



UFSM

Dissertação de Mestrado

**PROTOTIPAÇÃO E ANÁLISE DE CIRCUITOS
MULTIPLICADORES ARRAY DE BAIXO CONSUMO**

Leonardo Londero de Oliveira

PPGEE

Santa Maria, RS, Brasil

2005

**PROTOTIPAÇÃO E ANÁLISE DE CIRCUITOS
MULTIPLICADORES ARRAY DE BAIXO CONSUMO**

**por
Leonardo Londero de Oliveira**

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia Elétrica, Área de Concentração em Processamento de Energia, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Engenharia Elétrica**.

PPGEE

Santa Maria, RS, Brasil

2005

**Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica**

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

**PROTOTIPAÇÃO E ANÁLISE DE CIRCUITOS
MULTIPLICADORES ARRAY DE BAIXO CONSUMO**

elaborada por
Leonardo Londero de Oliveira

como requisito parcial para obtenção do grau de
Mestre em Engenharia Elétrica

COMISSÃO EXAMINADORA:

Prof. Dr. João Baptista dos Santos Martins
(Presidente/Orientador)

Prof. Dr. Eduardo Antonio César da Costa
(Co-orientador)

Prof. Dr. Sérgio Bampi

Santa Maria, 23 de março de 2005

O48p Oliveira, Leonardo Londero de
Prototipação e análise de circuitos multiplicadores Array
de baixo consumo / por Leonardo Londero de Oliveira. –
2005.

80 f. : il.

Dissertação (mestrado) – Universidade Federal de
Santa Maria, Centro de Tecnologia, Programa de Pós-
Graduação em Engenharia Elétrica, RS, 2005

Orientador: João Baptista dos Santos Martins

Co-orientador: Eduardo Antonio César da Costa

1. Engenharia elétrica 2. Energia elétrica 3. Circuitos
elétricos 4. Circuitos integrados 5. Circuitos
multiplicadores 6. Consumo de potência 7. Arquitetura
de multiplicadores 8. Protótipos I. Martins, João Baptista
dos Santos, orientador II. Costa, Eduardo Antonio César
da, co-orientador III. Título.

CDU 621.3.049.77

Maristela Eckhardt - CRB-10/737
Biblioteca Central da UFSM

Ao meu pai

Dioní Jorge Costa de Oliveira

e à minha mãe

Maria Oneida Londero de Oliveira

Agradecimentos

Reservo este espaço do meu trabalho para lembrar as pessoas que foram essenciais para a conclusão de mais esta etapa da minha vida.

A todos os amigos do Grupo de Microeletrônica (GMICRO). A convivência neste ambiente de trabalho foi decisiva para o desenvolvimento desta dissertação. Em especial quero lembrar os amigos Cesar Prior e Diego Salengue pelos valiosos suportes computacionais que resultaram na fabricação dos *chips* e companheirismo. Ao forte apoio recebido durante a minha passagem em Lisboa pela Ana de Jesus e o Doutor Miguel Silveira.

Aos Doutores do GMICRO André Aita, Cesar Rodrigues e João Baptista por proporcionarem as condições necessárias para a execução das atividades de pesquisa e, apostarem no potencial da UFSM, enfrentando diversos obstáculos para a criação e manutenção de um Grupo de Pesquisa. Em especial ao meu Orientador João Baptista pelo apoio oferecido nestes anos e por acreditar na minha capacidade na realização deste trabalho.

Ao amigo e Co-orientador Doutor Eduardo Costa, que mesmo não residindo em Santa Maria, fez-se presente de maneira

indispensável na construção deste projeto. Quero lembrar ainda os Doutores Sergio Bampi e José Monteiro também pelas valiosas contribuições e trabalhos desenvolvidos em conjunto.

Ao CNPq e a FAPERGS pelo apoio financeiro, fundamental para a realização deste trabalho.

À minha família pelo apoio irrestrito durante todos estes anos de estudo. Em especial aos meus pais pelos anos de compreensão e carinho, fundamentais nesta caminhada.

À minha companheira Olga Rohde por trilhar este caminho ao meu lado, com paciência, carinho e incentivo à minha profissão.

Sumário

Sumário	i
Lista de Tabelas	iii
Lista de Figuras	iv
RESUMO	v
ABSTRACT	vi
1. Introdução	1
1.1. Motivação	2
1.2. Objetivos	4
1.3. Metodologia	5
1.4. Contribuições do Trabalho	6
1.5. Organização do Trabalho	6
2. Potência e Atraso em Circuitos CMOS	8
2.1. Consumo de Potência em Circuitos CMOS	9
2.2. Consumo de Potência de Curto-Circuito	11
2.3. Consumo de Potência Estática	13
2.4. Consumo de Potência Dinâmica	15
2.5. Glitching	16
2.6. Atraso em Circuitos CMOS	17
2.7. Análise Estática de Atraso	18
2.8. Análise de Folga Temporal	18
2.9. Análise Dinâmica de Atraso	19
2.10. Resumo	20
3. Arquiteturas de Multiplicadores	21
3.1. Multiplicadores Array na Base 2^m em Complemento de 2	22
3.2. Multiplicador Binário em Complemento de 2	22
3.3. Multiplicador Híbrido em Complemento de 2	24
3.4. Introdução ao código Híbrido	25
3.5. Operação de multiplicação em código Híbrido	27
3.6. Multiplicador Booth Modificado	28
3.7. Resumo	31
4. Síntese Física das Arquiteturas	32
4.1. Fluxo de Projeto para a Síntese dos Multiplicadores	34
4.2. Resumo	44

5.	Estratégia de Verificação e Análise dos Resultados	45
5.1.	Estrutura de Projeto de Testes.....	47
5.2.	Desenvolvimento do Fluxo de Projeto para a Obtenção dos Parâmetros de Área, Atraso e Consumo de Potência.....	55
5.3.	Resultados de Consumo de Potência	58
5.4.	Resumo.....	61
6.	Conclusões e Trabalhos Futuros	63
7.	Bibliografia	66
	Anexo 1	72
	Anexo 2	74
	Anexo 3	79

Lista de Tabelas

TABELA 1 – REPRESENTAÇÕES DOS CÓDIGOS BINÁRIO, HÍBRIDO ($M=2$) E GRAY	25
TABELA 2 – NÚMERO DE TRANSIÇÕES PARA OS CÓDIGOS BINÁRIO, HÍBRIDO ($M=2$) E GRAY	26
TABELA 3 – RESULTADOS DE ÁREA E ATRASO PARA A PRIMEIRA SÍNTESE DOS MULTIPLICADORES	42
TABELA 4 – RESULTADOS DE ÁREA E ATRASO PARA A SEGUNDA SÍNTESE DOS MULTIPLICADORES	43
TABELA 5 – RESULTADOS EXPERIMENTAIS DE DISSIPACÃO DE POTÊNCIA PARA OS CHIPS MULTIPLICADORES	58
TABELA 6 – DISSIPACÃO DE POTÊNCIA PARA DIFERENTES PROFUNDIDADES DE VETORES	61

Lista de Figuras

FIGURA 1 – CORRENTE DE CURTO-CIRCUITO DURANTE TRANSIENTES	12
FIGURA 2 – REPRESENTAÇÃO DAS FONTES DE CORRENTE ESTÁTICA	14
FIGURA 3 – PORTAS XOR EM CASCATA	16
FIGURA 4 – EXEMPLO DE UMA MULTIPLICAÇÃO NA BASE 4 ($M=2$) EM COMPLEMENTO DE 2 UTILIZANDO O CÓDIGO BINÁRIO	23
FIGURA 5 – ARQUITETURA DO MULTIPLICADOR ARRAY BINÁRIO DE 8 BITS EM COMPLEMENTO DE 2	24
FIGURA 6 – CONVERSÃO ENTRE OS CÓDIGOS BINÁRIO E HÍBRIDO	27
FIGURA 7 – EXEMPLO DE UMA MULTIPLICAÇÃO NA BASE 4 ($M=2$) EM COMPLEMENTO DE 2 UTILIZANDO O CÓDIGO HÍBRIDO	28
FIGURA 8 – EXEMPLO DE UMA MULTIPLICAÇÃO DE 8 BITS UTILIZANDO O ALGORITMO BOOTH MODIFICADO	29
FIGURA 9 – ARQUITETURA DE 8 BITS DO MULTIPLICADOR BOOTH MODIFICADO	31
FIGURA 10 – DIAGRAMA REPRESENTADO AS VÁRIAS ETAPAS PARA OBTENÇÃO E VERIFICAÇÃO DAS ARQUITETURAS DE MULTIPLICADORES	33
FIGURA 11 – LAYOUT DO CIRCUITO MULTIPLICADOR ARRAY BINÁRIO DE 8 BITS	39
FIGURA 12 – LAYOUT DO CIRCUITO MULTIPLICADOR ARRAY HÍBRIDO DE 8 BITS	39
FIGURA 13 – LAYOUT DO CIRCUITO MULTIPLICADOR BOOTH MODIFICADO DE 8 BITS	40
FIGURA 14 – FOTO DO PROTÓTIPO DO CIRCUITO MULTIPLICADOR ARRAY BINÁRIO DE 8 BITS	40
FIGURA 15 – FOTO DO PROTÓTIPO DO CIRCUITO MULTIPLICADOR ARRAY HÍBRIDO DE 8 BITS	41
FIGURA 16 – FOTO DO PROTÓTIPO DO CIRCUITO MULTIPLICADOR BOOTH DE 8 BITS	41
FIGURA 17 – DIAGRAMA DOS VETORES DE TESTE UTILIZADOS	49
FIGURA 18 – REPRESENTAÇÃO DE VETORES RANDÔMICOS EM FORMATO SLS	49
FIGURA 19 – EXEMPLO DE LINHA DE COMANDOS DO PROGRAMA GERAVET	50
FIGURA 20 – PLACA DE PERIFÉRICOS DIO2	52
FIGURA 21 – DIAGRAMA DE PORTAS DE ENTRADA E SAÍDA DO PROGRAMA EM VHDL PARA INSERÇÃO DOS VETORES DE TESTE	52
FIGURA 22 – VETORES RANDÔMICOS RESULTANTES DE SIMULAÇÃO ELÉTRICA COM O PROGRAMA ELDO	53
FIGURA 23 – VETORES RANDÔMICOS RESULTANTES DA SIMULAÇÃO COM O PROGRAMA MODEL SIM	53
FIGURA 24 – TELA DO OSCIOSCÓPIO DIGITAL COM A SAÍDA DA FPGA	54
FIGURA 25 – ESQUEMA DE MONTAGEM DOS TESTES EXPERIMENTAIS	58
FIGURA 26 – COMPARAÇÃO DE VELOCIDADE	77
FIGURA 27 – COMPARAÇÃO DE PRECISÃO E CAPACIDADE	77
FIGURA 28 – LAYOUT DO CIRCUITO MULTIPLICADOR ARRAY BINÁRIO DE 8 BITS PARA A SEGUNDA SÍNTESE	79
FIGURA 29 – LAYOUT DO CIRCUITO MULTIPLICADOR ARRAY HÍBRIDO DE 8 BITS PARA A SEGUNDA SÍNTESE	79
FIGURA 30 – LAYOUT DO CIRCUITO MULTIPLICADOR BOOTH MODIFICADO DE 8 BITS PARA A SEGUNDA SÍNTESE	80

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Santa Maria, RS, Brasil

PROTOTIPAÇÃO E ANÁLISE DE CIRCUITOS MULTIPLICADORES ARRAY DE BAIXO CONSUMO

AUTOR: LEONARDO LONDERO DE OLIVEIRA
ORIENTADOR: JOÃO BAPTISTA DOS SANTOS MARTINS
CO-ORIENTADOR: EDUARDO ANTONIO CÉSAR DA COSTA
Data e Local de Defesa: Santa Maria, 23 de março de 2005.

Este trabalho apresenta a prototipação e análise de novas arquiteturas de circuitos multiplicadores digitais sob o ponto de vista físico. São analisados circuitos recentemente apresentados no meio científico e comparados com o estado da arte. As novas arquiteturas efetuam operações de multiplicação com sinal e mantêm a mesma regularidade de um multiplicador *array* convencional. As arquiteturas podem operar com números na base 2^m , onde m é o número de bits, o que permite a redução do número de linhas de produtos parciais, tendo-se desta forma, ganhos significativos em desempenho e redução do consumo de potência. Tais características, tornam-se um fator atrativo para as implementações dos circuitos multiplicadores no nível físico.

Para a maioria dos sistemas, testes de funcionalidade envolvem a necessidade de provar que o circuito é funcionalmente equivalente a uma determinada especificação. Estes devem ser usualmente os primeiros testes que um projetista deve construir como parte do processo de desenvolvimento do circuito. Desta forma, também faz parte deste trabalho o desenvolvimento de um fluxo de análise dos circuitos. Este fluxo deve percorrer desde os testes funcionais, passando por etapas de síntese física em nível de transistores e em FPGA, até a prototipação em Silício das arquiteturas. Neste trabalho, as arquiteturas de multiplicadores foram submetidas aos mesmos vetores de teste em todas as etapas de projeto, que envolvem desde a verificação da funcionalidade lógica, até a extração do consumo de potência no nível físico.

Palavras-chave: consumo de potência, síntese física, prototipação, teste de funcionalidade.

ABSTRACT

*Master Dissertation
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Santa Maria, RS, Brasil*

PROTOTIPAÇÃO E ANÁLISE DE CIRCUITOS MULTIPLICADORES ARRAY DE BAIXO CONSUMO

(PROTOTYPING AND ANALYSES OF LOW POWER ARRAY MULTIPLIER CIRCUITS)

AUTHOR: LEONARDO LONDERO DE OLIVEIRA

SUPERVISOR: JOÃO BAPTISTA DOS SANTOS MARTINS

CO-SUPERVISOR: EDUARDO ANTONIO CÉSAR DA COSTA

Date and Local: March 23TH of 2005, Santa Maria.

This work presents the prototyping and analysis of new multiplier architectures under the physical level of abstraction. Circuits recently presented in the academic community are analyzed and compared against the state of the art. The new architectures operate on signed multiplication and maintain the pure form of an array multiplier. These architectures are extended for radix-2^m encoding, where m is the number of the bits, which leads to a reduction of the number of partial lines. The proposed approach significantly improves the state of the art, enabling gains in performance and power consumption. Such aspects are attractive for the implementations of the new multipliers in the physical level.

For the most of the systems, functionality tests are used in order to verify if a circuit is functionally equivalent to a given specification. These types of tests have to be the first part of the development of the circuit. Thus, we have developed in this work a flow of the circuit analysis. This flow covers since the functional tests, passing through the stages of physical synthesis in transistors level and FPGA, until the prototyping in Silicon of the architectures. In this work, we have used the same test vectors in all the stages of the project which involves, since the verification of the logical functionality until the extraction of the power consumption in physical level.

Keywords: power dissipation, physical synthesis, prototyping, functionality tests.

1. INTRODUÇÃO

A rápida e crescente inovação em dispositivos VLSI (*Very Large Scale Integration*) impulsiona a utilização das tecnologias de fabricação aos seus limites, dentre as quais a verificação da dissipação de potência e o teste tornam-se os dois problemas de maior preocupação [12].

Desde o advento dos dispositivos eletrônicos portáteis, existe uma grande preocupação na indústria em manter estes dispositivos operando pelo maior tempo possível sob as mesmas condições. Entretanto, devido ao fato da tecnologia das baterias não ter evoluído significativamente nos últimos anos, existem muitos esforços no sentido de projetar circuitos que consumam cada vez menos energia e prolongar o tempo de vida útil das baterias. A redução da tensão de referência dos circuitos tem sido uma das alternativas encontradas. Entretanto esta alternativa esbarra em questões de desempenho do circuito, como a margem de ruído.

O crescente aumento na integração da tecnologia VLSI tem permitido que haja uma tendência para o desenvolvimento de aplicações móveis. Estes tipos de equipamentos portáteis, como *notebooks*, telefones celulares, *handhelds*, aparelhos biomédicos portáteis, entre outros, necessitam de um alto desempenho e baixo consumo de potência. Neste contexto, aspectos de baixa dissipação de potência, exigem a pesquisa de novas arquiteturas confiáveis que

possam ser testadas e fabricadas e que levem em consideração as necessidades de sistemas eletrônicos portáteis e de funções diversas [9].

Neste trabalho propõe-se a análise de desempenho e consumo de potência de novas arquiteturas de circuitos multiplicadores. As novas arquiteturas de multiplicadores foram propostas em [9] no nível lógico. No presente trabalho esta abordagem é estendida ao nível físico, onde são realizadas medições nos circuitos multiplicadores após as suas prototipações e posterior comparações de desempenho e consumo de potência entre as diferentes arquiteturas implementadas.

1.1. MOTIVAÇÃO

O principal objetivo da análise de potência em circuitos CMOS (*Complementary Metal Oxide Semiconductor*) é atingir níveis de precisão tão próximos quanto possíveis da análise de potência experimental, ou seja, utilizando um componente real [2] e, em um tempo razoável. O simulador elétrico SPICE (*Simulation Program for Integrated Circuits Emphasis*) [33] é a ferramenta de simulação que mais se aproxima deste panorama, conseguindo atingir níveis de precisão em torno dos 5% [2]. Logo a seguir se tem a simulação no nível de portas, método pelo qual se obtêm resultados próximos aos 25% da análise em Silício [2]. De outro modo, a análise considerando a transferência de nível de registradores ou RTL (*Register Transfer Level*) aproxima-se apenas em 50% de resultados experimentais [2].

Diante do quadro exposto, é que surge a motivação em validar, no nível físico, os circuitos multiplicadores apresentados em [9].

Assim, será possível comparar os resultados obtidos neste trabalho no nível físico, nos *chips* prototipados, com os resultados obtidos em [9] no nível lógico.

No nível físico uma das principais dificuldades para a validação dos circuitos integrados recai sobre os vetores de teste empregados, bem como as ferramentas utilizadas. O impacto do consumo de potência durante a aplicação de testes foi analisado em [41]. O calor dissipado durante a aplicação de um teste pode ser significativamente maior que durante a operação normal do circuito. Desta forma, o projeto e teste de circuitos complexos impõem extremos desafios para as ferramentas e metodologias atuais.

Normalmente, uma das motivações para testes recai no fato de que o consumo de potência medido durante a verificação de consumo do circuito é mais elevado do que o consumo de potência do circuito quando este se encontra em operação normal, podendo ser até duas vezes maior [13]. A primeira razão para tal fato é que a eficiência dos testes está relacionada com a taxa de inserção dos vetores, desta forma, a atividade de chaveamento de todos os nós do circuito no modo de teste é diversas vezes maior que na operação normal. A segunda razão se prende ao fato de que os blocos inseridos no circuito original para reduzir a complexidade posterior dos testes (DFT – *Design for Testability*) estão freqüentemente ociosos durante a operação normal. Entretanto, estes são intensivamente utilizados no modo de teste. Uma terceira razão diz respeito à inserção sucessiva de diferentes tipos de vetores de dados nas entradas de um circuito. Dependendo do tipo de vetor que é inserido em um sistema, pode

corresponder a uma significativa correlação entre os dados. Entretanto, a correlação entre vetores de teste pode não atingir tais níveis dependendo da profundidade destes vetores.

O algoritmo de testes desenvolvido neste trabalho procura levar em consideração os pontos mencionados acima. Utiliza-se o mesmo conjunto de vetores de entrada dos multiplicadores nos níveis lógico, elétrico e experimental para se ter uma comparação justa entre os diversos níveis de abstração abordados. Além disto, são utilizados vetores de teste com diferentes tamanhos para se observar o efeito da profundidade dos vetores no consumo de potência. Finalmente, o aspecto da correlação entre os vetores é considerado a partir da utilização de vetores com comportamento senoidal e randômico.

1.2. OBJETIVOS

O principal objetivo deste trabalho é analisar o desempenho sob o nível físico de arquiteturas de multiplicadores digitais. Neste contexto, são realizados testes de funcionalidade e de dissipação de potência, levando em consideração os vários níveis de abstração destes circuitos. As arquiteturas destes operadores aritméticos estão apresentadas em [9]. A principal característica das novas arquiteturas de multiplicadores é a regularidade e redução do número de linhas de produtos parciais o que contribui para o aumento de desempenho e redução do consumo de potência.

O aspecto de regularidade apresentado nas novas arquiteturas dos multiplicadores abordados se torna um fator atrativo para as suas implementações no nível físico. Neste trabalho este aspecto é levado

em consideração a partir da implementação destas arquiteturas no nível de *layout* e posterior análise no nível experimental com os circuitos prototipados.

1.3. METODOLOGIA

Os circuitos multiplicadores abordados neste trabalho foram prototipados pela *foundry* MOSIS [19] na tecnologia 0.5 μ m. São realizados testes e medições para avaliação do consumo de potência e desempenho dos multiplicadores.

As ferramentas utilizadas neste projeto fazem parte do Ambiente Mentor Graphics, onde os dados de área são analisados através do software IC Station [26] e potência utilizando o simulador elétrico ELDO [24] e o simulador Mach PA [40]. A avaliação de desempenho dos multiplicadores é realizada a partir da ferramenta SSTVelocity. A validação lógica dos circuitos é feita com o *software* ModelSim [28]. Este mesmo *software* foi utilizado para projetar o modo pelo qual os vetores de teste foram inseridos nos circuitos prototipados. A linguagem VHDL (VHSIC – *Very High Speed Integrated Circuits – Hardware Description Language*) [34] foi utilizada para esta tarefa. Para a validação física dos circuitos multiplicadores e implementação dos vetores de teste foi também utilizado o ambiente ISE da XILINX [29], responsável por sintetizar e transferir os códigos para uma placa de prototipação contendo um *chip* FPGA (*Field Programmable Gate Array*) SPARTAN IIE [31].

Algumas ferramentas acadêmicas também foram utilizadas no fluxo deste trabalho. Estas e outras ferramentas estão descritas com um maior nível de profundidade no Anexo 2.

1.4. CONTRIBUIÇÕES DO TRABALHO

A partir do estudo realizado neste trabalho para análise do consumo de potência e desempenho no nível físico de novas arquiteturas de circuitos multiplicadores, apresentam-se as seguintes contribuições:

- Verificação de parâmetros de área, atraso e consumo de potência no nível físico das novas arquiteturas de circuitos multiplicadores *array* em diferentes codificações;
- Estabelecimento de uma metodologia de implementação e análise de circuitos digitais em nível lógico, elétrico e experimental (FPGA e ASIC);
- Implementação de um algoritmo em FPGA para os testes experimentais dos circuitos integrados fabricados.

1.5. ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em 6 Capítulos. O Capítulo seguinte apresenta os principais conceitos sobre atraso e consumo de potência em circuitos CMOS. O Capítulo 3 apresenta o embasamento teórico para os circuitos multiplicadores utilizados neste trabalho. O processo de obtenção dos *layouts* das arquiteturas de multiplicadores é abordado no Capítulo 4. O Capítulo 5 apresenta em detalhes a metodologia empregada para o desenvolvimento do fluxo de projeto e

análise das arquiteturas de multiplicadores, bem como o desenvolvimento do processo e os resultados decorrentes da análise dos multiplicadores. Neste capítulo também é realizada uma comparação entre os dados obtidos na análise experimental e em simulação. Finalmente, no Capítulo 6 são apresentadas as principais conclusões do trabalho e as sugestões para trabalhos futuros.

2. POTÊNCIA E ATRASO EM CIRCUITOS CMOS

Com o aumento do uso de equipamentos portáteis a comunicação *wireless*, a dissipação de potência tem sido o ponto de maior preocupação nos circuitos VLSI (*Very Large Scale Integration*) atuais [17]. Por exemplo, o custo de produtos eletrônicos para o consumidor associados com o encapsulamento e resfriamento dos *chips* impõem limitações severas à dissipação média de potência. Uma excessiva dissipação de potência em circuitos integrados não apenas desencoraja seu uso em ambientes portáteis, mas também causa superaquecimento, o que prejudica o desempenho e reduz o tempo de vida do *chip* [37]. A aceitação do mercado para aplicações móveis e computadores portáteis também depende do tempo de operação do dispositivo associado à capacidade da bateria. Desta maneira, a redução do consumo de energia elétrica e dissipação média de potência têm se tornado um dos tópicos de mais rápido crescimento na indústria eletrônica e uma das áreas com maiores desafios de pesquisa neste campo. Aquecimento e temperatura estão diretamente relacionados com a potência média [37]. De fato, um outro fator de interesse por projeto de circuitos integrados considerando a dissipação de potência são problemas de temperatura dos dispositivos. Com a redução do tamanho dos dispositivos, a densidade dos transistores

tende a aumentar, gerando o aumento do número de componentes dissipando potência por unidade de área. Além disto, a redução do tamanho provoca a redução do atraso de propagação, permitindo a operação dos circuitos em frequências mais altas, o que por sua vez leva a um aumento do consumo de potência.

Neste capítulo serão abordados os aspectos de consumo de potência e atraso de propagação em circuitos CMOS (*Complementary Metal Oxide Semiconductor*). Inicialmente serão apresentados alguns conceitos que envolvem as principais fontes de consumo de potência em circuitos CMOS. Também serão abordados os principais aspectos que envolvem a análise de atraso de propagação de circuitos.

2.1. CONSUMO DE POTÊNCIA EM CIRCUITOS CMOS

Existem três componentes que estabelecem a quantidade de potência dissipada em circuitos CMOS [1]:

- dissipação estática de potência devido à corrente de fuga ou outras correntes que fluem continuamente pela fonte de potência do circuito;
- o consumo de potência de curto-circuito, que ocorre devido à corrente direta da fonte de alimentação para o terra durante o processo de comutação de uma porta lógica;
- dissipação dinâmica de potência devido à corrente de comutação durante a carga e descarga das capacitâncias de saída.

Considerando as componentes acima, pode-se então especificar, de acordo com a Equação (1), o consumo de potência total em circuitos CMOS:

$$(1) \quad P_{TOTAL} = P_{SC} + P_{static} + P_{din}$$

onde P_{SC} é a potência de curto-circuito, P_{static} representa a parcela devida ao consumo estático e P_{din} é a potência dinâmica.

A principal razão para a popularidade da lógica CMOS é que as portas tradicionais não possuem consumo estático quando suas saídas não estão comutando entre os níveis lógicos. Entretanto, para qualquer saída de uma porta CMOS que tenha o seu valor lógico alterado, haverá uma potência dissipada no *gate* dos transistores [15]. A razão inicial para esta dissipação de potência é o movimento de cargas elétricas para carregar e descarregar capacitâncias de carga externas e capacitâncias parasitas internas. Em adição a isto, para sinais de entrada com tempos de subida e descida finitos, é possível que um caminho de corrente direto entre a fonte de tensão e o terra seja formado temporariamente enquanto que esteja ocorrendo uma alteração na saída.

Em relação à dissipação de potência dos circuitos são utilizados alguns conceitos que definem a maneira pela qual esta componente pode ser considerada. Estes conceitos estão relacionados com a energia, a potência média, a potência instantânea e a potência de pico e estão detalhados no Anexo 1. A seguir são apresentados detalhes das 3 principais fontes de consumo de potência em circuitos CMOS.

2.2. CONSUMO DE POTÊNCIA DE CURTO-CIRCUITO

O consumo de potência de **curto-circuito** ocorre quando flui uma corrente diretamente da fonte de alimentação para o terra. Isto ocorre quando um circuito CMOS estático é chaveado por um sinal de entrada com tempos de subida e descida não-zero, tendo-se os transistores tipos PMOS (*Positive Channel Metal Oxide Semiconductor*) e NMOS (*Negative Channel Metal Oxide Semiconductor*) conduzindo simultaneamente por um curto intervalo de tempo. Este consumo adicional de potência é usualmente menor que 20% daquele devido à carga e descarga das capacitâncias parasitas [16], mas pode tornar-se substancial, particularmente se a entrada mudar de maneira lenta, ou seja, se os tempos de subida ou descida dos sinais de entrada dos circuitos forem altos. Na realidade, não é correto assumir valores de tempo de subida e descida como sendo nulos para as formas de onda de entrada [17]. Como resultado disto, é criado um caminho para a corrente diretamente de V_{DD} para GND em um curto período de tempo durante a comutação, onde os transistores PMOS e NMOS estão a conduzir simultaneamente. De acordo com [17], a corrente de curto-circuito pode ser aproximada sem maiores prejuízos por uma forma triangular, como pode ser visto na FIGURA 1.

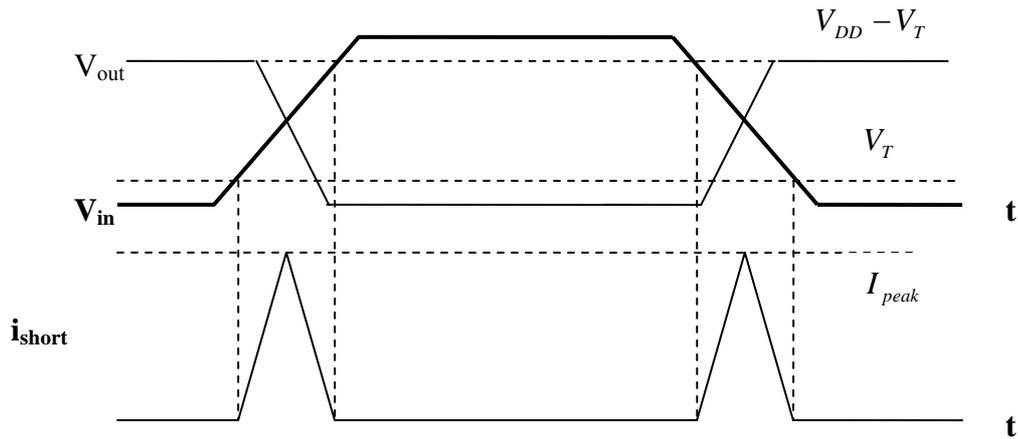


FIGURA 1 – Corrente de curto-circuito durante transientes

Baseado nesta similaridade geométrica e considerando $V_{DD} \gg |V_T|$, onde V_T representa a tensão de limiar, a Equação (2) apresenta a energia consumida por período de chaveamento.

$$(2) \quad E_{dp} = V_{DD} \frac{I_{peak} t_r}{2} + V_{DD} \frac{I_{peak} t_f}{2} = \frac{t_r + t_f}{2} V_{DD} I_{peak}$$

A corrente de pico I_{peak} é determinada pela corrente de saturação dos dispositivos e é diretamente proporcional à largura dos transistores. Para diminuir a energia E_{dp} e conseqüentemente a potência dissipada, devem ser evitados valores elevados de tempos de subida e descida, t_r e t_f respectivamente. Em geral, a potência dissipada devido a correntes de curto-circuito é muito menor que a potência dinâmica e, portanto pode ser ignorada para cálculos de primeira ordem, desde que os circuitos sejam bem selecionados.

2.3. CONSUMO DE POTÊNCIA ESTÁTICA

Correntes de fuga ou outras correntes que fluem continuamente pela fonte de potência causam a dissipação de potência estática. Idealmente, o consumo de potência estática de um circuito CMOS é nulo. Entretanto há sempre uma corrente de fuga presente. Esta contribuição no consumo total de potência está se tornando um fator de preocupação à medida que a tecnologia de processo de semicondutores atinge valores abaixo de $0.1\mu\text{m}$ [42]. Estudos demonstram que para o caso de um circuito inversor, utilizando tecnologia de 70nm, submetido a simulações operando a 125°C , as correntes de fuga podem chegar a contribuir em 49% para o consumo de potência total do circuito [43].

Correntes de fuga podem ocorrer quando um transistor está no estado desligado e outro transistor ativo carrega (*up/down*) o dreno em relação ao potencial de substrato. Considerando um circuito inversor como exemplo com uma tensão de entrada alta, tem-se neste caso, o transistor NMOS conduzindo e o transistor PMOS no estado desligado, com a saída do circuito em nível lógico baixo. Apesar de o transistor PMOS apresentar-se no estado desligado nesta situação, sua tensão fonte-substrato será igual a $-V_{DD}$, pois a tensão de saída está em 0V enquanto que a tensão de substrato em relação ao transistor PMOS está em V_{DD} como pode ser observado na FIGURA 2.

Outra fonte importante de corrente de fuga é a corrente sub-limiar, representada na FIGURA 2. Um transistor MOS pode possuir uma corrente fluindo do dreno para a fonte, mesmo quando a tensão V_{GS} é menor que tensão de limiar [17]. Esta corrente é tanto maior

quanto mais perto de 0V for a tensão de limiar para um $V_{GS}=0V$. Desta maneira ocorre um aumento do consumo de potência estático. Uma maneira de minimizar este efeito é manter a tensão de limiar V_t acima de 0,5V. A potência estática total é o produto da corrente de fuga do dispositivo e da fonte de tensão, dada pela equação abaixo:

$$(3) \quad P_{static} = I_{leakage} \cdot V_{DD}$$

onde $I_{leakage}$ representa a soma de todas as correntes de fuga presentes no circuito.

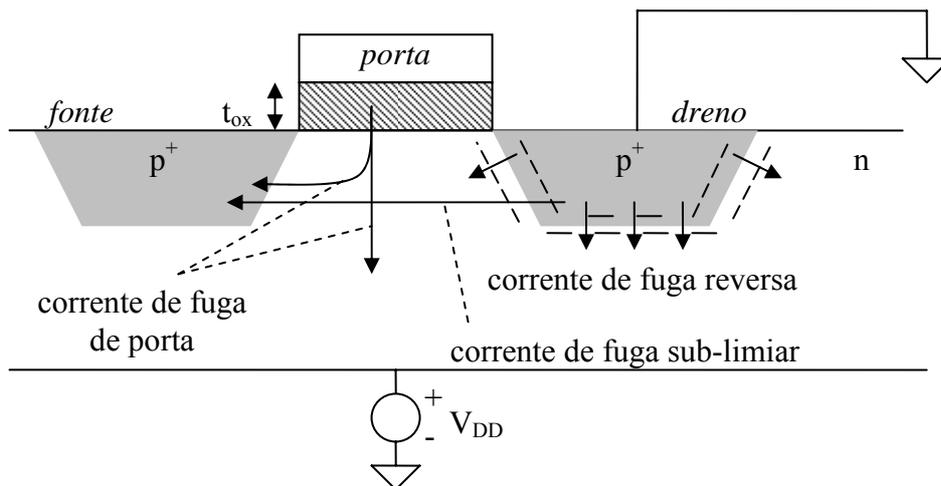


FIGURA 2 – Representação das fontes de corrente estática

2.4. CONSUMO DE POTÊNCIA DINÂMICA

A **dissipação dinâmica** ocorre durante o processo de comutação dos transistores PMOS e NMOS devido a **corrente de curto-circuito** e pelo processo de carga e descarga da capacitância de saída. Para a tecnologia CMOS empregada neste trabalho, a potência dinâmica ainda é a fonte dominante de consumo de potência, muito embora isto tenda a mudar para a integração futura em maior escala de integração [14].

A componente de comutação dinâmica de dissipação de potência (P_{din}) em uma transição na saída de uma porta carregada por um capacitor C_L é dada de acordo com a equação (4),

$$(4) \quad P_{din} = \frac{1}{2} C_L A V_{DD}^2 = \frac{1}{2} a f C_L V_{DD}^2$$

onde A é a atividade do nó de saída, medida em eventos/segundo para uma carga/descarga completa. No caso de projetos síncronos, a atividade A não é simplesmente f (frequência), mas em geral uma probabilidade de atividade normalizada a (menor do que 1 para modelo de atraso zero) é computada como função da estatística de entrada e modelos lógicos, pois nem todos os nós mudam em um determinado ciclo de relógio. Se, em um circuito, uma simples transição é realizada a cada ciclo de relógio na taxa f_{clk} , então a potência é dada por $\frac{1}{2} C_L V_{DD}^2 f_{clk}$. Entretanto, há casos em que a transição do sinal ocorre em diferentes taxas de frequência, tendo-se que considerar o valor do número de transições por ciclo de relógio ou

o fator de atividade a de transição dos nós, como mostrado na Equação (4).

2.5. GLITCHING

A dissipação de potência dinâmica também é influenciada por *glitching*, característico de sistemas com lógicas complementares em cascata [44]. A principal razão para a ocorrência do efeito de *glitching* recai sobre o fato de que as portas lógicas possuem um atraso de propagação do sinal diferente de zero [17]. Na realidade, o atraso de propagação finito entre um de um bloco lógico para o próximo, pode causar transições espúrias, chamadas de *glitches* ou *hazards*. Estas transições espúrias complicam a estimação da energia de circuitos síncronos. A principal razão para isto recai no fato de que os *glitches* são muito dependentes do atraso do circuito e, desta maneira, qualquer tentativa de se estimar a energia consumida, precisa incluir um modelo preciso de atraso [45]. Desta maneira, um determinado nó de um circuito pode exibir múltiplas transições em um único pulso de *clock* antes de estabelecer de fato o valor lógico correto. Considerando a duas portas XOR em cascata, como mostrado na FIGURA 3 e que as portas possuam atrasos idênticos e unitários. As entradas A, B e C chegam às portas ao mesmo tempo.

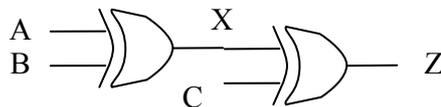


FIGURA 3 – Portas XOR em cascata

Desde que haja um atraso de propagação finito na primeira porta XOR, a segunda porta inicialmente calcula a saída Z de acordo com o valor que estava previamente armazenado em X. Quando o valor correto X, resultante da lógica da primeira porta, é liberado na saída, a segunda porta avalia novamente o sinal X. Desta maneira, a primeira avaliação torna-se redundante, consumindo potência sem gerar uma saída Z útil. Esta atividade de chaveamento extra contribui para a dissipação total de energia com cerca de 20% do valor global, mas pode chegar até a 70% da energia consumida no circuito no caso de somadores combinacionais [46].

2.6. ATRASO EM CIRCUITOS CMOS

A verificação de atraso tem por objetivo avaliar se as restrições de velocidade de operação impostas ao projeto estão sendo alcançadas ou não. Ainda, a verificação de atraso preocupa-se com a estimativa do **atraso crítico** dos circuitos e com a **máxima frequência de operação**, no caso de circuitos temporizados [8]. Tipicamente, a primeira tarefa de uma análise de atraso é determinar os atrasos em todos os caminhos do circuito.

O resultado de tal análise é verificar o caminho crítico do projeto, ou maior atraso. De posse desta informação, estabelece-se a velocidade máxima na qual o circuito irá operar. Como qualquer outro tipo de verificação, a precisão do teste do atraso é completamente dependente da precisão dos modelos de circuito adotados. O pacote de desenvolvimento de projetos disponibilizado pela Mentor Graphics, nomeadamente ADK [5], inclui todos os arquivos de bibliotecas

necessários para proceder a análise estática de atraso com a ferramenta SSTVelocity [7]. Existem várias estratégias de análise de atraso presentes na literatura [1]. A seguir são apresentadas algumas destas abordagens.

2.7. ANÁLISE ESTÁTICA DE ATRASO

A análise estática de atraso faz a verificação do atraso do circuito pela adição de atrasos de propagação ao longo de caminhos entre elementos sensíveis ao *clock* em um circuito. Nesta técnica, as somas são confrontadas com as restrições de atraso especificadas para cada caminho do circuito. Devido a sua orientação para caminhos, as ferramentas de análise estática de atraso podem determinar e reportar estatísticas de atraso tal como o número total de caminhos, atrasos entre pinos em um caminho e os caminhos mais críticos do circuito. Devido ao fato de a análise estática de atraso não realizar uma simulação funcional, os requerimentos para a construção de modelos são relativamente simples, sendo que apenas informações de atraso para cada parte do projeto são necessárias, tais como atrasos entre pinos e restrições de frequência. Desta forma, vetores de teste e modelos funcionais não são necessários desde que a análise não avalie o atraso em termos de funções lógicas [7].

2.8. ANÁLISE DE FOLGA TEMPORAL

A análise de folga temporal (*slack analysis*) compara o atraso entre caminhos de dados que terminam em dispositivos síncronos e seus correspondentes caminhos de *clock* [22]. Sinais de dados e de

clock que se propagam ao longo destes caminhos precisam ser sincronizados, caso contrário, ocorrem violações nas restrições de atraso. Este tipo de análise verifica as restrições de atraso enquanto perfaz estes caminhos, determina se ocorrem quebras nas restrições, e calcula o máximo período de tempo pelos quais caminhos de dados podem tanto causar ou evitar violações. Se a diferença entre o período especificado nas restrições de atraso e aquele calculado pela ferramenta como necessário para o circuito resultar em um valor negativo, indica que uma violação realmente ocorreu, enquanto que um valor positivo estabelece que a violação não ocorreu.

2.9. ANÁLISE DINÂMICA DE ATRASO

A análise dinâmica verifica o atraso do circuito pela aplicação de vetores de teste que são projetados para detectar violações de atraso. Esta aproximação é uma extensão da simulação e garante que o atraso do circuito é testado em um contexto funcional. Este tipo de teste traz como resultado erros de atraso que funcionalmente existem no circuito e prevê erros que ocorram em caminhos não usados do circuito [22]. Para testar o atraso do circuito utilizando condições de pior caso, as ferramentas de análise dinâmica de atraso avaliam o circuito utilizando o mínimo e máximo atrasos de propagação para cada componente no projeto. Para disponibilizar uma análise realmente do pior caso, a ferramenta precisa determinar quando aplicar um atraso mínimo ou máximo baseada nas ligações e se outros tipos de restrições estão sendo avaliados.

2.10. RESUMO

Este capítulo apresentou as principais fontes de consumo de potência em circuitos CMOS. Foram abordados aspectos relacionados à dissipação de potência dinâmica e estática. Dentre as componentes dinâmicas, *hazards* podem provocar um grande aumento na dissipada, mas podem ser reduzidos, igualando-se os comprimentos dos caminhos de dados pelos quais um sinal do circuito tem de passar. Tal técnica é utilizada por diversas ferramentas de síntese automática como o Leonardo Spectrum [6], utilizado neste trabalho. Já as componentes de potência estática e de curto-circuito têm uma menor influência no consumo global do circuito. Duas estratégias de análise de atraso também foram apresentadas neste capítulo: *slack analysis* e análise dinâmica.

3. ARQUITETURAS DE MULTIPLICADORES

Entre os operadores aritméticos, os circuitos multiplicadores são os mais comuns em muitas operações envolvendo multiplicação e acumulação (MAC), como os algoritmos da área de Processamento Digital de Sinais (DSP), por exemplo. Nos circuitos DSP, os multiplicadores são os responsáveis pela maior parte do consumo de potência. Diante disto, apresenta-se o interesse na aplicação de técnicas que possam reduzir o consumo de potência dos circuitos multiplicadores. Novas arquiteturas de circuitos multiplicadores de baixa potência do tipo *array*, que operam em complemento de 2 na base 2^m , foram recentemente apresentadas em [9]. A principal característica destes novos multiplicadores é a redução da atividade de chaveamento e operação em diferentes codificações de dados. Outro aspecto a ser destacado é a regularidade dos circuitos, que facilita além das suas implementações em *layout*, a aplicação de outras técnicas de redução de potência, como *pipelining*, por exemplo.

As novas arquiteturas de multiplicadores propostas em [9] apresentam significativas reduções de potência quando comparadas com o estado da arte (multiplicador Booth). Entretanto, todos os resultados até então apresentados apontam para o nível lógico. O presente trabalho estende o estudo apresentado em [9] propondo a

implementação, prototipação e análise destes novos multiplicadores em níveis mais baixos de abstração.

A seguir são apresentadas as principais características das arquiteturas de multiplicadores alvo deste trabalho.

3.1. MULTIPLICADORES ARRAY NA BASE 2^m EM COMPLEMENTO DE 2

A idéia da multiplicação na base 2^m é operar grupos de m bits ao invés de realizar a multiplicação bit a bit como em uma operação de multiplicação convencional [35]. Cada um destes grupos pode ser visto como uma representação de um dígito na base 2^m . Desta maneira, a arquitetura de multiplicadores segue a operação básica de multiplicação de números representados nesta base. As arquiteturas de multiplicadores na base 2^m são compostas por elementos básicos que proporcionam as operações dos produtos parciais. Os circuitos somadores básicos são responsáveis pelas somas destes produtos parciais. A seguir são apresentadas as principais características dos multiplicadores em complemento de 2 na base 2^m .

3.2. MULTIPLICADOR BINÁRIO EM COMPLEMENTO DE 2

Considerando dois operandos A e B de tamanho W , tem-se a operação na base 2^m em complemento de dois dada pela equação (5).

$$(5) \quad A \times B = A' \times B' - A' b_{W-1} \frac{b_{\frac{W}{m}-1}}{m} 2^{W-m} - a_{W-1} a_{\frac{W}{m}-1} \sum_{j=0}^{\frac{W}{m}-1} b_j 2^{W-m+j}$$

onde $A' = \sum_{i=0}^{W-m-2} a_i 2^{i-m}$ e $B' = \sum_{i=0}^{W-m-2} b_i 2^{i-m}$.

Esta operação é ilustrada na FIGURA 4.

Para os $W-m$ bits menos significativos dos operandos, pode-se efetuar a multiplicação sem sinal. Uma multiplicação com sinal é realizada quando estão envolvidos os bits mais significativos de um dos operandos (ou dos dois operandos simultaneamente). A técnica de extensão do sinal é utilizada para manter a regularidade da operação.

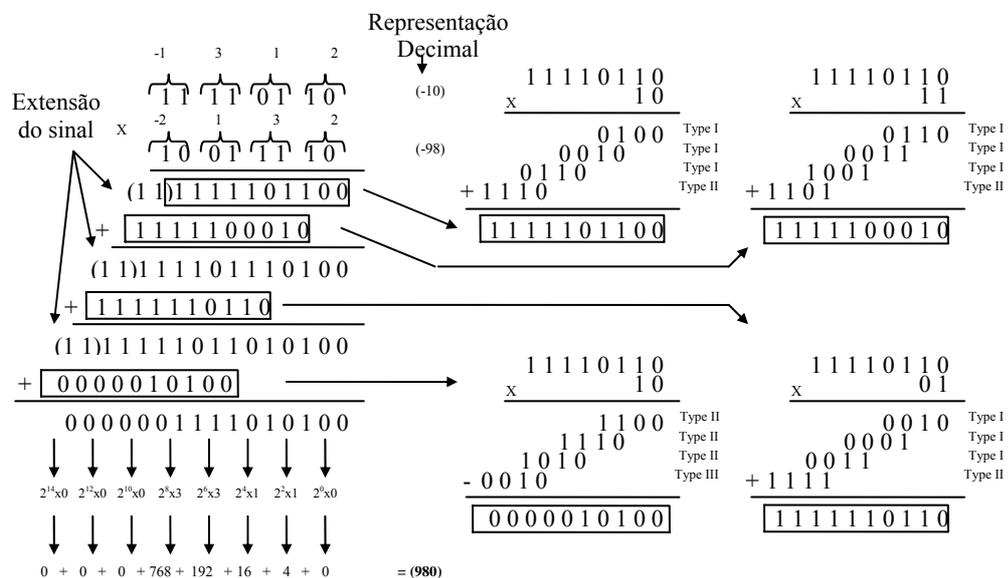


FIGURA 4 – Exemplo de uma multiplicação na base 4 ($m=2$) em complemento de 2 utilizando o código Binário

A FIGURA 5 mostra um exemplo da arquitetura de 8 bits na base 4 em complemento de 2 do multiplicador *array* Binário.

Para esta arquitetura, três tipos de módulos multiplicadores são necessários. Os módulos *Type I* são os módulos sem sinal. Os módulos *Type II* operam o produto parcial de m -bits de um valor sem

sinal com um valor em complemento de 2. Finalmente, os módulos *Type III* são módulos que operam sobre dois valores com sinal. Apenas um módulo *Type III* é requerido para os multiplicadores *array* na base 2^m , Para estes mesmos multiplicadores são necessários $2\frac{w}{m} - 2$ módulos *Type II* e $(\frac{w}{m} - 2)^2$ módulos *Type I*.

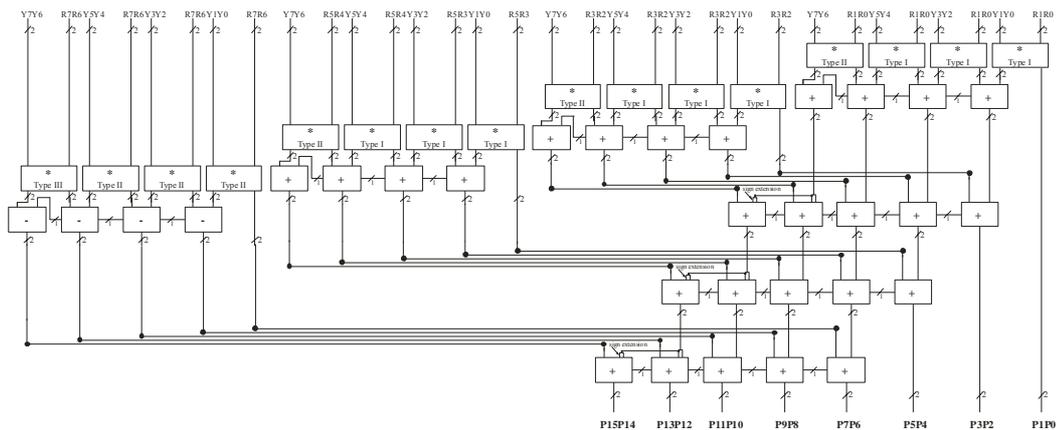


FIGURA 5 – Arquitetura do multiplicador *array* Binário de 8 bits em complemento de 2

3.3. MULTIPLICADOR HÍBRIDO EM COMPLEMENTO DE 2

Em [30] foi apresentada uma nova arquitetura de circuito multiplicador operando em um código diferente do Binário. A este novo circuito foi dado o nome de multiplicador Híbrido. A principal característica do circuito multiplicador operando neste código é a possibilidade de redução da atividade de comutação em grupos de m bits dos operandos. A seguir é apresentada uma introdução ao código Híbrido, bem como da operação de multiplicação com este código.

3.4. INTRODUÇÃO AO CODIGO HÍBRIDO

A idéia do código Híbrido é dividir os operandos em grupos de m bits, codificar cada grupo utilizando o código Gray e utilizar o comportamento do código Binário para propagar o *carry* entre os grupos. Desta forma o número de transições em cada grupo pode ser minimizado e uma estrutura regular pode ser obtida, onde os grupos menos significativos do resultado dependam somente dos grupos menos significativos dos operadores. A TABELA 1 mostra as representações dos códigos Binário, Híbrido ($m=2$) e Gray para números de 4 bits.

TABELA 1 – Representações dos códigos Binário, Híbrido ($m=2$) e Gray

Decimal	Binário	Híbrido ($m=2$)	Gray
0	0000	0000	0000
1	0001	0001	0001
2	0010	0011	0011
3	0011	0010	0010
4	0100	0100	0110
5	0101	0101	0111
6	0110	0111	0101
7	0111	0110	0100
8	1000	1100	1100
9	1001	1101	1101
10	1010	1111	1111
11	1011	1110	1110
12	1100	1000	1010
13	1101	1001	1011
14	1110	1011	1001
15	1111	1010	1000

Como pode ser observado na TABELA 1, o código Híbrido apresenta um compromisso entre a mínima dependência das entradas de dados apresentada pelo código Binário e a característica de baixa

atividade de chaveamento apresentada pelo código Gray. A TABELA 2 apresenta o número de transições dos códigos Binário, Gray e Híbrido para uma determinada seqüência de contagem.

TABELA 2 – Número de transições para os códigos Binário, Híbrido ($m=2$) e Gray

Número de Bits	Número de Transições			Diferença	
	Binário	Gray	Híbrido ($m=2$)	Hib→Bin	Hib→Gray
n = 4 bits (0,1,...,15,0)	30	16	20	-33,3%	+25%
n = 8 bits (0,1,...,255,0)	510	256	340	-33,3%	+32,8%
n = 16 bits (0,1,...,65535,0)	131070	65536	87380	-33,3%	+33,3%

Como pode ser observado na TABELA 2, o código Híbrido apresenta um valor percentual intermediário entre os códigos Binário e Gray em termos de número de transições. Desta forma, para sistemas onde a capacitância chaveada no barramento de dados é significativa e onde os dados apresentam um alto grau de correlação, a utilização do código Híbrido pode reduzir o consumo de potência em até cerca de um terço em relação ao código Binário [9]. O número de transições para seqüências de contagem, como apresentadas na TABELA 2, é calculado de acordo com as equações (6), (7) e (8), para os códigos Binário, Gray e Híbrido respectivamente.

$$(6) \quad n^{\circ}_{de_trans.}(Bin)=2^{n+1} - 2$$

$$(7) \quad n^{\circ}_{de_trans.}(Gray)=2^n$$

$$(8) \quad n^{\circ}_{de_trans.}(Hib)=2^m \frac{2^n - 1}{2^m - 1}$$

Outra característica apresentada pelo código Híbrido $m=2$ é a facilidade de mudança de representação para o código Binário, como mostra o exemplo da FIGURA 6, para uma representação $m=2$. Desta forma, o processo de codificação/decodificação dos dados utiliza um *hardware* de reduzida complexidade com uma porta EXOR ligada a cada grupo de $m=2$ bits. Neste caso, o código Híbrido também pode ser utilizado como método de codificação para os barramentos de endereços.

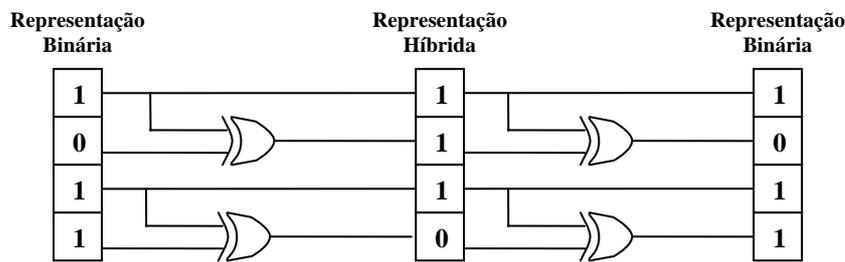


FIGURA 6 – Conversão entre os códigos Binário e Híbrido

3.5. OPERAÇÃO DE MULTIPLICAÇÃO EM CÓDIGO HÍBRIDO

No multiplicador Híbrido tipo *array*, a operação é realizada em cada grupo de m bits, onde m representa o tamanho do grupo de bits do operando. Os termos dos produtos parciais são expressos na representação na base 2^m , A FIGURA 7 ilustra um exemplo de uma multiplicação de 8 bits na base 4 ($m=2$) em código Híbrido.

A estrutura do multiplicador *array* Híbrido é semelhante à estrutura do multiplicador *array* Binário apresentada na FIGURA 5. A diferença consiste nos módulos operadores, que operam diretamente em código Híbrido para este multiplicador.

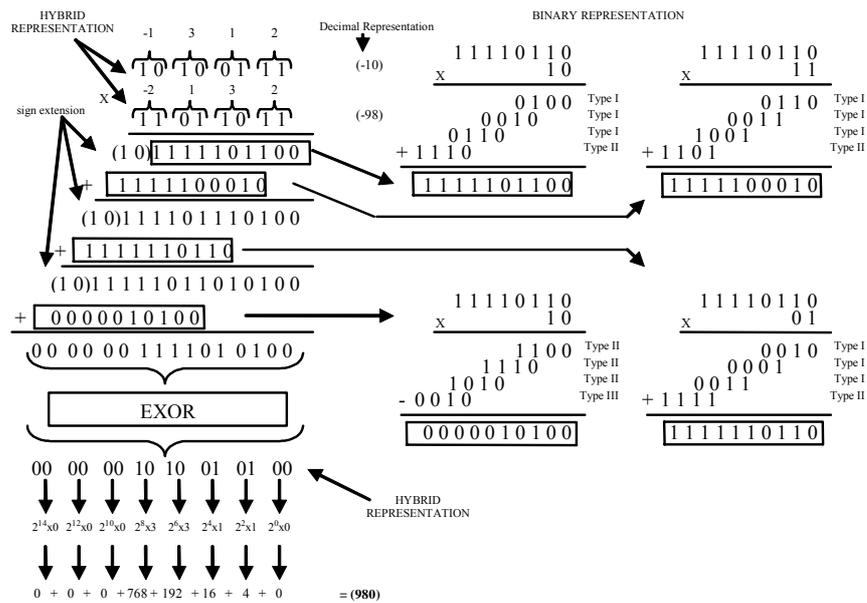


FIGURA 7 – Exemplo de uma multiplicação na base 4 ($m=2$) em complemento de 2 utilizando o código Híbrido

3.6. MULTIPLICADOR BOOTH MODIFICADO

O algoritmo Booth verifica a detecção do início, do interior e do final de um cadeia de valores no nível lógico 1. Este algoritmo apresenta como principal característica a possibilidade de operar sobre números representados em complemento de 2. No algoritmo Booth original, 2 bits são verificados ao mesmo tempo, determinando o tipo de operação a ser efetuada. Desta forma, em um termo multiplicador, o conjunto de dígitos que compõe a representação na base 2 é dado por (-1,0,1). Este conjunto é utilizado para ajustar os termos multiplicando. Uma codificação do termo multiplicando é utilizada para simplificar os produtos parciais.

O algoritmo Booth Modificado alcança um melhor desempenho através da codificação na base 4. Neste algoritmo são gerados os

sinais de controle (0, +X, +2X, -X e -2X) a partir do operando do multiplicador para cada grupo de 3 bits, como mostra o exemplo da FIGURA 8 para uma operação de 8 bits. Um circuito multiplexador produz os produtos parciais de acordo com o sinal de controle codificado.

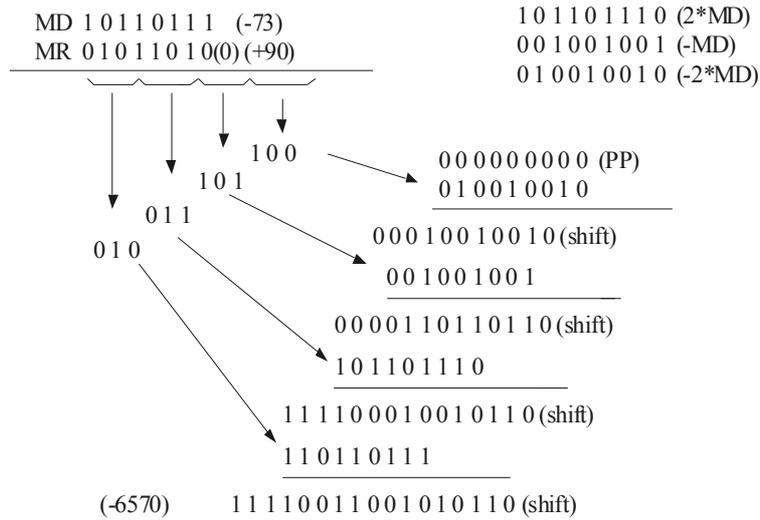


FIGURA 8 – Exemplo de uma multiplicação de 8 bits utilizando o algoritmo Booth Modificado

Como pode ser observado no exemplo da FIGURA 8, para reduzir o número de produtos parciais no multiplicador Booth Modificado, realiza-se uma operação em grupos de 3 bits no termo multiplicador (MR). De fato, o termo multiplicador é verificado de 2 em 2 bits com sobreposição do terceiro bit. Na varredura dos bits menos significativos do termo multiplicador, considera-se a existência de um valor zero virtual a direita do bit menos significativo. Deve-se observar que no algoritmo Booth também é utilizada a técnica de extensão do sinal. Para este algoritmo, utilizam-se dois bits extras para

tornar o *array* mais regular. Um circuito multiplexador produz os produtos parciais de acordo com o sinal de controle codificado.

A arquitetura do algoritmo Booth na base 4 (Booth Modificado) é apresentada em [23]. Nesta arquitetura é possível reduzir o número de produtos parciais pela codificação do multiplicador em complemento de 2. A geração do complemento de 2 para o termo multiplicando é realizada por cada circuito multiplexador e codificador mostrados na FIGURA 9, para uma operação de 8 bits. Como pode ser observado nesta figura, são necessários 4 blocos contendo um codificador e um multiplexador para calcular os termos dos produtos parciais. Esses circuitos produzem o termo multiplicando de acordo com os 3 bits no termo multiplicador (MR). Os termos do produto parcial são deslocados pelos circuitos somadores, que são também utilizados para produzir o resultado final.

O aspecto comum entre a arquitetura Booth e as arquiteturas propostas em [9] é o fato de que em cada passo do algoritmo dois bits são processados ao mesmo tempo. Entretanto as células Booth básicas não são simples circuitos somadores como nas arquiteturas *array* propostas em [9]. Estas células têm de realizar operações de soma, subtração e deslocamento à esquerda, dos bits do termo multiplicando. Esta complexidade torna difícil a implementação da arquitetura Booth em bases maiores o que 4 [9].

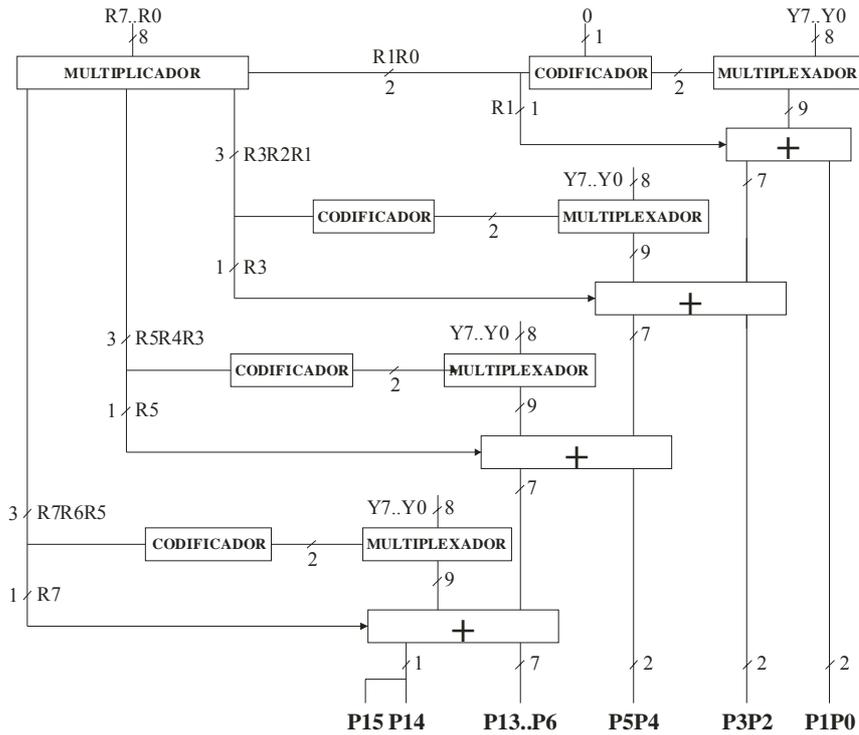


FIGURA 9 – Arquitetura de 8 bits do multiplicador Booth Modificado

3.7. RESUMO

Neste capítulo foram apresentadas as arquiteturas *array* Binário e Híbrido propostas em [9] e as arquiteturas Booth Modificado. Foi introduzido o código Híbrido e mostrado que esta codificação representa um caso intermediário entre o código Binário e o Gray em termos de número de transições. Esta característica torna a codificação Híbrida atrativa para a implementação de circuitos buscando a redução do consumo de potência. Os três circuitos apresentados operam números em complemento de dois na base 2^m e utilizam palavras de 8 bits. O aspecto comum entre a arquitetura Booth e as arquiteturas *array* é o fato de que em cada passo do algoritmo dois bits são processados ao mesmo tempo.

4. SÍNTESE FÍSICA DAS ARQUITETURAS

As arquiteturas de multiplicadores propostas em [9] e resumidas no Capítulo anterior foram submetidas ao processo de síntese física utilizando a geração automática de *layout*. O principal conjunto de ferramentas de CAD (*Computer-Aided Design*) pertence ao ambiente Mentor Graphics. Destaca-se ainda a utilização da ferramenta ISE (*Integrated Software Environment*) pertencente a XILINX. Ainda, como auxílio para o desenvolvimento do fluxo de projeto, algumas ferramentas acadêmicas foram utilizadas. Estas ferramentas serão apresentadas ao longo deste Capítulo. Durante a etapa de síntese das arquiteturas de multiplicadores, um pacote de desenvolvimento, análise e prototipação, programado para trabalhar com as ferramentas do ambiente Mentor Graphics, foi utilizado. Este pacote de desenvolvimento (ADK – *ASIC Design Kit*) possui as bibliotecas, regras de projeto e arquivos de processo apropriados para a tecnologia de fabricação de CI's utilizada neste trabalho, qual seja AMI 0.5 μ m e é disponibilizado pelo HEP (*Higher Educational Program*) da Mentor Graphics. Esta tecnologia está disponível para submissão de projetos a MOSIS [19]. A área do núcleo dos circuitos é extraída a partir das informações de *layout* fornecidas pelo *software* IC Station.

Neste capítulo será abordado o fluxo de projeto para a síntese das arquiteturas de multiplicadores.

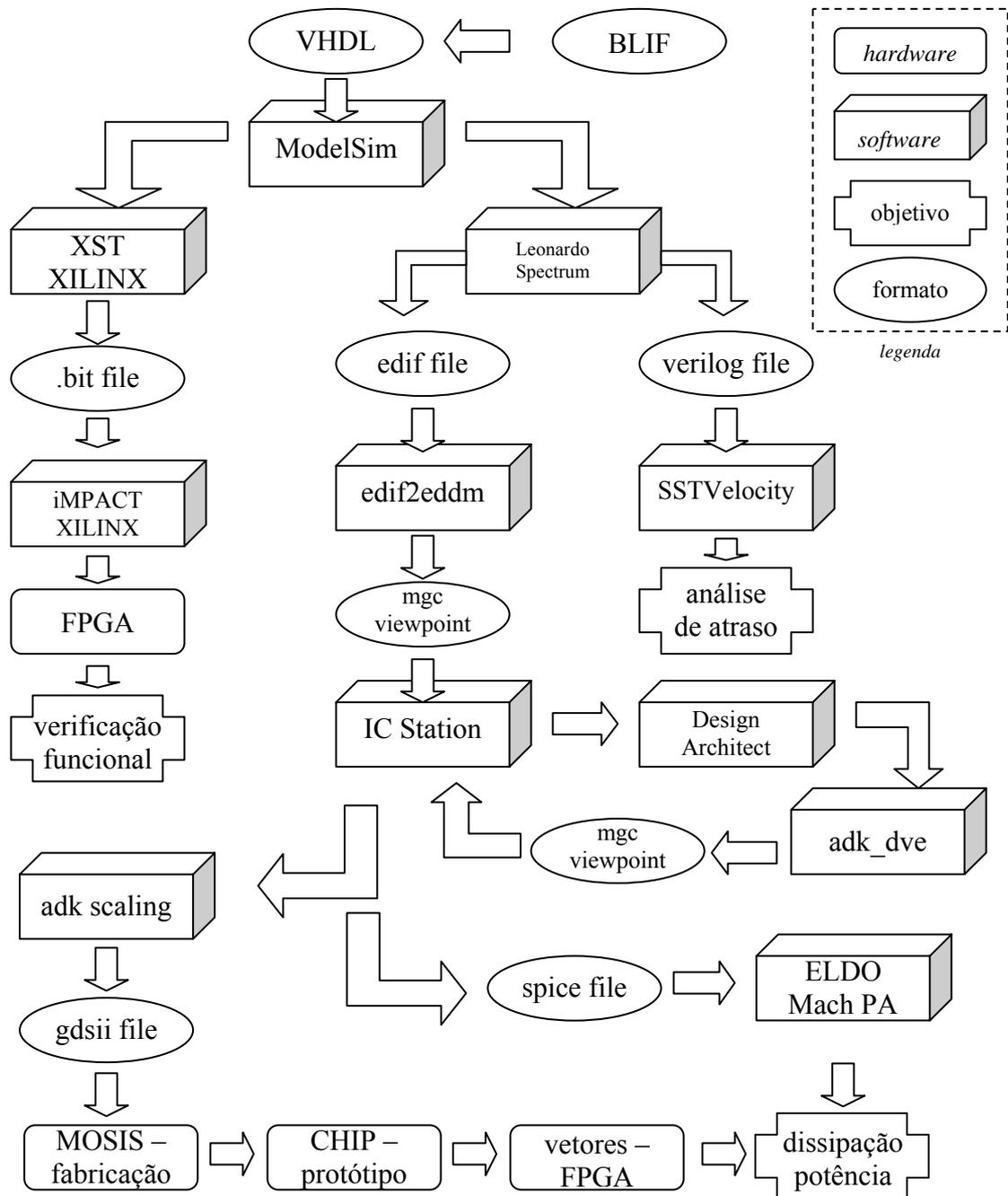


FIGURA 10 – Diagrama representado as várias etapas para obtenção e verificação das arquiteturas de multiplicadores

4.1. FLUXO DE PROJETO PARA A SÍNTESE DOS MULTIPLICADORES

No fluxo de projeto, uma série de ferramentas comerciais e acadêmicas foi utilizada, bem como equipamentos de teste e medição. Dentre as ferramentas utilizadas, destacam-se: ModelSim [28], XST [29], iMPACT [29], Leonardo Spectrum [6], SSTVelocity [7], IC Station [26], Design Architect [27], ELDO [24] e Mach PA [40]. Todas as ferramentas mencionadas fazem parte integrante do fluxo de projeto, mostrado na FIGURA 10. O Anexo 2 apresenta a definição de cada uma das ferramentas mencionadas, bem como dos equipamentos utilizados para testes e medições. Em geral, as diversas ferramentas disponíveis para um projetista podem ser classificadas 3 classes globais [17]:

- Análise e verificação: estas ferramentas examinam o comportamento do circuito e ajudam a determinar se a resposta está dentro dos limites especificados;
- Implementação e síntese: auxiliam o projetista na geração e otimização dos esquemáticos ou *layouts* dos circuitos;
- Testabilidade: são técnicas que provêm uma combinação dos métodos de projetos e ferramentas de CAD para validar a funcionalidade de um projeto já fabricado.

É importante ressaltar que há uma diferença entre simulação e verificação [17]. Quando o que está em estudo é **simulação**, o valor de um parâmetro de simulação como uma margem de ruído, atraso de propagação, ou energia dissipada é determinado pela aplicação de um padrão de vetores de excitação ao modelo do circuito de escolha e,

extraindo estes parâmetros diretamente dos sinais das formas de onda obtidas. Enquanto esta aproximação é muito flexível, ela possui a desvantagem que os resultados são fortemente dependentes da excitação de entrada do circuito. A **verificação**, por outro lado, compromete-se a extrair os parâmetros de sistema diretamente da descrição do circuito. Esta aproximação possui a vantagem de que o resultado é independente da escolha dos vetores de excitação. Do lado negativo, esta aproximação requer um entendimento inerente do estilo de projeto de circuito utilizado.

Tomando-se por base a classificação anteriormente descrita, podem-se dividir as ferramentas utilizadas neste trabalho, e mostradas no diagrama da FIGURA 10 como segue:

Análise e verificação: ModelSim, ELDO, Mach PA, SSTVelocity;

Implementação e síntese: XST e iMPACT (XILINX), adk_scaling, adk_dve, Design Architect, IC Station, edif2eddm, Leonardo Spectrum;

Testabilidade: FPGA.

O diagrama apresentado na FIGURA 10 divide-se em dois ramos principais, sendo o da esquerda atribuído às etapas de utilização do ambiente ISE da XILINX. O ramo da direita desenvolve os procedimentos de obtenção dos *layouts* dos circuitos, testes de atraso, simulações elétricas, prototipação e testes experimentais.

A primeira etapa de desenvolvimento consiste na descrição dos circuitos multiplicadores em formato *blif* e tradução para VHDL. A partir desta descrição dos circuitos, segue-se para a primeira verificação em *hardware* dos multiplicadores. Esta etapa é mostrada no ramo da esquerda da FIGURA 10. Inicialmente utiliza-se o programa ISE da XILINX como principal ferramenta de desenvolvimento. Após, verifica-se a sintaxe do código por intermédio do programa ModelSim, uma ferramenta de simulação lógica. O programa XST é utilizado para gerar o arquivo com a extensão *.bit* necessário para que o programa iMPACT possa carregar a descrição VHDL para dentro do dispositivo FPGA XILINX SPARTAN IIE utilizado neste trabalho. Com os multiplicadores devidamente prototipados em FPGA, procede-se então à verificação da funcionalidade das arquiteturas com o auxílio de um osciloscópio. Esta mesma etapa de uso da FPGA também se faz necessária para o ajuste dos vetores de teste que são posteriormente inseridos nas entradas dos protótipos.

O ramo da direita da FIGURA 10 é onde se processa a maior parte do fluxo de projeto. O programa Leonardo Spectrum é utilizado nesta etapa com dois propósitos. O primeiro deles diz respeito a gerar uma descrição em formato *verilog* das arquiteturas de multiplicadores. Isto para servir de arquivo de entrada da ferramenta de análise de atraso SSTVelocity, onde são extraídos os dados de atraso de cada circuito. O segundo objetivo de utilização do programa Leonardo Spectrum é gerar o arquivo *edif* (*electronic design interchange format*) das arquiteturas de multiplicadores. Este formato serve de

base à geração dos *layouts* e posterior simulação elétrica dos multiplicadores. A criação do *view point* é uma etapa do fluxo do projeto desenvolvida no ambiente Mentor Graphics e, representa o local onde são armazenadas informações sobre interconectividade entre as diferentes ferramentas utilizadas, iniciando o compartilhamento de informações entre os programas. O *view point* é produzido pelo script `edif2eddm` (*eddm – electronic design database model*). A síntese das arquiteturas utilizando o programa Leonardo Spectrum é realizada duas vezes para cada circuito. A primeira síntese envolve o código VHDL das arquiteturas de multiplicadores operando de acordo com um sinal de sincronismo. Todos os circuitos são submetidos às mesmas condições de síntese e otimizações de área e atraso. A otimização de área analisa a lógica presente no circuito e decide pela substituição de algumas células por outras mais otimizadas e que estão presentes na biblioteca de *starnd cells* da *foundry*. A otimização de atraso procura diminuir a existência de *glitches* durante a operação do circuito pela tentativa de equalização do tempo que os sinais de dados necessitam para chegarem às portas. A característica do sinal de sincronismo é necessária para etapa anterior, onde os circuitos são sintetizados para um dispositivo FPGA. O segundo grupo de circuitos sintetizados com o programa Leonardo Spectrum utiliza apenas lógica combinacional na implementação das arquiteturas e não sofre otimizações de *timing*, apenas de área. O Anexo 3 contém os *layouts* destas implementações. Estes circuitos não são sintetizados em FPGA, sendo a sua verificação funcional limitada a simulações lógicas com o simulador ModelSim e elétricas com o

simulador Mach PA. A ferramenta IC Station é utilizada para gerar o *layout* de cada arquitetura de multiplicador. Inicialmente se tem o processo de síntese do núcleo do circuito, ou seja, apenas a parte dedicada a desempenhar as funções lógicas descritas no código VHDL. Neste ponto são realizadas as atividades de *autofloorplan*, auto posicionamento e auto roteamento. Ainda com o objetivo de completar o layout do circuito, o programa Design Architect é utilizado para fazer as ligações sob o nível de esquemáticos entre o núcleo do circuito e os PADS de entrada e saída. Novamente é criado um *view point* com estas informações de interconectividade. Este *view point* é criado para ser utilizado com o programa IC Station para a geração do *padframe*. O *view point* é gerado através do script *adk_dve*. Os *layouts* completos das arquiteturas de multiplicadores estão apresentados na FIGURA 11, FIGURA 12 e FIGURA 13. Uma vez gerado o layout ocorrem as etapas de extração dos circuitos para a simulação elétrica e fabricação dos protótipos. Um *netlist* em formato SPICE é construído com base nas informações contidas nas bibliotecas da tecnologia trabalhada. Com isto é possível proceder à simulação elétrica com o simulador ELDO e extração das informações de dissipação de potência. O outro ponto diz respeito à geração de um arquivo em formato *gdsii* padrão contendo as informações das máscaras. Este arquivo é enviado para a MOSIS onde é gerenciado o processo de prototipação dos multiplicadores de acordo com as informações de máscaras contidas neste arquivo. A FIGURA 14, a FIGURA 15 e a FIGURA 16, mostram fotografias dos protótipos dos circuitos multiplicadores.

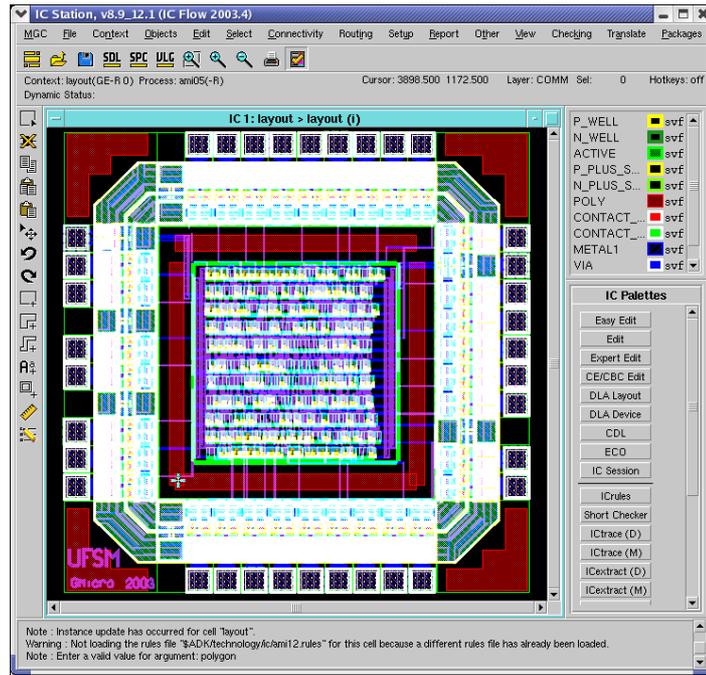


FIGURA 11 – *Layout* do circuito multiplicador *array* Binário de 8 bits

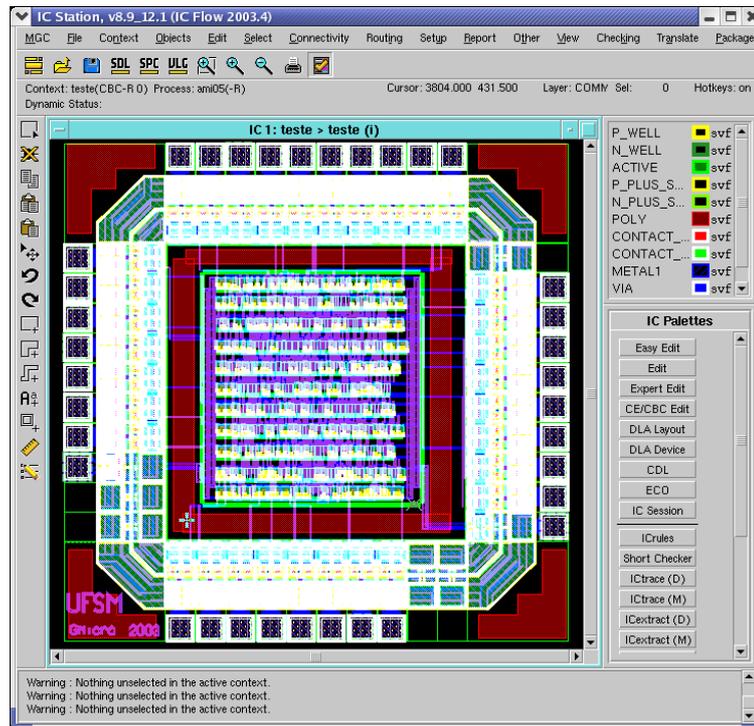


FIGURA 12 – *Layout* do circuito multiplicador *array* Híbrido de 8 bits

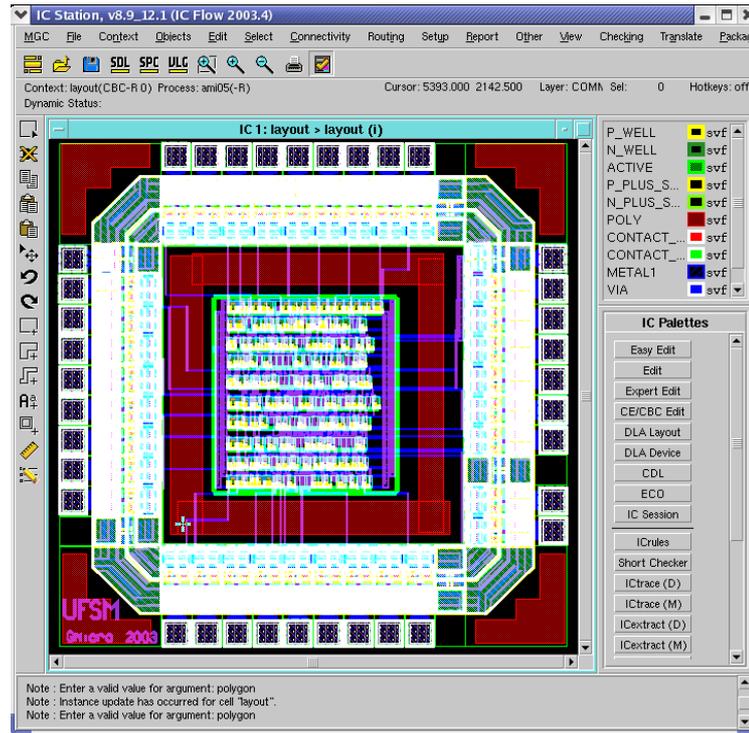


FIGURA 13 – *Layout* do circuito multiplicador Booth Modificado de 8 bits

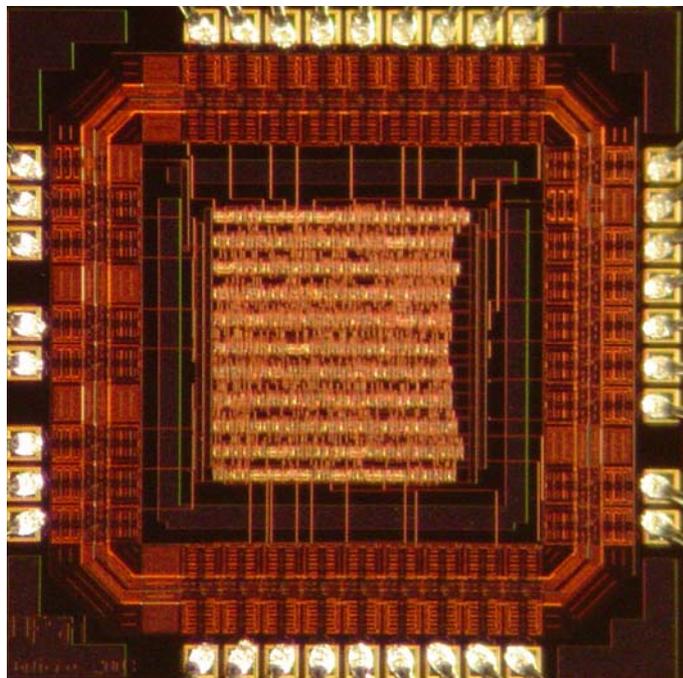


FIGURA 14 – Foto do protótipo do circuito multiplicador *array* Binário de 8 bits

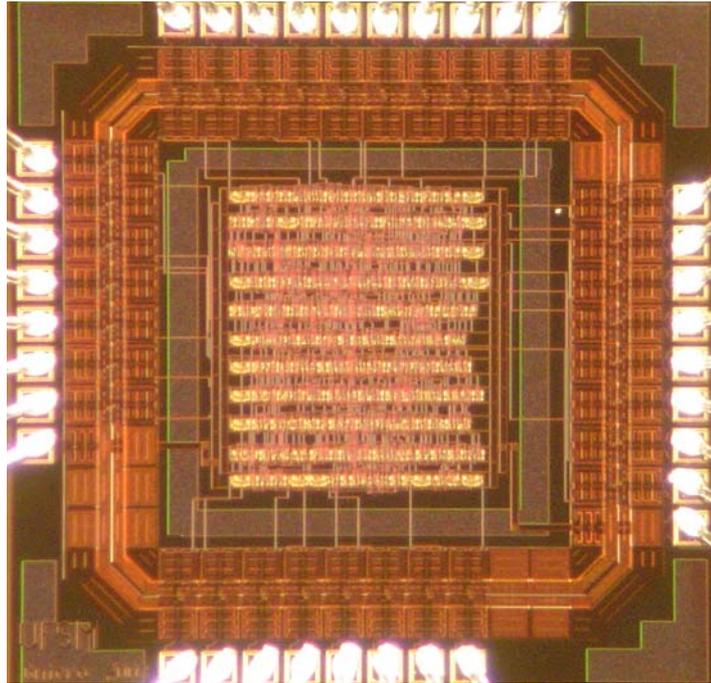


FIGURA 15 – Foto do protótipo do circuito multiplicador *array* Híbrido de 8 bits

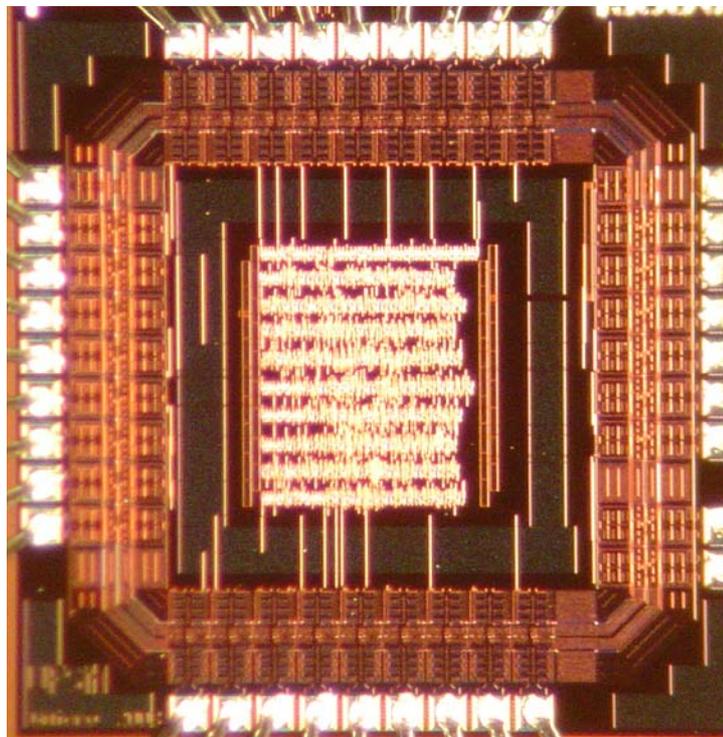


FIGURA 16 – Foto do protótipo do circuito multiplicador Booth de 8 bits

A TABELA 3 apresenta os resultados de área, número de transistores e atraso para cada arquitetura. O campo porcentagem apresenta valores de comparação que tomam por base a arquitetura Booth Modificado. O multiplicador *array* Híbrido apresenta valores de área mais elevados devido a maior complexidade de seus elementos básicos. Deve-se observar na TABELA 3 que mesmo o multiplicador Híbrido apresentando menor largura em relação ao multiplicador Binário, a sua significativa maior altura compromete a área total desta arquitetura. Os resultados apresentados na TABELA 3 estão de acordo com os *layouts* apresentados na FIGURA 11, FIGURA 12 e FIGURA 13.

TABELA 3 – Resultados de área e atraso para a primeira síntese dos multiplicadores

	altura (μm)	largura (μm)	área total (mm^2)	%	número transistores	%	atraso (ns)	%
Booth	546,0	587,1	0,320	–	2162	–	17,71	–
Array Binário	662,4	650,1	0,431	+34,68	3006	+39,04	15,99	-9,71
Array Híbrido	657,3	702,9	0,462	+44,37	2966	+37,19	17,29	-2,37

Os circuitos multiplicadores primeiramente sintetizados pelo programa Leonardo Spectrum, cujos resultados são apresentados na TABELA 3, foram submetidos a otimizações de área e atraso. As otimizações de atraso foram realizadas no sentido de diminuir o efeito dos *glitches* na estrutura do circuito. A otimização de área diminui o tamanho dos circuitos o máximo possível. Nesta etapa de otimização de área o programa Leonardo Spectrum verifica a lógica e substitui células do projeto original (*and's, nor's...*) por células existentes na

biblioteca da AMI, geralmente superportas ou SCCG (*Static CMOS Complex Gates*) [39].

Uma nova síntese para cada arquitetura com o programa Leonardo Spectrum, sem utilizar as otimizações acima descritas, foi então realizada. Este segundo processo de geração de *layout* foi implementado no sentido de verificar o impacto do resultado da síntese automática utilizando as otimizações de área e atraso realizadas pelo *software* Leonardo Spectrum. Os dados de área e atraso apresentados na TABELA 4 se referem a esta segunda síntese das arquiteturas de multiplicadores.

TABELA 4 – Resultados de área e atraso para a segunda síntese dos multiplicadores

	altura (μm)	largura (μm)	área total (mm^2)	%	número transistores	%	atraso (ns)	%
Booth	480	540	0,26	–	1691	–	16,94	–
Array Binário	588	630	0,37	+42,92	2323	+37,37	15,96	-5,78
Array Híbrido	579	702	0,41	+56,81	2487	+47,07	17,28	+2,00

Pode-se notar pela TABELA 4 que os dados apresentam valores de número de transistores e área menores que os resultados obtidos com a síntese anterior, cujos resultados foram apresentados na TABELA 3. Isto ocorre devido ao fato de que estes circuitos não possuem o *hardware* adicional empregado na primeira síntese para a implementação do sinal de *clock* dos circuitos, tampouco a inserção de *buffers* necessários na etapa de otimização de atraso. Particularmente em relação à área, a altura é a maior responsável pelos níveis elevados de área apresentados pelos circuitos multiplicadores *array*. Apesar do multiplicador *array* Binário apresentar maior valor de área, esta

arquitetura apresenta menor valor de atraso como mostrado na TABELA 3 e na TABELA 4. Isto devido à menor quantidade de termos de produtos parciais e menor valor de níveis lógicos apresentado por este circuito. O fato de o multiplicador Booth necessitar de um esquema de codificação e um multiplexador, que produzem o termo multiplicando de acordo com os três bits no termo multiplicador, faz com que esta arquitetura apresente um maior atraso em relação ao multiplicador *array* Binário. Embora o multiplicador Híbrido também reduza as linhas de produtos parciais, esta arquitetura apresenta um maior atraso em relação às demais arquiteturas de acordo com a TABELA 4. Isto se explica pelo fato do multiplicador Híbrido utilizar portas EXOR nas entradas dos circuitos somadores para conversão dos dados. Esta lógica adicional faz aumentar o seu caminho crítico, embora não represente um aumento significativo. Ainda assim, para os resultados da primeira síntese, mostrados na TABELA 3, o multiplicado Híbrido sustenta uma pequena vantagem.

4.2. RESUMO

A geração automática de *layout* foi um dos principais pontos abordados neste Capítulo. Foi apresentado o fluxo de análise implementado para as arquiteturas de circuitos multiplicadores e a metodologia utilizada neste trabalho, desde a descrição dos circuitos em VHDL até a etapa de prototipação. A tecnologia de implementação das arquiteturas foi a AMI 0.5 μ m. Foram apresentadas as ferramentas de síntese e simuladores dos ambientes de projeto (Mentor Graphics e XILINX).

5. ESTRATÉGIA DE VERIFICAÇÃO E ANÁLISE DOS RESULTADOS

A principal estratégia de verificação utilizada neste trabalho consiste na utilização de diferentes ferramentas de CAD (*Computer Aided Design*) no sentido de validar e testar os circuitos multiplicadores alvo deste trabalho.

A simulação elétrica é o método mais preciso para a verificação das restrições de atraso de um circuito CMOS (*Complementary Metal Oxide Semiconductor*). Simuladores elétricos como o SPICE representam o circuito como uma rede de elementos passivos e ativos (resistores, capacitores, indutores e fontes controladas) e resolvem o sistema de equações diferenciais ordinárias relativo a descrição do circuito para cada intervalo de tempo (*time step*) da simulação que está em funcionamento. Infelizmente, simuladores sob o nível elétrico requerem uma grande quantidade de tempo mesmo para circuitos relativamente pequenos, o que torna difícil a sua utilização para a determinação precisa do desempenho de projetos digitais no estado da arte. Uma alternativa à simulação elétrica é a simulação lógica de atraso. Este tipo de simulação é realizada por simuladores elétricos simplificados como o SSTVelocity [7], por exemplo. A simulação de atraso é mais rápida que a simulação elétrica porque utiliza modelos

menos precisos, o que por sua vez verifica resultados mais distantes da realidade dos resultados obtidos por uma simulação elétrica.

O processo para obtenção dos resultados deste trabalho fez uso de diversas ferramentas de análise e de síntese de circuitos integrados como foi apresentado no diagrama da FIGURA 10. O teste destes circuitos mereceu especial atenção por se tratar de uma parte importante deste processo. Os testes realizados em um projeto recaem em duas grandes categorias [1]. A primeira delas verifica se o circuito em questão realiza de fato a função pretendida, isto é, no caso dos circuitos deste trabalho, se os mesmos calculam corretamente as operações de multiplicação sobre os números envolvidos nos vetores de teste. Em outras palavras, estes testes asseguram que todas as portas lógicas no circuito ativam a função desejada. Estes tipos de testes são geralmente utilizados nas etapas iniciais do projeto e podem ser denominados como **testes funcionais**. A segunda categoria de testes verifica se todas as portas lógicas e registradores no *chip* funcionam corretamente. Estes testes são utilizados após o processo de fabricação em silício do circuito e são chamados de **testes industriais**.

Muitas vezes estes testes podem ser compactados em um único. Entretanto, o fluxo natural de projeto leva em consideração a funcionalidade antes das preocupações com a fabricação. Neste trabalho, adota-se a realização em separado das duas categorias de testes acima apresentadas. Em uma delas verifica-se a fase de prototipação em FPGA e simulações lógicas utilizando o programa ModelSim. A outra diz respeito aos testes elétricos com os *netlists* extraídos e os testes experimentais com os *chips*.

Neste Capítulo serão apresentadas a maneira como foi realizada a validação funcional destas arquiteturas de multiplicadores e as características do processo de implementação dos vetores de teste. A seguir, serão apresentados os resultados de dissipação de potência obtidos com a análise física em nível de simulação e experimental com os multiplicadores *array* Binário e em código Híbrido e com o multiplicador Booth Modificado. Todos os circuitos foram desenvolvidos em 8 bits. O atraso dos multiplicadores é extraído a partir da ferramenta SSTVelocity através da análise do caminho crítico. Os resultados de consumo de potência são apresentados com os valores obtidos com o simulador elétrico ELDO e Mach PA, de acordo com a análise dos circuitos extraídos para o formato SPICE. O simulador elétrico Mach PA foi utilizado para a obtenção dos resultados de dissipação de potência dos circuitos multiplicadores resultantes da síntese das arquiteturas sem otimizações de atraso. A seguir são apresentados os principais aspectos para a obtenção dos parâmetros de área, atraso e consumo de potência. Após, são apresentados os resultados obtidos nos processos de simulação e medição dos chips prototipados.

5.1. ESTRUTURA DE PROJETO DE TESTES

Os testes de circuitos integrados atualmente encontram-se entre os processos mais onerosos e com mais problemas envolvidos dentro de um ciclo de projeto, revelando desta maneira uma constante necessidade de inovação em testes de *chips* [13].

Para a maioria dos sistemas, testes de funcionalidade envolvem a necessidade de provar que o circuito é funcionalmente equivalente a uma determinada especificação, sendo usualmente os primeiros testes que um projetista deve construir como parte do processo de desenvolvimento do circuito.

Neste trabalho foi desenvolvido um procedimento de testes que teve por objetivo principal submeter as arquiteturas de multiplicadores aos mesmos vetores de teste em todas as etapas de projeto. A idéia surgiu da necessidade de testar os circuitos prototipados sem a utilização de um analisador lógico que não era um equipamento disponível, mas sim utilizando os recursos presentes.

A FIGURA 17 apresenta os diferentes formatos de vetores de teste utilizados neste trabalho. A descrição dos vetores inicial é em formato SLS (*Switch Level Simulator*). Os vetores neste formato foram utilizados em [18] para análise do consumo de potência dos multiplicadores. Desta forma, neste trabalho estes vetores foram utilizados como ponto de partida como mostra o diagrama da FIGURA 17.

A FIGURA 18 apresenta a sintaxe dos vetores em formato SLS para um exemplo de 2 bits em código Binário para uma profundidade de 50 vetores. As especificações *l* e *h* representam níveis lógicos baixo e alto, respectivamente.

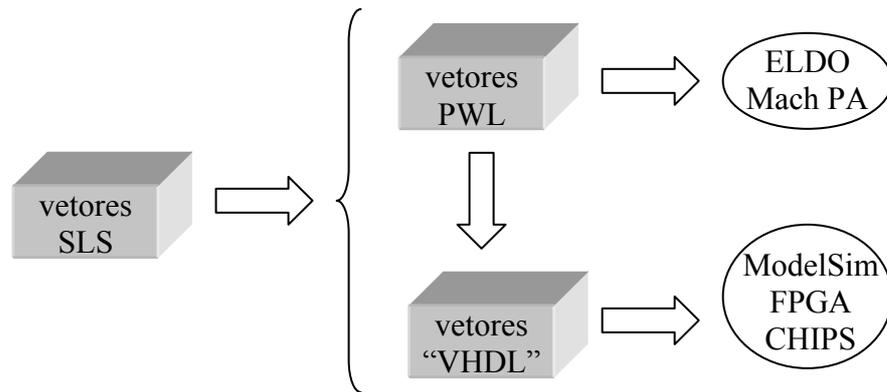


FIGURA 17 – Diagrama dos vetores de teste utilizados

```

set R7 = 1*2 1*2 h*2 h*2 h*2 h*2 1*2 h*2 h*2 h*2 h*2 h*2
1*2 h*2 1*2 1*2 h*2 1*2 h*2 1*2 1*2 h*2 h*2 1*2 h*2 h*2
h*2 h*2 h*2 h*2 1*2 h*2 1*2 h*2 1*2 1*2 h*2 1*2 h*2 1*2
h*2 1*2 h*2 1*2 h*2 h*2 1*2 h*2 h*2 h*2 h*2
set R6 = 1*2 1*2 1*2 h*2 h*2 h*2 1*2 1*2 h*2 1*2 1*2 1*2
h*2 1*2 h*2 h*2 h*2 1*2 1*2 1*2 1*2 1*2 h*2 1*2 h*2 h*2
1*2 h*2 1*2 h*2 1*2 h*2 h*2 h*2 h*2 1*2 h*2 h*2 h*2 1*2
h*2 1*2 1*2 1*2 h*2 h*2 1*2 h*2 h*2 1*2 1*2
.
.
.

```

FIGURA 18 – Representação de vetores randômicos em formato SLS

No nível mais baixo de abstração, outros formatos de vetores são necessários para testes dos multiplicadores. Em particular, os formatos PWL (*Piece Wise Level*) e VHDL são utilizados, como mostrado na FIGURA 17. Para a conversão do formato dos vetores da descrição SLS para as descrições PWL e VHDL, foram utilizados programas acadêmicos.

O programa *geravet* lê um arquivo com vetores de teste em formato SLS e gera um arquivo em formato PWL para cada *bit* de entrada. É possível definir qual será a tensão do nível lógico alto (5V neste trabalho), o nome das variáveis dos *bits* de saída (R para o exemplo da FIGURA 19), a frequência em MHz em que os vetores serão gerados, bem como os tempos de subida e descida da forma de onda (*tr* e *tf*). A FIGURA 19 exemplifica uma linha de comando utilizada para se obter os vetores em formato PWL utilizando o arquivo *rand8b100hib* em formato SLS.

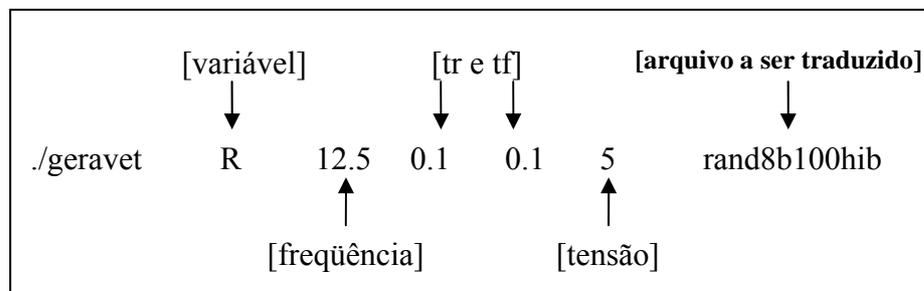


FIGURA 19 – Exemplo de linha de comandos do programa *geravet*

O programa *spice2vdhl* utiliza arquivos em formato PWL como entrada e retorna uma matriz de vetores em formato VHDL contendo os valores lógicos de cada *bit*. Esta matriz de vetores é instanciada para um programa em VHDL que a manipula de maneira a mostrar os resultados idênticos aos vetores em formato PWL. Não há neste programa opções de escolha de frequência, tempos de subida e descida. Estes valores são definidos por constantes dentro do código VHDL, tornando a matriz gerada pelo *spice2vdhl* independente destes fatores.

A FIGURA 21 mostra o diagrama de pinos da entidade VHDL responsável por manipular os vetores de teste. As portas de saída da direita representam as entradas R0 a R7 e T0 a T7, de 8 bits cada, que servem de fonte de excitação, tanto para os testes lógicos, quanto para os testes experimentais. As portas da esquerda são pinos necessários para controlar a placa DIO2 [32]. Esta placa, mostrada na FIGURA 20, abriga uma CPLD (*Complex Programmable Logic Device*) e contém diversos componentes como *leds*, botões, chaves e *display* que auxiliam em testes de circuitos digitais.

Após a conversão de formato dos vetores, os mesmos podem ser utilizados para simulações dos multiplicadores. Para tal finalidade, a ferramenta ELDO é utilizada para simulação elétrica, a ferramenta ModelSim é utilizada para simulação lógica e um osciloscópio com 16 canais digitais auxilia na visualização das saídas da FPGA. A FIGURA 22 mostra o formato dos vetores na ferramenta ELDO para 2 entradas em código Binário de 8 bits cada para 50 vetores randômicos. A FIGURA 23 mostra o formato dos vetores na ferramenta ModelSim para 2 entradas em código Binário de 8 bits cada para 100 vetores randômicos. Finalmente a FIGURA 24 mostra a tela do osciloscópio digital com a saída da FPGA de 2 entradas em código Binário de 8 bits cada para 100 vetores randômicos.

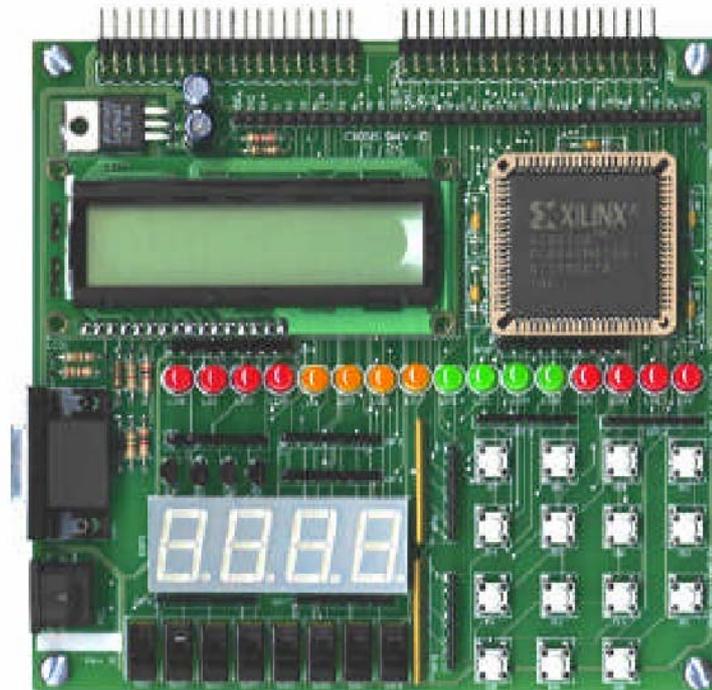


FIGURA 20 – Placa de periféricos DIO2

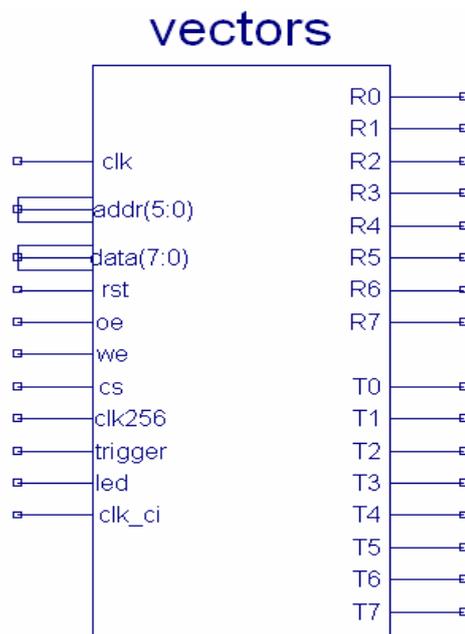


FIGURA 21 – Diagrama de portas de entrada e saída do programa em VHDL para inserção dos vetores de teste

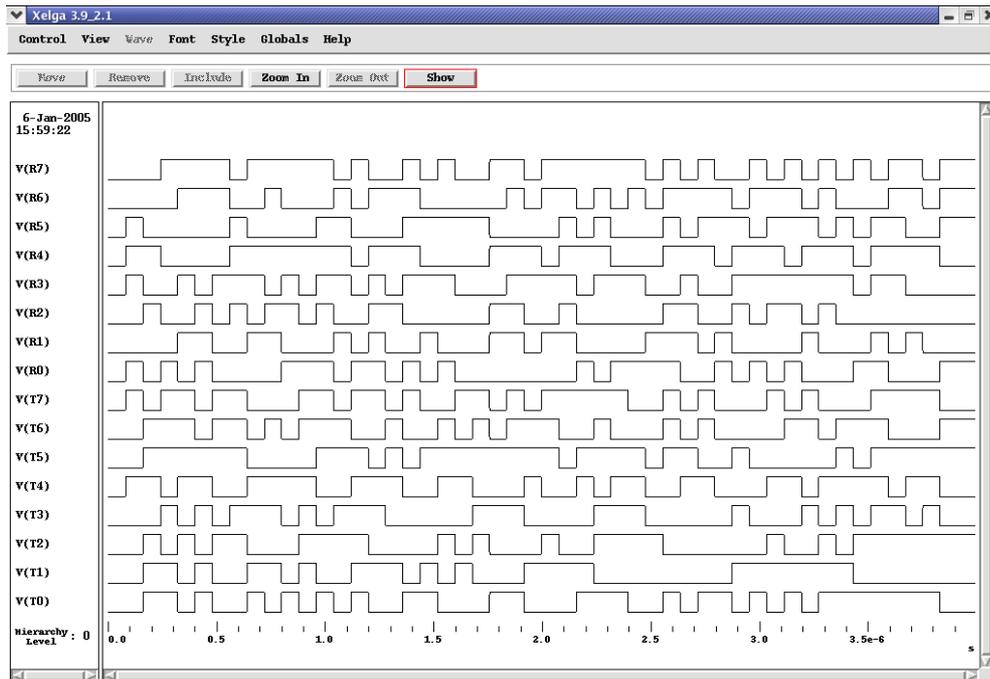


FIGURA 22 – Vetores randômicos resultantes de simulação elétrica com o programa ELDO

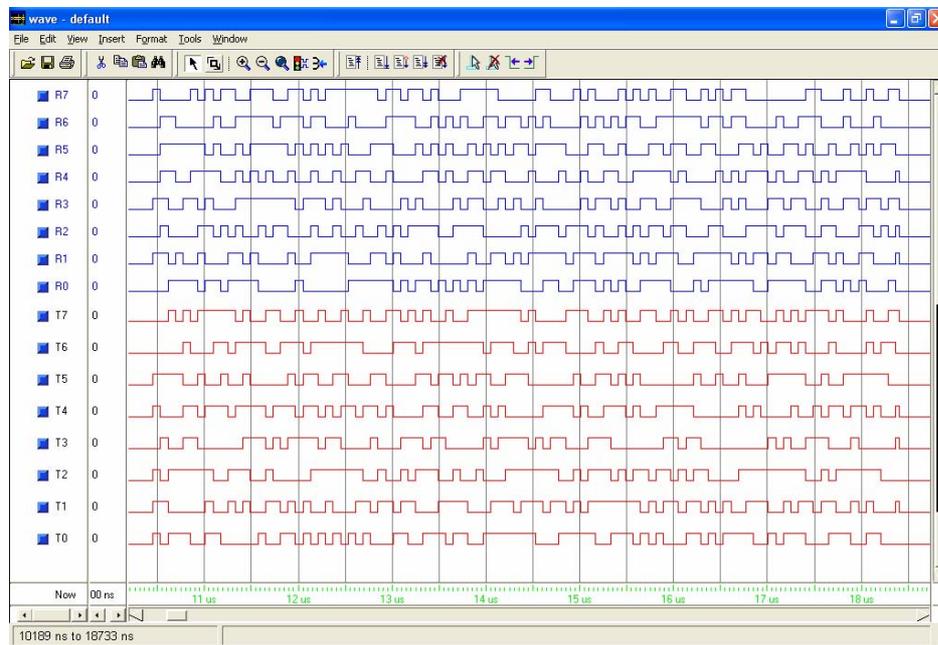


FIGURA 23 – Vetores randômicos resultantes da simulação com o programa ModelSim

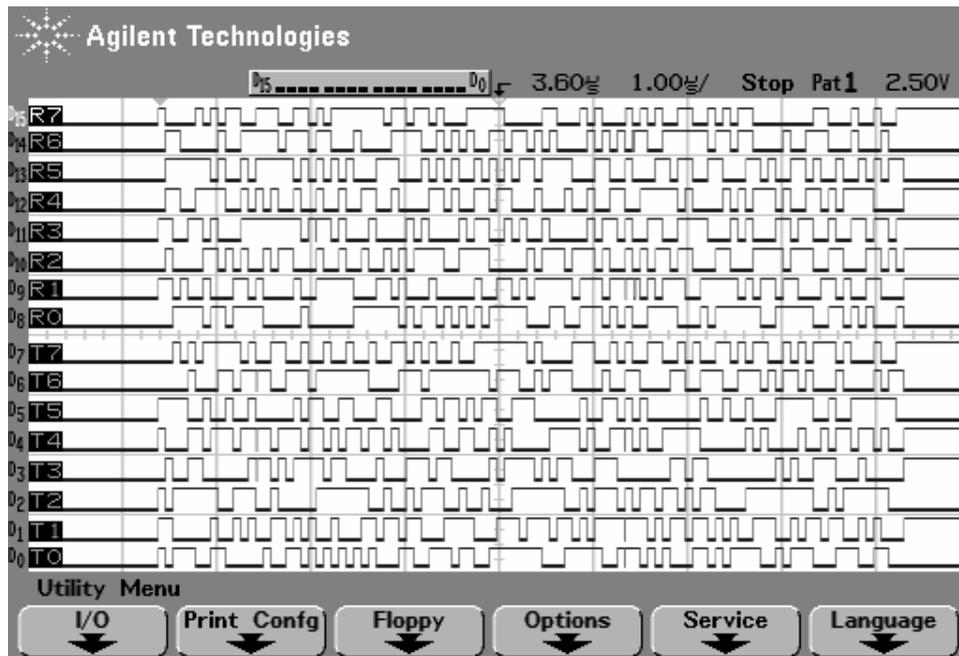


FIGURA 24 – Tela do osciloscópio digital com a saída da FPGA

Deve ser observado que o tempo requerido para uma simulação lógica dos circuitos utilizando o programa ModelSim para um grupo de vetores com uma profundidade x ou $10x$ não é muito diferente e não apresenta valores absolutos de utilização de máquina proibitivos sob o ponto de vista prático. Entretanto, uma simulação elétrica das arquiteturas consome muito tempo, podendo em determinadas circunstâncias alcançar mais de 200 horas, o que se traduz em uma dificuldade muito grande no momento de simular os circuitos para uma profundidade de vetores elevada. De uma forma parecida, o mesmo problema pode ser atribuído à utilização de uma FPGA. A barreira neste caso é armazenar todos os vetores dentro do dispositivo. Diante deste quadro, foram então utilizados vetores de teste até uma profundidade igual a 100. Este número permitiu uma simulação adequada dos circuitos em um tempo satisfatório

Em termos de validação funcional, as arquiteturas de multiplicadores foram submetidas aos mesmos vetores de testes utilizados para a validação lógica. Após a extração em um *netlist* SPICE, os circuitos foram simulados eletricamente com o simulador ELDO. No âmbito da simulação elétrica, foram escolhidas saídas aleatórias e observadas as entradas que geraram estas saídas. Os números de saída foram então manualmente verificados quanto ao seu correto valor em complemento de dois. O mesmo procedimento foi aplicado para os testes experimentais.

5.2. DESENVOLVIMENTO DO FLUXO DE PROJETO PARA A OBTENÇÃO DOS PARÂMETROS DE ÁREA, ATRASO E CONSUMO DE POTÊNCIA

Para a obtenção dos dados de área foi utilizada a ferramenta IC Station. As arquiteturas de multiplicadores foram submetidas ao processo de síntese automática, onde as etapas como o posicionamento automático das células, definição das bandas de roteamento, geração do *padframe* e roteamento dos canais de ligação entre as células dos circuitos, foram realizadas de maneira a seguir o mesmo padrão para todas as arquiteturas.

O auto posicionamento é realizado pelo *software* para estabelecer os locais onde cada célula deve ser posicionada. Esta etapa representa o processo de gerar um layout a partir de um projeto pré-existente [5]. Imediatamente anterior ao auto posicionamento é realizada a etapa de *autofloorplan*, onde são definidos os limites em termos de área em que

as células devem ser organizadas. Após esta etapa, as células básicas que compõem o circuito são inseridas entre as bandas definidas na etapa de *autofloorplan*. O roteamento automático ocorre em duas etapas. A primeira delas é após o auto posicionamento, quando são realizadas as interconexões entre os blocos básicos que constituem o circuito e são preenchidas as bandas de condução criadas na etapa de *autofloorplan*. A segunda etapa diz respeito ao roteamento das portas de entrada e saída do núcleo do circuito que se conectam diretamente aos PADS existentes no *padframe*.

O tipo de encapsulamento utilizado para os protótipos foi o cerâmico de 40 pinos ou DIP40 [36]. Os vetores de teste utilizados experimentalmente foram inseridos nas entradas dos *chips* de maneira cíclica, ou seja, cada grupo de 100 vetores era repetido após uma inserção completa. Esta medida tornou-se necessária devido a limitações de armazenamento da FPGA e capacidade de medição do multímetro para os intervalos de tempo utilizados nas simulações.

Os circuitos foram simulados eletricamente utilizando o programa ELDO. A tensão de referência para a tecnologia AMI 0.5 μ é 5V. Os modelos dos transistores foram adquiridos através da página da MOSIS na internet que, após cada rodada de fabricação, realiza testes para extração dos parâmetros elétricos de simulação.

Foram utilizados vetores de teste randômicos e correlacionados ou em formato senoidal. Os vetores randômicos apresentam valores aleatórios dados para cada uma das duas entradas de 8 bits. Os vetores correlacionados descrevem duas ondas em formato senoidal defasadas em 90 graus.

Os circuitos *latches* estão continuamente sendo excitados pelo sinal de *clock* e alguns deles requerem potência da fonte pela transição de seus transistores, de maneira que a potência destes circuitos seqüenciais está em sincronismo com o sinal de *clock*. O mesmo não é verdade para as portas dentro da lógica combinacional. Mesmo que as entradas para um bloco lógico combinacional sejam atualizadas pelos *latches*, as portas internas deste bloco podem sofrer diversas transições antes de estabelecer o resultado em estado estacionário para aquele período de *clock* [37].

A FIGURA 25 apresenta o esquema de montagem empregado para a obtenção dos resultados de consumo de potência dos testes experimentais. Um cabo de transmissão de dados utilizado comercialmente em discos rígidos foi utilizado para comunicar a placa de FPGA e a placa de testes que recebe os circuitos prototipados, servindo então para a inserção dos vetores de teste nos *chips*.

O multímetro apresentado foi configurado para a leitura de corrente média e conectado em série entre a fonte de tensão e o pino de VDD do circuito sobre teste. A fonte de tensão contínua foi ajustada em 5V.

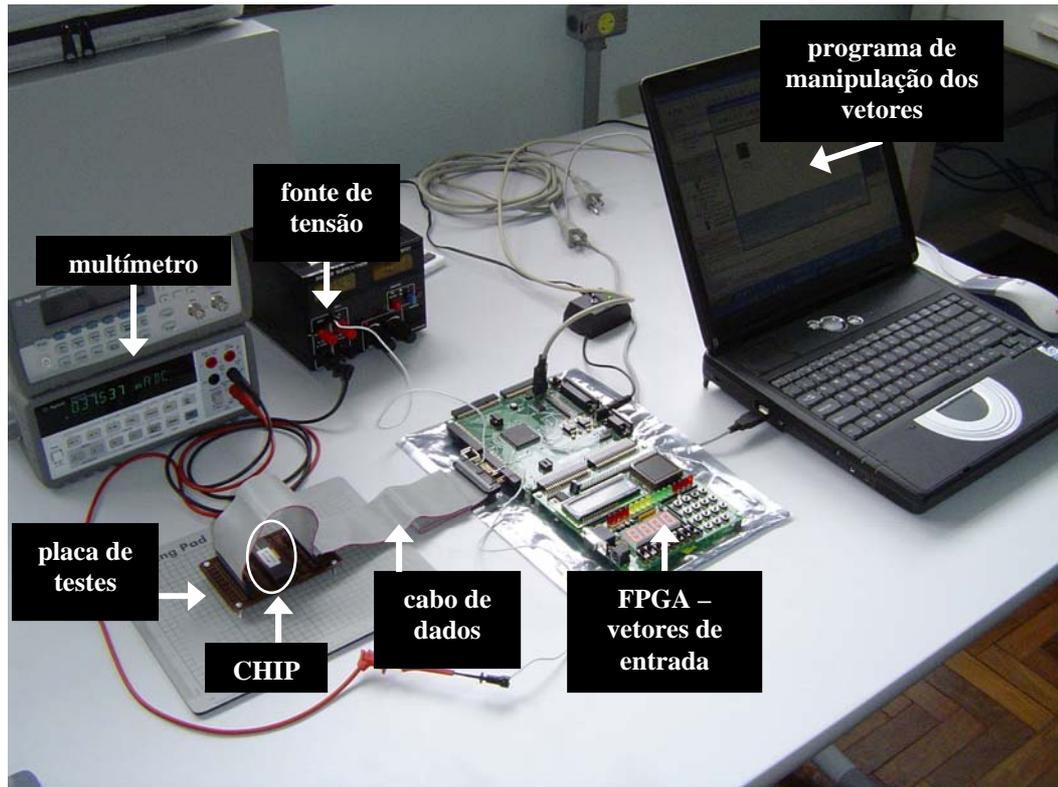


FIGURA 25 – Esquema de montagem dos testes experimentais

5.3. RESULTADOS DE CONSUMO DE POTÊNCIA

A TABELA 5 apresenta os resultados de dissipação de potência para os testes experimentais com os protótipos. A frequência de 6,5MHz foi utilizada para a excitação dos vetores de teste.

TABELA 5 – Resultados experimentais de dissipação de potência para os chips multiplicadores

	Randômico (mW)	%	Senoidal (mW)	%
Booth	167,5	–	138,0	–
Array Binário	154,5	-7,76	131,0	-5,07
Array Híbrido	161,5	-3,58	121,0	-12,31

Deve-se observar na TABELA 5 que para sinais randômicos, aplicados às entradas dos multiplicadores, onde o aspecto da correlação não está presente, tem-se uma maior redução de potência para o multiplicador *array* Binário. Isto se explica pela maior regularidade e menor quantidade de *glitching* apresentada por esta arquitetura. O aspecto da maior regularidade também é benéfico para o multiplicador Híbrido, que apresenta uma maior redução de potência em relação ao multiplicador Booth. Este aspecto se torna ainda mais relevante quando considerada a correlação dos sinais. Neste caso, a aplicação de vetores em formato senoidal às entradas dos multiplicadores, faz com que o multiplicador Híbrido apresente a maior redução do consumo de potência, como mostrado na TABELA 5. Este resultado está em acordo com a definição do código Híbrido, que apresenta uma menor atividade de chaveamento para sinais correlacionados. Além disto, é importante verificar a diferença entre os resultados absolutos de dissipação de potência obtidos a partir dos testes com os *chips* na TABELA 5 e, os valores de simulação elétrica na TABELA 6. Um dos fatores que provoca esta diferença é a interferência dos PADS nos resultados experimentais. Enquanto que na simulação este aspecto não foi considerado, o teste experimental leva em conta todos os fatores e imperfeições inerentes a um processo de medição real, tais como a capacitância das interconexões, interferência de outros equipamentos, precisão dos instrumentos de medida e os PADS do circuito. Algumas ações mitigadoras podem ser tomadas para reduzir este efeito e alcançar uma maior aproximação entre os resultados simulados e medidos. Entre elas está o tratamento

da questão de isolamento do núcleo do circuito dos PADS, considerando a medição da dissipação de potência. Um método indireto é prever a medição experimental do consumo de dois PADS através de um curto circuito pré-estabelecido do desenho das máscaras de *layout* que seguem para a fábrica. Com isto, é possível determinar a dissipação de potência para dois PADS e estimar qual seria a potência total multiplicando-se este resultado, dividido por dois, pelo número de PADS total do circuito. Uma outra abordagem pode ser conferida ao estabelecimento de contatos do núcleo do circuito com o meio externo, acessíveis através de microponteiras. Este processo, entretanto, necessita de equipamentos muito mais sofisticados.

Tem sido observado neste trabalho que o aspecto da profundidade dos vetores aplicados às entradas dos circuitos multiplicadores pode provocar um impacto na dissipação de potência. A TABELA 6 apresenta resultados de dissipação de potência para as três arquiteturas de multiplicadores com diferentes tamanhos de vetores. Cada circuito foi simulado pela ferramenta Mach PA para 4 diferentes profundidades de vetores: 50, 100, 500 e 1000. Como pode ser observado na TABELA 6, para a arquitetura Híbrida além do aspecto da correlação do sinal, a profundidade dos vetores utilizados também provoca um impacto na redução do consumo de potência. Quando são aplicados vetores em formato senoidal com profundidades de 500 e 100 vetores a arquitetura Híbrida apresenta melhores valores. Entretanto, a arquitetura Binária apresenta os melhores valores de consumo de potência, independente do tipo e da profundidade dos

vetores aplicados. Estes resultados obtidos no baixo nível de abstração confirmam os resultados obtidos em [9] no nível lógico.

TABELA 6 – Dissipação de potência para diferentes profundidades de vetores

		Randômico (mW)	Dif. %	Senoidal (mW)	Dif. %
Booth	50	4,90	–	3,40	–
	100	4,85	–	3,33	–
	500	4,81	–	2,27	–
	1000	4,72	–	1,31	–
Array Binário	50	3,90	-20,48%	2,78	-18,16%
	100	4,02	-17,00%	2,63	-20,97%
	500	3,96	-17,62%	1,72	-24,05%
	1000	4,04	-14,28%	1,08	-17,52%
Array Híbrido	50	4,75	-3,02%	3,23	-4,90%
	100	4,88	+0,69%	2,93	-12,17%
	500	4,87	+1,28%	1,82	-19,88%
	1000	4,88	+3,53%	1,10	-16,09%

]

5.4. RESUMO

Diferentes ferramentas de CAD foram utilizadas no sentido de validar e testar os circuitos multiplicadores alvo deste trabalho. Foi apresentada a maneira pela qual os vetores de teste foram gerados e os modos de inserção considerando os diversos níveis de abstração dos circuitos multiplicadores. Os mesmos vetores de teste foram implementados na simulação lógica com o *software* ModelSim, na simulação elétrica com o simulador ELDO e em testes experimentais através da utilização de uma placa de FPGA. Os vetores de teste são utilizados nas etapas de validação funcional e aquisição de dados de consumo de potência.

Neste capítulo, ainda foram abordados os resultados obtidos com as arquiteturas de multiplicadores. Foram realizados testes de simulação e testes experimentais e adquiridos dados consumo de potência. Embora as arquiteturas *array* Binário e Híbrido possuam perdas em termos de área ocupada para o multiplicador Booth Modificado, ambas as arquiteturas obtiveram ganhos em consumo de potência. O multiplicador *array* Binário manteve uma vantagem em termos de dissipação de potência nos testes experimentais e nas simulações elétricas para todos os resultados obtidos com a inserção de vetores de teste em formato senoidal. A vantagem do multiplicador *array* Híbrido em relação ao Binário mereceu especial atenção devido aos resultados obtidos por simulação quando se fez aumentar a profundidade dos vetores de teste correlacionados.

6. CONCLUSÕES E TRABALHOS FUTUROS

A excessiva dissipação de potência em circuitos integrados pode desestimular seu uso em ambientes portáteis, reduzindo a aceitação do mercado para aplicações móveis. Desta maneira, a redução do consumo de energia elétrica e dissipação média de potência têm se tornado um dos tópicos de mais rápido crescimento na indústria eletrônica e uma das áreas com maiores desafios de pesquisa neste campo.

A redução do número de transições no sentido de minimizar o consumo de potência é o principal objetivo de técnicas que usam diferentes esquemas de codificação em barramentos. Neste trabalho foram apresentadas novas arquiteturas de multiplicadores em diferentes codificações de dados. Estas novas arquiteturas de multiplicadores foram apresentadas em [9] no nível lógico e foram validadas neste trabalho no nível físico. Os resultados obtidos no nível físico comprovam os resultados apresentados em [9] no nível lógico.

Os testes de simulação nos circuitos multiplicadores tipo *array* revelaram que estas arquiteturas, apesar de precisarem de mais área e possuir um maior número de transistores, dissiparam menos potência em todas as simulações e testes experimentais realizados com vetores em formato senoidal. De fato, o efeito da correlação destes vetores em

formato senoidal, observado nos resultados de dissipação de potência, está diretamente ligado ao número de vetores. Neste sentido, verificou-se que para o circuito multiplicador Híbrido, o efeito da correlação dos sinais se faz mais presente, tendo esta arquitetura apresentado um menor consumo de potência quando utilizados vetores em formato senoidal com maiores profundidades. Entretanto, os melhores resultados observados foram para a arquitetura de multiplicador Binário, que independente do tipo e tamanho do vetor, apresentou os melhores resultados de redução de consumo de potência.

Os resultados experimentais comprovaram a assertiva da influência da correlação dos sinais, visto que o multiplicador *array* Híbrido consumiu menos potência do que o Binário para a inserção contínua de um grupo de vetores correlacionados. Este aspecto serve como indicativo de que para um maior número de vetores de teste, poderá ocorrer uma inversão da vantagem do consumo de potência entre as duas arquiteturas de multiplicadores *array*, com o Híbrido tirando o maior proveito.

Os testes de circuitos integrados atualmente encontram-se entre os processos mais onerosos e com mais problemas envolvidos dentro de um ciclo de projeto, revelando desta maneira uma constante necessidade de inovação em testes de *chips* [13]. Embora não tenha sido o principal enfoque deste trabalho, desenvolveu-se também um fluxo de análise de circuitos integrados digitais. Partindo-se de uma descrição em VHDL, passou-se por simulações lógicas, síntese física,

simulações elétricas e testes experimentais, dentro de um ambiente profissional de desenvolvimento de circuitos integrados.

Em continuidade aos trabalhos realizados nesta dissertação, sugerem-se as seguintes atividades a serem desenvolvidas:

- implementação das arquiteturas de 16 bits dos circuitos multiplicadores trabalhados seguindo o fluxo de projeto apresentado, com verificação das arquiteturas trabalhadas utilizando $m=4$ e, estudo da influência do código Híbrido em arquiteturas de maior número de bits;
- implementação de um bloco de controle de acesso a uma memória externa no código VHDL do programa de manipulação dos vetores de teste. Este procedimento libera espaço no dispositivo FPGA e aumenta a capacidade de inserção de vetores em termos de profundidade, além de permitir o aprimoramento do código. Desta forma, novas funções de verificação/programação podem ser inseridas no código;
- análise do comportamento destes circuitos inseridos em unidades aritméticas lógicas e circuitos da área de Processamento Digital de Sinais (*Digital Signal Processing – DSP*).

7. BIBLIOGRAFIA

- [1] WEST, N. H. E.; ESHRAGHIAN, K. **Principles of CMOS VLSI Design – A Systems Perspective**. Santa Clara, CA – USA: AT&T, 1994.
- [2] PAYNE, D. **Circuit Simulation for DSM IC Power Analysis**, 2000. Disponível em <<http://www.mentor.com/dsm>>. Acesso em: 10 mar. 2003.
- [3] UYEMURA, J. P. **Circuit Design for CMOS VLSI**. USA: Kluwer Academic Publishers, 1992.
- [4] UYEMURA, J. P. **Fundamentals of CMOS Digital Integrated Circuits**. USA: Addison-Wesley Publishing Company, 1988.
- [5] **Designing ASICs with the ADK Design Kit and Mentor Graphics Tools Version 2.0**. Disponível em <<http://www.mentor.com>> condicionado a contrato com a empresa Mentor Graphics ©. Acesso em: 10 mar. 2003.
- [6] **Leonardo Spectrum User's Manual**. Disponível em <<http://www.mentor.com>> condicionado a contrato com a empresa Mentor Graphics ©. Acesso em: 8 out. 2003.
- [7] **SSTVelocity User's Manual**. Disponível em <<http://www.mentor.com>> condicionado a contrato com a empresa Mentor Graphics ©. Acesso em: 8 out. 2003.
- [8] GÜNTZEL, J. A. **Functional Timing Analysis of VLSI Circuits Containing Complex Gates**. 2000. 182f. Tese (Doutorado em

- Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
- [9] COSTA, E. A. C. **Operadores Aritméticos de Baixo Consumo para Arquiteturas de Circuitos DSP**. 2002. 193f. Tese (Doutorado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.
- [10] UFSM. **Estrutura e Apresentação de Monografias, Dissertações e Teses**. 5ª ed. Santa Maria. 2000.
- [11] CALLAWAY, T.K. SWARTZLANDER, E.E., Jr. **Optimizing Multipliers for WSI**, Fifth Annual IEEE International Conference on Wafer Scale Integration (Proceedings), San Francisco – CA USA , p. 85 – 94, Jan. 1993.
- [12] CHENG-Wen Wu **On Energy Efficiency of VLSI Testing**, Sixth Asian Test Symposium (ATS '97 Proceedings), Akita Japan, p. 132 – 137, Nov. 1997.
- [13] GIRARD, P. **Survey of low-power testing of VLSI circuits**, IEEE Design & Test of Computers, Montpellier – France, p. 80 – 90, May-June 2002.
- [14] WILLIAMS, T.W. DENNARD, R.H. KAPUR, R. MERCER, M.R. MALY, M. **Iddq test: sensitivity analysis of scaling**, Test Conference, Washington, DC – USA, p. 786 – 792, Oct. 1996.
- [15] MARTIN, Kenneth W. **Digital Integrated Circuit Design**, New York, New York – USA: Oxford University Press, 2000.
- [16] VEENDRICK, H. **Short-Circuit Dissipation of Static CMOS Circuits and Its Impact on the Design of Buffer Circuits**,

- IEEE Journal of Solid-State Circuits, New York –USA, v. 19, p. 468 – 473, Aug. 1984.
- [17] RABAEY, Jan M. **Digital Integrated Circuits – A Design Perspective**, USA, Prentice Hall, 1996.
- [18] COSTA, E.; BAMPI, S.; MONTEIRO, J. **A New Architecture for Signed Radix-2^m Pure Array Multipliers**, Computer Design: VLSI in Computers and Processors, Proceedings, p. 112 – 117, Sept. 2002.
- [19] <http://www.mosis.com/about/whatis.html>, Acesso em: 12 jan./05.
- [20] COSTA, E.; BAMPI, S.; MONTEIRO, **Power Efficient Arithmetic Operand Encoding**, 14th Symposium on Integrated Circuits and Systems Design, Pirenópolis Brazil, p. 201 – 206, Sept. 2001.
- [21] **QuickPath Training Workbook**. Disponível em <<http://www.mentor.com>> condicionado a contrato com a empresa Mentor Graphics ©. Acesso em: 8 out. 2003.
- [22] **QuickPath User's and Reference Manual**. Disponível em <<http://www.mentor.com>> condicionado a contrato com a empresa Mentor Graphics ©. Acesso em: 8 out. 2003.
- [23] ABU-KHATER, I.S.; BELLAOUAR, A.; ELMASRY, M.I. **Circuit Techniques for CMOS Low-Power High-Performance Multipliers**, IEEE Journal of Solid-State Circuits, p. 1535 – 1546, Oct. 1996.
- [24] **ELDO User's Manual**. Disponível em <<http://www.mentor.com>> condicionado a contrato com a empresa Mentor Graphics ©. Acesso em: 8 out. 2003.

- [25] **XELGA User's Manual**. Disponível em
<<http://www.mentor.com>> condicionado a contrato com a
empresa Mentor Graphics ©. Acesso em: 8 out. 2003.
- [26] **IC Station User's Manual**. Disponível em
<<http://www.mentor.com>> condicionado a contrato com a
empresa Mentor Graphics ©. Acesso em: 8 out. 2003.
- [27] **Design Architect-IC User's Manual**. Disponível em
<<http://www.mentor.com>> condicionado a contrato com a
empresa Mentor Graphics ©. Acesso em: 8 out. 2003.
- [28] **ModelSim User's Manual**. Disponível em
<<http://www.mentor.com>> condicionado a contrato com a
empresa Mentor Graphics ©. Acesso em: 8 out. 2003.
- [29] http://www.xilinx.com/support/sw_manuals/xilinx6/index.htm
Acesso em: 5 jan. 2005.
- [30] COSTA, E.; BAMPI, S.; MONTEIRO, J. **A New Architecture
for 2's Complement Gray Encoded Array Multiplier**, 15th
Symposium on Integrated Circuits and Systems Design,
Proceedings, p. 14 – 19, Sept. 2002.
- [31] **Digilent 2E System Board – Reference Manual**. Disponível em
<<http://www.digilentinc.com>>. Acesso em: 23 out. 2004.
- [32] **Digilent DIO2 – Reference Manual**. Disponível em
<<http://www.digilentinc.com>>. Acesso em: 23 out. 2004.
- [33] NAGEL, W. **SPICE2, A Computer Program to Simulate
Semiconductor Circuits**. Berkeley, California: University of
California, Department of Electrical Engineering and Computer
Sciences, 1975. 63p. (UCB/ERL M75/520).

- [34] RUSHTON, Andrew, **VHDL for Logic Synthesis** – Second Edition, John Wiley & Sons, England, 1998.
- [35] HWANG, K., **Computer Arithmetic – Principles, Architecture and Design**. School of Electrical Engineering, 1979.
- [36] **Package Handbook – DIP40** Disponível em <http://www.mosis.com/Technical/Packaging/Ceramic/menu-pkg-ceramic.html> Acesso em: 8 out 2003.
- [37] NAJM, F.N. **A Survey of Power Estimation Techniques in VLSI Circuits**, IEEE Transactions on Large Scale Integration (VLSI) Systems, Urbana – IL, USA, p. 446 – 455, Dec. 1994.
- [38] SHEN, A.; GHOSH, A.; DEVADAS, S.; KEUTZER, K. **On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks**, IEEE/ACM International Conference on Computer-Aided Design – ICCAD-92, Santa Clara – CA, USA, p. 402 – 407, Nov. 1992.
- [39] REIS, R. A. da L. **Concepção de Circuitos Integrados**. Porto Alegre – RS, Brasil, Editora Sagra Luzzatto, 2000.
- [40] **Mach TA User's and Reference Manual**. Disponível em <http://www.mentor.com> condicionado a contrato com a empresa Mentor Graphics ©. Acesso em: 8 out. 2003.
- [41] ZORIAN, Y. **A Distributed BIST Control Scheme for Complex VLSI Devices**, VLSI Test Symposium, Atlantic City – NJ, p. 4 – 9, Apr. 1993.
- [42] Kim, N. S.; Blaauw, D.; Mudge, T. **Leakage Power Optimization Techniques for Ultra Deep Sub-Micron Multi-**

- Level Caches**, International Conference on Computer Aided Design ICCAD-2003. p. 627 – 632, Nov. 2003.
- [43] Kim, C.H.; Roy, K. **Dynamic V_{TH} Scaling Scheme for Active Leakage Power Reduction**, Design, Automation and Test in Europe Conference and Exhibition, Proceedings, Paris – France p. 163 – 167, Mar. 2002.
- [44] Gowan, M. K.; Biro, L. L.; Jackson, D. B. **Power Considerations in the Design of the Alpha 21264 Microprocessor**, Design Automation Conference, p. 726 – 731, Jun. 1998.
- [45] Penzes, P. L.; Martin, A. J. **An energy Estimation Method for Asynchronous Circuits with Application to an Asynchronous Microprocessor**, Design, Automation and Test in Europe Conference and Exhibition, Paris, France, p. 640 – 647, March 2002.
- [46] Shen, A.; Ghosh, A.; Devadas, S.; Keutzer, K. **On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks**, International Conference on Computer-Aided Design ICCAD-92, Santa Clara, CA – USA, p. 402 – 407, Nov. 1992.

Anexo 1

Em relação à dissipação de potência dos circuitos são utilizados alguns conceitos que definem a maneira pela qual esta componente pode ser considerada:

- Energia dissipada: é a energia total fornecida ao circuito, que é dissipada devido a correntes elétricas (DC) e devido à atividade de comutação;
- Potência média: a potência média equivale a toda distribuição de potência sob o período de análise. A divisão do montante de energia pelo tempo sob o qual um determinado circuito permaneceu em teste fornece a potência média [13];
- Potência instantânea: a potência instantânea é a potência consumida em um dado instante de tempo. Comumente ela é definida com a potência consumida logo após a aplicação de um sinal sincronizado de *clock*. Potências instantâneas elevadas podem sobrecarregar a distribuição de potência de um sistema no Silício ou no encapsulamento, causando danos [13];
- Potência de pico: é o maior valor de potência em dado instante. Este valor é muito importante para definir determinadas características de um dispositivo. A potência de pico define os componentes térmicos e limites elétricos

e ainda, os requisitos para o encapsulamento do *chip*. Se a potência de pico exceder um determinado limite, os projetistas não podem mais garantir que o sistema como um todo funcionará corretamente. De fato, o espaço de tempo para determinar a potência de pico está relacionado com a capacidade térmica do *chip*, de maneira que forçar esta janela para um período de *clock* é muitas vezes uma abordagem simplista. Por exemplo, considerando um circuito que possui um pico de potência durante apenas um ciclo, mas consome potência dentro da capacidade térmica do chip para todos os outros ciclos. Neste caso, o circuito não será danificado, pois a energia consumida – que corresponde ao tempo de potência consumida em um ciclo – não será suficiente para elevar a temperatura acima do limite de capacidade térmica do chip (a menos que o pico de potência consumida seja muito maior que o normal). Para que um *chip* seja danificado é necessário que haja um alto pico de potência não por um, mas por vários ciclos.

Anexo 2

Softwares utilizados e equipamentos de medição

- ModelSim: simulador lógico que permite analisar circuitos descritos em linguagem VHDL. Este *software* é parte integrante do ambiente Mentor Graphics e serviu de base para as etapas de projeto posteriores do fluxo apresentado, como mostra a FIGURA 10;
- XST: é a ferramenta de síntese utilizada pelo ISE. Foi inicialmente utilizada na etapa de testes em hardware das arquiteturas em FPGA e gera como arquivo de saída o formato *.bit* como pode ser observado na FIGURA 10;
- iMPACT: esta ferramenta é responsável por carregar a FPGA com o arquivo *.bit* da etapa anterior, fazendo a comunicação entre o computador e o *hardware* de destino. Também faz parte do ISE;
- Leonardo Spectrum: é uma ferramenta de síntese de linguagem de alto nível como VHDL, por exemplo, para FPGA (*Field Programmable Gate Array*) ou ASIC [6]. Esta ferramenta foi utilizada no processo de concepção dos circuitos integrados no ambiente Mentor Graphics, sendo responsável por juntar os vários códigos fonte de um projeto VHDL, tanto comportamental quanto estrutural, em um único arquivo de um formato intermediário (EDIF).

Neste arquivo de formato intermediário o projeto é sintetizado em células pré-existentes de uma dada tecnologia;

- SSTVelocity: é uma ferramenta de análise estática de atraso projetada para a verificação completa de Circuitos Integrados de Aplicação Específica (ASIC – *Application Specific Integrated Circuits*) [7]. Esta ferramenta analisa circuitos síncronos alimentados por sinais de *clock* síncronos e assíncronos. Dentre os arquivos de entrada suportados pela ferramenta, estão os formatos VHDL, VERILOG e EDIF. A ferramenta é capaz de analisar caminhos críticos e verifica os limites de operação de frequência do circuito pela imposição de atrasos de propagação ao longo destes caminhos. A soma destes tempos de atraso é então confrontada com as especificações de atraso inseridas no programa de maneira que possíveis violações são então apresentadas em um relatório;
- IC Station: esta é uma das ferramentas mais importantes no desenvolvimento do fluxo de projeto. É responsável pela visualização e edição de *layouts*. Nesta ferramenta ocorre a etapa inicial de geração das máscaras do circuito integrado, começando por apresentar graficamente os limites nos quais todo o circuito integrado deverá estar inserido (*autofloorplan*), seguido do processo de posicionamento das células (*autoplace*) e ligação das conexões (*autoroute*);

- *Design Architect*: é um completo editor de esquemáticos projetado para interagir com as demais ferramentas do Ambiente Mentor Graphics [27]. Este programa foi utilizado na etapa de geração do *padframe* no sentido de estabelecer simbolicamente as ligações entre o núcleo dos circuitos e os PADS;
- ELDO: simulador elétrico tipo SPICE. Este simulador é capaz de manipular circuitos até trezentos mil transistores e ter um aumento na velocidade de simulação entre 3 e 10 vezes perante os simuladores SPICE comerciais, mantendo a mesma precisão e confiabilidade [24]. O simulador ELDO foi inicialmente empregado neste trabalho. Foi utilizado para extrair dos circuitos informações sobre a dissipação de potência, conforme pode ser observado na FIGURA 10;
- Mach PA: este simulador lê *netlists* SPICE e analisa o consumo de potência dos circuitos sob teste [40]. São aceitos a maioria dos comandos utilizados no simulador ELDO. O grande diferencial deste simulador é a velocidade de análise dos circuitos. O Mach PA retorna resultados de análise de potência com a precisão de um simulador SPICE, a taxas de até 1000X mais rápido que tal simulador [2]. A FIGURA 26 apresenta a avaliação de uma SRAM com 16000 dispositivos e mostra que a simulação com o Mach PA é 400X mais veloz que a SPICE [2]. A velocidade e a capacidade deste simulador

permitem que o Mach PA possa ser utilizado para teste de grandes blocos ou chips completos, realizando com precisão simulações de dissipação de potência (FIGURA 27);

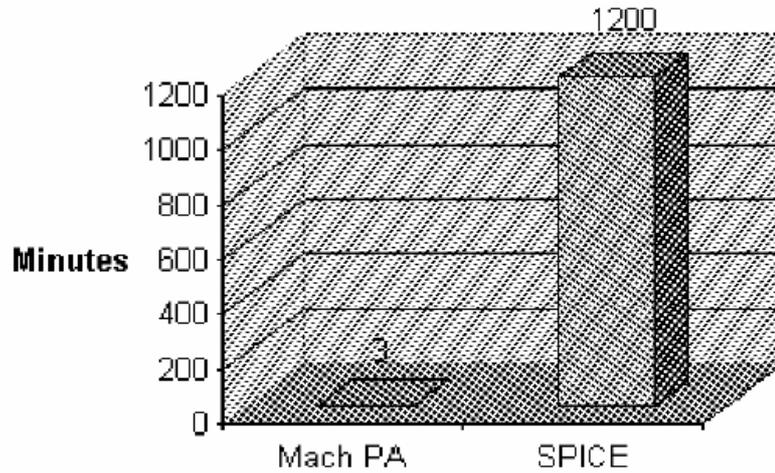


FIGURA 26 – Comparação de velocidade

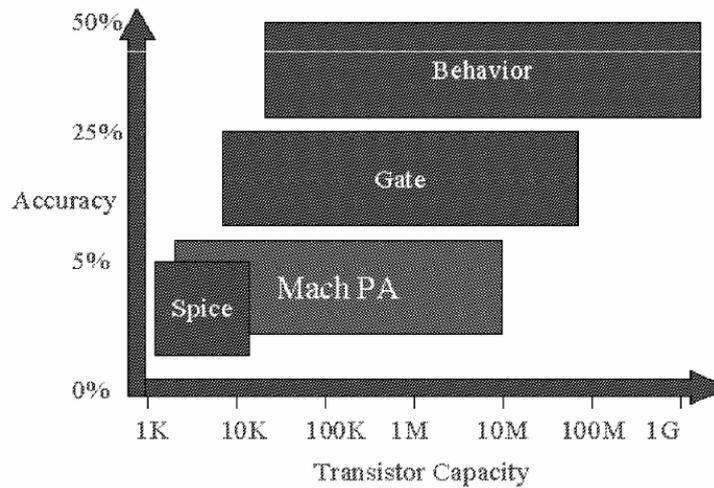


FIGURA 27 – Comparação de precisão e capacidade

- Placa Digilab IIE: esta placa de prototipação contém uma FPGA XILINX SPARTAN 2E XC2S200E [31]. Este dispositivo foi utilizado para os testes iniciais em *hardware* das arquiteturas de multiplicadores, bem como a posterior verificação de dissipação de potência através da inserção de vetores de teste nos *chips* prototipados;
- Fonte de tensão regulável EMG 18131: utilizada para alimentar os circuitos integrados;
- Osciloscópio Agilent 54622D: equipamento utilizado para verificação funcional em FPGA e protótipos e ajuste dos vetores de entrada;
- Multímetro Digital Agilent 34401A 6 ½: utilizado para extração dos valores de corrente DC dos *chips*;
- Microscópio Stemi 2000-C ZEISS: equipamento utilizado para conferência do *bounding diagram* e captação do *layout* dos protótipos.

Anexo 3

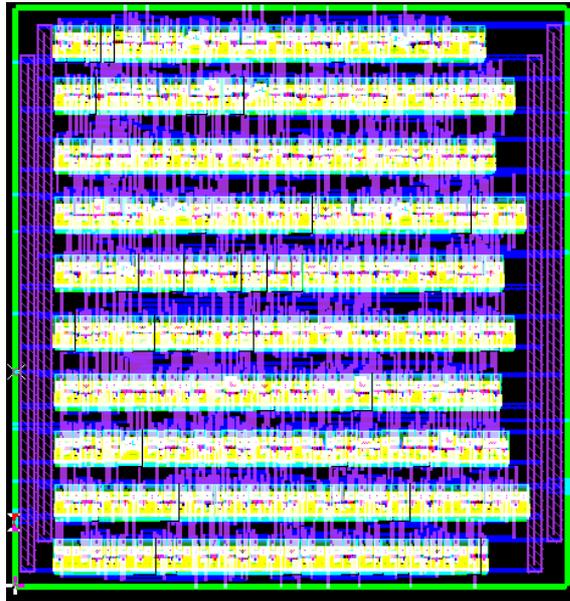


FIGURA 28 – Layout do circuito multiplicador array Binário de 8 bits para a segunda síntese

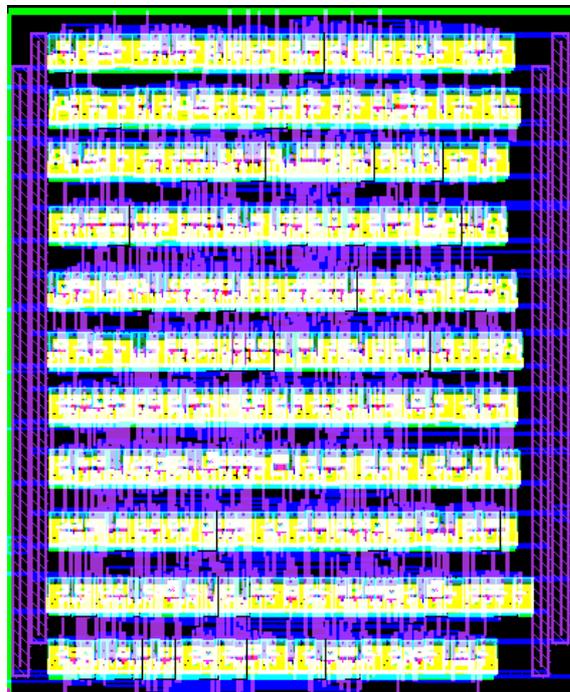


FIGURA 29 – Layout do circuito multiplicador array Híbrido de 8 bits para a segunda síntese

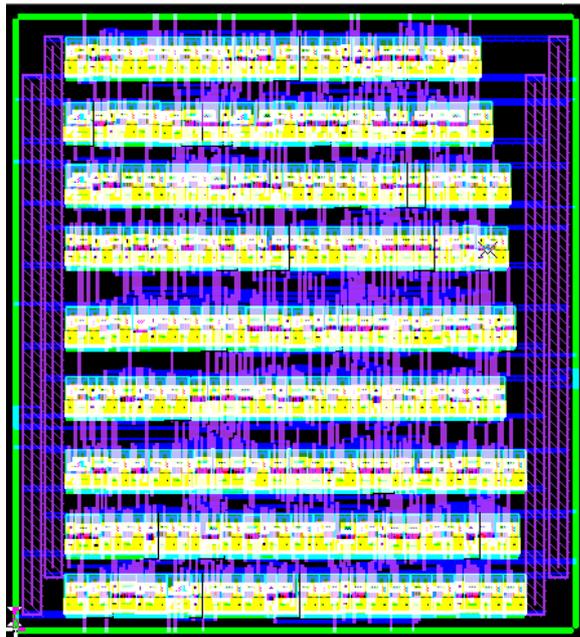


FIGURA 30 – Layout do circuito multiplicador Booth Modificado de 8 bits para a segunda síntese