned}
 F_2(A, B, C) &= (IP_1 IP_2 IP_3 IP_4 IP_7 IP_8)(IP_5 + IP_1 IP_3 IP_7 IP_8) \\
 &\quad (IP_6 + IP_2 IP_4 IP_7 IP_8) \\
 &\quad (\text{Axioma booleano de distributividade}) \tag{4.18}
 \end{aligned}$$

$$\begin{aligned}
 F_2(A, B, C) &= IP_1 IP_2 IP_3 IP_4 IP_5 IP_7 IP_8 + IP_1 IP_2 IP_3 IP_4 IP_7 IP_8 + \\
 &\quad IP_1 IP_2 IP_3 IP_4 IP_6 IP_7 IP_8 + IP_1 IP_2 IP_3 IP_4 IP_7 IP_8 \\
 &\quad (\text{Axioma booleano de distributividade}) \tag{4.19}
 \end{aligned}$$

$$\begin{aligned}
F_2(A, B, C) &= IP_1 IP_2 IP_3 IP_4 IP_5 IP_7 IP_8 + IP_1 IP_2 IP_3 IP_4 IP_7 IP_8 + \\
&IP_1 IP_2 IP_3 IP_4 IP_6 IP_7 IP_8 \\
&\text{(Axioma booleano de idempotência)}
\end{aligned} \tag{4.20}$$

Passo 5: Escolher o produto com o menor número de implicantes primos. Logo, a função mínima, que representa a Figura 4.6 é formada pela soma dos implicantes primos: $IP_1 IP_2 IP_3 IP_4 IP_7 IP_8$.

4.6 Métodos de Minimização na Forma POSE e SOPE

Os métodos de minimização na forma POSE e SOPE são formados pela utilização simultânea dos dois conjuntos universais de operadores.

4.6.1 Minimização Simétrica

O método de minimização simétrica é aplicável numa função $F(A, B)$ se, para cada mintermo $m_{A,B}^i$ existe o mintermo $m_{B,A}^i$. Logo, a função $F(A, B)$ é equivalente a $F(\text{Máximo}(A, B), \text{Mínimo}(A, B))$.

Exemplo 1: Minimizar a Tabela 4.17 pelo método simétrico.

Na Equação 4.21 é mostrada a função de síntese da Tabela 4.17 (a).

$$\begin{aligned}
F(A, B) &= A^3 \star^3 B^3 + A^2 \star^2 B^1 + A^1 \star^1 B^3 + A^1 \star^2 B^2 + A^2 \star^3 B^2 + \\
&A^1 \star^2 B + A \star^1 B^2 + A^3 \star^1 B^1 + A \star^2 B^1 + A^1 \star^3 B^1 + \\
&A \star^2 B^3 + A^2 \star^1 B + A^3 \star^2 B + A \star^3 B
\end{aligned} \tag{4.21}$$

Pelo axioma de comutatividade foi obtida a Equação 4.22.

$$\begin{aligned}
F(A, B) &= \underbrace{A^3 \star^3 B^3}_{m_{0,0}^3} + \underbrace{A^2 \star^2 B^1}_{m_{0,1}^2} + \underbrace{A^1 \star^2 B^2}_{m_{1,0}^2} + \underbrace{A^1 \star^1 B^3}_{m_{0,2}^1} + \underbrace{A^3 \star^1 B^1}_{m_{2,0}^1} + \\
&\underbrace{A^1 \star^2 B}_{m_{1,2}^2} + \underbrace{A \star^2 B^1}_{m_{2,1}^2} + \underbrace{A^2 \star^3 B^2}_{m_{1,1}^3} + \underbrace{A \star^1 B^2}_{m_{1,3}^1} + \underbrace{A^2 \star^1 B}_{m_{3,1}^1} + \\
&\underbrace{A^1 \star^3 B^1}_{m_{3,2}^3} + \underbrace{A \star^2 B^3}_{m_{2,3}^2} + \underbrace{A^3 \star^2 B}_{m_{3,2}^2} + \underbrace{A \star^3 B}_{m_{3,3}^3}
\end{aligned} \tag{4.22}$$

Na Equação 4.22 a linha abaixo dos mintermos mostra que tais mintermos cumprem a condição para a minimização simétrica. Logo a tabela reduzida é mostrada na Tabela 4.17 (b) com seis mintermos. E para o melhor entendimento foi feita a seguinte igualdade $C = A + B$ e $D = A \cdot B$.

Tabela 4.17: Exemplo de minimização simétrica.

$A \setminus B$	0	1	2	3
0	3	2	1	0
1	2	3	2	1
2	1	2	3	2
3	0	1	2	3

$C \setminus D$	0	1	2	3
0	3			
1	2	3		
2	1	2	3	
3	0	1	2	3

Exemplo 2: Minimizar a Meio Somador projetado na Seção 3.2 pelo método simétrico. E para o melhor entendimento foi feita a seguinte igualdade $C = A + B$ e $D = A \cdot B$.

Tabela 4.18: Tabela do meio somador MVL de $N = 4$.

$A \setminus B$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$C \setminus D$	0	1	2	3
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1

Como as Tabelas verdade de Q_1 e C_{out} mostradas na Tabela 4.18 têm simetria obteve-se a Tabela 4.19 e para o melhor entendimento foi feita a seguinte igualdade $C = A + B$ e $D = A \cdot B$.

Tabela 4.19: Tabela Simplificada do meio somador MVL de $N = 4$.

$C \setminus D$	0	1	2	3
0	0			
1	1	2		
2	2	3	0	
3	3	0	1	2

$C \setminus D$	0	1	2	3
0	0			
1	0	0		
2	0	0	1	
3	0	1	1	1

As funções de síntese minimizadas são mostrada nas Equações 4.23 e 4.24.

$$Q_1 = C \star^1 D^1 + C^1 \star^2 D^1 + C \star^2 D^2 + C^1 \star^3 D^2 + C \star^3 D^3 + C^2 \star^1 D^3 + C^3 \star^2 D^3 \quad (4.23)$$

$$C_{out} = C^3 \star^1 D^3 + C^2 \star^1 D + C^2 \star^1 D^3 + C^2 \star^1 D^2 \quad (4.24)$$

A simulação que mostra o correto funcionamento do circuito lógico do meio somador mostrado na Figura 4.7 foi implementado em VHDL a partir das Equações 4.23 e 4.24 sendo os dois sinais de entrada e os dois sinais de saída números MVL. Na simulação apresentada na Figura 4.8 pode-se observar que a primeira coluna "Cont" identifica a simulação como MVL, a segunda coluna "Signal" mostra os nomes dos sinais sendo eles entradas e saídas, a terceira coluna "Value" representa os valores das entradas e das saídas aos 7580ns de simulação e a última coluna corresponde aos valores MVL de cada sinal. Para a simulação do circuito digital foi considerada a representação digital quaternária $N = 4$, sendo as entradas do meio somador "a" e "b" e as saídas "cout" e "soma" que correspondem a A, B, C_{out} e Q_1 das Equações 4.23 e 4.24 depois de serem substituídos os literais C e D. Finalmente no eixo horizontal mostra-se o tempo de simulação.

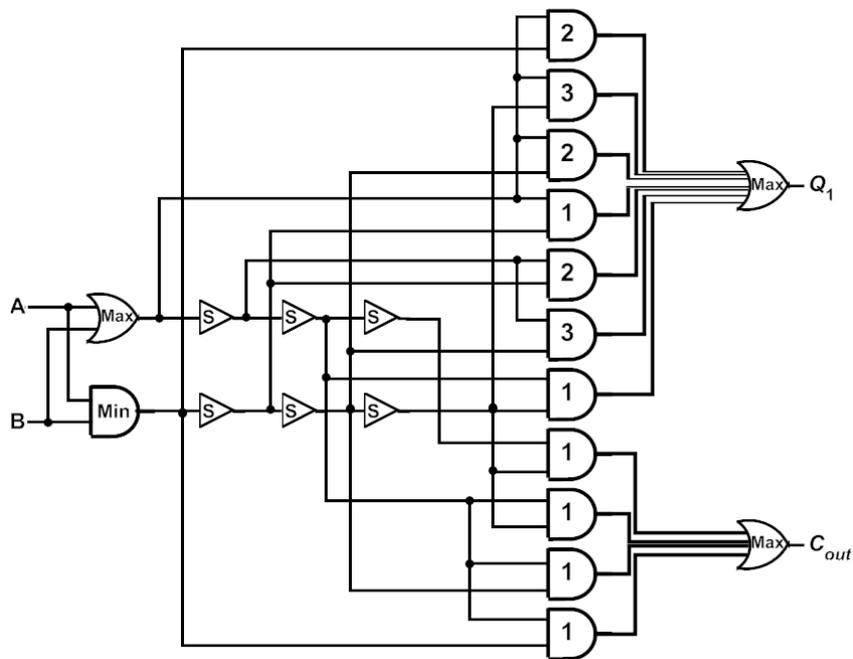


Figura 4.7: Circuito minimizado do meio somador de $N = 4$.

Contex	Signa	Value	7580ns															
			0ns	2000ns	4000ns	6000ns	8000ns	10000ns	12000ns	14000ns	16000							
mv1	a	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
mv1	b	1	0				1				2				3			
mv1	cout	1	0								1	0	1	0	1	1		
mv1	soma	0	0	1	2	3	1	2	3	0	2	3	0	1	3	0	1	2

Figura 4.8: Simulação do meio somador MVL de $N = 4$.

4.6.2 Método Janela

O método de simplificação janela é utilizado para formar um grupo no Mapa de Karnaugh Estendido quando faltam alguns mintermos. Como pode-se ver na Tabela 4.20 para obter o termo $A \star^1 A$ falta o $m_{1,1}^1$ que vem a ser nossa janela, em funções dessa forma agrupa-se utilizando os conceito de Mapas-KE e sintetiza-se o mintermo janela com o operador dual. Logo aplica-se nas duas expressões o operador dual.

Exemplo 1: Simplificar a Tabela 4.20 utilizando o método janela.

Tabela 4.20: Exemplo 1 de simplificação pelo método janela de $N = 4$.

$A \setminus B$	0	1	2	3
0	0	0	0	0
1	1	0	1	1
2	0	0	0	0
3	0	0	0	0

A função de síntese da Tabela 4.20 é mostrada na Equação 4.25 e a forma simplificada é mostrada na Equação 4.26.

$$F(A, B) = A \star^1 B^1 + A \star^1 B^3 + A \star^1 B^2 \quad (4.25)$$

$$F(A, B) = A \star^1 A \cdot A^3 +^0 B^3 \quad (4.26)$$

Exemplo 2: Simplificar a Tabela 4.21 utilizando o método janela.

Tabela 4.21: Exemplo 2 de simplificação pelo método janela de $N = 4$.

$A \setminus B$	0	1	2	3
0	0	0	0	2
1	0	0	0	2
2	0	0	0	2
3	2	2	2	0

A função de síntese da Tabela 4.21 é mostrada na Equação 4.27 e a forma simplificada é mostrada na Equação 4.28.

$$F(A, B) = A^2 \star^2 B^3 + A^1 \star^2 B^3 + A \star^2 B^3 + A^3 \star^2 B^2 + A^3 \star^2 B^1 + A^3 \star^2 B \quad (4.27)$$

$$F(A, B) = (A^3 \star^2 A^3 + B^3 \star^2 B^3) \cdot A^1 +^0 B^1 \quad (4.28)$$

Aplicações MVL

Neste capítulo são projetados circuitos MVL com seus respectivos equivalentes na Álgebra de Chaveamento. Para mostrar a corretude dos circuitos projetados são simulados em VHDL. Também são mostrados os circuitos lógicos em MVL e na Álgebra de Chaveamento para serem comparados em termos de entradas, saídas, conexões e portas lógicas. Nas comparações realizadas são consideradas as conexões e portas lógicas de maneira ilustrativa posto que não temos a implementação física das portas lógicas porém dependendo da tecnologia utilizada pode mudar o número de transistores e portas lógicas.

Nas Seções 5.1, 5.2 e 5.3 são projetados: um decodificador, um somador completo e um multiplexer respectivamente. Veja que todos os circuitos são projetados em MVL e na Álgebra de Chaveamento. Finalmente, na Seção 5.4 são discutidos os ganhos e as perdas.

5.1 Decodificador

A Figura 5.1 mostra o diagrama de bloco para um decodificador MVL de $N = 4$ que é composto por duas entradas A, B e seis saídas Q_1, Q_2, \dots, Q_6 . Também é mostrado o equivalente na Álgebra de Chaveamento que é composto por quatro entradas A, B, C, D e 16 saídas Q_1, Q_2, \dots, Q_{16} . As tabelas verdade que descrevem tais decodificadores são mostradas na Tabela 5.1.

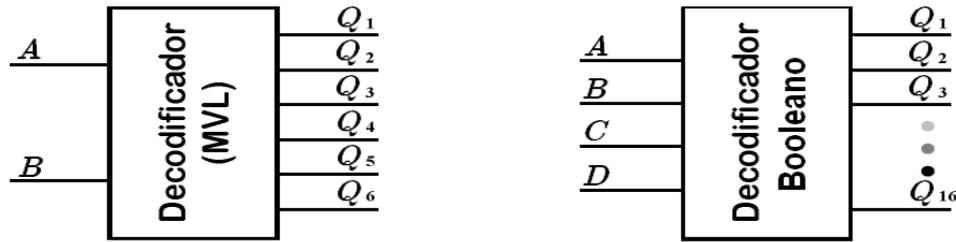


Figura 5.1: Decodificadores: MVL de $N = 4$ e seu equivalente na Álgebra de Chaveamento.

Tabela 5.1: Tabela verdade dos decodificadores MVL de $N = 4$ e seu equivalente na Álgebra de Chaveamento.

Entradas		MVL						Entradas				BINÁRIO															
A	B	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	A	B	C	D	Saídas															
												Q ₁ , Q ₂ , Q ₃ , ..., Q ₁₆															
0	0	1	0	0	0	0	0	0	0	0	0	1000000000000000															
0	1	2	0	0	0	0	0	0	0	0	1	0100000000000000															
0	2	3	0	0	0	0	0	0	0	1	0	0010000000000000															
0	3	0	1	0	0	0	0	0	0	1	1	0001000000000000															
1	0	0	2	0	0	0	0	0	1	0	0	0000100000000000															
1	1	0	3	0	0	0	0	0	1	0	1	0000010000000000															
1	2	0	0	1	0	0	0	0	1	1	0	0000001000000000															
1	3	0	0	2	0	0	0	0	1	1	1	0000000100000000															
2	0	0	0	3	0	0	0	0	1	0	0	0000000010000000															
2	1	0	0	0	1	0	0	0	1	0	0	0000000001000000															
2	2	0	0	0	2	0	0	0	1	0	1	0000000000100000															
2	3	0	0	0	3	0	0	0	1	0	1	0000000000010000															
3	0	0	0	0	0	1	0	0	1	1	0	0000000000001000															
3	1	0	0	0	0	2	0	0	1	1	0	0000000000000100															
3	2	0	0	0	0	3	0	0	1	1	1	0000000000000010															
3	3	0	0	0	0	0	1	0	1	1	1	0000000000000001															

5.1.1 Síntese do Decodificador MVL

A função MVL que sintetiza o decodificador MVL baseado na forma padrão de Soma de operações *Produto Estendido* (SOPE), é obtido mintermo a mintermo através da Tabela 5.1, como mostrado na Seção 3.2. As equações que descrevem o decodificador MVL são mostradas nas Equações 5.1, 5.2, 5.3, 5.4, 5.5 e 5.6; o circuito lógico obtido das equações é mostrado na Figura 5.2.

$$Q_1 = A^1 \star^1 B^1 + A^2 \star^2 B^1 + A^3 \star^3 B^1 \tag{5.1}$$

$$Q_2 = A^1 \star^1 B^2 + A^1 \star^2 B^2 + A^2 \star^3 B^2 \tag{5.2}$$

$$Q_3 = A \star^1 B^3 + A^1 \star^2 B^3 + A^1 \star^3 B^3 \tag{5.3}$$

$$Q_4 = A^3 \star^1 B + A \star^2 B + A^1 \star^3 B \tag{5.4}$$

$$Q_5 = A^2 \star^1 B^1 + A^3 \star^2 B^1 + A \star^3 B^1 \tag{5.5}$$

$$Q_6 = A^2 \star^1 B^2 \tag{5.6}$$

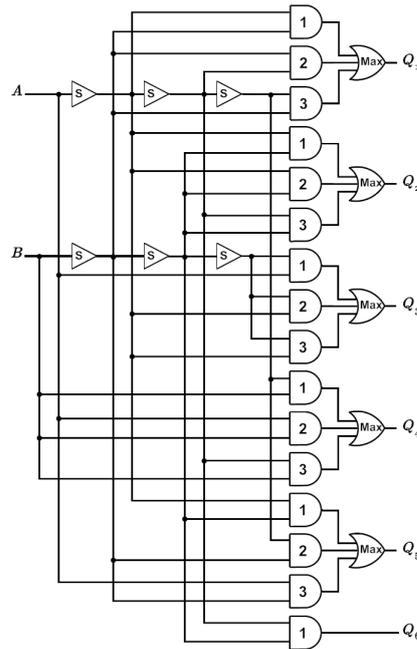


Figura 5.2: Circuito lógico do decodificador MVL de $N = 4$.

5.1.2 Simulação do Decodificador MVL

A simulação que mostra o correto funcionamento do circuito lógico do decodificador MVL mostrado na Figura 5.2 foi implementado em VHDL a partir das equações que o sintetizam sendo os dois sinais de entrada e os seis sinais de saída elementos de D . Na simulação apresentada na Figura 5.3 pode-se observar que a primeira coluna "Cont" identifica a simulação como MVL, na segunda coluna "Signal" mostram-se os nomes dos sinais sendo eles entradas e saídas, a terceira coluna "Value" representa os valores das entradas e das saídas aos 5740ns de simulação e a última coluna corresponde aos valores MVL de cada sinal. Para a simulação do circuito digital foi considerada a representação digital quaternária $N = 4$, sendo as entradas do decodificador "a" e "b" e as saídas "q1", "q2", "q3", "q4", "q5" e "q6" que correspondem a A , B , Q_1 , Q_2 , Q_3 , Q_4 , Q_5 e Q_6 da Tabela 5.1. Finalmente no eixo horizontal mostra-se o tempo de simulação.

Contex	Signa	Valu	5740ns															
			0ns	2000ns	4000ns	6000ns	8000ns	10000ns	12000ns	14000ns	16000ns							
mv1	a	1	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
mv1	b	1	0			1			2			3						
mv1	q1	0	1	0		2	0		3	0		0						
mv1	q2	3	0	2	0		3	0			1		0					
mv1	q3	0	0		3	0			1		0		2	0				
mv1	q4	0	0			1		0		2		0		3	0			
mv1	q5	0	0		1	0		2		0		3		0				
mv1	q6	0	0														1	

Figura 5.3: Simulação do decodificador MVL de $N = 4$.

5.1.3 Síntese do Decodificador Booleano

A função Booleana que sintetiza o decodificador binário baseado na forma padrão de Soma de Operações Produto (SOP), é obtido mintermo a mintermo da Tabela 5.1. As equações que descrevem o decodificador binário são mostradas nas Equações 5.7, 5.8, ..., 5.22; o circuito lógico obtido das equações é mostrado na Figura 5.4.

$$Q_1 = \bar{A} \bar{B} \bar{C} \bar{D} \quad (5.7)$$

$$Q_2 = \bar{A} \bar{B} \bar{C} D \quad (5.8)$$

$$Q_3 = \bar{A} \bar{B} C \bar{D} \quad (5.9)$$

$$Q_4 = \bar{A} \bar{B} C D \quad (5.10)$$

$$Q_5 = \bar{A} B \bar{C} \bar{D} \quad (5.11)$$

$$Q_6 = \bar{A} B \bar{C} D \quad (5.12)$$

$$Q_7 = \bar{A} B C \bar{D} \quad (5.13)$$

$$Q_8 = \bar{A} B C D \quad (5.14)$$

$$Q_9 = A \bar{B} \bar{C} \bar{D} \quad (5.15)$$

$$Q_{10} = A \bar{B} \bar{C} D \quad (5.16)$$

$$Q_{11} = A \bar{B} C \bar{D} \quad (5.17)$$

$$Q_{12} = A \bar{B} C D \quad (5.18)$$

$$Q_{13} = A B \bar{C} \bar{D} \quad (5.19)$$

$$Q_{14} = A B \bar{C} D \quad (5.20)$$

$$Q_{15} = A B C \bar{D} \quad (5.21)$$

$$Q_{16} = A B C D \quad (5.22)$$

5.1.4 Comparação de Resultados

Após comparar os circuitos lógicos do decodificador MVL de $N = 4$ mostrado na Figura 5.2, e seu equivalente da Álgebra de Chaveamento mostrado na Figura 5.4, foram obtidos os dados mostrados na Tabela 5.2, onde pode-se observar que o decodificador de $N = 4$ mostra menor número de entradas, saídas e conexões; também pode-se observar um incremento no número de portas lógicas.

A comparação realizada mostra que a implementação de circuitos digitais em MVL é efetiva para reduzir o número de entradas, saídas e conexões melhorando a área e as interconexões do circuito integrado, e apresenta desvantagens no número de portas lógicas.

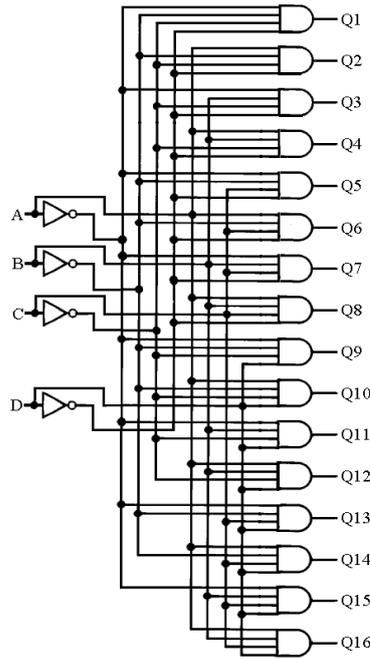


Figura 5.4: Circuito lógico do decodificador Booleano equivalente ao circuito MVL mostrado na Figura 5.2.

Tabela 5.2: Comparação dos decodificadores MVL de $N = 4$ e seu equivalente na Álgebra de Chaveamento.

	No. de entradas	No. de saídas	Portas lógicas	Conexões
MVL ($N=4$)	2	6	27	26
Binário ($N=2$)	4	16	20	56

5.2 Somador Completo

A Figura 5.5 mostra o diagrama de bloco para um somador completo MVL de $N = 4$ que é composto por três entradas A, B, C_{in} e duas saídas Q, C_{out} . Também é mostrado o equivalente na Álgebra de Chaveamento que é composto de cinco entradas $A_0, A_1, B_0, B_1, C_{in}$ e três saídas Q_0, Q_1, C_{out} . As tabelas verdade que descrevem tais decodificadores são mostrados na Tabela 5.3.

5.2.1 Síntese do Somador Completo MVL

A função MVL que sintetiza o somador completo MVL na forma padrão de Soma de operações *Produto Estendido* (SOPE), é baseado no meio somador da Seção 4.17 como é mostrado na Figura 5.6 que corresponde a Tabela 5.3. Assim as equações que descrevem o somador completo MVL são mostradas nas Equações 5.23, 5.24 onde as saídas Q_1 e C_{out} do meio somador

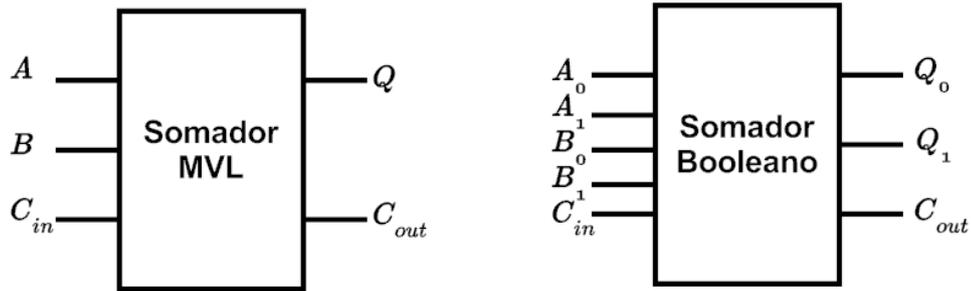


Figura 5.5: Somador completo: MVL de $N = 4$, e seu equivalente na Álgebra de Chaveamento.

Tabela 5.3: Tabela verdade do somador completo MVL de $N = 4$, e seu equivalente na Álgebra de Chaveamento.

MVL					BINÁRIO							
Entradas			Saídas		Entradas					Saídas		
A	B	C_{in}	C_{out}	Q	A_0	A_1	B_0	B_1	C_{in}	C_{out}	Q_0	Q_1
0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	1	0	0	0	1
0	2	0	0	2	0	0	1	0	0	0	1	0
0	3	0	0	3	0	0	1	1	0	0	1	1
1	0	0	0	1	0	1	0	0	0	0	0	1
1	1	0	0	2	0	1	0	1	0	0	1	0
1	2	0	0	3	0	1	1	0	0	0	1	1
1	3	0	1	0	0	1	1	1	0	1	0	0
2	0	0	0	2	1	0	0	0	0	0	1	0
2	1	0	0	3	1	0	0	1	0	0	1	1
2	2	0	1	0	1	0	1	0	0	1	0	0
2	3	0	1	1	1	0	1	1	0	1	0	1
3	0	0	0	3	1	1	0	0	0	0	1	1
3	1	0	1	0	1	1	0	1	0	1	0	0
3	2	0	1	1	1	1	1	0	0	1	0	1
3	3	0	1	2	1	1	1	1	0	1	1	0
0	0	1	0	1	0	0	0	0	1	0	0	1
0	1	1	0	2	0	0	0	1	1	0	1	0
0	2	1	0	3	0	0	1	0	1	0	1	1
0	3	1	1	0	0	0	1	1	1	1	0	1
1	0	1	0	2	0	1	0	0	1	0	1	0
1	1	1	0	3	0	1	0	1	1	0	1	1
1	2	1	1	0	0	1	1	0	1	1	0	1
1	3	1	1	1	0	1	1	1	1	1	0	1
2	0	1	0	3	1	0	0	0	1	0	1	1
2	1	1	1	0	1	0	0	1	1	1	0	0
2	2	1	1	1	1	0	1	0	1	1	0	1
2	3	1	1	2	1	0	1	1	1	1	1	0
3	0	1	1	0	1	1	0	0	1	1	0	0
3	1	1	1	1	1	1	0	1	1	1	0	1
3	2	1	1	2	1	1	1	0	1	1	1	0
3	3	1	1	3	1	1	1	1	1	1	1	1

são consideradas como MS , C_o respectivamente. C é o máximo de A , B e D é o mínimo de A e B ; o circuito lógico obtido das equações é mostrado na Figura 5.7.

$$Q = MS \star^1 C_{in}^1 + MS \star^2 C_{in}^2 + MS \star^3 C_{in}^3 + MS^1 \star^1 C_{in} + MS^1 \star^2 C_{in}^1 + MS \star^3 C_{in}^2 \quad (5.23)$$

$$C_{out} = C^3 \star^1 D^3 + C^2 \star^1 D + C^2 \star^1 D^3 + C^2 \star^1 D^2 + C^2 \star^1 C_{in}^3 + C^3 \star^1 D \star^1 C_{in}$$

$$C_{out} = C_o + C^2 \star^1 C_{in}^3 + C^3 \star^1 D \star^1 C_{in} \quad (5.24)$$

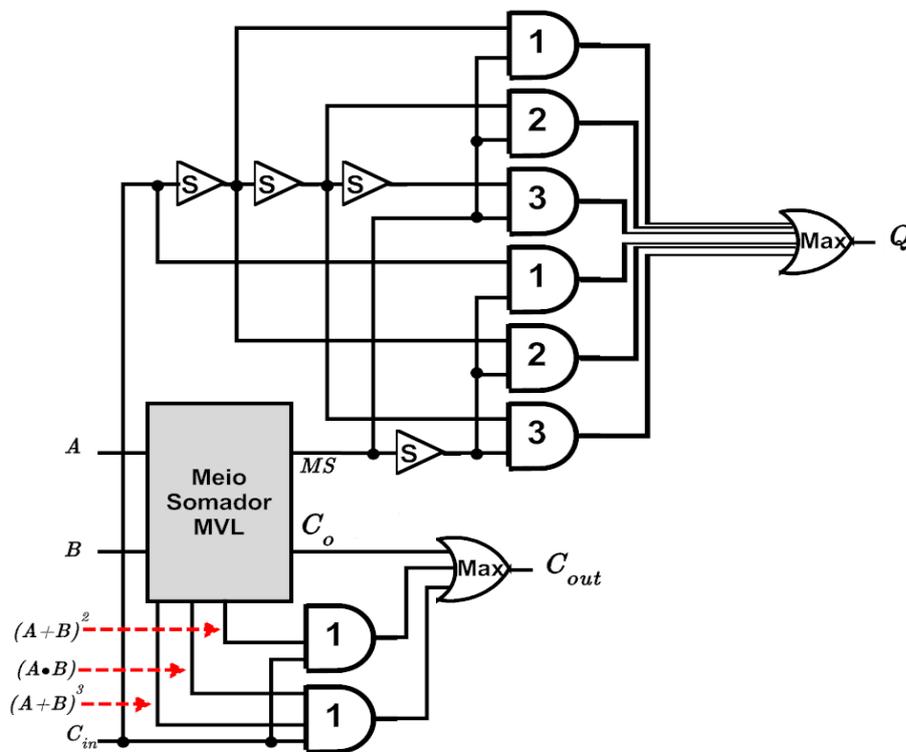


Figura 5.6: Circuito lógico do somador completo MVL de $N = 4$ baseado no meio somador.

5.2.2 Simulação do Somador Completo MVL

A simulação que mostra o correto funcionamento do circuito lógico do Somador Completo MVL mostrado na Figura 5.7 foi implementado em VHDL a partir das equações que o sintetizam sendo os três sinais de entrada e os dois sinais de saída elementos de D . Na simulação apresentada na Figura 5.8 pode-se observar que a primeira coluna "Cont" identifica a simulação como MVL, na segunda coluna "Signal" mostra-se os nomes dos sinais sendo eles entradas e saídas, a terceira coluna "Value" representa os valores das entradas e das saídas aos 15880ns de simulação e a última coluna corresponde aos valores MVL de cada sinal. Para a simulação do

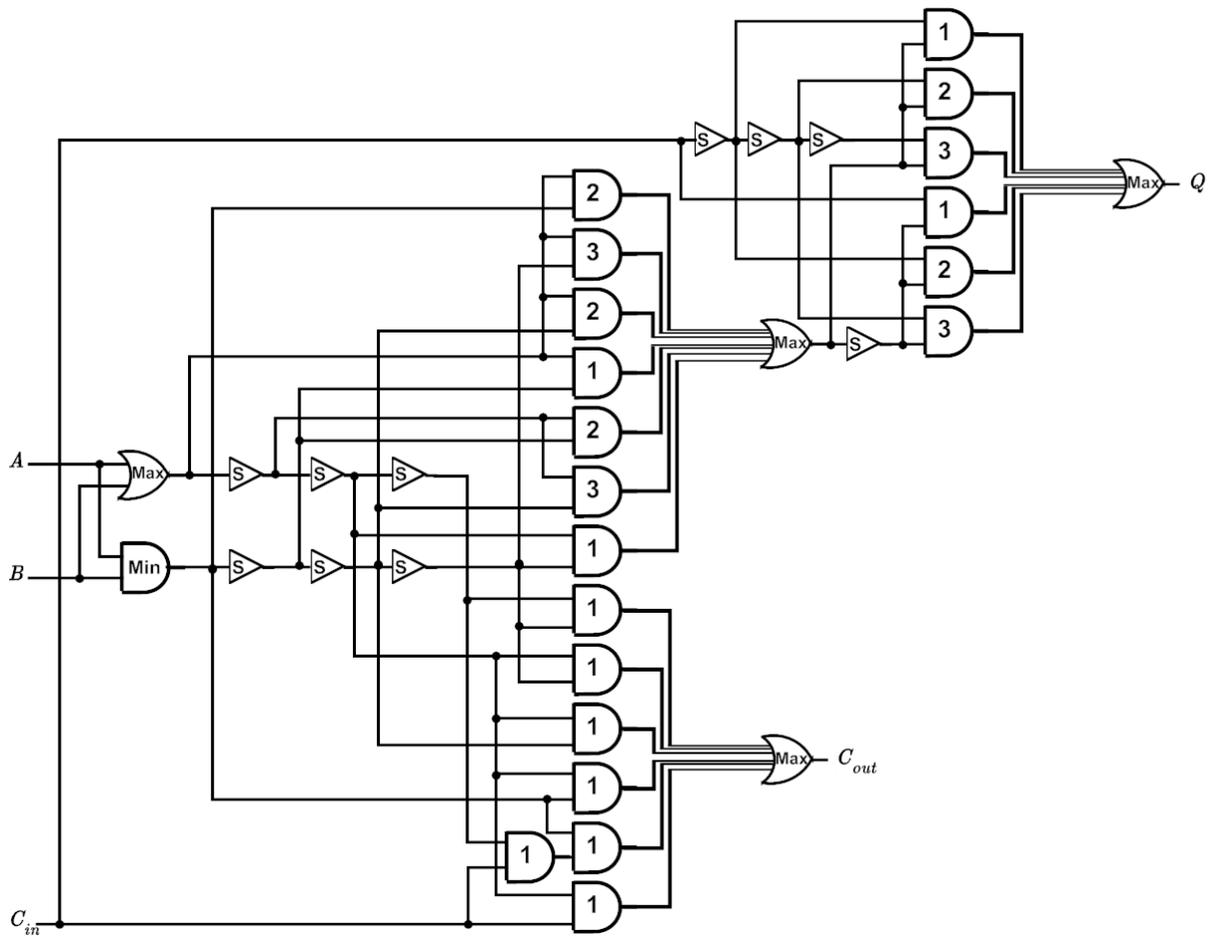


Figura 5.7: Circuito lógico do somador completo MVL de $N = 4$.

circuito digital da Figura 5.7 foi considerada a representação digital quaternária $N = 4$, sendo as entradas do somador “a”, “b”, “cin”, “cout” e “soma” que correspondem a: A, B, C_{in}, C_{out} , e Q da Tabela 5.3. Finalmente, no eixo horizontal mostra-se o tempo de simulação.

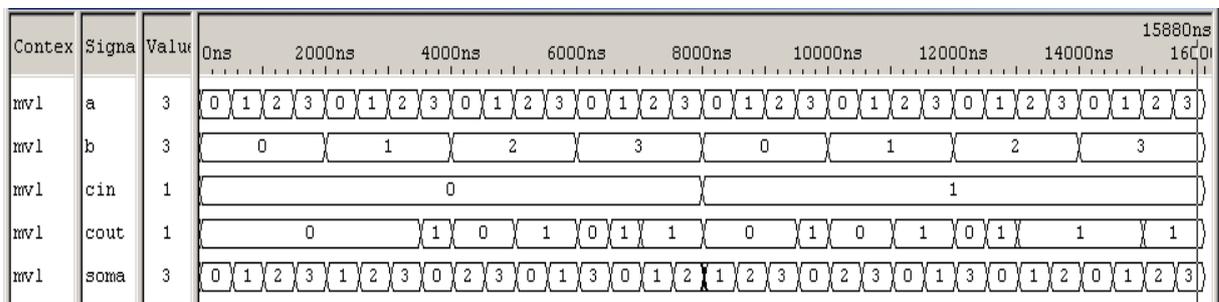


Figura 5.8: Simulação do somador MVL de $N = 4$.

5.2.3 Síntese do Somador Completo Booleano

A função Booleana que sintetiza o somador completo binário baseado na forma padrão de Soma de Operações Produto (SOP), é obtido mintermo a mintermo da Tabela 5.3. As equações que descrevem o somador completo binário são mostradas nas Equações 5.25, 5.26 e 5.27; o circuito lógico obtido das equações é mostrado na Figura 5.9.

$$Q_1 = A_1 B_1 C_{in} + \overline{A_1} \overline{B_1} C_{in} + A_1 \overline{B_1} \overline{C_{in}} + \overline{A_1} B_1 \overline{C_{in}} \quad (5.25)$$

$$D = A_1 B_1 + A_1 C_{in} + B_1 C_{in}$$

$$Q_0 = A_0 B_0 D + \overline{A_0} \overline{B_0} D + A_0 \overline{B_0} \overline{D} + \overline{A_0} B_0 \overline{D} \quad (5.26)$$

$$C_{out} = A_0 B_0 + A_0 D + B_0 D \quad (5.27)$$

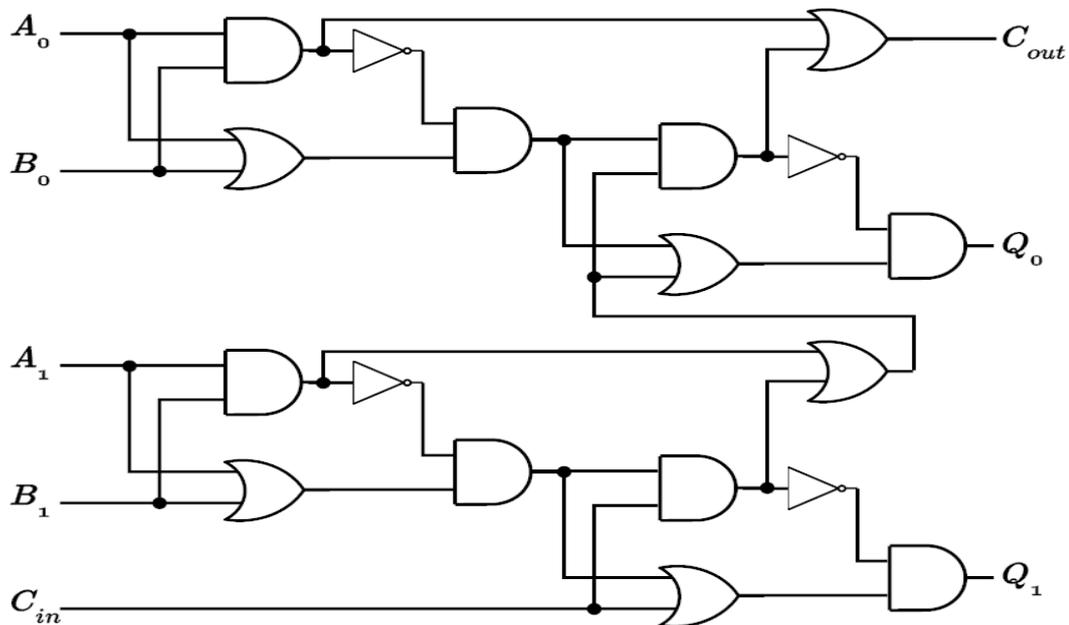


Figura 5.9: Circuito lógico do somador completo Booleano.

5.2.4 Comparação de Resultados

Após comparar os circuitos lógicos do somador completo MVL de $N = 4$ mostrado na Figura 5.7, e seu equivalente da Álgebra de Chaveamento mostrado na Figura 5.9, foram obtidos os dados mostrados na Tabela 5.4, onde pode-se observar que o somador completo de $N = 4$ mostra menor número de entradas e saídas; também pode-se observar um incremento no número de portas e conexões lógicas.

Tabela 5.4: Comparação dos somadores MVL de $N = 4$ e seu equivalente na Álgebra de Chaveamento.

	No. de entradas	No. de saídas	Portas lógicas	Conexões
MVL (N=4)	3	2	35	32
Binário (N=2)	5	3	18	12

A comparação realizada mostra que a implementação de circuitos digitais em MVL é efetiva para reduzir o número de entradas e saídas melhorando a área e as interconexões entre circuitos integrados, e apresenta desvantagens no número de portas lógicas e conexões internas.

5.3 Multiplexer

A Figura 5.10 mostra o diagrama de bloco para um multiplexer MVL de $N = 4$ que é composto por cinco entradas A, B, C, D, C_{trl} e uma saída O . Também é mostrado o equivalente na Álgebra de Chaveamento que é de dez entradas $A_0, A_1, B_0, B_1, C_0, C_1, D_0, D_1, C_{trl0}$ e C_{trl1} com duas saídas O_0, O_1 . As tabelas verdade que descrevem tais decodificadores são mostrados na Tabela 5.5.

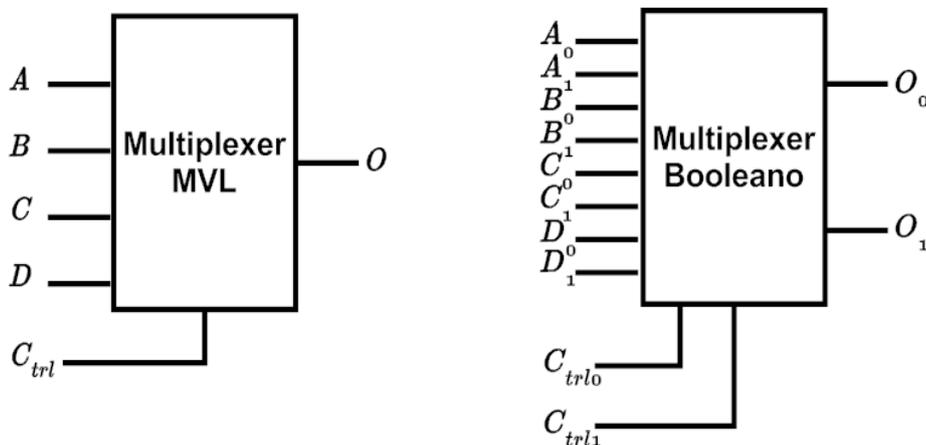


Figura 5.10: Multiplexer: MVL de $N = 4$ e seu equivalente na Álgebra de Chaveamento.

5.3.1 Síntese do Multiplexer MVL

A função MVL que sintetiza o multiplexer MVL na forma padrão de Soma de Operações Produto Estendido (SOPE), é obtido minitermo a minitermo da Tabela 5.5, como mostrado na Seção 3.2. A equação que descreve o multiplexer MVL é mostrado na Equação 5.28 e o circuito lógico obtido da equação é mostrado na Figura 5.11.

Tabela 5.5: Tabela verdade dos multiplexer MVL de $N = 4$, e seu equivalente na Álgebra de Chaveamento.

MVL		BINÁRIO			
Entradas	Saídas	Entradas		Saídas	
C_{trl}	O	C_{trl0}	C_{trl1}	O_0	O_1
0	A	0	0	A_0	A_1
1	B	0	1	B_0	B_1
2	C	1	0	C_0	C_1
3	D	1	1	D_0	D_1

$$\begin{aligned}
 O = & C_{trl}^1 \star^1 A + C_{trl}^2 \star^2 A + C_{trl}^3 \star^3 A + \\
 & C_{trl} \star^1 B + C_{trl}^1 \star^2 B + C_{trl}^2 \star^3 B + \\
 & C_{trl}^3 \star^1 C + C_{trl} \star^2 C + C_{trl}^1 \star^3 C + \\
 & C_{trl}^2 \star^1 D + C_{trl}^3 \star^2 D + C_{trl} \star^3 D
 \end{aligned}
 \tag{5.28}$$

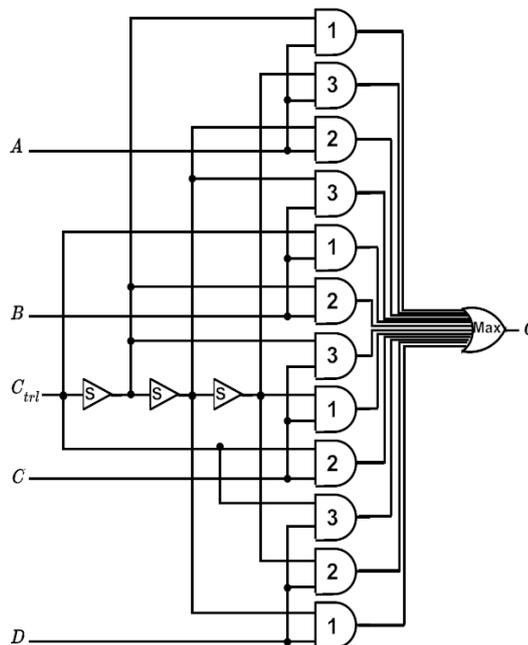


Figura 5.11: Circuito lógico do multiplexer MVL de $N = 4$.

5.3.2 Simulação do Multiplexer MVL

A simulação que mostra o correto funcionamento do circuito lógico do multiplexer MVL mostrado na Figura 5.11 foi implementado em VHDL a partir da equação que o sintetiza sendo

os cinco sinais de entrada e o sinal de saída elementos de **D**. Na simulação apresentada na Figura 5.12 pode-se observar que a primeira coluna "Cont" identifica a simulação como MVL, na segunda coluna "Signal" mostra-se os nomes dos sinais sendo eles entradas e saídas, a terceira coluna "Value" representa os valores das entradas e das saídas aos 12320ns de simulação e a última coluna corresponde aos valores MVL de cada sinal. Para a simulação do circuito digital foi considerada a representação digital quaternária $N = 4$, sendo as entradas do multiplexer "a", "b", "c", "d", "ctrl" e a saída "o", que correspondem a A, B, C, D e C_{trl} da Tabela 5.5. Finalmente no eixo horizontal mostra-se o tempo de simulação.

Cont	Signal	Value	12320ns																													
			0ns	2000ns				4000ns				6000ns				8000ns				10000ns				12000ns				14000ns				16000
mv1	a	0	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3						
mv1	b	1	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0						
mv1	c	2	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1						
mv1	d	3	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2						
mv1	ctrl	3	0				1				2				3																	
mv1	o	3	0	1	2	3	1	2	3	0	2	3	0	1	3	0	1	2														

Figura 5.12: Simulação do multiplexer MVL de $N = 4$.

5.3.3 Síntese do Multiplexer Booleano

A função Booleana que sintetiza o multiplexer binário baseado na forma padrão de Soma de Operações Produto (SOP), é obtido mintermo a mintermo da Tabela 5.1. As equações que descrevem o decodificador binário são mostradas nas Equações 5.29 e 5.30; o circuito lógico obtido das equações é mostrado na Figura 5.13.

$$O_0 = A_0 \overline{C_{trl0}} \overline{C_{trl1}} + B_0 \overline{C_{trl0}} C_{trl1} + C_0 C_{trl0} \overline{C_{trl1}} + D_0 C_{trl0} C_{trl1} \quad (5.29)$$

$$O_1 = A_1 \overline{C_{trl0}} \overline{C_{trl1}} + B_1 \overline{C_{trl0}} C_{trl1} + C_1 C_{trl0} \overline{C_{trl1}} + D_1 C_{trl0} C_{trl1} \quad (5.30)$$

5.3.4 Comparação de Resultados

Após comparar os circuitos lógicos do multiplexer MVL de $N = 4$ mostrado na Figura 5.11, e seu equivalente da Álgebra de Chaveamento mostrado na Figura 5.13, foram obtidos os dados mostrados na Tabela 5.6, onde pode-se observar que o multiplexer de $N = 4$ mostra menor número de entradas, saídas e portas lógicas; também pode-se observar um incremento no número de conexões.

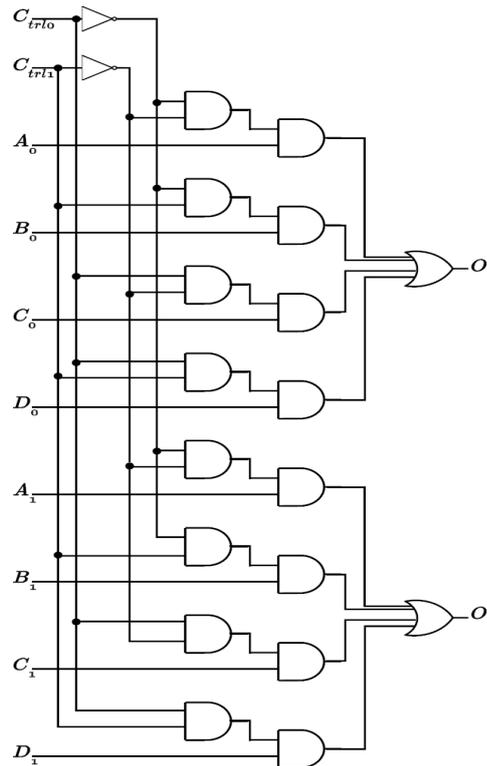


Figura 5.13: Circuito lógico do multiplexador Booleano equivalente ao circuito MVL mostrado na Figura 5.11.

Tabela 5.6: Comparação dos multiplexador MVL de $N = 4$ e seu equivalente na Álgebra de Chaveamento.

	No. de entradas	No. de saídas	Portas lógicas	Conexões
MVL (N=4)	5	1	19	16
Binário (N=2)	12	2	20	14

A comparação realizada mostra que a implementação de circuitos digitais em MVL é efetiva para reduzir o número de entradas, saídas e portas lógicas melhorando a área e as interconexões do circuito integrado, e apresenta desvantagens no número de conexões.

5.4 Discussão das Comparações

De acordo com as Tabelas 5.2, 5.4 e 5.6 podem ser feitas as seguintes observações:

O número de entradas e saídas foram reduzidas em comparação aos circuitos equivalentes projetados na Álgebra de Chaveamento, quanto ao número de portas lógicas, no primeiro e no segundo caso elas aumentaram e no último caso elas diminuíram. Quanto ao número de conexões, no primeiro caso elas diminuíram e no segundo e terceiro caso elas aumentaram.

Pode-se dizer que o aumento ou a diminuição das portas lógicas ou das conexões varia dependendo do circuito a ser projetado.

A Figura 5.14, traz a relação dos circuitos MVL com seus equivalentes na Álgebra de Chaveamento. Pode-se perceber que depois de utilizar a Álgebra MVL, o número de entradas e saídas ficaram em um intervalo entre 38% e 67%. O número de portas e conexões ficaram num intervalo entre 46% e 267%. Com isso, pode-se notar que a lógica MVL reduz o número de entradas e o número de saídas mas o número de conexões e de portas lógicas depende do problema.

Assim, na Figura 5.14, pode-se observar que em todos os casos os números de entradas e saídas foram reduzidos, mas no número de portas lógicas e conexões existe variação devido ao tipo do circuito. Por exemplo, o "vai-um" do somador completo é um problema binário, razão pela qual o incremento da representação digital não melhora o circuito. É importante considerar que o número de portas pode mudar devido a tecnologia usada para projetar os circuitos.

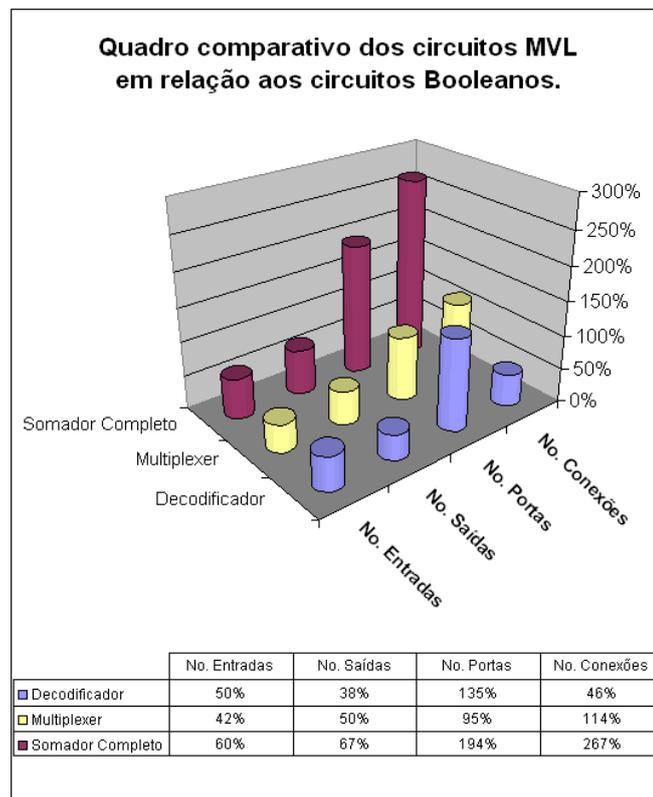


Figura 5.14: Quadro resumo das comparações.

Considerações Finais

6.1 Conclusões

Nesta pesquisa foram definidos dois operadores *Produto Estendido* e *Máximo Estendido*. Os operadores *Máximo*, *Mínimo* e *Sucessor* já conhecidos, com os dois operadores propostos formaram dois conjuntos universais de operadores.

Foram definidas duas formas padrão de funções MVL (SOPE Soma de Operações Produto Estendido e POSE Produto de Operações Soma Estendida), uma para cada conjunto universal de operadores, assim como os mintermos e maxtermos MVL. Os dois conjuntos universais de operadores tem o princípio da dualidade.

A metodologia de síntese apresentada é similar à já conhecida da Álgebra de Chaveamento, obtida mintermo a mintermo na forma SOPE ou maxtermo a maxtermo na forma POSE. Os métodos da análise de circuitos também mantem semelhança aos da Álgebra de Chaveamento.

A Álgebra de MV proposta permite a simplificação de funções MVL. Para a simplificação de funções MVL foram estendidos os métodos Mapa de Karnaugh, Quine McCluskey e Algoritmo Petrick da Álgebra de Chaveamento. Além dos métodos estendidos foi definido a simplificação pela aplicação do operador *Sucessor*, o método janela e a simplificação simétrica.

A funcionalidade da Álgebra de MV proposta e dos métodos de simplificação foi testada na projeção de um decodificador MVL, um somador completo MVL e um multiplexer MVL, com as respectivas simulações em VHDL.

Finalmente, dos circuitos projetados conclui-se que ao incrementar a base de representação numérica se reduz o número de entradas e saídas, isto reduz as interconexões entre circuitos integrados sendo útil na construção de circuitos VLSI. A metodologia apresentada nesta pesquisa

tem vantagem em relação a outras metodologias, devido à simplicidade dos operadores e a fácil extensão dos conceitos já estabelecidos no projeto de circuitos lógicos da Álgebra de Chaveamento para a Álgebra de MV.

6.2 Trabalhos Futuros

Nesta dissertação foi apresentada uma Álgebra de Múltiplos Valores e sua aplicação nos circuitos combinacionais, assim como os métodos de simplificação para os circuitos digitais combinacionais. Pelo fato de ser uma álgebra nova existem muitos trabalhos que podem ser desenvolvidos. A seguir são apresentados alguns deles:

- A implementação física das portas lógicas propostas nesta pesquisa.
- A utilidade da Álgebra de MV proposta na projeção dos circuitos seqüências de Múltiplos Valores, assim como expandir os conceitos de simplificação de circuitos seqüências da Álgebra de Chaveamento para a Álgebra de MV proposta.
- A minimização de redes MVL, o objetivo desta pesquisa é a minimização de nós MVL, onde cada nó é uma função MVL, e procura-se reduzir o número de literais e nós.
- A elaboração de um software para a síntese de funções MVL, que permita realizar a análise do tempo e a procura do pior caso no tempo de execução.
- A representação de imagens com funções MVL. Com a lógica booleana consegue-se representar imagens binárias supondo que o valor 0 é o fundo e o valor 1 é o componente.
- A aplicação dos métodos de simplificação propostos na técnica de mineração de dados Rough Sets.
- A comparação quantitativamente e qualitativamente com outras álgebras de MV na literatura.

Referências Bibliográficas

- [1] Claude Elwood Shannon. A Symbolic Analysis of Relay and Switching Circuits. Master's thesis, Massachusetts Institute of Technology, 1940.
- [2] Lukasiewicz. *On Three Valued-logic*. L. Borkowski, Select Works, pp. 169-171, North-Holland, Amsterdam 1920.
- [3] Grzegorz Malinowski. *A Companion to Philosophical Logic*, chapter Many-Valued Logic, pages 545–561. Blackwell, May 2002.
- [4] Emil Leon Post. Introduction to a General Theory of Elementary Propositions. *American Journal of Mathematics*, 1921.
- [5] Stanley L. Hurst. Two Decades of Multiple-Valued Logic - An Invited Tutorial. *International Symposium on Multiple-Valued Logic (ISMVL)*, (18):165–175, May 1988.
- [6] George Epstein. The Lattice Theory of Post Algebras. *Transactions of the American Mathematical Society*, 95(2):300–317, May 1960.
- [7] Dexter Kozen. On Kleene Algebras and Closed Semirings. In B. Rovan, editor, *Mathematical Foundations of Computer Science 1990*, volume 452. Springer-Verlag, May 1990.
- [8] Mou Hu. A Multiple-Valued Algebra for Modeling MOS VLSI Circuits at Switch-Level. *International Symposium on Multiple-Valued Logic (ISMVL)*, (18):329–336, May 1988.
- [9] Tatsuki Watanabe; Masayuki Matsumoto; Tekken Li. CMOS Four-Valued Logic Circuits Using Charge-Control Technique. *International Symposium on Multiple-Valued Logic (ISMVL)*, (18):90–97, May 1988.
- [10] Noboru Takagi; Masao Mukaidono. Kleene-Stone Logic Functions. *International Symposium on Multiple-Valued Logic (ISMVL)*, (20):93–100, May 1990.

- [11] Corina Reischer; Dan A. Simovici. Chain-Based Ockham Algebras. *International Symposium on Multiple-Valued Logic (ISMVL)*, (19):456–462, May 1989.
- [12] Zhang Dongmou. Medium Algebra MA and Medium Propositional Calculus MP*. *International Symposium on Multiple-Valued Logic (ISMVL)*, (19):289–294, May 1989.
- [13] Hurst; Stanley L. Multi-Valued Logic - It's Status and It's Future. In *IEEE transactions on computers*, volume C-33, 1984.
- [14] Atul K. Jain; Ron J. Bolton; Mostafa. CMOS Multiple-Valued Logic Design - Part I: Function Realization. *IEEE Transactions on Circuits and Systems*, Vol. 40 No 8, 1993.
- [15] Z. Tang; O. Ishizuka. A Learning Multiple-Valued Logic Network Algebra, Algorithm and Applications. *IEEE Transactions on Computer*, Vol. 47, No 2, 1998.
- [16] I. Thoidis; D. Soudris; I. Karafyllidis; S. Christoforidis; A. Thanailakis. Quaternary Voltage Mode CMOS Circuits for Multiple-Values Logic. *IEE Proc.-Circuits Devices Syst*, Vol. 145, No. 2, 1998.
- [17] Mahmood Bahraini; George Epstein. Three-Valued Karnaugh Maps. *International Symposium on Multiple-Valued Logic (ISMVL)*, (18):178–185, May 1988.
- [18] Tsutomu Sasao. EXMIM: A Simplification Algorithm for Exclusive-Or-Sum-Products Expressions for Multiple-Valued Input Two-Valued Output Functions. *International Symposium on Multiple-Valued Logic (ISMVL)*, (20):128–135, May 1990.
- [19] G. W. Dueck. RCM-MVL: A Recursive Consensus MVL Minimization Algorithm. *International Symposium on Multiple-Valued Logic (ISMVL)*, (20):136–143, May 1990.
- [20] Konrad Lei; Zvonko G. Vranesic. On the Synthesis of 4 Valued Current Mode CMOS Circuits. *International Symposium on Multiple-Valued Logic (ISMVL)*, (21):147–151, May 1991.
- [21] Okihiko Ishizuka; Hiroshi Takarabe; Zheng Tang; Hiroki Matsumoto. Synthesis of Current-Mode Pass Transistor Networks. *International Symposium on Multiple-Valued Logic (ISMVL)*, (21):139–146, May 1991.
- [22] Xunwei Wu; Franklin Prosser. Ternary CMOS Sequential Circuits. *International Symposium on Multiple-Valued Logic (ISMVL)*, (18):307–313, May 1988.
- [23] Takahiro Hanyu; Tatsuo Higuchi. Design of a Highly Parallel AI Processor Using New Multiple-Valued MOS Devices. *International Symposium on Multiple-Valued Logic (ISMVL)*, (18):300–306, 1988.

- [24] Michitaka Kameyama; Tsutomu Sekibe; Tatsuo Higuchi. Design of Highly Parallel Residue Arithmetic Circuits Based on Multiple-Valued Bidirectional Current-Home MOS Technology. *International Symposium on Multiple-Valued Logic (ISMVL)*, (18):6–13, May 1988.
- [25] M. H. Abd-El-Barr; T.D. Hoank; Z. G. Vranesic. Programmable Realization of Multi-Valued Multi-Threshold Functions. *International Symposium on Multiple-Valued Logic (ISMVL)*, (19):42–53, May 1989.
- [26] Hsu Liang Ho; Kenneth C. Smith. Switched Capacitor Circuits in The Implementation of Multiple-Valued Logic. *International Symposium on Multiple-Valued Logic (ISMVL)*, (19):202–209, May 1989.
- [27] Takahiro Hanyu; Tatsuo Higuchi. Design of a High-Density Multiple-Valued Content-Addressable Memory Based on Floating-Gate MOS Devices. *International Symposium on Multiple-Valued Logic (ISMVL)*, (20):18–23, May 1990.
- [28] Turgay Temel; Avni Morgul. Implementation of Multi-Valued Logic Gates Using Full Current-Mode CMOS Circuits. *Analog Integrated Circuits and Signal Processing*, 2004.
- [29] M. Abd-El-Barr; G. A. Hamid; M. N. Hasan. *Synthesis of MVL Functions Using Input and Output Assignments*. IEEE Proc. Circuits Devices Syst., June 1998.
- [30] Yunjian Jiang. Multi-Valued Logic Network Minimization and Its Applications. Master's thesis, University of California at Berkeley, 2000.
- [31] Ékué Kinvi-Boh. *Conception de Circuits en Logique Ternaire: de la Caractérisation au Niveau Transistor à la Modélisation Architecturale*. PhD thesis, Université de Rennes 1, França, 2006.
- [32] John F Wakerly. *Digital Design: Principles and Practices*. Prentice Hall, Upper Saddle River, New Jersey, 1999.
- [33] H. H. Loomis Jr; R. H. Wyman Jr. On Complete Sets of Logic Primitives. volume EC-14, April 1965.
- [34] Pawel Kerntopf; Marek A. Perkowski; Mozammel H. A. Khan. On Universality of General Reversible Multiple-Valued Logic Gates. *International Symposium on Multiple-Valued Logic (ISMVL)*, (34):68–73, May 2004.
- [35] Victor P. Nelson; H. Troy Nagle; Bill D. Carroll; J. David Irwin. *Digital Logic Circuit Analysis And Design*. Prentice Hall Englewood Cliffs, 2002.

- [36] Herbert Luque Peralta; M. A. Tito; M. G. Turqueti; E. M. Martins; M. E. Romero. Síntesis de Circuitos en Lógica Multi-Nivel. *V Congreso de Informática y Sistemas de Sudamérica*, 2006.
- [37] Herbert Luque Peralta; E. M. Martins; M. E. Romero; R. R. Dos Santos. El Principio de Dualidad en la Lógica de Múltiples Valores. *Anais do ANDESCON 2008 - IEEE SECCION PERU*, 2008.
- [38] Sajjan G. Shiva; Shiva G. Shiva. *Digital Logic and Microprocessor Design with VHDL*. CRC Press, 1998.
- [39] Enoch O Hwang. *Digital Logic and Microprocessor Design with VHDL*. La sierra University, Riverside, 2005.
- [40] Edward Joseph McCluskey. *Algebraic Minimization and The Design of Two-Terminal Contact Networks*. PhD thesis, Massachusetts Institute of Technology, June 1956.
- [41] Parag K. Lala. *Principles of Modern Digital Design*. Wiley-interscience, 2007.