

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

RICARDO CUNHA GONÇALVES DA SILVA

**Lógica Quaternária de Alto
Desempenho e Baixo Consumo
para Circuitos VLSI**

Tese apresentada como requisito parcial para a
obtenção do grau de Doutor em
Microeletrônica

Prof. Dr. Henri Boudinov
Orientador

Prof. Dr. Luigi Carro
Co-orientador

Porto Alegre, dezembro de 2007.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Silva, Ricardo Cunha Gonçalves da

Lógica Quaternária de Alto Desempenho e Baixo Consumo para Circuitos VLSI / Ricardo Cunha Gonçalves da Silva – Porto Alegre: Programa de Pós-Graduação em Microeletrônica, 2007.

91 f.:il.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica. Porto Alegre, BR – RS, 2007. Orientador: Henri Boudinov; Co-orientador: Luigi Carro.

1.Lógica Multivalorada. 2.Lógica Quaternária 3.Circuitos VLSI. 4.FPGA. I. Boudinov, H. II. Carro, Luigi. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Profa. Valquiria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PPGC: Prof^a Luciana Porcher Nedel

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço a todos que contribuíram para esse trabalho, em especial aos orientadores Henri e Luigi, aos professores Rogério Maltêz, Fernanda Lima, Renato Ribas e aos colegas Cristiano Lazzari, Mateus Dalponte, Artur Coelho e Carlos Cima.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS	7
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
2 CIRCUITOS DE LÓGICA MULTIVALORADA	15
2.1 Lógica multivalorada	15
2.2 Circuitos existentes de lógica multivalorada	17
2.2.1 Circuitos multivalorados modo-corrente.....	18
2.2.2 Circuitos multivalorados modo-tensão.....	21
3 CIRCUITOS QUATERNÁRIOS PROPOSTOS	26
3.1 Circuitos Elementares	26
3.1.1 Inversor Quaternário.....	28
3.1.2 MIN e MAX.....	30
3.1.3 DLC.....	32
3.1.4 Multiplexador.....	32
3.2 Circuitos Aritméticos	36
3.2.1 Contador.....	36
3.2.2 Somadores Parciais (2:2).....	38
3.2.3 Somadores Completos (5:2).....	43
3.3 Look-Up Tables	45
4 SIMULAÇÕES	49
4.1 Transistores	49
4.2 Circuitos Elementares	50
4.2.1 Inversor.....	50
4.2.2 NMIN/NMAX.....	56
4.2.3 DLC.....	56
4.2.4 MUX.....	57
4.3 Circuitos Aritméticos	58
4.3.1 Somador MIN-MAX.....	58

4.3.2	Somador Direto	59
4.3.3	Somador Parcial 2:2 MUX	59
4.3.4	<i>Carry</i> de Saída	61
4.3.5	Somador Completo 5:2.....	61
4.3.6	Equivalentes Binários.....	62
4.4	Análise dos Resultados e Discussões	62
5	APLICAÇÃO DE LÓGICA QUATERNÁRIA EM FPGA.....	65
5.1	FPGA	65
5.1.1	Arquitetura de um FPGA.....	65
5.2	FPGA Quaternário.....	67
5.2.1	Simulações Elétricas.....	68
5.2.2	Mapeamento Lógico.....	69
5.2.3	Discussões.....	75
6	CONCLUSÕES E IDÉIAS DE CONTINUAÇÃO DO TRABALHO	77
	REFERÊNCIAS.....	80
	APÊNDICE A PARÂMETROS SPICE DA TECNOLOGIA TSMC 0,18μM	85
	APÊNDICE B TABELA VERDADE DA FUNÇÃO SOMA3	87
	APÊNDICE C RESULTADO DA SIMULAÇÃO DA OPERAÇÃO SOMA3 NA FERRAMENTA ABC USANDO 2LUTS COMO BLOCO BÁSICO.....	88

LISTA DE ABREVIATURAS E SIGLAS

ASIC	Application Specific Integrated Circuit
CLB	Configurable Logic Block
CMOS	Complementary Metal Oxide Semiconductor
CMVL	Current-mode Multiple Valued Logic
DLC	Down Literal Circuit
FA	Full Adder
HA	Half Adder
LUT	Look-Up Table
MESFET	Metal Silicon Field Effect Transistor
MOS	Metal Oxide Silicon transistor
MUX	MUltipleXer
MVL	Multiple Valued Logic
NMOS	N-type Metal Oxide Semiconductor transistor
PMOS	P-type Metal Oxide Semiconductor transistor
QLUT	Quaternary Look-Up Table
QMUX	Quaternary MUltipleXer
SRAM	Static Random Access Memory
VLSI	Very Large Scale of Integration

LISTA DE FIGURAS

Figura 2.1: Funções lógicas MIN e MAX quaternárias	17
Figura 2.2: Comparador de correntes.....	19
Figura 2.3: Somador quaternário modo-corrente	20
Figura 2.4: Inversor quaternário: Circuito, tensão de limiar dos transistores e relação entre sinal de entrada, os estados de operação dos transistores controlados pela entrada e sinais de saída.....	22
Figura 2.5: Inversor ternário diametral	22
Figura 2.6: Circuitos a) MAX e b) MIN	23
Figura 2.7: A estrutura básica de um Neuron MOS	24
Figura 2.8: DLC usando vMOS.....	24
Figura 2.9: Sinais de entrada, símbolo e sinais de saída para os circuitos detectores de limiar de 0,5, 1,5 e 2,5	25
Figura 3.1: Estrutura de um circuito de lógica estática binário.....	26
Figura 3.2: Inversor binário.....	27
Figura 3.3: Esquemático de um circuito de lógica estática quaternário	27
Figura 3.4: Circuito Inversor e sua tabela verdade	28
Figura 3.5: Id x Vgs dos transistores usados no circuito inversor quaternários.....	29
Figura 3.6: Curvas I_{DS} x V_{GS} esquemáticas dos 8 diferentes transistores usados	30
Figura 3.7: Circuito NMIN e sua tabela verdade.....	30
Figura 3.8: Circuito NMAX de duas entradas e a sua tabela verdade	31
Figura 3.9: DLC1, DLC2 e DLC3	32
Figura 3.10: Representação de um multiplexador de 2 entradas (MUX 2:1) e seu esquemático.....	33
Figura 3.11: Diagrama de blocos e esquemático de um MUX 4:1.....	33
Figura 3.12: Representação de um QMUX 4:1	33
Figura 3.13: Multiplexador binário de 8 entradas MUX 8:1.....	34
Figura 3.14: Diagrama de blocos de um multiplexador quaternário de 16 entradas (QMUX 16:1).....	34
Figura 3.15: Esquemático do circuito multiplexador 4:1.....	35
Figura 3.16: Osciladores em anel quaternários construídos usando a) inversores em número ímpar, 5 Sucessores ($4n+1$ com $n=1$) e 3 Sucessores	36
Figura 3.17: Circuitos registradores construídos com número par de inversores e com 4 circuitos sucessores	37
Figura 3.18: Célula básica de um contador quaternário.....	38
Figura 3.19: Diagrama de blocos de um somador quaternário implementado segundo as equações da tabela 3.6	39
Figura 3.20: Um dos ramos da soma direta. Nesse exemplo, $A=1V$, $B=3V$ e $A+B=0V$	40
Figura 3.21: Somador direto.....	41

Figura 3.22: Somador parcial 2:2 implementado usando QMUX 4:1	42
Figura 3.23: Circuito avaliador de <i>Carry out</i> parcial.....	43
Figura 3.24: Somadores parciais 4:2 baseados em somadores parciais 2:2, dispostos em cascata e árvore	44
Figura 3.25: Somador parcial 2:2 final	44
Figura 3.26: Diagrama de blocos do circuito avaliador do <i>carry out</i> quaternário.....	45
Figura 3.27: Circuito de avaliação do <i>carry out</i> quaternário	45
Figura 3.28: Look-up table: Tabela verdade de uma função binária de 2 variáveis e a sua implementação em um MUX.....	46
Figura 3.29: Tabela verdade de uma função com duas entradas quaternárias e a implementação dessa função usando transistores de passagem.....	47
Figura 3.30: Esquemático de um circuito 2QLUT	48
Figura 4.1: Curva $I_{ds} \times V_{gs}$ dos 8 transistores usados	49
Figura 4.2: Curvas $I_{ds} \times V_{gs}$ para os transistores do circuito inversor.....	50
Figura 4.3: Análise DC do inversor quaternário.....	51
Figura 4.4: Transientes do sinal de saída para um inversor com os transistores nas dimensões mínimas.....	52
Figura 4.5: Transientes do sinal de saída para um inversor com larguras de canal de 0,54 μ m, 0,54 μ m, 0,54 μ m, 0,27 μ m, 0,54 μ m e 0,27 μ m para os transistores T1, T2, T3, T4, T5 e T6, respectivamente	54
Figura 4.6: Diferentes maneiras de contatar o substrato dos transistores em lógica quaternária.....	55
Figura 4.7: Transientes de saída do circuito repetidor.....	55
Figura 4.8: Tensões de saída de cada inversor em um oscilador em anel com 3 inversores quaternários	56
Figura 4.9: Entradas e saídas da simulação dos circuitos NMIN e NMAX.....	57
Figura 4.10: Funções de transferência dos circuitos DLCs.....	57
Figura 4.11: Equivalente binário do somador completo 5:2 quaternário.....	62
Figura 5.1: Arquitetura básica de um FPGA	66
Figura 5.2: Bloco lógico básico de um FPGA.....	67
Figura 5.3: Operação de soma de 3 números quaternários e sua implementação usando 2QLUTs	70
Figura 5.4: Operação de soma de 3 números quaternários e sua implementação usando 3QLUTs	71
Figura 5.5: Potência dissipada nos circuitos simulados usando diferentes blocos básicos	73
Figura 5.6: Número de transistores usados nos circuitos simulados usando diferentes blocos básicos.....	74
Figura 5.7: Atraso crítico dos circuitos simulados usando diferentes blocos básicos....	74

LISTA DE TABELAS

Tabela 2.1: Funções lógicas binárias de uma variável.....	15
Tabela 2.2: Funções lógicas binárias de duas variáveis.....	16
Tabela 2.3: Funções lógicas ternárias de 1 variável	24
Tabela 3.1: Tensões de limiar dos transistores usados	29
Tabela 3.2: Tabelas verdades dos diferentes DLC quaternárias.....	32
Tabela 3.3: Saídas DLC e seus negados (NDLC) para o circuito QMUX 4:1	35
Tabela 3.4: Seqüência de oscilação dos osciladores da figura 3.16.	37
Tabela 3.5: Tabela verdade de um somador parcial de 2 entradas	38
Tabela 3.6: Simetrias da tabela verdade do somador de 2 entradas	39
Tabela 3.7: Fontes de tensão ligadas às entradas do QMUX para realizar as funções Sucessoras	42
Tabela 4.1: Atrasos de propagação do somador MIN-MAX	58
Tabela 4.2: Atrasos de propagação do somador direto	59
Tabela 4.3: Atrasos do somador parcial 2:2 para transições na entrada A	60
Tabela 4.4: Atrasos do somador parcial 2:2 para transições na entrada B.....	60
Tabela 4.5: Atrasos de propagação do <i>carry</i> parcial	61
Tabela 4.6: Comparação entre as diversas realizações da função inversora.....	63
Tabela 4.7: Comparação entre as diversas realizações da função repetidora	63
Tabela 4.8: Comparação entre os diversos osciladores em anel	63
Tabela 4.9: Comparação entre as realizações do somador parcial 2:2	64
Tabela 4.10: Comparação entre os somadores quaternário e equivalente binário.....	64
Tabela 5.1: Comparação entre LUTs binárias e quaternárias	69
Tabela 5.2: Resultados do mapeamento lógico das funções soma3, soma4, soma5, soma6, soma22 e prod22 em LUTs quaternárias e binárias	72

RESUMO

Desde a década de 60, o aprimoramento das técnicas de fabricação de circuitos integrados que usam lógica binária tem levado ao aumento exponencial na densidade de dispositivos, melhoria do desempenho, redução da energia consumida e redução dos custos de fabricação dos circuitos integrados no estado da arte. Esse avanço tem sido alcançado historicamente pela miniaturização dos dispositivos que, já em escala nanométrica, começam a encontrar limites físicos para a sua redução. Com o intuito de dar continuidade ao avanço tecnológico, muitos trabalhos têm proposto a compactação da informação através do uso de lógica não binária como solução alternativa para a melhoria de desempenho de circuitos no estado da arte. Nesse sentido, diversos trabalhos foram desenvolvidos em diferentes tecnologias que vão de circuitos bipolares a dispositivos quânticos, entretanto, até o presente momento, nenhuma tecnologia demonstrou ao mesmo tempo os requisitos de desempenho, consumo, área e confiabilidade, necessários à aplicação em circuitos de alta escala de integração. Este trabalho apresenta uma nova família de circuitos de lógica quaternária com alto desempenho, baixos consumo e área e que usa tecnologia CMOS. Os circuitos desenvolvidos neste trabalho fazem uso de três fontes de alimentação e até oito diferentes transistores com diferentes tensões de limiar para realizar a lógica quaternária. São apresentados circuitos elementares como inversores e circuitos literais e com eles construídos circuitos aritméticos e multiplexadores. Os circuitos são simulados com a ferramenta SPICE usando a tecnologia TSMC 0,18 μm e os resultados são comparados com circuitos equivalentes em lógica binária. Na comparação de um somador completo quaternário de quatro *bits*, por exemplo, com o circuito equivalente em lógica binária, a implementação quaternária apresenta melhoria 55% na velocidade, 63% no consumo de potência e utiliza pouco mais de duas vezes o número de transistores. Este trabalho também propõe o uso de lógica quaternária em FPGA e são desenvolvidos blocos lógicos programáveis quaternários. Resultados de mapeamento lógico de circuitos aritméticos em blocos lógicos programáveis apresentam grande redução em área e consumo de potência na implementação quaternária quando comparado aos equivalentes binários. Em alguns circuitos quaternários, o consumo de potência e o número de transistores usados são reduzidos a 3% do consumo e do número de transistores usados nos circuitos equivalentes binários, enquanto o atraso crítico é duas vezes maior do que o atraso crítico binário.

Palavras-Chave: Lógica Multivalorada, Lógica Quaternária, Circuitos VLSI, FPGA.

Low-Power High-Performance Quaternary for VLSI Circuits

ABSTRACT

Since the decade of 60, the improvement of techniques for manufacturing integrated circuits that use binary logic has led to the exponential increase in the density of devices, improving performance, reducing energy consumption and reducing costs of manufacture of integrated circuits in the state of the art. This breakthrough has been achieved historically by the miniaturization of devices, already in nano, starting to reach physical limits to their reduction. In order to give continuity to technological advancement, many studies have proposed the compaction of information through the use of non-binary logic as an alternative for the performance improvement of the state of the art circuits. Accordingly, several studies have been developed in different technologies ranging from bipolar circuits to quantum devices, however, at the moment, no technology demonstrated at the same time the performance requirements, consumption, area and reliability necessary for the application in very large scale of integration. This paper presents a new family of quaternary logic circuits with high performance, low consumption and area, which uses CMOS technology. The circuits developed in this work make use of three power supplies and up to eight different transistors with different threshold voltages, to perform the quaternary logic. Elementary circuits such as inverters and literal circuits are presented and used to implement multiplexers and arithmetic circuits. The circuits are simulated with the SPICE tool using TSMC 0.18 μm technology and the results are compared with equivalent circuits in binary logic. Comparison of a quaternary full adder of four *bits*, for example, with the equivalent circuit in binary logic shows 55% improvement in speed and 63% in the power consumption for the quaternary implementation and it uses little more than twice the number of transistors. This paper also proposes the use of quaternary logic in FPGA and quaternary configurable logic blocks are developed. Logical mapping results of arithmetic circuits in configurable logic blocks show great reduction in area and power consumption of the quaternary implementation compared to the equivalent binary. In some quaternary circuits, the consumption of power and the number of transistors used are reduced to 3% of consumption and the number of transistors used in the binary equivalent circuits, while the critical delay is two times higher than the binary critical delay.

Keywords: Multiple Valued Logic, Quaternary Logic, VLSI Circuits, FPGA.

1 INTRODUÇÃO

Desde a década de 60, o aprimoramento das técnicas de fabricação de circuitos integrados, que usam lógica binária, tem levado ao avanço exponencial da redução das dimensões, aumento da densidade, melhoria do desempenho, redução da energia consumida e redução dos custos de fabricação dos dispositivos microeletrônicos. Esse avanço pode ser verificado pela “lei de Moore”, uma famosa previsão datada de 1965, feita pelo co-fundador da Intel, Gordon Moore (1965), que prevê a duplicação do número de dispositivos por chip a cada ano (mais tarde essa previsão foi ajustada para 18 meses (MOORE, 1975)). Essa lei empírica tem sido verificada na prática desde então, embora seja apenas uma tendência tecnológica e econômica e não uma lei física. As leis da Física, por sua vez, vêm apresentando barreiras cada vez mais difíceis de transpor e, em consequência, a lei de Moore, como conhecemos hoje, pode estar chegando ao fim, como já admitem alguns especialistas da indústria (FRANK, 2002). No que diz respeito aos níveis de consumo de potência e desempenho dos transistores convencionais, os limites vêm de leis fundamentais da termodinâmica e da mecânica estatística (FRANK, 2005).

Com os limites físicos cada vez mais próximos, a busca por soluções alternativas à tecnologia padrão existente (tecnologia planar do silício e circuitos digitais) tem se tornado um tópico em franco crescimento. Algumas soluções que abandonam parcial ou totalmente a tecnologia existente em silício têm sido propostas e demonstradas, contudo, ainda estão muito distantes dos requisitos de desempenho e confiabilidade necessários à aplicação industrial. Soluções que se baseiam na atual tecnologia planar do silício, partem de uma tecnologia já estabelecida, de comprovada confiabilidade e desempenho e são mais facilmente aceitas devido ao baixo custo de migração e ao aproveitamento de todo o conhecimento desenvolvido durante a era do silício.

A fim de continuar a melhoria dos atuais circuitos integrados e retardar a chegada de limites físicos, vários pesquisadores têm proposto a compactação de mais informação em uma única porta ou linha metálica usando mais de dois níveis discretos de sinais. O interesse nesse tipo de lógica, chamada lógica multivalorada ou lógica não-binária, tem aumentado recentemente e já existem realizações bem sucedidas comercialmente, como as memórias *Flash Multi-cell*, onde um *bit* pode guardar mais de 2 diferentes valores lógicos (BAUER et al., 1995). Essas memórias não binárias são produzidas por empresas como Intel, AMD, STMicroelectronics, Kingston, Fujitsu, Micron e Samsung entre outras.

O uso de lógica não binária em tecnologia CMOS se apresenta como uma promissora alternativa à atual lógica binária pela potencial melhoria na capacidade de processamento e por possíveis vantagens tecnológicas. O aumento da capacidade de processamento vem do uso de funções lógicas sem correspondência em lógica binária e entre as possíveis vantagens tecnológicas, consta a redução da rede de interconexões entre os componentes dos circuitos e destes com o mundo externo.

A alocação das conexões metálicas dentro do chip é um problema que está se tornando mais importante a cada nova geração de circuitos integrados. Com o aumento da densidade de componentes em um chip, também aumenta o número de fios necessários para as interconexões e a solução para alocar essas interconexões passa por aumentar o número de camadas de metais nos chips. A previsão é de aumento de mais de uma camada de metal a cada nova geração daqui para frente, segundo a Intel (KOBINSKY et al., 2006). O aumento das camadas metálicas adiciona custos à fabricação dos chips e reduz o rendimento das lâminas, além de afetar o desempenho devido ao aumento do produto resistência vezes a capacitância associado à rede de interconexões. Nos chips fabricados pela AMD em tecnologia SOI de 90nm, por exemplo, podem ser usadas até nove camadas de metal. No processador AMD Opteron Dual-core, por exemplo, onde são usados 233 milhões de transistores, é necessário aproximadamente 1,3 bilhão de contatos metálicos e existem cerca de 7 km de linhas metálicas por chip. A solução de utilizar lógica não binária é uma alternativa para reduzir esses números. Com a possibilidade de transmissão em cada interconexão de não apenas 2 níveis lógicos, mas sim 3 ou mais, o número de linhas metálicas necessárias pode ser reduzido significativamente. A redução das linhas metálicas, que é alcançada de forma natural com o uso de lógica não binária, pode resultar não só em aumento do desempenho, mas também redução do consumo (pela redução da energia dissipada nos metais) e redução dos custos de fabricação dos chips (pela necessidade de um menor número de camadas metálicas). Outro problema relacionado à conectividade é o reduzido número de pinos de saída frente à complexidade dos circuitos integrados. Diversas técnicas têm sido desenvolvidas para permitir a comunicação dos chips (nos quais pode haver centenas de milhões de transistores) com o mundo externo através de algumas centenas de pinos. Se a cada pino é permitida a transmissão de mais de dois níveis lógicos, uma melhor comunicação entre chips seria possível ou o número de pinos poderia ser reduzido, diminuindo o tamanho dos chips encapsulados.

Além das vantagens tecnológicas, o potencial ganho em capacidade de processamento é a característica mais poderosa oferecida pela lógica não binária para a melhoria em desempenho dos atuais circuitos. Utilizando lógica não binária, inúmeros trabalhos teóricos demonstram a realização de operações lógicas com maior eficiência do que usando lógica binária (CHALMERS, 2007; LABLANS, 2007). Com a realização de operações usando menor número de componentes, é possível o ganho em velocidade, consumo e área ao mesmo tempo. É também uma característica importante o fato de existir, em lógica não binária, operações sem correspondência em lógica binária. Essas operações podem ser usadas para resolver problemas de maneiras alternativas, com maior eficiência do que usando lógica binária.

Fazendo uso de lógica não binária, diversos trabalhos têm sido desenvolvidos nas últimas décadas, como está mostrado em detalhes no capítulo 2, mas apesar do expressivo número de realizações, ainda não existe uma tecnologia que tenha se mostrado suficientemente boa para substituir a atual tecnologia binária. Tendo em vista as vantagens em potencial do uso de lógica não binária e a falta de bons circuitos

disponíveis para a sua realização, esse trabalho propõe uma nova família de circuitos para a implementação de lógica quaternária de alto desempenho e baixo consumo.

Os circuitos desenvolvidos neste trabalho utilizam transistores com diferentes tensões de limiar e três fontes de alimentação para realizar lógica quaternária. Circuitos básicos são demonstrados, e a partir destes, circuitos aritméticos são projetados e simulados para comparação com os existentes circuitos de lógica binária.

Este trabalho também apresenta a aplicação de lógica quaternária em circuitos de lógica programável. O estudo é feito em circuitos que usam malhas de elementos lógicos programados por campo elétrico (FPGA). Essa aplicação, devido à suas características de grande demanda por conectividade e programabilidade, se constitui uma aplicação natural para lógica não binária.

O trabalho começa com uma breve introdução à lógica não binária e apresenta as principais realizações até o momento nessa área (capítulo 2). No capítulo 3 estão descritos os circuitos desenvolvidos neste trabalho. São apresentados os transistores usados, portas lógicas primitivas, circuitos aritméticos e circuitos para aplicação direta em lógica programável. O capítulo 4 apresenta as simulações elétricas dos circuitos apresentados no capítulo 3 e alguns resultados comparando-os com circuitos equivalentes em lógica binária. No capítulo 5 está apresentada a aplicação de lógica quaternária em circuitos FPGAs com simulações elétricas comparativas e alguns resultados de mapeamento lógico. Finalmente, o capítulo 6 traz as conclusões e idéias de continuação do trabalho.

2 CIRCUITOS DE LÓGICA MULTIVALORADA

2.1 Lógica multivalorada

O desenvolvimento e a aplicação da lógica binária estão na base de toda a revolução que a tecnologia da informação proporcionou ao mundo nas últimas décadas. Essa revolução teve início quando Claude Elwood Shannon expressou as características de chaves elétricas em álgebra booleana (no Instituto de Tecnologia de Massachusetts na década de 40 do século passado). A partir daí, a eletrônica digital se desenvolveu e evoluiu de chaves eletro-mecânicas aos atuais circuitos VLSI. Quase a totalidade dos circuitos que usam níveis discretos comercializados nos dias de hoje usa lógica binária, mas apesar do sucesso obtido em sua aplicação, não é necessariamente a mais eficiente e poderosa lógica de chaveamento. Existem características vantajosas inerentes à lógica não binária que não podem ser alcançados por lógica binária, como a existência de um maior número funções lógicas e a possibilidade do uso de diferentes operações, como por exemplo, aritmética residual (WEI et al., 1998) de maneira mais eficiente.

Uma breve apresentação das funções lógicas existentes em lógica não binária é suficiente para dar uma boa noção das novas possibilidades de seu uso e das suas potencialidades. O número de funções lógicas para um número n de entradas em uma lógica de base B é dado por $B^{(B^n)}$. Em lógica binária, as possibilidades de saída para uma função de uma variável binária são apenas quatro e estão mostradas na tabela 2.1:

Tabela 2.1: Funções lógicas binárias de uma variável

<i>Variável</i>	<i>Função 1</i>	<i>Função 2</i>	<i>Função 3</i>	<i>Função 4</i>
0	0	0	1	1
1	0	1	0	1

Entre essas possíveis funções lógicas de 1 variável, pode-se encontrar a função de negação (função 3 da tabela 2.1) que é uma das funções de maior importância em lógica binária. Para duas variáveis binárias, o número de funções é 16 ($2^{(2^2)}$) como se pode ver na tabela 2.2:

Tabela 2.2: Funções lógicas binárias de duas variáveis

Variável A	Variável B	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Entre elas podemos destacar as funções AND (F2), OR (F8), NAND (F15), NOR (F9) e XOR (F7). A partir desse pequeno conjunto de funções, toda a eletrônica digital é construída. Com um número tão limitado de funções lógicas, certas tarefas não são realizadas da forma mais eficiente possível. Em contrapartida, em lógica não binária, existe uma grande gama de novas funções que não possuem correspondência em lógica binária com o mesmo número de variáveis e podem ser usadas para a realização de operações lógicas de maneira mais eficiente. Os primeiros desenvolvimentos em circuitos integrados de lógica não binária foram realizações em lógica ternária, ou seja, usando números de base 3 (DUNDERDALE, 2002; NAZARALE et al., 1974). Em lógica ternária, para uma função de 1 entrada ternária e 1 saída ternária, podemos ter 27 (3^3) diferentes funções. Utilizando como notação 0, 1 e 2 para os três valores lógicos, as possíveis funções lógicas estão mostradas na tabela 2.3.

Tabela 2.3: Funções lógicas ternárias de 1 variável

entrada	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	...
0	0	0	0	0	0	0	0	0	0	1	1	1	1	...
1	0	0	0	1	1	1	2	2	2	0	0	0	1	...
2	0	1	2	0	1	2	0	1	2	0	1	2	0	...

...	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27
...	1	1	1	1	1	2	2	2	2	2	2	2	2	2
...	1	1	2	2	2	0	0	0	1	1	1	2	2	2
...	1	2	0	1	2	0	1	2	0	1	2	0	1	2

Pode-se perceber a partir da tabela 2.3 que uma série de novas funções surge quando é utilizada a base três, sendo a maioria delas sem equivalente em lógica binária, como as funções cíclicas Sucessor (F16) e Antecessor (F20). Com duas entradas ternárias e uma saída ternária, o número de funções lógicas possíveis é 19.683 (3^{3^2}) e novas operações podem ser vistas como por exemplo as funções MAX e MIN, cujas saídas são o maior e o menor valor das duas entradas, respectivamente.

Usando lógica quaternária (4 níveis lógicos distintos), o número de funções existentes para operação de 1 entrada quaternária e 1 saída quaternária é 256. Para uma

função de 2 variáveis quaternárias esse número cresce para 4^{16} . Como exemplos, na figura 2.1 estão apresentadas as funções lógicas MIN e MAX quaternárias.

		MAX (X,Y)			
		X			
Y	0	0	1	2	3
	1	1	1	2	3
	2	2	2	2	3
	3	3	3	3	3

		MIN (X,Y)			
		X			
Y	0	0	0	0	0
	1	0	1	1	1
	2	0	1	2	2
	3	0	1	2	3

Figura 2.1: Funções lógicas MIN e MAX quaternárias

2.2 Circuitos existentes de lógica multivalorada

Nas últimas 4 décadas, muitos trabalhos têm sido desenvolvidos usando circuitos não binários para realizar operações lógicas. Uma visão geral dos primeiros circuitos desenvolvidos pode ser encontrada em Hurst (1984), Smith (1981) e Butler (1995). Antes disso, as atrações matemáticas de representações numéricas multivaloradas já tornavam esse tema um tópico de estudo no campo teórico desde a década de 20 (ŁUKASIEWICZ, 1920). Na prática, o desenvolvimento de circuitos não binários teve início nos anos 70, com os primeiros trabalhos em projetos de circuitos ternários (DUNDERDALE et al., 2002; NAZARALE, 1974). Depois, um grande número de circuitos não-binários foi desenvolvido usando lógica de injeção integrada bipolar (I²L) (PUGSLEY et al., 1978; DAO et al., 1977), lógica de emissor acoplado bipolar (ECL) (ETIEMBLE et al., 1987), dispositivo de acoplamento de carga (CCD) (KERKHOFF et al., 1979; KERKHOFF 1981) e MESFETs (TRONT et al., 1979). Essas tecnologias, no entanto, permaneceram nos laboratórios e nunca se mostraram suficientemente promissoras para despertar o interesse da indústria, que ao mesmo tempo avançava rapidamente em sistemas digitais (binários). De fato, até hoje não está desenvolvida uma solução para o uso de lógica não binária compatível com os requerimentos necessários à alta escala de integração. Esses requerimentos se referem a circuitos que possuam, ao mesmo tempo, baixo custo, confiabilidade, pequena área e alto desempenho.

Acompanhando a evolução dos sistemas eletrônicos, a lógica não binária se desenvolveu também em tecnologia MOS planar. Circuitos multivalorados começaram a ser concebidos em tecnologias NMOS e CMOS usando níveis lógicos representados por sinais discretos de tensão ou de corrente.

Além de tecnologia planar do silício, outras tecnologias não convencionais foram aplicadas em lógica não binária. Entre elas, podem-se citar circuitos que usam dispositivos quânticos funcionais como diodos e transistores de tunelamento ressonante (WAHO, 1995; BABA, 1999) e também circuitos baseados em transistores de efeito de campo construídos com nano tubos de carbono (RAYCHOWDHURY et al., 2005).

Nas seções seguintes, estão apresentadas em maiores detalhes as principais realizações em lógica não binária existentes na literatura.

2.2.1 Circuitos multivalorados modo-corrente

Circuitos de lógica multivalorada (MVL) modo-corrente têm sido construídos desde os primeiros circuitos MVL bipolares, entretanto, foi com o desenvolvimento de circuitos modo-corrente CMOS, introduzidos em 1983 (FREITAS et al.), que essa tecnologia começou a apresentar alguns requisitos necessários para integração em muito alta escala (VLSI). Com o crescente avanço e sucessivas realizações nessa área desde então, como pode ser visto em (CURRENT, 1994), MVL modo-corrente passou a ser considerada a mais promissora e com maior potencial de comercialização entre as diversas tecnologias desenvolvidas para realizar lógica não binária (DUBROVA, 1999).

Circuitos MVL modo-corrente usam, como níveis lógicos, correntes elétricas que são múltiplos inteiros de um valor de corrente de referência e podem ser usadas de modo bidirecional. Entre as vantagens em potencial de se usar esses circuitos, a principal é a facilidade de se realizar a operação de soma simplesmente por ligações elétricas entre diferentes ramos do circuito, aplicando a lei das correntes de Kirchhoff.

Os primeiros circuitos quaternários modo-corrente (CMVL) usando tecnologia CMOS foram desenvolvidos e apresentados como um híbrido entre lógica binária e processamento analógico de sinais. Essa tecnologia, em sua concepção, não pretendia substituir a lógica binária por completo e sim buscar aplicações onde MVL fosse realmente superior à lógica binária em desempenho e tornar essa tecnologia compatível com a tecnologia CMOS binária, possibilitando com isso, a construção de circuitos ao mesmo tempo binários e não binários, que usam o tipo de lógica mais eficiente para cada tarefa. Para isso, circuitos que fazem a interface entre a parte binária e a quaternária do circuito, chamados de codificadores e decodificadores, são extremamente importantes.

Em lógica multivalorada modo corrente, as correntes podem ser copiadas, escalonadas, complementadas, ou ter o sinal trocado algebricamente usando circuitos espelho de corrente realizados em tecnologia MOS. A função espelho de corrente junto com circuitos comparadores de corrente (ou comparadores de limiar), estão na base desses circuitos multivalorados modo-corrente. O comparador de corrente é uma componente chave no projeto de circuitos CMVL, pois a limitação no comparador determina a habilidade da tecnologia em discriminar os diferentes níveis lógicos. A forma mais simples de comparador de corrente, mostrada na figura 2.2, pode ser vista em mais detalhes em (FREITAS et al., 1983; 1984).

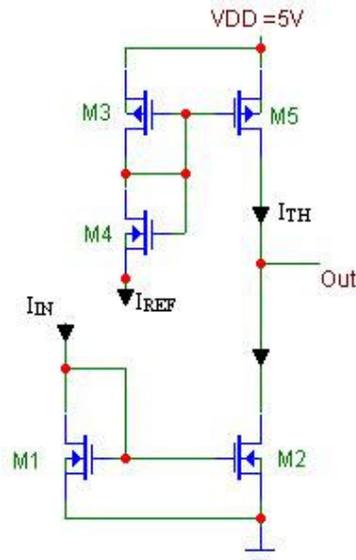


Figura 2.2: Comparador de correntes

No comparador, a corrente de entrada passa pelo transistor NMOS M1, que tem seu dreno conectado à sua porta e a fonte ligada na terra. A corrente de entrada determina a tensão na porta do M1 e, por consequência, a tensão na porta do M2 que é um transistor NMOS conectado de modo a replicar a corrente que passa em M1 (espelho de corrente). Da mesma forma o transistor PMOS M5 replica a corrente no par M3-M4 que é a corrente de referência a ser comparada à corrente de entrada. Os drenos do PMOS M5 e do NMOS M2 são ligados e nesse nó é estabelecida a tensão de saída do circuito comparador. A saída tem valor lógico ALTO quando a corrente de entrada é menor do que a corrente de referência e tem valor BAIXO quando a corrente de entrada é maior do que a corrente de referência.

O circuito comparador converte um sinal de corrente em um sinal de tensão. Essa conversão é usada como interface de comunicação entre circuitos binários e circuitos quaternários modo-corrente. Baseados no circuito comparador de corrente, foram desenvolvidos os circuitos codificador e decodificador (FREITAS et al., 1983; 1984; 1984) entre níveis lógicos binários e quaternários onde um *bit* quaternário corresponde a 2 *bits* binários.

O circuito lógico somador completo também foi demonstrado em vários trabalhos usando MVL modo-corrente (FRIEDMAN et al., 1977; CURRENT, 1980; WHEATON et al., 1984; CURRENT et al., 1985; CURRENT, 1990). Esses somadores, como são erroneamente chamados na literatura de somadores completos quaternários, são circuitos com 3 entradas e 2 saídas. Em lógica de base b , um somador completo deve ter $b+1$ entradas e duas saídas. Esse circuito soma as duas entradas quaternárias A e B mais o *carry* de entrada, que é binário, e gera o *bit* quaternário de soma S mais o *bit* binário representando o *carry* de saída C . Um exemplo de somador com base em comparadores e espelhos de corrente está mostrado na figura 2.3. Nesse circuito, a corrente I_{in} é a soma das correntes A e B e *Carry* de entrada do somador. A e B podem assumir valores de 0 , I , $2I$ e $3I$ e o *carry* de entrada pode ter os valores 0 e I . Dessa forma I_{in} pode assumir 8 valores de corrente, 0 , I , $2I$, $3I$, $4I$, $5I$, $6I$ e $7I$. Sete transistores NMOS idênticos replicam a corrente I_{in} e usando 7 comparadores de corrente, nos valores $0,5I$, $1,5I$, $2,5I$, $3,5I$, $4,5I$, $5,5I$ e $6,5I$, são usados para determinar os valores

lógicos dos nós A, B, C, D, E, F e G, respectivamente. O valor de D é usado para determinar o *carry* de saída controlando a chave que conecta o *carry* de saída à uma corrente I. Quando D está BAIXO o *carry* de saída está ALTO (valor lógico I) e vice-versa. Para determinar o valor de soma, uma lógica é feita usando todos os valores de A a G da seguinte maneira:

$$\bar{X} = (A + \bar{D})(E)$$

$$\bar{Y} = (B + \bar{D})(F)$$

$$\bar{Z} = (C + \bar{D})(G)$$

X, Y e Z são usados para controlar as chaves que ligam a saída a três correntes de valor I cada. Essas correntes são somadas formando a saída que pode assumir valores 0, I, 2I e 3I.

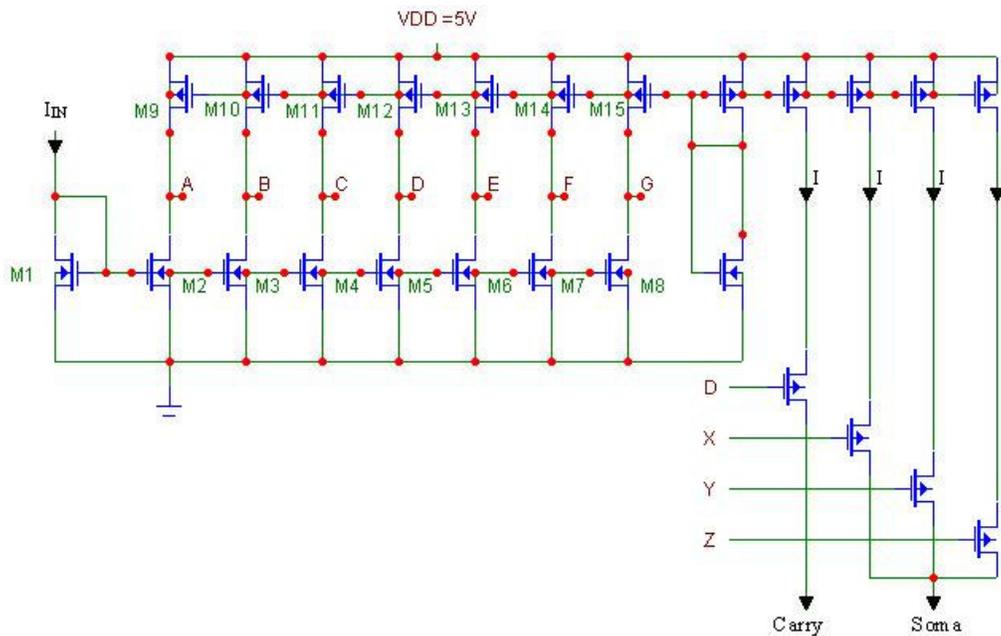


Figura 2.3: Somador quaternário modo-corrente

Além de somadores, *latches* modo-corrente (CURRENT, 1989; CURRENT et al. 1991), somador com *latch* (CURRENT, 1990), conversor analógico-digital quaternários modo-corrente (CURRENT, 1992) e circuitos multiplicadores (KAWAHITO et al., 1987) são outros exemplos de circuitos quaternários CMOS modo-corrente bidirecionais encontrados na literatura.

A partir do desenvolvimento dessa família, a área de circuitos não binários modo corrente cresceu e uma série de novos circuitos e famílias lógicas foram desenvolvidas. Entre as realizações estão somadores completos com dígito assinado (GONZALEZ, 1998; 2005), lógica diferencial *dual-rail* (HANYU et al., 1998; 1999), *Dual Rail Source Coupled*, *Dynamic Source-coupled logic* (HANYU et al., 2003).

Embora existam realizações em lógica não binária modo-corrente apresentando bons desempenhos, esse tipo de circuito tem uma desvantagem inerente à sua natureza que é o consumo estático de potência, pois necessitam de uma corrente elétrica fluindo para definir seus níveis lógicos e realizar operações lógicas. Uma vez que a dissipação de

potência é um problema cada vez mais severo nos circuitos integrados em ultra-alta escala de integração, grande parte do esforço em aprimorar tais circuitos é empregado em técnicas para reduzir o consumo de potência para tornar viável sua aplicação, entretanto, nenhuma solução definitiva para esse problema foi encontrada até agora. Algumas soluções buscam reduzir o consumo estático pela redução dos valores nominais das correntes representativas dos níveis lógicos, mas tal solução reduz as margens de ruídos e a detecção do nível lógico é prejudicada.

2.2.2 Circuitos multivalorados modo-tensão

Circuitos não binários que transmitem e processam sinais de tensão também foram demonstrados usando tecnologias NMOS e CMOS. As tensões podem ser obtidas no circuito através de fontes externas ou através de circuitos divisores de tensão. As principais realizações são apresentadas a seguir.

Em 1986, um método de implementar circuitos lógicos quaternários modo-tensão usando transistores MOS de canal-n foi apresentado por Yasuda et al. (1986). Os circuitos usam 3 diferentes transistores de enriquecimento com diferentes tensões de limiar e um transistor de depleção. Esse trabalho apresentou circuitos lógicos básicos como inversor quaternário, NMIN, NMAX, Sucessor e também um circuito genérico capaz de realizar qualquer função lógica quaternária. A figura 2.4.a) mostra o circuito inversor quaternário que é a forma elementar dessa tecnologia e onde são baseados os outros circuitos lógicos. A tensão de saída é determinada por uma divisão da tensão de alimentação (VDD) entre o transistor de depleção (T_0) e os demais transistores do circuito. Os transistores de enriquecimento têm V_{ts} de 0,5, 1,5 e 2,5 relativos aos sinais lógicos 0, 1, 2 e 3 e seus valores estão apresentados na figura 2.4.b). Dessa forma, para uma tensão V aplicada à porta dos transistores, estarão conduzindo os transistores cujo V_t é menor do que V . Usando apropriadamente os três tipos de transistores de enriquecimento, se pode ter para os quatro valores de entrada do circuito, quatro diferentes resistências equivalentes no ramo inferior do divisor de tensão. Dimensionando os transistores de forma correta, é realizado o circuito inversor quaternário. A tabela verdade e o diagrama de estados de operação dos transistores controlados pela entrada estão mostrados na figura 2.4.c). Os demais circuitos são realizados seguindo o mesmo princípio usado na realização do inversor. Esses circuitos foram simulados em Spice2 e implementados em tecnologia $4\mu\text{m}$ com VDD entre 5V e 12V. O resultado para o atraso em todos os transientes apresenta como pior caso 48ns e a área do circuito pode ser estimada a partir da área das portas que é igual a $25L^2$ (com $L = 4\mu\text{m}$).

Circuitos em tecnologia NMOS como os apresentados em (YASUDA et al., 1986) têm a desvantagem do consumo estático, pois uma corrente elétrica fluindo entre VDD e terra é necessária para a avaliação da tensão de saída dos circuitos. O consumo estático é uma característica inerente aos circuitos modo-corrente, mas em modo tensão é possível evitá-lo.

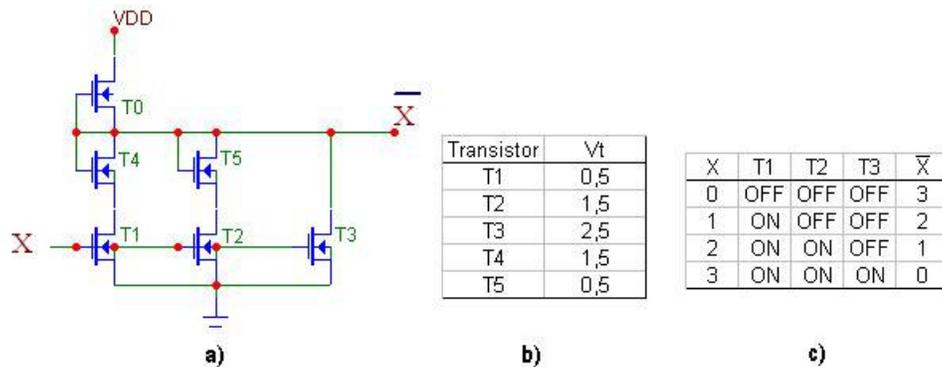


Figura 2.4: Inversor quaternário: a) Circuito, b) tensão de limiar dos transistores e c) relação entre sinal de entrada, os estados de operação dos transistores controlados pela entrada e sinais de saída (YASUDA et al., 1986)

Nesse sentido, uma série de circuitos modo-tensão sem consumo estático significativo foi desenvolvida. Uma família de circuitos lógicos ternários de baixo consumo foi apresentada em (HEUNG et al., 1985). Nessa tecnologia, são usados transistores de depleção e enriquecimento PMOS e NMOS e duas fontes externas de alimentação para determinar os três valores lógicos, que são $-1V$, $0V$ e $+1V$. A figura 2.5 mostra um inversor ternário onde são usados 4 transistores, 2 PMOS e 2 NMOS, um de depleção e um de enriquecimento para cada tipo, para realizar uma inversão diametral, ou seja, para $-1V$, $0V$ e $+1V$ na entrada, temos $+1V$, $0V$ e $-1V$ na saída. Ao aplicar $+1V$ na entrada, o transistor Q2 conduz e a saída é ligada à fonte de alimentação de $-1V$. Aplicando $-1V$ na entrada, o transistor Q1 conduz e liga a saída à fonte de alimentação de $+1V$. Quando a entrada é mantida a $0V$, os dois transistores de depleção mantêm a saída ligada em $0V$.

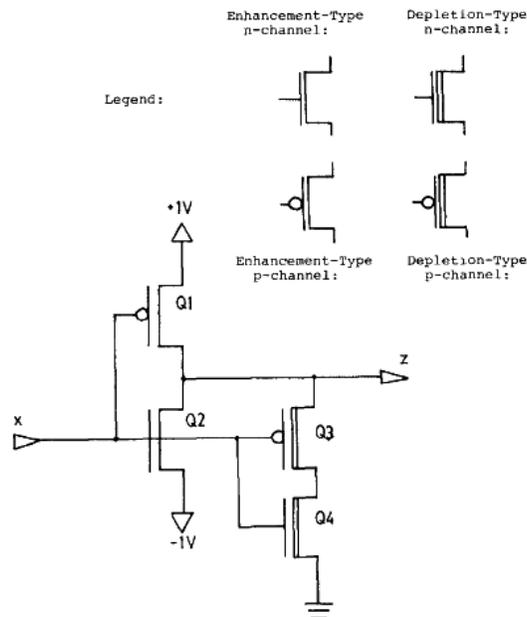


Figura 2.5: Inversor ternário diametral

Com essa tecnologia, são apresentados em (HEUNG et al., 1985), os circuitos inversor ternário positivo, inversor ternário negativo e as portas AND e OR ternárias. Esses circuitos básicos são usados para realizar um somador completo ternário.

Outra família de circuitos CMOS modo-tensão, desenvolvidos por Thoidis e colaboradores (THOUIDIS et al., 1998; 2001; 2004), apresentou somadores quaternários de baixo consumo que também usam transistores de enriquecimento e depleção. Essa família se baseia na lógica e nos circuitos desenvolvidos em (WATANABI et al., 1897), onde são apresentados circuitos que operam funções lógicas similares de forma dinâmica. Em (WATANABI et al., 1897), são usados duas fases de relógio para implementar circuitos dinâmicos usando uma técnica de controle de carga similar à técnica de pré-carga usada em circuitos binários. Os circuitos apresentados por Thoidis et al. e colaboradores apresentam formas mais eficazes de realizar as operações descritas em (WATANABI et al., 1897) e de forma assíncrona.

Os circuitos de Thoidis e colaboradores, usam 2 transistores PMOS e 2 transistores NMOS, um de depleção e um de enriquecimento em cada tipo. Os níveis lógicos são representados por sinais de tensão de 0V, 1V, 2V e 3V. Como exemplos de circuitos dessa família, estão mostrados na figura 2.6 os circuitos MIN e MAX que usam apenas 3 transistores para realizar cada uma dessas funções. Os transistores PMOS1 e NMOS1 são os do tipo depleção e PMOS e NMOS são transistores de enriquecimento com V_{ts} entre 0 e 1V. Nesses circuitos, uma característica inovadora é que os sinais de entrada são aplicados na porta e também na fonte de alguns transistores. No circuito MAX (fig. 2.6.a), o transistor P2 transmite o sinal de Y para a saída Z_1 sempre que X é menor do que Y. Se X é maior ou igual a Y, P2 abre e o sinal de saída é igual ao sinal da entrada X. O circuito MIN (fig. 2.6.b) opera de modo similar onde N1 liga a saída à entrada X quando X é menor do que Y, enquanto P2 e N2 transmitem Y à saída sempre que Y é menor ou igual a X.

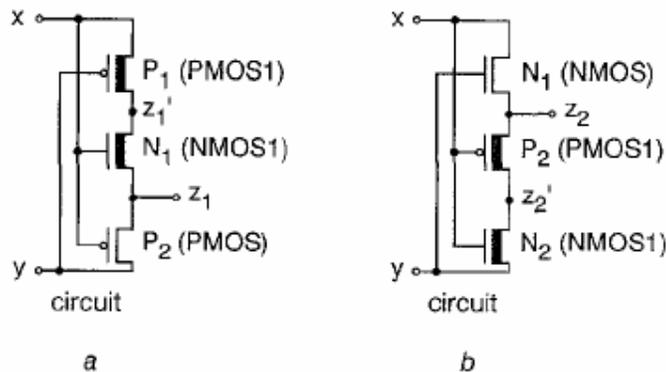


Figura 2.6: Circuitos a) MAX e b) MIN (THOUIDIS et al., 1998)

Outros circuitos básicos apresentados por Thoidis e colaboradores são os circuitos Literais, conversores de nível, soma truncada e diferença truncada. Todos esses circuitos são empregados na construção do primeiro somador completo quaternário estático, sem consumo estático de potência apresentado na literatura (THOUIDIS et al., 2004). Esse somador foi demonstrado em simulações com ferramentas SPICE, usando tecnologia 0,7 μm . As simulações mostram um atraso crítico de 15,9 ns, o que ainda é muito alto em relação aos circuitos binários que na mesma condição apresentam atrasos da ordem de 3 ns. Outro ponto negativo é que nessa tecnologia não é apresentada uma família

completa de circuitos, capaz de realizar qualquer função lógica e tampouco o circuito elementar inversor foi demonstrado.

Um novo tipo de circuito que usa transistores Neuron MOS (vMOS) (SHIBATA et al., 1993) foi desenvolvido e usado para realizar lógica multivalorada. Alguns circuitos funcionais foram apresentados em (SHEN et al., 2000; SUZUKI et al., 1993) e mais recentemente novos circuitos foram apresentados usando transistores de dupla passagem vMOS (PARK et al., 2004). Neuron MOS é um tipo de transistor com várias entradas aplicadas em uma porta de transistor que transmitem a informação para outra porta flutuante (floating gate). Esse transistor se caracteriza por ter a tensão de limiar variável alcançada controlando as tensões de pelo menos uma de suas entradas. Circuitos usando vMOS para realizar lógica multivalorada têm a vantagem de usar o processo CMOS com dupla camada de poli-silício sem a necessidade de usar múltiplas implantações iônicas para ajustes dos $V_{t,s}$. A estrutura básica de um vMOS está mostrada na figura 2.7.

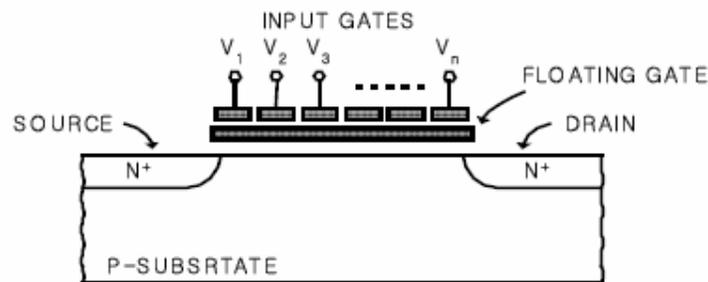


Figura 2.7: A estrutura básica de um Neuron MOS

Um exemplo de como o vMOS pode ser usado para realizar circuitos lógicos pode ser visto na realização dos circuitos detectores de limiar (DLCs) apresentados em (PARK et al., 2004). O circuito DLC usando vMOS está mostrado na figura 2.8. Com o mesmo circuito se pode realizar o detector de limiar para 0,5V, 1,5V e 2,5V ao se aplicar diferentes tensões nas portas de entrada de referência do circuito (V_{b1} e V_{b2}) como mostra a tabela 2.3.

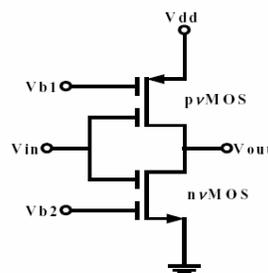


Figura 2.8 DLC usando vMOS

• Transistor de dupla passagem é um tipo de configuração para lógica de transistores de dupla passagem que substitui com vantagem os usuais transistores de dupla passagem NMOS e CMOS em margem de erro e degradação do sinal, respectivamente (SUZUKI, 1993).

Tabela 2.3: Valores de tensão aplicados em V_{b1} e V_{b2} para cada DLC

$V_{DD}=3.3V$	V_{TC}	V_{b1}	V_{b2}
$V_{th}(0.5)$	0.5V	3V	2V
$V_{th}(1.5)$	1.5V	2.2V	0.8V
$V_{th}(2.5)$	2.5V	1V	0V

A figura 2.9 mostra o símbolo de cada detector de limiar e suas funções de transferência.

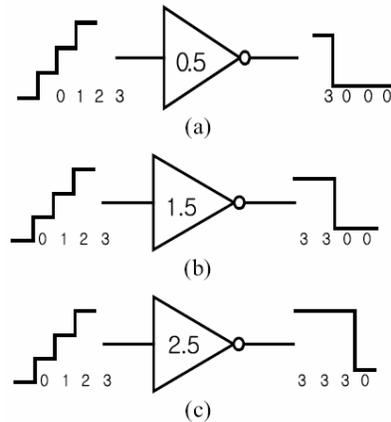


Figura 2.9: Sinais de entrada, símbolo e sinais de saída para os circuitos detectores de limiar de (a) 0,5, (b) 1,5 e (c) 2,5

Usando ν MOS como transistores de dupla passagem, foram desenvolvidos os circuitos MIN/NMIN, MAX/NMAX, soma truncada e diferença truncada. Todos esses circuitos foram simulados em (PARK et al., 2004) com HSPICE usando tecnologia de 0,35 μ m. Como principais resultados foram apresentados o consumo de 4,898 μ W dos circuitos NMIN e NMAX na frequência de 0,25 MHz. Essa tecnologia se mostra promissora, mas ainda apresenta níveis de consumo muito altos e baixo desempenho quando comparados aos circuitos binários.

Mais recentemente, foram desenvolvidos circuitos multivalorados usando lógica de limiar capacitiva (CTL) (SCHMID, 2004). CTL surgiu nos últimos anos como uma implementação capacitiva de funções lógicas de limiar e foi extensivamente descrita em (OZDEMIR et al., 1996; LEBLEBICI et al., 1998). Foram demonstrados em (SCHMID et al., 2004) circuitos literais, NOT, NAND e NOR.

Neste capítulo, foram apresentadas as principais realizações em lógica não binária e se pode ver uma diversidade de tecnologias e circuitos desenvolvidos nessa área. Apesar do expressivo número de realizações, ainda não existe uma tecnologia que tenha se mostrado suficientemente boa para substituir a atual tecnologia binária. Existem circuitos de bom desempenho em lógica quaternária modo-corrente, mas o consumo estático torna inviável sua aplicação em larga escala de integração e a solução natural, que é o uso de circuitos modo-tensão, ainda não apresenta circuitos que combinem alto desempenho e baixo consumo, além de pequena, robustez e completude. Tendo em vista as vantagens em potencial do uso de lógica não binária e a falta de bons circuitos disponíveis para a sua realização, esse trabalho propõe uma nova família de circuitos para a implementação de circuitos quaternários CMOS modo-tensão de alto desempenho e baixo consumo.

3 CIRCUITOS QUATERNÁRIOS PROPOSTOS

3.1 Circuitos Elementares

Os circuitos propostos neste trabalho utilizam tecnologia CMOS para realizar lógica quaternária de modo tensão. Para isso são usados três diferentes fontes de alimentação, que juntamente com o terminal comum (0V) formam os quatro níveis lógicos, e transistores com diferentes tensões de limiar (V_i). Os circuitos elementares quaternários podem ser entendidos como uma generalização dos circuitos básicos binários de lógica estática. Circuitos de lógica estática, de modo geral, são constituídos por uma rede de transistores entre a saída e VDD e outra rede de transistores entre a saída e a terra (RABAEY, 1996), como mostra a figura 3.1. Para cada sinal de entrada (ou conjunto de sinais de entrada), apenas uma rede está conduzindo e a outra está aberta.

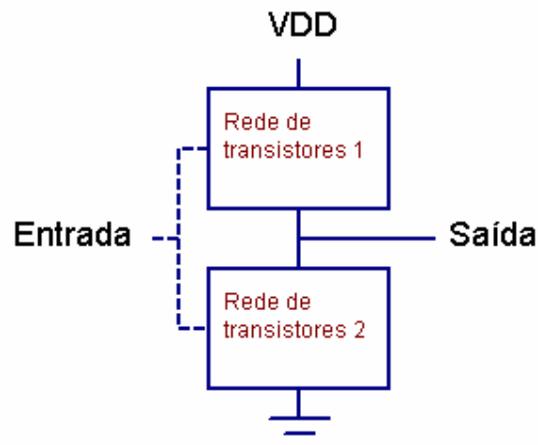


Figura 3.1: Estrutura de um circuito de lógica estática binário

No circuito inversor binário, cada rede de transistores é constituída por apenas um transistor. Esse circuito é implementado usando um transistor PMOS, cuja tensão de limiar é negativa, e um transistor NMOS, cuja tensão de limiar é positiva, ligados conforme esquemático da figura 3.2.

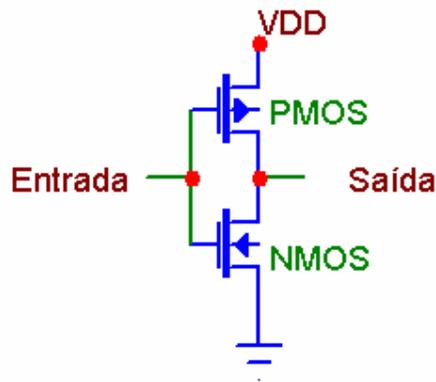


Figura 3.2: Inversor binário

Em lógica quaternária modo-tensão, existem quatro valores de tensão representando os níveis lógicos. Os sinais de tensão são fornecidos por três fontes externas de alimentação (VDD1, VDD2 e VDD3) mais o terminal comum. Dessa forma, é necessária uma rede de transistores para ligar cada um dessas fontes à saída. A figura 3.3 mostra a estrutura de um circuito de lógica estática de quatro níveis.

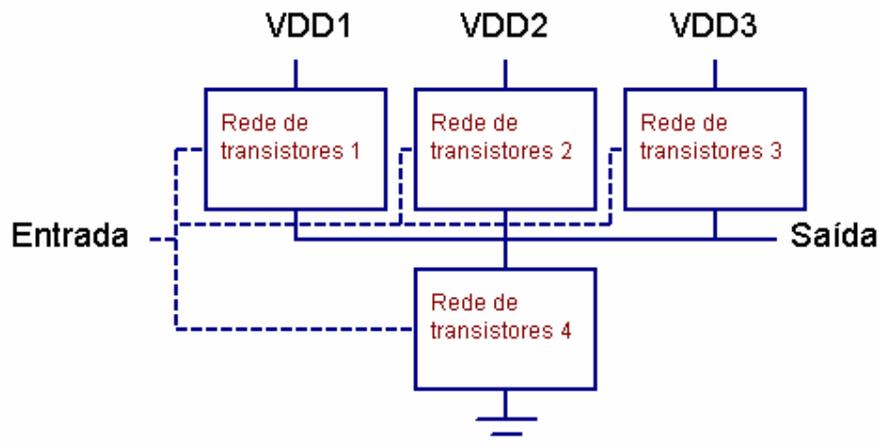


Figura 3.3: Esquemático de um circuito de lógica estática quaternário

Ao contrário das redes binárias, que precisam de apenas um valor de tensão de referência para conduzirem acima ou abaixo dele, as redes de transistores quaternários devem conduzir em um intervalo finito de tensões na entrada, ou seja, precisam de dois valores de tensão de referência. Cada rede deve estar fechada para apenas um nível elétrico aplicado na entrada e aberta para os níveis lógicos menores ou maiores do que esse. Essa característica pode ser alcançada combinando transistores PMOS e NMOS em cada rede de transistores e usando diferentes tensões de limiar. Utilizando diferentes transistores com diferentes tensões de limiar pode-se implementar redes de transistores para realizar funções quaternárias unívocas, como pode ser visto em maiores detalhes na descrição do inversor quaternário a seguir.

3.1.1 Inversor Quaternário

Em lógica não binária, a função negação perde o sentido. Com o uso de mais de dois níveis lógicos, a função usualmente referida como negação é a inversão diametral cuja tabela verdade está mostrada na figura 3.4.b). O circuito inversor quaternário é construído utilizando seis transistores, sendo três PMOS e três NMOS, cada um com uma diferente tensão de limiar, conectados conforme o esquemático mostrado na figura 3.4.a). Neste circuito, os níveis lógicos 0, 1, 2 e 3 correspondem à sinais de tensão de 0V, 1V, 2V e 3V.

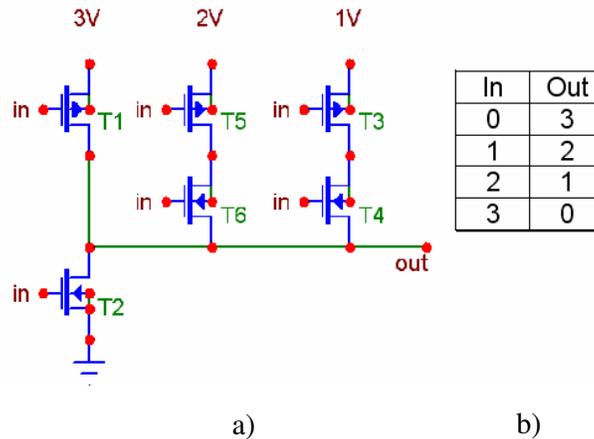


Figura 3.4: (a) Circuito Inversor e (b) sua tabela verdade

As tensões de limiar dos transistores do inversor quaternário são ajustadas de modo às suas curvas características se pareçam com as curvas da figura 3.5. Da forma como são dispostos os transistores no circuito inversor quaternário, é assegurado que apenas uma fonte de tensão está conectada à saída para cada valor de tensão aplicado nas entradas do circuito. A partir das curvas da figura 3.5, pode-se entender melhor como funciona o circuito. Ao aplicar na entrada do inversor um sinal de 0V, o transistor T1 está conduzindo e conecta a fonte de tensão de 3V à saída. Ao mesmo tempo, os transistores T2, T4 e T6 estão abertos, impedindo o caminho entre qualquer outra fonte de tensão e a saída e entre quaisquer duas fontes de tensão. Dessa forma, com a entrada mantida em 0V, a saída vai para 3V. Com 1V aplicado à entrada, o transistor T1 abre e a fonte de 3V é desligada da saída. Ao mesmo tempo, T6 está conduzindo e junto com T5 que também está conduzindo, ligam a fonte de 2V à saída. Quando à entrada, é aplicada uma tensão de 2V, o transistor T5 deixa de conduzir desconectando a saída da fonte de 2V. Por sua vez, T4 passa a conduzir e junto com T3, ligam a saída do circuito à fonte de 1V. Com 3V na entrada, o transistor T2 conecta a saída em 0V enquanto os transistores T1, T3 e T5 impedem a ligação da saída às demais fontes de tensão.

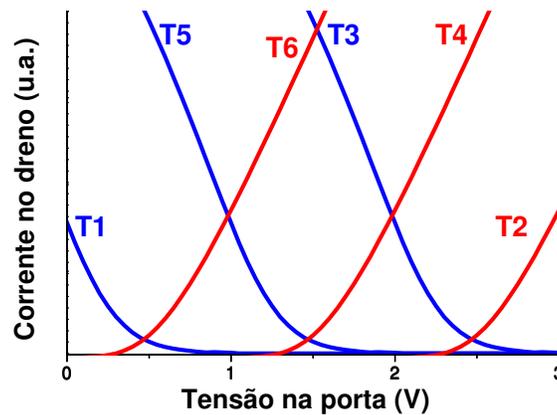


Figura 3.5: $I_d \times V_{gs}$ dos transistores usados no circuito inversor quaternários

Além dos 6 tipos de transistores usados no circuito inversor, outros dois tipos de transistores são também usados na realização dos circuitos desenvolvidos neste trabalho. Os valores de tensão de limiar para cada um dos transistores, relativos aos níveis lógicos, estão mostrados na tabela 3.1, assumindo os níveis 0, 1, 2 e 3.

Tabela 3.1: Tensões de limiar dos transistores usados

	<i>T1</i>	<i>T2</i>	<i>T3</i>	<i>T4</i>	<i>T5</i>	<i>T6</i>	<i>T7</i>	<i>T8</i>
Tensão de limiar	-2,5	2,5	1,5	0,5	-0,5	-1,5	-1,5	1,5
Tipo	PMOS	NMOS	PMOS	NMOS	PMOS	NMOS	PMOS	NMOS

As tensões de limiars mostradas na tabela 3.1 são relativas às fontes dos transistores, que, por sua vez, podem ser ligadas a qualquer uma das fontes de alimentação. Dessa forma, a tensão de limiar real¹ de cada transistor depende das suas conexões no circuito. No circuito inversor quaternário, por exemplo, as tensões de limiar dos transistores são deslocadas em relação aos valores da tabela 3.1 devido às conexões dos transistores nas diferentes tensões de alimentação, como se pode ver nas curvas da figura 3.5. As tensões de limiar para os transistores T1, T2, T3, T4, T5 e T6 no circuito inversor são 0,5V, 2,5V, 2,5V, 1,5V, 1,5V e 0,5V, respectivamente.

As tensões de limiar apresentados na tabela 3.1 correspondem à valores de tensão exatamente no meio entre dois níveis adjacentes, todavia, esses valores podem ser ajustados para se obter melhor desempenho. Se a tecnologia permite a redução do V_t dos transistores até 0,2V, as tensões de limiar dos transistores podem ser ajustadas para -2,2, 2,2, 1,8, 0,2, -0,2, -1,8, -1,2 e 1,2 para os transistores T1, T2, T3, T4, T5, T6, T7 e T8, respectivamente. As curvas de corrente no dreno em função da tensão na porta dos oito transistores usados estão mostradas na figura 3.6.

¹ Tensão de limiar real, neste trabalho, é definida como a tensão aplicada na porta do transistor conectado ao circuito em relação à qual o transistor está conduzindo ou não.

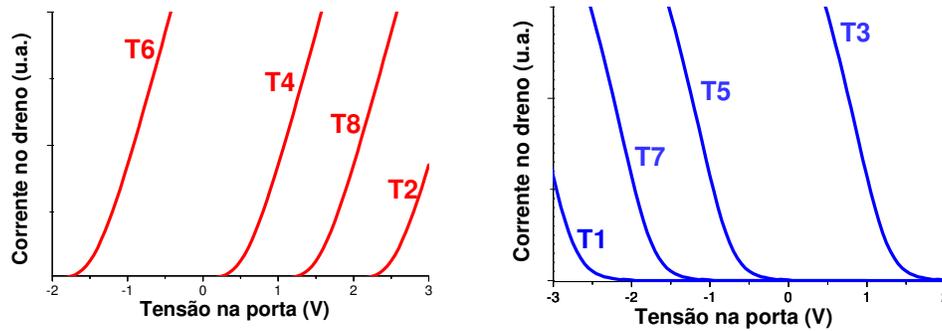


Figura 3.6: Curvas $I_{DS} \times V_{GS}$ esquemáticas dos 8 diferentes transistores usados

3.1.2 MIN e MAX

Em lógica quaternária, as funções conjunção e disjunção, que em binário representam as funções lógicas AND e OR, são substituídas pelas funções MIN e MAX. As funções MIN e MAX apresentam na saída o menor e o maior valor, respectivamente, entre as entradas. A implementação dessas funções proposta nesse trabalho é baseada no circuito inversor, que pode ser visto como uma função MIN e MAX de apenas uma entrada. Da mesma forma que nas funções binárias NAND e NOR, os circuitos naturais dessas funções em tecnologia CMOS são negadas e os circuitos apresentados realizam de fato as funções lógicas NMIN e NMAX.

O circuito que realiza a função lógica NMIN com duas entradas está mostrado na figura 3.7.a) juntamente com sua tabela verdade (figura 3.7.b)). Esse circuito foi desenvolvido a partir do circuito inversor quaternário, de forma similar à implementação de uma NAND binária que é baseada no inversor binário. O número de transistores usados é seis vezes o número de entradas do circuito.

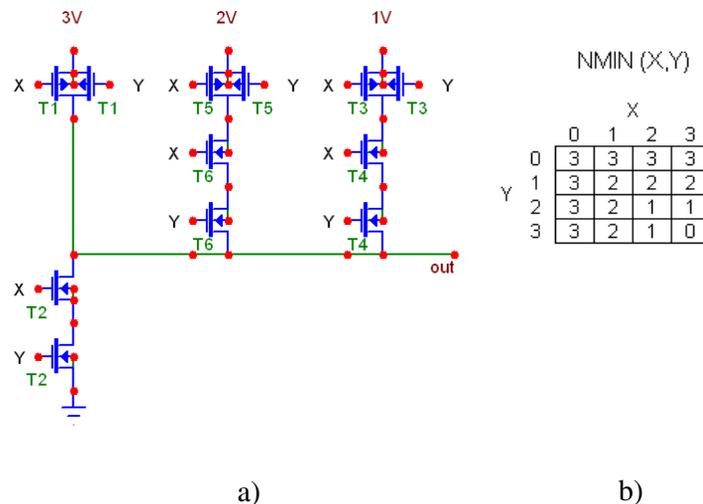


Figura 3.7: (a) Circuito NMIN e (b) sua tabela verdade

As duas entradas do circuito NMIN estão representadas na figura 3.7 como os sinais de porta X e Y. A operação do circuito se dá como segue: se pelo menos uma das duas entradas for mantida em 0V, pelo menos um dos transistores T1 está conduzindo e leva

a saída para 3V não importando o valor da outra entrada, da mesma forma como acontece no inversor quaternário. Ao mesmo tempo, pelo menos um dos transistores NMOS em cada ramo está aberto e desligam a saída das outras fontes de tensão. Os transistores NMOS dispostos em série entre cada um dos ramos 0V, 1V e 2V e a saída, fazem com que essas fontes estejam desconectadas da saída se pelo menos um dos dois transistores não está conduzindo, ou seja, sempre que um dos sinais é menor do que a tensão de limiar real desses transistores. De forma similar, os transistores PMOS, dispostos em paralelo em cada ramo, fazem com o sinal das fontes de tensão passem através deles se pelo menos uma das entradas for menor do que o valor de limiar real desses transistores. Esses transistores são responsáveis por desligar a fonte de tensão da saída sempre que as duas entradas são maiores do que as tensões de limiar desses transistores. Nessa configuração, temos na saída sempre o inverso (inversão diametral) do menor valor entre as duas entradas.

O circuito que realiza a função lógica NMAX com duas entradas está mostrado na figura 3.8.a) juntamente com sua tabela verdade (figura 3.8.b)). Esse circuito é feito a partir do circuito inversor quaternário, de forma similar à implementação de uma porta NOR binária a partir de um inversor binário. O número de transistores usados também é seis vezes o número de entradas do circuito.

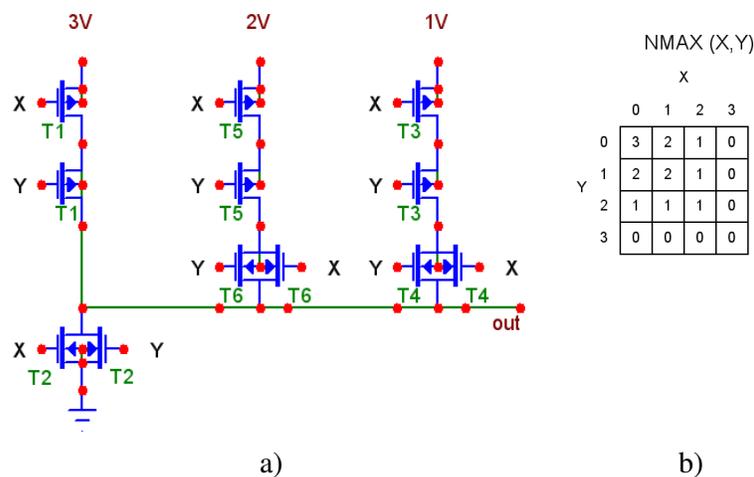


Figura 3.8: (a) Circuito NMAX de duas entradas e (b) a sua tabela verdade

A porta lógica MAX é um circuito de varias entradas onde a maior delas é copiada na saída. O circuito apresentado na figura 3.8 é a negação da função MAX e o sinal de saída é igual ao inverso do maior valor entre as entradas (inversão diametral). Os transistores nesse circuito são dispostos de modo que quando ambas as entradas estão em 0V, a saída é mantida em 3V pelo fechamento dos dois transistores T1. Quando em pelos menos uma das entradas é aplicado um sinal igual ou maior do que 1V, pelo menos um dos transistores T1 abre. Nessa situação o caminho entre a saída e a fonte de alimentação de 3V é aberto, ao mesmo tempo em que outra fonte de tensão é ligada à saída. Se a maior das entradas é 1V, um transistor T6 está conduzindo juntamente com os dois T5 e a saída é ligada em 2V. Se a maior das entradas é 2V, pelo menos um transistor T4 está conduzindo junto com os dois T3 e a saída é ligada à fonte de tensão de 1V. Com pelo menos uma das entradas em 3V, a saída é ligada em 0V, enquanto estão abertos os transistores T1, T3 e T5 controlados por essa mesma entrada, o que abre o caminho entre as outras fontes de tensão e à saída.

3.1.3 DLC

A função lógica DLC (Down Literal Circuit) é uma importante função em lógica não binária e é comumente usada como uma das funções básicas para a construção de circuitos complexos. Essa função possui uma entrada quaternária e uma saída binária que é levada ao nível lógico ALTO sempre que a entrada for menor do que um certo valor de referência i e ao nível lógico BAIXO, sempre que a entrada for maior ou igual a i . Em lógica quaternária existem três diferentes DLCs com i igual a 1, 2 e 3 (DLC1, DLC2 e DLC3 respectivamente). As tabelas verdades para os três tipos de DLC estão mostradas na tabela 3.2

Tabela 3.2: Tabelas verdades dos diferentes DLC quaternários

entrada	saídas		
	DLC1	DLC2	DLC3
0	3	3	3
1	0	3	3
2	0	0	3
3	0	0	0

A implementação de uma função DLC proposta neste trabalho utiliza apenas dois transistores numa configuração similar à de um inversor binário, como mostra a figura 3.9. O circuito DLC1 é feito usando um transistor PMOS T1 e um transistor NMOS T4. O sinal de entrada é aplicado às portas dos dois transistores e a saída é ligada ao dreno do NMOS e à fonte do PMOS. A fonte do NMOS é ligada em 0V enquanto o dreno do PMOS é ligado em 3V. Dessa forma, o transistor PMOS conduz quando a entrada está em 0V e o NMOS conduz quando a entrada está em 1V, 2V ou 3V. Os circuitos DLC2 e DLC3, são implementados substituindo T1 e T4 do DLC1 por T7 e T8 e T5 e T2, respectivamente.

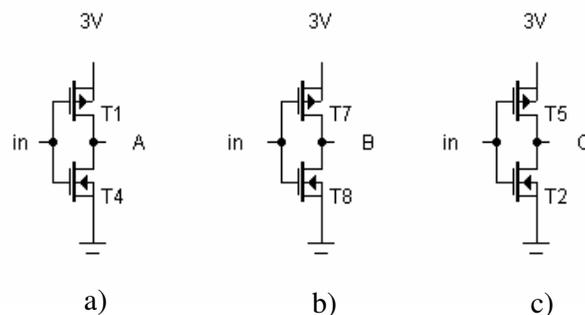


Figura 3.9: (a) DLC1 (b)DLC2 e (c) DLC3

3.1.4 Multiplexador

Além de portas lógicas básicas, é importante a demonstração de um circuito capaz de realizar qualquer função lógica, a fim de garantir a completude da família lógica proposta. Um circuito usado para realizar qualquer função lógica é o multiplexador

(MUX). Em lógica de níveis discretos, o multiplexador é um circuito que possui múltiplas entradas e só uma saída e a sua função é transmitir à saída o sinal de apenas uma das suas entradas. A seleção da entrada transmitida à saída é feita através de sinais de controle ou entradas de controle. Um multiplexador binário de duas entradas binárias, A e B, é uma conexão de transistores de passagem onde a saída é A ou B dependendo do valor de uma terceira entrada Z que é o sinal de controle. Esse circuito é chamado de MUX 2:1 e tem seu esquemático mostrado na figura 3.10.

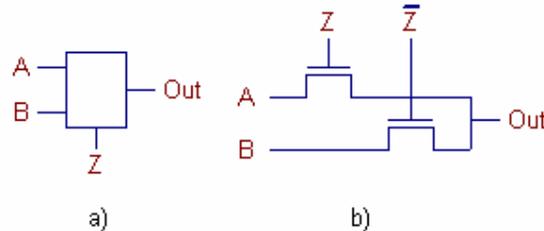


Figura 3.10: a) Representação de um multiplexador de 2 entradas (MUX 2:1) e b) seu esquemático

Um multiplexador binário de quatro entradas binárias, A, B, C e D, pode ser construído usando multiplexadores de duas entradas como blocos básicos, como mostra a figura 3.11.a). Esse multiplexador necessita de dois sinais de controle, Z1 e Z2, para selecionar apenas uma das entradas e transmiti-la à saída. O esquemático de um multiplexador de 4 entradas (MUX 4:1) está mostrado na figura 3.11.b). Note que foram suprimidos transistores de passagem redundantes nesse esquemático.

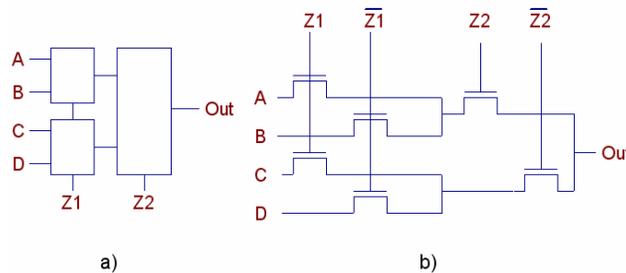


Figura 3.11: a) Diagrama de blocos e b) esquemático de um MUX 4:1

Em um multiplexador quaternário, todas as entradas, a saída e os sinais de controle são quaternários. Dessa forma, usando um único sinal de controle, se pode realizar um multiplexador quaternário de quatro entradas. Dessa forma, o mais simples multiplexador quaternário é capaz de realizar todas as 256 funções quaternárias possíveis de uma variável. A figura 3.12 mostra a representação em bloco de um multiplexador quaternário, onde A, B, C, D, Out e Z são sinais quaternários.

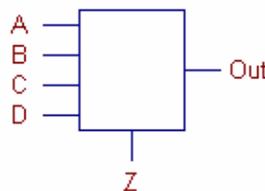


Figura 3.12: Representação de um QMUX 4:1

Em lógica binária, para implementar a mesma quantidade de funções lógicas seriam necessários 2 MUX de 8 entradas. Cada MUX de oito entradas pode realizar 256 funções, mas como a saída do MUX quaternário é quaternária, equivale a dois *bits* binários. Um MUX binário de 8 entradas necessita de 3 sinais de controle como mostra a figura 3.13.

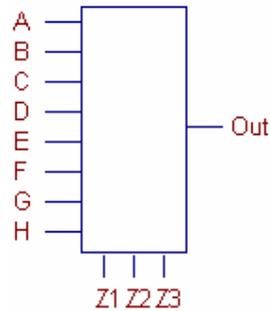


Figura 3.13: Multiplexador binário de 8 entradas MUX 8:1

Assim como em lógica binária, o multiplexador quaternário fundamental pode ser usado como bloco básico para a construção de multiplexadores de mais de uma entrada. A figura 3.14 mostra como pode ser implementado um QMUX 16:1 a partir de um QMUX 4:1. O multiplexador de 16 entradas quaternárias (A, B, C, ..., P) é capaz de realizar todas as 4^{16} possíveis funções de duas variáveis quaternárias usando apenas 2 sinais de controle.

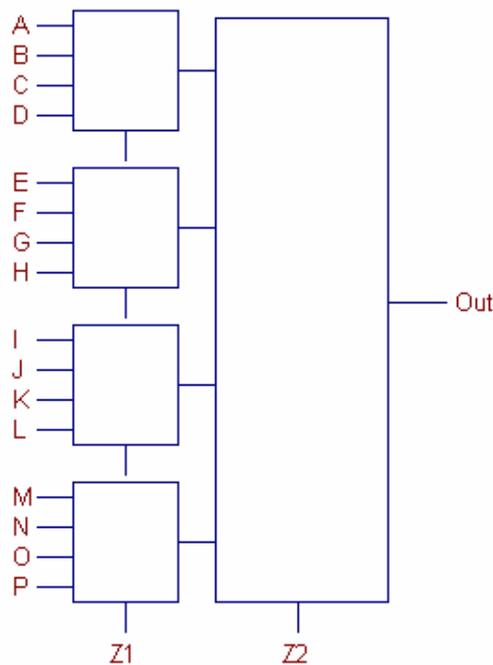


Figura 3.14: Diagrama de blocos de um multiplexador quaternário de 16 entradas (QMUX 16:1)

A implementação do QMUX também usa transistores de passagem para ligar qualquer uma das entradas à saída. A figura 3.15 mostra um esquemático da implementação de um QMUX 4:1 onde as entradas são A, B, C e D e podem representar qualquer valor entre 0, 1, 2 e 3. O sinal de controle é representado por IN e a saída do circuito é representada por OUT. Nesse circuito o sinal de controle (IN), passa por 3 DLCs. Ao sair dos DLCS, os três sinais passam por inversores binários. Dessa forma, para cada valor de IN, temos um diferente conjunto de 6 sinais elétricos (ver tabela 3.3) que controlam os transistores de passagem situados entre as entradas e a saída. Os sinais de saída dos DLCs e inversores são aplicados aos transistores de passagem de modo que quando o sinal de controle recebe os sinais de 0V, 1V, 2V e 3V, a saída copia os sinais de A, B, C e D, respectivamente.

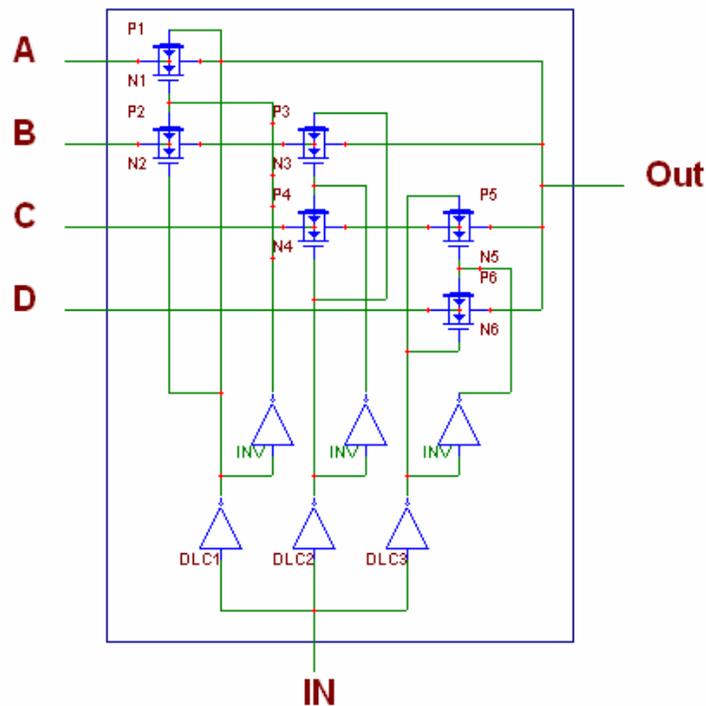


Figura 3.15: Esquemático do circuito multiplexador 4:1

Em um caso especial, esse circuito pode ser dedicado a realizar uma única função com as entradas sempre constantes e, nesse caso, apenas um transistor de passagem e não um par PMOS-NMOS pode ser usado, resultando em redução na área do circuito. Para realizar um inversor diametral, por exemplo, basta conectar as fontes de tensão de 0V, 1V, 2V e 3V nos terminais de entrada D, C, B e A, respectivamente.

Tabela 3.3: Saídas DLC e seus negados (NDLC) para o circuito QMUX 4:1

<i>In</i>	<i>DLC1</i>	<i>DLC2</i>	<i>DLC3</i>	<i>NDLC1</i>	<i>NDLC2</i>	<i>NDLC3</i>
0	3	3	3	0	0	0
1	0	3	3	3	0	0
2	0	0	3	3	3	0
3	0	0	0	3	3	3

3.2 Circuitos Aritméticos

3.2.1 Contador

O contador é um circuito que armazena ou mostra o número de vezes que um determinado evento ou processo ocorre. Existem diversos tipos de contadores entre síncronos, assíncronos, contadores Johnson ou por décadas. Como uma prova de conceitos de como a tecnologia apresentada aqui pode ser usada em circuitos síncronos, um contador controlado por relógio é apresentado. Antes disso, são mostrados circuitos osciladores em anel e registradores quaternários.

Um circuito oscilador em anel pode ser construído usando circuitos inversores em número ímpar ligando a saída de um inversor à entrada do seguinte inversor na cadeia, como mostra a figura 3.16.a). Esse oscilador em anel funciona da mesma forma que um oscilador em anel binário com a única diferença que o sinal pode oscilar entre os níveis lógicos 0 e 3 ou entre os níveis lógicos 1 e 2. Um oscilador desse tipo pode ser construído tanto com o circuito inversor quanto com um circuito QMUX programado para realizar a função inversão. Uma vez que o circuito QMUX pode ser usado para realizar qualquer função, é possível usá-lo para realizar oscilações entre quaisquer dois níveis lógicos, ou mesmo entre mais de 2 níveis lógicos. Com o QMUX realizando a função de sucessor, por exemplo, é possível fazer uma oscilação que passe pelos 4 níveis lógicos, usando para isso uma seqüência de $4n+1$ ($n \geq 1$) Sucessores (figura 3.16.b)). Também é possível uma oscilação no sentido contrário usando $4n-1$ Sucessores em série (figura 3.16.c)). Dessa forma a oscilação diminui ciclicamente. Essa característica, sem correspondência em lógica binária, potencializa o uso de lógica multivalorada para novas aplicações ou novas maneiras de resolver velhos problemas. A tabela 3.4 mostra como os sinais oscilam ao longo do tempo nos osciladores mostrados na figura 3.16.

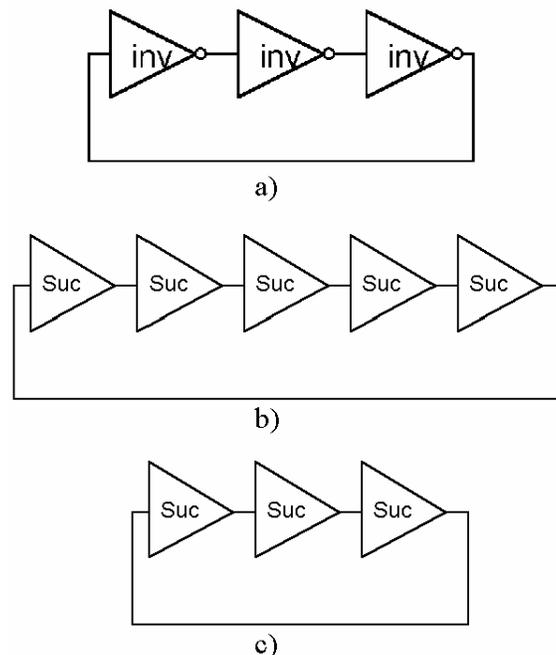


Figura 3.16: Osciladores em anel quaternários construídos usando a) inversores em número ímpar, b) 5 Sucessores ($4n+1$ com $n=1$) e c) 3 Sucessores

Tabela 3.4: Seqüência de oscilação dos osciladores da figura 3.16. $t_1 < t_2 < t_3 \dots$

Tipo de oscilador	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
a) Inversores	0	3	0	3	0	3	0	3	0
	1	2	1	2	1	2	1	2	1
b) 5 Sucessores	0	1	2	3	0	1	2	3	0
c) 3 Sucessores	0	3	2	1	0	3	2	1	0

De forma similar se pode construir um circuito para armazenar um dado. Usando um número par de inversores ligados em anel, um sinal pode ser armazenado e esse circuito pode ser chamado de célula de memória estática ou registrador. Pode-se construir uma célula de memória capaz de gravar qualquer um dos quatro níveis lógicos usando inversores (figura 3.17.a). Se pode ler dois sinais armazenados nesse anel que podem ser 0 e 3 ou 1 e 2 e existe também a possibilidade de armazenamento dos 4 níveis lógicos no mesmo anel usando 4 sucessores (ou antecessores) no lugar dos inversores (figura 3.17.b)). Dessa maneira podemos acessar não só o dado, mas também o seu sucessor, antecessor e inverso que também estarão armazenados no mesmo anel.

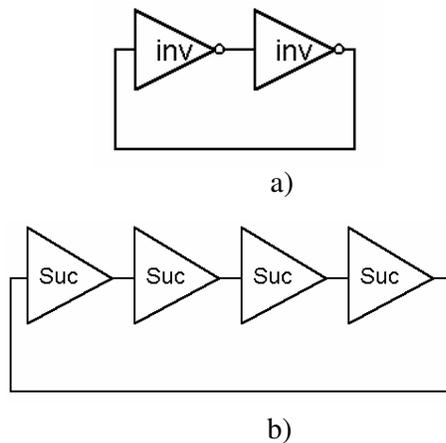


Figura 3.17: Circuitos registradores construídos com a) número par de inversores e b) com 4 circuitos sucessores

Aplicando os conceitos de oscilador e registrador, construímos um contador quaternário síncrono. A célula básica desse contador está mostrada na figura 3.18. O contador é constituído por parte de um oscilador em anel usando sucessores e parte de um registrador, controlado por dois relógios construídos com fases complementares. Dessa forma, a cada ciclo de relógio o sucessor levanta ciclicamente um nível lógico e o registrador armazena esse novo sinal. A fim de generalizar o contador para contar n bits, é usado um relógio baseado na saída do *bit* anterior para cada um dos *bits*. Isso é feito ao passar o sinal de saída do *bit* anterior por um circuito DLC1, que garante uma frequência de oscilação de $1/4$ da frequência do *bit* anterior. Dessa forma, sempre que um *bit* faz a transição entre os níveis lógicos 3 e 0, o *bit* seguinte aumenta um nível lógico ciclicamente e assim por diante.

3.2.2 Somadores Parciais (2:2)

Além das portas lógicas e circuitos básicos, é importante a demonstração de um circuito aritmético ou equivalente em complexidade. Em lógica quaternária, um somador pode ser realizado de inúmeras maneiras e nem sempre é fácil fazer uma correspondência direta com o circuito equivalente binário. Uma vez que os sinais podem representar 4 níveis, tanto as entradas quanto as saídas (*bit* de soma e *bit carry* de saída (*carry out*)) são quaternárias. Dessa forma, um somador completo, que em lógica binária é a soma de 2 entradas mais o *carry* de entrada (*carry in*), em lógica quaternária deve ser um circuito capaz de somar 4 entradas quaternárias mais o *carry in* quaternário (somador completo 5:2). Além do somador total, podemos ter somadores parciais de 2, 3 ou 4 entradas. Diferentes maneiras de realizar o somador quaternário foram desenvolvidas neste trabalho e estão apresentadas as mais relevantes.

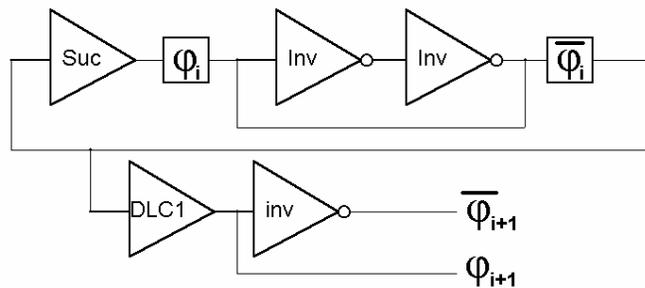


Figura 3.18: Célula básica de um contador quaternário. O relógio do *bit* $i+1$ é $1/4$ do *bit* i

O somador parcial de 2 entradas (sem *carry in*), é a soma de 2 sinais quaternários. Esse circuito tem duas saídas, uma representando a soma (*bit* 0) e outra representando o *carry out* (*bit* 1) como mostra a tabela 3.5. Existem 16 combinações possíveis que resultam em somas entre 00 e 12 (0 e 6 no sistema decimal) e o *carry out* é binário.

Tabela 3.5: Tabela verdade de um somador parcial de 2 entradas.

A	B	Soma	Carry out
0	0	0	0
0	1	1	0
0	2	2	0
0	3	3	0
1	0	1	0
1	1	2	0
1	2	3	0
1	3	0	1
2	0	2	0
2	1	3	0
2	2	0	1
2	3	1	1
3	0	3	0
3	1	0	1
3	2	1	1
3	3	2	1

3.2.2.1 Somador MIN-MAX

A implementação do somador parcial pode ser feita a partir da tabela 3.5 segundo a simetria representada pelas equações da tabela 3.6. Sempre que uma das entradas é igual a zero, a soma tem o valor da outra entrada, ou seja, quando o menor valor entre A e B é 0 ($\text{MIN}(A,B)=0$), a soma é igual ao maior valor entre A e B ($\text{MAX}(A,B)$). O *carry out*, nesse caso, é sempre 0 pois a soma das duas entradas não pode ser maior do que 3. Quando o menor valor entre as duas entradas é 1 ($\text{MIN}(A,B)=1$), a soma é igual ao sucessor do maior valor ($\text{Suc}[\text{MAX}(A,B)]$). O *carry out* nesse caso fica condicionado a ser 1 se $\text{MAX}(A,B)=3$ e 0 nos outros casos. Se o menor valor entre as duas entradas é igual a 2 ($\text{MIN}(A,B)=2$), então a soma é igual ao sucessor do sucessor do maior valor ($\text{Suc}^2[\text{MAX}(A,B)]$) e o *carry out* é sempre 1. Quando o menor valor entre as duas entradas é 3 ($\text{MIN}(A,B)=3$), a soma é igual ao terceiro sucessor do maior valor entre A e B ($\text{SUC}^3[\text{MAX}(A,B)]$) e o *carry out* é sempre 1. Como só temos um caso em que o mínimo entre as duas entradas é 3 ($A=3$ e $B=3$), a soma é sempre igual a 2.

Tabela 3.6: Simetrias da tabela verdade do somador de 2 entradas.

<i>SE</i>	<i>Soma</i>	<i>Carry out</i>
$\text{Min}(A,B) = 0$	$\text{Max}(A,B)$	0
$\text{Min}(A,B) = 1$	$\text{Suc} [\text{Max}(A,B)]$	0 1 (se $\text{Max}(A,B)=3$)
$\text{Min}(A,B) = 2$	$\text{Suc}^2 [\text{Max}(A,B)]$	1
$\text{Min}(A,B) = 3$	$\text{Suc}^3 [\text{Max}(A,B)]=2$	1

Um somador de duas entradas pode ser implementado por um circuito que resolva as equações da tabela 3.6. A figura 3.19 mostra um diagrama de blocos do circuito somador baseado nessa simetria. Nesse circuito, as duas entradas A e B entram ao mesmo tempo em um circuito $\text{MIN}(A,B)$ e $\text{MAX}(A,B)$. A saída do circuito $\text{MAX}(A,B)$ é levada a entrada de 3 circuitos sucessores colocados em série. A saída do circuito $\text{MIN}(A,B)$ é usada como sinal de um QMUX 4:1 que tem em suas entradas os sinais do $\text{MAX}(A,B)$ e seus sucessores. O *carry out* é avaliado usando um multiplexador controlado por $\text{MIN}(A,B)$ e $\text{MAX}(A,B)$.

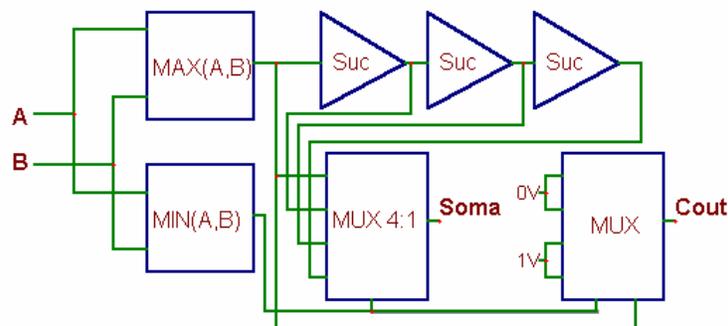


Figura 3.19: Diagrama de blocos de um somador quaternário implementado segundo as equações da tabela 3.6

3.2.2.2 Somador Direto

O somador parcial de dois números pode ser realizado de forma mais simples e direta usando apenas transistores de passagem e DLCs. Partindo da tabela verdade (tabela 3.3), cada linha pode ser implementada independentemente usando seis transistores de passagem entre a fonte de alimentação correspondente ao resultado da soma e a saída. Esses transistores de passagem são controlados por seis DLCs e seus negativos. Cada entrada, A e B, passa por 3 DLCs (DLC1, DLC2 e DLC3) e depois por inversores gerando sinais como os da tabela 3.3. Esse conjunto de sinais garante que, para cada par de entradas, somente um caminho entre fonte de alimentação e a saída esteja em curto. São 16 caminhos no total, correspondendo às 16 linhas da tabela verdade. Como exemplo, está mostrado na figura 3.20, o caminho da soma de $A=1$ e $B=3$ (soma igual a 0).

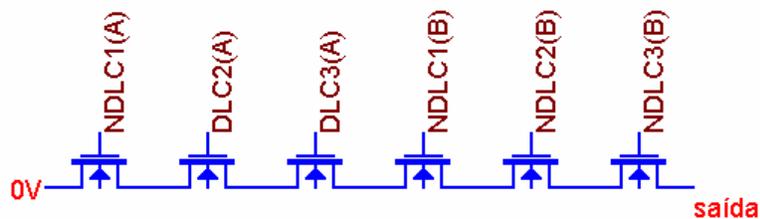


Figura 3.20: Um dos ramos da soma direta. Nesse exemplo, $A=1V$, $B=3V$ e $A+B=0V$

Na figura 3.20, todos os transistores de passagem são do tipo NMOS para transmitir o sinal de 0V sem degradação. Dos 6 transistores, 3 são controlados por A e 3 por B. As saídas dos circuitos DLCs e seus inversores são usados de modo a todos os transistores conduzirem quando $A=1$ ($NDLC1(A)=3$, $DLC2(A)=3$ e $DLC3(A)=3$) e $B=3$ ($NDLC1(B)=3$, $NDLC2(B)=3$ e $NDLC3(B)=3$). Uma vez que todos estão conduzindo ao mesmo tempo, o sinal de 0V é transmitido à saída. Em qualquer outro caso (A diferente de 1 ou B diferente de 3), pelo menos 1 dos transistores não está conduzindo e o caminho entre a fonte de alimentação e a saída está aberto. Os outros ramos são construídos da mesma forma. Para as somas iguais a 1V, 2V e 3V são usados transistores de passagem PMOS a fim de evitar degradação do sinal. Os primeiros transistores de cada ramo (junto à fonte de alimentação) são controlados pela saída do $DLC1(A)$ ou seu inverso, os segundos transistores de cada ramo são controlados pela saída do $DLC2(A)$ ou seu inverso e assim por diante, na mesma ordem em que aparecem na figura 3.20. A figura 3.21 mostra, esquematicamente, todos os 16 ramos da soma direta. Como notação, colocamos um "N" na porta do transistor sempre que o sinal de entrada for a saída do DLC negada.

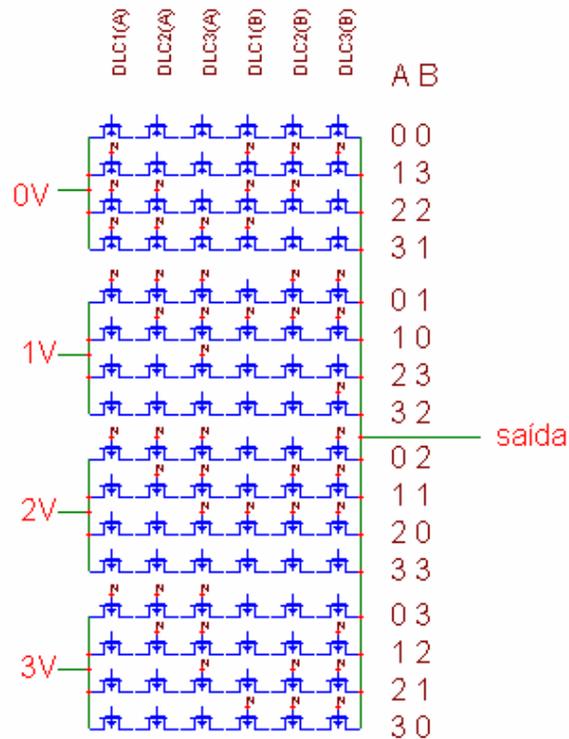


Figura 3.21: Somador direto

No somador direto, o *carry out* é avaliado de forma parecida usando como fontes de alimentação somente 0V e 1V. Tanto o circuito da soma quanto o circuito do *carry* podem ser consideravelmente reduzidos se removidos os transistores redundantes. Transistores redundantes podem ser considerados quando dois ramos que se ligam à mesma fonte de alimentação e são acionados por sinais idênticos. Nesse caso, se pode eliminar um dos dois transistores e deixar que os dois ramos compartilhem o mesmo transistor.

3.2.2.3 Somador QMUX

Uma vez que temos a disposição um circuito que pode realizar qualquer função lógica como o QMUX, podemos usá-lo para implementar um circuito somador. Entre as inúmeras possibilidades está o uso de vários QMUX para implementar as seguintes equações:

Se $B = 0$, então $S = A$;

Se $B = 1$, então $S = \text{Suc}(A)$;

Se $B = 2$, então $S = \text{Suc}^2(A)$;

Se $B = 3$, então $S = \text{Suc}^3(A)$.

A partir das equações acima, podemos construir um somador parcial (somente a parte da soma) como mostra a figura 3.22. Nesse somador, primeiro são calculados os Sucessores de A através de circuitos QMUX 4:1 (ver figura 3.12) configurando as entradas conforme a tabela 3.7.

Tabela 3.7: Fontes de tensão ligadas às entradas do QMUX para realizar as funções Sucessoras

	Suc (A)	Suc ² (A)	Suc ³ (A)
A	1V	2V	3V
B	2V	3V	0V
C	3V	0V	1V
D	0V	1V	2V

Após o cálculo dos sucessores de A, os três sinais sucessores junto com o próprio sinal de A são inseridos em outro QMUX 4:1 controlado por B. Dessa forma B determina qual dos sinais de entrada do QMUX 4:1 deve passar à saída, que é o valor da soma de A + B.

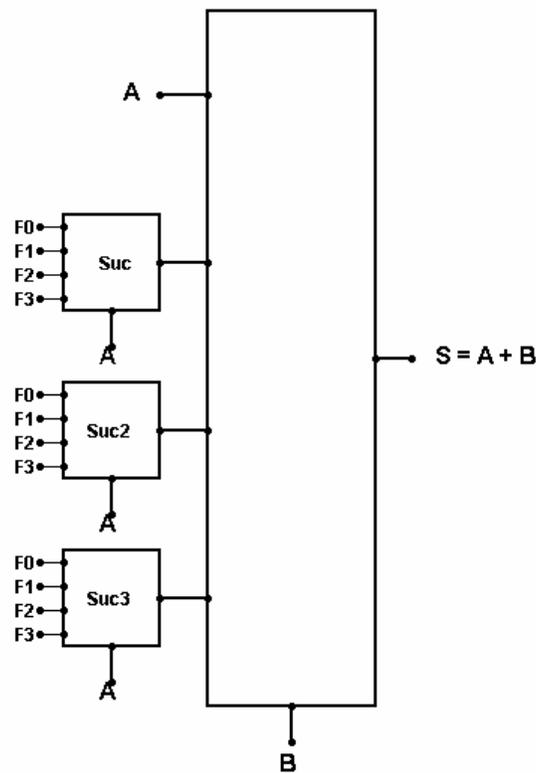


Figura 3.22: Somador parcial 2:2 implementado usando QMUX 4:1

A avaliação do *carry* de saída é feita de maneira parecida, mas usando outra simetria. Conforme a tabela 3.5, podemos calcular o *carry out* parcial (de 2 entradas) segundo as equações:

Se $B = 0$, então $Cout = 0$

Se $B = 1$, então $Cout = NDLC3(A)$

Se $B = 2$, então $Cout = NDLC2(A)$

Se $B = 3$, então $Cout = NDLC1(A)$

Dessa forma, no lugar dos três QMUX 4:1 controlados por A (necessários para avaliar a soma) podemos colocar 3 DLCs e 3 inversores, como mostra a figura 3.23.

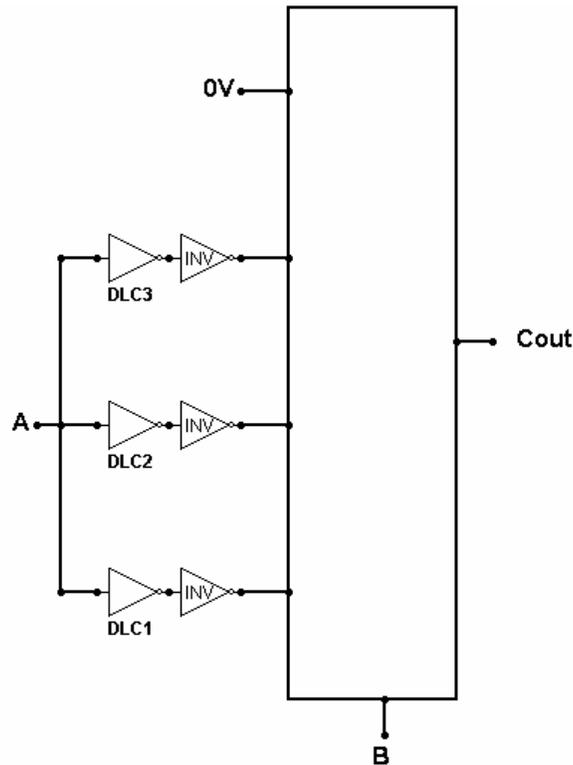


Figura 3.23: Circuito avaliador de *Carry out* parcial

3.2.3 Somadores Completos (5:2)

Em lógica quaternária, um somador completo é um circuito capaz de somar 5 números quaternários (4 entradas mais o *carry in*) resultando em dois *bits* de saída (*carry out* e soma) quaternários. Existem diferentes maneiras de se realizar o somador completo que pode ser baseado em somadores parciais ou diretamente usando diferentes QMUX. A análise dos diferentes somadores completos é feita primeiramente usando apenas quatro entradas, uma vez que o *carry* de entrada pode ser adicionado ao final, separadamente. Com o uso de somadores parciais 2:2 como bloco básico, constrói-se o somador parcial 4:2. Os somadores parciais 2:2 podem ser arranjados em cascata ou em árvore para somar quatro números quaternários A, B, C e D, como mostra a figura 3.24.

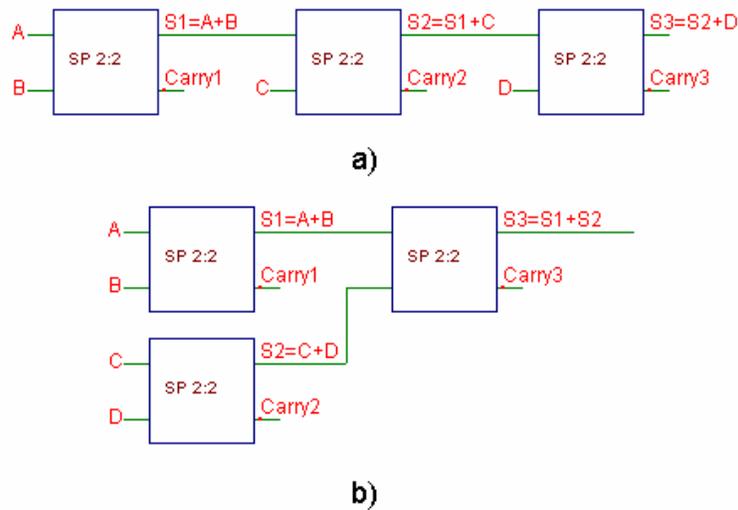


Figura 3.24: Somadores parciais 4:2 baseados em somadores parciais 2:2, dispostos em a) cascata e b) árvore

No arranjo em cascata dos somadores parciais 2:2, a soma de $A + B$ é o sinal $S1$ e o *carry* parcial é chamado de *carry1*. $S1$ é usado como entrada no segundo somador parcial 2:2 e somado com C , resultando no sinal de soma $S2$ e no *carry* parcial *carry2*. $S2$ é então usado como entrada em um terceiro somador parcial 2:2 e somado com D , resultando na soma $S3$ e no *carry* parcial $S3$. No arranjo em árvore, A e B são somados em um *carry* parcial 2:2, resultando na soma $S1$ e no *carry* parcial *carry1*. Ao mesmo tempo, C e D são somados, gerando $S2$ e *carry2*. Um terceiro somador parcial 2:2 soma $S1$ e $S2$, resultando nos sinais $S3$ e *carry3*.

Para construir um somador completo 5:2, basta usar o valor $S3$ calculado usando o somador parcial 4:2 e somá-lo com o quinto número, que pode ser o *carry in*. O resultado dessa última soma parcial 2:2 é o sinal de soma total dos 5 números e o *carry* parcial *carry4*, como mostra a figura 3.25.

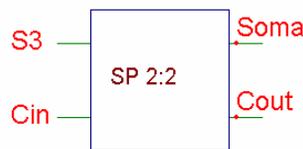


Figura 3.25: Somador parcial 2:2 final

A avaliação do *carry out* quaternário é a soma de todos os *carry* parciais, sendo incluído aqui um quarto *carry* (*carry4*) resultado da última soma. O diagrama de blocos do circuito que avalia o *carry out* a partir dos *carry* parciais está mostrado na figura 3.26.

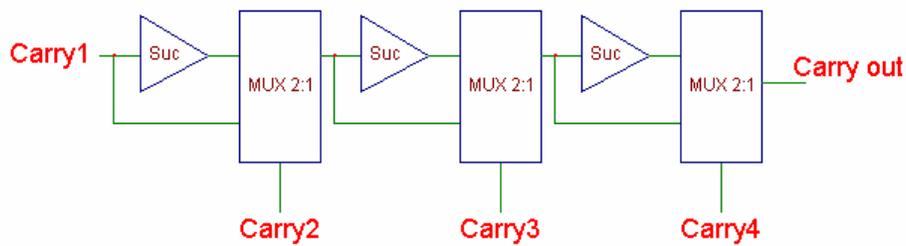


Figura 3.26: diagrama de blocos do circuito avaliador do *carry out* quaternário

O circuito de avaliação do *carry out* faz a soma de todos os *carry* parciais assumindo que cada *carry* parcial é binário e representa o número 1 quando está no nível lógico ALTO e 0 quando está no nível lógica BAIXO. Iniciando pelo *carry1*, é feita a soma de uma unidade para cada *carry* parcial no nível ALTO. A soma de uma unidade é feita usando um circuito sucessor e um MUX 2:1 que é controlado por um *carry* parcial e seleciona o sinal anterior, quando o *carry* parcial é 0, ou o sucessor do sinal anterior, quando o *carry* parcial é 1. O circuito usado para calcular o *carry out* quaternário está mostrado na figura 3.27.

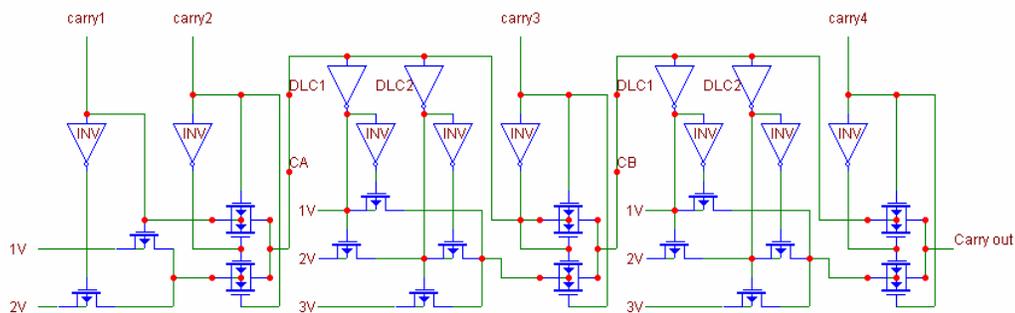


Figura 3.27: circuito de avaliação do *carry out* quaternário

3.3 Look-Up Tables

Um método de realizar funções lógicas genéricas a partir de suas tabelas verdades é conhecido como LUT (do inglês, Look-Up Table) (BROWN et al., 1992). Uma LUT é implementada através de um circuito multiplexador onde os sinais de controle desse multiplexador correspondem às variáveis de entrada da função descrita na tabela verdade e as entradas do multiplexador assumem valores constantes correspondentes aos valores de saída da função, como está mostrado no exemplo da figura 3.28. A figura 3.28.a) mostra a tabela verdade de uma função binária de duas variáveis ($Z = f(A, B)$) e a figura 3.28.b) mostra a implementação dessa função através de um multiplexador 4:1 binário.

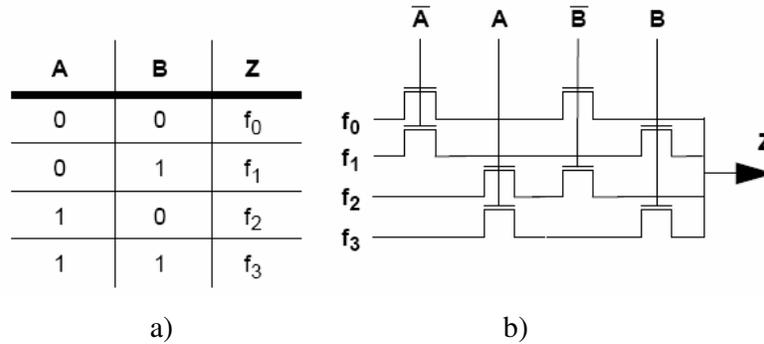


Figura 3.28: Look-up table: a) Tabela verdade de uma função binária de 2 variáveis e b) a sua implementação em um MUX

Look-Up Tables podem ser implementadas em qualquer operação, em qualquer base discreta e estão na base de diversos circuitos de lógica programável disponíveis comercialmente. Um grande número desses circuitos usam LUTs binárias de 4 e 6 entradas (4LUT e 6LUT). Uma 4LUT é implementada usando um multiplexador de 16 entradas, que usa quatro sinais de controle enquanto uma 6LUT é implementada através de um multiplexador de 64 entradas controladas por seis sinais.

Uma função quaternária de duas entradas possui 16 diferentes combinações entre elas e o multiplexador necessário para implementar essa função deve ter 16 entradas quaternárias e apenas dois sinais de controle quaternários. A figura 3.29.a) apresenta a tabela verdade para uma função de duas entradas quaternárias e uma saída quaternária enquanto a figura 3.29.b) mostra a primeira proposta de multiplexador usado para implementar a LUT quaternária de duas variáveis (2QLUT) (SHEIKHOLESLAMI et al., 1998). Em lógica quaternária existe uma dificuldade adicional relacionada às chaves do multiplexador, pois os sinais de controle não mais podem ser aplicados diretamente às portas dos transistores de passagem que são chaves unidirecionais e, portanto, estão abertos ou fechados em relação a um único ponto de referência. Dessa forma, o sinal de controle deve passar por algum circuito intermediário onde esse sinal é “demodulado” e aplicado aos transistores de passagem. Os circuitos ideais para esse fim são os circuitos delta literais (SHEIKHOLESLAMI et al., 1998), que na figura 3.29.b), estão representados por “X”. Uma função delta literal tem saída no nível lógico ALTO sempre que X for igual a n e no nível lógico BAIXO quando X for diferente de n.

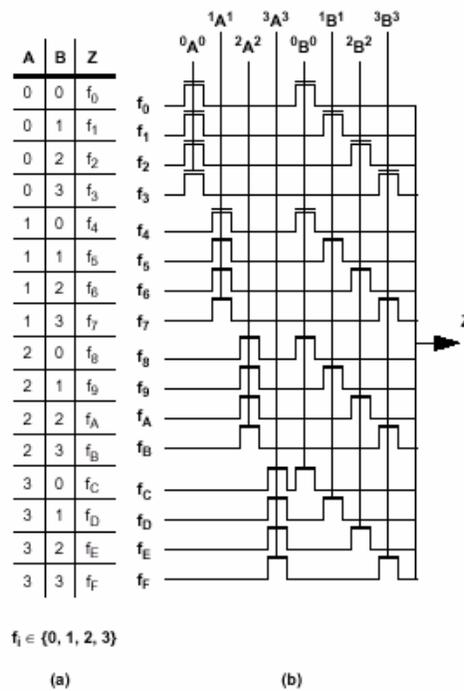


Figura 3.29: a) tabela verdade de uma função com duas entradas quaternárias e (b) a implementação dessa função usando transistores de passagem

Na realização do circuito multiplexador proposto neste trabalho, as funções delta literal foram substituídas por funções DLC. Apesar de se mostrarem teoricamente como os circuitos ideais para a função de demodulação, a implementação das funções delta literal não necessariamente é a mais eficiente e os melhores resultados foram obtidos usando os circuitos DLC.

A implementação de uma LUT quaternária para uma função de uma variável se faz pela utilização de um circuito QMUX (figura 3.15) onde todos os sinais de entrada são constantes. Para a realização de uma 2QLUT (LUT quaternária de 2 entradas), são usados três circuitos DLC e três inversores binários para cada sinal de entrada como mostra a figura 3.30.

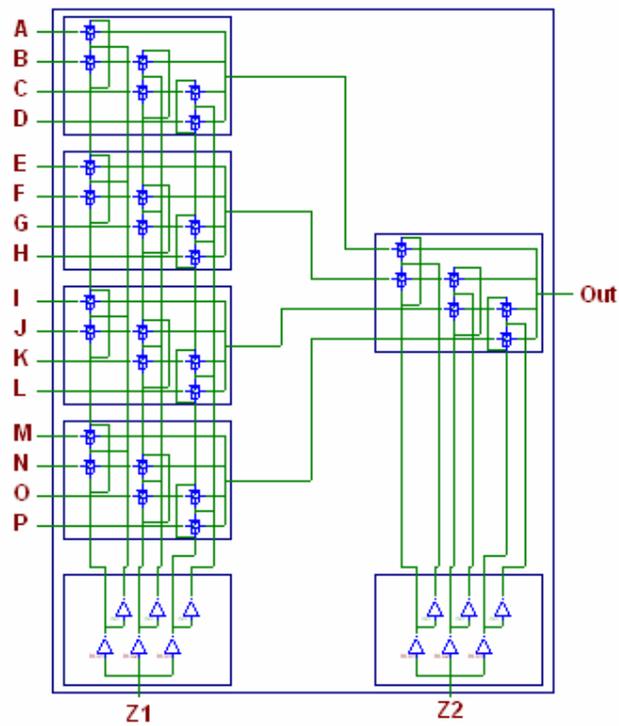


Figura 3.30: Esquemático de um circuito 2QLUT

Essa implementação faz uso do mesmo conjunto de transistores de passagem usados no QMUX, que é replicado cinco vezes de forma similar ao esquemático da figura 3.29. De forma similar se pode implementar uma nQLUT, com $n > 2$, a partir do QLUT e 2QLUT.

4 SIMULAÇÕES

4.1 Transistores

A verificação da funcionalidade e desempenho dos circuitos apresentados neste trabalho foi feita através de simulações elétricas. Foram usadas ferramentas SPICE para simular os circuitos usando a tecnologia TSMC 0,18 μm , cujos parâmetros estão mostrados no apêndice A.

Todos os circuitos foram projetados usando oito tipos de transistores, quatro PMOS e quatro NMOS, conforme descritos no capítulo 3. Cada transistor possui os mesmos parâmetros SPICE da tecnologia 0,18 μm , a menos da tensão de limiar. As curvas $I_{ds} \times V_{gs}$ dos oito transistores usados nos circuitos apresentados nesse trabalho estão mostrados na figura 4.1. Nessa simulação, todos os transistores estão ligados entre VDD de 3 volts e o terra.

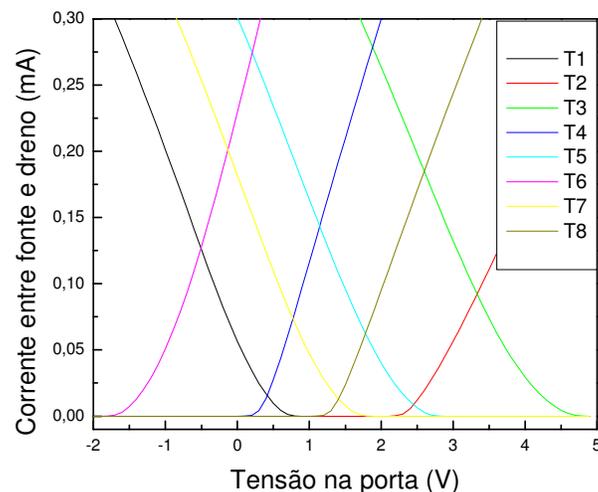


Figura 4.1: Curva $I_{ds} \times V_{gs}$ dos 8 transistores usados

As curvas da figura 4.1, mostram que as tensões de limiar de cada transistor, usando a alimentação de 3V, estão entre os valores de tensão dos níveis lógicos, quando representados por 0, 1V, 2V e 3V. Os transistores T6 e T3 estão conduzindo para

qualquer tensão entre 0 e 3V. Os transistores NMOS T4, T8 e T2 estão fechados para níveis lógicos maiores do que 1, 2 e 3, respectivamente, enquanto os transistores PMOS T1, T7 e T5 estão abertos para níveis lógicos menores do que 1, 2 e 3, respectivamente.

Com as conexões usadas para realizar o circuito inversor (figura 3.4), as curvas I_{ds} x V_{gs} dos seis transistores usados ficam como mostra a figura 4.2. Essas curvas foram esquematicamente mostradas anteriormente na figura 3.5.

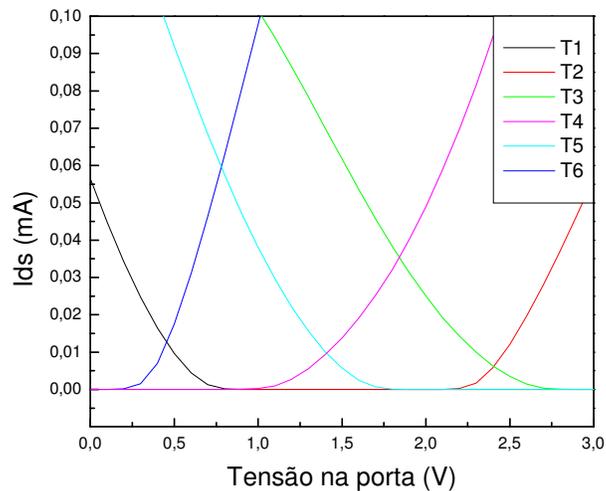


Figura 4.2: Curvas I_{ds} x V_{gs} para os transistores do circuito inversor

4.2 Circuitos Elementares

4.2.1 Inversor

A simulação do inversor envolve uma varredura na tensão de entrada do circuito e medida da sua tensão de saída. Em todas as simulações do circuito inversor foram usados como capacitâncias de saída, as portas de dois transistores (1 PMOS e 1 NMOS) nas dimensões mínimas ($L=0,18\mu\text{m}$ e $W=0,27\mu\text{m}$). A simulação DC mostra a funcionalidade do circuito inversor (figura 4.3). Nessa análise, a tensão de entrada é variada entre 0V e 3V em intervalos finitos e cada tensão aplicada é mantida constante até a estabilização do sinal de saída do circuito. A tabela verdade do circuito pode ser extraída por essa simulação. Conforme a figura 4.3, com a aplicação de 0V na entrada do inversor, a tensão na saída é exatamente 3V. Com tensões aplicadas de 1V, 2V e 3V, a tensão na saída é exatamente 2V, 1V e 0V, respectivamente.

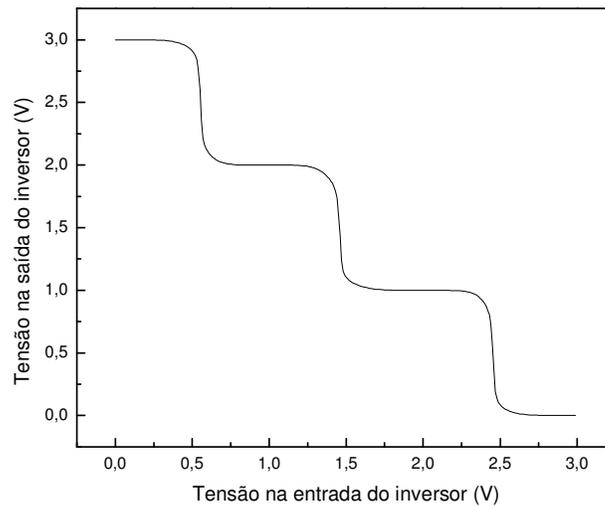


Figura 4.3: Análise DC do inversor quaternário

A verificação de desempenho é realizada através da análise de transiente dos sinais. Uma simulação de transientes permite fazer uma transição nas entradas do circuito entre quaisquer dois níveis lógicos em forma de rampa e são monitoradas as transições nos sinais de saída decorrentes da variação nas entradas. O teste de desempenho do circuito inversor, envolve a simulação de todas as possíveis transições de níveis lógicos na entrada, que são 12 no total, nomeadas 0-1, 0-2, 0-3, 1-0, 1-2, 1-3, 2-0, 2-1, 2-3, 3-0, 3-1 e 3-2 onde 0-1 significa variação da tensão de entrada de 0V para 1V, 0-2 de 0V para 2V e assim por diante. Em todas as transições são usadas rampas de subida ou descida com 10ps de duração. A figura 4.4 mostra os transientes do sinal de saída para todas as possíveis transições na entrada para um inversor onde todos os transistores do circuito são feitos com as dimensões mínimas, ou seja, com o comprimento de canal de $0,18\mu\text{m}$ e largura de canal igual a $0,27\mu\text{m}$.

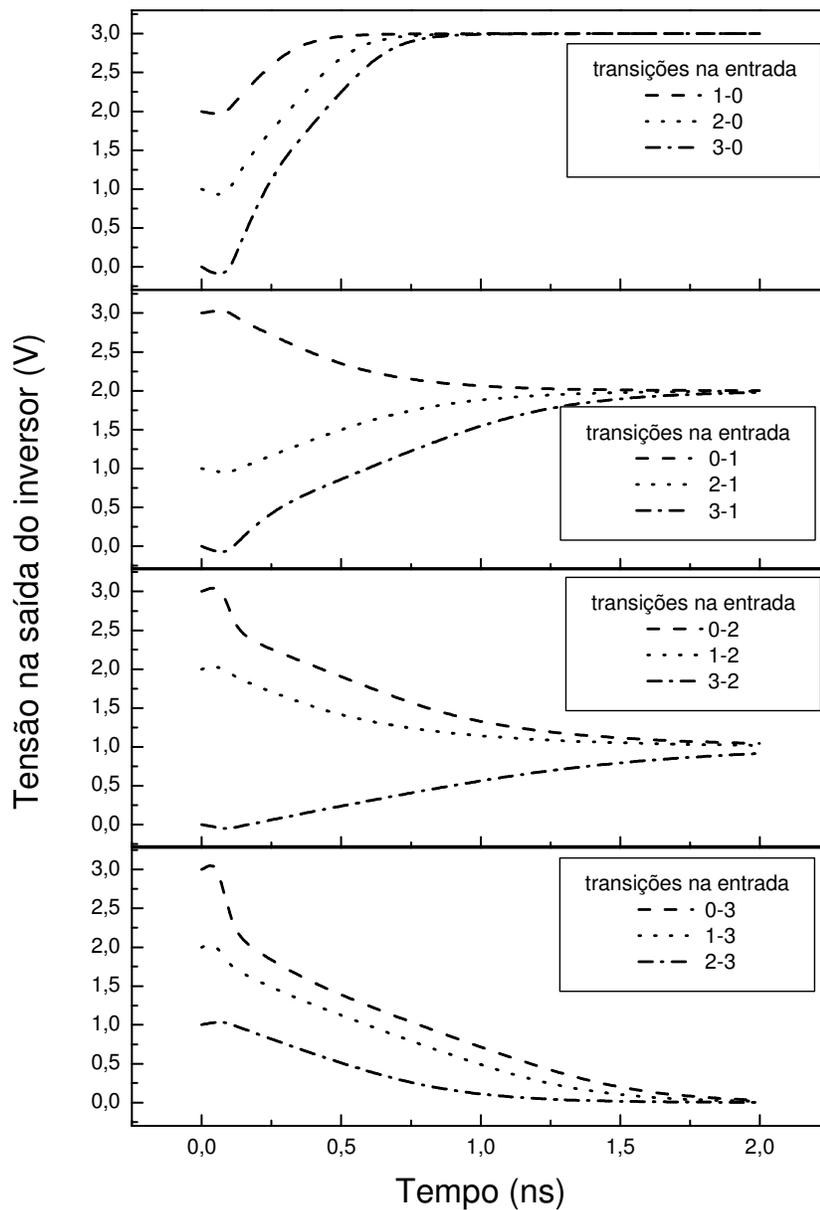


Figura 4.4: Transientes do sinal de saída para um inversor com os transistores nas dimensões mínimas

Os sinais de saída apresentaram um atraso de propagação menor do que 2ns a partir da transição na entrada do circuito. Os tempos medidos para o atraso na saída são contados a partir da transição na entrada até o tempo necessário para que o sinal de saída estabeleça o valor de tensão correto, com uma margem de erro de 0,5V. Os transientes da figura 4.4 apresentam um valor médio para o atraso de propagação de aproximadamente 650ps. O atraso máximo foi de 1180ps e o menor de 220ps, uma significativa diferença que pode ser corrigida pelo redimensionamento dos transistores.

O consumo de potência, que é a soma dos consumos das três fontes de alimentação, é medido à 250MHz e apresenta valor médio de $7,5\mu\text{W}$.

O resultado da simulação para esse circuito com o redimensionamento dos transistores pode ser visto na figura 4.5. O redimensionamento dos transistores leva em conta a diferença de mobilidade dos transistores, que é menor para os transistores do tipo PMOS, e a assimetria do circuito em relação à carga que deve ser carregada por cada fonte. Os transientes de saída para o circuito inversor com larguras de canal de $0,54\mu\text{m}$, $0,54\mu\text{m}$, $0,54\mu\text{m}$, $0,27\mu\text{m}$, $0,54\mu\text{m}$ e $0,27\mu\text{m}$ para os transistores T1, T2, T3, T4, T5 e T6, respectivamente, estão mostrados na figura 4.5

Os transientes mostrados na figura 4.5 apresentam uma média de 471ps no atraso de propagação. O consumo de potência médio para esse inversor é de $7,6\mu\text{W}$ à 250MHz. O atraso crítico nesse caso é de 773ps e o menor de 263ps, o que reduz de 525ps para 302ps a diferença entre a média dos atrasos e o atraso crítico. O circuito ainda pode ser melhorado pelo dimensionamento ótimo de todos os transistores, entretanto, esse procedimento requer uma otimização para cada circuito o que dificulta a automação do design.

Ensaio testando diferentes formas de ligação do contato do substrato foram realizados. Essa característica pode ser aproveitada em tecnologias SOI onde é possível contatar os substratos dos transistores individualmente e obter com isso superior desempenho. As simulações feitas até aqui usaram transistores com os terminais de substrato ligados como na figura 4.6.a). No circuito inversor, em cada ramo, o substrato do transistor que está ligado à fonte de alimentação é ligado à fonte de alimentação e o transistor ligado à saída tem o seu substrato ligado ao nó intermediário do ramo. Nessa condição, o valor médio do atraso de propagação é de 470ps (figura 4.5). Na configuração mostrada na figura 4.6.b), onde o substrato do transistor ligado à fonte de alimentação é ligado ao nó intermediário e o do transistor ligado à saída é ligado na saída, o atraso médio simulado foi de 350ps, o que representa uma melhoria de desempenho do circuito. Usando a configuração mostrada em 4.6.c), onde o substrato de todos os transistores PMOS estão ligados em 3V e o de todos os transistores NMOS estão ligados no terra, o atraso médio simulado foi de 840ps, o que mostra que a conexão usual em lógica binária não é a ideal em lógica quaternária.

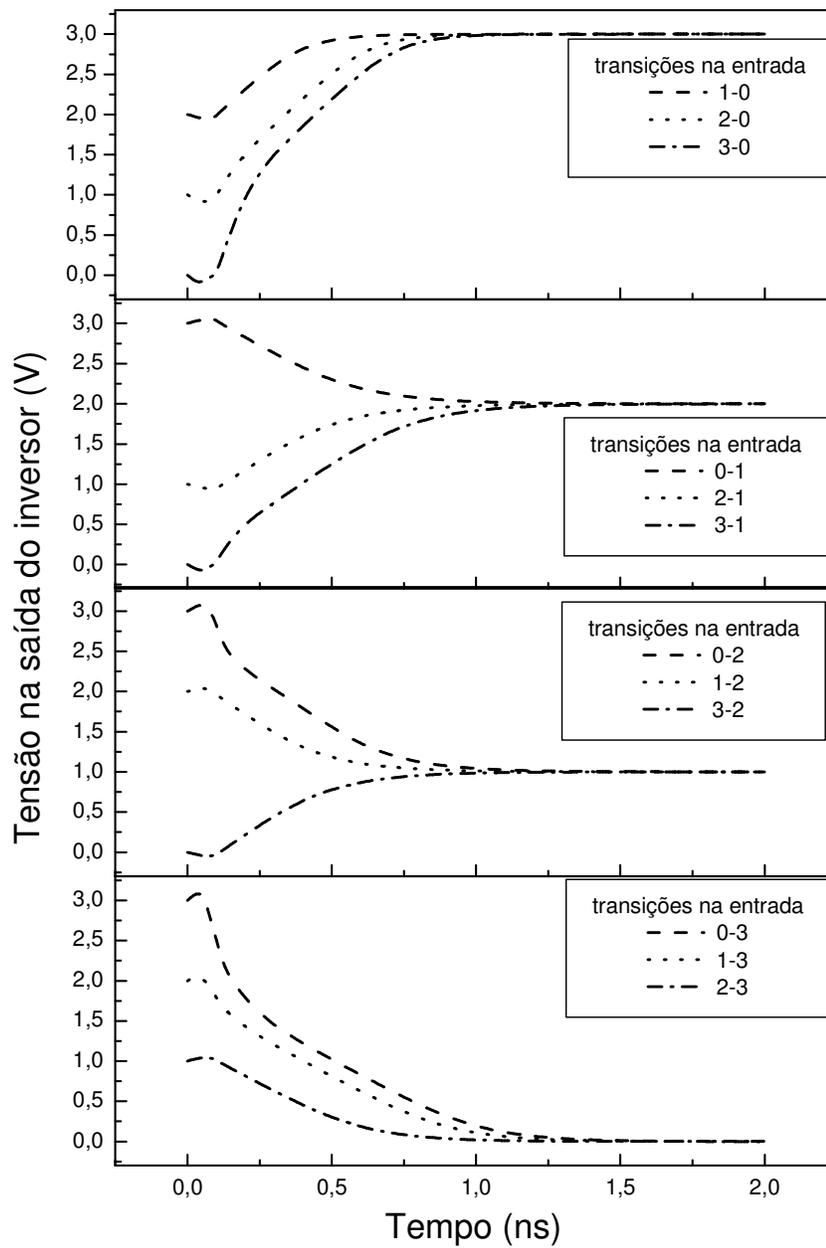


Figura 4.5: Transientes do sinal de saída para um inversor com larguras de canal de $0,54\mu\text{m}$, $0,54\mu\text{m}$, $0,54\mu\text{m}$, $0,27\mu\text{m}$, $0,54\mu\text{m}$ e $0,27\mu\text{m}$ para os transistores T1, T2, T3, T4, T5 e T6, respectivamente

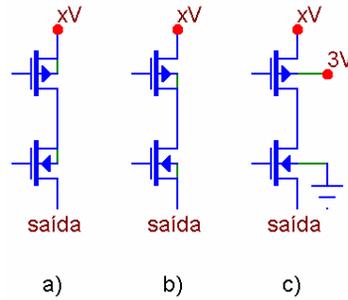


Figura 4.6: Diferentes maneiras de contatar o substrato dos transistores em lógica quaternária

A partir do circuito inversor, o repetidor e o oscilador em anel também foram simulados. O repetidor pode ser construído com dois inversores dispostos em série. O sinal de entrada é invertido duas vezes e, portanto, repetido na saída. O circuito repetidor simulado usa as mesmas dimensões de transistores que o circuito inversor cujas curvas são mostradas na figura 4.5. Uma varredura por todas as possíveis transições na entrada pode ser feita usando a seqüência 0-1-0-2-0-3-1-3-2-1-2-3-0. A figura 4.7 mostra os transientes de saída do repetidor quando aplicada a seqüência acima com intervalo de 4ns entre cada transição. A simulação mostra um atraso de propagação médio de 1,055ns e apresenta um consumo médio de potência de 19,5 μ W à 250MHz para o circuito repetidor.

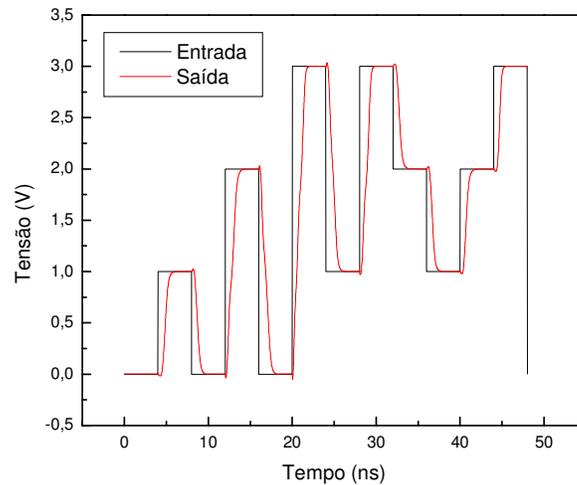


Figura 4.7: Transientes de saída do circuito repetidor

O oscilador em anel é realizado usando três inversores em série ligados em forma de anel onde a saída do último inversor é ligada na entrada do primeiro. Todos os nós de saída dos inversores, quando esse circuito é ligado às fontes de alimentação, oscilam entre os níveis lógicos 1 e 2 como mostra a figura 4.8.

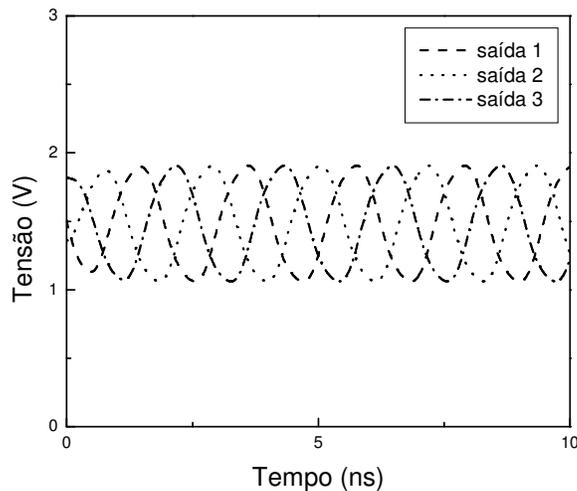


Figura 4.8: Tensões de saída de cada inversor em um oscilador em anel com 3 inversores quaternários

O circuito oscilador em anel com três inversores apresenta frequência de oscilação de 468 MHz com consumo médio de potência de $45,4\mu\text{W}$. Um oscilador em anel com cinco apresenta frequência de 260MHz e consumo de $47\mu\text{W}$.

4.2.2 NMIN/NMAX

Os circuitos NMIN e NMAX são simulados de forma semelhante às simulações anteriores, porém esses circuitos realizam funções de duas variáveis com 16 combinações possíveis entre elas. A simulação funcional dos circuitos NMIN e NMAX está mostrada na figura 4.9. Nessa simulação, uma varredura entre todas as 16 combinações das duas entradas é feita e o sinal de saída é medido.

Os atrasos de propagação e consumo de potência desses circuitos apresentam 701ps e 626ps para NMIN e NMAX, respectivamente. O consumo medido à 250MHz é de $15,5\mu\text{W}$ para NMIN e $11,5\mu\text{W}$ para NMAX.

4.2.3 DLC

Os circuitos DLCs, que serão usados na implementação do multiplexador também são simulados. As análises DC dos DLCs estão mostradas na figura 4.10. Nessa simulação, a tensão na entrada é varrida de 0 a 3V e o sinal de saída é medido e a função de transferência desses circuitos pode ser verificada.

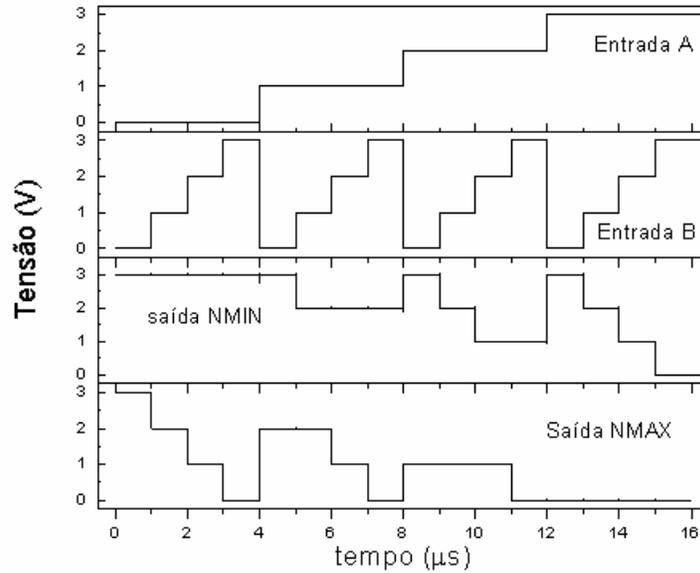


Figura 4.9: Entradas e saídas da simulação dos circuitos NMIN e NMAX

A análise dos transientes dos três circuitos DLC apresenta atrasos de propagação de 274ps, 260ps e 301ps e consumo médio de potência de $14,7\mu\text{W}$, $19,9\mu\text{W}$ e $15,1\mu\text{W}$ para os circuitos DLC1, DLC2 e DLC3, respectivamente.

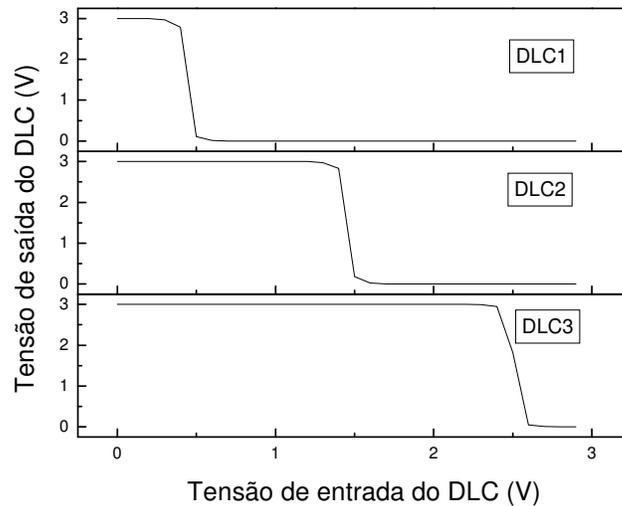


Figura 4.10: Funções de transferência dos circuitos DLCs

4.2.4 MUX

O circuito multiplexador pode realizar qualquer função lógica de uma entrada e para verificar seu desempenho qualquer função pode ser escolhida. Ao aplicar adequadamente os sinais de entrada do MUX, é simulada a função inversão diametral para comparar o circuito multiplexador com o circuito inversor apresentado anteriormente. O atraso médio do circuito é de 487ps consumindo em média $79,9\mu\text{W}$ à 250MHz.

Da mesma forma que com o circuito inversor, o multiplexador implementando a função de inversão também é usado para construir os circuitos repetidor e oscilador. O circuito repetidor que usa dois inversores apresenta atraso médio de 1064ps para um consumo de 160,5 μ W à 250MHz. O multiplexador quando usado com sinais constantes, pode ser reduzido pela eliminação dos pares de transistores de passagem sem degradação do sinal, usando apenas 18 transistores. Nesse caso, os atrasos têm média de 512ps e consumo médio de potência de 74,5 μ W à 250MHz. O repetidor que usa dois inversores com MUX reduzidos apresenta atraso médio de 1120ps, com atraso crítico de 1326ps. O consumo desse repetidor é de e 153,6 μ W à 250MHz. Outra forma de realizar o repetidor com MUX é selecionar diretamente as entradas para repetir o sinal de controle na saída, e o desempenho é aproximadamente o mesmo do inversor MUX.

O circuito oscilador em anel, construído usando três MUX inversores apresenta frequência de oscilação de 320MHz e consumo de 367 μ W. O oscilador também é simulado com MUX reduzido e apresenta frequência de 259MHz e potência de 273 μ W. Usando cinco circuitos Sucessores, a frequência de oscilação é de 140MHz.

4.3 Circuitos Aritméticos

4.3.1 Somador MIN-MAX

O somador MIN-MAX é simulado mantendo uma entrada constante e variando a outra. Os resultados das simulações dos transientes estão mostrados na tabela 4.1.

Tabela 4.1: Atrasos de propagação do somador MIN-MAX

Transição na entrada A	Atrasos de propagação (ps)			
	entrada B=0	entrada B=1	entrada B=2	entrada B=3
0-1	1001	1537	1362	1665
0-2	1671	1985	2043	1682
0-3	2218	2707	1781	2119
1-0	1228	1438	1682	2864
1-2	954	1502	1676	1380
1-3	1839	2189	1414	1979
2-0	1991	2130	1752	3248
2-1	1059	2008	1874	1315
2-3	1007	1880	1572	1560
3-0	2439	1083	2154	3452
3-1	2008	1414	2264	1740
3-2	1484	2293	1665	1426

O atraso crítico para esse somador é de 3452ps com média de 1807ps e consumo médio de 230 μ W à 250MHz. Esse somador usa 104 transistores.

4.3.2 Somador Direto

Os atrasos de propagação para todos os transientes possíveis obtidos na simulação do somador direto estão mostrados na tabela 4.2.

Tabela 4.2: Atrasos de propagação do somador direto

Transição na entrada A	Atrasos de propagação (ps)			
	entrada B=0	entrada B=1	entrada B=2	entrada B=3
0-1	400	2110	1841	1131
0-2	2786	2948	1375	1841
0-3	1888	466	804	175
1-0	1410	1772	793	3228
1-2	2459	2471	1503	816
1-3	2413	1014	1935	3590
2-0	1142	1830	3858	2436
2-1	1503	1970	2436	851
2-2	2343	1527	525	2517
3-0	1830	641	2284	2308
3-1	2389	4347	2855	1177
3-2	396	2844	1154	1340

O atraso crítico simulado é de 4347ps e a média é de 1827ps. Esse circuito consome em média 478 μ W à 250MHz e utiliza 68 transistores.

4.3.3 Somador Parcial 2:2 MUX

O somador parcial 2:2 MUX foi simulado em todas as transições possíveis, variando as duas entradas. Os resultados dos atrasos medidos estão mostrados nas tabelas 4.3 e 4.4, onde a tabela 4.3 mostra os atrasos de propagação do somador parcial 2:2 para as transições da entrada que controla os multiplexadores que realizam as funções sucessoras (entrada A na figura 3.22) enquanto a tabela 4.4 mostra os resultados para as transições na entrada que controla o MUX que avalia a soma das duas entradas (entrada B na figura 3.22).

Tabela 4.3: Atrasos do somador parcial 2:2 para transições na entrada A

Transição na entrada A	Atrasos de propagação (ps)			
	Entrada B=0	entrada B=1	entrada B=2	entrada B=3
0-1	46	658	601	878
0-2	110	775	971	508
0-3	164	970	513	233
1-0	55	622	662	785
1-2	45	659	981	657
1-3	120	960	596	648
2-0	129	778	566	792
2-1	41	745	1034	538
2-3	54	971	697	569
3-0	179	231	508	815
3-1	105	620	1027	557
3-2	41	1043	550	461

Tabela 4.4: Atrasos do somador parcial 2:2 para transições na entrada B

Transição na entrada B	Atrasos de propagação (ps)			
	entrada A=0	entrada A=1	entrada A=2	entrada A=3
0-1	480	454	503	741
0-2	524	531	755	392
0-3	496	711	326	163
1-0	285	399	457	564
1-2	445	468	752	496
1-3	457	664	350	489
2-0	345	475	340	578
2-1	399	466	846	429
2-3	399	652	494	431
3-0	373	82	315	590
3-1	443	270	834	443
3-2	396	662	408	338

A assimetria entre as duas entradas pode ser vista nos tempos dos atrasos de propagação das tabelas 4.3 e 4.4. O atraso crítico quando variada a entrada A é de 1043ps, maior do que os 846ps verificados quando variamos a entrada B. A média dos atrasos para a transição na entrada B, de 477ps também é inferior à média dos atrasos para as transições na entrada A, que é de 547ps. O consumo de potência médio para o somador parcial 2:2 MUX é de 80,5 μ W à 250MHz e o circuito utiliza 72 transistores.

4.3.4 Carry de Saída

A avaliação do *bit* de *carry* de saída pode ser feito independentemente e em paralelo ao cálculo da soma. O *carry* parcial, simulado variando uma das entradas mantendo a outra constante, apresenta média de atrasos de 163ps. Os atrasos obtidos estão mostrados na tabela 4.5. O caminho crítico do *carry* total é de 380ps.

Tabela 4.5: Atrasos de propagação do *carry* parcial (ns)

A	B=0	B=1	B=2	B=3
0-1	0	0,19	0	0,12
0-2	0	0	0,17	0,12
0-3	0	0	0,17	0,13
1-0	0	0	0	0,18
1-2	0	0	0,15	0
1-3	0	0,19	0,15	0
2-0	0	0	0,19	0,22
2-1	0	0	0,25	0
2-3	0	0,19	0	0
3-0	0	0,12	0,18	0,2
3-1	0	0,15	0,29	0
3-2	0	0,28	0	0

B	A=0	A=1	A=2	A=3
0-1	0	0	0	0,17
0-2	0	0	0,16	0,16
0-3	0	0,16	0,09	0,07
1-0	0	0	0	0
1-2	0	0	0,16	0,09
1-3	0	0,12	0,08	0
2-0	0	0	0,1	0,18
2-1	0	0	0,18	0
2-3	0	0,11	0	0
3-0	0	0,08	0,1	0,18
3-1	0	0,13	0,22	0
3-2	0	0,32	0	0

4.3.5 Somador Completo 5:2

O somador completo 5:2 é construído usando um somador parcial 4:2 e um somador parcial 2:2. O somador parcial 4:2, que é um arranjo de três somadores parciais 2:2, pode ser realizado arranjando-os em cascata ou em árvore. As simulações mostram que o arranjo em árvore resulta em atrasos menores do que o arranjo em cascata e esse é o arranjo usado no somador completo. A simulação para esse circuito mostra o caminho crítico com atraso de 1.26ns. O consumo médio do circuito em 200MHz é de 415 μ W e são usados 297 transistores nesse circuito.

Foram simulados a soma de 4 números com mais de 1 *bit* cada. A soma com números de 2 *bits* apresenta atraso crítico de 1,45ns e consumo de 1,11mW em 200MHz. As simulações com números de 3 e 4 *bits* apresenta atrasos de 1,66ns e 2,27ns e consumo de 1,95mW e 2,68mW em 200MHz, respectivamente. O número de transistores usados é de 701, 1105 e 1509 para dois, três e quatro *bits*, respectivamente.

4.3.6 Equivalentes Binários

Tendo em vista a realização do somador completo, podemos compará-lo com um circuito equivalente em lógica binária. Um número quaternário de 1 *bit* equivale a 1 número binário de 2 *bits*, assim, um somador quaternário de 4 números de n *bits* equivale a um somador binário de 4 números de $2n$ *bits*. Portanto, um somador completo 5:2 quaternário equivale ao somador binário mostrado na figura 4.11.

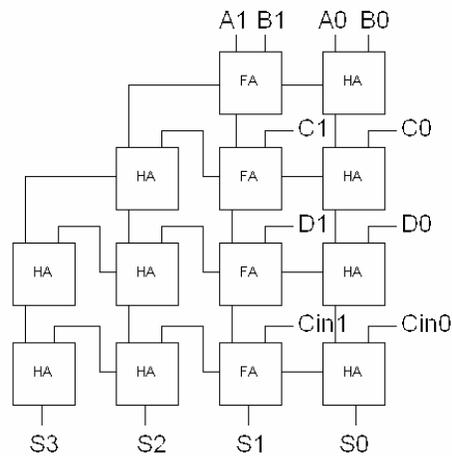


Figura 4.11: Equivalente binário do somador completo 5:2 quaternário

Para somar cinco números binários de dois *bits* são usados nove meio-somadores (HA) e quatro somadores completos (FA) binários arranjados conforme esquemático da figura 4.11. Na simulação, são usados como HA uma porta XOR e como FA o circuito somador em espelho descrito em (RABAEY, 1996). O circuito binário equivalente ao somador completo 5:2 quaternário é simulado realizando a soma de quatro números de 2 *bits*, 4 *bits*, 6 *bits* e 8 *bits* e os resultados mostram atrasos críticos de 1919ps, 2609ps, 3498ps e 4173ps, consumos de 0,98mW, 2,03mW, 3,13mW e 4,25mW à 200MHz e número de transistores usados de 222, 390, 558 e 726, respectivamente.

4.4 Análise dos Resultados e Discussões

Ao longo desse capítulo são apresentados os resultados de simulações de desempenho de diversos circuitos quaternários e uma vez que temos diferentes circuitos realizando a mesma operação lógica, uma comparação entre os resultados se faz necessária.

Para realizar a operação de inversão, se pode usar o circuito inversor quaternário CMOS ou o circuito multiplexador e os resultados dessas realizações estão mostrados na tabela 4.6, juntamente com o circuito inversor binário para comparação. A comparação entre os inversores quaternários mostra que o circuito inversor quaternário

que foi desenvolvido especificamente para realizar essa função tem a vantagem sobre a implementação do multiplexador em área, consumo de potência e atraso médio.

Tabela 4.6: Comparação entre as diversas realizações da função inversora

circuito	atrasos (ps)		consumo (μ W)	transistores usados
	crítico	médio	à 250 MHz	
inversor CMOS	773	471	7,6	6
inversor MUX completo	584	487	79,9	24
inversor MUX reduzido	642	512	74,7	18
inversor binário	108	105	33,8	2

Os circuitos repetidores e osciladores que também são construídos usando inversores, têm as comparações entre as diferentes implementações apresentadas nas tabelas 4.7 e 4.8.

Tabela 4.7: Comparação entre as diversas realizações da função repetidora

circuito repetidor usando:	Atrasos (ps)		Consumo (μ W)	Nº de transistores usados
	Crítico	Médio	à 250 MHz	
2 inversores CMOS quaternários	1414	1055	19,5	12
2 inversores MUX completos	1195	1064	160,5	48
2 inversores MUX reduzidos	1326	1120	153,6	36
1 inversor MUX completo	499	392	77,7	24
1 inversor MUX reduzido	612	407	66,9	18

Na comparação entre os repetidores, o circuito inversor apresenta vantagens em consumo e área, enquanto o inversor MUX possui melhor desempenho. Em relação ao desempenho, o circuito repetidor que usa apenas um MUX para copiar a entrada de controle na saída é a melhor opção.

Tabela 4.8: Comparação entre os diversos osciladores em anel

circuito oscilador em anel	Freqüência de oscilação (MHz)	Consumo (μ W)	nº de transistores usados
3 inversores CMOS	468	45	18
3 inversores MUX completos	320	367	72
3 inversores MUX reduzidos	259	273	54
5 Sucessores	140	1296	120
3 inversores binários	2200	936	6

A comparação entre os osciladores confirma os resultados anteriores quando comparados os circuitos MUX e inversor quaternário. A nova forma de realizar a oscilação passando pelos quatro níveis lógicos ciclicamente, realizada por cinco sucessores dispostos em anel, apresenta menor desempenho e maior consumo, comparada às demais.

Os resultados das simulações para as diferentes formas de realizar o somador parcial estão sumarizados na tabela 4.9.

Tabela 4.9: Comparação entre as realizações do somador parcial 2:2

somador parcial 2:2	atrasos (ps)		consumo (μ W) à 250 MHz	n° de transistores usados
	crítico	médio		
MIN-MAX	3452	1807	230	104
Direto	4347	1827	478	68
MUX	1043	512	81	72

A comparação entre os somadores parciais mostra a maior eficiência da implementação que usa multiplexadores. Esse somador tem melhor desempenho e menor consumo e por essa razão é usado na realização do somador completo 5:2.

Os resultados das simulações dos somadores completos 5:2 quaternários de 1, 2, 3 e 4 *bits* e seus equivalentes binários estão mostrados na tabela 4.10.

Tabela 4.10: Comparação entre os somadores quaternário e equivalente binário

número de bits quaternários	Quaternário			Binário		
	Atraso (ns)	Consumo (mW) em 200MHz	Transistores	Atraso (ns)	Consumo (mW) em 200MHz	Transistores
1	1,26	0,726	297	1,92	0,98	222
2	1,45	1,11	701	2,61	2,03	390
3	1,66	1,95	1105	3,5	3,13	558
4	2,27	2,68	1509	4,17	4,25	726

O somador quaternário tem melhor desempenho e menor consumo quando comparado ao seu equivalente binário, embora use um maior número de transistores. Para somar números maiores do que quatro *bits*, técnicas de aceleração do *carry* são empregadas em lógica binária e ainda não foram desenvolvidas para lógica quaternária.

5 APLICAÇÃO DE LÓGICA QUATERNÁRIA EM FPGA

Apesar do desenvolvimento de diversas tecnologias e trabalhos em lógica não binária e da melhoria considerável desses circuitos, especialmente nos últimos anos, ainda não há consenso sobre possíveis aplicações onde a lógica quaternária pode suplantará com vantagens naturais circuitos de lógica binária. A busca por tal aplicação tem sido discutida ao longo dos anos pelos pesquisadores da área e não existe além das memórias *Flash* quaternárias, outra aplicação incontestada delineada.

Um dos principais benefícios do uso de lógica não binária é a compactação da informação. Cada fio pode transmitir mais de dois sinais elétricos distintos e pode-se obter com isso uma grande redução no número de interconexões em um circuito integrado, portanto, aplicações com grandes demandas de interconexões podem ser consideradas aplicações alvo. Tendo em vista as características inerentes ao uso de lógica não binária e os circuitos desenvolvidos neste trabalho, como multiplexadores e LUTs, a aplicação de lógica quaternária em malhas de portas programadas por campo elétrico, conhecidos como FPGAs (do inglês, *Field Programmable Gate Arrays*) torna-se evidente. Um FPGA é um circuito que tem um alto custo com interconexões e, devido à necessidade de ser um circuito de aplicação genérica, não está otimizado ao extremo como o caso dos circuitos integrados de aplicação específica (ASIC - *Application Specific Integrated Circuit*). Por essas razões, este trabalho sugere a aplicação de lógica não binária em FPGAs como sendo uma aplicação natural.

5.1 FPGA

Um FPGA, ao contrário de um ASIC, não possui uma lógica pré-determinada e pode ser programado para implementar praticamente qualquer projeto. Circuitos ASIC geralmente são destinados a otimizar o desempenho em aplicações específicas e reduzir a complexidade e custo de fabricação em circuitos de alto volume de produção. Entretanto, circuitos ASIC possuem elevados custo e tempo de projeto. Em contrapartida, circuitos implementados em FPGA podem ser realizados em relativo curto prazo, não necessitam projeto de *layout* nem fabricação de máscaras ou fabricação de circuitos integrados, o que reduz os custos não recorrentes e tempo para ser lançado no mercado. Por essas razões, circuitos FPGA são vantajosos em aplicações com relativamente baixo volume e são cada vez mais usados.

5.1.1 Arquitetura de um FPGA

A arquitetura básica de um FPGA consiste de uma malha bidimensional de blocos lógicos e flip-flops, nos quais o usuário deve configurar a funcionalidade de cada bloco por meio da programação das entradas e saídas do circuito e das interconexões entre os blocos (BROWN et al., 1992). Existem diversas famílias de FPGAs que diferem entre si

pela maneira física de implementar a programabilidade do usuário, arranjo dos fios e estruturas dos blocos lógicos (DESCHAMPS et al., 2006). Um exemplo de arquitetura básica de um FPGA pode ser visto na figura 5.1.

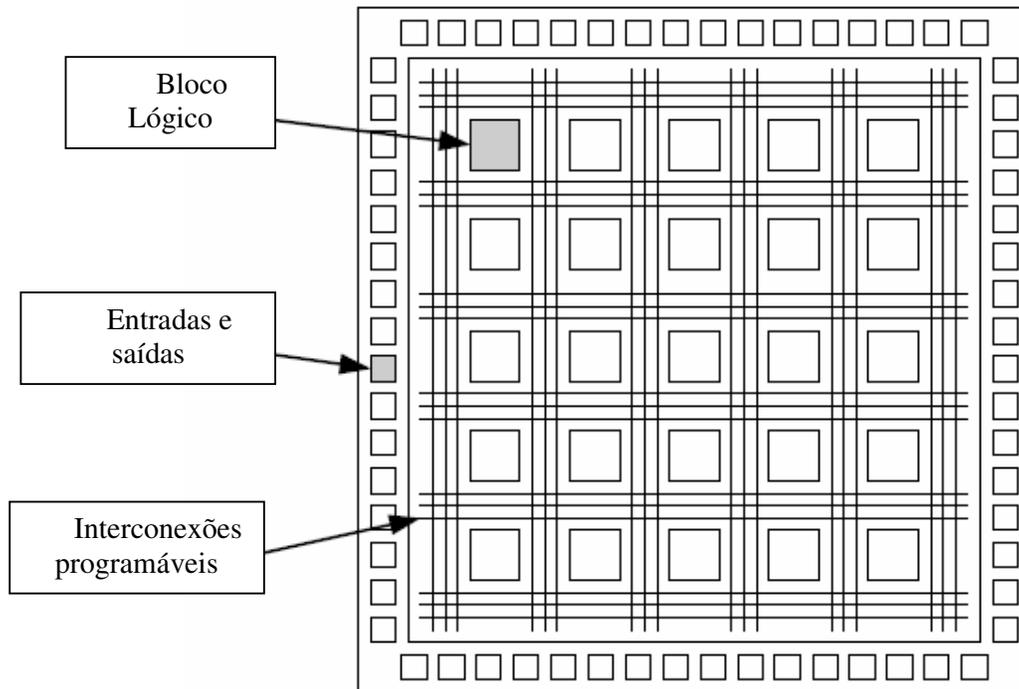


Figura 5.1: Arquitetura básica de um FPGA (DESCHAMPS, 2006)

Existem diversos tipos de FPGAs com diferentes arquiteturas e as interconexões programáveis podem ser implementadas em diferentes tecnologias, como SRAM, fusível reverso (*antifuse*) ou EPROM/EEPROM. As interconexões programáveis são constituídas basicamente por fios e chaves colocadas no cruzamento entre fios que corretamente abertas ou fechadas, determinam os caminhos dos sinais elétricos entre os blocos programáveis. O uso de lógica quaternária em FPGAs não tem influência sobre a maneira como são construídas essas chaves. Os blocos de entrada e saída são constituídos basicamente por repetidores e podem ser facilmente substituídos por repetidores quaternários. O bloco lógico programável, ou bloco lógico configurável, geralmente é construído usando LUTs e um registrador que opcionalmente armazena o sinal de saída da LUT.

Um bloco lógico básico exemplo de um FPGA está mostrado na figura 5.2. Nesse exemplo, o bloco lógico configurável (CLB) utiliza Look-up Table de quatro entradas. Comercialmente, existem FPGAs construídos utilizando LUTs de diferentes tamanhos, mas basicamente esses blocos são como mostrados na figura 5.2.

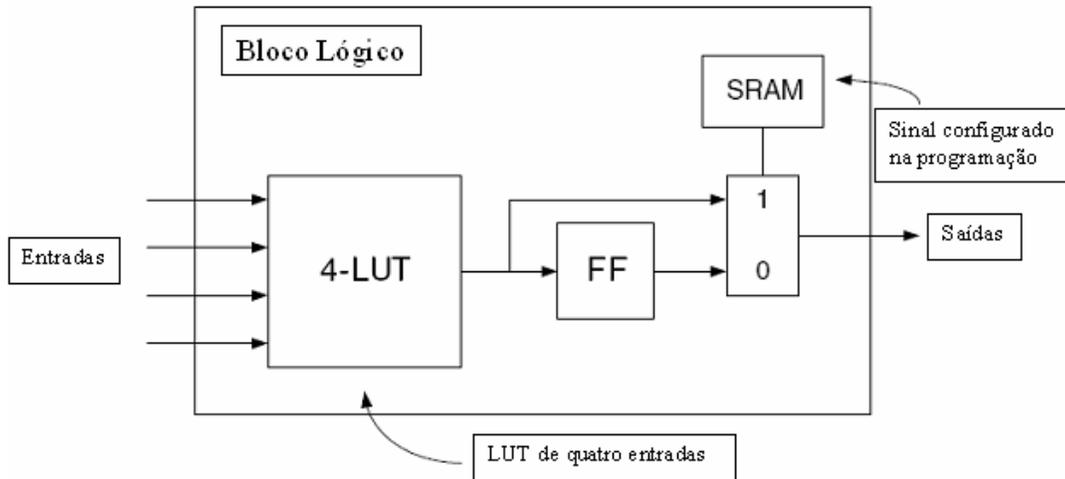


Figura 5.2: Bloco lógico básico de um FPGA (DESCHAMPS, 2006)

O sinal de saída da LUT, que realiza a função lógica pré-programada pelo usuário, é transmitido a um flip-flop ou diretamente levado a saída do CLB, dependendo de um outro sinal pré-programado em uma célula SRAM. Os sinais de saída desse bloco são levados à um bloco de saída ou à entrada de outros blocos configuráveis conforme a programação das interconexões. Com isso, um FPGA é capaz de implementar funções de qualquer complexidade e existem diferentes arquiteturas apropriadas para diferentes aplicações.

A arquitetura de um FPGA, que é destinada a dar maior flexibilidade ao projeto, possui desvantagens inerentes a sua construção frente aos circuitos ASIC. As interconexões programáveis, por exemplo, implicam em um elevado custo em área, consumo e desempenho em FPGA.

Com o advento das tecnologias submicrométricas, as interconexões estão se tornando o aspecto dominante do atraso nos circuitos no estado da arte (DALLY, 2002). Isso é devido ao fato que rapidez, densidade e potência obedecem à lei de Moore, enquanto o produto da resistência pela capacitância aumenta e conseqüentemente leva a um aumento do atraso causado pela rede de fios. Mesmo após a mudança no material para interconexões do alumínio para o cobre e o uso de isolantes com baixa permissividade elétrica, o problema permanece e está a se tornar cada vez mais significativo (METZE et al., 2003; HAVEMANN, 2007). Em FPGAs, as interconexões têm um papel ainda mais crítico, porque elas não só dominam o atraso (PRAGASAM, 2007), mas também severamente impactam o consumo de potência (LI, 2005) e área (SINGH et al., 2002). Para FPGAs com milhões de portas, a área destinada às interconexões programáveis no chip pode chegar a 85-90% da área do chip (KELLY et al., 2005).

5.2 FPGA Quaternário

Considerando o alto custo das interconexões em um FPGA, este trabalho propõe o uso de lógica quaternária como uma solução para a melhoria desses circuitos. Com o uso de lógica não binária, a informação pode ser compactada e a quantidade de fios reduzida, uma vez que cada fio pode transmitir mais do que dois sinais lógicos distintos. Essa característica, por si só, pode levar a redução do custo total de interconexões,

reduzindo área, potência e atraso, o que é especialmente importante em FPGAs e por essa razão, essa aplicação pode ser naturalmente vantajosa para lógica não binária.

Além dos benefícios causados pela redução dos fios no FPGA, existe uma redução pela metade no número de flip-flops no chip. Os flip flops também tem um papel significativo no consumo do FPGA, pois possuem chaves controladas pelos relógios do circuito e mantêm-se chaveando, independente de haver sinal sendo transmitido através do flip-flop. Uma vez que os sinais de saída das LUTs são quaternários, um flip-flop quaternário substitui dois flip-flops binários e independente do custo de sua implementação, a redução pela metade do número de chaves acionadas pelo relógio implica em significativa redução do consumo de potência do circuito.

Lógica não binária direcionada à FPGA já foi sugerida em (ZILIC et al., 1993; KELLY et al., 2005; SHEIKHOLESLAMI et al., 1998). Entretanto, a maioria dos esforços para realizar FPGAs não binários tem apresentado circuitos de modo corrente empregados nos blocos lógicos configuráveis. Esses circuitos apresentaram algum sucesso na redução da área, mas seu excessivo consumo estático impediu sua aceitação como uma alternativa viável aos circuitos padrões binário.

Este trabalho que mostra novas realizações em LUTs quaternárias, também apresenta suas aplicações em blocos programáveis de FPGAs. Os circuitos simulados para FPGA utilizam a mesma tecnologia TSMC 0,18 μ m, com VDD de 1,8V, a tensão nominal de funcionamento para circuitos nessa tecnologia.

5.2.1 Simulações Elétricas

A proposta deste trabalho é realizar um circuito de lógica programável utilizando lógica quaternária e a mesma estrutura de um FPGA binário comercial. A tensão máxima de alimentação é de 1,8V e os níveis lógicos 1, 2 e 3 correspondem a sinais de tensão de 0,6V, 1,2V e 1,8V, respectivamente. As simulações se concentram nas LUTs, que são os componentes geradores das funções nos blocos lógicos programáveis. São investigadas LUTs binárias, usadas nos FPGAs comerciais e LUTs quaternárias (QLUTs) apresentadas neste trabalho. São comparadas LUT binárias de 2, 4 e 6 entradas, chamadas 2LUT, 4LUT e 6LUT, respectivamente, e LUTs quaternárias de 1, 2 e 3 entradas quaternárias, denominadas 1QLUT, 2QLUT e 3QLUT, respectivamente.

Simulações de todas as possíveis combinações de sinais de entrada apresentam atrasos críticos de propagação de 287ps, 503ps e 592ps e consumos de 28 μ W, 155 μ W e 507 μ W em 500MHz para 1QLUT, 2QLUT e 3QLUT, respectivamente. A fim de estimar a área dos circuitos, apresentamos como fator de comparação o número de transistores usados que são 24, 84 e 528 para 1QLUT, 2QLUT e 3QLUT, respectivamente.

As LUTs binárias foram também simuladas usando a mesma tecnologia e para comparar circuitos capazes de realizar o mesmo número de funções lógicas, os resultados para potência e área devem ser dobrados. Um circuito 1QLUT é equivalente a dois circuitos 2LUT em paralelo, enquanto os circuitos 2QLUT e 3QLUT são equivalentes a dois circuitos 4LUT e dois circuitos 6LUT em paralelo, respectivamente. Uma 2LUT apresenta atraso de propagação de 67ps, consumo de potência de 31 μ W em 500MHz e utiliza 16 transistores. Os circuitos 4LUT e 6LUT, têm atraso de propagação de 146ps e 261ps, consumo de 145 μ W e 505 μ W em 500MHz e utilizam 68 e 264 transistores, respectivamente. Um sumário dos resultados está apresentado na tabela 5.1.

Os resultados mostram que os circuitos quaternários possuem menor consumo e área do que os circuitos equivalentes binários enquanto o desempenho dos circuitos binários é superior. Entretanto, uma vez que são comparados apenas as LUTs, o atraso e consumo adicionados pela rede de fios do circuito não são levados em conta, tampouco os consumos dos flip-flops.

Tabela 5.1: Comparação entre LUTs binárias e quaternárias

Quaternário			Equivalente Binário
QLUT	287	Atraso (ps)	67
	28	Potência (μ W)	62
	24	Nº de transistores	32
2QLUT	503	Atraso (ps)	146
	155	Potência (μ W)	290
	84	Nº de transistores	136
3QLUT	592	Atraso (ps)	261
	507	Potência (μ W)	1010
	288	Nº de transistores	528

O resultado das simulações elétricas apresenta a lógica quaternária com características competitivas com os atuais circuitos binários. A comparação entre os blocos lógicos, entretanto, pode não ser representativa do ganho proveniente do uso de lógica quaternária e circuitos maiores devem ser também comparados. Uma vez que existe um número maior de funções lógicas existentes em lógica quaternária do que em lógica binária, para um mesmo número de variáveis ou sinais de controle, o mapeamento de funções lógicas em circuitos contendo apenas LUTs pode dar uma maior dimensão da vantagem de usar lógica não binária em FPGAs.

5.2.2 Mapeamento Lógico

A programação de um FPGA é feita mapeando o circuito, através de suas funções lógicas, em seus blocos lógicos programáveis e programando as interconexões para transmitir os sinais entre os diferentes blocos lógicos e entre blocos lógicos e os blocos de entrada e saída do chip. Existem ferramentas automatizadas, distribuídas pelos fabricantes, para realizar esse mapeamento de acordo com a estrutura de cada FPGA. Em um nível mais geral, existem softwares de mapeamento lógico que fazem a otimização da implementação de funções lógicas usando como blocos fundamentais qualquer porta lógica pré-estabelecida, como inversores, portas AND, OR ou usando apenas LUTs, que é o caso de alguns FPGAs. Essas ferramentas, no entanto, existem apenas em lógica binária e não é possível mapear automaticamente funções quaternárias.

Além da inexistência de softwares de mapeamento quaternário, ferramentas de tradução de lógica binária para lógica quaternária também não são encontrados. Dessa forma, a comparação entre as implementações binária e quaternária de funções lógicas mapeadas em LUTs não podem ser feitas de maneira automatizada. Apesar disso,

funções quaternárias podem ser traduzidas em funções binárias através de suas tabelas verdades. Isso é feito atribuindo a cada sinal quaternário, dois sinais binários. Uma função de três entradas e duas saídas quaternárias, por exemplo, pode ser transformada facilmente em uma função binária de seis entradas e quatro saídas. As funções binárias, agora representadas por suas tabelas verdades podem ser usadas em ferramentas de mapeamento usando apenas LUTs e se obtêm através desse mapeamento o número de LUTs necessárias para implementar tais funções em um FPGA binário.

O mapeamento quaternário em QLUTs deve ser feito manualmente e, portanto, apenas pequenas funções lógicas puderam ser usadas para fins de comparações com a lógica binária. Neste trabalho, foram usadas algumas funções quaternárias de até seis variáveis quaternárias e comparadas suas implementações em QLUTs com a implementação em LUTs binárias das funções correspondentes em lógica binária. O resultado do número de LUTs necessários para cada implementação combinado com os resultados da tabela 5.1 provê uma razoável idéia dos seus custos de implementação.

O primeiro circuito estudado é um somador de três números quaternários (Soma3). Esse circuito soma os números A, B e D e o resultado são dois *bits*, um *bit* de soma e um *bit* de *carry*. A equação da soma está mostrada na figura 5.3.a) e o esquemático desse circuito está mostrado na figura 5.3.b). A tabela verdade dessa função está mostrada no apêndice B, como exemplo.

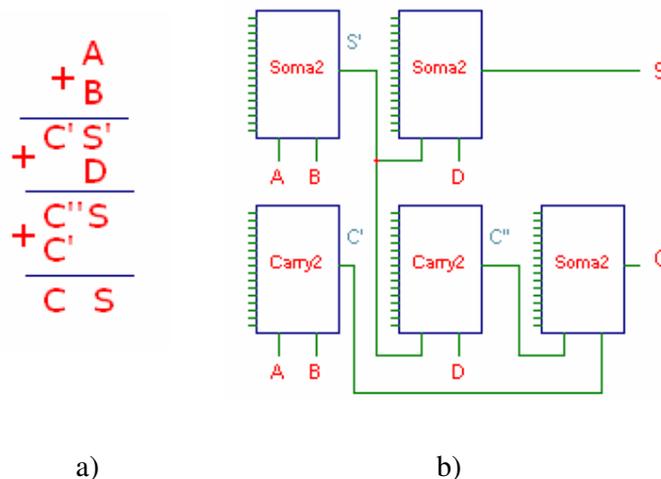


Figura 5.3: a) Operação de soma de 3 números quaternários e b) sua implementação usando 2QLUTs

No circuito esquemático, soma2 significa soma de dois números e carry2 significa a avaliação do carry de saída da soma de dois números. Cada bloco representa um circuito 2QLUT e a implementação do circuito soma3 necessita de cinco circuitos 2QLUT. O caminho crítico desse circuito passa por três 2QLUTs em série. Utilizando os valores da tabela 5.1, desconsiderando o atraso e potência dissipada pelas interconexões, se pode ver que esse circuito utiliza 420 transistores e consome 0,775mW em 500MHz. O atraso crítico desse circuito é de 1,51ns.

Quando mapeada em 3QLUTs a soma3 pode ser implementada como mostra a figura 5.4.

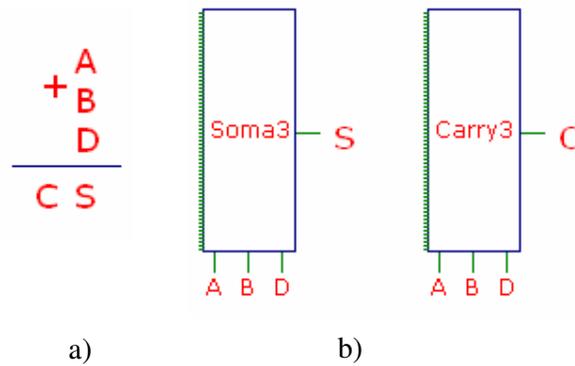


Figura 5.4: a) Operação de soma de 3 números quaternários e b) sua implementação usando 3QLUTs

A implementação da função soma3 em 3QLUT pode ser feita usando apenas um bloco para a avaliação da soma e um bloco para a avaliação da *carry* de saída, uma vez que esse circuito admite até três entradas quaternárias. Esse circuito utiliza 576 transistores e consome 1,01mW em 500MHz. O atraso é de 0,59ns, o mesmo atraso do circuito 3QLUT.

A correspondência em lógica binária é feita usando a ferramenta de mapeamento ABC (MICHSHENKO, 2007). O arquivo de entrada do ABC é a tabela verdade da função soma3 em lógica binária. Como mostra o Apêndice B, a cada variável quaternária, são atribuídas duas variáveis binárias e o circuito soma3 que possui três entradas e duas saídas quaternárias, em lógica binária possui seis entradas e quatro saídas. Com a tabela verdade, são simuladas as implementações da função usando 2LUT, 4LUT e 6LUT com a ferramenta ABC. Usando apenas circuitos 2LUT, são necessários 62 blocos e o caminho crítico passa por sete 2LUTs em série, como mostra o resultado da simulação no ABC (ver apêndice C). Com isso o número de transistores usados é de 992, o atraso crítico é de 0,47ns e a potência consumida é de 1,92mW em 500MHz. Usando 4LUT como bloco básico, a implementação utiliza 18 blocos com três em série no caminho crítico. O número de transistores usados nesse caso é de 1224, o atraso é de 0,44ns e o consumo é de 2,61mW em 500MHz. Com o circuito 6LUT como bloco fundamental, são necessários apenas quatro blocos em paralelo. O número de transistores usados é de 1056, o atraso crítico é de 0,26ns e o consumo é de 2,02mW em 500MHz. Com essa simulação, pode-se comparar implementações da mesma função utilizando diferentes blocos lógicos. Para o circuito soma3, o número de transistores usados e a potência consumida são menores usando 2QLUT como bloco básico enquanto o circuito mais rápido é o que usa 6BLUT.

Aumentando o grau de complexidade das funções, no que diz respeito ao número de entradas, saídas e blocos necessários para a implementação, foram simulados somadores de 4 (soma4), 5 (soma5) e 6 (soma6) números quaternários de 1 *bit*, um somador de 2 números quaternários de 2 *bits* (soma22) e um produto de dois números quaternários de 2 *bits* (prod22). Cada função teve seu equivalente binário extraído e simulado no ABC da mesma forma que a função soma3. O sumário dos resultados está mostrado na tabela 5.2. A comparação entre potência consumida, atrasos críticos e número de transistores usados para cada função nas diferentes implementações estão mostrados nas figuras 5.5, 5.6 e 5.7.

Tabela 5.2: Resultados do mapeamento lógico das funções soma3, soma4, soma5, soma6, soma22 e prod22 em LUTs quaternárias e binárias

Soma3

bloco básico	nº de blocos	nº de blocos em série	Atraso (ns)	Potência (mW)	nº de transistores
3QLUT	2	1	0,592	1,014	576
2QLUT	5	3	1,509	0,775	420
6LUT	4	1	0,261	2,02	1056
4LUT	18	3	0,438	2,61	1224
2LUT	62	7	0,469	1,922	992

Soma22

bloco básico	nº de blocos	nº de blocos em série	Atraso (ns)	Potência (mW)	nº de transistores
3QLUT	4	2	1,184	2,028	1152
2QLUT	7	3	1,509	1,085	588
6LUT	11	2	0,522	5,555	2904
4LUT	19	3	0,438	2,755	1292
2LUT	66	8	0,536	2,046	1056

Soma4

bloco básico	nº de blocos	nº de blocos em série	Atraso (ns)	Potência (mW)	nº de transistores
3QLUT	4	2	1,184	2,028	1152
2QLUT	8	3	1,509	1,24	672
6LUT	20	3	0,783	10,1	5280
4LUT	53	5	0,73	7,685	3604
2LUT	197	13	0,871	6,107	3152

Prod22

bloco básico	nº de blocos	nº de blocos em série	Atraso (ns)	Potência (mW)	nº de transistores
3QLUT	16	4	2,368	8,112	4608
2QLUT	23	6	3,018	3,565	1932
6LUT	60	3	0,783	30,3	15840
4LUT	151	5	0,73	21,895	10268
2LUT	405	13	0,871	12,555	6480

Soma5

bloco básico	nº de blocos	nº de blocos em série	Atraso (ns)	Potência (mW)	nº de transistores
3QLUT	4	2	1,184	2,028	1152
2QLUT	11	3	1,509	1,705	924
6LUT	53	4	1,044	26,765	13992
4LUT	140	6	0,876	20,3	9520
2LUT	414	15	1,005	12,834	6624

Soma6

bloco básico	nº de blocos	nº de blocos em série	Atraso (ns)	Potência (mW)	nº de transistores
3QLUT	7	3	1,776	3,549	2016
2QLUT	19	5	2,515	2,945	1596
6LUT	190	5	1,305	95,95	50160
4LUT	378	7	1,022	54,81	25704
2LUT	973	20	1,34	30,163	15568

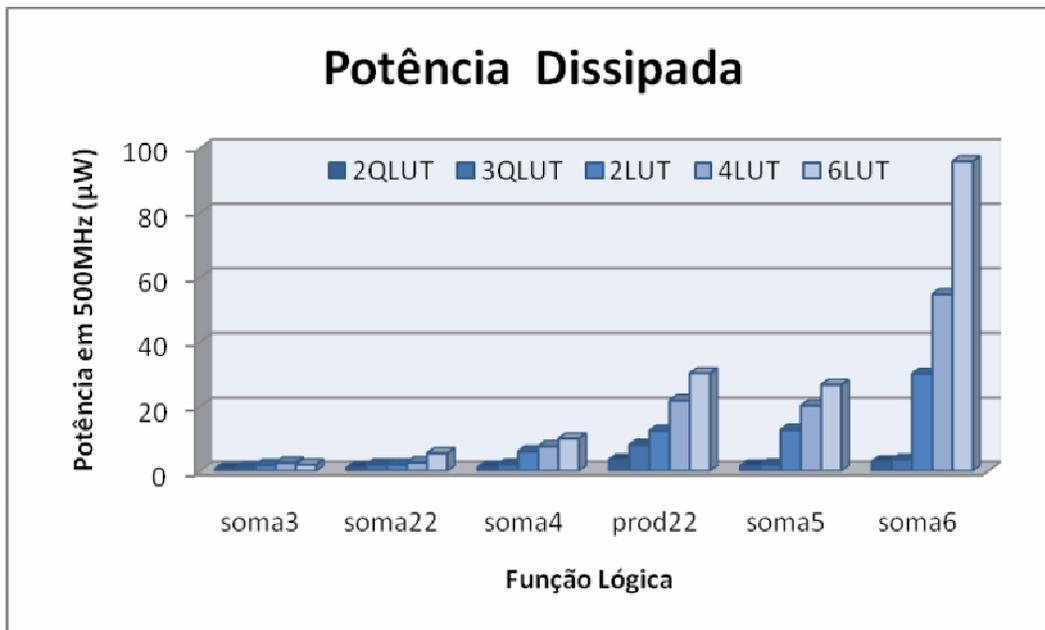


Figura 5.5 Potência dissipada nos circuitos simulados usando diferentes blocos básicos

A partir dos resultados, pode-se ver que as implementações quaternárias consomem menos potência do que as implementações binárias em todos os circuitos simulados e a redução em potência é maior para circuitos mais complexos. Para o circuito soma6, por exemplo, o consumo da implementação usando 2QLUT é apenas 3% do consumo do circuito implementado usando 6LUT.

Para o número de transistores usados, também se verifica uma vantagem para as implementações quaternárias e essa vantagem também aumenta para circuitos mais complexos. Para o circuito soma6, o número de transistores usados pela implementação do circuito que usa 2QLUT como bloco básico é pouco mais de 3% do número de transistores da implementação que usa 6LUT como bloco básico.

Em relação ao desempenho, as implementações binárias têm vantagem em relação às implementações quaternárias em todos os circuitos, entretanto, não é verificado um aumento da diferença em função da complexidade do circuito. No circuito soma6, o atraso crítico da implementação 2QLUT é 2,46 vezes maior do que a implementação 4LUT.

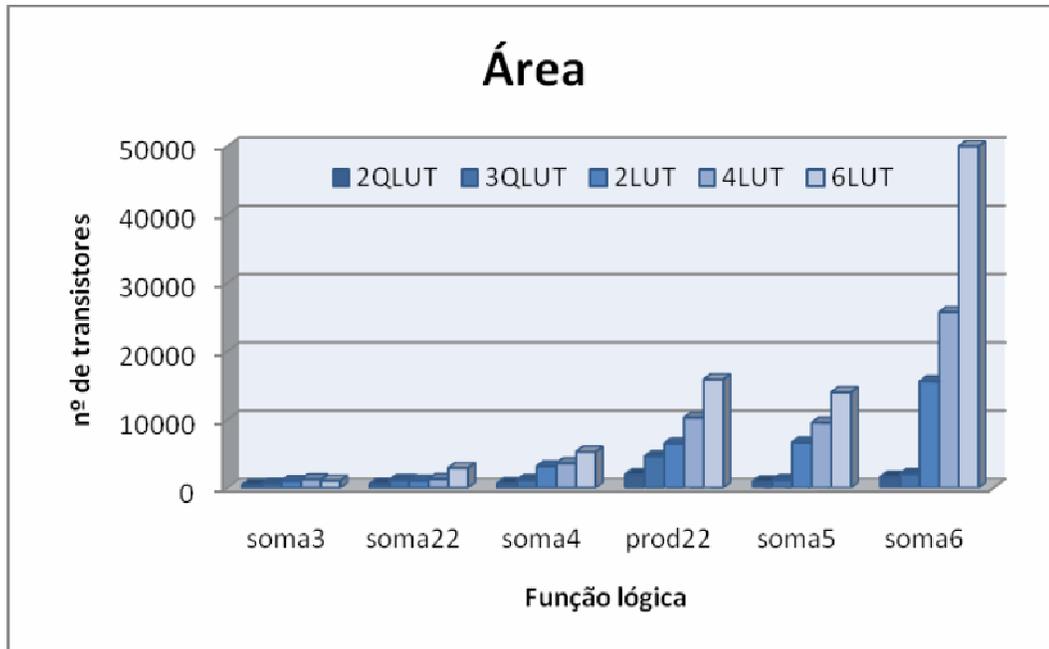


Figura 5.6: Número de transistores usados nos circuitos simulados usando diferentes blocos básicos

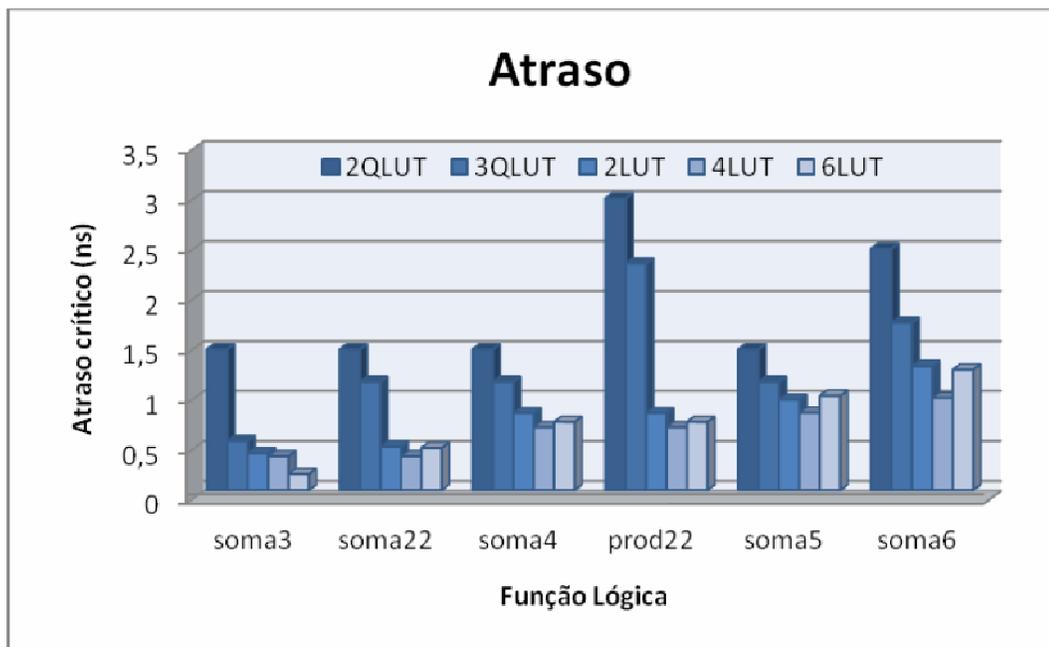


Figura 5.7: Atraso crítico dos circuitos simulados usando diferentes blocos básicos

Através da métrica custo energético de implementação, o atraso vezes a potência dissipada, se pode ver que em alguns casos a implementação quaternária é vantajosa e em alguns casos a implementação binária é melhor. Para os circuitos mais complexos, no entanto, os resultados mostram as implementações quaternárias com menor custo energético do que as binárias. Para o circuito soma6, por exemplo, o custo energético da implementação 2QLUT é 5% do custo energético da implementação 6LUT.

Os resultados mostram que apenas a comparação entre as características dos blocos lógicos não é suficiente para avaliar completamente os custos do uso do FPGA quaternário. Uma vez que existem diferentes funções em lógica quaternária que não possuem correspondência em lógica binária, as operações lógicas podem ser realizadas de maneira diferente. A partir dos resultados obtidos com circuitos mais complexos, pode-se ver que à medida que aumenta a complexidade do circuito as implementações em lógica quaternária se tornam mais vantajosas frente às implementações binárias.

5.2.3 Discussões

As simulações elétricas e de mapeamento lógico confirmam a funcionalidade e desempenho dos circuitos desenvolvidos e sua potencial aplicação em FPGAs. Também é importante ressaltar que o principal ganho com o uso de lógica quaternária é a compactação da informação, onde cada fio pode carregar o equivalente a dois *bits* binários e a avaliação desse ganho não pode ser verificada apenas pelos experimentos realizados nesse trabalho, entretanto, cabe elucubrar sobre as suas eventuais implicações.

No que diz respeito à área dos circuitos, os resultados mostram uma grande redução no número de transistores usados quando empregada a lógica quaternária no lugar de lógica binária. Entretanto, a substancial redução da área dos blocos lógicos é apenas uma pequena parte do potencial de ganho em área desse circuito. O maior ganho deve vir da redução da área das interconexões que em FPGAs modernos, podem responder por mais de 85% da área do chip (KELLY et al., 2005). Se a cada fio for permitido transmitir 4 níveis discretos de sinal, as interconexões poderiam ser reduzidas, a grosso modo, pela metade, e por conseqüência a área usada pelas interconexões. Como exemplo, pode-se considerar o circuito 3QLUT que possui área próxima à metade da área do circuito equivalente 6LUT. Somado à redução na área das interconexões, pode-se estimar que o circuito quaternário deve ter em torno de metade da área de um circuito de mesma capacidade binário. A redução pela metade na área do chip, por si só já traz uma redução talvez à metade no custo de fabricação do chip, o que pode tornar os circuitos quaternários atrativos ao mercado.

Em relação ao consumo de potência, os blocos lógicos quaternários apresentam uma grande melhoria em relação aos blocos lógicos binários. Essa melhoria se deve à redução no número de transistores e à reduzida tensão média transmitida, uma vez que alguns sinais transmitidos são frações do VDD binário. Outro ponto a considerar é o consumo de potência dos flip-flops, que são responsáveis por uma grande parte do consumo total em um FPGA. Com o uso de lógica quaternária, o número de flip-flops poderá ser reduzido à metade. A redução da potência dissipada nos fios também deve ser reduzida consideravelmente com o uso de lógica quaternária.

Comparando desempenhos, os blocos lógicos quaternários são inferiores aos correspondentes binários. O atraso crítico para o bloco 3QLUT é cerca de 2,3 vezes maior do que o atraso do correspondente 6LUT. Para a realização dos circuitos propostos, essa proporção não aumenta com o aumento da complexidade do circuito. Entretanto, apesar do maior atraso crítico, um FPGA quaternário pode ter essa relação diminuída ou até ser mais rápido do que os atuais binários se levado em conta a redução da área e no número de chaves no caminho dos fios. A redução na área das interconexões leva a uma aproximação dos blocos lógicos programáveis e, conseqüentemente, à redução do comprimento médio dos fios entre os blocos. A redução do número de fios e do comprimento médio dos fios leva a uma redução do

produto da capacitância pela resistência associada à rede de interconexões, e por consequência, a uma redução nos atrasos causados pelos fios que é um problema cada vez mais importante. Além da redução dos fios, a compactação da informação pode levar também a redução no número de chaves em que cada sinal deve passar até chegar ao próximo bloco lógico. Essas chaves que são usadas para realizar a programabilidade das interconexões adicionam atrasos no caminho dos sinais elétricos. A redução no número de chaves e nos comprimentos dos fios pode compensar o atraso causado pelo bloco lógico.

Uma vez que os circuitos quaternários apresentam uma reduzida área e consumo em comparação com o equivalente binário, melhorias ainda podem ser alcançadas pelo redimensionamento dos transistores nos circuitos quaternários.

6 CONCLUSÕES E IDÉIAS DE CONTINUAÇÃO DO TRABALHO

Este trabalho propõe uma nova família de circuitos quaternários que usa 3 tensões de alimentação e 8 tipos de transistores com diferentes tensões de limiar. Diversos circuitos foram desenvolvidos e simulados apresentando reduzido consumo de potência, desempenho comparável aos atuais circuitos binários e compatibilidade com os processos de fabricação dos circuitos integrados no estado da arte.

O circuito inversor quaternário CMOS apresenta atraso crítico 7 vezes maior do que o atraso crítico do inversor binário CMOS. O consumo do circuito binário é 4 vezes o consumo do quaternário na mesma frequência.

A comparação entre circuitos repetidores quaternários mostra que a implementação que usa um multiplexador que realiza a função de repetição consome 4 vezes mais do que a implementação que usa dois circuitos inversores CMOS e usa o dobro do número de transistores. Comparando os atrasos, os circuitos inversores apresentam atraso crítico quase três vezes maior do que o atraso do multiplexador.

Na construção de osciladores em anel, a maior frequência de oscilação, menor consumo e área foram verificados usando três inversores CMOS, entre as implementações quaternárias. O oscilador em anel binário que usa um terço do número de transistores da melhor implementação quaternária consome mais de 20 vezes do que esta e alcança uma frequência de oscilação apenas cinco vezes maior.

Nas implementações dos circuitos acima, baseados no inversor, as realizações quaternárias apresentam vantagens muito significativas na redução do consumo de potência em relação aos circuitos binários, entretanto o desempenho é inferior e o número de transistores usados é maior.

A implementação de somadores completos quaternários apresenta significativas vantagens em relação aos circuitos equivalentes em lógica binária. Para a soma de 5 números quaternários de 1 *bit*, o atraso é 65% e o consumo é 74% do equivalente binário e usa 1,33 vezes mais transistores. Na comparação de um somador completo quaternário de cinco números de quatro *bits* quaternários com o circuito equivalente em lógica binária, a implementação quaternária apresenta 55% do atraso crítico, 63% do consumo e utiliza pouco mais de duas vezes o número de transistores do que o circuito equivalente binário. O resultado mostra que circuitos quaternários podem obter maior

vantagem frente aos circuitos equivalentes binários à medida que aumenta a complexidade do circuito.

A aplicação de lógica quaternária em FPGAs foi apresentada como uma aplicação natural para o uso de lógica não binária devido ao grande custo das interconexões e necessidades de reconfiguração desses circuitos. Foram simuladas e comparadas look-up tables (LUTs), que são à base dos blocos lógicos programáveis de diversas famílias de FPGAs.

A comparação entre diferentes LUTs apresenta redução no consumo de potência e no número de transistores usados nas LUTs quaternárias em relação às LUTs binárias, que por sua vez apresentam superior desempenho com menores atrasos críticos. O circuito 1QLUT apresenta 45% do consumo e 75% do número de transistores usados no equivalente binário com um atraso crítico 4 vezes maior. O circuito 2QLUT apresenta 53% do consumo e 61% do número de transistores e atraso 3,4 vezes maior do que o equivalente binário. O circuito 3QLUT apresenta 50% do consumo, 54% do número de transistores e atraso crítico 2,3 vezes maior do que a sua contrapartida binária. Com esses resultados fica evidente que também existe uma melhoria nos parâmetros de comparação da implementação quaternária em relação à equivalente binária à medida que a complexidade do circuito aumenta.

Foi realizado o mapeamento lógico de algumas funções utilizando somente LUTs como blocos básicos. As funções simuladas foram a soma de 3, 4, 5 e 6 números quaternários de 1 *bit*, a soma de 2 números quaternários de 2 *bits* e o produto de dois números quaternários de 2 *bits*.

A potência consumida pelos circuitos é menor para as implementações quaternárias em todos os circuitos estudados e a redução no consumo é maior à medida que aumenta a complexidade do circuito. Para a função soma3, a de menor complexidade estudada, a potência consumida pela implementação binária com 6LUT é duas vezes a potência consumida pela implementação quaternária com 3QLUT. Para a função soma6, a função mais complexa estudada, o consumo da implementação com 6LUT é 27 vezes maior do que o consumo da implementação com 3QLUT. A implementação binária com menor consumo de potência para a função soma6, a que usa 2LUT, consome 10 vezes mais potência do que a implementação quaternária com menor consumo (2QLUT).

Em relação à área, as implementações quaternárias apresentam menores números de transistores usados do que as implementações binárias para todas as funções estudadas. Os resultados também mostram maiores vantagens para os circuitos quaternários à medida que aumenta a complexidade dos circuitos. Para a função soma6, o número de transistores usados é 24 vezes maior para implementação que usa 6LUT em comparação com a implementação quaternária que usa 3QLUT como bloco básico.

Considerando o desempenho dos circuitos, as implementações binárias levam vantagem em relação às implementações quaternárias para todas as funções estudadas. O atraso crítico da implementação quaternária da função soma3 usando 3QLUT é 2,3 vezes maior do que o atraso da implementação binária que usa 6LUT. Para a função soma6 o atraso quaternário é 1,4 vezes o do binário, o que mostra uma melhoria do desempenho dos circuitos quaternários em relação ao correspondente binário à medida que aumenta a complexidade do circuito.

Comparando os parâmetros de atraso, consumo e área, pode-se ver que as implementações quaternárias apresentam melhorias em relação às implementações

binárias à medida que aumenta a complexidade dos circuitos. O mapeamento de funções lógicas de complexidades maiores só é possível de maneira prática com o uso de ferramentas computacionais e não existem, até o momento, ferramentas de síntese e mapeamento lógico quaternárias para esse fim. O desenvolvimento de tais ferramentas deve ser um dos próximos passos no caminho da realização de um FPGA quaternário.

Apesar dos resultados alcançados e possíveis melhorias relacionadas com a redução dos fios, os circuitos quaternários apresentados neste trabalho possuem alguns pontos sensíveis, que incluem a necessidade de realização dos transistores com diferentes tensões de limiar, redução na margem de ruído e a necessidade de três diferentes sinais de alimentação. Entretanto, existe uma completa compatibilidade dos circuitos quaternários propostos com a atual tecnologia binária CMOS e não existe nenhum obstáculo tecnológico que torne ineficaz a realização desses circuitos.

As diferentes tensões de limiar podem ser alcançadas de duas maneiras, através de diversas implantações iônicas ou pelo uso de portas flutuantes. Com o uso de implantação iônica, algumas etapas de implantação são adicionadas ao processo, entretanto, não necessariamente isso implica em aumento de custos de fabricação. Uma vez que com o uso de lógica quaternária, o número de camadas metálicas pode ser reduzido, e os custos envolvidos com a construção de camadas de metal são maiores do que os custos de implantação iônica, essa troca pode resultar em redução nos custos totais de fabricação do circuito integrado. Uma alternativa à implantação iônica é o uso de portas flutuantes para programar a tensão de limiar dos transistores eletricamente de forma similar como é feito em memórias *flash*, mas existe um custo em área associado a essa tecnologia.

A redução das margens de ruído é uma desvantagem inerente ao uso de lógica não binária que usam frações de VDD em sua alimentação. Em aplicações onde o ruído é um fator crítico, a solução é o aumento de VDD, o que aumenta o consumo de potência, mas, por outro lado, reduz os atrasos de propagação. Considerando as três fontes de alimentação, a rede de fios destinada à alimentação do circuito deve ser aumentada, entretanto, esse aumento não impacta consideravelmente na área do circuito, pois esses fios são construídos em camadas superiores e não necessitam de área de silício para distribuir os sinais de tensão. Uma vez que as interconexões mais críticas em um FPGA são as interconexões programáveis, o uso de diferentes redes de alimentações não é um fator limitante no projeto. Circuitos modernos já usam mais de uma fonte de alimentação para os transistores e também diferentes tensões de limiar em técnicas para a redução de consumo.

A partir deste trabalho diferentes linhas de pesquisa podem ser investigadas no sentido de aprimorar os circuitos de lógica quaternária e torná-los mais atrativos mercadologicamente. Uma linha de pesquisa pode buscar a realização dos primeiros circuitos quaternários usando a família de circuitos desenvolvida nesse trabalho. Esses circuitos podem ser realizados em qualquer tecnologia CMOS a fim de provar os conceitos e determinar a melhor forma de realizar as diferentes tensões de limiar. Outra linha de pesquisa pode ir na direção de desenvolver novos circuitos de lógica quaternária como multiplicadores, filtros e os demais blocos usados nos FPGAs. O desenvolvimento de ferramentas computacionais para automatização do processo de desenvolvimento, síntese e verificação de lógica quaternária também será necessário.

REFERÊNCIAS

- BABA, T. Development of Quantum Functional Devices for Multiple-Valued Logic Circuits. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 29., 1999. **Proceedings...** [S.l.]: IEEE, 1999. p. 2-8.
- BAUER, M. et al. A Multi-cell 32 Mb Flash Memory. In: IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE, ISSCC, 42., 1995. **Digest of Technical papers.** [S.l.]: IEEE, 1995. p. 132–133.
- BROWN, T. S. et al. **Field Programmable Gate Arrays.** Norwell, MA: Kluwer Academic Publisher, 1992.
- BUTLER, J. T. Multiple-Valued Logic. **IEEE Potentials**, [S.l.], v. 14, n. 2, p. 11 – 14, 1995.
- CHALMERS UNIVERSITY OF TECHNOLOGY; GÖTEBORG UNIVERSITY. Resources for Many-Valued Logic. Disponível em: <<http://www.cs.chalmers.se/~reiner/mvl-web/>>. Acesso em: 22 nov. 2007.
- CURRENT, K. W. et al. Characteristics of Integrated Quaternary Threshold Logic Full Adders. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 10., 1980. **Proceedings...** [S.l.]: IEEE, 1980. p. 24-30.
- CURRENT, K. W.; FREITAS, D. A.; EDWARDS, F. A. CMOS Quaternary Threshold Logic Full Adder Circuits. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 15., 1985. **Proceedings...** [S.l.]: IEEE, 1985. p. 318-322.
- CURRENT, K. W. A CMOS Quaternary Latch. **IEEE Electronics Letters**, Piscataway, v. 25, n. 13, p. 856-858, 1989.
- CURRENT, K. W. A CMOS Quaternary Threshold Logic Full Adder Circuit with Transparent Latch. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 20., 1990. **Proceedings...** [S.l.]: IEEE, 1990. p. 168-173.
- CURRENT, K. W.; HURLSTON, M. E. A Bidirectional Current-Mode CMOS Multiple-Valued Logic Memory Circuit. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 21., 1991. **Proceedings...** [S.l.]: IEEE, 1991. p. 196-202.
- CURRENT, K. W. Algorithmic Analogue-to-Quaternary Converter Circuit Using Current-Mode CMOS. **Electronics Letters**, UK, v. 28, n. 12, p. 1111-1112, 1992.
- CURRENT, K. W. Current-Mode CMOS Multiple-Valued Logic Circuits. **IEEE Journal of Solid-State Circuits**, Piscataway, v. 29, n. 2, p. 95-107, 1994.

- DAO, T. T.; MCCLUSKEY, E. J.; RUSSEL, K. L. Multivalued Integrated Injection Logic. **IEEE Transactions on Computers**, Los Alamos, v. 26, n. 12, p. 1233-1241, 1977.
- DALLY, W. J. Computer Architecture is All About Interconnect. In: International Symposium on High-Performance Computer Architecture, 8., 2002. **Proceedings...** [S.l.]: IEEE, 2002.
- DESCHAMPS, J. P.; BIOL, G. J.; SUTTER, G. D. **Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems**. Hoboken, NJ: John Wiley & Sons, 2006.
- DUBROVA, E. Multiple-Valued Logic in VLSI: Challenges and Opportunities. In: IEEE NORCHIP CONFERENCE, 17., 1999. **Proceedings...** [S.l.]: IEEE, 1999. p. 340-350.
- DUNDERDALE, H. Current-Mode Circuits for the Many Functions of a Ternary Variable. **Electronics Letters**, UK, v. 6, p. 15-16, 2002.
- ETIEMBLE, D.; MOHSSINE, M.; ISRAEL, M. Comparison of Binary and Multivalued ECL IC for Implementation of Threshold Functions. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 17., 1987. **Proceedings...** [S.l.]: IEEE, 1987. p. 134-141.
- FRANK, D. J. Power-constrained CMOS scaling limits. **IBM Journal of Research and Development**, Armonk, NY, v. 46, n. 2/3, p. 235-244, 2002.
- FRANK, M. P. Approaching the Physical Limits of Computing. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 35., 2005. **Proceedings...** [S.l.]: IEEE, 2005.
- FREITAS, D. A.; CURRENT, K. W. A Quaternary Encoder-Decoder Circuit Design Using CMOS. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 13., 1983. **Proceedings...** [S.l.]: IEEE, 1983. p. 190-195.
- FREITAS, D. A.; CURRENT, K. W. A CMOS Current Comparator Circuit. **Electronics Letters**, UK, v. 19, n. 17, p. 695-697, 1983.
- FREITAS, D. A.; CURRENT, K. W. CMOS Circuits for Quaternary Encoding and Decoding. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 14., 1984. **Proceedings...** [S.l.]: IEEE, 1984. p. 164-168.
- FREITAS, D. A.; CURRENT, K. W. A Simple High-Gain CMOS Voltage Comparator Circuit. **International Journal of Electronics**, [S.l.], v. 57, n. 2, p. 195-198, 1984.
- FRIEDMAN, N.; SALAMA, C.; THOMPSON, P. Realization of Multivalued Integrated Injection of Logic Full Adder. **IEEE Journal of Solid-State Circuits**, Piscataway, v. SC-12, n. 5, p. 532-534, 1977.
- GONZALEZ, A. F.; MAZUMDER, P. Multiple-Valued Signed-Digit Adder Using Negative Differential-Resistance Devices. **IEEE Transactions on Computers**, Los Alamos, v. 47, p. 947-959, 1998.
- GONZALEZ, A. F. et al. CMOS Implementation of Multiple-Valued Logic Signed-Digit Full Adder Based on Negative-Differential-Resistance Devices. **IEEE Journal of Solid-State Circuits**, Piscataway, v. 36, p. 924-932, 2005.
- HANYU, T.; SAITO, T.; KAMEYAMA, M. Asynchronous Multiple-Valued VLSI System Based on Dual-Rail Current-Mode Differential Logic. In: INTERNATIONAL

- SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 28., 1998. **Proceedings...** [S.l.]: IEEE, 1998. p. 134-139.
- HANYU, T.; IKE, T.; KAMEYAMA, M. Self-Checking Multiple-Valued Circuit Based on Dual-Rail Current-Mode Differential Logic. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 29., 1999. **Proceedings...** [S.l.]: IEEE, 1999. p. 275-279.
- HANYU, T.; MOCHIZUKI, A.; KAMEYAMA, M. Multiple-Valued Dynamic Source-Coupled Logic. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 33., 2003. **Proceedings...** [S.l.]: IEEE, 2003. p. 207-212.
- HAVEMANN, B. Performance & Productivity Improvements of Cu/Low k Interconnects. Disponível em: <http://www.shareholder.com/novellus/downloads/nikkei.pdf>. Acesso em: 22 nov. 2007.
- HEUNG, A.; MOUFTAH, H. T. Depletion/Enhancement CMOS for a Low Power Family of Three-Valued Logic Circuits. **IEEE Journal of Solid-State Circuits**, v. SC-20, n. 2, p. 609-616, 1985.
- HURST, S. L. Multiple-Valued Logic - Its Status and its Future. **IEEE Transactions on Computers**, New York, v. C-33, n.12, p. 1160-1179, 1984.
- KAWAHITO, S. et al. A High-Speed Compact Multiplier Based on Multiple-Valued Bi-Directional Current-Mode Circuits. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 17., 1987. **Proceedings...** [S.l.]: IEEE, 1987. p. 172-180.
- KELLY, P. M. et al. Exploiting Binary Functionality in Quaternary Look-Up Tables for Increased Functional Density in Multiple-Valued Logic FPGAs. **Electronics Letters**, UK, v. 41, n.6, p. 300-302, 2005.
- KERKHOFF, H. G.; DIJKSTRA, H. The Application of CCDs in Multiple-Valued Logic. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 9., 1979. **Proceedings...** [S.l.]: IEEE, 1979. p. 304-309.
- KERKHOFF, H. G. Multiple-Valued Charge-Coupled Devices. **IEEE Transactions on Computers**, Los Alamos, v. 30, p. 644-652, 1981.
- KOBRINSKY, M. J. et al. On-Chip Optical Interconnects. **Intel Technology Journal**, [S.l.], v.8, n.2, p. 129-143, 2004.
- LABLANS, P. Ternarylogic LLC. Disponível em: <http://www.ternarylogic.com/>. Acesso em: 22 nov. 2007.
- LEBLEBICI, Y.; GURKAYNAK, F. K. Modular Realization of Threshold Logic Gates for High Performance Digital Signal Processing Applications. In: IEEE ASIC CONFERENCE, 11., 1998. **Proceedings...** [S.l.]: IEEE, 1998. p. 281-285.
- LI, F.; LIN, Y.; He, L. Power Modeling and Characteristics of Field Programmable Gate Arrays. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, New York, v. 24, n. 11, 2005.
- ŁUKASIEWICZ, J. O logice trójwartościowej. **Ruch Filozoficzny**, v. 5, p. 169-171, 1920.

- MANY-Valued Logic. In: ZALTA, E. N. (Ed.). **Stanford Encyclopedia of Philosophy**. Disponível em: <<http://plato.stanford.edu/entries/logic-manyvalued/#Pro>>. Acesso em: 22 nov. 2007.
- METZE, G. et al. Heterogeneous Integration. **Tech Trend Notes**, [S.l.], v. 12, n. 2, p. 3, 2003.
- MICHSHENKO, A. **ABC**: A System for Sequential Synthesis and Verification. Disponível em: <<http://www.eecs.berkeley.edu/~alanmi/abc/>>. Acesso em: 22 nov. 2007.
- MOORE, G. E. **Cramming more components onto integrated circuits**. New York: McGraw-Hill, 1965. p. 114-117.
- MOORE, G. E. Progress in digital integrated electronics. In: INTERNATIONAL ELECTRON DEVICES MEETING, 1975. **Proceedings...** [S.l.]: IEEE, 1975. p. 11-13.
- NAZARALE, J.; MORAGA, C. Minimum Realization of Ternary Threshold Functions. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 4., 1974. **Proceedings...** [S.l.]: IEEE, 1974. p. 347-358.
- OZDEMIR, H. et al. A Capacitive Threshold-Logic Gate. **IEEE Journal of Solid-State Circuits**, Piscataway, v. 31, n. 8, p. 1141-1150, 1996.
- PARK, S. J. et al. Design of Quaternary Logic Gate Using Double Pass-Transistor Logic with Neuron MOS Down Literal Circuit. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 34., 2004. **Proceedings...** [S.l.]: IEEE, 2004. p. 193-203.
- PRAGASAM, R. **Spartan FPGAs – The Gate Array Solution**. Disponível em: <http://www.xilinx.com/support/documentation/application_notes/xapp120.pdf>. Acesso em: 17 out. 2007.
- PUGSLEY, J. H.; SILIO, C. B. Some I²L Circuits for Multiple-Valued Logic. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 8., 1978. **Proceedings...** [S.l.]: IEEE, 1978. p. 23-31.
- RABAEY, J. M. **Digital Integrated Circuits: A Design Perspective**. Upper Sadle River: Prentice Hall, NJ, 1996.
- RAYCHOWDHURY, A.; ROY, K. Carbon Nanotube Based Voltage-Mode Multiple-Valued Logic Design. **IEEE Transactions on Nanotechnology**, [S.l.], v. 4, p. 168-179, 2005.
- SCHMID, A.; LEBLEBICI, Y. Realisation of Multiple-Valued Functions Using the Capacitive Threshold Logic Gate. **IEE Proceedings – Computers and Digital Techniques**, [S.l.], v. 151, n. 6, p. 435-447, 2004.
- SHEIKHOESLAMI, A.; YOSHIMURA, R.; GULAK, P. G. Look-Up Tables for Multiple-Valued, Combinational Logic. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 28., 1998. **Proceedings...** [S.l.]: IEEE, 1998. p. 264-269.
- SHEN, J. et al. Multi-Valued Logic Pass Gate Network Using Neuron-MOS Transistors. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 30., 2000. **Proceedings...** [S.l.]: IEEE, 2000. p. 15-20.

- SHIBATA, T.; OHMI, T. Neuron MOS Voltage-Mode Circuit Technology for Multiple-Valued Logic. **IEICE Transactions on Electronics**, [S.l.], v. E76-C, n. 3, p.347-356, 1993.
- SINGH, A.; MAREK-SADOWSKA, M. Efficient Circuit Clustering for Area and Power Reduction in FPGAs. In: INTERNATIONAL SYMPOSIUM ON FIELD PROGRAMMABLE GATE ARRAYS, 10., 2002. **Proceedings...** [S.l.]: ACM SIGDA, 2002. p. 59-66.
- SMITH, K. C. The Prospects for Multivalued Logic: A Technology and Applications View. **IEEE Transactions on Computers**, New York, v. C-30, n.9, p. 619-634, 1981.
- SUZUKI, M. et al. A 1.5ns 32b CMOS ALU in Double Pass-Transistor Logic. In: IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE, ISSCC, 40., 1993. **Digest of Technical papers**. [S.l.]: IEEE, 1993. p. 90-91.
- THOUIDIS, I. M. et al. Quaternary Voltage-Mode CMOS Circuits for Multiple-Valued Logic. **IEE Proceedings-Circuits, Devices and Systems**, [S.l.], v. 145, n. 2, p.71-77, 1998.
- THOUIDIS, I. M. et al. The Circuit Design of Multiple-Valued Logic Voltage-Mode Adders. In. IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, 4., 2001. **Proceedings...** [S.l.]: IEEE, 2001. p.162-165.
- THOUIDIS, I. M.; SOUDRIS, D.; THANAILAKIS, A. Voltage-Mode Multiple-Valued Logic Adder Circuits. **IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences**, [S.l.], v. E87-C, n. 6, p. 1054-1061, 2004.
- TRONT, J. G.; GIVONE D. D. Multiple-Valued Logic Gates Using MESFETs. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 9., 1979. **Proceedings...** [S.l.]: IEEE, 1979. p. 175-181.
- WAHO, T. Resonant Tunneling Transistor and its Application of Multiple-Valued Logic Circuits. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 25., 1995. **Proceedings...** [S.l.]: IEEE, 1995. p. 130-138.
- WATANABI, T.; MATSUMOTO, M.; LI, T. New Logical-Sum and Logical-Product Circuits Using CMOS Transistors and their Applications to Four-Valued Combinational Circuits. **International Journal of Electronics**, [S.l.], v. 63, n. 2, p. 215-227, 1987.
- WEI, S.; SHIMIZU, K. Residue Arithmetic Circuits Based on the Signed-Digit Multiple-Valued Arithmetic Circuits. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 28., 1998. **Proceedings...** [S.l.]: IEEE, 1998. p. 276-281, 1998.
- WHEATON, L. B.; CURRENT, K. W. Quaternary Threshold Logic Full Adder Circuit with Complementary Inputs. **International Journal of Electronics**, [S.l.], v. 56, n. 4, p. 539-545, 1984.
- YASUDA, Y. et al. Realization of Quaternary Logic Circuits by N-channel MOS Devices. **IEEE Journal of Solid-State Circuits**, [S.l.], v. SC-21, n. 1, p. 162-168, 1986.
- ZILIC, Z.; VRANESIC, Z.G. Multiple-Valued Logic in FPGAs. In: MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS, 36., 1993. **Proceedings...** [S.l.]: IEEE, 1993. p.1553-1556.

APÊNDICE A PARÂMETROS SPICE DA TECNOLOGIA TSMC 0,18µM

```

.MODEL CMOSN NMOS
+VERSION = 3.1
+XJ = 1E-7
+K1 = 0.5887327
+K3B = 2.8964027
7
+DVT0W = 0
+DVT0 = 1.6035635
+U0 = 274.5483998
2.168557E-18
+UC = 4.924907E-11
+AGS = 0.4277991
+KETA = -6.584602E-3
+RDSW = 126.9339775
+WR = 1
1.580422E-8
+XL = 0
7.164555E-9
+DWB = 1.044806E-8
+CIT = 0
+CDSCB = 0
6.028975E-5
+DSUB = 0.0206431
+PDIBLC2 = 2.791109E-3
+PSCBE1 = 5.115147E9
+DELTA = 0.01
+PRT = 0
+KT1L = 0
+UB1 = -7.61E-18
+WL = 0
+WWN = 1
+LLN = 1
+LWL = 0
+CGDO = 7.97E-10
+CJ = 9.497923E-4
+CJSW = 2.40081E-10
+CJSWG = 3.3E-10
+CF = 0
0.6702203
+PK2 = 1.701633E-3
3.490594E-3
+PU0 = 4.3426803
1.220531E-23
+PVSAT = 1.760964E3
2.692994E-3
TNOM = 27
NCH = 2.3549E17
K2 = 3.296042E-3
W0 = 1E-7
DVT1W = 0
DVT1 = 0.4698306
UA = -1.311544E-9
VSAT = 1.203681E5
B0 = 2.545419E-7
A1 = 5.123947E-4
PRWG = 0.4797458
WINT = 0
XW = -1E-8
VOFF = -0.0948017
CDSC = 2.4E-4
ETA0 = 2.975363E-3
PCLM = 0.9449545
PDIBLCB = -0.1
PSCBE2 = 4.180201E-9
RSH = 6.5
UTE = -1.5
KT2 = 0.022
UC1 = -5.6E-11
WLN = 1
WWL = 0
LW = 0
CAPMOD = 2
CGSO = 7.97E-10
PB = 0.8
PBSW = 0.8
PBSWG = 0.8
PVTH0 = -1.983354E-3
WKETA = 8.977957E-4
PUA = -5.90438E-12
PETA0 = 1.003159E-4
LEVEL = 49
TOX = 4.2E-9
VTH0 = 0.379924
K3 = 1E-3
NLX = 1.69233E-
7
DVT2W = 0
DVT2 = 0.0456741
UB =
A0 = 2
B1 = 5E-6
A2 = 0.6436736
PRWB = -0.2
LINT =
DWG = -
NFACTOR = 2.1860065
CDSCD = 0
ETAB =
PDIBLC1 = 0.1898778
DROUT = 0.792674
PVAG = 0
MOBMOD = 1
KT1 = -0.11
UA1 = 4.31E-9
AT = 3.3E4
WW = 0
LL = 0
LWN = 1
XPART = 0.5
CGBO = 1E-12
MJ = 0.3797493
MJSW = 0.1097977
MJSWG = 0.1097977
PRDSW = -
LKETA = -
PUB =
PKETA = -

```

```

.MODEL CMOSP PMOS
+VERSION = 3.1
+XJ = 1E-7
0.4038864
+K1 = 0.6116151
+K3B = 13.8926043
8.639126E-8
+DVT0W = 0
+DVT0 = 0.4951048
+UO = 110.3464702
1.117289E-21
+UC = -9.13705E-11
+AGS = 0.3994475
1.782397E-6
+KETA = 0.0184627
+RDSW = 244.5613257
0.4127198
+WR = 1
2.584371E-8
+XL = 0
3.040166E-8
+DWB = 2.450519E-9
+CIT = 0
+CDSCB = 0
2.596891E-4
+DSUB = 9.340337E-4
1.086516E-3
+PDIBLC2 = -9.292461E-6
+PSCBE1 = 5.44446E10
+DELTA = 0.01
+PRT = 0
+KT1L = 0
+UB1 = -7.61E-18
+WL = 0
+WWN = 1
+LLN = 1
+LWL = 0
+CGDO = 6.44E-10
+CJ = 1.127453E-3
+CJSW = 2.000464E-10
+CJSWG = 4.22E-10
+CF = 0
+PK2 = 2.107981E-3
5.038916E-3
+PUO = -1.2214418
PVSAT = -50
TNOM = 27
NCH = 4.1589E17
K2 = 0.0242617
W0 = 1E-6
DVT1W = 0
DVT1 = 0.2760469
UA = 1.346641E-9
VSAT = 2E5
B0 = 4.956417E-7
A1 = 0.4459567
PRWG = 0.5
WINT = 0
XW = -1E-8
VOFF = -0.0910412
CDSC = 2.4E-4
ETA0 = 1.117742E-3
PCLM = 1.466934
PDIBLCB = -1E-3
PSCBE2 = 5.581708E-10
RSH = 7.5
UTE = -1.5
KT2 = 0.022
UC1 = -5.6E-11
WLN = 1
WWL = 0
LW = 0
CAPMOD = 2
CGSO = 6.44E-10
PB = 0.8404323
PBSW = 0.8528941
PBSWG = 0.8528941
PVTH0 = 2.091041E-3
WKETA = 0.0442974
PUA = -5.88677E-11
PETA0 = 6.864588E-5
LEVEL = 49
TOX = 4.2E-9
VTH0 = -
K3 = 0
NLX =
DVT2W = 0
DVT2 = 0.1
UB =
A0 = 1.8340524
B1 =
A2 = 0.3
PRWB = -
LINT =
DWG = -
NFACTOR = 1.8219343
CDSCD = 0
ETAB = -
PDIBLC1 =
DROUT = 0
PVAG = 2.3008893
MOBMOD = 1
KT1 = -0.11
UA1 = 4.31E-9
AT = 3.3E4
WW = 0
LL = 0
LWN = 1
XPART = 0.5
CGBO = 1E-12
MJ = 0.4083303
MJSW = 0.3315737
MJSWG = 0.3315737
PRDSW = 5.6088997
LKETA = -
PUB = 1E-21+
PKETA = -4.57561E-
3

```

APÊNDICE B TABELA VERDADE DA FUNÇÃO SOMA3

Quaternário 2QLUT					Binário									
A	B	D	S	C	a0	a1	b0	b1	d0	d1	s0	s1	c0	c1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	1	0	1	0	0
0	0	2	2	0	0	0	0	0	1	0	1	0	0	0
0	0	3	3	0	0	0	0	0	1	1	1	1	0	0
0	1	0	1	0	0	0	0	1	0	0	0	1	0	0
0	1	1	2	0	0	0	0	1	0	1	1	1	0	0
0	1	2	3	0	0	0	0	1	1	0	1	1	0	0
0	1	3	0	1	0	0	0	1	1	1	1	0	0	1
0	2	0	2	0	0	0	1	0	0	0	1	0	0	0
0	2	1	3	0	0	0	1	0	0	1	1	1	0	0
0	2	2	0	1	0	0	1	0	1	0	0	0	0	1
0	2	3	1	1	0	0	1	0	1	1	0	1	0	1
0	3	0	3	0	0	0	1	1	0	0	1	1	0	0
0	3	1	0	1	0	0	1	1	0	1	0	0	0	1
0	3	2	1	1	0	0	1	1	1	0	0	1	0	1
0	3	3	2	1	0	0	1	1	1	1	1	1	0	1
1	0	0	1	0	0	1	0	0	0	0	0	1	0	0
1	0	1	2	0	0	1	0	0	0	1	1	0	0	0
1	0	2	3	0	0	1	0	0	1	0	1	1	0	0
1	0	3	0	1	0	1	0	0	1	1	0	0	0	1
1	1	0	2	0	0	1	0	1	0	0	1	0	0	0
1	1	1	3	0	0	1	0	1	0	1	1	1	0	0
1	1	2	0	1	0	1	0	1	1	0	0	0	0	1
1	1	3	1	1	0	1	0	1	1	1	1	0	1	0
1	2	0	3	0	0	1	1	0	0	0	1	1	0	0
1	2	1	0	1	0	1	1	0	0	1	0	0	0	1
1	2	2	1	1	0	1	1	0	1	0	0	1	0	1
1	2	3	2	1	0	1	1	0	1	1	1	0	0	1
1	3	0	0	1	0	1	1	1	0	0	0	0	0	1
1	3	1	1	1	0	1	1	1	1	0	1	0	1	0
1	3	2	2	1	0	1	1	1	1	1	1	1	0	1
2	0	0	2	0	1	0	0	0	0	0	1	0	0	0
2	0	1	3	0	1	0	0	0	0	1	1	1	0	0
2	0	2	0	1	1	0	0	0	1	0	0	0	0	1
2	0	3	1	1	1	0	0	0	1	1	0	1	0	1
2	1	0	3	0	1	0	0	1	0	0	1	1	0	0
2	1	1	0	1	1	0	0	1	0	1	0	0	0	1
2	1	2	1	1	1	0	0	1	1	0	0	1	0	1
2	1	3	2	1	1	0	0	1	1	1	1	0	0	1
2	2	0	0	1	1	0	1	0	0	0	0	0	0	1
2	2	1	1	1	1	0	1	0	0	1	0	1	0	1
2	2	2	2	1	1	0	1	0	1	0	1	0	0	1
2	2	3	3	1	1	0	1	0	1	1	1	1	0	1
2	3	0	1	1	1	0	1	1	0	0	0	1	0	1
2	3	1	2	1	1	0	1	1	0	1	1	0	0	1
2	3	2	3	1	1	0	1	1	1	0	1	1	0	1
2	3	3	0	2	1	0	1	1	1	1	0	0	1	0
3	0	0	3	0	1	1	0	0	0	0	1	1	0	0
3	0	1	0	1	1	1	0	0	0	1	0	0	0	1
3	0	2	1	1	1	1	0	0	1	0	0	1	0	1
3	0	3	2	1	1	1	0	0	1	1	1	1	0	0
3	1	0	0	1	1	1	0	1	0	0	0	0	0	1
3	1	1	1	1	1	1	0	1	0	1	0	1	0	1
3	1	2	2	1	1	1	0	1	1	0	1	0	0	1
3	1	3	3	1	1	1	0	1	1	1	1	1	0	1
3	2	0	1	1	1	1	1	0	0	0	0	1	0	1
3	2	1	2	1	1	1	1	0	0	1	1	0	0	1
3	2	2	3	1	1	1	1	0	1	0	1	1	0	1
3	2	3	0	2	1	1	1	1	0	1	1	0	0	1
3	3	0	2	1	1	1	1	1	0	0	1	0	0	1
3	3	1	3	1	1	1	1	1	0	1	1	1	0	1
3	3	2	0	2	1	1	1	1	1	0	0	0	1	0
3	3	3	1	2	1	1	1	1	1	1	0	1	1	0

APÊNDICE C RESULTADO DA SIMULAÇÃO DA OPERAÇÃO SOMA3 NA FERRAMENTA ABC USANDO 2LUTS COMO BLOCO BÁSICO

.model soma3	.names x4 x5 n34	.names n62 n63 n61
.inputs x0 x1 x2 x3 x4 x5	11 1	00 1
.outputs z0 z1 z2 z3	.names n27 n36 n35	.names x2 n38 n62
.names n13 n20 n12	01 1	01 1
00 1	.names n37 n40 n36	.names x2 n42 n63
.names n14 n15 n13	11 1	11 1
00 1	.names n38 n39 n37	.names n14 n65 n64
.names x2 x4 n14	00 1	01 1
00 1	.names x4 x5 n38	.names x0 x5 n65
11 1	01 1	01 1
.names n16 n18 n15	.names x0 x2 n39	.names n67 n73 n66
00 1	00 1	01 1
.names x0 n17 n16	.names n41 n42 n40	.names x0 n68 n67
11 1	00 1	10 1
.names x1 x3 n17	.names x0 x2 n41	.names n69 n71 n68
11 1	11 1	00 1
.names x0 n19 n18	.names x4 x5 n42	.names n19 n70 n69
01 1	10 1	11 1
.names x1 x3 n19	.names n45 n46 n44	.names x2 x4 n70
00 1	01 1	11 1
.names n21 n24 n20	.names x3 x5 n45	.names n17 n72 n71
00 1	10 1	11 1
.names n22 n23 n21	.names x1 n47 n46	.names x2 x4 n72
00 1	10 1	00 1
.names x0 n17 n22	.names x3 x5 n47	.names n74 n75 n73
01 1	01 1	01 1
.names x0 n19 n23	.names n49 n50 n48	.names n14 n22 n74
11 1	01 1	01 1
.names x2 x4 n24	.names x3 x5 n49	.names n76 n77 n75
01 1	11 1	00 1
10 1	.names x1 n51 n50	.names x0 n14 n76
.names n26 n35 n25	00 1	10 1
00 1	.names x3 x5 n51	.names x0 n70 n77
.names n27 n28 n26	00 1	01 1
01 1	.names n53 n56 z2	.names n12 n25 z0
.names x1 x3 n27	01 1	11 0
00 1	.names n54 n55 n53	.names n44 n48 z1
11 1	00 1	00 0
.names n29 n32 n28	.names x1 n51 n54	.names n58 n66 z3
11 1	10 1	01 0
.names n30 n31 n29	.names x1 n49 n55	.end
00 1	01 1	
.names x4 x5 n30	.names x4 n41 n56	
00 1	11 1	
.names x0 x2 n31	.names n27 n59 n58	
01 1	00 1	
.names n33 n34 n32	.names n60 n64 n59	
00 1	00 1	
.names x0 x2 n33	.names x0 n61 n60	
10 1	10 1	