

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

**MARLON LEANDRO MORAES**

**VALIDAÇÃO DE UMA TÉCNICA PARA O AUMENTO  
DA ROBUSTEZ DE SoC's A FLUTUAÇÕES DE  
TENSÃO NO BARRAMENTO DE ALIMENTAÇÃO**

**PORTO ALEGRE  
2008**

**MARLON LEANDRO MORAES**

**VALIDAÇÃO DE UMA TÉCNICA PARA O AUMENTO  
DA ROBUSTEZ DE SoC's A FLUTUAÇÕES DE  
TENSÃO NO BARRAMENTO DE ALIMENTAÇÃO**

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

**Orientador: Prof. Dr. Fabian Luis Vargas**

PORTO ALEGRE  
2008

# **VALIDAÇÃO DE UMA TÉCNICA PARA O AUMENTO DA ROBUSTEZ DE SoC's A FLUTUAÇÕES DE TENSÃO NO BARRAMENTO DE ALIMENTAÇÃO**

**MARLON LEANDRO MORAES**

Esta dissertação foi julgada para a obtenção do título de MESTRE EM ENGENHARIA e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul.

---

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

---

Prof. Dr. Rubem Dutra Fagundes – PUCRS

---

Prof. Dr. Eduardo Augusto Bezerra – PUCRS

# Dedicatória



*Este trabalho é dedicado às pessoas que sempre torceram incondicionalmente pela minha felicidade e sucesso, que festejaram e vibraram com todas as minhas conquistas e vitórias, mas que também sofreram e choraram ao meu lado em todas as minhas derrotas.*

*Esta dissertação é dedicada vocês, Leoni & Sandra, meus queridos e tão amados pais.*

# Agradecimentos

*Agradeço aos meus pais, Leoni e Sandra, pelo apoio incondicional nesta e em todas as outras escolhas da minha vida;*

*À minha irmã, Thaiana, pelo incentivo, carinho e bom humor nos momentos de angústia e stress;*

*À minha namorada, Denise, pela compreensão, paciência e amor, sobretudo nestes tempos atribulados de final de mestrado.*

*Aos meus colegas, e acima de tudo, grandes amigos do Laboratório de Sistemas, Sinais & Computação - SiSC, pelas inúmeras e tão decisivas contribuições para esta dissertação, e também, por tornarem estes dois anos de trabalho, esforço e dedicação tão prazerosos, quanto uma das nossas Confrarias SiSCanas.*

*Aos amigos “butequeiro” do Laboratório de Ensino e Pesquisa – LEP, Edison, Carlos e Andriago, pelas placas e pelos longos e flexíveis prazos nos empréstimos de equipamentos.*

*Ao professor e amigo Júlio Cesar Marques de Lima, pelos conselhos sinceros e soldas mais que perfeitas.*

*Ao senhor Steve Rhoad, desenvolvedor do processador Plasma, pelo rápido e preciso suporte técnico.*

*Ao professor Fabian Luis Vargas pelas oportunidades de aprendizagem e crescimento profissional.*

*À toda a equipe do Instituto Nacional da Tecnologia Industrial – INTI pela atenção e disponibilidade de equipamentos, fundamentais para a realização dos ensaios desta dissertação em Buenos Aires – Argentina.*

*À toda a equipe do INESC de Portugal, em especial ao professor Jorge Semião, pelas informações absolutamente fundamentais para a execução deste trabalho.*

*Ao Programa Nacional de Microeletrônica – CNPq, pelo financiamento desta etapa tão importante da minha vida.*

*A todos vocês que de alguma forma contribuíram para a execução deste trabalho, o meu mais sincero MUITO OBRIGADO!!!*

# Resumo

Tendo em vista que o barramento de alimentação ( $V_{CC}$  e Gnd) afeta diretamente a integridade de sinal de sistemas em *chip* (*Systems-on-Chip*, *SoC*) através de oscilações de tensão que podem induzir a erros funcionais, este trabalho tem por objetivo validar uma técnica inovadora, denominada CDCDC (*Clock Duty Cycle Dynamic Control*). Esta técnica visa aumentar a robustez de circuitos integrados (CI) digitais síncronos a tais oscilações de tensão.

A técnica em questão realiza o controle dinâmico do ciclo de trabalho (*duty-cycle*) do sinal de relógio (*clock*) de acordo com a presença de perturbações (ruídos) nas linhas de alimentação.

Este controle dinâmico do sinal de relógio realiza o prolongamento ou a redução do ciclo de trabalho, permitindo assim que o circuito síncrono apresente uma maior robustez às flutuações dos níveis de tensão nas linhas de alimentação, sem que haja redução da frequência do sinal de relógio. Garante-se desta forma, a manutenção do desempenho do sistema mesmo quando este estiver operando em ambientes expostos ao ruído.

Considerando que a interferência eletromagnética (EMI) é uma das principais causas de oscilações no barramento de alimentação de circuitos integrados (CI's), o que por sua vez compromete drasticamente a confiabilidade dos sistemas através da redução da margem de sinal/ruído, este trabalho tem por objetivo validar a utilização da técnica CDCDC para o aumento da robustez de CI's operando expostos à EMI.

# Abstract

Considering that the power-supply bus ( $V_{CC}$  and Gnd) can degrade signal integrity of Systems-on-Chip (SoC) by means of voltage oscillations that conduct to functional failure, the goal of this work is validate a new technique, named CDCDC (Clock Duty Cycle Dynamic Control), which aims at improving digital synchronous integrated circuits (IC) robustness to this kind of disruption.

The considered technique performs the dynamic control of the clock duty cycle according to the presence of perturbations (noise) on the power-bus.

This dynamic control of the clock signal performs the stretching or shrinking of the duty cycle, which allows the synchronous circuit to present higher robustness to power-supply fluctuations, while maintaining the same clock pace. Conversely, conventional approaches, in this case, use to reduce clock rate to ensure proper signal integrity.

Considering that the electromagnetic interference (EMI) is one of the main sources of voltage oscillations in the power-bus of integrated circuits (ICs), which in turn may compromise the reliability of electronic systems due to the reduction of signal-to-noise ratio, this work aims at validating the use of the CDCDC technique to increase the robustness of ICs operating under the EMI exposition.

# Lista de Figuras

Figura 2.1 - Modelo dos três universos (7).....	22
Figura 2.2 - Conceito de latência de falha e latência de erro.....	22
Figura 2.3 - Notação e emulação do modelo de falha <i>Stuck-at</i> (12). ....	25
Figura 2.4 - Modelo de falha <i>Transistor-Level Stuck</i> (12). ....	25
Figura 2.5 - Modelo de falha <i>bridging</i> (12).....	26
Figura 2.6 - Curva da banheira (7). ....	28
Figura 3.1 - Exemplo de Interferência Eletromagnética conduzida (19). ....	33
Figura 3.2 - Exemplo de Interferência Eletromagnética irradiada (19).....	33
Figura 3.3 - Exemplos de acoplamentos de EMI conduzida e irradiada (21). ....	34
Figura 3.4 - Exemplo de Integridade Sinais (23).....	35
Figura 3.5 - Retificador de monofásico (3). ....	37
Figura 3.6 - Gerador de teste com fontes de alimentação de chaveamento interno (4).....	39
Figura 3.7 - Gerador de teste baseado em uma fonte de alimentação programável (4). ....	39
Figura 3.8 - Características de placa de circuito impresso de teste (2). ....	41
Figura 3.9 - Disposição das camadas de uma placa de teste (2).....	41
Figura 3.10 - Exemplos sinais de onda contínua e amplitude modulada (2).....	43
Figura 3.11 - <i>Setup</i> de teste com células TEM e GTEM (33). ....	45
Figura 3.12 - <i>Setup</i> de teste de imunidade conduzida (35).....	46
Figura 3.13 - Princípio de injeção de distúrbios RF em um pino (35). ....	47
Figura 3.14 - Princípio de injeção de distúrbios RF em mais de um pino (35).....	47
Figura 3.15 - Estrutura da placa de teste para testes de injeção de RF (35).....	48
Figura 3.16 - Placa de teste para CI's com baixo número de pinos (35).....	49
Figura 3.17 - Placa de teste para CI's com alto número de pinos (35).....	49
Figura 4.1 - Arquitetura básica de um FPGA(38). ....	52
Figura 5.1 - Diminuição do ciclo de vida dos produtos eletrônicos (63). ....	56
Figura 5.2 - Arquitetura genérica de um SoC (70) ....	57
Figura 5.3 - Estrutura genérica de interconexão baseada em fios ponto-a-ponto (70).....	60
Figura 5.4 - Estrutura de interconexão em barramento com arbitro central (70). ....	61
Figura 6.1 - Exemplo de falha de atraso entre dois sinais. ....	63
Figura 6.2 - Cadeia de 77 inversores com variação de tensão de alimentação (88).....	66
Figura 6.3 - Arquiteturas básica do sistema CDCM (88).....	69
Figura 6.4 - Exemplo de arquitetura para o bloco BIPS (88). ....	69
Figura 6.5 - Exemplo de arquitetura para o bloco CSL (88). ....	70
Figura 6.6 - Simulações do bloco CSL para variações de $V_{CC}$ (89). ....	71
Figura 7.1 - Lógica do circuito descrito em VHDL “Clock Manager”. ....	74
Figura 7.2 - Fluxograma do circuito de teste B02. ....	76
Figura 7.3 - Fluxograma do circuito de teste B02 – Modificado. ....	77
Figura 7.4 - Kit de desenvolvimento <i>Spartan-3 Starter Kit Board</i> (91).....	79
Figura 7.5 - Diagrama de blocos do <i>setup</i> de teste completo. ....	79
Figura 7.6 - Lógica de alimentação da plataforma de desenvolvimento. ....	80
Figura 7.7 - Lógica de alimentação modificada da plataforma de desenvolvimento. ....	81
Figura 7.8 - Saídas do circuito B02 – Modificado (borda de subida). ....	83
Figura 7.9 - Saídas do circuito B02 – Modificado (borda de descida).....	84

Figura 7.10 - Saídas coerentes do circuito B02 - Modificado. ....	85
Figura 8.1 - Composição da placa de circuito impresso. ....	88
Figura 8.2 - Esquemático genérico da plataforma de testes. ....	89
Figura 8.3 - Vista inferior da plataforma de testes. ....	90
Figura 8.4 - Vista superior da plataforma de testes. ....	90
Figura 8.5 - Diagrama da placa de alimentação e injeção de falhas. ....	91
Figura 8.6 - Placa de alimentação e injeção de falhas. ....	92
Figura 8.7 - Diagrama temporal de escrita nos conversores digitais-analógicos. ....	92
Figura 8.8 - Diagrama de blocos da arquitetura básica do processador Plasma (95). ....	94
Figura 8.9 - Divisão de frequência do processador Plasma. ....	95
Figura 8.10 - Instanciamento da divisão de frequência no circuito <i>Clock Manager</i> . ....	95
Figura 8.11 - Saída de relógio de saída do circuito <i>Clock Manager</i> . ....	96
Figura 8.12 - Hierarquia de arquivos do processador Plasma. ....	97
Figura 8.13 - Estrutura hierarquica do circuito Sistema. ....	98
Figura 8.14 - Lógica de ajuste dos tempos dos <i>softwares</i> de teste. ....	99
Figura 9.1 - Teste de redução dos níveis de tensão de alimentação. ....	103
Figura 9.2 - Configurações dos testes de redução dos níveis de tensão de alimentação. ....	104
Figura 9.3 - Seqüência de procedimentos do teste de redução de tensão de alimentação. ....	106
Figura 9.4 - Gráfico comparativo do número de ocorrência de falhas. ....	107
Figura 9.5 - Teste de Interferências Eletromagnéticas Irradiadas. ....	108
Figura 9.6 - Plataforma de testes e caixa metálica inseridas na célula TEM. ....	109
Figura 9.7 - Configuração dos testes de EMI Irradiada (14). ....	109
Figura 9.8 - Seqüência de procedimentos dos testes de EMI Irradiada. ....	112
Figura 9.9 - Gráfico comparativo do número de ocorrência de falhas. ....	113

# Lista de Tabelas

Tabela 2.1 - Causas usuais em sistemas computacionais (10). .....	24
Tabela 2.2 - Medidas de dependabilidade (7).....	27
Tabela 3.1 - Níveis de Tensão de Teste (10). .....	36
Tabela 3.2 - Níveis de tensão e duração recomendadas para interrupções (4).....	38
Tabela 3.3 - Níveis de tensão e duração recomendados para quedas de tensão (4). .....	38
Tabela 3.4 - Níveis de tensão e duração recomendados para variação de tensão (4).....	38
Tabela 3.5 - Conteúdo das camadas da placa de circuito impresso.....	41
Tabela 3.6 - Conexão dos pinos do circuito integrado (2).....	42
Tabela 3.7 - Intervalos de frequência aplicados nos testes (2). .....	43
Tabela 3.8 - Níveis de potência de injeção (35). .....	47
Tabela 7.1 - Descrição funcional dos pinos do circuito <i>Clock Manager</i> . .....	75
Tabela 7.2 - Combinações de entrada do circuito <i>Clock Manager</i> . .....	75
Tabela 7.3 - Características do Circuito Teste B02. ....	76
Tabela 7.4 - Características do Circuito Teste B02 – Modificado. ....	77
Tabela 7.5 - Redução percentual da tensão de entrada.....	81
Tabela 7.6 - Redução percentual da tensão do núcleo.....	82
Tabela 8.1 - Descrição funcional dos pinos do circuito de Alimentação. ....	93
Tabela 8.2 - Combinações de entrada e saída do circuito de Alimentação. ....	93
Tabela 8.3 - Arquivos e funções do processador Plasma (95).....	97
Tabela 8.4 - Nomenclatura adotadas para lógica de ajuste dos tempos. ....	100
Tabela 8.5 - Descrição dos comandos do <i>software</i> de gerenciamento de teste. ....	100

# Lista de Siglas

ALU – Arithmetic and Logic Unit  
AM – Amplitude Modulation  
ANSI – American National Standards Institute  
ASIC – Application Specific Integrated Circuit  
BCD – Binary Coded Decimal  
BCI – Bulk Current Injection  
BIPS – Built-In Power supply voltage transient Sensor  
CDCDC – Clock Duty Cycle Dynamic Control  
CDCM – Clock Duty-Cycle Modulation  
CF – Curve Fitting  
CI – Circuito Integrado  
CLB – Configurable Logic Block  
CMOS – Complementary Metal-Oxide-Semiconductor  
CPLD – Complex Programmable Logic Device  
CRR – Circuito de Reconstituição de Relógio  
CSL – Clock Stretching Logic  
CW – Continuous Wave  
DAC – Digital to Analog Converter  
DC – Direct Current  
DDR SDRAM – Double-Data-Rate Synchronous Dynamic Random Access Memory  
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 & 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

WA – Work Always

WN – Work when Needed

# Sumário

Parte I - Fundamentos Teóricos.....	16
1. Introdução.....	17
1.1. Motivação.....	17
1.2. Objetivos.....	17
1.3. Apresentação dos Capítulos.....	18
2. Conceitos de Tolerância a Falhas.....	20
2.1. Introdução.....	20
2.2. Falha, Erro ou Defeito.....	21
2.2.1. Latência.....	22
2.2.2. Classificação de Falhas.....	23
2.3. Defeitos e Modelos de Falhas.....	24
2.4. Medidas Relacionadas ao Tempo Médio de Funcionamento.....	27
3. Compatibilidade Eletromagnética.....	29
3.1. Introdução.....	29
3.2. Conceitos Básicos.....	30
3.3. Ambiente Eletromagnético.....	31
3.4. Interferência Eletromagnética.....	31
3.4.1. Fontes de Interferências Eletromagnéticas.....	32
3.4.2. Interferência Eletromagnética Conduzida versus Irradiada.....	32
3.4.3. Acoplamento de Interferência Eletromagnética.....	33
3.5. Efeitos da Interferência Eletromagnética na Eletrônica.....	34
3.6. Normas IEC.....	36
3.6.1. Norma IEC 61.000-4-17.....	36
3.6.2. Norma IEC 61.000-4-29.....	37
3.6.3. Norma IEC 62.132.....	39
4. Tecnologia dos Dispositivos Reconfiguráveis.....	50
4.1. Introdução.....	50
4.2. Arquitetura dos FPGA's.....	51
5. System on Chip (SoC).....	55
5.1. Introdução.....	55
5.2. Linguagens para Descrição de SoC's.....	56
5.3. Arquitetura Genérica de um SoC.....	57
5.4. Interconexão de Núcleos Baseada em Fios Ponto-a-Ponto Dedicados.....	59
5.5. Interconexão de Núcleos Baseada em Barramentos Compartilhados.....	60
6. Controle Dinâmico do Ciclo de Trabalho do Sinal de Relógio (CDCDC).....	63
6.1. Introdução.....	63
6.2. Modulação do Ciclo de Trabalho.....	65
6.3. Descrição da Técnica CDCDC.....	66
Parte II - Metodologia.....	72
7. Estudo de Caso I.....	73
7.1. Introdução.....	73
7.2. Circuito Clock Manager.....	73
7.3. Circuito de Teste B02.....	75

7.4.	Circuito de Teste B02 – Modificado .....	76
7.5.	Plataforma Spartan-3 Starter Kit Board.....	78
7.6.	<i>Setup</i> de Teste Completo .....	79
7.7.	Níveis de Tensão da Plataforma de Desenvolvimento .....	80
7.8.	Redução da Tensão de Entrada.....	80
7.9.	Redução da Tensão do Núcleo do FPGA .....	81
7.10.	Experimento com o B02 – Modificado (borda de subida) .....	82
7.11.	Experimento com o B02 – Modificado (borda de descida).....	83
7.12.	Conclusões.....	85
8.	Estudo de Caso II.....	87
8.1.	Introdução .....	87
8.2.	Plataforma de Testes.....	87
8.3.	Placa de Alimentação e Injeção de Falhas.....	91
8.4.	Circuito de Alimentação .....	92
8.5.	Processador Plasma .....	93
8.6.	Modificações Estruturais no Processador Plasma .....	94
8.6.1.	Adição do Sinal de Relógio com Ciclo de Trabalho Variável .....	94
8.6.2.	Remoção da Divisão de Frequência .....	95
8.6.3.	Alteração do Ciclo de Trabalho do Bloco Multiplicador .....	96
8.7.	Sistema de Gerenciamento de Testes .....	97
8.8.	<i>Softwares</i> de Teste .....	98
8.9.	Ajuste dos Tempos de Execução .....	99
8.10.	Microcontrolador Gerenciador de Teste.....	100
Parte III - Resultados e Conclusões .....		101
9.	Resultados.....	102
9.1.	Introdução .....	102
9.2.	Testes de Redução dos Níveis de Tensão de Alimentação.....	102
9.3.	Testes Interferência Eletromagnética Irradiada.....	107
10.	Conclusões Finais .....	114
10.1.	Trabalhos Futuros .....	116
Referências Bibliográficas.....		117
Anexo A – Esquemático da Plataforma de Teste .....		124
Anexo B – Esquemático da Placa de Alimentação e Injeção de Falhas.....		134
Anexo C – Código do Circuito <i>Clock Manager</i> .....		136
Anexo D – Código do Circuito de Alimentação .....		140
Anexo E – Código do Circuito Sistema de Gerenciamento de Testes .....		144
Anexo F – Código <i>Software</i> do Microcontrolador de Gerenciamento de Teste .....		147
Anexo G – Planilhas dos Testes de Redução de Tensão .....		151
Anexo H – Planilha dos Testes de EMI Irradiada .....		153
Anexo I – Publicações .....		158

# **Parte I - Fundamentos Teóricos**

# 1. Introdução

## 1.1. Motivação

O número de aplicações críticas embarcadas cresceu significativamente nos últimos anos, proporcionando assim um considerável aumento das pesquisas relacionadas às técnicas capazes de agregarem confiabilidade e robustez a *Systems-on-Chip* (SoC's).

Aliado a isto, oscilações nos níveis de tensão do barramento de alimentação de circuitos digitais síncronos de elevada densidade de integração e alto desempenho provocam importantes atrasos de propagação (do inglês, *delay*) nos sinais internos destes dispositivos, reduzindo significativamente sua integridade de sinal e, por sua vez, causando a degradação de desempenho, a perda de confiabilidade e, em casos extremos, levando estes sistemas a ocorrências falhas funcionais.

É baseado neste cenário que propomos a implementação e a validação da técnica de Controle Dinâmico do Ciclo de Trabalho (CDCDC), proposta por João Paulo Cacho Teixeira, *et al* (1), para o aumento da robustez de *Systems-on-Chip* (SoC's) sujeitos a flutuações no barramento de alimentação sem a degradação do desempenho destes sistemas.

## 1.2. Objetivos

Esta dissertação tem como objetivo principal a implementação e a validação da técnica de Controle Dinâmico do Ciclo de Trabalho para o 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 implementados em FPGA's do tipo SRAM quando expostos a reduções dos níveis de tensão de alimentação;

- Analisar o comportamento de circuitos digitais síncronos implementados em FPGA's do tipo SRAM quando expostos a Interferências Eletromagnéticas (EMI) irradiadas;
- Desenvolver uma nova plataforma de testes com controle independente dos níveis de tensão dos dispositivos sob teste;
- Realizar testes de Interferências Eletromagnéticas (EMI) e redução de tensão seguindo as normas *International Electrotechnical Commission (IEC)* (2) (3) (4) para demonstrar a aplicabilidade da técnica de Controle Dinâmico do Ciclo de Trabalho para o aumento da 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 testes de *Systems-on-Chip* robustos, pois apresenta a validação de uma técnica que mantém o desempenho do sistema inalterado, mesmo quando este está sujeito 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* (SoC's);
- Capítulo 6: apresenta a técnica de Controle Dinâmico do Ciclo de Trabalho (CDCDC) proposta por João Paulo Cacho Teixeira, *et al* (1).

## Parte II – Metodologia

- Capítulo 7: apresenta a metodologia adotada nesta dissertação para a implementação da técnica de Controle Dinâmico do Ciclo de Trabalho em FPGA's do tipo SRAM, através de uma plataforma de desenvolvimento comercial.
- Capítulo 8: apresenta a metodologia adotada nesta dissertação para a implementação da técnica de Controle Dinâmico do Ciclo de Trabalho em FPGA's do tipo SRAM, através de uma plataforma de testes desenvolvida no laboratório de Sistemas, Sinais & Computação (SiSC).

## Parte III – Resultados e Conclusões

- Capítulo 9: apresenta os resultados dos testes de redução de tensão de alimentação e Interferência Eletromagnética Irradiada, realizados nesta dissertação, para a validação da técnica de Controle Dinâmico do Ciclo de Trabalho.
- 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

### 2.1. Introdução

Tolerância a falhas é a habilidade de um circuito e/ou sistema continuar a execução correta das suas tarefas (sem degradação de desempenho), mesmo diante da ocorrência de falhas em seu *hardware* e/ou em *software* (5) evitando assim prejuízos físicos e materiais.

Neste capítulo abordaremos os conceitos clássicos de tolerância a falhas, relacionados diretamente ao que esta dissertação propõe. A seguir são apresentados alguns conceitos formais retirados da literatura (5) (6) (7) (8) (9).

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

- a) **Falhas Permanentes:** ocorrem no meio físico e são provocadas através de falhas no processo de fabricação e/ou pelo envelhecimento dos componentes do sistema; curtos circuitos, nós abertos e *stuck-at* são exemplos de falhas permanentes.
- b) **Falhas Transientes:** ocorrem durante a vida útil dos componentes e são provocadas por adversidades e/ou fenômenos ambientais aleatórios onde o sistema está implementado; variações de tensão de alimentação e interferências eletromagnéticas são exemplos de falhas transientes.
- c) **Falhas Intermitentes:** é caracterizada pela ocorrência temporária e cíclica do erro a partir de variações das condições externas e/ou ambientais do sistema; vibrações e variações da temperatura são exemplos de falhas intermitentes.

***Stuck-at:*** tipo de falha permanente onde um nó do circuito está sempre no mesmo nível lógico seja ele zero (*stuck-at-zero*) ou um (*stuck-at-one*).

***Bit flips:*** tipo de falha transiente ocasionada por interferências externas, que resultam em uma mudança temporária no nível lógico em um determinado nó do circuito. Esta mudança pode ocorrer em ambos os sentidos de zero para um ou de um para zero.

**Erro:** define-se que um sistema está 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 medido desde a ocorrência da falha até a manifestação da mesma.

**Dependabilidade:** esse termo é uma tradução literal do termo inglês *dependability*, que indica a qualidade e a confiança depositada no serviço fornecido por um dado 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

Estamos interessados que um determinado sistema atenda satisfatoriamente as suas especificações de projeto e as necessidades dos seus usuários. Um defeito (do inglês, *failure*) é definido como um desvio da especificação. Define-se que um sistema está em estado errôneo, ou em erro, se o processamento posterior a partir deste estado pode levá-lo a um defeito. Finalmente define-se falha ou falta (do inglês, *fault*) como a causa física ou algorítmica do erro.

Falhas são inevitáveis, pois componentes físicos envelhecem e/ou sofrem com interferências externas, sejam ambientais ou humanas, projetos de *software* e *hardware* são vítimas de sua alta complexidade e também da fragilidade humana em trabalhar com grande volume de detalhes ou ainda com a deficiência de especificações.

A Figura 2.1 apresenta uma simplificação, sugerida por Dhiraj K. Pradhan (7), e também adotada nesta dissertação para os conceitos de falha, erro e defeito. Nela falhas estão associadas ao universo físico, erros ao universo da informação e defeitos ao universo do usuário.

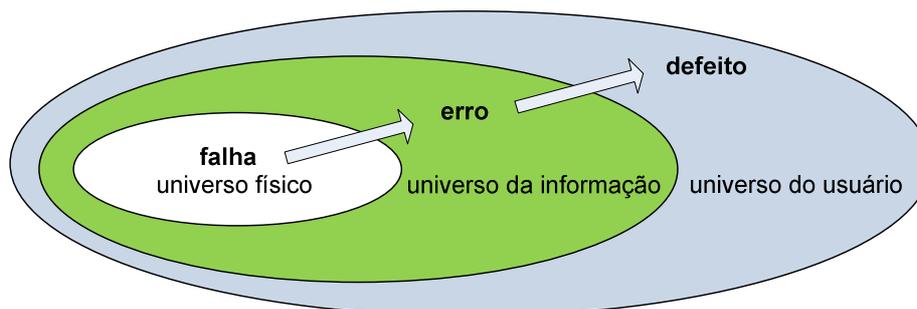


Figura 2.1 - Modelo dos três universos (7).

Um exemplo para este modelo de três universos (7) seria um *chip* de memória, que apresenta uma falha do tipo *stuck-at-zero* em um de seus *bits* (falha no universo físico). Esta falha pode provocar uma interpretação errada da informação armazenada em uma estrutura de dados (erro no universo da informação). Como resultado deste erro, por exemplo, o sistema pode negar autorização de embarque para todos os passageiros de um voo (defeito no universo do usuário). É interessante observar que uma falha não necessariamente leva a um erro (pois a porção da memória sob falha pode nunca ser usada) e um erro não necessariamente conduz 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 desde a ocorrência da falha até a manifestação do erro provocado por esta falha. Seguindo esta linha de raciocínio, define-se como latência de erro o período de tempo desde a ocorrência do erro até a manifestação do defeito.

Baseando-se no modelo de três universos (7), o tempo total medido desde a ocorrência da falha até o aparecimento do defeito é a soma das latências de falha e de erro. A Figura 2.2 apresenta o conceito de latência de falha e latência de erro.

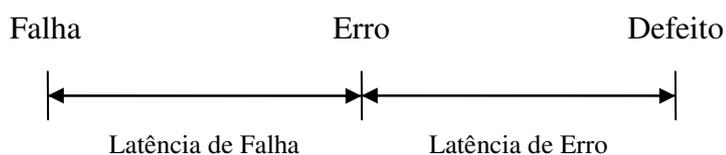


Figura 2.2 - Conceito de latência de falha e latência de erro.

### 2.2.2. Classificação de Falhas

Existem diversas classificações para falhas na literatura (5) (6) (7), entretanto estas são geralmente classificadas em falhas físicas (aquelas de que padecem os componentes) e falhas humanas (que compreendem falhas de projeto e falhas de interação).

Grande parte das causas de falhas são atribuídas a problemas de especificação, problemas de 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.

Para definir uma falha, além do agente causador consideram-se também 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 circuito;
- d) **Valor:** determinado ou indeterminado no tempo.

Existe uma crescente ocorrência de falhas provocadas por interação humana maliciosa, ou seja, por ações que visam propositalmente provocar danos aos sistemas. Essas falhas não são tratadas por técnicas de tolerância a falhas, mas sim por técnicas de segurança computacional (do inglês, *security*). Entretanto deve-se considerar que um sistema tolerante a falhas deve também ser seguro a intrusões e ações maliciosas.

Falhas de *software* e também de projeto são consideradas atualmente o mais grave problema em computação crítica. Sistemas críticos são, tradicionalmente, construídos de forma a suportar e tolerar falhas físicas. Tendo em vista isto, 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. Um exame de estatísticas disponíveis em (10) confirma essas considerações conforme mostrado na Tabela 2.1.

Tabela 2.1 - Causas usuais em sistemas computacionais (10).

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

## 2.3. Defeitos e Modelos de Falhas

Os testes em circuitos ou sistemas eletrônicos são realizados com o intuito de detectar falhas eventualmente presentes nestes dispositivos. Conseqüentemente, para a realização destes testes é necessário o emprego de modelos de falhas baseados em falhas reais definidas a partir de mecanismos físicos e *layouts* reais. Segundo Paul H. Bardell (11), um modelo de falha especifica a série de defeitos físicos que podem ser detectados através de um procedimento de teste. Um bom modelo de falha, segundo Charles E. Stroud (12), deve ser computacionalmente eficiente em relação ao dispositivo de simulação e refletir fielmente o comportamento dos defeitos que podem ocorrer durante o processo de projeto e manufatura, bem como o comportamento das falhas que podem ocorrer durante a operação do sistema. Estes modelos são utilizados na emulação de falhas e defeitos durante a etapa de simulação do projeto.

Neste contexto, nos últimos anos surgiram diversos modelos de falhas baseados nos principais defeitos físicos dos circuitos e sistemas eletrônicos, alguns destes serão apresentados nos itens a seguir.

### a) Modelo de falha *Gate-Level Stuck-at*

Este modelo de falha define que as portas de entrada e/ou saída do circuito podem estar fixadas em nível lógico '0' (*stuck-at-zero*) ou fixadas em nível lógico '1' (*stuck-at-one*). Salienta-se que as falhas *stuck-at* são emuladas como se as portas de entradas e/ou saídas estivessem desconectadas e fixadas ao nível lógico '0' (*stuck-at-zero*) ou ao nível lógico '1' (*stuck-at-one*) (10). A Figura 2.3 apresenta as formas de notação e emulações utilizadas para falhas *stuck-at*.

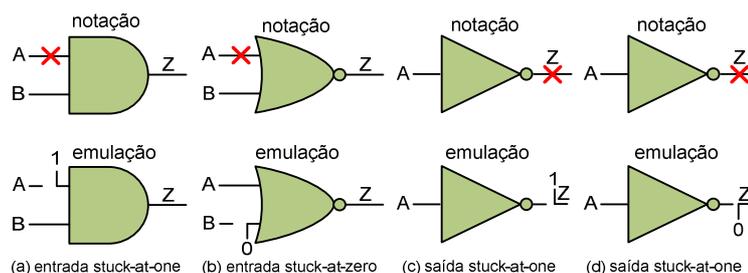


Figura 2.3 - Notação e emulação do modelo de falha *Stuck-at* (12).

### b) Modelo de Falha Transistor-Level Stuck

Este modelo reflete o comportamento exato das falhas de transistores em circuitos *CMOS* (*Complementary Metal-Oxide-Semiconductor*) e define que qualquer transistor pode estar em *stuck-on* (*s-on*) ou em *stuck-off* (*s-off*) (10).

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

Alternativamente, falhas *stuck-on* podem ser emuladas desconectando o pino de *gate* de um determinado *MOSFET* (*Metal-Oxide Semiconductor Field Effect Transistor*) do circuito e conectando-o ao nível lógico '1' para transistores *NMOS* (*Negative Metal Oxide Semiconductor*) ou ao nível lógico '0' para transistores *PMOS* (*Positive Metal Oxide Semiconductor*). O raciocínio inverso desta lógica pode ser realizado para emular falhas do tipo *stuck-off*. A Figura 2.4 apresenta um exemplo de emulação do modelo de falha *Transistor-Level Stuck* em uma porta lógica NOR.

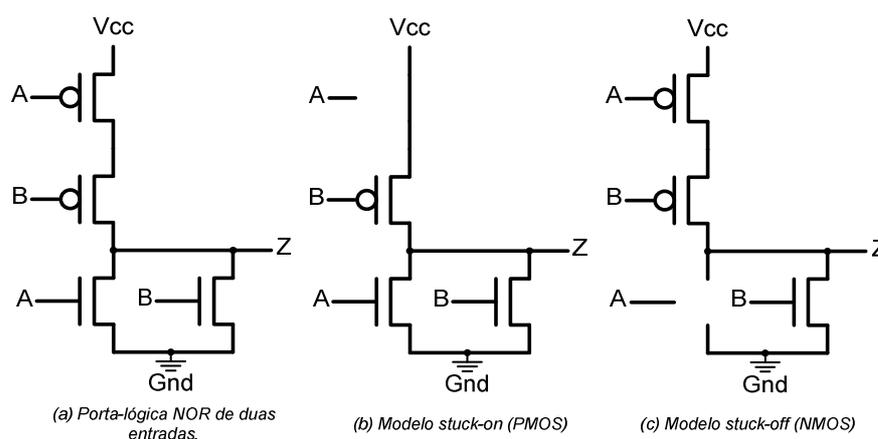


Figura 2.4 - Modelo de falha *Transistor-Level Stuck* (12).

### c) Modelo de Falha *Bridging*

Este modelo inclui outro importante conjunto de falhas, tais como rompimentos e/ou curtos entre trilhas de um determinado circuito.

Basicamente, a presença deste tipo de falha é resultante da deposição excessiva (*over-etching*) e/ou reduzida (*under-etching*) de material condutor durante o processo de fabricação das trilhas de circuitos VLSI (*Very Large Scale Integration*) ou ainda em PCB (*Printed Circuit Board*) (13). Além destes, outro modelo de falha *bridging* é definido a partir do comportamento observado em curtos circuitos ocorridos em ASIC's (*Application Specific Integrated Circuit*) e FPGA's sendo denominado *dominant-AND/OR bridging*.

A Figura 2.5 apresenta os modelos de falha *wired-AND/wired-OR bridging* e *dominant bridging*.

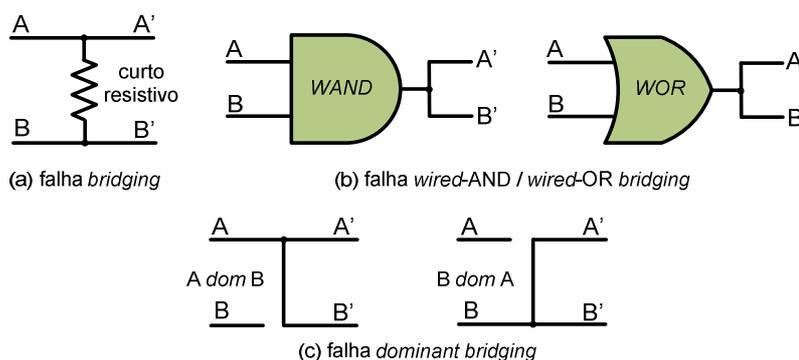


Figura 2.5 - Modelo de falha *bridging* (12).

Observa-se que embora falhas *transistor-level* e *bridging* reflitam com maior fidelidade o comportamento das falhas presentes em circuitos, sua emulação e avaliação em simuladores são computacionalmente mais complexas em relação às tradicionais falhas *stuck-at*(10).

#### d) Modelo de Falha *Delay*

Este modelo, alvo principal desta dissertação, representa outra importante classe de falha. Ao contrário dos demais modelos de falha aqui apresentados, os circuitos que apresentam falha de *delay* executam suas operações corretamente do ponto de vista lógico combinacional, entretanto, estas operações lógicas não são executadas ao longo do circuito na frequência de operação nominal requerida pelo projeto inicial, ocasionando assim um erro de *timing* proveniente dos diferentes tempos de propagação entre os sinais internos do circuito(13).

Este tipo de falha origina-se, em um caráter permanente, a partir de um *over* e/ou *under-etching* durante o processo de fabricação de circuitos *MOSFET's* com canais muito mais estreitos e/ou longos do que os pretendidos no projeto inicial. Entretanto, existe também a possibilidade de circuitos *MOSFET's* fabricados sem a presença de *over* e/ou *under-etching* apresentarem falhas de *delay* em caráter transiente.

Falhas de *delay* transientes são abordadas como foco principal de análise nesta dissertação e estas podem ser geradas através de interferências nas linhas de alimentação do circuito (quedas de tensão), variações de temperatura muito acentuadas e/ou provocadas por interferência eletromagnética (*EMI*).

Assim, o teste de *delay* concentra-se em encontrar e expor toda e qualquer falha que possa existir no dispositivo. O objetivo básico deste tipo de teste 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 interno do dispositivo apresenta erro de *timing*.

Tipicamente, o teste de *delay* consiste na aplicação seqüencial de 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 internos do circuito para detecção da falha (10).

## 2.4. Medidas Relacionadas ao Tempo Médio de Funcionamento

As medidas para avaliação de dependabilidade mais usadas na prática são: taxa de defeitos, MTTF (do inglês, *mean time to failure*), MTTR (do inglês, *mean time to repair*), MTBF (do inglês, *mean time between failure*). Estas medidas estão por sua vez relacionadas a outro parâmetro importante chamado confiabilidade (7). A Tabela 2.2 apresenta uma definição informal dessas medidas.

Tabela 2.2 - Medidas de dependabilidade (7).

Medida	Significado
Taxa de defeitos ( <i>failure rate</i> , <i>hazard function</i> , <i>harzard 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 de dependabilidade são determinadas estatisticamente pelos fabricantes, através da observância do comportamento dos componentes e dispositivos fabricados e deveriam ser fornecidas, tanto para os componentes e dispositivos eletrônicos, quanto 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 2.6 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.

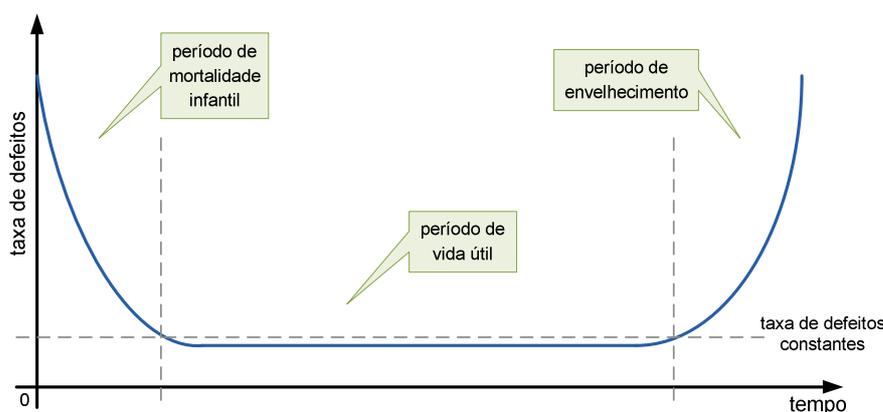


Figura 2.6 - Curva da banheira (7).

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.

É questionável se a curva da banheira pode ser aplicada também para componentes de *software*. Entretanto, pode ser observado que os componentes de *software* também apresentam uma fase de mortalidade infantil ou taxa de erros acentuados no início da sua fase de testes, que decresce rapidamente até a sua entrada em operação. A partir desse momento, o *software* apresenta uma taxa de erros constante até que, eventualmente, precise sofrer alguma alteração ou sua plataforma de *hardware* torne-se obsoleta. Nesse momento, a taxa de erros volta a crescer (14). Intencionalmente mencionamos, quando referido a *software*, taxa de erros ao contrário de defeitos, já que erro é o termo usualmente empregado quando se trata de programas incorretos.

## 3. Compatibilidade Eletromagnética

### 3.1. Introdução

O uso de dispositivos eletrônicos vem assumindo um papel indispensável no cotidiano de milhares de pessoas. Cada vez mais, equipamentos e produtos eletrônicos, como telefones celulares, dispositivos de computadores sem-fio, além de transmissões de rádio e televisão, dividem o mesmo ambiente eletromagnético, operando de maneira simultânea, silenciosa e invisível para a conveniência e satisfação das necessidades humanas modernas (15).

Entretanto, apesar dos inegáveis benefícios e facilidades que estes sistemas proporcionam ao nosso dia-a-dia, o acréscimo de Interferência Eletromagnética (do inglês, *Electromagnetic Interference – EMI*) por parte destes dispositivos, aos ambientes que estes estão inseridos, é uma situação inevitável e cada vez mais acentuada. Exemplos destas interferências são as perturbações visíveis na imagem de um televisor quando um veículo motorizado ruidoso (em radiação eletromagnética) passa nas suas proximidades ou ainda, quando ouvimos no nosso receptor de rádio perturbações oriundas de um aspirador de pó.

Neste cenário, o estudo da Compatibilidade Eletromagnética (do inglês, *Electromagnetic Compatibility – EMC*), que segundo Robert Bosch (16) 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, vem assumindo um importante papel no projeto e desenvolvimento de sistemas embarcados.

A metodologia adotada para análise da EMC possui dois pontos de vista a serem explorados. No primeiro caso, denominado emissão, o foco de análise se concentra na EMI que o dispositivo e/ou sistema embarcado emana para o meio em que está inserido. Já o segundo caso, denominado susceptibilidade, o foco de análise é centrado no comportamento do dispositivo e/ou sistema diante da presença de EMI oriunda de outros dispositivos e/ou sistemas.

A importância destes estudos é evidenciada pelo emprego de diversas técnicas de projeto (como *layouts* 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 Européias (17) relacionadas à EMC, ou nos regulamentos das companhias de aviação comercial que proíbem, aos passageiros, a utilização de aparelhos eletrônicos durante os vôos, para impossibilitar 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:** é definida como o fenômeno no qual a energia eletromagnética (interferência ou distúrbio) emana de uma determinada fonte geradora para um dispositivo ou sistema “vítima” (17).
- **Nível de Emissão:** é definido como o nível de interferência eletromagnética emitida a partir de um determinado dispositivo, equipamento ou sistema (17).
- **Limite de Emissão:** é definido como o nível máximo de emissão de uma fonte geradora de interferência eletromagnética(17).
- **Imunidade:** é a capacidade de um dispositivo, equipamento ou sistema executar suas funções na presença de uma perturbação, ruído ou interferência eletromagnética sem degradação de desempenho (17).
- **Nível de Imunidade:** é 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 (17).
- **Limite de Imunidade:** é 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 (17).
- **Susceptibilidade (eletromagnética):** é a situação inversa de imunidade (a falta), ou seja, é a incapacidade de um dispositivo, equipamento ou sistema executar suas funções na presença de uma perturbação eletromagnética (17).
- **Limite de Interferência:** é definido como o nível de interferência eletromagnética máxima admissível de um determinado dispositivo, equipamento ou sistema (17).

- **Nível de Compatibilidade Eletromagnética:** é definido como o nível de interferência eletromagnética utilizado como referência para a fixação de limites de emissão e imunidade (17).

### 3.3. Ambiente Eletromagnético

Um ambiente eletromagnético (do inglês, *Electromagnetic Environment – EME*) pode ser definido como a totalidade de fenômenos e interações eletromagnéticas existentes em um determinado local (17).

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

Seguindo esta mesma linha de raciocínio, toda e qualquer alteração que este ambiente sofra, seja estrutural ou comportamental, provocará mudanças na sua distribuição espacial de campos eletromagnéticos e conseqüentemente nas características de EMC dele e dos dispositivos e/ou sistemas nele inseridos.

### 3.4. Interferência Eletromagnética

As Interferências Eletromagnéticas (do inglês, *Electromagnetic Interference – EMI*) são perturbações causadas a um circuito, dispositivo e/ou sistema eletrônico por radiações eletromagnéticas emitidas a partir de uma fonte externa (17). Estes tipos de radiações podem provocar no circuito “vítima” à limitação do seu desempenho, como também a interrupção temporária ou permanente das suas funções.

Para compreender como estas Interferências Eletromagnéticas ocorrem, quais são as suas conseqüências e as possíveis medidas para minimizar ou extinguir seus efeitos, é necessário compreender o modo como o acoplamento ou ligação condutiva é formada, quais são os dispositivos causadores e/ou receptores e a também quais as frequências envolvidas neste fenômeno. Estes aspectos são apresentados sucintamente a seguir.

### 3.4.1. Fontes de Interferências Eletromagnéticas

Um importante ponto no estudo da Compatibilidade Eletromagnética (EMC) está associado às causas dos ruídos, interferências ou distúrbios eletromagnéticos (EMI) que os dispositivos, equipamentos e/ou sistemas estão sujeitos. Segundo David A. Weston (18) as fontes de EMI podem ser classificadas em naturais e artificiais, como segue:

- a) **Fontes naturais:** são fontes geradoras de ruídos, interferências ou distúrbios eletromagnéticos provenientes de fenômenos naturais como descargas elétricas atmosféricas e explosões solares (17).
- b) **Fontes artificiais:** são fontes geradoras de ruídos, interferências ou distúrbios eletromagnéticos provenientes da ação do homem, isto é, são provocados por fenômenos não naturais como acionamentos de cargas indutivas (motores elétricos), de cargas resistivas (aquecedores), de lâmpadas fluorescentes, aparelhos de microondas, telefones celulares e etc. (17).

### 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” (17). Estes fenômenos podem ser divididos, de acordo com o meio de propagação, em dois tipos (19):

- a) **Interferência Conduzida:** é definida como aquela na qual a energia é transferida, da fonte geradora ao circuito ou dispositivo “vítima”, através de um ou mais condutores de alimentação, pela referência de tensão (terra), ou ainda por pinos de entrada ou saída de sinais (17). Este tipo de interferência ocorre de forma freqüente em redes elétricas de sinal alternado (20), já que este tipo de rede alimenta, paralelamente, muitos outros sistemas ruidosos (eletromagneticamente) como motores, contactoras e relés. A Figura 3.1 apresenta um exemplo de interferência eletromagnética conduzida através da rede elétrica, onde uma máquina elétrica afeta o funcionamento de um equipamento eletrônico (microcomputador). Este tipo de interferência é muito observado em ambientes industriais, comerciais e domésticos.

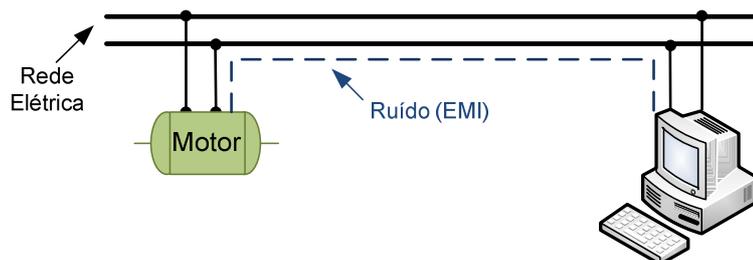


Figura 3.1 - Exemplo de Interferência Eletromagnética conduzida (19).

- b) **Interferência Irradiada:** é definida como aquela na qual a energia é transferida, da fonte geradora ao circuito ou dispositivo “vítima”, através do espaço sob a forma de ondas eletromagnéticas (17). Neste tipo de interferência tem-se um elemento irradiando um campo eletromagnético (fonte geradora de interferência) que se propaga na atmosfera e atinge equipamentos eletrônicos e sistemas de telecomunicações (“vítima”) podendo-os levar a falhas ou ainda a interrupção das suas funções (20). A Figura 3.2 apresenta um exemplo de interferência eletromagnética irradiada.

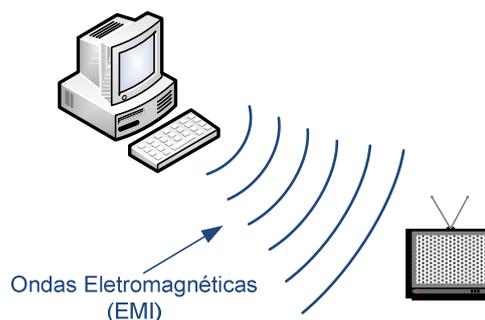


Figura 3.2 - Exemplo de Interferência Eletromagnética irradiada (19).

### 3.4.3. Acoplamento de Interferência Eletromagnética

O caminho de acoplamento é definido como o meio no qual uma determinada fonte de interferência utiliza para transferir parte ou a totalidade da sua energia eletromagnética a outro circuito e/ou dispositivo “vítima” (17). No caso do estudo e análise da Compatibilidade Eletromagnética (EMC) o caminho de acoplamento pode ser do tipo conduzido, irradiado ou ainda uma combinação destas duas.

A Figura 3.3 apresenta 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”.

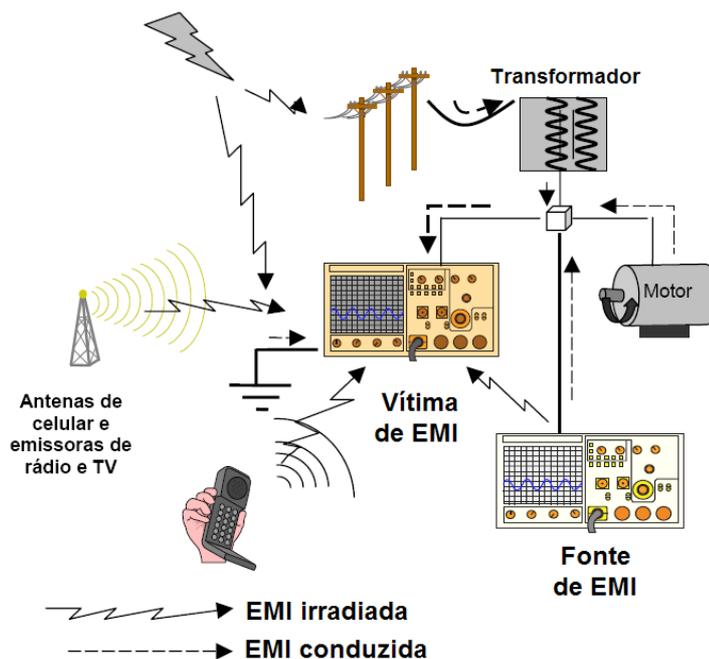


Figura 3.3 - Exemplos de acoplamentos de EMI conduzida e irradiada (21).

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.

Seguindo este raciocínio, 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” (22).

### 3.5. Efeitos da Interferência Eletromagnética na Eletrônica

A Integridade de Sinal (do inglês, *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 (23).

Seguindo esta mesma linha de raciocínio, um sinal com boa integridade apresenta, entre as suas características, níveis de tensão e tempos de transição de acordo com as exigências de projeto do sistema em que este sinal está associado. A Figura 3.4 apresenta exemplos de sinais e suas características de integridade.

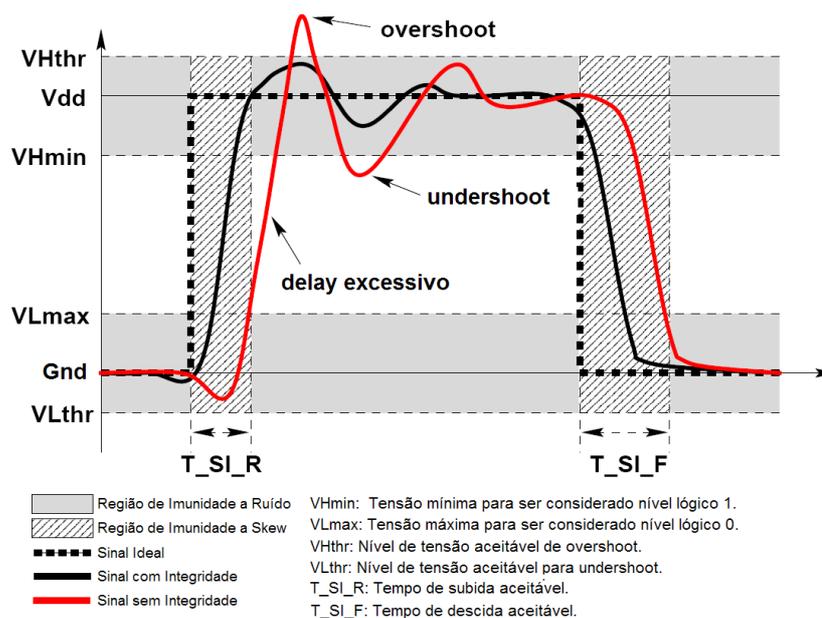


Figura 3.4 - Exemplo de Integridade Sinais (23).

Entretanto, a convivência de equipamentos de tecnologias diferentes em um mesmo ambiente, agregada à precariedade das instalações elétricas, ou ainda erros de projetos (19) 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 assim, a possibilidade de distúrbios como:

- a) **Flutuações nas linhas de alimentação:** momentâneas reduções dos níveis de tensão das linhas de alimentação ( $V_{cc}$ ) de um dispositivo e/ou sistema eletrônico (24) (25);
- b) **Ground Bounce:** momentâneas elevações dos níveis de tensão das linhas de referência de tensão ( $Gnd$ ) de um dispositivo e/ou sistema eletrônico (24) (25);
- c) **Ruído das Linhas de Alimentação:** variações de corrente de carga provenientes do rápido chaveamento de circuitos (26) (27);
- 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 (28) (29) (30).

Exemplos desta realidade são evidenciados em plantas industriais, onde a presença de EMI é muito freqüente em função da coabitação de máquinas e motores elétricos (fontes geradoras de ruído eletromagnético) e dispositivos eletrônicos como computadores e redes digitais. Neste cenário, a degradação dos sinais de dispositivos e/ou sistemas eletrônicos é um agente decisivo na ocorrência de falhas funcionais como *bit-flips* e atrasos de propagação na rede de distribuição de sinais.

## 3.6. Normas IEC

A *International Electrotechnical Commission* (IEC) (31) foi criada em junho de 1906 na Inglaterra e é, atualmente, a principal organização mundial no desenvolvimento e publicação de normas técnicas internacionais para as áreas relacionadas à engenharia elétrica e eletrônica.

Através da cooperação internacional dos seus membros, a IEC promove a definição e avaliação de padrões relativos às áreas elétricas e eletrônicas, facilitando assim o comércio e removendo barreiras técnicas (31). A seguir comentaremos brevemente as normas adotadas no desenvolvimento desta dissertação.

### 3.6.1. Norma IEC 61.000-4-17

Ondulações de tensão (*ripples*) podem influenciar decisivamente na confiabilidade e no desempenho de equipamentos e/ou dispositivos elétricos e eletrônicos alimentados com fontes de corrente contínua (DC). A norma IEC 61.000-4-17 tem como objetivo estabelecer, em laboratório, uma base para testes 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 (3).

De uma forma geral, esta norma define características para testes de ondulações de tensão como: a forma da onda da tensão, níveis e amplitudes de tensão, os geradores de teste, e também o *setup* e os procedimentos adequados de teste.

A Tabela 3.1 apresenta os níveis de tensão definidos, nesta norma técnica, para testes em fontes de alimentação DC de equipamentos e/ou dispositivos elétricos e/ou eletrônicos.

Tabela 3.1 - Níveis de Tensão de Teste (10).

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.	

Os níveis de teste são tensões pico-a-pico expressas em pontos percentuais relativas à tensão nominal DC ( $U_{DC}$ ). A amplitude da tensão de *ripple* ( $U_{m\acute{a}x} - U_{m\acute{i}n}$ ) é apresentada na Figura 3.5.

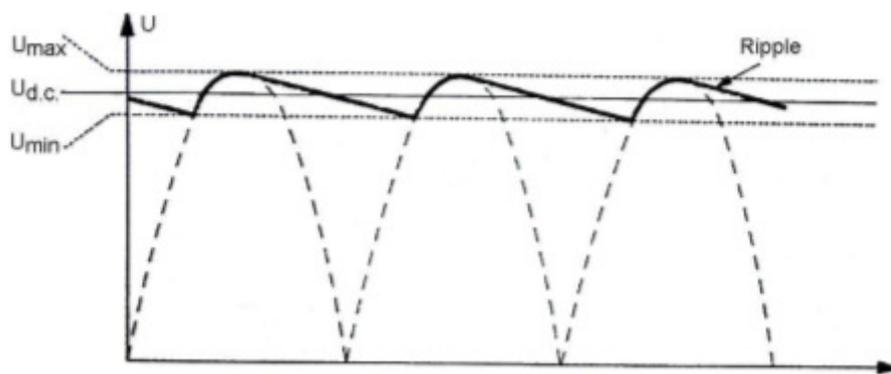


Figura 3.5 - Retificador de monofásico (3).

Para melhor compreensão da Tabela 3.1 e da Figura 3.5, pode-se utilizar o exemplo de um microcontrolador com tensão nominal de alimentação ( $U_{DC}$ ) igual a 3,3V. Supondo que as linhas de alimentação deste dispositivo podem estar sujeitas a ondulações de 10% (nível 3), o valor de pico de  $U_{m\acute{a}x}$  será igual a 3,47V (5% acima de  $U_{DC}$ ) e o valor de  $U_{m\acute{i}n}$  será igual 3,13V (5% abaixo de  $U_{DC}$ ) (32).

### 3.6.2. Norma IEC 61.000-4-29

Tanto a operação correta quanto o desempenho, de equipamentos e/ou dispositivos elétricos e eletrônicos, podem sofrer degradação quando estes estão sujeitos a distúrbios nas suas linhas de alimentação.

Neste sentido, a norma IEC 61.000-4-29 objetiva estabelecer um método básico para testes de imunidade de equipamentos e/ou dispositivos alimentados por fontes de corrente contínua externas de baixa tensão.

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

- a) **Queda de tensão:** é caracterizada como uma súbita redução na tensão de alimentação, do equipamento e/ou dispositivo, seguida da sua recuperação em um curto período de tempo (4).
- b) **Pequena Interrupção:** é caracterizada como o desaparecimento momentâneo da tensão por um período de tempo não maior que um minuto. Na prática, quedas de tensão superiores a 80% são consideradas interrupções (4).

- c) **Variação de tensão:** é caracterizada como uma mudança gradual da tensão de alimentação para um valor maior ou menor do que a tensão nominal ( $U_T$ ), podendo ser a sua duração curta ou longa (4).

As tabelas abaixo apresentam os níveis de tensão percentuais relativos à tensão nominal ( $U_T$ ) e os tempos de duração sugeridos pela norma para testes em equipamentos e dispositivos elétricos e eletrônicos.

Tabela 3.2 - Níveis de tensão e duração recomendadas para interrupções (4).

Teste	Nível de Tensão (% $U_T$ )	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.		

Tabela 3.3 - Níveis de tensão e duração recomendados para quedas de tensão (4).

Teste	Nível de Tensão (% $U_T$ )	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 3.4 - Níveis de tensão e duração recomendados para variação de tensão (4).

Teste	Nível de Tensão (% $U_T$ )	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.		

A norma IEC 61.000-4-29 sugere topologias de conexão entre os geradores de ruídos de alimentação e os dispositivos sobre teste (do inglês, *Device Under Test* - DUT). A Figura 3.6 e a Figura 3.7 apresentam exemplos destas topologias.

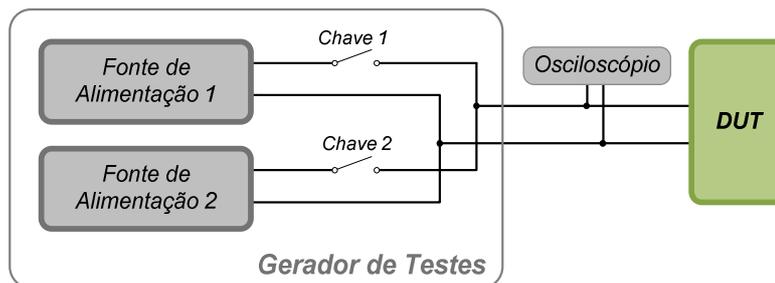


Figura 3.6 - Gerador de teste com fontes de alimentação de chaveamento interno (4).

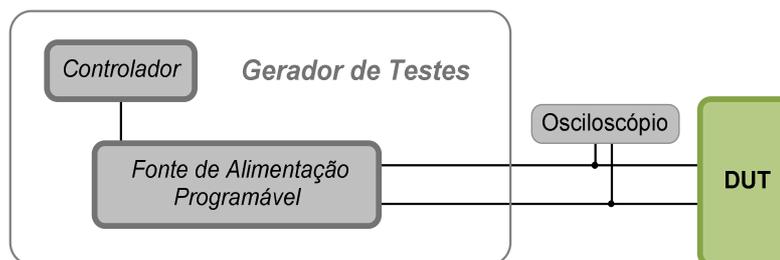


Figura 3.7 - Gerador de teste baseado em uma fonte de alimentação programável (4).

### 3.6.3. Norma IEC 62.132

A definição de níveis de imunidade eletromagnética em circuitos integrados é fundamental para a manutenção da qualidade e confiabilidade de projetos de placas de circuito impresso e também para a integração destes dispositivos em equipamentos e sistemas eletrônicos.

Neste sentido, a família de normas técnicas IEC 62.132 tem como objetivo definir, em linhas gerais, uma metodologia de testes para estimar a imunidade de circuitos integrados a distúrbios eletromagnéticos na faixa de frequências de 150kHz a 1GHz.

Tendo em vista a 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 totalmente 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 (2);
- **IEC 62.132-2:** Medições de imunidade eletromagnética em circuitos integrados sujeitos a distúrbios irradiados na faixa de frequência de 150kHz a 1GHz (33);
- **IEC 62.132-3:** Medições de imunidade eletromagnética em circuitos integrados sujeitos a distúrbios irradiados na faixa de frequência de 10kHz a 1GHz gerados por injeção de corrente (do inglês, *Bulk Current Injection – BCI*) (34);

- **IEC 62.132-4:** Medições de imunidade eletromagnética em circuitos integrados sujeitos a radiofrequência (RF) conduzida de até um 1GHz (35).
- **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 (36).

Como estudo de caso para esta dissertação, focamos nosso trabalho nos itens 1, 2 e 4 desta norma, sendo detalhadas a seguir.

### **IEC 62.132 – Parte 1**

A norma IEC 62.132-1 (2) fornece informações e definições gerais para medições de imunidade eletromagnética conduzida e irradiada em circuitos integrados. Nesta parte da norma, são fornecidas informações relativas às condições e procedimentos de testes, os equipamentos apropriados, além das necessidades e características do *setup* de teste.

- a) **Placa de Circuito Impresso:** As topologias das placas de circuito impresso (do inglês, *Printed Circuit Board – PCB*) utilizadas para testes 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 testes algumas características específicas relativas ao *layout* 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 (2). Estas características são apresentadas na Figura 3.8.

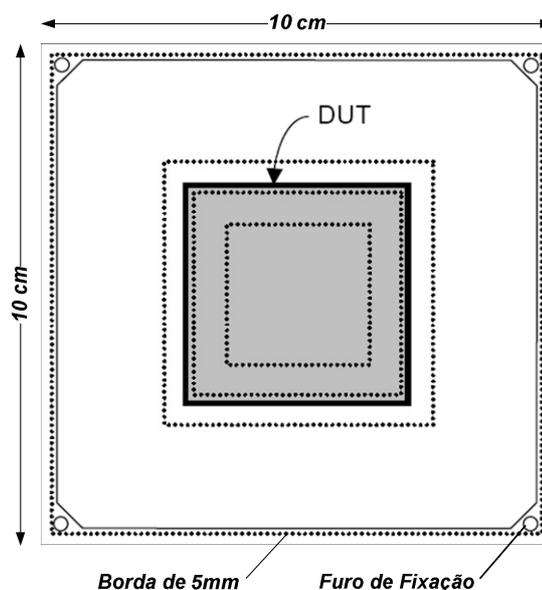


Figura 3.8 - Características de placa de circuito impresso de teste (2).

Aliado a estas características, uma placa de testes de imunidade eletromagnética em circuitos integrados deve dispor de uma elevada blindagem para garantir que somente o DUT esteja sujeito aos distúrbios eletromagnéticos. Tendo em vista esta necessidade, o número de camadas da placa (*layers*) deve ser igual ou superior a quatro (2). A Figura 3.9 e a Tabela 3.5 apresentam a disposição destas camadas e os seus conteúdos.

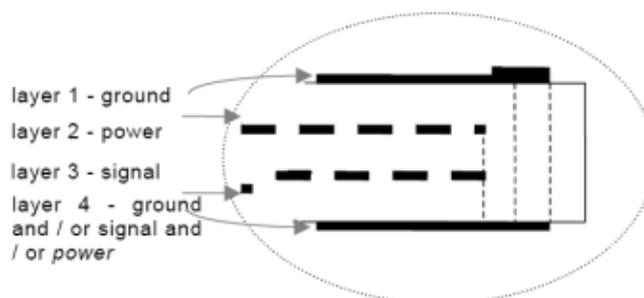


Figura 3.9 - Disposição das camadas de uma placa de teste (2).

Tabela 3.5 - Conteúdo das camadas da placa de circuito impresso.

Camada ( <i>layer</i> )	Conteúdo
1 - superior ( <i>top</i> )	Contém somente o circuito integrado que está sobre teste (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 ( $V_{CC}$ ) 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 $V_{CC}$ e <i>Gnd</i> (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.

- b) **Pinos não utilizados do CI:** A norma IEC 62.132-1 recomenda que os pinos do circuito integrado sob teste, a exceção daqueles que possuam uma funcionalidade especial definida pelo fabricante, devem ser isolados ou conectados a uma carga. A Tabela 3.6 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.

Tabela 3.6 - Conexão dos pinos do circuito integrado (2).

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

Os pinos não se enquadram em nenhuma das categorias listadas na Tabela 3.6 deverão ser conectados a cargas conforme exige sua funcionalidade e posteriormente comentado no relatório do ensaio.

- c) **Fonte de Alimentação:** O dispositivo sob teste deve ser alimentado por uma fonte de tensão que não esteja sujeita aos distúrbios eletromagnéticos aplicados. Aliado a isto, as linhas de alimentação devem ser filtradas de acordo com as especificações e recomendações do fabricante do circuito integrado.
- d) **Procedimento de Teste:** O procedimento de teste também é bastante relevante para a garantia da confiabilidade dos testes. Sendo assim, as seguintes recomendações devem ser seguidas:
- Verificar a operação normal da placa de testes e do DUT, isto é, realizar uma verificação das funcionalidades do sistema sem a aplicação de distúrbios eletromagnéticos;
  - A faixa de frequência abordada pela norma IEC 62.132-1 é de 150kHz a 1GHz. Entretanto, na prática, a faixa de frequência dos

testes 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 testes sigam os valores apresentados na Tabela 3.7.

Tabela 3.7 - Intervalos de frequência aplicados nos testes (2).

Faixa de Frequência (MHz)	0,15 – 1	1 – 100	100 – 1000
Intervalos Lineares (MHz)	≤ 0,1	≤ 1	≤ 10
Intervalos Logarítmicos	≤ 5% de incremento		

É importante ressaltar também 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 (2);

- O sinal de distúrbio usado para os testes de imunidade eletromagnética em circuitos integrados deve estar de acordo com as exigências da metodologia de teste adotada, isto é, sinais de onda contínua (do inglês, *continuous wave* – CW), sinais de 1kHz com amplitude modulada (do inglês, *Amplitude Modulation* – AM) em 80% ou ainda um pulso modulado com a taxa da repetição 1kHz (2). A Figura 3.10 apresenta exemplos destes tipos de distúrbios:

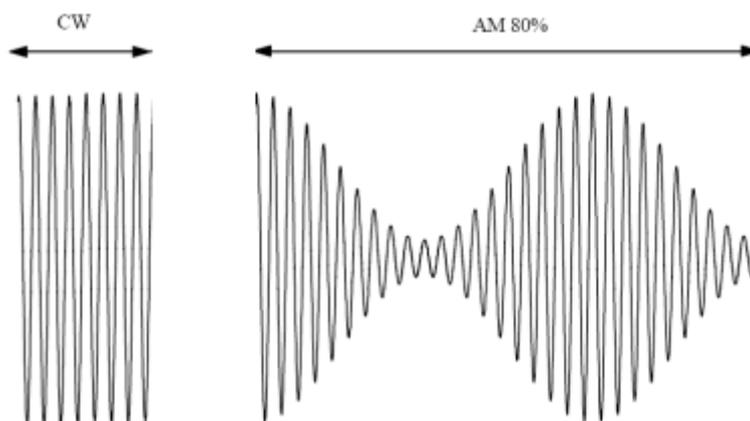


Figura 3.10 - Exemplos sinais de onda contínua e amplitude modulada (2).

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

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

- O monitoramento do circuito integrado deve ser realizado através de testes específicos considerando todas as suas funções operacionais. Neste sentido, 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 (2).
- e) **Relatório de Teste:** O relatório de teste deve ser detalhado o suficiente para especificar todos os parâmetros e variáveis nas quais o circuito integrado esteve sujeito durante o teste e permitir assim a sua repetibilidade (2). São exemplos de informações relevantes para o relatório de teste os seguintes dados:
- Diagrama do circuito;
  - Descrição da placa de teste na qual o CI está inserido (*layout*);
  - Condições de operação do CI (tensão de alimentação e sinais de saída);
  - Descrição do tipo de *software* executado no CI;

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

- **Classe A:** Todas as funções do CI 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 CI não executa durante a exposição conforme projetada, entretanto retorna automaticamente à operação normal depois que a exposição é removida;
- **Classe D:** Uma função do CI não executa durante a exposição conforme projetada e não retorna à operação normal até que a exposição seja removida e o CI seja reiniciado pela ação do operador;
- **Classe E:** Uma ou mais funções do CI não executam durante e após a exposição conforme projetadas, sendo sua operação normal impossível de ser retomada.

## **IEC 62.132 – Parte 2**

A norma IEC 62.132-2 define uma metodologia de teste 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 (do inglês, *Transverse Electromagnetic*) e GTEM (do inglês, *Gigahertz Transverse Electromagnetic*) (33).

Nesta metodologia o dispositivo sob teste (DUT) é montado em uma placa circuito impresso (PCB) e fixado na parede da célula através de uma porta de acoplamento (33). Este acoplamento na parede permite que a placa de teste e conseqüentemente o circuito integrado se tornem parte integrante da célula, ao contrário de testes convencionais nos quais a placa de teste é inserida no interior da célula alterando a orientação do campo elétrico (33).

A Figura 3.11 apresenta as configurações utilizadas em células TEM e GTEM para a realização dos testes de imunidade eletromagnética a distúrbios irradiados.

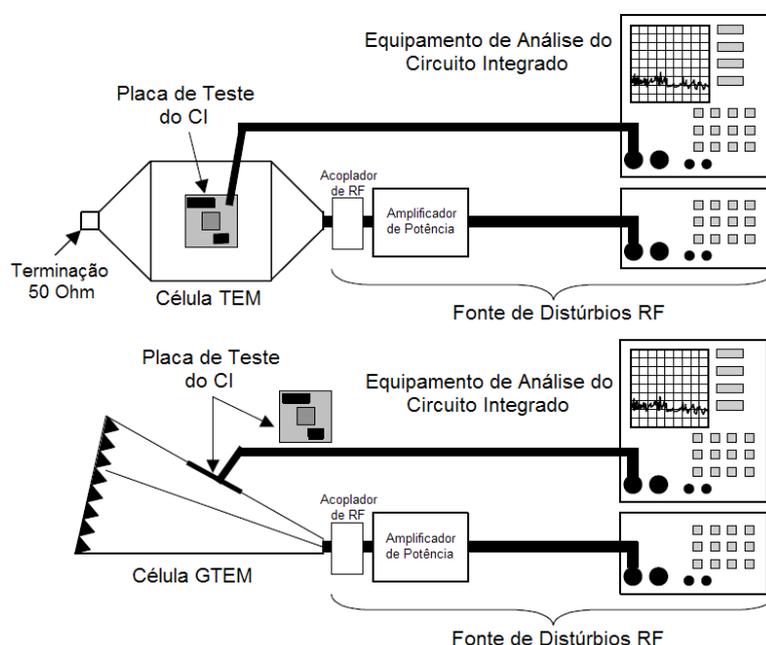


Figura 3.11 - Setup de teste com células TEM e GTEM (33).

#### IEC 62.132 – Parte 4

A norma IEC 62.132-4 descreve uma metodologia de teste para medições de imunidade em circuitos integrados sujeitos a presença de distúrbios de radiofrequência (RF) conduzidos (35).

Além de fornecer um padrão de teste com elevado grau de repetibilidade e correlação nas medidas, a norma também estabelece uma base comum para a avaliação de dispositivos semicondutores utilizados em equipamentos que operam em ambientes sujeitos a presença de ondas de radiofrequência indesejadas.

Segundo a norma, o nível de imunidade eletromagnética mínimo, para um determinado circuito integrado, é dependente do nível máximo de distúrbios toleráveis ao

sistema eletrônico no qual este dispositivo está inserido. Sendo assim, o valor do nível de imunidade é dependente dos parâmetros específicos do sistema e da aplicação.

O teste de injeção conduzida de RF para medições de imunidade eletromagnética em circuitos integrados é realizado, de acordo com a norma, na faixa de frequência de 150 kHz de até 1 GHz seguindo o *setup* apresentado na Figura 3.12.

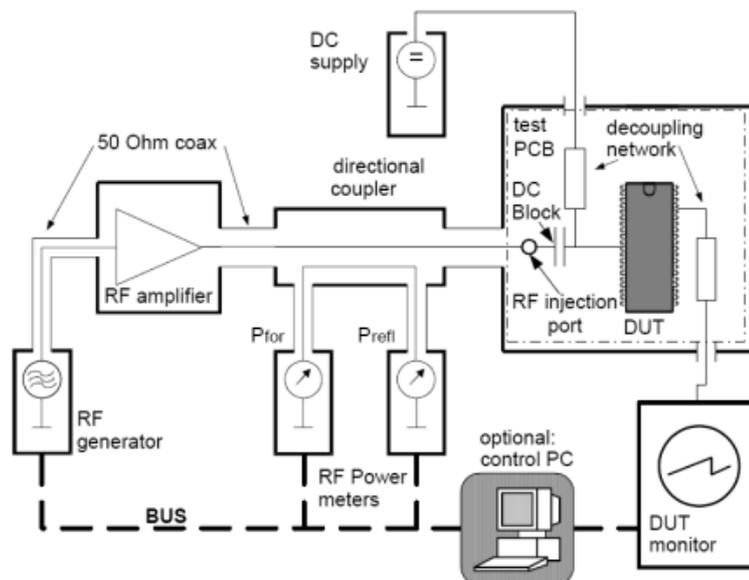


Figura 3.12 - *Setup* de teste de imunidade conduzida (35).

Neste *setup* apresentado na Figura 3.12, a injeção de distúrbios é realizada através de pinos do circuito integrado, que recebem um sinal amplificado proveniente de um gerador de radiofrequência. A norma define o emprego de amplificadores com elevadas potências (de 10 a 50W) para garantir níveis de distúrbios adequados aos circuitos integrados sob teste. Também é recomendada a utilização de outros equipamentos como acopladores direcionais de 50 *ohms* para o casamento de impedâncias, e medidores de potência para a análise das potências incidente e refletida.

A injeção de distúrbios pode ser realizada sob um ou mais pinos do circuito integrado sob teste. No caso da injeção em um único pino do circuito integrado, a potência de RF é injetada através de uma porta que pode conter em série um capacitor para bloqueio DC e um resistor para limitar a corrente e/ou simular uma carga real ao sistema. Os valores recomendados pela norma IEC 62.132 para estes componentes são 6,8nF e 100 ohms embora possam ser alterados conforme a necessidade e característica do CI. A Figura 3.13 apresenta o princípio da injeção de distúrbios de radiofrequência em um único pino do circuito integrado.

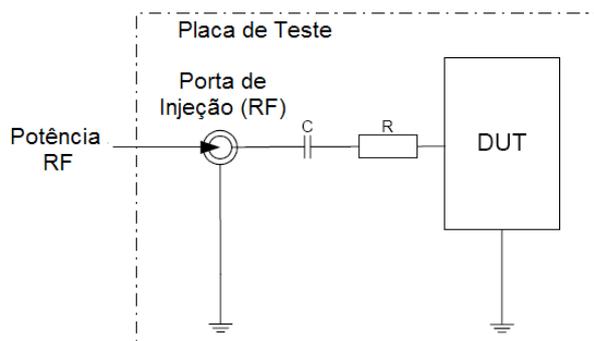


Figura 3.13 - Princípio de injeção de distúrbios RF em um pino (35).

Caso o teste de imunidade necessite injetar distúrbios de radiofrequência em mais de um pino do circuito integrado sob teste, esta injeção deve seguir os princípios do modo diferencial apresentado na Figura 3.14.

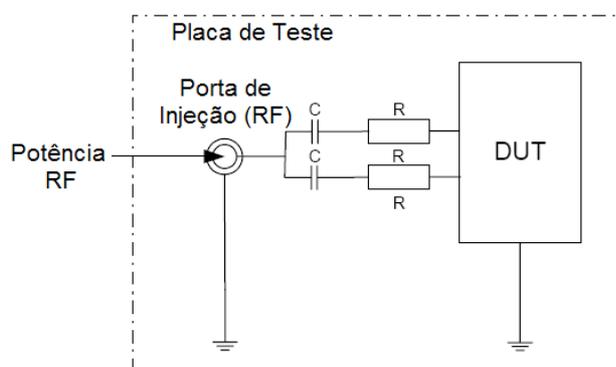


Figura 3.14 - Princípio de injeção de distúrbios RF em mais de um pino (35).

Conforme já foi referenciado anteriormente, os níveis de potência são muito importantes e decisivos para a qualidade e confiabilidade dos testes de imunidade a distúrbios conduzidos em circuitos integrados. Neste sentido, a norma define que 5 watts ou 37dBm<sup>1</sup> é a potência máxima aplicada de uma onda contínua (CW) a um pino do circuito integrado sem proteção externa. Entretanto, se o pino do circuito integrado for projetado para operar com proteção externa, estes níveis de potência podem ser modificados conforme valores apresentados na Tabela 3.8.

Tabela 3.8 - Níveis de potência de injeção (35).

Potência (Watts)	Proteção Externa	Exemplos de Dispositivos
1...5	Nenhuma ou somente um pequeno capacitor	Grandes chaves ( <i>switches</i> ), circuitos de alimentação, <i>transceiver</i> de barramento.
0,1...0,5	R-L-C, filtro passa baixa	Dispositivos de condicionamento de sinal, sensores, <i>drivers</i> de linhas de comunicação.
0,01...0,05	Sem conexão direta	Microntroladores, microprocessadores e memórias

1 O decibel (dB) é uma notação destinada a medir níveis de potência de forma relativa. A medida mais comum para expressar a potência de radiofrequência (RF) é o dBm (dB miliwatt), onde zero dBm é definido como 1mW de potência.

A configuração estrutural de teste de imunidade a radiofrequência conduzida em circuitos integrados é constituída de duas partes. A primeira delas refere-se aos componentes externos a placa de teste como:

- Gerador de sinais;
- Amplificador de potência;
- Cabos coaxiais;
- Conectores;
- Acoplador direcional;
- Medidor de potência incidente e refletida

Já a segunda parte apresentada na Figura 3.15 refere-se diretamente aos componentes contidos na placa de teste como:

- Portas de injeção de RF para conexão de cabos coaxiais;
- Placa de teste com plano de terra;
- Circuito integrado sob teste (DUT);
- Conector de injeção de RF;
- Capacitor e resistor de bloqueio;

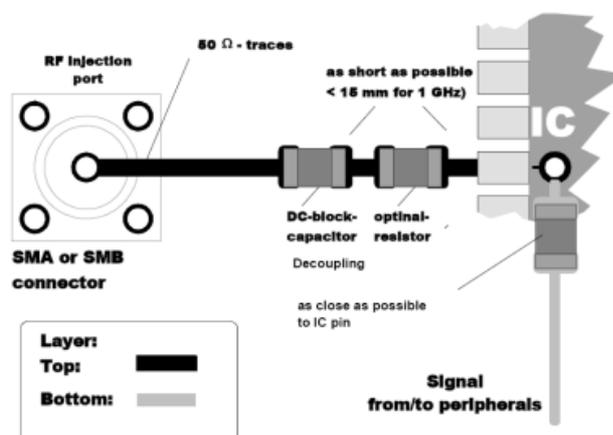


Figura 3.15 - Estrutura da placa de teste para testes de injeção de RF (35).

No sentido de garantir a confiabilidade dos testes de imunidade, a norma IEC 62.132-4 recomenda que a injeção de distúrbios de RF seja realizada o mais perto possível do dispositivo sob teste e em trilhas com impedância igual a 50 ohms. Segundo a norma esta recomendação pode ser adotada através de:

- Uma placa de teste com as menores dimensões possível, tornando assim, o conector de transmissão de RF mais perto do dispositivo sob teste (abordagem usada em circuitos integrados com baixo número de pinos). A Figura 3.16 apresenta o *layout* recomendado pela norma para este tipo de placa de teste.

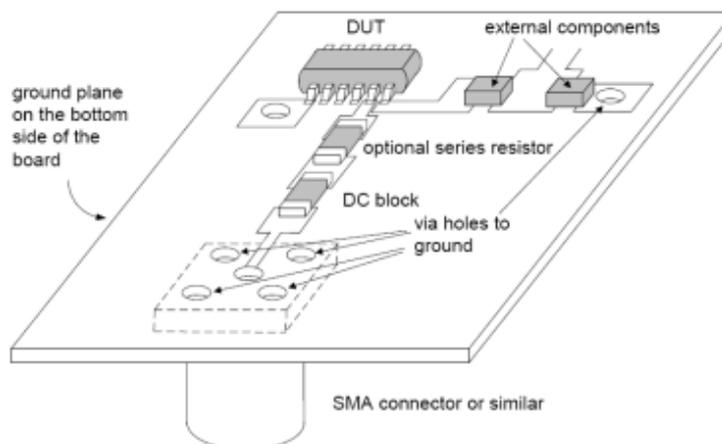


Figura 3.16 - Placa de teste para CI's com baixo número de pinos (35).

- b) Uma placa principal que possua trilhas com impedâncias iguais a 50 ohms integrada a uma placa auxiliar com trilhas de menores dimensões (abordagem usada em circuitos integrados com alto número de pinos). Neste caso, a placa auxiliar, contendo o circuito integrado sob teste, pode ser conectada a placa principal através dos pinos do conector de injeção, sendo a conexão de terra fornecida pelo contato entre o plano de terra da placa principal e o plano de terra da placa auxiliar. A Figura 3.17 apresenta o *layout* recomendado pela norma para este tipo de placa de teste.

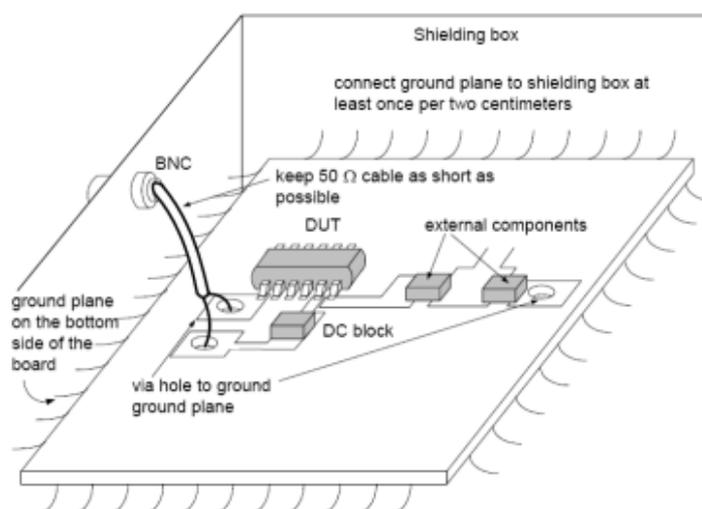


Figura 3.17 - Placa de teste para CI's com alto número de pinos (35).

# 4. Tecnologia dos Dispositivos Reconfiguráveis

## 4.1. Introdução

Determinadas aplicações computacionais desenvolvidas e implementadas tanto em *software* quanto em *hardware* necessitam freqüentemente 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 (do inglês, *Instruction Set Processor - ISP*), associado a uma memória. Os *ISP's* 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 PLA's (do inglês, *Programmable Logic Arrays*) e PAL's (do inglês, *Programmed Array Logic*), e principalmente através dispositivos reconfiguráveis como CPLD's (do inglês, *Complex Programmable Logic Device*) e FPGA's (do inglês, *Field Programmable Gate Array*), sendo este último adotado nesta dissertação como dispositivo sob teste.

De fato, 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).

## 4.2. Arquitetura dos FPGA's

Os FPGA's foram introduzidos em 1985 pela Xilinx Inc. (37). 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 (38).

Os FPGA's são chamados assim porque, além de disporem de uma estrutura similar aos PAL's<sup>2</sup>, também possuem uma disposição de portas semelhante às encontradas em ASIC (do inglês, *Application Specific Integrated Circuit*). Estas características fazem dos FPGA's excelentes dispositivos para o uso em prototipação de ASIC's, ou ainda, dependendo do projeto, em substituição a estes dispositivos.

A arquitetura básica dos FPGA's, apresentada na Figura 4.1, 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 FPGA's com outros dispositivos e/ou com o mundo externo, realizado através de blocos de entradas e/ou saídas também configuráveis pelo usuário.

---

<sup>2</sup> A tecnologia PAL (do inglês, *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.

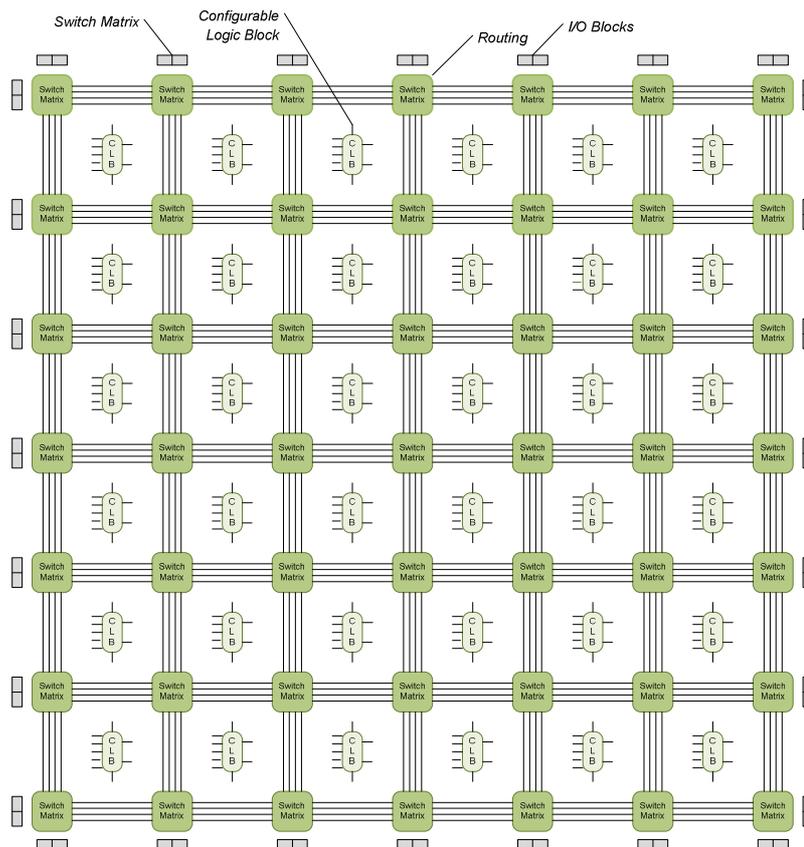


Figura 4.1 - Arquitetura básica de um FPGA(38).

Cada fabricante possui em sua linha de produtos FPGA's 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 os blocos lógicos (do inglês, *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 seqüenciais 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 freqüentemente 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 (39);

- **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 (em *ohms*), e capacitância (em *farads*), delimitam as características elétricas destes circuitos integrados (40). Existem basicamente três tipos de tecnologias de programação;
  1. **Tecnologia SRAM (do inglês, *Static Random Access Memory*):** usa uma célula de memória RAM (do inglês, *Random Access Memory*) estática para programar e controlar transistores de tecnologia CMOS (do inglês, *Complementary Metal Oxide Semiconductor*) ou multiplexadores (39). 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 (do inglês, *Programmable Read Only Memory*), EPROM (do inglês, *Erasable Programmable Read Only Memory*), EEPROM (do inglês, *Electrically Erasable Programmable Read Only Memory*) ou outro meio de armazenamento para prover o vetor de configuração do dispositivo (39). 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 (41). Existe um grande número de fabricantes que empregam a tecnologia SRAM em seus dispositivos a exemplo de Xilinx (37), Plessey (42), Algotronix (43), Altera (44), Concurrent Logic (45), Toshiba (46) e Atmel (47);
  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 (39). 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 as 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 20 volts. A tecnologia de antifusível é encontrada nos FPGA's produzidos pela Actel (48), Quicklogic (49) e Crosspoint (50);

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 (do inglês, *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 EPROM's, 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 (14). A tecnologia de porta flutuante baseada em EEPROM é usada em dispositivos fabricados pela AMD (51), Lattice (52) e Altera (53).

Os FPGA's 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 ALU's (do inglês, *Arithmetic and Logic Unit*), memórias e decodificadores.

Todas estas características de flexibilidade de arquitetura e reconfigurabilidade tornam os FPGA's excelentes dispositivos para projetos que necessitam de uma rápida inserção no mercado, sem que a preocupação com o custo de produção seja deixada de lado (38).

## 5. System on Chip (SoC)

### 5.1. Introdução

O 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. Em 1980, a maioria dos circuitos integrados (CI's) ditos complexos eram compostos por dezenas de milhares de transistores. Atualmente, é possível encontrar CI's que contenham dezenas de milhões de transistores (54). Entretanto, estima-se que até 2012 existam CI's contendo 4 bilhões de transistores, operando a uma frequência de 10GHz (55). Este avanço tecnológico permite a implementação de sistemas computacionais completos em um único CI, denominados SoC's (do inglês, *Systems-on-Chip*) (56) (57).

Os SoC's 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 (do inglês, *Micro-Electro-Mechanical Systems*) integrados no mesmo CI (58) (59) (55). 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, PDA's, aparelhos de jogos eletrônicos, aparelhos de DVD's, máquinas fotográficas digitais (integração de sensores ópticos com lógica digital e analógica), entre outros produtos (60) (61) (62). 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.

Tanto a evolução tecnológica quanto o curto *time-to-market* possuem grande efeito na diminuição do ciclo de vida dos produtos eletrônicos. Segundo R.A Bergamaschi, *et al* (63), esta 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 5.1 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.

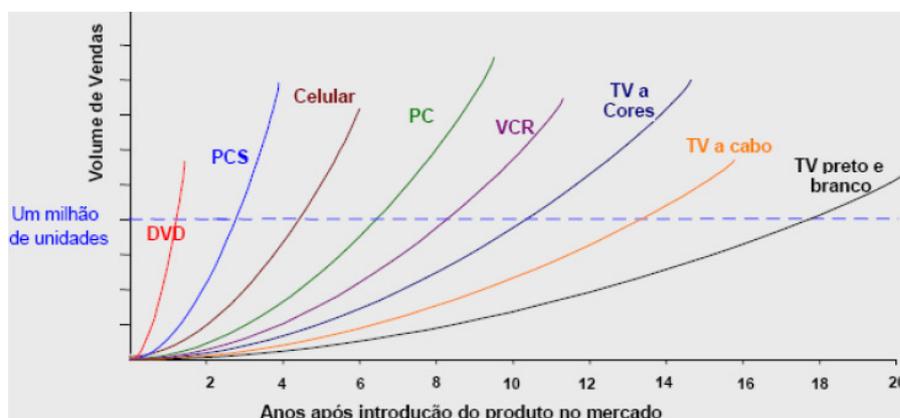


Figura 5.1 - Diminuição do ciclo de vida dos produtos eletrônicos (63).

Sendo assim, é possível 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. Dentro deste cenário, as companhias que desenvolvem e comercializam SoC's priorizam em seus projetos a utilização de núcleos de propriedade intelectual (do inglês, *Intellectual Property Cores*), ou simplesmente núcleos, a fim de aumentar a produtividade, minimizando o tempo de desenvolvimento, e conseqüentemente, o *time-to-market* de seus produtos (56).

## 5.2. Linguagens para Descrição de SoC's

O projeto concorrente de *hardware* e *software* é uma importante característica dos SoC's. 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 (64) e VHDL (*VHSIC Hardware Description Language*) (65), 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++ (66), são baseadas em um modelo de execução de instruções seqüenciais 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 (67), o SystemVerilog (64) e o SpecC (68). 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 SoC's 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*.

### 5.3. Arquitetura Genérica de um SoC

Como descrito 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 SoC's a partir da integração de núcleos heterogêneos. Segundo a Associação das Indústrias de Semicondutores (62) estima-se que em 2012, 90% da área dos circuitos integrados VLSI seja preenchida por núcleos. Logo, é possível inferir que a produção de SoC's 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 (55).

A Figura 5.2 apresenta 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 (69).

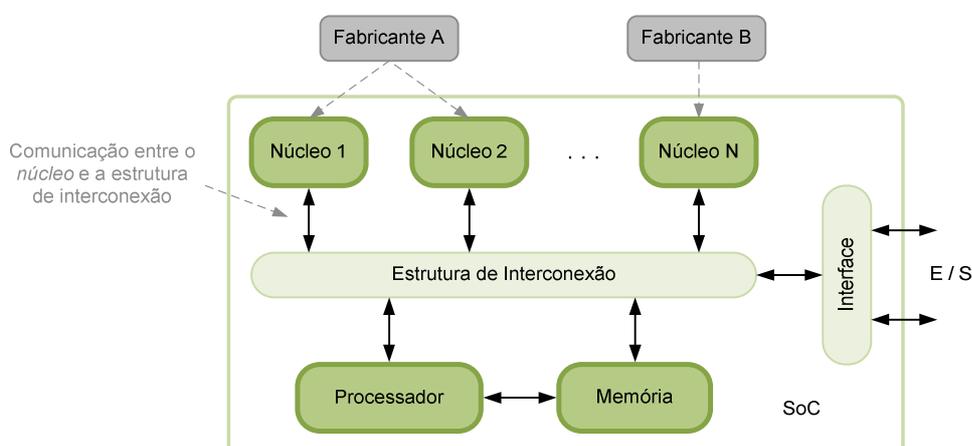


Figura 5.2 - Arquitetura genérica de um SoC (70) .

A arquitetura genérica de um SoC apresentada na Figura 5.2 é composta por núcleos IP que podem ser, por exemplo, módulos de telecomunicação, processadores DSP (do inglês, *Digital Signal Process*), decodificadores de MPEG2 (do inglês, *Motion Picture Experts Group Layer-2*), decodificadores MP3 (do inglês, *Motion Picture Experts Group Layer-3*), entre outros. Os núcleos IP, deste exemplo, provêm de fabricantes distintos (A, B) e estão integrados a uma estrutura de interconexão 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 (do inglês, *Universal Asynchronous Receiver Transmitter*) ou ainda uma porta USB (do inglês, *Universal Serial Bus*).

Gupta e Zorian (71) 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*.

- i. **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 projeto do seu SoC. Em contrapartida, esta flexibilidade repassa também ao projetista a responsabilidade do atendimento das restrições temporais (*timing*) do núcleo e do sistema (SoC).
- ii. **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.
- iii. **Hard Core** é geralmente uma descrição de um *layout* 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. Sendo 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 SoC através de duas estruturas de interconexão: fios ponto-a-ponto dedicados ou barramentos compartilhados, simples ou hierárquicos. Entretanto, antes de descrever estas estruturas citadas acima, torna-se necessário definir alguns conceitos que caracterizam as mesmas.

- **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 (72);
- **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.

## 5.4. Interconexão de Núcleos Baseada em Fios Ponto-a-Ponto Dedicados

Na interconexão baseada em fios dedicados (do inglês, *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.

Entretanto, o número de fios dedicados aumenta proporcionalmente ao número de núcleos adjacentes interligados. Caso pretenda-se interligar um núcleo a muitos outros através deste tipo de interconexão, pode-se gerar um congestionamento de fios (73). Tal característica pode ser considerada uma limitação à medida que o projeto SoC's pode contar com dezenas

ou centenas de núcleos (55) (72). Outro fator limitante está no fato que o projeto deste tipo de estrutura é específico e, portanto, a sua reutilização é bastante limitada. A Figura 5.3 apresenta uma estrutura genérica de interconexão de núcleos baseada em fios ponto-a-ponto.



Figura 5.3 - Estrutura genérica de interconexão baseada em fios ponto-a-ponto (70).

## 5.5. Interconexão de Núcleos Baseada em Barramentos Compartilhados

Uma estrutura de interconexão mais reutilizável é a baseada em barramentos de dados compartilhados (do inglês, *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 (74).

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 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. Por outro lado, um componente escravo é uma unidade que apenas responde às solicitações dos núcleos mestres, enviando ou recebendo dados (75). 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. Dentre os métodos existentes destacam-se o centralizado e o distribuído (75). 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 5.4 apresenta uma estrutura de interconexão baseada em barramento com árbitro.

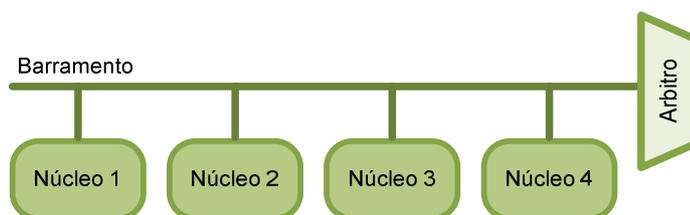


Figura 5.4 - Estrutura de interconexão em barramento com árbitro central (70).

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

- 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 (74) (76) (77).
- 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 (60), 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 (78), Avalon da Altera (79), CoreConnect da IBM (80) e OPB da Xilinx (81). 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.

Diversos autores prevêm que as estruturas de interconexão citadas acima tornar-se-ão 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* (82).

# 6. Controle Dinâmico do Ciclo de Trabalho do Sinal de Relógio (CDCDC)

## 6.1. Introdução

A integridade de sinal é a habilidade de um circuito e/ou sistema eletrônico gerar respostas corretas mesmo quando exposto à ambientes ruidosos (23). Entretanto, variações de temperatura e principalmente flutuações nas linhas de alimentação (quedas no  $V_{CC}$  e elevações no  $Gnd$ ) podem gerar a degradação de sinais, como por exemplo, a perda de sincronismo induzida pela diferença entre os atrasos ao longo dos diversos caminhos de propagação (críticos) e a rede de distribuição de sinal de relógio (do inglês, *clock*).

Neste sentido, se um determinado sinal, de um circuito digital síncrono, possuir um atraso de propagação muito elevado em seu caminho crítico, quando comparado ao período de sinal de relógio, este circuito poderá ser induzido a uma ocorrência de falha (diferença booleana), conforme apresenta a Figura 6.1.

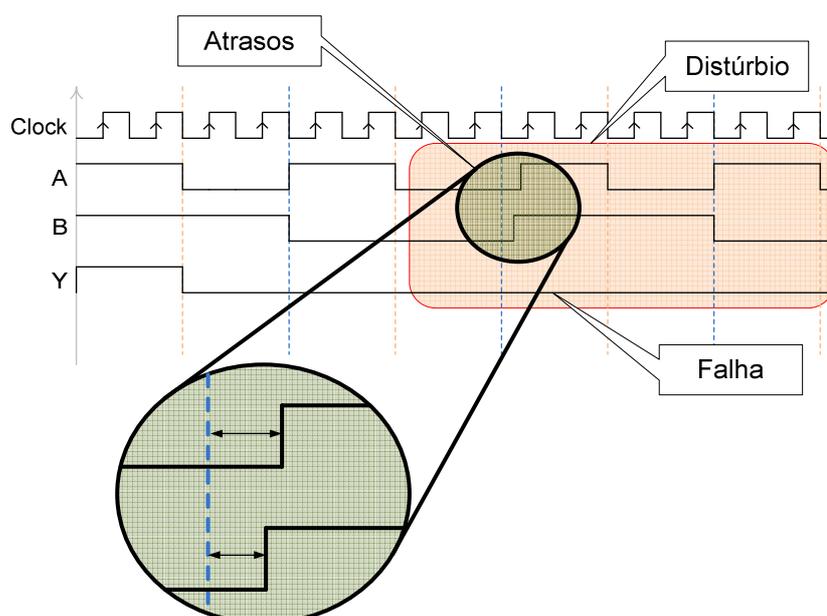


Figura 6.1 - Exemplo de falha de atraso entre dois sinais.

Estima-se que, em circuitos de alta integração e velocidade de chaveamento, flutuações da ordem de 10% nas linhas de alimentação podem gerar atrasos de propagação de aproximadamente 10% nos tempos dos sinais, causando assim erros funcionais (30).

Neste cenário, muitas soluções de projeto e fabricação têm sido propostas para garantir e/ou manter a integridade de sinais em *Systems-on-Chip* (SoC), a exemplo de: Modelagens de *layout* 3D (83); simulações RLC em redes de alimentação de *chips* (28); emprego de capacitores de desacoplamento (26) (84); redução nas quedas resistivas de tensão (28) (85); adição de *buffers* (83), blindagens de trilhas (86) e monitoração das atividades de emissões eletromagnéticas internas ao chip (87). Contudo, nenhuma destas técnicas garante a perfeita solução para o efeito de dessincronismo descrito acima.

Diante disto, e a fim de garantir a integridade de sinal de circuitos e/ou sistemas digitais síncronos, projetistas e usuários reduzem a frequência nominal do sinal de relógio para níveis onde os atrasos de propagação, provenientes de distúrbios nas suas linhas de alimentação, não tenham influência sobre o comportamento do circuito e/ou sistema. Embora esta medida seja bastante confiável e eficiente, para algumas aplicações críticas, cujo desempenho é um parâmetro absolutamente relevante, esta solução se torna inviável e intolerável (1).

Com o objetivo de garantir a integridade de sinal e conseqüentemente o funcionamento de SoC's sem degradação do desempenho, isto é, com a manutenção da frequência nominal do sinal de relógio, uma nova técnica denominada Controle Dinâmico do Ciclo de Trabalho (*do inglês, Clock Duty Cycle Dynamic Control - CDCDC*) foi proposta no INESC<sup>3</sup> por João Paulo Cacho Teixeira, *et al* (1).

O princípio de funcionamento desta metodologia está na adaptação dinâmica do sinal de relógio (*do inglês, clock*) aos sinais do circuito e/ou sistema digital síncrono com problemas de atraso de propagação (*do inglês, delay*), através da modulação do ciclo de trabalho (*do inglês, duty-cycle*). Para tal, as linhas de alimentação são monitoradas, e na presença de flutuações de tensão superiores aos níveis definidos como limite (*do inglês, threshold*), a modulação do ciclo de trabalho do sinal de relógio é ativada aumentando assim a integridade de sinal e garantindo um funcionamento do circuito e/ou sistema sem falhas (88).

---

<sup>3</sup> O INESC (Instituto de Engenharia de Sistemas e Computadores) de Porto – Portugal é uma associação privada que desenvolve atividades de pesquisa e desenvolvimento, consultoria, formação avançada e transferência de tecnologia nas áreas de Telecomunicações e Multimídia, Sistemas de Energia, Sistemas de Produção, Sistemas de Informação e Comunicação e Optoeletrônica.

## 6.2. Modulação do Ciclo de Trabalho

Conforme referido, reduções nas linhas de alimentação ( $V_{CC}$ ) podem acentuar os atrasos de propagação dos sinais de circuitos digitais síncronos levando-os a erros funcionais, principalmente quando a frequência do sinal de relógio destes circuitos é mantida constante no seu valor nominal.

Além de oscilações nas linhas de alimentação ( $V_{CC}$ ), a instabilidade em circuitos síncronos também pode ser proveniente de fenômenos na referência de tensão ( $Gnd$ ). Quando nós internos a lógica de um dispositivo mudam o seu estado, uma carga remanescente ( $C_L$ ) é drenada através da malha de referência ( $Gnd$ ). A corrente elétrica associada a esta carga é adicionada as correntes de chaveamento do circuito que fluem do  $V_{CC}$  para as interconexões de terra, devido aos simultâneos chaveamentos da rede N e P. Esta corrente total resultante fluindo através da malha de terra induz uma variação de tensão, denominada *ground bounce* ( $V_{GB}$ ) (89). O pior caso deste fenômeno ocorre quando um grande número de nós do circuito é chaveado simultaneamente. Neste caso, as correntes de chaveamento resultantes, quando somadas, fluem através da malha de referência de terra reduzindo a integridade de sinal e aumentando a probabilidade de mau funcionamento do sistema. Por uma questão de simplificação, tanto as oscilações nas linhas de alimentação ( $V_{CC}$ ) como na de referência de terra ( $Gnd$ ) são normalizadas usando o parâmetro *gamma* conforme a equação (6.1).

$$\gamma(V_{CC}) = \frac{\Delta V_{CC}}{V_{CC\ NOM}} \quad \text{ou} \quad \gamma(Gnd) = \frac{\Delta Gnd}{V_{CC\ NOM}} \quad (6.1)$$

Onde:  $\Delta V_{CC}$  é a diferença entre o valor de  $V_{CC}$  nominal ( $V_{CC\ NOM}$ ) e a queda de tensão de  $V_{CC}$ , e  $\Delta Gnd$  é a diferença entre a elevação da referência de terra e a referência propriamente dita ( $Gnd=0V$ ).

A Figura 6.2 apresenta os resultados de simulação SPICE, realizadas no INESC, para uma cadeia de 77 inversores desenvolvida em tecnologia CMOS 130nm, com tensão nominal igual a 1,2V e conectada a um *flip-flop* tipo D (sensível a níveis lógicos). Esta cadeia de inversores foi adotada para ilustrar o comportamento temporal do chaveamento de sinais em caminhos longos, isto é, em caminhos com grande número de níveis lógicos.

A Figura 6.2 (a) apresenta o colapso funcional do circuito com 83,33% da tensão nominal ( $\gamma(V_{CC})=16,67\%$ ) e ciclo de trabalho do sinal de relógio igual a 50%. Já na Figura 6.2 (b), onde o ciclo de trabalho do sinal de relógio é igual a 80%, o colapso funcional do circuito ocorre somente com 73,33% da tensão nominal ( $\gamma(V_{CC})=26,67\%$ ).

Conforme esperado, o efeito de dessincronização entre os sinais e as linhas de sincronismo (*clock*) ocorre posteriormente quando o circuito está sob a técnica de modulação do ciclo de trabalho do sinal de relógio (prolongamento). É possível verificar, na Figura 6.2 (a) e (b), que o instante de dessincronização ocorre, respectivamente, em  $t=31\text{ns}$  ( $V_{CC}=0,98\text{V}$ ) e  $t=40\text{ns}$  ( $V_{CC}=0,88\text{V}$ ) (88).

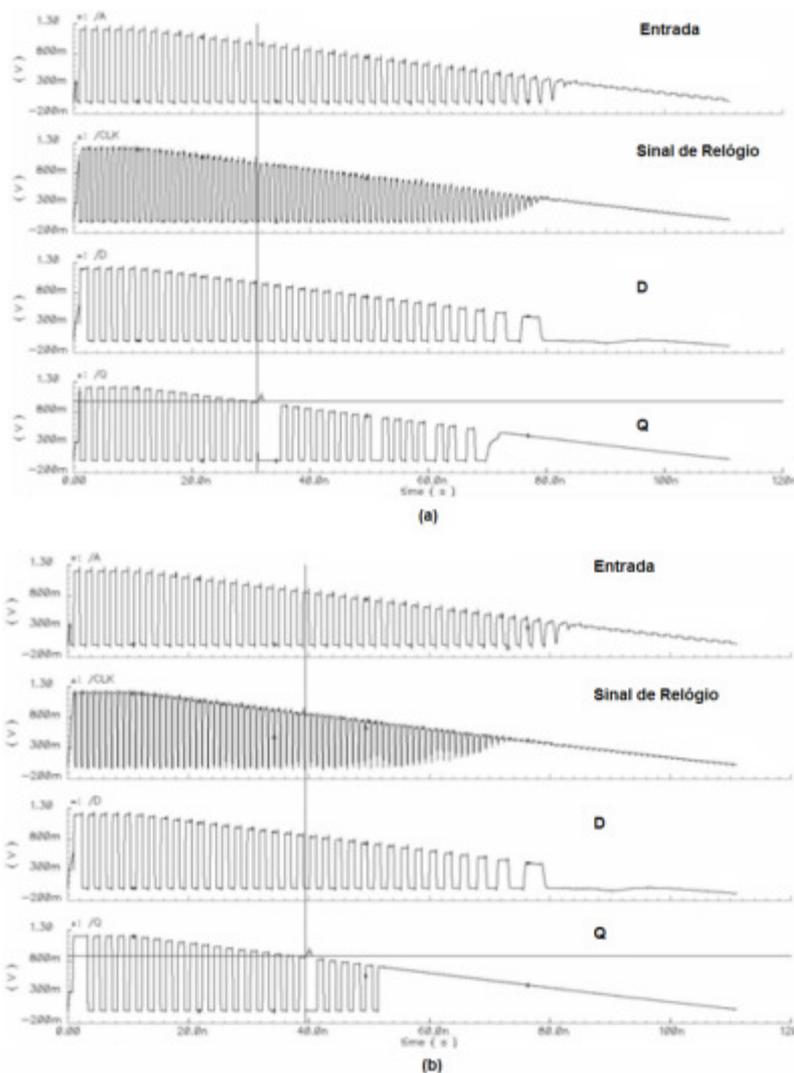


Figura 6.2 - Cadeia de 77 inversores com variação de tensão de alimentação (88).

### 6.3. Descrição da Técnica CDCDC

A metodologia proposta do Controle Dinâmico do Ciclo de Trabalho (CDCDC) é baseada nos seguintes pressupostos:

- O ciclo de trabalho do sinal de relógio é geralmente definido como sendo de 50%, para minimizar os efeitos do *jitter* associado às longas redes de distribuição de sinal de relógio e aos efeitos de variações de parâmetros

durante o processo de fabricação. Além de também permitir uma distribuição linear simétrica ao longo do tempo para circuitos projetados com elementos de memória ativos por bordas (do inglês, *edge triggered*) ou por níveis lógicos (do inglês, *level triggered*) (88).

- O ciclo de trabalho do sinal de relógio pode, contudo, ser temporariamente modificado para um ou mais subcircuitos, sendo que o valor original de 50% é mantido para as demais partes do *System-on-Chip* (SoC). Este fato não induz ao SoC um erro funcional por perda de sincronismo se, e somente se, a frequência do sinal de relógio for mantida a mesma para todos os subcircuitos do SoC, os quais também devem operar de forma síncrona (1).
- A modulação do ciclo de trabalho do sinal de relógio tem limites definidos. A título de exemplo, pode se considerar que um Circuito de Reconstituição de Relógio (CRR) é baseado em blocos do tipo malha de captura de fase (do inglês, *Phase Locked Loop* – PLL). O funcionamento correto deste CRR depende da extensão do prolongamento do ciclo de trabalho do sinal de relógio. Aliado a isto, a estabilidade do bloco PLL depende da margem, a qual deve ser igual ou superior a 30° e que corresponde a uma variação de aproximadamente 8,33% (relativa à 360°). Estas condições limitam o prolongamento do ciclo de trabalho do sinal de relógio de 50% (valor nominal) até aproximadamente 91,67% (valor máximo teórico alongado), garantindo assim a operação correta do PLL e conseqüentemente do circuito e/ou sistema. Com o objetivo de simplificar a definição da margem de fase, admite-se, em circuitos PLL, que o valor limite de variação é de 10%, sendo a faixa de excursão do ciclo de trabalho limitada em até 90% (88).

No sentido de implementar a metodologia proposta, algumas funcionalidades internas ao chip devem ser adicionadas. O sistema CDCM (do inglês, *Clock Duty-Cycle Modulation*) (88) implementa tais funcionalidades, monitorando localmente variações de  $V_{CC}$  e conseqüentemente acionando o CDCDC. A arquitetura da rede de alimentação, bem como a forma na qual as diferentes alimentações dos núcleos e/ou módulos dos SoC's são constituídos, irá determinar quantos moduladores de ciclo de trabalho do sinal de relógio (do inglês, *Clock Duty-Cycle* – CDC) devem ser inseridos.

A Figura 6.3 apresenta duas arquiteturas do sistema CDCM. A primeira, denominada WA (do inglês, *Work Always*), possui somente um bloco de lógica de prolongamento de sinal de relógio (do inglês, *Clock Stretching Logic* – CSL), utilizando apenas um circuito simples

que introduz um pequeno atraso de propagação ao sinal de relógio. Nesta arquitetura, o sinal de relógio modulado apresenta um prolongamento mínimo (ligeiramente superior a 50%) quando o valor de  $V_{CC}$  é nominal. Entretanto quando  $V_{CC}$  diminui, o bloco CSL prolonga o CDC de acordo com a amplitude da redução de tensão.

Já a segunda arquitetura apresentada na Figura 6.3 (b) é denominada WN (do inglês, *Work when Needed*) e possui características mais complexas. Sobre condições normais de tensão de alimentação, o multiplexador (MUX) é selecionado para receber a saída do PLL (ou gerador de *clock*). Entretanto, quando o bloco sensor de variação de tensão (do inglês, *built-in power supply voltage transient sensor – BIPS*) detecta alguma oscilação em  $V_{CC}$  (ou elevação na referência de terra) abaixo (ou acima) de um determinado limiar de tensão (*threshold*), o bloco CSL, inserido na entrada do multiplexador, é selecionado. Um *flip-flop* tipo D também é utilizado para garantir que o chaveamento entre os blocos PLL e CSL (e vice-versa) é executado sem a ocorrência de falhas na saída do multiplexador. Quando o ruído transiente na rede de alimentação cessa, o modulador CSL inicia gradualmente o restabelecimento do ciclo de trabalho do sinal de relógio (50%) e o BIPS chaveia o multiplexador selecionando o sinal do PLL.

Embora esta arquitetura, tenha um menor impacto sobre o desempenho do circuito com  $V_{CC}$  nominal quando o valor  $V_{CC}$  situa-se abaixo do limiar de tensão (*threshold*), ela exige um número maior de circuitos para a sua implementação e acrescenta um considerável atraso de propagação para o sinal de relógio modulado. Tendo em vista isto, esta arquitetura só pode ser utilizada quando este atraso de propagação for inexpressivo quando comparado ao período do sinal de relógio.

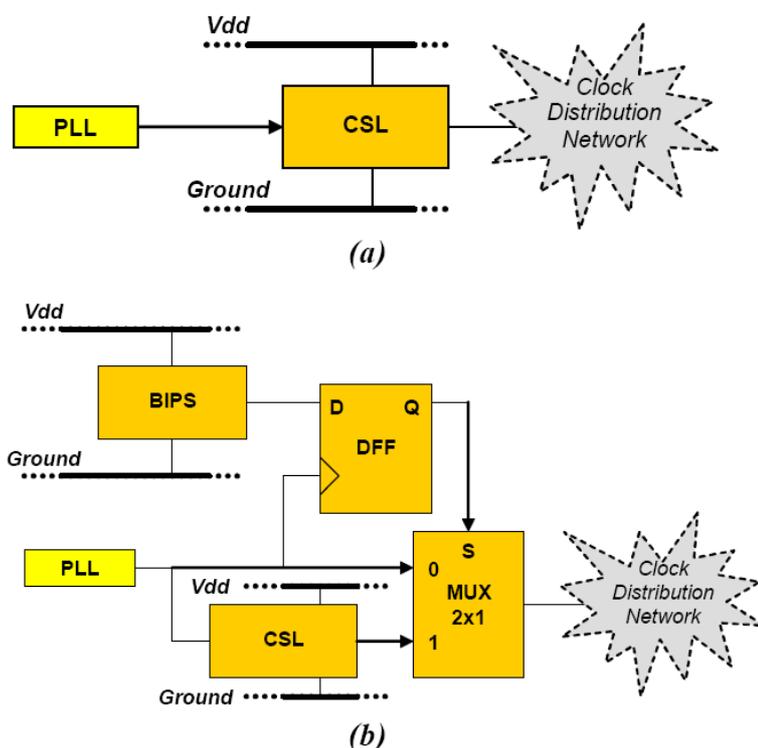


Figura 6.3 - Arquiteturas básica do sistema CDCM (88).

Muitas arquiteturas podem ser utilizadas na implementação dos blocos BIPS e CSL. A Figura 6.4 apresenta um exemplo de uma possível implementação em tecnologia CMOS para parte do sensor de variação de tensão do bloco BISP (88). O princípio de funcionamento deste circuito é dado pelo divisor de tensão formado pela cadeia de transistores M1-M3 que polariza o transistor M5 próximo a sua região de corte. Neste contexto, quando a diferença de potencial, dada por  $V_{CC}-Gnd$ , é reduzida abaixo do *threshold*, M5 é forçado para a sua região de corte, fazendo com que a saída do pseudo-inversor CMOS M4-M5 atinja  $V_{CC}$ . Os dois inversores restantes atuam neste circuito como *buffers*, sendo a saída “ $V_{CC}$  Sensor” usada para gerar o sinal de seleção do multiplexador (através do *flip-flops*).

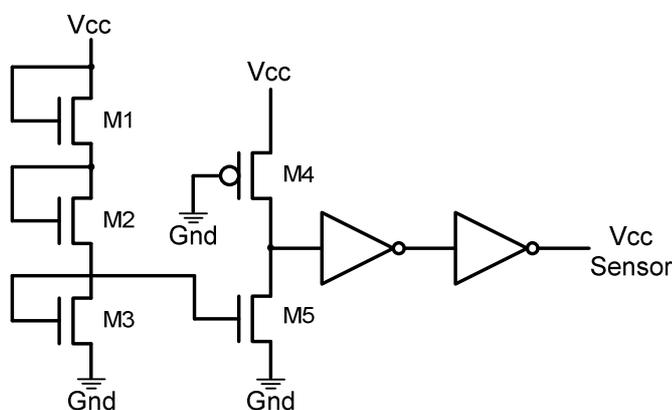


Figura 6.4 - Exemplo de arquitetura para o bloco BIPS (88).

A Figura 6.5 apresenta o exemplo de uma possível implementação do bloco CSL (88), cujo princípio de funcionamento é baseado no inversor CMOS formado pelos transistores M1-M2. O transistor M2 tipo PMOS tem sua capacidade de condução reduzida devido a sua associação com o transistor M3. Esta redução de velocidade de condução associada a uma redução de  $V_{CC}$  gera, sem alterar o período, uma modulação do ciclo de trabalho do sinal de relógio. Esta topologia de circuito é utilizada para atrasar a transição sinal de relógio do nível lógico alto para o baixo.

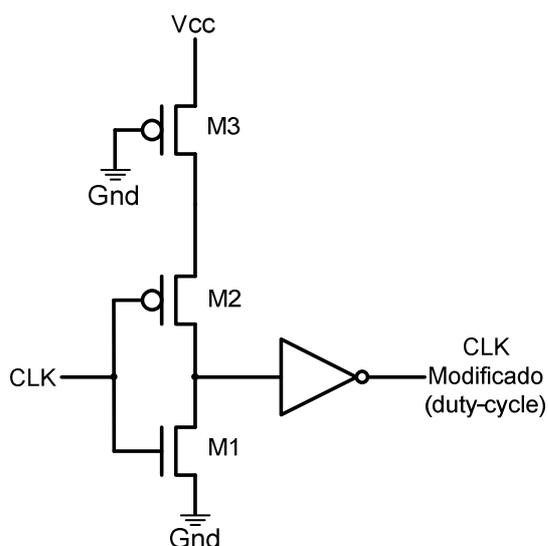


Figura 6.5 - Exemplo de arquitetura para o bloco CSL (88).

Na Figura 6.6 os resultados de simulação SPICE, realizadas no INESC em tecnologia CMOS 350nm, apresentam a capacidade do bloco CSL em modular o ciclo de trabalho do sinal de relógio proporcionalmente as reduções de tensão  $V_{CC}$ .

A Figura 6.6 (a) e (b) apresenta, respectivamente, o prolongamento e o encurtamento do ciclo de trabalho, gerado pelo bloco CSL sujeito a variações de tensão na faixa de 3,3V (valor nominal) a 1,1V, de um sinal relógio de 200 MHz.

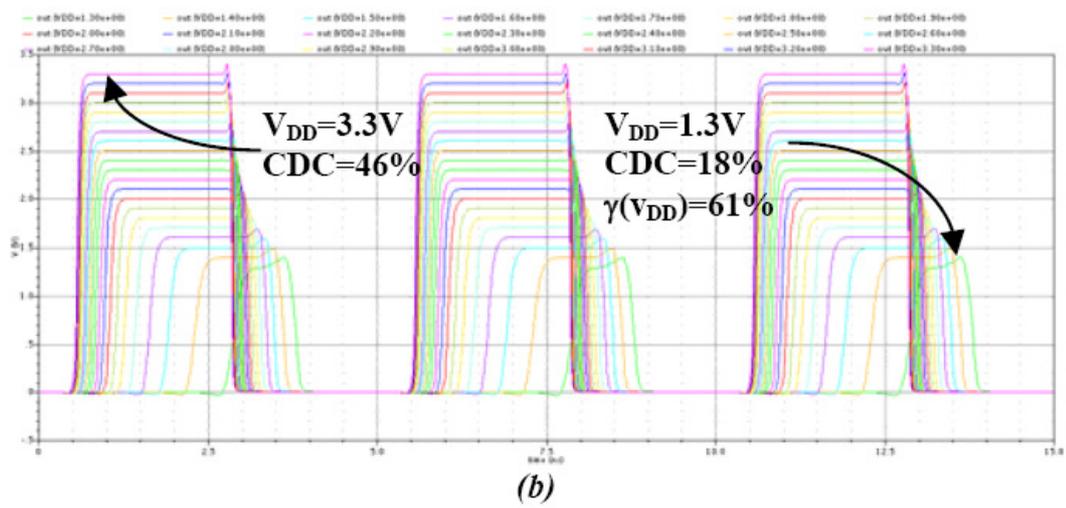
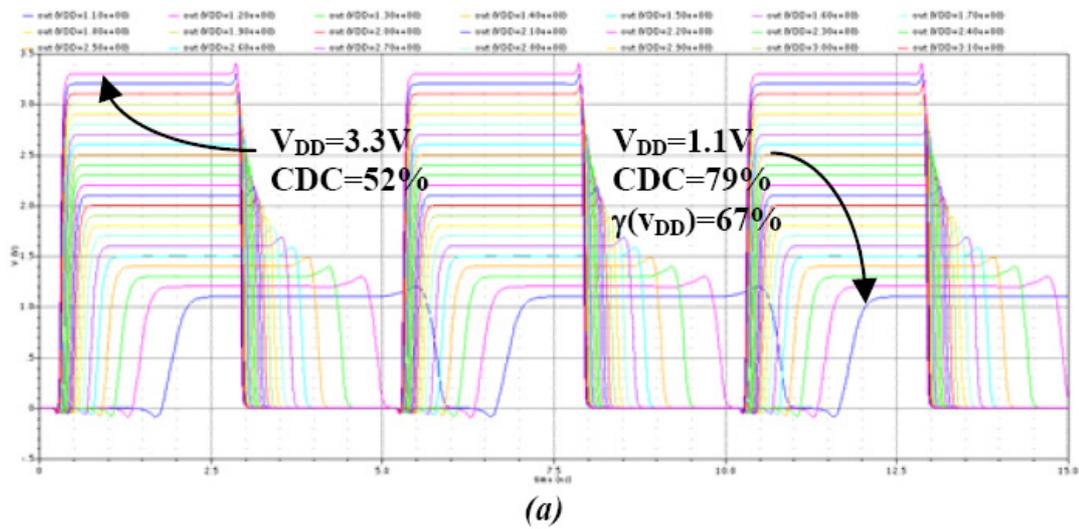


Figura 6.6 - Simulações do bloco CSL para variações de  $V_{CC}$  (89).

## **Parte II - Metodologia**

## 7. Estudo de Caso I

### 7.1. Introdução

Após as fases preliminares de revisão bibliográfica referenciadas nos capítulos e seções anteriores, optamos pela validação da técnica de Controle Dinâmico do Ciclo de Trabalho (CDCDC) em *Systems-on-Chip* (SoC's) como foco principal desta dissertação.

Entretanto o projeto e desenvolvimento de um circuito integrado em tecnologia CMOS contendo na sua estrutura a técnica CDCDC (88)(89), além de necessitar de um tempo de projeto incompatível com o tempo disponível para o desenvolvimento desta dissertação (doze meses), também necessitava de recursos técnicos e instrumentais com elevados custos financeiros indisponíveis em nosso laboratório (Laboratório de Sistemas, Sinais e Computação – SiSC).

Neste cenário, a implementação da técnica de Controle Dinâmico do Ciclo de Trabalho para robustez de *Systems-on-Chip* sujeitos a flutuações no barramento de alimentação em FPGA's do tipo SRAM tornou-se uma alternativa bastante viável sob o ponto de vista técnico e econômico, sendo sua metodologia apresentada a seguir.

### 7.2. Circuito Clock Manager

O circuito *Clock Manager*, descrito em VHDL (do inglês, *VHSIC Hardware Description Language*), foi desenvolvido para realizar a modulação dinâmica do ciclo de trabalho do sinal de relógio de circuitos digitais síncronos embarcados em dispositivos de lógica reprogramável como FPGA's.

Este circuito de modulação tem a sua estrutura lógica básica formada por contadores, sensíveis a transições lógicas (bordas), que ao dividirem a frequência do sinal de relógio de referência, possibilitam a geração de sinais de sincronismo com uma frequência reduzida (quatro vezes menor que a referência), mas com um ciclo de trabalho variável dinamicamente. A Figura 7.1 apresenta um diagrama temporal da lógica de construção dos sinais do circuito

*Clock Manager* onde: (a) é o sinal de relógio de entrada (referência); (b) é o sinal de relógio, gerado pelo circuito *Clock Manager*, com ciclo de trabalho igual a 50%; (c) é o sinal de relógio, gerado pelo circuito *Clock Manager*, com ciclo de trabalho igual a 25%; (d) é o sinal de relógio, gerado pelo circuito *Clock Manager*, com ciclo de trabalho igual a 75%; e (cont) é o estado atual do contador interno da lógica de geração dos sinais (b), (c) e (d).

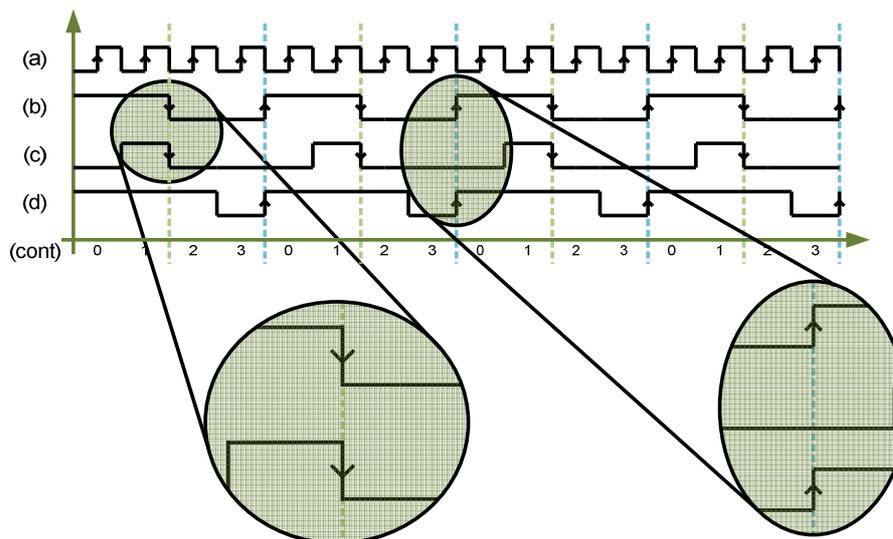


Figura 7.1 - Lógica do circuito descrito em VHDL “Clock Manager”.

Também é possível verificar nos detalhes da Figura 7.1 que os sinais (c) e (d) são síncronos ao sinal (b) em instantes de tempo distintos. Este comportamento é proveniente do fato destes sinais serem associados a circuitos síncronos com células de memória (*flip-flop's*) de características diferentes. O sinal (c), com ciclo de trabalho igual a 25%, é utilizado em circuitos com *flip-flop's* sensíveis a borda de subida do sinal de relógio, e por este motivo, tem a sua transição de nível lógico baixo para o alto retardada em relação à borda de subida do sinal (b), ao contrário das bordas de descida, dos sinais (b) e (c), que são mantidas síncronas.

Já o sinal (d), com ciclo de trabalho igual a 75%, possui um raciocínio inverso e é inserido em circuitos síncronos compostos por *flip-flop's* sensíveis a bordas de descida, e por este motivo, tem a sua transição do nível lógico alto para o baixo atrasada em relação à borda de descida do sinal (b), ao contrário das bordas de subida, dos sinais (b) e (d), que são mantidas síncronas.

A Figura 7.1 apresenta a descrição funcional dos pinos que compõe o circuito de modulação dinâmica do ciclo de trabalho do sinal de relógio *Clock Manager*.

Tabela 7.1 - Descrição funcional dos pinos do circuito *Clock Manager*.

Nome	Descrição
clock_in	Pino de entrada da referência de sinal de relógio do circuito (oscilador externo).
duty_selecao	Pinos de entrada para seleção (dois <i>bits</i> ) do ciclo de trabalho do sinal de relógio da saída variável (clock_b_pin) do circuito.
clock_a_pin	Pino de saída do sinal de relógio com ciclo de trabalho fixo e igual a 50%.
clock_b_pin	Pino de saída do sinal de relógio com ciclo de trabalho variável igual a 50, 25 ou 75%.

A Tabela 7.2 apresenta as combinações de entrada do circuito *Clock Manager* para seleção e geração dos sinais de relógio com ciclo de trabalho variável.

Tabela 7.2 - Combinações de entrada do circuito *Clock Manager*.

Entradas		Ciclos de Trabalho de Saída	
duty_selecao(1)	duty_selecao(0)	clock_a_pin	clock_b_pin
0	0	50%	50%
0	1	50%	25%
1	0	50%	75%

### 7.3. Circuito de Teste B02

No sentido de validar o circuito *Clock Manager*, juntamente com o conceito de controle e ajuste dinâmico do ciclo de trabalho do sinal de relógio em *System-on-Chip* (SoC) selecionamos um circuito de teste para análises preliminares. Este circuito, denominado B02, foi desenvolvido pelo *CAD Group* do Instituto Politécnico de Torino na Itália e pertence a um conjunto de circuitos descritos em VHDL para testes (*benchmark's*) denominado de ITC'99 Benchmark (2nd release) (90).

Além de ser um circuito digital síncrono e gerar saídas iguais quando excitado com o mesmo conjunto de entradas (requisitos básicos para o projeto), o circuito B02 possui características estruturais de uma máquina de estados finitos (do inglês, *Finite State Machine* – FSM), conforme apresenta a Figura 7.2, e sua aplicação funcional é o reconhecimento de números BCD (do inglês, *Binary Coded Decimal*) (90). A Tabela 7.3 apresenta um resumo das principais características deste circuito.

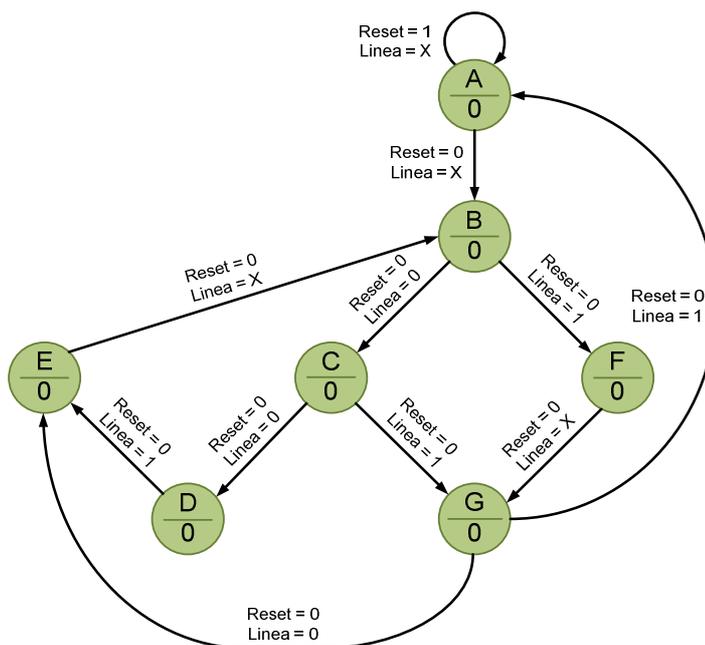


Figura 7.2 - Fluxograma do circuito de teste B02.

Tabela 7.3 - Características do Circuito Teste B02.

Número de Entradas	Número de Saídas	N.º de Portas Lógicas	N.º de Níveis Lógicos do Caminho Crítico
3 entradas (1 para dados, 1 para relógio e 1 para reset)	1	42 Portas Lógicas + 4 FF's.	20

## 7.4. Circuito de Teste B02 – Modificado

O circuito B02 possuía, entre suas características originais, uma única entrada de sinal de relógio. Tendo em vista esta característica, e que a técnica de CDCDC tem como objetivo, para aumentar a robustez do circuito sob teste, variar o ciclo de trabalho do sinal de relógio de apenas algumas partes e/ou blocos do circuito (caminho crítico), mantendo as demais regiões do circuito com os seus sinais de relógio inalterados. O circuito B02 foi modificado, sendo acrescentado a este uma entrada para sinal de relógio com ciclo de trabalho variável.

Além do circuito B02 originalmente possuir apenas uma entrada de sinal de relógio, seus sinais de saída também eram bastante limitados, isto é, o circuito possuía apenas um *bit* de sinal de saída, tornando a conferência das seqüências dos estados do circuito bastante difícil e com um elevado risco de um mascaramento de falhas.

Para solucionar esta limitação estrutural, também adicionamos, aos sinais de saída do circuito B02, dois *bits* gerando assim uma assinatura única para cada estado da FSM.

A Figura 7.3 apresenta o fluxograma do circuito de teste B02 – Modificado onde o estado atual da FSM possui uma assinatura de três *bits* única, e os estados futuros são dependentes do estado atual e dos sinais de entrada *Reset* e *Linea*.

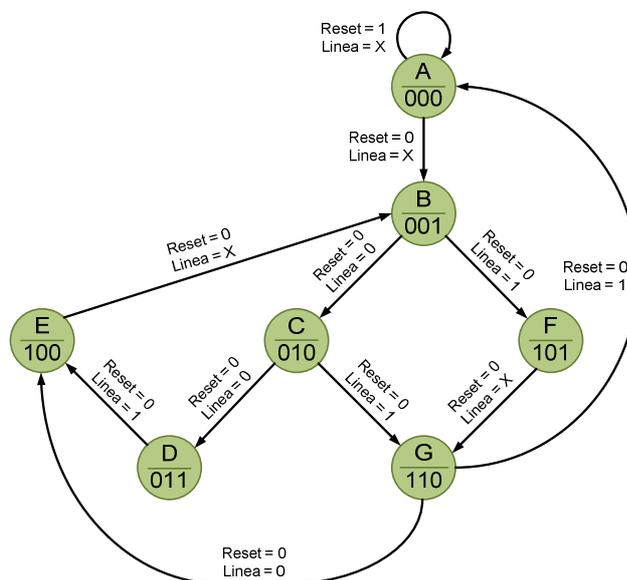


Figura 7.3 - Fluxograma do circuito de teste B02 – Modificado.

As alterações no circuito B02 modificaram não apenas o número de entradas e saídas do circuito, mas também o número de portas lógicas, o número de *flip-flop's* e o número de níveis lógicos do circuito (caminho crítico). A Tabela 7.4 apresenta as principais características do circuito B02 – Modificado.

Tabela 7.4 - Características do Circuito Teste B02 – Modificado.

Número de Entradas	Número de Saídas	N.º de Portas Lógicas	N.º de Níveis Lógicos do Caminho Crítico
4 entradas (1 para dados, 2 para relógio e 1 para reset)	3	66 Portas Lógicas + 6 FF's.	22

Estas diferenças quantitativas do circuito B02 – Modificado são atribuídas ao fato das alterações realizadas na estrutura do circuito de teste também alteraram a lógica de decodificação de saídas dos estados. Justificando assim o número divergente de portas lógicas e de *flip-flop's* do circuito, bem como o número de níveis lógicos no caminho crítico do circuito quando comparado ao circuito B02 original.

Embora os circuitos, original e modificado, tenham algumas diferenças estruturais, salientamos que as modificações realizadas não alteraram o comportamento, nem tampouco a lógica seqüencial dos estados do circuito B02. Estas modificações apenas permitiram uma melhor análise dos estados do circuito e teste, evitando assim um indesejado mascaramento de falhas.

## 7.5. Plataforma Spartan-3 Starter Kit Board

As fases iniciais de projetos envolvendo *Systems-on-Chip* são caracterizadas pelas constantes mudanças estruturais e comportamentais tanto do *software* quanto do *hardware*.

Aliado a isto, fatores decisivos como o elevado custo financeiro de componentes eletrônicos e da manufatura de placas de circuito impresso (do inglês, *printed circuit board*), o também elevado tempo de projeto de esquemático e *layout* tornam absolutamente recomendável o emprego de plataformas de desenvolvimento comerciais, evitando assim desnecessários desperdícios de recursos financeiros e humanos.

Seguindo esta linha de raciocínio, para os testes preliminares desta dissertação foi utilizada uma plataforma comercial produzida pela Xilinx Inc. denominada *Spartan-3 Starter Kit Board* (91).

Com um FPGA da família *Spartan-3* este kit de desenvolvimento, apresentada na Figura 7.4, possui características bastante adequadas para o desenvolvimento de projetos de *Systems-on-Chip*, dentre as quais destacam-se:

- FPGA *Spartan-3* XC3S200 (200k portas, 12 multiplicadores de 18-bits, 216 *Kbits* de *block* RAM interna e frequência de operação máxima de 500MHz);
- Memória *Flash* serial de 2Mbit (XCF02S);
- Memória SRAM de 1Mbyte (10ns de tempo de acesso).
- 8 chaves, 4 botões, 9 LED's, e 4 dígitos de *displays* de 7 segmentos;
- Comunicação serial padrão RS-232;
- Oscilador de 50Mhz;
- Conector VGA para vídeo;
- Conectores PS/2 para mouse/teclado;
- Conectores de expansão;
- 3 Reguladores de tensão (1,2v – 2,5v – 3,3v);
- Conector JTAG para programação e *debug*;

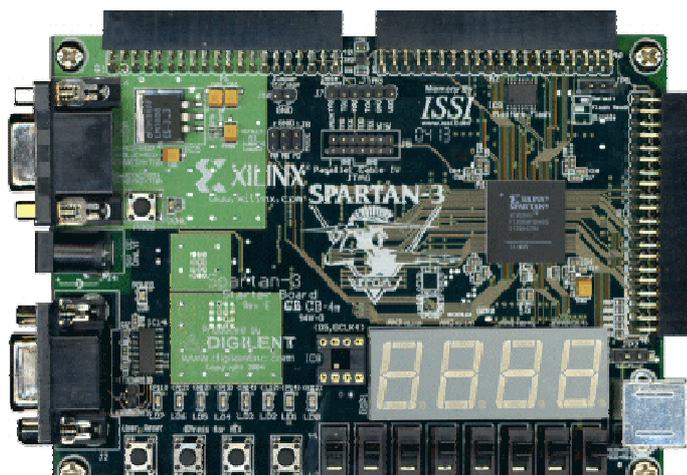


Figura 7.4 - Kit de desenvolvimento *Spartan-3 Starter Kit Board* (91).

## 7.6. *Setup* de Teste Completo

Para a validação da técnica de Controle Dinâmico de Ciclo de Trabalho do sinal de relógio foi realizada a integração dos módulos desenvolvidos nas etapas anteriores do projeto.

A Figura 7.5 apresenta o diagrama de blocos do *setup* de teste completo, onde é possível verificar que duas plataformas de desenvolvimento *Spartan-3 Starter Kit Board* foram utilizadas para a realização dos testes. A primeira plataforma de desenvolvimento (*Kit 1*), composta por um oscilador externo e pelo circuito *Clock Manager* embarcado em seu FPGA, gera os sinais de relógio e as seqüências de entradas conforme as frequências e os ciclos de trabalho indicados na Figura 7.5, já a segunda plataforma (*Kit 2*) possui, embarcado em seu FPGA, o circuito de teste B02 – Modificado e tem as suas linhas de alimentação sujeitas a flutuações de tensão através de uma fonte de tensão externa.

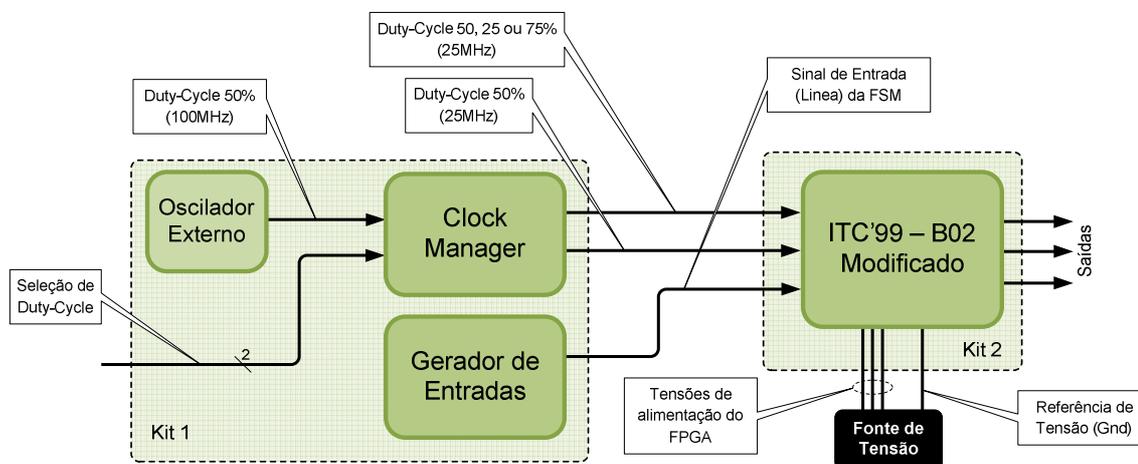


Figura 7.5 - Diagrama de blocos do *setup* de teste completo.

## 7.7. Níveis de Tensão da Plataforma de Desenvolvimento

A família de FPGA's *Spartan-3* necessita, para o seu bom funcionamento, de três níveis de tensão, sendo que os dois primeiros, com amplitude igual a 2,5 e 3,3 volts, alimentam a lógica de entrada e saída do dispositivo (periferia), já o último, com amplitude igual a 1,2 volts, alimenta o núcleo do dispositivo (91).

Para fornecer de uma forma robusta e confiável estes três níveis de tensão, ao FPGA e demais componentes da placa, a plataforma desenvolvimento *Spartan-3 Starter Kit Board* dispõe em seu esquemático de três reguladores de tensão contínua, alimentados por uma fonte externa (5volts). A Figura 7.6 um diagrama de blocos da lógica de alimentação da desta plataforma de desenvolvimento.

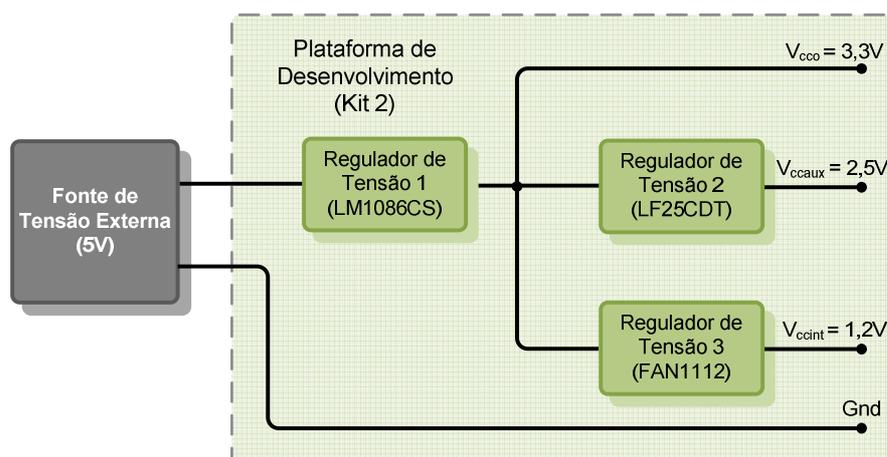


Figura 7.6 - Lógica de alimentação da plataforma de desenvolvimento.

## 7.8. Redução da Tensão de Entrada

Com o objetivo de tornar os sinais internos do circuito sob teste *B02-Modificado* com tempos de propagação mais lentos (em níveis lógicos), e conseqüentemente mais susceptíveis a falhas, reduzimos o nível de tensão de entrada da plataforma de desenvolvimento onde este circuito estava embarcado (*Kit 2*). Esta redução dos níveis tensão de alimentação, relativa à tensão nominal de entrada (5 volts) da plataforma de desenvolvimento, foi realizada de forma gradual e progressiva conforme apresenta a Tabela 7.5

Tabela 7.5 - Redução percentual da tensão de entrada.

Redução Percentual (%)	Tensão Aplicada (V)
5	4,75
10	4,50
15	4,25
20	4
25	3,75
30	3,5
35	3,25

O FPGA *Spartan-3* (XC3S200) juntamente com os demais componentes da plataforma de desenvolvimento *Spartan-3 Starter Kit Board* se mostraram bastante robustos quando submetidos a estas variações de tensão de entrada, pois o circuito sob teste B02 – Modificado, embarcado neste dispositivo, não apresentou falhas funcionais durante a execução destes testes.

## 7.9. Redução da Tensão do Núcleo do FPGA

Tendo em vista a não ocorrência de falhas funcionais no circuito de teste B02 – Modificado durante os testes de redução dos níveis de tensão de entrada da plataforma de desenvolvimento, julgamos necessária a remoção do regulador de tensão 3 (FAN1112), responsável pela alimentação do núcleo do FPGA *Spartan-3*. A Figura 7.7 apresenta o diagrama de blocos das linhas de alimentação da plataforma de desenvolvimento após a retirada do regulador de tensão, e sua substituição por uma fonte de tensão contínua externa e variável.

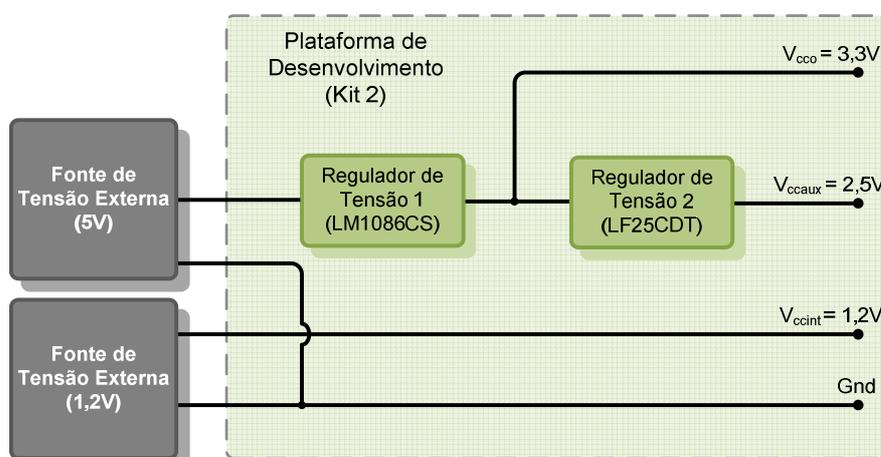


Figura 7.7 - Lógica de alimentação modificada da plataforma de desenvolvimento.

A remoção deste componente permitiu a redução gradual e progressiva da tensão de alimentação do núcleo do FPGA. Os valores de percentuais de redução, relativos à tensão nominal do núcleo do FPGA (1,2 volts), são apresentados na Tabela 7.6.

Tabela 7.6 - Redução percentual da tensão do núcleo.

Redução Percentual (%)	Tensão Aplicada (V)
5	1,14
10	1,08
15	1,02
20	0,96
25	0,9
30	0,84
35	0,78
40	0,72

## 7.10. Experimento com o B02 – Modificado (borda de subida)

Na busca por falhas funcionais do circuito B02 – Modificado que pudessem validar a técnica de CDCDC, reduzimos os níveis de tensão de entrada da plataforma de desenvolvimento (*Kit 2*) e do núcleo do FPGA para, respectivamente, amplitudes iguais a 3,2 e 0,72 volts.

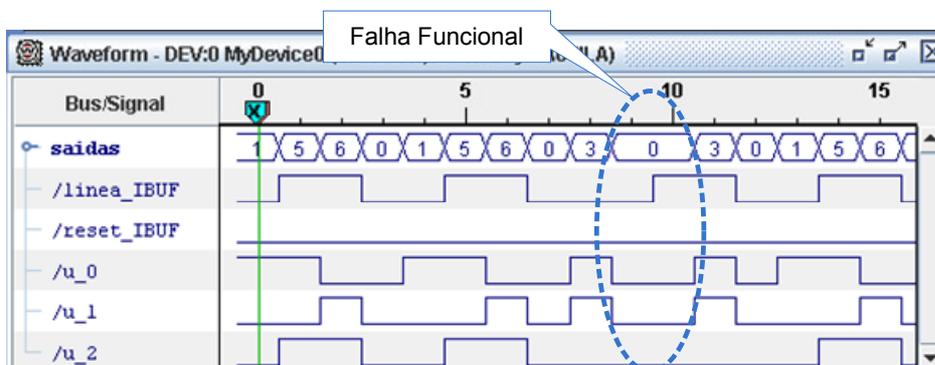
A Figura 7.8, obtida através da ferramenta de análise *ChipScope Pro 7.1i* (92), apresenta os resultados das saídas do circuito B02 – Modificado composto somente por *flip-flop's* sensíveis a borda de subida (transição do sinal de relógio do nível lógico baixo para o alto). É importante ressaltar que estas saídas, do circuito sob teste, foram geradas com ciclos de trabalho do sinal de relógio distintos, mas a partir do mesmo conjunto de entradas padrão (sinal *linea\_IBUF*).

A Figura 7.8 (a) apresenta as saídas do circuito B02 – Modificado, onde todos os *flip-flop's* do circuito estiveram sujeitos a um sinal de relógio com ciclo de trabalho igual a 50% e uma ocorrência de falha funcional na lógica da máquina de estados finitos foi registrada.

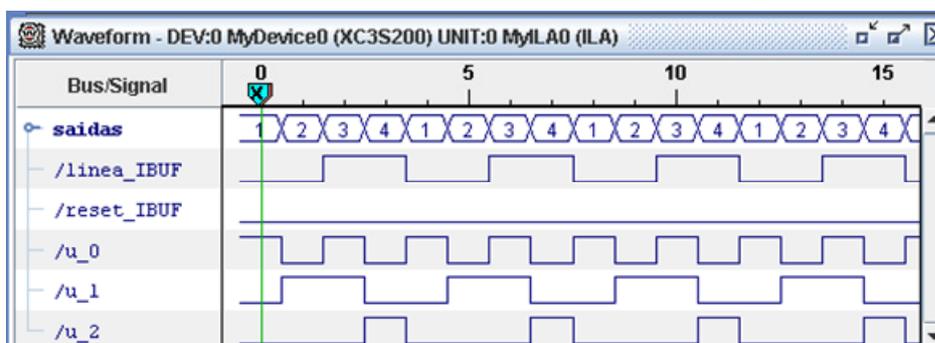
Por outro lado, a Figura 7.8 (b) apresenta as saídas deste mesmo circuito de teste, sem nenhuma ocorrência de falha funcional, mas com um ciclo de trabalho igual a 25% para os *flip-flop's* do caminho crítico e 50% para os demais *flip-flop's* do circuito de teste B02 – Modificado.



de estados finitos, onde todos os *flip-flop's* estavam sujeitos a um sinal de relógio com 50% de ciclo de trabalho. Já a Figura 7.9 (b) apresenta as saídas do mesmo circuito, sem registros de falhas funcionais, mas com um sinal de relógio de 75% de ciclo de trabalho para o *flip-flop* do caminho crítico e 50% para os demais *flip-flop's*.



Sinal de relógio com 50% de ciclo de trabalho;



Sinal de relógio com 75% de ciclo de trabalho;

Figura 7.9 - Saídas do circuito B02 – Modificado (borda de descida).

Analisando os resultados obtidos em ambos os testes preliminares realizados com o circuito B02 – Modificado, foi possível observar uma divergência nas seqüências dos estados da FSM. Contudo, após um estudo comportamental do circuito, concluiu-se que estas seqüências, embora divergentes entre si, são saídas válidas e atestam o funcionamento correto e coerente da máquina de estados finitos do circuito B02 – Modificado. A Figura 7.10 apresenta todas as seqüências de saída válidas deste circuito para o padrão de entrada usado neste experimento.

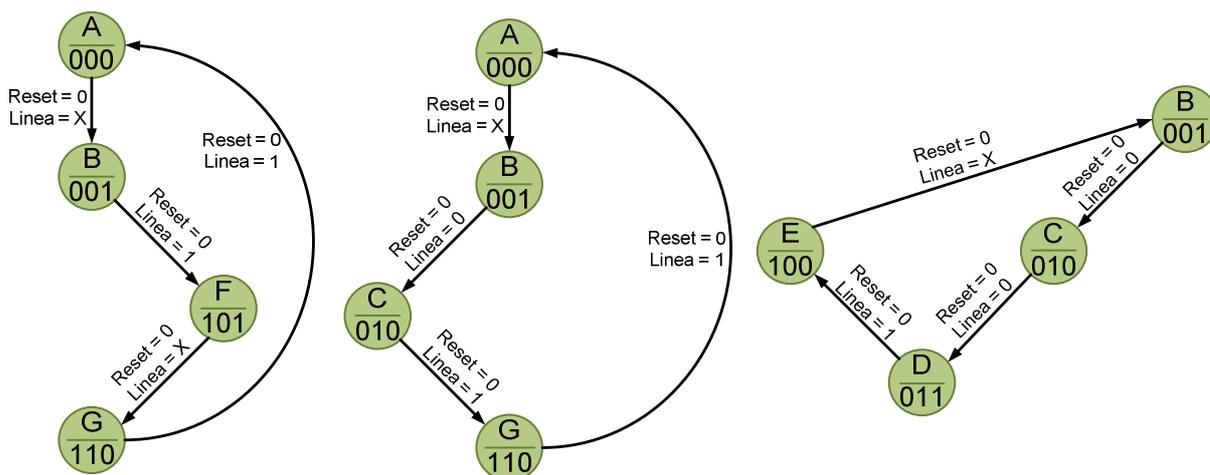


Figura 7.10 - Saídas coerentes do circuito B02 - Modificado.

## 7.12. Conclusões

Com base nos resultados experimentais preliminares apresentados anteriormente, podemos obter algumas conclusões, também preliminares, sobre o circuito de teste utilizado (B02 – Modificado), o comportamento e características de robustez às reduções nas linhas de alimentação da plataforma de desenvolvimento *Spartan-3 Starter Kit Board* e do seu FPGA, e principalmente sobre a validação da técnica de Controle Dinâmico do Ciclo de Trabalho (CDCDC) do sinal de relógio, foco principal desta dissertação. Estas conclusões preliminares são apresentadas a seguir:

- Nestes experimentos preliminares a plataforma de desenvolvimento e os seus componentes se mostraram bastante robustos, apresentando somente falhas funcionais quando expostos a reduções dos níveis de tensão da ordem de 35% e 40% para, respectivamente, a periferia e o núcleo do FPGA.
- As reduções nos níveis de tensão de alimentação dos FPGA's podem gerar importantes atrasos de propagação na rede de distribuição de sinais internos do circuito sob teste embarcado, levando-o a falhas funcionais, principalmente, quando este circuito opera com um ciclo de sinal de relógio igual a 50%.
- A técnica de Controle Dinâmico do Ciclo de Trabalho, implementada através do *Clock Manager*, permite a adaptação do sinal de relógio aos atrasos de propagação dos sinais internos do circuito sob teste, tornando-o mais robusto a oscilações dos níveis de tensão de alimentação, e por consequência, reduzindo a ocorrência de falhas funcionais.

- A plataforma de desenvolvimento possui *Spartan-3 Starter Kit Board* limitações estruturais que dificultaram e/ou impossibilitaram algumas abordagens mais complexas da validação da técnica CDCDC. Neste sentido, o projeto e desenvolvimento de uma nova plataforma com dois FPGA's, e com níveis de tensão de alimentação controláveis e independentes tornou-se imprescindível e fundamental.
- O circuito de teste B02 – Modificado foi fundamental para a validação preliminar da técnica CDCDC, entretanto sua funcionalidade é bastante restrita. Neste sentido concluiu-se que a escolha de um circuito mais complexo e funcional é absolutamente necessária para o desenvolvimento desta dissertação.

## 8. Estudo de Caso II

### 8.1. Introdução

Num segundo momento, com o objetivo de validar a técnica Controle Dinâmico do Ciclo de Trabalho (CDCDC) para o aumento da robustez de *Systems-on-Chip* (SoC's) sujeitos a flutuações de tensão no barramento de alimentação, e operando simultaneamente em um ambiente de interferência eletromagnética (EMI), elaboramos um segundo *estudo de caso*, cuja estrutura será detalhada a seguir.

### 8.2. Plataforma de Testes

Embora a plataforma de desenvolvimento *Spartan-3 Starter Kit Board* tenha sido de suma importância para a execução do primeiro estudo de caso desta dissertação. Este *hardware* possui limitações estruturais que impossibilitam algumas abordagens mais complexas relativas a esta dissertação, já que o seu projeto não segue nenhuma norma de teste 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, o projeto e desenvolvimento de uma nova plataforma de testes, com características estruturais mais complexas, robustas e funcionais, tornou-se absolutamente indispensável para a execução do trabalho descrito nesta dissertação.

Esta nova plataforma de testes, projetada e desenvolvida pela equipe do Laboratório SiSC<sup>4</sup>, foi baseada nas normas de teste de susceptibilidade de circuitos integrados a Interferências Eletromagnéticas (EMI) irradiadas e conduzidas IEC 62.132-1, 62.132-2 e 62.132-4, e possui, conforme apresenta a Figura 8.1, seis camadas (do inglês, *layers*) cujas características estruturais são apresentadas a seguir:

---

<sup>4</sup> A equipe do Laboratório SiSC é composta por professores e alunos da Faculdade de Engenharia da PUCRS. No projeto desta plataforma de testes 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 teste da placa, além de um plano de terra (*Gnd*) cobrindo toda área do *layer*;
- Camada 2 (*Inner 1*) – Possui somente os planos de alimentação ( $V_{CC}$ ) dos circuitos integrados sob teste;
- 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 testes irradiados, além de trilhas de alimentação, sinais, e um plano de terra (*Gnd*) cobrindo toda a área do *layer*.

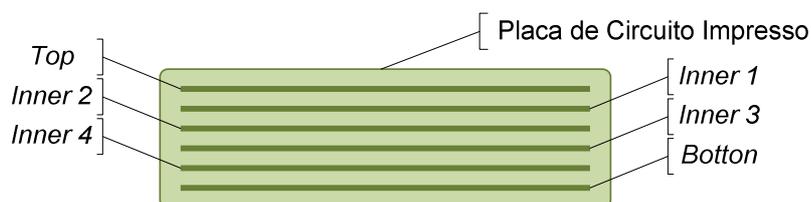


Figura 8.1 - Composição da placa de circuito impresso.

Além destes cuidados relativos ao atendimento das normas de teste de susceptibilidade a Interferências Eletromagnéticas (EMI) em circuitos integrados a plataforma de testes, cujo diagrama esquemático genérico é apresentado na Figura 8.2, dispõe em sua arquitetura dos seguintes componentes:

- 2 FPGA's Xilinx XC3S500E (500k portas, 256 pinos, 360 *Kbits* de *block* RAM interna, 20 multiplicadores e 4 DCM's);
- 1 FPGA Xilinx XC3S200 (200k portas, 144 pinos, 216 *Kbits* de *block* RAM interna, 12 multiplicadores e 2 DCM's);
- 4 memórias SRAM (do inglês, *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 LED's;
- 2 conectores JTAG independentes para programação e *debug* dos FPGA's;
- *Jumper's* para seleção e controle independente dos níveis de tensão alimentação;

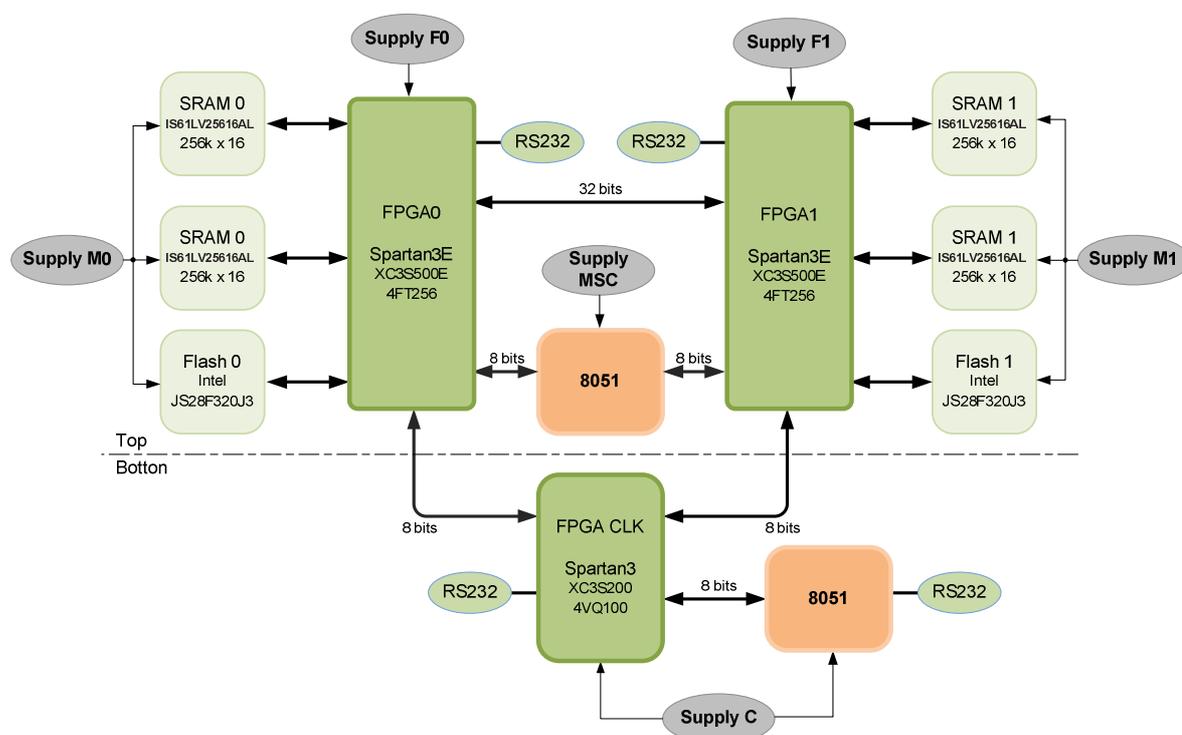


Figura 8.2 - Esquemático genérico da plataforma de testes.

A Figura 8.3 e a Figura 8.4 apresentam, respectivamente, as vistas inferior e superior da plataforma de testes cujos principais componentes são destacados.

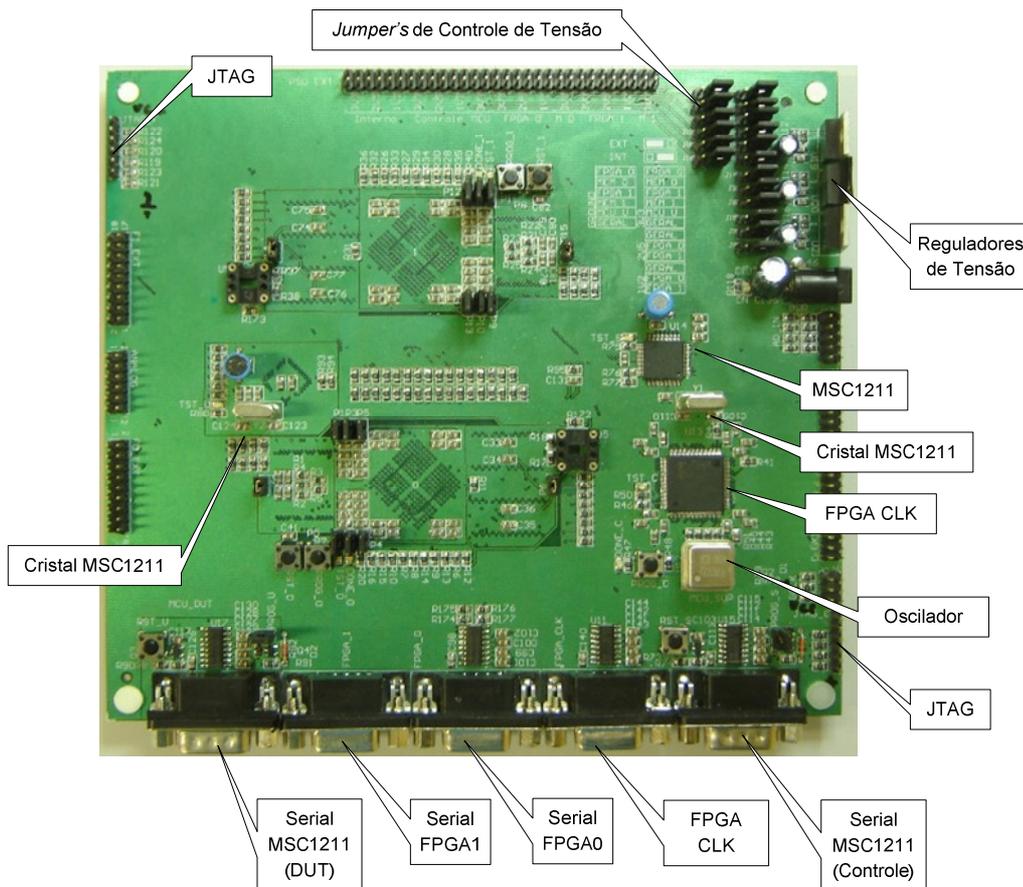


Figura 8.3 - Vista inferior da plataforma de testes.

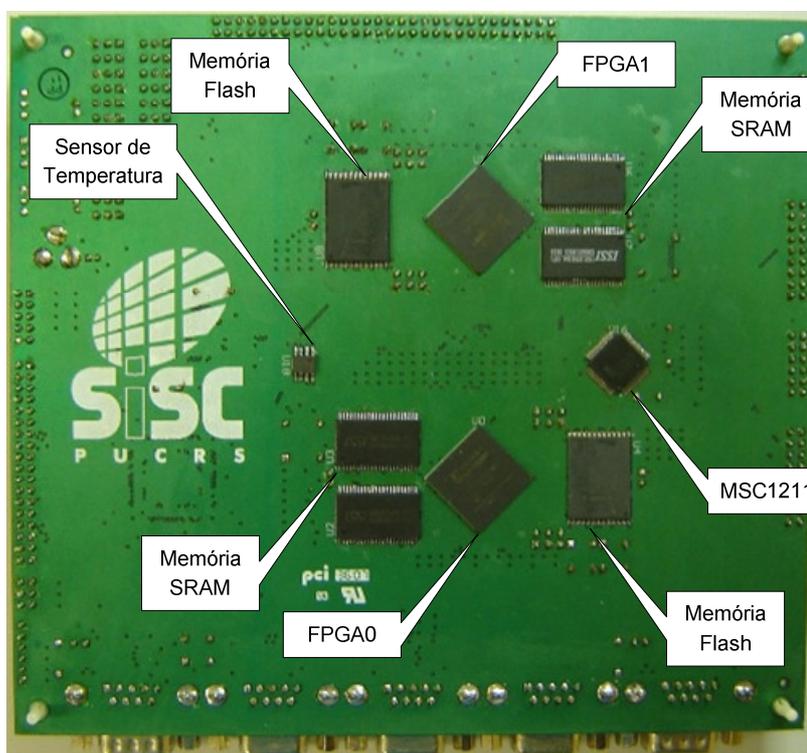


Figura 8.4 - Vista superior da plataforma de testes.

### 8.3. Placa de Alimentação e Injeção de Falhas

No primeiro estudo de caso apresentado nesta dissertação, à injeção de falhas via variação (redução) dos níveis de tensão de alimentação do FPGA sob teste era realizada manualmente através de fontes variáveis de tensão contínua.

Esta injeção de falhas manual, embora tenha sido bem sucedida experimentalmente, tornava a execução dos testes bastante lenta e sujeita a muitas imprecisões, decorrentes do manuseio dos potenciômetros de ajuste de tensão das fontes variáveis empregadas nos testes.

Tendo em vista estas características, e na busca pela garantia de qualidade e eficiência dos testes do segundo estudo de caso desta dissertação, se fez necessário o desenvolvimento de uma placa de injeção de falhas de alimentação para os circuitos integrados da plataforma de testes (FPGA's).

Depois de pesquisas e estudos relacionados às características e necessidades deste tipo de injeção de falhas (88) (89), concluiu-se que as reduções nas linhas de alimentação dos circuitos integrados sob teste deveriam ter um comportamento bastante preciso, com transições rápidas, e principalmente com repetibilidade.

A Figura 8.5 apresenta um diagrama esquemático genérico do circuito de alimentação e injeção de falhas, desenvolvido para os testes desta dissertação, onde é possível verificar que os níveis de tensão de alimentação e falha, do FPGA sob teste (DUT), são gerados pelo FPGA CLK associado a conversores digitais-analógicos (do inglês, *Digital to Analog Converter* - DAC) conectados em série a *buffer's* (amplificadores operacionais).

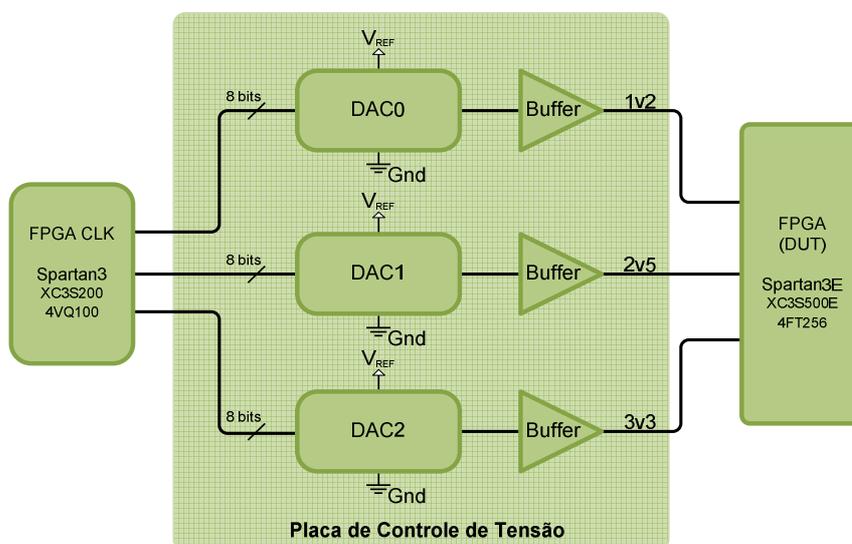


Figura 8.5 - Diagrama da placa de alimentação e injeção de falhas.

A Figura 8.6 apresenta, respectivamente, a vista superior e inferior da placa de alimentação e injeção de falhas desenvolvida para os testes do segundo estudo de caso desta dissertação.

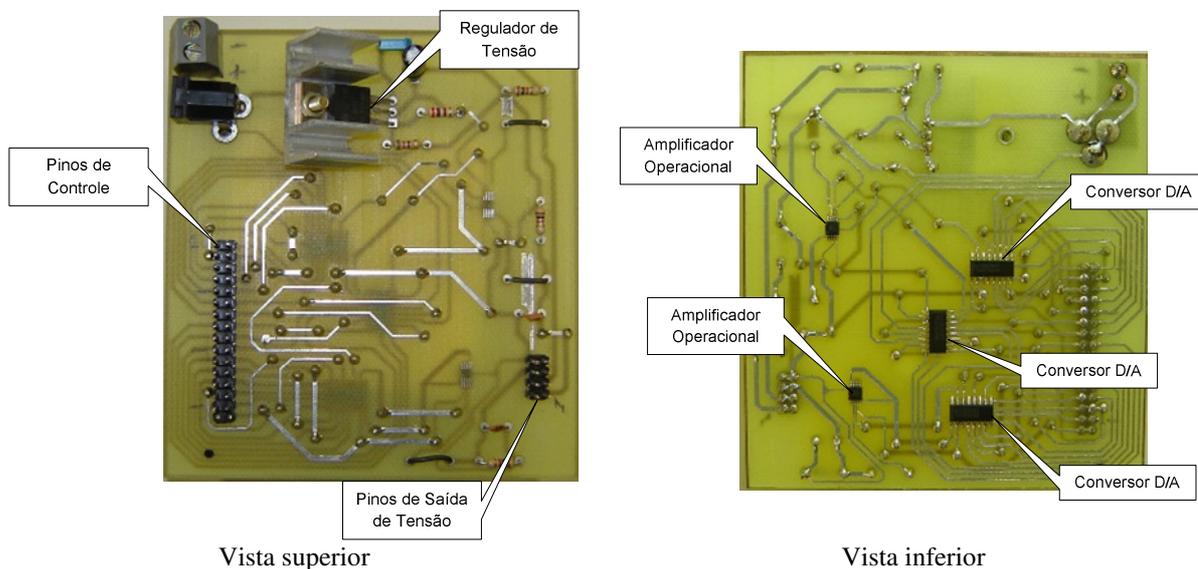


Figura 8.6 - Placa de alimentação e injeção de falhas.

## 8.4. Circuito de Alimentação

O circuito de Alimentação foi desenvolvido com o objetivo de gerar os sinais de controle e dados, apresentados no diagrama temporal da Figura 8.7, para cada um dos três conversores digitais analógicos da placa de alimentação e injeção de falhas através de uma máquina de estados finitos (FSM).

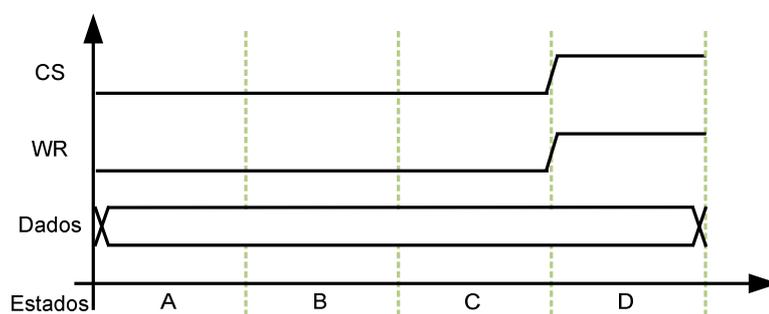


Figura 8.7 - Diagrama temporal de escrita nos conversores digitais-analógicos.

A Tabela 8.1 e Tabela 8.2 apresentam respectivamente a descrição funcional dos pinos e as combinações de entrada e saída do circuito de Alimentação.

Tabela 8.1 - Descrição funcional dos pinos do circuito de Alimentação.

Nome	Descrição
clock_in	Pino de entrada da referência de sinal de relógio do circuito (oscilador externo).
reset_in	Pino de <i>reset</i> da máquina de estados finitos do circuito.
selecao_ruido	Vetor de 2 <i>bits</i> para a seleção do tipo de saída dos DAC's.
cs_0	Pino de seleção do DAC0.
wr_0	Pino de habilitação de escrita no DAC0.
cs_1	Pino de seleção do DAC1.
wr_1	Pino de habilitação de escrita no DAC1.
cs_2	Pino de seleção do DAC2.
wr_2	Pino de habilitação de escrita no DAC2.
dado_0	Vetor de 8 <i>bits</i> de dados que define a amplitude do sinal de saída do DAC0.
dado_1	Vetor de 8 <i>bits</i> de dados que define a amplitude do sinal de saída do DAC1.
dado_2	Vetor de 8 <i>bits</i> de dados que define a amplitude do sinal de saída do DAC2.

Tabela 8.2 - Combinações de entrada e saída do circuito de Alimentação.

Entradas		Tipo de Saída
selecao_ruido(1)	selecao_ruido (0)	
0	0	Sinal contínuo fixo nos níveis de tensão nominal (1,2V – 2,5V – 3,3V).
0	1	Sinal quadrado oscilando entre as tensões nominais e as tensões reduzidas de teste.
1	0	Sinal contínuo fixo nos níveis de tensão reduzida de teste.

## 8.5. Processador Plasma

Conforme citado em seções anteriores, o circuito de teste utilizado no primeiro estudo de caso desta dissertação apresentava características funcionais e estruturais bastante simples e restritas. Tendo em vista isto, a busca por um novo circuito de teste mais complexo e robusto foi necessária para a execução do segundo estudo de caso desta dissertação.

O *softcore* Plasma é um processador RISC (do inglês, *Reduced Instruction Set Computer*) de 32 bits, totalmente compatível com o conjunto de instruções da arquitetura MIPS I™, exceto pelas instruções de *load/store* desalinhadas (93) (94).

Com três estágios de *pipeline* este processador, descrito em VHDL, possui diversos blocos que implementam funcionalidades como UART, *Timer*, além de controladores de interrupções, *Ethernet*, memória SRAM, memória DDR SDRAM e memória *Flash* (95).

A Figura 8.8 apresenta um diagrama de blocos da arquitetura básica deste processador embarcado.

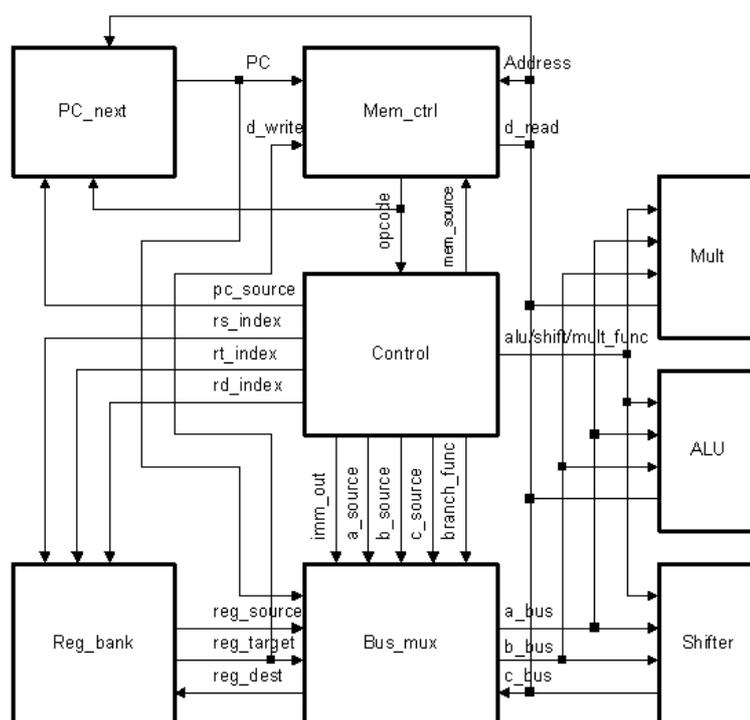


Figura 8.8 - Diagrama de blocos da arquitetura básica do processador Plasma (95).

## 8.6. Modificações Estruturais no Processador Plasma

Desenvolvido por Steve Rhoads, o processador Plasma é distribuído gratuitamente através do site *Opencores* (96) sob a filosofia de código “livre” e “aberto”, o que foi fundamental para a realização das modificações estruturais necessárias, em alguns blocos internos do processador, para a sua adaptação a técnica de Controle Dinâmico de Ciclo de Trabalho, o que seria impossível em *softcores* comerciais como o Microblaze (97) pela suas características.

### 8.6.1. Adição do Sinal de Relógio com Ciclo de Trabalho Variável

Assim como o circuito de teste utilizado no primeiro estudo de caso desta dissertação, o processador Plasma também possuía, entre suas características estruturais originais, uma única entrada de sinal de relógio. Tendo em vista isto, e que a técnica de CDCDC tem como objetivo variar o ciclo de trabalho do sinal de relógio de apenas algumas partes e/ou blocos do circuito para aumentar a robustez do circuito sob teste, mantendo as demais regiões do circuito com os seus ciclos de trabalhos de sinais de relógio inalterados. O processador

embarcado Plasma foi modificado, sendo acrescentada na sua estrutura, uma entrada adicional de sinal de relógio com ciclo de trabalho variável (proveniente do circuito *clock manager*).

### 8.6.2. Remoção da Divisão de Frequência

O processador Plasma possuía, na sua estrutura original, um divisor da frequência do sinal de relógio de entrada conforme apresenta a Figura 8.9. Embora esta divisão de frequência fosse fundamental para a lógica de escrita e leitura da memória SRAM externa, este condicionamento do sinal de relógio impossibilitava a implementação da técnica CDCDC, já que os sinais gerados pelo circuito *clock manager* com ciclo de trabalho diferente de 50%, perdiam esta característica após serem introduzidos no processador (divisor de frequência).

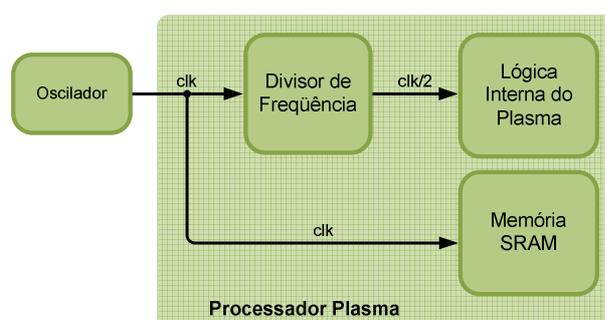


Figura 8.9 - Divisão de frequência do processador Plasma.

Com base nestas necessidades e características do processador e da lógica de escrita e leitura da memória SRAM, foi removido do interior do processador o processo de divisão de frequência do sinal de relógio, sendo este instanciado no interior do circuito *Clock Manager* para a garantia do sincronismo e principalmente da variação do ciclo de trabalho do sinal de relógio. A Figura 8.10 apresenta a estrutura divisão de frequência do sinal de relógio instanciada no interior do circuito *Clock Manager*, juntamente com a adição de uma nova entrada de sinal de relógio para a memória SRAM na estrutura do processador Plasma.

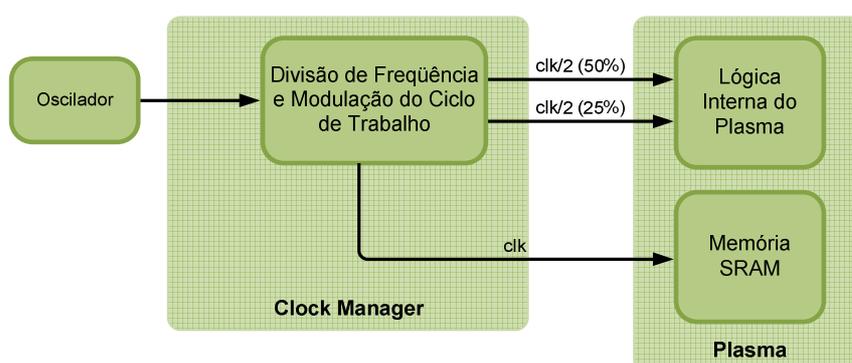


Figura 8.10 - Instanciamento da divisão de frequência no circuito *Clock Manager*.

A Figura 8.11 apresenta seis sinais de relógio gerados pelo circuito *Clock Manager*, os três primeiro com a técnica de Controle Dinâmico de Ciclo de Trabalho e os três últimos com o ciclo de trabalho fixo (Originais).

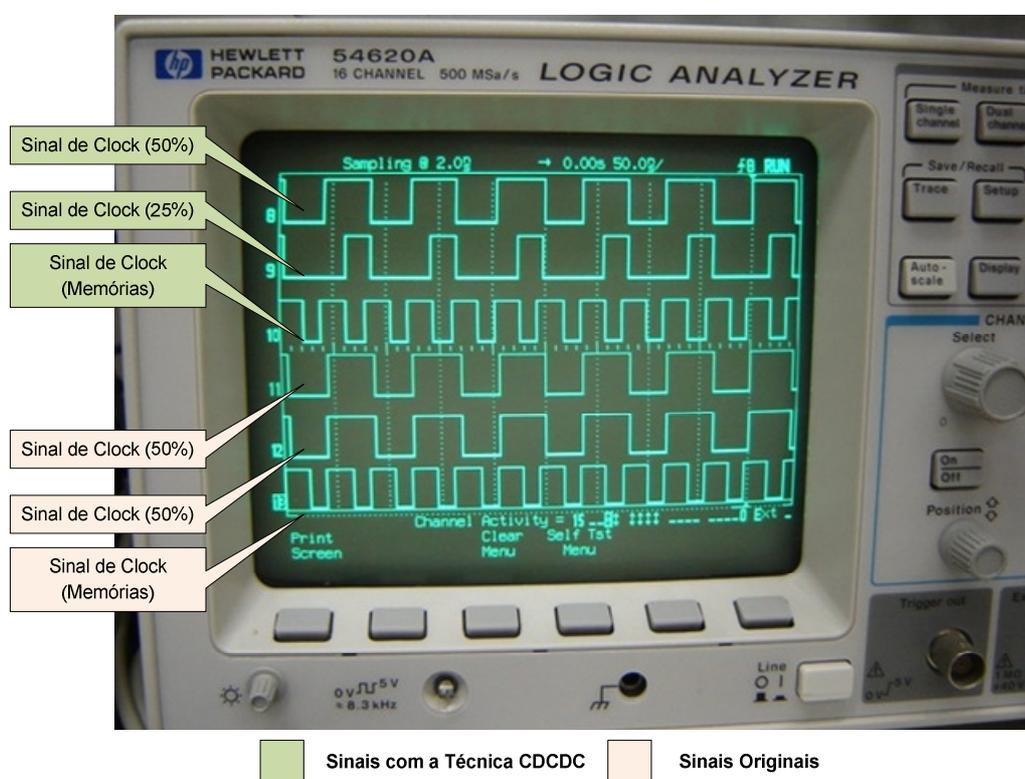


Figura 8.11 - Saída de relógio de saída do circuito *Clock Manager*.

### 8.6.3. Alteração do Ciclo de Trabalho do Bloco Multiplicador

O processador Plasma, usado neste segundo estudo de caso desta dissertação como circuito sob teste, foi desenvolvido de forma modular através de uma estrutura hierárquica de arquivos apresentada na Tabela 8.3 e na Figura 8.12 (95).

Com base nas características hierárquicas de descrição do processador, nas características de sincronismo e nos tipos de *flip-flop's* que compõem cada bloco, e principalmente nas características dos *softwares* de teste utilizados (que ainda serão apresentados nesta dissertação), optamos por implementar a técnica de Controle Dinâmico de Ciclo de Trabalho do sinal de relógio apenas no bloco multiplicador (*mult.vhd*) do processador, através da adição de entradas extras de sinal relógio nos blocos que interligam a hierarquia superior do processador (*plasma\_if.vhd*) ao bloco multiplicador (*mult.vhd*), conforme destacado na Figura 8.12.

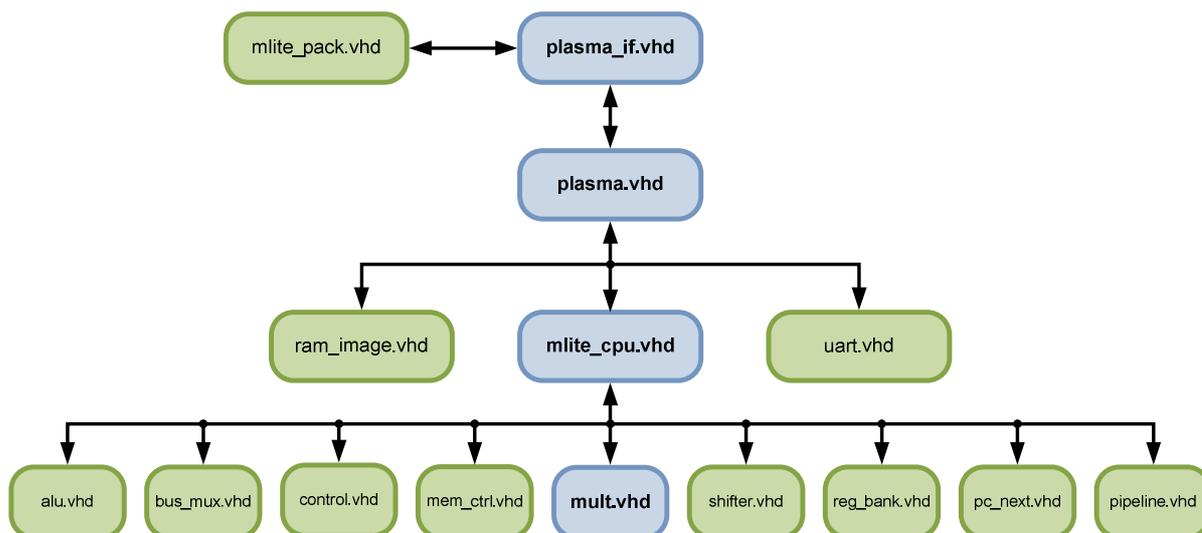


Figura 8.12 - Hierarquia de arquivos do processador Plasma.

Tabela 8.3 - Arquivos e funções do processador Plasma (95).

Arquivo	Função
mlite_pack.vhd	Constantes e funções do processador.
plasma_if.vhd	Hierarquia superior do processador.
plasma.vhd	Núcleo do processador com módulos de RAM e UART integrados.
ram_image.vhd	RAM interna para FPGA's Xilinx.
uart.vhd	Controlador de UART
mlite_cpu.vhd	Hierarquia superior do núcleo do processador.
alu.vhd	Unidade lógica e aritmética
bus_mux.vhd	Unidade de multiplexação do barramento
control.vhd	Decodificador de <i>Opcode</i>
mem_ctrl.vhd	Controlador de Memória
mult.vhd	Unidade de multiplicação e divisão
pc_next.vhd	Unidade <i>Program Counter</i>
reg_bank.vhd	Banco de 32 registradores de 32-bit.
shifter.vhd	Unidade <i>Shifter</i> .
pipeline.vhd	Unidade <i>Pipeline</i> .

## 8.7. Sistema de Gerenciamento de Testes

Tendo em vista que tanto o circuito *Clock Manager*, quanto o circuito de Alimentação necessitavam ser implementados no mesmo dispositivo (FPGA CLK) para a execução dos testes do segundo estudo de caso desta dissertação, foi desenvolvido um circuito de nível hierárquico superior para o instanciamento destes dois importantes módulos.

A Figura 8.13 apresenta a estrutura hierárquica do circuito *Sistema de Gerenciamento de Testes*, onde ambos os circuitos foram instanciados sem qualquer alteração funcional e/ou estrutural.

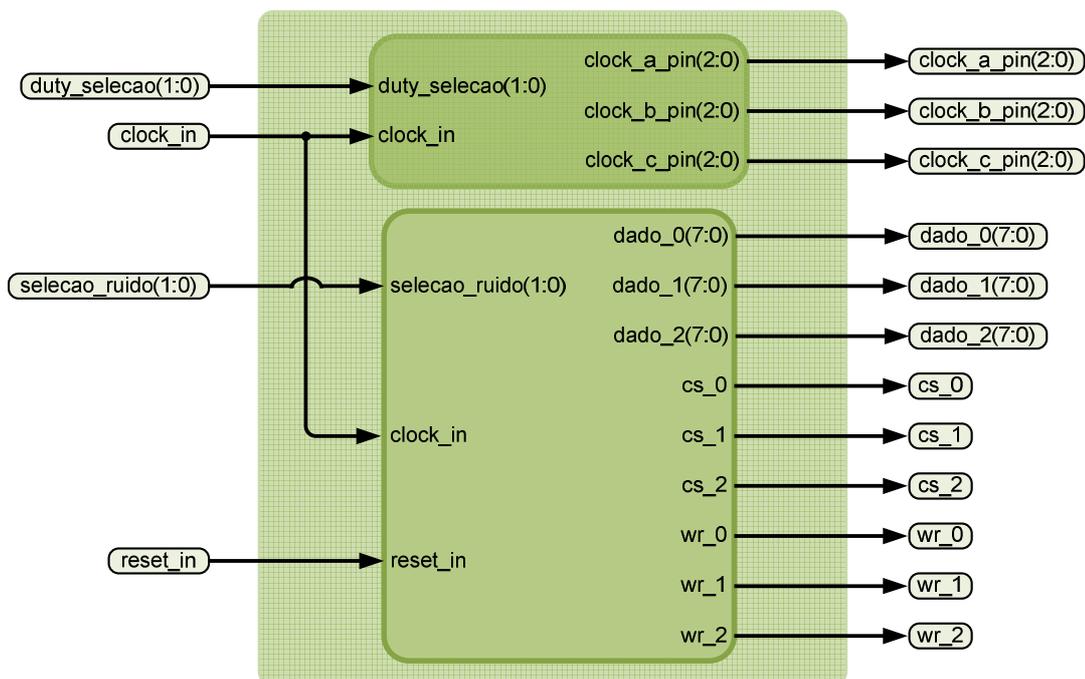


Figura 8.13 - Estrutura hierárquica do circuito Sistema.

## 8.8. Softwares de Teste

O Sistema Operacional em Tempo Real Plasma (do inglês, *Plasma Real-Time Operating System* – RTOS), como o próprio nome ressalta, foi projetado e desenvolvido, por Steve Rhoads, para ser executado especificamente sobre processador embarcado Plasma.

Com suporte a interrupções, *threads*, semáforos, filas de mensagens, *timers* e chaveamento preemptivo de contexto, este sistema operacional possui aproximadamente 1500 linhas de código C e 240 linhas de código *assembly*, que compilados geram aproximadamente 8 *Kbytes* de código executável.

Adicionalmente o Plasma RTOS também dispõe de bibliotecas que implementam funções C-ANSI, funções de ponto flutuante, além de *drivers* para a comunicação através da porta serial do processador, cujo acréscimo no tamanho de código executável é aproximadamente igual a 7 *Kbytes* (98).

Motivado por estas características de compatibilidade com o processador Plasma, e também pelo seu suporte ao compilador GCC, o Plasma RTOS foi utilizado como sistema operacional para a execução multitarefa das três aplicações de teste apresentadas abaixo:

- **Ajuste de Curvas** (do inglês, *Curve Fitting* – CF): realiza o ajuste de dez pontos decimais inteiros a cinco tipos diferentes de curvas, através do método dos mínimos quadrados;

- **Multiplicação de Matrizes (MM)**: realiza a multiplicação de duas matrizes de quarta ordem;
- **Whirlpool (W)**: realiza a encriptação de mensagens com até  $2^{256}$  bits de comprimento, produzindo uma mensagem codificada de 512 bits (99).

Além de serem consagradas aplicações no meio acadêmico (32) (14) (100) (101), os *softwares* de teste empregados neste segundo estudo de caso desta dissertação também necessitam de um elevado índice de multiplicações para a sua correta execução, o que foi decisivo para a sua escolha, já que a técnica de Controle Dinâmico de Ciclo de Trabalho do sinal de relógio foi implementada no bloco multiplicador do processador Plasma.

## 8.9. Ajuste dos Tempos de Execução

O comportamento das falhas funcionais provenientes das reduções dos níveis de tensão de alimentação se mostrou, nos testes preliminares desta dissertação, aleatória e de difícil repetibilidade, sendo necessária para o registro de ocorrências de falhas, a execução de testes com longos tempos de duração (quando comparados ao período de sinal de relógio).

Aliada a estas características de falhas, as aplicações utilizadas como *software* de teste neste segundo estudo de caso, mesmo quando gerenciadas por um sistema operacional, possuem diferentes tempos de execução e conseqüentemente diferentes tempos de finalização, provocando assim uma subutilização do processador.

Diante deste cenário e buscando um índice de ocorrência de falhas favorável para a validação da técnica de Controle Dinâmico de Ciclo de Trabalho do sinal de relógio, implementada no bloco multiplicador do processador embarcado Plasma, ajustamos o sistema operacional Plasma RTOS de forma que as três aplicações, embora possuíssem tempos de execução diferentes, concluíssem suas tarefas em tempos aproximadamente iguais. A Figura 8.14 apresenta a lógica de ajuste dos tempos de finalização dos softwares de teste (baseada em loop) e a Tabela 8.4 apresenta a nomenclatura adotada para esta lógica.

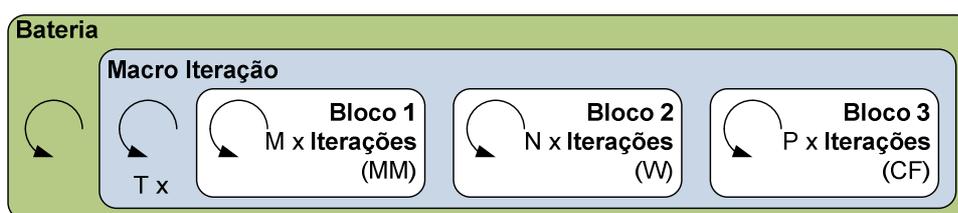


Figura 8.14 - Lógica de ajuste dos tempos dos *softwares* de teste.

Tabela 8.4 - Nomenclatura adotadas para lógica de ajuste dos tempos.

Nome	Descrição
Iteração	Execução completa de uma aplicação para um determinado vetor de entrada diferente.
Bloco	Execução completa de um agrupamento de iterações, onde M=2368, N=1448 e P=320.
Macro Iteração	Execução completa de um agrupamento de todos os blocos, onde T=2.
Bateria	Execução completa de todos os macros iterações.

## 8.10. Microcontrolador Gerenciador de Teste

Tendo em vista que o esquemático da plataforma de testes, apresentado na seção 8.2, dispõe de interconexões entre os pinos de entrada e saída do FPGA e do microcontrolador; que as funcionalidades dos circuitos *Clock Manager* e de *Alimentação*, instanciados no circuito *Sistema de Controle*, são selecionadas e acionadas através da combinação lógica dos seus pinos de entrada; e que o microcontrolador também possui uma porta de comunicação serial padrão RS232, desenvolvemos um *software*, em linguagem C, para a seleção e habilitação das funcionalidades destes circuitos. A Tabela 8.5 apresenta um resumo dos comandos do *software* de gerenciamento de teste e suas descrições.

Tabela 8.5 - Descrição dos comandos do *software* de gerenciamento de teste.

Comando	Descrição
N	Habilita o ciclo de trabalho do sinal de relógio igual a 50%.
R	Habilita o ciclo de trabalho do sinal de relógio igual a 25%.
F	Habilita o ciclo de trabalho do sinal de relógio igual a 75%.
V	Habilita os níveis de tensão de alimentação nominal.
Q	Habilita os níveis de tensão de alimentação pulsante.
D	Habilita os níveis de tensão de alimentação reduzidos.
S	<i>Reset</i> da placa de alimentação e injeção de falhas.
0	Coloca o FPGA0 em modo de programação.
1	Coloca o FPGA1 em modo de programação.

## **Parte III - Resultados e Conclusões**

# 9. Resultados

## 9.1. Introdução

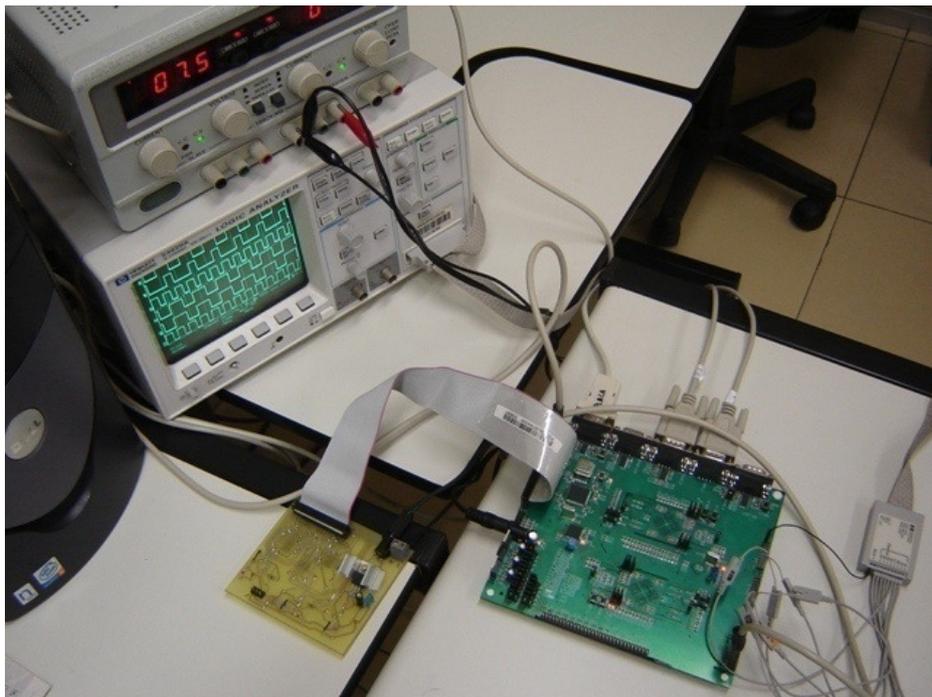
Este capítulo tem como objetivo demonstrar a aplicabilidade da técnica de Controle Dinâmico de Ciclo de Trabalho em *Systems-on-Chip* através de testes de redução de tensão de alimentação e testes de Interferência Eletromagnética Irradiada, cujos procedimentos e configurações serão apresentados a seguir.

## 9.2. Testes de Redução dos Níveis de Tensão de Alimentação

Realizado nas dependências do Laboratório de Sistemas, Sinais & Computação (SiSC) da Faculdade de Engenharia da PUCRS, conforme apresenta a Figura 9.1, os testes de redução dos níveis de tensão de alimentação seguiu as configurações apresentadas no diagrama da Figura 9.2.



(a)



(b)

Figura 9.1 - Teste de redução dos níveis de tensão de alimentação.

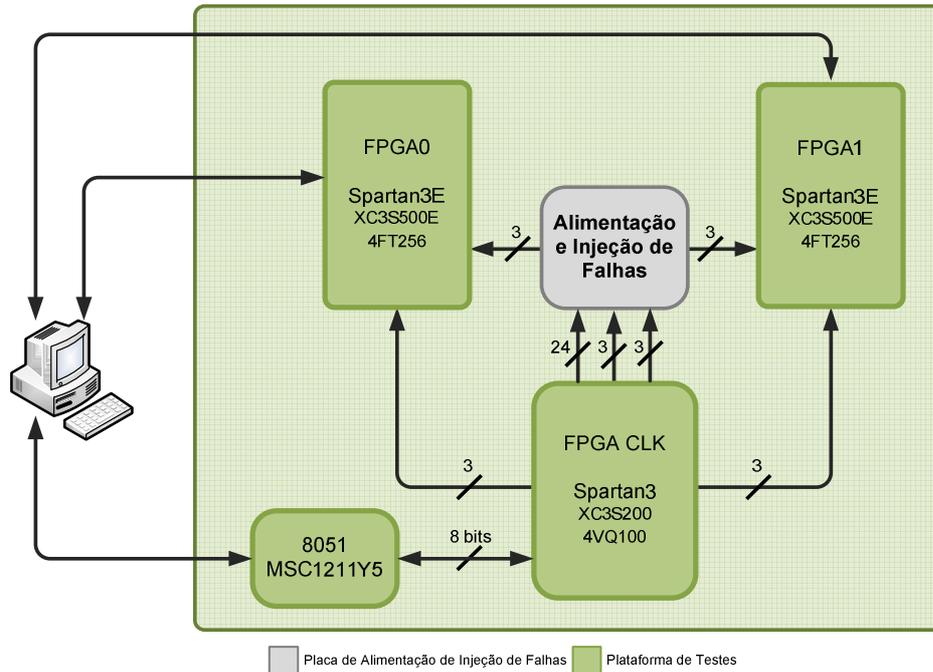


Figura 9.2 - Configurações dos testes de redução dos níveis de tensão de alimentação.

A Figura 9.3 apresenta a seqüência de procedimentos adotados nos testes de redução dos níveis de tensão de alimentação onde:

- **Início:** A placa de alimentação e injeção de falhas e a plataforma de testes são conectadas ao microcomputador supervisor através de comunicação serial e energizadas.
- **Configuração do FPGA CLK:** O *bitstream* do circuito *Sistema de Gerenciamento de Teste* é carregado no FPGA CLK via cabo JTAG através do *software* Impact desenvolvido pela Xilinx;
- **Configuração FPGA0 e FPGA1:** O processador Plasma é carregado nos FPGA's via cabo JTAG através do *software* Impact, sendo o FPGA0 carregado com o processador com a técnica de Controle Dinâmico de Ciclo de Trabalho e o FPGA1 com o processador original;
- **Download das Aplicações de Teste:** O sistema operacional Plasma RTOS é carregado via porta serial (RS232) nas memórias SRAM associadas a cada FPGA (FPGA0 e FPGA1);
- **Execução das Aplicações:** As aplicações de teste (Ajuste de Curvas, Multiplicação de Matrizes e *Whirlpool*) são inicializadas pelo sistema operacional Plasma RTOS sendo os resultados de suas operações enviadas para o microcomputador supervisor via porta serial;

- **Redução dos Níveis de Tensão:** As tensões de alimentação dos FPGA's sob teste (FPGA0 e FPGA1) são reduzidas, através da placa de alimentação e injeção de falhas, dos níveis nominais (1,2V, 2,5V e 3,3V), para respectivamente 905mV, 1,847V e 2,478V;
- **Acionamento do CDCDC:** Acionada a técnica de Controle Dinâmico de Ciclo de Trabalho (CDCDC), isto é, redução do ciclo de trabalho do sinal de relógio para igual a 25% no FPGA0.
- **Fim da Execução das Aplicações:** O sistema operacional Plasma RTOS finaliza a execução das aplicações de teste Ajuste de Curvas, Multiplicação de Matrizes e Whirlpool;
- **Suspensão do CDCDC:** Suspensão da técnica de Controle Dinâmico de Ciclo de Trabalho (CDCDC), isto é, aumento do ciclo de trabalho do sinal de relógio para igual a 50% no FPGA0.
- **Aumento dos Níveis de Tensão:** Os níveis de tensão de alimentação dos FPGA's sob teste (FPGA0 e FPGA1) são restabelecidos para os valores nominais (1,2V, 2,5V e 3,3V) através da placa de alimentação e injeção de falhas;
- **Readback:** É realizado *readback* através do software Impact, isto é, o processo de comparação entre o *bitstream* carregado e o *bitstream* padrão para verificar a ocorrência de falhas no hardware (processador Plasma) embarcado nos FPGA's sob teste.
- **Salva Arquivo:** Se o *readback* não apresentou diferenças entre o *bitstream* padrão e o *bitstream* carregado os arquivos de saída (*log's*) são salvos e os resultados registrados na planilha de testes.



Figura 9.3 - Seqüência de procedimentos do teste de redução de tensão de alimentação.

As três aplicações executadas pelo sistema operacional Plasma RTOS imprimem, para cada iteração, os seus estados de saída via porta serial (RS232). Estas impressões foram armazenadas em arquivos de *log* e comparadas através do aplicativo *Limpador de Log*<sup>5</sup> a um arquivo padrão gerado pela execução das aplicações de teste no processador Plasma 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 estas registradas na planilha de teste. A Figura 9.4 apresenta o gráfico comparativo entre as ocorrências de falhas registradas na versão original do processador (Original) e na versão modificada com a técnica de Controle Dinâmico de Ciclo de Trabalho (Técnica CDCDC).

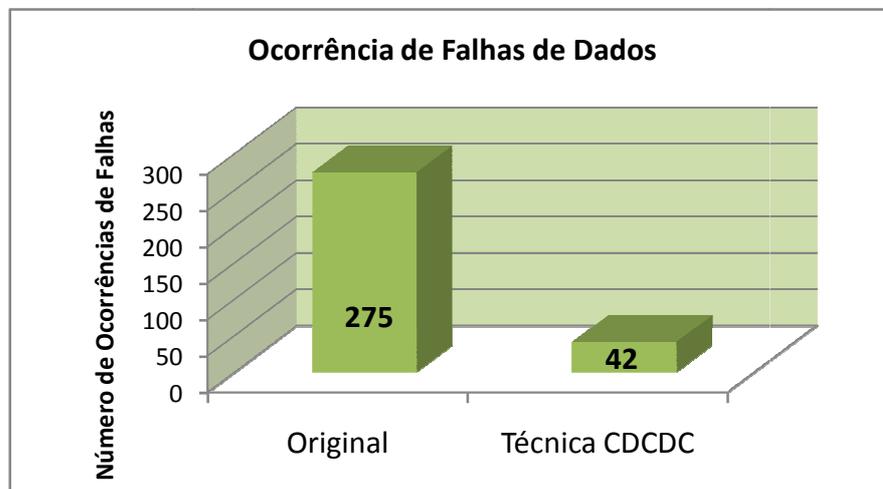


Figura 9.4 - Gráfico comparativo do número de ocorrência de falhas.

### 9.3. Testes Interferência Eletromagnética Irradiada

Realizados em janeiro de 2008 nos laboratórios do Instituto Nacional de Tecnologia Industrial (INTI) em Buenos Aires – Argentina, os testes de Interferência Eletromagnética (EMI) Irradiada desta dissertação seguiram a norma IEC 62.132-2. Sendo assim, foi necessário o emprego de equipamentos específicos como um gerador de sinais de alta frequência (1Hz-3GHz), um amplificador de sinais (1Hz-3GHz), um medidor de intensidade de campo, e principalmente uma célula TEM com resposta até 18GHz. A Figura 9.5 apresenta em destaque estes equipamentos.

<sup>5</sup> *Limpador de Log*: Este aplicativo foi desenvolvido em linguagem C, por Luciano Beiestorf Rocha e Cláudia Antunes Rocha para a comparação e análise de arquivos de *log* em suas dissertações de mestrado (15) (117).

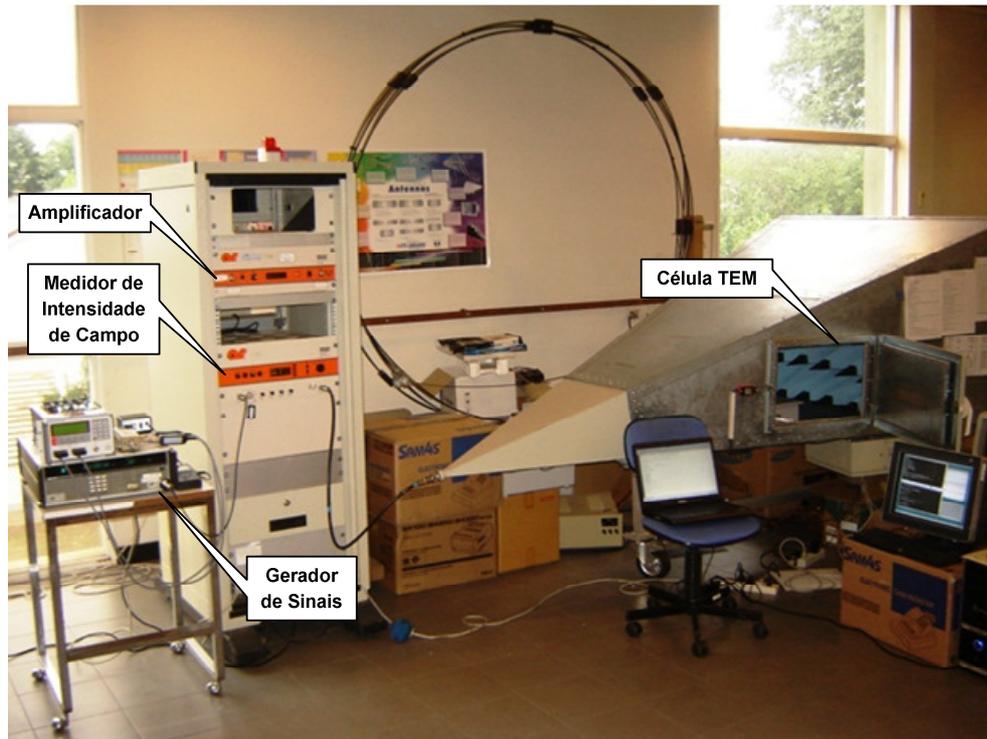


Figura 9.5 - Teste de Interferências Eletromagnéticas Irradiadas.

Além de necessitar destes importantes equipamentos, também utilizamos para os testes de EMI irradiada uma caixa metálica (dimensões iguais a 30x32x12cm) com uma abertura para a fixação da plataforma de testes (16x17cm) e outra para a passagem dos cabos de alimentação, gravação e comunicação (3x4cm).

Esta caixa metálica, com a carcaça conectada a referência de tensão (Gnd) da plataforma de testes, da placa de alimentação e injeção de falhas e da célula TEM, foi utilizada com a finalidade isolar e proteger os componentes e trilhas de sinais fixados do lado da placa que não deve estar sujeito a Interferências Eletromagnéticas (*bottom*), isto é, se comportar como uma *Gaiola de Faraday*<sup>6</sup>.

Após a fixação da plataforma de testes e da placa de alimentação e injeção de falhas no interior na caixa metálica, este conjunto foi inserido no interior da célula TEM conforme apresenta a Figura 9.6.

<sup>6</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 9.6 - Plataforma de testes e caixa metálica inseridas na célula TEM.

A Figura 9.7 apresenta a configuração dos testes de interferência eletromagnética realizados para esta dissertação.

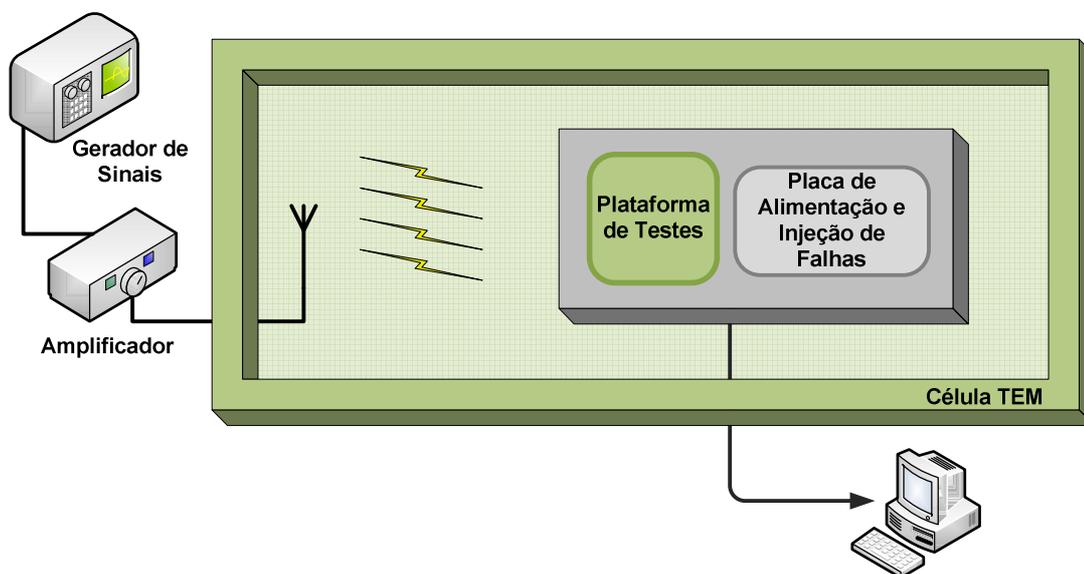


Figura 9.7 - Configuração dos testes de EMI Irradiada (14).

A Figura 9.8 apresenta a seqüência de procedimentos adotados nos testes de Interferência Eletromagnética (EMI) Irradiada onde:

- **Início:** A Placa de Alimentação e Injeção de Falhas e a Plataforma de Testes são conectadas ao microcomputador supervisor através de comunicação serial e energizadas.

- **Configuração do FPGA CLK:** O *bitstream* do circuito *Sistema de Gerenciamento de Teste* é carregado no FPGA CLK via cabo JTAG através do *software* Impact desenvolvido pela Xilinx;
- **Configuração FPGA0 e FPGA1:** O processador Plasma é carregado nos FPGA's via cabo JTAG através do *software* Impact, sendo um carregado com o processador com a técnica de Controle Dinâmico de Ciclo de Trabalho e o outro com o processador original;
- **Download das Aplicações de Teste:** O sistema operacional Plasma RTOS é carregado via porta serial (RS232) nas memórias SRAM associadas a cada FPGA (FPGA0 e FPGA1);
- **Execução das Aplicações:** As aplicações de teste (Ajuste de Curvas, Multiplicação de Matrizes e *Whirlpool*) são inicializadas pelo sistema operacional Plasma RTOS sendo os resultados de suas operações enviadas para o microcomputador supervisor via porta serial;
- **Redução dos Níveis de Tensão:** As tensões de alimentação dos FPGA's sob teste (FPGA0 e FPGA1) são reduzidas, através da placa de alimentação e injeção de falhas, dos níveis nominais (1,2V, 2,5V e 3,3V), para respectivamente 1,06V, 2,12V e 2,57V;
- **Acionamento da Interferência Eletromagnética:** Acionados o gerador ajustado em uma frequência igual a 850MHz e o amplificador de sinais gerando uma intensidade de campo eletromagnético de 187V/m no interior da célula GTEM.
- **Acionamento do CDCDC:** Acionada a técnica de Controle Dinâmico de Ciclo de Trabalho (CDCDC), isto é, redução do ciclo de trabalho do sinal de relógio para igual a 25%.
- **Fim da Execução das Aplicações:** O sistema operacional Plasma RTOS finaliza a execução das aplicações de teste Ajuste de Curvas, Multiplicação de Matrizes e Whirlpool;
- **Suspensão do CDCDC:** Suspensão da técnica de Controle Dinâmico de Ciclo de Trabalho (CDCDC), isto é, aumento do ciclo de trabalho do sinal de relógio para igual a 50%.
- **Suspensão da Interferência Eletromagnética:** Desligados o gerador e o amplificador de sinais.

- **Aumento dos Níveis de Tensão:** Os níveis de tensão de alimentação dos FPGA's sob teste (FPGA0 e FPGA1) são restabelecidos para os valores nominais (1,2V, 2,5V e 3,3V) através da placa de alimentação e injeção de falhas;
- **Readback:** É realizado *readback* através do software Impact, isto é, o processo de comparação entre o *bitstream* carregado e o *bitstream* padrão para verificar a ocorrência de falhas no hardware (processador Plasma) embarcado nos FPGA's sob teste.
- **Salva Arquivo:** Se o *readback* não apresentou diferenças entre o *bitstream* padrão e o *bitstream* carregado os arquivos de saída (*log's*) são salvos e os resultados registrados na planilha de testes.

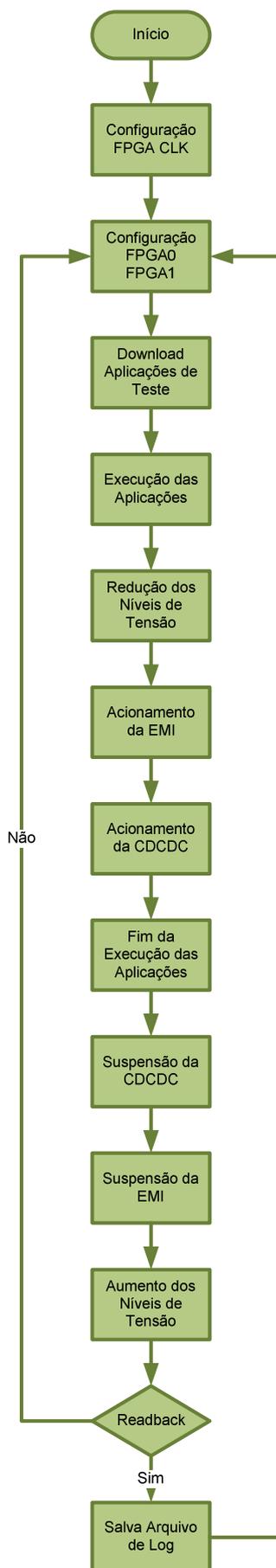


Figura 9.8 - Seqüência de procedimentos dos testes de EMI Irradiada

Assim como nos testes de redução de tensão, os testes de interferência eletromagnética irradiada as aplicações executadas pelo sistema operacional Plasma RTOS também imprimem, para cada iteração, os seus estados de saída via porta serial (RS232). Estas impressões também foram armazenadas em arquivos de *log* e comparadas através do aplicativo *Limpador de Log* com um arquivo padrão e suas ocorrências de falhas de dados registradas nas planilhas de teste. A Figura 9.9 apresenta o gráfico comparativo entre as ocorrências de falhas registradas na versão original do processador (Original) e na versão modificada com a técnica de Controle Dinâmico de Ciclo de Trabalho (Técnica CDCDC).

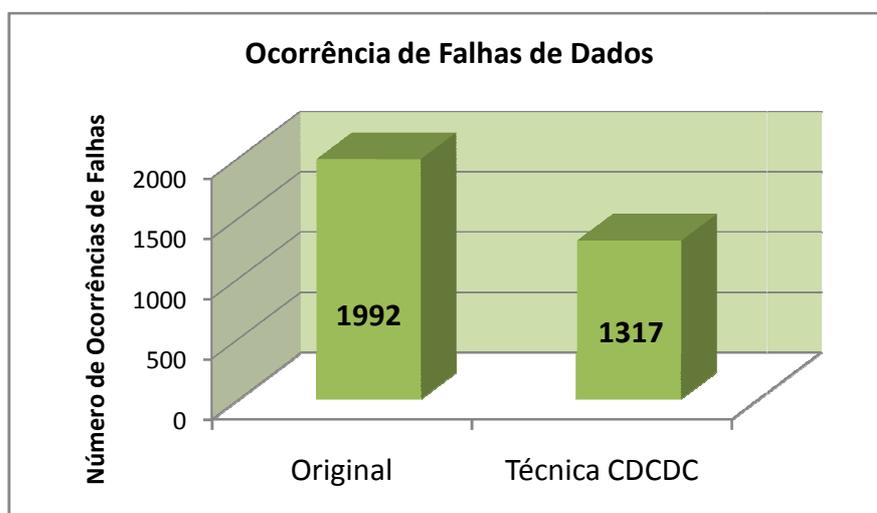


Figura 9.9 - Gráfico comparativo do número de ocorrência de falhas.

# 10. Conclusões Finais

Este trabalho teve como principal objetivo a validação da técnica de Controle Dinâmico de Ciclo de Trabalho para aumento da robustez de *Systems-on-Chip* sujeitos a flutuações nas linhas de alimentação, e ao ruído eletromagnético. Para cumprir estes objetivos, foi necessário realizar as etapas abaixo:

1. Pesquisar, implementar e avaliar o comportamento de circuitos digitais síncronos, quando estes são sujeitos a flutuações de tensão nas suas linhas de alimentação;
2. Desenvolver um circuito, descrito em VHDL e mapeado em FPGA, para a implementação da técnica de Controle Dinâmico de Ciclo de Trabalho, proposta por João Paulo Cacho Teixeira, *et al* (1).
3. Implementar em uma plataforma de desenvolvimento comercial um circuito de teste consagrado (*Benchmark - B02*) para a validação da técnica de Controle Dinâmico de Ciclo de Trabalho;
4. Projetar e desenvolver uma nova plataforma de testes seguindo as normas de teste de susceptibilidade a Interferências Eletromagnéticas (EMI) IEC 62.132, com controle independente dos níveis de tensão de alimentação dos dispositivos sob teste (FPGA's, memórias e microcontroladores);
5. Projetar e desenvolver um sistema de alimentação e injeção de falhas de alimentação para a nova plataforma de testes;
6. Pesquisar, implementar e avaliar alterações estruturais nos circuitos de testes (Processador Plasma) para permitir a validação da técnica de Controle Dinâmico de Ciclo de Trabalho em circuitos digitais síncronos de alto desempenho;
7. Implementar aplicativos de acordo com as necessidades impostas pela técnica de Controle Dinâmico de Ciclo de Trabalho;
8. Realizar os testes de redução dos níveis de tensão das linhas de alimentação dos circuitos integrados sob teste (FPGA's) no Laboratório do Grupo

SiSC/PUCRS, bem como, testes de Interferências Eletromagnéticas (EMI) Irradiada no laboratório do INTI, em Buenos Aires - Argentina.

Assim, as principais conclusões obtidas neste trabalho são relacionadas a seguir:

- As reduções nos níveis de tensão de alimentação de circuitos integrados, em especial no *core*, são absolutamente relevantes para a perda da integridade de sinal em sistemas digitais síncronos, podendo levá-los a perda de desempenho ou ainda a ocorrência de falhas funcionais dependendo da intensidade destas oscilações;
- As Interferências Eletromagnéticas (EMI) irradiadas com intensidades de campo iguais ou inferiores as utilizadas nos testes desta dissertação de mestrado, embora sejam um grave problema para a integridade de sinal de circuitos de alto desempenho e elevada densidade de integração, não são suficientes por si só para levarem o FPGA, e o sistema nele embarcado, a ocorrências de falhas funcionais;
- Conforme foi demonstrado nos testes desta dissertação, a combinação de fatores como Interferências Eletromagnéticas (EMI) Irradiadas e oscilações de tensão no barramento de alimentação de circuitos integrados podem degradar a integridade dos sinais significativamente, a ponto de ocorrências de falhas serem registradas. Neste sentido, concluímos que oscilações nos níveis de tensão de alimentação são mais importantes do que a interferência eletromagnética (EMI) irradiada, tendo em vista a perda de integridade de sinal em SoC's compostos por FPGA's Xilinx da família Spartan3E;
- Os dois circuitos mapeados nos FPGA's da plataforma apresentaram nos testes de Interferências Eletromagnéticas (EMI) irradiada níveis de susceptibilidades diferentes. Atribui-se a esta diferença o fato de que as trilhas e interconexões destes FPGA's, embora possuam um esquemático sob o ponto de vista funcional (lógico) idêntico, possuem caminhos internos distintos;
- Como conclusão final, a técnica de Controle Dinâmico de Ciclo de Trabalho (CDCDC) mostrou-se eficiente para o aumento da robustez de *Systems-on-Chip* a flutuações de tensão nas linhas de alimentação. Seu uso no processador Plasma reduziu a ocorrência de falhas em 75% nos testes de flutuação da tensão de alimentação e 44% nos testes interferência eletromagnética combinada com redução de tensão de alimentação.

## 10.1. Trabalhos Futuros

Como sugestão de trabalhos futuros, apresentamos atividades que inicialmente constavam do planejamento inicial, mas que devido ao tempo consumido nas etapas de implementação computacional e execução dos testes, não foram incluídas neste trabalho.

Embora o estudo de implementação e validação da técnica de Controle Dinâmico de Ciclo de Trabalho para o aumento da robustez de *Systems-on-Chip* tenha atingido muitos dos objetivos propostos, alguns pontos ainda necessitam de uma maior exploração, pois devido a limitações de tempo não foram incluídos neste trabalho. Neste sentido destacamos:

- Análise detalhada das características de uso do bloco multiplicador do processador Plasma por parte das aplicações de teste empregadas nesta dissertação;
- Testes em circuitos com elevados níveis de complexidade, funcionalidade e robustez, a exemplo do processador Plasma, mas com blocos internos formados por ambos os tipos de *flip-flop*: sensíveis à borda de descida e à borda de subida do sinal de relógio;
- Testes para a validação da técnica de Controle Dinâmico de Ciclo de Trabalho em *Systems-on-Chip* sujeitos a grandes variações de temperatura.

# Referências Bibliográficas

1. **Teixeira, João Paulo Cacho, et al.** *Aumento de Tolerância Dinâmica de Circuitos Eletrônicos Integrados Digitais a Variações de Tensão de Alimentação e de Temperatura*. 103.436 Portugal, 17 de Fevereiro de 2006. INESC-ID, PUCRS e Universidade de Vigo.
2. **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 : IEC - International Electrotechnical Commission, 2006. p. 47, Norma Técnica.
3. —. *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 : IEC - International Electrotechnical Commission, 2002. p. 27, Norma Técnica.
4. —. *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 : IEC - International Electrotechnical Commission, 2000. p. 37, Norma Técnica.
5. *Dependable Computing and Fault-Tolerance: Concepts and Terminology*. **Laprie, J. C.** [ed.] Ann Arbor. New York : IEEE, 1985. Proceedings. pp. 2-11.
6. **Anderson, T. e Lee, P.A.** *Fault Tolerance - Principles and Practice*. Englewood Cliffs : Prentice-Hall, 1981.
7. **Pradhan, Dhiraj K.** *Fault-Tolerant Computer System Design*. s.l. : Prentice Hall, 1996.
8. **Iyer, R. K. e Kalbarczyk, Z.** Hardware and Software Error Detection. [Online] 2002. [http://www.crhc.uiuc.edu/~kalbar/MotorolaCourse/HW&SW\\_ErrorDetection.pdf](http://www.crhc.uiuc.edu/~kalbar/MotorolaCourse/HW&SW_ErrorDetection.pdf).
9. **Bolzani, Letícia Maria Veiras.** *Explorando uma Solução Híbrida: Hardware + Software para a Detecção de Falhas em Systems-on-Chip (SoCs)*. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica. Porto Alegre : Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, 2005. Dissertação de Mestrado.
10. *Dependability: From concepts to limits*. **Laprie, J. C.** Johannesburg, South Africa : s.n., 1998. Proceedings of the IFIP International Workshop on Dependable Computing and its Applications. DCIA 98. pp. 108-126.
11. **Bardell, P. H.** *Built in Test for VLSI: Pseudorandom Techniques*. New York : s.n., 1987.
12. **Stroud, E. C.** *A Designer's Guide to Built-In Self-Test*. Boston : Kluwer Academic Publishers, 2002. pp. 15-27.
13. **Cortner, J. M.** *Digital Test Engineering*. United States of America : Wiley-Interscience, 1987. pp. 1-27.
14. **Benfica, Juliano D'Ornelas.** *Plataforma para Desenvolvimento de SoC (System-on-Chip) Robusto à Interferência Eletromagnética*. Programa de Pós-Graduação em Engenharia

Elétrica, Faculdade de Engenharia. Porto Alegre : Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, 2007. Dissertação de Mestrado.

15. **Bienert, R.W.** *EMC in Wireless Devices*. TUV Rheiland of North America : s.n., 2004.
16. **Bosch, R.** *Electromagnetic Compatibility – Definition and Need*. S. India - Suganya : s.n., 2004.
17. **IEC - International Electrotechnical Commission.** *International Electrotechnical Vocabulary - Part 161 (60050-161)*. Geneva, Switzerland : IEC - International Electrotechnical Commission, 1990. p. 66, Norma Técnica.
18. **Weston, D.A.** *Electromagnetic Compatibility; Principles and Applications*. 2ª. New York, USA : Marcel Dekker, Inc, 2001. pp. 1-17. ISBN 0-8247-8889-3.
19. **Souza, Antônio Marcos.** Eletromagnetismo, Notas de Aula. [Online] dezembro de 2007. <http://cict.inatel.br>.
20. **Krauss, J. D.; Fleisch, D.A.;** *Electromagnetics With Applications*. s.l. : Mc Graw-Hill, 1999.
21. **Zevzikovas, Marcos.** *Efeitos da Interferência Eletromagnética Conduzida em Equipamentos Eletromédicos no Ambiente Hospitalar*. Programa de Pós-Graduação em Engenharia Elétrica e de Computação. Campinas - São Paulo : Universidade Estadual de Campinas - UNICAMPI, 2004. p. 86, Dissertação de Mestrado.
22. **Kimmel, W.D.; Gerke, D.D.;** *Electromagnetic Compatibility in Medical Equipment: A Guide for Designers and Installers*. USA : IEEE and Interpharm Press, Inc., 1995. pp. IEEE and Interpharm Press, Inc. ISBN 0-935184-80-5 and 0-7803-1160-4.
23. **Nourani, Mehrdad e Attarha, Amir.** Signal Integrity: Fault Modeling and Testing in High-Speed SoCs. *Journal of Electronic Testing*. 4-5, Agosto de 2002, Vol. 18, pp. 539-554.
24. *Design in Hot-Carrier Reliability for High Performance Logic Applications.* **Fang, Peng; Tao, Jiang; Chen, J.F.; Hu, Chenming;**. Santa Clara, CA, USA : IEEE, 1998. Custom Integrated Circuits Conference. pp. 525-531. ISBN: 0-7803-4292-5.
25. **Leblebici, Yusuf.** Design Considerations for CMOS Digital Circuits With Improved Hot-Carrier Reliability. *Journal of Solid-State Circuits*. Julho de 1996, Vol. 31, 7, pp. 1014-1024.
26. *Estimation of Switching Noise on Power Supply Lines in Deep Sub-micron CMOS Circuits.* **Zhao, Shiyong e Roy, Kaushik.** Los Alamitos, CA, USA : IEEE Computer Society, 2000. 13th International Conference on VLSI Design. p. p. 168. ISSN: 1063-9667.
27. *EMI - Induced Soft Error Rate for COTS Microprocessor.* **Vargas, F.; Lopes, D. C.; Silva, J. Chaves da; Jr., D. Barros;**. 2004. 5th IEEE Latin American Test Workshop. pp. 169-172.
28. *Design for Signal Integrity: The New Paradigm for Deep-Submicron VLSI Design.* **Chen, H.H. e Wang, L.K.** 1997. International Symposium on VLSI Technology, Systems, and Applications. pp. 329-333.
29. *Interconnect Capacitance, Crosstalk, and Signal Delay for 0.35  $\mu\text{m}$  CMOS Technology.* **Cho, D.H.; Eo, Y.S.; Seung, M.H.; Kim, N.H.; Wee, J.K.; Kwon, O.K.; Park, H.S.** San Francisco, CA, USA : s.n., 1996. pp. 619-622.
30. *Dynamic Fault Test and Diagnosis in Digital Systems Using Multiple Clock Schemes and Multi-VDD Test.* **Rodriguez-Irago, M.; Andina, J. J. Rodriguez; Vargas, F.; Santos, M. B.; Teixeira, I. C.; Teixeira, J. P.;**. Los Alamitos, CA, USA : IEEE Computer Society, 2005.

11th IEEE International On-Line Testing Symposium - IOLTS. pp. 281-286. ISSN: 1530-1591.

31. **IEC - International Electrotechnical Commission.** *IEC - International Electrotechnical Commission.* [Online] International Electrotechnical Commission, 2008. [Citado em: 15 de janeiro de 2008.] <http://www.iec.ch/>.

32. **Lopes, Danniell Cavalcante.** *Estimação da Robustez de Sistemas Eletrônicos Via Injeção de Falhas por Interferências Eletromagnéticas.* Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica. Porto Alegre : Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, 2005. p. 97, Dissertação de Mestrado.

33. **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 : IEC - International Electrotechnical Commission, 2007. Norma Técnica.

34. —. *Integrated Circuits - Measurement of Electromagnetic Immunity, 150 kHz to 1 GHz - Part 3: Bulk Current Injection (BCI) Method (62.132-3).* Geneva, Switzerland : IEC - International Electrotechnical Commission, 2007. p. 37, Norma Técnica.

35. —. *Integrated Circuits - Measurement of Electromagnetic Immunity 150 kHz to 1 GHz - Part 4: Direct RF Power Injection Method (62.132-4).* Geneva, Switzerland : IEC - International Electrotechnical Commission, 2006. p. 49, Norma Técnica.

36. —. *Integrated Circuits - Measurement of Electromagnetic Immunity, 150 kHz to 1 GHz - Part 5: Workbench Faraday Cage Method (62.132-5).* Geneva, Switzerland : IEC - International Electrotechnical Commission, 2005. p. 45, Norma Técnica.

37. **Xilinx Inc.** Xilinx Data Book. [Online] 1999. <http://www.xilinx.com/partinfo/databook.htm>.

38. **Zeidman, Bob.** *Introduction to CPLD and FPGA Design.* s.l. : President The Chalkboard, 2001.

39. **Torok, Delfim Luiz.** *Projeto Visando a Prototipação do Protocolo de Acesso ao Meio em Redes Ethernet.* Porto Alegre : Programa de Pós-Graduação em Ciência da Computação da Universidade Católica do Rio Grande do Sul - PUCRS, 2001. Dissertação de Mestrado.

40. *Architecture of field programmable.* **Rose, J., El Gamal, A. e Sangiovanni-Vincentelli, A.** s.l. : Proceedings of the IEEE, 1993, Vol. 81, pp. 1013-1029.

41. **Aragão, A. C. O. S.** *Uma Arquitetura Sistólica para Solução de Sistemas Lineares Implementada com Circuitos FPGA's.* Instituto de Computação e Matemática Computacional. s.l. : Universidade de São Paulo - USP, 1998. Dissertação de Mestrado.

42. **Plessey Semiconductor.** *ERA60100 Preliminary Data Sheet.* 1989.

43. **Algotronix, Ltd.** *CAL 1024 Datasheet.* Edinburgh, Scotland : s.n., 1989.

44. **Altera.** *ACEX 1K, Embedded Programmable Logic Device Family Data Sheet.* 2001. Versão 4.1.

45. **Concurrent Logic.** *CFA6006 Field-Programmable Gate Array Data Sheet.* 1991.

46. *A Large Scale FPGA with 10K Core Cells with CMOS 0.8 μm 3-Layered Metal Process.* **Muroga, H.; Murata, H.; Sacki, Y.; Hibi, T.; Ohashi, Y.** 1991. Custom Integrated Circuits Conference - CICC'91. pp. 6.4.1-6.4.4.

47. **Atmel Corporation.** *Data Acquisition Systems Using Cache Logic FPGAs.* 1999. Application Note.
48. *An Architecture for Electrically Configurable Gate Arrays.* **El Gamal, A.; Greene, J.; Reyneri, J.; Rogoyski, E.** April de 1989, IEEE Journal of Solid-State Circuits, Vol. 24, pp. 394-398.
49. *A Very High-Speed Field Programmable Gate Array Using Metal-to-Metal Anti-Fuse Programmable Elements.* **Birkner, J.; Chan, A.; Chua, H. T.; Chao, A.; Gordon, K.; Kleinrnan, B.; Kolze, P. ; Wong, R.,** 1991. New Hardware Product Introduction at Custom Integrated Circuits Conference - CICC'91.
50. *An MPGA compatible FPGA architecture.* **Marple, D.; Cooke, L.,** 1992. ACM First International Workshop on Field Programmable Gate Array - FPGA 92. pp. 39-44.
51. **AMD Inc.** *Mach Devices High Density EE Programmable Logic Data Book.* 1990. Data Book.
52. **Baker, S.** Lattice Fields FPGA. *Eleci Fog Times.* Junho de 1991, 645, p. 1.
53. *A 5000-Gate CMOS EPLD with Multiple Logic and Interconnect Arrays.* **Wong, S.C.; So, H.C.; Ou, J.H.; Costello, J.,** Santa Clara, CA : Altera Corp., 1989. Custom Integrated Circuits Conference - CICC'89. pp. 5.8.1- 5.8.4.
54. **Reis, Ricardo Augusto da Luz.** *Concepção de Circuitos Integrados.* Porto Alegre : Sagra Luzzatto, 2000. p. 252. ISBN 8524106255.
55. **International Sematech.** *International Technology Roadmap for Semiconductors Update.* [Online] 2002. <http://www.itrs.net/>.
56. *Designing Systems-On-Chip Using Cores.* **Bergamaschi, R. A.; Lee, W. R.,** Los Angeles, California, United States : ACM, 2000. Design Automation Conference - DAC'00. pp. 420-425. ISBN: 1-58113-187-9.
57. *Automating the Design of SoCs Using Cores.* **Bergamaschi, R. A.; Bhattacharya, S.; Wagner, R.; Fellenz, C.; Muhlada, M.; White, F.; Daveau, J. M.; Lee, W. R.,** 2001. IEEE Design & Test of Computers. Vol. 18(5), pp. 32 - 45.
58. *System on Chip Design.* **Martin, G.; Chang, H.** Shanghai, China : IEEE, 2001. International Symposium on Integrated Circuits, Devices & Systems - ASIC'01. pp. 12-17. ISBN: 0-7803-6677-8.
59. *Global Modeling and Simulation of System-on-Chip Embedding MEMS Devices.* **Juneidi, Z.; Torki, K.; Martinez, S.; Nicolescu, G.; Courtois, B.; Jerraya, A.,** Shanghai, China : IEEE, 2001. International Symposium on Integrated Circuits, Devices & Systems - ASIC'01. pp. 666 - 669.
60. *Powering Networks on Chip: Energy-Efficient and Reliable Interconnect Design for SoCs.* **Benini, L.; De Micheli, G.** Montréal, P.Q., Canada : ACM, 2001. International Symposium on System Synthesis. pp. 33 - 38. ISBN:1-58113-418-5.
61. **Runner, S.; Sanaka, V.; Yu, E.,** Building an Infrastructure for IP Reuse. [Online] 15 de Maio de 2000. <http://www.eetimes.com>.
62. The National Technology Roadmap for Semiconductors. [Online] 1999. <http://public.itrs.net>.
63. *The A to Z of SoCs.* **Bergamaschi, R.A.; Cohn, J.** s.l. : IEEE / ACM, 2002. International Conference on Computer Aided Design - ICCAD. pp. 791-798. ISSN: 1092-3152.

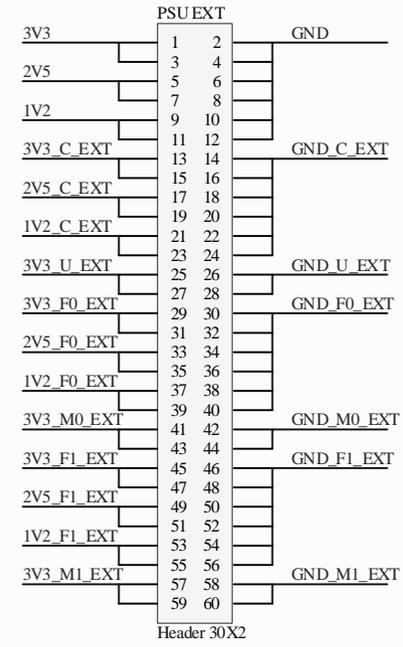
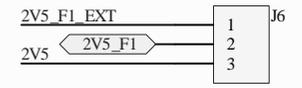
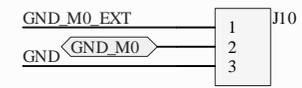
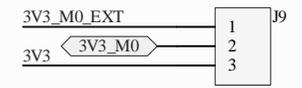
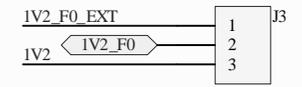
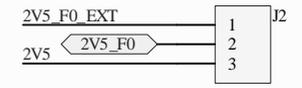
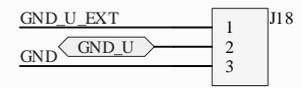
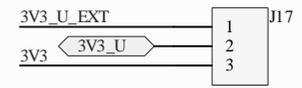
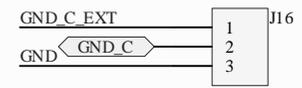
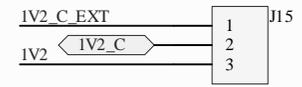
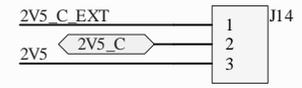
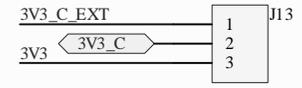
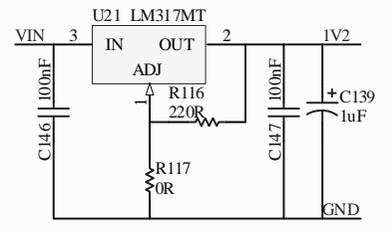
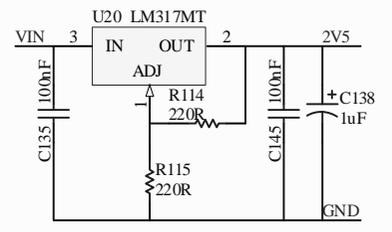
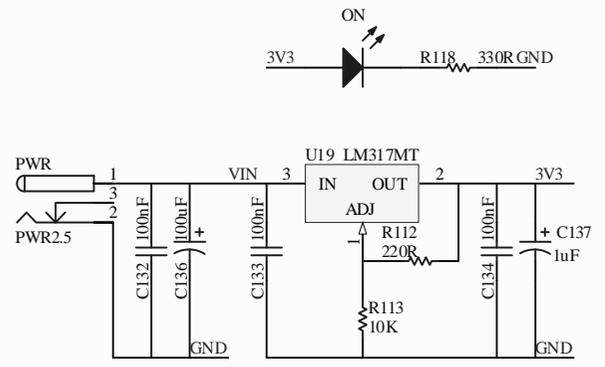
64. **Thomas, D. e Moorby, P.** *The Verilog Hardware Description Language*. s.l. : Kluwer Academic Publishers, 1991. p. 376.
65. **Perry, Douglas.** *VHDL*. 3rd Edition. New York, USA : McGraw-Hill, 1998. p. 493.
66. **Schildt, Herbert.** *C : Completo e Total*. 3. ed. São Paulo : Makron Books, 1997. p. 827.
67. **Open SystemC Initiative (OSCI).** SystemC v2.0.1 Language Reference Manual. [Online] 2003. <http://www.systemc.org>.
68. **Gajski, D.; Zhu, J.; Dömer, R.; Gerstlauer, A.; Zhao.** *Specification Language and Methodology*. Norwell, MA : Kluwer Academic Publishers, 2000. p. 336.
69. **Madiseti, V. K.; L, Shen;** Interface Design for Core-Based Systems. *IEEE Design & Test of Computers*. Outubro -Dezembro de 1997, Vol. 14, 4, pp. 45-51.
70. **Ost, Luciano Copello.** *Redes Intra-Chip Parametrizáveis com Interface Padrão para Síntese em Hardware*. Programa de Pós-Graduação em Ciência da Computação, Faculdade de Informática. Porto Alegre : Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, 2004. p. 134, Dissertação de Mestrado.
71. **Gupta, R. K.; Zorian, Y.;** Introducing Core-Based System Design. *IEEE Design & Test of Computers*. Outubro - Dezembro de 1997, Vol. 14, 4, pp. 15-25.
72. **Kumar, S.** On Packet Switched Network for Chip Communication. [ed.] Axel Jantsch e Hannu Tenhunen. *Networks on Chip*. s.l. : Kluwer Academic Publishers, 2003, 5, pp. 85-106.
73. **Zeferino, C. A.** *Redes-em-Chip: Arquiteturas e Modelos para Avaliação de Área e Desempenho*. PPGC. Porto Alegre : Universidade Federal do Rio Grande do Sul - UFRGS, 2003. Tese de Doutorado.
74. *A Generic Architecture for on-chip Acket-Switched Interconnections.* **Guerrier, P. e Greiner, A.**, 2000. Design Automation and Test in Europe - DATE'00. pp. 250-256.
75. **Hwang, K.** *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. New York : McGraw-Hill, 1993. p. 771.
76. *Networks on Chips: A New SoC Paradigm.* **Benini, L.; De Micheli, G.;** 1, Los Alamitos, CA, USA : IEEE Computer Society, Janeiro de 2002, Computer, Vol. 35, pp. 70-78. ISSN: 0018-9162.
77. *A Network on Chip Architecture and Design Methodology.* **Kumar, S.; Jantsch, A.; Soininen, J. P.; Fonsell, M.;** 2002. Computer Society Annual Symposium on VLSI - ISVLSI'02. pp. 105-112.
78. **ARM Corp.** *AMBA 2.0 Specification*. [Online] ARM Corp, 2007. <http://www.arm.com/products/solutions/AMBAHomePage.html>.
79. **Altera.** Avalon Bus Specification - Reference Manual. [Online] Altera, dezembro de 2007. [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).
80. **IBM.** The CoreConnect Bus Architecture. [Online] IBM, dezembro de 2007. <http://www.ibm.com/chips/products/coreconnect>.
81. **Xilinx Inc.** Designing Custom OPB Slave Peripherals for MicroBlaze. [Online] 8 de Fevereiro de 2002. [http://www.xilinx.com/ipcenter/processor\\_central/microblaze/doc/opb\\_tutorial.pdf](http://www.xilinx.com/ipcenter/processor_central/microblaze/doc/opb_tutorial.pdf).

82. *NOCGEN - Uma Ferramenta para Geração de Redes Intra-Chip Baseada na Infra-Estrutura HERMES*. **Moraes, Fernando Gehm; Ost, Luciano; Mello, Aline Vieira de; Palma, José Carlos Sant'anna; Calazans, Ney Laert Vilar**. Cartagena, Colômbia : s.n., 2004. X Workshop IBERCHIP. Vol. 1, pp. 210-216.
83. *Simulation, Modeling and Understanding the Importance of Signal Integrity*. **Green., L.** 6, s.l. : IEEE, Novembro de 1999, IEEE Circuits and Devices Magazine, Vol. 15, pp. pp. 7-10. ISSN: 8755-3996.
84. **Downing, R.; Gebler, P.; Katopis, G.;** Decoupling Capacitor Effects on Switching Noise. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*. Agosto de 1993, Vol. 16, 5, pp. 484-489.
85. *Full-Chip Verification of UDSM Designs*. **Saleh, R.; Overhauser, D.; Taylor, S.;** s.l. : IEEE / ACM, 1998. IEEE/ACM International Conference on Computer-Aided Design - ICCAD 98. pp. 453-460.
86. *Interconnect Optimization Strategies for High-Performance VLSI Designs*. **Kahng, A.B.; Muddu, S.; Sarto, E.;** 1999. Twelfth International Conference On VLSI Design. pp. 464-469.
87. *EMC Modeling and Simulation on Chip Level*. **Steinecke, T.; John, W.; Koehne, H.; Schmidt, M.;** 2001. IEEE International Symposium on Electromagnetic Compatibility - EMC. Vol. 2, pp. 1191-1196.
88. *Enhancing the Tolerance to Power-Supply Instability in Digital Circuits*. **Semião, J.; Freijedo, J.; Rodriguez Andina, J.J.; Vargas, F.; Santos, M.B.; Teixeira, I.C.; Teixeira, J.P.** Porto Alegre, Brasil : s.n., 2007. IEEE Computer Society Annual Symposium on VLSI - ISVLSI '07. pp. 207-212. ISBN: 0-7695-2896-1.
89. *Improving Tolerance to Power-Supply and Temperature Variations in Synchronous Circuits*. **Semião, J.; Freijedo, J.; Rodriguez-Andina, J. J.; Vargas, F.; Santos, M. B.; Teixeira, I. C.; Teixeira, J. P.** Krakow, Poland : s.n., 2007. IEEE Design and Diagnostics of Electronic Circuits and Systems - DDECS'07. pp. 1-6. ISBN: 1-4244-1162-9.
90. **CAD Group**. ITC'99 Benchmark (2nd release). [Online] Instituto Politécnico de Torino, 1999. [Citado em: ] <http://www.cad.polito.it/tools/itc99>.
91. **Xilinx Inc**. *Spartan-3 Starter Kit Board User Guide*. s.l. : Xilinx Inc., 2005. p. 64, Relatório Técnico.
92. **Xilinx Inc.;** *ChipScope Pro Software and Cores User Guide*. s.l. : Xilinx Inc., 2005. p. 222, Guia de Usuário.
93. **MIPS Technologies, Inc.;** *MIPS Technologies*. [Online] MIPS Technologies, Inc., 2008. <http://www.mips.com/>.
94. **Alecrim, Antonio A. de; Garibotti, Rafael Fraga;** *Memória Cache em uma Plataforma Multiprocessada*. Faculdade de Engenharia / Faculdade de Informática. Porto Alegre : Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, 2007. p. 67, Trabalho de Conclusão de Curso.
95. **Rhoads, Steve**. Plasma CPU Core. *Opencores*. [Online] 2007. <http://www.opencores.org/projects.cgi/web/mips/overview>.
96. **Opencores Community**. *Opencores*. [Online] 1999. <http://www.opencores.org/>.
97. **Xilinx Inc**. *MicroBlaze Processor Reference Guide*. s.l. : Xilinx Inc., 2007. p. 194, Relatório Técnico.

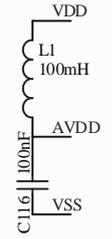
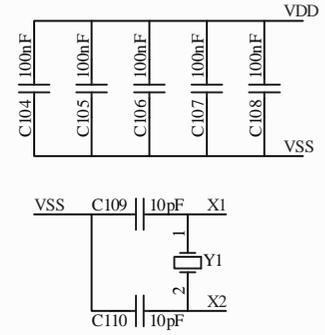
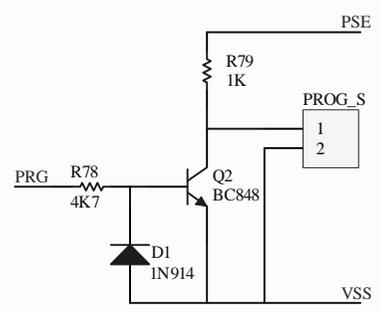
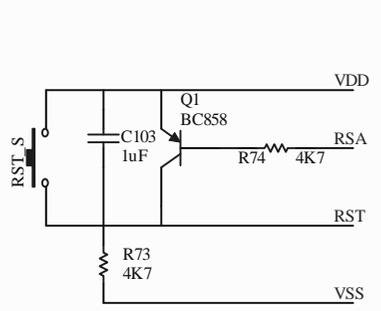
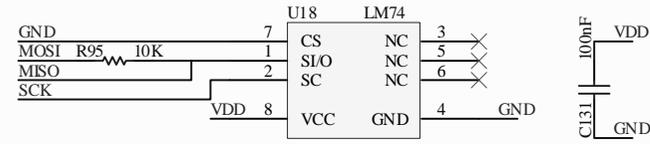
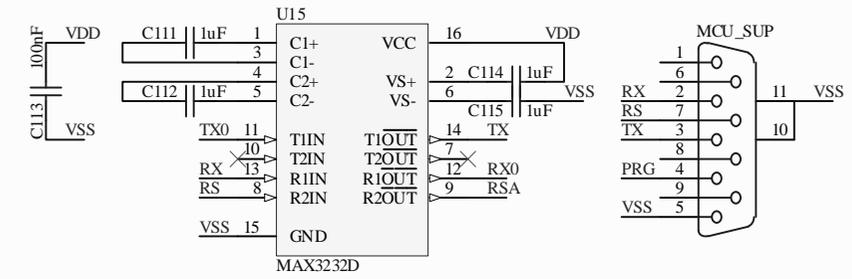
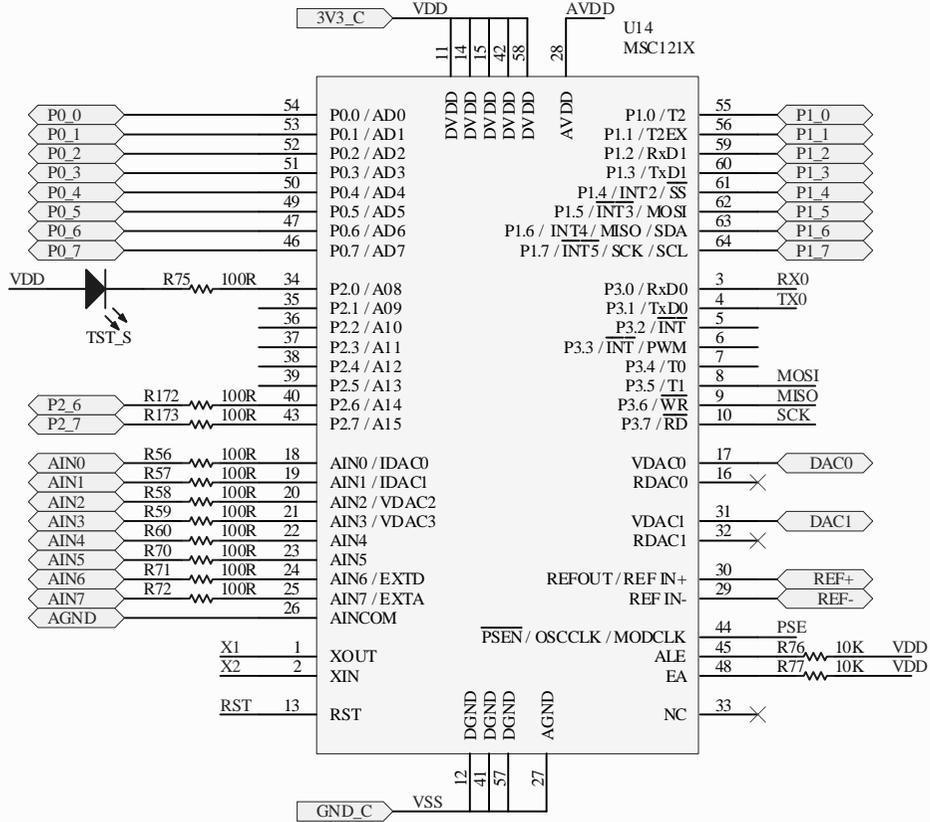
98. **Rhoad, Steve.** *Plasma CPU*. [Online] 2007. <http://plasmacpu.no-ip.org:8080>.
99. **Rijmen, Vincent; Barreto, Paulo S. L. M.;** *The Whirlpool Hash Function*. [Online] 2007. <http://paginas.terra.com.br/informatica/paulobarreto/WhirlpoolPage.html>.
100. **Rocha, Cláudia Antunes.** *Monitoramento do Fluxo de Controle de Processadores Embarcados Baseado em Profiling de Software*. Programa de Pós-Graduação em Engenharia Elétrica, Faculdade de Engenharia. Porto Alegre : Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, 2007. Dissertação de Mestrado.
101. **Piccoli, Leonardo Bisch.** *Soluções Híbridas de Hardware/Software para a Detecção de Erros em Systems-on-Chip (Soc) de Tempo Real*. Programa de Pós-Graduação em Engenharia Elétrica, Faculdade de Engenharia. Porto Alegre : Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, 2006. p. 160, Dissertação de Mestrado.
102. **Rocha, Luciano Beiestorf.** *Aplicação de Computação Móvel na Operação e Manutenção de Sistemas de Distribuição de Energia Elétrica*. Programa de Pós-Graduação em Engenharia Elétrica, Faculdade de Engenharia. Porto Alegre : Pontifícia Universidade Católica do Rio Grande do Sul, 2007. Dissertação de Mestrado.

## **Anexo A – Esquemático da Plataforma de Teste**

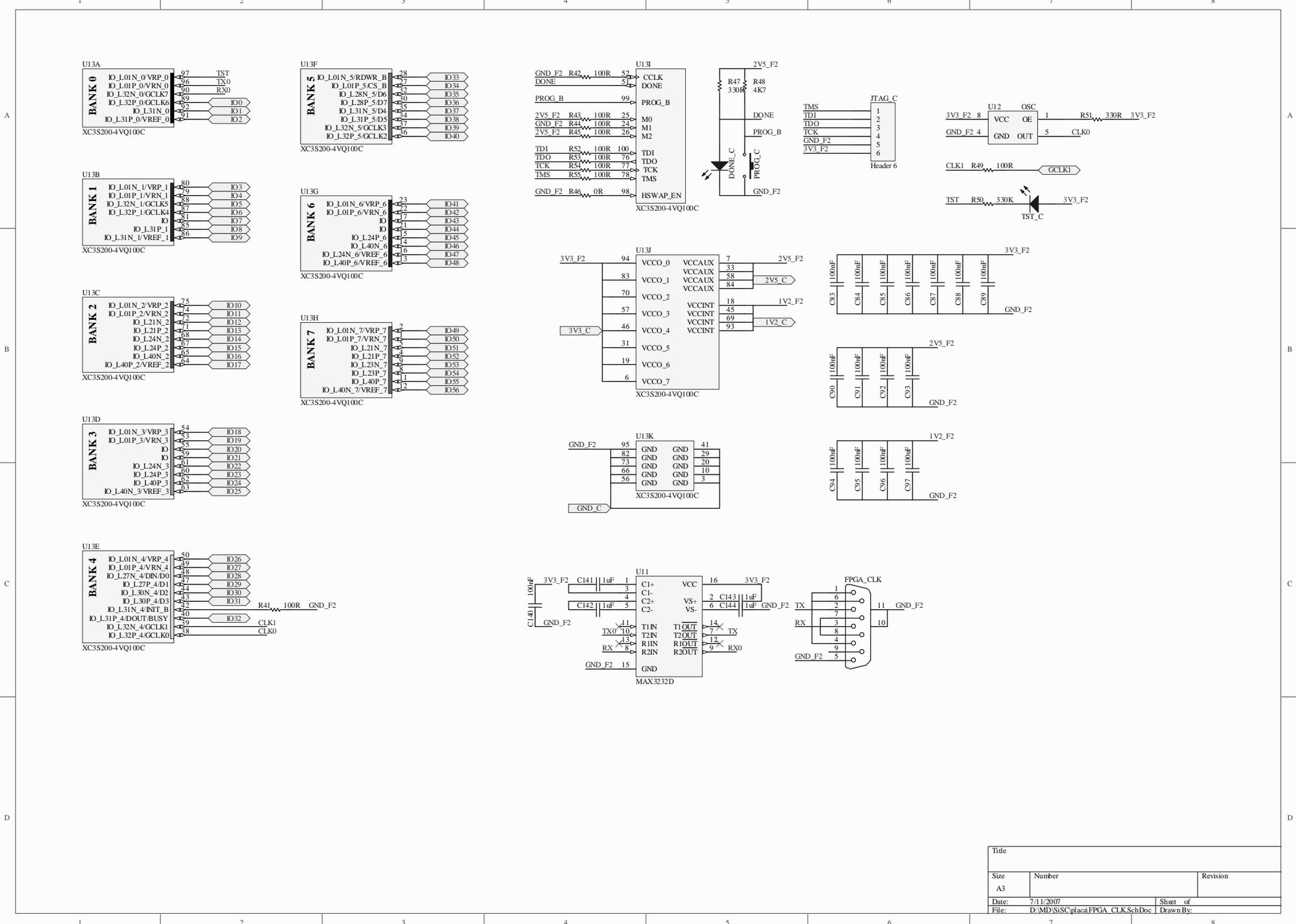




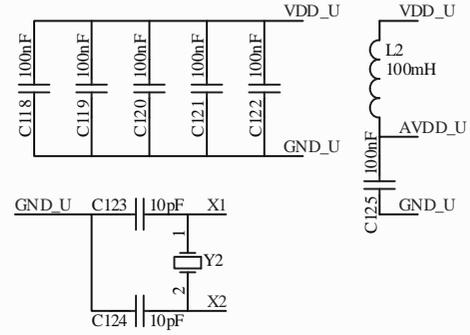
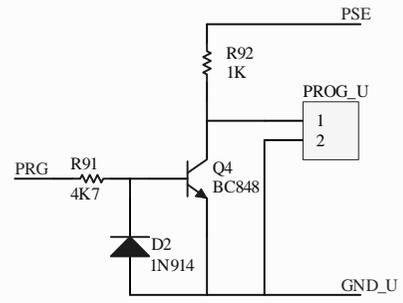
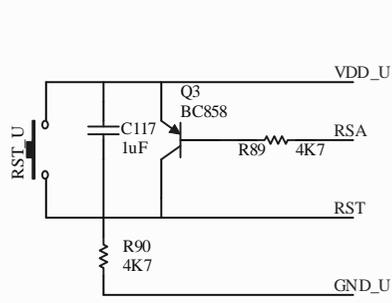
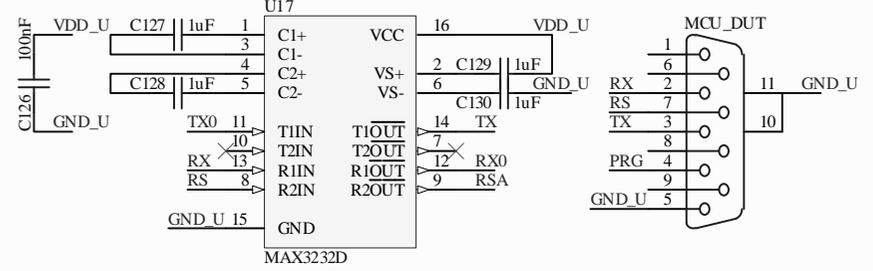
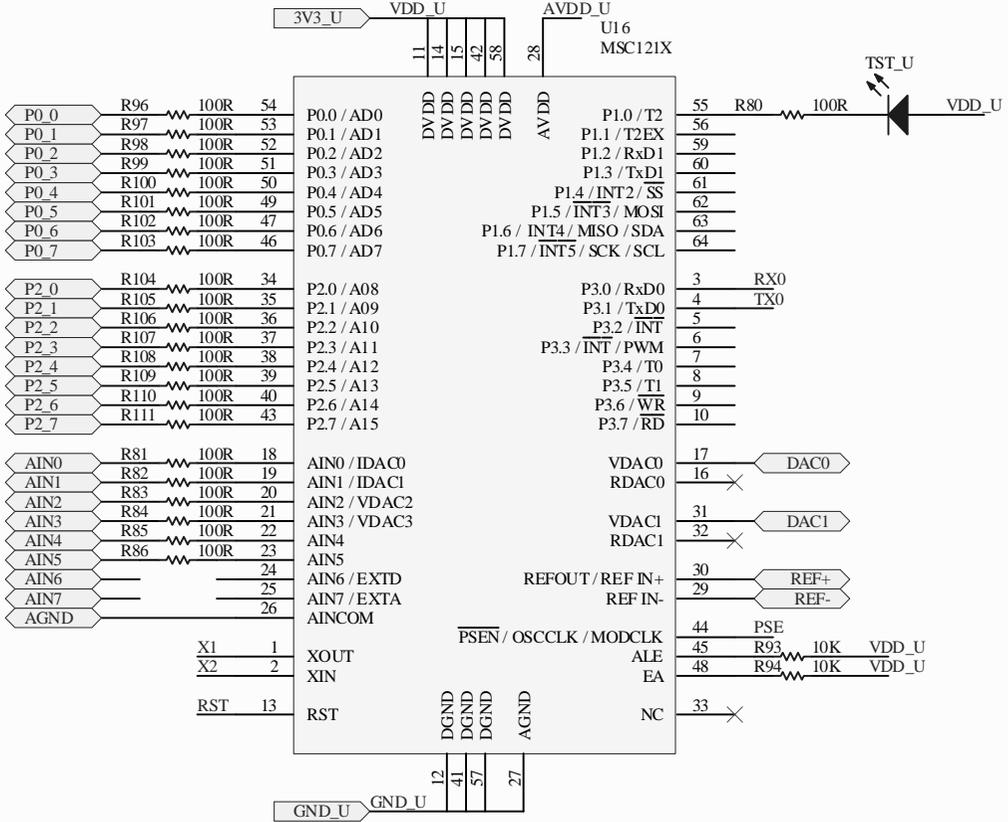
Title		
Size A4	Number	Revision
Date: 7/11/2007	Sheet of	
File: D:\MDS\ISCP\placa\PSU.SchDoc	Drawn By:	



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

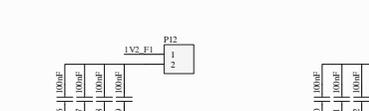
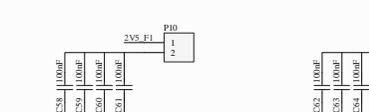
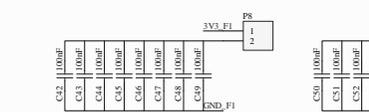
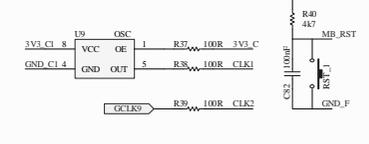
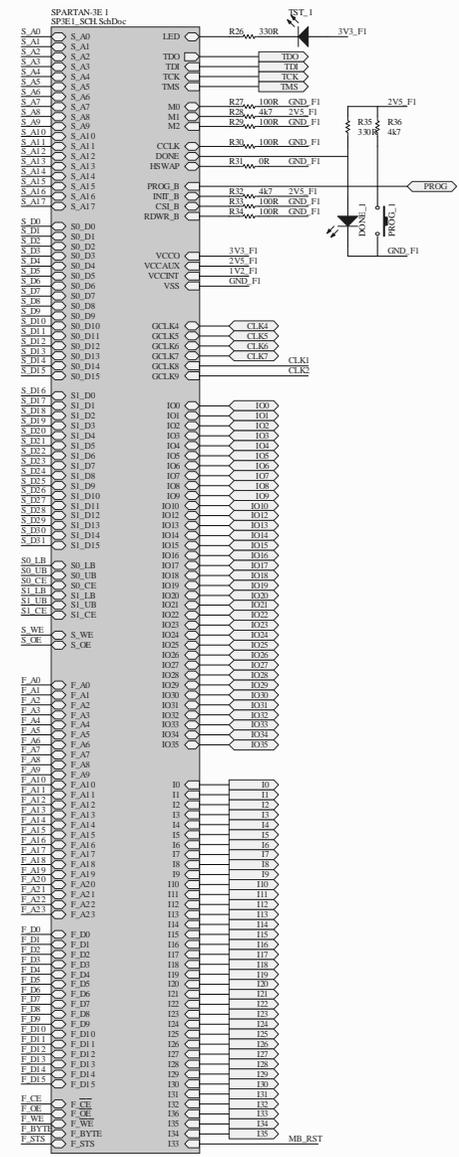
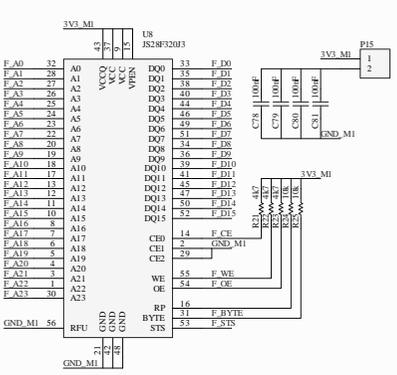
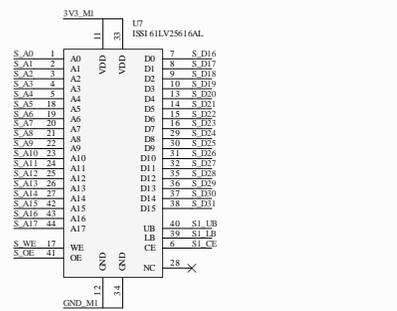
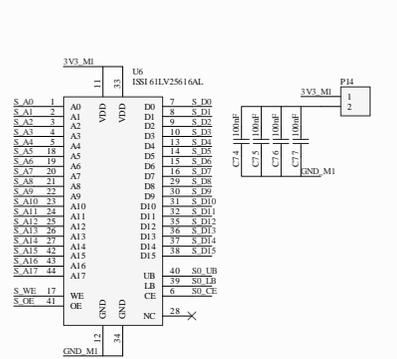


Title		
Size	Number	Revision
A3		
Date:	7/11/2007	Sheet of
File:	D:\MD\SiSC\placa\FPGA_CLK\SchDoc	Drawn By:



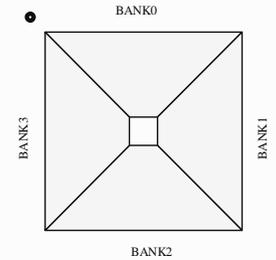
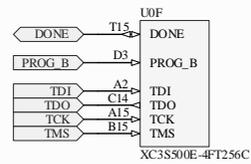
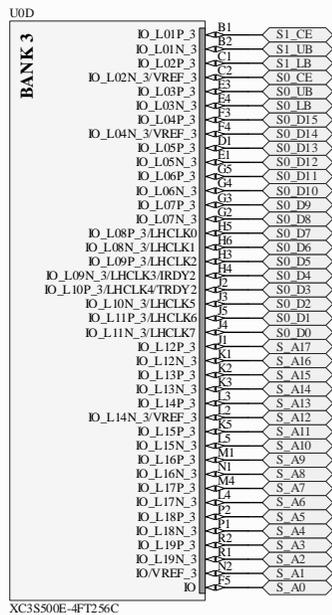
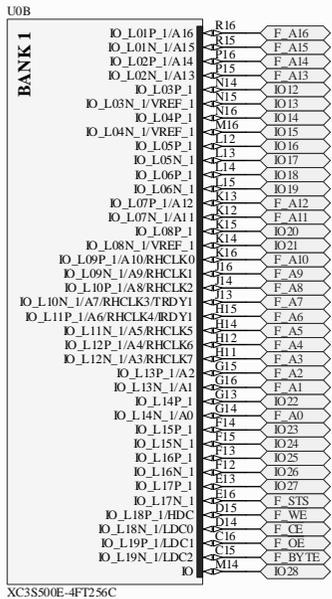
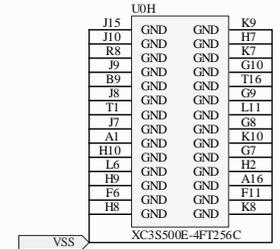
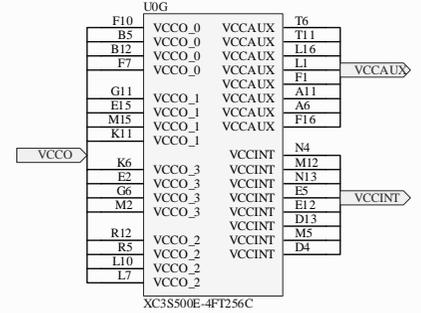
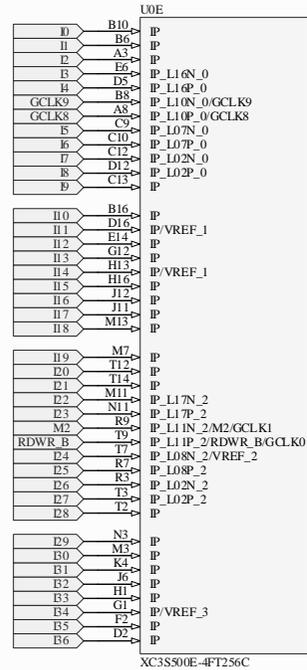
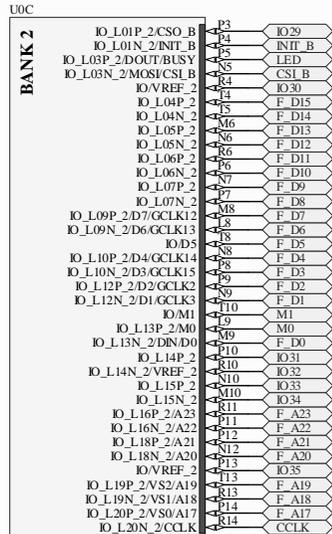
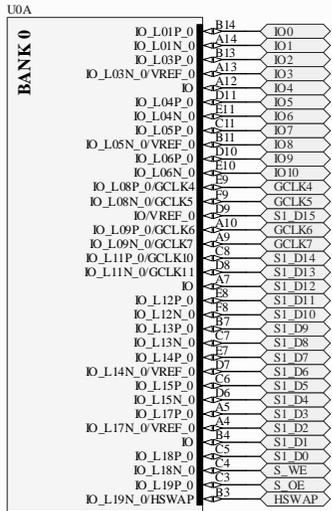
Title		
Size	Number	Revision
A4		
Date:	7/11/2007	Sheet of
File:	D:\MD\SISC\placa\MCU_DUT.SchDoc	Drawn By:



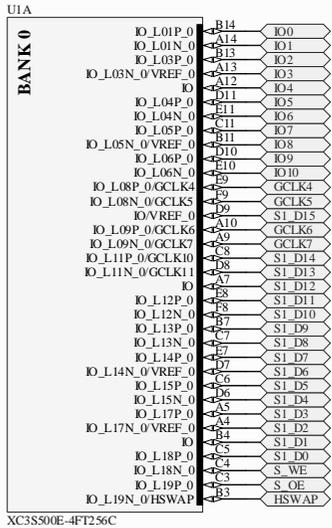


TDO, TDI, TCK, TMS  
JTAG  
Check Global da FPGA  
Reconfigura FPGA  
IO0, IO15  
Pinos de Input-Output  
R0, R12  
3V3\_C, GND\_C, 2V5\_F, 1V2\_F, GND\_F  
Alimentação da FPGA  
Alimentação das memórias

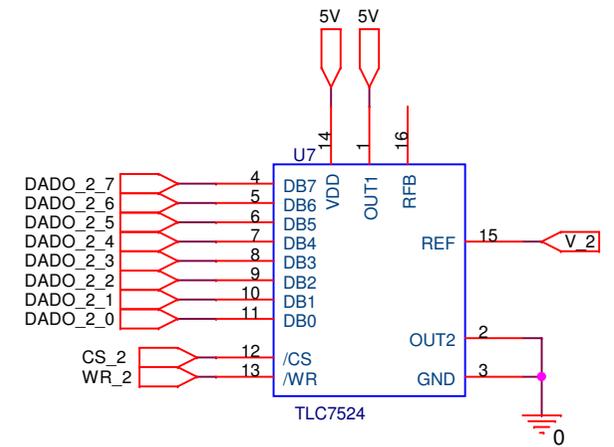
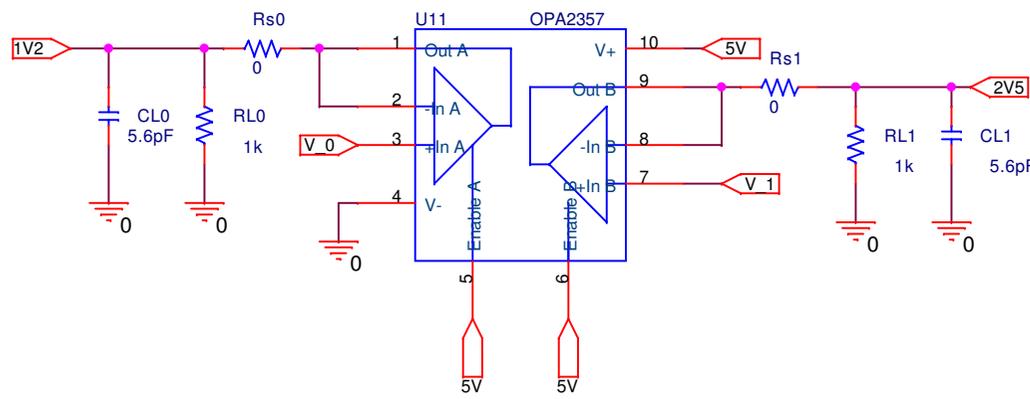
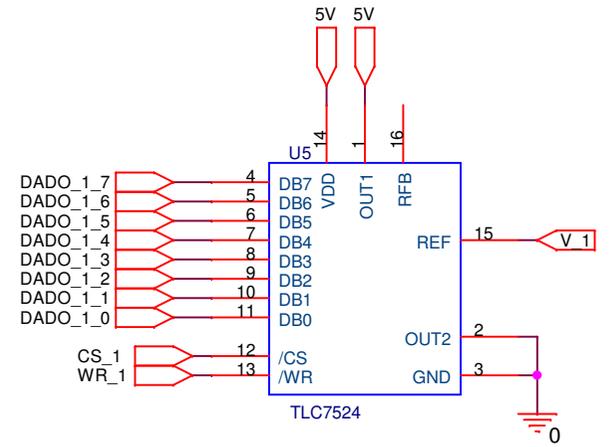
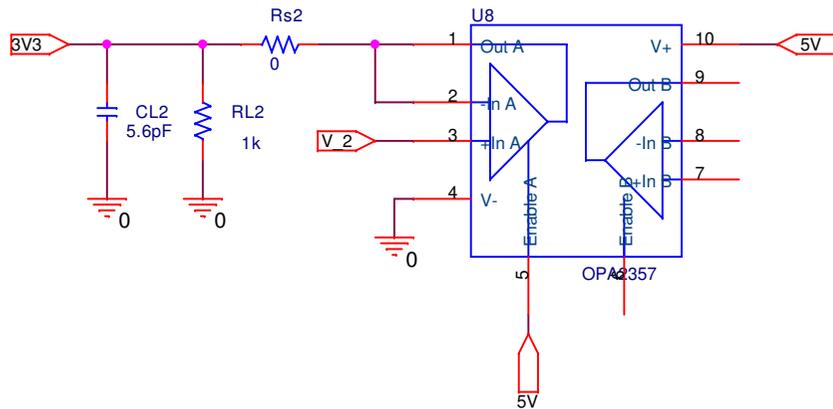
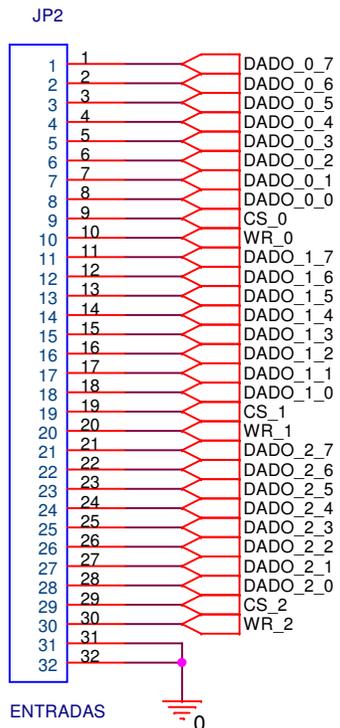
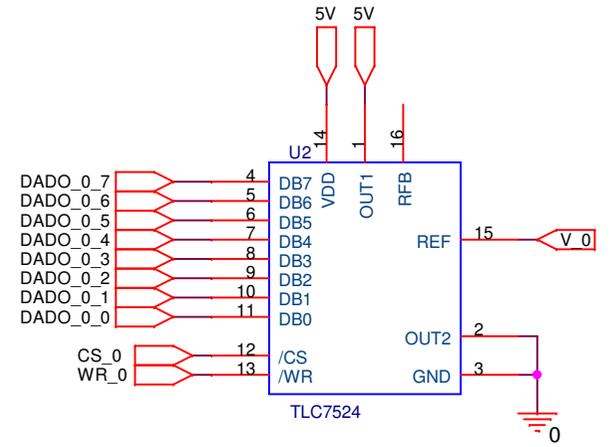
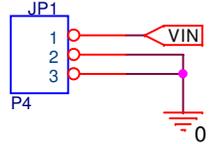
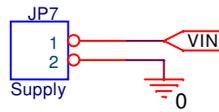
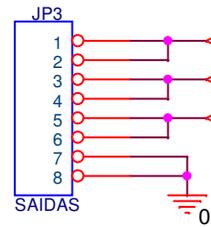
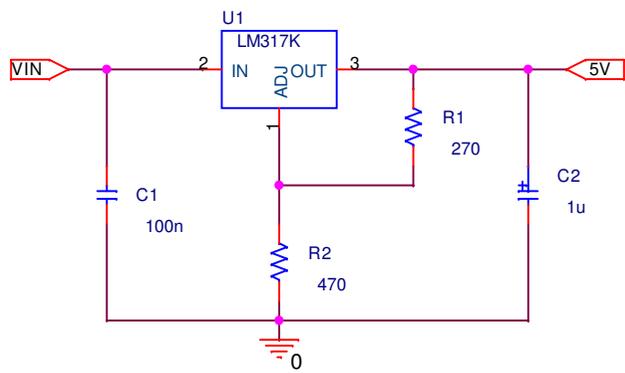
Size	Number	BLOCO FPGA 1	Revision
A2			
Date	7/11/2007		Sheet of
File	D:\MIDSS\place\FPGA1_SchDoc		Down By: MID



Title		
SPARTAN-3E 0		
Size	Number	Revision
A3		
Date:	7/11/2007	Sheet of
File:	D:\MD\SI\SC\placa\SP3E0_SCH.SchDoc	Drawn By: MD



## **Anexo B – Esquemático da Placa de Alimentação e Injeção de Falhas**



## **Anexo C – Código do Circuito *Clock* *Manager***

---

```
-- Laboratório de Sistemas Sinais & Computação - SiSC
-- Autor: Marlon Moraes
-- Projeto: CDCDC
-- Circuito: Clock Manager
-- Data de modificação: 20/12/07.
```

---

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
```

```
entity clock_manager is
port (
    clock_in:      in        std_logic:= '0';
    duty_selecao:  in        std_logic_vector(1 downto 0);
    clock_a_pin:   out       std_logic_vector(2 downto 0);
    clock_b_pin:   out       std_logic_vector(2 downto 0);
    clock_c_pin:   out       std_logic_vector(2 downto 0)
);
end clock_manager;
```

```
architecture behavioral of clock_manager is
```

```
constant DV:      integer:=4;
constant DV_50:   integer:=2;
constant DV_25r:  integer:=1;
constant DV_75f:  integer:=3;
```

```
signal clock_in_signal:  std_logic;
signal contador:         integer:=0;
signal clock_0_signal:   std_logic:= '1';
signal clock_1_signal:   std_logic:= '0';
signal clock_2_signal:   std_logic:= '0';
signal clock_3_signal:   std_logic:= '0';
```

```
begin
    sinais: process(clock_in)
    begin
        clock_in_signal <= clock_in;
    end process sinais;

    gerador: process(clock_in_signal)
    begin
        if clock_in_signal'event and clock_in_signal = '1' then
            if contador < DV-1 then
                contador <= contador+1;
            else
                contador <= 0;
            end if;
        end if;

        if clock_in_signal'event and clock_in_signal = '1' then
            if contador < DV_50 then
                clock_0_signal <= '1';
            else
                clock_0_signal <= '0';
            end if;
        end if;

        if clock_in_signal'event and clock_in_signal = '1' then
            if contador = DV_25r then
                clock_1_signal <= '1';
            else
                clock_1_signal <= '0';
            end if;
        end if;

        if clock_in_signal'event and clock_in_signal = '1' then
```

---

```
        if contador<DV_75f then
            clock_2_signal <= '1';
        else
            clock_2_signal <= '0';
        end if;
    end if;

    if clock_in_signal'event and clock_in_signal = '1' then
        clock_3_signal <= not clock_3_signal;
    end if;
end process gerador;

selecao: process(clock_in_signal)
begin
    if clock_in_signal'event and clock_in_signal = '1' then
        if duty_selecao="00" then
            clock_a_pin(0) <= clock_0_signal;
            clock_a_pin(1) <= clock_0_signal;
            clock_a_pin(2) <= clock_0_signal;

            clock_b_pin(0) <= clock_0_signal;
            clock_b_pin(1) <= clock_0_signal;
            clock_b_pin(2) <= clock_0_signal;

            clock_c_pin(0) <= clock_3_signal;
            clock_c_pin(1) <= clock_3_signal;
            clock_c_pin(2) <= clock_3_signal;

        elsif duty_selecao="01" then
            clock_a_pin(0) <= clock_0_signal;
            clock_a_pin(1) <= clock_0_signal;
            clock_a_pin(2) <= clock_0_signal;

            clock_b_pin(0) <= clock_0_signal;
            clock_b_pin(1) <= clock_1_signal;
            clock_b_pin(2) <= clock_1_signal;

            clock_c_pin(0) <= clock_3_signal;
            clock_c_pin(1) <= clock_3_signal;
            clock_c_pin(2) <= clock_3_signal;

        elsif duty_selecao="10" then
            clock_a_pin(0) <= clock_0_signal;
            clock_a_pin(1) <= clock_0_signal;
            clock_a_pin(2) <= clock_0_signal;

            clock_b_pin(0) <= clock_2_signal;
            clock_b_pin(1) <= clock_0_signal;
            clock_b_pin(2) <= clock_2_signal;

            clock_c_pin(0) <= clock_3_signal;
            clock_c_pin(1) <= clock_3_signal;
            clock_c_pin(2) <= clock_3_signal;

        elsif duty_selecao="11" then
            clock_a_pin(0) <= clock_3_signal;
            clock_a_pin(1) <= clock_3_signal;
            clock_a_pin(2) <= clock_3_signal;

            clock_b_pin(0) <= clock_3_signal;
            clock_b_pin(1) <= clock_3_signal;
            clock_b_pin(2) <= clock_3_signal;

            clock_c_pin(0) <= clock_3_signal;
            clock_c_pin(1) <= clock_3_signal;
            clock_c_pin(2) <= clock_3_signal;
        end if;
    end if;
end process selecao;
end behavioral;
```

## **Anexo D – Código do Circuito de Alimentação**

```
-----
-- Laboratório de Sistemas Sinais & Computação - SiSC
-- Autor: Marlon Moraes
-- Projeto: CDCDC
-- Circuito: Circuito de Alimentação
-- Data de modificação: 20/12/07.
-----

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity supply is
port (
  clock_in:   in    std_logic:= '0';
  reset_in:  in    std_logic:= '1';
  selecao_ruído: in  std_logic_vector(1 downto 0);
  cs_0:      out   std_logic;
  wr_0:      out   std_logic;
  cs_1:      out   std_logic;
  wr_1:      out   std_logic;
  cs_2:      out   std_logic;
  wr_2:      out   std_logic;
  dado_0:    out   std_logic_vector(7 downto 0):=(others => '0');
  dado_1:    out   std_logic_vector(7 downto 0):=(others => '0');
  dado_2:    out   std_logic_vector(7 downto 0):=(others => '0')
);
end supply;

architecture behavioral of supply is

  signal clock_signal: std_logic;
  signal reset_signal: std_logic;
  signal dado_signal_0: std_logic_vector(7 downto 0):=(others => '0');
  signal dado_signal_1: std_logic_vector(7 downto 0):=(others => '0');
  signal dado_signal_2: std_logic_vector(7 downto 0):=(others => '0');

  signal contador_ruído: integer:=0;

  constant DIV_ruído: integer:=100;

  constant A: integer :=0;
  constant B: integer :=1;
  constant C: integer :=2;
  constant D: integer :=3;

  constant N1V2: std_logic_vector(7 downto 0):="01000101";
  constant N2V5: std_logic_vector(7 downto 0):="10010001";
  constant N3V3: std_logic_vector(7 downto 0):="10111111";

  constant D1V2: std_logic_vector(7 downto 0):="00111011";
  constant D2V5: std_logic_vector(7 downto 0):="01110100";
  constant D3V3: std_logic_vector(7 downto 0):="10001111";

begin

  sinais: process(reset_in, clock_in)
  begin
    reset_signal <= not (reset_in);
    clock_signal <= clock_in;
  end process sinais;

  FSM: process(reset_signal, clock_signal)
  variable stato: integer range 3 downto 0;
  begin
    if reset_signal='1' then
      stato:=A;
      cs_0<='0';
      wr_0<='0';

```

```
        cs_1<='0';
        wr_1<='0';
        cs_2<='0';
        wr_2<='0';
        dado_0<=(others=>'0');
        dado_1<=(others=>'0');
    dado_2<=(others=>'0');
elseif rising_edge(clock_signal) then
    case stato is
        when A =>
            cs_0<='0';
            wr_0<='0';
            cs_1<='0';
            wr_1<='0';
            cs_2<='0';
            wr_2<='0';
            dado_0<=dado_signal_0;
            dado_1<=dado_signal_1;
            dado_2<=dado_signal_2;
            stato:=B;
        when B =>
            stato:=C;
        when C =>
            stato:=D;
        when D =>
            cs_0<='1';
            wr_0<='1';
            cs_1<='1';
            wr_1<='1';
            cs_2<='1';
            wr_2<='1';
            stato:=A;
        end case;
    end if;
end process FSM;

dado: process(reset_signal,clock_signal)
begin
    if reset_signal='1' then
        dado_signal_0<=(others=>'0');
    elseif rising_edge(clock_signal) then
        if selecao_ruido ="00" then
            dado_signal_0 <= N1V2;
            dado_signal_1 <= N2V5;
            dado_signal_2 <= N3V3;

            elseif selecao_ruido = "01" then
                dado_signal_0 <= D1V2;
                dado_signal_1 <= D2V5;
                dado_signal_2 <= D3V3;

            elseif selecao_ruido = "10" then
                if contador_ruido < DIV_ruido then
                    contador_ruido <= contador_ruido + 1;
                else
                    contador_ruido<= 0;
                end if;
                if contador_ruido < (DIV_ruido/2) then
                    dado_signal_0 <= N1V2;
                    dado_signal_1 <= N2V5;
                    dado_signal_2 <= N3V3;
                else
                    dado_signal_0 <= D1V2;
                    dado_signal_1 <= D2V5;
                    dado_signal_2 <= D3V3;
                end if;

            elseif selecao_ruido = "11" then
                if contador_ruido < 255 then
                    contador_ruido <= contador_ruido + 1;
```

```
        else
            contador_ruido<= 0;
        end if;
    if dado_signal_0 > 0 then
        dado_signal_0 <= dado_signal_0 - 1;
        dado_signal_1 <= dado_signal_1 - 1;
        dado_signal_2 <= dado_signal_2 - 1;
    else
        dado_signal_0<=(others=>'1');
        dado_signal_1<=(others=>'1');
        dado_signal_2<=(others=>'1');
    end if;
end if;
end process dado;
end behavioral;
```

---

# **Anexo E – Código do Circuito Sistema de Gerenciamento de Testes**

```
-----  
-- Laboratório de Sistemas Sinais & Computação - SiSC  
-- Autor: Marlon Moraes  
-- Projeto: CDCDC  
-- Circuito: Sistema de Gerenciamento de Teste  
-- Data de modificação: 20/12/07.  
-----
```

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
entity sistema is  
  port(  
    clock_in : in std_logic;  
    reset_in : in std_logic;  
    clock_a_pin : out std_logic_vector(2 downto 0);  
    clock_b_pin : out std_logic_vector(2 downto 0);  
    clock_c_pin : out std_logic_vector(2 downto 0);  
    dado_0 : out std_logic_vector(7 downto 0);  
    dado_1 : out std_logic_vector(7 downto 0);  
    dado_2 : out std_logic_vector(7 downto 0);  
    cs_0 : out std_logic;  
    wr_0 : out std_logic;  
    cs_1 : out std_logic;  
    wr_1 : out std_logic;  
    cs_2 : out std_logic;  
    wr_2 : out std_logic;  
    duty_selecao : in std_logic_vector(1 downto 0);  
    selecao_ruido : in std_logic_vector(1 downto 0)  
  );  
end sistema;
```

```
architecture behavioral of sistema is
```

```
  component clock_manager  
    port(  
      clock_in:      in      std_logic:='0';  
      duty_selecao:  in      std_logic_vector(1 downto 0);  
      clock_a_pin:  out      std_logic_vector(2 downto 0);  
      clock_b_pin:  out      std_logic_vector(2 downto 0);  
      clock_c_pin:  out      std_logic_vector(2 downto 0)  
    );  
  end component;
```

```
  component supply  
    port(  
      clock_in:      in      std_logic:='0';  
      reset_in:      in      std_logic:='1';  
      selecao_ruido: in      std_logic_vector(1 downto 0);  
      cs_0:          out      std_logic;  
      wr_0:          out      std_logic;  
      cs_1:          out      std_logic;  
      wr_1:          out      std_logic;  
      cs_2:          out      std_logic;  
      wr_2:          out      std_logic;  
      dado_0:        out      std_logic_vector(7 downto 0):=(others => '0');  
      dado_1:        out      std_logic_vector(7 downto 0):=(others => '0');  
      dado_2:        out      std_logic_vector(7 downto 0):=(others => '0')  
    );  
  end component;
```

```
begin
```

```
  clock_manager_0:  clock_manager  
    port map(  
      clock_in      => clock_in,  
      duty_selecao  => duty_selecao,  
      clock_a_pin   => clock_a_pin,
```

```
        clock_b_pin => clock_b_pin,
        clock_c_pin => clock_c_pin
    );

supply_0: supply
    port map(
        clock_in      => clock_in,
        reset_in      => reset_in,
        selecao_ruido => selecao_ruido,
        cs_0           => cs_0,
        wr_0           => wr_0,
        cs_1           => cs_1,
        wr_1           => wr_1,
        cs_2           => cs_2,
        wr_2           => wr_2,
        dado_0         => dado_0,
        dado_1         => dado_1,
        dado_2         => dado_2
    );

end behavioral;
```

---

**Anexo F – Código *Software* do  
Microcontrolador de Gerenciamento  
de Teste**

```
-----
-- Laboratório de Sistemas Sinais & Computação - SiSC
-- Autor: Marlon Moraes
-- Projeto: CDCDC
-- Data de modificação: 20/12/07.
-----

#include <stdio.h>
#include "mcs1211.h"
#include "delay.h"
#include "display.h"
#include "serial_9600.h"

#define sel_0 P1_0
#define sel_1 P1_1
#define ruido_0 P1_2
#define ruido_1 P1_3
#define reset P1_4
#define prog_0 P2_6
#define prog_1 P2_7

char dado_serial;

void mensagem()
{
    serial_string("\n*****\n");
    serial_string(" Seleccione:\n");
    serial_string("    N - duty-cycle normal (50%) \n");
    serial_string("    R - duty-cycle rise-edge (25%) \n");
    serial_string("    F - duty-cycle fall-edge (75%) \n");
    serial_string("    V - tensao nominal (1.2V - 2.5V - 3.3V) \n");
    serial_string("    Q - queda de tensao pulsante (queda) \n");
    serial_string("    D - queda tensao fixa \n");
    serial_string("    T - rampa para teste \n");
    serial_string("    S - reset supply \n");
    serial_string("*****\n");
}

/***** Main *****/
void main (void)
{
    serial_init();
    serial_string("\nMarlon L. Moraes\n");
    serial_string("Controle do Clock Manager...\n");
    mensagem();

    prog_0 = 1;
    prog_1 = 1;

    sel_0 = 0;
    sel_1 = 0;

    ruido_0 = 0;
    ruido_1 = 0;

    while(1)
    {
        dado_serial=serial_getc();
        if (dado_serial == 'N')
        {
            sel_0 = 0;
            sel_1 = 0;
            serial_string("\nclock A 50% com divisao de frequencia por 4...\n");
            serial_string("clock B 50% com divisao de frequencia por 4...\n");
            serial_string("clock C 50% com divisao de frequencia por 2...\n");
            mensagem();
        }
        if (dado_serial == 'R')
```

```
{
sel_0 = 1;
sel_1 = 0;
serial_string("\nclock A 50% com divisao de frequencia por 4...\n");
serial_string("clock B 25% com divisao de frequencia por 4...\n");
serial_string("clock C 50% com divisao de frequencia por 2...\n");
mensagem();
}
if (dado_serial == 'F')
{
sel_0 = 0;
sel_1 = 1;
serial_string("\nclock A 50% com divisao de frequencia por 4...\n");
serial_string("clock B 75% com divisao de frequencia por 4...\n");
serial_string("clock C 50% com divisao de frequencia por 2...\n");
mensagem();
}

if (dado_serial == 'A')
{
sel_0 = 1;
sel_1 = 1;
serial_string("\nATENCAO!!!\n");
serial_string("\nclock A 50% com divisao de frequencia por 2...\n");
serial_string("clock B 50% com divisao de frequencia por 2...\n");
serial_string("clock C 50% com divisao de frequencia por 2...\n");
mensagem();
}

if (dado_serial == '0')
{
prog_0 = 0; ).
delay(100);
prog_0 = 1;
serial_string("\n-- A T E N C A O --\n");
serial_string("\nFPGA 0 resetada e esperando programação...\n");
mensagem();
}

if (dado_serial == '1')
{
prog_1 = 0; ).
delay(100);
prog_1 = 1;
serial_string("\n-- A T E N C A O --\n");
serial_string("\nFPGA 1 resetada e esperando programação...\n");
mensagem();
}

if (dado_serial == 'V')
{
ruído_0 = 0;
ruído_1 = 0;
serial_string("\ntensao de alimentacao nominal (1.2V 2.5V 3.3V)...\n");
mensagem();
}

if (dado_serial == 'T')
{
ruído_0 = 1;
ruído_1 = 1;
serial_string("\ rampa ... \n");
mensagem();
}

if (dado_serial == 'Q')
{
ruído_0 = 0;
ruído_1 = 1;
serial_string("\ queda de pulsante... \n");
```

```
    mensagem();
}

if (dado_serial == 'D')
{
    ruido_0 = 1;
    ruido_1 = 0;
    serial_string("\ queda de tensao fixa...\n");
    mensagem();
}

if (dado_serial == 'S')
{
    reset = 0;
    delay(500);
    reset = 1;
    serial_string("\ supply resetada...\n");
    mensagem();
}
}
}
```

---

## **Anexo G – Planilhas dos Testes de Redução de Tensão**

Tensões (V)			Readback		Falha no Arquivo de Saída							
1v2	2v5	3v3	F0	F1	F0			F1				
					M	W	F	M	W	F		
905mV	1,847V	2,478V	ok	ok			2					
			ok	ok				19	30	7		
			ok	ok			1					
			ok	ok			1					
			ok	ok			2					
			ok	ok								
			ok	ok			2					
			ok	ok			1					
			ok	ok								
			ok	ok			2					
			ok	ok			1					
			ok	ok						28	57	13
			ok	ok								
			ok	ok								
			ok	ok					1			
			ok	ok					2			
			ok	ok								
			ok	ok								
			ok	ok					1			
			ok	ok					2			
			ok	ok								
			ok	ok								
			ok	ok					1			
			ok	ok					1			
			ok	ok					1			
			ok	ok								
			ok	ok					2			
			ok	ok					1			
			ok	ok								
			ok	ok					2			
			ok	ok					2			
			ok	ok					1			
ok	ok				2		47	68	6			
ok	ok						1					
ok	ok						1					
ok	ok											
ok	ok						2					
ok	ok						1					
ok	ok						1					
ok	ok						2					

## **Anexo H – Planilha dos Testes de EMI Irradiada**

Frequência (MHz)	Campo (V/m)	Tensões (V)			Arquivo	Readback		Travou		Falha no Arquivo de Saída						Observações		
		1v2	2v5	3v3		F0	F1	F0	F1	F0			F1					
										M	W	F	M	W	F			
800	154	1,02	2	2,64	1199804827	ok	ok		ok				6				F0 com o Plasma modificado (técnica). F1 com o Plasma original. Entradas iguais a Porto Alegre.	
					1199805400	ok	ok		ok			7						
					1199805855	ok	ok		ok			8						
					1199806566	ok	ok		ok			7						
					1199807170	ok	ok		ok			7						
					1199807797	ok	ok		ok			6						
					1199808377	ok	ok		ok			7						
					1199815123	ok	ok		ok							16		F0 com o Plasma original e F1 com o Plasma modificado (técnica). Entradas iguais a Porto Alegre.
		1199816651	ok	ok		ok								F0 com o Plasma original e F1 com o Plasma modificado (técnica). Entradas embaralhadas.				
-	-	1,2	2,5	3,3	1199817243	ok	ok								Somente EMI.			
-	-	1,02	2	2,64	119981026	ok	ok		ok						F0 com o Plasma original e F1 com o Plasma modificado (técnica). Entradas embaralhadas e EMI.			
-	-				1199818591	ok	ok								Somente quedas de tensão.			
-	-	1,2	2,5	3,3	1199880837	ok	ok									F0 com o Plasma Original. F1 com o Plasma modificado (técnica). Entradas Embaralhadas. (Testes Realizados pela manhã)		
-	-	1,032	2,5	2,5	1199881431	ok	ok											
-	-				1199882034	ok	ok											
-	-				1199882581	ok	ok											
-	-				1199883143	ok	ok											
-	-				1199883695	ok	ok											
-	-				1199884323	ok	ok	ok	ok									
850	187	1,056	2,05	2,574	1199884725	ok	ok	ok	ok									
					1199885260	ok	ok	ok	ok									
					1199885756	ok	ok								1			
					1199886299	ok	ok											
					1199886840	ok	ok								2			
					1199887399	ok	ok								4			
		1199887967	ok	ok								2						
-	-				1199888505	ok	ok											
-	-				1199889299	ok	ok							1				
-	-	1,2	2,5	3,3	1199902514	ok	ok								Geração do arquivo gold.			
-	-				1199902901	ok	ok					5			F0 com o Plasma modificado (Técnica).			
-	-				1199903297	ok	ok					2						
-	-				1199903672	ok	ok											
					11999054722	ok	ok					8						
					1199905764	ok	ok					6						
					1199906100	ok	ok					6						





					1200060590	ok	ok					4								
-	-			2,64	1200061429	ok	ok													
850	10dBm	1,08	2,125	2,805	1200061898	ok	ok													
					1200062769	ok	ok													
					1200064148	ok	ok													
					1200065063	ok	ok													
					1200065721	ok	ok													
					1200069681	ok	ok													
					1200070177	ok	ok													
<p>F0 com Plasma Original.  F1 com Plasma Modificado (Técnica).  Sequência Modificada.</p>																				

## **Anexo I – Publicações**

Semião, Jorge. ; Rodriguez-Irago, Marcial ; Moraes, Marlon Leandro ; Vargas, F. ; Santos, M. B. ; Teixeira, I. C. ; Andina, Juan José Rodríguez ; Teixeira, J. P. . System-on-Chip Signal Integrity Enhancement to Power Supply Disturbances. In: 6th International Workshop on Electromagnetic Compatibility of Integrated Circuits - EMC COMPO'07, 2007, Torino. Proceedings of the 6th International Workshop on Electromagnetic Compatibility of Integrated Circuits (EMC COMPO'07). Los Alamitos Circle, CA - USA : IEEE Computer Society, 2007. v. 1.

F. Vargas, J. Benfica, M. Moraes, E. Gatti, L. Garcia, D. Lupi, F. Hernandez, J. Semião, J. Freijedo, M. B. Santos, I. C. Teixeira, J. J. Rodríguez Andina, J. P. Teixeira, “SoC Prototyping Environment for EMC-Dependability Measurements”, DECIDE 2007 – First International Workshop on Dependable Circuit Design, Buenos Aires, Argentina, December 6–7, 2007.

J. Semião, J. Freijedo, M. Moraes, M. Mallmann, C. Antunes, J. Benfica, F. Vargas, M. Santos, I. C. Teixeira, J. J. Rodríguez Andina, J. P. Teixeira, D. Lupi, E. Gatti, L. Garcia, F. Hernandez. “Power-Supply Instability Aware Clock Signal Modulation for Digital Integrated Circuits”, EMC Europe 08 – Hamburg, Germany, 2008.

# System-on-Chip Signal Integrity Enhancement Based on Dynamic Clock Management

J. Semião<sup>2</sup>, J. Freijedo<sup>1,4</sup>, M. Moraes<sup>3</sup>, F. Vargas<sup>3</sup>, M. B. Santos<sup>1</sup>,  
I. C. Teixeira<sup>1</sup>, J. J. Rodríguez Andina<sup>4</sup>, J. P. Teixeira<sup>1</sup>

<sup>1</sup> IST / INESC-ID Lisboa, R. Alves Redol, 9, 3º, 1000-029 Lisboa, Portugal

<sup>2</sup> Univ. of Algarve – UAlg / EST, Campus da Penha, 8005-139 Faro, Portugal

<sup>3</sup> PUCRS - Electrical Engineering Dept., Av. Ipiranga, 6681, 90619-900 Porto Alegre, Brazil

<sup>4</sup> Univ. of Vigo, Dept. de Tecnología Electrónica, Campus Universitario, 36310 Vigo, Spain

[vargas@computer.org](mailto:vargas@computer.org), [paulo.teixeira@ist.utl.pt](mailto:paulo.teixeira@ist.utl.pt)

**Abstract** - As IC technology scales down, interconnect issues are becoming one of the major concerns of gigahertz System-on-Chip (SoC) design. Voltage distortion (power supply noise) and delay violations (signal and clock skews) dramatically contribute to signal integrity loss. As a consequence, performance degradation, reliability problems and ultimately, functional error occur. In this paper, we propose a new methodology to enhance SoC signal integrity with respect to power/ground voltage transients, without degrading its performance. The underlying principle of the proposed methodology is to dynamically adapt the clock duty-cycle (CDC) according to the signal propagation delay through the logic whose power supply voltage is being disturbed. The methodology is based on architecture with a built-in power supply voltage sensor (BIPS), which monitors abnormal power grid activity, and a clock stretching logic (CSL) block, used to increase clock duty-cycle accordingly. Moreover, a model to accurately quantify CDC stretching as a function of  $V_{DD}$  and/or Ground fluctuations is proposed. It is shown that the proposed methodology makes, for a set of valuable applications, the digital circuit more robust to power line fluctuations while maintaining at-speed clock rate. Moreover, when clock frequency reduction is inevitable, it enhances circuit tolerance when the disturbances start to occur, allowing the clock generator to react and reduce its frequency. This improves signal integrity when fast  $V_{DD}$  disturbances occur. Practical experiments based on FPGA implementation for a simple benchmark circuit were performed to demonstrate the assertions.

## 1. INTRODUCTION

Signal integrity is the ability of a signal to generate correct responses in a circuit. It generally includes all effects that cause a circuit malfunction due to the distortion of the signal waveform [1]. According to this definition, a signal with good integrity presents: (i) voltage values at required levels and (ii) level transitions at required times. For instance, an input signal to a flip-flop with good signal integrity arrives early enough to guarantee the setup and hold times and it does not have spikes that may cause undesired logic transitions.

Various signal integrity problems have been studied for high-speed gigahertz nanometer System-on-Chip (SoC). The most important ones are: (a) *crossstalk* (signal distortion due to cross coupling effects between signals) [2,3]; (b) *overshoot/undershoot* (momentarily signal rising/decreasing above/below the power supply voltage ( $V_{DD}$ ) and ground ( $V_{SS}$ ) lines [4,5]; (c) *reflection* (echoing back a portion of a signal, at high-frequency circuits, where interconnections behave as transmission lines); (d) *electromagnetic interference* – EMI

(resulting from antenna properties) [6,7,24]; (e) *power-supply noise* [8,9]; and (f) *signal skew* (delay in arrival time to different receivers) [10-12].

In this work, we are concerned with the power supply disturbances, regardless of their origin, and their impact on digital circuit performance. Both  $V_{DD}$  depletion and ground elevation cause *propagation delay through signal path increase*. It is assumed that well-known techniques [13,14] are used to limit the impact of power supply disturbances on clock generation and distribution performance. In this scenario, if a signal propagation delay is large enough, it will induce a *de-synchronization effect* due to the increased difference between the critical path propagation delay and the clock distribution network delay. Hence, e.g., 10% power supply voltage fluctuations may translate in more than 10% timing inaccuracy, causing a functional error [12]. Moreover, noise margins are reduced in the presence of power-supply voltage transients. This problem gets worse with technology scaling down. Interestingly, research results on delay fault detection and diagnosis can be reused to enhance delay fault tolerance (and thus signal integrity) to power supply voltage transients [15].

Many design and fabrication solutions have been proposed to enhance signal integrity in the presence of power grid activity. However, none of them guarantees perfect solution for the de-synchronization effect described above. In addition to the uncertainties that result from using these solutions, their use often requires the knowledge of the *instantaneous maximum power supply current*,  $i_{DD}(t)$ , for delay analysis (as opposed to power dissipation that relates to average current). The estimation of this parameter for complex designs is frequently a burden task, since it is strongly dependent on input-pattern *sequence*, even if the output is not switching. Some of the proposed solutions include: 3-D layout modeling and parasitic extraction [16]; accurate RLC simulation of on-chip power grid [10] using decoupling capacitors [8,17] and to improve resistive voltage drop (IR-drop) [10,18]; insertion of buffers on the grid [16]; and shielding wires (e.g., grounding every other line) [19]. Buffer insertion and transistor resizing methods have also been proposed [20,21] to achieve better power-delay and area-delay trade-offs. Additionally, self-test methodologies and on-chip probes to monitor intra-packaging EM-emission activities [22] have been developed to test signal integrity in high-speed SoC. Finally, other techniques have been discussed in the literature [24]. For example, the reduction of the maximum distance between supply pins and the circuit supply connections [23]. This can be achieved, for instance, by implementing *finger-shaped power distribution networks* to feed  $V_{DD}$  and ground signals throughout the internal blocks of the IC.

Undesired voltage transients in power/ground

interconnects may be constrained, but not eliminated. In order to guarantee correct timing performance in their presence, the ultimate solution is to reduce the clock signal frequency. For some applications, such loss in performance cannot be tolerated. However, in a significant set of applications, a new solution can be applied, which does not require clock frequency reduction. The underlying principle of the proposed methodology is to dynamically adapt the clock duty-cycle (CDC) according to the signal propagation delay through the logic whose power supply voltage is being disturbed. The functionality to implement the underlying principle is as follows: (1) power supply voltage monitoring and (2) when voltage variation exceeds a user's defined threshold, CDC modulation. Hence, the correspondent architecture contains: (1) a **built-in** power supply voltage transient **g**enerator (BIPS) block to monitor power grid activity ( $V_{DD}$  undershoots and/or ground overshoots); (2) a **clock stretching** logic (CSL) block, to enhance the clock duty-cycle provided by the phase-locked loop (PLL) block to the controlled logic.

Therefore, the main advantage of the proposed methodology is to render the circuit more robust to power line fluctuations by maintaining *at-speed clock rate*. As such fluctuations are often due to circuit operation, they tend to cause local disturbances in the power grid. Hence, it is a waste to reduce the clock frequency in the overall system. The proposed methodology locally adapts the CDC without loosing circuit performance.

Another problem is that clock signal generators (like PLL) need time to reduce clock frequency, when a disturbance occurs and such reduction is mandatory for reliable operation. The proposed methodology can add some tolerance to the circuit when the disturbances start to occur, allowing clock generators to react and reduce their frequency. This also improves signal integrity when fast disturbances occur.

This paper is organized as follows: *Section 2* shows how stretching the clock duty-cycle can enhance the tolerance to power/ground voltage disturbances. *Section 3* describes the proposed methodology; and introduces the basic concepts related to the BIPS and the CSL blocks. This section also presents the applicability situations and the basic principles for the application of the proposed method. In *Section 4*, practical experiments based on a FPGA implementation of an ITC'99 benchmark circuit were accomplished to demonstrate the proposed approach effectiveness. Finally, *Section 5* outlines the main conclusions and directions for future work.

## 2. CLOCK DUTY CYCLE STRETCHING

As referred, lowering the power supply voltage,  $V_{DD}$ , enhances the propagation delay of signal paths. Hence, if the observation pace is kept invariant (at-speed circuit operation), lowering  $V_{DD}$  while maintaining nominal clock period,  $t_{CLK}$ , reduces circuit noise and time margins, which will, ultimately, induce system functional errors. The time margin is characterized by the *time slack*,  $t_{PM}$ . In this work, the time slack is defined as the time interval difference between the clock period and the time interval associated with the time response of the critical path in the slowest combinational module between registers.

Let us also consider another phenomenon, the *ground bounce*, also known as *simultaneous switching noise*. This phenomenon occurs when internal nodes of a logic device change state. When this happens, the charge remaining in the internal nodes ( $C_L$ ) is drained through the ground grid. The currents associated with this charge removal are added to the switching currents flowing from  $V_{DD}$  to the ground interconnects due to the simultaneous switching of the N- and P-networks. Thus, the resulting total current flowing through the ground grid induces a local voltage variation, namely, the ground bounce,  $V_{GB}$ . Worst case conditions exist when a large

number of nodes simultaneously switch. In this case, the resulting switching currents from each node capacitance are added together. The total current flowing through the ground lead reduces noise margins and increases the probability of system malfunction.

For the sake of simplicity, we normalize the *size of the disturbance* on  $V_{DD}$ /ground interconnects voltage using a *gamma parameter*:

$$\gamma(vdd) = \frac{\Delta V_{DD}}{V_{DDnom}} \quad \text{or} \quad \gamma(gnd) = \frac{\Delta GND}{V_{DDnom}} \quad (1)$$

where:  $\Delta V_{DD}$  is the difference between the nominal  $V_{DD}$  ( $V_{DDnom}$ ) and the depleted  $V_{DD}$ , and  $\Delta GND = \Delta V_{SS}$  is the difference between the elevated ground and  $V_{SS} = 0$  Volt.

Fig. 1 shows SPICE simulation results for a 77-inverter chain designed with the 130nm UMC CMOS technology ( $V_{DDnom} = 1.2V$ ). The inverter chain has been selected to illustrate the timing behavior of a switching long signal path. The clock period is  $t_{CLK} = 1$  GHz. This figure shows the *de-synchronization effect* when a signal path (simulated by the inverter chain) is terminated by a level-sensitive D-type flip-flop (D-FF). Simulations have been carried out assuming that the clock signal goes on-chip through a buffer (two inverters), prior to trigger the D-FF. Fig. 1.a shows circuit functional collapse at 83.33% of  $V_{DDnom}$  ( $\gamma(vdd) = 16.67\%$ ), with CDC = 50%. For Fig. 1(b), when the clock duty cycle is increased to CDC = 80%, circuit functional collapse arises only at 73.33% of  $V_{DDnom}$  ( $\gamma(vdd) = 26.67\%$ ).

As expected, the de-synchronization between the signal and clock lines occurs later in time when CDC is stretched as proposed in this paper. Figs. 1(a) and 1(b) depict the instant of de-synchronization:  $t=31ns$  ( $V_{DD} = 0.98$  volts) and  $t=40ns$  ( $V_{DD} = 0.88$  volts), respectively. Thus, by increasing CDC, the circuit has been rendered more robust to the considered power interconnect noise.

Note that *re-synchronization* may occur for lower values of  $V_{DD}$  (approx. 35ns, Fig. 1.a and 42ns in Fig. 1.b). However, this is not useful, as the correct behavior has been lost first for higher values of  $V_{DD}$  ( $t = 31ns$ , Fig. 1.a).

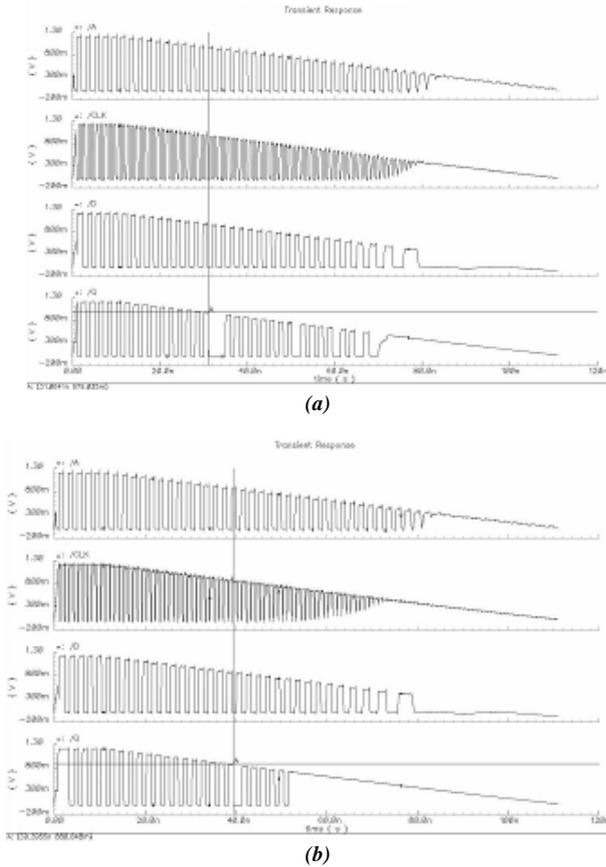
## 3. PROPOSED METHODOLOGY

The proposed methodology is based on the following assumptions:

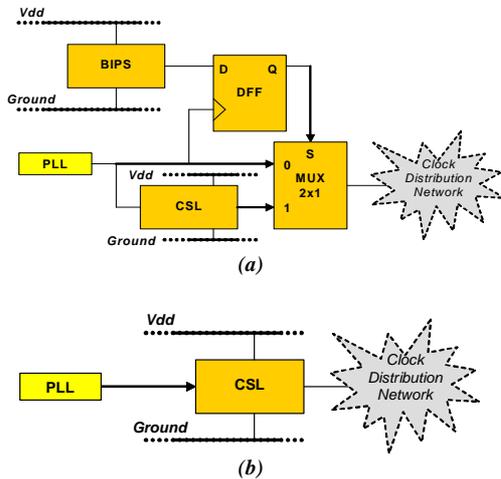
- CDC is generally set at 50% to minimize the *jitter effect* and uncertainties associated with *parameters spread* due to process variations; and to allow a *weighted-time distribution* for circuits designed with both rise- and fall-edge triggered flip-flops.
- The maximum value to which CDC can be stretched is significantly lower than 100% (otherwise, the combinational logic part collapses). In practice, the maximum allowed value is 80%;
- CDC may be temporarily modified for some functional parts of a SoC, but maintained unchanged for the other parts. This does not result in circuit functional error if the overall clock rate is kept unchanged and all the combinational parts of the circuit keep working synchronously.

In order to implement the proposed methodology, on-chip added functionality must be included. We refer as CDCM (or CDC Modulation) system the one that implements this added functionality. Locally, the CDCM system must monitor  $V_{DD}$  variations and trigger CDC variations accordingly. The power grid architecture, and the way its constituting blocks feed the different cores or SoC modules, will determine how many CDC modulators should be inserted. Here, we illustrate the

methodology using one power grid partition, one functional module and one CDCM system. The identification of the critical parts of the SoC that need to be monitored is a non trivial task, but it is beyond this work.



**Fig. 1.** 77-inverter chain terminated by a level-sensitive D-FF, with variable  $V_{DD}$  (voltage ramp) ((a) CDC=50%; (b) CDC=80%).



**Fig. 2.** Basic architectures of the CDCM system: (a) architecture WN (*Work when Needed*); (b) architecture WA (*Work Always*).

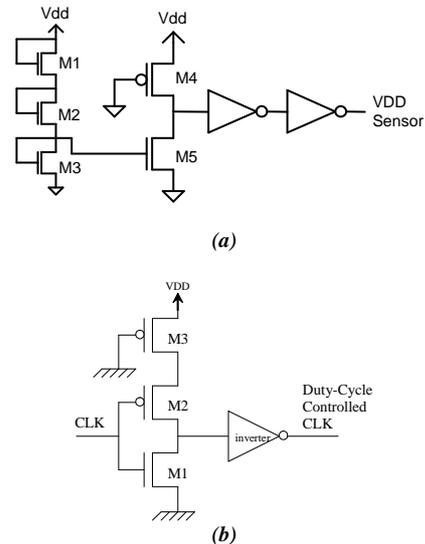
The block diagram depicted in Fig. 2 represents two high-level architectures of the CDCM system. For the WN (*Work when Needed*) architecture, under normal operation the MUX 2X1 is set to the PLL output (or clock generator). When the built-in power supply voltage transient sensor (BIPS) detects  $V_{DD}$  depletion (or ground elevation) below (or above) a given threshold voltage value, the clock stretching logic (CSL) block input of the MUX 2X1 is selected. The use of the D\_FF is to guarantee that the switching from the PLL clock to the CSL clock (and vice-versa) is done without creating glitches at the

output clock (MUX output). The D\_FF should be chosen with the opposite trigger edge that is controlled by the CSL block. The clock signal provided by the CSL block feeds now the circuit logic. This block operates synchronously with the PLL and at the same rate. When the voltage transient in the power-supply grid fades away, the CSL modulator starts gradually reestablishing the original clock duty-cycle (50%), and the BIPS switches the MUX 2X1 selection signal.

The WA (*Work Always*) architecture is a simplified scheme with only the CSL block. The clock signal provided by the CSL block will be always stretched, and thus has a CDC > 50%, even when  $V_{DD}$  has the nominal value. However, it requires less circuitry and adds less delay to the new clock signal.

Several architectures for the BIPS and CSL blocks can be implemented. Here, one possible implementation of these blocks in CMOS technology is shown in Fig. 3. Fig. 3.a describes an implementation example for the ( $V_{DD}$ - $V_{SS}$ ) sensor part in the BIPS block. The main purpose of this block is to detect when a disturbance (in  $V_{DD}$  or  $V_{SS}$ ) is critical for the circuit performance. The underlying concept is as follows: the M1-M3 voltage divider biases the M5 NMOS transistor near cut-off. When  $V_{DD}$ - $V_{SS}$  decreases below a given threshold value, M5 is driven into the cut-off region, and the output of the pseudo-CMOS M4-M5 inverter is driven to  $V_{DD}$ . The two remaining inverters act as a buffer. The output signal ( $V_{DD}$  sensor) is used to generate the MUX2X1 Selection signal (through the flip-flop).

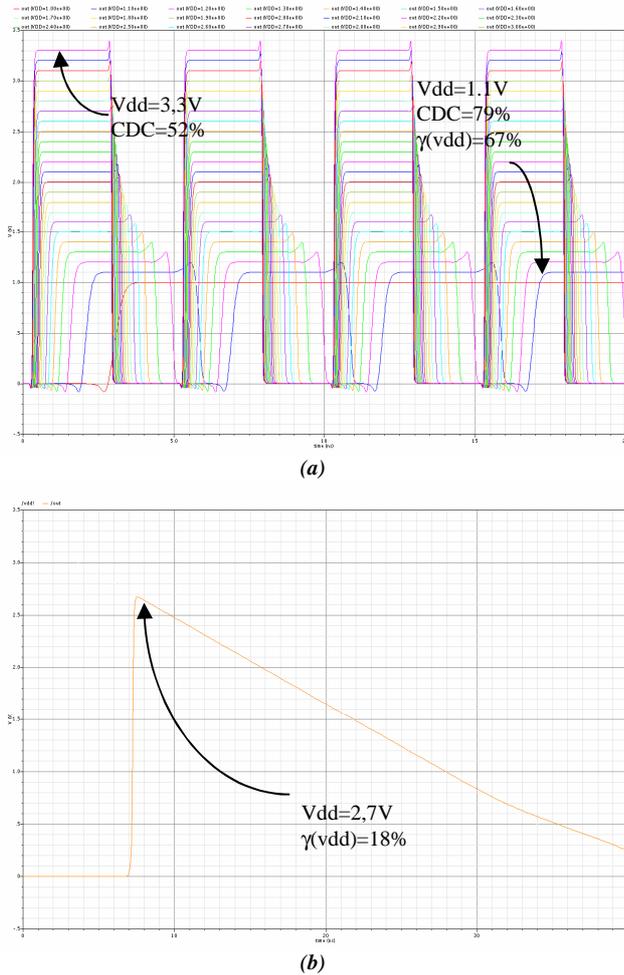
Fig. 3.b describes a possible implementation for the CSL block. The main purpose of this block is to increase CDC, to delay the trigger edge of the clock, allowing the memory cells connected to this modified clock to capture the error free data. In particular, the objective is to delay one of the switching transitions (in this case is the High-to-Low transition). The CDC modulator core is implemented by the M1-M2 inverter. Its pull-up PMOS transistor (M2) has a slow driving capability, due to M3. The result is a new modulated clock signal, with the same clock period, but with enhanced CDC in the presence of a reduction of  $V_{DD}$ . The output signal can be buffered if needed.



**Fig. 3.** Typical architectures for (a) BIPS; (b) CSL blocks.

In Fig. 4 simulation results (AMS 350 nm CMOS technology) show the CSL block capability to stretch CDC proportionally to the depletion of  $V_{DD}$ . For instance, Fig. 4(a) shows the corresponding stretched CDC several  $V_{DD}$  values, from nominal  $V_{DD}$  (3.3 Volt) to  $V_{DD} = 1.0$  Volt. Note that the stretched CDC is always larger than 50% (the original duty-cycle), and lower than 100%. When  $V_{DD}$  is depleted to the

lowest possible value, 1.1V in Figs. 4(a), CDC reaches 79% (these values are for 0.2GHz frequency and for a given transistors dimension). From these values, the CSL does not respond to further  $V_{DD}$  depletions. As consequence, the correct functionality collapses. In Fig. 4(b), the BIPS block shows that for  $V_{DD}$  values smaller than 2.7V (corresponding to a reduction in  $\gamma(vdd)=18\%$ ) the BIPS detects an abnormal  $V_{DD}$  reduction and signalizes its output with a High value.



**Fig. 4.** CSL and BIPS robustness to power supply voltage variations. CSL output as a function of  $V_{DD}$  decrease: 200MHz (a); and BIPS output as a function of  $V_{DD}$  decrease (b).

### 3.1. CDC Stretching Model

A model to accurately quantify *CDC stretching* as a function of  $V_{DD} / V_{SS}$  fluctuations has been developed. The amount of CDC variation, required to tolerate a given  $V_{DD}$  variation, depends on the designer's defined *time slack* value,  $t_{PM}$ . Typically, it may correspond, for nominal values of the circuit parameters, to 8-10% of  $T_{pd}$ , the time period required by the longest critical path to propagate the switching values. When the power supply range,  $(V_{DD} - V_{SS})$  decreases, it erodes  $t_{PM}$ . One part of the speed degradation may be "absorbed" by the reduction in  $t_{PM}$ , without causing a functional error; however, if this occurs, there is no guarantee of correct functionality. Therefore, in our approach, CDC variation, required to tolerate  $V_{DD}$  variation, must preserve the nominal  $t_{PM}$  value. If  $t_{CLK}$  is the clock signal period, then

$$t_{CLK} = T_{pd} + t_{PM} \quad (2)$$

In the absence of CDC modulation,  $CDC=50\%$  and thus

$$CDC = T_{on} / t_{CLK} = 50\% \quad (3)$$

where  $T_{on}$  is the time interval in which the clock signal assumes the '1' logic value. When a  $\Delta V_{DD}$  occurs, a new value of CDC, referred as  $CDC'$ , must be applied. In the worst-case situation (for which nominal  $t_{PM}$  is kept), the clock duty cycle must stretch by the same amount of time required for signal propagation through the logic path. Hence,

$$CDC' = \{T_{on} + T_{pd} \cdot \Delta T_{pd}\} / t_{CLK} \quad (4)$$

and

$$\Delta CDC(V_{DD}) = CDC' - CDC = [T_{pd} / t_{CLK}] \cdot \Delta T_{pd}(V_{DD}) \quad (5)$$

As a consequence, *CDC maximum variation follows  $T_{pd}(V_{DD})$  variation*, for which the authors have previously developed a simple analytical model [12,15]. The typical architectures of the BIPS and CSL blocks, depicted in Fig. 3, allow deriving such variation.

## 4. EXPERIMENTAL RESULTS

Practical experiments based on a FPGA implementation of an ITC'99 benchmark circuit were accomplished to demonstrate the proposed approach effectiveness. They are described hereafter.

This experiment aimed at demonstrating the following assumptions:

- 1) The circuit presents the same behavior (i.e., for the same input sequence, it yields the same output) when operating at a clock duty cycle (CDC) of 50% or 80%.
- 2) Stretching the CDC from 50 to 80% improves circuit robustness to power-bus fluctuations.
- 3) No matter a circuit is composed by one or two types of flip-flops (FF): rising-edge FFs and/or falling-edge FFs, it can be controlled by the proposed clock stretching approach.

For this experiment, we selected as circuit under test (CUT) the *ITC'99 Benchmark (2<sup>nd</sup> release) - B02 Circuit*, which is a FSM that recognizes BCD numbers. Table 1 summarizes the main characteristics of B02.

Number of inputs	Number of outputs	Number of gates	Number of logic levels in the critical (longest) path
3 inputs (1 for data, 1 for clock and 1 for reset)	1	38 logic gates + 3 FFs.	18

**Table 1.** ITC'99 Benchmark Circuit characteristics.

After identifying the critical path, we modified the circuit by adding 1 more input. This extra input allowed controlling the FF at the output of the critical path independently from the other two FFs in the circuit. As it will be seen hereafter, while the clock input to the two FFs that were not in the critical path was fixed to a CDC equal to 50%, the extra clock input that controlled the FF in the critical path moved, when activated, from 50 to 80% in order to increase circuit robustness to power-supply fluctuations.

Since the original B02 had only one output, we added two more outputs. By doing so, we aimed at performing a better discrimination between internal "erroneous" states from "correct" states during the experiment.

The whole experiment was performed in a Spartan3 FPGA mounted on a Xilinx prototyping board. In this FPGA, two different versions of B02 were implemented: in the first version, the circuit critical path was terminated by a rising-edge FF, whereas in the second version, this FF was replaced by another one triggered by the falling-edge segment of the clock signal.



power supply transient sensor (BIPS) block, that is used to monitor abnormal power interconnect activities ( $V_{DD}$  undershoots and ground overshoots) and a clock stretching logic (CSL) block, used to increase CDC proportionally to supply voltage variation. Typical implementations of the BIPS and CSL blocks have been proposed. The range of CDC variation to tolerate  $V_{DD}$  variation has also been identified and discussed.

Experimental results based on FPGA implementations demonstrate the *effectiveness* of the proposed methodology to react to power-supply fluctuations by stretching the clock signal fed to the controlled logic. It is a primary condition that the BIPS and CSL blocks must be, by design, more robust to power supply voltage transients than the logic to be controlled. As shown in Section 3, Fig. 4, by carefully adjusting internal BIPS/CSL blocks parameters it is possible to reach self-tolerance to power-supply oscillation up to 50%. It is worth noting that, as expected for any logic, this tolerance is operating frequency-dependent: as higher as operating clock rates are, less tolerant are the BIPS/CSL blocks to  $V_{DD}$ /GND oscillations.

At present, we are working on the implementation of a 3-stage pipeline microprocessor, whose internal blocks clock signal is controlled by the proposed dynamic clock management methodology.

**Acknowledgement:** The work reported in this paper has been partially funded by CNPq (Science and Technology Foundation, Brazil), FCT (Science and Technology Foundation, Portugal) under the POSC/EEA-ESE/57405 DynaTest Project, and by Xunta de Galicia, Spain, under PGIDIT05TIC30301PR Research Project.

## REFERENCES

- [1] M. Nourani, A. Attarha. "Signal Integrity: Fault Modeling and Testing in High-Speed SoCs". Journal of Electronic Testing: Theory and Applications (JETTA), 2002, pp. 539-554.
- [2] M. Cuvillo, S. Dey, X. Bai, Y. Zhao. "Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects". Int. Conf. on Computer Aided Design, pp. 297-303, Nov. 1999.
- [3] W. Chen, S. Gupta, M. Breuer. "Test Generation in VLSI Circuits for Crosstalk Noise". Int. Test Conf. (ITC'98), pp. 641-650, 1998.
- [4] P. Fang, J. Tao, J. Chen, C. Hu. "Design in Hot Carrier Reliability For High Performance Logic Applications". Proc. IEEE Custom Integrated Circuits Conf., pp. 25.1.1-25.1.7, Oct. 1998.
- [5] Y. Leblebici. "Design Considerations for CMOS Digital Circuits with Improved Hot-Carrier Reliability". IEEE Journal of Solid-State Circuits, vol. 31, no. 7, pp. 1014-1024, July 1996.
- [6] S. Kimothi, U. Nandwani. "Uncertainty Considerations in Compliance-Testing for Electromagnetic Interference". Annual Reliability and Maintainability Symposium, pp. 265-268, 1999.
- [7] F. Vargas, D. C. Lopes, E. Gatti, D. Prestes, D. Lupi. "On the Proposition of an EMI-Based Fault Injection Approach". Proc. 11<sup>th</sup> IEEE Int. On-Line Testing Symposium, pp. 207-208, 2005.
- [8] S. Zhao, K. Roy. "Estimation of Switching Noise on Power Supply Lines in Deep Sub-Micron CMOS Circuits". Proc. Int. Conf. on VLSI Design, pp. 168-173, Jan. 2000.
- [9] F. Vargas, D. C. Lopes, J. Chaves da Silva, D. Barros Jr. "EMI-Induced Soft-Error Rate Estimates for COTS Microprocessor". Proc. 5<sup>th</sup> IEEE Latin American Test Workshop, pp.169-172, 2004.
- [10] H. Chen, L. Wang. "Design for Signal Integrity: The New Paradigm for Deep-Submicron VLSI Design". Proc. Int. Symposium on VLSI Technology, pp. 329-333, June 1997.
- [11] D. Cho, Y. Eo, M. Seung, N. Kim, J. Wee, O. Kown, H. Park. "Interconnect Capacitance, Crosstalk and Signal Delay for 0.35 $\mu$ m CMOS Technology". Proc. Int. Meeting on Electron Devices, pp. 619-622, 1996.
- [12] M. Rodriguez-Irago, J. J. Rodríguez Andina, F. Vargas, M. B. Santos, I.C. Teixeira, J. P. Teixeira. "Dynamic Fault Test and Diagnosis in Digital Systems Using Multiple Clock Schemes and Multi-VDD Test". Proc. 11th IEEE Int. On-Line Testing Symposium (IOLTS), pp. 281-286, 2005.
- [13] A. V. Oppenheim, A. S. Willsky, S. H. Nawab. "Signals and Systems". 2nd Edition, 1997, Prentice Hall Inc.
- [14] K. Bernstein, K.M. Carrig, Ch.M. Durham, P.R. Hansen, D. Hogenmiller, E.J. Nowak, N.J. Rohrer. "High Speed CMOS Design Styles", Kluwer Academic Publishers, 1999.
- [15] D. Barros Júnior, M. Rodriguez-Irago, M.B. Santos, I.C. Teixeira, F. Vargas and J.P. Teixeira, "Fault Modeling and Simulation of Power Supply Voltage Transients in Digital Systems on a Chip", Journal of Electronic Testing, Theory and Application (JETTA), vol.21, pp. 349-363, Kluwer Academic Publishers, August, 2005.
- [16] L. Green. "Simulation, Modeling and Understanding the Importance of Signal Integrity". IEEE Circuit and Devices Magazine, pp. 7-10, Nov. 1999.
- [17] R. Downing, P. Gebler, G. Katopis. "Decoupling Capacitor Effects on Switching Noise". IEEE Trans. on Components, Hybrids and Manufacturing Technology, vol.16, no.5, pp. 484-489, Aug. 1993.
- [18] R. Saleh, D. Overhauser, S. Taylor. "Full-Chip Verification of UDSM Designs" Proc. Int. Conf. on Computer Aided Design (ICCAD), pp. 453-460, 1998.
- [19] A. Kahng, S. Muddu, E. Sarto. "Interconnect Optimization Strategies for High-Performance VLSI Designs". Proc. Int. Conf. on VLSI Design, pp. 464-469, Aug. 1999.
- [20] G. Tellez, M. Sarrafzadeh. "Minimal Buffer Insertion in Clock Trees with Skew and Slew Rate Constraints." IEEE Trans. on CAD, vol. 16, no. 4, pp. 333-342, April 1997.
- [21] Y. Jiang, S. Sapatnekar, C. Bamji and J. Kim. "Interleaving Buffer Insertion and Transistor Resizing into a Single Optimization". IEEE Trans. on VLSI Systems, vol. 6, no. 4, pp. 625-633, Dec. 1998.
- [22] T. Steinecke, W. John, . Koehne, M. Schmidt. "EMC Modeling and Simulation on Chip level". IEEE Int. Symposium on Electromagnetic Compatibility (EMC), 14-16 Aug. 2001.
- [23] S. Ben Dia, M. Ramdani, E. Sicard, editors, "Electromagnetic Compatibility of Integrated Circuits – Techniques for Low Emission and Susceptibility", Springer, 2006.
- [24] E. Sicard, F. Vargas, F. Hernandez, F. Fiori, J. P. Teixeira. "Design and Test on Chip for EMC". IEEE Design and Test of Computers, Issue. Nov/Dec. 2006, pp. 502-503.
- [25] IEC 61.000-4-29 International Standard: Test and Measurement Techniques – Voltage Dips, Short Interruptions and Voltage Variations on D.C. Input Power Port Immunity Tests. [www.iec.ch](http://www.iec.ch).

# SoC Prototyping Environment for EMC-Dependability Measurements

F. Vargas<sup>1</sup>, J. Benfica<sup>1</sup>, M. Moraes<sup>1</sup>, E. Gatti<sup>2</sup>, L. Garcia<sup>2</sup>, D. Lupi<sup>2</sup>, F. Hernandez<sup>3</sup>, J. Semião<sup>4,6</sup>, J. Freijedo<sup>5,6</sup>, M. B. Santos<sup>6</sup>, I. C. Teixeira<sup>6</sup>, J. J. Rodríguez Andina<sup>5</sup>, J. P. Teixeira<sup>6</sup>

<sup>1</sup> Electrical Engineering Dept. Catholic University – PUCRS. Porto Alegre – Brazil.

<sup>2</sup> INTI. Buenos Aires – Argentina.

<sup>3</sup> Universidad ORT / URSEC. Montevideo – Uruguay.

<sup>4</sup> Univ. of Algarve – UAlg / EST. Faro, Portugal.

<sup>5</sup> Univ. of Vigo, Dept. de Tecnología Electrónica. Vigo, Spain.

<sup>6</sup> IST / INESC-ID. Lisboa, Portugal.

[vargas@computer.org](mailto:vargas@computer.org), [lupi@inti.gov.ar](mailto:lupi@inti.gov.ar), [paulo.teixeira@ist.utl.pt](mailto:paulo.teixeira@ist.utl.pt)

## Abstract

We present a configurable standard platform devoted to measure the dependability of systems-on-chip (SoC) when operating in electromagnetic (EM) environments. The platform is composed of two boards compliant with the 62.132-2 and 62.132-4 IEC Std Parts, being conceived for radiated and conducted measurements, respectively. The SoC under test can be laid down on two types of ICs: two Spartan 500E FPGAs and an 8051-like microcontroller. Two practical experiments have been carried out. The first experiment was performed on a SoC designed to be robust to EM interference (EMI), according to the 62.132-2 IEC Std Part for radiated interference. The second experiment was carried out on a modified (noise-robust) version of the ISCAS'99 benchmark circuit B02, in order to validate a new adaptive clock signal control. This experiment was conducted according to the 61000-4-29 IEC Std part. The obtained results demonstrate the utility and benefits from using the proposed platform to estimate in an early stage of the design process the behavior of SoC operating in EM environment.

**Keywords:** System-on-Chip (SoC); Hardware and software early-reliability estimation; electromagnetic immunity measurement; IEC International Standard.

## 1. Introduction

The roadmap for standardization of immunity measurement methods has reached a high degree of success with the IEC 62.132 proposal [1]. Recently (2006), some extensions have been proposed through research publications, which aim at extending the *Bulk Current Injection Method* and the *Direct Power Injection Method* to 10 GHz [2].

At the same time, the technology scale down offers the possibility to design more complex integrated circuits (ICs) [3], with tenths of millions of transistors placed and routed in between more than one thousand I/O pins. The supply voltage is continuously decreasing, reaching less than 1 volt for the IC core, and less than 2 volts for the periphery and I/O pads. This scenario reduces noise

margins and increases circuit susceptibility to external electromagnetic (EM) waves [4,5].

There has been an increased demand for EMC models applicable to integrated circuits and hardware/software-based prototyping vehicles, in order to conduct compatibility analysis early in the system-on-chip (SoC) design process. It is at this point that we introduce our work. We propose hereafter an innovative (configurable) platform for measuring the EM susceptibility of SoCs prototyped during the design phase. Depending on the designer interest and the target application, the prototype immunity can be measured with respect to the hardware and/or the software parts of the SoC. In the best of our knowledge, this is the first time that this kind of platform is reported.

The remainder of this paper is divided as follows: *Section 2* presents the proposed platform. *Section 3* describes the case-studies and the practical experiments that have been carried out to demonstrate the utility and benefits from using the proposed platform. Finally, *Section 4* summarizes the main points of this work.

## 2. The Proposed Platform

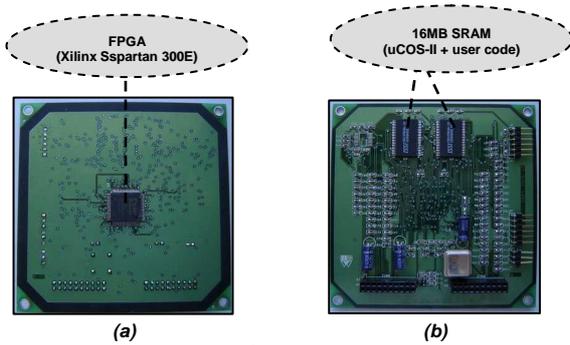
The proposed environment is composed of two boards for radiated and conducted electromagnetic immunity measurements. With this infrastructure, multiple embedded microprocessors like MicroBlaze<sup>1</sup> and PowerPC 601 running uCLinux or uCOS-II<sup>2</sup> [6,7] can be prototyped. Additionally to the hardware parts, several implementations of VHDL-described embedded intellectual property (IP) cores and C-code programs can also have their immunity response measured and compared to each other in order to leverage the final dependability level for the SoC on the design.

Fig. 1a presents a photograph of first board (Board I), designed and fabricated according the IEC 61.132-2

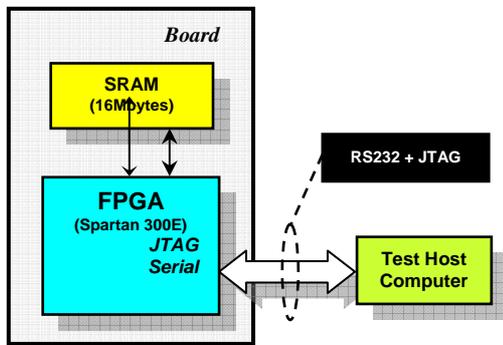
<sup>1</sup> MicroBlaze™ is a true 32-bit soft RISC processor optimized for use in Xilinx's FPGA architectures. The processor's main memory interface conforms to the IBM CoreConnect specification for the On-Chip Peripheral Bus (OPB).

<sup>2</sup> MicroC/OS-II has been certified to RTCA DO-178B Level A for use in avionics systems where failure could result in catastrophic loss of the aircraft, and approved for use in FDA Class III medical devices where failure could result in loss of life for the patient or clinician.

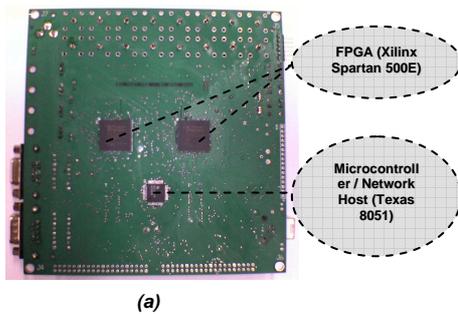
standard for radiated electromagnetic (EM) immunity measurement. The “test side” of this board is shown in Fig. 1a, which contains the IC under test (Xilinx FPGA, Spartan 300E). This side contains also the board ground layer. Fig. 1b shows “other side” of the board, which contains the remaining logic (SRAM memories, clock generator and voltage regulators, among other components). This board side also lays down the  $V_{DD}$  distribution network for the system. The two inner layers of the board are used for signal propagation. Around the board, it can also be observed a “ground ring” used to attach the onboard system ground with the TEM cell ground into a unique reference. Fig. 2 presents the basic blocks composing Board I.



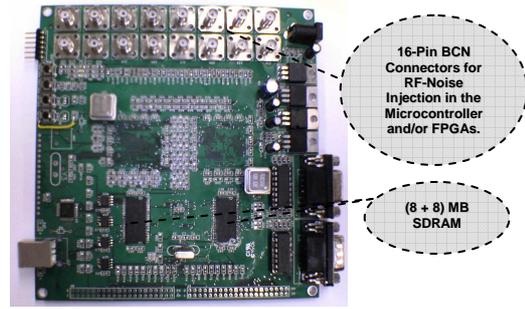
**Fig. 1.** Board I: 10x10cm<sup>2</sup> IEC 62.132-2 std compliant board comprised by four-layers: Gnd (top) / signal / signal / Vdd (bottom). (a) Top view; (b) Bottom view.



**Fig. 2.** Basic blocks of Board I.



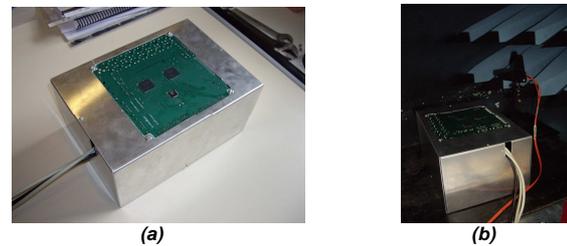
(a)



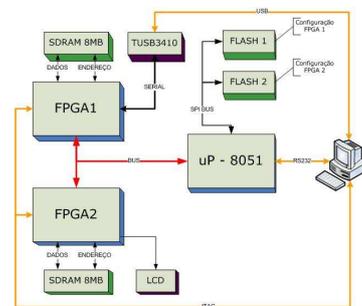
(b)

**Fig. 3.** Board II: IEC 62.132-4 std compliant board comprised of 6 layers for conducted immunity measurement. Views: (a) Top; (b) Bottom.

The second board contains two Xilinx Spartan 500E FPGAs, a Texas 8051-like microcontroller, 16MBytes of SDRAM, and 8MBytes of serial Flash memory, among other glue logic required for communication with the test host computer (see Fig. 3 for details). In this figure, side (a) contains the components under test, i.e., the parts whose EM measurements can to be performed; whereas side (b) contains the remainder of the logic (processor bus, memories, crystals, connectors for radio-frequency (RF) noise injection, connectors and logic for external environment communication-support, among other devices). Fig. 4 depicts the shielding box for radiated testing. The remainder logic of the board is protected inside the box, while the devices under test (side (a)) are placed externally, to be exposed to EM fields. Fig. 5 presents the basic blocks of Board II.



**Fig. 4.** Shielding box for radiated test: (a) General view; (b) Inside the GTEM Cell.



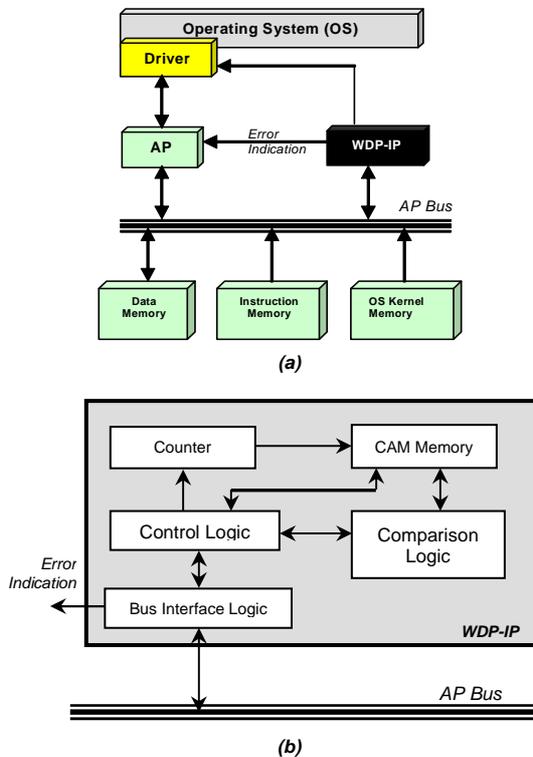
**Fig. 5.** Basic blocks of Board II.

### 3. Experimental Results

This section presents the two case-studies and the practical experiments that have been carried out to demonstrate the utility and benefits from using the proposed platform.

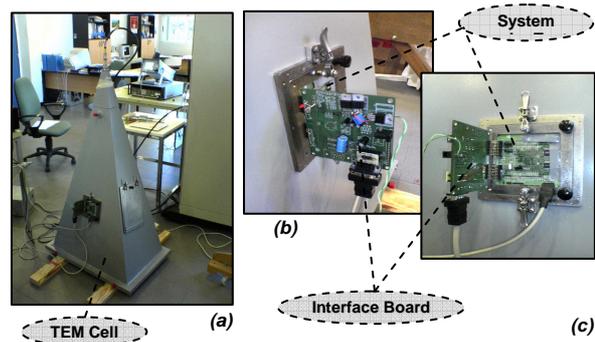
#### 3.1 First Case-Study

We conducted an experiment aiming at analyzing the radiated electromagnetic sensitivity of a watch-dog processor intellectual property (WDP-IP) core [8,9] designed to monitor the Xilinx MicroBlaze soft core processor running under the uCOS-II operating system control. This system is said to be the “*Test Vehicle*”, prototyped in Board I (Fig. 6a presents a general view of the system, whereas Fig. 6b depicts details of the WDP-IP core basic blocks). The whole SoC was described in VHDL language.



**Fig. 6.** SoC prototyped in Board I: (a) General architecture; (b) WDP-IP basic blocks.

To perform the experiment, we implemented three user tasks running in the processor under the time-shared basis: a *random prime numbers generator (PNG)*, a *bubble sort to reorder a matrix (BS)*, and a *digital filter (DT)*. This experiment was based on the *International IEC 62.132 Standard Part 2: Measurement of Radiated Immunity – TEM Cell Method*.



**Fig. 7.** Test environment showing TEM Cell and test vehicle prototyped in Board I. (a) General view; (b) and (c) Closer views detailing the test vehicle with the FPGA board side turned into the chamber.

Fig. 7 depicts the TEM-Cell and the test setup at the *Instituto Nacional de Tecnologia Industrial – INTI*, Buenos Aires, where the experiment was conducted. Dealing to minimize test procedure complexity, we arbitrarily decided to stop the experiment when we succeeded to obtain 330 measurements of system failure. This resulted in a total time of system exposition to radiated EMI of approximated 40 hours. The test conditions were as follows:

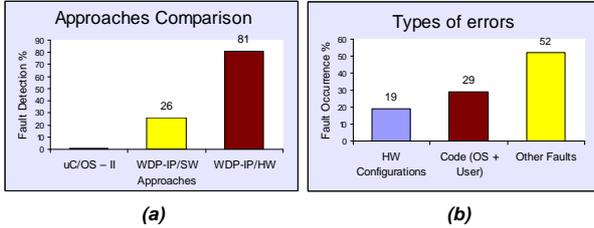
- a) EM field range: from 10 to 220V/m;
- b) Measured frequency range: from 150KHz to 3GHz (extended IEC 62.132-2);
- c) Signal Modulation Format: three different types were used: 80%, Without Modulation, and Pulsed Signal.

The Interface Board observed in Fig. 7b and 7c is used to perform communication between Board I and the external computer (test host). The Interface Board is responsible, for instance, for the RS232 serial and for the JTAG communications between the test engineer and Board I during measurements procedure.

It is worth noting that we have also implemented a second version of the WDP-IP. This version presented the same functionalities as the WDP-IP in hardware, but it was implemented purely in software (C-ansi) and compiled with the kernel of the uCOS-II OS.

Additionally, the fault detection capability of the proposed I-IP was compared against the uCOS-II OS native fault detection structures existing in its own kernel. In summary, test measurements were carried out on three different system configurations: (a) microprocessor + WDP-IP in hardware (VHDL); (b) microprocessor + WDP-IP in software (C); and (c) microprocessor + uCOS-II OS native fault detection structures (original uCOS-II OS kernel). Fig. 8a summarizes the measurements for this experiment. Note also that Fig.8b presents results related with the types of faults occurring in Board I: 19% of the faults were related with hardware configuration of the FPGA; 29%

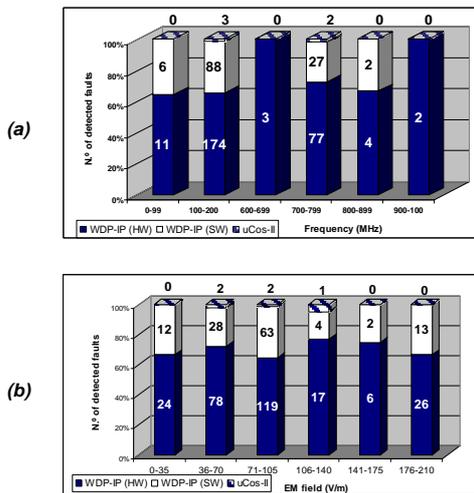
of the faults resulted from errors occurring during data handling or control-flow execution of the user code or during the execution of the operating system kernel. Finally, 52% of the faults occurred due to communication malfunction between the microprocessor under test and the Interface Board.



**Fig. 8.** Fault detection capability measured for the WDP-IP during IEC 62.132-2 test session. (a) Approaches comparison; (b) Classification of observed errors.

Fig. 9 presents the occurrence of faults as a function of the “modulated EM signal frequency” (Fig. 9a) and the “EM field incident on the board under test” (Fig. 9b). For instance, in the frequency range of 100-200MHz (Fig. 9a), the system under test presented 265 faults: 174 (65.7%) were detected by the WDP-IP in hardware, 88 (33.2%) were detected by the WDP-IP in software, and 3 (1.1%) were detected by the native fault detection structures existing in the kernel of the uCOS-II OS.

It is worth noting that even though measurements were performed up to an extended frequency range: from 150KHz to 3GHz (the std. IEC 62.132-2 rules measurements up to 1GHz), Fig. 9a depicts results only up to 1GHz because no fault occurrence was observed above this level.



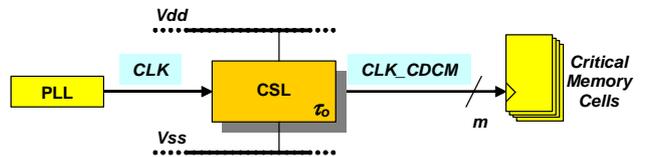
**Fig. 9.** Fault occurrence as a function of the modulated EM signal frequency (a) and the EM field incident on the board under test (b).

After analyzing the measurement results, we concluded that:

- The uCOS-II kernel native fault detection was very low (approx. 1%) because of the embedded structures were able to detect **only** those faults that resulted in an increase of the time allocated by the OS for the processor to run the task slices. Note that those faults that reduce the time allocated by the OS (for instance, resulting in a procedure aborting) are **not** detected by the kernel native structures.
- The WDP-IP software version was capable to detect (in addition to those faults that resulted in an increase/decrease of the time allocated by the OS for the processor to run the task slices) most of the faults that affected user memory elements (FFs and SRAM). However, it **failed** to signal most of those faults that changed the FPGA **configuration bitstream**. Several of these faults yielded system crash (processor should be reinitialized).
- The WDP-IP hardware version was capable to detect most of the faults that affected not only **user memory elements**, but also those that **corrupted FPGA configuration logic**. In addition to this, the WDP-IP also detected those faults that **corrupted (by increasing or reducing) the task slice execution time frames** defined by the OS.

### 3.2 Second Case-Study

For the second case-study, we conducted an experiment aiming at analyzing the capability of a new approach to enhance SoC signal integrity with respect to power/ground voltage transients, without degrading its performance [4,11,12]. The underlying principle of such approach is to dynamically adapt the *clock duty-cycle* (CDC) according to the signal propagation delay through the logic whose power supply voltage is being disturbed. Fig. 10 depicts the architecture; the *Clock Stretching Logic* (CSL) block performs power-supply disturbance monitoring and CDC stretching, using a simple circuit and introducing a limited clock delay,  $\tau_o$ , (referred to as the *intrinsic CDC Modulation: i.e., CDCM delay*). The modulated clock signal adds a minimum stretch at nominal  $V_{DD}$  (CDC slightly greater than 50%). When  $V_{DD}$  decreases, the CSL block stretches CDC according to  $V_{DD}$  reduction. In this case, the time slack is increased by the amount  $\tau_o$ , i.e.,  $t'_s = t_s + \tau_o (V_{DD})$ .



**Fig. 10.** Basic architecture of the CDC modulation system.

When the voltage transient in the power-supply grid fades away, the CSL block starts gradually reestablishing the original clock duty-cycle (i.e., slightly greater than 50%).

Therefore, this experiment aims at demonstrating the following assumptions:

- 1) The circuit presents the same behavior (i.e., for the same input sequence, it yields the same output) when operating at a clock duty cycle (CDC) of 50% or 80%.
- 2) Stretching the CDC from 50 to 80% improves circuit robustness to power-bus fluctuations.
- 3) No matter a circuit is composed by one or two types of flip-flops (FF): rising-edge FFs and/or falling-edge FFs, it can be controlled by the proposed clock stretching approach.

For this experiment, we selected as circuit under test (CUT) the *ITC'99 Benchmark (2<sup>nd</sup> release) – B02 Circuit*, which is a FSM that recognizes BCD numbers. Table 1 summarizes the main characteristics of B02.

Number of inputs	Number of outputs	Number of gates	Number of logic levels in the critical (longest) path
3 inputs (1 for data, 1 for clock and 1 for reset)	1	38 logic gates + 3 FFs.	18

**Table 1.** ITC'99 Benchmark Circuit characteristics.

The whole experiment was performed in Board II. In one of the FPGAs of this board, two different versions of B02 were implemented: in the first version, the circuit critical path was terminated by a rising-edge FF, whereas in the second version, this FF was replaced by another one triggered by the falling-edge segment of the clock signal. In the second FPGA of Board II, we implemented the logic function (i.e., the CSL block) that controls the CDC stretching.

After identifying the B02 critical path, we modified the circuit by adding 1 more input. The goal of this extra input was to allow the CSL block implemented in the second FPGA of Board II to control the FF at the output of the critical path independently from the other two FFs in B02. As it will be seen hereafter, while the clock input to the two FFs that were not in the critical path was fixed to a CDC equal to 50%, the extra clock input that controlled the FF in the critical path moved, when activated, from 50 to 80% in order to render the circuit more robust to power-supply fluctuations.

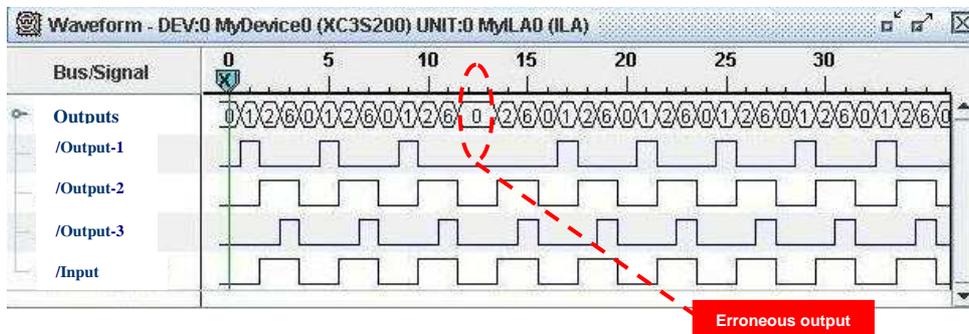
Since the original B02 had only one output, we added two more outputs. By doing so, we aimed at performing a better discrimination between internal “erroneous” states from “correct” states during the experiment.

The experiment was performed by applying voltage dips to the FPGA  $V_{DD}$  pins according to the IEC 61.000-4-29 International Standard [10] (by means of the BCN connectors shown in Fig. 3b). Then, we applied voltage dips of 36% to the 5V pins feeding the periphery (I/O pads) and 42% to the 1.2V pins feeding the core logic of the FPGA. Based on this condition, the resulting  $V_{DD}$  fluctuated from 5 to 3.2V (for the periphery) and from 1.2 to 0.7V (for the logic core). We tried also to apply deeper voltage dips, but the FPGA IC lost configuration and crashed down.

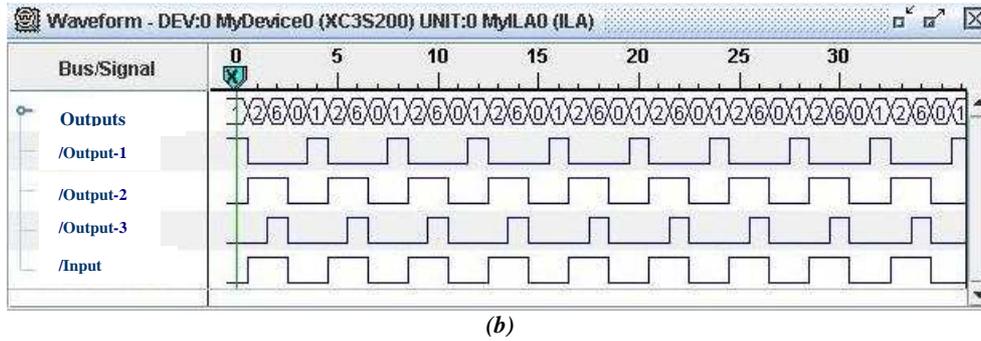
Figs. 11 and 12 display the measured output values for B02. Fig. 13 presents results for the first version of B02 (with the circuit critical path terminated by a rising-edge FF), whereas Fig. 12 summarizes results for the second B02 version (with the critical path terminated by a falling-edge FF).

With the purpose of illustrating this example, the expected (correct) B02 output sequence is: “0126”. This sequence is repeated till the moment when the circuit is reset. Then, another valid sequence is randomly started. Among the possible (valid) sequences, they are namely: “0126”, “0156” or “1234”.

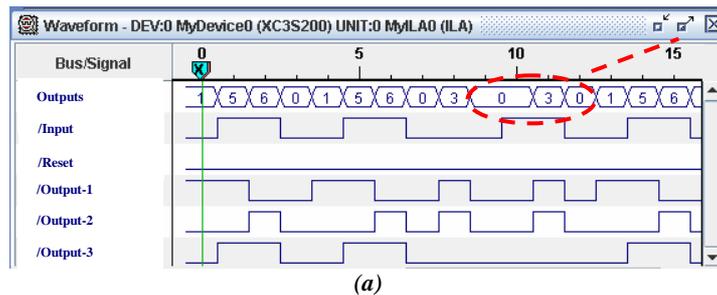
Note also that the clock stretching logic that controls the critical path FF (Figs. 11b and 12b) performs in two different ways: the **clock signal that controls the rising-edge FF** of Fig. 11b has its **low-level segment** stretched from 50 to 80% (i.e., delaying the rising edge of the clock signal); on the other hand, the **clock signal that controls the falling-edge FF** (Fig. 12b) has its **high-level segment** stretched from 50 to 80% (thus, delaying the clock signal falling-edge). Fig. 13 summarizes the original and the stretched clock signals for both cases.



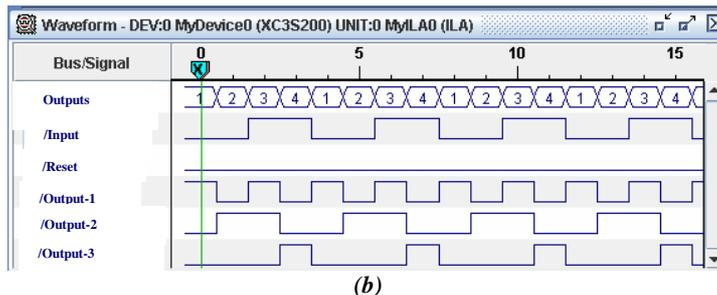
(a)



**Fig. 11.** CUT output for all memory elements implemented with rising-edge FF: **(a)** 50% CDC, erroneous output; **(b)** 80% CDC, correct output.

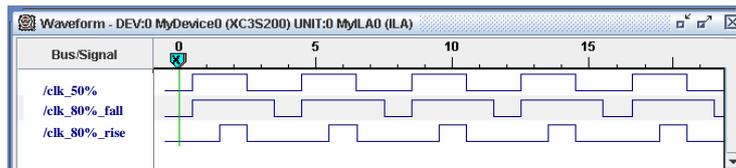


**(a)**



**(b)**

**Fig. 12.** B02 output for all memory elements implemented with rising-edge FF, *except* the FF in the critical path, which was implemented with falling-edge FF: **(a)** 50% CDC, erroneous output; **(b)** 80% CDC, correct output.



**Fig. 13.** Summary of the original and the stretched clock waveforms: Original clock (**clk\_50%**); Stretched clock activating the falling-edge FF in the critical path (**clk\_80%\_fall**); Stretched clock activating the rising-edge FF in the critical path (**clk\_80%\_rise**).

#### 4. Conclusions

There has been an increased demand for hardware/software-based prototyping vehicles in order to conduct compatibility analysis early in the system-on-chip (SoC) design process. In order to address this point, we presented a configurable standard environment for electromagnetic (EM) immunity measurement of prototype system-on-chip (SoC). In the best of our knowledge, this is the first time that this kind of platform is reported.

The environment is composed of two boards compliant with the 62.132-2 and 62.132-4 IEC Std Parts, being conceived for radiated and conducted measurements, respectively. The SoC under test can be prototyped on two types of ICs: two FPGAs and a microcontroller.

The underlying advantages of the proposed test platform rely on: **(a)** reduction of SoC design cost and time due to early-estimation of system behavior in the presence of EM noise according to recognized standards, and **(b)** allowance of measurements for

hardware (IP cores) as well as for software (user-code and operating system-kernel).

Practical experiments have been carried out. The obtained results demonstrate the utility and benefits from using the proposed platform to estimate the behavior of embedded systems operating in EM environment.

**Acknowledgement:** The work reported in this paper has been partially funded by CNPq (Science and Technology Foundation, Brazil).

### References

- [1] [www.iec.ch](http://www.iec.ch) (last access on 30/08/2007).
- [2] E. Sicard, F. Vargas, F. Hernandez, F. Fiori, J. P. Teixeira. "Design and Test on Chip for EMC". IEEE Design and Test of Computers, Issue. Nov/Dec. 2006, pp. 502-503.
- [3] Bernardi, P.; Veiras Bolzani; L. M.; Rebaudengo, M.; Sonza Reorda, M.; Vargas, F. L.; Violante, M. A New Hybrid Fault Detection Technique for Systems-on-a-Chip. IEEE Transactions on Computers, Feb. 2006, Vol. 55, No. 2. pp. 185-198.
- [4] Semião, J.; Rodriguez-Irago, M.; Piccoli, L.; Vargas, F.; Santos, M. B.; Teixeira, I. C.; Andina, J. J. R.; Teixeira, J. P. Digital Circuit Signal Integrity Enhancement by Monitoring Power Grid Activity. 8th IEEE Latin American Test Workshop (LATW'07), Cuzco, Peru, 11-14 March 2007.
- [5] Steinecke, T. Experimental Characterization of Switching Noise and Signal Integrity in Deep Submicron Integrated Circuits. IEEE International Symposium on Electromagnetic Compatibility, Washington - DC, USA, 21-25 August 2000. pp. 107-112.
- [6] Micrium – Empowering Embedded Systems. [www.ucos-ii.com](http://www.ucos-ii.com) (last access on 25/08/2007).
- [7] Validated Software Corporation. <http://www.validatedsoftware.com> (last access on 25/08/2007).
- [8] Vargas, F.; Piccoli, L.; Benfica, J.; Alecrim Jr., A.; Moraes, M. Time-Sensitive Control-Flow Checking for Multitask Operating System-Based SoCs. 13<sup>th</sup> IEEE International On-Line Testing Symposium (IOLTS'07), Crete, Greece, 9-11 July 2007.
- [9] Vargas, F.; Piccoli, L.; Benfica, J.; Alecrim Jr., A.; Moraes, M. Summarizing a Time-Sensitive Control-Flow Checking Monitoring For Multitask SoCs. IEEE International Conference on Field Programmable Technology (FPT'06), Bangkok, Thailand, 13-15 December 2006. pp. 249-252.
- [10] IEC 61.000-4-29 International Standard: Test and Measurement Techniques – Voltage Dips, Short Interruptions and Voltage Variations on D.C. Input Power Port Immunity Tests. [www.iec.ch](http://www.iec.ch).
- [11] M. Rodriguez-Irago, J. J. Rodríguez Andina, F. Vargas, M. B. Santos, I.C Teixeira, J. P. Teixeira. Dynamic Fault Test and Diagnosis in Digital Systems Using Multiple Clock Schemes and Multi-VDD Test. 11th IEEE Int. On-Line Testing Symposium (IOLTS), pp. 281-286, 2005.
- [12] Semião, J.; Rodriguez-Irago, M.; Piccoli, L.; Vargas, F.; Santos, M. B.; Teixeira, I. C.; Andina, J. J. R.; Teixeira, J. P. Signal Integrity Enhancement in Digital Circuits in the Presence of Power Grid Activity. IEEE Design and Test of Computers. (Scheduled for publication on Nov/Dec Issue, 2007.)

# Power-Supply Instability Aware Clock Signal Modulation for Digital Integrated Circuits

J. Semião<sup>2</sup>, J. Freijedo<sup>1,4</sup>, M. Moraes<sup>3</sup>, M. Mallmann<sup>3</sup>, C. Antunes<sup>3</sup>, J. Benfica<sup>3</sup>, F. Vargas<sup>3</sup>, M. Santos<sup>1</sup>, I. C. Teixeira<sup>1</sup>, J. J. Rodríguez Andina<sup>4</sup>, J. P. Teixeira<sup>1</sup>, D. Lupi<sup>5</sup>, E. Gatti<sup>5</sup>, L. Garcia<sup>5</sup>, F. Hernandez<sup>6</sup>

<sup>1</sup> IST / INESC-ID Lisboa, R. Alves Redol, 9, 3º, 1000-029. Lisboa, Portugal

<sup>2</sup> Univ. of Algarve – UAlg / EST, Campus da Penha, 8005-139. Faro, Portugal

<sup>3</sup> PUCRS - Electrical Engineering Dept., Av. Ipiranga, 6681, 90619-900. Porto Alegre, Brazil

<sup>4</sup> Univ. of Vigo, Dept. de Tecnología Electrónica, Campus Universitario, 36310. Vigo, Spain

<sup>5</sup> Inst. Nacional de Tec. Industrial (INTI), Av. General Paz, 5445 (San Martín). Buenos Aires, Argentina

<sup>6</sup> Universidad ORT, Calle Cuareim, 1451. Montevideo, Uruguay

[vargas@computer.org](mailto:vargas@computer.org), [paulo.teixeira@ist.utl.pt](mailto:paulo.teixeira@ist.utl.pt), [fernandez@uni.ort.edu.uy](mailto:fernandez@uni.ort.edu.uy)

**Abstract** - As IC technology scales down, interconnect issues are becoming one of the major concerns of gigahertz System-on-Chip (SoC) design. Voltage distortion (power supply noise) and delay violations (signal and clock skews) dramatically contribute to signal integrity loss. As a consequence, performance degradation, reliability problems and ultimately, functional error occur. In this paper<sup>1</sup>, we propose a new methodology to enhance SoC signal integrity with respect to power/ground voltage transients, without degrading its performance. The underlying principle of the proposed methodology is to dynamically adapt the clock duty-cycle (CDC) according to the signal propagation delay through the logic whose power supply voltage is being disturbed. The methodology is based on a clock stretching logic (CSL) block, which monitors abnormal power grid activity and increases clock duty-cycle accordingly. Moreover, a model to accurately quantify CDC stretching as a function of  $V_{DD}/Gnd$  fluctuations is proposed. Practical experiments based on the implementation of a 32-bit pipeline processor in a FPGA IC were performed and demonstrate the circuit robustness enhancement to power line fluctuations while maintaining at-speed clock rate.

## 1. INTRODUCTION

Signal integrity is the ability of a signal to generate correct responses in a circuit. It generally includes all effects that cause a circuit malfunction due to the distortion of the signal waveform [1]. According to this definition, a signal with good integrity presents: (i) voltage values at required levels and (ii) level transitions at required times. For instance, an input signal to a flip-flop with *good signal integrity* arrives early enough to guarantee the setup and hold times and it does not have spikes that may cause undesired logic transitions.

Various signal integrity problems have been studied for high-speed gigahertz nanometer System-on-Chip (SoC). The

most important ones are: (a) *crosstalk* (signal distortion due to cross coupling effects between signals) [2,3]; (b) *overshoot/undershoot* (momentarily signal rising/decreasing above/below the power supply voltage ( $V_{DD}$ ) and ground ( $V_{SS}$ ) lines [4,5]; (c) *reflection* (echoing back a portion of a signal, at high-frequency circuits, where interconnections behave as transmission lines); (d) *electromagnetic interference – EMI* (resulting from antenna properties) [6,7,24]; (e) *power-supply noise* [8,9]; and (f) *signal skew* (delay in arrival time to different receivers) [10-12].

In this work, we are concerned with the power supply disturbances, regardless of their origin, and their impact on digital circuit performance. Both  $V_{DD}$  depletion and ground elevation cause *propagation delay through signal path increase*. It is assumed that well-known techniques [13,14] are used to limit the impact of power supply disturbances on clock generation and distribution performance. In this scenario, if a signal propagation delay is large enough, it will induce a *de-synchronization effect* due to the increased difference between the critical path propagation delay and the clock distribution network delay. Hence, e.g., 10% power supply voltage fluctuations may translate in more than 10% timing inaccuracy, causing a functional error [12]. Moreover, noise margins are reduced in the presence of power-supply voltage transients. This problem gets worse with technology scaling down. Interestingly, research results on delay fault detection and diagnosis can be reused to enhance delay fault *tolerance* (and thus signal integrity) to power supply voltage transients [15].

Many design and fabrication solutions have been proposed to enhance signal integrity in the presence of power grid activity. However, none of them guarantees perfect solution for the de-synchronization effect described above. In addition to the uncertainties that result from using these solutions, their use often requires the knowledge of the *instantaneous maximum power supply current*,  $i_{DD}(t)$ , for delay analysis (as opposed to power dissipation that relates to average current). The estimation of this parameter for complex designs is frequently a burden task, since it is strongly dependent on input-pattern *sequence*, even if the output is not switching. Some of the proposed solutions include: 3-D layout modeling and parasitic extraction [16]; accurate RLC

<sup>1</sup> The work reported in this paper has been partially funded by CNPq (Science and Technology Foundation, Brazil), FCT (Science and Technology Foundation, Portugal) under the POSC/EEA-ESE/57405 DynaTest Project, and by Xunta de Galicia, Spain, under PGIDIT05TIC30301PR Research Project.

simulation of on-chip power grid [10] using decoupling capacitors [8,17] and to improve resistive voltage drop (IR-drop) [10,18]; insertion of buffers on the grid [16]; and shielding wires (e.g., grounding every other line) [19]. Buffer insertion and transistor resizing methods have also been proposed [20,21] to achieve better power-delay and area-delay trade-offs. Additionally, self-test methodologies and on-chip probes to monitor intra-packaging EM-emission activities [22] have been developed to test signal integrity in high-speed SoC. Finally, other techniques have been discussed in the literature [24]. For example, the reduction of the maximum distance between supply pins and the circuit supply connections [23]. This can be achieved, for instance, by implementing *finger-shaped power distribution networks* to feed  $V_{DD}$  and ground signals throughout the internal blocks of the IC.

Undesired voltage transients in power/ground interconnects may be constrained, but not eliminated. In order to guarantee correct timing performance in their presence, the ultimate solution is to reduce the clock signal frequency. For some applications, such lost in performance cannot be tolerated. However, in a significant set of applications, a new solution can be applied, which does not require clock frequency reduction. The underlying principle of the proposed methodology is to dynamically adapt the clock duty-cycle (CDC) according to the signal propagation delay through the logic whose power supply voltage is being disturbed. The functionality to implement the underlying principle is as follows: (1) power supply voltage monitoring and (2) when voltage variation exceeds a user's defined threshold, CDC modulation. Hence, the correspondent architecture contains a *clock stretching logic* (CSL) block, to monitor power grid activity ( $V_{DD}$  undershoots and/or ground overshoots) and to enhance the clock duty-cycle provided by the phase-locked loop (PLL) block to the controlled logic when required.

Therefore, the main advantage of the proposed methodology is to render the circuit more robust to power line fluctuations by maintaining *at-speed clock rate*. As such fluctuations are often due to circuit operation, they tend to cause local disturbances in the power grid. Hence, it is a waste to reduce the clock frequency in the overall system. The proposed methodology locally adapts the CDC without loosing circuit performance.

## 2. CLOCK DUTY CYCLE MODULATION

As referred, lowering the power supply voltage,  $V_{DD}$ , enhances the propagation delay of signal paths. Hence, if the observation pace is kept invariant (at-speed circuit operation), lowering  $V_{DD}$  while maintaining nominal clock period,  $t_{CLK}$ , reduces circuit noise and time margins, which will, ultimately, induce system functional errors. The time margin is characterized by the *time slack*,  $t_{PM}$ . In this work, the time slack is defined as the time interval difference between the clock period and the time interval associated with the time response of the critical path in the slowest combinational module between registers.

Let us also consider another phenomenon, the *ground bounce*, also known as *simultaneous switching noise*. This phenomenon occurs when internal nodes of a logic device change state. When this happens, the charge remaining in the internal nodes ( $C_L$ ) is drained through the ground grid. The currents associated with this charge removal are added to the switching currents flowing from  $V_{DD}$  to the ground interconnects due to the simultaneous switching of the N- and

P-networks. Thus, the resulting total current flowing through the ground grid induces a local voltage variation, namely, the ground bounce,  $V_{GB}$ . Worst-case conditions exist when a large number of nodes simultaneously switch. In this case, the resulting switching currents from each node capacitance are added together. The total current flowing through the ground lead reduces noise margins and increases the probability of system malfunction.

For the sake of simplicity, we normalize the *size of the disturbance* on  $V_{DD}/Gnd$  interconnects voltage using a *gamma parameter*:

$$\gamma(vdd) = \frac{\Delta V_{DD}}{V_{DDnom}} \quad \text{or} \quad \gamma(gnd) = \frac{\Delta GND}{V_{DDnom}} \quad (1)$$

where:  $\Delta V_{DD}$  is the difference between the nominal  $V_{DD}$  ( $V_{DDnom}$ ) and the depleted  $V_{DD}$ , and  $\Delta GND = \Delta V_{SS}$  is the difference between the elevated ground and  $V_{SS} = 0$  Volt.

## 3. PROPOSED METHODOLOGY

The proposed methodology is based on the following assumptions:

- a) CDC is generally set at 50% to minimize the *jitter effect* and uncertainties associated with *parameters spread* due to process variations; and to allow a *weighted-time distribution* for circuits designed with both rise- and fall-edge triggered flip-flops.
- b) The maximum value to which CDC can be stretched is significantly lower than 100% (otherwise, the combinational logic part collapses). In practice, the maximum allowed value is 80%;
- c) CDC may be temporarily modified for some functional parts of a SoC, but maintained unchanged for the other parts. This does not result in circuit functional error if the overall clock rate is kept unchanged and all the combinational parts of the circuit keep working synchronously.

Consider a synchronous IP core, with different modules, each fed by a subsection of the power grid infrastructure (Fig. 1). If circuit operation induces  $V_{DD}$  variations in module  $i$ , its timing performance is distorted, typically delayed. In order to allow the combinational blocks to finish their job, an additional time has to be given to the signals switching in critical paths. Hence, the underlying idea is to dynamically delay their capture, by the *critical memory cells*, in the presence of  $V_{DD}$  variation. Therefore, for a limited subset of the module's registers ( $m \ll k$ ), a CDC modulation block must be added to accommodate such delay.

In a synchronous circuit, CDC is generally set at 50% to minimize *jitter* and *process variations*, and to allow a *weighted-time distribution* for circuits designed with both rise- and fall-edge triggered flip-flops (FF). In order to prevent logic errors, we assume that CDC may be stretched up to 80%.

We refer as CDCM (or CDC Modulation) module the one that implements this added functionality. Locally, the CDCM module monitors  $V_{DD}$  variations and triggers CDC variations accordingly. For each IP module, static timing analysis is used to identify the critical paths,

allowing us to determine how many CDC modulators should be inserted (and where). Here, we illustrate the methodology using *one* power grid partition, *one* functional module and *one* CDCM system.

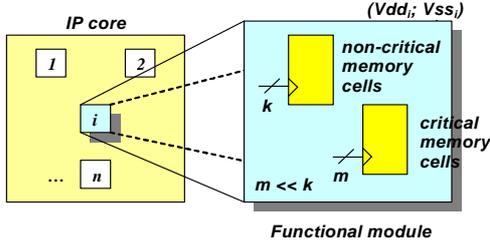


Fig. 1. Synchronous IP core under local  $V_{DD}$  disturbance

As shown in Fig. 2, the proposed CDCM architecture is based on a clock stretching Logic (CSL) block, which monitors power grid activity and enhances the clock signal ( $CLK$ ) delivered by the phase-locked loop ( $PLL$ ) block to the controlled logic accordingly. In this architecture, the CDCM module performs disturbance monitoring and CDC stretching using a simple circuit and introducing a limited clock delay,  $\tau_o$ , (referred to as the *intrinsic CDCM delay*). The modulated clock signal adds a minimum stretch at nominal  $V_{DD}$  (CDC slightly greater than 50%). When  $V_{DD}$  decreases, the CSL block stretches CDC according to  $V_{DD}$  reduction. In this case, the time slack is increased by the amount  $\tau_o$ , i.e.,  $t'_s = t_s + \tau_o(V_{DD})$ . When the  $V_{DD}$  voltage transient fades away, the CSL block starts gradually reestablishing the original clock duty-cycle (50%) to the controlled logic (*Critical Memory Cells*).

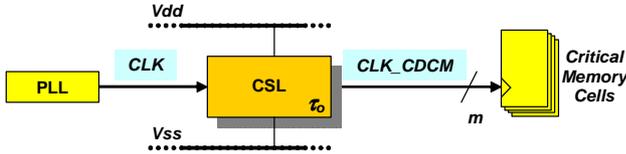


Fig. 2. Basic architecture of the CDCM system.

Several architectures for the CSL block can be implemented. One possible implementation in static CMOS technology is shown in Fig. 3. The CSL block modulate CDC by delaying *one* of the switching transitions (in this case, the **High-to-Low** transition). The M1-M2 inverter implements the CDC modulator core. Its pull-up PMOS transistor (M2) has a slow driving capability, due to M3. The result is a modulated clock signal, with identical clock period, but with enhanced CDC in the presence of a reduction of  $V_{DD}$ . The output signal is buffered to restore the fast switching capability. For circuits with both rising and falling-edge trigger clocks, two types of CSL circuits must be implemented.

In Fig. 4, simulation results (AMS 350 nm CMOS technology) show the CSL block capability to stretch

CDC proportionally to  $V_{DD}$  depletion, from nominal  $V_{DD}$  (3.3 Volt) down to  $V_{DD} = 1.0$  Volt. Note that the stretched CDC is always larger than 50% and lower than 100%. When  $V_{DD}$  is depleted to the lowest possible value (1.1V) CDC reaches 79%. Below 1.1 V, the correct functionality collapses.

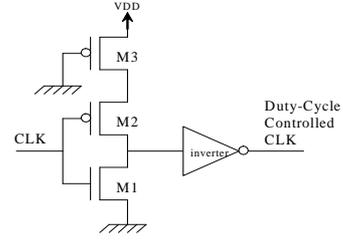


Fig. 3. Typical architecture for the CSL block.

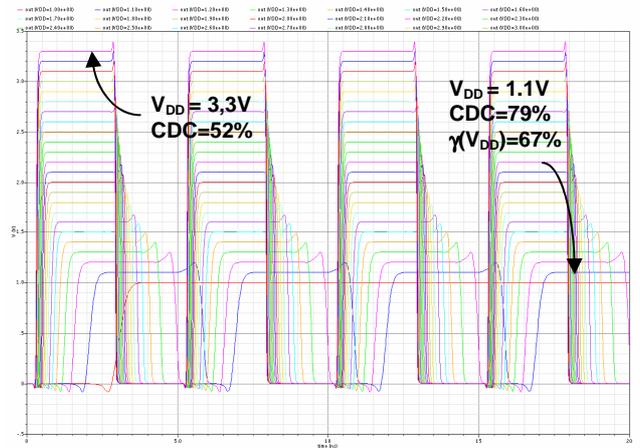


Fig. 4. CSL robustness to power supply voltage variations 200 MHz,  $V_{DD} \in (3.3; 1.0)$  V

### 3.1. CDC Modulation Model

A model to accurately quantify *CDC stretching* as a function of  $V_{DD}/V_{SS}$  fluctuations has been developed. The amount of CDC variation, required to tolerate a given  $V_{DD}$  variation, depends on the designer's defined *time slack* value,  $t_{PM}$ . Typically, it may correspond, for nominal values of the circuit parameters, to 8-10% of  $T_{pd}$ , the time period required by the longest critical path to propagate the switching values. When the power supply range,  $(V_{DD} - V_{SS})$  decreases, it erodes  $t_{PM}$ . One part of the speed degradation may be "absorbed" by the reduction in  $t_{PM}$ , without causing a functional error; however, if this occurs, there is no guarantee of correct functionality. Therefore, in our approach, CDC variation, required to tolerate  $V_{DD}$  variation, must preserve the nominal  $t_{PM}$  value. If  $t_{CLK}$  is the clock signal period, then

$$t_{CLK} = T_{pd} + t_{PM} \quad (2)$$

In the absence of CDC modulation,  $CDC=50\%$  and thus

$$CDC = T_{on} / t_{CLK} = 50\% \quad (3)$$

where  $T_{on}$  is the time interval in which the clock signal assumes the '1' logic value. When a  $\Delta V_{DD}$  occurs, a new value of CDC, referred as CDC', must be applied. In the worst-case

situation (for which nominal  $t_{PM}$  is kept), the clock duty cycle must stretch by the same amount of time required for signal propagation through the logic path. Hence,

$$CDC' = \{T_{on} + T_{pd} \cdot \Delta T_{pd}\} / t_{CLK} \quad (4)$$

and

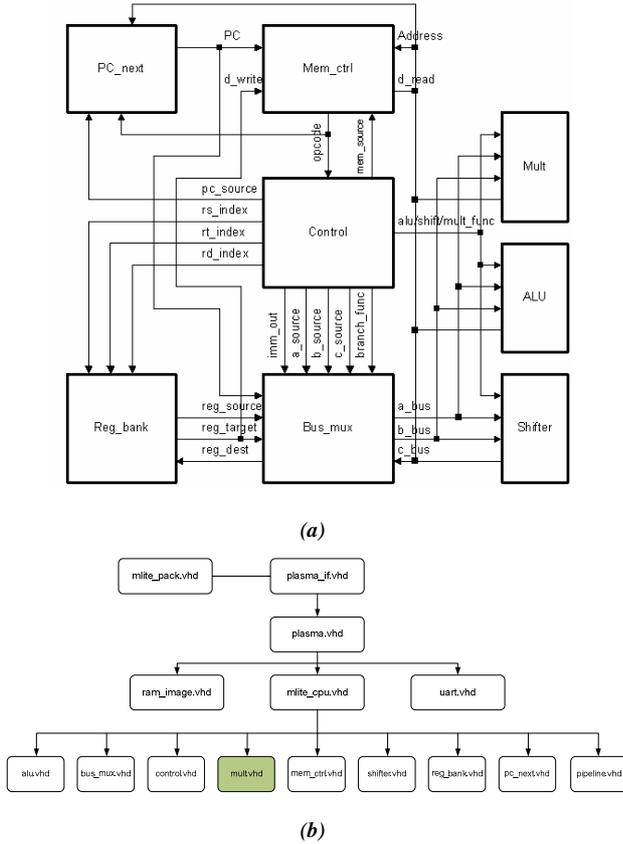
$$\Delta CDC(V_{DD}) = CDC' - CDC = [T_{pd} / t_{CLK}] \cdot \Delta T_{pd}(V_{DD}) \quad (5)$$

As a consequence, *CDC maximum variation follows  $T_{pd}(V_{DD})$  variation*, for which the authors have previously developed a simple analytical model [12,15]. The typical architecture of the CSL block, depicted in Fig. 4, allows deriving such variation.

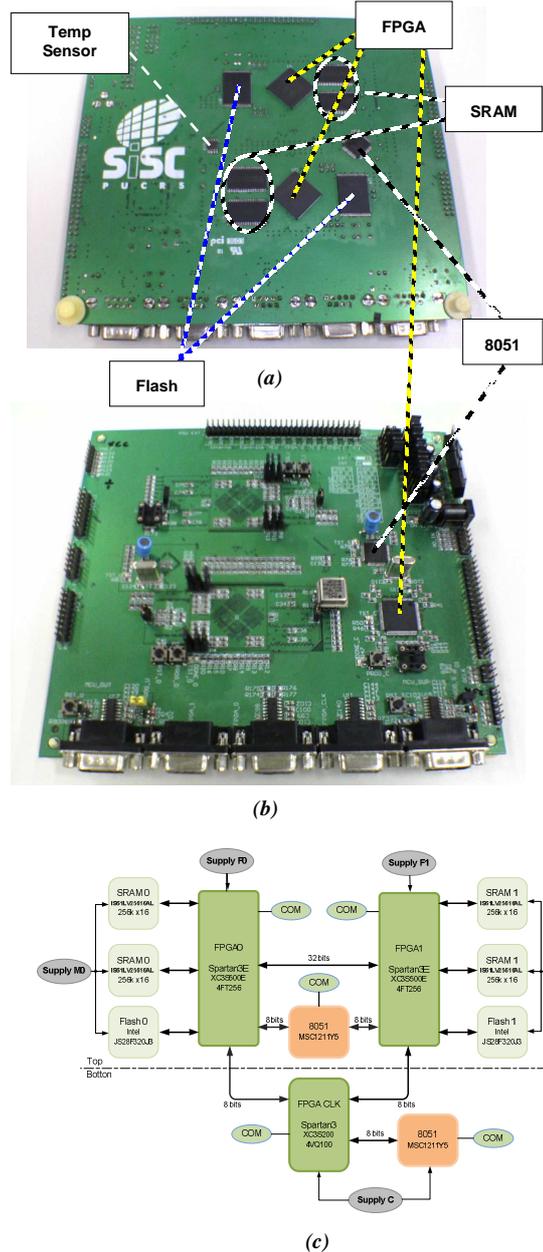
#### 4. EXPERIMENTAL RESULTS

Practical experiments were performed on a 32-bit microprocessor that was mapped into a FPGA IC. Aiming at demonstrating the proposed approach effectiveness, these experiments are described hereafter.

We selected the PLASMA soft-core microprocessor which was retrieved from the Internet public domain ‘‘Opencores’’ [26]. This is a 32-bit three-stage pipeline processor described in VHDL and mapped into an IEC 62.132-compliant test board [25] previously designed and implemented by the authors. Fig. 5 depicts the PLASMA general architecture and the VHDL basic blocks used to describe the processor. Fig. 6 shows a photo and describes the main ICs and connections of the test board.



**Fig. 5.** PLASMA processor: (a) general architecture, and (b) VHDL basic blocks used to describe the processor.



**Fig. 6.** General view of the IEC-compliant test board used to prototype the PLASMA processor: (a) test side, (b) the remaining glue logic, (c) block diagram.

Among the facilities available in this board, there is a temperature sensor that allows us to perform a burn-in test in conjunction with IEC test sessions<sup>2</sup> [25]. Additionally, the 2 FPGAs, 4 SRAMs, 2 Flashes and the 8051 microcontroller have separated power-supply bus. This infra-structure allows us to perform individual RF-conducted noise injection into any of these ICs, or any combination of them. To do so, a second board (not shown in Fig. 7) containing special connectors for RF-conducted noise injection is attached on the periphery pins of the board shown in Fig. 7b.

Two versions of this processor were implemented and prototyped in one of the FPGAs of Fig. 6a. The first version

<sup>2</sup> IEC 62.132, IEC 61.004-17 and IEC 61.004-29.

(namely, “*Original*”) had the CDC fixed to 50% during the whole experiment. The second version (“*Multiplier*”) had the CDC controlling the Multiplier Block modulated during  $V_{DD}$  fluctuations. To do so, we modified the VHDL code describing the PLASMA by adding an external pin to the processor so that we were able to control the CDC addressing the Multiplier Block. The remaining blocks of the processor had the CDC fixed to 50%. The CDC was modified in such a way that when the modulation was activated, the rising edge of the clock signal was delayed.

Note that:

(1) we selected the Multiplier Block to have the CDC controlled because this block was the only one in the PLASMA description (Fig. 6c) that had explicit clock signal declaration in the VHDL code. For the PLASMA remaining blocks, the clock signal was declared as asynchronous actions triggered after the instruction decoding from the Control Block. So, if other processor blocks had to be controlled, we would have to carry out complex changes in the VHDL code.

(2) All the D flip-flops (D-FFs) in the VHDL code of the Multiplier Block were declared as the type “*rising-edge D-FFs*”. So, when the CSL block modulated the clock signal, it actually reduced the CDC from 50% to 25%. This action was carried out by delaying the rising-edge of the clock signal. Thus, it delayed the D-FFs triggering instant. Fig. 7 depicts this situation.

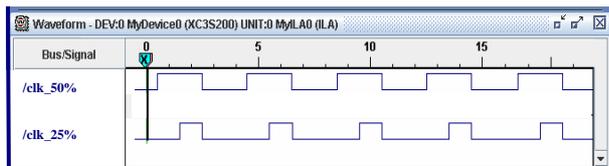


Fig. 7. Summary of the original and the stretched clock waveforms: Original clock ( $\text{clk}_{50\%}$ ); Stretched clock activating rising-edge FFs ( $\text{clk}_{25\%}$ ).

The experiment was performed by applying voltage dips to the FPGA  $V_{DD}$  pins according to the IEC 61.000-4-29 International Standard [25]. Then, we induced voltage dips of 37% to the 3.3V pins feeding the periphery (I/O pads) and 41% to the 1.2V pins feeding the core logic of the FPGA. These voltage dips were injected in the FPGA pins at a frequency range varying from 4 to 14 MHz. Based on this condition, the resulting  $V_{DD}$  fluctuated from  $\sim 3.3V$  to  $\sim 2.11V$  (for the periphery) and from  $\sim 1.2$  to  $\sim 0.75V$  (for the logic core). We tried also to apply deeper voltage dips, but the FPGA IC lost configuration and collapsed. Fig. 8 displays the injected noise captured with the oscilloscope at the FPGA  $V_{DD}$  input pins.

Fig. 9 summarizes the measured output values for the processor. For this experiment, the processor executed (in a time-shared basis) three application tasks under the control of its native operating system: Multiplication Matrix (MM), Curve Fitting (CF) and Cryptography (CR). The experiment was arranged in such a way that the PLASMA executed the three tasks with the same priority. This condition allocated 33% of the CPU time to run each of the tasks.

Additionally, the usage of the Multiplier Block by the tasks was intentionally made not equal: task CR was the one that used extensively this hardware resource. Next, the MM task was implemented in such a way to use it moderately. Finally, the CF was the application task that least used the Multiplier Block. As shown in Fig. 9b, this Multiplier Block

usage condition was reflected in terms of CDC Modulation effectiveness: the CR task was the one that presented the best robustness increase (from 155 fault occurrences to zero), followed by the MM and then, the CF task.

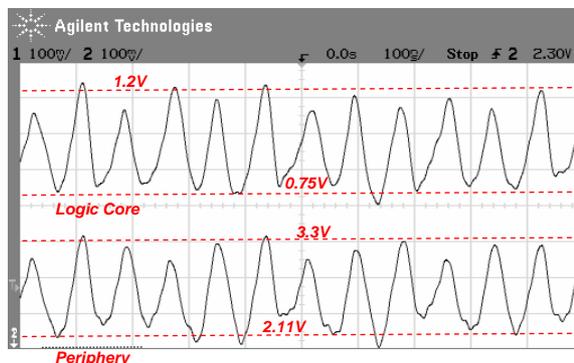


Fig. 8. IEC 61.000-4-29-compliant Injected noise captured with the oscilloscope at the FPGA  $V_{DD}$  input pins

Note that for the specific case of the CF task, the number of faults increased (from 26 to 40 occurrences). This can probably be explained: (a) by the size of the selected inputs and (b) by the CPU time allocated to run this task: we used four thousand five hundred different input test vectors divided equally among the three tasks. Since the CF task was the one that least used the Multiplier Block, probably a greater input test vector set running under a larger time basis would be required to demonstrate the *slight* increase of the PLASMA robustness when using the proposed technique. At present, this assumption is under investigation and we expect to present updated results in the upcoming work.

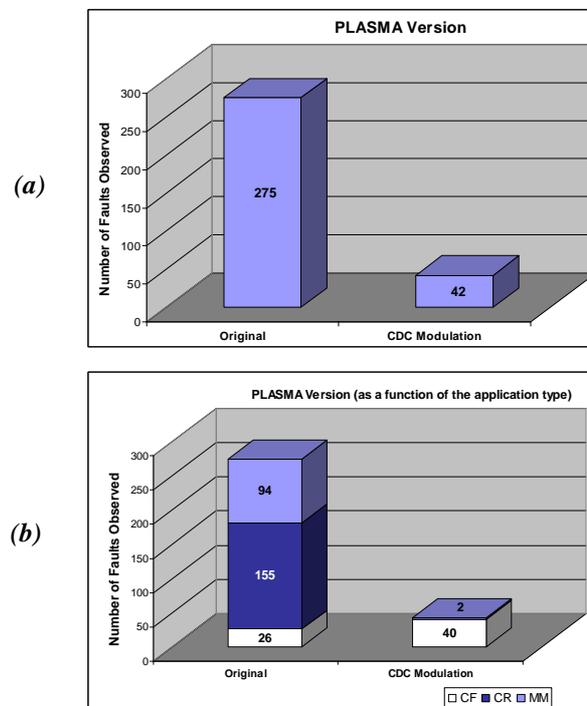


Fig. 9. Summary for the IEC 61.000-4-29-based experiment.

## 5. CONCLUSIONS

It was presented a new methodology aiming at enhancing SoC signal integrity to power-supply fluctuations. The goal is to keep circuit synchronism and nominal performance even in the presence of  $V_{DD}/GND$  disturbances that, due to increased signal path delays, may cause IC de-synchronization. Typical solutions define, during design phase, conservative (reduced) IC clock rates to ensure synchronization between signal propagation through logic with clock control under power-supply grid disturbance.

The methodology is based on a clock duty cycle modulation (CDCM) concept, which is built-up around a clock stretching logic (CSL) block. The CSL is used to monitor abnormal power interconnect activities ( $V_{DD}$  undershoots or ground overshoots) and to enhance CDC proportionally to supply voltage variation. This solution is implemented by delaying the rising (or falling) edge of the clock signal, depending on which kind of D-FF is used in the controlled logic. A typical implementation of the CSL block was proposed. The range of the CDC variation to tolerate  $V_{DD}$  disturbance has also been identified and discussed.

Experimental results based on the implementation of a 32-bit pipeline microprocessor into a FPGA demonstrate the *effectiveness* of the proposed methodology to react to power-supply fluctuations by stretching the clock signal fed to the controlled logic. It is a primary condition that the CSL block must be, by design, more robust to power supply voltage transients than the logic to be controlled. As shown in Section 3, Fig. 4, by carefully adjusting internal CSL block parameters it is possible to reach self-tolerance to power-supply oscillation up to 50% ( $\gamma(v_{dd}) = 67\%$ ). It is worth noting that, as expected for any logic, this tolerance is operating frequency-dependent: as higher as operating clock rates are, less tolerant is the CSL block to  $V_{DD}/GND$  oscillations.

**If accepted, we will include in the Final Version of this paper results for RF-conducted noise injection and GTEM cell according to the IEC standards 62.132-4 and 62.132-2, respectively. At present, these experiments are under development.**

## REFERENCES

- [1] M. Nourani, A. Attarha. "Signal Integrity: Fault Modelling and Testing in High-Speed SoCs". Journal of Electronic Testing: Theory and Applications (JETTA), 2002, pp. 539-554.
- [2] M. Cuviallo, S. Dey, X. Bai, Y. Zhao. "Fault Modelling and Simulation for Crosstalk in System-on-Chip Interconnects". Int. Conf. on Computer Aided Design, pp. 297-303, Nov. 1999.
- [3] W. Chen, S. Gupta, M. Breuer. "Test Generation in VLSI Circuits for Crosstalk Noise". Int. Test Conf. (ITC'98), pp. 641-650, 1998.
- [4] P. Fang, J. Tao, J. Chen, C. Hu. "Design in Hot Carrier Reliability for High Performance Logic Applications". Proc. IEEE Custom Integrated Circuits Conf., pp. 2511-2517, Oct. 1998.
- [5] Y. Leblebici. "Design Considerations for CMOS Digital Circuits with Improved Hot-Carrier Reliability". IEEE Journal of Solid-State Circuits, vol. 31, no. 7, pp. 1014-1024, July 1996.
- [6] S. Kimothi, U. Nandwani. "Uncertainty Considerations in Compliance-Testing for Electromagnetic Interference". Annual Reliability and Maintainability Symposium, pp. 265-268, 1999.
- [7] F. Vargas, D. C. Lopes, E. Gatti, D. Prestes, D. Lupi. "On the Proposition of an EMI-Based Fault Injection Approach". Proc. 11<sup>th</sup> IEEE Int. On-Line Testing Symposium, pp. 207-208, 2005.
- [8] S. Zhao, K. Roy. "Estimation of Switching Noise on Power Supply Lines in Deep Sub-Micron CMOS Circuits". Proc. Int. Conf. on VLSI Design, pp. 168-173, Jan. 2000.
- [9] F. Vargas, D. C. Lopes, J. Chaves da Silva, D. Barros Jr. "EMI-Induced Soft-Error Rate Estimates for COTS Microprocessor". Proc. 5<sup>th</sup> IEEE Latin American Test Workshop, 2004, pp.169-172.
- [10] H. Chen, L. Wang. "Design for Signal Integrity: The New Paradigm for Deep-Submicron VLSI Design". Proc. Int. Symposium on VLSI Technology, pp. 329-333, June 1997.
- [11] D. Cho, Y. Eo, M. Seung, N. Kim, J. Wee, O. Kown, H. Park. "Interconnect Capacitance, Crosstalk and Signal Delay for 0.35 $\mu$ m CMOS Technology". Proc. Int. Meeting on Electron Devices, pp. 619-622, 1996.
- [12] M. Rodríguez-Irago, J. J. Rodríguez Andina, F. Vargas, M. B. Santos, I.C. Teixeira, J. P. Teixeira. "Dynamic Fault Test and Diagnosis in Digital Systems Using Multiple Clock Schemes and Multi-VDD Test". Proc. 11th IEEE Int. On-Line Testing Symposium (IOLTS), pp. 281-286, 2005.
- [13] A. V. Oppenheim, A. S. Willsky, S. H. Nawab. "Signals and Systems". 2nd Edition, 1997, Prentice Hall Inc.
- [14] K. Berstein, K.M. Carrig, Ch.M. Durham, P.R. Hansen, D. Hogenmiller, E.J. Nowak, N.J. Rohrer. "High Speed CMOS Design Styles", Kluwer Academic Publishers, 1999.
- [15] D. Barros Júnior, M. Rodríguez-Irago, M.B. Santos, I.C. Teixeira, F. Vargas and J.P. Teixeira, "Fault Modeling and Simulation of Power Supply Voltage Transients in Digital Systems on a Chip", Journal of Electronic Testing, Theory and Application (JETTA), vol.21, pp. 349-363, Kluwer Academic Publishers, August, 2005.
- [16] L. Green. "Simulation, Modeling and Understanding the Importance of Signal Integrity". IEEE Circuit and Devices Magazine, pp. 7-10, Nov. 1999.
- [17] R. Downing, P. Gebler, G. Katopis. "Decoupling Capacitor Effects on Switching Noise". IEEE Trans. on Components, Hybrids and Manufacturing Technology, vol.16, no.5, pp. 484-489, Aug. 1993.
- [18] R. Saleh, D. Overhauser, S. Taylor. "Full-Chip Verification of UDSM Designs" Proc. Int. Conf. on Computer Aided Design (ICCAD), pp. 453-460, 1998.
- [19] A. Kahng, S. Muddu, E. Sarto. "Interconnect Optimization Strategies for High-Performance VLSI Designs". Proc. Int. Conf. on VLSI Design, pp. 464-469, Aug. 1999.
- [20] G. Tellez, M. Sarrafzadeh. "Minimal Buffer Insertion in Clock Trees with Skew and Slew Rate Constraints." IEEE Trans. on CAD, vol. 16, no. 4, pp. 333-342, April 1997.
- [21] Y. Jiang, S. Sapatnekar, C. Bamji and J. Kim. "Interleaving Buffer Insertion and Transistor Resizing into a Single Optimization". IEEE Trans. on VLSI Systems, vol. 6, no. 4, pp. 625-633, Dec. 1998.
- [22] T. Steinecke, W. John, . Koehne, M. Schmidt. "EMC Modeling and Simulation on Chip level". IEEE Int. Symposium on Electromagnetic Compatibility (EMC), 14-16 Aug. 2001.
- [23] S. Ben Dia, M. Ramdani, E. Sicard, editors, "Electromagnetic Compatibility of Integrated Circuits – Techniques for Low Emission and Susceptibility", Springer, 2006.
- [24] E. Sicard, F. Vargas, F. Hernandez, F. Fiori, J. P. Teixeira. "Design and Test on Chip for EMC". IEEE Design and Test of Computers, Issue. Nov/Dec. 2006, pp. 502-503.
- [25] IEC International Standards. [www.iec.ch](http://www.iec.ch).
- [26] [www.opencores.org](http://www.opencores.org).