

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE ENGENHARIA  
PROGRAMA DE POS-GRADUAÇÃO DE ENGENHARIA ELÉTRICA**

**LUÍS FERNANDO STIBORSKI CRISTOFOLI**

**ANÁLISE DA ROBUSTEZ DOS CIRCUITOS ASSÍNCRONOS EM  
AMBIENTE DE INTERFERÊNCIA ELETROMAGNÉTICA**

Porto Alegre  
2009

LUÍS FERNANDO STIBORSKI CRISTOFOLI

**ANÁLISE DA ROBUSTEZ DOS CIRCUITOS ASSÍNCRONOS EM  
AMBIENTE DE INTERFERÊNCIA ELETROMAGNÉTICA**

Dissertação apresentada como requisito para  
obtenção do grau de Mestre pelo Programa  
de Pós-graduação da Faculdade de Engenharia  
Elétrica da Pontifícia Universidade Católica  
do Rio Grande do Sul.

Orientador: Prof. Dr. Fabian Luiz Vargas

Coorientador: Prof. Ney Laert Vilar Calazans

Porto Alegre

2009

**ANÁLISE DA ROBUSTEZ DOS CIRCUITOS ASSÍNCRONOS EM  
AMBIENTE DE INTERFERÊNCIA ELETROMAGNÉTICA**

LUÍS FERNANDO STIBORSKI CRISTOFOLI

Dissertação apresentada ao Programa de Mestrado em Engenharia Elétrica, da Faculdade de Engenharia da Pontifícia Universidade Católica do Rio Grande do Sul, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica.

Fabian Luis Vargas, Dr.  
Orientador

Rubem Dutra Fagundes, Dr.  
Coordenador  
Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Dr. Fabian Luis Vargas  
Presidente – PUCRS

Prof. Dr. Ruben Dutra Fagundes  
PUCRS

Prof. Dr. Carlos Silva Cardenas  
PUCP

*Dedico esta dissertação a meus pais,  
cujo exemplo de honestidade e trabalho  
tem sido um norteador para a minha vida,  
para minha esposa, que tem  
me dado apoio nos momentos mais difíceis  
e mostrado a simplicidade de ter esperança  
e para meus filhos que me motivaram.*

# *Agradecimentos*

*Dedico meus sinceros agradecimentos para:*

- ✓ *aos professores doutores Fabian Vargas e Ney Calazans, pela orientação e incentivo;*
- ✓ *a equipe do Laboratório de Sistemas, Sinais e Computação - SiSC, pela ajuda em diversos momentos, contribuições e revisão deste trabalho;*
- ✓ *ao Laboratório de Ensino e Pesquisa - LEP, pelos empréstimos de equipamentos e placas;*
- ✓ *ao Sr. Christos P. Sotiriou, coordenador do projeto ASPIDA, pelo suporte técnico imprescindível;*
- ✓ *o coordenador do Programa de Pós-Graduação em Engenharia Elétrica, professor doutor Ruben Dutra Fagundes, pelo apoio manifestado;*
- ✓ *aos colegas do Mestrado em Engenharia Elétrica da PUCRS;*
- ✓ *e aos que de um forma ou de outra contribuíram neste trabalho.*

◀◀◀◀◀ *Muito Obrigado* ▶▶▶▶▶

*“Obstáculos e dificuldades fazem parte da vida.*

*E a vida é a arte de superá-los.*

*Os pensamentos são como pedras:*

*constróem, soterram e matam.”*

***Mestre DeRose***

# *Resumo*

Atualmente, grande parte dos equipamentos eletrônicos utilizam circuitos síncronos que são controlados por um sinal de relógio (*clock*) global. Este sinal estabelece o exato momento em que os registradores devem capturar os dados e assim, sincroniza as operações do sistema. Contudo, este tipo de circuito pode apresentar uma série de problemas como, por exemplo, grande sensibilidade ao ruído, além de apresentar altos índices de emissão eletromagnética e por conseguinte, afetar outros circuitos vizinhos com este tipo de ruído.

Neste contexto, os circuitos assíncronos surgem como uma alternativa extremamente viável e interessante no que diz respeito ao projeto de sistemas intrinsecamente mais robustos ao ruído. Entretanto, o uso de circuitos assíncronos em larga escala é nitidamente limitado pela maior complexidade de projeto e principalmente pela inexistência de ferramentas CAD capazes de darem suporte a todas as fases de desenvolvimento dos mesmos e a necessidade de mudança de paradigmas por parte dos projetistas.

Assim, o presente trabalho tem como principal objetivo comparar sistemas síncronos com assíncronos gerados a partir de uma dada técnica de dessincronização de forma a estabelecer a robustez associada a cada um dos circuitos. Esta técnica de dessincronização, desenvolvida em 2004 representa uma grande referência na área de projeto de circuitos assíncronos. Ela é baseada no fluxo de projeto de circuitos síncronos e representa uma solução bastante simples, capaz de gerar circuitos assíncronos a partir de descrições síncronas. Além disso, esta técnica pode ser implementada através do uso de ferramentas de CAD convencionais já existentes no mercado.

Finalmente, para validar a técnica de dessincronização acima mencionada, foram realizados vários experimentos de injeção de falhas através do uso de interferência eletromagnética (EMI) irradiada e conduzida de acordo com as normas IEC 62.132-2 e IEC 61.004-29. A plataforma de ensaios utilizada foi projetada e desenvolvida pela equipe do Laboratório SiSC (Sistemas, Sinais e Computação) da PUCRS. Sobre esta plataforma, uma placa SMD com seis camadas contendo vários FPGAs e lógica de controle, duas versões distintas do processador (*softcore*) DLX foram mapeadas em FPGA e o programa aplicativo carregado em memória BRAM. A análise dos resultados obtidos durante os experimentos de injeção de falhas indica que a técnica proposta é capaz de gerar eficientemente circuitos assíncronos e que estes, quando expostos a EMI, são sem dúvida mais robustos do que os circuitos síncronos.

# *Abstract*

Nowadays, the major part of electronic devices uses synchronous circuits controlled by a global clock signal. This signal establishes the exact moment the data should be captured by the registers synchronizing the system's operations. Meanwhile, this type of circuits can cause a series of problems as for example, very high noise sensibility and electromagnetic emission degree that can affect peripheral circuits.

In this context, asynchronous circuits have been proposed as an interesting alternative able to provide circuits intrinsically more robust to noise. Though, asynchronous circuits are not used in large scale, due to their high design's complexity and furthermore to the lack of CAD tools able to deal with all design's steps and involves changing most of the designers' mentality.

The proposed work aims at comparing the reliability of synchronous and asynchronous systems generated according to the desynchronization approach proposed in 2004. This technique represents a simple solution able to develop asynchronous circuits based on the synchronous circuit's design flow. It is important to note that the proposed approach can be implemented using commercial CAD tools.

Finally, in order to evaluate the technique proposed in this work, various fault injection campaigns have been performed applying irradiated and conducted electromagnetic interference (EMI) according to IEC 62.132-2 and IEC 61.004-29 norms, respectively. Indeed, a test platform has been developed by the SiSC Laboratory of the Catholic University of Rio Grande do Sul. This platform is composed of a 6-layers SMD board containing several FPGAs and control logic, two different versions of the DLX microprocessor mapped on a FPGA and an application software loaded into the BRAM memory. The analysis of the obtained results demonstrates that the proposed technique is able to efficiently generate asynchronous circuits that are certainly more robust and reliable when exposed to EMI than the synchronous ones.



# *Sumário*

**Lista de Figuras**

**Lista de Tabelas**

**Lista de Siglas**

<b>1</b>	<b>Introdução</b>	<b>21</b>
1.1	Motivação . . . . .	21
1.2	Objetivos . . . . .	22
1.3	Apresentação dos capítulos . . . . .	23
<b>2</b>	<b>Conceitos de tolerância a falhas</b>	<b>25</b>
2.1	Introdução . . . . .	25
2.2	Falha, erro ou defeito . . . . .	26
2.2.1	Latência . . . . .	27
2.2.2	Classificação das falhas . . . . .	28
2.3	Defeitos e modelos de falhas . . . . .	29
2.4	Medidas relacionadas ao tempo médio de funcionamento . . . . .	32
<b>3</b>	<b>Compatibilidade eletromagnética</b>	<b>35</b>
3.1	Introdução . . . . .	35
3.2	Conceitos Básicos . . . . .	36
3.3	Ambiente eletromagnético . . . . .	37
3.4	Interferência eletromagnética . . . . .	37

3.4.1	Fontes de interferência eletromagnética . . . . .	38
3.4.2	Interferência eletromagnética conduzida versus irradiada . . . . .	38
3.4.3	Acoplamento de interferência eletromagnética . . . . .	39
3.5	Efeitos da interferência eletromagnética na eletrônica . . . . .	40
3.6	Normas IEC . . . . .	41
3.6.1	Norma IEC 61.000 . . . . .	42
3.6.1.1	Norma IEC 61.000-4-17 . . . . .	42
3.6.1.2	Norma IEC 61.000-4-29 . . . . .	43
3.6.2	Norma IEC 62.132 . . . . .	45
3.6.2.1	IEC 62.132 - Parte 1 . . . . .	46
3.6.2.2	IEC 62.132 - Parte 2 . . . . .	51
<b>4</b>	<b>System on Chip (SoC)</b>	<b>53</b>
4.1	Introdução . . . . .	53
4.2	Linguagens para descrição de SoCs . . . . .	54
4.3	Arquitetura genérica de um SoC . . . . .	55
4.4	Interconexão de núcleos baseada em fios ponto-a-ponto dedicados . . . . .	57
4.5	Interconexão de núcleos baseada em barramentos compartilhados . . . . .	58
4.6	Interconexão de núcleos baseada em redes intra-chip (NoCs) . . . . .	60
<b>5</b>	<b>Tecnologia dos dispositivos reconfiguráveis</b>	<b>62</b>
5.1	Introdução . . . . .	62
5.2	Arquitetura dos FPGAs . . . . .	63
<b>6</b>	<b>Circuitos assíncronos</b>	<b>67</b>
6.1	Introdução . . . . .	67
6.2	Definições e conceitos básicos . . . . .	68
6.2.1	Princípio de funcionamento . . . . .	68

6.2.2	Característica de um operador assíncrono . . . . .	70
6.3	Implementação de circuitos assíncronos . . . . .	71
6.3.1	Problemas temporais . . . . .	71
6.3.2	Gestão dos problemas temporais . . . . .	72
6.4	Classificação . . . . .	72
6.4.1	Modelagem de atrasos . . . . .	73
6.4.2	Circuitos insensíveis ao atraso ( <i>Delay Insensitive - DI</i> ) . . . . .	73
6.4.3	Circuitos quase insensíveis ao atraso ( <i>Quasi-Delay Insensitive - QDI</i> ) . . . . .	74
6.4.4	Circuitos independente da velocidade ( <i>Speed Independent - SI</i> ) . . . . .	74
6.4.5	Circuitos <i>Micropipelines</i> . . . . .	75
6.4.6	Circuitos de Huffman . . . . .	75
6.5	Protocolos de comunicação . . . . .	75
6.5.1	Protocolo 2 fases . . . . .	76
6.5.2	Protocolo 4 fases . . . . .	76
6.6	Representação da informação . . . . .	77
6.6.1	Codificação de trilha única ( <i>Single Rail</i> ) . . . . .	77
6.6.2	Codificação de trilha dupla ( <i>Dual Rail</i> ) . . . . .	78
6.7	Elemento C-Müller . . . . .	79
6.8	Técnica de Dessincronização . . . . .	80
6.8.1	Introdução . . . . .	81
6.8.2	Etapas da Dessincronização . . . . .	82
6.8.3	Controlador dos <i>Latches</i> . . . . .	82
6.8.4	Substituição dos <i>Flip-Flops</i> . . . . .	83
6.8.5	Remoção do sinal de relógio ( <i>clock</i> ) . . . . .	85
6.8.5.1	Grafo da dependência dos dados . . . . .	85
6.8.5.2	Rede de controle . . . . .	85

6.8.5.3	Atraso da lógica combinacional . . . . .	86
6.8.5.4	Interligação da rede . . . . .	88
6.8.6	Resultado da temporização . . . . .	89
<b>7</b>	<b>Plataforma de ensaios</b>	<b>90</b>
7.1	Introdução . . . . .	90
7.2	Placa de ensaios . . . . .	90
7.3	Placa de alimentação e injeção de falhas . . . . .	93
7.4	Circuito gerenciador da alimentação do ensaio . . . . .	95
7.5	Conclusão . . . . .	96
<b>8</b>	<b>Estudo de Caso I</b>	<b>98</b>
8.1	Introdução . . . . .	98
8.2	Multiplicador com somadores . . . . .	98
8.3	Modificações estruturais no multiplicador . . . . .	99
8.4	Sistema de gerenciamento de ensaios . . . . .	100
8.5	Ensaio realizados . . . . .	101
8.6	Conclusão . . . . .	102
<b>9</b>	<b>Estudo de Caso II</b>	<b>104</b>
9.1	Introdução . . . . .	104
9.2	Projeto ASPIDA . . . . .	104
9.3	Processador DLX . . . . .	105
9.4	Processador DLX ASPIDA . . . . .	106
9.5	Processador DLX assíncrono . . . . .	108
9.6	Implementação DLX assíncrono na FPGA . . . . .	108
9.7	Sistema de gerenciamento de ensaios . . . . .	109
9.8	Softwares de ensaio . . . . .	112

9.9	Ensaio realizados . . . . .	113
9.9.1	Ensaio de redução dos níveis de tensão de alimentação . . . . .	113
9.9.2	Ensaio de queda dos níveis de tensão de alimentação . . . . .	116
9.9.3	Ensaio de interferência eletromagnética irradiada . . . . .	118
9.9.4	Ensaio de interferência eletromagnética emitida . . . . .	121
9.10	Conclusão . . . . .	125
<b>10</b>	<b>Conclusões finais</b>	<b>128</b>
10.1	Perspectiva de trabalhos futuros . . . . .	129
	<b>Referências</b>	<b>131</b>
	<b>Anexo A - Esquemático da plataforma de ensaio</b>	<b>140</b>
	<b>Anexo B - Esquemático da placa de alimentação e injeção de falhas</b>	<b>150</b>

# *Lista de Figuras*

1	Modelo dos três universos . . . . .	27
2	Conceito de latência de falha e latência de erro. . . . .	28
3	Notação e emulação do modelo de falha <i>Stuck-at</i> . . . . .	30
4	Modelo de falha <i>Transistor-Level Stuck</i> . . . . .	30
5	Modelo de falha <i>bridging</i> . . . . .	31
6	Curva da banheira . . . . .	34
7	Exemplo de interferência eletromagnética conduzida . . . . .	39
8	Exemplo de interferência eletromagnética irradiada . . . . .	39
9	Exemplos de acoplamentos de EMI conduzida e irradiada . . . . .	40
10	Exemplo de Integridade Sinais . . . . .	40
11	Retificador de monofásico . . . . .	43
12	Gerador de ensaio com fontes de alimentação de chaveamento interno . . . . .	46
13	Gerador de ensaio baseado em uma fonte de alimentação programável . . . . .	46
14	Características de placa de circuito impresso de ensaio . . . . .	47
15	Disposição das camadas de uma placa de ensaio . . . . .	47
16	Exemplos sinais de onda contínua e amplitude modulada . . . . .	50
17	Configuração de ensaio com células TEM e GTEM . . . . .	52
18	Diminuição do ciclo de vida dos produtos eletrônicos . . . . .	54
19	Arquitetura genérica de um SoC . . . . .	56
20	Estrutura genérica de interconexão baseada em fios ponto-a-ponto . . . . .	58
21	Estrutura de interconexão em barramento com arbitro central . . . . .	59
22	Interconexão de núcleos baseada em redes intra-chip (NoCs) . . . . .	60

23	Exemplo de uma rede intra-chip (NoCs) . . . . .	61
24	Arquitetura básica de um FPGA . . . . .	64
25	Circuito síncrono genérico . . . . .	69
26	Circuito assíncrono genérico . . . . .	70
27	Comunicação no circuito assíncrono . . . . .	70
28	Classificação dos circuitos assíncronos . . . . .	73
29	Modelos de atraso dos circuitos assíncronos . . . . .	73
30	Bifurcações isocrônicas ( <i>isochronics forks</i> ) . . . . .	74
31	Circuitos Micropipelines . . . . .	75
32	Protocolo 2 fases . . . . .	76
33	Protocolo 4 fases . . . . .	77
34	Codificação de trilha única ( <i>Single Rail</i> ) . . . . .	77
35	Codificação de trilha dupla - 4 fases ( <i>Dual Rail</i> ) . . . . .	78
36	Codificação de trilha dupla por valor - 2 fases ( <i>Dual Rail</i> ) . . . . .	79
37	Codificação de trilha dupla por paridade - 2 fases ( <i>Dual Rail</i> ) . . . . .	79
38	Elemento C-Müller . . . . .	80
39	Fluxo comparativo da dessincronização . . . . .	81
40	Circuito síncrono inicial . . . . .	82
41	Bloco diagrama do controlador do <i>latch</i> . . . . .	83
42	Grafo dos protocolos de <i>handshake</i> para dessincronização . . . . .	84
43	Exemplo de flip-flop onde escorregamento não é permitido . . . . .	84
44	Grafo da dependência dos dados do circuito exemplo . . . . .	85
45	Rede de controle do circuito exemplo . . . . .	86
46	Modelo de lógica com elementos de atraso . . . . .	87
47	Estrutura de um elemento de atraso assimétrico . . . . .	87
48	Modelo de lógica com detecção de conclusão . . . . .	88

49	Circuito dessincronizado . . . . .	88
50	Composição da placa de circuito impresso . . . . .	91
51	Esquemático genérico da plataforma de ensaios . . . . .	92
52	Vista inferior da plataforma de ensaios . . . . .	93
53	Vista superior da plataforma de ensaios . . . . .	94
54	Diagrama da placa de alimentação e injeção de falhas . . . . .	94
55	Placa de alimentação e injeção de falhas . . . . .	95
56	Diagrama temporal de escrita nos conversores digitais-analógicos . . . . .	95
57	Forma de onda da queda de tensão pulsada . . . . .	96
58	Circuito do multiplicador . . . . .	99
59	Circuito do multiplicador modificado . . . . .	100
60	Circuito de ensaio do multiplicador . . . . .	101
61	Processador DLX . . . . .	106
62	Processador DLX: leiaute das instruções . . . . .	107
63	Processador DLX: instruções - subconjunto 1 . . . . .	107
64	Processador DLX: instruções - subconjunto 2 . . . . .	108
65	Processador DLX: instruções - subconjunto 3 . . . . .	108
66	Processador DLX assíncrono . . . . .	109
67	Processador ASPIDA DLX: sistema de demonstração . . . . .	110
68	Processador ASPIDA DLX: sistema de ensaio . . . . .	111
69	Configurações dos ensaios de redução e queda dos níveis de tensão de ali- mentação . . . . .	114
70	Sequência de procedimentos dos ensaios de redução e queda de tensão de alimentação . . . . .	115
71	Gráfico comparativo das tensões de falha (Redução de Tensão) . . . . .	116
72	Gráfico comparativo das tensões de falha (Queda de Tensão) . . . . .	117
73	Configuração dos ensaios de interferência eletromagnética irradiada . . . . .	118



74	Posições da placa durante os ensaios . . . . .	119
75	Gráfico de ocorrência de falhas nos ensaios de interferência eletromagnética irradiada (DLX síncrono) . . . . .	120
76	Gráfico de ocorrência de falhas nos ensaios de interferência eletromagnética irradiada (DLX assíncrono) . . . . .	120
77	Gráfico comparativo dos ensaios de interferência eletromagnética irradiada	121
78	Configuração dos ensaios de interferência eletromagnética emitida . . . . .	122
79	Gráfico comparativo dos ensaios de interferência eletromagnética emitida .	122
80	DLX Síncrono: Interferência eletromagnética emitida . . . . .	124
81	DLX Assíncrono: Interferência eletromagnética emitida . . . . .	125

# *Lista de Tabelas*

1	Causas usuais em sistemas computacionais . . . . .	29
2	Medidas de dependabilidade . . . . .	33
3	Níveis de Tensão de Ensaio . . . . .	43
4	Níveis de tensão e duração recomendadas para interrupções . . . . .	44
5	Níveis de tensão e duração recomendados para quedas de tensão . . . . .	45
6	Níveis de tensão e duração recomendados para variação de tensão . . . . .	45
7	Conteúdo das camadas da placa de circuito impresso . . . . .	48
8	Conexão dos pinos do circuito integrado . . . . .	49
9	Intervalos de frequência aplicados nos ensaios . . . . .	49
10	Descrição funcional dos pinos do circuito de Alimentação . . . . .	96
11	Descrição dos comandos de controle . . . . .	97
12	Resposta ao comandos de controle ‘0’ . . . . .	97
13	Multiplicador: área utilizada na FPGA ( <i>overhead</i> ) . . . . .	102
14	Multiplicador: ensaio de queda de tensão . . . . .	103
15	Processador ASPIDA DLX: mapa de memória do sistema de demonstração .	110
16	Processador ASPIDA DLX: mapa de memória do sistema de ensaio . . . .	111
17	Processador DLX: área <i>Post-Synthesis</i> . . . . .	112
18	Processador DLX: temporização na prototipação . . . . .	112

# *Lista de Siglas*

ALU – Arithmetic and Logic Unit

AM – Amplitude Modulation

ASIC – Application Specific Integrated Circuit

BCI – Bulk Current Injection

CI – Circuito Integrado

CLB – Configurable Logic Block

CMOS – Complementary Metal-Oxide-Semiconductor

CPLD – Complex Programmable Logic Device

CW – Continuous Wave

DAC – Digital to Analog Converter

DC – Direct Current

DSP – Digital Signal Processing

DUT – Device Under Test

EEPROM – Electrically Erasable Programmable Read Only Memory

EMC – Electromagnetic Compatibility

EME – Electromagnetic Environment

EMI – Electromagnetic Interference

EPROM – Erasable Programmable Read Only Memory

FPGA – Field Programmable Gate Array

FSM – Finite State Machine

GCC – GNU Compiler Collection

GTEM – Gigahertz Transverse Electromagnetic

IC – Integrated Circuits

IEC – International Electrotechnical Commission

IP – Intellectual Property

ISP – Instruction Set Processor

MEMS – Micro-Electro-Mechanical Systems

MIPS – Microprocessor without Interlocked Pipeline Stages

MM – Multiplicação de Matrizes

MOS – Metal Oxide Semiconductor

MOSFET – Metal-Oxide Semiconductor Field Effect Transistor

MP3 – Motion Picture Experts Group Layer-3

MPEG2 – Motion Picture Experts Group Layer-2

MTBF – Mean Time Between Failure

MTTF – Mean Time to Failure

MTTR – Mean Time to Repair

NMOS – Negative Metal Oxide Semiconductor

OPB – On-Chip Peripheral Bus

PAL – Programmed Array Logic

PCB – Printed Circuit Board

PLA – Programmable Logic Arrays

PLL – Phase Locked Loop

PMOS – Positive Metal Oxide Semiconductor

PROM – Programmable Read Only Memory

PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul

RAM – Random Access Memory

RF – Radiofrequência

RISC – Reduced Instruction Set Computer

RTOS – Real-Time Operating System

SI – Signal Integrity

SiSC – Laboratório de Sistemas, Sinais e Computação

SoC – Systems-on-Chip

SRAM – Static Random Access Memory

TEM – Transverse Electromagnetic

UART – Universal Asynchronous Receiver Transmitter

USB – Universal Serial Bus

VHDL – VHSIC Hardware Description Language

VHSIC – Very High Speed Integrated Circuits

VLSI – Very Large Scale Integration

# 1 *Introdução*

Equipamentos com sistemas embarcados de tempo real são cada vez mais utilizados atualmente, estão por todos os lugares, e cada vez é exigido mais destes sistemas que tornem-se mais inteligentes e complexos, tendo como consequência o aumento do nível de integração [1], [2]. Com isto, temos o crescimento das pesquisas de técnicas para proporcionar a estes sistema uma melhora da performance com aumento da confiabilidade e robustez, baseando-se em tecnologias do tipo *Systems-on-Chip* (SoCs).

Por outro lado, temos o aumento da demanda de equipamentos alimentados por baterias, com baixo consumo, caracterizando-se como um fator agravante, onde oscilações nos níveis de tensão do barramento de alimentação de circuitos digitais síncronos, de elevada densidade de integração e de alto desempenho, provocam perturbações as quais podem causar a degradação do sistema. Este fato ocorre devido atrasos (*delay*) na propagação dos sinais, o que compromete o desempenho e a confiabilidade, uma vez que podem levar o equipamento a situações de falha [3] [4] .

## 1.1 *Motivação*

Os circuitos assíncronos, nos últimos anos, estão sendo pesquisados como uma alternativa interessante aos projetos de sistemas robustos a falhas, que apresentam baixo consumo e podem operar com tensões baixas, apresentando uma boa imunidade a interferência eletromagnética, quando comparado aos circuitos síncronos [5] [6] [7]. Porém, este tipo de circuitos não são utilizados em larga escala, devido ao fato de que não existem boas ferramentas de CAD que tratem por completo todas as etapas do projeto, e principalmente porque estes circuitos exigem que os projetistas mudem suas mentalidades [8].

Recentemente, em 2004, foi apresentada uma técnica intermediária e simples para obtenção de circuitos assíncronos, chamada de Dessincronização, que deriva pouco do fluxo

de projetos síncronos, utilizando ferramentas de CAD convencionais para transformar um circuito síncrono em assíncrono, desta forma permitindo a difusão dos circuitos assíncronos [8] [9].

Baseando-se neste cenário, nesta dissertação estaremos verificando a robustez dos sistemas assíncronos versus síncronos, gerados pela técnica de dessincronização, com o objetivo do aumento da robustez de *Systems-on-Chip* (SoCs) sujeitos a flutuações no barramento de alimentação e interferências eletromagnéticas (EMI) irradiadas, sem a degradação do desempenho destes sistemas.

## 1.2 Objetivos

Esta dissertação tem como objetivo principal a implementação e a comparação dos circuitos síncronos e assíncronos com relação ao aumento da robustez de *Systems-on-Chip* sujeitos a flutuações no barramento de alimentação. Além disto, este trabalho também possui os seguintes objetivos:

- Analisar o comportamento de circuitos digitais síncronos e assíncronos implementados em FPGAs do tipo SRAM quando expostos a reduções e quedas dos níveis de tensão de alimentação;
- Analisar o comportamento de circuitos digitais síncronos e assíncronos implementados em FPGAs do tipo SRAM quando expostos a interferências eletromagnéticas (EMI) irradiadas;
- Realizar ensaios de interferências eletromagnéticas (EMI) e redução de tensão seguindo as normas *Internacional Electrotechnical Commission* (IEC) [10] [11] [12] para analisar a robustez de *Systems-on-Chip*.

O tema desta dissertação é de grande importância para a pesquisa na área de integridade de sinais, além de projetos e ensaios de *Systems-on-Chip* robustos, pois apresenta a análise de desempenho de circuitos síncronos e assíncronos, mesmo quando estes estão sujeitos a flutuações no barramento de alimentação e/ou interferências eletromagnéticas irradiadas.

## 1.3 Apresentação dos capítulos

Esta dissertação foi dividida em três partes dispostas da seguinte forma:

### Parte I - Fundamentos teóricos

- Capítulo 2: apresenta os conceitos clássicos da área de tolerância a falhas;
- Capítulo 3: apresenta os principais conceitos sobre compatibilidade eletromagnética (EMC), juntamente com a apresentação das principais normas técnicas relacionadas às interferências eletromagnéticas (EMI) em circuitos integrados;
- Capítulo 4: apresenta uma breve introdução relacionada às principais tecnologias que envolvem os componentes reconfiguráveis (FPGAs);
- Capítulo 5: apresenta uma breve introdução relacionada aos principais conceitos que envolvem os *Systems-on-Chip* (SoCs);
- Capítulo 6: apresenta a uma introdução relacionada aos principais conceitos aplicados aos circuitos assíncronos e a técnica denominada dessincronização.

### Parte II - Metodologia

- Capítulo 7: apresenta a plataforma de ensaios utilizada nos ensaios realizados nesta dissertação, desenvolvida no laboratório de Sistemas, Sinais & Computação (SiSC).
- Capítulo 8: apresenta a metodologia adotada nesta dissertação para a análise da robustez de um circuito multiplicador implementado de forma síncrono e assíncrono, prototipado na plataforma de ensaios desenvolvida no laboratório de Sistemas, Sinais & Computação (SiSC).
- Capítulo 9: apresenta a metodologia adotada nesta dissertação para a análise da robustez de um processador DLX implementado de forma síncrono e assíncrono, prototipado na plataforma de ensaios desenvolvida no laboratório de Sistemas, Sinais & Computação (SiSC). Juntamente com os resultados dos ensaios de alteração de tensão de alimentação e compatibilidade eletromagnética, realizados nesta dissertação, para a análise da robustez de circuitos síncronos e assíncronos, objetivo desta dissertação.

### Parte III - Conclusões



- Capítulo 10: apresenta as conclusões obtidas no desenvolvimento desta dissertação e propõe alguns trabalhos futuros relacionados a este tema.

## 2 *Conceitos de tolerância a falhas*

Neste capítulo serão abordados os conceitos clássicos relacionados a área de tolerância a falhas, os quais relacionam-se com esta dissertação.

### 2.1 Introdução

Tolerância a falhas constitui-se em um conjunto de técnicas que têm por objetivo o prover um circuito e/ou sistema que permaneça corretamente funcional, mesmo perante a ocorrência de falhas, sem a degradação de sua performance, para isto a técnica de tolerância a falhas mais comum é a de redundância. Esta redundância, em um sistema computacional, pode ser: de *hardware* (ex: circuitos extras), *software* (ex. programas de diagnóstico) ou temporal (ex. duplicação de operações) [13], com a finalidade de evitar danos [14].

A seguir são apresentados os principais conceitos retirados da literatura [13] [14] [15] [16] [17]

**Falha:** pode ocorrer tanto no âmbito de *hardware* quanto de *software*, sendo esta a causa de um erro. Componentes envelhecidos e interferências externas são exemplos de fatores que podem levar o sistema à ocorrência de falhas. As falhas podem ser classificadas em permanentes, transientes e intermitentes:

- a) **Falhas permanentes:** ocorrem no meio físico. Falhas no processo de fabricação e/ou pelo envelhecimento dos componentes do sistema, que causam curtos circuitos, nós abertos e *stuck-at* são exemplos de falhas permanentes.
- b) **Falhas transientes:** são provocadas por adversidades e/ou fenômenos ambientais aleatórios onde o sistema está localizado e ocorrem durante a vida útil dos componentes. Variações na tensão de alimentação e interferências eletromagnéticas causadas por antenas, rádios PX, telefones celulares e descargas elétricas são

exemplos de falhas transientes.

- c) **Falhas intermitentes:** são ocorrências temporárias e/ou cíclicas, a partir de variações das condições externas e/ou ambientais do sistema. Estas falhas podem ser causadas por vibrações e variações da temperatura.

**Stuck-at:** falha permanente onde um nó do circuito permanece sempre num mesmo nível lógico: zero (*stuck-at-zero*) ou um (*stuck-at-one*).

**Bit flips:** falha transiente causada por interferências externas, que produzem a mudança temporária do nível lógico de um determinado nó do circuito. Pode ocorrer em ambos os sentidos, de zero para um ou de um para zero.

**Erro:** define-se que um sistema encontra-se em estado errôneo, ou em erro, se o processamento posterior a partir desse estado pode levar a um defeito.

**Defeito:** ocorre quando existe um desvio das especificações do projeto, esse não pode ser tolerado e deve ser evitado.

**Latência:** período de tempo compreendido entre a ocorrência da falha até a sua devida manifestação.

**Dependabilidade:** esse termo é uma tradução literal do termo inglês *dependability*, que indica a qualidade e a confiança depositada no serviço fornecido por um determinado sistema. Confiabilidade e disponibilidade são dois dos principais atributos da dependabilidade.

**Confiabilidade:** capacidade de atender as especificações do projeto dentro de condições definidas durante certo período de funcionamento e estar operacional no início desse período.

**Disponibilidade:** é a probabilidade de o sistema estar operacional quando a utilização deste for necessária.

## 2.2 Falha, erro ou defeito

De acordo com os conceitos para falha, erro e defeito, um sistema apresenta um defeito (*failure*) quando ocorre um desvio da especificação; o erro (*error*) é alguma alteração no sistema, que pode levá-lo a um estado diferente do especificado, assim o erro é o causador do defeito; e por fim, a falha (*fault*) é a causa primária de um defeito que venha a ocorrer no sistema.

A ocorrência de falhas são inevitáveis, devido ao fato de que os componentes físicos sofre do processo de envelhecimento e/ou estão expostos a interferências externas, sejam ambientais ou humanas. O mesmo podemos inferir com relação aos projetos de *software* e *hardware*, devido a sua sua alta complexidade e a incapacidade humana de trabalhar com grande volume de detalhes ou ainda especificações deficiente ou incompletas. No entanto, estes defeitos podem ser evitados com a utilização de técnicas de tolerância a falhas.

Na figura 1 é mostrada uma simplificação, sugerida por Dhiraj K. Pradhan [15], que será adotada nesta dissertação para os conceitos de falha, erro e defeito. Sendo falhas associadas ao universo físico, erros ao universo da informação e defeitos ao universo do usuário.

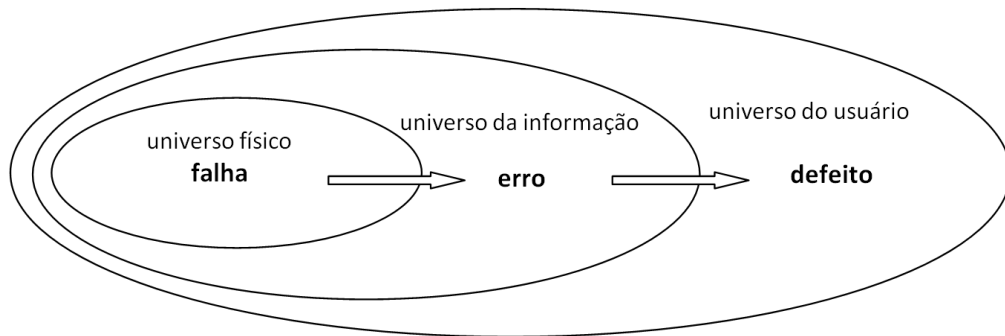


Figura 1: Modelo dos três universos [15].

Como exemplo para este modelo de três universos [15] podemos usar *chip* de memória, que apresenta uma falha do tipo *stuck-at-zero* em um de seus *bits* (falha no universo físico), que pode provocar uma interpretação errada da informação armazenada em uma de suas estrutura de dados (erro no universo da informação). Tendo como resultado deste erro, por exemplo, a não autorização do sistema para o embarque de todos os passageiros de um voo (defeito no universo do usuário). É interessante observar que uma falha não necessariamente conduz a um erro (pois a parte da memória sob falha pode nunca ser usada) e um erro não necessariamente leva a um defeito (no exemplo, a informação de voo lotado poderia eventualmente ser obtida a partir de outros dados redundantes da estrutura).

### 2.2.1 Latência

Define-se latência de falha como o período de tempo compreendido entre a ocorrência da falha e a manifestação do erro provocado por esta. E define-se latência de erro o

período de tempo compreendido entre a ocorrência do erro e a manifestação do defeito ocasionado por este.

Seguindo o modelo de três universos [15], o tempo total compreendido entre a ocorrência da falha e o aparecimento do defeito é a soma das latências de falha e de erro. A Figura 2 apresenta este conceito de latência de falha e latência de erro.

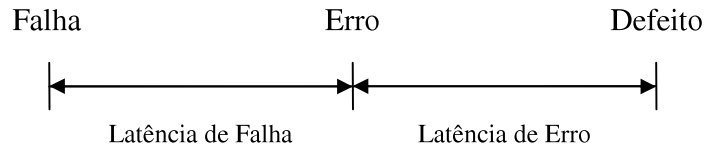


Figura 2: Conceito de latência de falha e latência de erro.

### 2.2.2 Classificação das falhas

Diversas classificações para falhas são apresentadas na literatura [13] [14] [15], entretanto aparecem classificadas geralmente como falhas físicas (aquelas de que padecem os componentes) e falhas humanas (que compreendem falhas de projeto e falhas de interação).

As principais causas de falhas são problemas de especificação, implementação, componentes defeituosos, imperfeições de manufatura, fadiga dos componentes físicos, além de distúrbios externos como radiação, interferência eletromagnética, variações ambientais (temperatura, pressão, umidade) e também problemas de operação.

Ainda para definir uma falha consideram-se os seguintes itens:

- a) **Natureza:** falha de *hardware*, falha de *software*, de projeto, de operação;
- b) **Duração ou persistência:** permanente ou temporária (intermitente ou transitória);
- c) **Extensão:** global ou local a um determinado módulo do sistema;
- d) **Valor:** determinado ou indeterminado no tempo.

As falhas provocadas por interação humana maliciosa vem crescendo, ou seja, por ações que propositalmente visam provocar danos aos sistemas. Essas falhas são tratadas

por técnicas de segurança computacional (*security*) e não por técnicas de tolerância a falhas. Entretanto um sistema tolerante a falhas deve considerá-las também para ser seguro a estas intrusões e ações maliciosas.

Falhas de *software* e projeto são atualmente consideradas o mais grave problema em computação crítica. Os sistemas críticos, normalmente são construídos de forma a suportar e tolerar falhas físicas. Assim é compreensível que falhas não tratadas e não previstas no projeto sejam as que mais danos causem aos sistemas, pois possuem um grande potencial de comprometer a sua confiabilidade e disponibilidade do sistema. Um exame de estatísticas disponíveis em [18] confirma essas considerações conforme mostrado na Tabela 1.

Tabela 1: Causas usuais em sistemas computacionais [18].

Sistemas Tradicionais				Redes cliente-servidor (não tolerantes a falhas)	
Não tolerantes a falhas		Tolerantes a falhas			
MTTF: 6 a 12 semanas		MTTF: 21 anos (tendem)		Disponibilidade média: 98%	
Indisponibilidade após defeito: 1 a 4 horas					
Defeitos:		Defeitos:		Defeitos:	
hardware	50%	software	65%	projeto	60%
software	25%	operações	10%	operações	24%
comunicação/ambiente	15%	hardware	8%	físicos	16%
operações	10%	ambiente	7%		

## 2.3 Defeitos e modelos de falhas

Com o intuito de realizar ensaios em circuitos ou sistemas eletrônicos com a finalidade de detectar prováveis falhas, modelos de falhas são utilizados para emulação destas falhas ou defeitos em um ambiente de simulação durante a etapa do projeto e devem atender aos seguintes requisitos [19] [20]:

1. Refletir com precisão o comportamento dos defeitos reais que podem acontecer durante a fabricação e no processo de manufatura, como também o comportamento de falhas que podem acontecer durante a operação do sistema.
2. Deve ter eficiência computacional com respeito ao ambiente de simulação de falha.

Serão descritas a seguir alguns modelos de falhas, que nos últimos anos surgiram baseados nos principais defeitos físicos encontrados nos circuitos.

- a) **Modelo de falha *gate-level stuck-at*** Este modelo de falha define que as portas de entrada e/ou saída podem estar presas ao nível lógico ‘0’ (*stuck-at-zero*) ou ao nível lógico ‘1’ (*stuck-at-one*). Note que as falhas *stuck-at* são emuladas como se as portas de entradas e/ou saídas estivessem desconectadas e ligadas ao nível lógico ‘0’ (*stuck-at-zero*) ou ao nível lógico ‘1’ (*stuck-at-one*) [18].

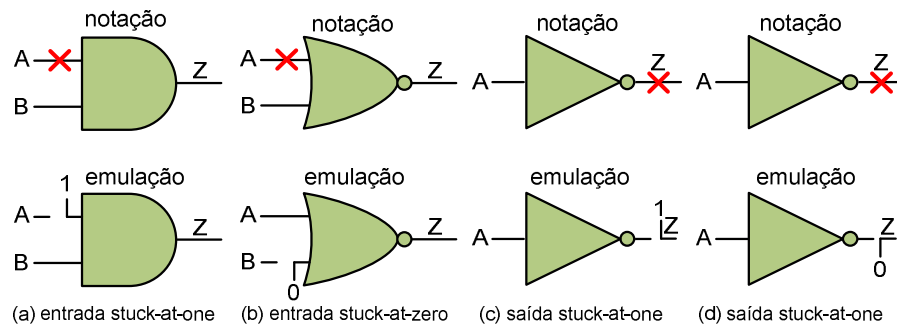


Figura 3: Notação e emulação do modelo de falha *Stuck-at* [20].

- b) **Modelo de Falha *Transistor-Level Stuck*** Este modelo reflete o comportamento exato das falhas de transistores em circuitos NMOS (*Negative Channel Metal-Oxide Semiconductor*) e define que qualquer transistor pode estar *stuck-on* (também denominado *stuck-short*) ou *stuck-off* (também denominado *stuck-open*) [18].

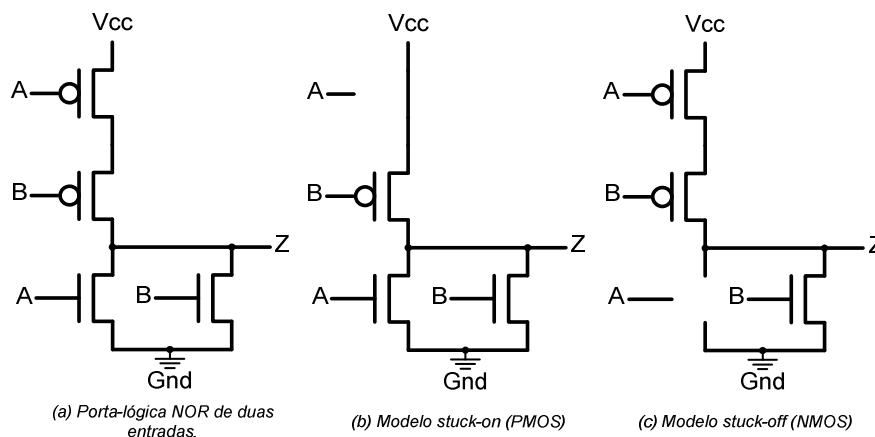


Figura 4: Modelo de falha *Transistor-Level Stuck* [20].

Salienta-se que as falhas *stuck-on* (*s-on*) podem ser emuladas através de um curto circuito entre o *source* e o *drain* do transistor e as falhas *stuck-off* (*s-off*) desconectando-se o transistor do circuito. Alternativamente, falhas *stuck-on* podem ser emuladas

desconectando a porta MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*) do sinal e conectando-a a lógica ‘1’ para transistores NFETS (*Negative Metal Oxide Semiconductor Field Effect Transistor*) ou a lógica ‘0’ em transistores PFETS (*Positive Metal Oxide Semiconductor Field Effect Transistor*).

Este procedimento fará com que o transistor esteja sempre conduzindo. Já no que diz respeito a falhas *stuck-off*, elas podem ser emuladas conectando a porta MOSFET a lógica ‘0’ para transistores NFET e a lógica ‘1’ para transistores PFET, assim o transistor nunca conduzirá.

- c) **Modelo de falha *bridging*** Este modelo inclui outro importante conjunto de falhas, tais como curtos entre trilhas e rompimento de trilhas (trilhas abertas). Estes tipos de falhas resultam basicamente de deposição excessiva (*overetching*) ou insuficiente (*under-etching*) durante o processo de fabricação do VLSI (*Very Large Scale Integration*) ou PCB (*printed circuit board*). Outro modelo de falha *bridging* ocorrem em ASICs (*Application Specific Integrated Circuit*) e FPGAs, definido como *dominant-AND/OR bridging*.

Observa-se que apesar das falhas *transistor-level* e *bridging* refletirem mais fielmente o comportamento das falhas presentes em circuitos, sua emulação e avaliação em simuladores é computacionalmente mais complexa em relação as tradicionais falhas *stuck-at* [18].

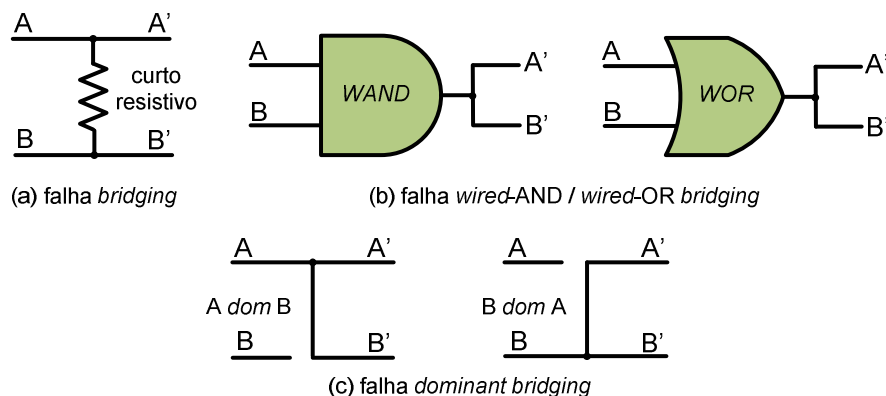


Figura 5: Modelo de falha *bridging* [20].

- d) **Modelo de falha *delay*** Este modelo representa outra importante classe de falhas que diferente do outros modelos de falha, os circuitos executam suas operações corretamente do ponto de vista lógico combinacional, entretanto, não são executadas na frequência de operação nominal especificada, ocasionando assim um erro de



*timing* proveniente dos diferentes tempos de propagação entre os sinais internos do circuito [21]. Este tipo de falha origina-se a partir de um *over* ou *under-etching* durante o processo de fabricação que origina MOSFETs com canais muito mais estreitos ou longos do que os pretendidos.

Assim, o ensaio de *delay* concentra-se em encontrar e expor todo e qualquer falha que possa existir no dispositivo. O objetivo básico deste tipo de ensaio é verificar o tempo de propagação dos sinais nos caminhos entre *flip-flops*, entre entradas primárias e *flip-flops* e finalmente entre *flip-flops* e saídas primárias, ou seja, verificar através da lógica combinacional se durante a operação na frequência requerida, algum caminho do sistema falhou.

Tipicamente, o ensaio de *delay* consiste na aplicação sequencial de dois vetores, tal que o caminho através da lógica combinacional é carregado com o primeiro vetor enquanto o segundo vetor gera a transição através dos caminhos para detecção da falha [18].

- d) **Modelo de falha simples versus múltiplos** Durante o processo de fabricação de um determinado dispositivo VLSI ou PCB múltiplos defeitos podem ser inseridos. Para ilustrar com mais clareza, as diferenças em termos de tempo de simulação dos modelos simples versus múltiplos, observe os exemplos abaixo descritos. Em um circuito com  $N$  portas de entradas e saídas, diante do modelo múltiplo de falha *stuck-at* deve-se emular  $3N-1$  e do modelo de falha simples apenas  $2N$  diferentes falhas. A mesma análise pode ser feita diante dos modelos *transistor-level stuck* e para o modelo *wired-AND/OR* ou *dominant bridging*. Já para o modelo *dominant-AND/OR bridging* múltiplo é necessário simular  $5N-1$  falhas e no simples  $4N$  falhas. Entretanto, a alta cobertura de falhas obtida a partir de modelos de falhas simples garante sua ampla utilização no desenvolvimento e avaliação de ensaios [18].

## 2.4 Medidas relacionadas ao tempo médio de funcionamento

As medidas para avaliação de dependabilidade mais usadas na prática são: taxa de defeitos, MTTF (*mean time to failure*), MTTR (*mean time to repair*), MTBF (*mean time between failure*). Estas medidas estão por sua vez relacionadas a outro parâmetro importante chamado confiabilidade [15]. A Tabela 2 apresenta uma definição informal dessas medidas.

Tabela 2: Medidas de dependabilidade [15].

Medida	Significado
Taxa de defeitos ( <i>failure rate, hazard function, hazard rate</i> )	Número esperado de defeitos em um dado período de tempo; é assumido um valor constante durante o tempo de vida útil do componente.
MTTF ( <i>mean time to failure</i> )	Tempo esperado até a primeira ocorrência de defeito.
MTTR ( <i>mean time to repair</i> )	Tempo médio para reparo do sistema.
MTBF ( <i>mean time between failure</i> )	Tempo médio entre os defeitos do sistema.

Estas medidas são determinadas estatisticamente, observando o comportamento dos componentes e dispositivos. Os fabricantes deveriam fornecer medidas de dependabilidade para os seus produtos, tanto para os componentes eletrônicos, como para os sistemas de computação mais complexos.

A taxa de defeitos de um componente e/ou dispositivo é mensurada em defeitos por unidade de tempo e é diretamente proporcional ao tempo de vida do componente e/ou dispositivo. Uma representação usual para a taxa de defeitos de componentes de *hardware* é dada pela curva da banheira. A Figura 6 apresenta esta curva onde podemos distinguir três fases:

- a) Mortalidade infantil: componentes fracos e mal fabricados;
- b) Vida útil: taxa de defeitos constantes;
- c) Envelhecimento: taxa de defeitos crescentes.

Os componentes de *hardware* só apresentam taxa de defeitos constante durante um período de tempo chamado de vida útil, que segue uma fase com taxa de defeitos decrescente chamada de mortalidade infantil. Para acelerar a fase de mortalidade infantil, os fabricantes recorrem a técnicas de *burn-in*, onde é efetuada a remoção de componentes fracos sendo estes substituídos por componentes que já sobreviveram à fase de mortalidade infantil através do processo de aceleração de operação.

Para componentes de *software* é questionável se a curva da banheira pode ser aplicada. Entretanto, pode ser observado que os componentes de *software* também apresentam uma

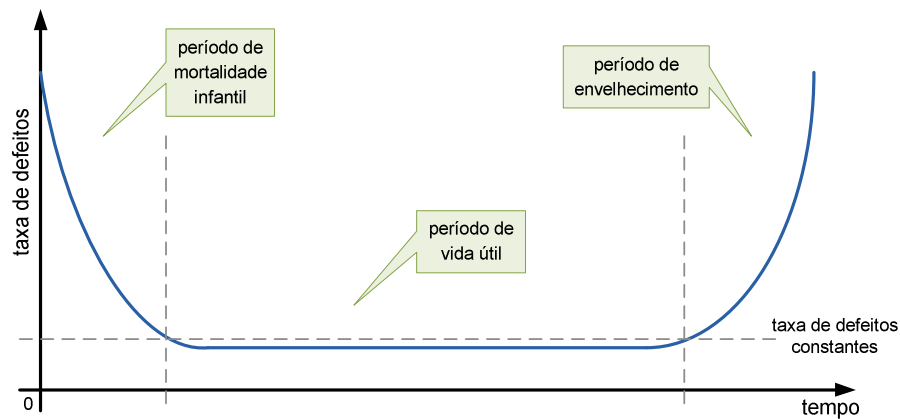


Figura 6: Curva da banheira [15].

fase de mortalidade infantil ou taxa de erros acentuados no início da sua fase de ensaios, que decresce rapidamente até a sua entrada em operação. A partir desse momento, o *software* apresenta um taxa de erros constante até que, eventualmente, precise sofrer alguma correção ou sua plataforma de *hardware* torne-se obsoleta. Nesse momento, a taxa de erros volta a crescer [22]. Quando referido a *software*, foi mencionado taxa de erros (*bugs*) ao contrário de falhas, já que erro é o termo usualmente empregado quando se trata de programas.

## 3 *Compatibilidade eletromagnética*

### 3.1 Introdução

O uso de dispositivos eletrônicos no cotidiano de milhares de pessoas assumiu um papel indispensável. Equipamentos e produtos eletrônicos, como telefones celulares, dispositivos de computadores sem-fio (*wireless*), além de transmissões de rádio e televisão, todos no mesmo ambiente eletromagnético, operando de maneira simultânea, silenciosa e invisível para a conveniência e satisfação das necessidades humanas da vida modernas [23]. Entretanto, apesar dos inúmeros benefícios e facilidades que estes equipamentos proporcionam ao nosso dia-a-dia, ocorre o aumento de Interferência Eletromagnética (*Electromagnetic Interference - EMI*) provocadas estes dispositivos, aos ambientes que estes estão inseridos, é uma situação inevitável e cada vez mais acentuada.

Neste cenário, o estudo da Compatibilidade Eletromagnética (*Electromagnetic Compatibility - EMC*), que segundo Robert Bosch [24] pode ser definida como a capacidade de um circuito, equipamento, dispositivo e/ou sistema operar satisfatoriamente no seu ambiente, sem que este, introduza e/ou sofra perturbações eletromagnéticas intoleráveis, tem um importante papel no projeto e desenvolvimento de sistemas embarcados.

Para a análise da EMC são adotados dois pontos de vista, o denominado emissão, onde a análise se concentra na EMI que o dispositivo e/ou sistema embarcado emite para o meio no qual está inserido. E o outro, denominado susceptibilidade, onde a análise verifica o comportamento do dispositivo e/ou sistema na presença de EMI proveniente de outros dispositivos e/ou sistemas.

Evidenciamos a importância destes estudos no emprego de diversas técnicas de projeto (como leiautes específicos, planos de terra, blindagem, conectores apropriados e filtragem dos pinos dos dispositivos), pela quantidade de publicações e normas técnicas relacionadas ao tema, e mais recentemente pelos requisitos das Diretivas Comunitárias Europeias [25] relacionadas à EMC, ou pelos requisitos das empresas automobilísticas para o forneci-

mento de equipamentos que serão instalados nos veículos durante sua fabricação, para que não apresentem defeito ou poluam o ambiente eletromagnético, ou nos regulamentos das companhias de aviação comercial que proíbem, aos passageiros, a utilização de aparelhos eletrônicos durante os voos, para evitar que fenômenos eletromagnéticos interfiram nos sistemas de navegação.

## 3.2 Conceitos Básicos

Nesta seção são definidos alguns conceitos importantes sobre Compatibilidade Eletromagnética necessários para o bom entendimento desta dissertação.

- **Emissão:** definida como o fenômeno no qual a energia eletromagnética interferência (ou distúrbio) é emitida por uma determinada fonte geradora para um dispositivo ou sistema “vítima” [25].
- **Nível de Emissão:** definido como o nível de interferência eletromagnética emitida a partir de um determinado dispositivo, equipamento ou sistema [25].
- **Limite de emissão:** definido como o nível máximo de emissão de uma fonte geradora de interferência eletromagnética [25].
- **Imunidade:** capacidade de um dispositivo, equipamento ou sistema executar suas funções na presença de uma perturbação, ruído ou interferência eletromagnética sem degradação de desempenho [25].
- **Nível de imunidade:** definido como a incidência máxima de ruído, perturbação ou interferência eletromagnética sobre um determinado dispositivo, equipamento ou sistema para que ele permaneça capaz de operar com o grau de desempenho exigido [25].
- **Limite de imunidade:** definido como o nível de imunidade mínimo sobre um determinado dispositivo, equipamento ou sistema para que ele permaneça capaz de operar com grau de desempenho exigido [25].
- **Susceptibilidade (eletromagnética):** situação inversa de imunidade, ou seja, é a incapacidade de um dispositivo, equipamento ou sistema executar suas funções na presença de uma perturbação eletromagnética [25].

- **Limite de interferência:** definido como o nível de interferência eletromagnética máxima admissível de um determinado dispositivo, equipamento ou sistema [25].
- **Nível de compatibilidade eletromagnética:** definido como o nível de interferência eletromagnética utilizado como referência para a fixação de limites de emissão e imunidade [25].

### 3.3 Ambiente eletromagnético

Um ambiente eletromagnético (*Electromagnetic Environment* - EME) é definido como a totalidade de fenômenos e interações eletromagnéticas existentes em um determinado local [25], tais como a rede de energia elétrica, o tipo de edificação ou instalação onde o equipamento está instalado, bem como outros equipamentos eletro-eletrônicos instalados e até o ambiente externo. Na medida em que ocorrem alterações no leiaute dos equipamentos, na edificação e/ou instalações e, principalmente, na instalação elétrica, teremos alteração no ambiente eletromagnético.

Neste contexto de EMC, é relevante a análise das alterações provocadas pela operação e/ou coabitação de dispositivos e sistemas eletrônicos aos parâmetros espaciais e temporais deste ambiente, como a força de campo elétrico, a densidade de potência, ou a densidade de energia.

### 3.4 Interferência eletromagnética

As interferências eletromagnéticas (*Electromagnetic Interference* - EMI) são perturbações causadas a um circuito, dispositivo e/ou sistema eletrônico por radiações eletromagnéticas emitidas a partir de uma fonte externa [25]. Que podem provocar no circuito “vítima” à degradação do seu desempenho, ou até mesmo a interrupção temporária ou permanente das suas funções.

Para compreender como estas interferências eletromagnéticas ocorrem, as suas consequências e as possíveis medidas para minimizar ou extinguir seus efeitos, é necessário compreender o modo como o acoplamento ou ligação condutiva, quais são as fonte causadores e/ou receptores e a também quais as frequências envolvidas.

### 3.4.1 Fontes de interferência eletromagnética

Segundo David A. Weston [26] as fontes de EMI podem ser classificadas em naturais e artificiais, como segue:

- a) **Fontes naturais:** são interferências ou distúrbios eletromagnéticos provenientes de fenômenos naturais como o ruído atmosférico, decorrente de descargas elétricas, ruídos cósmicos provocados por explosões do sol, etc [25].
- b) **Fontes artificiais:** são interferências ou distúrbios eletromagnéticos provenientes da ação do homem, como os provocados pelo acionamentos de cargas indutivas como motores elétricos, cargas resistivas como aquecedores, lâmpadas fluorescentes, equipamentos médicos de diatermia por radiofrequência (RF), aparelhos de micro-ondas, equipamentos de comunicação móvel, etc. [25].

### 3.4.2 Interferência eletromagnética conduzida versus irradiada

As emissões eletromagnéticas são fenômenos nos quais a energia de uma determinada fonte geradora flui para outro dispositivo ou circuito “vítima” [25]. Que podem ser divididos em dois tipos de acordo com o meio de propagação [27]:

- a) **Interferência conduzida:** aquela na qual a energia flui, da fonte geradora para o circuito ou dispositivo “vítima”, por um ou mais condutores de alimentação, pela tensão de referência (terra ou neutro), ou ainda por bornes de entrada ou saída de sinais [25]. E ocorre frequentemente em redes elétricas de corrente alternada [28], uma vez que este tipo de rede alimenta de forma paralela, muitos outros sistemas ruidosos (eletromagneticamente) como motores, contactoras e relés. A Figura 7 apresenta um exemplo de interferência eletromagnética conduzida através da rede elétrica, onde uma máquina elétrica afeta o funcionamento de um equipamento eletrônico (microcomputador).
- b) **Interferência irradiada:** aquela na qual a energia flui, da fonte geradora para o circuito ou dispositivo “vítima”, sob a forma de ondas eletromagnéticas pelo ar [25]. Tem-se um elemento irradiando um campo eletromagnético (fonte geradora de interferência) que se propaga na atmosfera e atinge equipamentos eletrônicos e sistemas (“vítima”), podendo levar a falha ou a interrupção de funções [28]. A Figura 8 apresenta um exemplo de interferência eletromagnética irradiada.

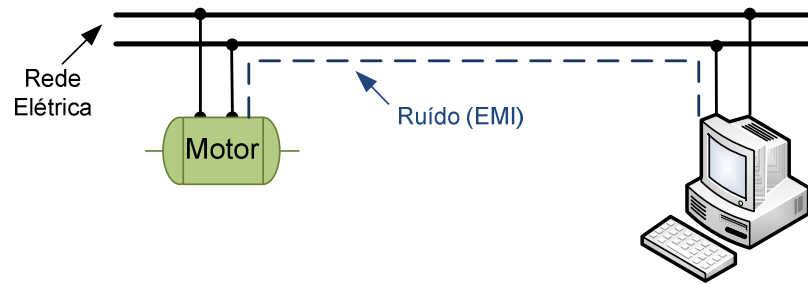


Figura 7: Exemplo de interferência eletromagnética conduzida [27].

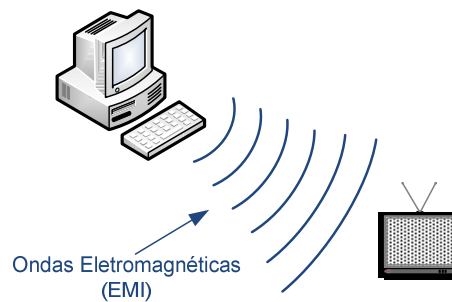


Figura 8: Exemplo de interferência eletromagnética irradiada [27].

### 3.4.3 Acoplamento de interferência eletromagnética

O meio utilizado por uma determinada fonte de interferência para transferir parte ou a totalidade da sua energia eletromagnética a outro circuito e/ou dispositivo “vítima” e definido como caminho de acoplamento [25]. No estudo e análise da Compatibilidade Eletromagnética (EMC) o caminho de acoplamento pode ser do tipo conduzido, irradiado ou ainda uma combinação de ambos.

Alguns exemplos de acoplamentos de interferência eletromagnética conduzida e/ou irradiada que podem estar presentes em um ambiente onde operam equipamentos e/ou dispositivos “vítima” são apresentados na figura 9.

Como o tipo de acoplamento existente entre uma determinada fonte geradora de interferência e um dispositivo e/ou circuito “vítima” é diretamente dependente de fatores como a frequência e o comprimento de onda da interferência eletromagnética, então os distúrbios com baixas frequências (grandes comprimentos de onda) propagam-se facilmente por meios condutivos, mas não tão eficientemente por meio irradiado. Entretanto, interferências eletromagnéticas com altas frequências (pequenos comprimentos de onda) se propagam eficientemente pelo ar, mas sua propagação é bloqueada pelas indutâncias das trilhas e fiações dos dispositivos e/ou circuitos “vítima” [30].



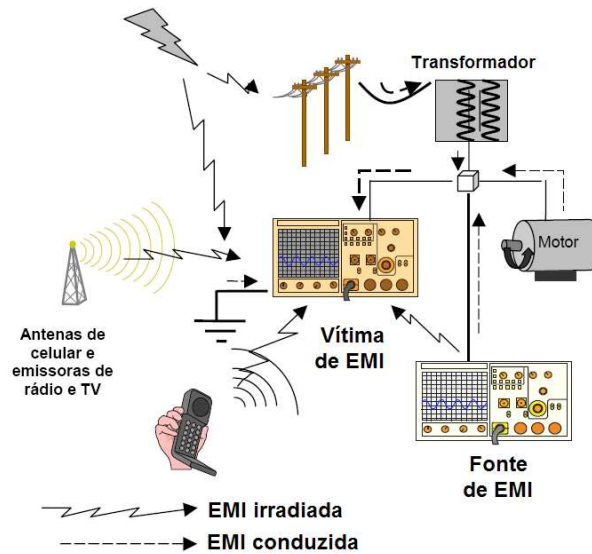


Figura 9: Exemplos de acoplamentos de EMI conduzida e irradiada [29].

### 3.5 Efeitos da interferência eletromagnética na eletrônica

A Integridade de Sinal (*Signal Integrity - SI*) é definida como a característica e/ou a habilidade de um determinado sinal gerar respostas corretas em um circuito e/ou sistema eletrônico [31]. Desta forma um sinal com boa integridade apresenta, entre as suas características, níveis de tensão e tempos de transição de acordo com as especificações de projeto do sistema em que este sinal está associado. A Figura 10 apresenta exemplos de sinais e suas características de integridade.

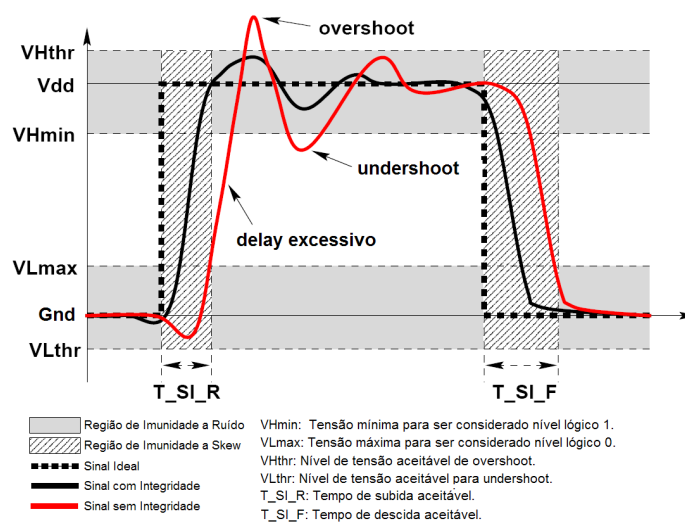


Figura 10: Exemplo de Integridade Sinais [31].

Porém, a presença de equipamentos de tecnologias diferentes em um mesmo ambiente, juntamente com a precariedade das instalações elétricas, ou ainda erros de projetos [27] contribuem decisivamente para a emissão e/ou susceptibilidade de *Systems-on-Chip*, de alto desempenho e alta densidade de integração, a interferências eletromagnética (EMI), potencializando distúrbios como:

- a) **Flutuações nas linhas de alimentação:** reduções momentâneas dos níveis de tensão das linhas de alimentação ( $V_{cc}$ ) de um dispositivo e/ou sistema eletrônico [32] [33];
- b) **Ground Bounce:** elevações momentâneas dos níveis de tensão das linhas de referência de tensão ( $Gnd$ ) de um dispositivo e/ou sistema eletrônico [32] [33];
- c) **Ruído das Linhas de Alimentação:** variações de corrente de carga provenientes do rápido chaveamento de circuitos [34] [35];
- d) **Skew:** diferenças nos tempos de propagação de dois ou mais sinais transmitidos simultaneamente através da rede de distribuição de um circuito e/ou sistema [36] [37] [38].

### 3.6 Normas IEC

A IEC (*International Electrotechnical Commission*) [39] é uma organização mundial, criada em junho de 1906 na Inglaterra, com a missão de preparar e publicar normas para elétrica, eletrônica e tecnologias relacionadas. Que servem como base para as normatizações nacionais e bem como referência para termos e acordos internacionais.

Através de seus membros, a IEC promove a cooperação internacional de todas as questões relacionadas com as normatizações eletrotécnicas, tal como o atendimento de conformidade as normas, nos campos de eletricidade, eletrônica e tecnologias relacionadas.

A IEC engloba todas as tecnologias associadas a eletroeletrônica, incluídas eletrônica, magnetismo e eletromagnetismo, eletroacústico, multimídia, telecomunicações, e produção e distribuição de energia, bem como as disciplinas associadas tal como a terminologia e símbolos, compatibilidade eletromagnética, medição e performance, dependabilidade, projeto e desenvolvimento, segurança e meio ambiente.

### 3.6.1 Norma IEC 61.000

A família de normas técnicas IEC 61.000 tem como objetivo abordar os assuntos básicos sobre EMC, englobando terminologia, descrições de fenômenos eletromagnéticos e o ambiente de EM, medição e técnicas de ensaios, e guias de instalação e mitigação. Esta é uma série grande e consideravelmente subdividida em padrões e relatórios técnicos, composta de nove partes, como pode ser visto a seguir. Note que os títulos das Partes 7 e 8 estão ainda abertos.

- **Parte 1:** Considerações gerais (introdução, princípios fundamentais, segurança), definições e terminologia;
- **Parte 2:** Descrição e classificação do ambiente, e níveis de compatibilidade;
- **Parte 3:** Limites de emissão e imunidade;
- **Parte 4:** Técnicas de medição e ensaios;
- **Parte 5:** Guias de instalação, métodos de mitigação e dispositivos;
- **Parte 6:** Padrões genéricos;
- **Parte 9:** Diversos.

Nesta dissertação foram aplicados da parte 4 os itens 17 e 29 desta norma, que são detalhadas a seguir.

#### 3.6.1.1 Norma IEC 61.000-4-17

A norma IEC 61.000-4-17 tem como objetivo estabelecer, em laboratório, uma base para ensaios e medições de equipamentos e/ou dispositivos elétricos e eletrônicos de baixa tensão sujeitos a ondulações de tensão (*ripples*) provenientes de sistemas externos de retificação de tensão ou carregadores auxiliares de bateria [11]. Define características para ensaios de ondulações de tensão: a forma da onda da tensão, níveis e amplitudes de tensão, os geradores de ensaios, e também o ambiente e os procedimentos adequados de ensaios.

Os níveis de tensão definidos, por esta norma técnica, para ensaios em fontes de alimentação DC de equipamentos e/ou dispositivos elétricos e/ou eletrônicos, são apresentados na Tabela 3.

Tabela 3: Níveis de Tensão de Ensaio [11].

Nível	Percentual da tensão nominal (%)
1	2
2	5
3	10
4	15
x	x

x: valor definido de acordo com a especificação do produto.

A amplitude da tensão de *ripple* ( $U_{\text{máx}} - U_{\text{mín}}$ ) é apresentada na Figura 11, sendo que os níveis de ensaio são tensões pico-a-pico expressas em pontos percentuais relativas à tensão nominal DC (UDC).

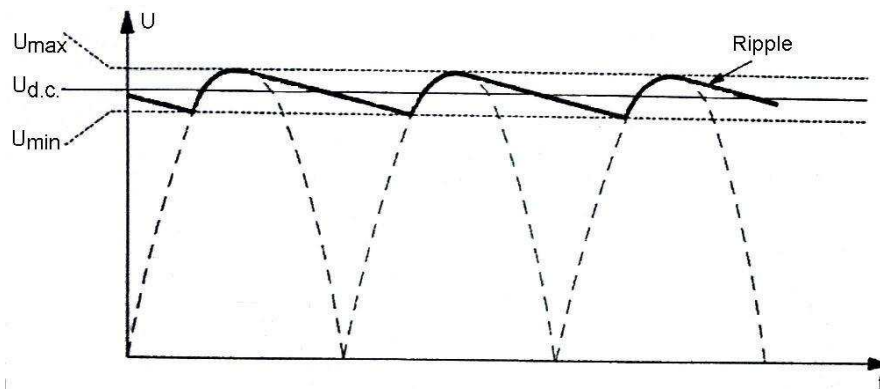


Figura 11: Retificador de monofásico [11].

Como exemplo temos um microcontrolador com tensão nominal de alimentação (UDC) igual a 3,3V, assim de acordo com a Tabela 3 e da Figura 11, e supondo que as alimentação deste dispositivo esteja sujeita a ondulações de 10% (nível 3), então teremos o valor de pico de  $U_{\text{máx}}$  igual a 3,47V (5% acima de UDC) e o valor de  $U_{\text{mín}}$  igual 3,13V (5% abaixo de UDC) (32).

### 3.6.1.2 Norma IEC 61.000-4-29

A norma IEC 61.000-4-29 tem como objetivo estabelecer estabelecer um método básico para ensaios relativos a imunidade de equipamentos e/ou dispositivos alimentados por fontes de corrente contínua externas de baixa tensão.

Faz-se necessário, para o bom entendimento desta dissertação, a definição de: queda de tensão, pequena interrupção e variação de tensão, pois existe diversos tipos de distúrbios relacionados às linhas de alimentação de equipamentos e dispositivos elétricos e eletrônicos.

- a) **Queda de tensão:** caracterizada como uma redução súbita na tensão de alimentação, do equipamento e/ou dispositivo, seguida da seu retorno a tensão nominal em um curto período de tempo [12].
- b) **Pequena Interrupção:** caracterizada como o desligamento momentâneo da tensão por um período de tempo menor que um minuto. Quedas de tensão superiores a 80% são na prática consideradas como interrupções [12].
- c) **Variação de tensão:** caracterizada como uma alteração gradual da tensão de alimentação para valores maiores ou menores que a tensão nominal (UT), sendo que a sua duração pode ser curta ou longa [12].

Nas Tabelas 4, 5 e 6 são apresentados os níveis de tensão percentuais relativos à tensão nominal (UT) e os tempos de duração sugeridos pela norma para ensaios em equipamentos e dispositivos elétricos e eletrônicos.

Tabela 4: Níveis de tensão e duração recomendadas para interrupções [12].

Ensaio	Nível de Tensão (% UT)	Duração (s)
Pequena Interrupção	0	0,001
		0,003
		0,01
		0,03
		0,1
		0,3
		1
		x

x: valor definido de acordo com a especificação do produto.

A norma IEC 61.000-4-29 sugere topologias de conexão entre os geradores de ruídos de alimentação e os dispositivos sobre ensaio (*Device Under Test - DUT*). As Figura 12 e 13 apresentam exemplos destas topologias.

Tabela 5: Níveis de tensão e duração recomendados para quedas de tensão [12].

Ensaio	Nível de Tensão (% UT)	Duração (s)
Queda de Tensão	40 a 70 ou x	0,01
		0,03
		0,1
		0,3
		1
		x

x: valor definido de acordo com a especificação do produto.

Tabela 6: Níveis de tensão e duração recomendados para variação de tensão [12].

Ensaio	Nível de Tensão (% UT)	Duração (s)
Variação de Tensão	85 a 120 ou 80 a 120 ou x	0,01
		0,03
		0,1
		0,3
		1
		x

x: valor definido de acordo com a especificação do produto.

### 3.6.2 Norma IEC 62.132

A família de normas técnicas IEC 62.132 tem como objetivo definir, em linhas gerais, uma metodologia de ensaios para estimar a imunidade de circuitos integrados a distúrbios eletromagnéticos na faixa de frequências de 150kHz a 1GHz. Como existe uma imensa gama de distúrbios eletromagnéticos que os circuitos integrados podem estar sujeitos quando estão em seu ambiente de operação, e que estes distúrbios podem ter características e causas distintas, a família de normas IEC 62.132 foi desenvolvida em cinco partes abordando os seguintes temas:

- **IEC 62.132-1:** Condições e definições gerais [10];
- **IEC 62.132-2:** Medições de imunidade eletromagnética em circuitos integrados sujeitos a distúrbios irradiados na faixa de frequência de 150kHz a 1GHz [40];
- **IEC 62.132-3:** Medições de imunidade eletromagnética em circuitos integrados

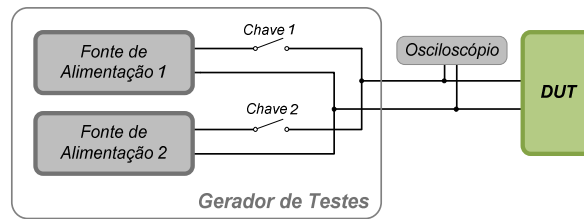


Figura 12: Gerador de ensaio com fontes de alimentação de chaveamento interno [12].

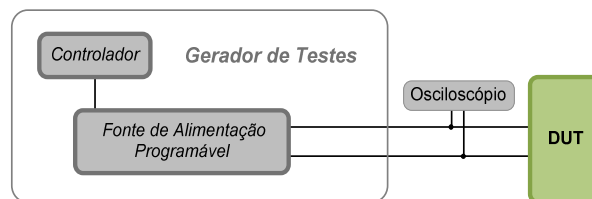


Figura 13: Gerador de ensaio baseado em uma fonte de alimentação programável [12].

sujeitos a distúrbios irradiados na faixa de frequência de 10kHz a 1GHz gerados por injeção de corrente (*Bulk Current Injection - BCI*) [41];

- **IEC 62.132-4:** Medições de imunidade eletromagnética em circuitos integrados sujeitos a radiofrequência (RF) conduzida de até um 1GHz [42];
- **IEC 62.132-5:** Medições de imunidade eletromagnética em circuitos integrados sujeitos a distúrbios conduzidos na faixa de frequência de 150kHz a 1GHz através do método de bancada da Gaiola de Faraday [43].

No estudo de caso desta dissertação, foram aplicados os itens 1, 2 desta norma, que são detalhadas a seguir.

### 3.6.2.1 IEC 62.132 - Parte 1

A norma IEC 62.132-1 [10] fornece informações e definições gerais para medições de imunidade eletromagnética conduzida e irradiada em circuitos integrados. Onde são fornecidas informações relativas às condições e procedimentos de ensaios, os equipamentos apropriados, além das necessidades e características da configuração (*setup*) de ensaio.

- a) **Placa de Circuito Impresso:** As topologias das placas de circuito impresso (*Printed Circuit Board - PCB*) utilizadas para ensaios de imunidade eletromagnética em

circuitos integrados podem variar de acordo com os métodos de injeção e medição dos distúrbios. Entretanto, a norma IEC 62.132-1 sugere para a garantia da eficiência, qualidade e confiabilidade dos ensaios algumas características específicas relativas ao leiaute desta placa como: dimensões de 10x10 cm; furos de fixação somente nos cantos da placa; bordas com largura de 5mm revestida de cobre e conectadas ao terra (*Gnd*); além de vias localizadas a uma distância mínima de 5mm da borda da placa [10]. Estas características são apresentadas na Figura 14.

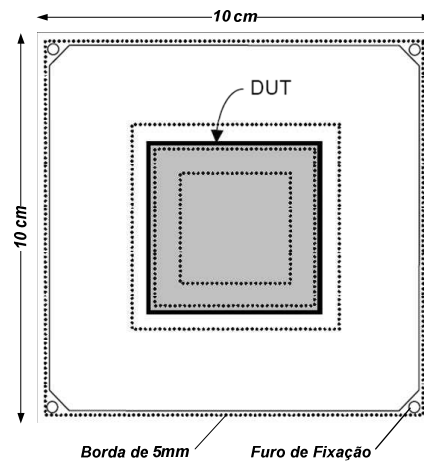


Figura 14: Características de placa de circuito impresso de ensaio [10].

A placa de ensaios de imunidade eletromagnética em circuitos integrados deve dispor de uma blindagem elevada de forma a garantir que somente o DUT esteja sujeito aos distúrbios eletromagnéticos. Assim sendo, o número de camadas da placa (*layers*) deve ser igual ou superior a quatro [10]. A Figura 15 e a Tabela 7 apresentam a disposição destas camadas e os seus conteúdos.

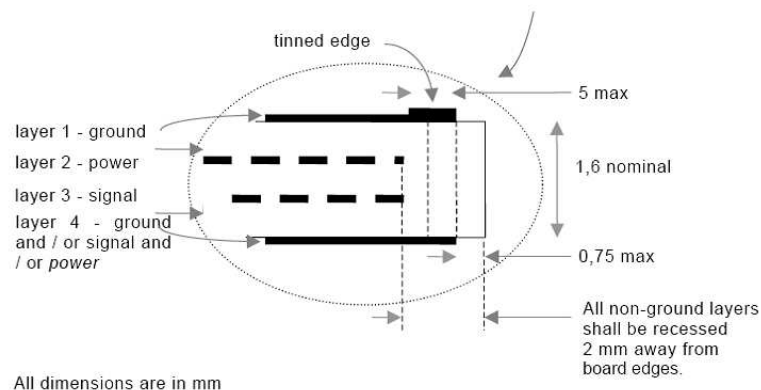


Figura 15: Disposição das camadas de uma placa de ensaio [10].

- b) **Pinos não utilizados do CI:** A norma IEC 62.132-1 recomenda que os pinos do circuito integrado sob ensaio, a exceção daqueles que possuam uma funcionalidade



Tabela 7: Conteúdo das camadas da placa de circuito impresso [10].

Camada ( <i>layer</i> )	Conteúdo
1 - superior ( <i>top</i> )	Contém somente o circuito integrado que está sobre ensaio (DUT) e um plano de terra ( <i>Gnd</i> ) cobrindo toda a área da placa. Esta camada não pode conter trilhas de sinais.
2	Contém somente um plano de alimentação (VCC) cobrindo toda a área da placa. Esta camada não pode conter trilhas de sinais.
3	Contém todas as trilhas de sinais utilizadas pelos componentes e circuitos da placa, além de sinais de VCC e Gnd (se necessários).
4 - inferior ( <i>bottom</i> )	Contém os demais componentes da placa, bem como qualquer tipo de sinal e/ou alimentação. Esta camada também deve dispor de um plano de terra para reforçar a blindagem da placa.

especial definida pelo fabricante, devem ser isolados ou conectados a uma carga. A Tabela 8 apresenta os valores padrões de carga recomendados pela norma para diferentes tipos de pinos de circuitos integrados. É importante ressaltar que estes valores recomendados pela norma podem ser alterados caso outros valores de carga sejam mais adequados a um determinado circuito integrado.

Os pinos não enquadrados em nenhuma das categorias listadas na Tabela 8 deverão ser conectados a cargas conforme exige sua funcionalidade e posteriormente informados no relatório do ensaio.

- c) **Fonte de Alimentação:** O DUT deve ser alimentado por uma fonte de tensão que não esteja sujeita aos distúrbios eletromagnéticos aplicados. E as linhas de alimentação devem ser filtradas de acordo com as especificações e recomendações do fabricante do circuito integrado.
- d) **Procedimento de ensaio:** O procedimento de ensaio também é bastante relevante para a garantia da confiabilidade dos ensaios. Sendo assim, as seguintes recomendações devem ser seguidas:
- Verificar a operação normal da placa de ensaios e do DUT, isto é, realizar uma verificação das funcionalidades do sistema sem a aplicação de distúrbios eletromagnéticos;

Tabela 8: Conexão dos pinos do circuito integrado [10].

Tipo de Pino	Carga no Pino
Analógicos	
Alimentação	Conforme indicação do fabricante.
Entrada	Se o CI não for internamente isolado é conectado em <i>pull-down</i> ( $10k\Omega$ ).
Sinais de Saída	Se o CI não for internamente isolado é conectado em <i>pull-down</i> ( $10k\Omega$ ).
Saída de Potência	Carga nominal conforme indicação do fabricante.
Digitais	
Alimentação	Conforme indicação do fabricante.
Entrada	Se o CI não for internamente isolado é conectado ao terra ( <i>Gnd</i> ). Se não puder ser aterrado, este é conectado em <i>pull-up</i> ( $10k\Omega$ ).
Saída	Conectado a um capacitor de 47pF referenciado ao terra ( <i>Gnd</i> ).
Controle	
Entrada	Se o CI não for internamente isolado é conectado ao terra ( <i>Gnd</i> ). Se não puder ser aterrado, este é conectado em <i>pull-up</i> ( $10k\Omega$ ).
Saída	Conforme indicação do fabricante.
Bidirecional	Conectado a um capacitor de 47pF referenciado ao terra ( <i>Gnd</i> ).
Analógico	Conforme indicação do fabricante.

- A faixa de frequência abordada pela norma IEC 62.132-1 é de 150kHz a 1GHz. Entretanto, na prática, a faixa de frequência dos ensaios de imunidade eletromagnética depende muito das frequências de interrupção do DUT, aliado também à metodologia de injeção de interferências. Neste sentido, a norma recomenda que os intervalos de frequência aplicados nestes ensaios sigam os valores apresentados na Tabela 9.

Tabela 9: Intervalos de frequência aplicados nos ensaios [10].

Faixa de Frequência (MHz)	0,15 - 1	1 - 100	100 - 1000
Intervalos Lineares (MHz)	$\leq 0,1$	$\leq 1$	$\leq 10$
Intervalos Logaritmos	$\leq 5\%$ de incremento		

Devemos ressaltar que as frequências críticas como, por exemplo, as frequências de sinais de sincronismo (*clock*) do sistema e/ou dos dispositivos auxiliares devem ser testadas usando intervalos menores a fim de se obter pontos de

susceptibilidade [10];

- O sinal de distúrbio usado para os ensaios de imunidade eletromagnética em circuitos integrados deve estar de acordo com as exigências da metodologia de ensaio adotada, ou seja, sinais de onda contínua (*continuous wave - CW*), sinais de 1kHz com amplitude modulada (*Amplitude Modulation - AM*) em 80% ou ainda um pulso modulado com a taxa da repetição 1kHz [40]. A Figura 16 apresenta exemplos destes tipos de distúrbios:

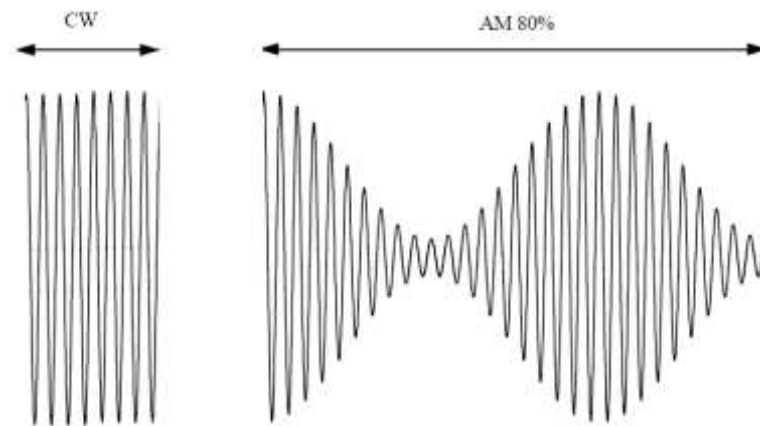


Figura 16: Exemplos sinais de onda contínua e amplitude modulada [10].

- O nível de potência incidente sobre o circuito integrado é diretamente relacionado ao tipo de sinal de distúrbio e conseqüentemente a metodologia de ensaio adotada(2). A equação 3.1 apresenta a relação entre as potências de uma onda contínua (PCW) e a potência de um sinal modulado em amplitude (PAM), onde  $m$  representa o índice de modulação.

$$P_{AM} = P_{CW} \frac{2 + m^2}{2(1 + m)^2} \quad (3.1)$$

- O monitoramento do circuito integrado deve ser realizado através de ensaios específicos considerando todas as suas funções operacionais. Desta forma, o comportamento funcional do circuito integrado deve ser estável ao longo do tempo, de tal forma que duas medições com estímulos e distúrbios semelhantes, mas separadas por um intervalo de tempo, devem produzir os mesmos resultados [10].
- e) **Relatório de ensaio:** O relatório de ensaio deve ser detalhado o suficiente para especificar todos os parâmetros e variáveis nas quais o circuito integrado esteve

sujeito durante o ensaio e permitir assim a sua repetibilidade [10]. São exemplos de informações relevantes para o relatório de ensaio os seguintes dados:

- Diagrama do circuito;
- Descrição da placa de ensaio na qual o CI está inserido (*layout*);
- Condições de operação do CI (tensão de alimentação e sinais de saída)
- Descrição do tipo de programa executado no CI;

Os resultados obtidos a partir dos destes ensaios de imunidade eletromagnética em circuitos integrados são dados fundamentais e também devem ser inseridos no relatório de ensaio seguindo a seguinte classificação:

- **Classe A:** Todas as funções do circuito integrado executam, conforme projetadas, durante e após a exposição a um distúrbio eletromagnético;
- **Classe B:** Todas as funções do circuito integrado executam durante a exposição, entretanto, uma ou mais delas podem sair das tolerâncias especificadas em projeto. Quando a exposição é removida, todas as funções retornam automaticamente aos limites normais;
- **Classe C:** Uma função do circuito integrado não executa durante a exposição conforme projetada, entretanto retorna automaticamente à operação normal depois que a exposição é removida;
- **Classe D:** Uma função do circuito integrado não executa durante a exposição conforme projetada e não retorna à operação normal até que a exposição seja removida e o circuito integrado seja reiniciado pela ação do operador;
- **Classe E:** Uma ou mais funções do circuito integrado não executam durante e após a exposição conforme projetadas, sendo sua operação normal impossível de ser retomada.

### 3.6.2.2 IEC 62.132 - Parte 2

A norma IEC 62.132-2 define uma metodologia de ensaio para medições de imunidade a distúrbios eletromagnéticos irradiados em circuitos integrados (CI) na faixa de frequência de 150kHz a 1GHz através do emprego de células TEM (*Transverse Electromagnetic*) e GTEM (*Gigahertz Transverse Electromagnetic*) [40].

O dispositivo sob ensaio (DUT) é montado em uma placa circuito impresso (PCB) e fixado na parede da célula através de uma porta de acoplamento [40]. Este acoplamento na

parede permite que a placa de ensaio e conseqüentemente o circuito integrado se tornem parte integrante da célula, ao contrário de ensaios convencionais nos quais a placa de ensaio é inserida no interior da célula alterando a orientação do campo elétrico [40].

A Figura 17 apresenta as configurações utilizadas em células TEM e GTEM para a realização dos ensaios de imunidade eletromagnética a distúrbios irradiados.

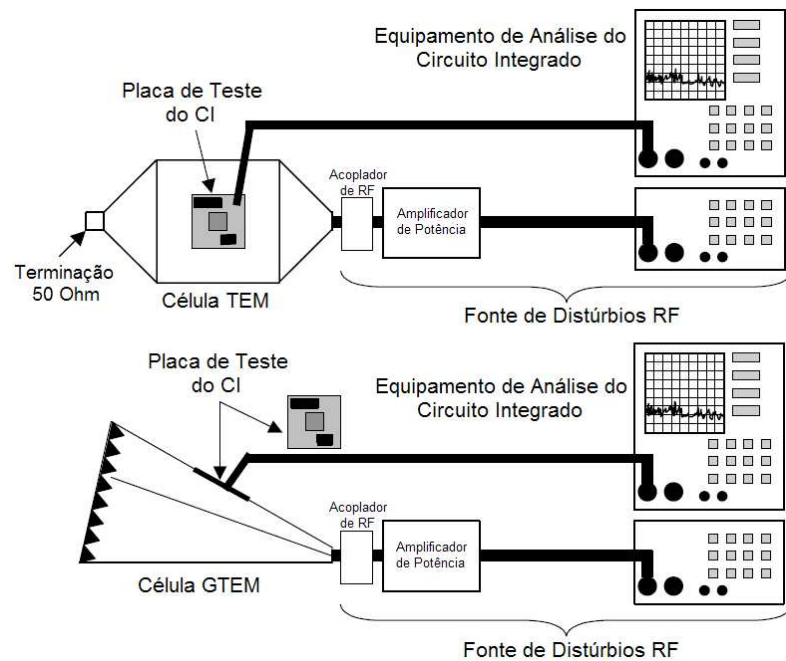


Figura 17: Configuração de ensaio com células TEM e GTEM [40].

## 4 *System on Chip (SoC)*

### 4.1 Introdução

Em 1980, a maioria dos circuitos integrados (CIs) ditos complexos eram compostos por dezenas de milhares de transistores. Atualmente, é possível encontrar CIs que contenham dezenas de milhões de transistores [44]. Entretanto, estima-se que até 2012 existam CIs contendo 4 bilhões de transistores, operando a uma frequência de 10GHz [45]. Este avanço tecnológico permite a implementação de sistemas computacionais completos em um único CI, denominados SoCs (*Systems-on-Chip*). Este aumento do número de transistores e da frequência de operação, o curto tempo de projeto e a redução do ciclo de vida dos produtos eletrônicos, caracterizam o atual cenário da indústria de semicondutores [46] [47].

Os SoCs podem ser compostos por processadores, memórias, módulos dedicados em hardware para realização de funções específicas, módulos de software e inclusive tecnologias mais recentes como MEMS (*Micro-Electro-Mechanical Systems*) integrados no mesmo CI [48] [49] [45]. A heterogeneidade apresentada por esses sistemas, ou seja, a possibilidade de combinar diferentes tecnologias no mesmo CI tem sido aproveitada em diversos segmentos da indústria. Tal fato é evidenciado pelo sucesso de produtos eletrônicos como telefones celulares, PDAs, aparelhos de jogos eletrônicos, aparelhos de DVDs, máquinas fotográficas digitais (integração de sensores ópticos com lógica digital e analógica), entre outros produtos [50] [51] [45]. Porém, combinar diferentes tecnologias no mesmo CI é um processo difícil, que implica no aumento de tempo e custo do projeto de novos produtos. Por outro lado, existe a necessidade da diminuição no tempo de lançamento do produto no mercado (*time-to-market*), o que vêm a contribuir para o domínio de mercado e aumento dos lucros.

A diminuição do ciclo de vida dos produtos eletrônicos é influenciada tanto a evolução tecnológica quanto o curto *time-to-market*. Segundo R.A Bergamaschi, et al [52], a diminuição do ciclo de vida dos produtos pode ser evidenciada pelo tempo que um dado

produto demora a ter o seu volume de vendas igual a um milhão de unidades. A Figura 18 apresenta um gráfico que relaciona a diminuição do ciclo de vida de produtos eletrônicos frente ao volume de vendas após a sua inserção no mercado. É possível observar, nesta mesma figura, que um televisor a cores demorou aproximadamente dez anos para vender um milhão de unidades, enquanto que, um aparelho de DVD demorou apenas um ano para atingir tal marca de vendas.

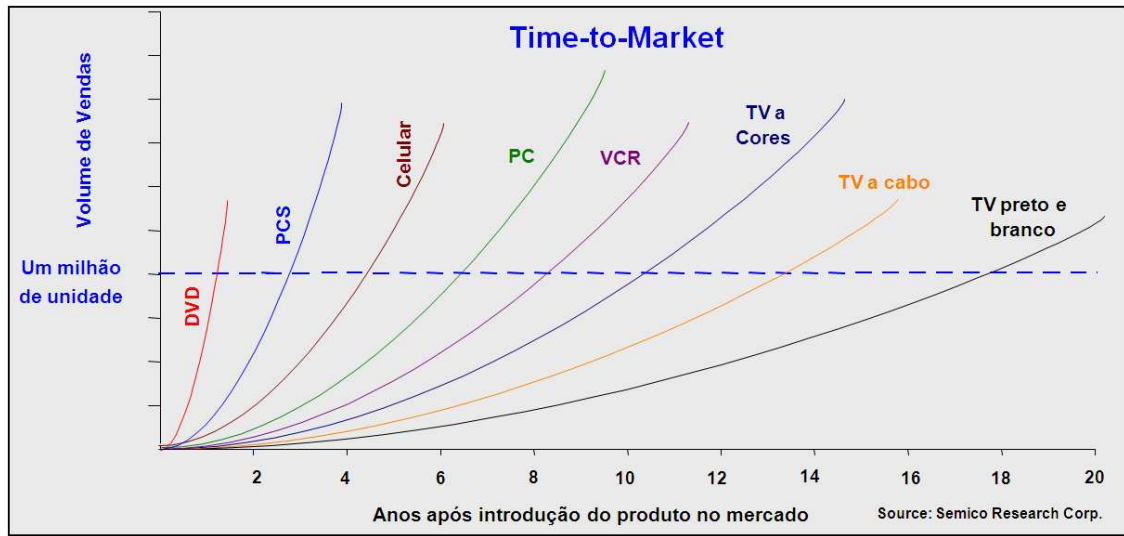


Figura 18: Diminuição do ciclo de vida dos produtos eletrônicos [52].

Podemos constatar que existe uma grande pressão em relação à diminuição dos tempos de desenvolvimento e de lançamento de um produto ao mercado antes que o mesmo já esteja ultrapassado tecnologicamente. Neste cenário, as empresas que desenvolvem e comercializam SoCs priorizam em seus projetos a utilização de núcleos de propriedade intelectual (*Intellectual Property Cores*), ou simplesmente núcleos, a fim de aumentar a produtividade, minimizando o tempo de desenvolvimento, e consequentemente, o *time-to-market* de seus produtos [46].

## 4.2 Linguagens para descrição de SoCs

O projeto concorrente de *hardware* e *software* é uma importante característica dos SoCs. O procedimento usual para o projeto destes sistemas é utilizar uma dada linguagem para descrever os módulos de hardware e outra linguagem para descrever os procedimentos de software. As linguagens de descrição de hardware, tais como *Verilog* [53] e *VHDL* (*VHSIC Hardware Description Language*) [54], têm como características comuns a hierarquia (descrição estrutural com utilização de componentes), o paralelismo

e a temporização. Já as linguagens para descrição de software, como C e C++ [55], são baseadas em um modelo de execução de instruções sequenciais adaptadas para a execução em processadores de propósito geral, sendo que estas geralmente não dispõem de suporte para paralelismo e temporização.

Exemplos de linguagens utilizadas para modelagem concorrente de *hardware* e *software* são o *SystemC* ([56], o *SystemVerilog* [53] e o *SpecC* [57]. Estas linguagens de descrição concorrente são derivadas de linguagens de descrição de *software* acrescidas de bibliotecas que simulam as características de paralelismo e temporização. A descrição de sistemas computacionais através destas linguagens concorrentes permite o aumento do nível de abstração do projeto de SoCs para os chamados níveis sistêmicos, onde os detalhes de baixo nível do projeto são abstraídos tornando o projeto das funcionalidades de cada núcleo e de suas interconexões mais amigável e eficiente do ponto de vista do *time-to-market*.

### 4.3 Arquitetura genérica de um SoC

Como já mencionado anteriormente, o mercado de semicondutores é caracterizado por produtos eletrônicos cada vez mais complexos e com tempo de vida útil cada vez mais reduzido. Com isso, tornou-se comum o desenvolvimento de SoCs a partir da integração de núcleos heterogêneos. Segundo a Associação das Indústrias de Semicondutores [45] estima-se que em 2012, 90% da área dos circuitos integrados VLSI seja preenchida por núcleos. Logo, é possível inferir que a produção de SoCs esta muito atrelada à criação e validação de núcleos que possam ser reutilizados em projetos distintos. Estima-se que desenvolver um núcleo que possa ser reutilizado por outros projetistas é substancialmente mais difícil (por um fator estimado entre 2 a 5 vezes maior) que desenvolvê-lo para um único projeto [45].

A arquitetura genérica de um SoC composto por núcleos não programáveis, processadores, memórias e interfaces de entrada e saída de dados que se comunicam através de uma estrutura de interconexão é apresenta na Figura 19 [58].

A Figura 19 apresentada a arquitetura genérica de um SoC composto por núcleos IP que podem ser, por exemplo, módulos de telecomunicação, processadores DSP (*Digital Signal Process*), decodificadores de MPEG2 (*Motion Picture Experts Group Layer-2*), decodificadores MP3 (*Motion Picture Experts Group Layer-3*), entre outros. Os núcleos IP provêm de fabricantes distintos (A, B) e estão integrados a uma estrutura de interconexão



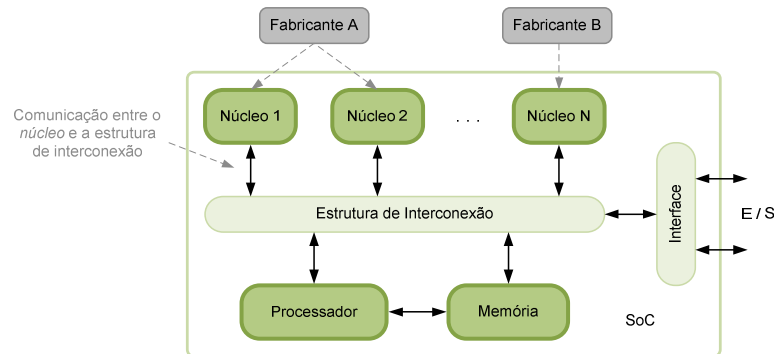


Figura 19: Arquitetura genérica de um SoC [59].

que pode ser um barramento ou ainda uma rede intra-chip. As interfaces com o mundo externo (E/S) são utilizadas para interconectar periféricos ao SoC, como por exemplo, uma porta UART (*Universal Asynchronous Receiver Transmitter*) ou ainda uma porta USB (*Universal Serial Bus*).

Gupta e Zorian [60] definem um núcleo como um módulo de *hardware* pré-projetado e pré-verificado, que pode ser usado na construção de uma aplicação maior ou mais complexa em um CI. Estes núcleos podem ser classificados em três categorias: *soft core*, *firm core* e *hard core*.

- **Soft core** é um núcleo desenvolvido em uma linguagem de descrição de *hardware* (*VHDL*, *Verilog* e *SystemC*). As principais vantagens apresentadas por este tipo de núcleo são a independência de tecnologia e a flexibilidade de projeto. Estas características possibilitam o projetista modificar o *soft core*, visando adequar as funcionalidades deste núcleo em prol do seu projeto. Em contrapartida, esta flexibilidade repassa também ao projetista a responsabilidade do atendimento das restrições temporais (*timing*) do núcleo e do sistema (SoC).
- **Firm core** é um *netlist* que apresenta uma flexibilidade menor em relação ao *soft core* e na maioria dos casos depende de uma tecnologia específica. As vantagens dos *firm cores* em relação aos *soft cores* são relativas a uma estimativa de desempenho mais próxima da realidade e uma melhor proteção da propriedade intelectual.
- **Hard core** é geralmente uma descrição de um leiaute posicionado e roteado em um ASIC ou FPGA. Para atingir um maior desempenho de processamento, um baixo consumo e uma área menor, estes são otimizados para uma determinada tecnologia.

Assim, os *hard cores* possuem a garantia dos tempos de propagação dos sinais no núcleo (*timing*), além de proverem alta proteção à propriedade intelectual. Como consequência destas características, os *hard cores* são fortemente dependentes da tecnologia e a sua flexibilidade é bastante limitada.

Os núcleos são usualmente interconectados aos SoCs através de duas estruturas de interconexão: fios ponto-a-ponto dedicados ou barramentos compartilhados, simples ou hierárquicos. Entretanto, torna-se necessário definir alguns conceitos que caracterizam as mesmas antes de descrever estas estruturas citadas acima.

- **Paralelismo:** relaciona-se à possibilidade de transferência e/ou recepção de dados entre dois ou mais pares de núcleos simultaneamente;
- **Consumo de energia:** determina a quantidade de energia consumida por um determinado circuito e/ou núcleo;
- **Escalabilidade:** refere-se à capacidade de interconectar componentes adicionais à estrutura de interconexão, sem comprometimento significativo no desempenho global do sistema [61];
- **Reusabilidade:** é a capacidade de utilizar uma dada estrutura de interconexão em projetos distintos. Essa estrutura deve proporcionar facilidades para que um grande número de núcleos possa trocar informações eficientemente. Isso, tanto para pares de núcleos como para comunicações concorrentes entre vários pares.

#### 4.4 Interconexão de núcleos baseada em fios ponto-a-ponto dedicados

Na interconexão baseada em fios ponto-a-ponto dedicados (*dedicated wires*), os núcleos são interligados diretamente um ao outro formando uma conexão chamada ponto-a-ponto. Este tipo de estrutura de interconexão é bastante eficaz se cada núcleo tem que se comunicar com um pequeno número de núcleos adjacentes, assim o desempenho oferecido pela estrutura ponto-a-ponto pode ser considerado bom, pois cada comunicação ocorre independentemente das demais. Caso pretenda-se interligar um núcleo a muitos outros através deste tipo de interconexão, o número de fios dedicados aumenta proporcionalmente ao número de núcleos, que pode gerar um congestionamento de fios [62]. Tal característica pode ser considerada uma limitação à medida que o projeto SoCs pode

contar com dezenas ou centenas de núcleos [45] [61]. Outro fator limitante está no fato que o projeto deste tipo de estrutura é específico e, portanto, a sua reutilização é bastante limitada. Na Figura 20 é apresentada uma estrutura genérica de interconexão de núcleos baseada em fios ponto-a-ponto.

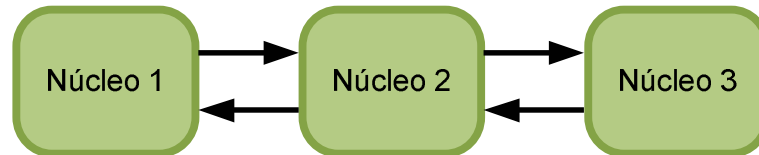


Figura 20: Estrutura genérica de interconexão baseada em fios ponto-a-ponto [59].

## 4.5 Interconexão de núcleos baseada em barramentos compartilhados

Uma estrutura de interconexão baseada em barramentos de dados compartilhados é mais reutilizável (*shared data bus*). Além da reutilização, a baixa área de silício e a baixa latência contribuem para que esta estrutura de interconexão de núcleos seja a mais utilizada atualmente [63].

Um barramento consiste em um conjunto de fios que se conectam fisicamente a diferentes núcleos do SoC de forma paralela, e sobre o qual os dados são transmitidos e recebidos. Os núcleos conectados a este tipo de interconexão podem ser classificados de duas formas, mestres ou escravos.

Um núcleo, dito mestre, é uma unidade que controla a transferência de dados no barramento, ou seja, é este núcleo que solicita aos demais, a transmissão ou a recepção dos dados. E um componente, denominado escravo, é uma unidade que apenas responde às solicitações dos núcleos mestres, enviando ou recebendo dados [64]. Como exemplo desta estrutura, pode-se citar um microprocessador, atuando como mestre, e uma memória que efetua o papel de escravo do barramento. Neste contexto, as informações são lidas ou escritas da/na memória a partir dos sinais de comando gerados pelo microprocessador.

A maioria dos barramentos define um método de arbitragem responsável pelo controle de acesso dos núcleos mestres a este. Dentre os métodos existentes destacam-se o cen-

tralizado e o distribuído [64]. No método centralizado, um dispositivo denominado como árbitro ou controlador de barramento é responsável pela atribuição de acesso ao barramento através de sinais de requisição e de permissão. Quando o árbitro percebe uma requisição de acesso direito, ele gera, quando for possível, um sinal de permissão ao mestre que solicitou o acesso ao barramento. Entretanto, o método distribuído ou descentralizado não possui um árbitro, sendo o papel de monitoramento das linhas de requisição delegado e desempenhado pelos próprios núcleos do barramento. Desta maneira, cada núcleo possui sua prioridade na ordem de requisições e pode, ou não, utilizar o barramento. A Figura 21 apresenta uma estrutura de interconexão baseada em barramento com árbitro.

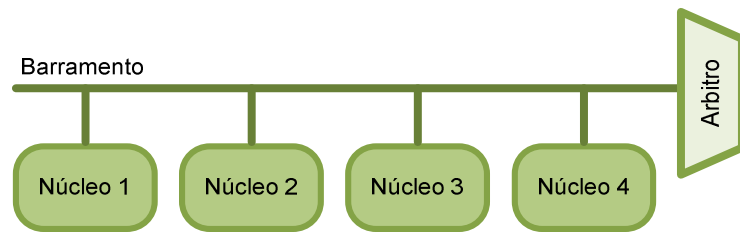


Figura 21: Estrutura de interconexão em barramento com árbitro central [59].

Apesar de barramentos serem reutilizáveis, podem-se citar algumas desvantagens significativas desta forma de interconexão de núcleos de SoCs:

- a) Ausência de paralelismo: O paralelismo de transferência de dados inexistente no uso de barramentos simples, já que todos os núcleos compartilham o mesmo canal de comunicação possibilitando assim, apenas uma transação de comunicação por vez. Esta desvantagem pode ser parcialmente contornada através do uso de, por exemplo, hierarquias de barramentos, onde o problema é apenas minimizado pelo incremento no número de transações simultâneas, porém de forma bastante limitadas (duas a três transações simultâneas).
- b) Baixa escalabilidade: A escalabilidade também é limitada a dezenas de núcleos [63] [65] [66].
- c) Aumento do consumo de energia. O emprego de interconexões via barramento gera um aumento no consumo de energia devido à existência de longos fios de conexão [50], além disso, o acréscimo de núcleos ao barramento aumenta a capacitância total do sistema, reduzindo conseqüentemente o desempenho elétrico do SoC.

Entre as arquiteturas de barramento de dados compartilhados intra-chip encontradas na literatura, destacam-se: AMBA da ARM [67], *Avalon* da Altera [68], *CoreConnect* da IBM [69] e OPB da Xilinx [70]. Estas arquiteturas de barramento geralmente estão vinculadas às arquiteturas de um determinado processador, tal como o AMBA vinculado ao processador ARM, o *CoreConnect* vinculado ao processador *PowerPC*, o *Avalon* vinculado ao processador *Nios* e o OPB vinculado ao processador *MicroBlaze*.

## 4.6 Interconexão de núcleos baseada em redes intra-chip (NoCs)

Diversos autores preveem que as estruturas de interconexão citadas acima irão tornar-se fatores limitantes para grandes projetos, em função do aumento da complexidade dos sistemas e do aumento dos requisitos de largura de banda. Uma possível solução para a integração de núcleos são as redes intra-chip [71].

Uma estrutura de interconexão que pode solucionar os problemas relacionados ao uso de barramentos, simples ou hierárquicos, são as redes intra-chip (*network on chip* - NoC)[65] [50]. NoCs herdam das redes de computadores e de sistemas distribuídos as características das camadas de protocolos e o conceito de ligação de nodos à rede. Nas NoCs, os núcleos do sistema são interligados por meio de uma rede composta por roteadores e canais ponto-a-ponto. A comunicação entre os núcleos ocorre pela troca de mensagens transferidas por meio de roteadores e canais intermediários até atingir o seu destino [65], conforme podemos ver nas Figura 22 e 23.

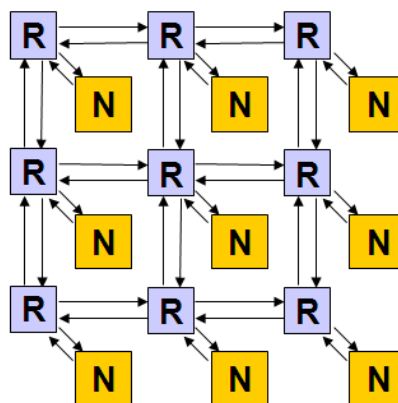


Figura 22: Interconexão de núcleos baseada em redes intra-chip (NoCs) [50].

Espera-se que os sistemas baseados em NoCs forneçam boas soluções para o reuso de núcleos [66], uma vez que as NoCs possuem as seguintes características: eficiência no

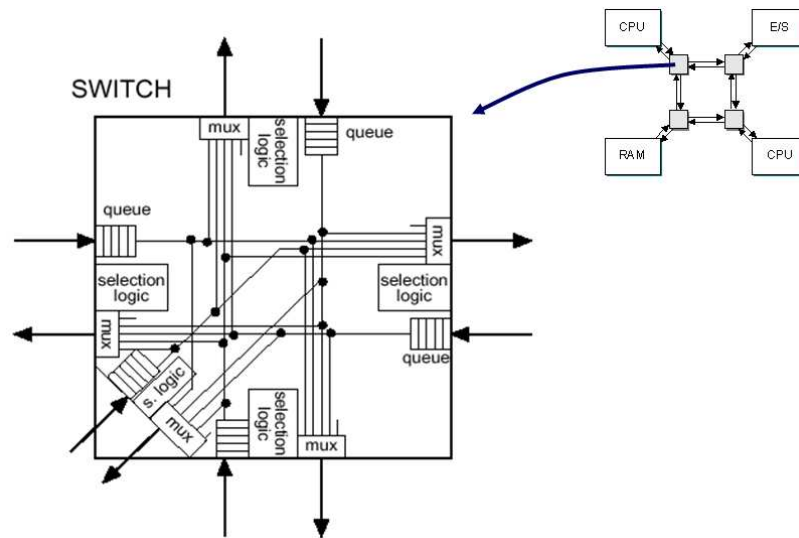


Figura 23: Exemplo de uma rede intra-chip (NoCs) [50].

consumo de energia [50]; largura de banda escalável, quando comparada à arquiteturas de barramento tradicionais; reusabilidade; decisões de roteamento distribuídas [63]; paralelismo na comunicação. Embora tenham como desvantagem maior custo em área de silício e latência na comunicação, esses problemas serão certamente atenuados pela grande disponibilidade de transistores e por soluções arquiteturais que permitirão reduzir a latência da rede e seus efeitos no desempenho da aplicação [62].

## 5 *Tecnologia dos dispositivos reconfiguráveis*

### 5.1 Introdução

Determinadas aplicações computacionais desenvolvidas e implementadas tanto em *software* quanto em *hardware* necessitam frequentemente de alterações estruturais e/ou funcionais. No caso de projetos desenvolvidos em *software* existe a necessidade do emprego de um *hardware* subjacente para a realização destas alterações. Este *hardware* normalmente é composto por um processador de conjunto de instruções (*Instruction Set Processor – ISP*), associado a uma memória. Os ISPs podem ser programados e reprogramados para executar uma ou mais aplicações específicas, sendo necessário apenas o preenchimento da sua memória com instruções.

Entretanto, no caso de implementações via *hardware*, as aplicações flexíveis são desenvolvidas através do emprego de dispositivos configuráveis como PLAs (*Programmable Logic Arrays*) e PALs (*Programmed Array Logic*), e principalmente através dispositivos reconfiguráveis como CPLDs (*Complex Programmable Logic Device*) e FPGAs (*Field Programmable Gate Array*), sendo este último adotado nesta dissertação como dispositivo sob ensaio.

Estes dispositivos revolucionaram a tradicional distinção entre *hardware* e *software*, visto que as funcionalidades e as estruturas dos projetos, em *hardware*, podem ser alteradas de forma total (todo o projeto é modificado), parcial (apenas uma parte do projeto é modificado) ou ainda de forma dinâmica (com o dispositivo em funcionamento, uma parte do projeto é modificada).

## 5.2 Arquitetura dos FPGAs

Os FPGAs foram introduzidos em 1985 pela Xilinx Inc. [72]. Desde então, um grande número de outras companhias vem desenvolvendo estes dispositivos a exemplo de Actel, Altera, Atmel, Plessey, Plus Logic, AMD, Quicklogic, Algotronix, Concurrent Logic, Crosspoint Solutions, entre outras [73].

Os FPGAs são chamados assim porque, além de disporem de uma estrutura similar aos PALs <sup>1</sup>, também possuem uma disposição de portas semelhante às encontradas em ASIC (*Application Specific Integrated Circuit*). Estas características fazem dos FPGAs excelentes dispositivos para o uso em prototipação de ASICs, ou ainda, dependendo do projeto, em substituição a estes dispositivos.

A arquitetura básica dos FPGAs, apresentada na Figura 24, consiste em uma matriz de elementos agrupados em blocos lógicos configuráveis. A configuração das interconexões, entre estes blocos, é realizada pelo usuário via *software* através de barramentos de chaves semelhantes às interconexões de uma PAL. O mesmo raciocínio ocorre no interfaceamento dos FPGAs com outros dispositivos e/ou com o mundo externo, realizado através de blocos de entradas e/ou saídas também configuráveis pelo usuário.

Cada fabricante possui em sua linha de produtos FPGAs com características e arquiteturas próprias, entretanto, em termos gerais todos seguem a arquitetura básica apresentada na Figura 4.1 que consiste em:

- Blocos de configuração lógica: São nos blocos lógicos (*Configurable Logic Block – CLB*) que as funções lógicas do projeto são de fato implementadas. Em algumas arquiteturas estes blocos lógicos possuem também recursos como *flip-flop* e/ou registradores para implementação de, por exemplo, circuitos sequenciais síncronos;
- Blocos de I/O configurável: Um bloco de I/O configurável é usado para receber sinais externos ao chip e/ou enviá-los para fora novamente. Este bloco consiste em um *buffer three-state* (alta impedância) que pode ser configurado bidirecionalmente. Além disto, há frequentemente um *flip-flop* na saída deste bloco, de modo que os sinais controlados por *clock* possam ter saídas diretamente conectadas aos pinos externos sem apresentar atrasos de propagação (*delay*) significativos [74];

---

<sup>1</sup> A tecnologia PAL (*Programmable Array Logic*) possui em sua estrutura apenas um único nível de programação, constituído de um plano de portas lógicas AND. Este nível alimenta um segundo plano de portas lógicas OR fixas, fazendo com que estes dispositivos tenham custos de produção menores e um melhor desempenho quando comparado a tecnologia PLA.



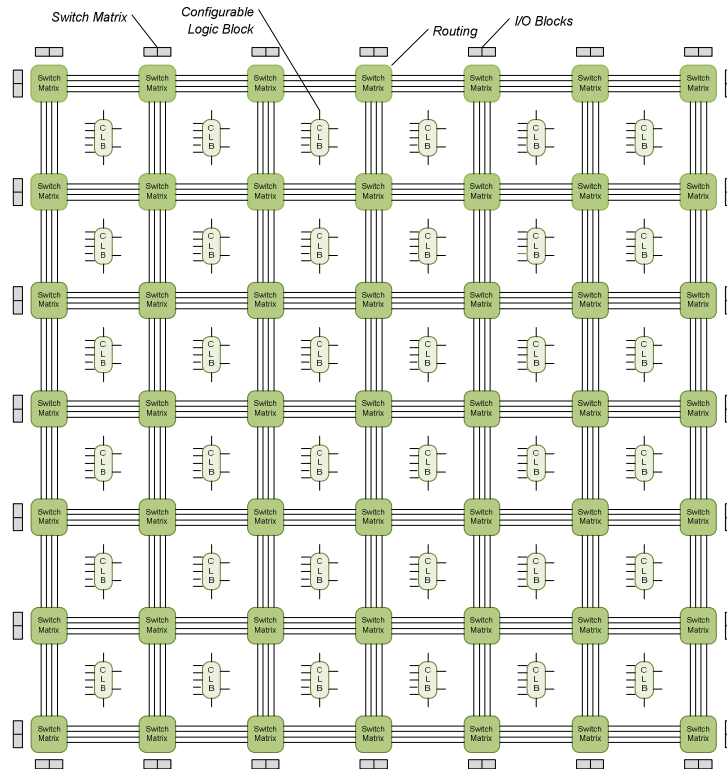


Figura 24: Arquitetura básica de um FPGA [73].

- Conexões globais: São interconexões que circundam os blocos lógicos e os blocos de I/O formando uma rede de linhas e colunas que se conectam através de chaves;
- Matrizes de conexão: São chaves de interconexão, programáveis na fase de roteamento automático, que permitem a interconexão entre os blocos lógicos através das conexões globais;
- Elementos programáveis: Em um FPGA, a matriz de conexão e os blocos lógicos são configurados e interconectados eletricamente através de chaves eletrônicas programáveis. As propriedades destas chaves, tais como tamanho, resistência, e capacitância, delimitam as características elétricas destes circuitos integrados [75]. Existem basicamente três tipos de tecnologias de programação:

1. Tecnologia SRAM (*Static Random Access Memory*): usa uma célula de memória RAM (*Random Access Memory*) estática para programar e controlar transistores de tecnologia CMOS (*Complementary Metal Oxide Semiconductor*) ou multiplexadores [74]. Tendo em vista que a tecnologia SRAM é volátil, isto é, o FPGA deverá ser configurado a cada vez que o dispositivo for alimentado, existe a necessidade intrínseca do uso de uma memória externa permanente,

- que pode ser do tipo PROM (*Programmable Read Only Memory*), EPROM (*Erasable Programmable Read Only Memory*), EEPROM (*Electrically Erasable Programmable Read Only Memory*) ou outro meio de armazenamento para prover o vetor de configuração do dispositivo [74]. Entretanto, os dispositivos com tecnologia SRAM possuem a vantagem de ser configurados rapidamente, além de requererem apenas a tecnologia padrão de circuitos integrados para a sua fabricação [76]. Existe um grande número de fabricantes que empregam a tecnologia SRAM em seus dispositivos a exemplo de Xilinx [72], Plessey [77], Algotronix [78], Altera [79], Concurrent Logic [80], Toshiba [81] e Atmel [82];
2. Tecnologia antifusível: o antifusível é um dispositivo que apresenta, no seu estado não programado, uma impedância bastante elevada entre seus terminais. Quando se aplica entre estes terminais uma tensão, o antifusível “queima” criando uma conexão de baixa impedância [74]. Entre as vantagens da tecnologia antifusível estão o tamanho reduzido, a baixa capacitância quando não programado, e a baixa resistência quando programado. Porém, como desvantagens, esta tecnologia não permite reprogramação, necessita de transistores com larga área a fim de suportar correntes de programação (aproximadamente 5mA), além da necessidade de um circuito de programação com alimentação extra para o fornecimento dos níveis de tensão de configuração, que são próximos de 20volts. A tecnologia de antifusível é encontrada nos FPGAs produzidos pela Actel [83], Quicklogic [84] e Crosspoint [85];
  3. Tecnologia de porta flutuante: Nesta tecnologia, os comutadores programáveis são baseados em transistores com portas (*gates*) flutuante iguais aos usados nas memórias EPROM e EEPROM. Cada célula da memória possui um transistor MOS (*Metal Oxide Semiconductor*) construído com dois gates, sendo que um deles se encontra não conectado (flutuante) ao barramento da memória e isolado por material de alta impedância. Esta tecnologia é semelhante à empregada em EPROMs, exceto pela remoção das cargas da porta flutuante, que pode ser feita eletricamente no circuito sem luz ultravioleta. Esta característica permite à fácil e rápida reconfigurabilidade do dispositivo, o que pode ser muito útil em determinadas aplicações. Entretanto, a tecnologia de portas flutuantes apresenta a desvantagem de que a célula de EEPROM é aproximadamente duas vezes o tamanho de uma célula de EPROM [22]. A tecnologia de porta flutuante baseada em EEPROM é usada em dispositivos fabricados pela AMD [86], Lattice [87] e Altera [88].

Os FPGAs também podem dispor em sua arquitetura básica, de circuitos de *clock* para o chaveamento de sinais de sincronismo para cada bloco lógico, além de recursos de lógica adicional tais como ALUs (*Arithmetic and Logic Unit*), memórias e decodificadores.

Todas estas características de flexibilidade de arquitetura e reconfigurabilidade tornam os FPGAs excelentes dispositivos para projetos que necessitam de uma rápida inserção no mercado, sem que a preocupação com o custo de produção seja desprezado [73].

## 6 *Circuitos assíncronos*

### 6.1 Introdução

Atualmente a maioria dos circuitos utilizados são síncronos, sendo o seu comportamento controlado por um sinal de relógio global (*clock*), que estabelece o momento no qual os registradores capturam os dados e sincroniza as operações do sistema. Com este modelo o sinal do relógio deve ser tal que respeite o caminho crítico e considere os atrasos máximos. Como consequências desta abordagem temos os problemas advindos do sinal de relógio: geração de ruído, maior consumo de potência, estrutura para distribuição deste sinal, escorregamento do sinal de relógio, etc.

Com a demanda por circuitos de baixo consumo e operando com baixa tensão de alimentação, o uso de circuitos síncronos exige maiores cuidados durante a elaboração do projeto, como a necessidade de técnicas e pesquisas na área de distribuição do sinal de relógio, controle da frequência e escorregamento do sinal [3] [4].

Neste cenário, os circuitos assíncronos estão sendo pesquisados nos últimos anos como uma alternativa e solução aos problemas e gargalos impostos pela tecnologia dos circuitos síncronos [5] [6] [7]. Apresentando os benefícios de não possuírem um sinal de relógio global, desempenho baseado na média e não no pior caso como nos circuitos síncronos, adaptável a variações de processo e do ambiente (tensão, temperatura, etc), componentes modularizáveis, baixo consumo e baixa emissão de ruído. Tendo como contrapartida os desafios referentes falta de ferramentas de CAD para o desenvolvimento, aumento considerável de área para reduzir transitórios e falta de projetistas com experiência em circuitos assíncronos.

Para melhor compreender o paradigma destes circuitos, a seguir serão apresentados as definições, características e formas de implementações. Assim, tomaremos como ponto de partida a definição extraída do livro de Lavagno e Sangiovanni-Vincentelli [89] para circuitos digitais síncronos e assíncronos, uma visão estrutural, também referenciada por

Calazans [6]:

Um **circuito digital assíncrono** é uma interconexão arbitrária de portas lógicas, com a restrição de que nenhum par de saídas de portas lógicas conecta-se entre si. Um **circuito digital síncrono** é um circuito digital assíncrono onde todos os laços de realimentação passam através de um elemento de memória controlado por um sinal de relógio do circuito. O comportamento do relógio deve ser tal que nenhum evento pode se propagar ao longo de um laço de realimentação sem “parar” em algum ponto deste pelo fato de encontrar um elemento de memória inativo, que impede de ir adiante. Além disso, nenhum evento deve poder atingir a entrada de um elemento de memória fora de certas janelas de tempo pré-especificadas antes e depois de uma transição de relógio, denominadas tempo de estabelecimento (*setup time*) e tempo de manutenção (*hold time*), respectivamente. Isto vale tanto para o ambiente onde opera o sistema (que controla as entradas externas), quanto para os circuitos combinacionais internos percorridos pelos sinais (L.Lavagno e L. Sangiovanni-Vincentelli, 1993)

## 6.2 Definições e conceitos básicos

Neste seção serão abordados os conceitos e definições básicas relacionadas ao paradigma dos circuitos assíncronos, tais como o princípio de funcionamento e as características de um operador assíncrono. Mais detalhes podem ser encontrados em [90] [91] [92] [6] [5] nos quais esta sessão está baseada e os quais para melhor compreensão desta dissertação devem ser consultados.

### 6.2.1 Princípio de funcionamento

A diferença principal entre os circuitos síncronos e assíncronos esta na forma de gestão da sequência de dados. A seguir será apresentada a forma básica de funcionamento deste dois paradigmas de circuitos.

- a) **Funcionamento do circuito síncrono** Os circuitos síncronos tem os seu comportamento controlado por um sinal de relógio global (*clock*), que estabelece o momento no qual os registradores capturam aos dados e sincronizam as operações do sistema. Com este modelo o sinal do relógio deve ser tal que respeite o caminho crítico e considere os atrasos máximos. A Figura 25 apresentada um circuito síncrono genérico.

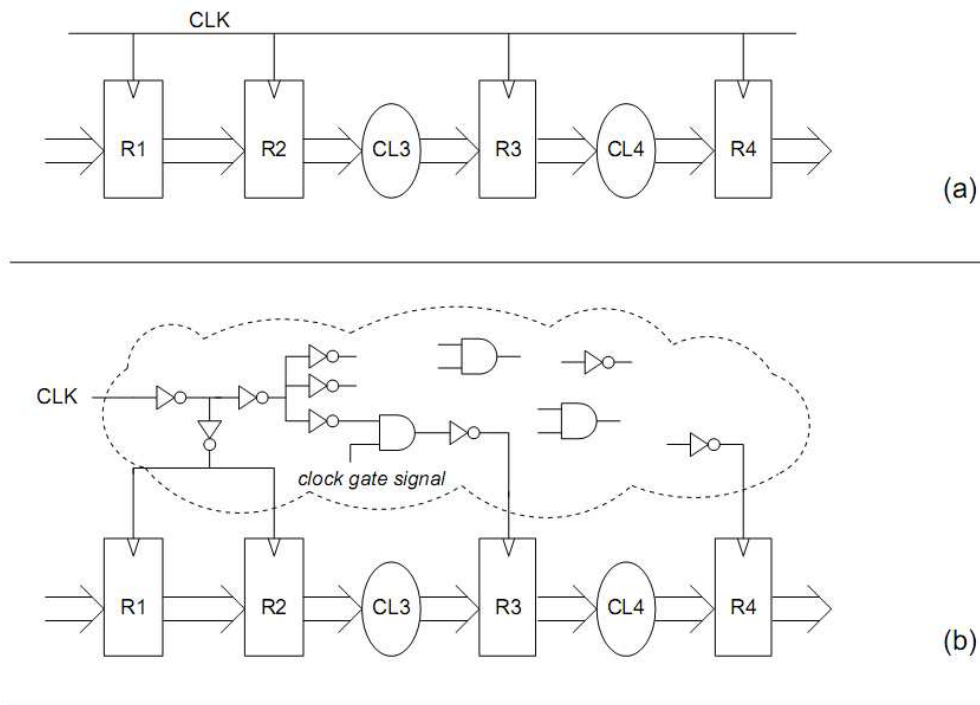


Figura 25: (a)Circuito síncrono genérico, (b)árvore de *clock* [5].

Neste tipo circuito o tempo é definido pela frequência do sinal de relógio. Por convenção utilizada, os componentes sequenciais operam na transição positiva do sinal do relógio (transição no nível lógico ‘0’ para ‘1’) ou transição negativa do sinal do relógio (transição no nível lógico ‘1’ para ‘0’). Quando ativo seus componentes utilizam os dados provenientes do elemento predecessor (dados de entrada) tratam e processam através dos seus circuitos combinatórios, os quais ficarão disponíveis para o elemento predecessor no próximo período do sinal de relógio utilizá-los (dados de saída). Assim sendo, conforme o sinal de relógio, todos os elementos liberam os dados periodicamente e simultaneamente.

- a) **Funcionamento do circuito assíncrono** O circuito assíncrono, diferentemente dos síncronos, tem suas atividades controladas localmente utilizando para isto canais de comunicação que detectam a presença de dados em suas entradas e saídas, sendo este protocolo de comunicação baseado em solicitação e reconhecimento (*handshaking* ou *request/acknowledge*). Na figura 26 é apresentado um circuito síncrono genérico e na figura 27 é apresentado de forma genérica a comunicação entre os circuitos assíncronos.

O operador assíncrono A2 deve proceder o seu cálculo utilizando o dado proveniente de A1. Quando o operador A1 termina de processar os dados, envia um sinal de requisição (*request*) ao operador A2 com a finalidade de indicar que existe um dado

válido disponível. Caso o operador A2 esteja disponível, então fará o processamento dos dados e vai indicar ao elemento A1 por um sinal de reconhecimento (*acknowledge*) que pode disponibilizar um novo dado.

Desta forma os elementos são comandados por um sinal de controle local e a validade dos dados deve ser assegurada localmente.

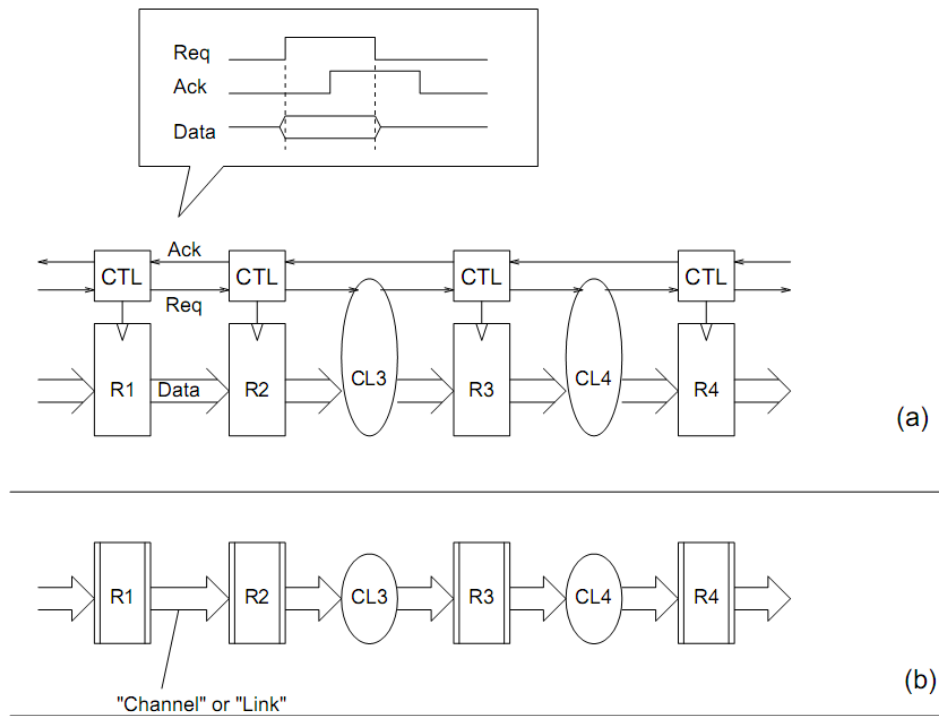


Figura 26: (a)Circuito assíncrono genérico, (b)fluxo de dados [5].

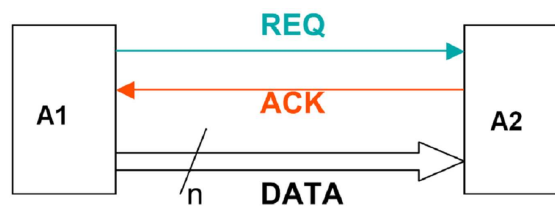


Figura 27: Comunicação no circuito assíncrono genérico.

## 6.2.2 Característica de um operador assíncrono

Na Figura 27 são apresentados dois operadores assíncronos que se comunicam com outras partes do circuito, e não somente entre si. Também é apresentada uma forma de comunicação onde o sinal de requisição está juntamente com os dados (*bundled data*). A seguir apresentamos as características de um operador assíncrono.

- a) **Latência:** corresponde ao tempo necessário para processar um dado de entrada e disponibilizar na saída. É igual ao tempo gasto através dos circuitos combinacionais.
- b) **Ciclo:** corresponde ao tempo necessário para processar uma sequência de dois dados sucessivos.
- b) **Profundidade do *pipeline*:** corresponde ao número de dados que um operador pode memorizar.

## 6.3 Implementação de circuitos assíncronos

Como os operadores assíncronos se comunicam entre si através de requisição e reconhecimento, um sinal de requisição de um operador ativa o outro operador ao qual está conectado. Os sinais devem estar válidos durante todo o tempo e não sendo permitidos transitórios (*hazard*) e corridas (*race*). Neste ponto a síntese de circuitos assíncronos é mais restrita, pois devem ser livres de transitórios [93] [90]. A seguir serão apresentadas as hipóteses destes problemas na implementação dos circuitos assíncronos bem como noções sobre os protocolos de comunicação e codificação de dados.

### 6.3.1 Problemas temporais

Durante um estado estável de um circuito, os dados estão certamente válidos, porém entre um estado válido e outro, fenômenos transitórios poderiam ser produzidos. Estes fenômenos podem ser devido a transição dos sinais de entrada ou aos atrasos existentes nos fios e portas lógicas do circuito. Portanto, como os fenômenos transitórios nas lógicas combinacionais dependem dos atrasos na lógica, na colocação das portas lógicas ou na implementação do circuito, uma atenção especial é dada a estes detalhes na concepção dos circuitos assíncronos.

Os transitórios podem ser classificados com a seguir:

- a) **Transitório estático:** ocorre quando um sinal deveria permanecer constante, mas altera o seu valor momentaneamente e retorna ao normal. Pode surgir nas saídas de portas lógicas OU/E quando os dois sinais de entrada alteram simultaneamente.
- b) **Transitório dinâmico:** ocorre quando um sinal deveria trocar de nível lógico somente uma vez, no entanto troca mais de uma vez.



- b) **Transitório combinacional:** ocorre quando aparece um funcionamento não esperado. Podem ser suprimidos na fase de especificação e implementação.
- c) **Transitório sequencial:** ocorre devido aos circuitos de realimentação, podem ser devidos aos fenômenos de corridas críticas.

Os transitórios podem ocorrer e não influenciar o perfeito funcionamento do circuito ou podem afetar o funcionamento e causar problemas ao circuito. Estes transitórios podem ser identificados e suprimidos durante a concepção dos circuitos, alterando a especificação das funções.

### 6.3.2 Gestão dos problemas temporais

Nos circuitos síncronos, o comportamento destes são controlados por um sinal de relógio global (*clock*), que estabelece o momento no qual os registradores capturam os dados e sincroniza as operações do sistema. Este sinal do relógio deve ser tal que respeite o caminho crítico e considere os atrasos máximos, deve também respeitar os tempos de transição (*setup*) e manutenção (*hold*) dos dados. Estas condições são suficientes para assegurar o perfeito funcionamento e garantir a validade dos dados. Como consequências desta abordagem, os circuitos síncronos são livres dos problemas de transitórios.

Por outro lado, os circuitos assíncronos são vulneráveis a estes problemas de transitórios. Assim uma atenção especial deve ser dada a concepção destes circuitos a fim de torná-los livres dos transitórios. Portanto surgem duas opções, a primeira é que os operadores sejam livres de transitórios e tenham a capacidade de indicar a validade dos dados de saída, sinalizando o fim do processamento dos dados através de um sinal de requisição (*request*). A outra opção consiste em fazer a imposições de certas restrições de tempo de maneira que os transitórios sejam tolerados de forma a não comprometer o funcionamento do sistema. A seguir serão abordadas as classificações dos circuitos e as hipóteses temporais.

## 6.4 Classificação

Conceber circuitos assíncronos robustos a transitórios, consiste em controlar as hipóteses temporais adotadas na etapa de concepção e elaboração do projeto do circuito. As diferentes hipóteses sobre os atrasos e modo de funcionamento permite classificar os circuitos assíncronos em categorias conforme apresentada na Figura 28.



Figura 28: Classificação dos circuitos assíncronos.

### 6.4.1 Modelagem de atrasos

Os projetos de circuitos assíncronos são diretamente influenciados pelo pressupostos adotados para atrasos que os modelos estabelecem. Na Figura 29 podemos ver os modelos de atraso de circuito descritos a seguir.

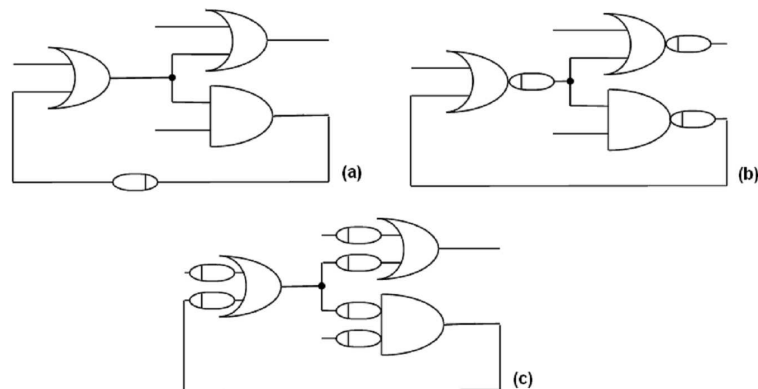


Figura 29: Modelos de atraso dos circuitos assíncronos: (a) atraso de realimentação, (b) atraso de porta, (c) atraso de fio [6].

- a) **Atraso de realimentação:** onde todo o laço de realimentação contém pelo menos um elemento de atraso.
- b) **Atraso de porta:** onde existe um elemento de atraso em cada saída de porta lógica.
- b) **Atraso de fio:** onde existe um elemento de atraso em cada entrada de porta lógica.

### 6.4.2 Circuitos insensíveis ao atraso (*Delay Insensitive - DI*)

Estes circuitos tem como pressuposto funcionarem corretamente independente dos atrasos de portas e fios, onde podem apresentar atrasos finitos de magnitude arbitrária.

Com esta imposição o operador receptor deve informar ao emissor que a informação foi recebida corretamente. Para construção deste tipo de circuito é necessário ter considerações bem restritas no projeto e utilização de estrutura de portas muito complexas, com consequência de aumento da área de silício.

### 6.4.3 Circuitos quase insensíveis ao atraso (*Quasi-Delay Insensitive - QDI*)

Para reduzir a complexidade do circuito, é possível introduzir certas considerações nas bifurcações (*forks*), sendo chamadas de isocrônicas (*isochronic*), onde os atrasos entre as conexões são consideradas iguais, conforme ilustrado na Figura 30. Este modelo de circuito é chamado de quase insensível a atrasos (Quasi-Delay Insensitive - QDI), que para o seu correto funcionamento é assumindo estes “*isochronics forks*”. É um modelo de portas simples sob a hipótese temporal, que permite implementar com o uso de portas padrões.

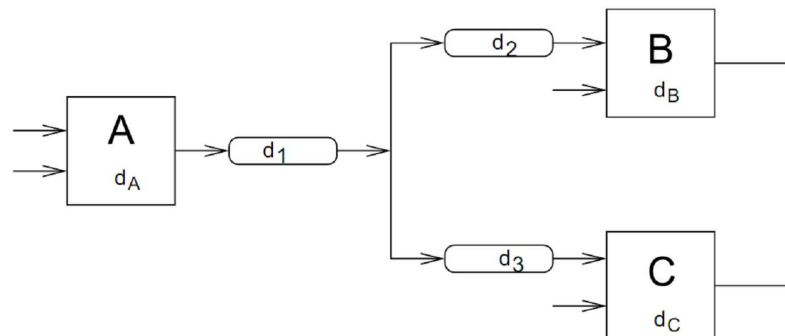


Figura 30: Bifurcações isocrônicas (*isochronics forks*) [5].

### 6.4.4 Circuitos independente da velocidade (*Speed Independent - SI*)

Neste modelo de circuitos assíncronos, assume-se atrasos finitos, mas de magnitude arbitrária nas portas lógicas, e os fios como elementos ideais. Este modelo é equivalente ao QDI mas não é possível implementá-los com a tecnologia atual, devido a hipótese sobre os *forks*, onde temos a hipótese de um atraso nulo sobre os fios, a qual não possível de obter-se.

### 6.4.5 Circuitos *Micropipelines*

Os circuitos *micropipelines* foram apresentados por Sutheland em 1989 [94]. Nestes, os circuitos combinacionais processam os dados, e a parte de controle gerencia o fluxo de dados conforme mostrado na Figura 31. Atrasos são inseridos nos sinais de requisição de forma tal a respeitar o tempo de latência do circuito combinacional. Este tipo de circuito permite a construção de um *pipeline* elástico, pois o número de dados presentes no *pipeline* pode variar.

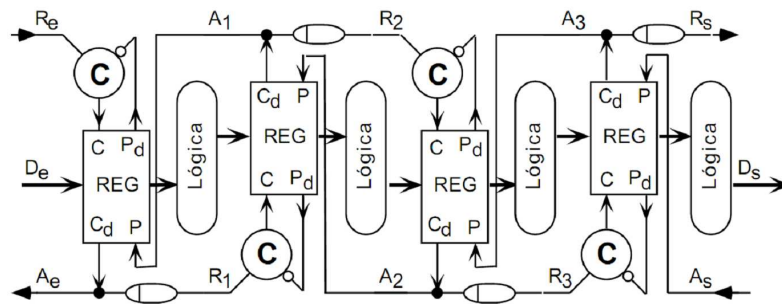


Figura 31: Circuitos Micropipelines [5].

### 6.4.6 Circuitos de Huffman

Nestes circuitos, conhecido por circuitos de modo fundamental, o modelo de atrasos é idêntico aos circuitos síncronos: se supõem que os atrasos em todos os elementos são nos terminais ou de valores conhecidos. O controle do caminho dos dados é feito pelo uso de relógios locais, onde o período é estimado pelo tempo de latência máxima do seu circuito combinatório. O modo de operação fundamental assume que os sinais nas entradas irão mudar somente quando o circuito está em uma condição estável, isto é, somente quando não houver mudanças nos sinais internos da rede.

## 6.5 Protocolos de comunicação

A comunicação nos circuitos assíncronos possui o protocolo baseado em solicitação e reconhecimento (*handshaking* ou *request/acknowledge*). O protocolo de comunicação utilizado pode ser de 2 fases ou 4 fases. O protocolo de 2 fases é sensível a transição do sinal de controle e o protocolo 4 fases é sensível ao nível do sinal de controle.

### 6.5.1 Protocolo 2 fases

Este protocolo é denominado NRZ (*Not Return to Zero*), ou ainda *half handshake*. Este protocolo é sensível a transição do sinal de controle, conforme demonstrado abaixo e na Figura 32.

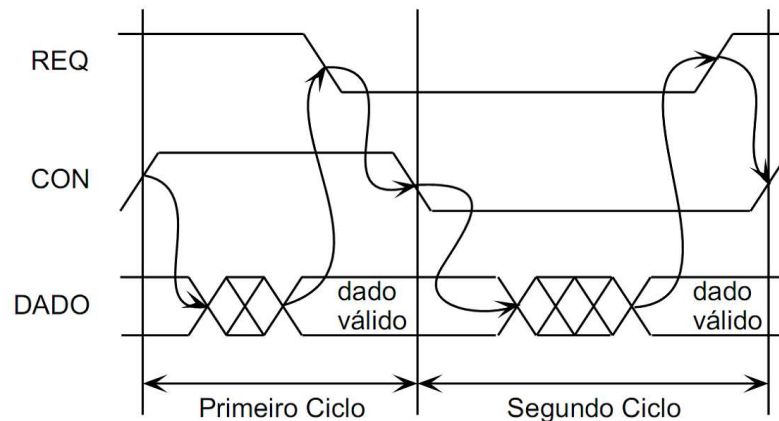


Figura 32: Protocolo 2 fases [6].

- **Fase 1:** O receptor é ativado: ele trata a informação e sinaliza o reconhecimento (*acknowledge*)
- **Fase 2:** O emissor é ativado: ele detecta o sinal de reconhecimento e envia um nova informação.

### 6.5.2 Protocolo 4 fases

Este protocolo é denominado RZ (*Return to Zero*), ou ainda *full handshake*. Este protocolo é sensível ao nível do sinal de controle, conforme demonstrado na Figura 33.

- **Fase 1:** O emissor (*sender*) envia um dado válido para o receptor (*receiver*).
- **Fase 2:** O receptor detecta a presença de dado válido e envia um sinal de reconhecimento (*acknowledge*) para o emissor.
- **Fase 3:** O emissor detecta o sinal de reconhecimento e envia um dado vazio (*empty*).
- **Fase 4:** O receptor detecta o dado vazio (*empty*) e retira o sinal de reconhecimento, colocado ao nível lógico inicial.

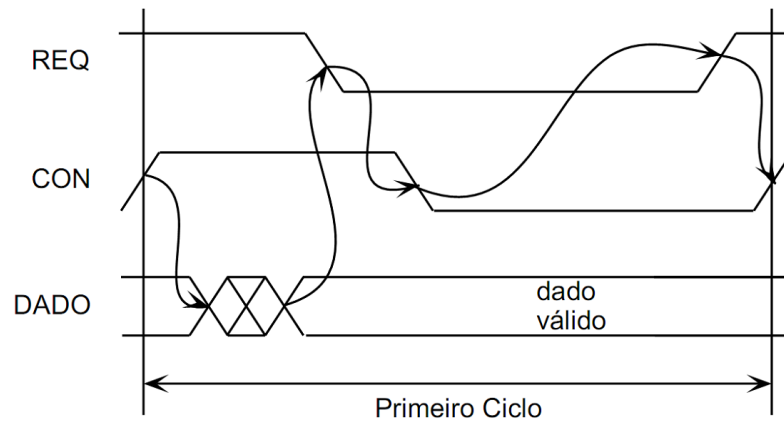


Figura 33: Protocolo 4 fases [6].

## 6.6 Representação da informação

Nos circuitos assíncronos a validade dos dados deve ser assegurada localmente, e para isto os dados são agrupados com a informação de sua validade. A seguir serão apresentados os principais tipos de codificação de dados utilizados.

### 6.6.1 Codificação de trilha única (*Single Rail*)

Neste modelo, também conhecido por *Bundled Data*, a codificação física da informação assume que cada *bit* está associado a um fio, onde o valor da tensão no fio determina o valor do bit, níveis lógicos '0' ou '1'. Juntamente com o barramento de dados é agrupado um fio para indicar a validade da informação, conforme ilustrado na Figura 34. Este tipo de codificação pode ser utilizada com o protocolo de 2 ou 4 fases.

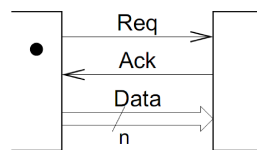


Figura 34: Codificação de trilha única (*Single Rail*) [5].

Apesar dos protocolos de 2 fase serem geralmente mais velozes do que os protocolos de 4 fases, por não terem o retorno a zero entre a transição de dados, o que torna mais eficiente em relação ao tempo e consumo de energia, na prática estes não são muito utilizados devido a complexidade exigida dos circuitos para responder a estímulos por transição ao invés de nível, no caso do protocolo 4 fases.

Outro ponto a ser evidenciado é que devido ao fato da validade da informação ser sinalizada por um fio separado, este tem que garantir que todos os dados estejam válidos

quando for realizada a sinalização de requisição (*request*), assim surge uma dificuldade de projeto, que consiste em fazer com que o atraso da propagação do sinal de requisição seja igual ou superior ao maior o atraso do barramento de dados.

### 6.6.2 Codificação de trilha dupla (*Dual Rail*)

Neste modelo, diferentemente da codificação de trilha simples, a codificação física da informação assume que cada *bit* está associado a um par de fios, onde a validade da informação e o sinal de requisição estão codificados juntamente. A forma de representação da informação depende do protocolo de comunicação utilizado conforme mostrado a seguir.

- **4 Fases:** Para esta implementação cada valor lógico é dado um código composto de um valor verdadeiro (d.t - true) e valor falso (d.f - false) (Figura 35), assim o valor lógico '1' é representado por d.t = '1' e d.f = '0' e o valor lógico '0' é representado por d.t = '0' e d.f = '1', ainda temos a representação do estado vazio (*empty*) por d.t = '0' e d.f = '0'. A representação d.t = '1' e d.f = '1' não é permitida. Assim sendo, entre os dados válidos deve ser intercalado um dado vazio, com isto não é permitida a passagem de um dado diretamente para outro, o que permite implementar um detector de validade de dados com um simples OR dos fios d.t e d.f. Com esta forma de codificação este protocolo se torna robusto, fazendo com que dois operadores possam se comunicar independente dos atrasos existentes entre eles.

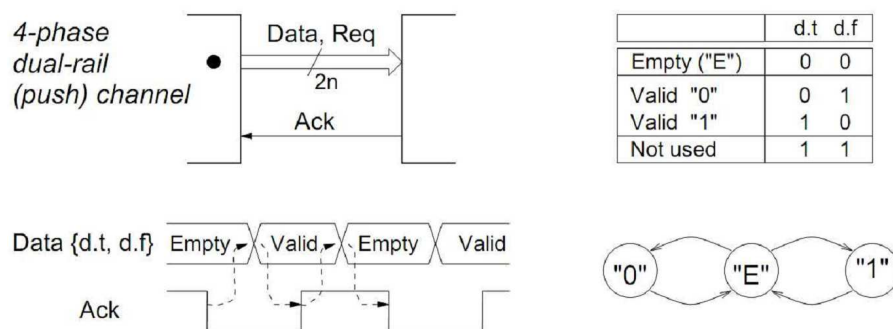


Figura 35: Codificação de trilha dupla - 4 fases (*Dual Rail*) [5].

- **2 Fases:** Nesta implementação o que determina a existência de um novo dado válido é a transição de um dos fios (d.t ou d.f), diferentemente do protocolo anterior, não existe o estado vazio (*empty*), este protocolo é dito sensível a transição. Duas formas de codificação pode ser utilizadas:

- a) Cada fio é dedicado a um valor e cada transição indica que um novo dado está válido. Na Figura 36, o fio d.f é dedicado ao valor lógico '0' e d.t ao valor lógico '1', assim um evento sobre d.f indica que existe um novo valor lógico '0' e um evento sobre d.t indica que existe um novo valor lógico '1'.

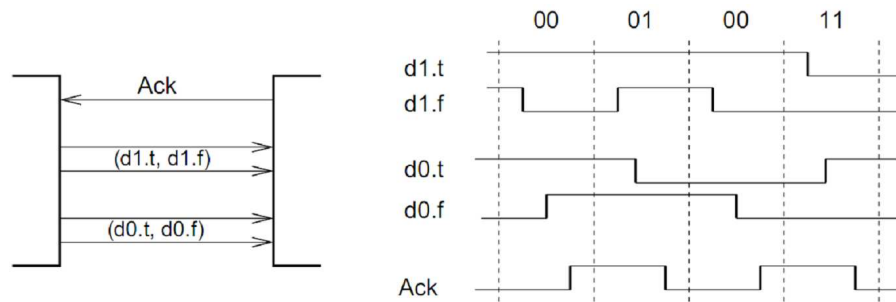


Figura 36: Codificação de trilha dupla por valor - 2 fases (*Dual Rail*) [5].

- b) Um fio é dedicado ao um valor e outro a paridade. Se o fio do valor altera significa que existe um novo dado como o valor alterado. Se o fio do paridade altera significa que o existe um novo dado mas o valor não alterou. Na Figura 37 temos um exemplo onde o fio d.f é dedicado ao valor lógico e d.t a paridade.

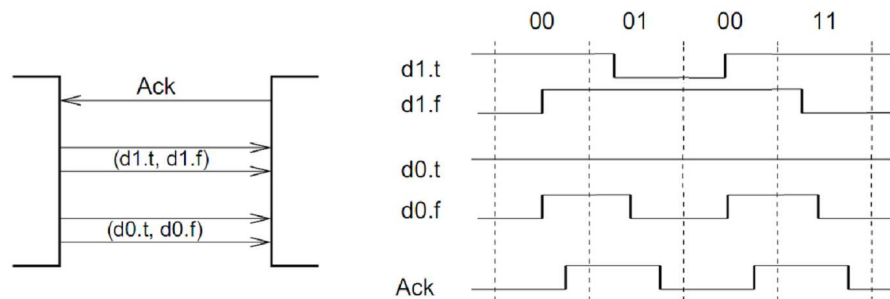


Figura 37: Codificação de trilha dupla por paridade - 2 fases (*Dual Rail*) [5].

## 6.7 Elemento C-Müller

Nos circuitos síncronos o sinal de relógio (*clock*) define o ponto onde os dados estão válidos e estáveis. Entre as transições deste sinal pode existir muitos sinais transitórios que poderão gerar múltiplas transições nos circuitos combinacionais, mas isto não afetará o funcionamento do sistema. No entanto nos circuitos assíncronos esta situação é diferente, pois em qualquer circunstância os sinais devem ser válidos o tempo todo, e qualquer transição é significativa, como consequência os transitórios e corridas devem ser evitados.

Müller [95] introduziu em 1965 um elemento essencial para a implementação dos



circuitos assíncronos. Chamado de *C-Element*, com o objetivo de obter um elemento que realiza-se a função semelhante ao E lógico (AND), porém que alteração no valor de saída somente ocorra quando os valores de entradas forem iguais, para evitar os sinais transitórios, e independente do número de entradas. Este elemento tornou-se indispensável para a implementação dos controles dos circuitos assíncronos, como por exemplo na gestão das requisições e reconhecimentos, e na detecção do barramento de dados com informação válidas. Na Figura 38 é apresentado a sua representação simbólica e a respectiva tabela verdade.

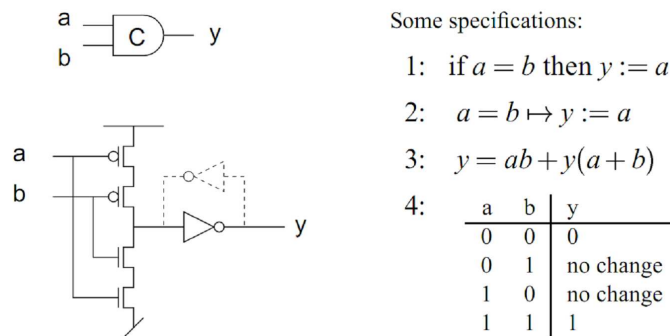


Figura 38: Elemento C-Müller [5].

## 6.8 Técnica de Dessincronização

Vários circuitos assíncronos foram implementados nos últimos anos, como AMULET [96], ASPRO [97], ASPIDA [8], MiniMIPS [98], bem como técnicas que permitem sintetizar automaticamente circuitos assíncronos, juntamente com e ferramentas de desenvolvimento dedicadas (Computer-Aided Design-CAD), como CAST [99], TABGRAM [100], BALSÁ [101] e TAST [102]. Porém, este tipo de circuitos não usados em larga escala, devido ao fato que não existirem boas ferramentas de CAD que tratem por completo todas as etapas do projeto, e principalmente por que estes circuitos exigem que os projetistas mudem suas mentalidades [8], em relação ao paradigma síncrono.

Diante deste fato, foi apresentada uma técnica que transforma um circuito síncrono em assíncrono, utilizando-se de ferramentas de CAD convencionais, chamada de Dessincronização [8] [103]. Esta técnica é um caminho intermediário e simples para obter circuitos assíncronos por derivar pouco do fluxo de projetos síncronos e desta forma permitir a difusão destes circuitos. Este tipo de circuito assíncrono será descrito a seguir por ter sido escolhido como tema principal desta dissertação, os demais modelos não serão descritos neste trabalho, para maiores informações pode ser consultada a bibliografia em referência.

### 6.8.1 Introdução

Dessincronização é uma metodologia para converter um circuito síncrono em um circuito assíncrono equivalente. O sinal de relógio global (*clock*) é trocado por uma rede interligada de controladores de registradores tipo *latch*, mantendo o caminho dos dados intacto. Esta transformação foi apresentada por Cortadella, A. Kondratyev, L. Lavagno e C.P. Sotiriou [8], e tem por premissa preservar a propriedade conhecida como fluxo-equivalente, o que significa que cada elemento sequencial no circuito dessincronizado irá possuir exatamente a mesma sequência de dados do circuito síncrono inicial. Isto permite a aplicação da metodologia e técnicas de teste síncrona. A principal vantagem deste método é que ele pode utilizar as ferramentas normais de EDA (*Electronic Design Automation*), assim o projetista não necessita ter qualquer conhecimento dos circuitos assíncronos. Esta é a única metodologia para concepção de circuitos assíncronos que permite a utilização de bibliotecas e ferramentas padrões a partir de especificação HDL [8] [103].

A etapa da transformação de dessincronização utiliza o circuito síncrono originado na *post-synthesis*, antes de proceder-se a próxima etapa do fluxo, denominada *backend*, ou seja, o projeto físico. Uma visão geral do fluxo dessincronização comparado com o fluxo convencional pode ser visto na Figura 39.

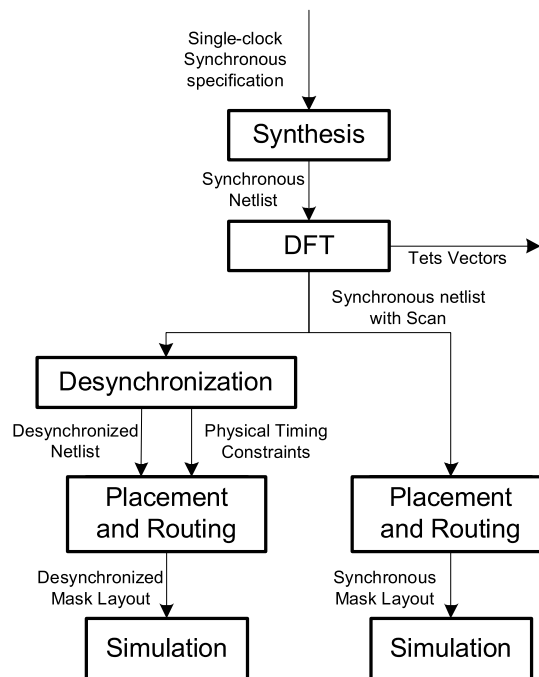


Figura 39: Fluxo comparativo da dessincronização [104].

## 6.8.2 Etapas da Dessincronização

O modelo de Dessincronização apresentado assume que o circuito possui blocos combinacionais (CL) e registradores implementados com *flip-flop* tipo D, e todos trabalham na mesma borda do sinal de relógio (*clock edge*). Este método se procede em três etapas descritas a seguir.

- Conversão do circuito síncrono baseado em *flip-flop* para um baseado em pares de *latches* mestre-escravo.
- Geração dos atrasos (*delays*) para a lógica combinacional.
- Implementação da rede de controle dos registradores e ligação ao circuito original.

Para exemplificar as etapas da dessincronização, descritas a seguir, será utilizado como ponto inicial um exemplo de circuito síncrono, como vemos na Figura 40. As nuvens indicam a lógica combinacional, que acopladas aos registradores *flip-flop*. As linhas tracejadas indicam as regiões do circuito, que contém uma lógica combinacional sempre acompanhada de um registrador para as saídas. As regiões podem ser especificadas pelo projetista quer manualmente ou automaticamente, por uma ferramenta de dessincronização.

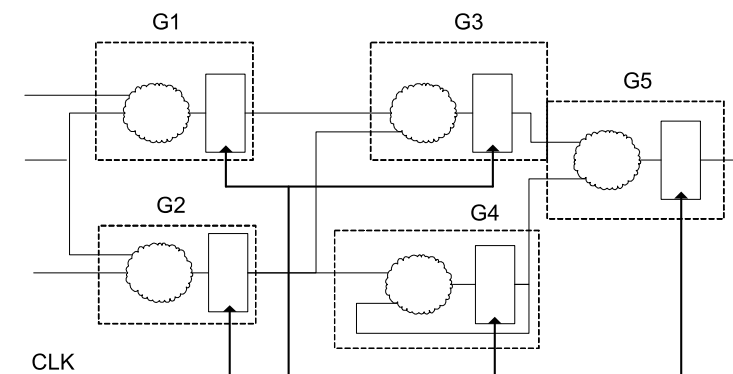


Figura 40: Circuito síncrono inicial [104].

## 6.8.3 Controlador dos *Latches*

O controlador dos *latches* é o principal elemento utilizado na rede de controle, circuito assíncrono com o respectivo protocolo de *handshake* implementado. Podemos ver na Figura 41 um bloco diagrama representativo do controlador. No lado esquerdo da figura, o sinal *ri*, sinal de entrada de requisição, indica que a região predecessora(s) terminou o processamento e possui dados disponíveis na saída, enquanto o sinal *ai*, a saída

de reconhecimento, indica que esta região processou os dados atuais de entrada e estes podem ser substituídos por novos. No lado direito, temos os sinais para o controlador(es) sucessor. Assim, sinal *ro*, sinal de saída de requisição, indica ao controlador da região destino a existência de dados de saída válido, enquanto que o sinal *ao*, sinal de entrada de reconhecimento, indica que a região destino processou esses dados. O sinal *g*, sinal de controle do *latch*, é utilizado habilitar a captura de dados pelo conjunto de latches, enquanto o sinal *rst*, é utilizado para a inicialização.

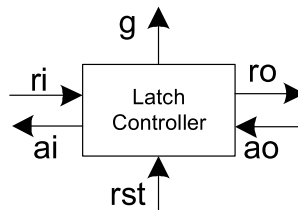


Figura 41: Bloco diagrama do controlador do *latch* [104].

Os controladores podem ter qualquer protocolo handshake para dessincronização, desde que mantenha-se o fluxo-equivalente [103], por exemplo, semi-desacoplado, completamente desacoplado ou outros tipos válidos. Os *Signal Transition Graphs* (STGs) dos protocolos pode ser visto na Figura 42. Os STGs são redes Petri [105], que representam as dependências e sequência dos sinais. Na Figura, os sinais A e B correspondem a dois controles de *latches* em sequência. O protocolo de comunicação pode ser de 2 ou 4 fases. O projeto ASPIDA utiliza controladores semi-desacoplado de 4 fases, por ter sido demonstrado que apresentam uma boa relação custo-benefício (*tradeoff*) entre os sinais concorrentes e a complexidade dos circuitos assíncronos [106]. Os protocolos indicados como *not alive* ou *not-flow-equivalent* na Figura não podem ser utilizados para a dessincronização por apresentarem *deadlocks* e *data overwriting* respectivamente.

#### 6.8.4 Substituição dos *Flip-Flops*

Sendo o projeto dos circuitos síncronos baseado em *flip-flop*, para aplicar a dessincronização, os *flip-flops* devem ser substituídos por *latches*. Os *Flip-flops* do tipo D são conceitualmente compostos por *latch* mestre-escravo (*master-slave*). Para realizar a dessincronização, a sua estrutura interna é explicitamente exposta, e mantendo o comportamento equivalente.

Esta transformação é importante, pois torna o circuito capaz de tolerar escorregamentos (*skews*) em diferentes regiões dos controladores. Um caso hipotético deste problema pode ser visto na Figura 43. Se dois *flip-flops* são controlados por diferentes controladores,

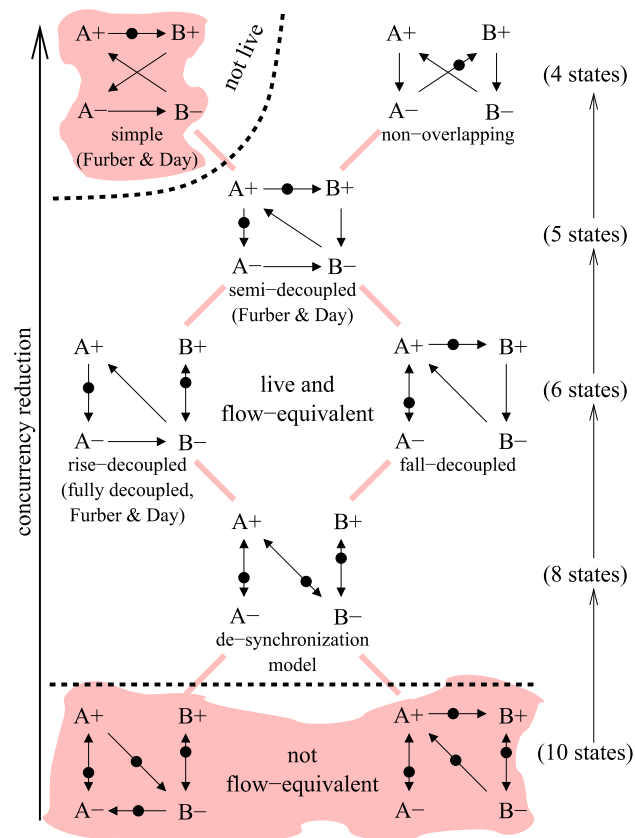


Figura 42: Grafo dos protocolos de *handshake* para dessincronização [104].

o escorregamento entre A e B não pode ser garantido, assim, ocorrerá a sobrescrita de dados.

Além disso, a conversão de um circuito baseado em *flip-flop* em um baseado em *latch* pode ter o desempenho melhorado, mas com o aumento de área. Este tipo conversão não é específica do processo de dessincronização. Este processo é conhecido para melhorar a performance nos sistemas síncronos [107], e por esta razão tem o seu valor por si próprio.

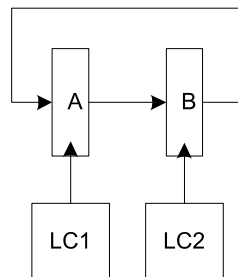


Figura 43: Exemplo de flip-flop onde escorregamento não é permitido [104].

### 6.8.5 Remoção do sinal de relógio (*clock*)

Como mencionado acima, a principal diferença entre um circuito síncrono e o dessincronizado é a substituição da rede do sinal de relógio global por uma rede de interligada de controle de *latches* dos registradores, que gera os sinais para os elementos do circuito dessincronizado. Esta seção descreve como esta rede é composta, como está ligado ao caminho de dados (*datapath*) do circuito síncrono e como os controladores devem ser ligados uns aos outros para que o funcionamento correto do circuito seja assegurado.

#### 6.8.5.1 Grafo da dependência dos dados

A rede de controladores de *latches* deve respeitar as dependências do fluxo de dados entre as várias partes do circuito. Assim, o primeiro passo na análise do circuito a ser dessincronizado deve ser a construção de um grafo que represente a dependência dos dados do circuito. Neste grafo, os nós representam as regiões do circuito e as setas a dependências dos dados. Cada dependência de dados entre duas regiões do circuito, ou seja, uma saída de dados de uma região para uma entrada de dados de outra, é indicado pela direção da seta ligando os dois nós. Figure 44 mostra o grafo correspondente à dependência dos dados do circuito síncrono da Figura 40.

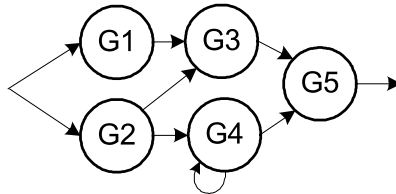


Figura 44: Grafo da dependência dos dados do circuito exemplo [104].

#### 6.8.5.2 Rede de controle

No grafo da dependência dos dados utilizado para construir a rede de controladores, cada região do circuito, representado por um nó no grafo da dependência dos dados, será controlado por um par de controladores de *latches* mestre-escravo. O controlador de *latch* mestre e escravo de um nó devem ser conectados a todos os seus nodos antecessores e sucessores, usando elementos de sincronização (elemento C-Muller). A Figura 45 mostra a rede de controladores resultante. Para os casos em que existem várias requisições de entrada ou de saída de reconhecimento, elementos C-Muller são usados para sincronização, como mostrado na figura 45.

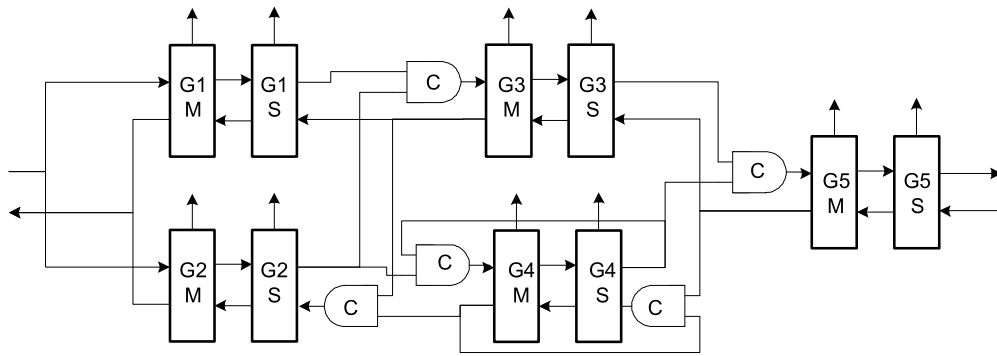


Figura 45: Rede de controle do circuito exemplo [104].

### 6.8.5.3 Atraso da lógica combinacional

O circuito dessincronizado deve respeitar as restrições temporais das sequências de dados do circuito síncrono original. Isto implica que a lógica combinacional de cada região deve ter tempo suficiente para computar os dados. Uma vez que o sinal de requisição indica que a região terminou o processamento e os dados estão válidos, esses sinais devem conter o atraso correspondente ao tempo do caminho crítico. Existem dois métodos possíveis para obter isto, ou utilizando elementos de atraso correspondentes ao tempo do caminho crítico ou modificando a lógica combinacional e incluindo elementos para detecção de conclusão, ou seja, verificação da validade dos dados.

- Elementos de atraso:** Nesta abordagem o sinal de requisição passa por um atraso antes de atingir o controlador destino. Assim, temos um atraso para cada região do circuito. A Figura 46 mostra de modo geral como os elementos de atraso são conectados ao circuito. Os *buffers* são compostos por árvores de *low-skew buffer*. As restrições temporais são satisfeitas se  $delem\_length + CT\_target \leq CT\_source + CL\_delay$ . Quando os *buffers* estão equilibrados ou não existem, tempo de latência zero,  $CT\_target = CT\_source$ , a relação fica reduzida a  $delem\_length \leq CL\_delay$ , portanto teremos o elemento de atraso correspondendo ao tempo do caminho crítico.

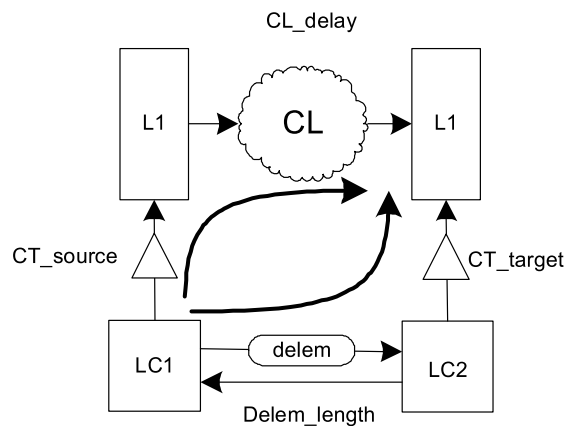


Figura 46: Modelo de lógica com elementos de atraso [104].

Os elementos de atraso utilizados na implementação são simétricos no caso de protocolo de comunicação de 2 fases, ou assimétricos para o protocolo de 4 fases. Os elementos de atraso simétricos têm os tempo de sinal baixo e alto iguais (*rise and fall time*), por sua vez, os elementos de atraso assimétricos têm o tempo de sinal alto bem maior do que o tempo de sinal baixo. Um exemplo de um elemento de atraso assimétrico pode ser visto na Figura 47. Um multiplexador pode ser inserido para configurá-lo após a prototipação. No caso dos elementos de atraso simétricos as portas E são substituídos por buffers ou pares de inversores.

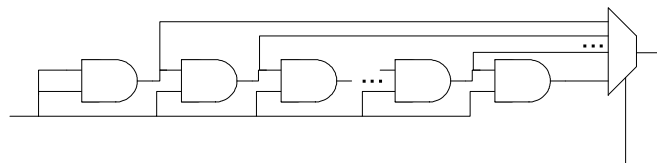


Figura 47: Estrutura de um elemento de atraso assimétrico [104].

- **Detecção de conclusão:** Uma alternativa aos elementos de atraso é detecção conclusão [108], como exemplificado na Figura 48. Neste caso, a lógica combinacional deve ser modificada e incluído elementos para detecção de conclusão (CD) e validade dos dados. Este sinal pode ser usado como requisição de entrada para um controlador. Esta abordagem tem como a principal vantagem de o circuito funcionar na melhor performance, um vez que os atrasos serão considerados tanto para as variações paramétricas, como da lógica combinacional. Sua principal desvantagem é que esta transformação significativa em aumento de área e consumo (aproximadamente duas vezes). Esta abordagem não foi seguida no projeto ASPIDA.



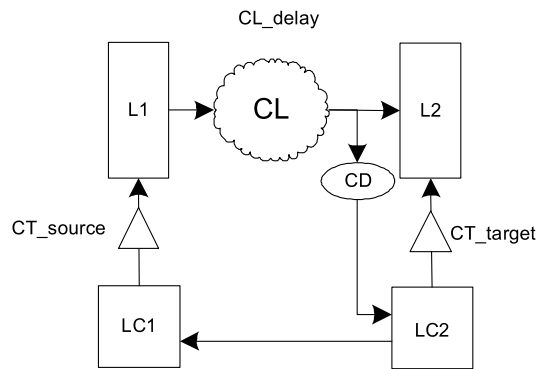


Figura 48: Modelo de lógica com detecção de conclusão [104].

#### 6.8.5.4 Interligação da rede

Na última etapa da dessincronização, a rede de controladores é ligado ao circuito síncrono original. Esta rede constitui-se de pares de controladores, elementos C-Muller e elementos de atraso. Na Figura 49 é mostrado o circuito dessincronizado derivado do circuito síncrono original da Figura 40. As linhas em negrito indicam a rede de controladores. Pode ser visto que cada região do circuito original tem uma lógica combinacional, representado por uma nuvem, um atraso elemento, um par de controladores e elementos C-Muller utilizados para sincronizar as múltiplas requisições de entrada ou sinais de reconhecimentos.

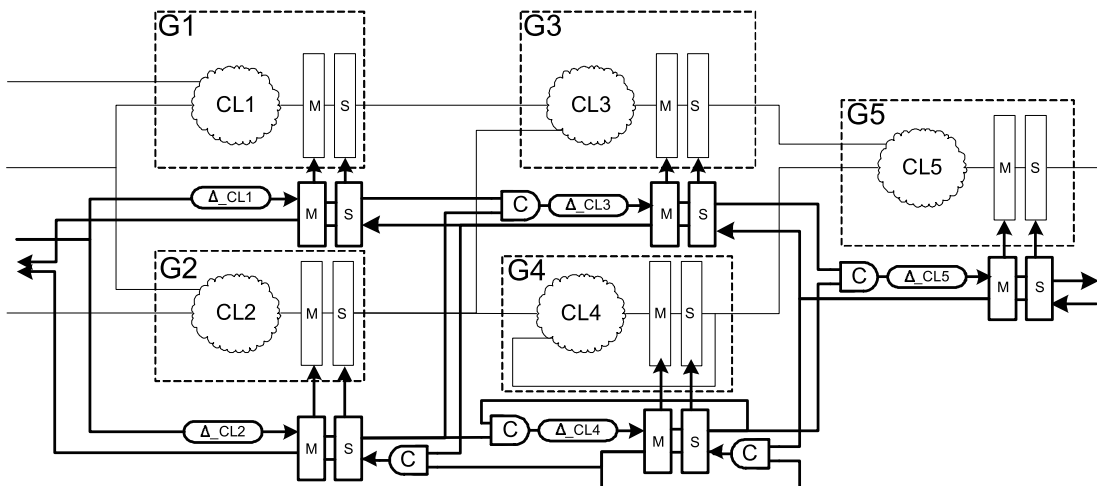


Figura 49: Circuito dessincronizado [104].

### 6.8.6 Resultado da temporização

Os circuitos dessincronizados não possuem sinal de relógio externo, a sincronização é obtida internamente pela rede de controladores de *latches*. Assim, a referência de temporização não é externa, mas resultado da autotemporização da rede interna de controladores do circuito.

Os elementos de atraso utilizados são influenciados pela vários fatores, variação do processo, da tensão de alimentação e da temperatura, os mesmos aos quais é submetida a lógica combinacional. Isto porque eles pertencem ao mesmo circuito e estão no mesmo *chip* e restrições especiais podem ser usados para obtê-los o mais próximo do modelo da lógica combinacional. Os atrasos devem incluir margens para suportar as variações, mas estas margens são mais amplas do que as suportadas pelos circuitos síncronos. Deste modo, as variações do processo e as variações dinâmicas de funcionamento, que afetam o desempenho, são melhor toleradas uma vez que o circuito dessincronizado, de certa forma, ajusta-se automaticamente e dinamicamente [8].

No caso de utilizar a técnica de detecção de conclusão, o circuito tolera não somente as variações intra-chip, mas também apresenta um desempenho melhor, isto devido ao fato de que o funcionamento é dinamicamente afetado pela lógica de dados, assim, o atraso não é sempre igual ao tempo do caminho crítico.

## 7 *Plataforma de ensaios*

### 7.1 Introdução

Embora as plataformas de desenvolvimento dos fabricantes de FPGA (Xilinx [109]) tenham sido importantes para os estudos preliminares para esta dissertação, as mesmas não são adequados aos ensaios que foram realizados. Estes *hardwares* possuem limitações estruturais que impossibilitam algumas abordagens mais complexas relativas a esta dissertação, já que o seu projeto não segue nenhuma norma de ensaio de susceptibilidade a Interferências Eletromagnéticas (EMI), o seu esquemático contém de apenas um FPGA, e principalmente, não existe um controle independente dos níveis de tensão de alimentação desta placa.

Neste sentido, a equipe do Laboratório SiSC <sup>1</sup> desenvolveu uma nova plataforma de ensaios em 2007 e uma placa de alimentação e injeção de falhas para permitir o controle da tensão de alimentação do DUT da plataforma e a respectiva injeção de falhas necessárias aos ensaios [4]. Esta plataforma tornou-se absolutamente indispensável para a execução do trabalho descrito nesta dissertação.

### 7.2 Placa de ensaios

Esta plataforma de ensaios possui características estruturais mais complexas, robustas e funcionais, sendo baseada nas normas de ensaio de susceptibilidade de circuitos integrados a Interferências Eletromagnéticas (EMI) irradiadas e conduzidas IEC 62.132-1 [10], 62.132-2 [40] e 62.132-4 [42], e possui, conforme apresenta a Figura 50, seis camadas (*layers*) cujas características estruturais são apresentadas a seguir.

---

<sup>1</sup>A equipe do Laboratório SiSC é composta por professores e alunos da Faculdade de Engenharia da PUCRS. No projeto desta plataforma de ensaios estiveram envolvidos diretamente os alunos Marlon Moraes e Marcelo Mallmann juntamente com os professores Fabian Luis Vargas e Juliano D’Ornelas Benfica,.

- Camada 1 (*Top*) – Possui os circuitos integrados (CI) sob ensaio da placa, além de um plano de terra (*Gnd*) cobrindo toda área do layer;
- Camada 2 (*Inner 1*) – Possui somente os planos de alimentação (VCC) dos circuitos integrados sob ensaio;
- Camada 3 (*Inner 2*) – Possui todas as trilhas de sinal e/ou alimentação dos componentes e/ou dispositivos da placa;
- Camada 4 (*Inner 3*) – Possui todas as trilhas de sinal e/ou alimentação dos componentes e/ou dispositivos da placa;
- Camada 5 (*Inner 4*) – Possui todas as trilhas de sinal e/ou alimentação dos componentes e/ou dispositivos da placa;
- Camada 6 (*Botton*) – Possui os demais componentes e/ou dispositivos da plataforma, isto é, nesta camada são fixados aqueles componentes e/ou dispositivos que não são sujeitos a EMI durante os ensaios irradiados, além de trilhas de alimentação, sinais, e um plano de terra (*Gnd*) cobrindo toda a área do *layer*.

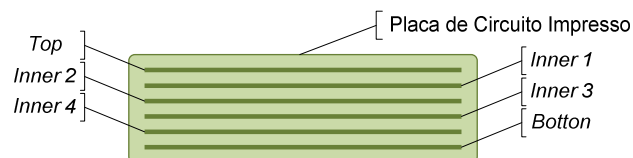


Figura 50: Composição da placa de circuito impresso.

Além destes cuidados relativos ao atendimento das normas de ensaio de susceptibilidade a Interferências Eletromagnéticas (EMI) em circuitos integrados, a plataforma de ensaios, cujo diagrama esquemático genérico é apresentado na Figura 51, dispõe em sua arquitetura dos seguintes componentes:

- 2 FPGAs Xilinx XC3S500E (500k portas, 256 pinos, 360 *Kbits* de *block* RAM interna, 20 multiplicadores e 4 DCMs);
- 1 FPGA Xilinx XC3S200 (200k portas, 144 pinos, 216 *Kbits* de *block* RAM interna, 12 multiplicadores e 2 DCMs);

- 4 memórias SRAM (*Static Random Access Memory*) IS61LV25616AL-10T, produzidas pela ISSI, que formam dois bancos de memória de 1Mbyte com configuração de 256x16 para cada FPGA;
- 2 memórias *Flash* Intel JS28F320J3 32Mbits e tempo de acesso de 110ns;
- 2 microcontroladores (*core* 8051) produzidos pela Texas Instruments;
- 3 osciladores de frequência igual a 49.152MHz (para cada FPGA);
- 2 cristais de frequência igual a 11.0592MHz (para cada microcontrolador);
- Comunicação serial padrão RS-232 (para cada FPGA e microcontrolador);
- 3 reguladores de tensão LM317 para o controle independente dos níveis de tensão de alimentação;
- 1 sensor de temperatura serial 12 bits LM74, produzido pela National Semiconductor;
- 4 botões e 4 LEDs;
- 2 conectores JTAG independentes para programação e depuração dos FPGAs;
- *Jumpers* para seleção e controle independente dos níveis de tensão alimentação;

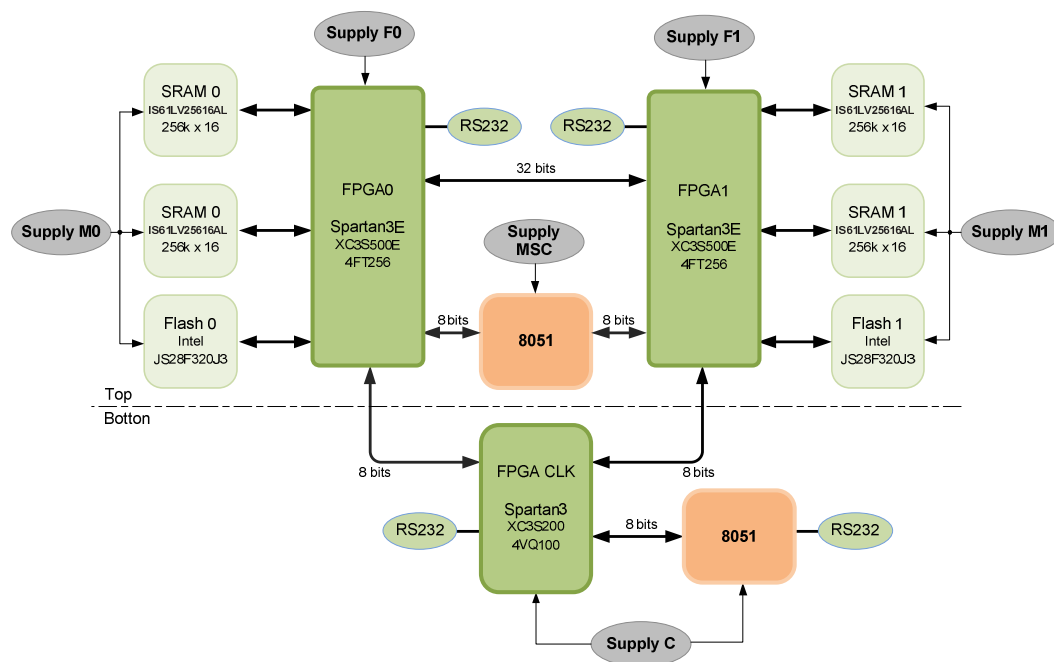


Figura 51: Esquemático genérico da plataforma de ensaios.

A Figura 52 e a Figura 53 apresentam, respectivamente, as vistas inferior e superior da plataforma de ensaios cujos principais componentes são destacados.

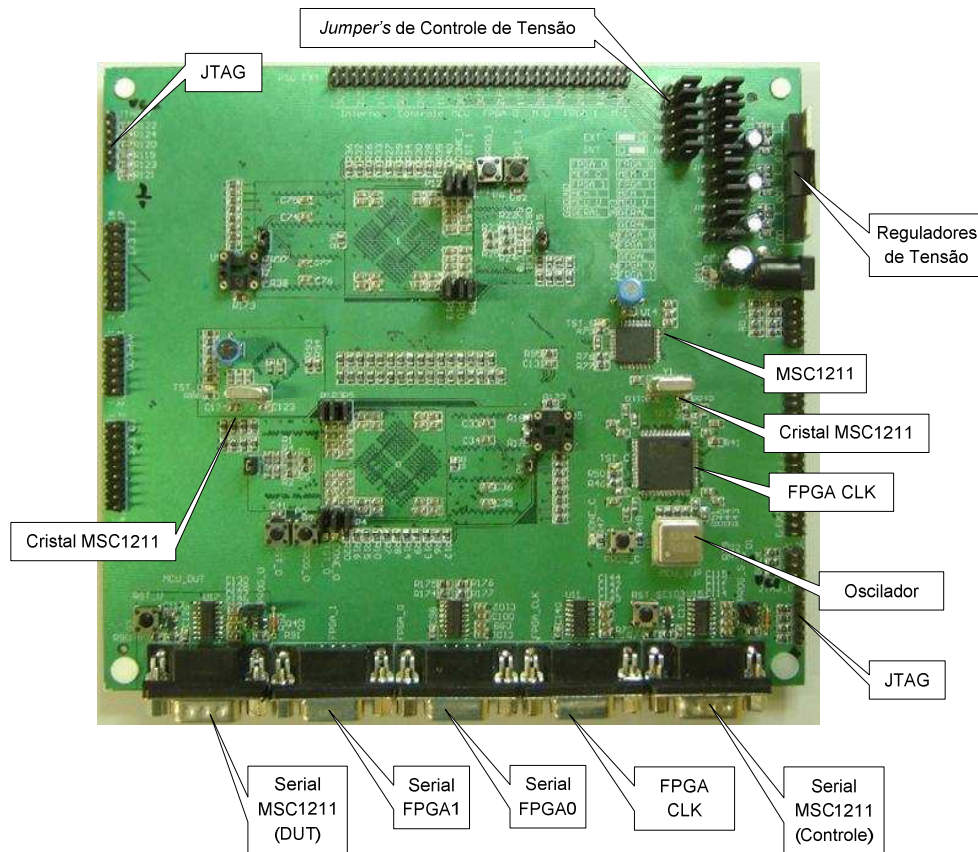


Figura 52: Vista inferior da plataforma de ensaios.

### 7.3 Placa de alimentação e injeção de falhas

Tendo em vista estas características, e na busca pela garantia de qualidade e eficiência dos ensaios dos estudos de caso desta dissertação, fez necessário utilizar uma placa de injeção de falhas de alimentação para os circuitos integrados da plataforma de ensaios (FPGAs).

De acordo com pesquisas e estudos relacionados às características e necessidades deste tipo de injeção de falhas [110] [111], tem-se que as reduções nas linhas de alimentação dos circuitos integrados sob ensaio devem ter um comportamento bastante preciso, com transições rápidas, e principalmente com repetibilidade.

A Figura 54 apresenta um diagrama esquemático genérico do circuito de alimentação e injeção de falhas, utilizado para os ensaios desta dissertação, onde é possível verificar que os níveis de tensão de alimentação e falha, do FPGA sob ensaio (DUT), são gerados

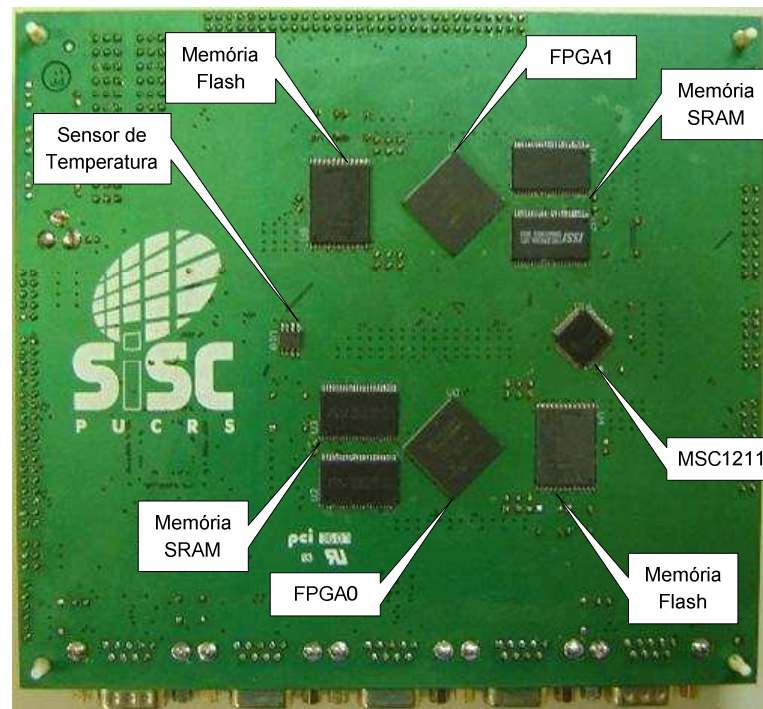


Figura 53: Vista superior da plataforma de ensaios.

pele FPGA CLK associado a conversores digitais-analógicos (*Digital to Analog Converter – DAC*) conectados em série a *buffers* (amplificadores operacionais).

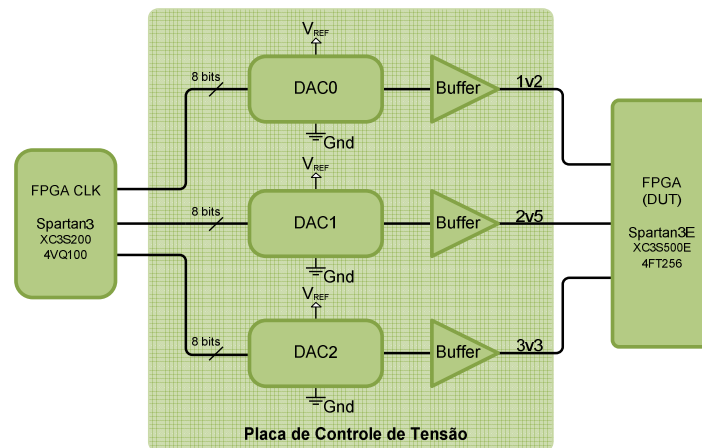


Figura 54: Diagrama da placa de alimentação e injeção de falhas.

A Figura 55 apresenta, respectivamente, a vista superior e inferior da placa de alimentação e injeção de falhas utilizada para os ensaios dos estudos de caso desta dissertação.

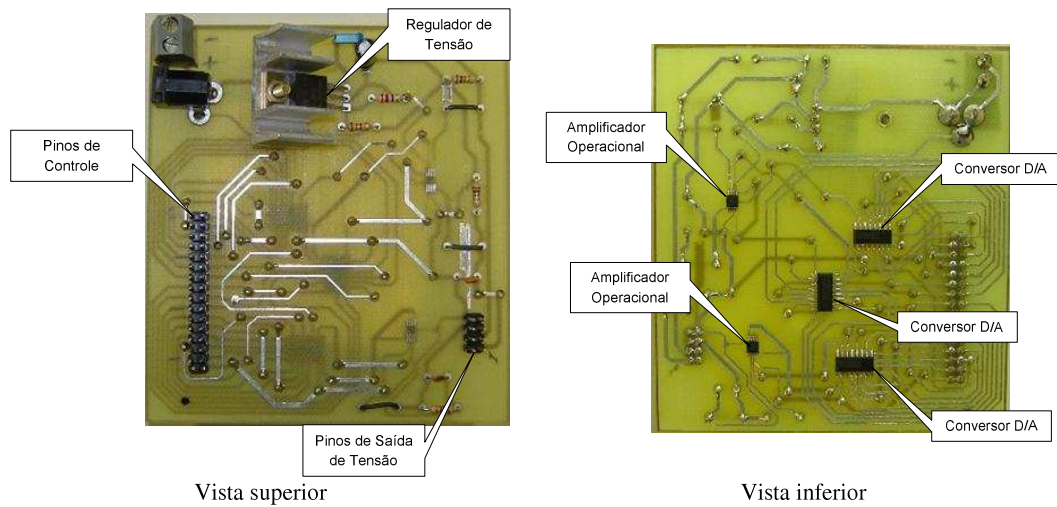


Figura 55: Placa de alimentação e injeção de falhas.

## 7.4 Circuito gerenciador da alimentação do ensaio

O circuito Gerenciador da Alimentação do Ensaio foi desenvolvido com o objetivo de receber através de uma UART, os comandos de controle formados por caracteres simples, sem necessidade de enviar enter (CR/LF) conforme listados na Tabela 11. Além disso, deveria gerar os sinais de controle e dados, apresentados no diagrama temporal da Figura 56, para cada um dos três conversores digitais analógicos da placa de alimentação e injeção de falhas através de uma máquina de estados finitos (*Finite State Machine - FSM*).

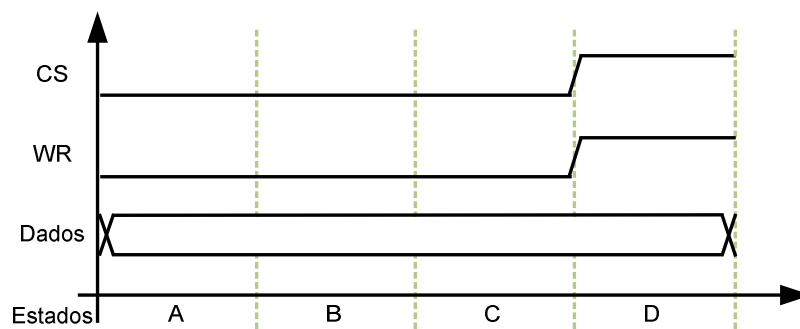


Figura 56: Diagrama temporal de escrita nos conversores digitais-analógicos.

A Tabela 56 apresenta respectivamente a descrição funcional dos pinos de entrada e saída do circuito de Alimentação.

O único comando que possui uma resposta é o comando '0', que retorna o modo de operação da tensão, o canal selecionado e os valores configurados para queda ( $V_{Low}$ ) nos



Tabela 10: Descrição funcional dos pinos do circuito de Alimentação.

Nome	Descrição
clock_in	Pino de entrada do sinal de relógio do circuito (oscilador externo).
reset_in	Pino de <i>reset</i> da máquina de estados finitos do circuito.
cs_x	Pino de seleção do DAC x [0,1 ou 2].
wr_x	Pino de habilitação de escrita no DAC x.
dado_x	Vetor de 8 <i>bits</i> que define a amplitude do sinal de saída do DAC x.
rxd_in	Pino para sinal de entrada serial da UART.
txd_out	Pino para sinal de saída serial da UART.
Fy_clock	Pino para sinal de clock da FPGA y [0 ou 1].
Fy_reset	Pino para <i>reset</i> da FPGA y.

canais 1, 2 e 3. Sendo composta por 10 *bytes* em ASCII com o formato xMCV3V2V1, descrito na Tabela 12. Os valores de tensão são retornados em base hexadecimal.

Quando a placa é inicializada, o modo de tensão nominal é selecionado, sendo valores das tensões de saída são ajustados para os valores nominais, sem nenhum canal de tensão selecionado. E O valor da largura dos pulsos para as quedas de tensão pulsadas esta configurados no VHDL, bem como o *duty-cycle*, como podemos verificar na Figura 57.

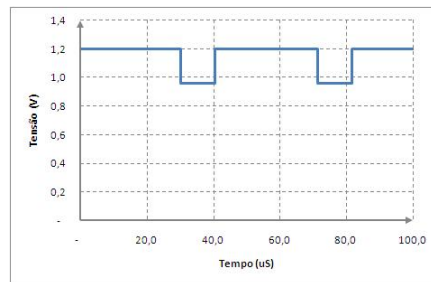


Figura 57: Forma de onda da queda de tensão pulsada.

## 7.5 Conclusão

Esta plataforma, composta de uma placa de ensaios, uma placa de alimentação e injeção de falhas e o circuito gerenciador da alimentação do ensaio, descritos neste capítulo, foi utilizada para a realização dos ensaios de Interferência Eletromagnética Conduzida, redução e quedas de tensão de alimentação, e ensaios de Interferência Eletromagnética Irradiada, em ambos os estudos de casos. A metodologia adotada e os respectivos resultados são apresentados nos capítulos que se segue.

Tabela 11: Descrição dos comandos de controle do circuito de Alimentação.

Comando	Descrição
0	Retorna o valor dos ajustes internos
1	Configura o canal 1 (tensão nominal 1,2V) para ajuste de VLow
2	Configura o canal 2 (tensão nominal 2,5V) para ajuste de VLow
3	Configura o canal 3 (tensão nominal 3,3V) para ajuste de VLow
+	Incrementa o valor do canal selecionado
-	Decrementa o valor do canal selecionado
V ou v	Configura o modo de tensão para nominal
Q ou q	Configura o modo de tensão para o valor ajuste de VLow ajustado
P ou p	Configura o modo de tensão para pulsado entre o valor nominal e o valor de ajuste de VLow ajustado.
Z ou z	Configura o modo de tensão para pulsado entre o valor nominal e o zero.
R ou r	Comando de <i>reset</i> para os FPGA 0 e 1.

Tabela 12: Resposta ao comandos de controle '0'.

Byte	Descrição
x	Indica inicio da resposta
M	Modo de operação do controle de tensão: 0 - Indica modo de tensão nominal 1 - Indica modo de tensão com valor VLow 2 - Indica modo pulsado com valor nominal e VLow 3 - Indica modo pulsado com valor nominal e zero
C	Canal selecionado para ajustes: 0 - Nenhum canal selecionado 1 - Canal 1 (tensão 1,2V) selecionado para a ajuste de VLow 2 - Canal 2 (tensão 2,5V) selecionado para a ajuste de VLow 3 - Canal 3 (tensão 3,3V) selecionado para a ajuste de VLow
V2	Valor ajustado para VLow no canal 3 (tensão nominal 3,3V)
V2	Valor ajustado para VLow no canal 2 (tensão nominal 2,5V)
V1	Valor ajustado para VLow no canal 1 (tensão nominal 1,2V)

## 8 *Estudo de Caso I*

### 8.1 Introdução

A partir da revisão bibliográfica referenciada nos capítulos anteriores, decidiu-se pela seleção de um circuito simples para ser implementado tanto no paradigma assíncrono como síncrono, com o intuito de iniciar as verificações e ensaios relativos a robustez dos circuitos assíncronos com relação aos circuitos síncronos, foco principal desta dissertação. Conseqüentemente, optamos por um multiplicador de 8 *bits* como primeiro estudo de caso.

Para a implementação dos circuitos para verificação de sua robustez quando sujeitos a flutuações no barramento de alimentação, optou-se em utilizar FPGAs do tipo SRAM, por ser uma alternativa bastante viável sob o ponto de vista técnico e econômico, sendo sua metodologia apresentada a seguir.

### 8.2 Multiplicador com somadores

O circuito a utilizado foi um Multiplicador de 8 *bits* em *hardware*, conforme Figura 58, onde é ilustrada uma versão de multiplicador de 4 *bits*, ilustrando os geradores de produtos (linhas de portas AND de duas entradas) e os somadores em cascata (linhas com meio somadores e somadores completos, *half adder* - HA, *full adder* - FA). Este circuito foi desenvolvido na versão assíncrona baseado em portas do tipo *Delay Insensitive Minterm Synthesis* (DIMS) [112] e prototipados primeiramente em FPGAs da Xilinx (*Spartan-3 Starter Kit* - XC3S200 [109]), para verificação funcional, e depois prototipado na plataforma de ensaios do SiSC (7.2). Para facilitar a prototipação, foram usadas estruturas do tipo *Relationally Placed Macros* (RPMs) para implementar as portas DIMS, usando ferramentas de suporte ao projeto fornecidas no ambiente de desenvolvimento ISE da Xilinx.

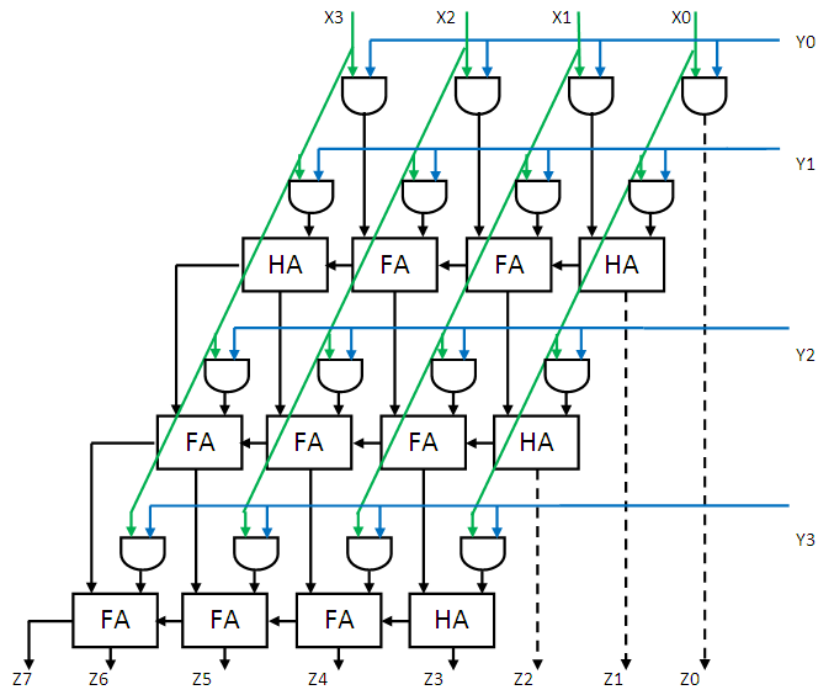


Figura 58: Circuito do multiplicador.

### 8.3 Modificações estruturais no multiplicador

O multiplicador foi modificado para ter o seu desempenho aumentado, bem como a sua complexidade, para isto foi introduzido um *pipeline*, alterando a arquitetura base do multiplicador de 8 *bits*. Esta modificação consiste na inserção de registradores separadores de estágio após cada linha de somadores do multiplicador, sendo ainda realizada a adequação do processo de *handshaking* entre os estágios do *pipeline* (Figura 59), no caso do circuito assíncrono. A seguir temos em detalhes as modificações realizadas para o propósito acima descrito:

- Inseridos registradores, que tem como função armazenar os valores parciais obtidos nos somadores, e os operandos, em cada estágio do multiplicador;
- Inseridos detectores de validade dos sinais oriundos das saídas dos somadores de cada estágio, com o objetivo de realizar o *handshaking* com o estágio anterior;
- A partir do segundo estágio do *pipeline*, os operandos são obtidos dos valores armazenados nos registradores do estágio anterior;
- O valor LSB (*Least Significant Byte*) do produto é gerado parcialmente em cada estágio e armazenado, sendo este propagado para os estágios seguintes;

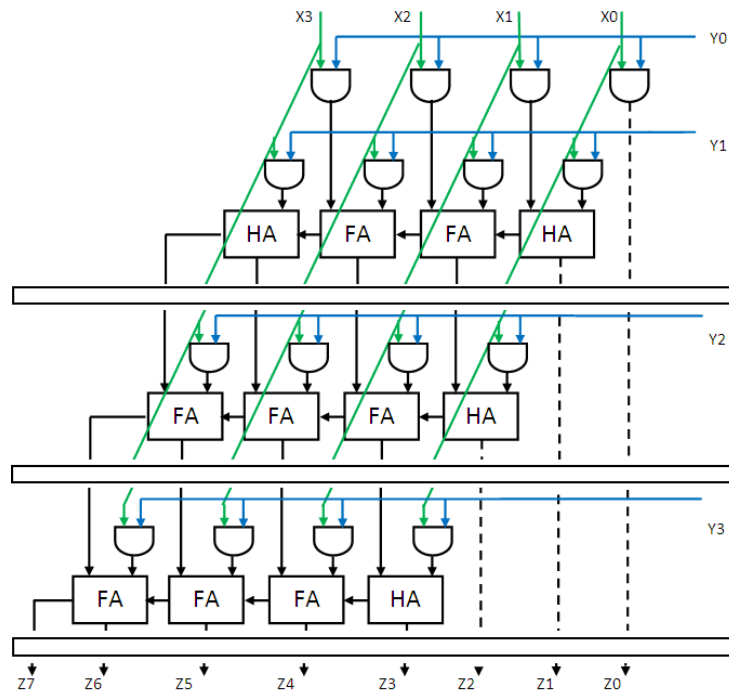


Figura 59: Circuito do multiplicador modificado.

## 8.4 Sistema de gerenciamento de ensaios

Com o intuito de realizar o gerenciamento dos ensaios foi implementada como um *pipeline* assíncrono de 3 estágios (contendo dois registradores separadores de estágios - Figura 60). O primeiro estágio contém um produtor de dados que se comunica com o multiplicador, que constitui o segundo estágio. O terceiro estágio contém um consumidor de dados. Toda a comunicação entre estágios se dá via *handshaking* assíncrono do tipo *dual-rail*, 4 fases. O Produtor e Consumidor operam com protocolos *single-rail*, *bundled-data*. Entre Produtor e o Multiplicador existe um conversor *single-to-dual* e entre o Multiplicador e o Consumidor existe um conversor *dual-to-single*.

O circuito consumidor e produtor são compostos por um *Built-In Logic Block Observer* (BILBO) [113], onde tem a função de *Linear Feedback Shift Register* (LFSR) para gerar os valores de entradas pseudo-aleatórios e *Multiple-Input Shift-Registers* (MISR) para realizar a assinatura dos resultados, respectivamente. Sendo que a versão do circuito síncrono foi obtido a partir do circuito assíncrono, onde foi substituído o protocolo de comunicação (*handshake*), pelo sinal de relógio global (*clock*), para que o comportamento de ambos os circuitos sejam o mais próximo possível, permitindo assim uma comparação com maior grau de confiabilidade.

Com o intuito de controlar o sistema sob ensaio foi incluído ao Circuito Gerenciador

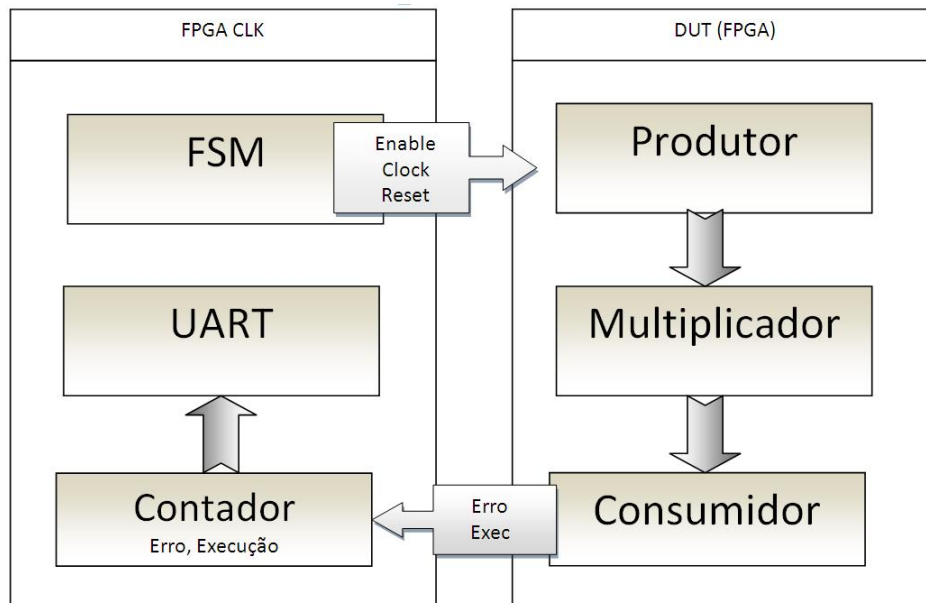


Figura 60: Circuito de ensaio do multiplicador.

da Alimentação do Ensaio (7.4), foram incluídos uma máquina de estados finitos (*Finite State Machine* - FSM) com a finalidade de gerar os sinais para controle dos BILBOs, além de contadores para realizar contagem do número de execuções e erros detectados pela comparação da assinatura do MISR, bem como comandos para enviar pela UART os respectivos valores.

## 8.5 Ensaio realizados

Para proceder o ensaio com a injeção de ruído conduzido através da flutuação da tensão no barramento de VDD segundo parâmetros especificados em norma IEC 61.004-29 [12], foi utilizada a plataforma de ensaio projetada e desenvolvida pela equipe do Laboratório SiSC - PUCRS (Sistemas, Sinais e Computação), vide detalhes na seção 7.2, concebida de acordo com as normas de ensaio de susceptibilidade de circuitos integrados a Interferências Eletromagnéticas (EMI) irradiadas e conduzidas IEC 61.004-29 [12].

Os FPGAs da plataforma de ensaios pertencem a Família Spartan-3 [72], os quais necessitam de três níveis de tensão de alimentação, sendo as tensões com amplitude igual a 2,5 e 3,3 volts destinada às lógicas de entrada e saída do dispositivo (periferia), e o núcleo do dispositivo (*core*) alimentado com tensão de amplitude igual a 1,2 volts [109]. Como o circuito avaliado esta no núcleo do dispositivo, e não depende das entradas e saídas do dispositivo, e não desejávamos termos falhas indesejáveis, foi aplicado a injeção de queda de tensão somente na tensão de alimentação do núcleo. Esta queda de tensão teve seus

níveis reduzidos de forma manual, gradual e progressiva utilizando-se o o circuito descrito na seção 7.4 do capítulo anterior, onde o valor inicial para a queda foi fixado em 0,96V, que corresponde a um redução percentual de 20%, e a partir desta inseriu-se comandos de incremento ou decremento corresponde a 19,5mV de acréscimo ou decréscimo da queda de tensão aplicada. Também o tempo da queda aplicada e o tempo de repetição foi fixado após ensaios experimentais preliminares, onde se verificou que se obtinha melhor resultados com o os valores conforme apresentados na figura 57, seguindo a norma IEC 61.004-29 [12].

## 8.6 Conclusão

Com base nos resultados experimentais preliminares apresentados anteriormente, podemos obter algumas conclusões, sobre os circuitos assíncronos e síncronos, o comportamento e as características de robustez às reduções tensão de alimentação são apresentadas a seguir:

1. Nestes experimentos preliminares a plataforma de ensaios e os seus componentes se mostraram bastante robustos, apresentando somente falhas funcionais quando expostos a queda nos níveis de tensão da ordem de 50% no núcleo do FPGA.
2. O Multiplicador na versão assíncrona apresenta um aumento na área utilizada da FPGA (*overhead*) de 185% em relação a versão síncrona, como pode ser visto na Tabela 13, isto se deve ao fato deste utilizar a codificação de dados com dois fios (*dual-rail*) e ao circuitos necessários para implementar o controle do fluxo de dados (*handshaking*).

Tabela 13: Multiplicador: área utilizada na FPGA (*overhead*).

Circuito	Tamanho
Assíncrono	2.105 LUTs
Síncrono	737 LUTs

3. O Multiplicador na versão assíncrona apresenta uma robustez maior em relação a versão síncrona, quando expostos a queda nos níveis de tensão no núcleo do FPGA, como pode ser visto na Tabela 14.
4. Ambos os Multiplicador, na versão assíncrona e síncrona, foram fundamentais para a verificação preliminar da robustez destes, entretanto sua funcionalidade e complexi-

Tabela 14: Multiplicador: ensaio de queda de tensão.

Circuito	Falhas %	
	Síncrono	Assíncrono
Frequência		
50,0 MHz	8,5%	7,2%
25,0 MHz	6,8%	5,0%
12,5 MHz	4,8%	0,6%

bilidade é muito restrita. Neste sentido concluiu-se que a escolha de um circuito mais complexo, tal qual um processador, é extremamente necessária para o desenvolvimento desta dissertação.



## 9 *Estudo de Caso II*

### 9.1 Introdução

Com o objetivo de analisar a robustez de circuitos implementados de forma síncronas e assíncronas mais complexos e com uma funcionalidade o mais próxima possível dos processadores utilizados atualmente, foi realizada uma procura para encontrar um processador que possuísse uma versão síncrona e outra assíncrona, com prototipação em FPGA, disponível para utilização a nível académico, com fontes abertos (*open core*). Neste sentido foi selecionado o processador DLX utilizado no projeto ASPIDA (ASynchronous oPen-source Ip of the Dlx Architecture) [8].

Este processador apresenta uma motivação extra para sua escolha devido ao fato de sua implementação ser recente, fazendo o uso de uma técnica que é um caminho intermediário e simples para obter circuitos assíncronos derivando pouco do fluxo de projetos síncronos. Nele utilizam-se ferramentas de CAD convencionais, permitindo transformar um circuito síncrono em assíncrono, desta forma contribuindo para a difusão dos circuitos assíncronos [8] [103].

### 9.2 Projeto ASPIDA

O projeto ASPIDA [114], sediado na Europa, teve por objetivos demonstrar a viabilidade industrial e o potencial de reutilização IP assíncronos, gratuitos, com código fonte aberto e qualidade industrial, projetados, testados e implementados com ferramentas EDA industriais, que incentivação as especificações de IP reutilizáveis abertos (*Open IP Reuse specifications*), desta maneira pode ser facilmente utilizadas em arquiteturas que adotem o *Open IP System-On-a-Chip*. A arquitetura implementa o conjunto de instruções DLX, uma arquitetura RISC (Reduced Instruction Set Computer) genérica bem conhecida e utilizada para fins educativos e com objetivos de pesquisa. Inspirado no êxito do *software* de fonte aberta, este projeto pretende promover a utilização de *Open IP Reuse, hardware*

de fonte aberta.

No projeto ASPIDA foi empreendido um esforço conjunto de FORTH-ICS, Politécnico de Torino, Manchester University e IHP Microelectronics [114].

O processador ASPIDA DLX implementa um IP assíncrono da Arquitetura de Conjunto de instruções DLX (ISA) que incorpora o suporte para conversão para ISA (*Instruction Set Architecture*), portanto pode ser facilmente convertido em qualquer ISA RISC. O fluxo de projeto baseado em ferramentas EDA existentes foi adotado em todos os passos para produzir um *netlist* portátil e distribuir todos os arquivos de HDL intermediários usados para projetos de alto nível e desenho de nível da porta. O produto final é independente da tecnologia e independente da regulação de tempo e em uma forma conveniente para a integração usando somente padrões, ferramentas e fluxos industriais convencionais, sem dependência de ferramentas assíncronas e conhecimento específico de projeto assíncrono. O processador DLX assíncrono foi implementado com sucesso tanto em ASIC como em versões para FPGA [115].

### 9.3 Processador DLX

O DLX é processador RISC, de 32 *bits* e com *pipeline* de 5 estágios, descrito no livro de Hennessy and Patterson's, *Computer Architecture: A Quantitative Approach* [116]. Amplamente conhecido, e a sua arquitetura de *pipeline* se assemelha com outros processadores RISC, tais como MIPS [117] e ARM [67]. Um diagrama detalhado de seus estágios do *pipeline* é mostrado na Figura 61.

O processador DLX tem a estrutura dos estágios de *pipeline* como segue:

1. *Instruction Fetch*(IF): a próxima instrução a ser executada é lida da memória de instruções do endereço apontado pelo registrador *Program Counter*(PC). O próximo endereço é calculado adicionando 4 ao valor do endereço corrente, caso não existe uma instrução de salto para ser executada.
2. *Instruction Decode*(ID): a instrução procedente do estágio IF é decodificada e os registradores são acessados. Saltos condicionais também são avaliados neste estágio e saltos relativos ou absolutos são calculados e enviados de volta ao estágio IF.
3. *Execute* (EX): as operações de lógicas e aritméticas são realizadas com os operandos procedentes dos estágio ID. O resultado destas operações podem ser escritos nos registradores ou em endereços de memória de dados acessados no estágio seguinte.

4. *Memory*(MEM): a memória de dados é acessada (leitura ou escrita).
5. *Write Back*(WB): o valor a ser escrito nos registradores (se algum) é enviado para o estágio ID.

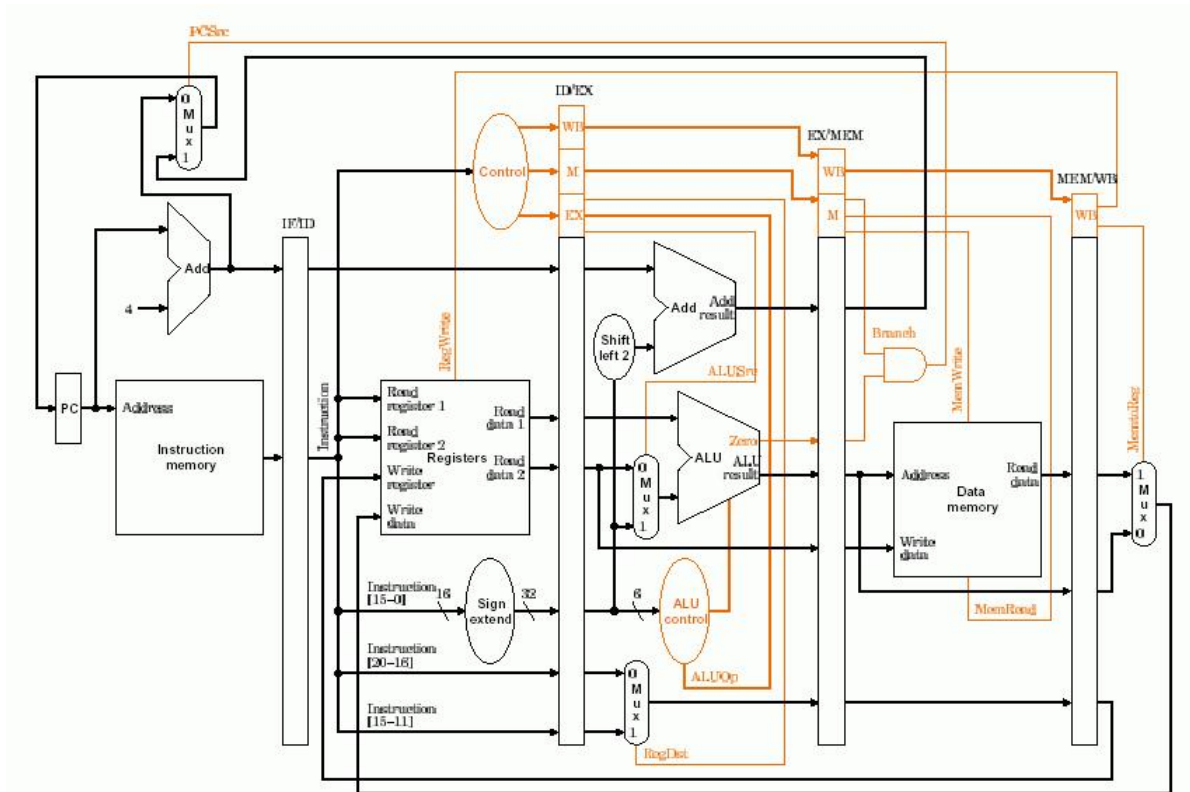


Figura 61: Processador DLX [116].

## 9.4 Processador DLX ASPIDA

O processador ASPIDA DLX (*open source*) tem suporte total para dados do tipo inteiro DLX ISA. As operações de ponto flutuante não são suportadas na versão atual do processador. O ASPIDA DLX contém duas interfaces de memória, de acordo com o modelo DLX original, que tem suporte para *byte* (8 bits - *byte*), *meia-palavra* (16 bits - *half word*) e *palavra inteira* (32 bits - *full word*). Os saltos (*branch*) seguem a semântica RISC convencional e necessitam de atraso, isto é a instrução seguida pelo salto sempre é realizada. Também possui um coprocessador de vetores interrupção, incluindo um registro de causa de interrupção e um registrador de exceção.

O ASPIDA DLX possui os três tipos de operação de DLX ISA:

1. *I-type*: operações de lógicas/aritméticas executadas entre um registro e um valor imediato. Os saltos condicionais também são instruções de I-type;
2. *R-type*: operações de lógicas/aritméticas executadas entre dois registros. As instruções de leitura e escrita também são instruções de R-type;
3. *J-type*: instruções de salto absoluto ou relativo.

O *layout* de cada tipo de instrução pode ser visto na Figura 62.

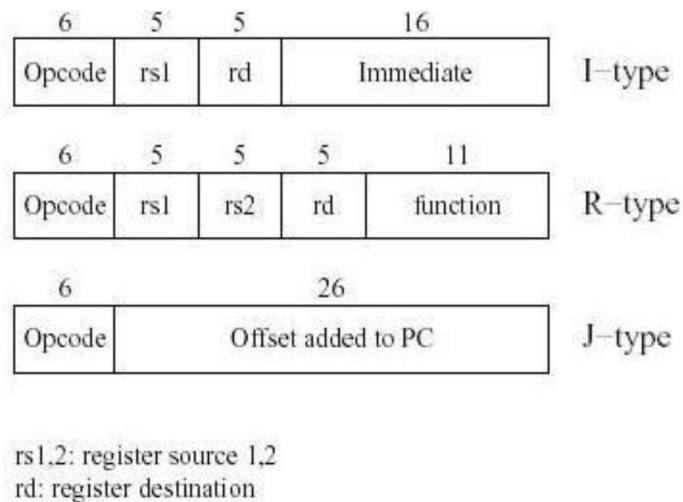


Figura 62: Processador DLX: leiaute das instruções [116].

O subconjunto de instruções DLX ISA é mostrado nas Figura 63, 64 e 65. As instruções suportadas estão assinaladas com uma marca (✓).

IR[2-0]	IR[5-3]							
	000	001	010	011	100	101	110	111
000	(rr alu)	(float)	j ✓	jal ✓	beqz ✓	bnez ✓	bftpt	bftpf
001	addi ✓	addui ✓	subi ✓	subui	andi ✓	ori ✓	xori ✓	lhi ✓
010	rfe ✓	trap ✓	jr ✓	jalr ✓	slli ✓	-	srli ✓	srai ✓
011	seqi ✓	snei ✓	slti ✓	sgti ✓	slei ✓	sgei ✓	-	-
100	lb ✓	lh ✓	-	lw ✓	lbu ✓	lhu ✓	lf	ld
101	sb ✓	sh ✓	-	sw ✓	-	-	sf	sd
110	-	-	-	-	-	-	-	-
111	-	-	-	-	-	-	-	-

Figura 63: Processador DLX: instruções - subconjunto 1 [116].

IR[28-26]	IR[31-29]							
	000	001	010	011	100	101	110	111
000	-	-	-	-	sll ✓	-	srl ✓	sra ✓
001	-	-	-	-	-	-	-	-
010	-	-	-	-	-	-	-	-
011	-	-	-	-	-	-	-	-
100	add ✓	addu ✓	sub ✓	subu ✓	and ✓	or ✓	xor ✓	-
101	seq ✓	sne ✓	slt ✓	sgt ✓	sle ✓	sge ✓	-	-
110	movi2s	movs2i	movf	movd	movfp2i	movi2fp	-	-
111	-	-	-	-	-	-	-	-

Figura 64: Processador DLX: instruções - subconjunto 2 [116].

IR[28-26]	IR[31-29]							
	000	001	010	011	100	101	110	111
000	addf	subf	multf	divf	addd	subd	multd	divd
001	cvtf2d	cvtf2i	cvtd2f	cvtd2i	cvti2f	cvti2d	mult	div
010	eqf	nef	ltf	gtf	lef	gef	multu	divu
011	eqd	ned	ltd	gtd	led	ged	-	-
100	-	-	-	-	-	-	-	-
101	-	-	-	-	-	-	-	-
110	-	-	-	-	-	-	-	-
111	-	-	-	-	-	-	-	-

Figura 65: Processador DLX: instruções - subconjunto 3 [116].

## 9.5 Processador DLX assíncrono

Na operação de dessincronizar do DLX, o sinal de relógio global (*clock*) é retirado e é substituído por controladores de *handshaking*. Os *flip-flops* são substituídos por pares de *latches*. Na Figura 66 podemos ver DLX dessincronizado com o respectivo caminho de dados (*datapath*). Como pode ser visto, os *latches* que separam os estágios do caminho de dados são localmente controlados, e são responsáveis por produzir os sinais apropriados para que os dados se movam seguramente de um estágio do *pipeline* ao estágio seguinte.

## 9.6 Implementação DLX assíncrono na FPGA

Os controladores de dessincronização, usados na versão FPGA, são compostos de uma parte mestre e uma parte escravo. São controladores semidescoplados, com protocolo de comunicação de 4 fase. Este circuito foi concebido a partir de uma especificação *Signal*

*Transition Graph* (STG) na ferramenta *petrify* [105], conforme descritos na seção 6.8.5.3. O controlador de dessincronização completo de DLX inclui as partes escravo e mestre juntamente aos *pulse delay* e *matched delay*. Os *pulse delay* são cadeias de inversores, onde foram utilizadas as restrições KEEP que são essenciais, pois de outra maneira os inversores seriam otimizados pelo software de prototipação ISE/Xilinx. Os *matched delay* são cadeias de portas AND, onde foram utilizados os atributos de síntese, os quais são essenciais para que a cadeia de portas, que implementam os atrasos assimétricos, não sejam otimizadas pela ferramenta de prototipação.

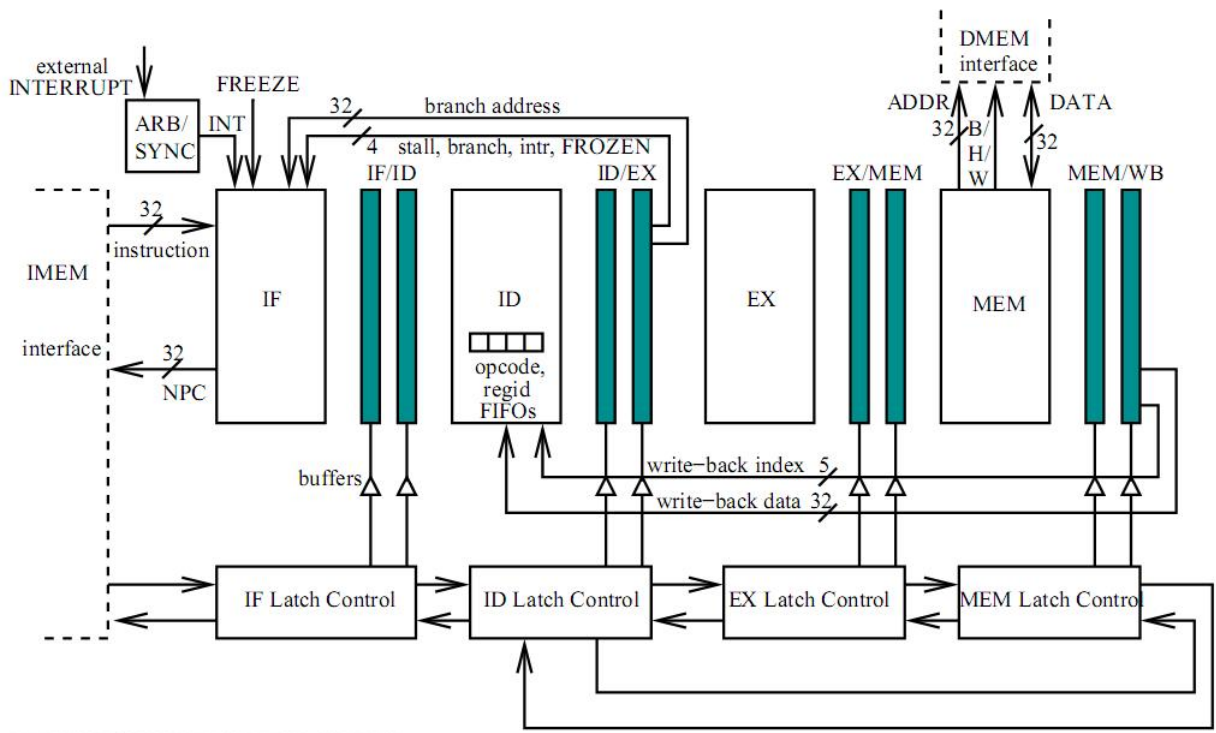


Figura 66: Processador DLX assíncrono [8].

## 9.7 Sistema de gerenciamento de ensaios

O sistema de demonstração do projeto ASPIDA DLX, disponibilizado na página do *OpenCore* na *internet* [115], consiste em uma interface para memória de instruções e dados para a CPU e uma para VRAM, e uma interface para um controlador de vídeo do tipo VGA, como podemos ver na Figura 67. Tendo o mapa de memória conforme a Tabela 15.

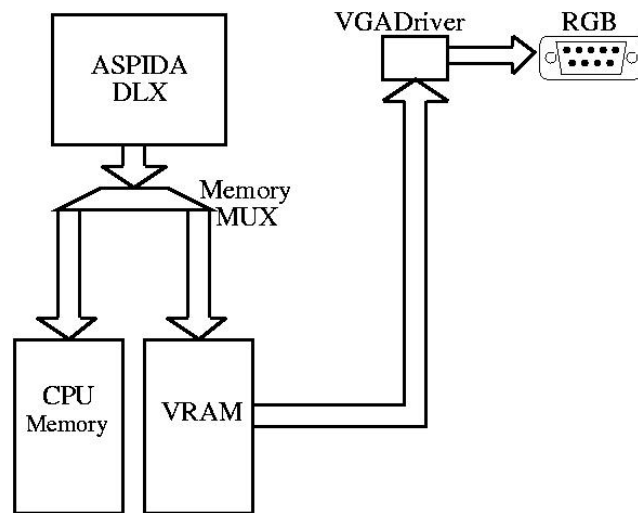


Figura 67: Processador ASPIDA DLX: sistema de demonstração [115].

[8]

Tabela 15: Processador ASPIDA DLX: mapa de memória do sistema de demonstração.

Posição	Descrição
0x0000	<i>STACK</i>
	<i>Instruction Memory</i>
	<i>Data Memory</i>
0x1000	<i>Video Memory</i>

Como este sistema não possuía interfaces que possibilitasse o registro dos eventos ocorridos com o respectivo envio de mensagens para o meio exterior, fez-se necessário alterá-la colocando-se no lugar da interface de vídeo uma interface do tipo UART, e como a memória de dados e instruções era pequena, permitindo apenas a execução de programas pequenos com no máximo 512 *bytes*, alterou-se também o tamanho da memória disponível, trocando-se os quatro componentes RAMB4\_S8\_S8 (*B*lock RAM de 4 *Kbits*, com acesso por *byte*) por RAMB16\_S9\_S9 (*B*lock RAM 16 *Kbits*, com acesso por *byte*), aumentando assim o tamanho da memória para 2 *Kbytes*. Desta forma o sistema para ensaios ficou com a estrutura conforme a Figura 68 e o mapa de memória conforme a Tabela 16.

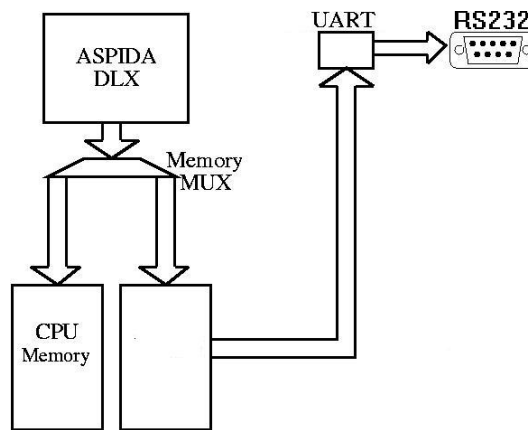


Figura 68: Processador ASPIDA DLX: sistema de ensaio.

Tabela 16: Processador ASPIDA DLX: mapa de memória do sistema de ensaio.

Posição	Descrição
0x0000	<i>STACK</i>
	<i>Instruction Memory</i>
	<i>Data Memory</i>
0x2000	<i>UART Memory</i>

Na Figura 17 é mostrado o resultado comparativo da *Post-Synthesis*, referente a área ocupada pelos elementos do processador DLX e o número total de portas equivalente<sup>1</sup>, síncrono e assíncrono, implementados na FPGA Xilinx, utilizando a ferramenta de síntese ISE 9.2i, e na Figura 18 apresenta-se o atraso mínimo e a frequência máxima dos respectivos circuitos. Salientamos que nos estudos do mestrado de Andrikos Nikolaos [104] para a implementação com a tecnologia ST CORE9 90nm, utilizando o fluxo de projeto da Synopsys-Avanti STMicroelectronics, obteve como resultado um aumento de área total de 13,5% e de consumo de 22%, comparando-se ao circuito síncrono. Por sua vez, no projeto ASPIDA, na prototipação de ASIC usando a tecnologia UMC 0.18- $\mu$ m CMOS, os resultados apontam uma pequeno aumento de área e consumo, sendo 1,5% em área e 0,4 em consumo, comparando-se ao circuito síncrono [8]. Fato este que devido as diferentes ferramentas e otimizações adotadas, indicando que esta técnica será promissora para os circuitos assíncronos.

<sup>1</sup> *Total equivalent gate count for design* é uma estimativa feita pela ferramenta ISE da Xilinx do número de portas normais necessárias para um circuito equivalente



Tabela 17: Processador DLX: área *Post-Synthesis*.

Componente	Dessincronizado	Síncrono
DLX	94,69%	96,90%
Elem. de Atraso	1,37%	-
STM Comandos	1,01%	0,81%
UART	2,80%	2,19%
Outros	0,13%	0,10
Total LUTs (4 input)	3.080	3.823
Total de portas equivalentes	550.352 (181,5%)	303.185 (100%)

Tabela 18: Processador DLX: temporização na prototipação.

Processador DLX	Dessincronizado	Síncrono
Período mínimo	19,925ns	19,625ns
Frequência máxima	50,188MHz	50,955MHz

Neste ponto é importante salientar que *core* do processador ASPIDA DLX, em ambas as versões: síncrona e assíncrona, foram mantidos tal e qual disponibilizados na página do projeto na *internet* do *open core* [118], apenas alterando-se a capacidade da memória e a interface de saída para uma UART.

## 9.8 Softwares de ensaio

Os *softwares* escolhidos para serem empregados nos ensaios deste segundo estudo de caso desta dissertação, são consagradas aplicações no meio acadêmico [119] [22] [120] [121], e também possuem um elevado índice de operações de multiplicação para a sua correta execução, bem como vários saltos. Isto foi decisivo para esta escolha, já que pretendemos avaliar o desempenho e a robustez dos processadores síncrono e assíncrono. Uma outra restrição na escolha dos *softwares* foi a limitação do DLX implementado, que opera somente com dados do tipo inteiro. Diante destas necessidades e restrições foram selecionados os *softwares* listados a seguir:

1. Multiplicação de Matrizes (MM): realiza a multiplicação de duas matrizes de quarta ordem, operando com números inteiros;
2. *InsertSort* (IS): realiza a ordenação de um vetor contendo dez números inteiros;

Devido ao fato do sistema não possuir a funcionalidade para carga do *software* para a execução pelo processador, e não haver tempo hábil para a respectiva implementação, foi necessária a inclusão dos parâmetros de inicialização dos blocos de memória contendo o código de máquina do *software* para o ensaio. Para compilar as aplicações foi utilizado o compilador gcc (*GNU Compiler Collection*), conforme disponibilizado na internet na página do projeto no *Open Core* [115]. Desta forma, foi necessário gerar um *bitstream* para cada versão de *software* utilizada durante os ensaios.

## 9.9 Ensaios realizados

Esta seção tem por objetivo apresentar os ensaios realizados para determinação e análise da robustez do circuitos síncronos e assíncronos, com os seus respectivos resultados. Foram realizados ensaios de interferência eletromagnética conduzida, redução e queda de nível de tensão de alimentação, e ensaios de interferência eletromagnética irradiada e emitida, cujos procedimentos e configurações serão apresentados em sequência.

### 9.9.1 Ensaios de redução dos níveis de tensão de alimentação

Os ensaios foram realizados nas dependências do Laboratório de Sistemas, Sinais e Computação (SiSC) da Faculdade de Engenharia da PUCRS. Os ensaios de redução ou queda dos níveis de tensão de alimentação seguiram as configurações apresentadas no diagrama da Figura 69, conforme a norma IEC 61.000-4-17 [11].

A Figura 70 apresenta a sequência de procedimentos adotados nos ensaios de redução ou queda dos níveis de tensão de alimentação listados a seguir.

- a) **Início:** A placa de alimentação e injeção de falhas e a plataforma de ensaio são conectadas à estação de trabalho supervisória (PC - *Desktop*) através de um canal de comunicação serial (RS-232), às respectivas fontes de alimentação, e na sequência são energizadas.

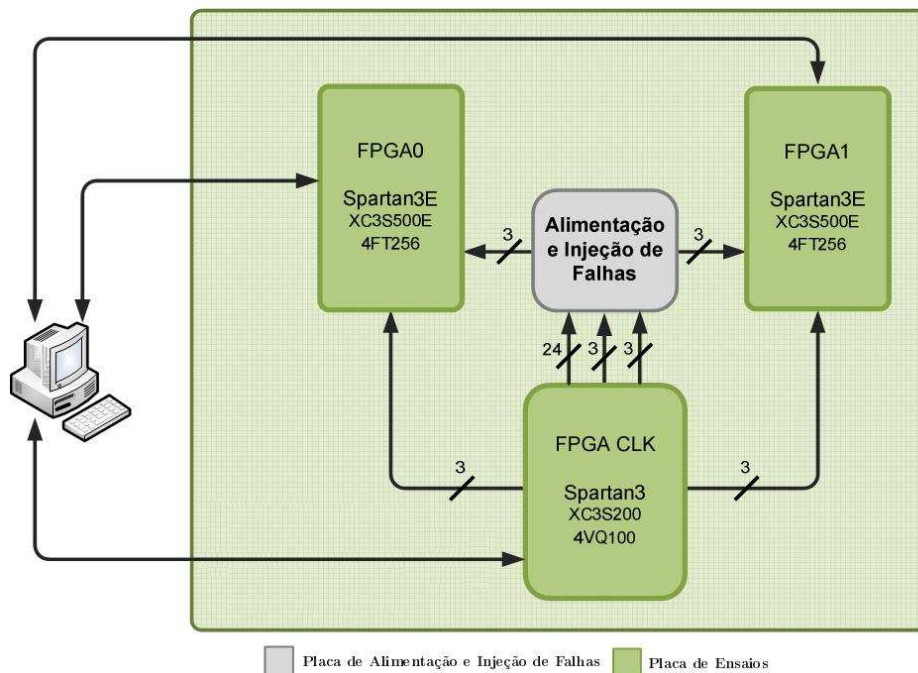


Figura 69: Configurações dos ensaios de redução e queda dos níveis de tensão de alimentação.

- b) **Configuração do FPGA CLK:** O *bitstream* do Circuito Gerenciador da Alimentação do Ensaio (7.4) é carregado no FPGA CLK via cabo JTAG através do *software Impact* fornecido pela Xilinx;
- c) **Configuração FPGA0 ou FPGA1:** O processador DLX é carregado nos FPGA 0 ou 1 via cabo JTAG através do *software Impact*, incluindo o *software* aplicativo do ensaio, na versão assíncrona ou síncrona;
- d) **Execução das aplicações:** As aplicações de teste (Multiplicação de Matrizes ou *InsertSort*) são inicializadas pelo Circuito Gerenciador da Alimentação do Ensaio, através de um comando de inicialização (*reset*) enviado pelo PC pela interface serial (RS-232) para o FPGA CLK.
- e) **Redução dos níveis de tensão:** O nível da tensão de alimentação do *core* do FPGA sob ensaio (FPGA0 ou FPGA1) é reduzida, através da placa de alimentação e injeção de falhas, do nível nominal (1,2V), inicialmente para 960mV, através de através de um comando para trocar para o modo de redução de tensão, enviado pelo PC pela interface serial (RS-232) para o FPGA CLK.

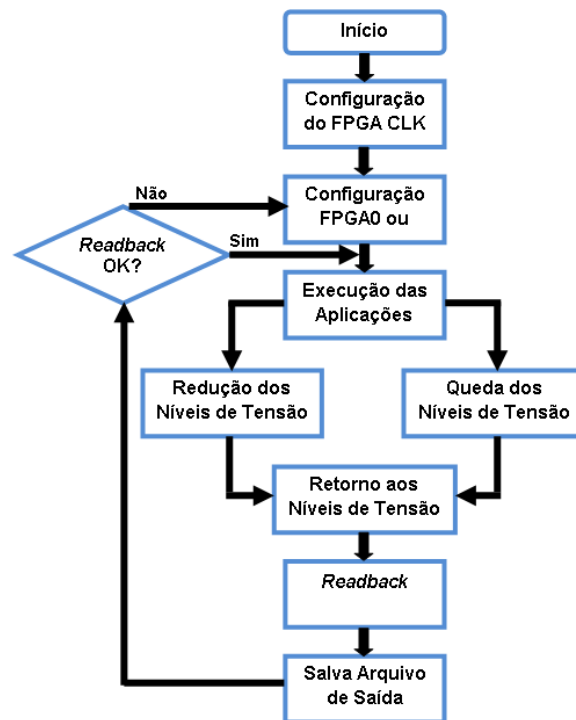


Figura 70: Sequência de procedimentos dos ensaios de redução e queda de tensão de alimentação.

Na sequência, verifica-se o comportamento do sistema pela estação de trabalho supervisória, que recebe as saídas do sistema sob ensaio, e gradualmente é reduzida o nível de tensão de alimentação, através de comandos para reduzir ou aumentar o nível de tensão de alimentação, enviado pelo PC pela interface serial (RS-232) para o FPGA CLK. O valor da redução do nível de tensão é diminuído até o momento em que as saídas do sistema sob ensaio comecem a apresentar erros.

- f) **Retorno aos níveis de tensão nominais:** O nível da tensão de alimentação do *core* do FPGA sob ensaio (FPGA0 ou FPGA1) é restabelecido para o valor nominal (1,2V) pela da placa de alimentação e injeção de falhas, através de um comando específico, retornando ao modo tensão nominal, enviado pelo PC pela interface serial (RS-232) para o FPGA CLK;
- g) **Readback:** É realizado processo de *readback* através do *software Impact*, pelo qual efetua-se a comparação entre o *bitstream* carregado e o *bitstream* padrão, para verificação de ocorrência de falhas no *hardware* (processador DLX) prototipado no FPGA sob ensaio.
- h) **Salva arquivo:** Os arquivos de saída (*logs*) gerados nos testes, são salvos e os resultados registrados na planilha de ensaios, com a informação se o *readback* apresentou

ou não diferenças entre o *bitstream* padrão e o *bitstream* carregado, e o valor do nível de redução ou queda de tensão de alimentação onde começaram as ocorrências de erros na saída do sistema.

A aplicação executada pelo processador DLX, síncrono ou assíncrona, imprime para cada iteração, os estados de saída via uma interface serial (RS232). Estas impressões foram armazenadas em arquivos de *log* e comparadas a um arquivo padrão gerado pela execução das aplicações de teste no processador DLX sem qualquer tipo de ruído nas linhas de alimentação do FPGA. As ocorrências de divergências entre os resultados dos arquivos de *log* e o arquivo padrão, foram consideradas falhas de dados, sendo registradas na planilha de ensaio o número de execuções realizadas até a ocorrência do primeiro erro.

O Gráfico 71 apresenta o comparativo entre o nível de tensão de alimentação onde começaram a ocorrência de falhas registradas na versão síncrona e assíncrona do processador DLX.

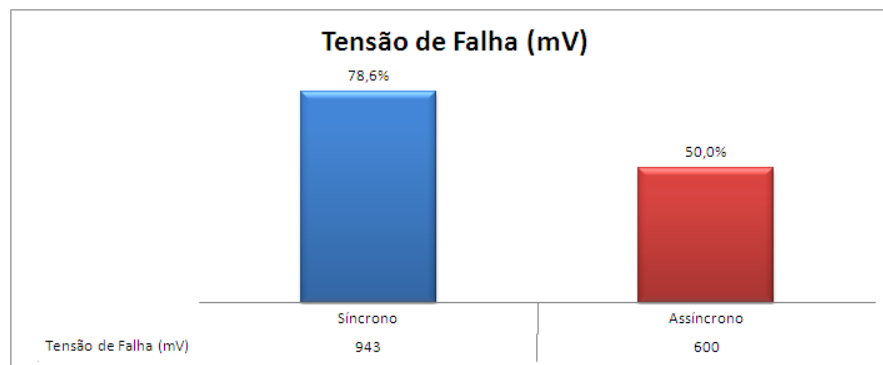


Figura 71: Gráfico comparativo das tensões de falha (Redução de Tensão).

Conforme pode ser constatado no Gráfico 71, ao reduzirmos o nível da tensão de alimentação do *core* do FPGA no circuito síncrono, este manteve-se plenamente funcional até o valor em torno de 943mV, o que representa 78,6% do valor da tensão nominal de operação do *core* do FPGA. Em contrapartida, ao aplicarmos o mesmo procedimento no circuito assíncrono, este por sua vez manteve-se plenamente funcional até o valor em torno de 600mV, o que representa 50,0% do valor da tensão nominal de operação do *core* do FPGA.

### 9.9.2 Ensaios de queda dos níveis de tensão de alimentação

De forma análoga ao descrito no item 9.9.1, foram realizados os ensaios de queda dos níveis de tensão de alimentação. Para isto foi adotado o mesmo procedimento descrito na

seção 9.9.1 e ilustrado na Figura 70, somente alterando-se o item (e), descrito a seguir, realizando-se as quedas do nível de tensão conforme a norma IEC 61.000-4-29 [12].

- e1) **Queda dos Níveis de Tensão:** As quedas do nível da tensão de alimentação do *core* do FPGA sob ensaio (FPGA0 ou FPGA1) são comandadas, através da placa de alimentação e injeção de falhas, do nível nominal (1,2V), inicialmente para 960mV, através de comando que altera para o modo de queda de tensão, enviado pelo PC pela interface serial (RS-232) para o FPGA CLK.

Em seguida, o comportamento do sistema é verificado através da estação de trabalho supervisória, observando as saídas do sistema sob ensaio, e gradualmente o nível de queda de tensão de alimentação é aumentado, utilizando-se de comandos para reduzir ou aumentar o nível de queda de tensão de alimentação, enviando-se pelo PC pela interface serial (RS-232) para o FPGA CLK. O valor de queda do nível de tensão é aumentado até o momento em que as saídas do sistema sob ensaio comecem a apresentar erros.

Tal qual no ensaio de redução do nível de tensão de alimentação, as impressões foram armazenadas em arquivos de *log* e comparadas, sendo as ocorrências de divergências entre os resultados dos arquivos de *log* e o arquivo padrão, consideradas falhas de dados, sendo registradas na planilha de ensaio o número de execuções realizadas até a ocorrência do primeiro erro.

O comparativo entre o nível de tensão de alimentação onde começaram a ocorrência de falhas registradas na versão síncrona e na versão assíncrona do processador DLX é apresentado no Gráfico 72.

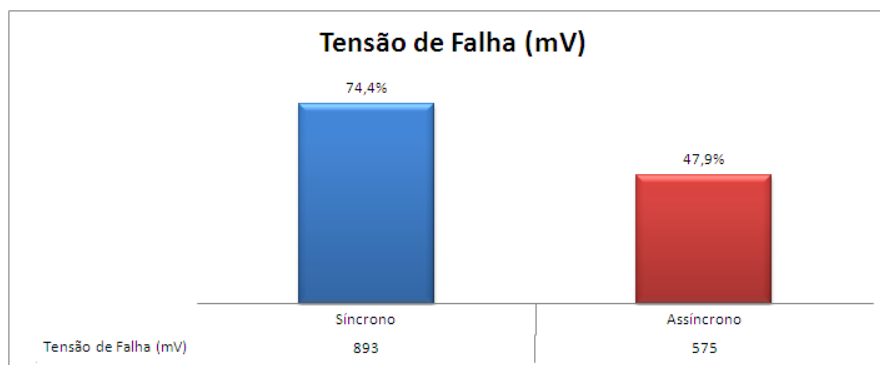


Figura 72: Gráfico comparativo das tensões de falha (Queda de Tensão).

Ao aumentarmos no nível da queda de tensão de alimentação do *core* do FPGA no circuito síncrono, verificou-se que este funcionou de maneira correta até o valor em torno

de 893mV, representando 74,4% do valor da tensão nominal de operação do *core* do FPGA, conforme pode ser constatado no Gráfico 72. Por outro lado, circuito assíncrono manteve-se plenamente funcional até o valor em torno de 575mV, representando 47,9% do valor da tensão nominal de operação do *core* do FPGA.

### 9.9.3 Ensaios de interferência eletromagnética irradiada

Realizados na Universidade Politécnica da Catalunha, no Grupo de Compatibilidade Electromagnética, Barcelona, em fevereiro de 2009. Estes ensaios de interferência eletromagnética irradiada seguiram a norma IEC 62.132-2 [40]. Os equipamentos empregados são específicos para este tipo de ensaio, sendo necessário um gerador de sinais de alta frequência (1Hz-3GHz), um amplificador de sinais, um medidor de intensidade de campo elétrico, uma antena, e uma câmara anecoica. Na Figura 73 podemos observar a configuração do ensaio.



Figura 73: Configuração dos ensaios de interferência eletromagnética irradiada.

Para os ensaios de EMI irradiada, a placa foi colocada em várias posições, horizontal, vertical, e também numa caixa metálica, conforme mostrado na Figura 74. Esta caixa tem as dimensões de 30x32x12cm, com uma abertura para a fixação da plataforma de ensaios (16x17cm) e outra para a passagem dos cabos de alimentação, configuração e

comunicação (3x4cm). Esta caixa metálica foi utilizada com a finalidade de isolar e proteger os componentes e trilhas de sinais fixados do lado da placa que não deveriam estar sujeitos à interferências eletromagnéticas (*bottom*), comportando-se como uma Gaiola de Faraday <sup>2</sup>

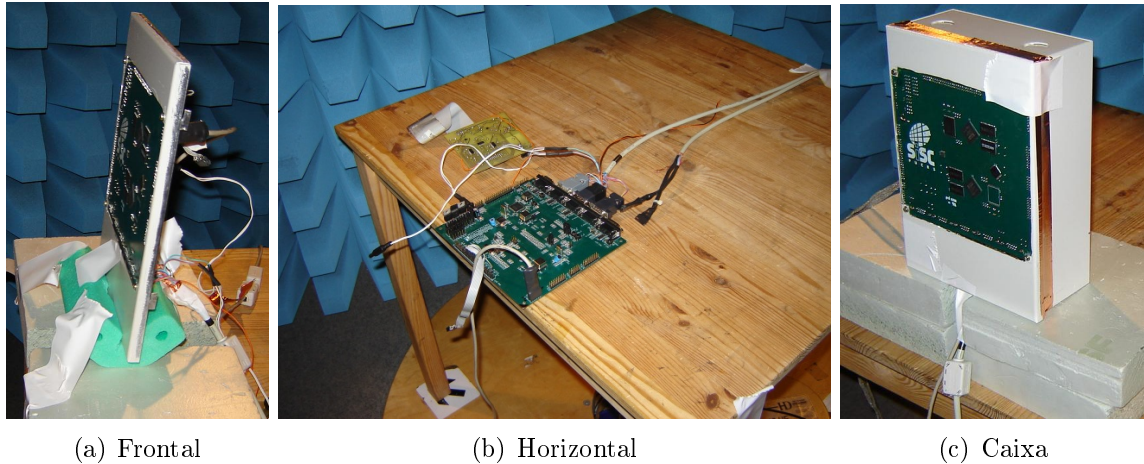


Figura 74: Posições da placa durante os ensaios

O procedimento adotado para a realização dos ensaios segue uma sequência similar àquelas descrita nos itens anteriores, apenas alterando-se os itens (e) e (f), conforme descrito a seguir.

- e2) **Acionamento da interferência eletromagnética:** Energiza-se o gerador e o amplificador de sinais, varia-se a frequência de 80MHz a 1GHz, e a intensidade de campo eletromagnético no interior da câmara anecoica entre 18 e 100V/m.
- f2) **Suspensão da interferência eletromagnética:** Desliga-se o gerador e o amplificador de sinais.

As saídas dos aplicativos de ensaios foram armazenadas em arquivos de *log* e comparadas, sendo as ocorrências de divergências entre os resultados dos arquivos de *log* e o arquivo padrão, consideradas falhas de dados, sendo registradas na planilha de ensaio a frequência e a intensidade de campo magnético a partir da qual começam a ocorrer falhas. Os gráficos 75 e 76 apresentam o resultado dos ensaios, sendo assinaladas com barras na cor vermelha as faixas de frequência onde ocorreram falhas.

<sup>2</sup> Uma Gaiola de Faraday é uma superfície condutora metálica que envolve uma dada região do espaço e que pode, em certas situações, impedir a propagação de interferências produzidas por campos elétricos e/ou eletromagnéticos externos.





Figura 75: Gráfico de ocorrência de falhas nos ensaios de interferência eletromagnética irradiada (DLX síncrono).



Figura 76: Gráfico de ocorrência de falhas nos ensaios de interferência eletromagnética irradiada (DLX assíncrono).

O comparativo entre a ocorrência de falhas registradas na versão síncrona e na versão assíncrona do processador DLX é apresentado no Gráfico 77, sendo estes dados normalizados em relação ao circuito assíncrono. Constatamos que quando submetido a interferências eletromagnéticas irradiadas, o circuito síncrono apresentou um desempenho inferior ao circuito assíncrono.

Conforme vemos no intervalo de 80 à 400Mhz os dois circuitos têm o mesmo número de falhas, sendo de nosso entendimento que este fato é devido a indução através dos cabos de alimentação e comunicação. Nesta faixa de frequência temos o comprimento de onda de 3,75m à 0,75m respectivamente ( $\lambda = c/f$ ,  $c =$  velocidade da luz), com pulsos na ordem de  $ns$ , assim sendo os condutores de alimentação e comunicação apresentam-se como antenas. Portanto, pelo acoplamento nesta faixa de frequência temos a indução de ruído nas linha do barramento de alimentação do FPGA (VDD e GND), bem como nos sinais de saída do sistema (UART).

Portanto, para efeito de análise consideramos a faixa de frequência acima de 400MHz, assim contabilizamos para o circuito síncrono um índice de falhas em torno de 314,5% superior ao circuito assíncrono, valor normalizado em relação ao circuito assíncrono.

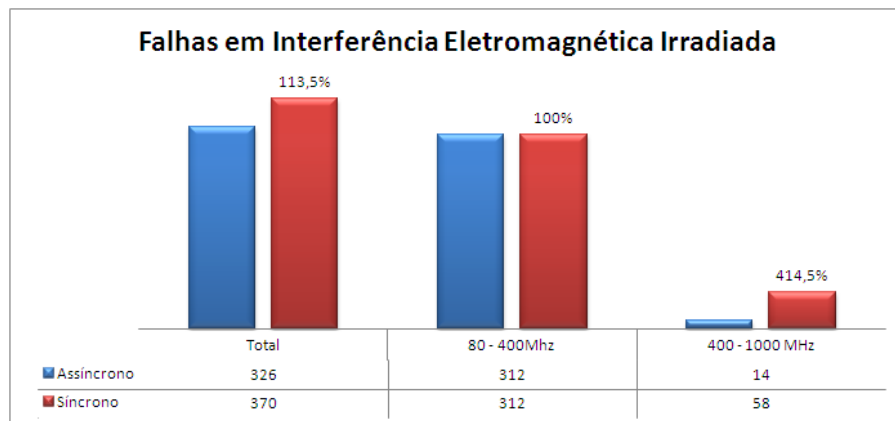


Figura 77: Gráfico comparativo dos ensaios de interferência eletromagnética irradiada.

#### 9.9.4 Ensaios de interferência eletromagnética emitida

Da mesma forma que os ensaios da seção anterior, os ensaios de interferência eletromagnética emitida foram realizado na Universidade Politécnica da Catalunha, no Grupo de Compatibilidade Electromagnética. Os equipamentos empregados são específicos para este tipo de ensaio, sendo necessário um medidor de intensidade de campo elétrico, duas antenas e uma câmara anecoica. Além da configuração apresentada no item 9.9.3, utili-

zada para medição de intensidade de *campo elétrico distante*, também foi utilizado a configuração mostrada na Figura 78 para a medição de intensidade de *campo elétrico próximo*.

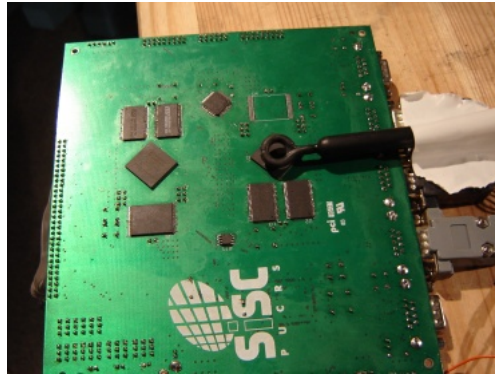


Figura 78: Configuração dos ensaios de interferência eletromagnética emitida.

O procedimento adotado para a realização dos ensaios segue uma sequência similar os dos itens anteriores, apenas alterando o item (e) e suprimindo item (f), conforme descrito a seguir.

- e3) **Medição da interferência eletromagnética emitida:** Acionado o medidor de intensidade de campo elétrico para realizar a varredura do espectro de frequência de 30 MHz a 1 GHz no interior da câmara anecoica.

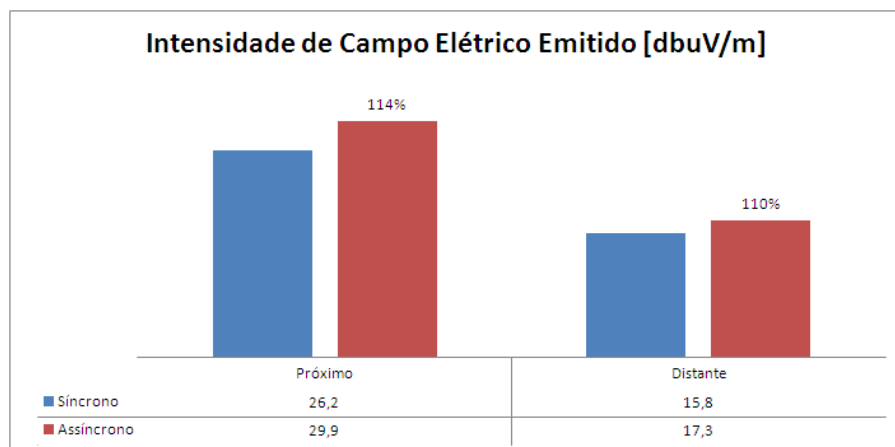


Figura 79: Gráfico comparativo dos ensaios de interferência eletromagnética emitida.

Nas Figuras 80 e 81 são mostrados os gráficos do campo elétrico emitido, respectivamente pelo circuito do processador DLX síncrono e assíncrono. A partir destas informações foi calculado a média da intensidade de campo elétrico no espectro de frequência

analisado, este resultado é apresentado no Gráfico 79, sendo os dados normalizados em relação ao circuito assíncrono. Podemos notar que existe uma diferença na intensidade de *campo elétrico distante*, sendo a emissão do circuito assíncrono  $1,5dB\mu V$  superior ao circuito síncrono e, utilizando a equação 9.1 para calcular o valor da intensidade de campo elétrico em  $\mu V$ , temos o circuito síncrono com uma emissão média de  $6,14\mu V$  e o circuito assíncrono com uma emissão média de  $7,32\mu V$ , portanto temos aproximadamente uma emissão 1,2 vezes superior. Fato este evidenciado também no *campo elétrico próximo*, onde temos o circuito assíncrono com uma emissão de  $3,7dB\mu V$  a mais que o circuito síncrono, ou seja, com o uso da equação 9.1, temos o circuito síncrono com uma emissão média de  $20,5\mu V$  e o circuito assíncrono com uma emissão média de  $31,4\mu V$ , o que representa uma emissão aproximadamente 1,5 vezes superior.

$$U = 20 \cdot \log_{10} \left( \frac{u}{u_0} \right) \quad (9.1)$$

onde, adotamos  $u_0 = 1V$  para obtemos  $dBV$  e  $u_0 = 1\mu V$  para obtemos  $dB\mu V$ , e consequentemente temos que  $0dB\mu V = -120dBV$ .

No nosso entendimento estas diferenças devem-se ao fato de que o circuito assíncrono, gerado pelo processo de dessincronização, prototipado em FPGA, ocupou em torno de 81% a mais que o circuito síncrono, conforme podemos verificar na Tabela 17. Ainda no intuito de compreender este aumento de emissão, que esta técnica de dessincronização apresentou, nos reportamos aos estudos de Andrikos Nikolaos [104] para a implementação na tecnologia ST CORE9 90nm, utilizando o fluxo de projeto da Synopsys-Avanti STMicroelectronics, no qual obteve como resultado um aumento de área total de 13,5% e de consumo de 22%, comparando-se ao circuito síncrono. E por sua vez, no projeto ASPIDA, na prototipação de ASIC na tecnologia UMC 0.18- $\mu m$  CMOS, os resultados apontam uma pequeno aumento de área e consumo, sendo 1,5% em área e 0,4% em consumo, comparando-se ao circuito síncrono [8]. Estes fatos corroboram para este raciocínio, pois devido ao aumento de área temos um consumo de corrente maior e consequentemente temos mais atividade no circuito, portanto um emissão maior.

Também destacamos o fato de que apesar de ocupar 2% a 3% do circuito prototipado, uma UART estava presente em ambos os circuitos, contribuindo desta forma para mascarar os resultados obtidos, pelo fato de conter linhas de distribuição do sinal de relógio (*clock*), uma vez que trata-se de um circuito implementado no paradigma síncrono.

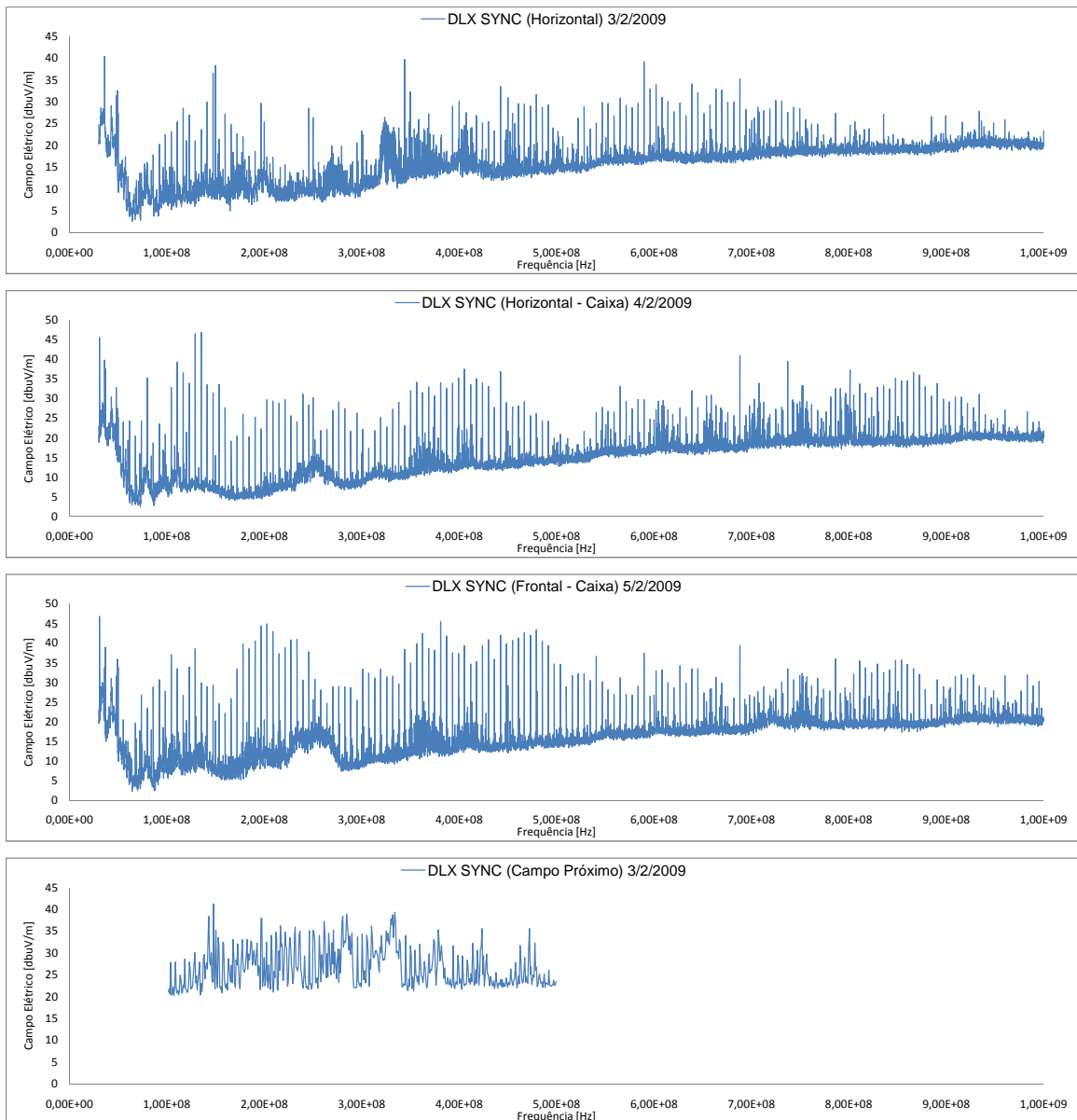


Figura 80: DLX Síncrono: Interferência eletromagnética emitida.

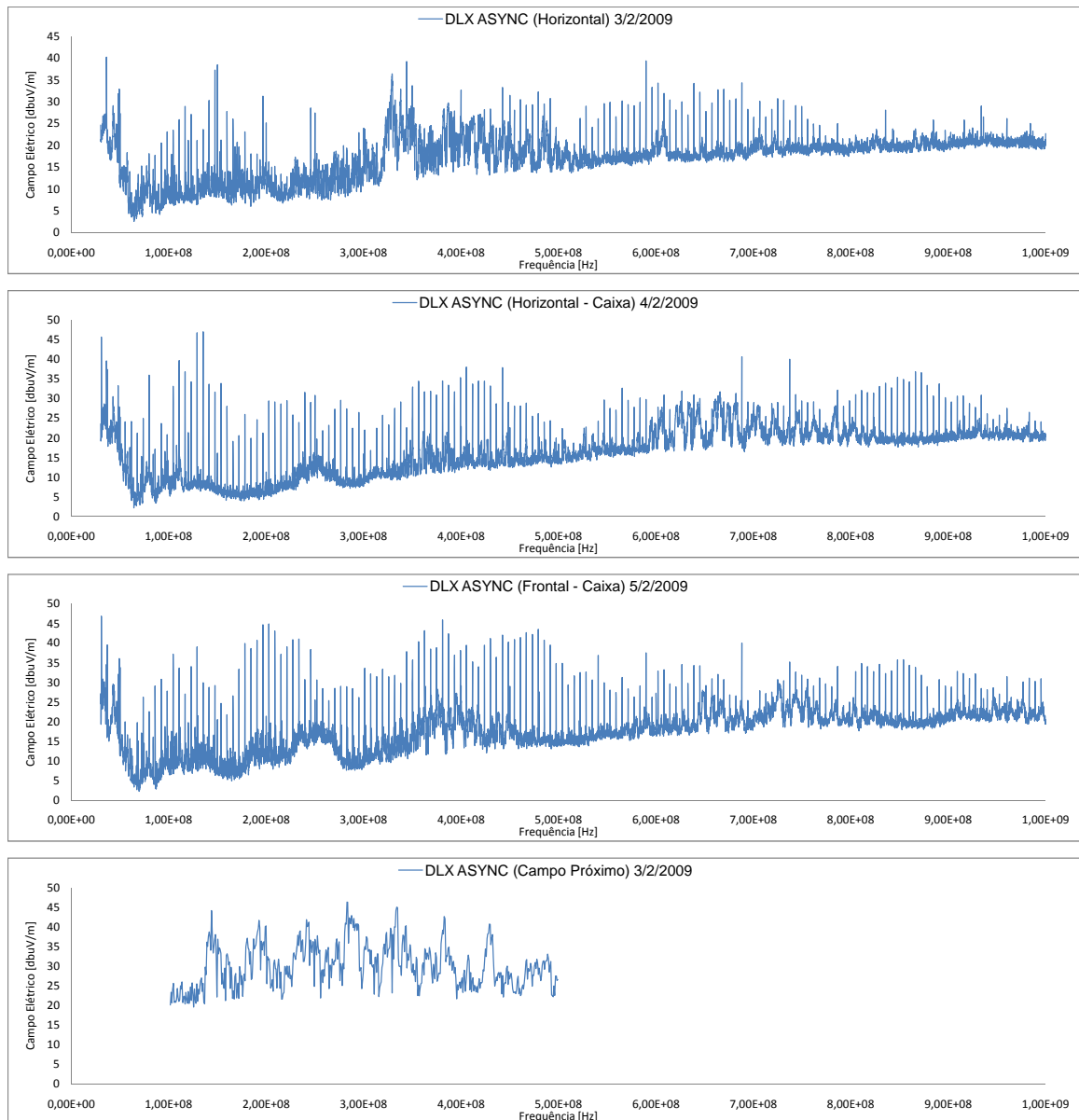


Figura 81: DLX Assíncrono: Interferência eletromagnética emitida.

## 9.10 Conclusão

A seguir teceremos as conclusões deste capítulo dividindo-as em duas grandes partes: alteração de tensão no barramento de alimentação e ensaios de compatibilidade eletromagnética em câmara anecoica.

### a) Alteração de tensão no barramento de alimentação

- **Ensaios de redução dos níveis de alimentação:** neste ensaio podemos constatar que os circuitos síncronos apresentam falhas quando a tensão de

alimentação do *core* atinge o valor em torno de 78,6% da tensão nominal. Em contrapartida, os circuitos assíncronos somente apresentam falhas em torno de 50,0% da tensão nominal.

- **Ensaio de queda dos níveis de alimentação:** de maneira similar, constatamos que os circuitos síncronos apresentam falhas quando a queda do nível de tensão de alimentação do *core* atinge o valor em torno de 74,4% da tensão nominal. E os circuitos assíncronos por sua vez, apresentam falhas com níveis de queda em torno de 47,9% da tensão nominal.

As oscilações nos níveis de tensão de alimentação de circuitos digitais síncronos, redução ou queda, provocam atrasos de propagação (*delay*) nos sinais internos destes circuitos, reduzindo a integridade de sinal significativamente e causando a degradação de desempenho, perda de confiabilidade, e podendo em caso extremos provocar a ocorrência de falhas funcionais. Por outro lado, os circuitos assíncronos apresentam uma robustez natural a essas exigências, uma vez que, possuem independência do sinal do relógio e os atrasos serão melhor assimilados pelo circuito. Este fato vem ao encontro dos resultados obtidos nos ensaios.

#### a) Compatibilidade eletromagnética

- **Ensaio de interferência eletromagnética irradiada:** na análise destes ensaios considerou-se a faixa de frequência acima de 400MHz, pois segundo o nosso entendimento no intervalo de frequência de 80 à 400Mhz ocorreu indução através dos cabos de alimentação e comunicação, devido ao fato de que os cabos para estes comprimentos de onda comportarem-se como antenas. Portanto, pelo acoplamento nesta faixa de frequência temos a indução de ruído nas linha do barramento de alimentação do FPGA (VDD e GND), bem como nos sinais de saída do sistema (UART).

Assim, tal como observado, o circuito síncrono apresentou um desempenho inferior ao circuito assíncrono, quando submetido a interferências eletromagnéticas irradiadas. Na faixa de frequência acima de 400MHz, contabilizamos para o circuito síncrono um índice de falhas em torno de 314,5% superior ao circuito assíncrono, valor normalizado em relação ao circuito assíncrono.

- **Ensaio de interferência eletromagnética emitida:** comparando-se a média da intensidade de campo elétrico no espectro de frequência analisado, verifica-se que o circuito assíncrono possui níveis de emissão maiores que o

circuito síncrono, tanto no *campo elétrico distante*, com valores aproximadamente 1,2 vezes superior ao circuito síncrono, como no *campo elétrico próximo*, com uma emissão em torno de 1,5 vezes a mais que o circuito síncrono. É de nosso entendimento que estas diferenças devem-se ao fato de que o circuito assíncrono, gerado pelo processo de dessincronização, ocupou em torno de 81% a mais área que o circuito síncrono. Neste sentido temos os resultados obtidos em estudos anteriores desta técnica [104] [8], nos quais na simulação de implementação com a tecnologia ST CORE9 90nm, obteve-se como resultado um aumento de área total de 13,5% e de consumo de 22%, em outro para prototipação de ASIC na tecnologia UMC 0.18- $\mu$ m CMOS, os resultados apontam uma pequeno aumento de área e consumo, sendo 1,5% em área e 0,4% em consumo, comparando-se ao circuito síncrono. Estes fatos corroboram para este raciocínio, pois devido ao aumento de área temos um consumo de corrente maior e conseqüentemente temos mais atividade no circuito, portanto uma emissão maior.

Aliado a este fato, temos uma UART que apesar de ocupar 2% a 3% do circuito prototipado, estava presente em ambos os circuitos, assim contribuindo para mascarar os resultados obtidos, pelo fato de conter linhas de distribuição do sinal de relógio (*clock*), uma vez que trata-se de um circuito implementado de modo síncrono.

Ao final dos ensaios nota-se que os circuitos assíncronos apresentam-se mais robusto que os circuitos síncronos, quando submetido a interferências eletromagnéticas conduzidas do tipo redução e queda do nível de tensão alimentação, e interferências eletromagnéticas irradiadas. No entanto as medições de interferência eletromagnética emitida, mostram que devido ao aumento de área utilizada no FPGA, o circuito assíncrono apresenta um emissão de campo elétrico superior.

No caso da medição de intensidade de campo elétrico emitido, o veículo utilizado nos ensaios, o FPGA, não apresentou-se como sendo o mais adequado para a realização deste tipo de ensaio, e desta forma podemos ter os resultados destes ensaios mascarados.



## *10 Conclusões finais*

Os circuitos assíncronos apresentam-se como uma alternativa interessante aos projetos de sistemas robustos a falhas, comparando-se aos circuitos síncronos, quando submetidos a interferência eletromagnética, embora não utilizados em larga escala. Existem fatores limitantes para a sua difusão, residindo principalmente na necessidade de ferramentas específicas de projeto, além de conhecimento em técnicas de projeto de circuitos assíncronos por parte dos projetistas. Uma solução alternativa e simples para obtenção de circuitos assíncronos, chamada de Dessincronização, foi apresentada por J. Cortadella, A. Kondratyev, L. Lavagno e C. Sotiriou em 2004. Esta técnica transforma um circuito síncrono em assíncrono, utilizando-se de ferramentas de CAD convencionais, derivando-se pouco do fluxo de projetos síncronos.

Assim, no presente trabalho, com o intuito de realizar uma análise comparativa da robustez entre os circuitos síncronos e assíncronos gerados pela técnica de dessincronização, validando a abordagem proposta, fez-se necessário a realização das etapas abaixo:

- Pesquisa dos circuitos e processadores assíncronos para posterior implementação e consequente avaliação do seu desempenho.
- O processador DLX foi portado para a plataforma de ensaio adotada, projetada e desenvolvida pela equipe do Laboratório SiSC - PUCRS.
- Compilação e implementação de programas aplicativos para os ensaios;
- Realização de ensaios de interferência eletromagnética irradiada e conduzida, redução e queda de tensão de alimentação, e interferência eletromagnética emitida.

A análise dos resultados obtidos nos ensaios realizados indicam que a técnica de dessincronização adotada mostrou-se favorável para a geração de circuitos assíncronos, os quais apresentaram-se mais robustos e mais confiáveis quando submetidos às interferências eletromagnéticas, quando comparados aos circuitos síncronos equivalentes. Esta melhora de

desempenho possui um custo, que reflete-se no aumento de área de silício utilizada, e neste caso estudado, tendo como consequência um aumento da emissão de campo elétrico.

Finalmente, no nosso entendimento, para ensaios de medição de intensidade de campo elétrico emitido, o FPGA, veículo utilizado nos ensaios, não apresenta-se como o mais adequado para a realização deste tipo de ensaio, o que de certa forma pode ter mascarado os resultados obtidos nestes ensaios.

## 10.1 Perspectiva de trabalhos futuros

Como sugestão de trabalhos futuros, apresentamos atividades que inicialmente constavam do planejamento inicial, ou foram constatadas durante o desenvolvimento, mas que devido ao tempo consumido nas etapas de implementação computacional e execução dos ensaios, não foram incluídas neste trabalho. Estas atividades poderão tanto facilitar quanto viabilizar a obtenção de melhores resultados em ensaios a serem realizados em trabalhos futuros, a saber:

- Produção de uma nova plataforma de ensaio com as características descritas a seguir: incorporar à placa o circuito de alimentação e injeção de falhas, tendo o controle de níveis de tensão mais refinado, incluir a opção de alimentação por bateria montada na placa, alterar o meio físico da interface de comunicação para fibra ótica, reduzir ao máximo o número de conectores metálicos, aumentar o número de sinais entre o FPGA alvo dos ensaios e o circuito gerenciador de ensaio (FPGA ou microprocessador), e incluir um dispositivo para medição de corrente elétrica fornecida ao FPGA sob ensaio;
- Implementação da técnica de dessincronização modificando a lógica combinacional para incluir a detecção de conclusão, possibilitando assim a realização comparativa do desempenho entre as técnicas para compensar o atraso da lógica combinacional, elementos de atraso ou detecção de conclusão;
- Implementação da técnica de dessincronização em um outro processador com maior complexidade e com suporte a instruções matemáticas de ponto flutuante, ficando como sugestão o processador Plasma, utilizado pelo SiSC em pesquisas recentes. Desta forma será possível observar o comportamento de um sistema operacional embarcado rodando em um processador assíncrono na presença de interferência eletromagnética;

- Realização de ensaios de variação de temperatura, para avaliação do desempenho dos circuitos quando submetidos a temperatura extremas, simulando ambientes hostis aos quais os equipamentos eletrônicos estão normalmente inseridos, como sugestão sugerimos valores entre  $-40^{\circ}C$  e  $90^{\circ}C$ .

## *Referências*

- [1] KATHAIL V., A. S. et al. Pico: automatically designing custom computers. *Computer*, v. 35, n. 9, p. 39–47, Sept. 2002.
- [2] WOLF, W. Embedded is the new paradigm(s). *Computer*, IEEE Computer Society, Los Alamitos, CA, USA, v. 37, n. 3, p. 99–101, 2004. ISSN 0018-9162.
- [3] SEMIÃO, J. et al. Power-supply instability aware clock signal modulation for digital integrated circuits. In: *EMC Europe 08*. Hamburg, Germany: [s.n.], 2008.
- [4] MORAES, M. L. *Validação de Uma Técnica Para o Aumento da Robustez de SoC's a Flutuações de Tensão no Barramento de Alimentação*. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2008. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.
- [5] SPARSØ, J.; FURBER, S. *Principles of Asynchronous Circuit Design: A Systems Perspective*. [S.l.]: Springer, 2001. 360 p.
- [6] CALAZANS, N. L. V. Automated logic design of sequential digital circuits. In: \_\_\_\_\_. [S.l.]: Imprinta, 1998. cap. Capítulo 3: Representação Seqüencial e Capítulo 5: Síntese Assíncrona, p. 342.
- [7] BERKEL, C. H. V.; JOSEPHS, M. B.; NOWICK, S. M. Applications of asynchronous circuits. *Proceedings of the IEEE*, v. 87, n. 2, p. 223–233, Feb. 1999.
- [8] CORTADELLA, J. et al. Desynchronization: Synthesis of asynchronous circuits from synchronous specifications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 25, n. 10, p. 1904–1921, Oct. 2006.
- [9] ANDRIKOS, N. et al. A fully-automated desynchronization flow for synchronous circuits. In: *Proc. 44th ACM/IEEE Design Automation Conference DAC '07*. [S.l.: s.n.], 2007. p. 982–985.
- [10] IEC International Electrotechnical Commission. *Integrated Circuits - Measurement of Electromagnetic Immunity, 150 kHz to 1 GHz - Part 1: General Conditions and Definitions (62.132-1)*. Geneva, Switzerland, 2006. P47, Norma Técnica.
- [11] IEC International Electrotechnical Commission. *Electromagnetic Compatibility (EMC) - Part 4-17: Testing and Measurement Techniques - Ripple on d.c. Input Power Port Immunity Test (61.000-4-17)*. Geneva, Switzerland, 2002. P27, Norma Técnica.
- [12] IEC International Electrotechnical Commission. *Electromagnetic Compatibility (EMC) - Part 4-29: Testing and Measurement Techniques - Voltage Dips, Short Interruptions and Voltage Variations on d.c. Input Power Port Immunity Tests (61.000-4-29)*. Geneva, Switzerland, 2000. P37, Norma Técnica.

- [13] LAPRIE, J. Dependable computing and fault-tolerance: Concepts and terminology. In: *15th IEEE Int. Symp. on Fault-Tolerant Computing*. New York: IEEE, 1985. s.n., p. 2–11.
- [14] ANDERSON T. E LEE, P. *Fault Tolerance - Principles and Practice*. [S.l.]: Prentice-Hall, 1981.
- [15] PRADHAN, D. K. *Fault-Tolerant Computer System Design*. [S.l.]: Prentice Hall, 1996.
- [16] IYER, R. K.; KALBARCZYK, Z. *Hardware and Software Error Detection*. 2002. Online. Disponível em: <[http://www.crhc.uiuc.edu/~kalbar/MotorolaCourse/HW\\_SW\\_ErrorDetection.pdf](http://www.crhc.uiuc.edu/~kalbar/MotorolaCourse/HW_SW_ErrorDetection.pdf)>.
- [17] BOLZANI, L. M. V. *Explorando uma Solução Híbrida: Hardware + Software para a Detecção de Falhas em Systems-on-Chip (SoCs)*. 329–333 p. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2005. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.
- [18] LAPRIE, J. C. Dependability: From concepts to limits. In: *Proceedings of the IFIP International Workshop on Dependable Computing and its Applications*. Johannesburg, South Africa: [s.n.], 1998. p. 108–126. DCIA 98.
- [19] BARDELL W. H. MCANNEY, J. S. P. H. *Built in Test for VLSI: Pseudorandom Techniques*. New York, USA: Wiley-Interscience, 1987. ISBN: 978-0-471-62463-9.
- [20] STROUD, E. C. *A Designer's Guide to Built-In Self-Test*. Boston: Kluwer Academic Publishers, 2002. 15-27 p.
- [21] CORTNER, J. M. *Digital Test Engineering*. United States of America: Wiley-Interscience, 1987. 1-27 p.
- [22] BENFICA, J. D. *Plataforma para Desenvolvimento de SoC (System-on-Chip) Robusto à Interferência Eletromagnética*. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2007. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.
- [23] BIENERT, R. *EMC in Wireless Devices*. [S.l.]: TUV Rheiland of North America, 2004.
- [24] BOSCH, R. *Electromagnetic Compatibility - Definition and Need*. S. India: Suganya, 2004.
- [25] IEC International Electrotechnical Commission. *International Electrotechnical Vocabulary - Part 161 ( 60050-161)*. Geneva, Switzerland, 1990. P66, Norma Técnica.
- [26] WESTON, D. *Electromagnetic Compatibility: Principles and Applications*. 2a. ed. New York, USA: Marcel Dekker, Inc, 2001. 1-17 p. ISBN 0-8247-8889-3.
- [27] SOUZA, A. M. *Eletromagnetismo, Notas de Aula*. Dezembro 2007. Online. Disponível em: <<http://cict.inatel.br>>.

- [28] KRAUSS, J. D.; FLEISCH, D. *Electromagnetics With Applications*. [S.l.]: Mc Graw-Hill, 1999.
- [29] ZEVZIKOVAS, M. *Efeitos da Interferência Eletromagnética Conduzida em Equipamentos Eletromédicos no Ambiente Hospitalar*. Dissertação (Mestrado) — Universidade Estadual de Campinas - UNICAMP, São Paulo, 2004. P. 86, Programa de Pós-Graduação em Engenharia Elétrica e de Computação - Dissertação de Mestrado.
- [30] KIMMEL, W.; GERKE, D. *Electromagnetic Compatibility in Medical Equipment: A Guide for Designers and Installers*. USA: IEEE and Interpharm Press, Inc., 2001. ISBN 0-935184-80-5 and 0-7803-1160-4.
- [31] NOURANI M, A. A. Signal integrity: Fault modeling and testing in high-speed socs. *Journal of Electronic Testing*, v. 18, n. 4-5, p. 539–554, Agosto 2002.
- [32] FANG PENG, T. J. C. J.; HU, C. Design in hot-carrier reliability for high performance logic applications. In: *Custom Integrated Circuits Conference*. Santa Clara, CA, USA: IEEE, 1998. (Custom Integrated Circuits Conference), p. 525–531. ISBN: 0-7803-4292-5.
- [33] LEBLEBICI, Y. Design considerations for cmos digital circuits with improved hot-carrier reliability. *Journal of Solid-State Circuits*, v. 31, p. 1014–1024, Julho 1996.
- [34] ZHAO SHIYOU, R. K. Estimation of switching noise on power supply lines in deep sub-micron cmos circuits. In: *13th International Conference on VLSI Design*. Los Alamitos, CA, USA: IEEE Computer Society, 2000. p. 168. ISSN: 1063-9667.
- [35] VARGAS, F. et al. Emi - induced soft error rate for cots microprocessor. In: *5th IEEE Latin American Test Workshop*. [S.l.: s.n.], 2004. p. 169–172.
- [36] CHEN, H. H.; WANG, L. K. Design for signal integrity: The new paradigm for deep submicron vlsi design. In: *Proc. Technical Papers VLSI Technology, Systems, and Applications 1997 International Symposium on*. [S.l.: s.n.], 1997. p. 329–333.
- [37] CHO, D. H. et al. Interconnect capacitance, crosstalk, and signal delay for 0.35 & cmos technology. In: *Proc. International Electron Devices Meeting*. [S.l.: s.n.], 1996. p. 619–622.
- [38] RODRIGUEZ-IRAGO, M. et al. Dynamic fault test and diagnosis in digital systems using multiple clock schemes and multi-vdd test. In: *11th IEEE International On-Line Testing Symposium - IOLTS*. Los Alamitos, CA, USA: IEEE Computer Society, 2005. p. 281–286. ISSN: 1530-1591.
- [39] IEC. *IEC - International Electrotechnical Commission*. Dezembro 2008. Online. Disponível em: <<http://www.iec.ch/>>.
- [40] IEC International Electrotechnical Commission. *Integrated Circuits - Measurement of Electromagnetic Immunity, 150 kHz to 1 GHz - Part 2: Measurement of Radiated Immunity - Tem-Cell and Wideband Tem-Cell Method (62.132-2)*. Geneva, Switzerland, 2007. Norma Tecnica.

- [41] IEC International Electrotechnical Commission. *Integrated Circuits - Measurement of Electromagnetic Immunity, 150 kHz to 1 GHz - Part 3: Bulk Current Injection (BCI) Method (62.132-3)*. Geneva, Switzerland, 2007. P. 37 - Norma Tecnica.
- [42] IEC International Electrotechnical Commission. *Integrated Circuits - Measurement of Electromagnetic Immunity, 150 kHz to 1 GHz - Part 4: Direct RF Power Injection Method (62.132-4)*. Geneva, Switzerland, 2006. P. 49 - Norma Tecnica.
- [43] IEC International Electrotechnical Commission. *Integrated Circuits - Measurement of Electromagnetic Immunity, 150 kHz to 1 GHz - Part 5: Workbench Faraday Cage Method (62.132-5)*. Geneva, Switzerland, 2005. P. 45 - Norma Tecnica.
- [44] REIS, R. A. d. L. *Concepção de Circuitos Integrados*. Porto Alegre: Sagra Luzzatto, 2000. 252 p. ISBN 8524106255.
- [45] International Sematech. *International Technology Roadmap for Semiconductors*. 2008. Online. Disponível em: <<http://www.itrs.net>>.
- [46] BERGAMASCHI, R. A.; LEE, W. R. Designing systems-on-chip using cores. In: *Design Automation Conference - DAC'00*. Los Angeles, California, United States: ACM, 2000. p. 420–425. ISBN: 1-58113-187-9.
- [47] BERGAMASCHI, R. A. et al. Automating the design of socs using cores. *IEEE Design & Test of Computers*, v. 18, n. 5, p. 32–45, Sept.–Oct. 2001.
- [48] MARTIN, G.; CHANG, H. System on chip design. In: *International Symposium on Integrated Circuits, Devices & Systems - ASIC'01*. Shanghai, China: IEEE, 2001. p. 12–17. ISBN: 0-7803-6677-8.
- [49] JUNEIDI, Z. et al. Global modeling and simulation of system-on-chip embedding mems devices. In: *International Symposium on Integrated Circuits, Devices & Systems - ASIC'01*. Shanghai, China: IEEE, 2001. p. 666 – 669.
- [50] BENINI, L.; MICHELI, G. D. Powering networks on chip: Energy-efficient and reliable interconnect design for socs. In: *International Symposium on System Synthesis*. Montréal, P.Q., Canada: ACM, 2001. p. 33 – 38. ISBN:1-58113-418-5.
- [51] RUNNER, S.; SANAKA, V.; YU, E. *Building an Infrastructure for IP Reuse*. Maio 2000. Online. Disponível em: <<http://www.eetimes.com/story/OEG20000515S0038>>.
- [52] BERGAMASCHI, R.; COHN, J. The a to z of socs. In: *International Conference on Computer Aided Design - ICCAD*. [S.l.]: IEEE / ACM, 2002. p. 791–798. ISSN: 1092-3152.
- [53] THOMAS D. E MOORBY, P. *The Verilog Hardware Description Language*. [S.l.]: Kluwer Academic Publishers, 1991. 376 p.
- [54] PERRY, D. *VHDL*. 3rd edition. ed. New York, USA: McGraw-Hill, 1998. 493 p.
- [55] SCHILDT, H. *C Completo e Total*. 3 ed. ed. São Paulo: [s.n.], 1997. 827 p.
- [56] SOCIETY, I. C. *IEEE Standard SystemC Language Reference Manual*. New York, USA: IEEE Computer Society, 2006. 423 p. ISBN 0-7381-4870-9 SS95505. Disponível em: <<http://standards.ieee.org/getieee/1666/download/1666-2005.pdf>>.

- [57] GAJSKI, D. et al. *Specification Language and Methodology*. Norwell, MA: Kluwer Academic Publishers, 2000. 336 p.
- [58] MADISSETTI, V. K.; L, S. Interface design for core-based systems. In: *IEEE Design & Test of Computers*. [S.l.: s.n.], 1997. v. 14, n. 4, p. 45–51.
- [59] OST, L. C. *Redes Intra-Chip Parametrizáveis com Interface Padrão para Síntese em Hardware*. 134 p. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2004. Faculdade de Informática, PPrograma de Pós-Graduação em Ciência da Computação.
- [60] GUPTA, R. K.; ZORIAN, Y. Introducing core-based system design. In: *IEEE Design & Test of Computers*. [S.l.: s.n.], 1997. v. 14, n. 4, p. 15–25.
- [61] KUMAR, S. Networks on chip. In: \_\_\_\_\_. [S.l.]: Kluwer Academic Publishers, 2003. cap. 5 - On packet switched networks for on-chip communication, p. 85–106.
- [62] ZEFERINO, C. A. *Redes-em-Chip: Arquiteturas e Modelos para Avaliação de Área e Desempenho*. Tese (Tese de Doutorado) — Universidade Federal do Rio Grande do Sul - UFRGS, Porto Alegre, 2003. PPGC.
- [63] GUERRIER, A. G. P. A generic architecture for on-chip packet-switched interconnections. In: *Design Automation and Test in Europe - DATE'00*. Paris, França: [s.n.], 2000. p. 250–256. ISBN: 0-7695-0537-6.
- [64] HWANG, K. *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. New York: McGraw-Hill, 1993. 771 p.
- [65] BENINI L., D. M. G. Networks on chips: a new soc paradigm. *Computer*, v. 35, n. 1, p. 70–78, Jan. 2002.
- [66] KUMAR, S. et al. A network on chip architecture and design methodology. In: *Computer Society Annual Symposium on VLSI - ISVLSI'02*. [S.l.: s.n.], 2002. p. 105–112.
- [67] ARM Corp. *AMBA 2.0 Specification*. 2007. Online. Disponível em: <<http://www.arm.com/products/solutions/AMBAHomePage.html>>.
- [68] ALTERA. *Avalon Bus Specification - Reference Manual*. Dezembro 2007. Online. Disponível em: <[http://www.altera.com/products/software/products/sopc/avalon/nio-avalon\\_features.html](http://www.altera.com/products/software/products/sopc/avalon/nio-avalon_features.html)>.
- [69] IBM. *The CoreConnect Bus Architecture*. Dezembro 2007. Online. Disponível em: <<http://www.ibm.com/chips/products/coreconnect>>.
- [70] Xilinx Inc. *Designing Custom OPB Slave Peripherals for MicroBlaze*. Fevereiro 2002. Online. Disponível em: <[http://www.xilinx.com/ipcenter/processor\\_central/microblaze/doc/opb\\_tutorial.pdf](http://www.xilinx.com/ipcenter/processor_central/microblaze/doc/opb_tutorial.pdf)>.
- [71] MORAES, F. G. et al. Nocgen - uma ferramenta para geração de redes intra-chip baseada na infra-estrutura hermes. In: *X Workshop IBERCHIP*. Cartagena, Colômbia: [s.n.], 2004. v. 1, p. 210–216.



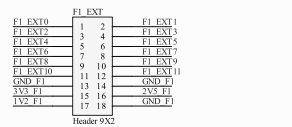
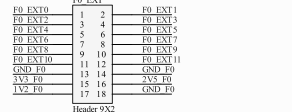
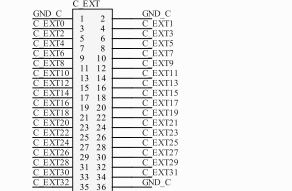
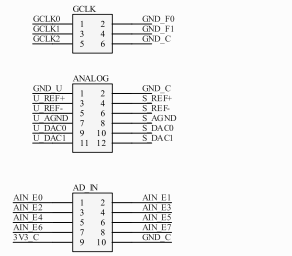
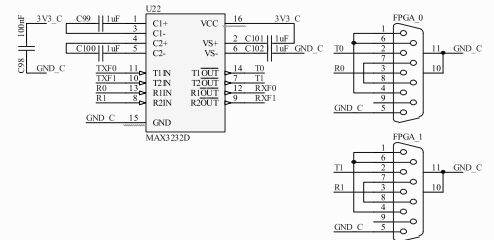
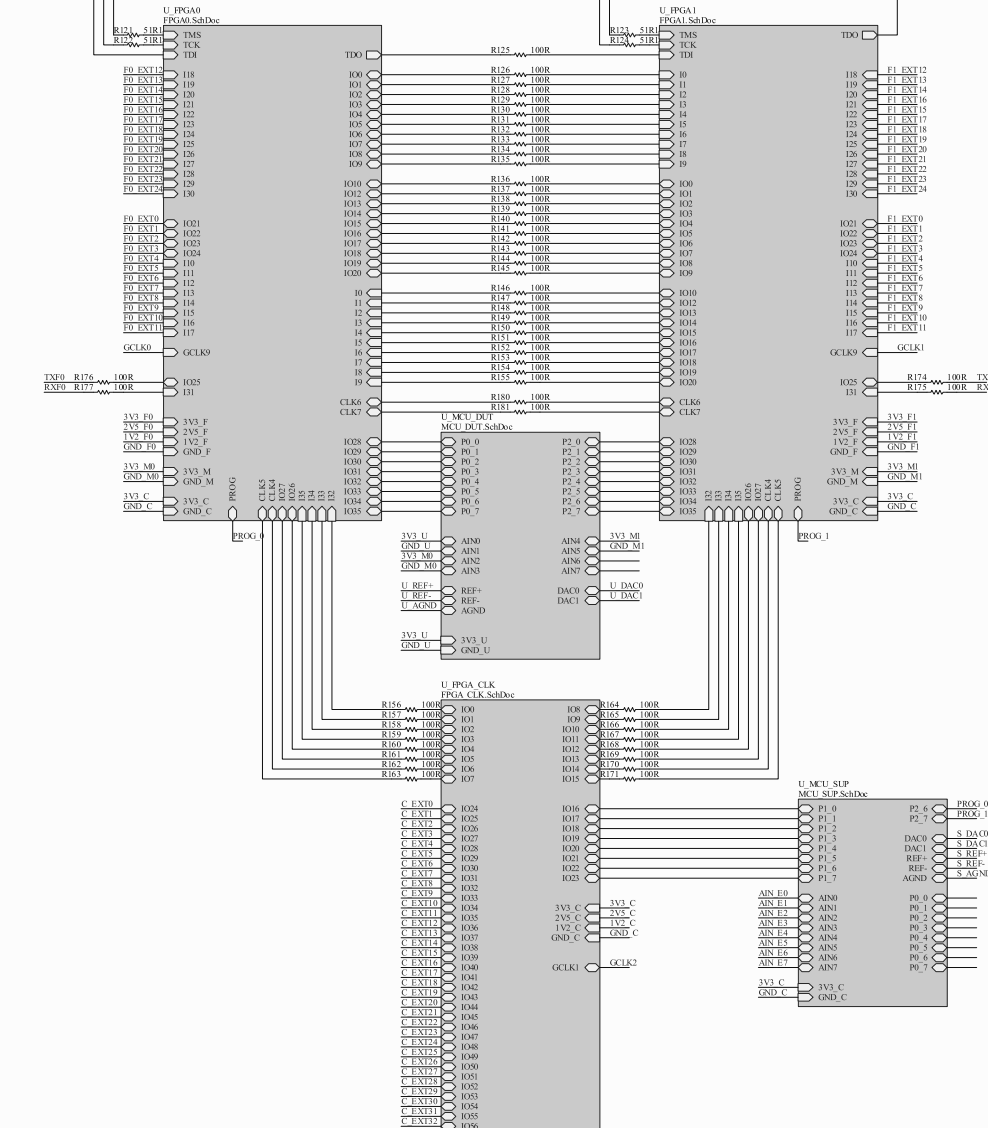
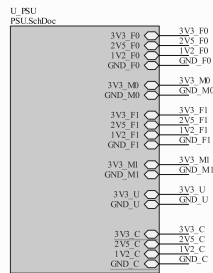
- [72] Xilinx Inc. *Xilinx Data Book*. Outubro 2008. Online. Disponível em: <<http://www.xilinx.com/support/documentation/index.htm>>.
- [73] ZEIDMAN, B. *Introduction to CPLD and FPGA Design*. 2001. Online. Disponível em: <<http://www.scribd.com/doc/11572884/Introduction-to-CPLD-and-FPGA-Design>>.
- [74] TOROK, D. L. *Projeto Visando a Prototipação do Protocolo de Acesso ao Meio em Redes Ethernet*. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2001. Faculdade de Informática, Programa de Pós-Graduação em Ciência da Computação.
- [75] ROSE, J.; GAMAL, A. E.; SANGIOVANNI-VINCENTELLI, A. Architecture of field programmable. In: *Proceedings of the IEEE*. [S.l.: s.n.], 1993. Vol. 81, p. 1013–1029.
- [76] ARAGÃO, A. C. O. S. *Uma Arquitetura Sistólica para Solução de Sistemas Lineares Implementada com Circuitos FPGA's*. Dissertação (Dissertação de Mestrado) — Universidade de São Paulo - USP, 1998. Instituto de Computação e Matemática Computacional.
- [77] Plessey Semiconductor. *ERA60100 Preliminary Data Sheet*. [S.l.]: Plessey Semiconductor, 1989.
- [78] Algotronix Ltd. *CAL 1024 Datasheet*. Edinburgh, Scotland: Algotronix Ltd., 1989.
- [79] ALTERA. *ACEX 1K, Embedded Programmable Logic Device Family Data Sheet*. [S.l.]: Altera, 2001. Versão 4.1.
- [80] Concurrent Logic. *CFA6006 Field-Programmable Gate Array Data Sheet*. [S.l.]: Concurrent Logic, 1991.
- [81] MUROGA, H. et al. A large scale fpga with 10k core cells with cmos 0.8 um 3-layered metal process. In: *Custom Integrated Circuits Conference - CICC'91*. [S.l.: s.n.], 1991. p. 6.4.1–6.4.4.
- [82] Atmel Corporation. *Data Acquisition Systems Using Cache Logic FPGAs*. [S.l.]: Atmel Corporation, 1999. Application Note.
- [83] GAMAL, A. E. et al. An architecture for electrically configurable gate arrays. *IEEE Journal of Solid-State Circuits* ., Vol. 24, p. 394–398, April 1989.
- [84] BIRKNER, J. et al. A very high-speed field programmable gate array using metal-to-metal anti-fuse programmable elements. In: *New Hardware Product Introduction at Custom Integrated Circuits Conference - CICC'91*. [S.l.: s.n.], 1991.
- [85] MARPLE, D.; COOKE, L. An mpga compatible fpga architecture. In: *ACM First International Workshop on Field Programmable Gate Array - FPGA 92*. [S.l.: s.n.], 1992. p. 39–44.
- [86] AMD Inc. *Mach Devices High Density EE Programmable Logic Data Book*. [S.l.]: AMD Inc, 1990. Data Book.
- [87] BAKER, S. Lattice fields fpga. In: *Eleci Fog Times*. [S.l.: s.n.], 1991. p. 1.

- [88] WONG, S. et al. A 5000-gate cmos epld with multiple logic and interconnect arrays. In: *Custom Integrated Circuits Conference - CICC '89*. Santa Clara, CA: Altera Corp, 1989. p. 5.8.1– 5.8.4.
- [89] LAVAGNO, L.; SANGIOVANNI-VINCENTELLI, L. *Algorithms for Synthesis and Testing of Asynchronous Circuits*. [S.l.]: Kluwer Academic Publishers, 1993. (Kluwer Academic Series in Engineering and Computer Science).
- [90] RENAUDIN, M. Asynchronous circuits and systems: a promising design alternative. *Microelectronic Engineering Journal*, v. 54, n. (1-2), p. 133–149, December 2000.
- [91] BERKEL, K. V.; JOSEPHS, M. B.; NOWICK, S. M. Scanning the technology: Applications of asynchronous circuits. In: *proceedings of the IEEE*. [S.l.: s.n.], 1999. v. 87, n. 2, p. 223–233.
- [92] HAUCK, S. Asynchronous design methodologies: An overview. In: *Proc. of the IEEE*. [S.l.: s.n.], 1995. v. 83, n. 1, p. 69–93.
- [93] DAVIS, A.; S.NOWICK. *An Introduction to Asynchronous Circuit Design*. [S.l.], September. Departement of Computer Science.
- [94] SUTHERLAND, I. Micropipelines. In: *Communication of the ACM*. [S.l.: s.n.], 1989. v. 32, n. 6.
- [95] MULLER, D.; BARTKY, W. A theory of asynchronous circuits. In: *Proc. Int. Symp. Theory of Switching*. [S.l.]: Harvard Univ. Press, 1959. p. 204–243.
- [96] GARSIDE, J. D. et al. Amulet3i - an asynchronous system-on-chip. In: *Proc. Sixth International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC 2000)*. Eilat, Israel: IEEE Computer Society Press, 2000. p. 162–175. ISBN 0769505864.
- [97] RENAUDIN, M.; VIVET, P.; ROBIN, F. ASPRO-216: A standard-cell QDI 16-bit RISC asynchronous microprocessor. In: *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*. [S.l.: s.n.], 1998. p. 22–31.
- [98] MARTIN, A. J. et al. The design of an asynchronous MIPS R3000 microprocessor. In: *Advanced Research in VLSI*. [S.l.: s.n.], 1997. p. 164–181.
- [99] MARTIN, A. J.; NYSTRÖM, M. Cast: Caltech asynchronous synthesis tools. In: *Proc. of Fourth Asynchronous Circuit Design Working Group Workshop*. Turku, Finland: [s.n.], 2004. Disponível em: <<http://www.async.caltech.edu/publications.html>>.
- [100] BERKEL, K. v. et al. The VLSI-programming language Tangram and its translation into handshake circuits. In: *Proc. European Conference on Design Automation (EDAC)*. [S.l.: s.n.], 1991. p. 384–389.
- [101] BARDSLEY, A.; EDWARDS, D. A. The Balsa asynchronous circuit synthesis system. In: *Forum on Design Languages*. Tübingen, Germany: [s.n.], 2000.
- [102] DUC, A. V. D. et al. Tast cad tools: Tutorial. In: *Int. Symp. on Advanced Research in Asynchronous Circuits and Systems ASYNC'02*. Manchester, UK: [s.n.], 2002. (TIMA internal report).

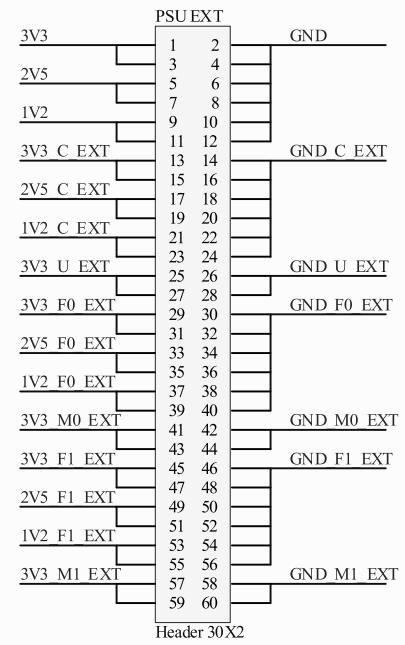
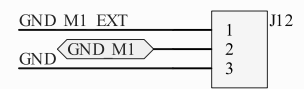
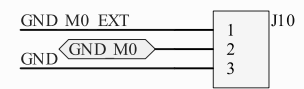
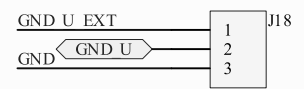
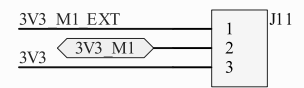
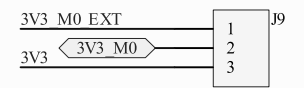
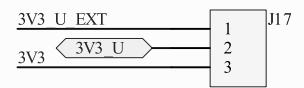
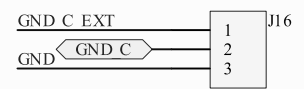
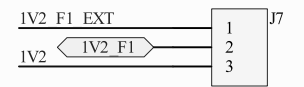
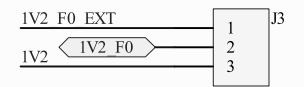
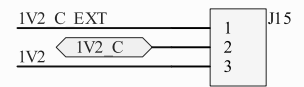
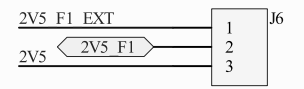
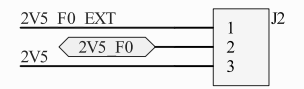
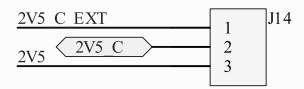
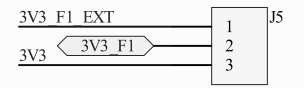
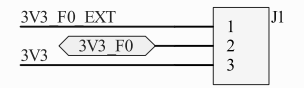
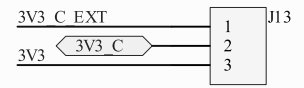
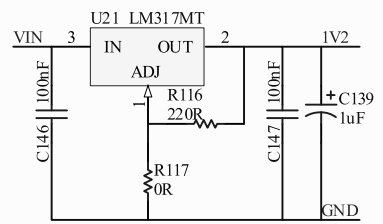
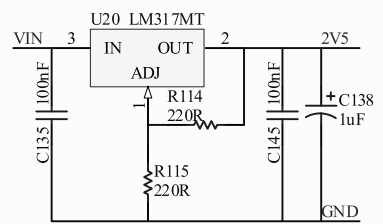
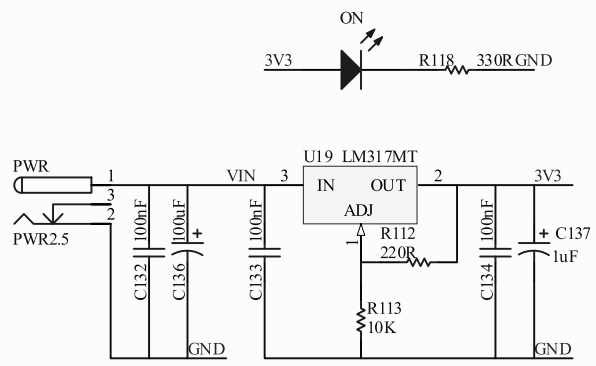
- [103] BLUNNO, I. et al. Handshake protocols for de-synchronization. In: *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*. [S.l.]: IEEE Computer Society Press, 2004. p. 149–158.
- [104] ANDRIKOS, N. *A Fully-Automated Desynchronization Flow for Synchronous Circuits*. Dissertação (Master's Thesis) — University of Crete, Heraklion, Greece, February 2006. P. 45.
- [105] MURATA, T. Petri nets: Properties, analysis and applications. In: *Proceedings of the IEEE*. [S.l.: s.n.], 1989. p. 541–580.
- [106] VLACHOS, E. *Study of asynchronous controllers' circuits in de-synchronized systems*. [S.l.], 2004. P. 337.
- [107] CHINNERY, D.; KEUTZER, K. Closing the gap between asic and custom: Tools and techniques for high-performance asic design. In: \_\_\_\_\_. [S.l.]: Kluwer Academic Publishers, 2002. cap. Reducing the timing overhead.
- [108] CORTADELLA, J. et al. Coping with the variability of combinational logic delays. In: *Proc. IEEE International Conference on Computer Design: VLSI in Computers and Processors ICCD 2004*. [S.l.: s.n.], 2004. p. 505–508.
- [109] Xilinx, Inc. *Spartan-3 FPGA Starter Kit Board User Guide*. 2008 June. Online. UG130 (v1.2). Disponível em: [http://www.xilinx.com/support/documentation/spartan-3\\_board\\_and\\_kit\\_documentation.htm](http://www.xilinx.com/support/documentation/spartan-3_board_and_kit_documentation.htm).
- [110] SEMIÃO, J. et al. Enhancing the tolerance to power-supply instability in digital circuits. In: *IEEE Computer Society Annual Symposium on VLSI - ISVLSI '07*. Porto Alegre, Brasil: [s.n.], 2007. ISBN: 0-7695-2896-1.
- [111] SEMIÃO, J. et al. Pimproving tolerance to power-supply and temperature variations in synchronous circuits. In: *IEEE Design and Diagnostics of Electronic Circuits and Systems - DDECS'07*. Krakow, Poland: [s.n.], 2007. ISBN: 1-4244-1162-9.
- [112] SPARSØ, J.; STAUNSTRUP, J. Delay-insensitive multi-ring structures. *Integr. VLSI J.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 15, n. 3, p. 313–340, 1993. ISSN:0167-9260.
- [113] KOENEMAN, J. M. B.; ZWIEHOFF, G. Built-in logic block observation techniques. In: *Proc. IEEE Int Test Conf*. [S.l.: s.n.], 1979. p. 37–41.
- [114] SOTIRIOU, C. et al. *Asynchronous Open-Source DLX Processor (ASPIDA)*. 2004. Online. Asynchronous DLX Demo at ASYNC 2004 Conference. Disponível em: <http://www.ics.forth.gr/carv/async/demo/>.
- [115] SOTIRIOU, C. *ASPIDA sync/async DLX Core: DLX ISA CPU with SYNCHRONOUS and ASYNCHRONOUS Implementations*. Sep 2005. Online. Disponível em: <http://www.opencores.org/projects.cgi/web/aspida/overview>.
- [116] HENNESY, J.; PATTERSON, D. *Computer Architecture: A Quantitative Approach*. San Mateo, CA: Morgan Kaufmann, 1990.

- [117] MIPS Technologies, Inc. *MIPS Technologies*. 2008. Online. Disponível em: <<http://www.mips.com/>>.
- [118] Opencores Community. *Opencores*. 2008. Online. Disponível em: <<http://www.opencores.org>>.
- [119] LOPES, D. C. *Estimação da Robustez de Sistemas Eletrônicos Via Injeção de Falhas por Interferências Eletromagnéticas*. 97 p. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2005. Programa de Pós-Graduação em Engenharia Elétrica, Faculdade de Engenharia - Dissertação de Mestrado.
- [120] ROCHA, C. A. *Monitoramento do Fluxo de Controle de Processadores Embarcados Baseado em Profiling de Software*. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2007. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.
- [121] PICCOLI, L. B. *Soluções Híbridas de Hardware/Software para a Detecção de Erros em Systems-on-Chip (Soc) de Tempo Real*. 160 p. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2006. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.
- [122] GREEN, L. Simulation, modeling and understanding the importance of signal integrity. *IEEE Circuits and Devices Magazine*, v. 15, n. 6, p. 7–10, Novembro 1999. ISSN: 8755-3996.
- [123] ROCHA, L. B. *Aplicação de Computação Móvel na Operação e Manutenção de Sistemas de Distribuição de Energia Elétrica*. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2007. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.
- [124] TEIXEIRA JOÃO PAULO CACHO, e. a. Aumento de tolerância dinâmica de circuitos eletrônicos integrados digitais a variações de tensão de alimentação e de temperatura. In: . [S.l.: s.n.], 2006.

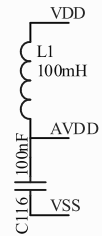
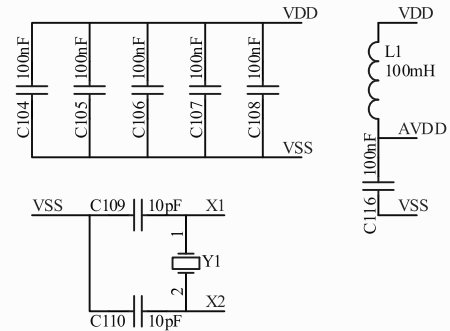
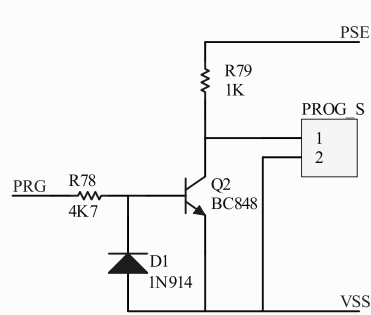
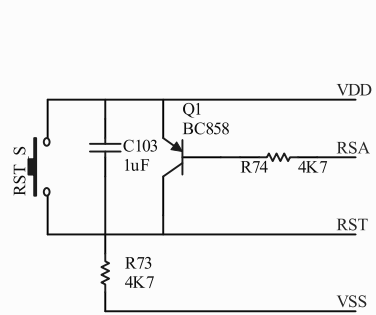
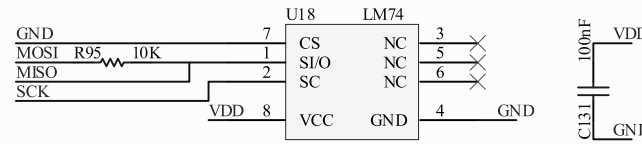
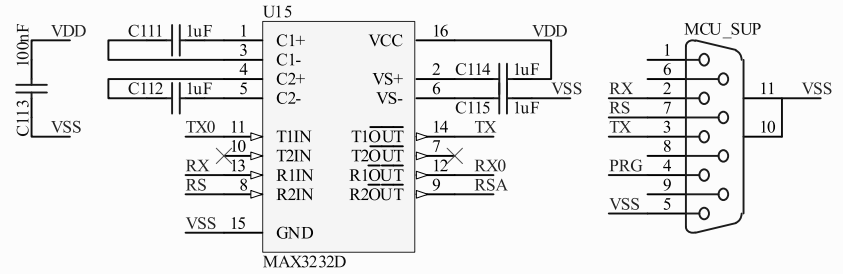
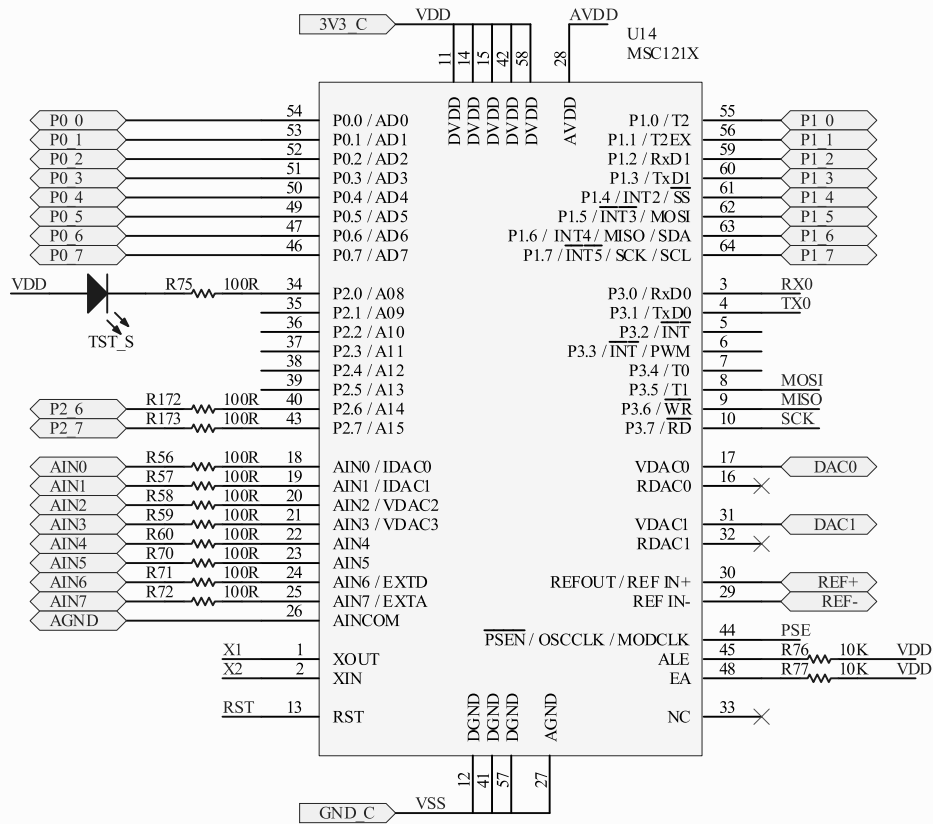
*ANEXO A - Esquemático da plataforma de  
ensaio*



Title		
Size	Number	Revision
A2		
Date	7/1/2007	Sheet of
File	D:\MDS\PCB\place\TOP_SCH\SchDoc	Drawn By

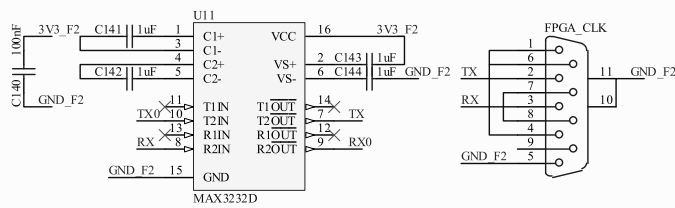
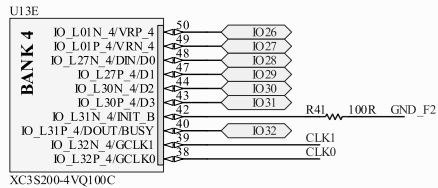
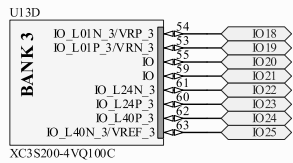
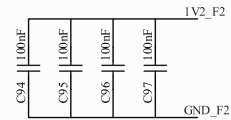
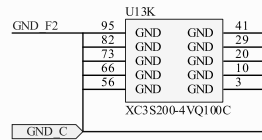
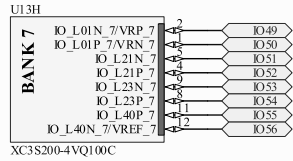
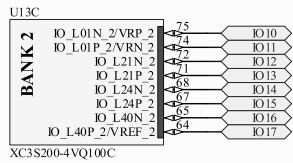
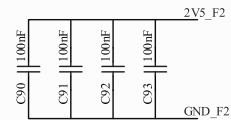
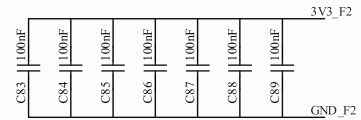
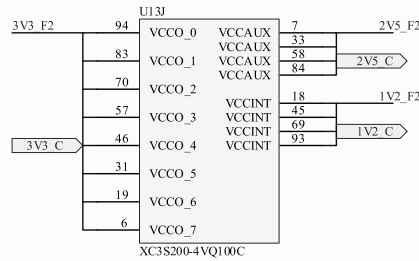
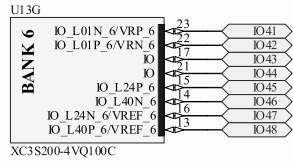
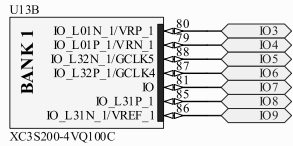
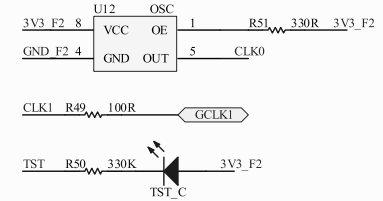
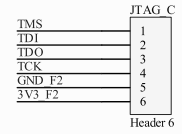
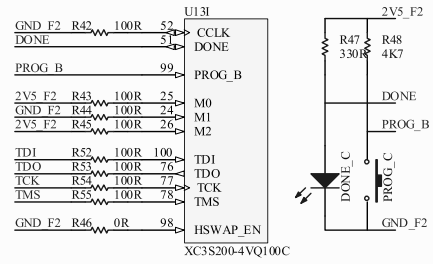
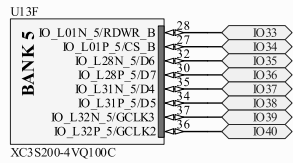
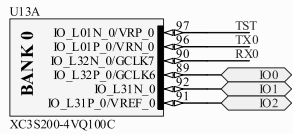


Title		
Size	Number	Revision
A4		
Date:	7/11/2007	Sheet of
File:	D:\MDS\IS\placa\PSU.SchDoc	Drawn By:

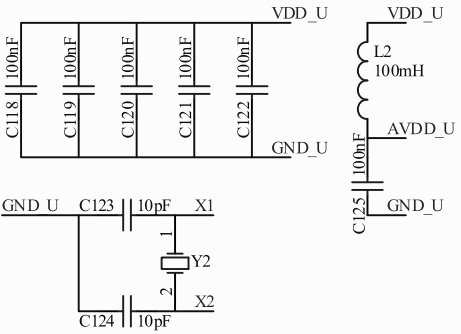
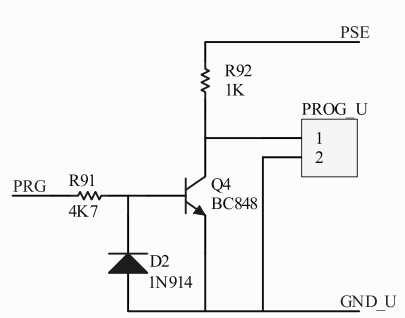
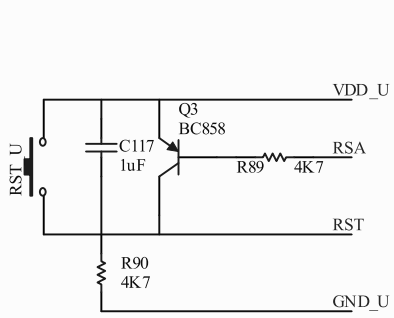
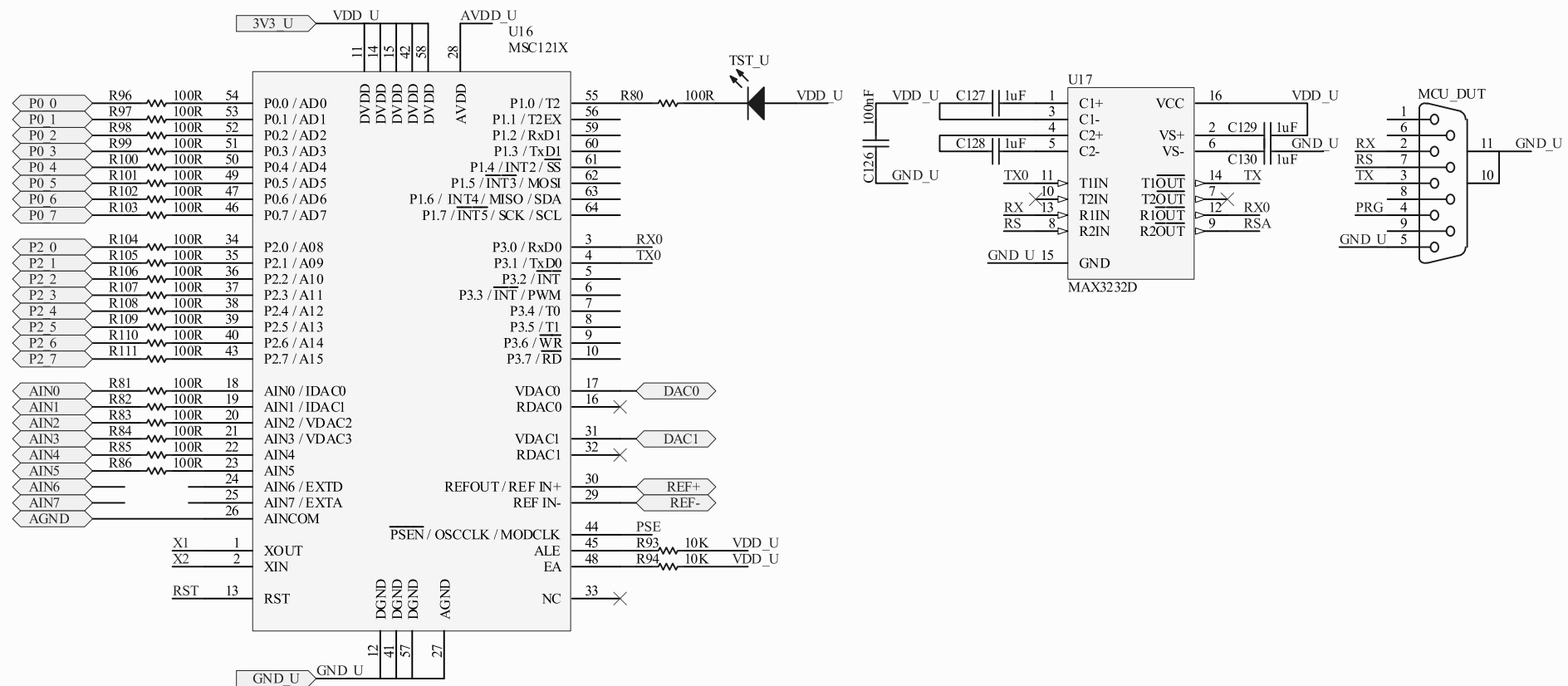


Title		
Size	Number	Revision
A4		
Date:	7/11/2007	Sheet of
File:	D:\MDS\IC\placa\MCU_SUP.SchDoc	Drawn By:

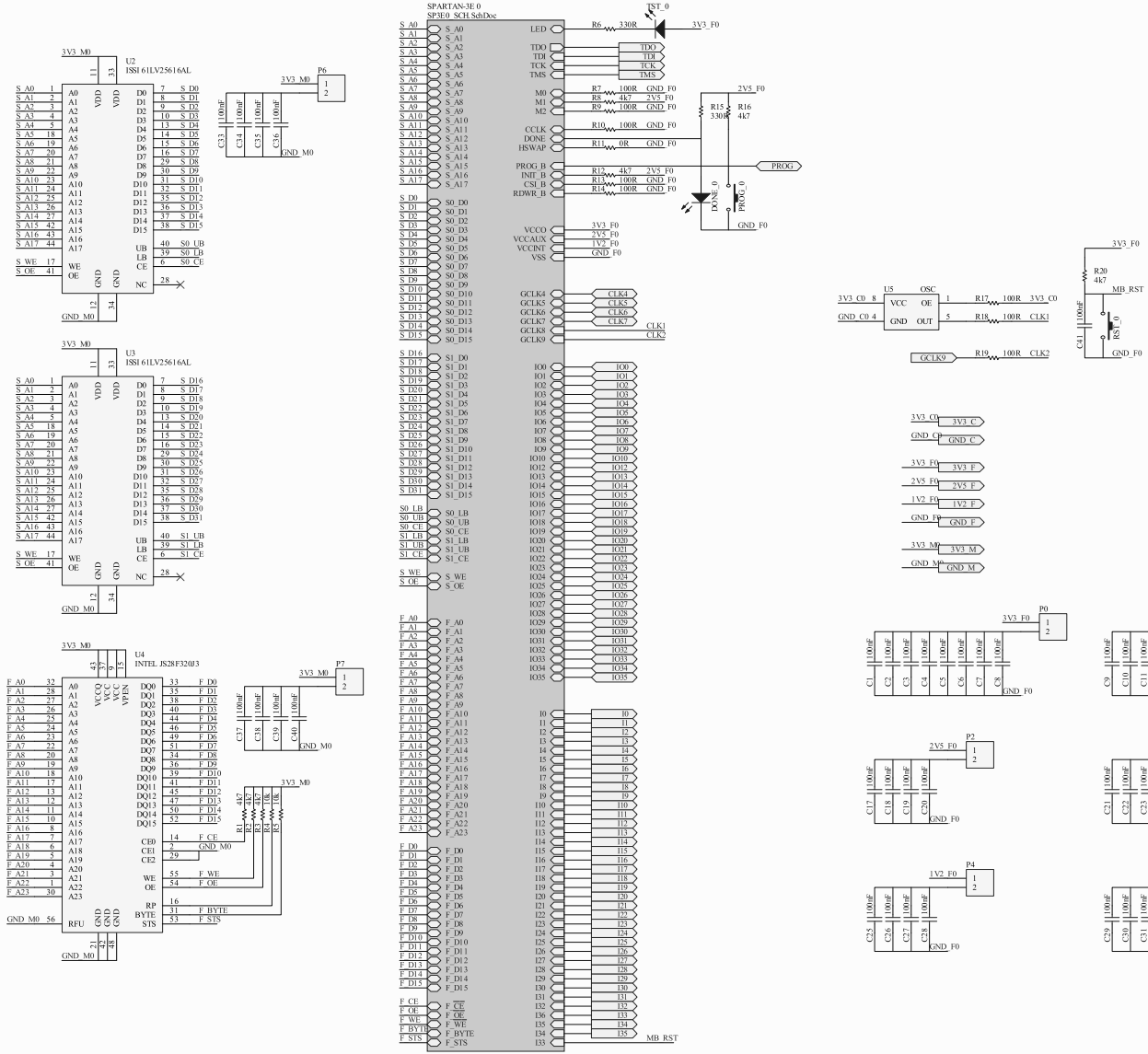




Title		
Size	Number	Revision
A3		
Date:	7/1/2007	Sheet of
File:	D:\MD\SiSC\placa\FPGA_CLK.SchDoc	Drawn By:

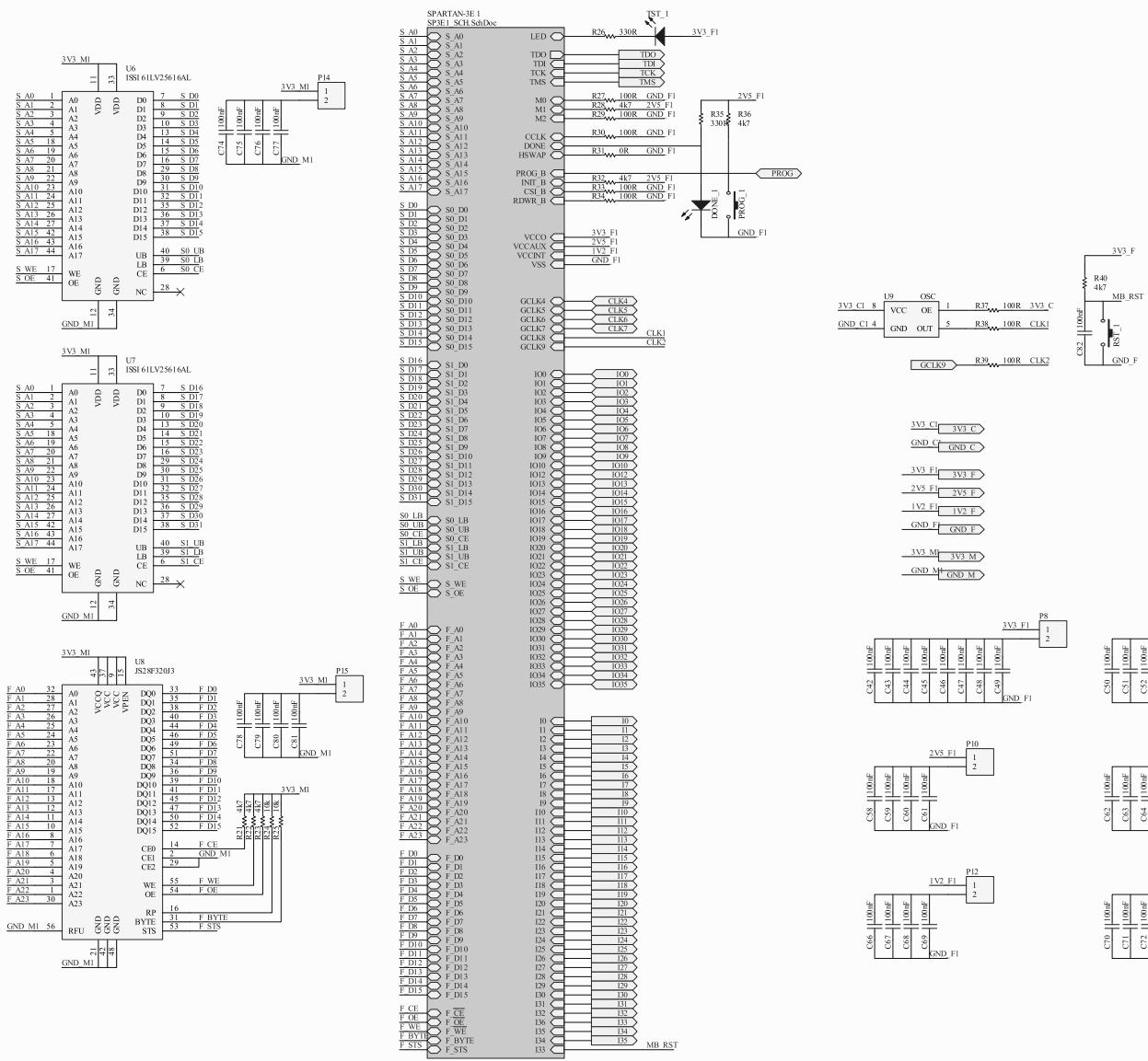


Title		
Size	Number	Revision
A4		
Date:	7/11/2007	Sheet of
File:	D:\MDS\SC\placa\MCU_DUT.SchDoc	Drawn By:



TDO, TDI, TCK, TMS  
 GCLK4, 5, 6  
 Clock Global da FPGA  
 PROG.  
 IO0, IO35  
 Pinos de Input-Output  
 B.D2  
 Pinos de Input-Output  
 3V3\_C, GND\_C  
 VCC e GND limpos para o oscilador  
 3V3\_F, 2V5\_F, 1V2\_F, GND\_F  
 Alimentação da FPGA  
 3V3\_M, GND\_M  
 ITAG  
 Clock Global da FPGA  
 Reconfigura FPGA  
 IO0, IO35  
 Pinos de Input-Output  
 Pinos de Input-Output  
 VCC e GND limpos para o oscilador  
 Alimentação da FPGA  
 Alimentação das memórias

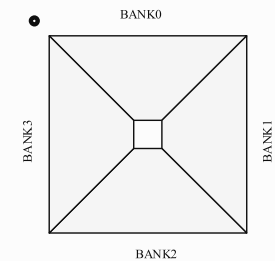
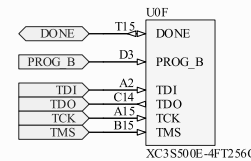
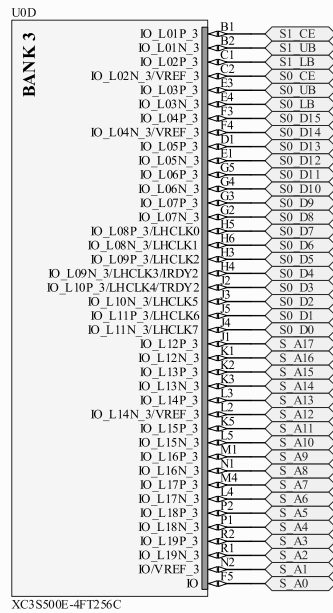
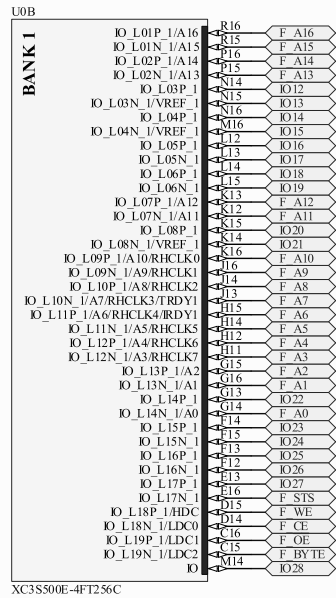
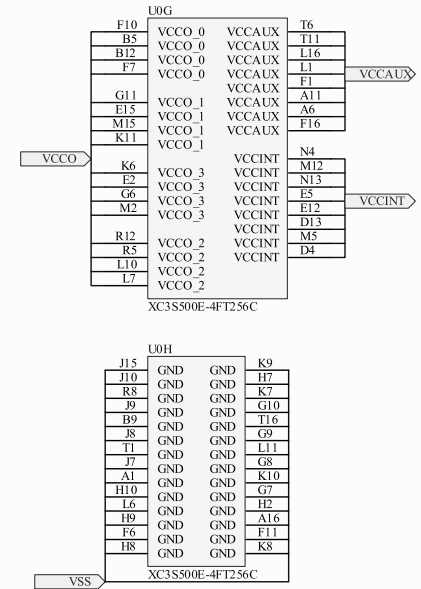
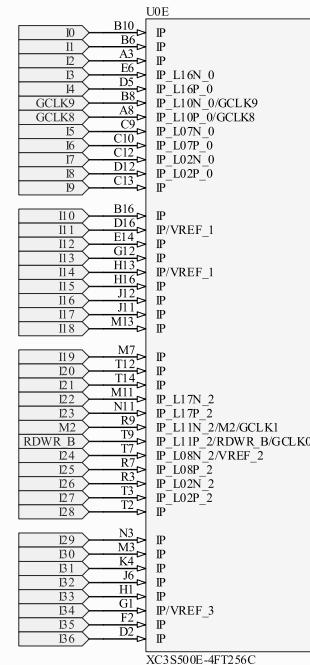
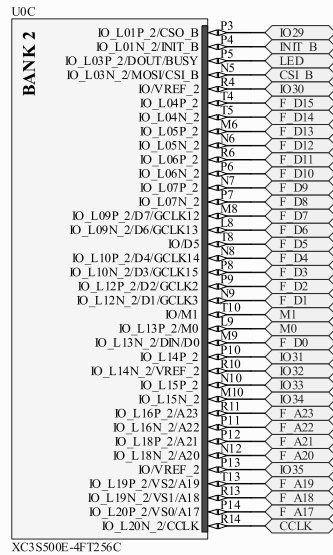
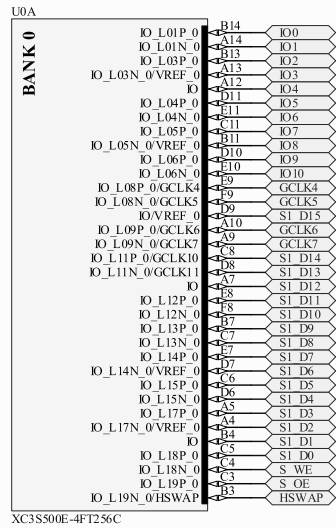
Title		
Size	Number	Revision
A2	BLOCO FPGA 0	
Date	7/11/2007	Sheet of
File	D:\MDSIS\placa\FPGA0_SchDoc	Drawn By: MD

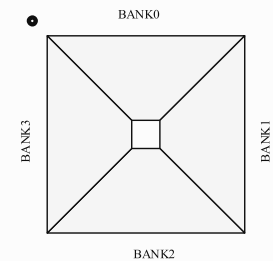
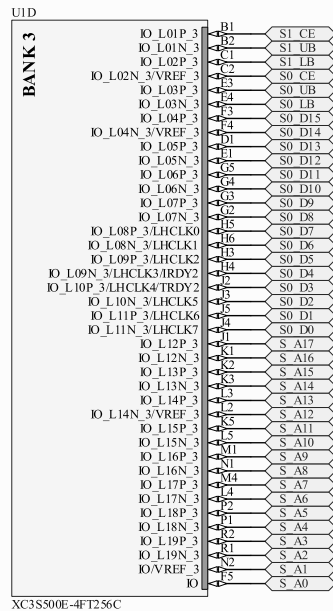
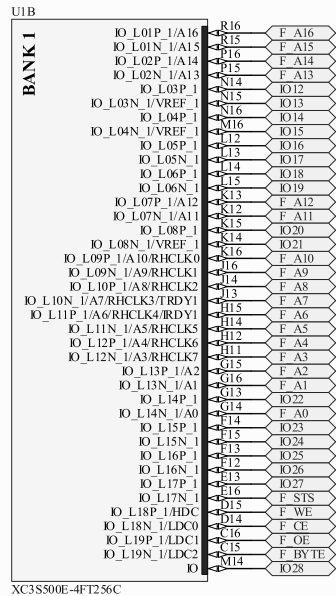
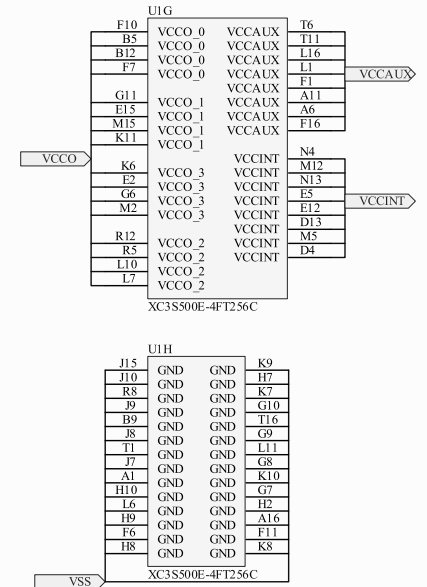
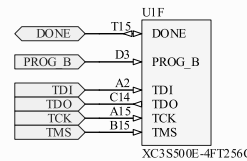
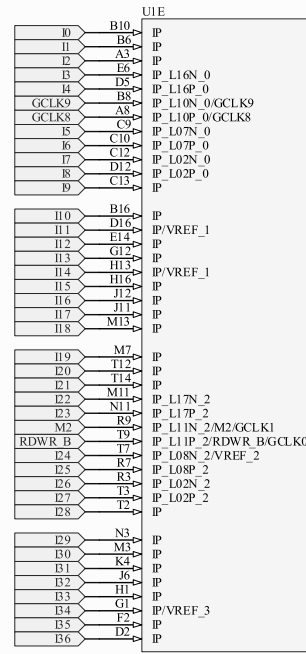
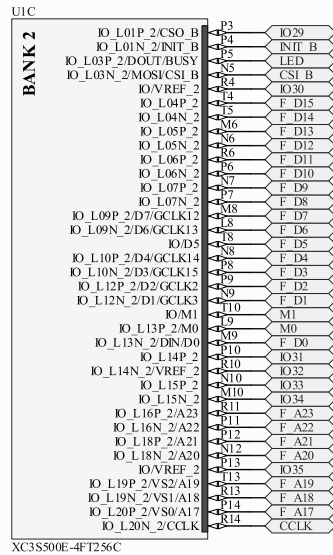
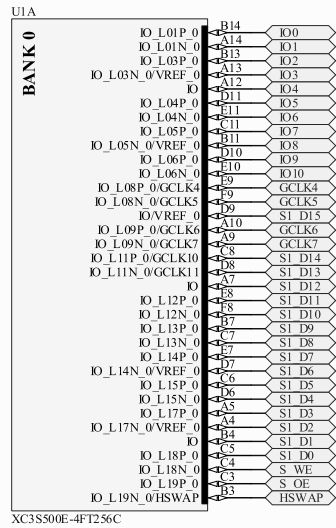


TDO, TDI, TCK, TMS  
GCLK4, 5, 6  
PROG  
IO0, IO35  
B.D2  
3V3\_C, GND\_C  
3V3\_F, 2V5\_F, 1V2\_F, GND\_F  
3V3\_M, GND\_M

ITAG  
Clock Global da FPGA  
Reconfigura FPGA  
Pinos de Input-Output  
Pinos de Input-Output  
VCC e GND limpos para o oscilador  
Alimentação da FPGA  
Alimentação das memórias

Title			
Size	Number	Revision	
A2	BLOCO FPGA 1		
Date	7/1/2007	Sheet of	
File	D:\MDSIS\placa\FPGA1_SchDoc	Drawn By: MD	





*ANEXO B - Esquemático da placa de  
alimentação e injeção de falhas*

