

UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

**Explorando Técnicas para Otimização da Eficiência
Energética de Circuitos Multiplicadores**

Douglas Machado Borges

Orientador: Prof^a. Dra. Cristina Meinhardt
Co-orientador: Prof. Dr. Vagner Rosa

Rio Grande, 2021

Ficha Catalográfica

B732e Borges, Douglas Machado.

Explorando técnicas para otimização da eficiência energética de circuitos multiplicadores / Douglas Machado Borges. – 2021.

104 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2021.

Orientadora: Dra. Cristina Meinhardt.

Coorientador: Dr. Vagner Rosa.

1. Circuitos Multiplicadores 2. Eficiência Energética 3. Computação Aproximada I. Meinhardt, Cristina II. Rosa, Vagner III. Título.

CDU 004.312

Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

ATA DE SESSÃO DE DEFESA DE DISSERTAÇÃO DE MESTRADO

Ata nº 16/2020

Na data de 21 de dezembro de 2020, às 13 horas, ocorreu a Sessão de Defesa de Dissertação de Mestrado de Douglas Machado Borges, que apresentou a dissertação intitulada “Explorando Técnicas para Otimização da Eficiência Energética de Circuitos Multiplicadores”, realizada sob a orientação da Profa. Dra. Cristina Meinhardt e coorientação do Prof. Dr. Vagner Santos da Rosa. A banca examinadora foi constituída pelos Profs. Dr. Paulo Francisco Butzen (PPGComp), Dra. Alexandra Lackmann Zimpeck (UCPel) e Dr. Leonardo Bandeira Soares (IFRS), sob a presidência da orientadora. Após a apresentação do trabalho, a banca arguiu o candidato e, a seguir, deliberou pela

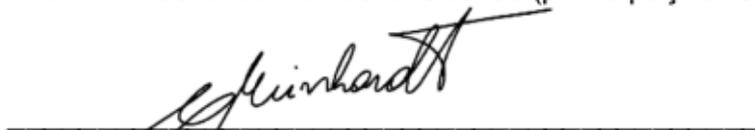
- () aprovação da Dissertação
- (x) aprovação da Dissertação, sugerindo modificações no texto
- () reprovação da Dissertação

Rio Grande, 21 de dezembro de 2020.



Prof. Dr. Paulo Francisco Butzen (participação remota)

Profa. Dra. Alexandra Lackmann Zimpeck (participação remota)

Prof. Dr. Leonardo Bandeira Soares (participação remota)

Profa. Dra. Cristina Meinhardt (participação remota)
Orientadora

Prof. Dr. Vagner Santos da Rosa (participação remota)
Coorientador

AGRADECIMENTOS

Agradeço a todos os meus familiares e amigos. Agradeço também a minha Orientadora e ao meu Co-Orientador pelos suporte e conhecimentos comigo compartilhados.

*“Faça ou não faça. Tentativa não há.”
(Yoda)*

RESUMO

BORGES, Douglas Machado . **Explorando Técnicas para Otimização da Eficiência Energética de Circuitos Multiplicadores**. 2021. 104 f. . Universidade Federal do Rio Grande - FURG, Rio Grande.

Circuitos multiplicadores são componentes de grande importância para os sistemas digitais. Um multiplicador eficiente e de baixo consumo energético é sempre requerido em projetos de hardware pelo fato de seu desempenho afetar substancialmente a performance de todo o sistema onde ele está inserido. Por isso, faz-se necessário avaliar as características elétricas deste componente. Neste contexto, o objetivo deste trabalho é investigar alternativas de projeto de multiplicadores energeticamente eficientes, explorando técnicas para redução energética, como redução da tensão de operação (operando em *near-threshold*) e aproximação adotando somadores aproximados. Os circuitos multiplicadores presentes no estado da arte foram reproduzidos utilizando um modelo preditivo de transistor e simulados eletricamente, buscando por formas de reduzir o consumo energético. A partir do levantamento bibliográfico decidiu-se estudar os multiplicadores Array, Baugh-Wooley, Booth e Vedic. Para fazer uma análise de desempenho mais abrangente foi adotado um fator denominado *Power-delay product* (PDP), que relaciona atraso e potência. Foram realizadas simulações em 8 cenários de teste. Nos dois primeiros, os multiplicadores exatos foram simulados em tensão nominal e em *near-threshold*. Os demais cenários abordaram a substituição do somador *Mirror* (MA) por somadores aproximados AXA e AMA em dois níveis (30% e 100%), com simulações em tensão nominal e de *near-threshold*. Os resultados demonstram que com a redução da tensão é possível atingir uma grande redução em consumo energético, porém, com aumentos significativos nos tempos de atraso. Em comparação, o uso do somador aproximados AMA2 em tensão nominal, proporcionou uma boa economia energética e uma pequena redução no atraso, mas com alguma perda de precisão numérica. A avaliação do impacto da computação aproximada na precisão, motivou a investigação de outros cenários de aproximação nos multiplicadores, buscando uma melhor relação entre redução energética e precisão. Visando aplicações tolerantes a erro, é perceptível que a adoção da computação aproximada pode resultar em cenários com melhor relação entre energia, atraso e precisão. Neste trabalho observou-se que a utilização do somador aproximado AMA2 proporcionou redução de até 30% em potência e até 4% em atraso para circuitos multiplicadores.

Palavras-chave: Circuitos Multiplicadores, Eficiência Energética, Computação Aproximada.

ABSTRACT

BORGES, Douglas Machado . **Exploring Techniques for Optimizing Energy Efficiency in Multiplier Circuits**. 2021. 104 f. . Universidade Federal do Rio Grande - FURG, Rio Grande.

Multiplier circuits are components of great importance for digital systems. An efficient and low energy consumption multiplier is always required in hardware projects because its performance substantially affects the performance of the entire system where it is inserted. Therefore, it is necessary to evaluate the electrical characteristics of this component. In this context, the objective of this work is to investigate alternatives for the design of energy efficient multipliers, exploring techniques for energy reduction, such as reducing the operating voltage (operating in near-threshold) and approximation by adopting approximate additions. The multiplier circuits in the state of the art were reproduced using a predictive transistor model and electrically simulated, looking for ways to reduce energy consumption. Based on the bibliographical survey, it was decided to study the Array, Baugh-Wooley, Booth and Vedic multipliers. To make a more comprehensive performance analysis, a factor called Power-delay product (PDP) was adopted, which relates delay and power. Simulations were performed in 8 test scenarios. In the first two, the exact multipliers were simulated at nominal voltage and at near-threshold. The other scenarios addressed the replacement of mirror adder (MA) by approximate AXA and AMA adders at two levels (30% and 100%), with simulations in nominal voltage and near-threshold. The results show that with the reduction of the voltage it is possible to achieve a great reduction in energy consumption, however, with significant increases in the delay times. In comparison, the use of the AMA2 at nominal voltage, provided good energy savings and a small reduction in delay, but with some loss of numerical accuracy. The evaluation of the impact of approximate computation on precision, motivated the investigation of other approximation scenarios in the multipliers, seeking a better relationship between energy reduction and precision. Aiming at error tolerant applications, it is noticeable that the adoption of approximate computation can result in scenarios with a better relationship between energy, delay and precision. In this work it was observed that the use of the AMA2 provided a reduction of up to 30% in power and up to 4% in delay for multiplier circuits.

Keywords: Multipliers, Energy optimization, Approximate computing.

LISTA DE FIGURAS

Figura 1	Meio-somador	20
Figura 2	Somador <i>Mirror</i> CMOS	21
Figura 3	<i>Ripple-Carry Adder</i>	22
Figura 4	<i>Carry-Lookahead Adder</i>	22
Figura 5	Multiplicador Array de 4 bits	25
Figura 6	Multiplicador Vedic de 2 bits	27
Figura 7	Multiplicadores Vedic de 4 bits	27
Figura 8	Ilustração de uma multiplicação Baugh-Wooley de 8 bits	28
Figura 9	Multiplicador Baugh-Wooley de 4 bits	29
Figura 10	Algoritmo de Booth para a multiplicação em complemento de dois	31
Figura 11	Multiplicador Booth de 4 bits	32
Figura 12	Somador <i>Mirror</i>	35
Figura 13	<i>Simplified Mirror Adder</i> (SMA)	36
Figura 14	<i>Approximate Mirror Adder 1</i> (AMA1)	36
Figura 15	<i>Approximate Mirror Adder 2</i> (AMA2)	37
Figura 16	<i>Exact XNOR Adder</i> (EXA)	38
Figura 17	<i>Approximate XOR-based Adder 1</i> (AXA1)	39
Figura 18	<i>Approximate XNOR-based Adder 2</i> (AXA2)	39
Figura 19	<i>Approximate XNOR-based Adder 3</i> (AXA3)	40
Figura 20	Corrente de curto-circuito	46
Figura 21	Previsão da evolução da potência dissipada do ITRS	46
Figura 22	Localização dos somadores aproximados no Array	48
Figura 23	Localização dos somadores aproximados no Baugh-Wooley	49
Figura 24	Localização dos somadores aproximados no Booth	49
Figura 25	Localização dos somadores aproximados no Vedic	50
Figura 26	Tempos de propagação	53
Figura 27	Fluxograma da aplicação de apoio	54
Figura 28	(a) Redução do número de transistores usando 30% de somadores aproximados (b) Redução do número de transistores usando 100% de somadores aproximados	58
Figura 29	(a) Tempos de propagação dos multiplicadores sem sinal em tensão nominal, (b) Tempos de propagação dos multiplicadores com sinal em tensão nominal	59

Figura 30	(a) Potência dos multiplicadores sem sinal em tensão nominal, (b) Potência dos multiplicadores com sinal em tensão nominal	60
Figura 31	PDP dos multiplicadores em tensão nominal	60
Figura 32	(a) Tempos de propagação dos multiplicadores sem sinal em <i>near-threshold</i> , (b) Tempos de propagação dos multiplicadores com sinal em <i>near-threshold</i>	61
Figura 33	(a) Potência dos multiplicadores sem sinal em <i>near-threshold</i> , (b) Potência dos multiplicadores com sinal em <i>near-threshold</i>	61
Figura 34	PDP dos multiplicadores em <i>near-threshold</i>	62
Figura 35	Tempo de propagação e potência máximos dos multiplicadores em <i>near-threshold</i> normalizados pelo nominal	62
Figura 36	(a) Tempos de propagação dos multiplicadores sem sinal com 100% de AXA2, (b) Tempos de propagação dos multiplicadores com sinal com 100% de AXA2	63
Figura 37	(a) Potência dos multiplicadores sem sinal com 100% de AXA2, (b) Potência dos multiplicadores com sinal com 100% de AXA2	64
Figura 38	PDP dos multiplicadores com 100% de AXA2	64
Figura 39	Tempo de propagação e potência máximos dos multiplicadores com 100% de AXA2 normalizados pelos exatos em tensão nominal	64
Figura 40	(a) ED total por bit dos multiplicadores sem sinal com 100% de AXA2, (b) ED total por bit dos multiplicadores com sinal com 100% de AXA2	65
Figura 41	(a) Tempos de propagação dos multiplicadores sem sinal com 30% de AXA2, (b) Tempos de propagação dos multiplicadores com sinal com 30% de AXA2	66
Figura 42	(a) Potência dos multiplicadores sem sinal com 30% de AXA2, (b) Potência dos multiplicadores com sinal com 30% de AXA2	67
Figura 43	PDP dos multiplicadores com 30% de AXA2	67
Figura 44	Tempo de propagação e potência máximos dos multiplicadores com 30% de AXA2 normalizados pelos exatos	68
Figura 45	(a) ED total por bit dos multiplicadores sem sinal com 30% de AXA2, (b) ED total por bit dos multiplicadores com sinal com 30% de AXA2	69
Figura 46	(a) Tempos de propagação dos multiplicadores sem sinal com 30% de AXA2 em NT, (b) Tempos de propagação dos multiplicadores com sinal com 30% de AXA2 em NT	70
Figura 47	(a) Potência dos multiplicadores sem sinal com 30% de AXA2 em NT, (b) Potência dos multiplicadores com sinal com 30% de AXA2 em NT	70
Figura 48	PDP dos dos multiplicadores com 30% de AXA2 em NT	71
Figura 49	Atraso e potência máximos dos multiplicadores com 30% de AXA2 em NT normalizados pelos exatos em tensão nominal	71
Figura 50	Atraso e potência máximos dos multiplicadores com 30% de AXA2 em NT normalizados por 30% AXA2	71
Figura 51	(a) Tempos de propagação dos multiplicadores sem sinal com 100% de AMA2, (b) Tempos de propagação dos multiplicadores com sinal com 100% de AMA2	72
Figura 52	(a) Potência dos multiplicadores sem sinal com 100% de AMA2, (b) Potência dos multiplicadores com sinal com 100% de AMA2	73

Figura 53	(a) PDP dos multiplicadores com 100% de AMA2	73
Figura 54	Atraso e potência máximos dos multiplicadores com 100% de AMA2 normalizados pelos exatos	74
Figura 55	(a) ED total por bit dos multiplicadores sem sinal com 100% de AMA2, (b) ED total por bit dos multiplicadores com sinal com 100% de AMA2	74
Figura 56	(a) Tempos de propagação dos multiplicadores sem sinal com 30% de AMA2, (b) Tempos de propagação dos multiplicadores com sinal com 30% de AMA2	75
Figura 57	(a) Potência dos multiplicadores sem sinal com 30% de AMA2, (b) Potência dos multiplicadores com sinal com 30% de AMA2	76
Figura 58	PDP dos multiplicadores com 30% de AMA2	76
Figura 59	Atraso e potência máximos dos multiplicadores com 30% de AMA2 normalizados pelos exatos	76
Figura 60	(a) ED total por bit dos multiplicadores sem sinal com 30% de AMA2, (b) ED total por bit dos multiplicadores com sinal com 30% de AMA2	77
Figura 61	(a) Tempos de propagação dos multiplicadores sem sinal com 30% de AMA2 em NT, (b) Tempos de propagação dos multiplicadores com sinal com 30% de AMA2 em NT	78
Figura 62	(a) Potência dos multiplicadores sem sinal com 30% de AMA2 em NT, (b) Potência dos multiplicadores com sinal com 30% de AMA2 em NT	79
Figura 63	PDP dos multiplicadores com 30% de AMA2 em NT	79
Figura 64	Atraso e potência máximos dos multiplicadores com 30% AMA2 em NT normalizados pelos exatos em tensão nominal	79
Figura 65	Atraso e potência máximos dos multiplicadores com 30% AMA2 em NT normalizados por 30% AMA2	80
Figura 66	Diagrama de Venn dos cenários de simulação	81

LISTA DE TABELAS

Tabela 1	Tabela verdade do Meio-somador	20
Tabela 2	Tabela verdade do Somador-completo	20
Tabela 3	Comparação entre as arquiteturas de multiplicadores	33
Tabela 4	Tabela verdade e distância de erro dos AMA's	37
Tabela 5	Contagem de transistores e de erros para cada um dos AMA's	38
Tabela 6	Tabela verdade e distância de erro dos AXA's	39
Tabela 7	Contagem de transistores e de erros para cada um dos AXA's	40
Tabela 8	Características elétricas dos somadores	41
Tabela 9	Comparação da proposta com trabalhos relacionados	43
Tabela 10	Exemplo de arco de atraso	53
Tabela 11	Número de transistores dos multiplicadores em cada cenário de simulação	58
Tabela 12	Distância de erro total, média e desvio padrão para os multiplicadores com 100% de somadores AXA2	66
Tabela 13	Distância de erro total, média e desvio padrão para os multiplicadores com 30% de somadores AXA2	69
Tabela 14	Distância de erro total, média e desvio padrão para os multiplicadores com 100% de somadores AMA2	74
Tabela 15	Distância de erro total, média e desvio padrão para os multiplicadores com 30% de somadores AMA2	77

LISTA DE ABREVIATURAS E SIGLAS

AMA	<i>Aproximate Mirror Adder</i>
AND	Célula lógica que representa a função booleana $A.B$
AXA	<i>Aproximate XOR/XNOR-based Adder</i>
CAS	<i>Controlled add/subtract</i>
Cin	<i>Carry In</i>
CLA	<i>Carry-Lookahead Adder</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
Cout	<i>Carry Out</i>
DRUM	<i>Dynamic Range Unbiased Multiplier</i>
DSP	<i>Digital Signal Processing</i>
ED	<i>Error distance</i>
FA	<i>Full Adder</i>
FinFET	<i>Fin-Shaped Field Effect Transistor</i>
FPGA	<i>Field Programmable Gate Array</i>
HA	<i>Half Adder</i>
HP	<i>High-performance</i>
L	Comprimento do canal do transistor
LSB	<i>Least Significant Bit</i>
LOA	<i>Lower-part-OR Adder</i>
MSB	<i>Most Significant Bit</i>
NAND	Célula lógica que representa a função booleana $\overline{A.B}$
NMOS	<i>N-channel Metal Oxide Semiconductor</i>
NOR	Célula lógica que representa a função booleana $\overline{A + B}$
OR	Célula lógica que representa a função booleana $A + B$
PDP	<i>Power-delay product</i>
PMOS	<i>P-channel Metal Oxide Semiconductor</i>

PTL	<i>Pass-Transistor Logic</i>
PTM	<i>Predictive Technology Model</i>
RCA	<i>Ripple-Carry Adder</i>
SMA	<i>Simplified Mirror Adder</i>
SoC	<i>System on Chip</i>
SPICE	<i>Simulation Program with Integrated Circuits Emphasis</i>
TED	<i>Total Error Distance</i>
tpHL	Tempo de propagação <i>high-to-low</i>
tpLH	Tempo de propagação <i>low-to-high</i>
VHDL	<i>Very High Description Language</i>
W	Largura do canal do transistor
W _n	Largura do canal do transistor NMOS
W _p	Largura do canal do transistor PMOS
XOR	Célula lógica que representa a função booleana $A \oplus B$
XNOR	Célula lógica que representa a função booleana $\overline{A \oplus B}$

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Objetivo	17
1.2	Organização do Trabalho	18
2	SOMADORES	19
2.1	Meio-Somador (<i>Half-Adder</i>)	19
2.2	Somador-Completo (<i>Full-Adder</i>)	19
2.3	<i>Ripple-Carry Adder</i>	21
2.4	<i>Carry-Lookahead Adder</i>	22
3	MULTIPLICADORES	24
3.1	Array	25
3.2	Vedic	25
3.3	Baugh-Wooley	28
3.4	Booth	30
4	COMPUTAÇÃO APROXIMADA APLICADA EM SOMADORES E MULTIPLICADORES	34
4.1	AMA (<i>Approximate Mirror Adder</i>)	35
4.2	AXA (<i>Approximate XOR/XNOR-based Adder</i>)	38
4.3	Análise dos somadores aproximados	40
4.4	Trabalhos relacionados com aproximação em multiplicadores	41
5	METODOLOGIA	44
5.1	Técnicas de redução energética	44
5.2	Cenários de teste	47
5.3	Análise das características elétricas e da precisão	51
6	RESULTADOS	57
6.1	Nominal	59
6.2	<i>Near-Threshold</i>	61
6.3	100% de AXA2	63
6.4	30% AXA2	66
6.5	30% AXA2 em NT	69
6.6	100% AMA2	72
6.7	30% AMA2	75
6.8	30% AMA2 em NT	78
6.9	Avaliação Geral	80

7 CONCLUSÃO	82
7.1 Trabalhos Futuros	84
7.2 Publicações	84
REFERÊNCIAS	86
APÊNDICE A REVISÃO BÁSICA DE LÓGICA COMBINACIONAL	92
A.1 Portas Lógicas	92
APÊNDICE B DESCRIÇÕES SPICE	100

1 INTRODUÇÃO

O crescimento no uso de dispositivos de computação pessoal (computadores portáteis e aplicativos baseados em áudio e vídeo em tempo real) e sistemas de computação sem fio, fizeram a dissipação de potência tornar-se um requisito crítico no desenvolvimento de sistemas digitais. Neste contexto, circuitos multiplicadores fazem-se necessários, principalmente no processamento digital de sinais (*Digital Signal Processor - DSP*), no processamento de imagens e em unidades aritméticas de microprocessadores (GOEL; GARG, 2011).

Um multiplicador rápido e de baixo consumo energético é frequentemente necessário na construção de sistemas digitais. A multiplicação é significativamente afetada pelo consumo energético. Geralmente, uma maior dissipação de potência implica em uma operação em alta temperatura, que tende a gerar falhas no sistema (GOEL; GARG, 2011).

O avanço das redes sem fio trouxe a necessidade de *DSP Systems on Chip* (SoCs) de baixa potência e alta velocidade. O multiplicador é um circuito de grande importância para estes componentes e possui uma influência bastante significativa nas características de desempenho e potência do sistema onde está inserido. Sendo assim, a escolha de uma determinada topologia de multiplicador pode trazer ganho ou perda de desempenho ao DSP (KHATIBZADEH; RAAHEMIFAR; AHAMDI, 2005).

Há muitas décadas vem-se desenvolvendo circuitos com o objetivo de realizar a operação aritmética de multiplicação. Em um primeiro momento, essa operação era realizada utilizando iterativamente um circuito somador. Porém, com o passar dos anos, surgiram algumas restrições em relação a taxa de *clock*, criando a necessidade de obter um hardware dedicado. Neste contexto, começaram a ser propostas implementações de multiplicadores matriciais (SJALANDER; LARSSON-EDEFORS, 2008a).

O funcionamento de todas as topologias de multiplicadores propostas na literatura pode ser dividido em 3 estágios: geração de produtos parciais, adição de produtos parciais e um estágio final de adição. A velocidade da operação de multiplicação pode ser aumentada diminuindo-se o número de produtos parciais, pois quanto menor for o número de produtos parciais menor será a quantidade de somas necessárias para obter-se o produto final, por isso muitos algoritmos e modificações foram propostos para acelerar este pro-

cedimento (MANJUNATH et al., 2015). O levantamento bibliográfico evidenciou alguns trabalhos que apresentam comparações de diferentes topologias de multiplicação. Pode-se destacar os seguintes trabalhos: (ABRAHAM; KAUR; SINGH, 2015), (SWEE; HIUNG, 2012), (SABEETHA et al., 2015). O primeiro apresenta um estudo sobre as particularidades das técnicas Booth Modificado, Vedic, Wallace e Dadda. O segundo apresenta um comparação entre as topologias Array, Wallace, Dadda, Reduced-Area e Booth Modificado. O último, por sua vez, exhibe os multiplicadores Bit Array, Carry-Save, Baugh-Wooley e Wallace Tree.

Na pesquisa, foi possível constatar que, na grande maioria dos trabalhos que abordam circuitos multiplicadores, é feita a descrição em nível arquitetural ou comportamental, adotando ferramentas de descrição em alto nível e linguagens como *Very High Description Language* (VHDL) e Verilog e, em maioria, com resultados de síntese para *Field Programmable Gate Array* (FPGA). O número de trabalhos onde utiliza-se simulações a nível elétrico considerando tecnologias nanométricas se mostrou significativamente menor. Isso pode estar relacionado com a complexidade existente em implementar grandes circuitos lidando apenas com transistores em simulações elétricas. O cenário citado serve de motivação para a realização deste trabalho tendo como foco as simulações a nível elétrico para uma caracterização dos tempos de atrasos e do consumo de energia.

1.1 Objetivo

O objetivo deste trabalho é apresentar os aspectos de projeto de circuitos multiplicadores, buscando utilizar de técnicas para aumentar a eficiência energética destes. Este trabalho explora duas técnicas para redução energética através da redução da tensão de operação (operando em *near-threshold*) e através da aproximação em nível lógico com a adoção de somadores aproximados. No presente trabalho são avaliadas as seguintes topologias de multiplicadores: Array, Baugh-Wooley, Booth e Vedic.

São objetivos parciais deste trabalho:

- Avaliar as características de atraso de propagação e consumo de energia de um conjunto de circuitos multiplicadores;
- Investigar os efeitos da redução de tensão nas características elétricas destes multiplicadores;
- Explorar a aproximação em nível lógico com a adoção de dois tipos de somadores aproximados;
- Relatar o impacto na precisão numérica dos multiplicadores ao adotar aproximação em todos os somadores internos dos circuitos multiplicadores;

- Avaliar uma alternativa de aproximação parcial nos somadores dos multiplicadores buscando melhor relação entre redução energética e precisão;
- Combinar as técnicas aproximação e redução de tensão para maior eficiência energética dos multiplicadores considerando os impactos em tempos de propagação e precisão;
- Prover uma discussão sobre os diferentes cenários de projeto dos circuitos multiplicadores.

1.2 Organização do Trabalho

Este trabalho está dividido em sete capítulos. O Capítulo 1 contém esta introdução, o Capítulo 2 apresenta circuitos somadores exatos, enquanto o Capítulo 3 apresenta os algoritmos de multiplicação e detalhes sobre o circuito dedicado a cada um deles. O Capítulo 4 apresenta o conceito de computação aproximada e implementações de circuitos que seguem este paradigma, já o Capítulo 5 apresenta os detalhes dos experimentos que foram realizados e as métricas usadas para a avaliação dos circuitos. O Capítulo 6 apresenta os resultados obtidos nos experimentos e, por fim, o Capítulo 7 traz as considerações finais.

2 SOMADORES

Existem na literatura diferentes circuitos digitais. Este capítulo apresenta as principais características de um somador, ou seja, de circuitos propostos e projetados com a funcionalidade de realização de somas. Estes componentes são muito importantes no projetos de multiplicadores e outros circuitos aritméticos. Nos multiplicadores, os somadores tem a tarefa de realizar a adição dos produtos parciais para que se obtenha o produto final da multiplicação. Ao considerar somadores, inicia-se as considerações de projeto observando o meio-somador, para expandir ao somador completo e, estes componentes serão utilizados no projeto de somadores de n -bits e outros circuitos aritméticos. Uma revisão de conceitos sobre portas lógicas básicas é disponibilizada no Apêndice A.

2.1 Meio-Somador (*Half-Adder*)

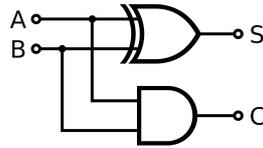
O meio-somador é um circuito que realiza a soma de dois dígitos binários. A Figura 1 mostra a implementação do circuito enquanto a Tabela 1 representa a tabela verdade do meio-somador. O meio-somador tem como entrada dois dígitos binários, A e B , e produz dois dígitos em suas saídas, um de soma e um de propagação do *carry* (C_{out}). Simplificando a tabela verdade, podemos representar o circuito meio-somador com a saída da soma sendo produzida por uma porta XOR entre os dígitos A e B , enquanto o *carry* é produzido por uma função AND dos mesmos dígitos.

Este circuito é simples e aplicado nos primeiros estágios de multiplicadores e somadores de n -bits, ou seja, para minimizar o número de componentes necessários para o cálculo do bit menos significativo em uma operação. Nestes casos, é conhecido que não haverá a situação de *carry-in* para a operação. Para os demais bits em operações aritméticas de n -bits, é necessário considerar a possibilidade da propagação de *carry*, adotando-se assim o somador-completo.

2.2 Somador-Completo (*Full-Adder*)

O somador-completo trata-se de um circuito que possui três sinais de entrada, A , B e *Carry In* (C_{in}), e gera como saída a *Soma* e o *Carry Out* (C_{out}). A Tabela 2 apresenta

Figura 1: Meio-somador



Fonte: Adaptado de (FLOYD, 2007)

Tabela 1: Tabela verdade do Meio-somador

A	B	Soma	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Fonte: O autor

todas as combinações possíveis de entrada e saída para um somador-completo. A soma de dois bits pode ser obtida combinando os sinais A e B através de uma porta lógica XOR, para adicionar o *Carry In* insere-se uma nova XOR entre ele e o resultado obtido anteriormente, como mostra a Equação 1. O *Carry Out* é obtido pela Equação 2, através de uma porta OR que tem como entradas as saídas de duas portas AND. A primeira das ANDs tem como entradas a saída da expressão $A \oplus B$ e o *Carry In*, enquanto a segunda recebe os bits A e B (PEDRONI, 2010).

Tabela 2: Tabela verdade do Somador-completo

A	B	C_{in}	Soma	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fonte: O autor

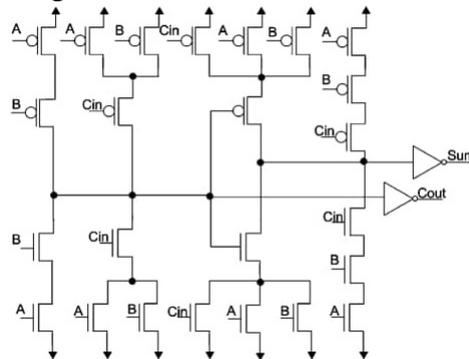
$$Soma = A \oplus B \oplus C_{in} \quad (1)$$

$$C_{out} = (A \oplus B).C_{in} + (A.B) \quad (2)$$

A nível de transistor existem diferentes topologias para descrever o somador-completo propostas com diferentes objetivos de otimização de atraso, ou de energia ou de qualidade dos sinais de saída, como pode-se ver em (NAVI et al., 2009). Neste trabalho, adotou-se o circuito apresentado na Figura 12, o *Mirror* (MA) CMOS. Este é composto por

28 transistores divididos em duas redes, *pull up* e *pull down*, sendo que estas redes se complementam logicamente. O somador *Mirror CMOS* tem como vantagem sua boa condutibilidade e robustez. Estas qualidades são muito importantes ao se trabalhar com tecnologias nanométricas de dimensões muito pequenas e que utilizam tensões baixas. A principal desvantagem fica sendo a sua alta capacitância de entrada, pelo fato de apresentar muitos transistores conectados.

Figura 2: Somador *Mirror CMOS*



Fonte: Adaptado de (NAVI et al., 2009)

O somador-completo permite a soma de dois bits e a implementação de somadores de $n - bits$, conectando as entrada de *carry-in* e *carry-out*. Para a implementação de somadores de $n - bits$, a estrutura mais simples é a conexão sequencial de n somadores completos conhecida como *Ripple-Carry Adder*.

Na literatura, também existem outras propostas de estruturas para acelerar a propagação de *carry*, como por exemplo o *Carry-Select Adder* (BEDRIJ, 1962) e o *Kogge-Stone Adder* (KOGGE; STONE, 1973). Neste trabalho, foram considerados para o projeto dos multiplicadores, os somadores *Ripple-Carry* e a *Carry-Lookahead*, detalhadas nas próximas seções.

2.3 *Ripple-Carry Adder*

O *Ripple-Carry Adder* (RCA) é um somador de n -bits onde o *carry* de saída de cada somador-completo é ligado à entrada de *carry* do somador no próximo estágio. A representação de um somador RCA de 4-bits é apresentada na Figura 3.

Nesta implementação, as saídas de um somador-completo não podem ser calculadas até que o *carry* de entrada seja recebido. Esta dependência da propagação do *carry* dos bits menos significativos provoca o atraso crítico deste circuito. O atraso para cada somador seria o período de tempo entre a colocação do *carry* de entrada e a definição do *carry* de saída, levando em conta que as entradas A e B já estejam definidas. O *carry* de entrada do estágio menos significativo é propagado por todos os somadores para que a soma seja totalmente realizada. O pior caso para o atraso seria um cenário onde todos os somadores

respectivamente.

$$G = A.B \quad (3)$$

$$P = A + B \quad (4)$$

Um *carry* de saída pode ser expresso em função de geração e propagação. Resumidamente, obtêm-se um *carry* de saída 1 se ele for gerado por um somador-completo ou se este somador propagar seu *carry* de entrada. A relação é expressa pela Equação 5.

$$C_{out} = G + P.C_{in} \quad (5)$$

Baseado nos conceitos apresentado, pode-se desenvolver as expressões para o bit de transporte (*carry*) de cada estágio de somador. Tomando como exemplo o CLA de quatro bits, obtêm-se as seguintes equações:

$$C_1 = G_0 + P_0.C_{in_0} \quad (6)$$

$$C_2 = G_1 + P_1.G_0 + P_1.P_0.C_{in_0} \quad (7)$$

$$C_3 = G_2 + P_2.G_1 + P_2.P_1.G_0 + P_2.P_1.P_0.C_{in_0} \quad (8)$$

$$C_4 = G_3 + P_3.G_2 + P_3.P_2.G_1 + P_3.P_2.P_1.G_0 + P_3.P_2.P_1.P_0.C_{in_0} \quad (9)$$

Os somadores mostrados neste capítulo são utilizados na implementação dos circuitos multiplicadores que serão mostrados no Capítulo 3. Os somadores de n -bits citados possuem características diferentes no que se refere a potência e tempo de resposta. O *Ripple-Carry Adder* (RCA) é um circuito mais simples que o CLA e por isso apresenta menor consumo de energia, no entanto seu tempo de resposta é muito elevado, dado que o resultado final da soma e o *Cout* só estarão disponíveis após a propagação dos *carrys* intermediários através de todos os somadores completos. Por sua vez, o *Carry-Lookahead Adder* (CLA) é um circuito com menor tempo de resposta pois utiliza uma metodologia que permite antecipar o *carry* gerado em cada estágio de soma, todavia essa lógica adicional torna o circuito mais complexo que o RCA, consumindo assim mais energia.

3 MULTIPLICADORES

Atualmente, existem diversos algoritmos e circuitos que possuem a finalidade de realizar operações de multiplicação computacional.

Observando o modo como a multiplicação de binários sem sinal é efetuado por nós, usando lápis e papel, percebe-se alguns pontos importantes:

1. A multiplicação envolve a geração de produtos parciais, um para cada dígito no multiplicador. Os produtos parciais são somados para gerar o produto final.
2. Os produtos parciais são definidos de maneira simples. Quando o bit multiplicador é 0, o produto é zero. Quando o multiplicador é 1, o produto é o multiplicando.
3. O produto final é obtido através da adição dos produtos parciais. Para a realização desta operação cada produto parcial é deslocado uma posição a esquerda em relação ao produto anterior.
4. A multiplicação de dois números binários de n -bits resulta em um produto de no máximo $2n$ bits de extensão.

A multiplicação de dois inteiros sem sinal (não negativos) é um problema um tanto que mais simples do que realizar a operação com números em complemento de dois, e embora reduza o espaço de números da solução, o ajuste final de sinais pode ser realizado como uma etapa simples adicional ao processo de multiplicação.

No que se refere a multiplicação de números em complemento de dois, o método descrito não funciona corretamente e pode gerar resultados incorretos (STALLINGS, 2010). A notação de complemento de dois é o modo mais utilizado para a representação de números inteiros com sinal nas ciências computacionais. Esta é muito usada porque seu uso descarta a necessidade de existir um circuito para examinar os sinais dos operandos e alterar o sinal do resultado de acordo com eles (MOHANTY, 2013). Para suprir a demanda das operações com números em complemento foram desenvolvidos novos algoritmos.

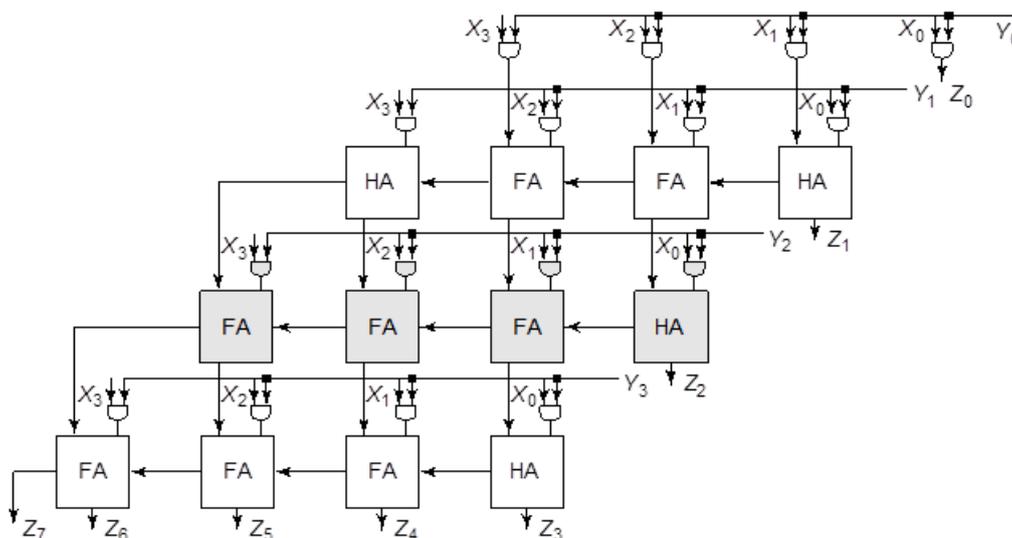
Neste capítulo são mostradas algumas das técnicas para realizar multiplicações de números sem e com sinal. Além disto é explicada a implementação em *hardware* dos

principais multiplicadores encontrados na literatura. As arquiteturas apresentadas neste trabalho destinadas a multiplicação sem sinal são o Array e o Vedic, enquanto o Baugh-Wooley e o Booth são propostos para as operações em complemento de dois.

3.1 Array

O multiplicador Array é uma arquitetura baseada no princípio de soma e deslocamento. A Figura 5 mostra o circuito para um multiplicador Array para 4 bits. Neste circuito, os produtos parciais são gerados usando portas AND e a soma dos produtos é feita usando meio somadores e somadores completos. Em um multiplicador de n -bits, são necessárias $n \times n$ portas AND para a geração dos produtos, $n \times (n - 2)$ FA's e n HA's. O pior atraso deste circuito está associado a sua largura, ou seja, quanto maior o número de bits da entrada, maior será o atraso no pior caso (SABEETHA et al., 2015).

Figura 5: Multiplicador Array de 4 bits



Fonte: (J. M. RABAEY; NIKOLIC, 2003)

3.2 Vedic

O multiplicador Vedic é proposto para multiplicação de números sem sinal e baseado na matemática védica. A matemática védica (SWAMI; MAHARAJA, 1986) é um antigo conjunto de técnicas para realizar cálculos baseado em regras simples e princípios, com as quais qualquer problema matemático pode ser resolvido. O sistema é formado por 16 sutras ou aforismos, que são na verdade, sentenças verbais que descrevem maneiras naturais de solucionar uma gama de problemas matemáticos (CHIDGUPKAR; KARAD, 2004). No que diz respeito a multiplicação, existe na matemática védica um algoritmo chamado Urdhava Tiryakbhyam, ou simplesmente Vertical e Cruzado, que é aplicável a

qualquer tipo de multiplicação. Este algoritmo se baseia na ideia de que a geração de todos os produtos parciais da multiplicação pode ser feita paralelamente com a adição dos mesmos.

O circuito baseado no algoritmo citado tem uma estrutura muito regular, o que facilita a criação do seu leiaute. Esta implementação também tem como vantagem o aumento mais lento no atraso e na área, comparado a outros multiplicadores, a medida que se adapta a arquitetura para realizar operações com mais bits de entrada (NARCHI et al., 2011).

Nas Equações 10, 11 e 12 é mostrada a operação mais básica do algoritmo. Ela é realizada entre dois números de dois bits, A e B , onde $A = a1a0$ e $B = b1b0$. No primeiro passo, os bits menos significativos são multiplicados para gerar o LSB do produto (vertical). Na segunda, o LSB do multiplicador é multiplicado pelo segundo bit do multiplicando, este produto é somado ao produto entre o LSB do multiplicando e o segundo bit do multiplicador (cruzado). A soma citada gera o segundo bit do produto final e um *carry* para a etapa seguinte. O *carry* gerado na segunda etapa é adicionado ao produto entre os dois MSB's, sendo que o resultado e o *carry* desta adição, correspondem ao terceiro e quarto bits do produto final, respectivamente (SATISH; RAJU, 2013). O resultado final é composto por $c2s2s1s0$.

$$s0 = a0b0 \quad (10)$$

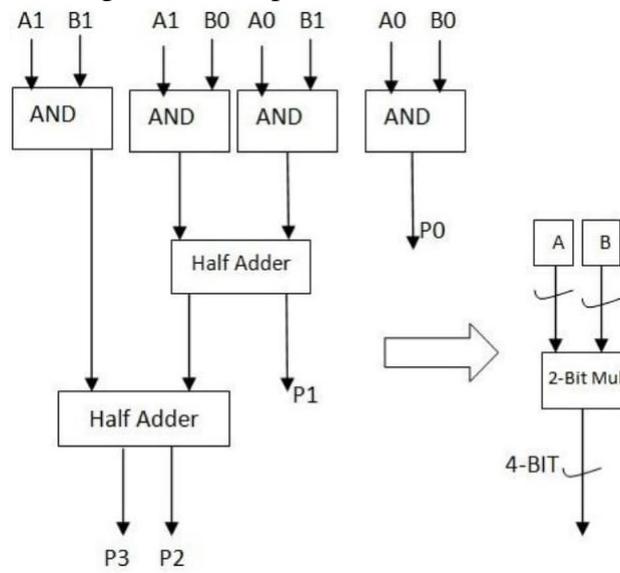
$$c1s1 = a1b0 + a0b1 \quad (11)$$

$$c2s2 = c1 + a1b1 \quad (12)$$

O módulo de multiplicação Vedic 2×2 , apresentado na Figura 6, é o bloco básico da técnica. Este bloco é implementado por quatro portas AND de duas entradas e dois meio somadores. Pode-se perceber que esse bloco é semelhante a um módulo projetado para a multiplicação de 2×2 bits no método Array convencional, seus maior atraso é igual ao atraso de dois meio somadores.

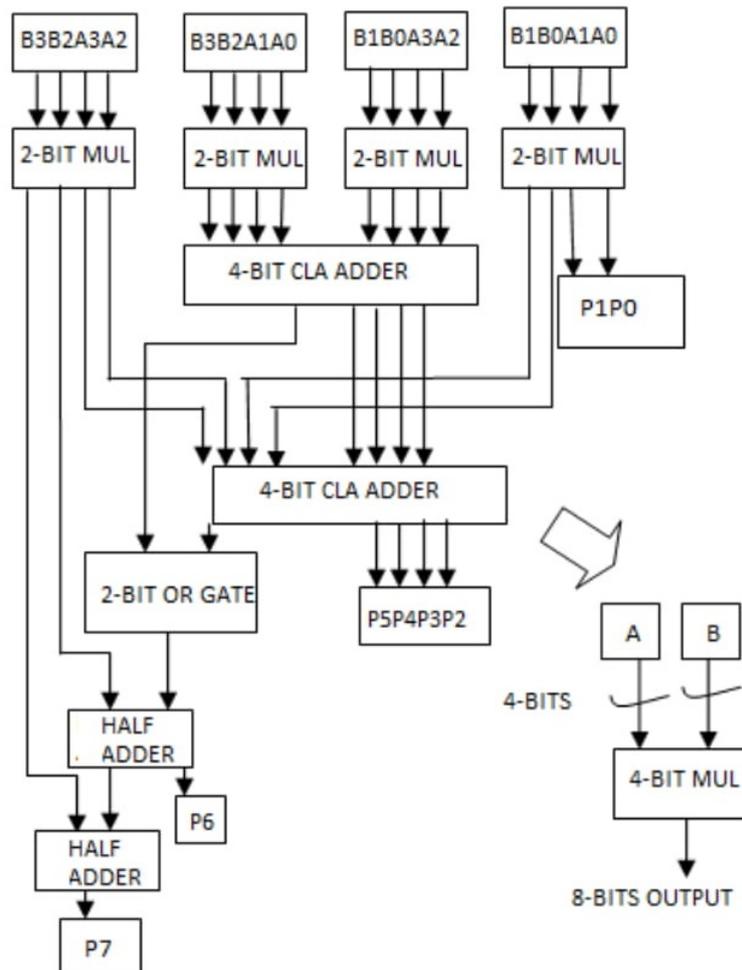
O bloco básico é usado para a construção de um multiplicador Vedic de 4 bits, sendo este último usado como componente para a implementação de um multiplicador de 8 bits. Resumidamente falando, para a construção de um Vedic de 2^n bits, se fazem necessários como componentes quatro multiplicadores de 2^{n-1} bits. A cada aumento nas dimensões do multiplicador também se faz necessário trocar os somadores de n -bits, internos do circuito, por outros com suporte ao maior número de entradas, como também adicionar meio somadores.(SATISH; RAJU, 2013). O multiplicador Vedic de 4 bits é exposto na Figura 7.

Figura 6: Multiplicador Vedic de 2 bits



Fonte: (K. BATHIJA et al., 2012)

Figura 7: Multiplicadores Vedic de 4 bits



Fonte: (K. BATHIJA et al., 2012)

3.3 Baugh-Wooley

O algoritmo de Baugh-Wooley proposto em (BAUGH; WOOLEY, 1973) maximiza a regularidade do multiplicador e permite que todos os produtos parciais tenham sinal positivo. Neste algoritmo, o AND de cada bit do multiplicador por cada bit do multiplicando produz os bits dos produtos parciais, que são adicionados através de uma árvore de somadores (BAUGH; WOOLEY, 1973).

A técnica de Baugh-Wooley foi desenvolvida para multiplicar números em complemento de dois. Quando dois números nessa codificação são multiplicados, cada um dos produtos parciais a ser adicionado é um número com sinal, este sinal precisa ser estendido até que se alcance a largura do produto final, com o objetivo de formar uma soma correta dos produtos através da árvore de somadores (MOHANTY, 2013).

A criação da matriz de produtos parciais para uma multiplicação de n -bits utilizando este algoritmo pode ser dividida em três etapas (SJALANDER; LARSSON-EDEFORS, 2008a):

1. O bit mais significativo das $n - 1$ primeiras linhas de produtos parciais e todos os bits da última linha de produtos, exceto o mais significativo, são invertidos.
2. O valor 1 é adicionado a coluna n .
3. O bit mais significativo do resultado final é invertido.

A Figura 8 mostra o procedimento citado tendo como exemplo uma multiplicação de 8x8 bits. As primeiras $n - 1$ linhas correspondem aos produtos parciais com magnitude e são gerados por uma porta NAND e $n - 1$ portas AND. A linha n é chamada de produto parcial com bit de sinal e é gerada por uma AND e $n - 1$ portas NAND (MOHANTY, 2013).

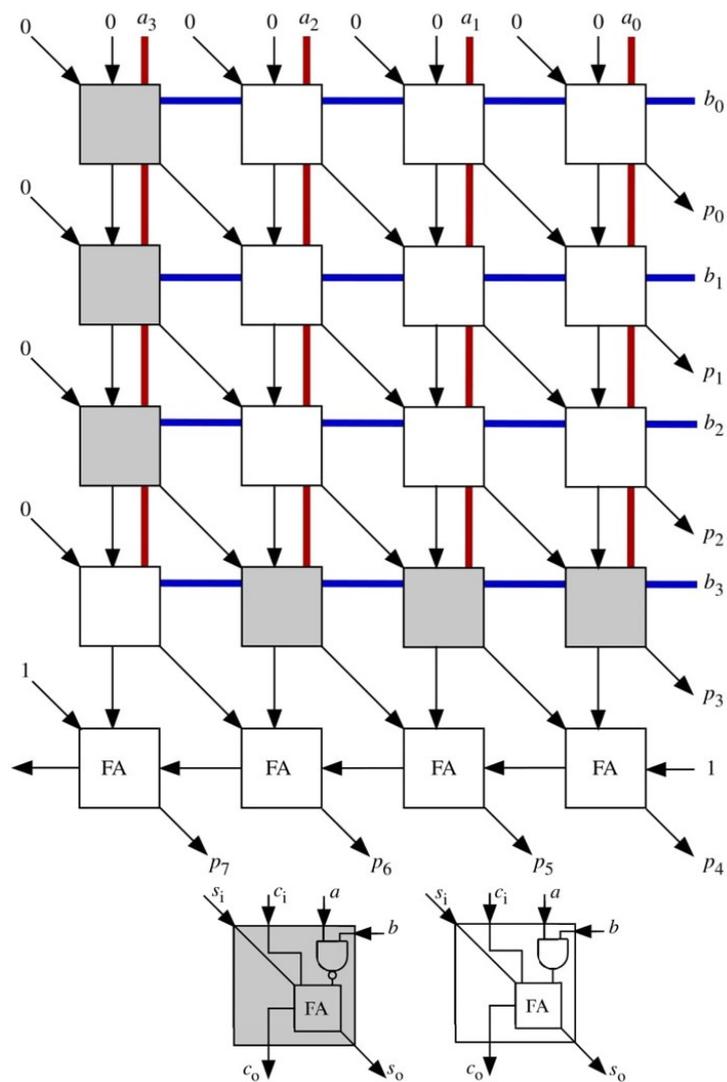
Figura 8: Ilustração de uma multiplicação Baugh-Wooley de 8 bits

	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀	
	X ₇	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	
	1	$\overline{p_{70}}$	p ₆₀	p ₅₀	p ₄₀	p ₃₀	p ₂₀	p ₁₀	p ₀₀
	$\overline{p_{71}}$	p ₆₁	p ₅₁	p ₄₁	p ₃₁	p ₂₁	p ₁₁	p ₀₁	
	$\overline{p_{72}}$	p ₆₂	p ₅₂	p ₄₂	p ₃₂	p ₂₂	p ₁₂	p ₀₂	
	$\overline{p_{73}}$	p ₆₃	p ₅₃	p ₄₃	p ₃₃	p ₂₃	p ₁₃	p ₀₃	
	$\overline{p_{74}}$	p ₆₄	p ₅₄	p ₄₄	p ₃₄	p ₂₄	p ₁₄	p ₀₄	
	$\overline{p_{75}}$	p ₆₅	p ₅₅	p ₄₅	p ₃₅	p ₂₅	p ₁₅	p ₀₅	
	$\overline{p_{76}}$	p ₆₆	p ₅₆	p ₄₆	p ₃₆	p ₂₆	p ₁₆	p ₀₆	
	$\overline{p_{77}}$	$\overline{p_{67}}$	$\overline{p_{57}}$	$\overline{p_{47}}$	$\overline{p_{37}}$	$\overline{p_{27}}$	$\overline{p_{17}}$	$\overline{p_{07}}$	
	$\overline{s_{15}}$	s ₁₄	s ₁₃	s ₁₂	s ₁₁	s ₁₀	s ₉	s ₈	s ₇
		s ₆	s ₅	s ₄	s ₃	s ₂	s ₁	s ₀	

Fonte: (SJALANDER; LARSSON-EDEFORS, 2008a)

A arquitetura baseada no algoritmo de Baugh-Wooley mostrada na Figura 9 é composta por dois tipos de blocos: os blocos cinzas, compostos por um somador e uma porta

Figura 9: Multiplicador Baugh-Wooley de 4 bits



Fonte: (RAJMOHAN; MAHESWRI, 2016)

NAND de duas entradas ligada a uma de suas entradas, e o bloco branco que é organizado da mesma forma, porém a porta utilizada é a AND. Os blocos cinzas representam a parte do multiplicador que dá suporte a operações com números com sinal representados em complemento de 2, enquanto os brancos realizam a multiplicação binária convencional. Cada bloco do circuito recebe quatro entradas, um bit do multiplicador (horizontal-azul), um bit do multiplicando (vertical-vermelho), o *carry* (vertical-preto) e a soma (diagonal-preto) proveniente dos blocos anteriores. Cada bloco produz duas saídas, a saída da soma (diagonal-preto) e a saída do *carry* (vertical-preto) (RAJMOHAN; MAHESWRI, 2016).

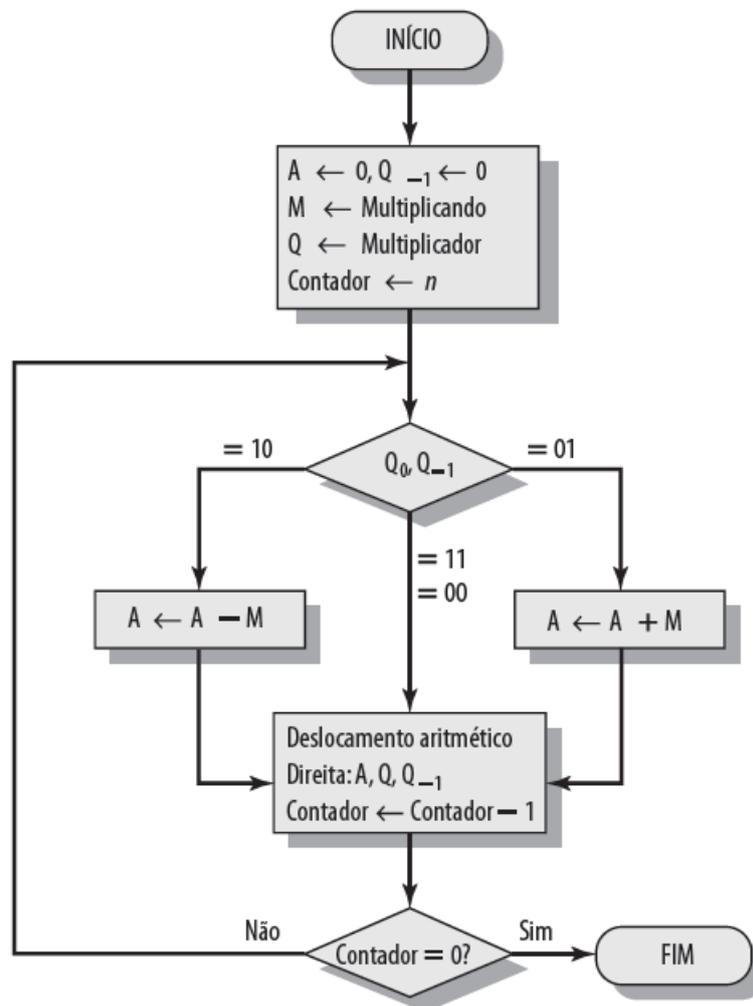
3.4 Booth

O algoritmo de Booth (BOOTH, 1951) é um algoritmo de multiplicação que pode operar dois números binários com sinal em uma notação de complemento de dois. Este algoritmo tem a capacidade de realizar menos adições e subtrações em comparação a multiplicação binária tradicional, sendo um processo de codificação que pode ser usado para minimizar o número de produtos parciais em uma operação de multiplicação (SAHOO; SAMANT, 2013). No método de multiplicação de Booth o termo multiplicador é analisado em pares de bits consecutivos iniciando pelo menos significativo, sendo cada par codificado em um dos dígitos $\{-1, 0, 1\}$. Cada dígito resultante da codificação é subsequentemente multiplicado pelo termo multiplicando, gerando assim um produto parcial da multiplicação (SJALANDER; LARSSON-EDEFORS, 2008b). As etapas do algoritmo apresentado são mostradas a seguir.

O algoritmo de Booth é representado na Figura 10 e pode ser descrito da seguinte forma. Primeiramente, o multiplicador e o multiplicando são colocados nos registradores Q e M, respectivamente. Há também um registrador de 1 bit colocado logicamente à direita do bit menos significativo (Q_0) do registrador Q e chamado Q_{-1} . Os resultados da multiplicação aparecerão nos registradores A e Q. A e Q_{-1} são inicializados em 0. A lógica de controle verifica os bits do multiplicador um de cada vez, sendo que à medida que cada bit é examinado, o bit à sua direita também é observado. Se os dois bits forem iguais (1–1 ou 0–0), então todos os bits dos registradores A, Q e Q_{-1} são deslocados à direita por 1 bit. Se os dois bits forem diferentes, então o multiplicando é somado ou subtraído do registrador A, dependendo se os dois bits forem 0–1 ou 1–0. Após a adição ou subtração, ocorre o deslocamento à direita. De qualquer forma, o deslocamento à direita é tal que o bit mais à esquerda de A (A_{n-1}) não apenas é deslocado para A_{n-2} , mas também permanece em A_{n-1} . Isso é exigido para preservar o sinal do número em A e Q. Esse é conhecido como um deslocamento aritmético, pois preserva o bit de sinal (STALLINGS, 2010).

Uma implementação de hardware para a multiplicação de números binários de 4 bits utilizando o algoritmo de Booth é mostrada em (HUANG; CAI; SHEN, 2015). O circuito,

Figura 10: Algoritmo de Booth para a multiplicação em complemento de dois

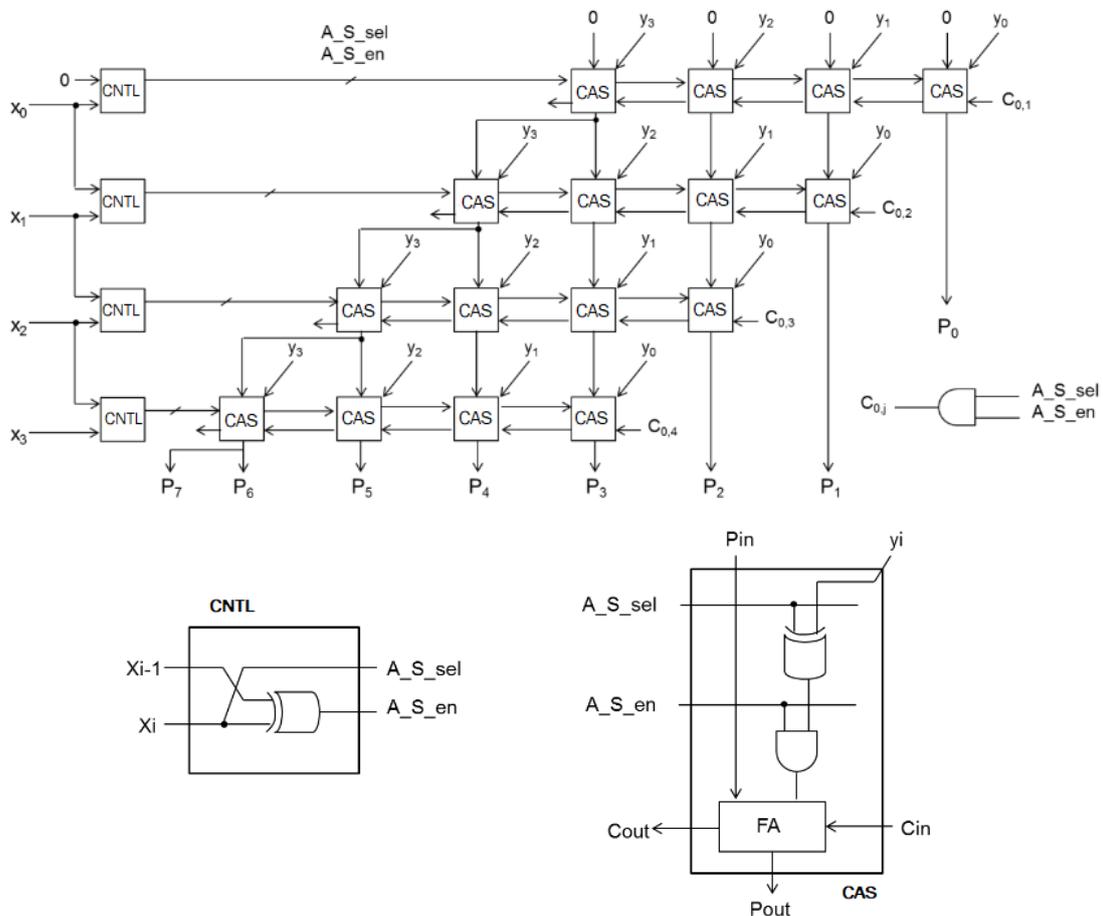


Fonte: (STALLINGS, 2010)

disposto na Figura 11, é composto por quatro linhas de células denominadas *Controlled add/ subtract* (CAS) conectadas em série, todas as células de uma dada linha estão ligadas a célula *Control* (CNTL) da respectiva linha. As células CAS em uma determinada linha i executam uma operação de adição, subtração ou apenas deslocamento aritmético do produto acumulado, dependendo dos sinais gerados pela célula de controle (CNTL) da linha correspondente.

A célula CNTL é composta por uma porta lógica XOR, esta célula é responsável por gerar os sinais *selection* e *enable* a partir de bit x_i e x_{i-1} do termo multiplicador. Se o *enable* for 0 não é realizada operação aritmética, se for 1 é realizada a operação definida pelo *selection*, que pode ser adição caso seja 0, ou subtração caso seja 1. A célula CAS é composta por uma porta lógica XOR, uma AND e um somador completo. As células CAS recebem os dois sinais de controle, um bit do produto parcial anterior e um do multiplicando, além do *carry* gerado pelo somador da célula a direita. O *carry* de entrada da célula mais a direita de cada uma das linhas é gerado pela combinação do *enable* e do *selection* através de uma porta AND (KOREN, 2008).

Figura 11: Multiplicador Booth de 4 bits



Fonte: Adaptado de (HUANG; CAI; SHEN, 2015)

Muitas arquiteturas foram propostas para executar a operação aritmética de multiplicação, cada uma delas apresenta vantagens e desvantagens em relação a área ocupada, complexidade de implementação, regularidade de leiaute e tempo de resposta. As implementações citadas neste capítulo foram escolhidas com base na existência, no estado da arte, de diversas propostas de alterações que visam aumentar a performance destes circuitos aritméticos. A Tabela 3 apresenta uma comparação das principais características dos multiplicadores observados.

Tabela 3: Comparação entre as arquiteturas de multiplicadores

Multiplicador	Área	Complexidade	Leiaute	Tempo de resposta
Array	Pequena	Simples	Regular	Alto
Baugh-Wooley	Pequena	Simples	Regular	Alto
Booth	Média	Complexo	Irregular	Baixo
Vedic	Muito grande	Mais complexo	Mais irregular	Muito baixo

Fonte: Adaptado de (ALEXANDER, 2012)

4 COMPUTAÇÃO APROXIMADA APLICADA EM SOMADORES E MULTIPLICADORES

A eficiência energética vem tornando-se cada vez mais importante no desenvolvimento de sistemas digitais. Neste cenário, os sistemas computacionais tornam-se gradativamente mais integrados e cresce o número de tarefas de alta demanda computacional a serem executadas por dispositivos móveis. Estas tarefas estão relacionadas a diferentes aplicativos, como, por exemplo, processamento de mídia (áudio, vídeo, gráfico e imagens), reconhecimento e mineração de dados. Uma semelhança entre as tarefas citadas é que, em muitos dos casos, um resultado inferior ao ideal já possui qualidade suficiente para o usuário final, caracterizando estas aplicações como tolerantes a pequenas imprecisões. Alguns dos motivos da possibilidade de tolerar erros são as limitações da percepção humana, as redundâncias de dados e a habilidade de lidar com entradas ruidosas (HAN; ORSHANSKY, 2013).

A tolerância a erros pode ser explorada por meio da computação aproximada, ramo de pesquisa que busca encontrar maneiras de obter economia de recursos, como área e energia, em troca de redução da qualidade do resultado computado pelo sistema digital (PALEM; LINGAMNENI, 2013).

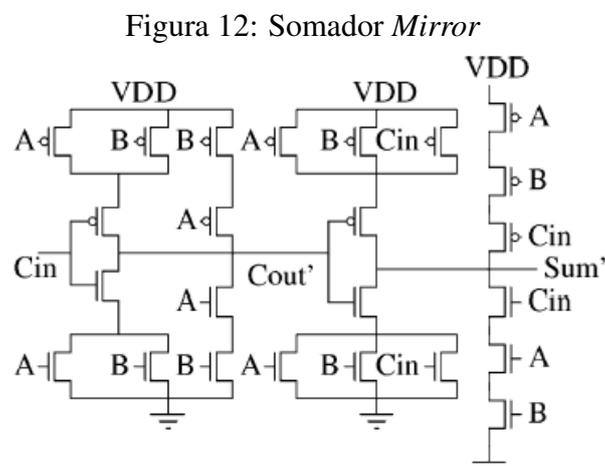
No estado da arte estão presentes algumas pesquisas sobre a aplicação de computação aproximada em somadores e multiplicadores. Entretanto, muitas delas propõem alterações em nível arquitetural ou descrição RTL. Poucos trabalhos abordam a computação aproximada em nível elétrico em somadores, e um número menor ainda de trabalhos é encontrado avaliando multiplicadores.

Uma maneira de aplicar aproximação em somadores é alterar a estrutura da célula somador de 1-bit (GUPTA et al., 2011). Em (YANG et al., 2013) são apresentados somadores aproximados baseados em portas XOR/XNOR e transistores de passagem, estes circuitos dispõem de número reduzido de transistores quando comparados ao somador exato *Mirror* CMOS. Dentre estes estudos, destacam-se os somadores AMA e AXA. Os *Approximate Mirror Adders* (AMAs) foram introduzidos a partir da redução lógica do somador *Mirror*, para isto foi feita a remoção de alguns transistores visando alcançar menor dissipação de potência e complexidade (GUPTA et al., 2011). Por sua vez, os *Approxi-*

mate XOR/XNOR-based adders (AXAs) propostos em (YANG et al., 2013), são baseados em um somador exato, composto por portas XOR/XNOR, implementado a partir de transistores de passagem. Embora o uso de transistores de passagem cause uma redução da margem de ruído, os AXAs são úteis quando uma menor precisão pode ser tolerada. Estes somadores aproximados serão melhor explicados nas Seções 4.1 e 4.2.

4.1 AMA (*Approximate Mirror Adder*)

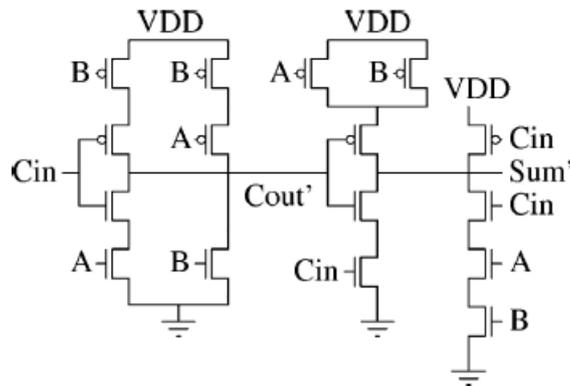
Os *Approximate Mirror Adders* são somadores aproximados obtidos através da redução da complexidade lógica, a nível de transistor, do somador *Mirror* (Figura 12). A redução do número de transistores gerou somadores imprecisos porém de baixo consumo energético. As simplificações do somador *Mirror* não só apresentam um número menor de transistores, como também uma redução nas capacitâncias dos nós internos desta unidade aritmética. Diminuir a complexidade lógica proporciona redução no consumo energético de duas maneiras: a primeira é proveniente da diminuição da área do hardware, enquanto a segunda é a redução dos caminhos críticos, devido a redução da complexidade, que acaba possibilitando redução da tensão sem que erros sejam induzidos pela temporização (GUPTA et al., 2011).



Fonte: (GUPTA et al., 2011)

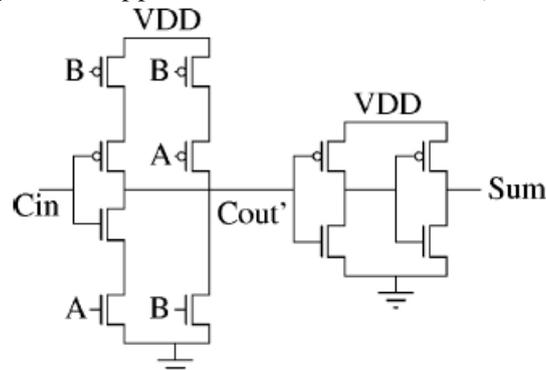
Para a simplificação do *Mirror* não se pode fazer remoções arbitrárias dos transistores, é necessário ter cuidado para que nenhuma combinação das entradas A , B e Cin , resulte em curto-circuito ou circuito aberto. Outro quesito neste processo é buscar introduzir o mínimo de erro na tabela verdade do somador-completo. Uma criteriosa seleção dos transistores a serem removidos, gerou o esquemático mostrada na Figura 13, denominado *Simplified Mirror Adder* (SMA). Esta aproximação possui oito transistores a menos que o MA convencional, porém apresenta um erro para saída $Cout$ e dois para a $Soma$ (GUPTA et al., 2011).

Uma análise da tabela verdade do somador-completo (Tabela 4) revelou que $Soma =$

Figura 13: *Simplified Mirror Adder (SMA)*

Fonte: (GUPTA et al., 2011)

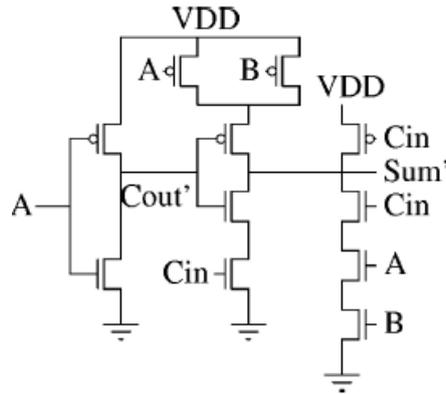
C_{out} em 75% das combinações de entrada. Sendo assim, um forma de simplificar o MA, é remover os transistores responsáveis pela *Soma* e igualar esta saída ao C_{out} . Apesar de poder-se igualar as duas saídas diretamente, foi introduzido um *buffer* entre os dois nós para evitar uma penalidade no atraso, em projetos onde dois ou mais destes somadores sejam conectados em série. A metodologia de igualar a *Soma* ao C_{out} foi aplicada no circuito SMA (Figura 13), gerando assim a aproximação 1 (AMA1), mostrada na Figura 14. O AMA1 apresenta um erro na saída C_{out} e três na *Soma*.

Figura 14: *Approximate Mirror Adder I (AMA1)*

Fonte: (GUPTA et al., 2011)

Observando novamente a Tabela 4, nota-se que $C_{out} = A$ em seis das oito configurações de entrada e que $C_{out} = B$ ocorre na mesma proporção. A partir disto criou-se a aproximação 2 (AMA2), onde é usado apenas um inversor com a entrada A para calcular o C_{out} , enquanto a *Soma* é calculada da mesma forma que no SMA (Figura 13). O circuito final é mostrado na Figura 15 e seu uso introduz dois erros em C_{out} e três na *Soma*.

Em projetos voltados a aproximação, deve-se utilizar alguma métrica para avaliar a precisão dos resultados aproximados em relação os exatos. Em (LIANG; HAN; LOMBARDI, 2013), foi proposta uma métrica denominada distância de erro (ED - *Distance*

Figura 15: *Approximate Mirror Adder 2 (AMA2)*

Fonte: (GUPTA et al., 2011)

Error) como figura de mérito para a computação inexata (YANG et al., 2013). Para uma determinada entrada, a distância de erro é definida como a distância aritmética entre uma saída inexata a e a saída correta b , como mostrado na Equação 13, onde i e j são os índices para os *bits* em a e b , respectivamente (YANG et al., 2013). Exemplificando, os valores inexatos “01” e “00” tem ED de “01” (1) e “10” (2), em relação ao valor exato “10”. Para avaliar a precisão do projeto aproximado, pode-se utilizar a distância de erro total, que trata-se do somatório das ED’s observadas para todas as possíveis combinações de entrada do circuito.

$$ED(a, b) = |a - b| = \left| \sum_i a[i] * 2^i - \sum_j b[j] * 2^j \right| \quad (13)$$

A redução da complexidade lógica do somador *Mirror*, em busca da diminuição do número de transistores e do consumo energético, gera circuitos com tabelas verdade diferentes do esperado, como mostrado na Tabela 4. Nesta tabela é evidenciada a distância de erro (ED) para cada uma dos vetores de entrada. Por sua vez, a Tabela 5 expõe o número de transistores e a contagem de falhas para cada aproximação do MA.

Tabela 4: Tabela verdade e distância de erro dos AMA’s

ENTRADAS A B Cin	EXATO		SMA			AMA1			AMA2		
	Cout	Soma	Cout	Soma	ED	Cout	Soma	ED	Cout	Soma	ED
0 0 0	0	0	0	0	0	0	1	1	0	0	0
0 0 1	0	1	0	1	0	0	1	0	0	1	0
0 1 0	0	1	1	0	1	1	0	1	0	0	1
0 1 1	1	0	1	0	0	1	0	0	0	1	1
1 0 0	0	1	0	0	1	0	1	0	1	0	1
1 0 1	1	0	1	0	0	1	0	0	1	0	0
1 1 0	1	0	1	0	0	1	0	0	1	0	0
1 1 1	1	1	1	1	0	1	0	1	1	1	0

Fonte: O autor

Tabela 5: Contagem de transistores e de erros para cada um dos AMA's

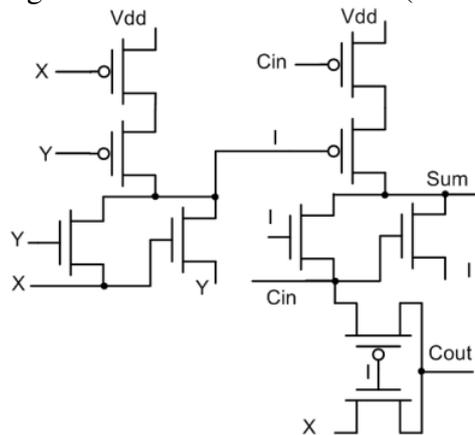
Somador	Número de transistores	Entrada com falha	Contagem de falhas
SMA	20	010, 100	2
AMA1	13	000, 010, 111	3
AMA2	15	010, 011, 100	3

Fonte: (SILVA; MEINHARDT, 2020)

4.2 AXA (*Approximate XOR/XNOR-based Adder*)

Os circuitos *Approximate XOR/XNOR-based Adders* são somadores aproximados que tem sua implementação fundamentada na modificação dos somadores exatos baseados em XOR/XNOR. O somador exato baseado em XOR foi proposto por (MAHMOUD; MAGDY, 1999), enquanto o baseado em XNOR foi apresentado em (LIN et al., 2007), ambos os circuitos tem sua implementação realizada com dez transistores. O somador apresentado em (LIN et al., 2007), fundamenta-se em portas XNOR de quatro transistores e é apresentado na Figura 16.

Figura 16: *Exact XNOR Adder (EXA)*



Fonte: (YANG et al., 2013)

Em (YANG et al., 2013) são apresentadas três implementações de somadores aproximados, embasados nos somadores exatos citados anteriormente, o AXA1 baseia-se no somador proposto em (MAHMOUD; MAGDY, 1999), enquanto os AXA2 e AXA3, no circuito apresentado em (LIN et al., 2007). A Tabela 6 apresenta uma compilação das tabelas verdade do somador exato e dos somadores AXA.

O primeiro somador aproximado (AXA1) é mostrado na Figura 17. Neste circuito as saídas são produzidas por dois inversores e dois blocos de transistores de passagem (*Pass-transistor Logic - PTL*) (ZIMMERMANN; FICHTNER, 1997). O arranjo apresentado é composto por um total de oito transistores e sua distância de erro é 4, conforme visto na Tabela 6 (YANG et al., 2013).

A Figura 18 apresenta o somador AXA2. Esta topologia é formada por uma porta XNOR de quatro transistores e um bloco de transistores de passagem, possuindo um total

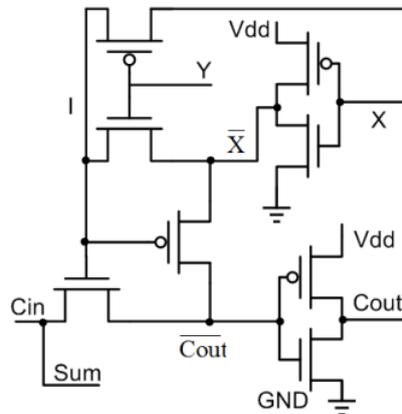
Tabela 6: Tabela verdade e distância de erro dos AXA's

Entrada A B Cin	Exato		AXA1			AXA2			AXA3		
	Cout	Soma	Cout	Soma	ED	Cout	Soma	ED	Cout	Soma	ED
0 0 0	0	0	0	0	0	0	1	1	0	0	0
0 0 1	0	1	0	1	0	0	1	0	0	1	0
0 1 0	0	1	1	0	1	0	0	1	0	0	1
0 1 1	1	0	0	1	1	1	0	0	1	0	0
1 0 0	0	1	1	0	1	0	0	1	0	0	1
1 0 1	1	0	0	1	1	1	0	0	1	0	0
1 1 0	1	0	1	0	0	1	1	1	1	0	0
1 1 1	1	1	1	1	0	1	1	0	1	1	0

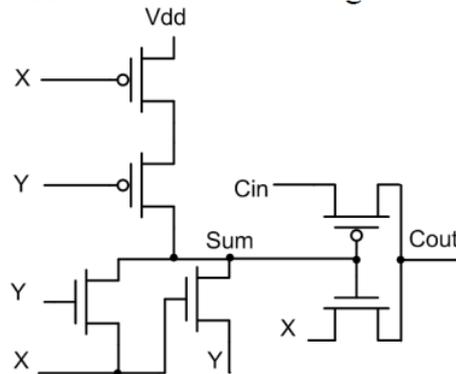
Fonte: (YANG et al., 2013)

de seis transistores. Referente as saídas, o valor da *Soma* é correto para 50% das operações possíveis, enquanto o *Cout* é exato para todas as possibilidades de entrada. A distância de erro para esta aproximação é 4.

Por fim, é mostrado na Figura 19 o somador AXA3. O circuito em questão é uma

Figura 17: *Approximate XOR-based Adder 1 (AXA1)*

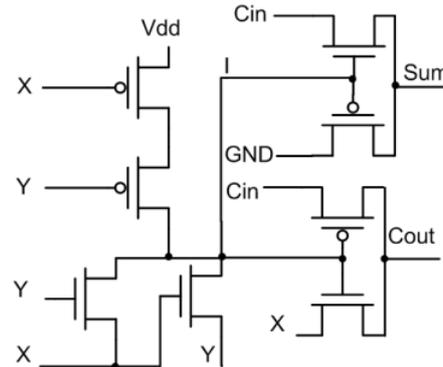
Fonte: (YANG et al., 2013)

Figura 18: *Approximate XNOR-based Adder 2 (AXA2)*

Fonte: (YANG et al., 2013)

modificação do AXA2, tendo por diferencial a introdução de um bloco de transistores de passagem, com o objetivo de aumentar a precisão da *Soma*. A implementação é dada por oito transistores no total, onde uma metade forma uma porta XNOR enquanto a outra compõe dois blocos PTL. A precisão da saída *Soma* é de 75%, enquanto a do *Cout* é de 100%. A distância de erro total apresentada por esse circuito é 2, como mostrado na Tabela 6.

Figura 19: *Approximate XNOR-based Adder 3 (AXA3)*



Fonte: (YANG et al., 2013)

A Tabela 7 mostra o número de transistores e os vetores de entrada que geram resultados imprecisos em cada um dos somadores AXA. Embora o circuito AXA2 apresentem contagem de falhas e distância de erro maiores que o circuito AXA3, o circuito AXA2 é o que possui o menor número de transistores entre as três versões. Resultados comparativos destes três somadores aproximados mostram que esta redução de dispositivos torna a versão AXA2 a com menor dissipação de energia (SILVA; MEINHARDT, 2020). Por este motivo, escolheu-se este circuito entre os AXAs para adoção nos multiplicadores.

Tabela 7: Contagem de transistores e de erros para cada um dos AXA's

Somador	Número de transistores	Entradas com falha	Contagem de falhas
AXA1	8	010, 011, 100, 101	4
AXA2	6	000, 010, 100, 110	4
AXA3	8	010, 100	2

Fonte: Adaptado de (YANG et al., 2013)

4.3 Análise dos somadores aproximados

Os somadores AMA avaliados em (SILVA; MEINHARDT, 2020), apresentaram redução significativa no consumo energético em comparação ao *Mirror*, o SMA oferece uma economia de 32,92 %, enquanto o AMA1 proporciona uma redução de 40,07%, por fim, o AMA2 chega a alcançar 74,47% de economia. No que se refere ao atraso crítico, as aproximações do *Mirror* também tiveram bons resultados, apresentando 22,05% (SMA),

29,70% (AMA1) e 26,49% (AMA2) de melhora em relação a sua versão exata. Tratando-se do PDP, o AMA2 apresentou uma diminuição de 81,23% quando comparado ao PDP do MA.

Os resultados mostrados em (SILVA; MEINHARDT, 2020) apontaram que o uso do somador aproximado AXA2 representa uma economia de 85,48% no consumo energético, quando comparado ao somador *Mirror* (MA) convencional, enquanto o AXA1 trouxe uma aumento de 103%. No que diz respeito ao atraso o AXA2 trouxe o melhor resultado, porém esse valor é 4,4 vezes o atraso máximo do MA. Em relação ao PDP, o AXA2 mostrou bons resultados, trazendo uma redução de 36,23% em relação ao somador *Mirror*.

A partir dos resultados observados foi decidido utilizar o somador de menor potência de cada grupo, ou seja, o AMA2 e o AXA2, como alternativas para reduzir o consumo energético de circuitos multiplicadores. Na Tabela 8 são mostrados os dados de caracterização elétrica dos somadores aproximados apresentados em (SILVA; MEINHARDT, 2020). A utilização dos somadores aproximados na composição de circuitos multiplicadores irá induzir erros nos bits de saída que compõem o resultado da multiplicação, sendo este erros toleráveis ou não dependendo do nível de precisão exigido pela aplicação.

Tabela 8: Características elétricas dos somadores

Somador	Atraso Máximo (ps)	Potência (nW)	PDP (aJ)
<i>Mirror</i>	44,35	485,51	21,53
SMA	34,57	325,68	11,26
AMA1	31,18	290,97	9,07
AMA2	32,60	123,95	4,04
EXA	324,41	262,61	85,14
AXA1	235,90	989,53	233,43
AXA2	194,87	70,48	13,73

Fonte: (SILVA; MEINHARDT, 2020)

4.4 Trabalhos relacionados com aproximação em multiplicadores

No que se refere a aproximação em somadores de n -bits pode-se citar o *Lower-part-OR Adder* (LOA) estudado em (MAHDIANI et al., 2010). Este somador é dividido em duas partes, uma opera os bits mais significativos e é composta por somadores de 1-bit exatos, e a outra lida com os menos significativos e é formada por portas OR. Tratando-se de multiplicadores, é possível aplicar computação aproximada modificando a arquitetura do multiplicador ou substituindo seus somadores exatos internos por aproximados.

Em (LIU; HAN; LOMBARDI, 2014), foi proposto um multiplicador que utiliza somadores aproximados e lógica baseada em portas OR para a recuperação de erros. Este circuito mostrou diminuição de 20% no atraso e 69% na potência, em relação ao multiplicador Wallace (WALLACE, 1964). Em (HASHEMI; BAHAR; REDA, 2015) foi apresen-

tada um estrutura denominada *Dynamic Range Unbiased Multiplier* (DRUM), esta tem como objetivo o descarte dos bits menos significativos das entradas antes da realização da multiplicação. Foi demonstrado que o componente proposto pode alcançar de 54% a 80% de redução energética.

No trabalho de (KIM; KIM, 2017) é apresentado uma versão aproximada do multiplicador Array (J. M. RABAEY; NIKOLIC, 2003), este circuito foi obtido a partir da introdução do somador aproximado VAXA, estudado em (KIM; KIM, 2016). O multiplicador proposto apresenta uma redução de 20% em área, atraso e consumo energético. A pesquisa de (REDDY et al., 2015) também propôs uma aproximação do Array, com a diferencial de que são utilizados somadores AMA (GUPTA et al., 2011). Esta pesquisa alcançou redução de 30% no produto entre atraso e potência, para multiplicadores de 4 bits, e até 64% para multiplicadores de 8 bits.

Em (GUO; SUN; KIMURA, 2018) é proposto um multiplicador aproximado que utiliza compressores baseados em OR. Esta proposta apresenta uma redução de 50,78% da potência e 53,19% da área, comparado ao multiplicador Wallace (WALLACE, 1964). O trabalho de (PANDEY et al., 2018) relata um multiplicador Dadda (DADDA, 1965) aproximado utilizando propostas de somadores PTL inexatos. Neste estudo foi obtida uma economia de 28% no número de transistores e de 32% no consumo energético.

Em (PRANAY; JANDHYALA, 2015) é apresentado um multiplicador de Booth Modificado (MACSORLEY, 1961) onde foram introduzidos somadores aproximados propostos em (LIU; HAN; LOMBARDI, 2014) e (KAHNG; KANG, 2012). O circuito oferece uma melhoria de 15,3% no atraso e de 5% no consumo. A pesquisa de (QIAN et al., 2016) expõe uma multiplicador Booth aproximado utilizando compressores 4-2 propostos em (MOMENI et al., 2015) e uma árvore de Wallace aproximada. Comparando ao Booth Modificado, o circuito proposto oferece até 37% de economia na energia para 8 bits, e 22% de redução em atraso para 16 bits.

A Tabela 9 apresenta uma comparação dos trabalhos citados, mostrando a técnica de aproximação, as arquiteturas de multiplicadores abordadas e o número de bits das entradas, além da tecnologia nanométrica utilizada para a implementação e simulação. Relacionando com os trabalhos apresentados, a proposta desta dissertação difere ao avaliar o efeito no consumo energético combinando duas estratégias: a aproximação dos somadores com a redução de tensão de operação a níveis próximos do limiar de *threshold*. Este trabalho também adota a substituição dos somadores exatos presentes nos circuitos multiplicadores por somadores aproximados do tipo AMA e AXA, e fornece uma avaliação de 4 topologias de multiplicadores. Apesar de limitar a avaliação a 4 bits, enquanto alguns trabalhos já apresentam dados para multiplicadores de até 32 bits, a metodologia proposta nesta dissertação foi a única encontrada até o momento a considerar tecnologias abaixo de 45nm e a adotar simulação elétrica no levantamento de dados de potência e atraso para os multiplicadores avaliados.

Tabela 9: Comparação da proposta com trabalhos relacionados

Trabalho	Técnica de aproximação	Multiplicador	# de bits	Tecnologia
(LIU; HAN; LOMBARDI, 2014)	Somador aproximado proposto	Proposto	8, 16	Processo STM 28nm CMOS
(HASHEMI; BAHAR; REDA, 2015)	descarte de bits menos significativos	Proposto	16, 24, 32	Compilador Synopsys DC e biblioteca de células de 65 nm
(KIM; KIM, 2017)	Somador aproximado (KIM; KIM, 2016)	Array	16	Design Compiler com biblioteca de células Nangate de 45 nm
(REDDY et al., 2015)	Somadores AMA	Array	4, 8, 16	Software EDA da tecnologia Cadence 180 nm
(GUO; SUN; KIMURA, 2018)	Compressor baseado em OR, HA aproximado	Proposto	8, 16	Synopsys Design Compiler com tecnologia SMIC 40nm
(PANDEY et al., 2018)	Somadores PTL aproximados	Dadda	8	Tecnologia 90 nm UMC
(PRANAY; JANDHYALA, 2015)	Somador aproximado propostos em (KAHNG; KANG, 2012) e (LIU; HAN; LOMBARDI, 2014)	Booth Modificado (Radix-4)	8	Cadence Tools e biblioteca de células CMOS padrão UMC 0,18um
(QIAN et al., 2016)	Compressores 4:2 (MOMENI et al., 2015), Wallace Tree aproximada	Booth Modificado (Radix-4)	8, 16, 32	Synopsys Design Compiler com biblioteca de células Nangate 45nm
Esta proposta	Somadores aproximados e redução de tensão	Array, Baugh-Wooley, Booth e Vedic	4	Descrição SPICE com 16nm PTM HP

Fonte: O autor

5 METODOLOGIA

Com objetivo de investigar topologias de multiplicadores energeticamente eficientes, foram escolhidos neste trabalho os multiplicadores Array (J. M. RABAEY; NIKOLIC, 2003), Vedic (K. BATHIJA et al., 2012), Baugh-Wooley (RAJMOHAN; MAHESWRI, 2016) e Booth (HUANG; CAI; SHEN, 2015), descritos no Capítulo 3 e com os circuitos apresentados nas Figuras 5, 7, 9 e 11, respectivamente. As técnicas de redução de energia escolhidas para a exploração são computação aproximada e operação em tensão *near-threshold*.

Para avaliar diferentes circuitos multiplicadores visando o projeto com eficiência energética, este projeto é organizado em 7 etapas:

1. Levantamento Bibliográfico;
2. Definição dos circuitos multiplicadores a serem avaliados;
3. Descrição dos circuitos na tecnologia CMOS 16nm HP PTM;
4. Aplicação de técnicas de redução energética;
5. Definição dos cenários de teste;
6. Análise das características elétricas;
7. Avaliação da qualidade das soluções dos circuitos que adotam aproximação.

As próximas seções detalharão os procedimentos metodológicos adotados. As descrições SPICE dos circuitos estão disponíveis no Apêndice B, assim como em um repositório online. Este repositório também armazena os códigos gerados para observação das métricas de avaliação definidas nesta dissertação. O repositório está disponível no link: <https://github.com/douglasbrgs/dissertacaoDouglasBorges>.

5.1 Técnicas de redução energética

O alto consumo de energia pode vir a causar problemas de duração em baterias, de confiabilidade e de aquecimento em circuitos. A potência de um circuito em tecnologia

CMOS pode ser dividido em três fatores: potência estática, potência dinâmica e potência de curto circuito (CHANDRAKASAN; SHENG; BRODERSEN, 1992). A Equação 14 apresenta a composição da potência total do circuito em função desses fatores (SOUDRIS; PIGUET; GOUTIS, 2002).

$$P_{total} = P_{estática} + P_{dinâmica} + P_{curtocircuito} \quad (14)$$

A potência estática é a potência associada a existência de corrente de fuga nos transistores MOS e pode ser determinada pela corrente de fuga (I_{fuga}) e pela tensão de alimentação do circuito, como mostra a Figura 15. A potência dinâmica é aquela consumida pela carga e descarga das capacitâncias (C) existentes nas conexões e nos *gates* dos transistores MOS. Ela é diretamente proporcional a frequência de chaveamento do circuito e quadraticamente proporcional a tensão de alimentação, como mostra a Equação 16, onde α significa a atividade de chaveamento, C a capacitância de carga, f a frequência de operação do célula e V a tensão de alimentação da célula. Finalmente, a potência de curto circuito é a potência consumida pela ocorrência da ligação direta entre a V_{dd} e GND , e será dependente da corrente de curto circuito ($I_{curtocircuito}$), como mostra a Equação 17.

$$P_{estática} = I_{fuga} \cdot V_{dd} \quad (15)$$

$$P_{dinâmica} = \alpha \cdot C \cdot f \cdot V_{dd}^2 \quad (16)$$

$$P_{curtocircuito} = I_{curtocircuito} \cdot V_{dd} \quad (17)$$

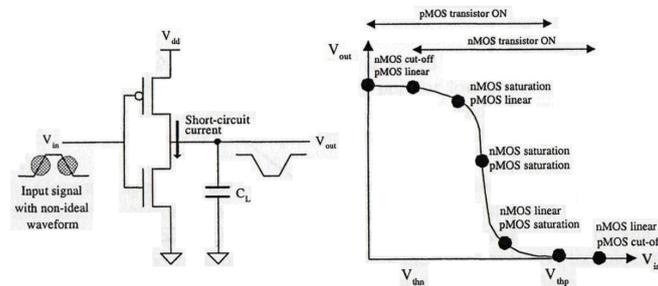
No caso de um inversor CMOS a corrente de curto-circuito aparece quando os transistores NMOS e PMOS estão permitindo a passagem de corrente pelos seus canais (Figura 20). Isso ocorre por causa do *slew rate* das ondas de entrada. O *slew rate* é a velocidade de varrimento, ou seja, a velocidade que a onda leva para saindo do nível baixo chegar no nível alto ou vice e versa.

As técnicas desenvolvidas para reduzir o consumo dinâmico da tecnologia CMOS adotam, em sua maioria, a ideia de *voltage scaling* introduzida em (CHANDRAKASAN; SHENG; BRODERSEN, 1992). Essa ideia surgiu da análise da equação da potência dinâmica. Como pode ser visto na equação 16, a potência dinâmica tem dependência quadrática com a tensão de alimentação.

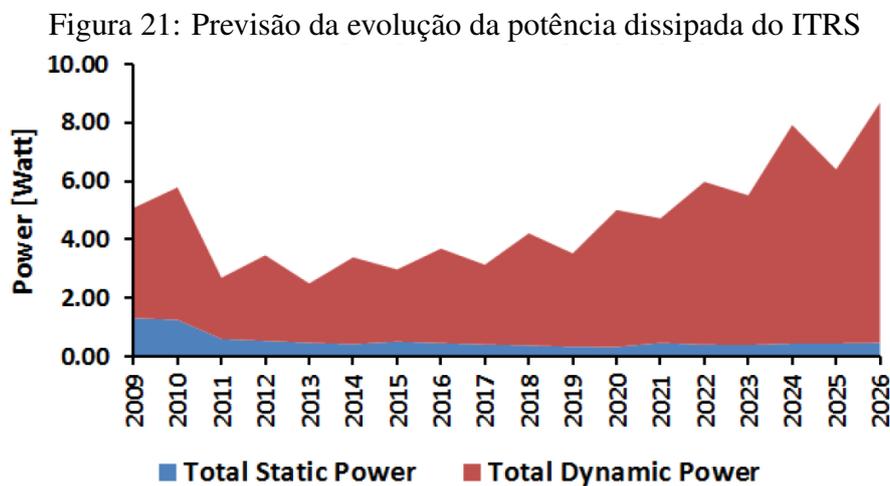
A Figura 21 mostra a projeção da ITRS sobre a progressão da potência estática e dinâmica nos próximos anos, destacando como a componente principal a potência dinâmica. Vale ressaltar que a parcela do consumo estático na tecnologia CMOS é muito pequena considerando apenas tecnologias maiores que 100 nm. Na busca de bons resul-

tados na redução do consumo, o foco principal das pesquisas é o consumo dinâmico.

Figura 20: Corrente de curto-circuito



Fonte: (SOUDRIS; FIGUET; GOUTIS, 2002)



Fonte: (KAHNG, 2013)

O número de transistores nos *System on Chip* (SoC) vem aumentando cada vez mais, objetivando a implementação de novas funcionalidades. No entanto, esse aumento na quantidade de dispositivos impacta negativamente no consumo energético, fazendo com que os limites de resfriamento impeçam os núcleos de operarem em frequência máxima. Uma manifestação desta limitação é descrita como *dark silicon*, nela uma grande parte do silício é desligada para que o SoC continue trabalhando em uma faixa de potência fixa (ESMAEILZADEH et al., 2011).

Uma maneira de evitar a *dark silicon* é otimizar a eficiência energética através do *voltage scaling* da tensão de alimentação (VDD). Como o VDD tem uma relação quadrática e linear com a potência dinâmica e a frequência, reduzir a tensão de alimentação pode refletir em melhorias significativas no consumo de energia. No entanto, como o desempenho é inversamente proporcional à diferença entre a tensão de alimentação do transistor e a tensão de limiar, o VDD só pode ser diminuído a um valor mínimo V_{min} de maneira que os ganhos em eficiência de energia não sejam afetados pela degradação resultante no desempenho. Em tecnologias submicrônicas, sob os efeitos das variações do processo, o au-

mento no atraso é grande o suficiente para causar falhas de temporização. Por este motivo, o V_{min} é geralmente definido como a tensão mínima que proporciona um funcionamento sem falhas. A tensão mínima definida, encontra-se logo acima da tensão de *threshold* do transistor, e é denominada tensão de *near-threshold*, esta permite a minimização da energia em troca de aumento moderado em atraso (PINCKNEY et al., 2012).

5.2 Cenários de teste

Para realizar a análise de desempenho dos circuitos estudados neste trabalho, foram criados oito diferentes casos de teste, definidos adotando os multiplicadores e os somadores inexatos. Dos somadores aproximados encontrados no estado da arte escolheu-se o AMA2 (GUPTA et al., 2013) e o AXA2 (YANG et al., 2013), apresentados nas Figuras 15 e 18 do Capítulo 4.

Inicialmente, visando a caracterização nominal dos circuitos multiplicadores, os componentes foram simulados com tensão de alimentação nominal (0,7V). Por sua vez, o segundo cenário considera os circuitos multiplicadores operando com uma tensão próxima a tensão de *threshold* da tecnologia utilizada, a tensão escolhida para este cenário de simulação foi 0,4 volts. Os próximos cenários exploram diferentes níveis de computação aproximada aplicadas nos somadores dos multiplicadores, conjuntamente também com operação em tensão reduzida. Ao todo são explorados mais 6 casos, onde duas técnicas de aproximação diferentes são aplicadas nos somadores (AXA2 e AMA2), em dois níveis de aproximação (30% e 100%) e com operação em tensão nominal e *near-threshold*. Assim, os oito cenários avaliados são:

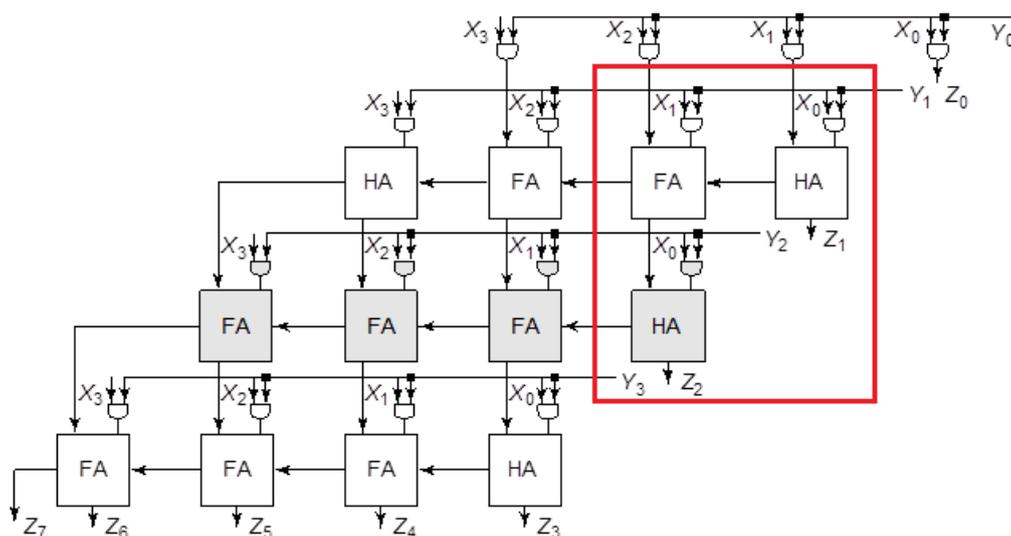
1. **Nominal:** foi usada a tensão nominal (0,7V) como fonte de alimentação dos circuitos;
2. **Near-Threshold:** a tensão usada foi de 0,4V;
3. **100% AXA2:** todos os somadores foram substituídos por somadores aproximados AXA2 e utilizou-se tensão nominal;
4. **30% AXA2:** neste cenário, 30% dos somadores foram substituídos por somadores aproximados AXA2 e utilizou-se tensão nominal;
5. **30% AXA2 em NT:** Assim como no 30% AXA2, 30% dos somadores foram substituídos por somadores aproximados AXA2. Entretanto, agora considera-se conjuntamente a aproximação e a operação com tensão reduzida para 0,4V;
6. **100% AMA2:** todos somadores foram substituídos por somadores aproximados AMA2 e utilizou-se tensão nominal;

7. **30% AMA2:** neste cenário, 30% dos somadores foram substituídos por somadores aproximados AMA2 e utilizou-se tensão nominal;
8. **30% AMA2 em NT:** Assim como no AMA2.30, 30% dos somadores foram substituídos por somadores aproximados AMA2. Entretanto, agora considera-se conjuntamente a aproximação e a operação com tensão reduzida para 0,4V.

A substituição de 100% dos somadores exatos por aproximados visou a avaliação da redução do consumo energético e o impacto infringido no resultado final, observado o grande prejuízo na qualidade do resultado neste experimento, foi definido arbitrariamente um valor de 30% para determinar a quantidade de somadores a serem substituídos pelos inexatos em cada circuito multiplicador. Esta porcentagem foi definida objetivando a introdução de um menor nível de imprecisão mas mantendo a ideia de alcançar uma redução energética significativa. Na Figura 22, são destacados os somadores que foram trocados pelos aproximados no multiplicador Array para introduzir 30% de aproximação, o posicionamento dos circuitos inexatos foi definido buscando deixá-los mais próximos das saídas dos bits menos significativos com o intuito de minimizar o impacto da imprecisão nos bits mais significativos. A mesma lógica de posicionamento foi aplicada nos multiplicadores Baugh-Wooley (Fig. 23), Booth (Fig. 24) e Vedic (Fig. 25).

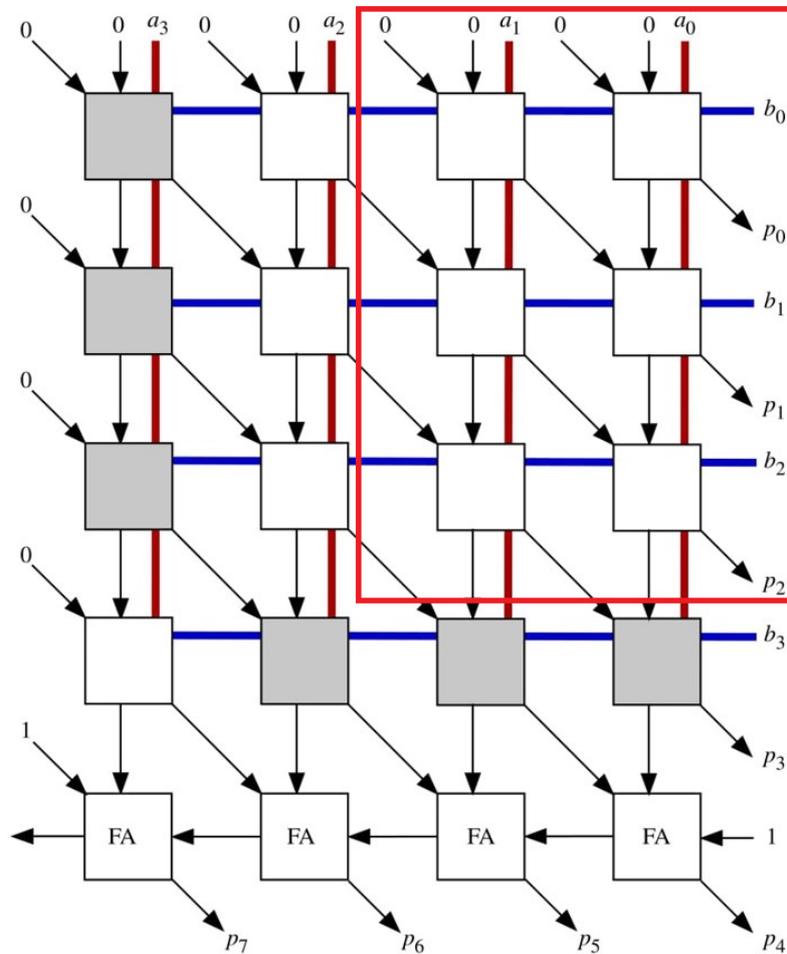
A técnica de operação com tensão reduzida foi avaliada através da observação das saídas corretas das multiplicações para todos os arcos de atrasos dos multiplicadores avaliados. A tensão reduzida impacta na frequência de operação dos circuitos como os resultados posteriormente demonstrarão, mas não compromete os valores finais da operação de multiplicação se a frequência de chaveamento das entradas respeitar o atraso crítico.

Figura 22: Localização dos somadores aproximados no Array



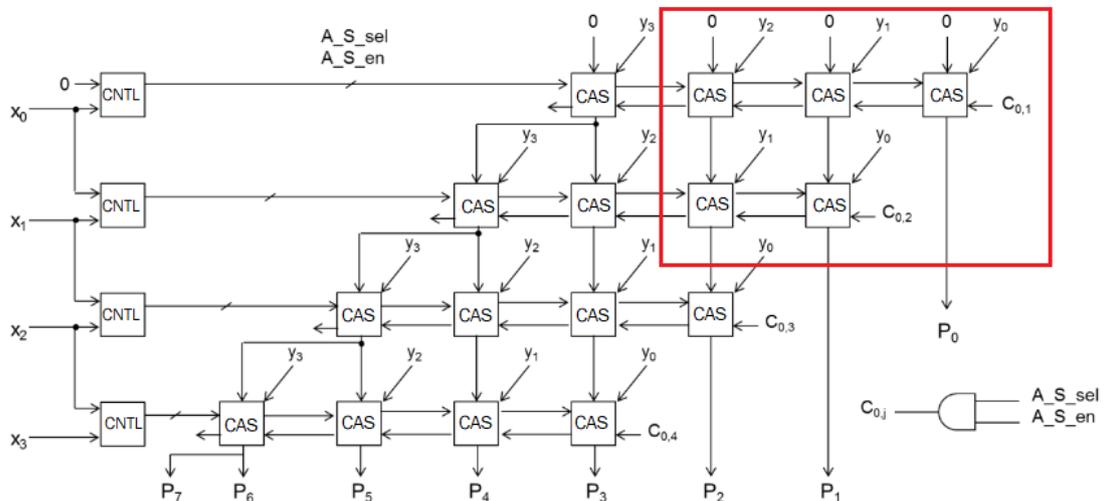
Fonte: Adaptado de (J. M. RABAEY; NIKOLIC, 2003)

Figura 23: Localização dos somadores aproximados no Baugh-Wooley



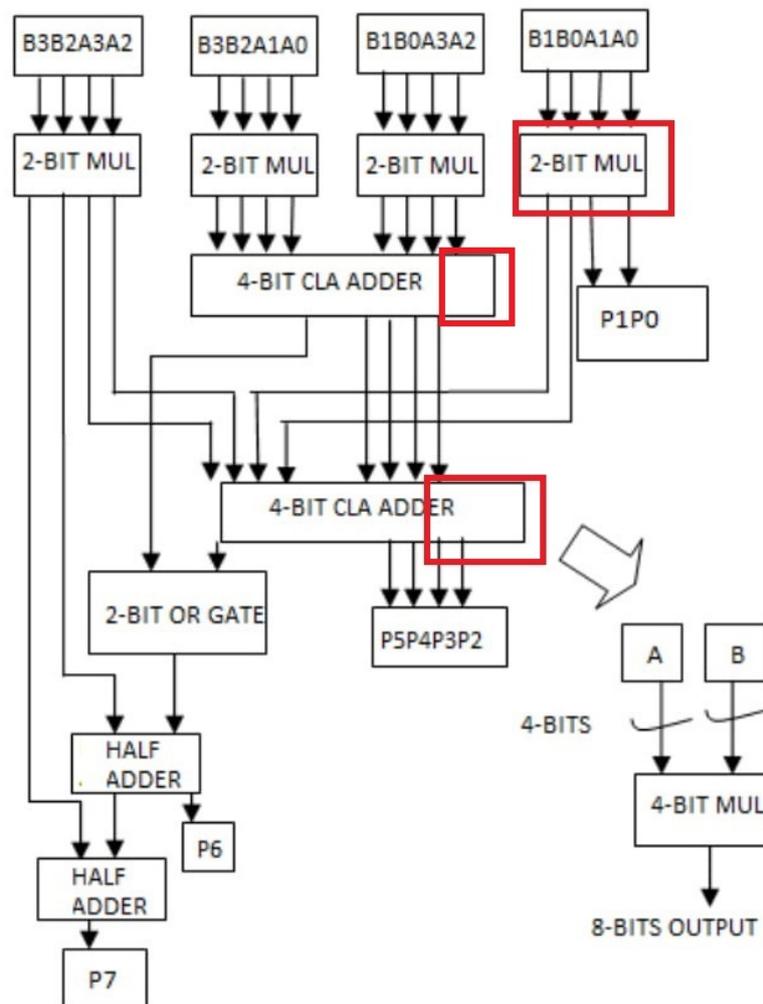
Fonte: Adaptado de (RAJMOHAN; MAHESWRI, 2016)

Figura 24: Localização dos somadores aproximados no Booth



Fonte: Adaptado de (HUANG; CAI; SHEN, 2015)

Figura 25: Localização dos somadores aproximados no Vedic



Fonte: Adaptado de (K. BATHIJA et al., 2012)

5.3 Análise das características elétricas e da precisão

A partir de uma revisão no estado-da-arte constatou-se que, na maioria dos trabalhos, os circuitos são avaliados através de alguns parâmetros específicos, sendo estes:

1. Número de transistores, que proporciona uma estimativa de área;
2. Tempo de propagação, que permite determinar o atraso crítico;
3. Consumo energético e/ou potência;
4. PDP: o produto entre a potência e o atraso (*Power-delay Product* - PDP);
5. Precisão das soluções aproximadas.

Para obter os dados relativos aos quesitos citados, os circuitos foram descritos em termos de seus componentes elétricos, através da linguagem SPICE. Tendo os circuitos descritos, é possível fazer simulações de seu funcionamento usando diferentes combinações de entrada. Com base nessas simulações, faz-se possível observar seu comportamento elétrico e obter dados, como tempo de propagação e consumo de energia. Observando as saídas esperadas e as saídas obtidas nos multiplicadores aproximados pode-se estimar a precisão das soluções aproximadas. Existe uma gama de ferramentas para a realização deste tipo de simulação, uma delas é o Ngspice (NGSPICE, 2018), uma ferramenta com distribuição gratuita e que não apresenta grandes dificuldades em sua utilização.

Configuração do Circuito

Antes de realizar-se as simulações, percebeu-se que não faria sentido considerar um multiplicador isoladamente pelo fato de que esse circuito é usado conectado a outros componentes que possuem suas próprias capacitâncias e resistências. Para reproduzir isso, conectou-se dois inversores em série em cada uma dos bits das entradas. Note que esse procedimento não altera o valor lógico da entrada, apenas introduz um comportamento realístico para os sinais de entrada. Em cada uma das saídas, foi conectado um capacitor de 1fF (um femto Faraday).

Tecnologia e Dimensionamento

Antigamente, associava-se a tecnologia de um transistor ao seu comprimento de seu canal (L). Embora em muitas tecnologias atuais o tamanho da tecnologia seja a dimensão L , existe ainda uma relação direta entre a redução da tecnologia e a redução do comprimento de canal do dispositivo. A dimensão L influencia diretamente na duração do trânsito de elétrons entre os terminais do transistor. Sendo assim, uma menor tecnologia proporciona um dispositivo com maior frequência de resposta. Neste trabalho foi utilizada a tecnologia preditiva de 16nm disponibilizada pela PTM (CAO et al., 2000), este

modelo é um dos mais atuais baseados tecnologia bulk CMOS. Assim como a tecnologia definida, o comprimento de canal neste modelo é de 16nm.

A largura do canal (W) também é um fator importante no dimensionamento de um transistor. Uma maior largura de canal implica no aumento da corrente que pode passar pelo dispositivo, caracterizando o dimensionamento dos dispositivos. Existem algumas técnicas de dimensionamento de transistores, dentre elas, este trabalho adota duas delas. A primeira técnica de dimensionamento usada neste trabalho, define que a largura de um transistor deve ser pelo menos o dobro do tamanho da tecnologia. No que diz respeito a transistores CMOS, é importante citar que os transistores NMOS são duas vezes mais rápidos que os PMOS, isto se deve a maior mobilidade dos elétrons, predominantes nestes dispositivos, comparada a mobilidade das lacunas, predominantes nos PMOS. Para compensar esta diferença, a largura dos transistores PMOS deve ser o dobro da dos NMOS (ZEGHBROECK, 2011). Seguindo a técnica citado, foram usados transistores NMOS com $W=32\text{nm}$ e PMOS com $W=64\text{nm}$.

Nos somadores completos também foi utilizada a técnica *Logical Effort* (SUTHERLAND; SPROULL; HARRIS, 1999). Esta abordagem define que a largura do canal (W) deve ser o tamanho da tecnologia multiplicado pelo número de transistores em série, além de englobar a técnica de mobilidade apresentada anteriormente.

Área

Pode-se ter uma estimativa da área ocupada por um componente fazendo-se uma estimativa do número de transistores presentes em suas estrutura. No entanto, este método não é tão preciso porque cada topologia possui um seu próprio leiaute e pode ser implementada usando diferentes modos de dimensionamento de cada dispositivo. Neste trabalho, a estimativa de área pela contagem de componentes será utilizada para permitir uma comparação de alto nível entre os multiplicadores, abstraindo otimizações de leiaute.

Características Elétricas

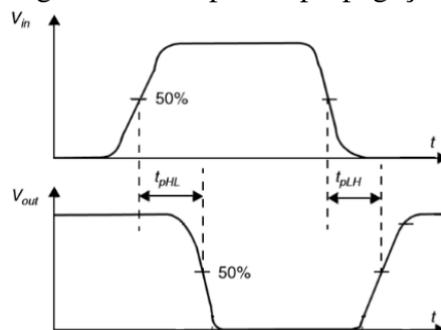
A caracterização elétrica dos multiplicadores conta com dois principais pontos de observação: o atraso e a potência. Este trabalho tem como foco principal a eficiência energética, entretanto, conhecer os atrasos críticos e as relações entre atraso e potência permite ao projetista melhores tomadas de decisão sobre a topologia de multiplicador mais adequada ao seu projeto.

Atraso

O tempo de atraso mostra o quanto um circuito é rápido. Este valor indica o tempo de resposta do circuito o que é fundamental para avaliar o desempenho. Para determinar o atraso crítico, este trabalho investigou o tempo de propagação dos multiplicadores para todos os arcos de atraso.

O tempo de propagação é compreendido como o intervalo de tempo que a saída do circuito demora para transicionar de valor (de 0 para 1 ou de 1 para 0), devido a uma transição na entrada. Estes tempos são medidos a partir de 50% da transição da onda de entrada até 50% da transição da onda da saída. Existem dois tipos de tempo de propagação: o *high-to-low* (tpHL) onde a saída vai de 1 para 0 e o *low-to-high* (tpLH) onde a saída vai de 0 para 1. A Figura 26 ilustra este conceito.

Figura 26: Tempos de propagação



Fonte: (J. M. RABAEY; NIKOLIC, 2003)

Para avaliar-se os tempos de atraso é necessário identificar arcos de atraso. Arcos são pares de combinações de entrada que são diferentes em apenas um bit e produzem saídas diferentes. A Tabela 10, mostra um exemplo de arco onde o bit a_0 passa de “0” para “1”, provocando uma alteração no valor da saída. Para as simulações, adota-se que cada bit das entradas é representado por uma fonte de tensão. A fonte que representa o bit de transição em um arco, inicia em um estado e muda após um período de tempo de dois nanosegundos, enquanto as demais permanecem constantes.

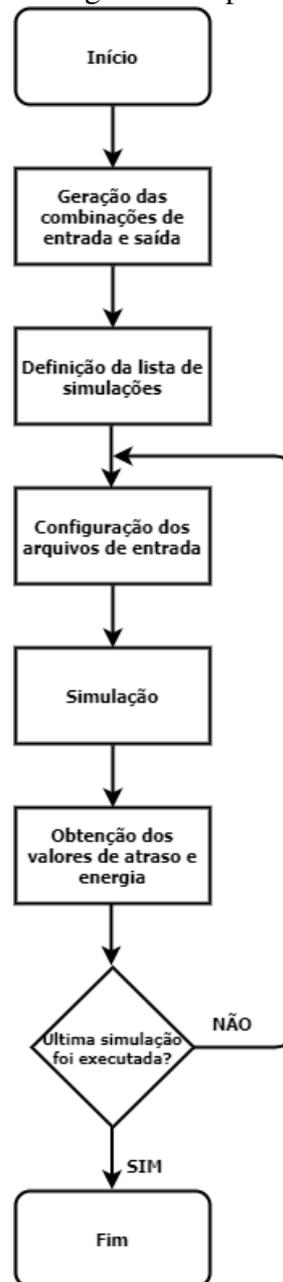
Tabela 10: Exemplo de arco de atraso

ID entrada	Entrada (AB)	Operação(A x B)	Saída
1	00000001	0000 x 0001	00000000
17	00010001	0001 x 0001	00000001

Fonte: O autor

Com o intuito de avaliar todos os arcos de atraso e a potência dissipada em cada uma destas transições, desenvolveu-se uma aplicação em C++, dedicada a validação de multiplicadores de 4 bits. Esta ferramenta configura os arquivos de entrada do simulador e faz a sua chamada para os diferentes casos de estudo. Na Figura 27 está disposto um fluxograma dos passos executados pela aplicação comentada. A etapa inicial consiste em armazenar todas as possíveis entradas e suas respectivas saídas em duas matrizes distintas. A matriz *Entrada* possui em cada uma de suas linhas uma concatenação das entradas *A* e *B*, enquanto a matriz *Saída* apresenta, na linha de mesmo índice, o produto binário das mesmas.

Figura 27: Fluxograma da aplicação de apoio



Fonte: O autor

A principal etapa desta aplicação é a identificação dos arcos. Para isso, são usados dois laços encadeados para comparar todos os casos de entrada entre si e verificar quais pares diferem em apenas um bit. Ao serem identificados, observa-se se os mesmos possuem saídas diferentes. Se a alternância entre esse par de entradas altera a saída, o arco é adicionado a uma lista de arcos de cada uma das saídas sensibilizadas pela transição. Tendo a lista de arcos pronta, a mesma é percorrida por um laço de repetição. A cada iteração as fontes de entrada do circuito são alteradas de acordo com o par formador do arco, é feita a chamada ao simulador elétrico para simular o comportamento do circuito para esta determinada configuração e com isso obtidos os valores de atraso e consumo ener-

gia. Nesta versão, adotou-se o simulador elétrico NGSpice, entretanto, toda a codificação é adaptável para funcionar corretamente com outros simuladores elétricos, tais como o HSPICE da Synopsys.

Ao todo, foram identificados 4840 arcos de atraso, nos multiplicadores para binários sem sinal, e 5088 nos multiplicadores de binários em complemento de dois, ambos com entradas de 4 bits. Os mesmos arcos observados nos circuitos exatos foram analisados nos circuitos aproximados, contudo alguns não puderam ser calculados devido a imprecisão induzida nas saídas não provocar alteração de nível lógico. Estes arcos foram desconsiderados para o estudo dos tempos de atraso e consumo de energia. Dos arcos calculados, consideraremos os maiores tempos de propagação para determinar o atraso crítico. Observa-se também a média dos atrasos como métrica secundária na comparação dos multiplicadores.

Potência

Observar a potência dissipada dos circuitos faz-se importante para escolher uma topologia que consuma menos energia. Isto vem-se tornando cada vez mais importante, devido ao grande avanço tecnológico dos dispositivos eletrônicos portáteis que fazem uso de baterias. Também é importante ressaltar que alto consumo pode gerar sobreaquecimento do dispositivo gerando falhas operacionais. O simulador elétrico Ngspice não possui uma função específica para medir a potência, mas determina a energia consumida em um determinado intervalo de tempo. Através da Equação 18 obtêm-se a energia, calculando a integral da corrente consumida da fonte durante o tempo de cada simulação (4ns). Nesta equação, i corresponde à corrente, Δt à variação de tempo e Vdd à tensão na fonte.

$$Energia = \int_0^{\Delta t} i(Vdd)dt \quad (18)$$

Depois de obtido o valor da energia, pode-se calcular a potência a partir da Equação 19, esta determina que a potência média é a razão entre a energia consumida ao longo do intervalo de tempo, multiplicada pela tensão de alimentação.

$$P(Vdd) = \frac{Energia}{\Delta t} * Vdd \quad (19)$$

Foram observados todos os valores de potência total para cada um dos arcos de atraso dos multiplicadores.

Power-delay Product

Potência e atraso são parâmetros de extrema importância para avaliar circuitos elétricos. Para verificar a eficiência relacionando estes dois quesitos é utilizado o fator *power-delay product*. O PDP consiste, simplesmente, no produto entre os mesmos, como mostrado na Equação 20. É possível calcular um PDP para cada arco de

atraso, relacionando o tempo de propagação deste com o consumo ocorrido durante sua simulação. Neste trabalho, adota-se o PDP referente ao arco de atraso com o maior tempo de propagação.

$$PDP = Atraso * Potencia \quad (20)$$

Precisão

A introdução de somadores aproximados provoca erro numérico e é indicada somente para aplicações tolerantes a erros, como processamento de imagens, vídeos e sons. Optou-se pela avaliação da diferença entre os resultados obtidos pelos circuitos exatos e os apresentados pelos multiplicadores aproximados. Esta métrica é denominada distância de erro (LIANG; HAN; LOMBARDI, 2013), sendo calculada como a diferença absoluta entre o valor esperado e o valor obtido (Equação 21).

Neste trabalho, foi calculada distância de erro para cada bit da saída dos multiplicadores e para o produto final como um todo, os dados usados para esta computação foram obtidos através da simulação de todas as possibilidades de entrada para multiplicadores de 4 bits, num total de 256 casos de teste. Um procedimento em linguagem C# foi implementado para converter as tensões de saída dos circuitos em valores binários, sendo 0,4V e 0,23V, os valores mínimos para o “1” lógico na simulações em tensão nominal (0,7V) e tensão de *near-threshold* (0,4V), respectivamente.

$$ED(a, b) = |a - b| = \left| \sum_i a[i] * 2^i - \sum_j b[j] * 2^j \right| \quad (21)$$

6 RESULTADOS

Foram realizados experimentos para avaliar os multiplicadores de 4 bits considerando 8 diferentes cenários:

1. **Nominal**: foi usada a tensão nominal (0.7V) como fonte de alimentação dos circuitos;
2. **Near-Threshold**: a tensão usada foi de 0,4V;
3. **100% AXA2**: todos os somadores foram substituídos por somadores aproximados AXA2 e utilizou-se tensão nominal;
4. **30% AXA2**: neste cenário, 30% dos somadores foram substituídos por somadores aproximados AXA2 e utilizou-se tensão nominal;
5. **30% AXA2 em NT**: Assim como no 30% AXA2, 30% dos somadores foram substituídos por somadores aproximados AXA2. Entretanto, agora considera-se conjuntamente a aproximação e a operação com tensão reduzida para 0,4V;
6. **100% AMA2**: todos os somadores foram substituídos por somadores aproximados AMA2 e utilizou-se tensão nominal;
7. **30% AMA2**: neste cenário, 30% dos somadores foram substituídos por somadores aproximados AMA2 e utilizou-se tensão nominal;
8. **30% AMA2 em NT**: Assim como no 30% AMA2, 30% dos somadores foram substituídos por somadores aproximados AMA2. Entretanto, agora considera-se conjuntamente a aproximação e a operação com tensão reduzida para 0,4V.

Inicialmente, foi feita uma estimativa do número de transistores empregado na implementação de cada circuito multiplicador avaliado. Em termos gerais, o número de transistores pode proporcionar uma ideia da área do circuito, porém, é importante observar que os circuitos possuem diferentes layouts, que podem ser providos adotando diferentes modos de dimensionamento de transistor.

Através da Tabela 11, pode-se observar o número de transistores utilizados na implementação dos multiplicadores nas metodologias de aproximação utilizadas. Na Figura 28 estão dispostos os percentuais de redução do número de transistores para cada grau de aproximação aplicado, analisando os dados conclui-se que o uso do somador AXA2 proporcionou a maior redução de dispositivos, substituindo 30% dos somadores AXA2, alcançou-se 20,50% de redução no Baugh-Wooley, quando a substituição foi de 100% dos somadores exatos, alcançou-se aproximadamente 70% de redução para o mesmo multiplicador.

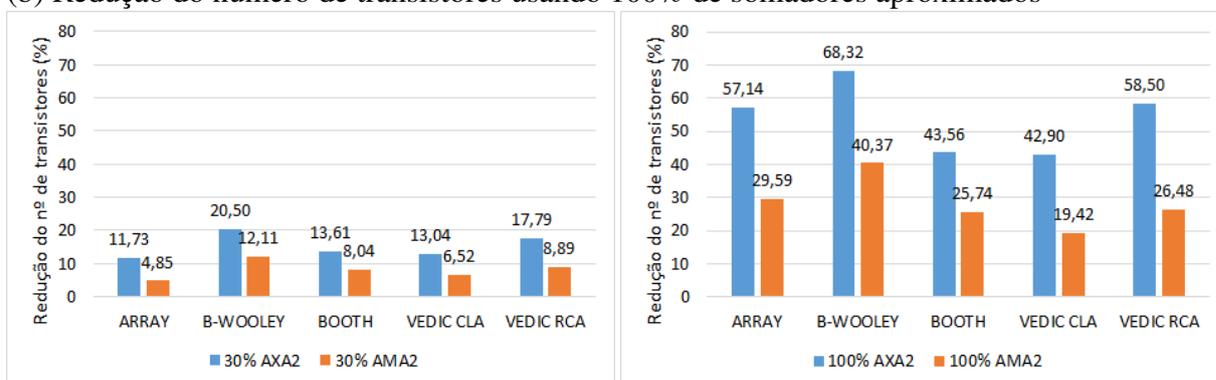
Os demais resultados obtidos serão apresentados primeiramente para cada um dos cenários e, depois, uma comparação geral entre os cenários avaliados, sempre observando os valores de atrasos, potência e PDP. Para os cenários com aproximação, também será apresentado resultados de distância de erro para a aplicação estudo de caso.

Tabela 11: Número de transistores dos multiplicadores em cada cenário de simulação

Multiplicadores	Nível de aproximação				
	Original	30% AXA2	30% AMA2	100% AXA2	100% AMA2
Array	392	346	373	168	276
Baugh-Wooley	644	512	566	204	384
Booth	808	698	743	456	600
Vedic CLA	690	600	645	394	556
Vedic RCA	506	416	461	210	372

Fonte: O autor

Figura 28: (a) Redução do número de transistores usando 30% de somadores aproximados
(b) Redução do número de transistores usando 100% de somadores aproximados



Fonte: O autor

6.1 Nominal

Na Figura 29 é possível observar os tempos de propagação médios e máximos dos multiplicadores para números sem sinal 29(a) e dos multiplicadores para números em complemento de dois 29(b). O Array tem o menor tempo de propagação médio entre todos os multiplicadores, apresentando redução de 15,38% em relação ao Vedic CLA e 25,13% em comparação ao Baugh-Wooley. Quando se considera apenas multiplicadores com sinal, Booth e Baugh-Wooley tem tempos médios semelhantes, existindo mínima vantagem para o Booth. No que se refere ao atraso máximo, nota-se o Vedic RCA como melhor opção, oferecendo redução de 14,46% comparado ao Array e 28,45% em relação ao Booth. O menor atraso crítico para multiplicadores com sinal fica por conta do Baugh-Wooley que apresenta vantagem de 3,2% em comparação a Booth.

Figura 29: (a) Tempos de propagação dos multiplicadores sem sinal em tensão nominal, (b) Tempos de propagação dos multiplicadores com sinal em tensão nominal

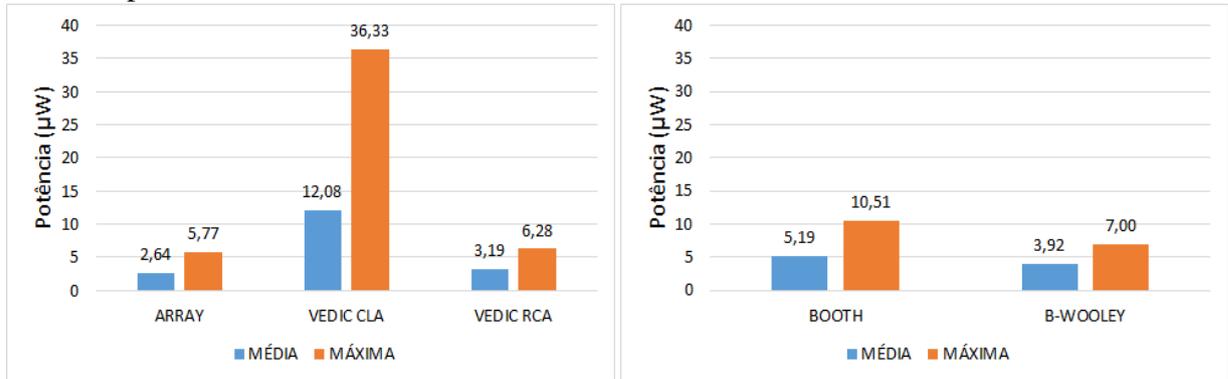


Fonte: O autor

Os resultados para potência média e máxima estão disposto na Figura 30 para os multiplicadores de números sem sinal na Figura 30 (a) e com sinal na Figura 30(b). O Array apresentou a menor potência média, tendo uma economia de 78,14% comparativamente ao Vedic CLA e 49,13% em relação ao Booth. O Baugh-Wooley teve o melhor resultado nos circuitos que operam entradas em complemento de 2, apresentando um potência média 24,47% menor que o Baugh-Wooley. No que diz respeito a potência máxima, pode-se observar o Array como detentor do melhor resultado, apresentando economia de 84,11% em relação ao Vedic CLA, e 45,09% comparado ao Booth. Nos multiplicadores com sinal o Baugh-Wooley teve melhor resultado, trazendo redução de 33,39% em relação ao Booth.

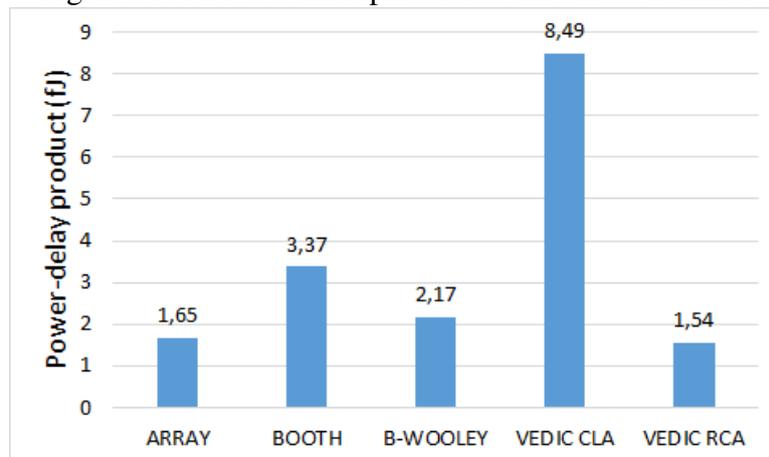
Através da Figura 31 é possível observar os resultados mensurados para o produto entre atraso e potência (PDP). O circuito com melhor resultado foi o Vedic RCA apresentando um PDP 81,86% menor que o Vedic CLA. O Baugh-Wooley teve um resultado 34,97% menor que Booth, ao considerar apenas os multiplicadores com sinal.

Figura 30: (a) Potência dos multiplicadores sem sinal em tensão nominal, (b) Potência dos multiplicadores com sinal em tensão nominal



Fonte: O autor

Figura 31: PDP dos multiplicadores em tensão nominal

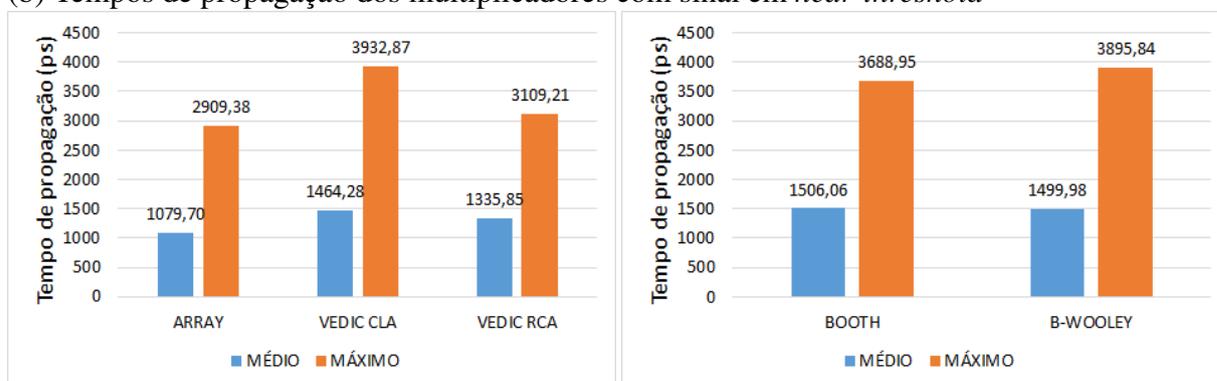


Fonte: O autor

6.2 Near-Threshold

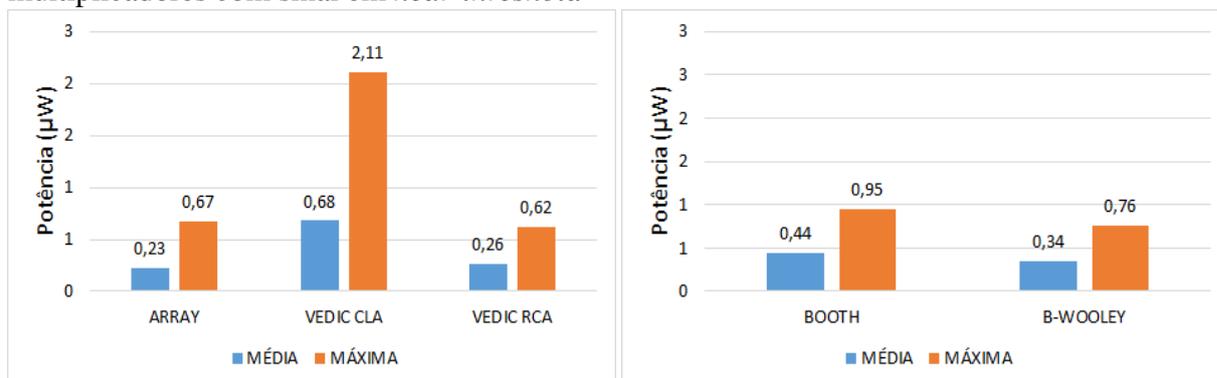
Analisando as figuras 32(a) e 32(b), é possível observar os tempos de propagação médio e máximo para os multiplicadores de números sem e com sinal, nesta ordem, ambos com tensão de entrada igual a 0,4V. O melhor atraso médio observado é o do Array, que é 26,26% menor que o do Vedic CLA e 28,31% menor que o do Booth. Para multiplicadores de entradas em complemento de 2, o melhor tempo de propagação médio é do Baugh-Wooley, com a pequena vantagem de 0,4% sobre o Booth. Tratando-se de tempos máximo percebe-se novamente o Array como melhor opção, com atrasos 26,02% e 25,32% melhores que Vedic CLA e Baugh-Wooley, respectivamente.

Figura 32: (a) Tempos de propagação dos multiplicadores sem sinal em *near-threshold*, (b) Tempos de propagação dos multiplicadores com sinal em *near-threshold*



Fonte: O autor

Figura 33: (a) Potência dos multiplicadores sem sinal em *near-threshold*, (b) Potência dos multiplicadores com sinal em *near-threshold*

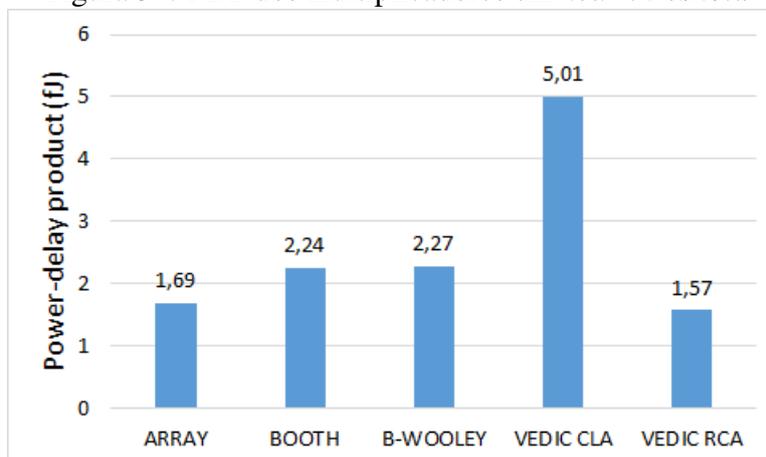


Fonte: O autor

A Figura 33 mostra os resultados de potência para os multiplicadores sem sinal 33(a) e com sinal 33(b), utilizando tensão de entrada de 0,4V. No geral, o Array teve a menor média, representando 66,18% de economia em relação ao Vedic CLA. Para multiplicadores com sinal, foi apresentada uma vantagem de 22,73% do Baugh-Wooley sobre o Booth. Examinando a potência máxima, pode-se notar que o Vedic RCA obteve o melhor

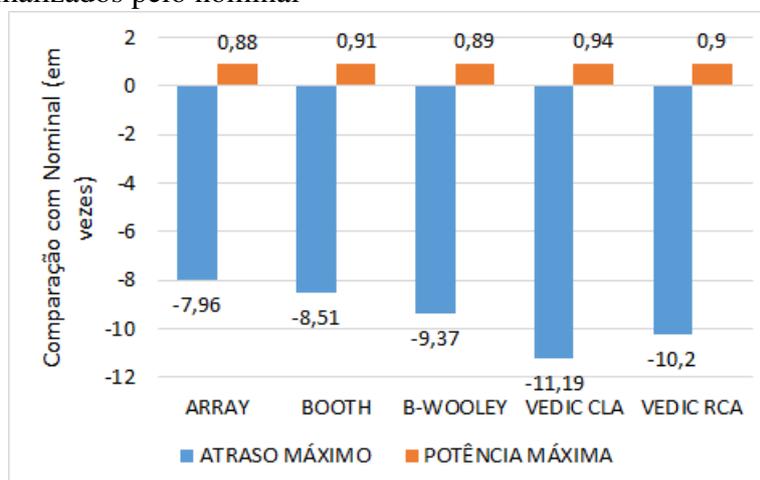
valor, sendo 70,62% mais econômico que o Vedic CLA no pior caso de consumo. Por sua vez, o Baugh-Wooley foi o multiplicador com sinal a trazer menor resultado para potência máxima, superando o Booth em 20%.

Figura 34: PDP dos multiplicadores em *near-threshold*



Fonte: O autor

Figura 35: Tempo de propagação e potência máximos dos multiplicadores em *near-threshold* normalizados pelo nominal



Fonte: O autor

A Figura 34 apresenta os resultados de PDP para os multiplicadores simulados com *near-threshold*. Pode-se concluir que o Vedic RCA apresenta o menor PDP, tendo resultado 2,19 vezes menor que o Vedic CLA. Para os multiplicadores com sinal, Booth e Baugh-Wooley apresentaram PDP's semelhantes, sendo o do Booth o menor deles.

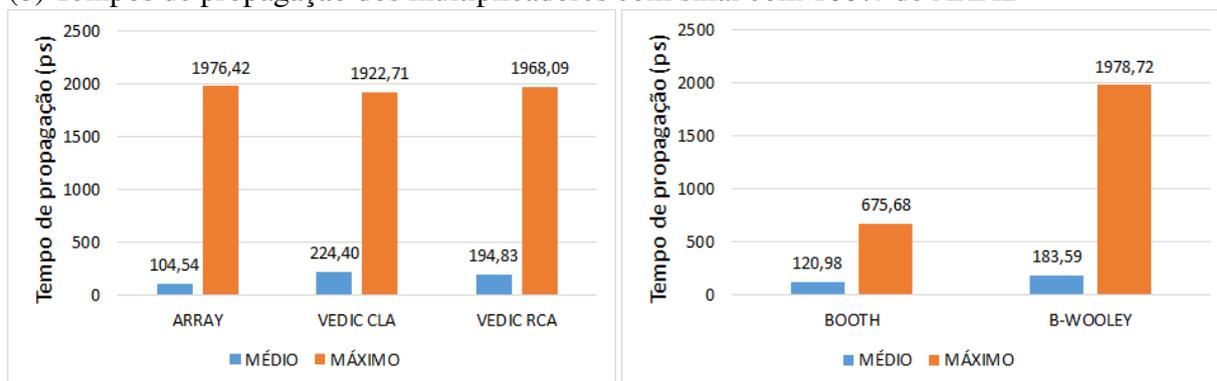
Na Figura 35 estão dispostos os valores máximos de tempo de propagação e potência dos multiplicadores em *near-threshold*, normalizados pelos resultados dos circuitos em tensão nominal. A redução da tensão de alimentação impactou consideravelmente no atraso crítico dos circuitos, trazendo acréscimos superiores a 8 vezes, no entanto a potência

máxima foi reduzida em valores próximos a 90%. Dentre os multiplicadores sem sinal, pode-se destacar que o Array teve o menor aumento no atraso crítico, obtendo uma redução de 88% no consumo. Para multiplicadores com sinal, o Booth apresentou o menor aumento no tempo de propagação máximo impactando em uma economia de 91% na potência em relação a sua versão nominal.

6.3 100% de AXA2

A Figura 36 mostra os resultados de atraso médio e máximo para os multiplicadores com 100% de somadores AXA2. O Array tem o menor atraso médio, com valor 53,41% menor que o tempo médio do Vedic CLA. Para multiplicadores com sinal, o Booth possui o melhor tempo médio tendo vantagem de 34,10% sobre o Baugh-Wooley. No que se refere a atraso máximo o Booth é a melhor opção, tendo atraso crítico 1,92 vezes menor que o Baugh-Wooley. O atraso máximo dos três multiplicadores sem sinal foi semelhantes, porém o Vedic CLA foi com o resultados ligeiramente inferior entre as três alternativas.

Figura 36: (a) Tempos de propagação dos multiplicadores sem sinal com 100% de AXA2, (b) Tempos de propagação dos multiplicadores com sinal com 100% de AXA2



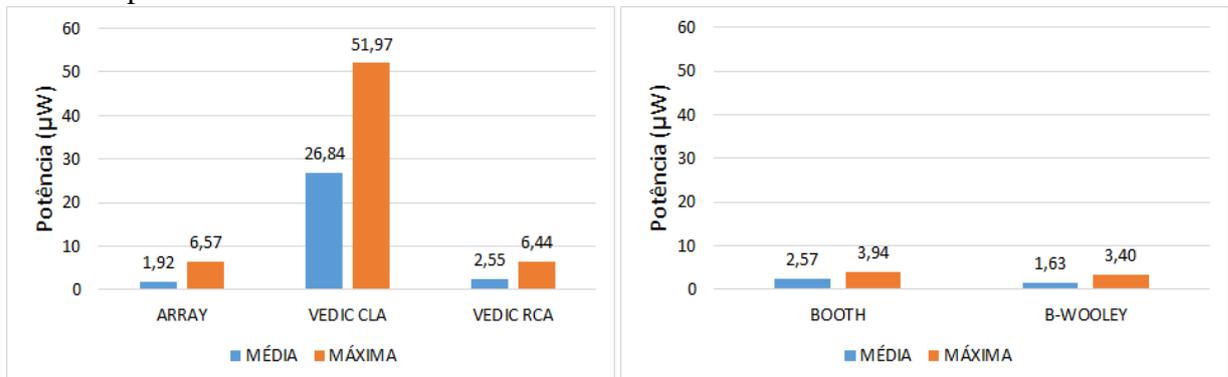
Fonte: O autor

Analisando a Figura 37, observa-se que Baugh-Wooley tem a menor potência média, com resultado 36,57% menor que o Booth e 15,46 vezes menor que o Vedic CLA. Dentre os multiplicadores de números sem sinal, o Array tem a menor potência média, apresentando uma dissipação 12,98 vezes menor que o Vedic CLA.

No que diz respeito a potência máxima, a melhor opção é o Baugh-Wooley que oferece 13,70% de redução em relação ao Booth e 93,46% em relação ao Vedic CLA. Para entradas sem sinal exclusivamente, o Vedic RCA teve o menor resultado, apresentando um valor 7,07 vezes menor que o Vedic CLA para potência máxima.

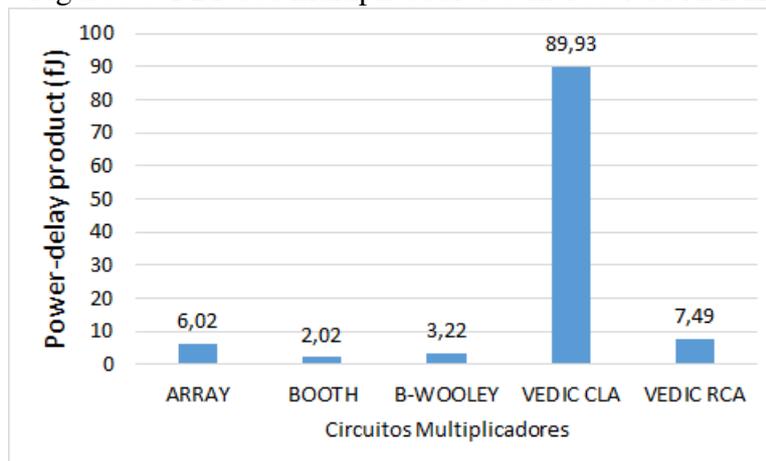
Na Figura 38 pode-se observar os resultados para o produto entre atraso e potência (PDP) para os multiplicadores com 100% de somadores AXA2. O Booth tem o menor PDP, sendo seu resultado 37,27% menor que o Baugh-Wooley e 97,75% menor que o Vedic CLA. Considerando apenas multiplicadores de binários sem sinal, o Array apresenta

Figura 37: (a) Potência dos multiplicadores sem sinal com 100% de AXA2, (b) Potência dos multiplicadores com sinal com 100% de AXA2



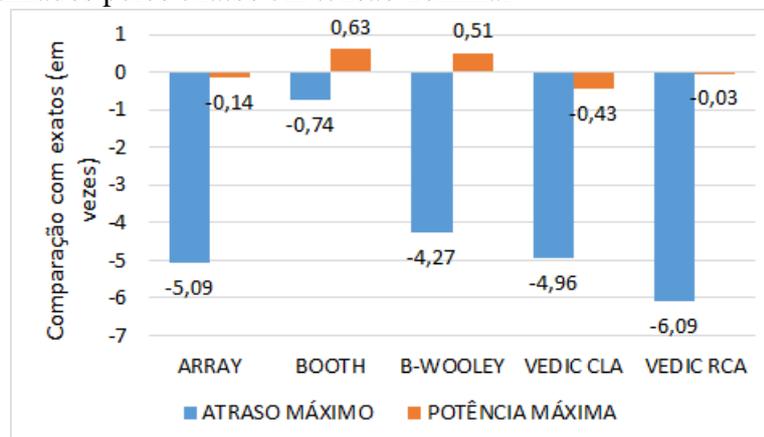
Fonte: O autor

Figura 38: PDP dos multiplicadores com 100% de AXA2



Fonte: O autor

Figura 39: Tempo de propagação e potência máximos dos multiplicadores com 100% de AXA2 normalizados pelos exatos em tensão nominal



Fonte: O autor

o menor PDP, oferecendo redução de 19,63% e 93,30% em relação ao Vedic RCA e ao Vedic CLA respectivamente.

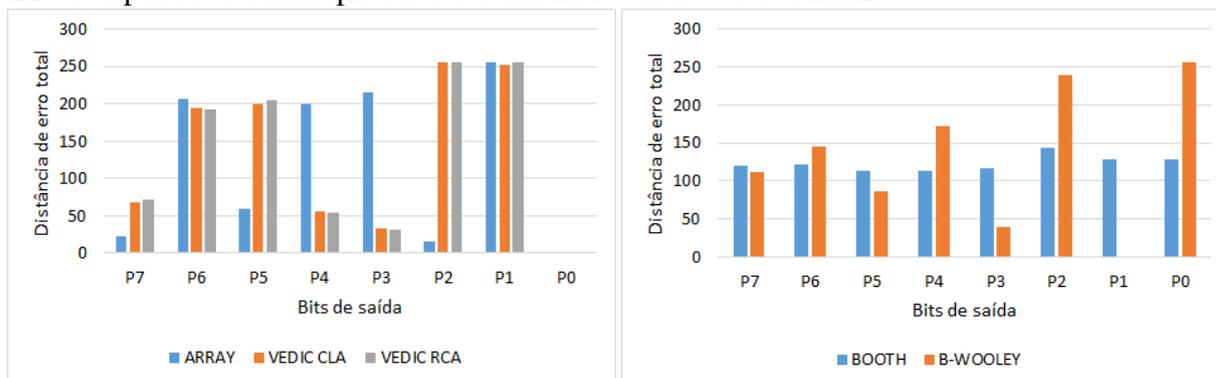
Na Figura 39 estão dispostos os resultados para atraso e potência máxima, dos multiplicadores com 100% de somadores AXA2, normalizados pelos resultados com somadores exatos em tensão nominal. Apenas os multiplicadores com sinal obtiveram redução na potência máxima. O Booth teve redução de 63% da potência em troca de um aumento de 74% no atraso crítico, enquanto o Baugh-Wooley alcançou economia de 51% na potência, com aumento de aproximadamente 4 vezes o tempo de propagação máximo.

Para avaliar o impacto causado pela introdução dos somadores aproximados, foram feitas simulações individuais para cada uma das combinações de entrada possíveis para os circuitos multiplicadores de 4 bits, totalizando 256 casos de teste.

A Figura 40 mostra o somatório da distância de erro para cada bit de saída para os multiplicadores. Observando-se a distância de erro total por bit, nota-se que os multiplicadores Array e Vedic apresentam menor erro no bit mais significativo (P7) comparado aos multiplicadores Booth e Baugh-Wooley, que são multiplicadores projetados para trabalhar com valores representados em complemento de dois. O multiplicador Booth apresenta um resultado regular de erro em todos os bits.

Entretanto, a Tabela 12 apresenta o erro considerando o resultado final por inteiro convertido para decimal. Lembra-se que o erro é dado pela diferença absoluta entre o valor esperado e o valor obtido. Observando o erro numérico para as saídas dos multiplicadores, o Booth foi o que apresentou menor erro médio e menor erro acumulado, dado pela soma dos erros encontrados. A dispersão dos resultados mostrou-se semelhante para todas as topologias avaliadas com desvio padrão na faixa de 26 a 29. Entretanto, este desvio padrão tem maior significado para a topologia Booth que tem a menor média. Ou seja, embora a topologia Booth tenha menor média, os resultados são mais dispersos em relação a média.

Figura 40: (a) ED total por bit dos multiplicadores sem sinal com 100% de AXA2, (b) ED total por bit dos multiplicadores com sinal com 100% de AXA2



Fonte: O autor

Tabela 12: Distância de erro total, média e desvio padrão para os multiplicadores com 100% de somadores AXA2

Multiplicador	Total	Média	Desvio Padrão
Array	15744	61,50	26,19
Booth	6094	23,80	28,67
B-Wooley	10848	42,38	26,39
Vedic CLA	20032	78,25	29,29
Vedic RCA	20212	78,95	28,21

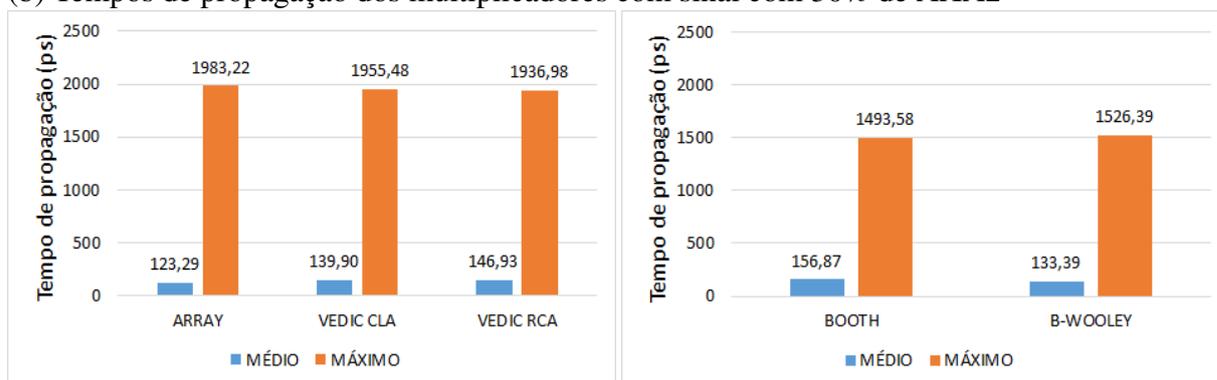
Fonte: O autor

6.4 30% AXA2

Buscando uma alternativa intermediária de exploração da aproximação, observa-se os resultados para os multiplicadores adotando 30% de aproximação. Na Figura 41 é possível observar os resultados de tempo de propagação médios e máximos para os multiplicadores, onde 30% dos somadores foram substituídos pelo somador AXA2. O Array tem o menor tempo de propagação médio, superando o Vedic RCA em 16,09% e o Booth em 21,41%. Analisando apenas multiplicadores com sinal, percebe-se que o Baugh-Wooley tem um tempo de propagação médio 14,97% menor que o Booth.

No que se refere a tempo de propagação máximo, nota-se que o Booth é a melhor opção, superando o Baugh-wooley em 21,49%. Contemplando unicamente os multiplicadores sem sinal, verifica-se resultados semelhantes entre as três topologias, sendo o Vedic RCA o circuito a apresentar o menor tempo de propagação máximo, 2,33% menor que o do Array.

Figura 41: (a) Tempos de propagação dos multiplicadores sem sinal com 30% de AXA2, (b) Tempos de propagação dos multiplicadores com sinal com 30% de AXA2

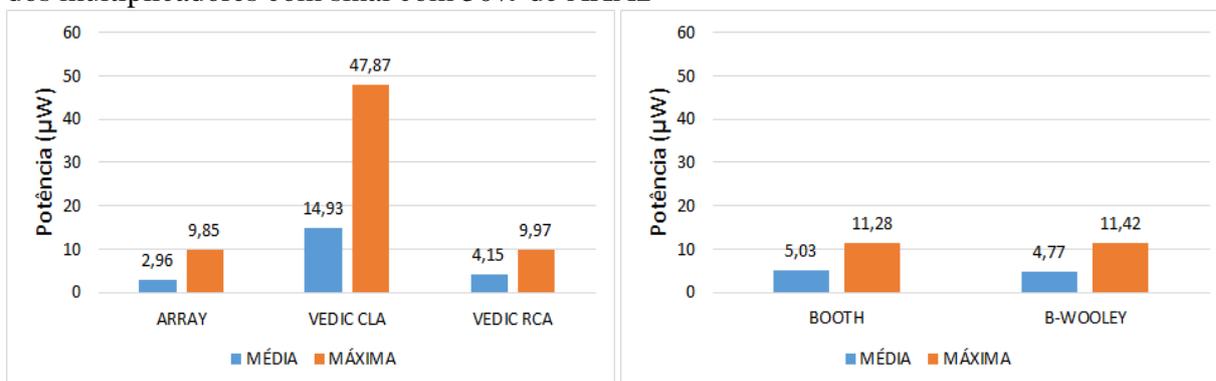


Fonte: O autor

Através da Figura 42 pode-se observar que o Array apresentou a menor potência média, correspondendo a 80,17% de economia em relação ao Vedic CLA. Dentre os multiplicadores com sinal, a menor potência média foi obtida pelo Baugh-Wooley, este oferece uma redução de dissipação de potência de 5,17% quando comparado ao Booth. Quanto a potência máxima, o Array e o Vedic RCA mostraram resultados semelhantes, com vanta-

gem para o Array, este apontou economia de 79,42% em comparação ao Vedic CLA. Em relação aos multiplicadores sem sinal, os dois circuitos apresentaram potência máxima semelhante, com pequena vantagem para o Booth sobre o Baugh-Wooley (1,22%).

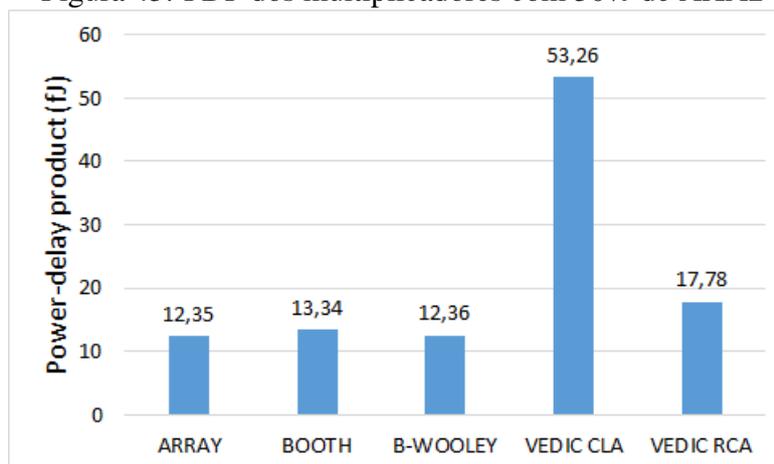
Figura 42: (a) Potência dos multiplicadores sem sinal com 30% de AXA2, (b) Potência dos multiplicadores com sinal com 30% de AXA2



Fonte: O autor

A Figura 43 mostra o produto entre atraso e potência para os multiplicadores com 30% de somadores AXA2. O Array e o Baugh-Wooley apresentaram valores praticamente iguais, sendo os menores PDP's para multiplicadores sem e com sinal respectivamente. O Vedic CLA mostrou resultado 3,31 vezes maior que o do Array, sendo assim a topologia que obteve pior resultado.

Figura 43: PDP dos multiplicadores com 30% de AXA2

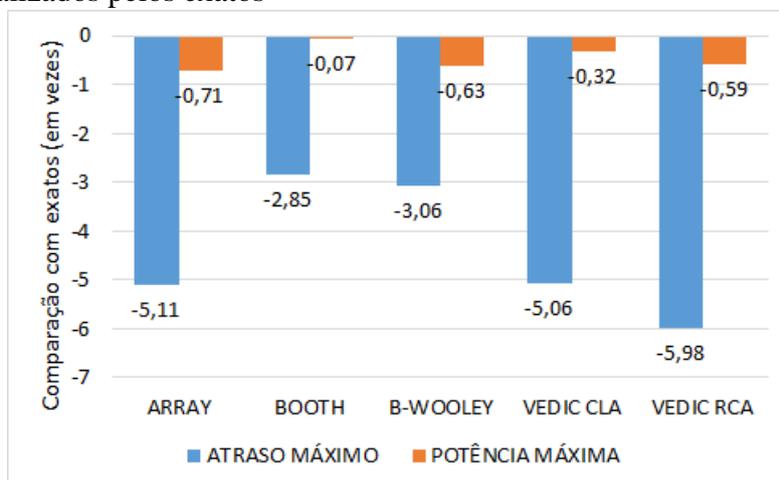


Fonte: O autor

Na Figura 44 é apresentada a normalização dos resultados máximos obtidos pelos multiplicadores com 30% de AXA2, em relação ao resultados apresentados pelos circuitos com somadores exatos. Nota-se que a aproximação não trouxe redução na potência máxima, e sim aumento de até 71%, como mostrado no multiplicador Array. O uso dos 30% de somadores aproximados, ainda implicou em aumento de até 6 vezes no atraso máximo.

Uma das possíveis razões identificadas para este aumento do consumo de energia é que o circuito AXA2, devido aos transistores de passagem, precisa geralmente de uma fonte adicional para aumentar a capacidade de condução da saída e regular o sinal em níveis lógicos bem definidos. Ao optarmos por não inserir esta fonte, que aumentaria ainda mais o consumo energético do multiplicador, os somadores aproximados com AXA2 acabaram operando em situações críticas, onde a carga vista na saída destes eram somadores exatos com maiores capacitâncias do que na versão 100% AXA2. Ao observar os sinais de corrente, notou-se que os circuitos somadores aproximados com AXA2 nesta versão tiveram longos períodos de consumo de energia dinâmica, e níveis lógicos irregulares.

Figura 44: Tempo de propagação e potência máximos dos multiplicadores com 30% de AXA2 normalizados pelos exatos

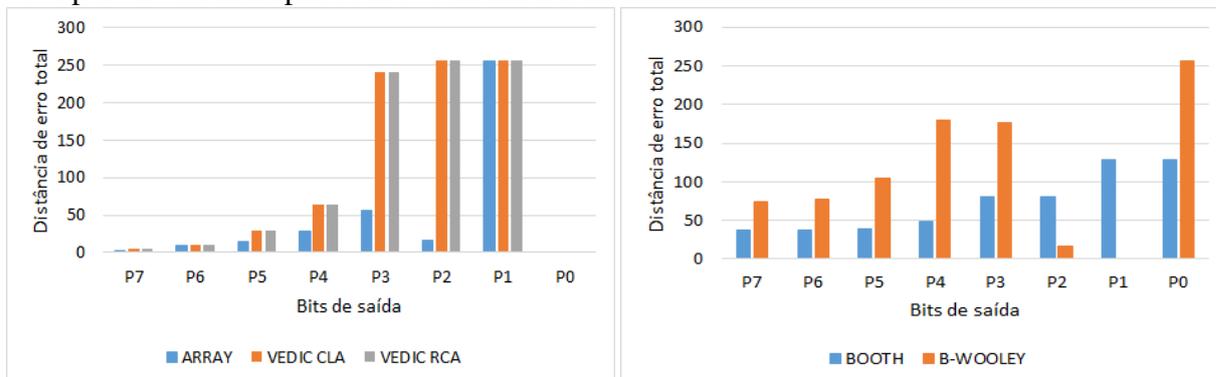


Fonte: O autor

A Figura 45 mostra a distância de erro total apresentada pelos multiplicadores com 30% de somadores AXA2. Observa-se que ao remover os somadores aproximados dos bits mais significativos, a distância de erro para os bits P4 a P7 diminui significativamente comparado com os resultados de 100% AXA (Figura 40). Entretanto, as alterações nos 30% dos somadores tiveram mais impacto no bit P3 dos multiplicadores Vedic, refletindo propagações dos somadores aproximados. Nos multiplicadores Booth e Baugh-Wooley também observa-se a redução da distância de erro para os bits mais significativos (de P5 a P7).

No que refere ao erro para o produto final da multiplicação, a Tabela 13 mostra uma redução considerável do erro acumulado e médio comparado com a versão 100% AXA2 (Figura 12). O cenário com 30% de somadores aproximados AXA2 apresentou uma redução de 94% do erro para o multiplicador Array, 87% para o Vedic, 82% para o Baugh-Wooley e 61% para o Booth. Entretanto, esta configuração demonstrou não ser recomendada para obtenção de eficiência energética nos multiplicadores devido aos resultados de potência elevados discutidos anteriormente.

Figura 45: (a) ED total por bit dos multiplicadores sem sinal com 30% de AXA2, (b) ED total por bit dos multiplicadores com sinal com 30% de AXA2



Fonte: O autor

Tabela 13: Distância de erro total, média e desvio padrão para os multiplicadores com 30% de somadores AXA2

Multiplicador	Total	Média	Desvio Padrão
Array	912	3,56	2,88
Booth	2400	9,38	20,97
B-Wooley	1988	7,77	4,207
Vedic CLA	2688	10,50	4,213
Vedic RCA	2688	10,50	4,213

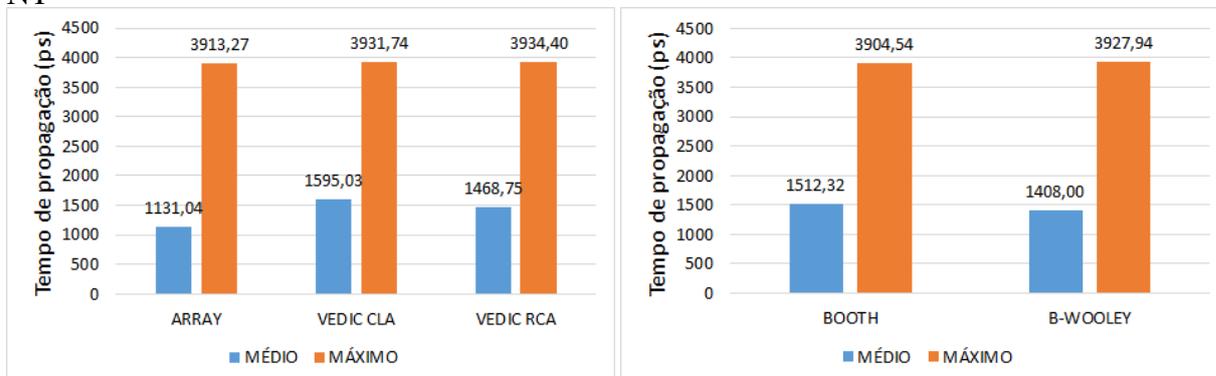
Fonte: O autor

6.5 30% AXA2 em NT

Considerando os favoráveis resultados de operação em NT para redução de energia, estes resultados exploram a adoção de aproximação em 30% dos somadores dos multiplicadores, utilizando somadores AXA, conjuntamente com a operação na tensão de 0.4V. Através da Figura 46 é possível observar os resultados de tempo de propagação médios e máximos para os multiplicadores com 30% de AXA2 em *near-threshold*. O Array foi o melhor em atraso médio, considerando todas as topologias, com resultado cerca de 29,09% melhor que o Vedic CLA. Considerando apenas os multiplicadores com sinal, o Baugh-Wooley superou o Booth em atraso médio, apresentando 6,9% de redução. O Booth apresentou o menor atraso crítico no geral, com resultado 0,6% melhor que o Baugh-Wooley e, 0,76% melhor que o Vedic RCA. O Array apresentou o menor atraso máximo entre os multiplicadores sem sinal, com resultado 0,54% melhor que o Vedic RCA.

A Figura 47 apresenta os valores médios e máximos de potência obtidos na simulação dos multiplicadores. Nota-se o Array como melhor opção considerando apenas o atraso médio, este circuito apresentou economia de 72,37% e 44,74%, comparado ao Vedic CLA e ao Booth, respectivamente. Uma análise exclusiva de multiplicadores para entradas em complemento de dois aponta o Baugh-Wooley como melhor opção, visando menor potência média, o circuito superou o Booth em 21,05%. Considerando a potência máxima

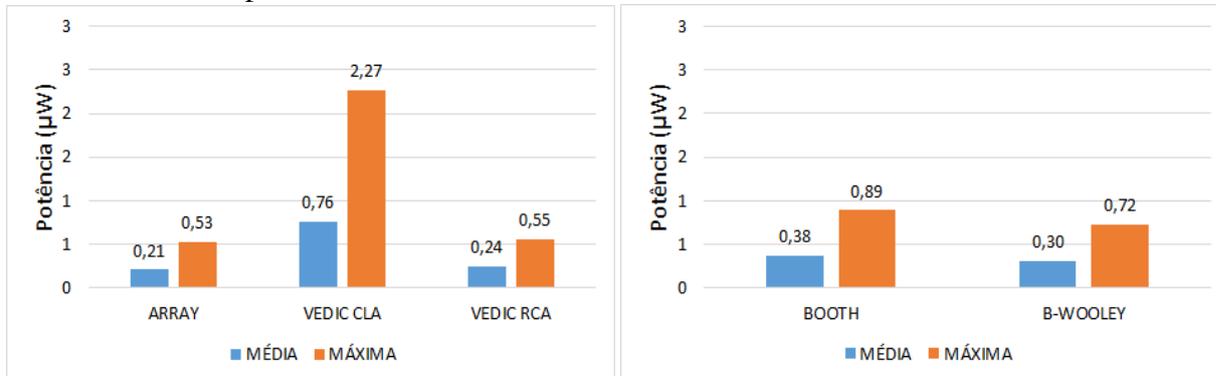
Figura 46: (a) Tempos de propagação dos multiplicadores sem sinal com 30% de AXA2 em NT, (b) Tempos de propagação dos multiplicadores com sinal com 30% de AXA2 em NT



Fonte: O autor

foram obtidos resultados parecidos para o Array e o Vedic RCA, sendo o primeiro o melhor circuito no geral, tendo redução de 76,65% em relação ao Vedic CLA. Já para multiplicadores com sinal, a menor potência máxima foi apresentada pelo Baugh-Wooley que traz uma economia de 19,10% em comparação ao Booth.

Figura 47: (a) Potência dos multiplicadores sem sinal com 30% de AXA2 em NT, (b) Potência dos multiplicadores com sinal com 30% de AXA2 em NT

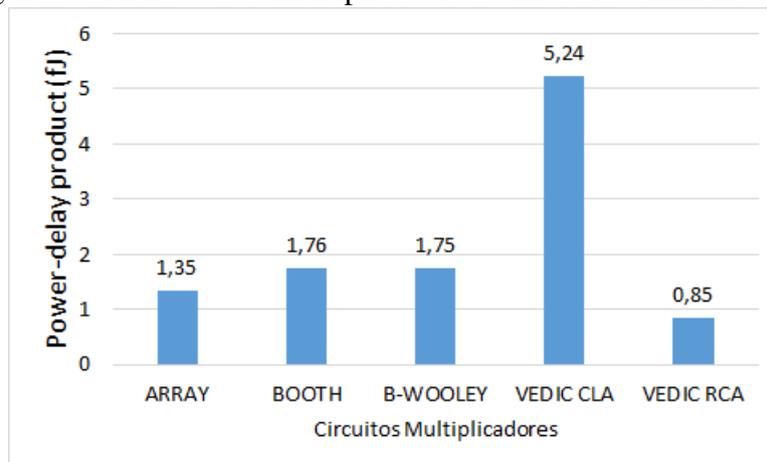


Fonte: O autor

Na Figura 48 pode-se observar o produto entre em atraso e potência para os multiplicadores com 30% de somadores AXA2 e *near-threshold*. Sobre o PDP, o menor resultado foi obtido pelo Vedic RCA, sendo 5,16 vezes menor que o PDP do Vedic CLA. Para multiplicadores de números com sinal, os resultados foram muito semelhantes, com pequena vantagem do Baugh-Wooley.

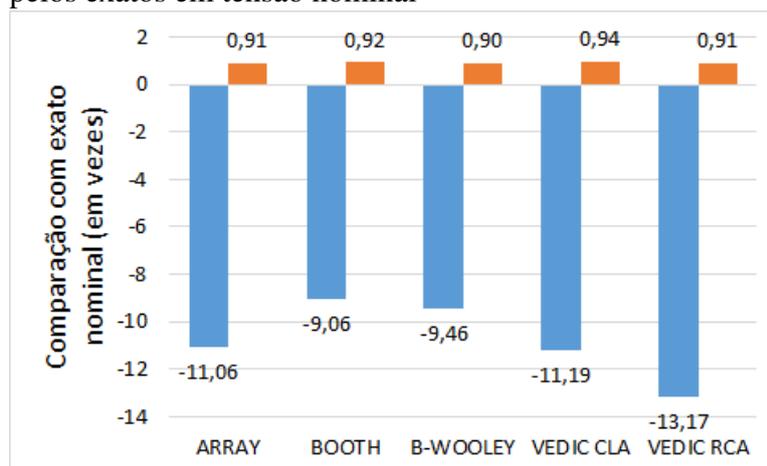
Na Figura 49 os resultados obtidos nos circuitos exatos com tensão nominal foram utilizados para normalizar os resultados mostrados pelos circuitos com 30% de somadores AXA2 em *near-threshold*. Analisando os valores apresentados, percebe-se uma redução de cerca de 90% do consumo energético no pior caso, em troca de um aumento em atraso crítico de até 13 vezes. O Vedic CLA teve a menor redução na potência (94%), porém seu atraso crítico aumentou em cerca de 11 vezes. Nos multiplicadores em complemento

Figura 48: PDP dos dos multiplicadores com 30% de AXA2 em NT



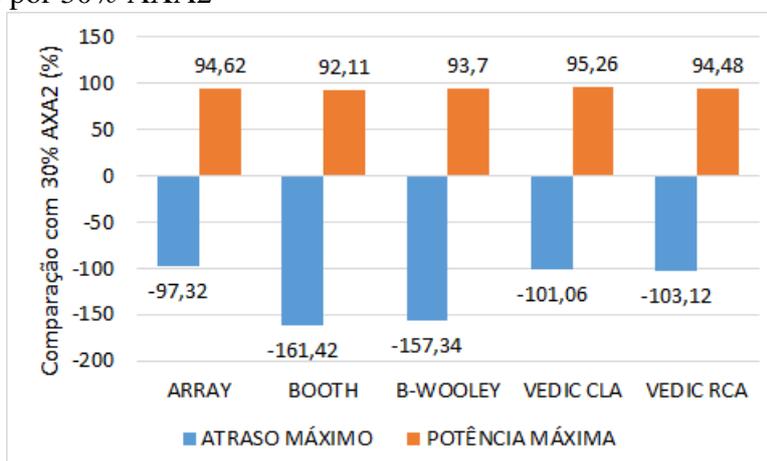
Fonte: O autor

Figura 49: Atraso e potência máximos dos multiplicadores com 30% de AXA2 em NT normalizados pelos exatos em tensão nominal



Fonte: O autor

Figura 50: Atraso e potência máximos dos multiplicadores com 30% de AXA2 em NT normalizados por 30% AXA2



Fonte: O autor

de dois a melhor redução energética foi de 92%, no Booth, em contrapartida houve um aumento de 9 vezes no atraso máximo.

A Figura 50 apresenta a normalização dos resultados máximos do cenário 30% AXA2 em NT relacionando-o ao cenário 30% AXA2 em tensão nominal. Nota-se que o Array alcançou uma redução de 94,62% na dissipação máxima, porém o atraso crítico teve um aumento de 97,32%. No que diz respeito aos multiplicadores com sinal, o Booth teve a maior redução de potência com 93,70% de economia, mas com 157,34% de acréscimo ao tempo de propagação máximo.

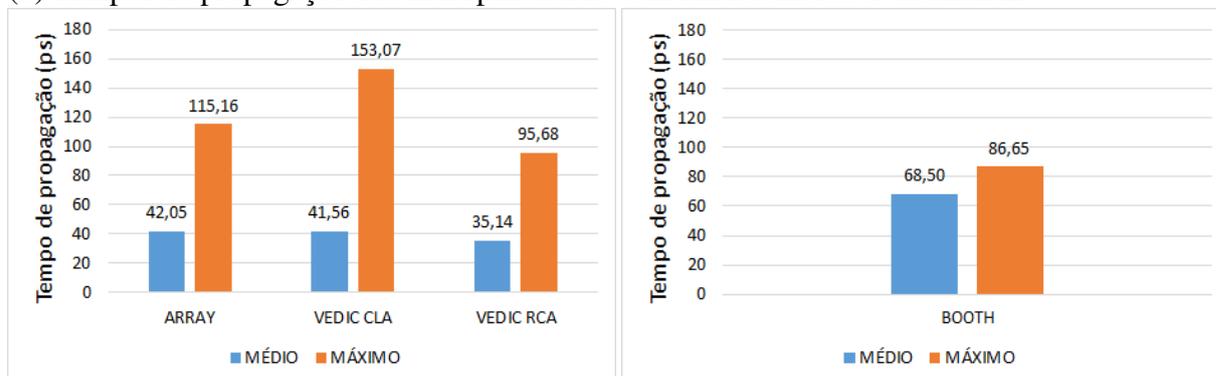
Assim, observa-se que o problema da versão 30% de AXA2 em tensão nominal pode ser contornado ao operar em tensão reduzidas. Salienta-se que os resultados de precisão não foram afetados pela operação near-threshold, sendo os mesmos apresentados na Figura 45 e Tabela 13. Assim, este cenário torna-se favorável considerando potência e precisão.

6.6 100% AMA2

Este cenário apresenta os resultados de atraso e potência para a simulação dos multiplicadores com 100% de somadores AMA2. Neste meio, aferiu-se que o erro induzido pela aproximação impediu a obtenção de dados para o multiplicador Baugh-Wooley, removido então da apresentação dos resultados.

Na Figura 51 é possível observar os resultados de tempo de propagação médio e máximo para os multiplicadores com 100% de somadores AMA2. Pode-se notar que o Vedic RCA tem o menor tempo de propagação médio, mostrando redução de 16,43% comparado ao Array e, 48,70% comparado ao Booth. O menor resultado para de tempo de propagação máximo foi obtido pelo Booth, sendo 43,39% menor que o Vedic CLA.

Figura 51: (a) Tempos de propagação dos multiplicadores sem sinal com 100% de AMA2, (b) Tempos de propagação dos multiplicadores com sinal com 100% de AMA2

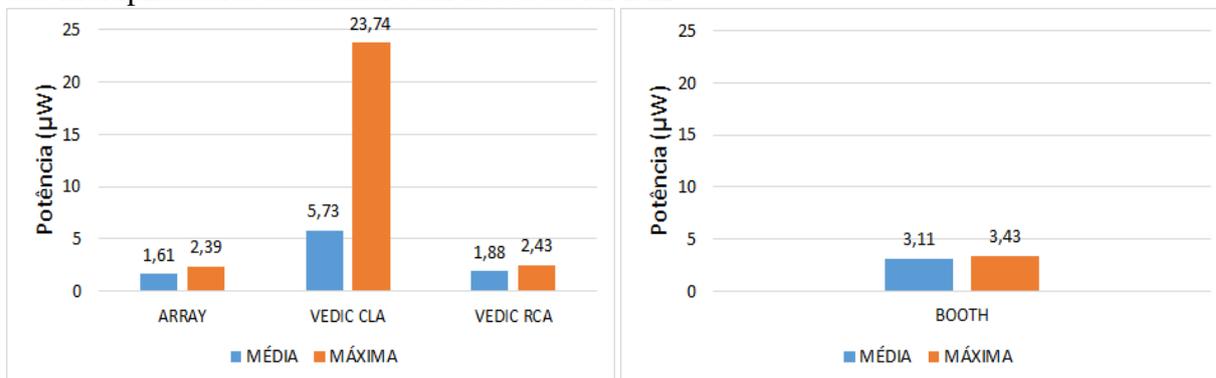


Fonte: O autor

Os resultados para potência média e máxima mostrados na Figura 52 apontam o Array como melhor opção nos dois quesitos avaliados. O multiplicador Array apresen-

tou redução de 71,90% na potência média e de 89,93% na potência máxima, ambas em comparação com o Vedic CLA.

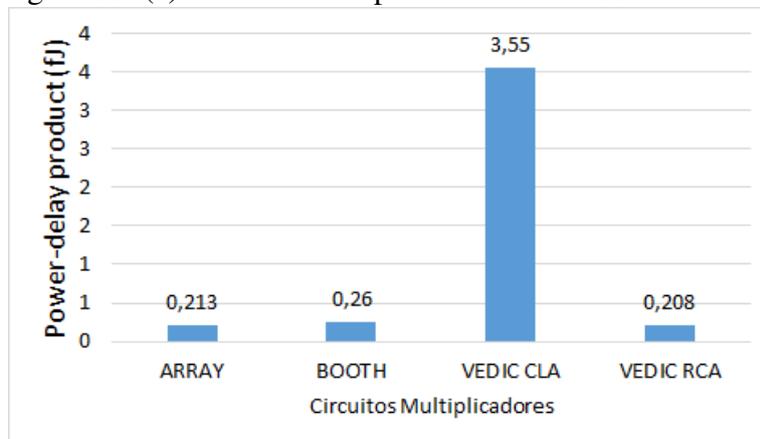
Figura 52: (a) Potência dos multiplicadores sem sinal com 100% de AMA2, (b) Potência dos multiplicadores com sinal com 100% de AMA2



Fonte: O autor

Os valores obtidos para o produto entre atraso e potência (PDP) estão dispostos na Figura 53. O Array e o Vedic RCA tiveram resultados similares, com mínima vantagem do Vedic RCA, sendo o PDP deste último, 94,14% menor que o do Vedic CLA.

Figura 53: (a) PDP dos multiplicadores com 100% de AMA2

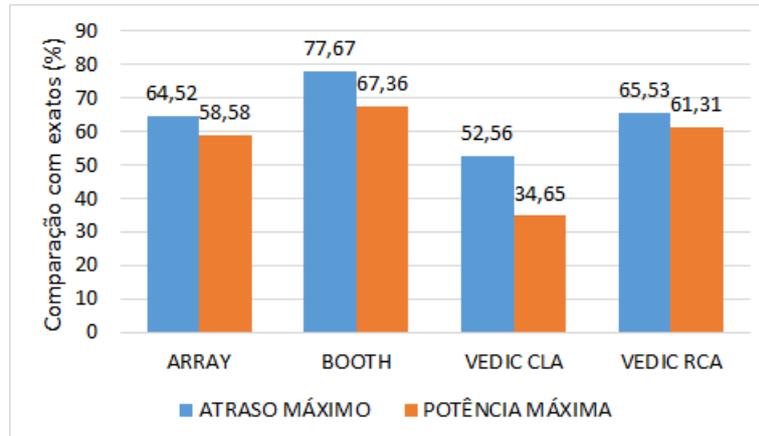


Fonte: O autor

Analisando a Figura 54, é possível observar a comparação entre os resultados obtidos para os multiplicadores com 100% de somadores AMA2 e os multiplicadores exatos. As maiores reduções de potência podem ser observadas no Booth e no Vedic RCA (67,36% e 61,31%), estes circuitos também tiveram redução no atraso crítico, 77,67% e 65,53%, respectivamente.

Na Figura 55 está disposta a distância de erro total, para cada um dos bits de saída, dos multiplicadores com 100% de somadores AMA2. Nota-se que multiplicadores sem sinal tem menor número de erros nos bits de P5 a P7, com erro decaindo do bit menos para o mais significativo. Os multiplicadores Booth e Baugh-Wooley apresentaram número de

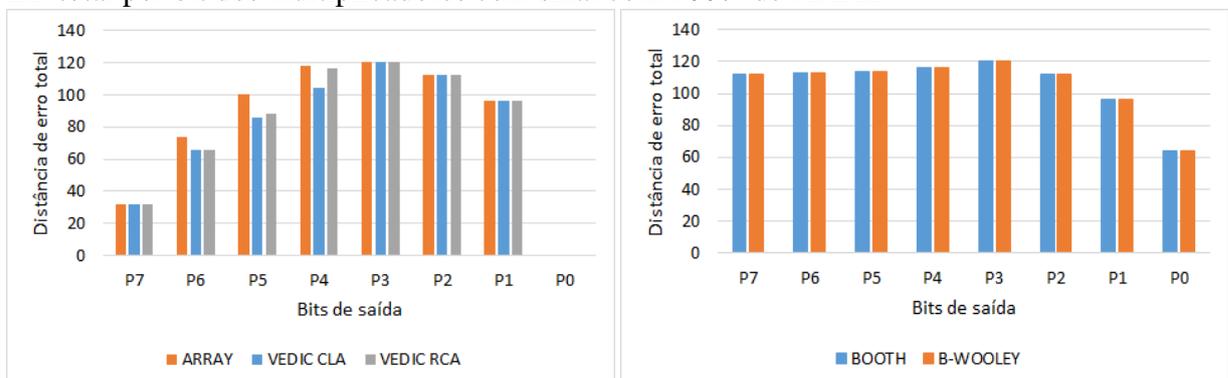
Figura 54: Atraso e potência máximos dos multiplicadores com 100% de AMA2 normalizados pelos exatos



Fonte: O autor

erros igual para todos os bits de saída, com um crescimento gradativo do erro partindo do bit P0 até o P3. Contudo a Tabela 14, mostra que considerando o resultado final em decimal, os multiplicadores para entradas em complemento de dois, têm menor distância de erro total, média e desvio padrão que os multiplicadores Array e Booth.

Figura 55: (a) ED total por bit dos multiplicadores sem sinal com 100% de AMA2, (b) ED total por bit dos multiplicadores com sinal com 100% de AMA2



Fonte: O autor

Tabela 14: Distância de erro total, média e desvio padrão para os multiplicadores com 100% de somadores AMA2

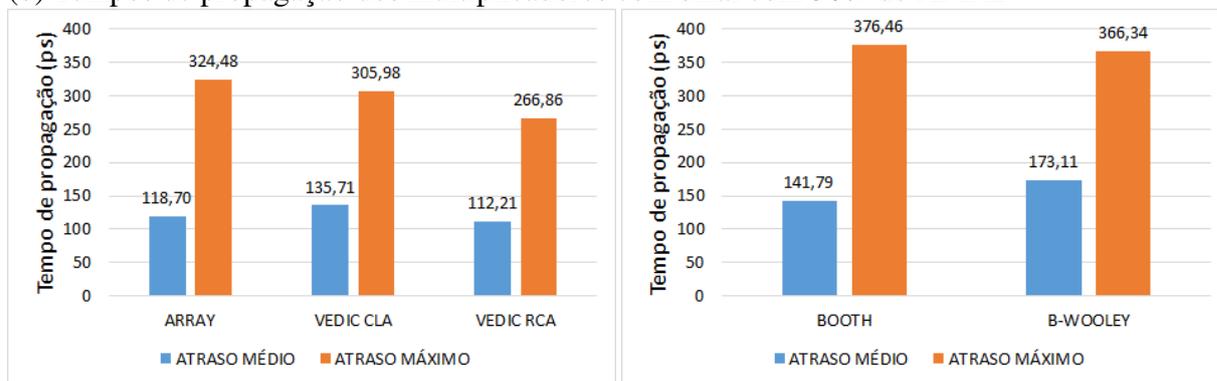
Multiplicador	Total	Média	Desvio Padrão
Array	6160	24,06	23,31
Booth	4152	16,22	14,60
B-Wooley	4096	16,00	14,36
Vedic CLA	13980	54,61	52,32
Vedic RCA	13436	52,48	51,77

Fonte: O autor

6.7 30% AMA2

Através da Figura 56, é possível observar valores de tempo de propagação médio e máximo para os multiplicadores com 30% de somadores substituídos pelo somador AMA2. O Vedic RCA teve o menor tempo de propagação médio, sendo 17,32% menor que o tempo do Vedic CLA e 35,18% menor que o do Baugh-Wooley. Entre os multiplicadores com sinal, o Booth obteve o menor tempo médio, superando o Baugh-Wooley em 18,09%. Em relação ao tempo de propagação máximo o Vedic RCA teve o melhor resultado dentre todos os circuitos, com atraso máximo 17,76% menor que o Array e, 29,11% menor que o Booth. Para os multiplicadores de entradas em complemento de dois, o atraso máximo do Baugh-Wooley foi 2,69% menor que o do Booth.

Figura 56: (a) Tempos de propagação dos multiplicadores sem sinal com 30% de AMA2, (b) Tempos de propagação dos multiplicadores com sinal com 30% de AMA2



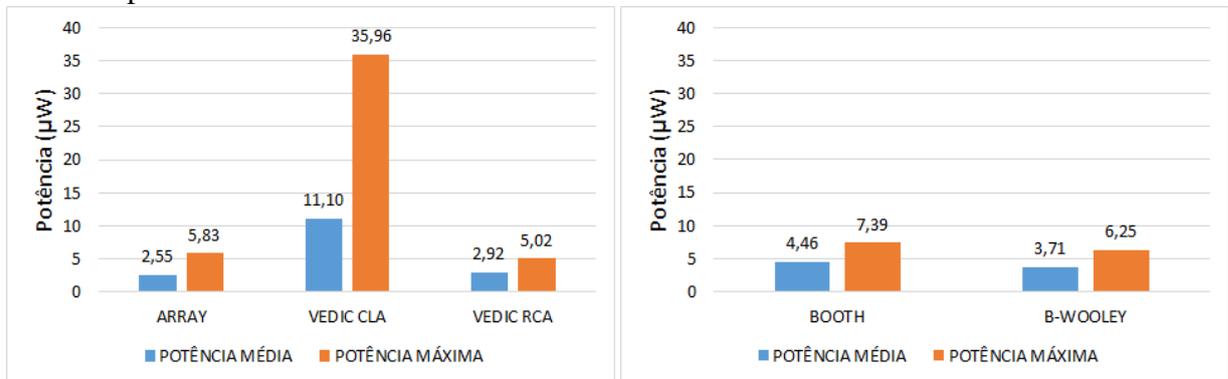
Fonte: O autor

A Figura 57 mostra a potência média e máxima para os multiplicadores com 30% de somadores AMA2. O Array teve a menor média, oferecendo economia de 77,03% em relação ao Vedic CLA. Quanto aos multiplicadores com sinal, o Baugh-Wooley apresenta uma redução de 16,82% na potência média, comparado ao Booth. No que diz respeito à potência máxima, o Vedic RCA é a melhor opção no geral, trazendo economia de 86,04% em comparação ao Vedic CLA, enquanto entre os multiplicadores de números com sinal, o Baugh-Wooley superou o Booth trazendo redução de 16,82% na potência máxima.

Na Figura 58 observa-se o produto entre atraso e potência para os multiplicadores com 30% de somadores substituídos pelo somador AMA2. O Vedic RCA tem o menor PDP, com resultado 2,33 vezes menor que o Vedic CLA. Considerando apenas multiplicadores com sinal, o PDP do Baugh-Wooley é 23,95% menor que o do Booth.

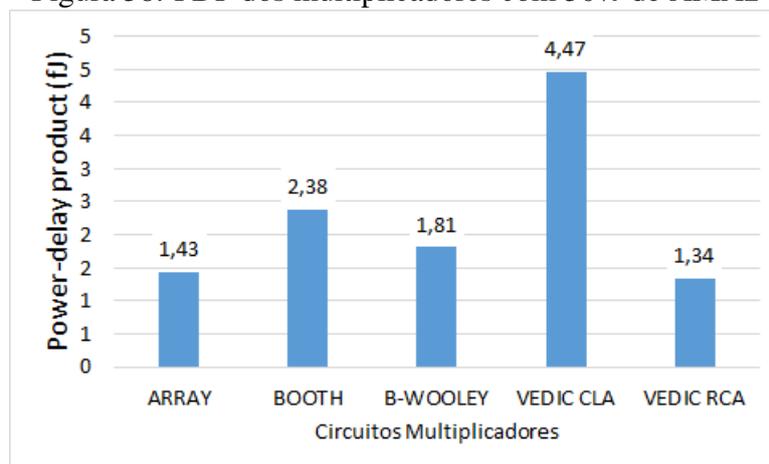
Na Figura 59 é possível observar os resultados obtidos pelo uso de 30% de somadores AMA2 nos circuitos multiplicadores, normalizados pelos resultados dos multiplicadores exatos. O Booth e o Vedic RCA tiveram redução de 29,69% e 20,06% da dissipação máxima, enquanto o tempo de propagação máximo aumentou 2,97% e 3,87%, respectivamente.

Figura 57: (a) Potência dos multiplicadores sem sinal com 30% de AMA2, (b) Potência dos multiplicadores com sinal com 30% de AMA2



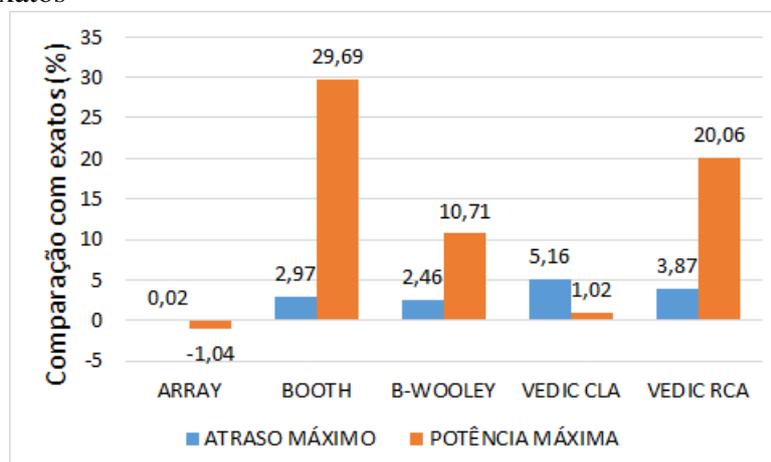
Fonte: O autor

Figura 58: PDP dos multiplicadores com 30% de AMA2



Fonte: O autor

Figura 59: Atraso e potência máximos dos multiplicadores com 30% de AMA2 normalizados pelos exatos

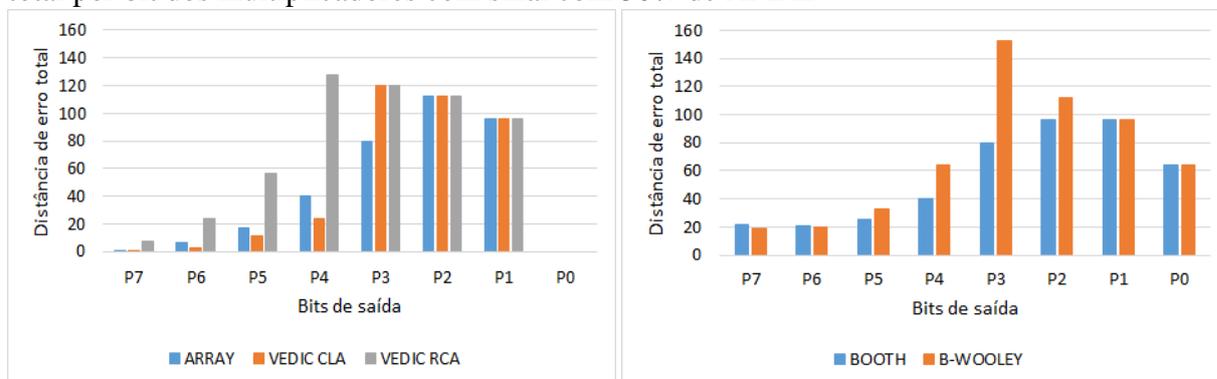


Fonte: O autor

A Figura 60 apresenta a distância de erro para cada uma das saídas nos multiplicadores com 30% de somadores AMA2. Os resultados apontam que em comparação ao erro gerado pela utilização de 100% de AMA2 (Figura 55), a utilização de apenas 30%, próximos aos bits menos significativos, produz uma redução importante no erro para os bits de P4 a P7, com exceção do Vedic RCA, que tem seu erro minimamente aumentado para a saída P4. Também pode-se notar que os circuitos Array e Vedic obtiveram maior precisão no bit mais significativo (P7) do que os multiplicadores destinados a entradas em complemento de dois. No entanto, o erro dos bits de P0 a P2 se manteve praticamente o mesmo nos dois níveis de aproximação, para ambos os multiplicadores, enquanto a saída P3 variou entre piora (Baugh-Wooley) e melhora (Array e Booth) da precisão.

Através da Tabela 15 pode-se analisar o erro considerando a saída dos multiplicadores em valores decimais. Verifica-se que houve grande redução no erro para todos os circuitos comparando com a versão 100% AMA2 (Tabela 14). A diminuição do erro chegou a 92% no Vedic RCA, 91% no Array, 89% no Vedic CLA, 74% no Booth e 66% no Baugh-Wooley.

Figura 60: (a) ED total por bit dos multiplicadores sem sinal com 30% de AMA2, (b) ED total por bit dos multiplicadores com sinal com 30% de AMA2



Fonte: O autor

Tabela 15: Distância de erro total, média e desvio padrão para os multiplicadores com 30% de somadores AMA2

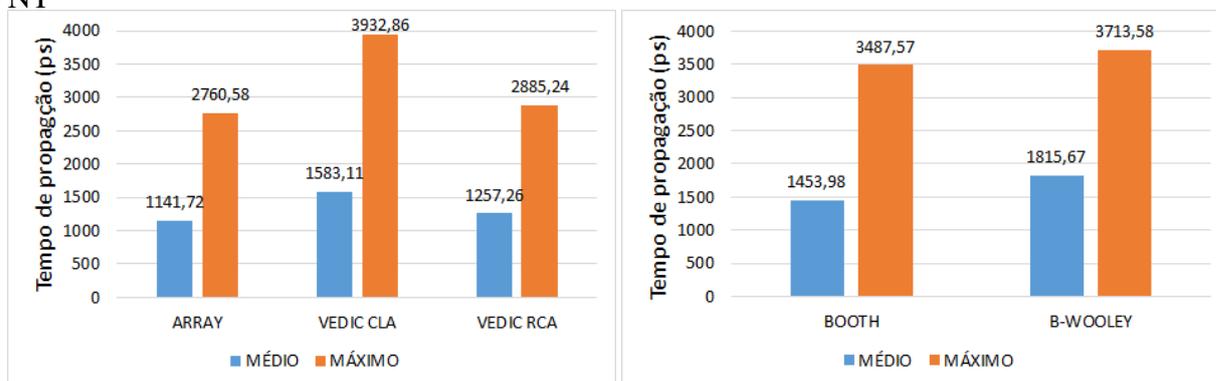
Multiplicador	Total	Média	Desvio Padrão
Array	544	2,13	2,00
Booth	1088	4,25	9,88
B-Wooley	1380	5,39	4,09
Vedic CLA	1488	5,81	4,66
Vedic RCA	1104	4,31	3,45

Fonte: O autor

6.8 30% AMA2 em NT

Finalizando os cenários avaliados, observa-se os resultados para a configuração 30% AMA2 em *near-threshold*. Os tempos de propagação médio e máximo para os multiplicadores com 30% de somadores AMA2, simulados em *near-threshold*, estão disposto na Figura 61. O Array apresentou o menor tempo médio, com valor 27,88% menor que o Vedic CLA e 37,12% menor que o Baugh-Wooley. Considerando apenas multiplicadores com sinal, o Booth foi o circuito que teve melhor atraso médio, superando o Baugh-wooley em 19,92%. Quanto ao tempo de propagação máximo, o Array teve o melhor resultado, sendo 29,80% menor que o Vedic CLA, já para multiplicadores em complemento de dois, o melhor resultado foi apresentado pelo Booth, 6,09% menor que o Baugh-Wooley.

Figura 61: (a) Tempos de propagação dos multiplicadores sem sinal com 30% de AMA2 em NT, (b) Tempos de propagação dos multiplicadores com sinal com 30% de AMA2 em NT



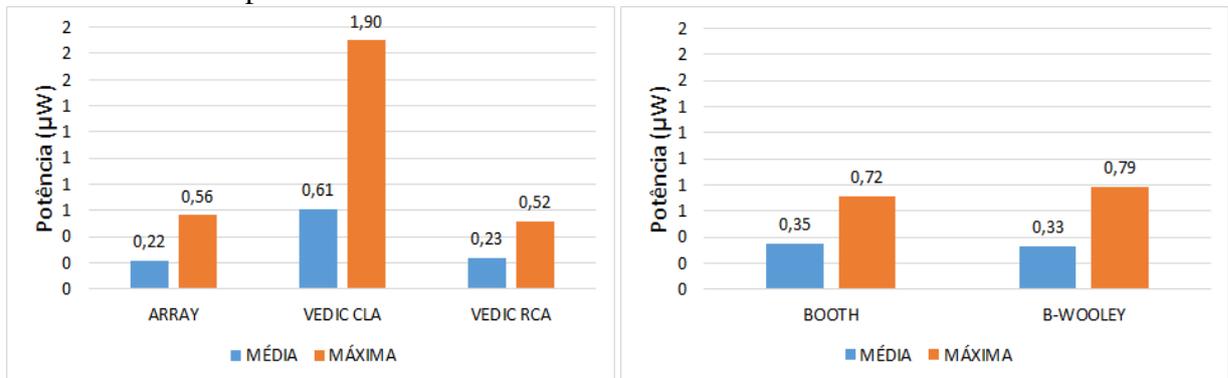
Fonte: O autor

Na Figura 62 podem ser observados os resultados obtidos para potência média e máxima. A menor potência média foi apresentada pelo Array, que trouxe uma redução de 63,93% em comparação ao Vedic CLA. Já para os multiplicadores com sinal, nota-se que o Baugh-Wooley tem uma potência média 5,71% menor que o Booth. No que diz respeito a potência máxima, pode-se perceber o Vedic RCA como melhor opção no geral, este apresenta uma economia de 72,63% em relação ao Vedic CLA. Ao se considerar apenas os multiplicadores com sinal o circuito que apresenta menor dissipação máxima é o Booth, trazendo redução de 8,86% em relação Baugh-Wooley.

Através da Figura 63 é possível observar os resultados para o PDP, onde percebe-se que o menor valor foi apresentado pelo Vedic RCA, com resultado 77,77% menor que o Vedic CLA. Considerando apenas multiplicadores com sinal, o menor produto entre atraso e potência foi obtido pelo Baugh-Wooley, sendo 13,08% menor que o PDP do Booth.

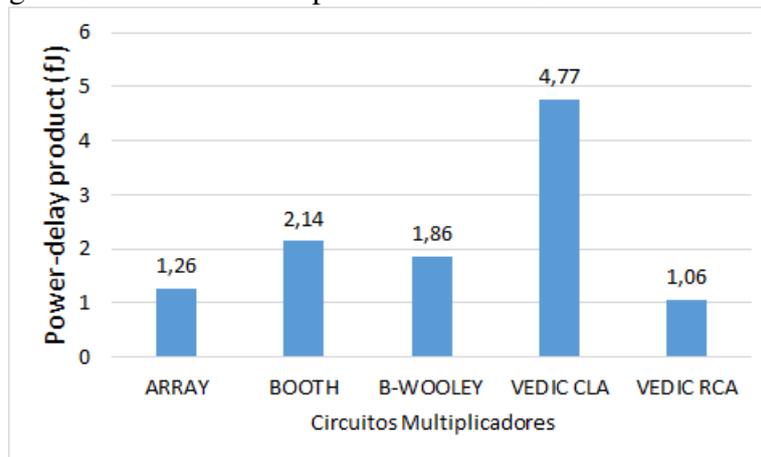
Na Figura 64, pode-se observar uma comparação entre os resultados apresentados pelos multiplicadores com 30% de somadores AMA2 e *near-threshold* e os multiplicadores exatos, simulados em tensão nominal. Os resultados apontaram que o Vedic CLA teve a

Figura 62: (a) Potência dos multiplicadores sem sinal com 30% de AMA2 em NT, (b) Potência dos multiplicadores com sinal com 30% de AMA2 em NT



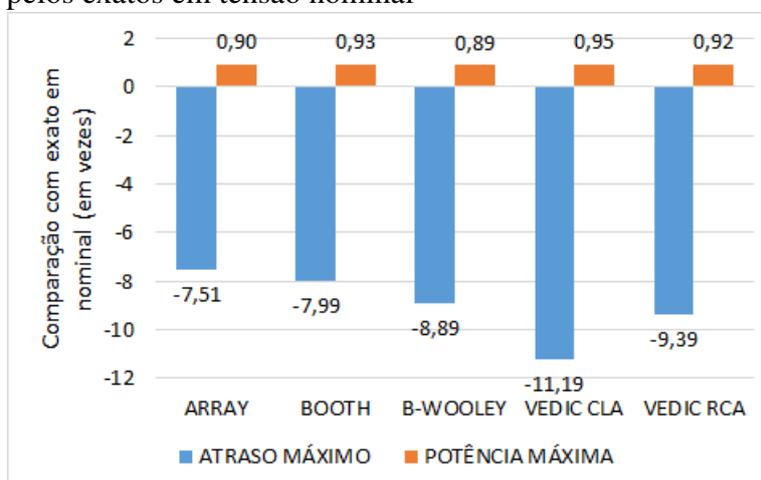
Fonte: O autor

Figura 63: PDP dos multiplicadores com 30% de AMA2 em NT



Fonte: O autor

Figura 64: Atraso e potência máximos dos multiplicadores com 30% AMA2 em NT normalizados pelos exatos em tensão nominal



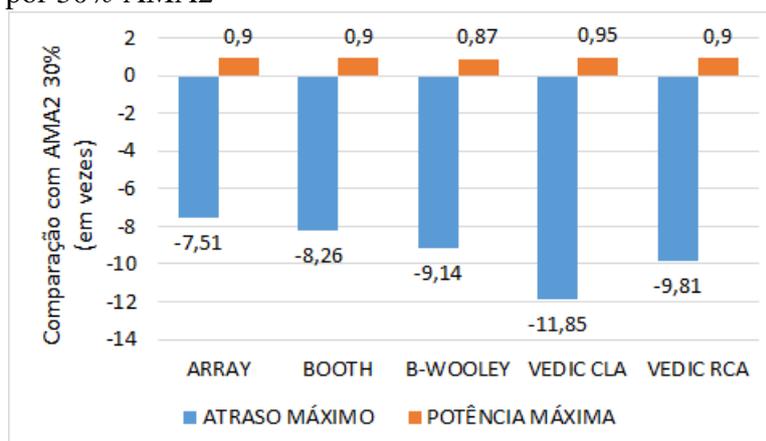
Fonte: O autor

maior redução de potência (95%), porém apresentou o maior acréscimo no atraso crítico, cerca de 11 vezes. Já para os multiplicadores de complemento de dois a menor redução foi a de 93% obtida pelo Booth, em contrapartida houve um acréscimo de 8 vezes no atraso máximo.

A Figura 65 mostra a normalização dos resultados obtidos com 30% de somadores AMA2 e *near-threshold*, utilizando os resultados de 30% de AMA2 em tensão nominal. O Vedic RCA obteve a maior redução da potência máxima (95%), porém, apresentou um aumento de aproximadamente 12 vezes do atraso crítico. O multiplicador com sinal que obteve o melhor resultado foi o Booth, alcançando uma redução de 90% da dissipação, com um acréscimo de aproximadamente 8 vezes do tempo de propagação máximo.

Destaca-se que a precisão dos resultados não é afetada pela operação em tensão de *near-threshold*, sendo as distâncias de erro iguais as mostradas na Figura 60 e na Tabela 15.

Figura 65: Atraso e potência máximos dos multiplicadores com 30% AMA2 em NT normalizados por 30% AMA2



Fonte: O autor

6.9 Avaliação Geral

Dois quesitos muito relevantes no projeto de sistemas digitais são o tempo de atraso e a potência, porém em projetos envolvendo computação aproximada, faz-se importante utilizar a precisão do resultado computado como fator de avaliação de desempenho. O diagrama mostrado na Figura 66 apresenta uma classificação dos cenários de simulação estudados neste trabalho, relacionando-os com os três quesitos de avaliação.

Os circuitos exatos simulados em tensão nominal foram usados como referência para a avaliação dos demais cenários de teste e, como este trabalho visa otimização energética, foi o cenário definido como o com maior compromisso entre atraso e precisão. Os circuitos exatos simulados em *near-threshold* mantém a precisão do cenário nominal, porém esta é acompanhada de uma redução de até 94% no consumo e de um acréscimo de até 11

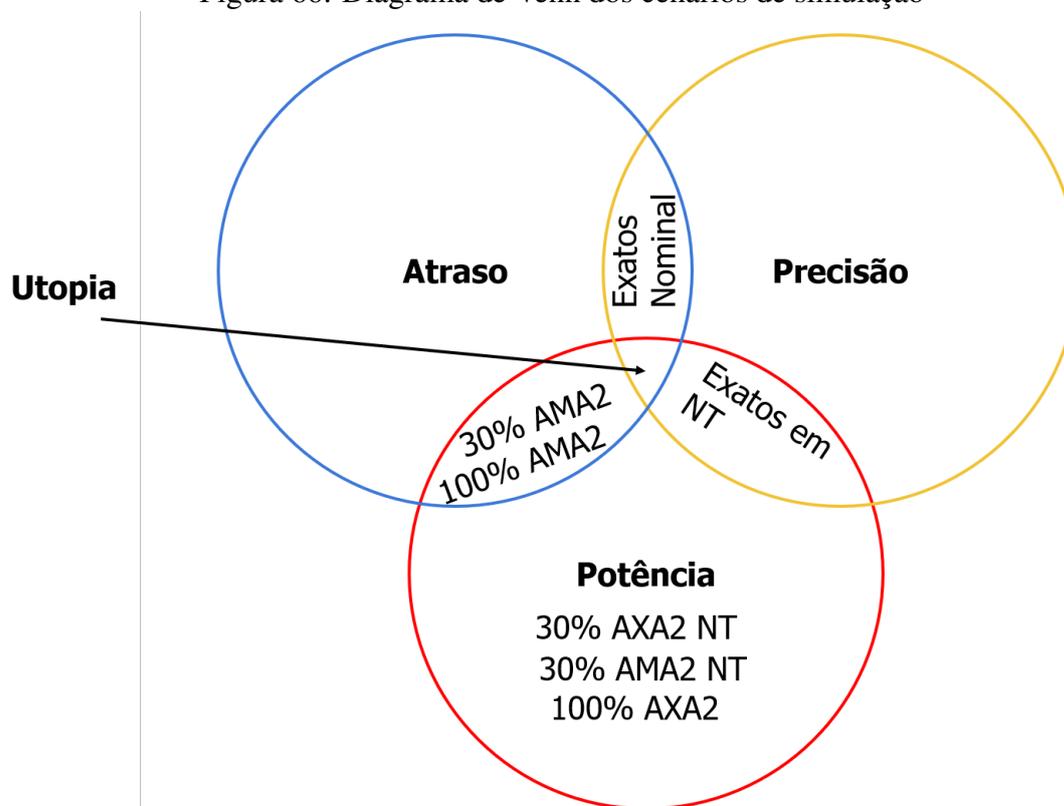
vezes nos tempos de atraso.

O cenário onde foram usados 100% de somadores AXA2 mostrou redução energética para os multiplicadores Booth (63%) e Baugh-Wooley (51%), no entanto introduziu aumento de até 6 vezes nos tempos de atraso, além de grande imprecisão no resultado final. Reduzindo a utilização dos somadores AXA2 para 30% é possível diminuir a imprecisão nos bits mais significativos do produto, porém passa-se a ter apenas prejuízos em atraso e potência, por este motivo o cenário não é mostrado no diagrama da Fig. 66.

Nas simulações utilizando 100% de somadores AMA2 obteve-se grandes reduções em atraso (até 78%) e potência (até 92%), todavia esse nível de aproximação introduziu alta imprecisão. A troca de apenas 30% de somadores exatos por aproximados AMA2 apresentou-se com uma solução para reduzir a imprecisão nos bits mais significativos da saída, ademais de atingir atraso e potência 30% menores em comparação aos dos circuitos exatos em nominal.

A combinação do uso de 30% de somadores aproximados (AXA2 ou AMA2) e *near-threshold* mostrou redução de até 95% na potência, mas teve prejuízo de até 13 vezes no atraso.

Figura 66: Diagrama de Venn dos cenários de simulação



Fonte: O autor

7 CONCLUSÃO

Este trabalho apresenta um apanhado sobre técnicas de multiplicação binária presentes no estado da arte. Foram mostrados os métodos com as quais a operação é realizada, além de como é composta a implementação de cada circuito dedicado. Embora, existam diferentes multiplicadores, todos tem como semelhança a presença de três etapas básicas: Geração de produtos parciais, adição de produtos parciais e um estágio de adição final (SHARMA; SINDAL, 2013). Foram abordadas topologias de multiplicadores destinados a multiplicação de números binários com e sem sinal. Dentre os circuitos que trabalham com números sem sinal foram mostrados os multiplicadores Array e Vedic. Já, em relação a topologias que operam com números em complemento de dois, foram apresentados o Baugh-Wooley e o Booth.

A motivação para o trabalho está embasada na grande necessidade da multiplicação em diversas funções computacionais, e no fato de que o desempenho dos circuitos multiplicadores afeta significativamente o desempenho dos sistemas onde estes estão inseridos. Estudos buscando otimização no consumo energético dos sistemas digitais são importantes no contexto em que a miniaturização dos componentes tenha possibilitado que sejam fabricados dispositivos computacionais cada vez menores e com capacidade para realizar diversas funções. Neste meio, as fontes de energia integradas a estes dispositivos, ou seja, as baterias, também precisam ter suas dimensões reduzidas mas sem perder a capacidade de alimentar satisfatoriamente o funcionamento dos dispositivos. Nos projetos digitais modernos é frequentemente requerido componentes com pequenas dimensões e com consumo energético e tempo de resposta satisfatórios.

Neste estudo foram avaliadas duas alternativas para aumentar a eficiência energética dos circuitos multiplicadores abordados: a computação aproximada e a operação em tensão de *near-threshold*. A introdução da computação aproximada foi feita através da substituição de somadores exatos MA por somadores aproximados AMA e AXA, que por sua vez possuem um número menor de transistores. A operação em *near-threshold* diz respeito a redução da tensão de operação da tecnologia nanométrica a um nível próximo do limiar de *threshold*. O diferencial desta proposta em relação as demais pesquisas sobre multiplicadores observadas no estado da arte, se dá pela abordagem simultânea de duas

técnicas para a otimização energética, além da adoção de uma tecnologia de transistor abaixo de 45nm.

A substituição de 100% dos somadores exatos por somadores AXA2 só produziu redução de dissipação para o Baugh-Wooley e para o Booth, enquanto refletiu em piora para os demais multiplicadores. No Baugh-Wooley a redução foi de 51%, seguida de um aumento de 4 vezes no atraso, enquanto no Booth houve redução de 63% no consumo e 74% de piora no *delay*. Utilizando-se 100% de somadores AMA2 obteve-se melhora no desempenho para todos os circuitos sendo o Booth o que apresentou melhores resultados, com redução de 78% e 67% no atraso e na potência, respectivamente. É importante frisar que a utilização de apenas somadores aproximados provoca um erro numérico muito grande que deve ser considerado de acordo com a aplicação alvo do multiplicador.

Para minimizar o erro gerado pela aproximação de 100% dos somadores, foram realizados experimentos onde a utilização dos circuitos inexatos foi limitada a 30% dos somadores. Os resultados apontaram que a utilização de 30% de somadores AXA2 acabou trazendo prejuízo ao desempenho dos multiplicadores. O Array foi o mais afetado negativamente em termos de potência, tendo aumento de 71% no consumo energético e de 5 vezes no *delay*. Enquanto isso, o uso de 30% de somadores AMA2 trouxe-se otimizações para praticamente todos os circuitos propostos. Dentre os resultados destacam-se as reduções na potência e no atraso de 30% e 3% para o Booth, 20% e 4% para o Vedic RCA e 11% e 2% para o Baugh-Wooley.

Seguindo a técnica de redução de tensão foram realizados experimentos onde o uso de 30% de somadores inexatos e *near-threshold* foram combinados. Individualmente, a operação em *near-threshold* proporcionou reduções entre 88% e 94% na potência, com aumento na faixa de 8 a 11 vezes para o atraso máximo. A maior economia em consumo foi alcançada pelo Vedic CLA (94%), seguida por um aumento de 11 vezes no *delay*. Em comparação aos circuitos exatos nominais, a otimização energética obtida usando AXA2 e *near-threshold* ficou entre 90% e 94% enquanto o acréscimo no atraso apresentou-se entre 9 e 13 vezes. Por fim, o uso de redução de tensão e AMA2 aumentou o atraso entre 7,5 e 11 vezes, enquanto proporcionou redução de consumo energético na faixa de 89% a 95%. Em ambos os cenários com redução de tensão, o Vedic CLA teve a maior otimização energética, variando entre 94% e 95% de redução.

Considerando-se todos os experimentos observados pode-se concluir que a utilização de 30% de somadores AMA2 apresentou-se como a melhor alternativa para a otimização energética dos circuitos multiplicadores avaliados. O uso deste somador proporcionou tanto melhorias em eficiência energética (chegando a 30%) quanto em tempo de propagação (atingindo até 4%), além de apresentar uma distância de erro até 92% menor que sua variante com 100% de somadores aproximados.

7.1 Trabalhos Futuros

Nesta seção são apresentadas três possibilidades de trabalhos futuros:

- utilização de outros somadores aproximados como componentes de circuitos multiplicadores;
- avaliação do impacto da aproximação no processamento de imagem;
- implementação de uma lógica de correção de erro para circuitos aproximados.

O impacto da aproximação no processamento de imagem pode ser avaliado através de um estudo de caso utilizando um filtro. Basicamente, a aplicação de um filtro pode ser realizada através de somas e multiplicações nos valores dos píxeis de uma imagem, sendo assim, é possível introduzir somadores inexatos nos componentes que realizam estas operações, visando redução energética em troca de alguma perda na qualidade da imagem processada.

Neste trabalho pode-se observar que com a substituição dos somadores exatos por aproximados, os multiplicadores passaram a apresentar erros recorrentes nos bits de saída, ou seja, em situações onde se esperava um bit 0 (zero) o resultado apresentou um bit 1 (um) e vice-versa. Como forma de amenizar esses erros pode-se conectar um inversor as saídas que apresentam um quantidade muito elevada de erros, com foco nas que representam os bits mais significativos, assim pode-se aumentar a precisão de um circuito aproximado ao custo da adição de um circuito simples como o inversor.

7.2 Publicações

Ao longo do mestrado, os trabalhos realizados resultaram nas seguintes publicações listadas abaixo. Os resultados finais desta dissertação serão divulgados em futuras publicações.

BORGES, D. M. ; BORBA, A. O. ; ROSA, V. S. ; MEINHARDT, Cristina . Performance Evaluation of Arithmetic Blocks at 16nm Technology. In: WCAS - Workshop on Circuits and System Design, 2019, São Paulo.

BORGES, D. M. ; MEINHARDT, C. ; ROSA, V. S. . Comparison of Baugh-Wooley and Booth Radix-2 Multiplier Architectures in 16nm Technology. In: WCAS - Workshop on Circuits and Systems Design, 2018, Bento Gonçalves.

BORGES, D. M. ; MEINHARDT, C. . Comparação das Arquiteturas de Multiplicadores Baugh-Wooley e Radix-2 Booth na Tecnologia de 16nm. In: Iberchip - Iberchip Workshop, 2018, Puerto Vallarta.

BORBA, A. O. ; BORGES, D. M. ; MEINHARDT, C. ; ROSA, V. S. . Avaliação de Circuitos Aritméticos em Tecnologias Nanométricas. REVISTA JÚNIOR DE

INICIAÇÃO CIENTÍFICA EM CIÊNCIAS EXATAS E ENGENHARIA, v. 1, p. 9-18, 2018.

REFERÊNCIAS

ABRAHAM, S.; KAUR, S.; SINGH, S. Study of various high speed multipliers. **International Conference on Computer Communication and Informatics (ICCCI)**, [S.l.], p.1–5, Jan 2015.

ALEXANDER, S. A Review of Different Multipliers in Digital Circuits. **International Journal of MC Square Scientific Research**, [S.l.], v.4, n.1, 2012.

BAUGH, C. R.; WOOLEY, B. A. A Two's Complement Parallel Array Multiplication Algorithm. **IEEE Transactions on Computers**, [S.l.], v.C-22, n.12, p.1045–1047, Dec 1973.

BEDRIJ, O. J. Carry-Select Adder. **IRE Transactions on Electronic Computers**, [S.l.], v.EC-11, n.3, p.340–346, 1962.

BOOTH, A. D. A Signed Binary Multiplication Technique. **The Quarterly Journal of Mechanics and Applied Mathematics**, [S.l.], v.4, n.2, p.236–240, 1951.

BORBA, A.; BORGES, D.; MEINHARDT, C.; ROSA, V. Avaliação de Circuitos Aritméticos em Tecnologias Nanométricas. **ICCEEg**, [S.l.], 2018.

CAO, Y.; SATO, T.; ORSHANSKY, M.; SYLVESTER, D.; HU, C. New paradigm of predictive MOSFET and interconnect modeling for early circuit simulation. **Proceedings of the IEEE 2000 Custom Integrated Circuits Conference (Cat. No.00CH37044)**, [S.l.], p.201–204, May 2000.

CHANDRAKASAN, A.; SHENG, S.; BRODERSEN, R. Low-power CMOS digital design. **IEEE Journal of Solid-State Circuits**, [S.l.], v.27, n.4, p.473–484, Apr. 1992.

CHIDGUPKAR, P. D.; KARAD, M. T. The Implementation of Vedic Algorithms in Digital Signal Processing. **Global J. of Engg. Edu**, [S.l.], v.8, n.2, 2004.

DADDA, L. Some Schemes for Parallel Multipliers. **Alta Frequenza**, [S.l.], p.349–356, 1965.

ESMAEILZADEH, H.; BLEM, E.; AMANT, R. S.; SANKARALINGAM, K.; BURGER, D. Dark silicon and the end of multicore scaling. In: ANNUAL INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE (ISCA), 2011., 2011. **Anais...** [S.l.: s.n.], 2011. p.365–376.

FLOYD, T. **Sistemas Digitais: Fundamentos e Aplicações**. [S.l.]: BOOKMAN COMPANHIA ED, 2007.

GOEL, N.; GARG, L. Comparative Analysis of 4-bit CMOS Multipliers. **International Conference on VLSI, Communication and Instrumentation (ICVCI)**, [S.l.], 2011.

GUO, Y.; SUN, H.; KIMURA, S. Design of Power and Area Efficient Lower-Part-OR Approximate Multiplier. In: TENCON 2018 - 2018 IEEE REGION 10 CONFERENCE, 2018. **Anais...** [S.l.: s.n.], 2018. p.2110–2115.

GUPTA, V.; MOHAPATRA, D.; PARK, S. P.; RAGHUNATHAN, A.; ROY, K. IMPACT: IMPrecise adders for low-power approximate computing. In: IEEE/ACM INTERNATIONAL SYMPOSIUM ON LOW POWER ELECTRONICS AND DESIGN, 2011. **Anais...** [S.l.: s.n.], 2011. p.409–414.

GUPTA, V.; MOHAPATRA, D.; RAGHUNATHAN, A.; ROY, K. Low-Power Digital Signal Processing Using Approximate Adders. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.l.], v.32, n.1, p.124–137, 2013.

HAN, J.; ORSHANSKY, M. Approximate computing: An emerging paradigm for energy-efficient design. In: IN PROCEEDINGS OF 18TH IEEE EUROPEAN TEST SYMPOSIUM (ETS), 2013. **Anais...** [S.l.: s.n.], 2013.

HASHEMI, S.; BAHAR, R. I.; REDA, S. DRUM: A Dynamic Range Unbiased Multiplier for approximate applications. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN (ICCAD), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.418–425.

HUANG, W.; CAI, Z.; SHEN, S. **Comparison of Different Architectural Design Methods for Multiplier**. Disponível em: <http://venividiwiki.ee.virginia.edu/mediawiki/index.php/ClassECE6332Fall15GroupMultiplier>, Acesso em: 11 de dez. 2020.

J. M. RABAEY, A. C.; NIKOLIC, B. **Digital Integrated Circuits**. [S.l.]: Prentice Hall Publications, 2003.

K. BATHIJA, R.; S. MEENA, R.; SARKAR, S.; SAHU, R. Low Power High Speed 16x16 bit Multiplier using Vedic Mathematics. **International Journal of Computer Applications**, [S.l.], v.59, p.41–44, 12 2012.

KAHNG, A. B. The ITRS design technology and system drivers roadmap: Process and status. In: ACM/EDAC/IEEE DESIGN AUTOMATION CONFERENCE (DAC), 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.1–6.

KAHNG, A. B.; KANG, S. Accuracy-configurable adder for approximate arithmetic designs. In: DAC DESIGN AUTOMATION CONFERENCE 2012, 2012. **Anais...** [S.l.: s.n.], 2012. p.820–825.

KHATIBZADEH, A.; RAAHEMIFAR, K.; AHAMDI, M. A novel multiplier for high-speed applications. In: IEEE INTERNATIONAL SOC CONFERENCE, 2005., 2005. **Proceedings...** [S.l.: s.n.], 2005. p.305–308.

KIM, S.; KIM, Y. Energy-efficient hybrid adder design by using inexact lower bits adder. In: IEEE ASIA PACIFIC CONFERENCE ON CIRCUITS AND SYSTEMS (APCCAS), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.355–357.

KIM, S.; KIM, Y. High-performance and energy-efficient approximate multiplier for error-tolerant applications. In: INTERNATIONAL SOC DESIGN CONFERENCE (ISOCC), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.278–279.

KOGGE, P. M.; STONE, H. S. A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations. **IEEE Transactions on Computers**, [S.l.], v.C-22, n.8, p.786–793, 1973.

KOREN, I. **High-Speed Multiplication - III**. Disponível em: <https://slideplayer.com/slide/5101318/>, Acesso em 11 de dez. 2020.

LIANG, J.; HAN, J.; LOMBARDI, F. New Metrics for the Reliability of Approximate and Probabilistic Adders. **IEEE Transactions on Computers**, [S.l.], v.62, p.1760–1771, 2013.

LIN, J.; HWANG, Y.; SHEU, M.; HO, C. A Novel High-Speed and Energy Efficient 10-Transistor Full Adder Design. **IEEE Transactions on Circuits and Systems I: Regular Papers**, [S.l.], v.54, n.5, p.1050–1059, May 2007.

LIU, C.; HAN, J.; LOMBARDI, F. A low-power, high-performance approximate multiplier with configurable partial error recovery. In: DESIGN, AUTOMATION TEST IN EUROPE CONFERENCE EXHIBITION (DATE), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.1–4.

MACSORLEY, O. L. High-Speed Arithmetic in Binary Computers. **Proceedings of the IRE**, [S.l.], v.49, n.1, p.67–91, Jan 1961.

MAHDIANI, H. R.; AHMADI, A.; FAKHRAIE, S. M.; LUCAS, C. Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications. **IEEE Transactions on Circuits and Systems I: Regular Papers**, [S.l.], v.57, n.4, p.850–862, 2010.

MAHMOUD, H.; MAGDY, A. A 10-transistor low-power high-speed full adder cell. In: 2014, 1999. **Anais...** [S.l.: s.n.], 1999. v.1, p.43–46.

MANJUNATH; HARIKIRAN, V.; KOPPARAPU, M.; S, S.; K, S. Design and Implementation of 16x16 Modified Booth Multiplier. In: 2015. **Anais...** [S.l.: s.n.], 2015.

MISHRA, S. S.; KUMAR AGRAWAL, A.; K NAGARIA, R. A comparative performance analysis of various CMOS design techniques for XOR and XNOR circuits. **International Journal on Emerging Technologies**, [S.l.], v.1, 01 2010.

MOHANTY, P. An Efficient Baugh-Wooley Architecture for Signed and Unsigned Fast Multiplication. **NIET Journal of Engineering; Technology**, [S.l.], v.1, 2013.

MOMENI, A.; HAN, J.; MONTUSCHI, P.; LOMBARDI, F. Design and Analysis of Approximate Compressors for Multiplication. **IEEE Transactions on Computers**, [S.l.], v.64, n.4, p.984–994, 2015.

NARCHI, P.; KERUR, S. S.; NIDAGUNDI, J. C.; KITTUR, H. M.; A, G. V. Implementation of Vedic Multiplier for Digital Signal Processing. **IJCA Proceedings on International Conference on VLSI, Communications and Instrumentation (ICVCI)**, [S.l.], n.16, p.1–5, 2011. Full text available.

NAVI, K.; MAEEN, M.; FOROUTAN, V.; TIMARCHI, S.; KAVEHEI, O. A novel low-power full-adder cell for low voltage. **Integration, the VLSI journal**, [S.l.], v.42, n.4, p.457 – 467, 2009.

NGSPICE. **Ngspice circuit simulator**. Disponível em: <http://ngspice.sourceforge.net/>, Acesso em: 26 de nov. 2018.

PALEM, K.; LINGAMNENI, A. Ten years of building broken chips: The physics and engineering of inexact computing. **ACM Trans. Embed. Comput. Syst**, [S.l.], v.12, 2013.

PANDEY, A.; KARRI, M. R.; YADAV, P.; Y.B., N. K.; M.H., V. Design and Analysis of Approximate Multipliers for Error-Tolerant Applications. In: IEEE INTERNATIONAL SYMPOSIUM ON SMART ELECTRONIC SYSTEMS (ISES) (FORMERLY INIS), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.94–97.

PEDRONI, V. **Eletronica Digital Moderna e Vhdl**. [S.l.]: ELSEVIER EDITORA, 2010.

PINCKNEY, N.; SEWELL, K.; DRESLINSKI, R. G.; FICK, D.; MUDGE, T.; SYLVESTER, D.; BLAAUW, D. Assessing the performance limits of parallelized near-threshold computing. In: DAC DESIGN AUTOMATION CONFERENCE 2012, 2012. **Anais...** [S.l.: s.n.], 2012. p.1143–1148.

PRANAY, B. M.; JANDHYALA, S. Accuracy Configurable Modified Booth Multiplier Using Approximate Adders. In: IEEE INTERNATIONAL SYMPOSIUM ON NANO-ELECTRONIC AND INFORMATION SYSTEMS, 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.281–285.

QIAN, L.; WANG, C.; LIU, W.; LOMBARDI, F.; HAN, J. Design and evaluation of an approximate Wallace-Booth multiplier. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.1974–1977.

RAJMOHAN, V.; MAHESWRI, O. Design of Compact Baugh-Wooley Multiplier Using Reversible Logic. **Circuits and Systems**, [S.l.], v.7, p.1522–1529, 2016.

REDDY, K. M.; KUMAR, Y. B. N.; SHARMA, D.; VASANTHA, M. H. Low power, high speed error tolerant multiplier using approximate adders. In: INTERNATIONAL SYMPOSIUM ON VLSI DESIGN AND TEST, 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.1–6.

SABEETHA, S.; AJAYAN, J.; SHRIRAM, S.; VIVEK, K.; RAJESH, V. A study of performance comparison of digital multipliers using 22nm strained silicon technology. **2015 2nd International Conference on Electronics and Communication Systems (ICECS)**, [S.l.], p.180–184, Feb 2015.

SAHOO, B. C.; SAMANT, S. K. **Design and Power Estimation of Booth Multiplier Using Different Adder Architectures**. 2013. 56p. Electronics Communication Engineering — National Institute of Technology, Rourkela.

SATISH, D.; RAJU, B. R. A High Speed 16*16 Multiplier Based On Urdhva Tiryakbhyam Sutra. **IJSEAT**, [S.l.], v.1, n.5, 2013.

SHARMA, N.; SINDAL, R. Modified Booth Multiplier using Wallace Structure and Efficient Carry Select Adder. **International Journal of Computer Applications**, [S.l.], v.68, p.39–42, 04 2013.

SILVA, F. G. R. G. da; BUTZEN, P. F.; MEINHARDT, C. PVT variability analysis of FinFET and CMOS XOR circuits at 16nm. In: IEEE INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS AND SYSTEMS (ICECS), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.528–531.

SILVA, P. H. A.; MEINHARDT, C. Comparação de Topologias de Full Adders para Computação Aproximada. In: XI COMPUTER ON THE BEACH, 2020, Baln. Camboriú, SC, Brasil. **Anais...** [S.l.: s.n.], 2020.

SJALANDER, M.; LARSSON-EDEFORS, P. High-speed and low-power multipliers using the Baugh-Wooley algorithm and HPM reduction tree. **2008 15th IEEE International Conference on Electronics, Circuits and Systems**, [S.l.], p.33–36, 2008.

SJALANDER, M.; LARSSON-EDEFORS, P. The Case for HPM-Based Baugh-Wooley Multipliers. , [S.l.], 2008.

SOUDRIS, D.; PIGUET, C.; GOUTIS, C. **Design CMOS Circuits for Low Power**. [S.l.]: Springer, 2002.

STALLINGS, W. **Arquitetura e organização de computadores**. [S.l.]: Pearson Education Inc., 2010. v.8.

SUTHERLAND, I.; SPROULL, B.; HARRIS, D. **Logical Effort: Designing Fast CMOS Circuits**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

SWAMI, J.; MAHARAJA, S. B. K. T. **Vedic Mathematics**. Varanasi, India: Motilal Banarsidas, 1986.

SWEE, K. L. S.; HIUNG, L. H. Performance comparison review of 32-bit multiplier designs. **2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)**, [S.l.], v.2, p.836–841, June 2012.

WALLACE, C. S. A Suggestion for a Fast Multiplier. **IEEE Transactions on Electronic Computers**, [S.l.], v.EC-13, n.1, p.14–17, Feb 1964.

YANG, Z.; JAIN, A.; LIANG, J.; HAN, J.; LOMBARDI, F. Approximate XOR/XNOR-based adders for inexact computing. In: IEEE INTERNATIONAL CONFERENCE ON NANOTECHNOLOGY (IEEE-NANO 2013), 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.690–693.

ZEGHBROECK, B. V. **Principles of Semiconductor Devices**. Disponível em: http://ecee.colorado.edu/~bart/book/book/chapter7/ch7_7.htm >, *Acesso em* : 11dez.2020.

ZIMMERMANN, R.; FICHTNER, W. Low-power logic styles: CMOS versus pass-transistor logic. **IEEE Journal of Solid-State Circuits**, [S.l.], v.32, n.7, p.1079–1090, 1997.

APÊNDICE A REVISÃO BÁSICA DE LÓGICA COMBINACIONAL

A multiplicação é uma operação mais complexa que a adição e a subtração, tanto quando realizada a nível de software quanto a nível de hardware (STALLINGS, 2010). Partindo deste pressuposto, para que se possa compor um circuito multiplicador, são utilizadas portas lógicas com diferentes funções além de somadores. Nas próximas seções serão mostrados os componentes básicos mais relevantes na execução deste trabalho.

A.1 Portas Lógicas

Inversor

O inversor realiza a operação de inversão ou complementação. Esta operação é compreendida como a troca de um nível lógico para o nível oposto, ou seja, quando um nível alto é aplicado na entrada de um inversor, a saída mostra um nível baixo e quando um nível baixo é aplicado na entrada a saída apresenta um nível alto. O símbolo que representa esta função e a sua implementação em lógica complementar CMOS são mostrado na Figura 67. A Tabela 16 mostra a tabela verdade deste circuito (FLOYD, 2007).

Tabela 16: Tabela verdade do inversor

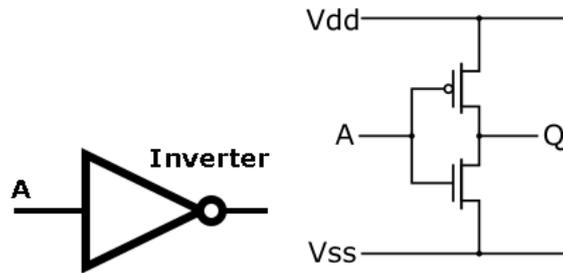
A	Q
0	1
1	0

Fonte: O autor

Porta NAND

A porta NAND produz uma saída em nível baixo apenas quando todas as entradas estiverem em nível alto. Quando alguma das saídas for nível baixo a saída será nível alto. A Figura 68.(a) mostra uma NAND de duas entradas, nesta porta a saída *Out* será nível baixo apenas quando as entradas *A* e *B* estiverem em nível baixo e será nível alto quando *A*, ou *B*, ou ambas estiverem em nível baixo (FLOYD, 2007). O funcionamento desta

Figura 67: (a) Símbolo do inversor (b) Implementação CMOS



Fonte: Adaptado de (FLOYD, 2007)

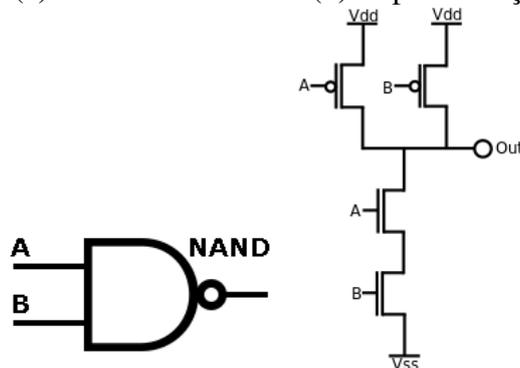
porta é descrito na Tabela 17. Na Figura 68(b) é exibido o circuito CMOS referente a essa função lógica.

Tabela 17: Tabela verdade da NAND

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

Fonte: O autor

Figura 68: (a) Símbolo da NAND (b) Implementação CMOS



Fonte: Adaptado de (FLOYD, 2007)

Porta NOR

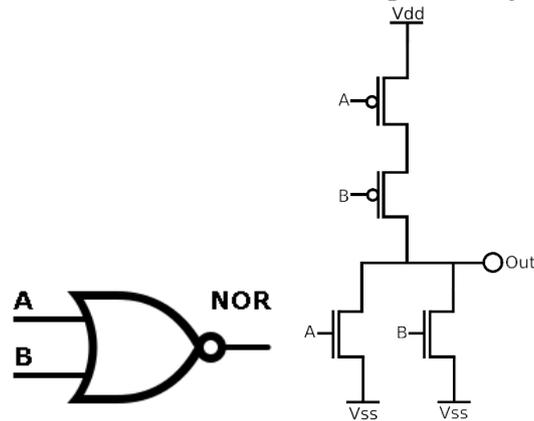
A porta NOR produz uma saída de nível baixo quando alguma de suas saídas for nível alto. A saída só terá nível alto quando todas as entradas tiverem nível baixo. A Figura mostra uma porta NOR de duas entradas, para esta, quando a entrada *A*, ou a entrada *B*, ou ambas forem nível alto, a saída *Out* será nível baixo. A saída só será nível alto quando as entradas *A* e *B* estiverem em nível baixo, como mostrado na Tabela 18 (FLOYD, 2007). Na Figura 69(b) podemos ver o circuito que descreve essa função em CMOS.

Tabela 18: Tabela verdade da NOR

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Fonte: O autor

Figura 69: (a) Símbolo da NOR (b) Implementação CMOS



Fonte: Adaptado de (FLOYD, 2007)

Portas AND

A porta AND realiza uma operação conhecida como multiplicação lógica. uma porta AND produz uma saída em nível alto somente quando todas as entradas apresentarem nível alto. Se pelo menos uma das entradas estiver em nível baixo, a saída será nível baixo. Para uma porta AND de duas entradas, a saída *Out* será nível alto apenas quando as entradas *A* e *B* também forem nível alto e será nível baixo quando uma ou ambas as entradas forem nível baixo. Uma porta AND pode ter qualquer número de entradas maior que dois, sendo que a saída só será nível alto quando ambas entradas forem nível alto. A Figura 70 apresenta o símbolo da porta apresentada e a Tabela 19 as combinações de entrada e saída para uma AND de duas entradas (FLOYD, 2007). A AND tem um comportamento contrário ao da NAND, logo, sua implementação em nível elétrico é feita utilizando esta última com sua respectiva saída ligada a um inversor. A saída do inversor produz o sinal final.

Figura 70: Símbolo da AND



Fonte: Adaptado de (FLOYD, 2007)

Tabela 19: Tabela verdade da AND

A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

Fonte: O autor

Porta OR

Uma porta OR apresenta um nível alto em sua saída quando pelo menos uma das entradas tiver em nível alto. Quando todas as entradas tiverem nível baixo, a saída será nível baixo. A Figura 71 mostra uma porta OR de duas entradas, nesta a saída *Out* será nível alto quando a entrada *A* ou a entrada *B* estiverem em nível alto, ou quando ambas estiverem em alto. A saída será nível baixo quando as duas entradas estiverem em nível baixo. O comportamento de uma porta OR de duas entradas é mostrado na Tabela 20 (FLOYD, 2007). A função OR é inversa a função NOR, apresenta anteriormente. Sendo assim, a implementação a nível de circuito é feita conectando a saída da NOR a um inversor e adotando a saída deste como sinal final.

Figura 71: Símbolo da OR



Fonte: Adaptado de (FLOYD, 2007)

Tabela 20: Tabela verdade da OR

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

Fonte: O autor

Porta OR exclusivo (XOR)

O símbolo padrão para a porta OR exclusivo é mostrado na Figura 72. A saída de uma porta XOR é nível alto apenas quando as duas entradas apresentam níveis lógicos diferentes. A porta XOR apresenta o seguinte comportamento: a saída *Out* tem nível alto quando a entrada *A* estiver em nível baixo e a entrada *B* em nível alto ou vice-versa. Quando as entradas estão em níveis iguais a saída *Out* mostra nível baixo. O funcionamento descrito

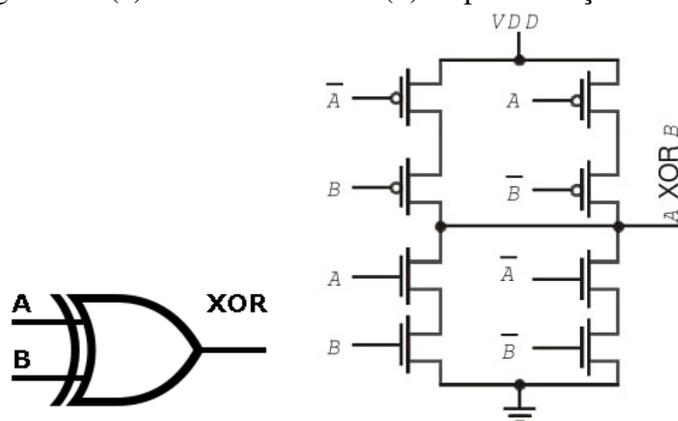
é mostrado na Tabela 21 (FLOYD, 2007). A função XOR tem diferentes implementações, uma das mais tradicionais é mostrada na Figura 72(b).

Tabela 21: Tabela verdade da XOR

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Fonte: O autor

Figura 72: (a) Símbolo da XOR (b) Implementação da XOR

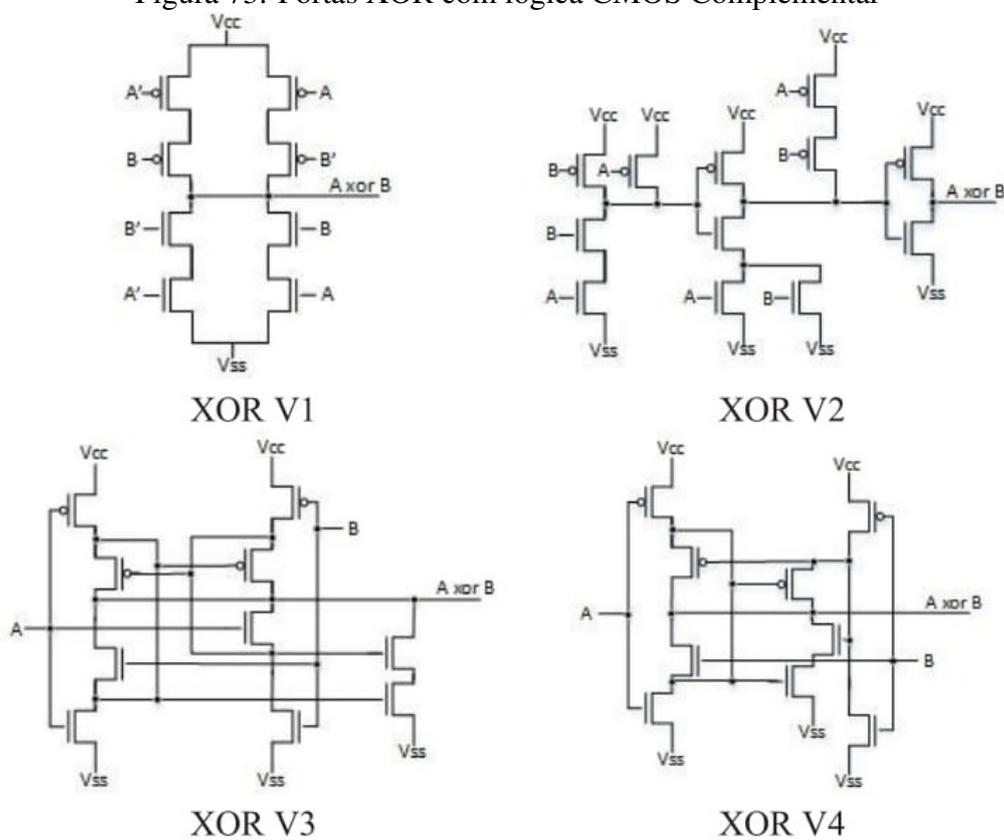


Fonte: Adaptado de (FLOYD, 2007), (MISHRA; KUMAR AGRAWAL; K NAGARIA, 2010)

As portas lógicas XOR são blocos fundamentais para a implementação de vários circuitos aritméticos como somadores, multiplicadores e compressores. O desempenho deste circuitos complexos é significativamente afetado pela performance individual de cada XOR incluída no mesmo.

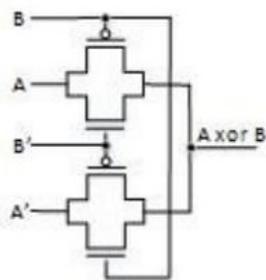
No estado da arte estão presentes diferente implementações para este componente. Assim sendo, é necessária uma escolha meticulosa da opção será empregada no projeto para que se possa obter um menor número de transistores, menor consumo de energia e um menor caminho crítico, além de um sinal sem distorções (MISHRA; KUMAR AGRAWAL; K NAGARIA, 2010). Nas Figuras 73 e 74 são mostradas diferentes implementações da porta lógica. As portas da primeira figura foram implementadas usando a lógica complementar CMOS enquanto as da segunda são baseadas em *Pass-Transistor Logic* (PTL). O método PTL é a melhor maneira de implementar circuitos para aplicação de baixo consumo.

Figura 73: Portas XOR com lógica CMOS Complementar

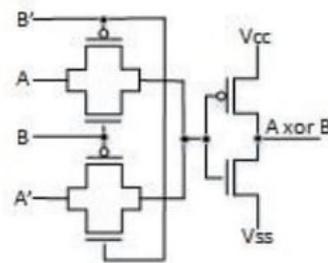


Fonte: (SILVA; BUTZEN; MEINHARDT, 2016)

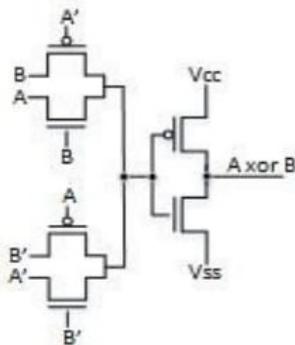
Figura 74: Portas XOR usando lógica PTL



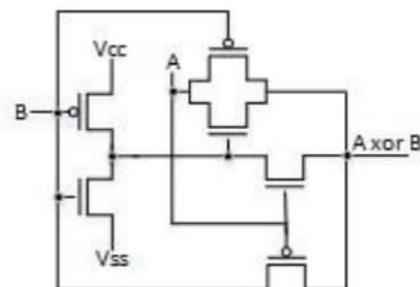
XOR V5



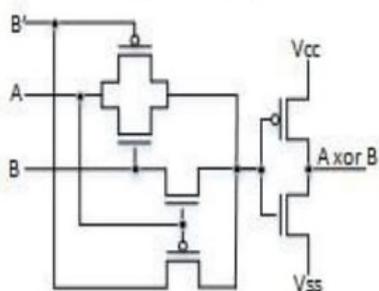
XOR V6



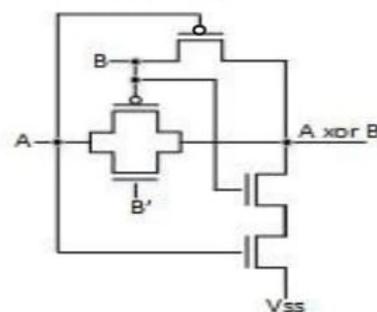
XOR V7



XOR V8



XOR V9



XOR V10

Fonte: (SILVA; BUTZEN; MEINHARDT, 2016)

Porta NOR exclusivo (XNOR)

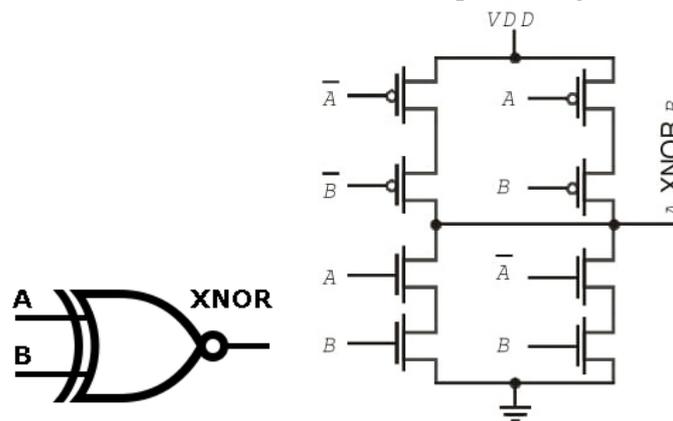
O símbolo da porta XNOR é semelhante ao da XOR, com exceção do pequeno círculo colocado na saída, como pode-se ver na Figura 75(a), este indica que a saída da XNOR é o oposto da saída da XOR. Para uma porta XNOR, a saída *Out* é nível baixo quando a entrada *A* for nível baixo e a entrada *B* for nível alto, ou vice-versa. Quando ambas entradas estão em nível igual a saída fica em nível alto. A Tabela 22 mostra esse comportamento(FLOYD, 2007). A Figura 75(b) mostra uma implementação desta função.

Tabela 22: Tabela verdade da XNOR

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	1

Fonte: O autor

Figura 75: (a) Símbolo da XNOR (b) Implementação da XNOR



Fonte: Adaptado de (FLOYD, 2007), (MISHRA; KUMAR AGRAWAL; K NAGARIA, 2010)

APÊNDICE B DESCRIÇÕES SPICE

Figura 76: Descrição SPICE do multiplicador Array (Parte 1)

```
base.txt [3]
1 *multiplicador array
2 .include library.cir
3 .include 16nm_HP.pm
4 *fonte de tensao-----
5 Vvdd vdd gnd 0.7
6 Vvdd2 vdd2 gnd 0.7
7 *entradas-----
8 .include fontes.txt
9 Xinva0 f0in fonte0 vdd2 gnd INV
10 Xinva1 f1in fonte1 vdd2 gnd INV
11 Xinva2 f2in fonte2 vdd2 gnd INV
12 Xinva3 f3in fonte3 vdd2 gnd INV
13 Xinva4 f4in fonte4 vdd2 gnd INV
14 Xinva5 f5in fonte5 vdd2 gnd INV
15 Xinva6 f6in fonte6 vdd2 gnd INV
16 Xinva7 f7in fonte7 vdd2 gnd INV
17 Xinvb0 fonte0 f0 vdd2 gnd INV
18 Xinvb1 fonte1 f1 vdd2 gnd INV
19 Xinvb2 fonte2 f2 vdd2 gnd INV
20 Xinvb3 fonte3 f3 vdd2 gnd INV
21 Xinvb4 fonte4 f4 vdd2 gnd INV
22 Xinvb5 fonte5 f5 vdd2 gnd INV
23 Xinvb6 fonte6 f6 vdd2 gnd INV
24 Xinvb7 fonte7 f7 vdd2 gnd INV
25 *LINHA0-----
26 *---f3=a0
27 Xand00 f3 f7 p0 vdd gnd AND
28 Xand01 f3 f6 P0_1 vdd gnd AND
29 Xand02 f3 f5 P0_2 vdd gnd AND
30 Xand03 f3 f4 P0_3 vdd gnd AND
31 *LINHA1-----
32 *---f2=a1
33 Xand10 f2 f7 P1_0 vdd gnd AND
34 Xand11 f2 f6 P1_1 vdd gnd AND
35 Xand12 f2 f5 P1_2 vdd gnd AND
36 Xand13 f2 f4 P1_3 vdd gnd AND
37 *SOMADORES0-----
38 Xha00 P1_0 P0_1 p1 cout00 vdd gnd HA
39 Xfa01 cout00 P1_1 P0_2 soma01 cout01 vdd gnd FA_CMOS
40 Xfa02 cout01 P1_2 P0_3 soma02 cout02 vdd gnd FA_CMOS
41 Xha03 P1_3 cout02 soma03 cout03 vdd gnd HA
42 *LINHA2-----
43 *---f1=a2
```

Fonte: o autor

Figura 77: Descrição SPICE do multiplicador Array (Parte 2)

```

44 Xand20 f1 f7 P2_0 vdd gnd AND
45 Xand21 f1 f6 P2_1 vdd gnd AND
46 Xand22 f1 f5 P2_2 vdd gnd AND
47 Xand23 f1 f4 P2_3 vdd gnd AND
48 *SOMADORES1-----
49 Xha10 P2_0 soma01 p2 cout10 vdd gnd HA
50 Xfa11 cout10 P2_1 soma02 soma11 cout11 vdd gnd FA_CMOS
51 Xfa12 cout11 P2_2 soma03 soma12 cout12 vdd gnd FA_CMOS
52 Xfa13 cout12 cout03 P2_3 soma13 cout13 vdd gnd FA_CMOS
53 *LINHA3-----
54 *----f0=a3
55 Xand30 f0 f7 P3_0 vdd gnd AND
56 Xand31 f0 f6 P3_1 vdd gnd AND
57 Xand32 f0 f5 P3_2 vdd gnd AND
58 Xand33 f0 f4 P3_3 vdd gnd AND
59 *SOMADORES2-----
60 Xha20 P3_0 soma11 p3 cout20 vdd gnd HA
61 Xfa21 cout20 P3_1 soma12 p4 cout21 vdd gnd FA_CMOS
62 Xfa22 cout21 P3_2 soma13 p5 cout22 vdd gnd FA_CMOS
63 Xfa23 cout22 P3_3 cout13 p6 p7 vdd gnd FA_CMOS
64 *----saidas-----
65 Cload0 p0 gnd 1f
66 Cload1 p1 gnd 1f
67 Cload2 p2 gnd 1f
68 Cload3 p3 gnd 1f
69 Cload4 p4 gnd 1f
70 Cload5 p5 gnd 1f
71 Cload6 p6 gnd 1f
72 Cload7 p7 gnd 1f
73 *simulacao-----
74 .tran 0.001 4n
75 *measure-----
76 .include measure.txt
77 .end
78

```

spice file length : 2.569 lines : 78 Ln : 58 Col : 34 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Fonte: o autor

Figura 78: Descrição SPICE do multiplicador Baugh-Wooley

```

1 *multiplicador Baugh-Wooley
2 .include biblioteca.cir
3 .include l6nm_HP.pm
4 *fonte de tensao-----
5 Vvdd vdd gnd 0.7
6 Vvdd2 vdd2 gnd 0.7
7 *carrys iniciais-----
8 Vc0 c0 gnd 0
9 Vc1 c1 gnd 0
10 Vc2 c2 gnd 0
11 Vc3 c3 gnd 0
12 *sommas iniciais no eixo x-----
13 Vsi1 si1 gnd 0
14 Vsi2 si2 gnd 0
15 Vsi3 si3 gnd 0
16 Vsi4 si4 gnd 0
17 *sommas iniciais no eixo y-----
18 Vs1 s1 gnd 0
19 Vs2 s2 gnd 0
20 Vs3 s3 gnd 0
21 Vs4 s4 gnd 0.7
22 Vs5 s5 gnd 0.7
23 *valores a serem multiplicados-----
24 .include fontes.txt
25 Xinva0 f0in a0 vdd2 gnd INV
26 Xinva1 f1in a1 vdd2 gnd INV
27 Xinva2 f2in a2 vdd2 gnd INV
28 Xinva3 f3in a3 vdd2 gnd INV
29 Xinva4 f4in a4 vdd2 gnd INV
30 Xinva5 f5in a5 vdd2 gnd INV
31 Xinva6 f6in a6 vdd2 gnd INV
32 Xinva7 f7in a7 vdd2 gnd INV
33 Xinvb0 a0 f0 vdd2 gnd INV
34 Xinvb1 a1 f1 vdd2 gnd INV
35 Xinvb2 a2 f2 vdd2 gnd INV
36 Xinvb3 a3 f3 vdd2 gnd INV
37 Xinvb4 a4 f4 vdd2 gnd INV
38 Xinvb5 a5 f5 vdd2 gnd INV
39 Xinvb6 a6 f6 vdd2 gnd INV
40 Xinvb7 a7 f7 vdd2 gnd INV
41 *linha 1-----
42 Xcelula11 c0 si1 f0 f7 soma11 cout11 vdd gnd grayCell
43 Xcelula12 c1 si2 f1 f7 soma12 cout12 vdd gnd whiteCell
44 Xcelula13 c2 si3 f2 f7 soma13 cout13 vdd gnd whiteCell
45 Xcelula14 c3 si4 f3 f7 p0 cout14 vdd gnd whiteCell
46 *linha 2-----
47 Xcelula21 cout11 sj1 f0 f6 soma21 cout21 vdd gnd grayCell
48 Xcelula22 cout12 soma11 f1 f6 soma22 cout22 vdd gnd whiteCell
49 Xcelula23 cout13 soma12 f2 f6 soma23 cout23 vdd gnd whiteCell
50 Xcelula24 cout14 soma13 f3 f6 p1 cout24 vdd gnd whiteCell
51 *linha 3-----
52 Xcelula31 cout21 sj2 f0 f5 soma31 cout31 vdd gnd grayCell
53 Xcelula32 cout22 soma21 f1 f5 soma32 cout32 vdd gnd whiteCell
54 Xcelula33 cout23 soma22 f2 f5 soma33 cout33 vdd gnd whiteCell
55 Xcelula34 cout24 soma23 f3 f5 p2 cout34 vdd gnd whiteCell
56 *linha 4-----
57 Xcelula41 cout31 sj3 f0 f4 soma41 cout41 vdd gnd whiteCell
58 Xcelula42 cout32 soma31 f1 f4 soma42 cout42 vdd gnd grayCell
59 Xcelula43 cout33 soma32 f2 f4 soma43 cout43 vdd gnd grayCell
60 Xcelula44 cout34 soma33 f3 f4 p3 cout44 vdd gnd grayCell
61 *linha 5-----
62 X_FA_1 cout41 sj4 cout52 p7 cout51 vdd gnd FA_CMOS
63 X_FA_2 cout42 soma41 cout53 p6 cout52 vdd gnd FA_CMOS
64 X_FA_3 cout43 soma42 cout54 p5 cout53 vdd gnd FA_CMOS
65 X_FA_4 cout44 soma43 sj5 p4 cout54 vdd gnd FA_CMOS
66 *saidas-----
67 cload p0 gnd 1f
68 cload1 p1 gnd 1f
69 cload2 p2 gnd 1f
70 cload3 p3 gnd 1f
71 cload4 p4 gnd 1f
72 cload5 p5 gnd 1f
73 cload6 p6 gnd 1f
74 cload7 p7 gnd 1f
75 *simulacao-----
76 .tran 0.001 4n
77 .include measure.txt
78 .end

```

Fonte: o autor

Figura 79: Descrição SPICE do multiplicador Booth

```

1  *multiplicador booth|
2
3  .include circuitos.cir
4  .include 16nm_HP.pm
5
6  *-----fonte de tensao-----
7  Vvdd vdd gnd 0.7
8  Vvdd2 vdd2 gnd 0.7
9  *-----iniciais-----
10 Va a gnd 0
11 Vp1 pin1 gnd 0
12 Vp2 pin2 gnd 0
13 Vp3 pin3 gnd 0
14 Vp4 pin4 gnd 0
15 *-----valores-----
16 .include fontes.txt
17
18 Xinva0 f0in a0 vdd2 gnd INV
19 Xinva1 flin a1 vdd2 gnd INV
20 Xinva2 f2in a2 vdd2 gnd INV
21 Xinva3 f3in a3 vdd2 gnd INV
22 Xinva4 f4in a4 vdd2 gnd INV
23 Xinva5 f5in a5 vdd2 gnd INV
24 Xinva6 f6in a6 vdd2 gnd INV
25 Xinva7 f7in a7 vdd2 gnd INV
26
27 Xinvb0 a0 f0 vdd2 gnd INV
28 Xinvb1 a1 f1 vdd2 gnd INV
29 Xinvb2 a2 f2 vdd2 gnd INV
30 Xinvb3 a3 f3 vdd2 gnd INV
31 Xinvb4 a4 f4 vdd2 gnd INV
32 Xinvb5 a5 f5 vdd2 gnd INV
33 Xinvb6 a6 f6 vdd2 gnd INV
34 Xinvb7 a7 f7 vdd2 gnd INV
35 *-----linha1-----
36 Xxor1 a f3 enb1 vdd gnd XOR
37 Xand1 f3 enb1 c01 vdd gnd AND
38
39 Xcas11 c01 f7 f3 enb1 pin1 p0 c11 vdd gnd CAS
40 Xcas12 c11 f6 f3 enb1 pin2 sum12 c12 vdd gnd CAS
41 Xcas13 c12 f5 f3 enb1 pin3 sum13 c13 vdd gnd CAS
42 Xcas14 c13 f4 f3 enb1 pin4 sum14 c14 vdd gnd CAS
43 *-----linha2-----
44 Xxor2 f3 f2 enb2 vdd gnd XOR
45 Xand2 f2 enb2 c02 vdd gnd AND
46
47 Xcas21 c02 f7 f2 enb2 sum12 p1 c21 vdd gnd CAS
48 Xcas22 c21 f6 f2 enb2 sum13 sum22 c22 vdd gnd CAS
49 Xcas23 c22 f5 f2 enb2 sum14 sum23 c23 vdd gnd CAS
50 Xcas24 c23 f4 f2 enb2 sum14 sum24 c24 vdd gnd CAS
51 *-----linha3-----
52 Xxor3 f2 f1 enb3 vdd gnd XOR
53 Xand3 f1 enb3 c03 vdd gnd AND
54
55 Xcas31 c03 f7 f1 enb3 sum22 p2 c31 vdd gnd CAS
56 Xcas32 c31 f6 f1 enb3 sum23 sum32 c32 vdd gnd CAS
57 Xcas33 c32 f5 f1 enb3 sum24 sum33 c33 vdd gnd CAS
58 Xcas34 c33 f4 f1 enb3 sum24 sum34 c34 vdd gnd CAS
59 *-----linha4-----
60 Xxor4 f1 f0 enb4 vdd gnd XOR
61 Xand4 f0 enb4 c04 vdd gnd AND
62
63 Xcas41 c04 f7 f0 enb4 sum32 p3 c41 vdd gnd CAS
64 Xcas42 c41 f6 f0 enb4 sum33 p4 c42 vdd gnd CAS
65 Xcas43 c42 f5 f0 enb4 sum34 p5 c43 vdd gnd CAS
66 Xcas44 c43 f4 f0 enb4 sum34 p6 c44 vdd gnd CAS
67 *-----saidas-----
68 Cload0 p0 gnd 1f
69 Cload1 p1 gnd 1f
70 Cload2 p2 gnd 1f
71 Cload3 p3 gnd 1f
72 Cload4 p4 gnd 1f
73 Cload5 p5 gnd 1f
74 Cload6 p6 gnd 1f
75 *simulacao
76 .tran 0.001 4n
77
78 *measure
79 .include measure.txt
80
81 .end
82

```

spice file length: 2429 lines: 82 Ln: 1 Col: 21 Sel: 0|0 Windows (CR LF) UTF-8 INS

Fonte: o autor

Figura 80: Descrição SPICE do multiplicador Vedic

```

1  *multiplicador vedic
2
3  .include library.cir
4  .include 16nm_HP.pm
5  Vvdd vdd gnd 0.7
6  Vvdd2 vdd2 gnd 0.7
7  Vcin cin gnd 0
8
9  *descricao de fontes
10 .include fontes.txt
11 =====BARREIRA DE INVERSORES=====
12 Xinv0 f0in fonte0 vdd2 gnd INV
13 Xinv1 f1in fonte1 vdd2 gnd INV
14 Xinv2 f2in fonte2 vdd2 gnd INV
15 Xinv3 f3in fonte3 vdd2 gnd INV
16 Xinv4 f4in fonte4 vdd2 gnd INV
17 Xinv5 f5in fonte5 vdd2 gnd INV
18 Xinv6 f6in fonte6 vdd2 gnd INV
19 Xinv7 f7in fonte7 vdd2 gnd INV
20 Xinvb0 fonte0 f0 vdd2 gnd INV
21 Xinvb1 fonte1 f1 vdd2 gnd INV
22 Xinvb2 fonte2 f2 vdd2 gnd INV
23 Xinvb3 fonte3 f3 vdd2 gnd INV
24 Xinvb4 fonte4 f4 vdd2 gnd INV
25 Xinvb5 fonte5 f5 vdd2 gnd INV
26 Xinvb6 fonte6 f6 vdd2 gnd INV
27 Xinvb7 fonte7 f7 vdd2 gnd INV
28 =====
29 Xv1 f3 f2 f7 f6 p0 p1 p13 p14 vdd gnd VEDIC2
30 Xv2 f1 f0 f7 f6 p21 p22 p23 p24 vdd gnd VEDIC2
31 Xv3 f3 f2 f5 f4 p31 p32 p33 p34 vdd gnd VEDIC2
32 Xv4 f1 f0 f5 f4 p41 p42 p43 p44 vdd gnd VEDIC2
33
34 Xcla1 cin p21 p22 p23 p24 p31 p32 p33 p34 p51 p52 p53 p54 p55 pg gg vdd gnd CLA4
35 Xcla2 cin p51 p52 p53 p54 p13 p14 p41 p42 p2 p3 p4 p5 p65 pg gg vdd gnd CLA4
36
37 Xor p65 p55 p71 vdd gnd OR
38 Xha1 p71 p43 p6 p81 vdd gnd HA
39 Xha2 p81 p44 p7 cout vdd gnd HA
40 =====CAPACITORES DE SAIDA=====
41 cload0 p0 gnd 1f
42 cload1 p1 gnd 1f
43 cload2 p2 gnd 1f
44 cload3 p3 gnd 1f
45 cload4 p4 gnd 1f
46 cload5 p5 gnd 1f
47 cload6 p6 gnd 1f
48 cload7 p7 gnd 1f
49
50 *====simulacao====
51 .tran 0.001 4n
52 .include measure.txt
53 .end

```

spice file length: 1.602 lines: 53 Ln: 1 Col: 1 Sel: 0|0 Windows (CR LF) UTF-8 INS

Fonte: o autor