

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

TÉCNICA DE PROJETO PARA AUMENTO DA ROBUSTEZ DE CIRCUITOS
ASSÍNCRONOS FRENTE AO RUÍDO ELETROMAGNÉTICO CONDUZIDO

AARON HENGLES

PORTO ALEGRE
2011

AARON HENGLES

TÉCNICA DE PROJETO PARA AUMENTO DA ROBUSTEZ DE CIRCUITOS
ASSÍNCRONOS FRENTE AO RUÍDO ELETROMAGNÉTICO CONDUZIDO

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

Orientador: Prof. Dr. Fabian Luis Vargas

PORTO ALEGRE

2011

AGRADECIMENTOS

Este trabalho de mestrado deve não só ao meu esforço, mas principalmente pela colaboração de muitas pessoas que este trabalho foi possível. E com sentimento de gratidão que cito alguns nomes:

Antes de tudo, quero agradecer a Deus. Por suas bênçãos todos os dias da minha vida. ‘Pois dEle, por Ele e para Ele são todas as coisas. A Ele seja a glória para sempre! Amém.’

Ao meu orientador, Prof. Dr. Fabian Luis Vargas, por compartilhar o tema de pesquisa e ser um interlocutor disposto a oferecer estímulos e, principalmente, a percorrer novos caminhos, ouvir com interesse e ânimo todos os questionamentos e problemas que surgiram durante esse trabalho. Pela sua capacidade e ousadia pela inovação, mesmo com riscos advindos dessa atitude. Por sua amizade e compreensão, principalmente em momentos difíceis de dúvidas em qual caminho a seguir no início dessa etapa que chegou ao fim.

À profa. Dr. Leticia Bolzani, por seu fundamental trabalho ao meu lado, fazendo parte do laboratório de pesquisas SiSC, oferecendo, durante o meu período de mestrado, muitos exemplos, sugestões e críticas fundamentais para a elaboração de um bom trabalho de mestrado. Por sua participação em minha banca, sempre com críticas construtivas a respeito do trabalho. Finalmente pela sua amizade no dia a dia dos dois anos no qual me dediquei a este trabalho.

À minha família, por toda compreensão, apoio, carinho e amor, em todos os dias em que estiveram ao meu lado incondicionalmente. Ao meu pai, Eliseu Hengles Junior, exemplo de um homem reto, que com orgulho sempre me apoiou. A minha mãe, Betina Concha Vásquez Hengles, com seus conselhos e sua fibra. E finalmente meus irmãos que me apoiaram, principalmente minha irmã Thaísa Hengles em suas correções à minha dissertação.

A todos os amigos e amigas que estiveram ao meu lado nessa etapa, principalmente a Marília Gaelzer Müller que esteve mais próxima me apoiando em momentos difíceis.

Os amigos André Luís Del Mestre, Eduardo Chielle e Rodrigo Remor por ajudas referentes ao trabalho.

A todos agradeço, profundamente, e dedico os resultados obtidos neste trabalho.

AARON HENGLES

**TÉCNICA DE PROJETO PARA AUMENTO DA ROBUSTEZ DE CIRCUITOS
ASSÍNCRONOS FRENTE AO RUÍDO ELETROMAGNÉTICO CONDUZIDO**

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

BANCA EXAMINADORA:

Prof. Dr. Sebastião Cícero Pinheiro Gomes - FURG

Prof. Dra. Leticia Bolzani Pöhls – PUCRS

Prof. Dr. Fabian Luis Vargas – PUCRS

RESUMO

Atualmente, sistemas eletrônicos embarcados (*System-on-Chip -SoC*) são cada vez mais populares, com custos cada vez menores e performance cada vez mais elevada usados em aplicações críticas. Por estarazão, é necessário que estes sistemas sejam extremamente confiáveis e robustos. Observa-se que a grande maioria dos *SOCs* utilizados atualmente faz uso do paradigma síncrono, o qual se baseia em um sinal de relógio global para sincronizar todo o circuito. Porém, é importante salientar que essa arquitetura apresenta sérios problemas relacionados à compatibilidade eletromagnética (*Electromagnetic Compatibility - EMC*), tanto no que tange à emissão quanto à susceptibilidade.

Neste contexto, circuitos assíncronos representam uma alternativa extremamente viável capaz de amenizar e até mesmo solucionar tais problemas de EMC, pois circuitos assíncronos tendem intrinsecamente a serem mais robustos ao ruído eletromagnético. A grande dificuldade frente ao paradigma assíncrono está fundamentada ao fato de que projetistas não estão preparados para essa mudança de paradigma bem como uma carência de ferramentas de CAD voltadas para o desenvolvimento deste tipo de circuito no mercado.

Assim, esta dissertação de mestrado visa propor uma metodologia de projeto de circuitos assíncronos que correlacione o tipo de ruído eletromagnético existente no meio onde o circuito será operado com a confiabilidade esperada para o mesmo. Basicamente, este objetivo é alcançado através do aumento progressivo dos atrasos da lógica de controle dos circuitos de *handshaking* entre estágios de um circuito pipeline assíncrono.

Ao final, este trabalho apresenta resultados de experimentos práticos realizados para validar a metodologia proposta através da injeção de ruído conduzido nas linhas de alimentação (*Power Supply Disturbances - PSD*) de diferentes versões do circuito assíncrono adotado como estudo-de-caso. Os experimentos práticos foram realizados de acordo com o *Standard* internacional IEC 61.000-4-29, que define parâmetros para a geração de ruído nas linhas de alimentação de circuitos e sistemas eletrônicos integrados.

Palavras-chave: projeto de circuitos assíncronos robustos, compatibilidade eletromagnética (EMC), ruído eletromagnético conduzido nas linhas de alimentação, standard internacional IEC 61.000-4-29.

ABSTRACT

Nowadays, electronic systems (System-on-Chip -SoC) are becoming more and more popular, with reduced costs and high performance. For this reason, it is mandatory that such systems become more reliable and robust than ever. Most of the SOCs currently adopted make use of the asynchronous paradigm, which is based on a global clock signal to synchronize the whole system. However, this architecture presents serious problems related to the electromagnetic compatibility (EMC), namely, high electromagnetic emission level and reduced susceptibility.

In this context, asynchronous circuits represent an interesting alternative, capable to solve or at least minimize the above mentioned problems. Such condition is observed because asynchronous circuits tend to become intrinsically more robust to electromagnetic interference. The most important drawback of asynchronous circuits is that designers are not prepared for this change of paradigm as well to the lack of CAD tools to develop this type of circuit.

In this scenario, the present work proposes a new methodology to increase the robustness of asynchronous circuits when exposed to electromagnetic interference (EMI). This goal is attained by increasing progressively the delay of the control logic of the handshaking circuits between stages of an asynchronous pipeline circuit.

This work concludes by presenting experiments aiming to validate the proposal. In these experiments, conducted electromagnetic interference is applied to the power supply lines of different versions of the circuit adopted as the case-study. Such noise is generated according to the international standard IEC 61000-4-29, which defines rules for generating voltage dips, short interruptions and voltage variations on the DC power port of electronic systems and integrated circuits.

Keywords: design of robust asynchronous circuits, electromagnetic compatibility (EMC), noise injection on power-supply lines, international standard IEC 61000-4-29.

LISTA DE FIGURAS

Figura 1: Modelo de falhas.....	21
Figura 2: Latências de falha e erro.....	24
Figura 3: Curvas de tensão.....	26
Figura 4: Curva abaixo da tensão nominal.....	26
Figura 5: Interferência conduzida.....	33
Figura 6: Interferência radiada.....	33
Figura 7: Exemplos de interferências.....	34
Figura 8: Exemplo de integridade de sinal.....	35
Figura 9: <i>Ripple</i>	38
Figura 10: Restrição a circuitos síncronos.....	43
Figura 11: Circuito síncrono genérico.....	44
Figura 12: Circuito assíncrono genérico.....	44
Figura 13: Controladores mestre escravo.....	45
Figura 14: Classificação assíncrona.....	48
Figura 15: <i>Isochronics forks</i>	49
Figura 16: Circuito micropipeline.....	50
Figura 17: Modelos de atraso.....	51
Figura 18: Protocolo de quatro fases.....	52
Figura 19: Protocolo de duas fases.....	52
Figura 20: Codificação de trilha única.....	53
Figura 21: Codificação de trilha dupla em um protocolo de quatro fases.....	54
Figura 22: Codificação de trilha dupla por valor de duas fases.....	54
Figura 23: Codificação de trilha dupla por paridade de duas fases.....	55
Figura 24: Elemento C-muller.....	56
Figura 25: Comparação dos fluxos.....	58
Figura 26: Circuito Síncrono e Assíncrono.....	59
Figura 27: Circuito assíncrono inicial.....	60
Figura 28: Protocolos de handshake.....	61
Figura 29: Controlador de trava.....	61
Figura 30: Flip-Flop onde não é permitido escorregamento.....	63
Figura 31: Dependência de dados.....	64
Figura 32: Rede de controladores resultante.....	64

Figura 33: Elementos de atraso	65
Figura 34: Lógica de detecção e conclusão	66
Figura 35: Circuito resultante	67
Figura 36: Placa do circuito impresso	69
Figura 37: Plataforma de ensaio	70
Figura 38: Vista superior da placa de teste.....	71
Figura 39: Vista inferior da placa de testes	72
Figura 40: Injetor de falhas	72
Figura 41: Classificação da plataforma.....	73
Figura 42: Configuração do gerador.	75
Figura 43: Processador DLX – <i>pipelines</i>	78
Figura 44: DLX dessincronizado.....	81
Figura 45: Comparação entre os dois processadores	81
Figura 46: ASPIDA DLX.....	82
Figura 47: ASPIDA DLX para ensaio.	83
Figura 48: Configuração dos experimentos.....	84
Figura 49: Fluxo de execução	85
Figura 50: Tensões para ensaios	87
Figura 51: Gráfico comparativo entre a versão síncrona e versão assíncrona.	88
Figura 52: Resultados da norma de ripple.....	88
Figura 53: Modelo de grafos para <i>pipelines</i>	92
Figura 54: Gráfico de transição dos controladores	93
Figura 55: Estrutura interna dos controladores	95
Figura 56: Restrições de tempo para controladores	96
Figura 57: Implementação dos atrasos	97
Figura 58: Mostra da implementação de atraso.....	99
Figura 59: Setup de ensaios.	100
Figura 60: Fluxo de execução.....	101
Figura 61: Gráfico de falhas funcionais.....	104
Figura 62: Falhas Funcionais	105
Figura 63: Confiabilidade do processador DLX em função de diferentes valores para a lógica de atraso e quedas de tensão de alimentação.	106

LISTA DE TABELAS

Tabela 1: Medidas de dependabilidade.....	28
Tabela 2: Padrões IEC 61.000.....	37
Tabela 3: Definições de <i>ripple</i>	37
Tabela 4: Níveis de tensão e duração recomendada para queda de tensão.....	39
Tabela 5: Níveis de tensão e duração recomendada para pequena interrupção.....	39
Tabela 6: Níveis de tensão e duração recomendada para variação de tensão.....	40
Tabela 7: Tabela verdade <i>C-element</i>	55
Tabela 8: Características da implementação do gerador.....	73
Tabela 9: Instruções DLX – conjunto 1.....	79
Tabela 10: Instruções DLX – conjunto 2.....	80
Tabela 11: Comparação de versões.....	81
Tabela 12: Relação de área original.....	103
Tabela 13: Custos de atraso.....	103
Tabela 14: Indicação de atrasos.....	105
Tabela 15: Representação de falhas em ciclo de relógio.....	106
Tabela 16: Falhas funcionais com diferentes atrasos e variações.....	107

LISTA DE SIGLAS

ARM – Advanced RISC Machines
ASIC - *Application Specific Integrated Circuit*
ASPIDA - *ASynchronous Open-source Ip of the Dlx Architecture*
CLB - *Configurable Logic Block*
CLK – *Clock*
DI – *Delay Insensitive*
DSP - *Digital Signal Process*
EEPROM - *Electrically Erasable Programmable Read Only Memory*
EMC - *Compatibilidade Eletromagnética*
EME - *ElectromagneticEnvironment*
EMI – *EletromagneticInterference*
EMI – *EletromagneticInterference*
EPROM - *Erasable Programmable Read Only Memory*
FPGAs - *Field Programmable Gate Array*
GALS - *Globally Asynchronous Locally Synchronous*
GCC - *GNU Compiler Collection*
Gnd - *Ground Bounce*
IEC - *InternacionalElectrotechnical Commission*
ISA - *Instruction Set Architecture*
MEMS - *Micro-Electro Mechanical Systems*
MG - *Marked Graph*
MIPS - *Millions of Instructions Per Second*
MOS - *Metal Oxide Semiconductor*
MP3 - *Motion Picture Experts Group Layer – 3*
MPEG2 - *Motion Picture Experts Group Layer – 2*
MTBF - *Mean Time BetweenFailure*
MTTF - *Mean Time to Failure*
MTTR - *Mean Time toRepair*
NMOS - *Negative Channel Metal Oxide Semiconductor*
PAL - *Programmed Array Logic*
PCB - *Printed Circuit Board*
PLA - *Programmable Logic Array*

PMOS - *Positive Metal Oxide Semiconductor*
PROM - *Programmable Read Only Memory*
QDI *Quasi-Delay Insensitive*
RAM - *Random Access Memory*
RISC - *Reduce Instruction Set Computer*
SI - *SpeedIndependent*
SISC – *Sistemas, Sinais e Computação*
SoC - *Systems on Chip*
SRAM - *Static Random Access Memory*
STG - *Signal Transition Graph*
UART - *Universal Assynchronous Receiver Transmitter*
USB - *Universal Serial Bus*
VHDL - *VHSIC Hardware Description Language*
VSLI - *Very Large Scale Integration*

SUMÁRIO

1 INTRODUÇÃO	17
1.1 Motivação	17
1.2 Objetivos.....	18
1.3 Organização do trabalho	19
2 TOLERÂNCIA A FALHAS	21
2.1 Introdução	21
2.2 Conceito referentes à falhas	22
2.2.1 Classificação de falhas	23
2.3 Modelo de falhas	25
2.4 Mecanismos de injeção de falhas.....	27
2.4.1 Injeção de falhas por hardware	27
2.5 Medidas de dependabilidade.....	28
2.6 Conclusões do capítulo.....	28
3 COMPATIBILIDADE ELETROMAGNÉTICA	30
3.1 Introdução	30
3.2 Conceitos	31
3.3 Interferência eletromagnética.....	32
3.3.1 Fontes de interferência eletromagnética	33
3.3.2 Tipos de interferência	34
3.4 Interferência eletromagnética na eletrônica	35
3.5 Normatização	36
3.5.1 Norma 61.000-4-17	37
3.5.2 Norma 61.000-4-29	38
3.6 Conclusões do capítulo.....	40
4 CIRCUITOS ASSÍNCRONOS	41
4.1 Introdução	41
4.2 Princípios de funcionamento do paradigma assíncrono	43
4.3 Fenômenos temporais.....	45
4.4 Interação com o ambiente.....	47
4.5 Classificação	48
4.6 Modelagem de atraso para circuitos.....	50

4.7	Protocolo de comunicação	51
4.8	Codificação dos dados	53
4.9	<i>C-element de Muller</i>	55
4.10	Conclusões do capítulo	56
5	TÉCNICA DE DESSINCRONIZAÇÃO	57
5.1	Introdução	57
5.2	Modelo de dessincronização	58
5.2.1	Passos da dessincronização	59
5.3	Gráficos de transição de sinal	60
5.4	<i>Latch controller</i> - controlador	61
5.5	Substituição dos flip-flops	62
5.6	Remoção do relógio	63
5.6.1	Grafo de dependência	63
5.6.2	Atraso para lógica combinacional	64
5.7	Interligação da rede	66
5.8	Conclusões do capítulo	67
6	PLATAFORMA DE ENSAIOS	68
6.1	Introdução	68
6.2	Plataforma de ensaios	68
6.3	Injetor de ruído eletromagnético	72
6.3.1	Especificação	73
6.3.2	Fluxograma de configuração	74
6.4	Conclusão do capítulo	75
7	ANÁLISE DA ROBUSTEZ DOS PARADIGMAS	76
7.1	Introdução	76
7.2	Processador DLX	77
7.2.1	Aspida DLX	79
7.3	Processador DLX dessincronizado	80
7.4	Experimentos	82
7.4.1	Aplicação do estudo de caso	83
7.4.2	Ambiente dos experimentos	84
7.5	Resultados	87
7.6	Conclusões do capítulo	88

8METODOLOGIA PROPOSTA	90
8.1 Introdução	90
8.2Inserção de lógica de atraso baseada em grafos de transição de sinais.....	91
8.3Lógica de controle.....	94
8.3.1 Controladores.....	95
8.3.2 Lógica de atraso	96
8.3.3 Adição de atrasos	97
8.4Experimentos para validação da proposta	99
8.5Resultados	103
8.6Conclusões do capítulo.....	107
9 CONCLUSÃO	109
9.1Trabalhos futuros	110
10 REFERÊNCIAS.....	111
11 ANEXOS	118
11.1Plataforma de ensaios	118
11.2Publicações científicas referentes ao trabalho	126

1 INTRODUÇÃO

Atualmente é possível observar um aumento significativo na complexidade e desempenho dos sistemas embarcados utilizados nos mais variados tipos de aplicações presentes em diferentes segmentos do mercado [1]. Além disso, dado o sempre crescente número de aplicações móveis, observa-se um aumento da necessidade de sistemas com baixo consumo de energia. Neste contexto, surgem vários estudos que visam aumentar o desempenho, a confiabilidade e a robustez dos mesmos, principalmente no que tange circuitos microprocessados para aplicações críticas, como aeroespaciais e médicas. É importante salientar que a grande maioria dos sistemas embarcados atualmente utilizados, adota o paradigma síncrono, no qual apenas um sinal de relógio global controla todo o circuito. Apesar da grande popularidade, o paradigma síncrono apresenta sérios problemas relacionados à sensibilidade ao ruído, à emissão eletromagnética e ao consumo de energia. Estas características podem degradar o sistema, visto que podem provocar atrasos (*delay*) na propagação dos sinais levando o mesmo a apresentar algum tipo de falha [2] [3] [4].

Neste contexto, surgem os circuitos assíncronos como uma alternativa para solucionar os problemas mencionados anteriormente relacionados ao paradigma síncrono. Circuitos assíncronos são naturalmente *low power* e mais robustos [5]. Estes operam de forma mais confiável em voltagens mais baixas e utilizam menos energia que a proposta síncrona, reduzindo também a emissão eletromagnética devido ao fato de não apresentarem um relógio global. Assim, é possível explorar estas características para projetar circuitos com parâmetros de desempenho mais elevados, com menor consumo de energia e finalmente que emitam menos Interferência Eletromagnética (*Electromagnetic Interference - EMI*) [6].

1.1 MOTIVAÇÃO

Com o aumento das aplicações críticas embarcadas, aumentou também a preocupação com as técnicas capazes de agregar confiabilidade e robustez a Sistemas embarcados (*Systems on Chip - SoCs*). A oscilação nos níveis de tensão nas linhas de alimentação quando inseridos em ambientes reais considerados hostis estão muitas

vezes expostos e suscetíveis a degradação do sistema, podem gerar atrasos na propagação dos sinais. Com isso, ocorre a degradação do sistema, perda de sua confiabilidade e no pior dos casos, a queda total do sistema. Finalmente, isso pode gerar erros funcionais que muitas vezes são intoleráveis para certas aplicações.

Os problemas anteriormente apresentados tornam-se mais significativos em circuitos síncronos, dado ao atraso gerado no caminho crítico do circuito. Com o objetivo de superar essas limitações de projeto, vários grupos de pesquisa voltam-se para o desenvolvimento de projetos assíncronos. Esses circuitos não assumem pressuposto de discretização no tempo sendo o tempo uma variável contínua [6]. Entretanto, esses circuitos possuem características positivas quando citamos sistemas tolerantes a falhas, com menor consumo de energia (aplicações *low power*) e que operam com tensões mais baixas [6]. Em [10], é apresentado o primeiro trabalho concreto a respeito de *QuasiDelayInsensitive* (QDI) assíncrono em termos de EMC.

Todavia circuitos puramente assíncronos possuem limitações que privam o seu uso dentre os quais são citadas poucas ferramentas CAD para o seu desenvolvimento, e poucos projetistas preparados para atuar com esse tipo de circuito, sendo a mentalidade dos projetistas voltada para circuitos síncronos. Uma alternativa para esses problemas citados anteriormente é uma técnica de dessincronização em [Lavagno]. et. al. propõe uma técnica de dessincronização para que o circuito não precise de ferramentas CAD próprias, geralmente necessárias quando se trata do projeto de circuitos assíncronos convencionais. Finalmente, essa técnica é capaz de aumentar a robustez dos circuitos quando comparados ao circuito síncrono similar, apresentando bom funcionamento em situações em que normalmente a versão síncrona não funcionaria [13] [14].

1.2 OBJETIVOS

Este trabalho de mestrado tem como objetivo analisar o comportamento de circuitos síncronos e assíncronos quando expostos ao ruído gerado nas linhas de alimentação. Em mais detalhes, serão analisados aspectos relacionados à robustez dos mesmos no que diz respeito aos atrasos relacionados gerados a partir da variação da tensão no barramento de alimentação [7].

Posteriormente com base no recolhimento destas informações, é então desenvolvida uma proposta a partir de ensaios realizados de interferências eletromagnéticas e redução dos níveis de tensão seguindo as normas da IEC (*Internacional*

Electrotechnical Commission) [8] [9]. Dada as características da proposta de dessincronização, que faz a utilização de elementos de retardamento para a substituição do relógio global. Essa proposta é baseada na modificação dos atrasos referentes ao caminho crítico da lógica combinacional do circuito, ou seja, nos elementos de retardamento, propondo um nível de atraso maior que o atraso original do circuito, de forma que ele esteja funcional mesmo com a aplicação de um determinado nível de ruído, com algum máximo de falhas gerado, definido pelo *Standard IEC*.

Com isto espera-se obter um circuito mais robusto e confiável, quando exposto a ambientes hostis. A proposta da dissertação é de grande importância para a área de pesquisa relacionada à robustez e a integridade dos sinais frente à flutuação no barramento de alimentação.

1.3 ORGANIZAÇÃO DO TRABALHO

Esta dissertação foi dividida em três grandes partes subdivididas da seguinte forma:

Parte I – Desenvolvimento da proposta

- **CAPÍTULO 2** –Apresenta os principais conceitos de tolerância a falhas e suas técnicas e as principais técnicas propostas na literatura;
- **CAPÍTULO 3** –Demonstra os conceitos sobre a *EMC* (Compatibilidade eletromagnética, e *EMI* (Interferência eletromagnética), e normas IEC, todos focados principalmente na proposta;
- **CAPÍTULO 4** -Apresenta uma introdução dos principais conceitos relacionados aos circuitos assíncronos
- **CAPÍTULO 5** –Apresenta a técnica de dessincronização aplicada a circuitos síncronos proposta por [Lavagno]. et.al.

Parte II – Validação

- **CAPÍTULO6** –Apresenta a plataforma de ensaios utilizada nos experimentos do trabalho
- **CAPÍTULO7** –Apresenta o processador adotado, mostrando uma análise de comportamento que faz a comparação entre o circuito assíncrono e sua contraparte dessincronizada.
- **CAPÍTULO8** –Apresenta uma metodologia proposta para o aumento da robustez em circuitos assíncronos, apresentando os resultados obtidos da verificação da proposta.

Parte III – CONCLUSÃO

- **CAPÍTULO9** –Apresenta a conclusão da metodologia proposta a partir dos resultados obtidos e propõe trabalhos futuros.

2 TOLERÂNCIA A FALHAS

Para uma melhor visão do trabalho, se faz necessário introduzir os principais assuntos referentes aos modelos de falhas existentes com conceitos clássicos referente à área de tolerância a falhas, bem como as técnicas de injeção de falhas que serão abordados nas próximas sessões deste trabalho.

2.1 INTRODUÇÃO

Um circuito é dito tolerante a falhas quando ele permanece com todas as suas funcionalidades mesmo com a presença de falhas e sem prejudicar o seu desempenho. Um exemplo muito utilizado em aviões, cuja necessidade de ser tolerante a falhas é grande é a redundância cuja técnica é muito utilizada. A redundância pode ser realizada por três métodos:

- I. Hardware – com a utilização de circuitos extras;
- II. Software – programas que vão fazer o diagnóstico em tempo real;
- III. Temporal – duplicação de operações.

Um conceito importante que devemos ter em mente é saber diferenciar e ter um conceito bem formado a respeito de defeito (*failure*), erro (*error*) e falha (*fault*). A figura 1 exemplifica os modelos.

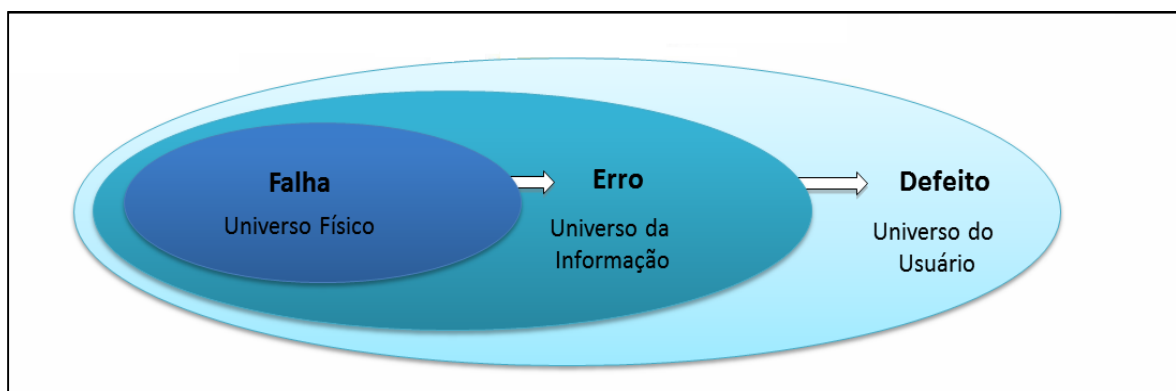


Figura 1 – Modelos de Falhas [15]

A figura vista anteriormente foi sugerida por **Dhiraj K. et. al** [15] e mostra os defeitos sendo referenciados ao universo do usuário ocorrendo quando há um desvio da especificação. Os erros são relacionados ao universo da informação que ocorre quando há alteração no sistema que pode ser levado a um estado diferente de sua especificação, sendo assim erro é o que causa o defeito. E por fim, falhas ao universo físico que é a causa primária de um defeito que venha a ocorrer no sistema. Uma falha não necessariamente leva a um estado de erro, pois a linha de código com falha pode nunca ser executada. Um erro também não necessariamente leva a um estado de defeito, pois talvez uma informação nunca seja usada.

Todos os sistemas estão suscetíveis a apresentar falhas, seja por estarem expostos as condições de interferência externa como ambientais e humanas, seja pelo fato de todos os sistemas envelhecerem e com o tempo vir apresentando cada vez mais falhas. As técnicas de tolerância a falhas surgem para que os defeitos sejam evitados, buscando sistemas mais confiáveis.

Um exemplo muito utilizado para fazer a distinção dos três modelos: Supondo que um determinado computador tem algum problema na fonte que altere a tensão que alimenta os seus componentes eletrônicos. Até o momento descrito se tem uma falha. Mas se esta alteração de tensão fizer que alguns bits da memória tenham os seus valores trocados de zero para um, a falha se transformou em erro. Se o erro anterior não for tratado e o sistema travar ou alterar alguma informação de um banco de dados, por exemplo, o erro transformou-se em defeito. Se o sistema foi afetado pelo problema, este não é mais um erro e sim defeito.

2.2 CONCEITOS REFERENTES ÀS FALHAS

Falhas transientes que ocorrem durante a vida útil dos componentes são provocadas por uma não conformidade e por fenômenos aleatórios ambientais onde o sistema está operando. Podemos citar como exemplo de falhas transientes componentes eletrônicos como rádios, telefones celulares, dentre outros componentes eletrônicos que causam variações na tensão de alimentação e geram interferências eletromagnéticas.

Já a falha intermitente dá em ocorrências cíclicas e temporárias dadas pelas condições externas e do ambiente em que o circuito está. Um exemplo que podemos citar é quando um componente que foi projetado para ser utilizado a temperaturas ditas

normais, é utilizado em um ambiente com temperaturas elevadas gerando erros em sua saída.

Devemos citar outros termos importantes para um bom entendimento desse trabalho retirados de [17] [18] [19] como:

- **Confiabilidade:** Capacidade de atender as especificações de projeto dentro de condições definidas durante certo período de funcionamento e estar operacional no início desse período;

- **Disponibilidade:** É dada pela probabilidade do sistema estar operacional quando for necessária a sua utilização;

- **Dependabilidade (*dependability*):** É a indicação da qualidade e confiança dada no serviço fornecido por algum sistema. A confiabilidade e a disponibilidade citadas anteriormente são dois atributos da dependabilidade;

- **Latência:** Período de tempo compreendido entre a ocorrência do erro até a sua devida manifestação. Como podemos observar na figura 2 descrita por Pradhan [15], temos a soma das latências de falha e erro para termos o tempo total entre a ocorrência da falha e o surgimento do defeito.

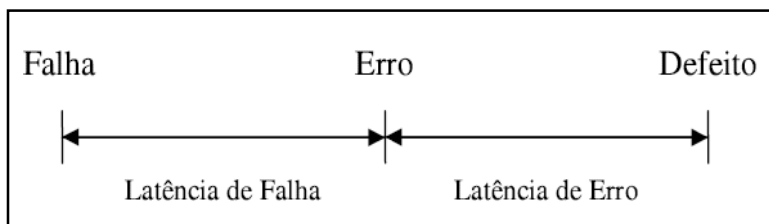


Figura 2 – Latências de falha e erro [15]

2.2.1 CLASSIFICAÇÃO DE FALHAS

Como citamos anteriormente, falhas podem ocorrer por vários motivos externos e internos, como também do seu próprio envelhecimento [16]. Dado isso, as

falhas foram classificadas em falhas permanentes, falhas transientes e falhas intermitentes [17].

As falhas permanentes são falhas que ocorrem no processo de fabricação do circuito e pelo seu envelhecimento, ocorrem no meio físico e podemos citar como exemplo curto circuitos, *stuck-at* e nós abertos.

As falhas são divididas principalmente em falhas físicas e falhas humanas [18]. As falhas físicas são aquelas de que padecem os componentes e as falhas humanas são as que compreendem falhas de projeto e falhas de interação.

Os principais motivos relacionados a falhas são problemas causados pela especificação, implementação, imperfeições da produção, fadiga dos componentes físicos e componentes defeituosos além de distúrbios gerados pelo meio como interferência eletromagnética, radiação e variações de temperatura, pressão e umidade como problemas de operação. Exemplo disso seria um fotógrafo que com sua câmera, ou seja, seu componente eletrônico, pode tanto fotografar no deserto com altas temperaturas, como em Alpes em temperaturas baixas, em cada situação os projetistas tem que levar em conta que em cada ambiente os componentes vão reagir de alguma forma podendo apresentar falhas caso essas situações não sejam analisadas.

Devemos citar outros itens a fim de definir uma falha [16] [17]:

- **Natureza** –Falha de software, falha de hardware, de projeto, de operação;
- **Valor** –Determinado ou indeterminado no tempo;
- **Persistência ou Duração** –Permanente ou temporária (intermitente ou transitória);
- **Extensão** –Global ou local a um determinado módulo do circuito.

As falhas de projeto e software são os principais problemas relacionados aos sistemas críticos. Sistemas críticos são idealizados para que tolerem falhas físicas. As falhas que não são previstas no projeto ou que não foram tratadas são as que mais geram danos nos sistemas possuindo um grande potencial de comprometer sua confiabilidade e sua disponibilidade.

2.3 MODELOS DE FALHAS

Com o objetivo de analisar sistemas e detectar falhas presentes em dispositivos, são realizados testes em circuitos integrados ou sistemas eletrônicos, para isso é necessário modelar as falhas se baseando em falhas reais definidas a partir de mecanismos físicos e layouts reais. Um modelo de falha é a especificação de uma série de defeitos físicos que são detectados através de um procedimento de teste [19].

Em [19] [20] os autores demonstram que os modelos de falhas devem atender os seguintes requisitos:

- I. Refletir com precisão o comportamento dos defeitos reais que acontecem durante os processos de fabricação e de manufatura, como também o comportamento das falhas que podem vir a acontecer durante a operação do sistema;
- II. Ter eficiência computacional respeitando o ambiente de simulação da falha.

Baseado nesses conceitos, modelos de falhas surgiram nos últimos anos baseados em defeitos físicos dos circuitos e dos sistemas eletrônicos. Dentre os modelos mais importantes temos:

- **Modelo de falha *delay*** – do ponto de vista lógico combinacional, os circuitos executam suas operações corretamente, todavia eles não são executados na frequência de operação nominal especificada [21]. O que ocorre em virtude disso é um erro de *timing* devido aos diferentes tempos de propagação dos sinais internos do circuito. Muitas biografias passam a falsa impressão que as transições de sinal acontecem em forma de degrau, mas não é o que realmente acontece na prática. A partir da figura 3(a) de **Tummeltshamme e Steininger** [22], podemos analisar que essa transição ocorre com uma curva exponencial. Onde t é uma constante de atraso e RC é a constante de tempo. Tudo isso é levado em conta para circuitos síncronos para determinar em que frequência o relógio do sistema irá operar. Na figura 3(b), nós temos a demonstração do que ocorre quando injetamos ruídos em um circuito, neste caso diminuindo a tensão nominal de 1.5V para 1.1V,

agora temos um atraso equivalente a $t + \varepsilon$, sendo ε um atraso adicional. Circuitos síncronos possuem margem para que isso ocorra e ele continue sem apresentar falhas, mas caso seja maior que a margem definida acarretará em falhas de atraso.

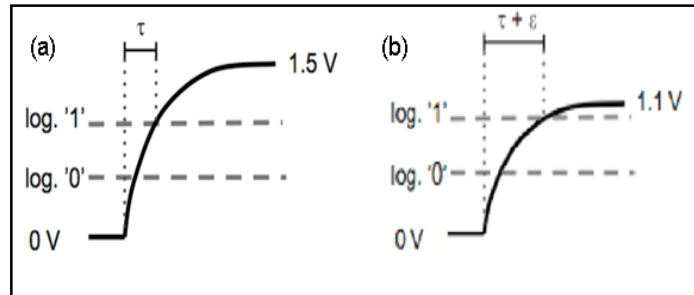


Figura 3 – (a) Curva exponencial original.

(b) Curva com diminuição da tensão. [22].

- **Modelo de falha de nível lógico** –É para analisar se há quedas de tensão na qual o nível lógico 1 não pode ser atingido acarretando em uma falha de nível lógico, dado que a tensão de *threshold* não diminui proporcionalmente junto com a tensão de alimentação. A figura 4 mostra o que de fato acontece nesse tipo de falha. Poderemos observar futuramente através de nossos experimentos que esse modelo de falha ocorre em nosso circuito síncrono em estudo.

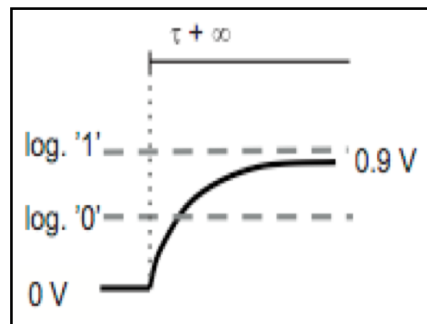


Figura 4 – Curva com tensão de 0.9V. [22].

2.4 MECANISMOS DE INJEÇÃO DE FALHAS

Técnicas de injeção de falhas é uma área do conhecimento de suma importância para o escopo desse trabalho, pois é através de uma dessas técnicas que realizamos toda a verificação da nossa proposta. É importante destacar também que a técnicas de injeção de falhas que é escolhida para o uso depende de fatores como a etapa do projeto na qual o sistema se encontra os tipos de falhas e níveis de controlabilidade esperado [24]. Vamos nos ater principalmente à técnica que será mais utilizada em nossa proposta.

As técnicas são subdivididas em: Injeção de falhas baseadas em simulações, que são utilizadas durante as fases iniciais de projeto, possuindo baixo custo; Injeção de falhas por software que é quando já se possui um protótipo e também não possui um contexto tão complexo e permite injeção de falhas acessíveis por software; E por fim, injeção de falhas por hardware que iremos citar com mais detalhes, sendo a injeção por hardware a utilizado no trabalho.

2.4.1 INJEÇÃO DE FALHAS POR HARDWARE

Este tipo de injeção de falhas acontece por meio físico através do uso de circuitos adicionais e pode ser realizada com contato como introduzindo ruído nas linhas de alimentação do circuito ou sem contato através de radiação [25].

A injeção de falhas sem contato pode ser realizada com a indução de correntes elétricas no circuito e dá-se o nome de interferência eletromagnética irradiada. Para a realização de experimentos desse tipo, são necessários uso de equipamentos específicos mais complexos e que possuem maior custo, sendo que a injeção de falhas sem contato é a mais complexa por ser difícil ter exatidão no ponto de recebimento da interferência.

A injeção de falhas por contato pode ser aplicada nas linhas de alimentação do circuito, ou diretamente nos pinos do circuito. As injeções ocorrem por meio de pontas de prova e por soquetes e são do tipo *pin-level*. Esse tipo de injeção é mais controlável, tendo que se ter sempre o cuidado para não danificar o circuito, como por exemplo, com tensões acima do que o circuito permite. Como na injeção sem contato há a necessidade da utilização de equipamentos externos, às vezes bem complexos e também com custo elevado [22]. Essa é uma técnica que permite atingir partes inatingíveis do sistema que outras técnicas citadas anteriormente não permitiam.

Cabe salientar que utilizaremos uma plataforma de geração de ruído por hardware que provocará distúrbio nas linhas de alimentação do circuito em teste. Pois a partir dessa técnica podemos simular com mais exatidão os problemas causados a esses tipos de circuitos, podendo assim avaliar melhor nossa técnica de projeto.

2.4.2 MEDIDAS DE DEPENDABILIDADE

Segundo **Pradhan** [15] as medidas de dependabilidade estão todas relacionadas à confiabilidade do sistema, dentre elas podemos citar as mais importantes como a MTBF (*Mean Time Between Failure*), MTTF (*Mean Time To Failure*), MTTR (*Mean Time To Repair*) e taxa de falhas.

A taxa de falhas de um sistema computacional é medida em defeitos por alguma unidade de tempo específica, essa taxa é proporcional ao tempo de vida do sistema e são determinadas de forma estatística. Neste trabalho, será utilizada esta leitura de taxa de falhas por unidade de tempo, com variação nas tensões de alimentação segundo definições de alguns *Standards*. A tabela 1 mostrada abaixo demonstra as principais medidas de dependabilidade.

Tabela 1 – Medidas [15].

MEDIDA	SIGNIFICADO
Taxa de falhas (failure rate, hazard function , hazard rate).	Número de falhas por tempo.
MTTF	Tempo até a primeira ocorrência de um defeito.
MTTR	Tempo médio para o reparo do sistema.
MTBF	Tempo médio entre os defeitos do sistema.

2.6 CONCLUSÕES DO CAPÍTULO

Conceitos importantes foram apresentados nesse capítulo no que refere - se aos circuitos tolerantes às falhas, os quais baseiam o estudo de caso e proposta, sendo uma das motivações do trabalho. O capítulo teve como principal objetivo apresentar os mais

importantes conceitos na área em questão, bem como os tipos de injeção de falhas existentes, dando ênfase aos assuntos que serão utilizados ao longo dessa dissertação de mestrado.

3 COMPATIBILIDADE ELETROMAGNÉTICA

Neste capítulo, serão abordados os principais conceitos referentes à Compatibilidade Eletromagnética (EMC - *Eletromagnetic Compatibility*) que estão mais relacionados à nossa proposta. O capítulo será iniciado com uma pequena introdução referente ao assunto. Serão mostrados também os conceitos principais da Interferência Eletromagnética (EMI - *Eletromagnetic Interference*) e os tipos de interferências eletromagnéticas do qual os equipamentos estão sujeitos. Também apresentaremos as principais normas que foram utilizadas na realização deste trabalho.

3.1 INTRODUÇÃO

No contexto atual, é possível observar que o uso sempre crescente de equipamentos eletrônicos, tais como celular, dispositivos *wireless*, televisão, rádio, *ebooks*, câmeras digitais, entre outros. Todos os circuitos eletrônicos como os que foram citados geram interferências eletromagnéticas EMI, o que faz com que os mesmos apresentem uma necessidade de aumento de tolerância a esses tipos de ruídos gerados.

Em mais detalhes, é necessário garantir que estes tipos de sistemas continuem operando sem erros independentes das condições as quais são expostos. Dentre essa abordagem dois conceitos são extremamente importantes. O primeiro é relacionado à emissão, sendo definido como o fenômeno no qual a interferência eletromagnética é emitida por uma determinada fonte geradora para outro sistema. O segundo conceito é chamado de susceptibilidade que é a análise do comportamento do sistema em meio à emissão de outros dispositivos para com ele [26]. A EMC pode ser definida pela capacidade de sistemas computacionais operarem de modo satisfatório em seu ambiente sem que ele sofra perturbações eletromagnéticas intoleráveis e sem que o mesmo não produza interferências acima das permitidas, para que não afete outros sistemas vizinhos [27].

Dado que um sistema não pode interferir nos demais sistemas vizinhos a ponto de interferir em seu funcionamento, é necessário esse controle e um estudo aprofundado sobre o tema. O estudo sobre EMI tem dois pontos de vista. A da emissão onde o foco

está nos níveis de interferência que o sistema gera para o meio e susceptibilidade que é o quanto esse sistema é afetado pelas interferências desse meio.

Grande parte da proposta tem como base os problemas causados pela EMI, que é quando a partir de uma fonte externa, o circuito se perturba por causa de emissões eletromagnéticas que são recebidas dessa fonte e que é definida pela IEC em [27]. Essas perturbações podem fazer com que aconteça um erro no sistema. O foco principal que iremos abordar é da interferência conduzida, que através do barramento de alimentação, o nosso circuito é afetado por uma fonte geradora que dê ruído [27]. Ocorre principalmente em corrente alternada, por alimentar de forma paralela a outros tipos de sistemas que gerem ruídos para o circuito.

3.2 CONCEITOS

Os conceitos descritos a seguir são definições apresentadas pela IEC (*International Electrotechnical Commission*) [27]:

- I. **Imunidade:** Capacidade de um dispositivo, equipamento ou sistema de executar suas funções na presença de certa perturbação, ruído ou interferência eletromagnética sem degradação de desempenho;
- II. **Emissão:** É o fenômeno no qual a energia eletromagnética (interferência ou distúrbio) emana de uma determinada fonte geradora para um dispositivo ou sistema vítima;
- III. **Nível de Emissão:** É o nível de interferência eletromagnética emitida a partir de um determinado dispositivo, equipamento ou sistema;
- IV. **Nível de Imunidade:** É 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;
- V. **Limite de Emissão:** É o nível máximo de emissão de uma fonte geradora de interferência eletromagnética;

- VI. **Limite de Imunidade:** É o nível de imunidade mínimo sobre um determinado dispositivo, equipamento ou sistema para que ele permaneça capaz de operar com grau de desempenho exigido;
- VII. **Limite de Interferência:** É o nível de interferência eletromagnética máxima admissível de um determinado dispositivo, equipamento ou sistema;
- VIII. **Susceptibilidade (eletromagnética):** É a incapacidade de um dispositivo, equipamento ou sistema executar suas funções na presença de uma perturbação eletromagnética;
- IX. **Nível de Compatibilidade Eletromagnética:** É o nível de interferência eletromagnética utilizado como referência para a fixação de limites de emissão e imunidade.

3.3 INTERFERÊNCIA ELETROMAGNÉTICA

Primeiramente é necessário definir o significado de um ambiente eletromagnético [27], do inglês (*Electromagnetic Environment – EME*) o qual é a união de todos os fenômenos e interações eletromagnéticas que ocorrem em um determinado meio. Um exemplo para isso seria um escritório com toda a sua rede elétrica, o tipo de instalação no qual estão inseridos, os equipamentos eletrônicos como computadores, notebooks, frigobar, dentre outros, e o próprio meio externo vizinho que interfere nesse ambiente.

A definição de interferência eletromagnética EMI segundo a IEC [27], é definida como um conjunto de perturbações que afetam um determinado circuito, dispositivo, sistema eletrônico por radiações eletromagnéticas emitidas a partir de uma fonte externa, o que pode levar o dispositivo a apresentar falhas, ter uma degradação ou se tornar inoperante.

3.3.1 FONTES DE INTERFERÊNCIA ELETROMAGNÉTICA

As fontes de interferência eletromagnética se subdividem em fontes naturais e fontes artificiais [27] [28]:

- I. **Fontes artificiais** – São os distúrbios eletromagnéticos ou interferências causadas pela ação do homem. Exemplos que podemos citar são motores elétricos que podem gerar interferência nas fontes de alimentação, lâmpadas eletrônicas, celulares e variadas máquinas industriais. A Figura 5 mostra uma figura clássica desse tipo de interferência, no caso uma interferência conduzida [29]. Já a figura 6 demonstra um exemplo de um computador gerando interferências para um aparelho de televisão.

- II. **Fontes naturais** – São distúrbios eletromagnéticos ou interferências causadas de forma natural por fenômenos naturais como o próprio ruído eletromagnético, descargas elétricas, dentre outros [27].

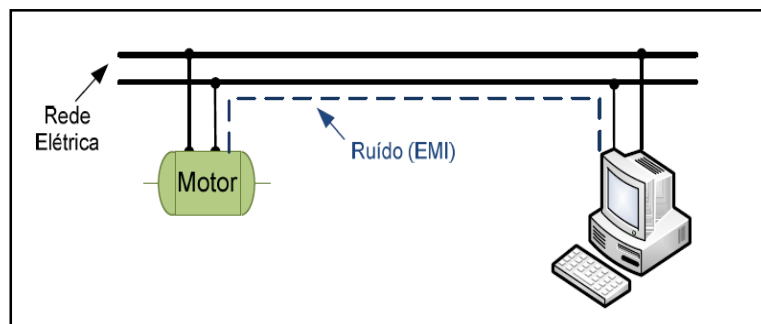


Figura 5 – Interferência conduzida [29].

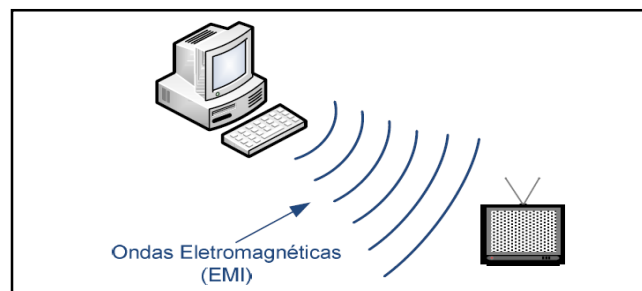


Figura 6 – Interferência radiada [29].

3.3.2 TIPOS DE INTERFÊRENCIA

O sistema eletrônico pode ser atingido por interferências eletromagnéticas de duas formas distintas conforme mostram as figuras 5 e figura 6, de forma conduzida ou de forma irradiada [30]. Em mais detalhes Interferência Eletromagnética pode ser classificada como:

- I. **Interferência eletromagnética conduzida** – A interferência é transferida da fonte geradora para vítima por meio dos condutores da alimentação elétrica, pela referencia de tensão ou pelos terminais de entrada e de saída. Esse tipo de ruído ocorre mais frequentemente pelas correntes alternadas, pois a rede elétrica alimenta vários circuitos ditos ruidosos [30];
- II. **Interferência eletromagnética irradiada** – Este tipo de interferência é diferente da anterior devido ao seu meio de propagação, pois ela emite da fonte geradora para o circuito vítima pelo ar por ondas eletromagnéticas. Elementos como cabos e trilhas do circuito agem como antenas gerando ruído para o sistema vítima [30].

A figura 7 mostra exemplos dos dois tipos de interferência em um ambiente eletromagnético.

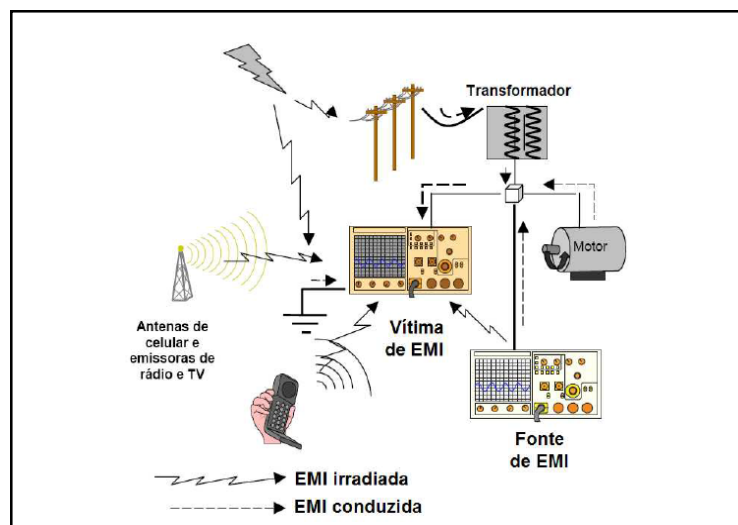


Figura 7–Exemplos de interferências [31]

3.4 INTERFERÊNCIAS ELETROMAGNÉTICAS NA ELETRÔNICA

Nourani define a integridade de sinal (*Signal Integrity*) como sendo a característica ou habilidade que um sinal possui para gerar respostas corretas em um circuito ou sistema eletrônico [31]. Dado isso, para que um sistema eletrônico opere como foi projetado, é necessário que seus sinais elétricos apresentem características, como níveis de tensão, tempos de subida e descida compatíveis com sua especificação de projeto.

A figura 8 abaixo exemplifica os sinais e as características necessárias no que diz respeito a integridade dos mesmos. Em vermelho temos um sinal que não se encaixa nas especificações de projeto, em pontilhado a sua representação ideal, mas que é impossível devido às características dos circuitos. Finalmente a curva em preto demonstra o sinal real dentro das especificações.

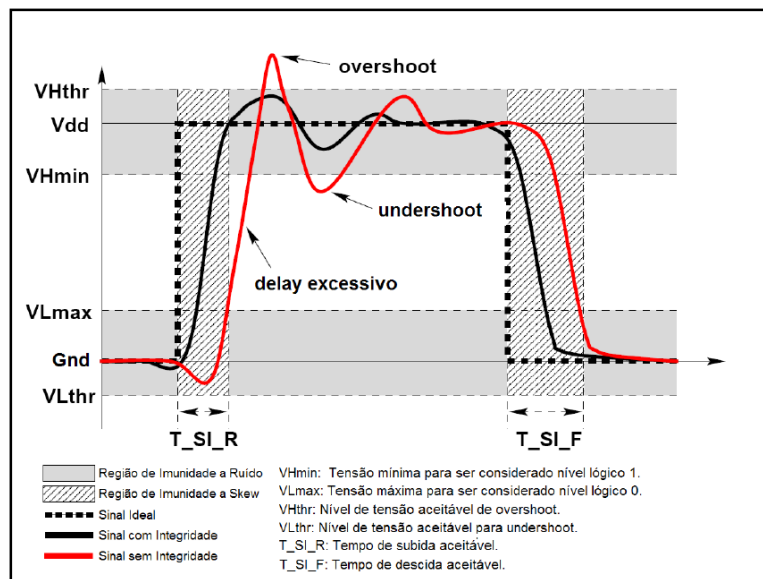


Figura 8 – Exemplo de integridade de sinal [31].

Mas como foi citado anteriormente, em um ambiente eletromagnético, temos diversos equipamentos gerando ruídos e interferências externas. A própria instalação elétrica que contribui ainda mais com as interferências, contribuindo assim para emissões indesejadas no meio e com isso causando efeitos como:

- I. **Ruído nas linhas de alimentação** – Devido ao rápido chaveamento que ocorrem nos circuitos, ocorrem variações de corrente que geram ruídos nas linhas de alimentação [33];
- II. **Flutuação nas linhas de alimentação**—É a redução temporária nos níveis de tensão nas linhas de alimentação (Vcc) de um sistema eletrônico comum, devido a precariedade que temos em nossa rede elétrica [34] [35];
- III. **Skew**— É a diferença nos tempos da propagação de dois ou mais sinais transmitidos através da rede de distribuição de um sistema computacional [36] [37];
- IV. **Ground Bounce** – São elevações da tensão que ocorrem na tensão Gnd de um sistema eletrônico[38].

3.5 NORMALIZAÇÃO

Com o grande aumento de sistemas eletrônicos que geram ruídos e interferências eletromagnéticas, foi necessário buscar uma padronização e uma organização no que diz respeito a essas emissões. Com isso, com o objetivo de realizar uma cooperação de forma internacional na área eletroeletrônica, surgiu uma organização de cunho mundial denominada IEC, Comissão Eletrotécnica Internacional (*International Electrotechnical Commission*).

Essa organização reúne tecnologias de magnetismo, eletromagnetismo, eletroacústica, telecomunicações, dentre outras. Ela também é responsável pelas normatizações eletrotécnicas internacionais.

Neste trabalho de mestrado estudaremos a família de normas IEC 61.000, pois essa norma tem o objetivo de abordar assuntos básicos sobre EMC. A tabela 2 mostra nove partes na qual essa norma está subdividida em padrões e relatórios técnicos. Esta dissertação utilizou os itens 17 e 29 da parte 4, que enfatizaremos a seguir.

Tabela 2 – Padrões IEC 61.000

Parte 1	Considerações gerais (introdução, princípios fundamentais, segurança definições e terminologia).
Parte 2	<i>Descrição e classificação do ambiente e níveis de compatibilidade</i>
Parte 3	<i>Limites de emissão e imunidade</i>
Parte 4	<i>Técnicas de medição e ensaio</i>
Parte 5	Guias de instalação, métodos de mitigação e dispositivos
Parte 6	Padrões genéricos
Parte 7	Em aberto
Parte 8	Em aberto
Parte 9	Diversos

3.5.1 NORMA IEC 61.000-4-17

Equipamentos eletrônicos em geral estão sujeitos a ondulações de tensão, também conhecido como *Ripples*, tanto em sistemas com alimentação em redes elétricas, quanto proveniente de carregadores auxiliares de bateria. Esta norma determina características para ensaios de ondulações de tensão [39].

A tabela 3 demonstra os níveis de tensão que são definidos para ensaios em fonte de tensão DC.

Tabela 3 – Definições de *Ripple*

Nível	Tensão nominal (%)
1	2
2	5
3	10
4	15
X	Valor qualquer definido em projeto

A figura9 mostra um *Ripple*, onde temos a tensão nominal DC indicada na figura por $U_{d.c.}$, a amplitude de tensão do *Ripple* sendo $U_{max} - U_{min}$. Como exemplo, caso utilizemos a norma no nível 2 com um percentual de 2%, nós teremos U_{max} como sendo a tensão nominal $DC+1\%$ e U_{min} como sendo $DC-1\%$.

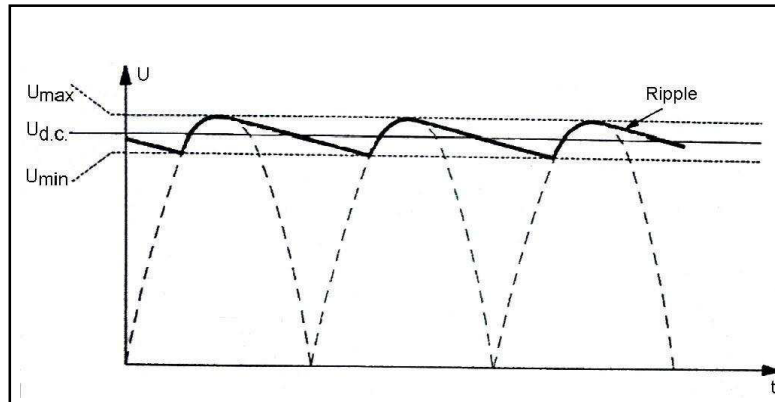


Figura 9 – Ripple.

Um cuidado que devemos ao fazer ensaios com ripple, está relacionado ao U_{max} , pois ele não pode ser maior a ponto de danificar o circuito, então temos esse valor limitado por um valor que não prejudique o circuito em teste.

3.5.2 NORMA 61000-4-29

Esta norma tem como principal objetivo estabelecer um método básico para a realização de testes de imunidade de equipamentos ou dispositivos eletrônicos alimentados por fontes de corrente contínua externas de baixa tensão. Há diversos tipos de distúrbios e ruídos relacionados às linhas de alimentação de equipamentos e dispositivos eletroeletrônicos. Tendo isso em vista, iremos fazer uma pequena definição de queda de tensão, pequena interrupção e variação de tensão, mostrando posteriormente o que a norma sugere quanto a isso.

- I. **Queda de tensão** – A queda de tensão se caracteriza por uma súbita redução na tensão de alimentação do dispositivo eletrônico, seguido do retorno de sua tensão nominal em um curto período de tempo [40]. A tabela 4 apresenta os níveis de tensão percentuais relativos à tensão nominal (U_T) e o tempo sugerido pela norma em quedas de tensão.

Tabela 4 – Níveis de tensão e duração recomendada [40]

Teste	Nível de tensão (%)	Duração (s)
Pequena Interrupção	0	0.001
		0.003
		0.01
		0.03
		0.1
		0.3
		1
		Especificação do produto

II. Pequena interrupção – Caracteriza-se pelo desaparecimento momentâneo da tensão por um período de tempo que não exceda um minuto, pois quedas de tensão maiores que 80% são consideradas interrupções [40]. A tabela 5 apresenta os níveis de tensão percentuais relativos a tensão nominal (UT) e o tempo sugerido pela norma pequena interrupção.

Tabela 5 – Níveis de tensão e duração recomendada [40]

Teste	Nível de tensão (%)	Duração (s)
Queda de Tensão	40 a 70 ou X	0.01
		0.03
		0.1
		0.3
		1
		Especificação do produto

III. Variação de tensão - Consiste na mudança gradual da tensão nominal de alimentação para um valor acima ou abaixo, podendo ser de duração curta ou longa. A tabela 6 apresenta os níveis de tensão percentuais

relativos à tensão nominal (UT) ao tempo sugerido pela norma em variação de tensão.

Tabela 6 – Níveis de tensão e duração recomendada [40]

Teste	Nível de tensão (%)	Duração (s)
Variação de Tensão	85 a 120	0.01
	ou	0.03
	80 a 120	0.1
	ou	0.3
	X	1
		Especificação do produto

3.6 CONCLUSÕES DO CAPÍTULO

O objetivo foi apresentar os conceitos mais importantes relacionados ao nosso trabalho no que se refere à compatibilidade eletromagnética. Discorremos a respeito do crescimento do número de aparelhos que geram interferências eletromagnéticas, deixando ambientes eletromagnéticos cada vez mais ruidosos. Com isso, houve a necessidade de uma maior pesquisa na área em questão, buscando modelos que simulam as emissões geradas por esses sistemas computacionais.

No final do capítulo foram apresentadas as normas existentes e que serão utilizadas nessa dissertação de mestrado, que têm como objetivo a padronização dos componentes eletrônicos no que se refere à compatibilidade eletromagnética.

4 CIRCUITOS ASSÍNCRONOS

Este capítulo apresenta alguns conceitos básicos relacionados ao desenvolvimento desse trabalho. Primeiramente teremos uma breve introdução referente aos circuitos assíncronos, tratando posteriormente dos conceitos básicos referentes ao assunto e à dissertação.

4.1 INTRODUÇÃO

Atualmente observa-se que a grande maioria dos projetos, projetistas e sistemas de projeto de sistemas digitais VLSI atualmente lidam apenas com o paradigma síncrono. Os projetistas têm pouquíssima familiaridade com o paradigma assíncrono. Dada a natureza dos circuitos síncronos, o caminho crítico de cada parte do circuito deve ser respeitado considerando seus atrasos máximos. Esse tipo de abordagem apresentam algumas limitações no que diz respeito à geração de ruído, maior consumo de potência e escorregamento do sinal de relógio, todos oriundos do seu sinal de relógio. Sistemas digitais síncronos pressupõem um modelo discreto no tempo, enquanto sistemas assíncronos assumem um modelo contínuo de tempo.

O avanço atual do paradigma assíncrono pode ser observado a partir de uma análise dos últimos anos. Basicamente surgiram seis processadores no estilo assíncrono, todos desenvolvidos no âmbito acadêmico, sendo um deles uma versão comercial. Estes apresentam vantagens claras sobre versões síncronas como a taxa de potência dissipada que é medida em MIPS (*Millions of Instructions Per Second*) por Watt. Devido a sempre crescente necessidade de circuitos que apresentam baixo consumo de energia (*Low Power*) e operando com baixa tensão de alimentação. O uso ligado a circuitos síncronos começa a exigir maiores cuidados na elaboração de seus projetos referentes às técnicas de distribuição do sinal de relógio, controle de frequência e escorregamento de sinal [2].

A proposta assíncrona se justifica a medida que permitem a superação de inconvenientes do estilo síncrono sem perdas significativas intrínsecas das quais são associadas aos circuitos síncronos. Com isso, a pesquisa relacionada a circuitos assíncronos vem aumentando consideravelmente, como uma alternativa aos circuitos

síncronos. Os circuitos assíncronos apresentam como principais benefícios não possuírem um sinal global de relógio, desempenho baseado na média e não no pior caso, maior adaptação a variações do ambiente, como tensão e temperatura, baixo consumo de energia e baixa emissão de ruído [91]. O projeto assíncrono não é algo muito disseminado, pois possuem algumas desvantagens como seu projeto ser mais complexo que o projeto síncrono, não há muitas ferramentas CAD disponíveis no mercado para poder trabalhar com sistemas assíncronos e por possuir poucos projetistas preparados para lidar com esses conceitos.

Vale a pena introduzir uma definição estrutural dos circuitos digitais notando que a definição estrutural do circuito síncrono nada mais é que um caso especial de circuito assíncrono. Essa definição foi retirada do livro de **Lavagno et. al.**[92]:

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

A definição mostrada impõe restrições sobre a estrutura interna de circuitos assíncronos, já para circuitos síncronos ela coloca restrições sobre a estrutura interna e sobre o ambiente onde o circuito opera. A figura 10 demonstra exemplos das restrições que circuitos síncronos devem respeitar em relação ao relógio, apresentando dois exemplos de violação.

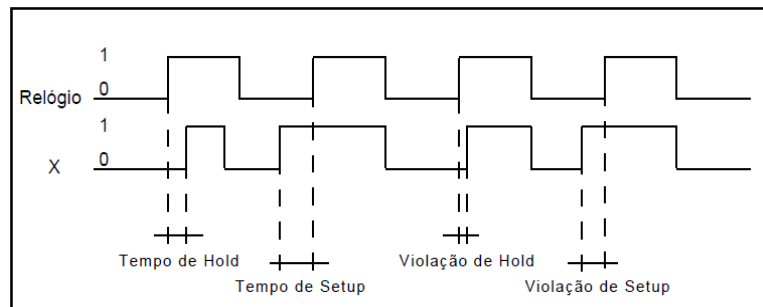


Figura 10–Restrição a circuitos síncronos.

4.2 PRINCÍPIOS DE FUNCIONAMENTO DO PARADIGMA ASSÍNCRONO

Temos como a principal diferença entre circuitos síncronos e assíncronos sendo relacionada à forma de gestão de dados. Os princípios de funcionamento de cada um dos paradigmas é apresentado a seguir:

- I. **Funcionamento do circuito síncrono** –O circuito síncrono tem como controle o sinal de relógio global. Esse sinal estabelece a sincronização do sistema, onde os registradores irão captar os dados. O tempo determinado pelo relógio deve respeitar o caminho crítico do circuito lógico, considerando atrasos máximos, sendo assim, o tempo é definido pela frequência do sinal de relógio [6]. A figura 11 apresenta um circuito síncrono genérico.

Dada uma convenção, os componentes seqüenciais operam na transição positiva do sinal de relógio, ou seja, transição no nível lógico zero para um, ou transição negativa de sinal do relógio, ou seja, transição do nível lógico um para nível lógico zero. Os dados são liberados periodicamente através do sinal de relógio.

- II. **Funcionamento do circuito assíncrono** – O circuito assíncrono, como dito anteriormente, não possuindo relógio global, possui a rede de controladores que faz o uso de canais de comunicação fazendo a detecção da presença de dados em suas entradas e saídas, possuindo assim protocolos de comunicação de solicitação e reconhecimento (*request, acknowledge*), ou chamados também de protocolos de *handshaking*. Temos na figura 12 um circuito assíncrono genérico.

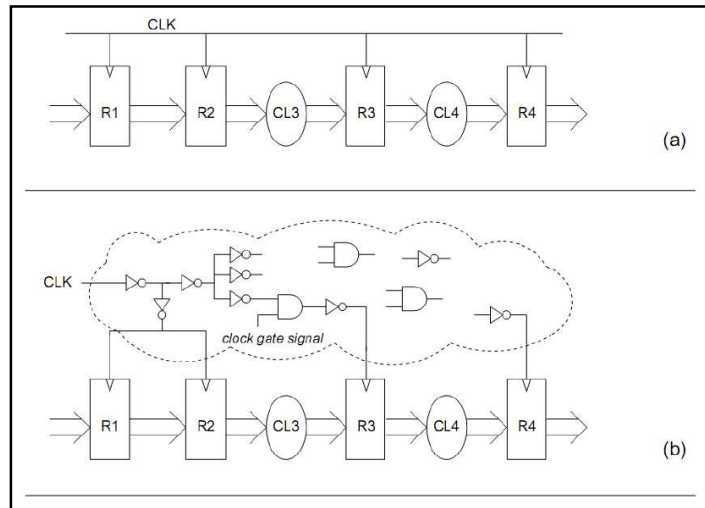


Figura 11 – Circuito síncrono genérico [6]

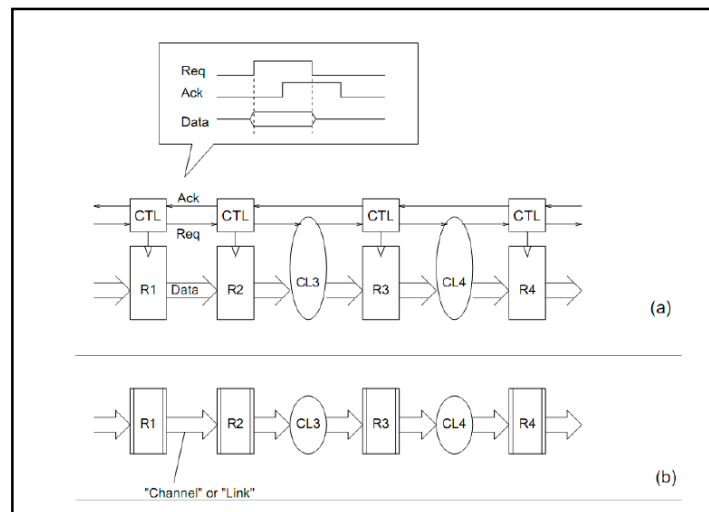


Figura 12– Circuito assíncrono genérico.

Como podemos facilmente perceber, temos uma única diferença entre os dois circuitos: A substituição do relógio global por protocolos de comunicação de *handshaking*. A figura 13 apresenta os controladores mestre-escravo de forma genérica. Quando o controlador mestre termina de processar os dados, ele envia um sinal de requisição (*request*) ao controlador escravo para indicar a presença de um dado válido disponível. Se o controlador escravo estiver ocupado, ele não responderá, mas caso esteja disponível, ele fará então o processamento dos dados e vai indicar ao controlador mestre por um sinal de reconhecimento (*acknowledge*) que pode disponibilizar um novo

dado. Portanto, os elementos são comandados localmente e a validade dos dados também deve ser assegurada localmente. Tais operados assíncronos apresentados além de se comunicar entre si, se comunicam com outras partes do circuito. Existem dois protocolos de comunicação, mas que serão tratados em outras sessões.



Figura 13 – Controladores mestre escravo.

Algumas características dos operadores assíncronos apresentados na figura acima são:

- I. **Latência** – É dado pelo tempo necessário para processar um dado de entrada e disponibiliza-lo em sua saída. É igual ao tempo gasto através dos circuitos combinacionais.
- II. **Ciclo** – É dado pelo tempo necessário para processar um sequência de dois dados sucessivos.
- III. **Profundidade do pipeline** – É dado pelo número de dados que um operador pode memorizar.

4.3 FENÔMENOS TEMPORAIS

Transitórios (*hazards*) são exemplos de fenômenos temporais que podem afetar o funcionamento de circuitos assíncronos. Os transitórios em circuitos síncronos podem ocorrer desde que no instante da amostragem, todos os valores estejam estáveis. Em sistemas assíncronos, um valor transitório pode levar o circuito a apresentar um estado inválido ou indesejado [93]. Eles são devido à transição dos sinais de entrada ou ao

atraso existente nos fios ou nas portas lógicas do circuito. Porém existe algumas técnicas que podem garantir o desenvolvimento de circuitos combinacionais com comportamento livre de transitórios.

Os transitórios são classificados como:

- I. **Transitório estático** – Ocorrem quando temos um sinal que deveria permanecer constante, mas altera o seu valor por um momento retornando ao seu valor original. Podem ocorrer nas saídas de portas lógicas ou quando dois sinais de entrada alteram seu valor simultaneamente;
- II. **Transitório dinâmico** – Ocorrem quando um sinal deveria trocar de nível lógico somente uma vez, mas troca mais de uma vez;
- III. **Transitório combinacional** – Ocorrem quando aparece um funcionamento que não é esperado. Podem ser suprimidos na fase de especificação e implementação;
- IV. **Transitório seqüencial** – Ocorrem devido aos circuitos de realimentação.

Devemos nos assegurar que os operadores de circuitos assíncronos sejam livres de transitórios e tenham a capacidade de indicar a validade dos dados de saída, sinalizando o fim de seu processamento de dados por um sinal de requisição (*request*). Para assegurar essa estabilidade em propostas assíncronas é fazendo restrições de tempo, de modo que os transitórios sejam tolerados de forma a não comprometer o bom funcionamento do sistema.

Outro fenômeno temporal que pode prejudicar o bom funcionamento dos circuitos digitais é a metaestabilidade. É um fenômeno difícil de ser tratado em projetos assíncronos, sendo um fenômeno que pode ocorrer em dispositivos de armazenamento [94]. Quando um dado a ser registrado em um elemento de armazenamento altera o seu valor simultaneamente ou muito próximo a transição do sinal que habilita a amostragem, a saída pode apresentar problemas. Isso ocorre caso as restrições de tempos de *setup* e *hold* do elemento de memória não forem respeitadas [94].

4.4 INTERAÇÃO COM O AMBIENTE

Uma etapa fundamental para a determinação de qual estilo de projeto assíncrono adotar é sobre a imposição de restrições à interação entre o circuito e seu ambiente. Quanto menor o conjunto de restrições impostas a essa interação, mais complexo é o processo de projeto. A interação do circuito com o seu ambiente são classificados em três categorias:

- I. **Modo Fundamental** –Quando um circuito parte de um estado estável, ou seja, onde todos os sinais de entrada, saída e internos estão estáveis, é permitida a mudança de um único sinal de entrada. Após essa mudança é necessário que todos os sinais alcancem novamente um estado estável antes que o ambiente possa realizar outra mudança unitária de alguma entrada do circuito;
- II. **O modo rajada (*burst mode*)** –O modo rajada nada mais é do que uma extensão do modo fundamental. Ele é uma forma restrita de permitir múltiplas alterações de entradas e saídas. Quando um circuito em modo rajada se encontra em um estado estável, ele aguarda por um conjunto definido de alterações nas suas entradas. As alterações desse conjunto podem acontecer em ordem arbitrária. Quando o conjunto de alterações na entrada é observado, ele computa um conjunto de saídas. O ambiente não pode gerar nenhuma nova alteração na sua entrada até que o circuito se estabilize, ou seja, gere o conjunto de saídas e estabilize seus sinais internos. Para funcionar de acordo com as restrições de ambiente, tanto circuitos operando em modo fundamental quanto circuitos operando em modo rajada utilizam elementos de atraso em seus sinais de realimentação. Isso ocorre por que esses sinais são vistos como novas entradas e devem obedecer o tempo de estabilização do circuito;
- III. **Modo entrada e saída** –Com esse modo é permitida a alteração de mais de um sinal simultaneamente. O circuito opera sobre essas entradas e pode gerar ou não sinais de saída correspondentes. Assim que o ambiente observa

que a saída correspondente foi definida, esse pode gerar novas entradas. Uma forma de o circuito interagir com o ambiente no modo entrada e saída é utilizando um protocolo de *handshake*, onde cada conjunto de entradas gerado pelo ambiente é assinalado com um sinal de requisição (*request*). Uma vez que as entradas são processadas, o circuito responde com um sinal de reconhecimento, o qual é usado pelo ambiente como sinalização para geração de novos dados.

4.5 CLASSIFICAÇÃO DOS CIRCUITOS ASSÍNCRONOS

Projetar circuitos assíncronos livre de transitórios, consiste em controlar as hipóteses temporais que são adotadas na etapa de concepção e elaboração do projeto do circuito assíncrono. Circuitos assíncronos são classificados conforme as diferentes hipóteses de atrasos e seus modos de funcionamento. A figura 14 mostra a classificação dos circuitos assíncronos e são definidas a seguir.

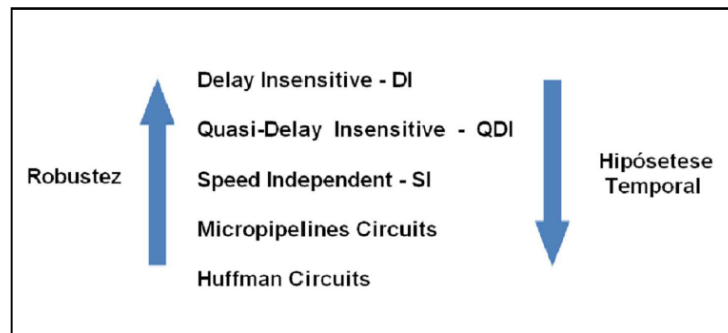


Figura 14 – Classificação assíncrona.

- I. **Delay Insensitive-** São chamados de circuitos insensíveis ao atraso e funcionam corretamente independentes dos atrasos nas portas e fios, onde podem apresentar atrasos finitos de magnitude arbitrária. O operador que recebe os dados deve informar ao emissor o recebimento dos dados. Ocorre um aumento considerável de área de silício nessa proposta, pois para construção deste tipo de circuito é necessário ter considerações bem restritas e a utilização de estruturas de porta complexas [6].

- II. **Quasi-Delay Insensitive** – São chamados de circuitos quase insensíveis ao atraso e tem como objetivo a redução da complexidade no circuito. É introduzido certas considerações nas bifurcações (*forks*), sendo chamadas de isocrônicas (*isochronic*), considerando os atrasos entre as conexões iguais. É um modelo de portas simples sob a hipótese temporal, o que permite implementar com o uso de portas padrões. A figura 15 mostra essas bifurcações isocrônicas [6].

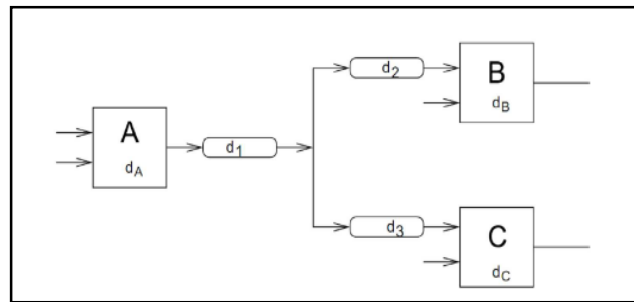


Figura 15 – *Isochronic forks* [93].

- III. **Speed independent** – São os circuitos independentes da velocidade, que assume atrasos infinitos, mas de magnitude arbitrária nas portas lógicas, e os fios como elementos ideais. Este modelo é equivalente ao modelo anterior, mas é impossível sua implementação com a tecnologia atual, dada a hipótese sobre os *forks* com um atraso nulo sobre os fios [6].
- IV. **Circuitos Micropipelines** – Apresentados por **Sutheland** [95]. A parte de controle gerencia o fluxo de dados enquanto os circuitos combinacionais processam os dados. Nos sinais de requisição são inseridos atrasos para que se respeite o tempo de latência do circuito combinacional. Este modelo permite a construção de um pipeline elástico, já que o número de dados presente no pipeline pode variar. A figura 16 mostra um exemplo genérico deste modelo [6].

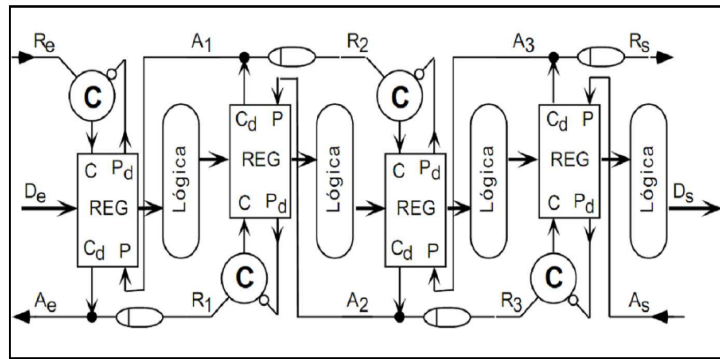


Figura 16 – Circuito micropipeline [6].

- V. **Circuito de Huffman** – É conhecido por circuito de modo fundamental, sendo o seu atraso idêntico ao de circuitos com o paradigma síncrono, supondo que os atrasos em todos os elementos são nos terminais ou de valores conhecidos. O controle do caminho dos dados é feito pelo uso de relógios locais, onde o tempo é estimado pelo tempo de latência máxima do seu circuito combinatório. Supõe-se que os sinais nas entradas irão alterar apenas quando o circuito está em uma condição estável, ou seja, quando não houver mudanças nos sinais internos da rede [6].

4.6 MODELAGEM DE ATRASO PARA CIRCUITOS

Os modelos de atrasos estabelecidos são base para os projetos assíncronos. A figura 17 apresenta os modelos de atrasos existentes:

- I. **Atraso de realimentação** – Existe pelo menos um elemento de atraso para cada laço de realimentação;
- II. **Atraso de porta** – Existe um elemento de atraso em cada saída da porta lógica;
- III. **Atraso de fio** – Existe um elemento de atraso em cada entrada da porta lógica.

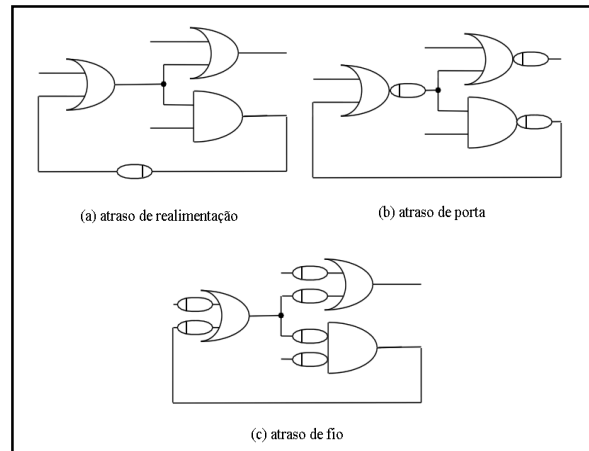


Figura 17 – Modelos de atraso [93].

4.7 PROTOCOLOS DE COMUNICAÇÃO

O protocolo de comunicação assíncrono mais comum e simples é o *handshake* como dito anteriormente. Ele utiliza dois sinais: *Orequest* e *acknowledge*, que controla o processo de transmissão de dados ou de sincronização. Com base no número de sinalizações envolvidas no protocolo de handshake ele pode ser de duas fases sendo sensível a transição do sinal de controle, ou de quatro fases sendo sensível ao nível do sinal de controle:

- **Protocolo de quatro fases** – Inicia uma transmissão com a requisição de transmissão (req+) e eventual disponibilização do dado. Quando o receptor recebe esta requisição, ele processa e sinaliza através de um reconhecimento (ack+). O transmissor retira a requisição (req-) que quando percebida pelo receptor, faz com que ele retire o reconhecimento (ack-) [6] [93]. A figura 18 ilustra esse protocolo.

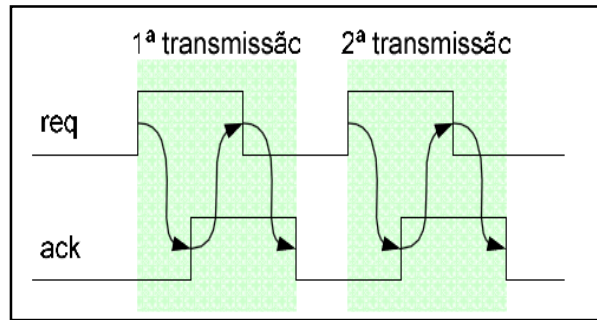


Figura 18 – Protocolo de quatro fases.

- **Protocolo de duas fases** – Já o protocolo de duas fases possui metade das transições do protocolo de quatro fases. O transmissor sinaliza o início da comunicação invertendo o valor do sinal de requisição (req+, req-). O receptor, quando processa o pedido, responde invertendo o sinal de reconhecimento (ack+, ack-) [6] [93]. A figura 19 ilustra esse protocolo.

Cada sinalização mostrada em cada um dos protocolos acima é abstraída nas bibliografias como a movimentação de uma ficha (*token*). No protocolo de duas fases cada transmissão é caracterizada pela movimentação de duas fichas (req+, ack+) ou (req-, ack-), sendo que os protocolos (req) e (ack) fluem em sentidos opostos. Já o protocolo de quatro fases se caracteriza pela movimentação de quatro fichas (req+, ack+, req-, ack-), sendo que as fichas (req+) e (req-) fluem em um sentido oposto das fichas (ack+) e (ack-) [6].

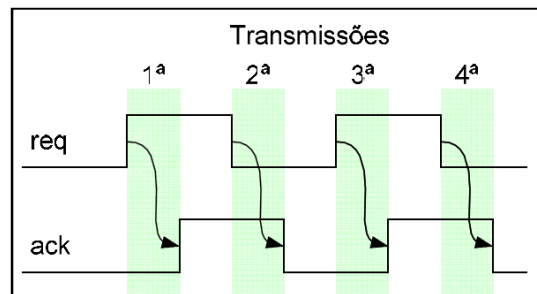


Figura 19 – Protocolo de duas fases.

4.8 CODIFICAÇÃO DOS DADOS

A codificação mais utilizada é conhecida como codificação de trilha única (*single rail*). No paradigma assíncrono, a utilização do protocolo de *handshake* em conjunto com dados codificados em trilha única é chamada de protocolo *bundled data*. O sinal de requisição é responsável por sinalizar a validade dos dados transmitidos. Para o funcionamento desse protocolo é necessário que o atraso do sinal de requisição seja maior ou igual ao atraso de todos os sinais de dados. Então temos que o sinal de requisição só pode acontecer após os dados válidos ficarem estáveis na entrada do receptor. A figura 20 mostra a codificação de trilha única.

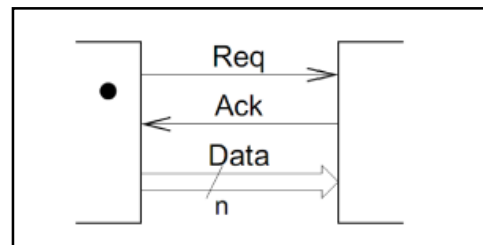


Figura 20 – Codificação de trilha única [6]

Com o objetivo de eliminar a restrição de temporização imposta ao sinal de requisição em um canal *bundled data*, projetistas exploram possibilidades de codificação que são capazes de carregar o dado juntamente com sua validade, o que permite desenvolver protocolos de comunicação onde não há a necessidade de impor restrições temporais na troca dos dados.

Já na codificação de trilha dupla, outra forma muito utilizada, onde os sinais são utilizados para cada bit de dados da representação binária. Um modo de representar um bit de dados com a utilização de trilha dupla associado a um protocolo de quatro fases é mostrado na figura 21. Quando temos a utilização de dados de trilha dupla associada ao protocolo de quatro fases, sempre teremos entre dois dados válidos, um espaçador.

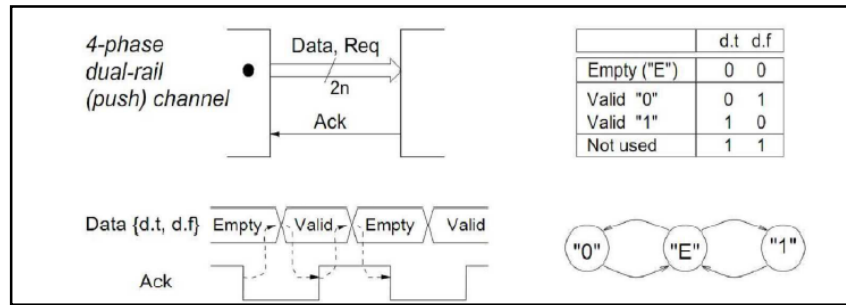


Figura 21 – Codificação de trilha dupla em um protocolo de quatro fases [6].

Outra forma de codificar dados em trilha dupla é utilizando o protocolo de *handshake* de duas fases. Uma das diferenças para a outra forma é que não precisamos inserir espaçadores entre os dados válidos. O funcionamento dessa abordagem requer que para mudar de um dado válido x para um dado válido y , não se alcance nenhum valor intermediário que represente outro dado válido, fazendo o uso da noção de fases. Há duas formas de representar esse tipo de codificação:

- I. **Por valor** – Cada fio é dedicado a um valor e cada transição indica que um novo dado está válido. Na figura 22 abaixo, o fio d.f é dedicado ao valor lógico zero, o fio d.t ao valor lógico um, assim um evento sobre d.f indica que existe um novo valor lógico zero e um evento sobre d.t indica que existe um novo valor lógico igual a um. A figura 22 demonstra essa codificação;

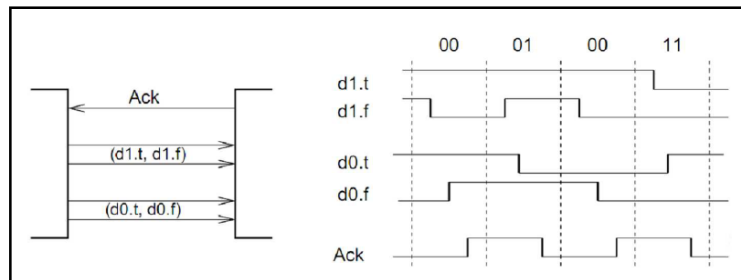


Figura 22 – Codificação de trilha dupla por valor de duas fases [6].

- II. **Por paridade** – Um fio é dedicado para o valor e outro dedicado para a paridade. Caso o fio do valor altere indica que há um novo dado com o valor alterado. Caso o fio de paridade se altere, indica que existe um

novo dado mas o valor não alterou. Na figura 23 abaixo temos um exemplo. O fio d.f é dedicado ao valor lógico e d.t é dedicado a paridade.

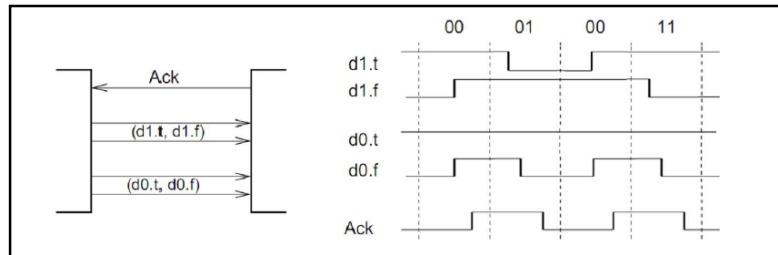


Figura 23 – Codificação de trilha dupla por paridade de duas fases [6].

4.9 O C-ELEMENT DE MULLER

Um componente extremamente importante em circuitos assíncronos e para o entendimento do nosso trabalho é o *C-element* de Muller [96]. Esse componente funciona como um sincronizador de eventos, produzindo um evento na sua saída quando todas as suas entradas recebem eventos específicos [95]. Este elemento tornou-se indispensável na implementação dos controles dos circuitos assíncronos. A tabela 7 mostra a tabela verdade do elemento *C-muller*.

Tabela 7 – Tabela verdade *C-element*.

a	b	Z_i
0	0	0
0	1	Z_{i-1}
1	0	Z_{i-1}
1	1	1

Na tabela 7 acima, nós temos a e b como sinais de entrada e Z_i sendo o sinal de saída. O *C-element* gera zero em sua saída quando as duas entradas são iguais a zero, um na saída quando as duas entradas são iguais a um, e mantém o valor anterior para qualquer outra configuração de entrada. Na figura 24 temos um exemplo de implementação *C-elemente* sua representação [6].

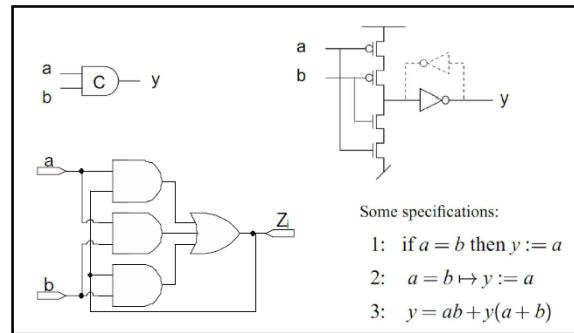


Figura 24 – Elemento C-muller [6].

4.10 CONCLUSÕES DO CAPÍTULO

O capítulo teve como objetivo apresentar as principais características referentes aos circuitos assíncronos, introduzindo os principais conceitos relativos a essa área do conhecimento. Foi realizada referência também às desvantagens do uso desse tipo de paradigma, motivando projetistas a desenvolverem propostas que atenuem as dificuldades encontradas em circuitos assíncronos. Os conceitos que aqui foram abordados serão necessários para o entendimento do capítulo que segue e que discorre a respeito da dessincronização de circuitos puramente síncronos.

5 TÉCNICA DE DESSINCRONIZAÇÃO

Neste capítulo será apresentada uma técnica de dessincronização proposta por **Cortadella, A. Kondratyev, L. Lavagno e C.P. Sotiriou** [12]. Primeiramente será apresentada uma breve introdução a respeito do tema e, então, serão mostradas as etapas no processo de dessincronização, apresentando suas características e dando ênfase nos assuntos principais que envolvem a dissertação e a premissa de se manter o fluxo de equivalência. Finalmente será apresentado como a rede é interligada, apresentando os benefícios do uso dessa técnica em sistemas que necessitam uma maior confiabilidade.

5.1 INTRODUÇÃO

Atualmente circuitos assíncronos não são utilizados em larga escala por dois motivos principais: O primeiro é devido ao fato de que os projetistas teriam que mudar sua mentalidade, fundamentada toda ela em circuitos síncronos [12], e o segundo é a falta de ferramentas CAD para o desenvolvimento desse tipo de circuito. Dentre os circuitos que foram desenvolvidos com o paradigma assíncrono e com a utilização de ferramentas CAD dedicadas, podemos citar ASPRO [97], MiniMIPS [98]. Mas como citado anteriormente, elas exigem o uso de *Computer-Aided Design-CAD*, que são ferramentas dedicadas como TAST [99], CAST[100], Balsa [101].

A técnica de dessincronização é um novo paradigma que tem como objetivo automatizar o processo de transformação de um circuito síncrono, para um circuito assíncrono equivalente, tendo em vista se desfazer das habilidades especiais e ferramentas dedicadas que o paradigma assíncrono exige.

Uma dessas técnicas é apresentada transformando circuitos síncronos que com suas características síncronas em circuitos assíncronos equivalentes [12]. Essa técnica propicia a utilização das ferramentas convencionais utilizadas em circuitos síncronos. Essa técnica é apresentada servindo como base para o tema da dissertação no qual é utilizado um circuito que foi dessincronizado por esse método. Uma das maiores preocupações dos desenvolvedores dessa técnica é que o circuito assíncrono resultante irá possuir a mesma sequência de dados do circuito síncrono utilizado, conhecido como fluxo equivalente.

A figura 25 mostra uma comparação entre o fluxo convencional e o fluxo de dessincronização. Em primeiro lugar, a especificação Verilog foi sintetizada utilizando *Synopsys Design Compiler* e vetores de testes foram gerados. Em seguida o netlist síncrono foi dessincronizado usando a ferramenta de dessincronização *drdesync*. Foi utilizada a mesma ferramenta para execução física de ambos designs. Finalmente, a simulação funcional foi realizada utilizando o ambiente Cadence Verilog-XL Simvision [102].

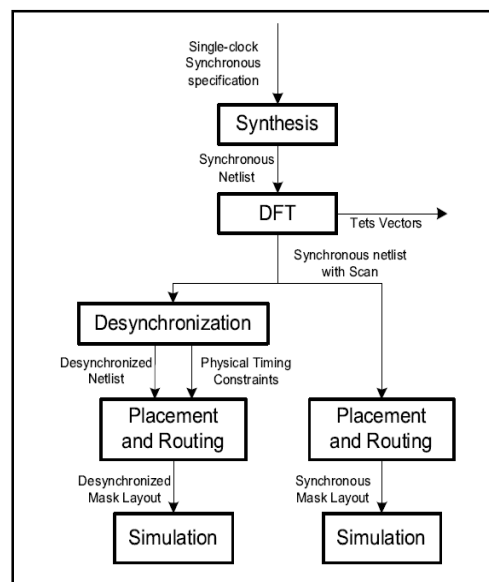


Figura 25: Comparação dos fluxos. [102]

5.2 MODELO DE DESSINCRONIZAÇÃO

O modelo de dessincronização que será apresentado tem como objetivo a substituição do relógio global por um conjunto de controladores assíncronos que garantem ao sistema um comportamento equivalente. O modelo pressupõe que o circuito possua blocos com a lógica combinacional e registradores implementados com Flip-Flop do tipo D, estes trabalhando na mesma frequência de relógio. A figura 26 possui um exemplo genérico onde temos FF como sendo Flip-Flop D e CL como a lógica combinacional e seu sistema dessincronizado.

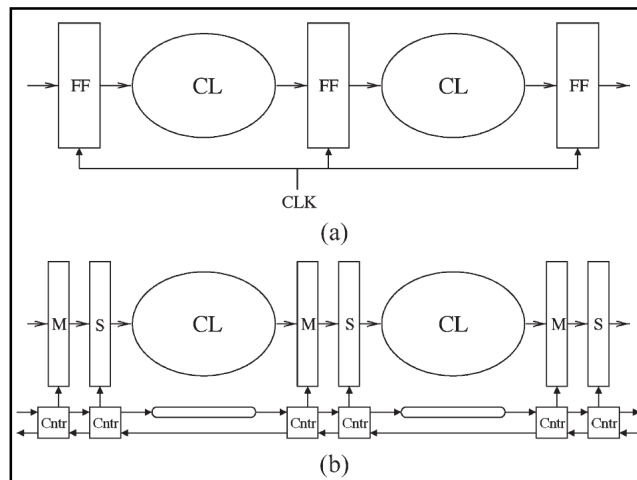


Figura 26– (a) Circuito síncrono.
(b) Circuito Dessincronizado [12].

5.2.1 Passos da Dessincronização

A dessincronização se dá em três etapas:

- I. A substituição dos Flip-Flop do tipo D do circuito em uma rede de controladores do tipo mestre-escravo. (Fig.35(b)).
- II. Geração de atrasos relacionados a lógica combinacional do circuito. Os atrasos são indicados como retângulos na Fig.35 (b). O atraso corresponde ao atraso do caminho crítico da lógica combinacional. O atraso é como um detector de conclusão.
- III. Implementação da rede de controladores locais.

Para podermos demonstrar de forma mais concisa cada etapa da dessincronização, vamos iniciar mostrando à figura 27 um circuito síncrono dividido em cinco pipelines funcionais. As nuvens demonstram a lógica combinacional que estão interligados com os flip-flops do tipo D. Os blocos demonstrados com linhas tracejadas indicam o conjunto da lógica combinacional acompanhado com o registrador de saída controlados pelo relógio global *CLK*. Cada bloco tracejado é especificado automaticamente como demonstrado em [102], sendo essencial para dessincronização.

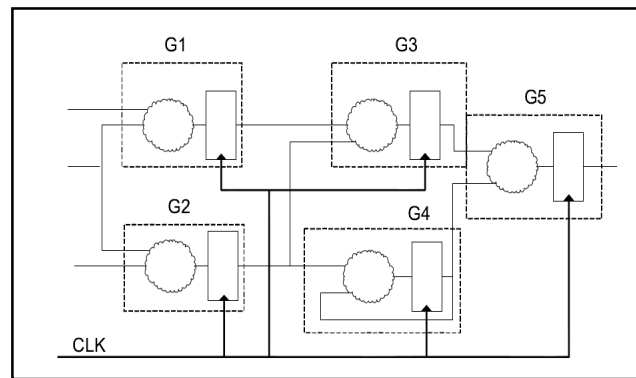


Figura 27– Circuito assíncrono inicial [102].

5.3 Gráficos de transição de sinal

O modelo utilizado na dessincronização para especificar os controladores assíncronos foi baseado em redes de Petry [103]. É chamado de *Signal Transition Graph* (STG). O STG é um modelo formal para os diagramas de tempo que pode especificar causalidade, concorrência e a transição dos sinais. Uma classe específica de redes de Petry que tem uma maior importância no trabalho é o MG (*marked graph*), no qual a lógica combinacional é representada por transições e os Flip-Flops são representados por lugares.

Os STGs são utilizados para a escolha do protocolo de handshake e para verificar se o circuito irá manter o fluxo equivalente. Foi verificado que os controladores podem ter qualquer protocolo de handshake, como exemplos podemos citar o semi-desaclopado e o completamente desaclopado, dentre outros tipos, desde que seja mantido o fluxo de equivalência. Podemos observar através da figura 28 vários protocolos que podem ser utilizados na dessincronização. Os controles de *latches* são compreendidos na figura pelos sinais A e B. O protocolo de comunicação pode ser de quatro ou duas fases. O projeto DLX, base dessa dissertação, utiliza controladores semi-desaclopado de quatro fases. Ele indicou ter uma boa relação custo benefício entre os sinais concorrentes e a complexidade que esse protocolo apresenta frente aos circuitos assíncronos. Como podemos ver na figura, dois tipos de protocolos não são permitidos: O protocolo *not alive* por apresentar *deadlocks*, e o *not-flow-equivalent*, por apresentar *data overwriting*.

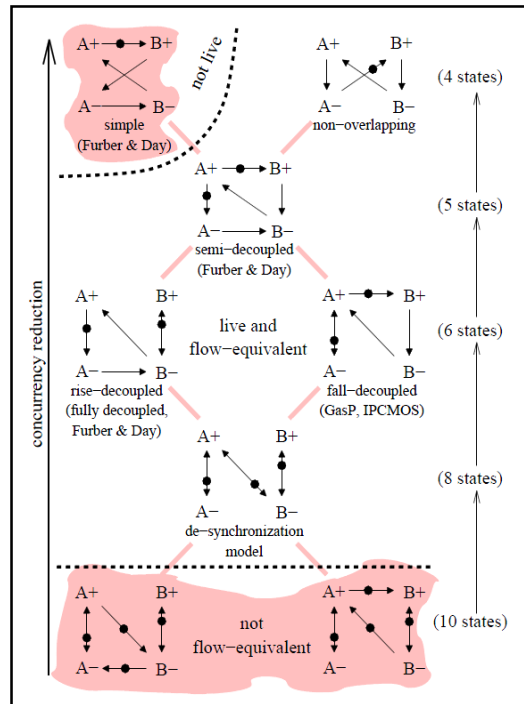


Figura 28 - Diferentes graus de concorrência nos protocolos de handshake. [12]

5.4 LATCH CONTROLLER – Controlador

O principal elemento usado na rede de controladores mestre-escravo é o controlador de trava que é implementado conforme o protocolo de *handshake*. A figura 29 apresenta um controlador de trava genérico.

Ao lado esquerdo da figura temos os sinais que se comunicam com o controlador antecessor, enquanto ao lado direito temos os sinais que se comunicam com o controlador sucessor.

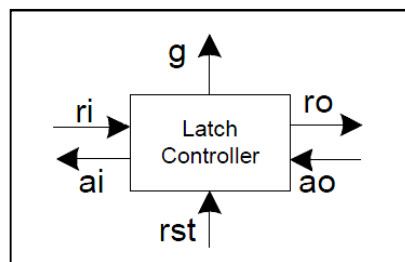


Figura 29 – Controlador de trava [102]

Abaixo temos a descrição de cada um dos sinais do controlador na figura anterior:

- **ri** – Sinal de entrada de requisição que indica se o sucessor possui dados disponíveis;
- **ai** – Saída de reconhecimento que indica se a região processou os dados e está apta a receber novos dados;
- **ro** – Sinal de saída que indica ao controlador sucessor à existência de dados válidos;
- **ao** – Sinal de entrada que indica se foram processados os dados do controlador sucessor;
- **g** – Habilita a captura de dados pelo conjunto de *latches*;
- **rst** – Reinicia.

5.5 SUBSTITUIÇÃO DOS FLIP FLOPS

Dado que o projeto síncrono é baseado em *flip-flop*, eles devem ser substituídos por controladores mestre-escravo (*master-slave*). O conceito referente à *flip-flops* do tipo D, é que eles representam um conjunto de controladores mestre escravo, sendo sua estrutura interna explicitamente exposta, sempre mantendo o fluxo equivalente.

A transformação é essencial, pois torna o circuito tolerante a escorregamentos (*skews*) nas diferentes regiões dos controladores. Um caso problemático pode ser observado na figura 30. Caso dois *flip-flops* sejam controlados por diferentes controladores, o escorregamento entre A e B não é garantido, ocorrerá assim a sobrescrita de dados.

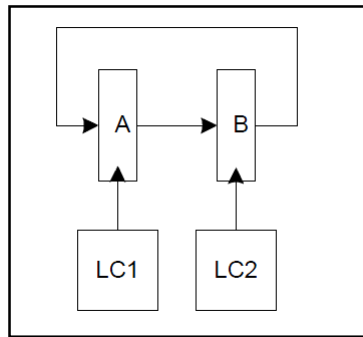


Figura 30 – Exemplo de loop de *flip-flop* onde não é permitido escorregamento [102].

A conversão de um circuito baseado em *flip-flop* em um baseado em controladores mestre-escravo pode ter o desempenho melhorado, mas com um aumento na área. Este processo é conhecido na melhora da performance em sistemas síncronos, não sendo específica no processo de dessincronização [102].

5.6 REMOÇÃO DO RELÓGIO

Como foi mencionada anteriormente, a principal diferença entre um circuito síncrono e sua contraparte assíncrona é que o circuito dessincronizado passa a ter uma rede de controladores mestre-escravo, que gera os sinais para os elementos do circuito dessincronizado. Essa seção tem como objetivo apresentar como essa rede é composta, como é a interligação do caminho de dados (*datapath*) do circuito síncrono e como é a ligação entre os controladores para que o funcionamento do circuito seja assegurado e o fluxo equivalente mantido.

5.6.1 GRAFO DE DEPENDÊNCIA

A rede de controladores mestre-escravo deve respeitar as dependências de fluxo de dados entre as diversas partes do circuito. Assim, o primeiro passo da análise de um circuito para dessincronização é a construção de um gráfico que represente a dependência do circuito [102]. A cada dependência de dados entre duas regiões do circuito, ou seja, uma saída de dados de uma região para a entrada de dados de outra região é representado pela direção da seta ligando os dois nós. Mantendo como base o

exemplo apresentado na figura 27, apresentamos na figura 31 a dependência de dados do exemplo.

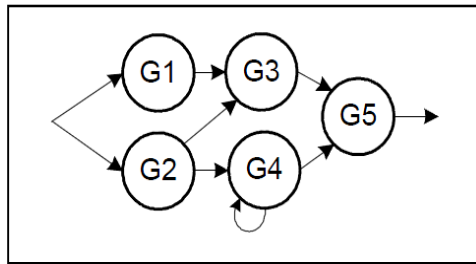


Figura 31 – Dependência de dados da figura 34.

O grafo de dependência de dados mostrado é utilizado para construir a rede de controladores. O nó no grafo de dependência de dados representa cada região do circuito e será controlado por um par de controladores mestre-escravo. O controlador mestre-escravo de um nó deve ser conectado a todos os seus nodos antecessores e sucessores, usando elementos de sincronização, como o elemento C-muller citado no capítulo anterior. Podemos ver a rede de controladores resultante na figura 32. Os elementos C-muller são utilizados onde temos mais de uma requisição de entrada e saída, como podemos ver na figura.

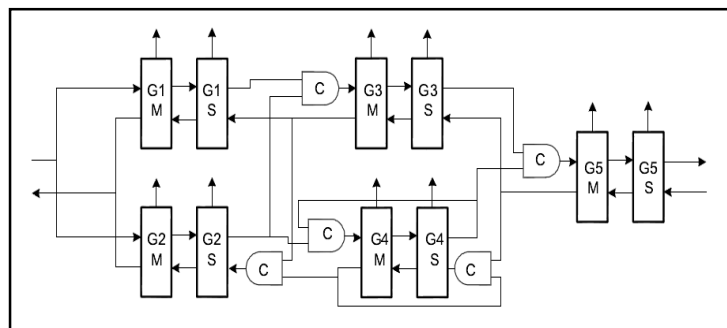


Figura 32 – Rede de controladores resultante [102].

5.6.2 ATRASO PARA LÓGICA COMBINACIONAL

O circuito dessincronizado deve respeitar as restrições de configuração dos seus elementos sequenciais. Isto implica que as nuvens referentes à lógica combinacional têm que ter tempo suficiente para calcular os seus dados. Quando o sinal de requisição

indica que a região terminou o processamento e os dados estão válidos, esses sinais devem conter um atraso maior ou igual ao caminho crítico do circuito. Dois métodos são utilizados para alcançar esse objetivo. Uma forma é utilizando um atraso que imita o atraso da lógica combinacional e a outra é com a utilização de um detector de conclusão.

O primeiro método, com o atraso imitando a lógica combinacional, o sinal de requisição passa por um atraso antes de atingir o controlador de destino dos dados. Para cada parte do circuito teremos um atraso correspondente. A figura 33 apresenta como os atrasos são conectados ao circuito. As restrições de tempo são satisfeitas se:

$$\text{delem_length} + \text{CT_target} \leq \text{CT_source} + \text{CL_delay}$$

Para buffers equilibrados temos:

$$\text{CT_target} \leq \text{CT_source}$$

Ficando:

$$\text{delem_length} \leq \text{CL_delay}$$

O elemento de atraso corresponde ao tempo do caminho crítico.

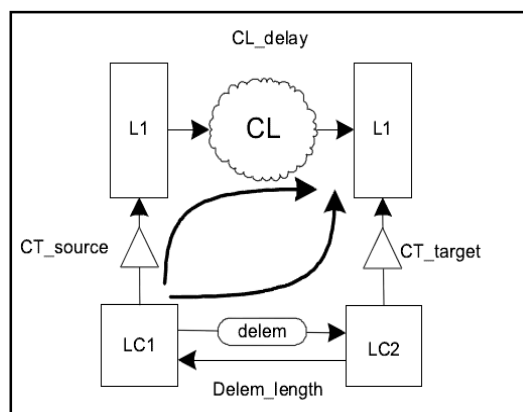


Figura 33 – Elementos de atraso [102].

Os elementos de retardamento utilizados são implementados como simétrico no protocolo de *handshaking* de duas fases ou assimétrico no protocolo de *handshaking* de quatro fases.

Uma alternativa para o método que indicamos anteriormente é a lógica de detecção e conclusão [12]. Neste caso a lógica combinacional é transformada de tal forma que gera sinais de detecção e conclusão, indicando a conclusão da computação e a validade dos dados. Este sinal pode ser usado como uma solicitação de entrada para um controlador. Esse circuito tem a vantagem de permitir que o circuito trabalhe em tempo real. Sua principal desvantagem é que a transformação da lógica induz a uma sobrecarga de área muito grande e um maior consumo de energia. A figura 34 mostra a lógica de detecção e conclusão.

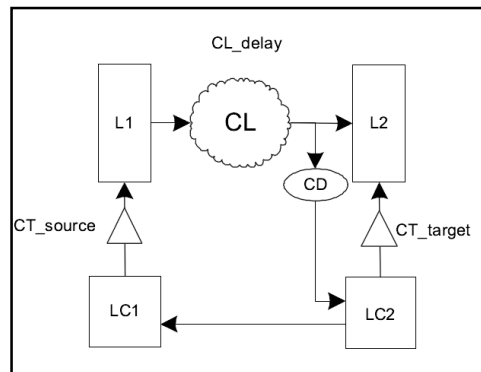


Figura 34 – Lógica de detecção e conclusão [12].

5.7 INTERLIGAÇÃO DA REDE

Por fim, na última etapa da dessincronização a rede de controladores mestre-escravo é religada ao circuito síncrono original. Esta rede é formada por pares de controladores, elementos C-muller e elementos de atraso. Temos apresentado na figura 35 o circuito resultante das etapas de dessincronização. As linhas em negrito indicam a rede de controladores e as nuvens representam a lógica combinacional

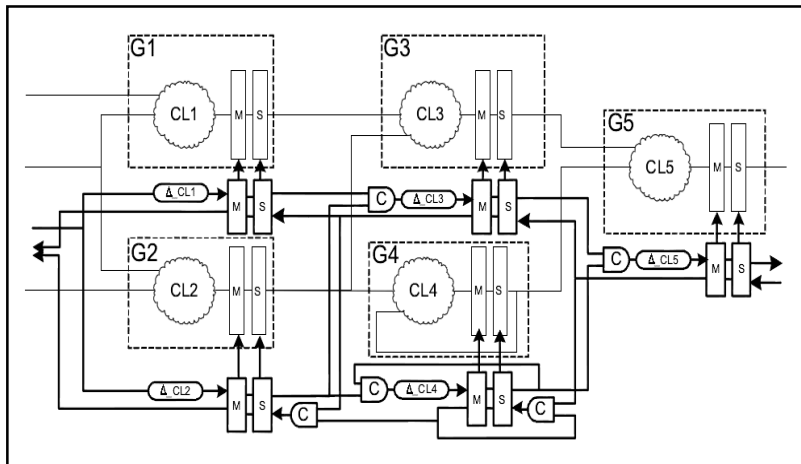


Figura 35 – Circuito resultante [12].

5.8 CONCLUSÕES DO CAPÍTULO

Este capítulo apresentou as etapas do processo de dessincronização de circuitos síncronos, transformando-os em uma classe específica de circuitos assíncronos. A técnica tem como objetivo sanar as deficiências provenientes de circuitos assíncronos, como a falta de projetistas na área do conhecimento assíncrono, bem como a falta de ferramentas para o uso deste paradigma.

6 PLATAFORMA ADOTADA DURANTE OS EXPERIMENTOS

Tendo como primeiro objetivo desta dissertação a comparação entre um processador síncrono e sua equivalência dessincronizada, foi necessário a utilização de uma plataforma de testes. Essa plataforma também foi utilizada para validar a proposta para o aumento de robustez de circuitos assíncronos frente ao ruído eletromagnético conduzido. Esse capítulo faz uma apresentação da plataforma de testes utilizada em nosso trabalho. Primeiramente será apresentada a placa de ensaios utilizada para prototipar os processadores, posteriormente mostraremos o injetor de ruído conduzido.

6.1 INTRODUÇÃO

Os hardwares existentes e kits de desenvolvimento dos fabricantes de FPGA, como a *Xilinx*[74], não permitem um estudo aprofundado sobre interferência eletromagnética EMI necessários nesta dissertação, pois possuem limitações estruturais que impossibilita abordagens mais complexas. Para a plena execução desta proposta, é necessário a realização de vários experimentos práticos, que por sua vez, estão baseados no uso de plataformas de teste. Com este objetivo, foi desenvolvido no ano de 2009 pela equipe do Laboratório SISC uma plataforma de ensaios, e em 2010 foi desenvolvido um injetor de ruído que foi primordial para o desenvolvimento dessa dissertação. Esta plataforma de ensaios possui características estruturais mais complexas, robustas e funcionais, sendo baseada nas normas de ensaio de susceptibilidade de circuitos integrados às interferências eletromagnéticas (EMI) conduzidas, possuindo características apropriadas como entradas independentes para alimentação elétrica de seus principais componentes.

6.2 PLATAFORMA DE ENSAIOS

Foi desenvolvida uma placa baseada nas normas de ensaios de susceptibilidade de circuitos integrados à EMI irradiada e conduzida IEC 62132-1, IEC 62.132-2 e IEC 62132-4. A placa possui seis camadas (*layers*) como mostra a figura 36.

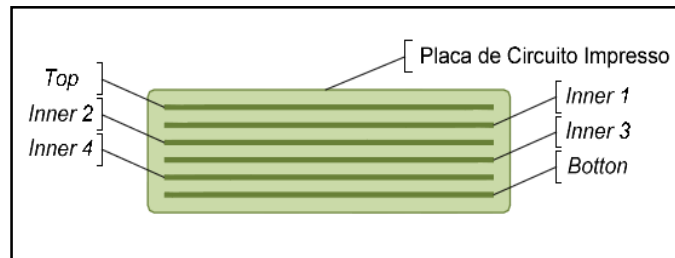


Figura 36 – Camadas da placa do circuito impresso.

A seguir é apresentada a descrição da distribuição das camadas da placa de circuito impresso:

- I. **Camada 1 (*Top*)** – Possui os CIs sob teste da plataforma, além de um plano de terra (GND) cobrindo toda área da camada;
- II. **Camada 2 (*Inner 1*)** – Possui somente os planos de alimentação (V_{CC}) dos circuitos integrados sob teste;
- III. **Camada 3 (*Inner 2*)** – Possui todas as trilhas de sinal e/ou alimentação dos componentes e/ou dispositivos da placa;
- IV. **Camada 4 (*Inner 3*)** – Possui todas as trilhas de sinal e/ou alimentação dos componentes e/ou dispositivos da placa;
- V. **Camada 5 (*Inner 4*)** – Possui todas as trilhas de sinal e/ou alimentação dos componentes e/ou dispositivos da placa;
- VI. **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*.

A figura 37 apresenta um diagrama de bloco da plataforma de testes. Fotos da placa são apresentadas nas figuras 38 e 39 com destaque aos seus principais componentes.

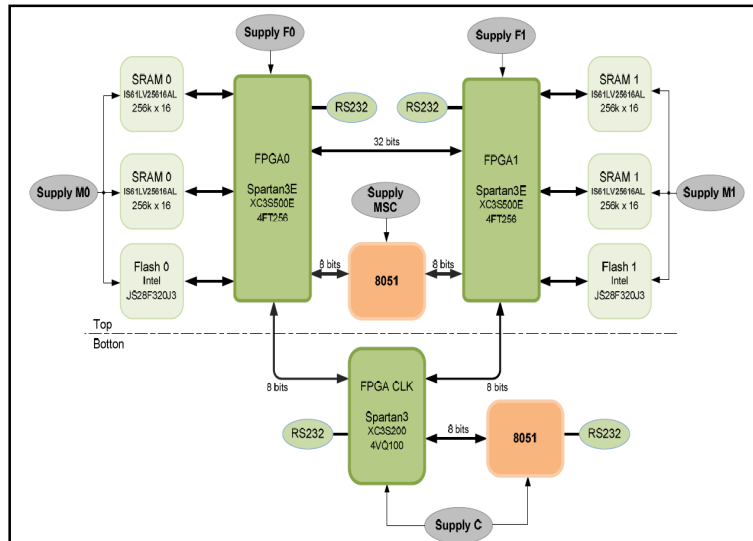


Figura 37 - Plataforma de ensaio.

Sua arquitetura dispõe dos seguintes componentes:

- I. Uma FPGA Xilinx XC3S200 (200k portas, 144 pinos, 216 Kbits de block RAM interna, 12 multiplicadores e 2 DCM's);
- II. Dois FPGAs Xilinx XC3S500E (500k portas, 256 pinos, 360 Kbits de block RAM interna, 20 multiplicadores e 4 DCM's);
- III. Quatro memórias SRAM IS61LV25616AL-10T, produzidas pela ISSI, que formam dois bancos de memória de 1Mbyte com configuração de 256x16 para cada FPGA;
- IV. Duas memórias Flash Intel JS28F320J3 32Mbits e tempo de acesso de 110ns;
- V. Dois microcontroladores 8051 produzidos pela Texas Instruments;
- VI. Três osciladores de frequência igual a 49.152MHz (para cada FPGA);

- VII. Dois cristais de frequência igual a 11.0592MHz (para cada microcontrolador);
- VIII. Comunicação serial padrão RS-232 (para cada FPGA e microcontrolador);
- IX. Três reguladores de tensão LM317 para o controle independente dos níveis de tensão de alimentação;
- X. Um sensor de temperatura serial 12 bits LM74, produzido pela National Semiconductor;
- XI. Quatro botões e 4 LED's;
- XII. Dois conectores JTAG independentes para programação e debug dos FPGA's;
- XIII. Jumper's para seleção e controle independente dos níveis de tensão alimentação.

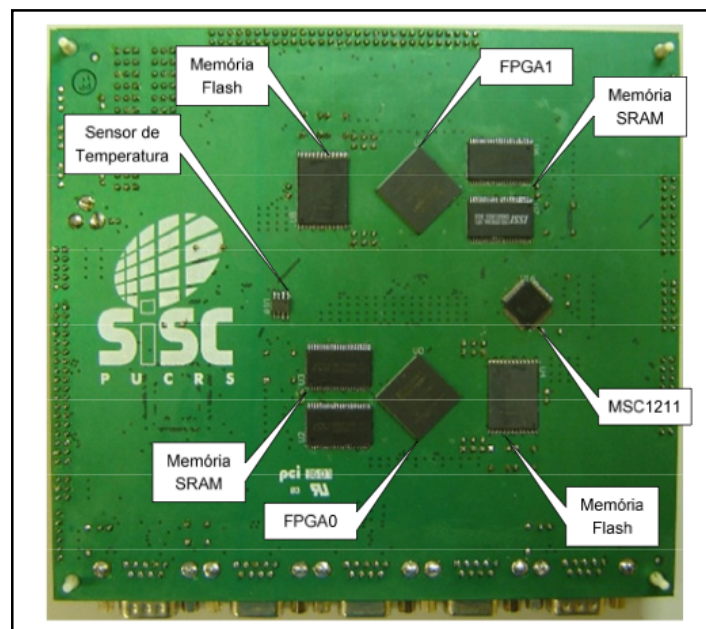


Figura 38 – Vista superior da plataforma de testes.

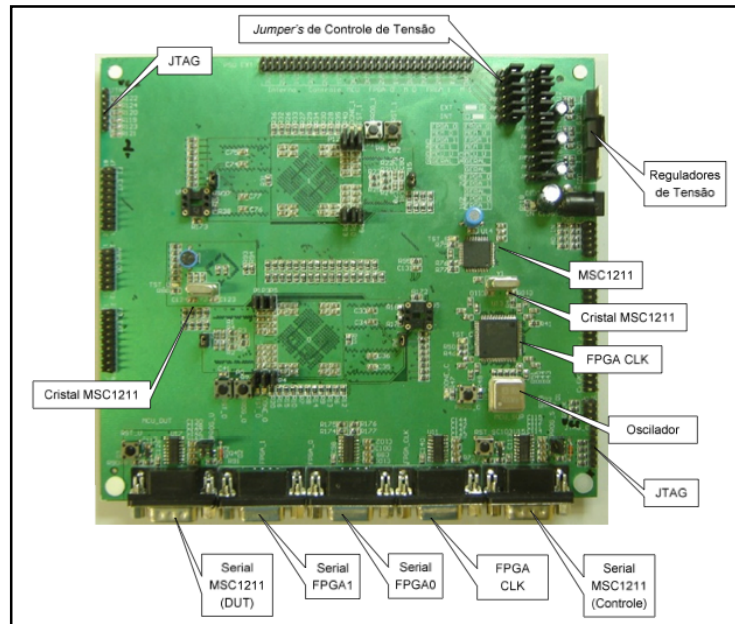


Figura 39– Vista inferior da plataforma de testes.

6.3 INJETOR DE RUÍDO ELETROMAGNÉTICO CONDUZIDO

O injetor de ruído foi desenvolvido por Darcio Pinto Prestes em seu trabalho de mestrado [104], a qual foi concebida para avaliar trabalhos desenvolvidos no grupo de pesquisas SiSC, da PUCRs, ao qual faço parte. O injetor de ruído Devastator permite a injeção de ruído eletromagnético conduzido para a realização de experimentos envolvendo a avaliação de técnicas de tolerância a falhas, bem como a robustez de circuitos integrados e sistemas [104]. Ela permite também que seu funcionamento seja controlado externamente com o objetivo de fazer experimentos automatizados. A figura 40 mostra a plataforma.



Figura 40 – Injetor de falhas [104]

A figura 41 mostra como a plataforma é classificada no que diz respeito aos diferentes mecanismos de injeção de falhas ou *Power Supply Disturbances* (PSD).

Em nosso trabalho utilizaremos a norma 61000-4-29 para a variação de tensão e simulação de um *Ripple*.

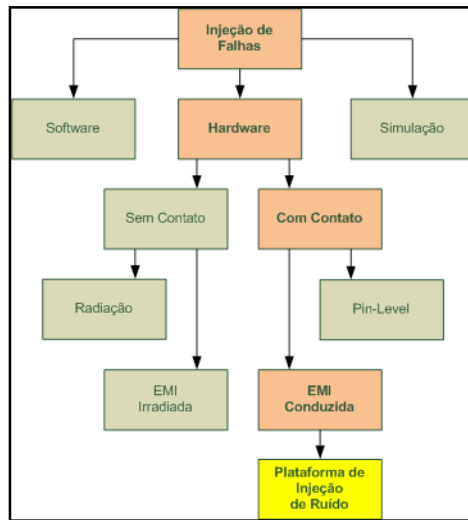


Figura 41 – Classificação da plataforma [104]

6.3.1 ESPECIFICAÇÃO

As características técnicas principais do gerador de ruído são mostrados na tabela 8.

Tabela 8 – Características da implementação do gerador [104]

<i>Característica</i>	<i>Unidade</i>	<i>Valor</i>
Tensão máxima de saída	V	9,99
Corrente máxima de saída	A	2,00
Temporização mínima máxima	ms	1 2000
Tempo de resposta de habilitação	ms	20
Tempo máximo de subida do sinal	ns/V	185
Regulagem de carga	%	< 5,00

Os parâmetros da plataforma são definidos abaixo retirados de [104]:

- I. **Tensão máxima de saída:** É a tensão máxima que o gerador pode disponibilizar na sua saída. O hardware da plataforma foi projetado para fornecer um sinal que excursiona desde -15,0 até +15,0 V, porém foi implementada uma limitação para este valor, por software, para que o mesmo se mantivesse dentro da faixa de 0,00 a 9,99 V;
- II. **Corrente máxima de saída:** É a corrente máxima que é possível drenar do gerador. Ela é limitada pelo transformador de alimentação e nesta implementação seu valor é de 2,00 A;
- III. **Temporização mínima | máxima:** É o tempo durante o qual o gerador mantém determinado nível de tensão em sua saída. O hardware permite uma temporização mínima (e múltipla) de 100 μ s, porém através do software esta temporização foi implementada numa faixa que vai de 1 até 2000 ms, em passos de 1 ms;
- IV. **Tempo de resposta de habilitação:** É o tempo decorrido entre o acionamento da entrada *enable* e o início da modulação do ruído na componente DC da saída do gerador. Esta entrada de controle é utilizada quando o gerador é gerenciado a partir de um equipamento externo;
- V. **Tempo máximo de subida do sinal:** É o tempo de subida e descida máximo que o sinal de saída consegue excursionar. A plataforma consegue ir de 0,00 a 5,00 V em 925 ns;
- VI. **Regulagem de carga:** Neste caso refere-se à variação na tensão de saída quando, ajustada em 10 V e a partir da ausência de carga, conectamos uma carga resistiva de 10 ohms, estabelecendo uma corrente de 1,00A aproximadamente;

7.3.2 FLUXOGRAMA DE CONFIGURAÇÃO

Para a utilização do gerador é necessário seguir as etapas apresentadas na figura

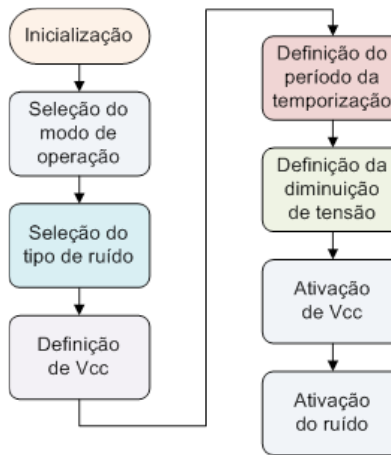


Figura 42 – Configuração do gerador [104]

6.4 CONCLUSÃO DO CAPÍTULO

A plataforma de ensaios, constituída de um equipamento de alimentação e injeção de falhas, bem como a placa de ensaios que foram descritos, foram utilizadas para a realização deste trabalho na execução de ensaios de EMI e interferência eletromagnética conduzida tanto na comparação do circuito, quanto na validação e verificação da proposta.

7 ANÁLISE DA ROBUSTEZ DOS PARADIGMAS

Este capítulo apresenta a comparação entre as duas versões de um mesmo circuito. A versão original síncrona e a versão obtida a partir do processo de dessincronização. A princípio será apresentada uma introdução referente aos conceitos de dessincronização e os motivos pelo qual foi escolhido esse paradigma, posteriormente apresentaremos o processador síncrono usado e os experimentos realizados. Finalmente serão demonstrados os resultados obtidos. A análise tem como objetivo, demonstrar que esse novo paradigma possui as características benéficas dos circuitos assíncronos e demonstrar que em ambientes hostis, o circuito assíncrono apresenta algumas falhas.

7.1 INTRODUÇÃO

Com base na bibliografia referenciada nos capítulos anteriores, o estudo de caso tem o objetivo de analisar a robustez dos circuitos síncronos, comparados com sua versão dessincronizada. A escolha pelo processador DLX, utilizado no projeto ASPIDA (*ASynchronous Open-source Ip of the Dlx Architecture*) [12], se deu por três motivos principais. O primeiro e principal motivo deu-se por estar disponível a nível acadêmico e possuir código aberto para realizar diversas modificações para sua análise e para apresentação da proposta. O segundo motivo é por sua implementação recente e complexidade para conseguirmos aproximar da complexidade dos processadores utilizados atualmente. E por fim, pela necessidade da utilização de um processador que existisse na sua forma síncrona e assíncrona com a disponibilidade de ser prototipado em FPGAs, para a realização de ensaios e modificações.

Com essas versões de processadores, foi possível a utilização de ferramentas CAD convencionais utilizadas em projetos síncronos e que temos disponibilidade no laboratório de pesquisa SiSC.

Como foi mencionado anteriormente, utilizou-se o projeto ASPIDA [105]. Houve a participação de outras instituições como o Politécnico de Torino, FORTH-ICS, Manchester University e IHP Moelectronics [105]. O projeto teve

como maior objetivo demonstrar a viabilidade industrial e o potencial de reutilização IP assíncronos, com código aberto e com qualidade industrial. Eles deveriam ser projetados, testados e implementados com a utilização de ferramentas tradicionais industriais para especificações de IP reutilizáveis abertos (*Open IP reuse specifications*), e assim poder ser usados em arquiteturas que utilizem *Open IP system-On-a_Chip*. É uma arquitetura do tipo RISC (*reduce instruction set computer*) e instruções DLX. Os pesquisadores pretendem que este projeto promova a utilização de hardware de fonte aberta(*Open IP reuse*).

O processador ASPIDA DLX pode ser facilmente convertido em qualquer ISA RISC, pois implementa um IP assíncrono da arquitetura do conjunto de instruções DLX (ISA) que incorpora o suporte para conversão para ISA (*Instruction Set Architecture*). Com o objetivo de produzir um *netlist* portátil e distribuir todos arquivos de HDL intermediários usados para projeto de alto nível e desenho do nível de porta o fluxo de projeto, foi baseado em ferramenta EDA, sendo seu produto final independente da tecnologia e independente da regulamentação de tempo. Como consequência, os projetistas conseguiram o objetivo de apresentar um produto que fosse compatível com as ferramentas e fluxos industriais convencionais sem a dependência de ferramentas assíncronas e sem a necessidade de projetistas com conhecimento de circuitos assíncronos, sendo implementado tanto em ASIC como em FPGAs [105].

Na próxima seção será falado um pouco do processador original sem as modificações referentes ao processador ASPIDA DLX, sendo que as características que apresentaremos são características que as duas versões possuem. Posteriormente falaremos do processador ASPIDA DLX e de sua versão assíncrona que fora utilizados com algumas modificações para as comparações entre os dois circuitos.

7.2 PROCESSADOR DLX

O processador DLX tem como principais características [106]:

- Arquitetura do tipo RISC (*Reduced Instruction Set Code*);
- 5 estágios de *pipeline*;
- Arquitetura de 32 bits;

É um processador muito usado em nível acadêmico por ter semelhanças com processadores RISC como MIPS [107] e ARM [5]. É mostrado a partir da figura 43, os estágios dos *pipelines* do processador.

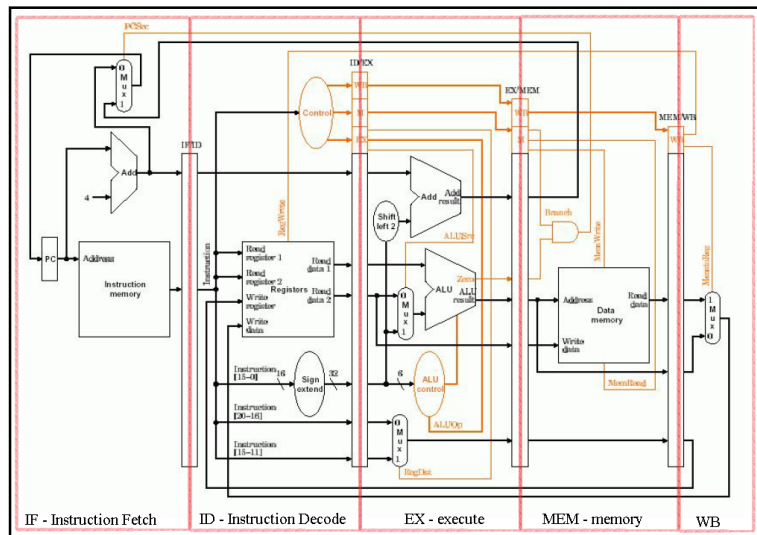


Figura 43 – Processador DLX – pipelines [106]

Pipelines do processador DLX [106]:

- I. **IF-Instruction Fetch** :O PC (*Program Counter*) aponta para a instrução a ser lida na memória de instruções. Verifica se há uma instrução de salto a ser executada, instruções estas analisadas no estágio ID, caso não exista, o próximo endereço é calculado adicionando quatro bits ao valor do endereço anterior;
- II. **ID – Instruction Decode**:Nessa fase a instrução que foi buscada anteriormente é decodificada e os registradores são acessados. É verificado se há saltos condicionais, relativos ou absolutos sendo calculados e enviados de volta ao estágio anterior;
- III. **EX-Execute**:realização das operações lógicas e aritméticas, sendo que os operandos são recebidos do estágio anterior. Resultados das operações são escritos nos registradores ou em endereços de memória;

- IV. **MEM-Memory:** Acessos de escrita e leitura na memória;
- V. **WB-Write Back:** Valores escritos em registradores são enviados ao estágio ID.

7.2.1 ASPIDA DLX

As principais características referentes a esse processador são de possuir suporte total a dados do tipo inteiro, não possuindo suporte para operação de ponto flutuante, interface de memória de byte, meia palavra e palavra inteira. Como a versão original, ele também possui um co-processador de vetores de interrupção.

Possui três tipos de operação. A operação *I-type*, que são operações lógicas e aritméticas executadas entre um registrador e um valor imediato, sendo os saltos condicionais também definidos como esse tipo de instrução. A operação *R-type* também é uma operação lógica e aritmética, mas executadas entre dois registradores, sendo leitura e escrita definidas como esse tipo de instrução. Já as operações do tipo *J-type* são as instruções referentes a saltos, salto absoluto ou relativo. As tabelas 9 e 10 mostram os conjuntos de instruções suportadas pelo processador, muitas instruções foram omitidas nas tabelas, mas as instruções omitidas não são suportadas na versão ASPIDA DLX.

Tabela 9 – Instruções DLX – conjunto 1 [105]

IR [5-3]								
IR [2-0]	000	001	010	011	100	101	110	111
000	-	-	j	jal	beqz	bnez	-	-
001	addi	addui	subi	-	andi	ori	xori	lhi
010	rfe	trap	jr	jalr	slli	-	srli	srai
011	seqi	snei	slti	sgti	slei	sgei	-	-
100	lb	lh	-	lw	lbu	lhu	-	-
101	sb	sh	-	sw	-	-	-	-

Tabela 10 – Instruções DLX – conjunto 2 [105]

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

7.3 PROCESSADOR DLX DESSINCRONIZADO

Como foi descrito anteriormente na técnica de dessincronização, o relógio global é substituído por uma rede de controladores mestre-escravo. O processador DLX passou pelo processo de dessincronização e podemos observar através da figura 44 a ausência do relógio. Foi dado destaque em amarelo para a rede de controladores mestre-escravo que substituíram o relógio global do circuito original. Apesar de o processador apresentarcinco estágios de pipeline, podemos observar também que são quatro estágios funcionais.

O protocolo assíncrono utilizado nesse projeto é o protocolo de quatro fases com controladores semidescoplado. O circuito final foi obtido a partir de uma especificação STG (*Signal Transition Graph*). O STG é uma subclasse de redes de Petry utilizado em sistemas concorrentes, e foi utilizado na dessincronização do circuito na criação de um modelo de sincronização, o qual é definido a partir de uma série de regras para que o circuito mantenha seu fluxo de equivalência sendo considerado válido [108].

A tabela 11 mostra a comparação entre os dois processadores no que diz respeito à energia consumida, área utilizada e velocidade. Pode ser visto que eles apresentam praticamente a mesma área, velocidade e consumo de energia. Já na figura 45 nós apresentamos primeiramente à forma genérica do circuito DLX síncrono e logo abaixo, na mesma figura, o mesmo processador após passar pelas etapas de dessincronização

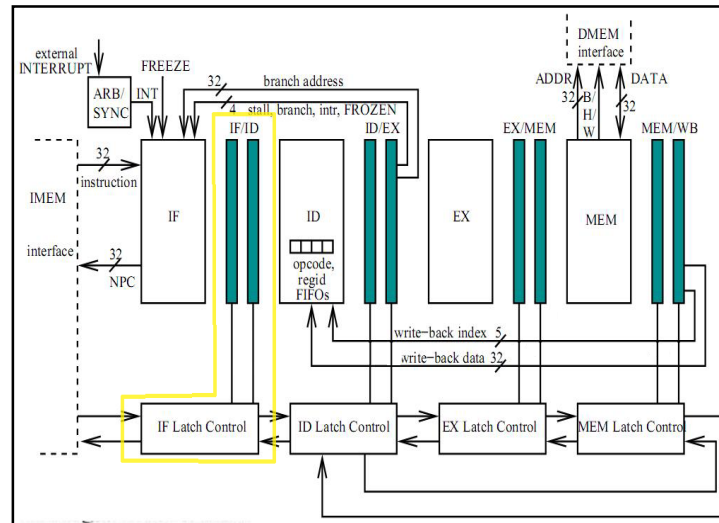


Figura 44 – DLX dessincronizado [108]

O novo paradigma não apresenta menos consumo de energia como os circuitos assíncronos normalmente apresentam, pois possui características que o diferenciam dos circuitos puramente assíncronos, para que o novo paradigma possa fazer uso das mesmas ferramentas utilizadas em circuitos síncronos.

Tabela 11 – comparação de versões [12].

	DLX - síncrono	DLX dessincronizado
Cycle time	4.4ns	4.45ns
Dyn. Power Cons.	70.9mW	71.2mW
Área	372.656 μm^2	378.058 μm^2

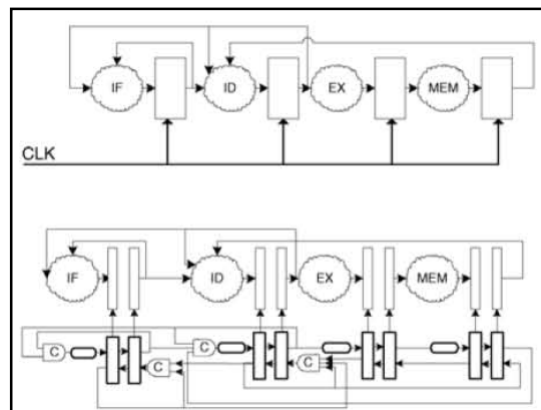


Figura 45 – Comparação entre os dois processadores [102].

7.4 EXPERIMENTOS

A descrição original do projeto ASPIDA DLX, disponível no *website OpenCore* na internet foi utilizado nessa proposta, possui em sua interface uma memória de instruções e dados para a CPU, outra para a VRAM e um interface para um controlador VGA de vídeo. A figura 46 mostra o esquema de entrada e saída do processador ASPIDA DLX [105].

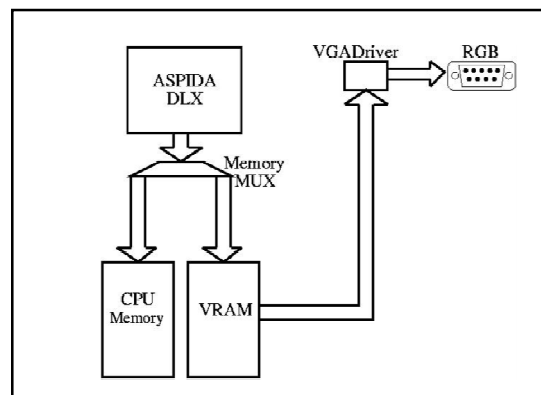


Figura 46 – ASPIDA DLX .

Devido à estrutura de entrada e saída proposta no projeto ASPIDA, foi necessário a alteração. Foi realizada a troca da interface de vídeo por uma interface UART. Essa troca se deve ao fato da necessidade de uma saída para os dados em teste e que também fosse compatível com a plataforma de ensaios que nós estamos utilizando e que foi citado anteriormente. Outra troca que foi necessária também, foi alterar a memória de 512 bytes para 2 Kbytes, essa alteração foi necessária para ser possível a realização do nosso estudo de caso [13] [14]. A figura 47 mostra essa nova estrutura e pode ser comparada com a figura 46 que mostra a versão original [13] [14].

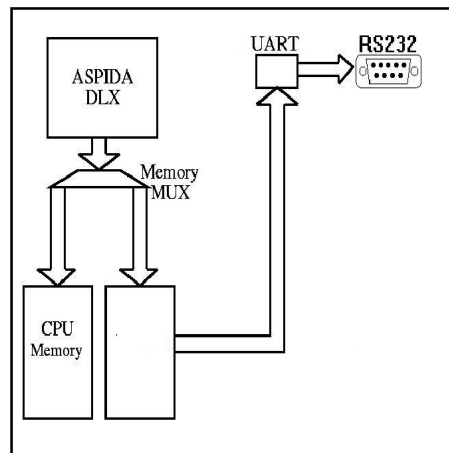


Figura 47 – ASPIDA DLX para ensaio.

É importante salientar que o core do processador ASPIDA DLX foi mantido em ambas as versões síncrona e assíncrona, da mesma forma que foi obtida na página da *open cores*. As únicas alterações para a realização destes experimentos foi o aumento da capacidade de memória e a interface de saída para uma melhor adequação e verificação da proposta.

7.4.1 APLICAÇÃO DO ESTUDO DE CASO

Para podermos realizar os ensaios tanto na comparação de circuitos síncronos frente ao circuito dessincronizado, quanto para a validação da proposta de modificação nos atrasos do processador, foi necessário a escolha de um software que realizasse os ensaios. Com a bibliografia referenciada na área de teste e confiabilidade de sistemas como em [13] [14] [90] [109] [110], foi incentivado que fosse utilizada nesse trabalho a multiplicação de uma matriz de dez linhas e dez colunas, operando com números inteiros, dado que o processador não possui suporte para ponto flutuante, sendo esta uma limitação. A multiplicação de matrizes foi decisiva, pois nosso foco está na avaliação do desempenho e da robustez do processador assíncrono, pois ela possui um grande número de multiplicações e uma boa quantidade de saltos para a realização da computação.

A aplicação foi carregada a partir dos blocos de memória contendo o código de máquina do software para o ensaio, utilizando-se do compilador *gcc* (*GNU Compiler Collection*) disponibilizado no site da *opencores* [105].

7.4.2 AMBIENTE DOS EXPERIMENTOS

Os experimentos realizados têm como meta analisar a robustez dos circuitos síncronos e assíncronos frente ao ruído eletromagnético. Nesta seção iremos apresentar como esses experimentos foram realizados. A figura 48 abaixo demonstra a configuração dos experimentos.

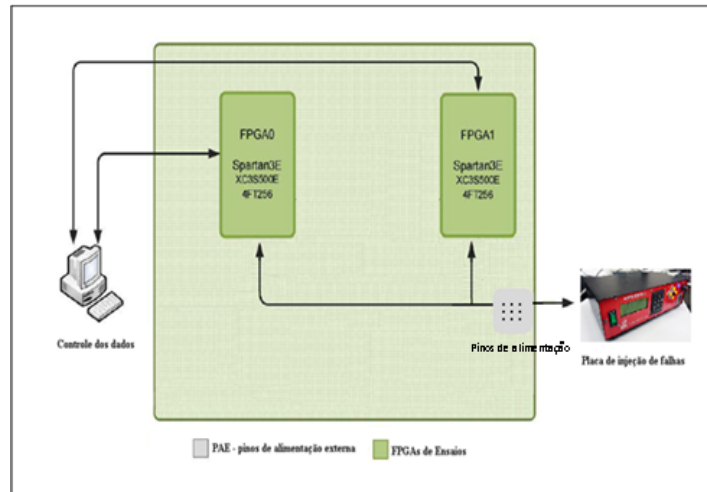


Figura 48 – Configuração dos experimentos.

Através da figura podemos observar o uso das duas FPGAs na placa de ensaios. O gerador de ruídos e um computador que captura os dados gerados pelos processadores que executam multiplicações de matrizes com valores conhecidos para a comparação de seus resultados.

Os experimentos foram realizados na Faculdade de Engenharia da Pontifícia Universidade Católica do Rio Grande do Sul no Laboratório SiSC (Sistemas, Sinais e Computação). Os experimentos baseiam-se na NORMA IEC 61.000-4-17 conforme descrito na seção 3.5.1. A figura 49 apresenta os passos dos procedimentos adotados durante os experimentos.



Figura 49 – Fluxo de execução.

A descrição das etapas referente à figura 49 é apresentada abaixo:

- I. **Início** – A plataforma de ensaios é conectada a estação de trabalho e monitoramento (computador portátil), por meio de um canal de comunicação USB - serial (RS-232), o injetor de ruído e a placa de ensaios são energizadas;
- II. **Configuração do Injetor de Ruídos**–O gerador de ruídos é configurado com base na seção 7.3.2.Foi escolhida a opção queda de tensão de alimentação, logo depois o nível de tensão nominal é configurado para 1.2V, o período de temporização em 20ms e a amplitude primeiramente definida como zero;
- III. **Configuração das FPGAs** –Nesta etapa o processador DLX síncrono é carregado em uma das FPGAs, e o DLX assíncrono na outra FPGA existente. Eles são carregados através do cabo JTAG USB e com a utilização do software ADEPT da *Digilent*;

- IV. **Início da multiplicação de matriz** –A aplicação é iniciada através do botão de reset da plataforma de ensaios;
- V. **Aplicação do *Ripple***–O ruído começa a ser inserido, com o injetor de ruído passando do modo *standby* para o modo de operação;
- VI. **Aumento nos níveis de *Ripple***–O ruído baseado em *Ripple* conforme foi visto anteriormente é aplicado. A amplitude do ruído passa a ser aumentada a cada etapa;
- VII. **Queda do sistema** – É verificado se cada um dos circuitos está funcionando corretamente. Caso algum dos circuitos tenha parado de funcionar é feita a verificação em que nível de ruído o circuito caiu, continuando os experimentos no processador que ainda está em funcionamento.
- VIII. **Geração do *log***–É gerado um *log*, que são todos os dados de saída gerados pelos processadores, ou seja, os resultados referentes à multiplicação de matriz;
- IX. **Análise dos dados** –Os *logs* de saída gerados pelos processadores são analisados.

Na figura 50 temos apresentado o *Ripple* que alimentamos os circuitos simulando ruídos nas linhas de alimentação. Na figura 50 (a) nós temos a alimentação nominal do circuito, referente à 1.2V. Conforme vamos aumentando o *Ripple*, notem pela figura 50 (b) que chegamos a um limite na tensão máxima do circuito de 1.32V. Esse valor não pode ser incrementado, pois é a tensão máxima permitida para o FPGA.

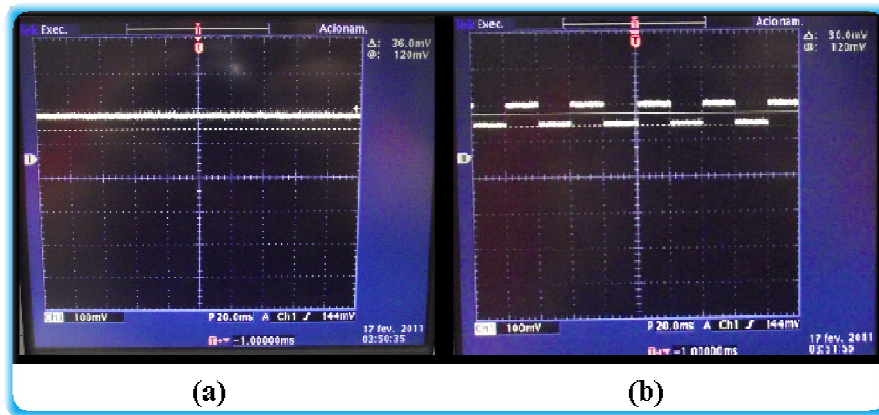


Figura 50 – (a) Tensão nominal inicial (b) Tensão com ripple.

7.5 RESULTADOS

Ao todo foram realizados 200 ensaios com o circuito síncrono e mesma quantidade de ensaios referentes ao circuito dessincronizado. Como dito anteriormente, foi aplicado um *Ripple* da norma IEC 61.000-4-17, para simular tanto ruídos gerados pela alimentação de redes elétricas, como também provenientes de carregadores auxiliares de baterias. Na tabela 3 da seção 3.5.1 podemos ver que utilizamos todos os níveis ate chegarmos ao nível X, onde os níveis são determinados pelo projeto.

Os circuitos síncronos se mostraram funcionais até um *Ripple* de 16,67%, enquanto os circuitos assíncronos chegaram a suportar um *Ripple* de 36,67%, lembrando que os níveis superiores do *ripple*, ou seja, o U_{max} ficou limitado ao máximo do circuito que é de 10%. Com a versão síncrona, então, chegamos a um U_{max} de 1,32V e U_{min} de 1000mV, ou seja, 83,33% da tensão nominal. Já a versão assíncrona com um U_{max} de 1,32 e U_{min} de 760mV, ou seja, 63,33% da tensão nominal. A figura 51 apresenta um gráfico de comparação das tensões entre os dois circuitos. Em verde nós temos apresentado à tensão máxima, em azul a tensão mínima do *ripple*, em vermelho, para dar ênfase nas diferenças entre as versões, é mostrada a amplitude do ruído. Já a figura 52 mostra o *Ripple* atingido por cada circuito.

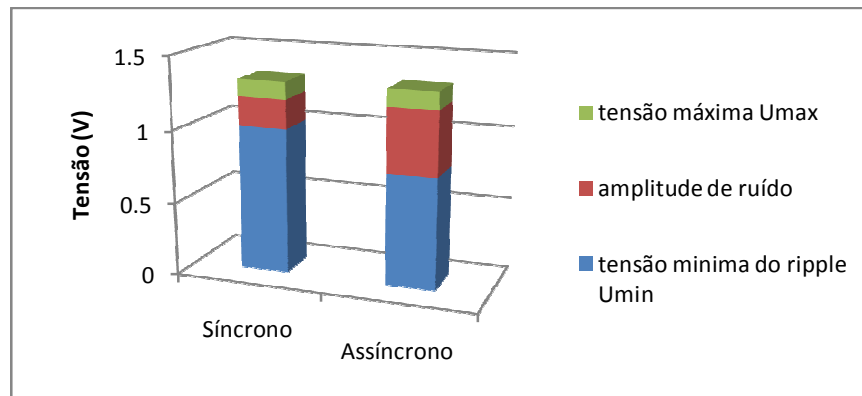


Figura 51 –Gráfico comparativo entre a versão síncrona e versão assíncrona.

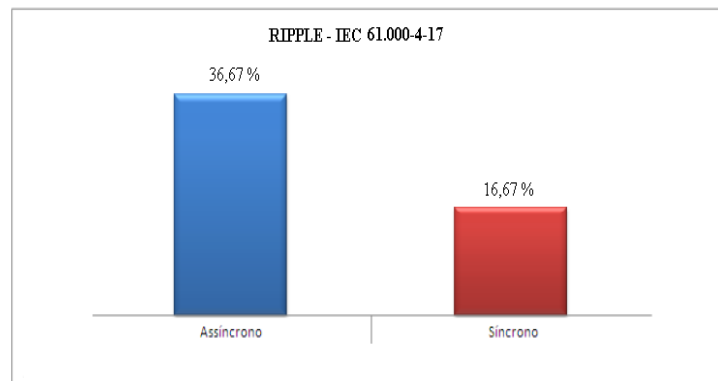


Figura 52 – Ripple.

7.6 CONCLUSÕES DO CAPÍTULO

Neste capítulo podemos ver primeiramente as diferenças entre os projetos DLX síncrono e sua contraparte assíncrona, ou seja, após passar pelas etapas de dessincronização.

Podemos comprovar com os experimentos realizados que a implementação assíncrona é significativamente mais robusta que a implementação síncrona, suportando um *ripple* maior que o dobro do suportado na versão original.

As ondas geradas nas simulações de ruído por *ripple* no circuito digital síncrono provocaram atrasos em sua propagação dos sinais internos do circuito. Esse fato reduziu a integridade do sinal causando a degradação do sistema, perda de confiabilidade e o surgimento de falhas funcionais até a queda do sistema por completo. O circuito digital assíncrono por possuir independência do sinal de relógio e atrasos pré-determinados,

esses são mais bem assimilados pelo circuito, apresentando a robustez natural para circuitos assíncronos.

Apesar de mostrar a maior robustez, foi constatado que o circuito assíncrono em condições extremas de ruído apresentou algumas falhas funcionais, constatados pelos *logs* gerados na fase de testes dos circuitos. Devido a esse fato, o capítulo que segue apresenta uma proposta para melhorar esses níveis de falhas funcionais geradas quando simulamos ruídos extremos.

8 METODOLOGIA PROPOSTA

Nesse capítulo será apresentada uma nova metodologia capaz de aumentar a robustez de circuitos assíncronos. Essa proposta é baseada no uso de grafos de transição de sinais para a tomada de decisão do ponto de inserção dos atrasos na lógica de controle dos circuitos de *handshaking*, visando o aumento da robustez de circuitos assíncronos à interferência eletromagnética conduzida. Após a seleção dos pontos de inserção de atrasos, a técnica busca identificar qual é o valor do atraso ideal para atender ao *standard* internacional IEC 61.0004-29. Após a apresentação da proposta é feita a verificação com testes realizados nas versões descritas e, por fim, uma conclusão do assunto referido.

8.1 INTRODUÇÃO

Por meio do capítulo anterior foi possível observar que circuitos assíncronos são recomendados quando há necessidade de uma maior robustez, já que circuitos assíncronos mostraram maior robustez que o circuito síncrono original.

Através de experimentos complementares realizados conforme visto no capítulo anterior, foi observado que:

- I. Circuitos síncronos apresentam falhas permanentes quando operam com uma queda de alimentação superior a 20%;
- II. Circuitos síncronos tendem a apresentar falhas transientes quando a queda na tensão de alimentação é incrementada durante o funcionamento do circuito, mesmo para variações de queda de tensão inferiores a 20%.

Porém, constata-se que o circuito assíncrono mesmo apresentando uma robustez maior que o circuito original síncrono, o novo paradigma assíncrono apresenta uma confiabilidade menor à medida que aumentamos a quantidade de ruído nas linhas de alimentação do circuito. Embora o circuito assíncrono permaneça em funcionamento a mais de 40% de queda de tensão de alimentação, ele apresenta em torno de mil falhas

transientes por hora nesta faixa de ruído. Em algumas aplicações em que o ambiente em que os circuitos estão expostos apresenta muito ruído, é necessário que estas mantenham um nível mínimo de confiabilidade mesmo com a presença dessas perturbações.

Visando aumentar a robustez de circuitos síncronos, é de consenso geral entre os projetistas o uso de frequências do sinal de relógio inferiores àqueles valores máximos que estes circuitos poderiam suportar. Esta medida é tomada de forma a garantir uma margem de segurança mínima frente ao ruído e as variações (temperatura, umidade e interferência eletromagnética, por exemplo) inerentes ao meio e às variações do processo de fabricação de circuitos integrados. Mas isso não pode ser considerado em circuitos assíncronos, pois estes não possuem sinal de *relógio*. Para circuitos assíncronos com as características que utilizamos em nossos estudos, uma forma para conseguir algo semelhante é aumentando o atraso nos sinais de comando gerados pelos controladores de *handshake* nos circuitos assíncronos.

A proposta descrita a seguir está baseada no aumento do atraso dos circuitos controladores de *handshake*, e tem por finalidade aumentar ainda mais a robustez de circuitos assíncronos frente a grandes amplitudes de ruído. Ao mesmo tempo, esta proposta também apresenta uma técnica que busca relacionar o aumento no atraso dos controladores de *handshake* com a confiabilidade necessária para que estes circuitos atendam *Standards* internacionais para o projeto de eletrônica embarcada imune ao ruído eletromagnético, a saber: IEC 61.0004-29 e IEC 61.0004-17. Neste contexto, faremos uma breve apresentação da lógica de controle, pois é ela quem controla os relógios locais do circuito assíncrono. A seguir apresentaremos a lógica de atraso, ou seja, a definição do atraso referente aos atrasos existentes no circuito tanto pela sua lógica combinacional como também por outros fatores.

8.2 INSERÇÃO DE LÓGICA DE ATRASO BASEADA EM GRAFOS DE TRANSIÇÃO DE SINAIS.

Esta seção tem por objetivo apresentar o uso do Grafo de Transição de Sinais como ferramenta de apoio para tomada de decisão dos pontos adequados para inserção de elementos de atraso nos controladores de *handshake* de circuitos assíncronos.

Iremos fazer a utilização de Grafos Marcados (*Marked Graphs MG*). Eles são uma subclasse de Redes de Petri [111] que são modelos de decisão livre de sistemas concorrentes.

O grafo marcado é uma tripla $(\Sigma, \rightarrow, Mo)$, onde Σ representa o conjunto de eventos, $\rightarrow \subseteq (\Sigma \times \Sigma)$ é o conjunto de arcos e Mo é a marcação inicial que atribui um número de fichas (*tokens*) para os arcos do grafo marcado.

Um evento é ativado quando todos os seus arcos antecessores diretos possuem uma ficha. Um evento é habilitado quando ocorre a eliminação das fichas de cada arco predecessor e adicionam-se fichas para cada arco sucessor. Sendo uma sequência de eventos σ viável se ela pode ser atingida a partir de Mo , sendo denotado por $Mo \xrightarrow{\sigma}$. Uma marcação M' é alcançável a partir de M se existe σ tal que $M \xrightarrow{\sigma} M'$. O conjunto de marcações alcançáveis a partir de Mo é denotado por $[Mo]$.

Para uma maior compreensão, a figura 53 apresenta um pipeline linear. Os pontos pretos que se verificam na figura representam símbolos de dados, enquanto os pontos brancos representam bolhas (estágios do *pipeline* que já computaram seus dados e estão prontos para receber novos dados provenientes do estágio anterior do *pipeline*).

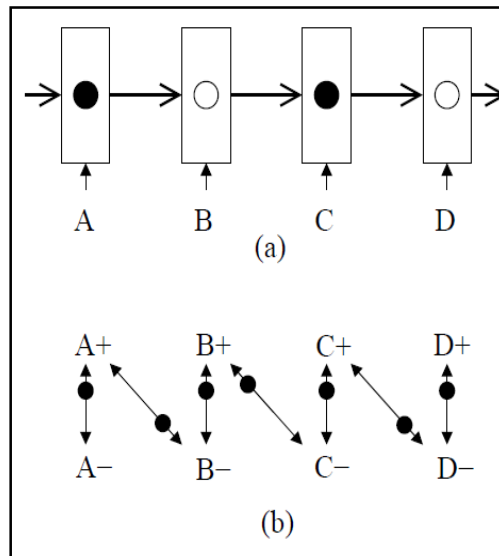


Figura 53 – Modelo de grafos para *pipelines*.

Assume-se que todas as travas se tornam transparentes quando o sinal de controle é elevado. A transição A+ significa elevação de sinal, ou seja, a requisição de

uma ficha do arco A para o arco antecessor. Já A- significa a baixa de sinal, ou seja, que o arco A já está com a ficha proveniente do arco antecessor. A figura 53 (b) apresenta a representação por grafos marcados genéricos do exemplo (a), sendo que o grafo apresentado é um modelo simplificado, por não demonstrar mais de um arco que apresente a mesma dependência.

O grafo da figura 54 (b) é definido pelo protocolo escolhido no projeto DLX ASPIDA dessincronizado, ou seja, com controladores de quatro fases semi-desacoplado, e demonstra o fluxo dos controladores de *latches* mestre-escravo do circuito em estudo, no qual se faz a inserção de atrasos. A escolha foi baseada com o intuito de um modelo que mantivesse o fluxo equivalente de dados e que evitasse impasses. Impasses são todas as situações indesejadas ao sistema, como por exemplo, *deadlocks*. O modelo é o mais adequado para o tipo de implementação escolhida pelos projetistas [102].

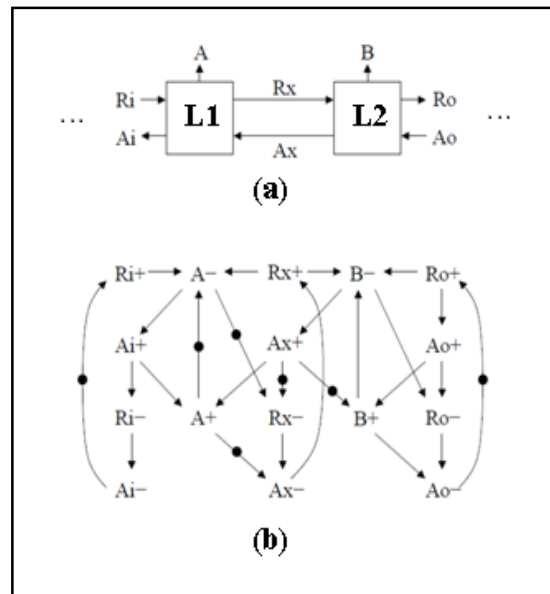


Figura 54 – Gráfico de transição dos controladores.

Os pontos nos quais poderíamos adicionar a lógica de atraso podem ser indicados através da figura 54 (a), que apresenta um dos quatro conjuntos de controladores mestre-escravo que o processador possui. Um local possível seria no caminho de dados entre os próprios controladores mestre-escravo L1 e L2 apresentados na figura. Outro local seria entre um dos controladores mestre-escravo L1 ou L2 com

um controlador antecessor ou sucessor, que são apresentados de forma intuitiva na figura, como por exemplo, entre o controlador L1 apresentado na figura e o controlador L2 antecessor a ele, mas que faz parte do conjunto mestre-escravo antecessor.

Com a inserção de atrasos entre os controladores mestre-escravo L1 e L2, obtemos dependências não permitidas que podem ser evidenciados pelos grafos marcados. Podemos observar a dependência gerada com a inserção de atrasos em L1 e L2 dada por $Ax+ \rightarrow A-$. Essa dependência nos diz que para **A** completar a leitura da ficha é preciso primeiro esperar que a ficha de dados seja armazenado em **Ax**. Caso este não esteja presente, **A** pode ler um dado inválido fazendo com que uma ficha correta seja perdida. Já com a inserção de atraso entre os controladores e seu antecessor, como por exemplo, entre os sinais **Ri** e **Ro** (antecessor) da mesma forma iremos ter uma dependência, mas dada por $Ri+ \rightarrow A-$, sendo essa uma dependência possível, como pode ser observada no modelo da figura 54.

Para garantir que o aumento de atraso no funcionamento dos controladores de *handshake* resultasse no aumento da robustez dos circuitos assíncronos frente ao ruído eletromagnético conduzido, optou-se pela escolha de inserção de atrasos apenas nos locais onde o modelo de Grafos de Transição de Sinais permitisse. A desobediência à esta regra não resulta em um incremento do aumento da robustez dos circuitos assíncronos, tornando ineficaz o aumento de área e consumo de potência resultantes da inserção da lógica de atraso dos controladores de *handshake*.

8.3 LÓGICA DE CONTROLE

Como os circuitos assíncronos não possuem um sinal de relógio, sendo o circuito assíncrono em estudo controlado por um conjunto de controladores mestre-escravo, se fez necessário um estudo sobre como toda essa lógica funciona, para que as escolhas referentes à lógica e principalmente ao atraso que vamos adicionar no circuito estejam corretas. Nessa seção iremos apresentar a técnica proposta para a escolha de onde e como devem ser adicionados os elementos de atraso nos circuitos controladores de *handshake*. Esta abordagem será realizada através da estrutura de controle do DLX ASPIDA dessincronizado.

8.3.1 CONTROLADORES

Na seção 3.4 foi apresentado o controlador no processo de dessincronização do circuito e temos indicado na seção como o controlador faz o controle das travas mestre-escravo do circuito. Também iremos apresentar a estrutura mais interna que temos no processador de estudo. A figura 55 mostra um dos controladores das travas mestre-escravo apresentados na figura 29 da seção 5.4. Ele faz a utilização de controladores *semidescoplados* de quatro fases. Note também que ele faz a utilização do elemento C-Muller para o controle das travas.

No total, o processador DLX ASPIDA dessincronizado faz uso de oito controladores, sendo que eles apresentam todos a mesma estrutura apresentada na figura 55, cada controlador controla um *latch*. A escolha por esse tipo de implementação dos controladores se deu por uma boa troca entre simplicidade do controlador e seu desempenho, simplicidade essa que é refletida no baixo atraso gerado pelos controladores.

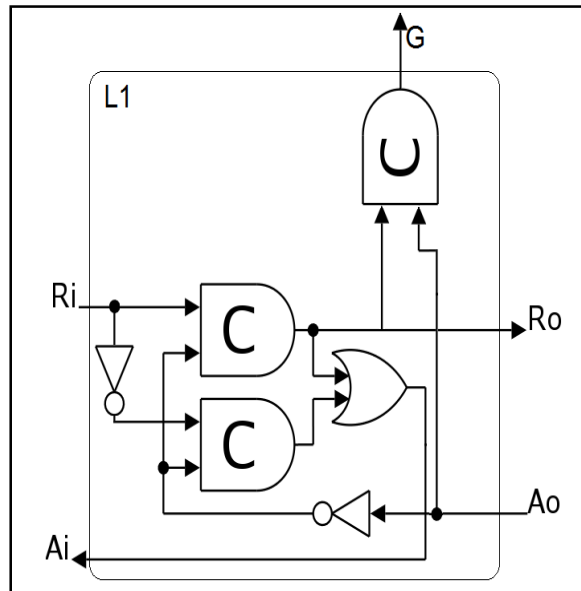


Figura 55 – Estrutura interna dos controladores.

8.3.2 LÓGICA DE ATRASO

Todo o projeto é otimizado de forma incremental para atender todos os requisitos de tempo. Como os blocos computacionais contêm atraso, é necessário que seja feito um cálculo do maior tempo da lógica combinacional, conhecido como o caminho crítico de cada parte do circuito, mas não apenas os atrasos da lógica combinacional, pois temos atrasos também referentes aos *latches*.

A figura 56 mostra o diagrama de tempo para o comportamento dos controladores, os sinais I e S representam respectivamente as entradas e saídas dos *latches*. O sinal L representa o controle do *latch*.

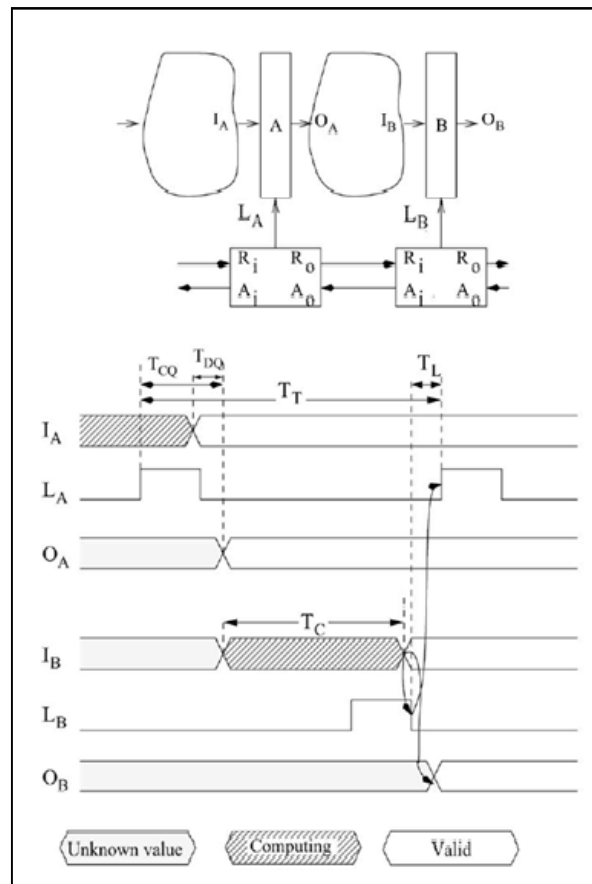


Figura 56 – Restrições de tempo para controladores [12]

Note que a estrutura do circuito DLX ASPIDA dessincronizado é um pouco diferente, pois ela possui dois controladores mestre escravo, o que não é apresentado na figura, mas como o foco está na lógica de atraso, esta é similar a apresentada na figura

abaixo. Como todos os controladores do processador DLX são construídos com a mesma estrutura, eles possuem o mesmo atraso em sua lógica. Nos temos que:

$$\underline{\underline{TT \geq TCQ + TC + TL}}$$

Onde:

TCQ ► atraso de propagação relógio local;

TC ► atraso do bloco computacional;

TL ► atraso dos *latches*.

Como foi utilizado o protocolo de quatro fases, os elementos de atraso utilizados na implementação do circuito são assimétricos. Os elementos de atraso assimétrico possuem o tempo de sinal alto maior que o tempo de sinal baixo, diferentemente do que ocorreria se fossem utilizados atrasos simétricos que possuem o tempo de sinal alto e baixo equivalente. Podemos ver na figura 57 a implementação no circuito do atraso assimétrico. O multiplexador visto na figura é inserido para a configuração após a prototipação.

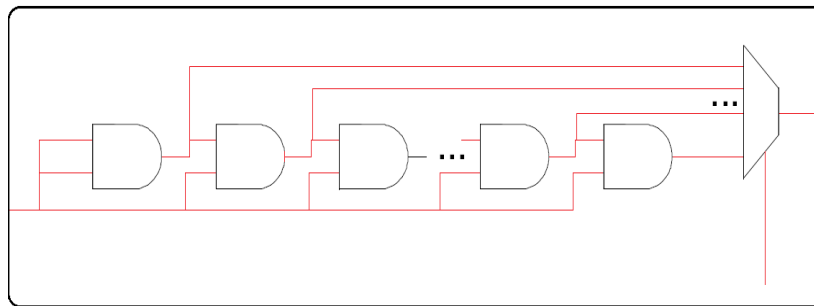


Figura 57 - Implementação dos atrasos.

8.3.3 ADIÇÃO DE ATRASOS

Um grande cuidado quando implementamos as adições de atraso, foi no momento da prototipação, pois para efeitos de performance e otimização, a ferramenta tende a otimizar o código, fazendo com que as adições de atraso se perdessem. Como os

atrasos são baseados em cadeias de portas AND, foram utilizados atributos de síntese para que não ocorra otimização pela ferramenta de prototipação.

Em nosso processador, possuímos quatro *pipelines* funcionais IF, ID, EX e MEM, cada um com sua estrutura de controladores mestre-escravo e seu atraso correspondente. Como foi apresentado na seção anterior que a lógica de atraso é em função principal da lógica combinacional do circuito, sendo os outros fatores (propagação do relógio local e atraso dos *latches*) constantes ao projeto, temos como fator determinante das diferenças de atrasos em cada *pipeline* como sendo a lógica combinacional.

Em projetos síncronos a definição do tempo de relógio se dá pelo tempo necessário para se concluir os caminhos críticos em cada *pipeline* do circuito. Um bom projeto é aquele que possui um valor bem próximo dos tempos do caminho crítico, assim o circuito ganha em desempenho. Como o processador assíncrono advém de um circuito síncrono dessincronizado é intrínseco que os *pipelines* possuíssem praticamente o mesmo atraso, e é o que foi constatado para os *pipelines* IF, ID e EX. O pipeline MEM possui metade do tempo de atraso dos outros estágios de pipeline. Dada essas considerações, nossa proposta se baseia em um aumento proporcional em cada estágio do pipeline.

A figura 58 mostra de forma simplificada a estrutura do processador ASPIDA DLX assíncrono em (b). Podemos ver destacado a rede de controladores mestre-escravo que substitui o relógio global. Em (a) nos temos indicado em vermelho a nossa proposta que aumenta o atraso no circuito.

Como foi visto na seção 9.2, a decisão de seleção dos pontos de inserção da lógica de atraso baseia-se em gráficos de transição de sinais, e que definiram os atrasos mostrados na figura 58. A decisão do local de inserção da lógica de atraso dos controladores de *handshaking* baseou-se no gráfico de transição definidos no trabalho de Cortadella, A. Kondratyev, L. Lavagno [12], que estabelece as dependências possíveis dos controladores da lógica de *handshaking* do processador, como apresentado na figura 54.

Inserções de atrasos na lógica de *handshaking* provocam dependências que, muitas vezes, não são suportadas. Essas dependências que surgem estão relacionadas ao local de inserção dos atrasos e ao protocolo dos controladores de *handshaking*.

Um exemplo de local que provoca dependências que não podem ser suportadas é entres os controladores L1 e L2, mostrados na figura 58 como exemplo. A inserção de

atrasos nesse local provoca dependências do tipo $(Ax+ \rightarrow A-)$. Como podemos ver na figura 54 essa é uma dependência que não pode ser assegurada e foi verificado através de experimentos que a inserção nesse local não traz benefícios ao circuito, apresentando as mesmas quantidades de falhas ou até mesmo uma quantidade maior de falhas do que a do circuito original.

Foram escolhidos então os locais mostrados em vermelho pela figura 58. A inserção de atrasos no local indicado inclui as dependências $(Ri+ \rightarrow A-)$ e $(Ro+ \rightarrow B-)$. Essas dependências podem ser suportadas, como mostra a figura 54.

Com o aumento dos atrasos há um ganho em confiabilidade mas com um pequeno aumento em área e uma diminuição em sua performance, que poderá ser observado mais a frente.

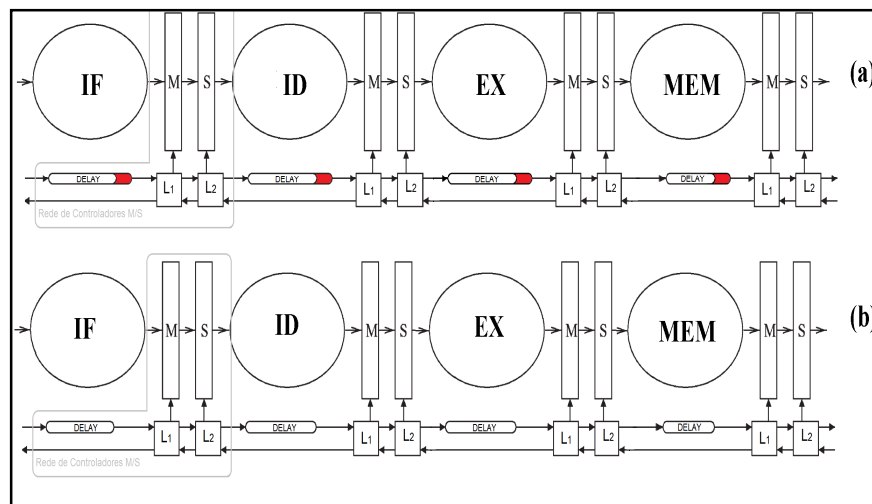


Figura 58 – (a) Inserção dos atrasos (b) Circuito original.

8.4 EXPERIMENTOS PARA VALIDAÇÃO DA PROPOSTA

Os experimentos tiveram como objetivo primeiramente validar a proposta apresentada com relação à robustez de circuitos assíncronos frente ao ruído eletromagnético conduzido, e após isso, com uma série de experimentos apresentarem qual deve ser o atraso a ser acrescentado ao circuito para que ele atenda a um dado nível de ruído definido pelo *standard* IEC. O ambiente de ensaios é similar ao apresentado na figura 48. A figura 59 abaixo apresenta o *setup* de ensaios realizados. Nela podemos

observar o computador que recolhe os dados gerados pelo processador na placa de ensaios que possui o processador em teste, o injetor de falhas e a placa de ensaios.

A simulação escolhida para esses ensaios foi um ruído do tipo queda de tensão de alimentação descrita anteriormente na tabela 6 da seção 3.5.2. Os ruídos foram gerados a partir das linhas de alimentação do processador, conforme a norma IEC 61000-4-29.

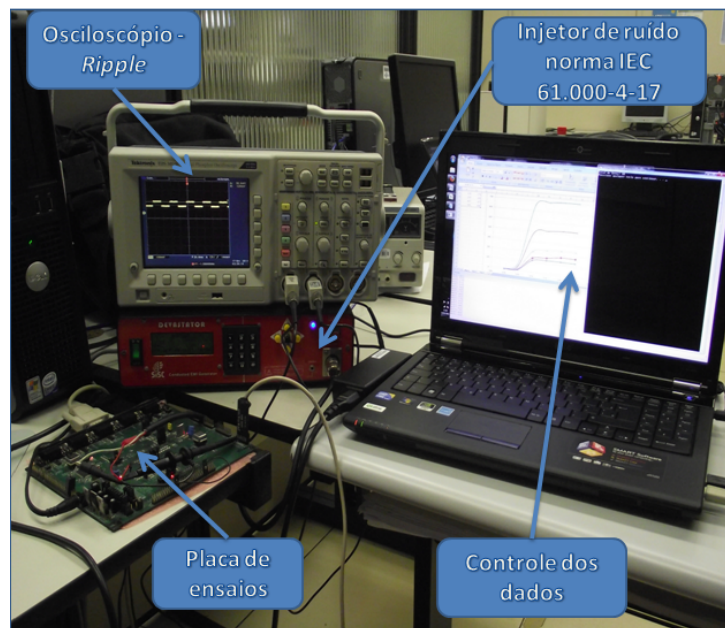


Figura 59 – Setup de ensaios.

A princípio foi escolhida uma amplitude próximo da queda do processador, para avaliar o quanto de falhas funcionais (*failure rate*) o circuito original apresenta sob essas condições de ruídos extremos. Como os objetivos desses experimentos são diferentes dos objetivos antes apresentados do qual se buscava verificar e comparar a robustez do circuito síncrono e sua contraparte assíncrona, estes são realizados de uma forma um pouco diferente da apresentada anteriormente. A seguir a figura 60 apresenta a sequência dos procedimentos dos ensaios.

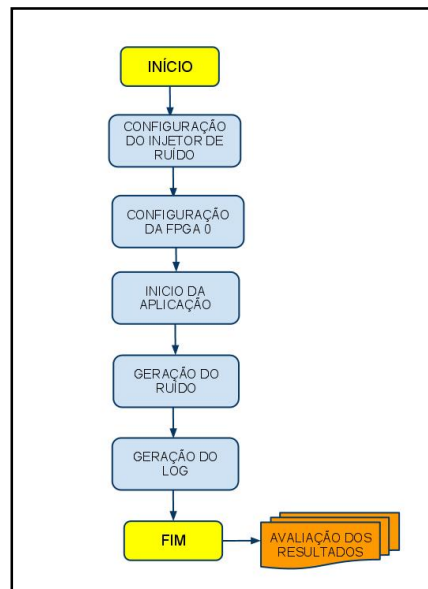


Figura 60 – Fluxo de execução.

A descrição das etapas referente à figura 60 é apresentada abaixo:

- I. **Início** – A plataforma de ensaios é conectada à estação de trabalho e monitoramento (computador portátil), por meio de um canal de comunicação USB - serial (RS-232), o injetor de ruído e a placa de ensaios são energizadas.
- II. **Configuração do injetor de ruído** - O gerador de ruídos é configurado com base na seção 7.3.2 em queda de tensão de alimentação, logo depois o nível de tensão nominal é configurado para 1.2V, o período de temporização em 1ms e a amplitude utilizada nesses ensaios foram diversas e o gerador de ruído a principio foi deixado em *standby*.
- III. **Configuração da FPGA 0** – Nesta etapa o processador DLX assíncrono é carregado em uma das FPGAs, nós utilizamos sempre a FPGA zero. O processador é carregado através do cabo JTAG USB e com a utilização do software ADEPT da *Digilent*;

- IV. **Início da aplicação** – Como nos ensaios realizados anteriormente, utilizamos a multiplicação da multiplicação de uma matriz com dez linhas e colunas, nessa etapa a multiplicação é iniciada com o botão *reset* da placa de ensaios;
- V. **Geração do ruído** –O ruído começa a ser inserido, com o injetor de ruído passando do modo *standby* para o modo de operação;
- VI. **Geração do log** – Após uma hora de ensaio é gerado um *log* com todos os resultados gerados pelo processador em teste;
- VII. **Análise dos dados** – Essa etapa é feita após o fim das aplicações a partir do *log* gerado no fim do experimento. O experimento é baseado em uma multiplicação de matriz em loop com o conhecimento dos resultados. A partir disso, foi criado um software em Java para automatizar o processo e contar o número de falhas ocorridas no período de uma hora.

Outro fator importante a ser considerado em nossa proposta é a área ocupada pelos elementos de atraso inseridos ao longo dos circuitos controladores de *handshake*. Neste sentido, podemos considerar que a inserção de elementos de atraso não é um fator crítico em nossa proposta, já que estes elementos de atraso, na versão original, representam menos de 2% de todo o circuito.

A tabela 12 apresenta a área ocupada pelos diferentes blocos funcionais do circuito assíncrono original. A tabela 13 apresenta um resumo da ocupação de área e de consumo de potência para diferentes configurações dos elementos de atraso em comparação com o circuito original (visto na tabela 12). Por exemplo, se desejássemos incrementar em 30% o atraso dos circuitos controladores de *handshake* com relação ao processador DLX original, isto representaria um acréscimo de 27,74% sobre a área que contém os elementos de atraso no circuito original. Em outras palavras, a área total ocupada pelos elementos de atraso passaria de 1,37% para 1,75% da área ocupada pelo processador DLX: $1,37 \times 1,2774 = 1,75\%$. Disto, conclui-se que o incremento tanto na área quanto no consumo de potência são praticamente desprezíveis quando o aumento da lógica de atraso varia no intervalo de 10 a 50% com relação à lógica do processador

original (no pior caso, para um aumento do atraso da ordem de 50%, observou-se um incremento de 41,60%, passando a área dos elementos de atraso de 1,37 para 1,94% da área total do processador). Como poderá ser observado nas próximas seções deste trabalho, para valores da lógica de atraso dos controladores de *handshaking* acima de 50%, não foram observadas melhorias substanciais no tocante à confiabilidade do circuito e por esta razão, tais valores não foram considerados na tabela 13.

Tabela 12 – Relação de área original do circuito assíncrono DLX ASPIDA.

Componente	Original
DLX	94,69%
UART	2,80%
Outros	1,14%
Elementos de atraso	1,37%

Tabela 13 – Custos de atraso

Versão	Acréscimo em área dos elementos de atraso com relação ao DLX Assíncrono	Aumento no consumo
10%	1,50%	0.035%
20%	1,63%	0.07%
30%	1,75%	0.105%
40%	1,82%	0.14%
50%	1,94%	0.175%

8.5 RESULTADOS

Os primeiros experimentos tiveram como objetivo demonstrar o ganho de robustez da nossa proposta e a procura de um atraso ideal que não comprometesse muito o sistema em desempenho e que tivesse o maior ganho para suportar ruídos extremos como os simulados.

Os ensaios mostrados a seguir foram de queda na tensão de alimentação (*voltage dip*) com período de 1ms, ciclo de trabalho (*duty cycle*) de 50% e amplitude de 40%, tendo o intuito de mostrar a imunidade de equipamentos ou dispositivos eletrônicos alimentados por fontes de corrente contínua externas de baixa tensão. Foram realizados ensaios na versão do DLX ASPIDA com o atraso proposto e com uma versão com um aumento de 50% em relação aos atrasos iniciais. Foram realizadas 100 horas de experimentos com essa configuração. Os resultados obtidos podem ser visualizados pelo gráfico da figura 61. No eixo horizontal temos a versão original e a nossa proposta. No eixo vertical, é apresentada a média de número de falhas por hora (*failure rate*, λ).

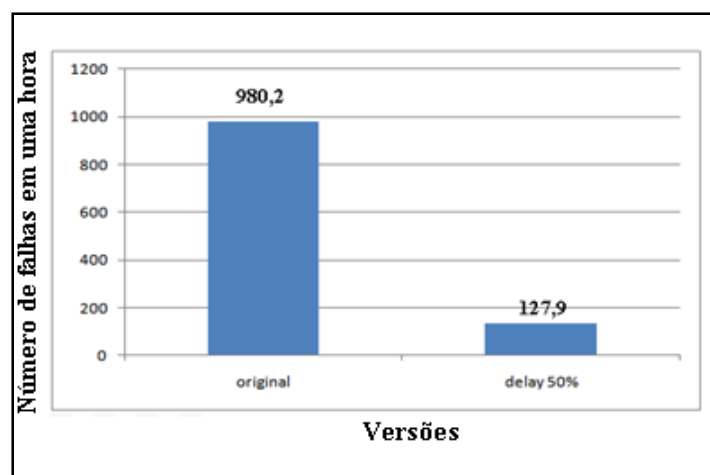


Figura 61 – Gráfico referente às falhas funcionais (*failure rate*).

Com a verificação do ganho de robustez apresentado, foram realizados ensaios com aumentos progressivos de atrasos de 10%, 20%, 30%, 40%, 75% e 100%. Na tabela 14 temos os experimentos realizados em todas as versões do circuito dessincronizado com alteração nos atrasos. Pode ser observado um ganho grande com atraso de 10% e com a diminuição gradativa no gráfico indicado na figura 62.

Através dos resultados obtidos é possível notar uma grande diminuição nas falhas funcionais, mesmo com o acréscimo de pouco atraso(10%), e que as falhas tendem a se estabilizar por volta de um acréscimo de 50% de atraso.

Com o intuito de apresentar o numero de falhas presentes no circuito com o mesmo desempenho, a tabela X apresenta as versões do circuito assíncrono com o número de experimentos realizados levando em consideração o ciclo de relógio. A

terceira coluna da tabela X nós temos apresentado o numero de falhas tendo como base o mesmo número de ciclos do processador em cada versão.

Tabela 14 – Indicação de atrasos.

Atraso acrescentado à lógica de <i>handshaking</i>	Número médio de falhas por hora (<i>failure rate</i> , λ)
Versão original	980.2
10%	698.6
20%	422.5
30%	239.4
40%	176.9
50%	130.3
75%	128.2
100%	127.9

Para medir a confiabilidade de cada abordagem, foi necessário realizar ensaios com diversos níveis de queda de tensão de alimentação. Primeiramente, foi necessário descobrir a partir de que valor de queda de tensão de alimentação o processador DLX começa a apresentar falhas. O gráfico da figura 63 apresenta uma curva para cada nível de atraso acrescentando. Essa fase de ensaios continuou com experimentos de queda de tensão de alimentação com temporização em 1ms e amplitude variável para analisar o comportamento das diferentes implementações do processador DLX em ambientes com diferentes níveis de ruído.

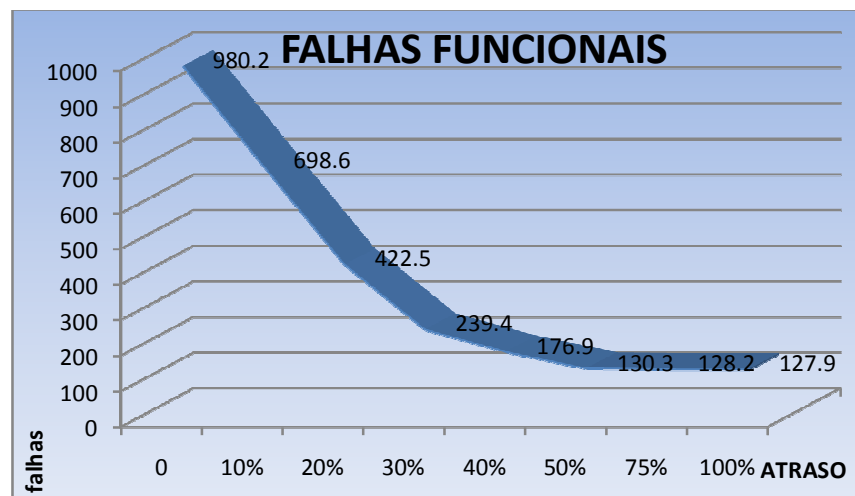


Figura 62 – Falhas Funcionais.

Tabela 15 – Representação de falhas em ciclo de relógio.

Versão	Tempo de experimentos (em ciclos de relógio).	Número de falhas observadas. (3,82 $\times 10^{13}$ ciclos de relógio).
Original	$5,10 \times 10^{13}$	888,18
Atraso 10%	$4,84 \times 10^{13}$	628,74
Atraso 20%	$4,59 \times 10^{13}$	380,25
Atraso 30%	$4,33 \times 10^{13}$	215,46
Atraso 40%	$4,08 \times 10^{13}$	159,21
Atraso 50%	$3,82 \times 10^{13}$	130,34

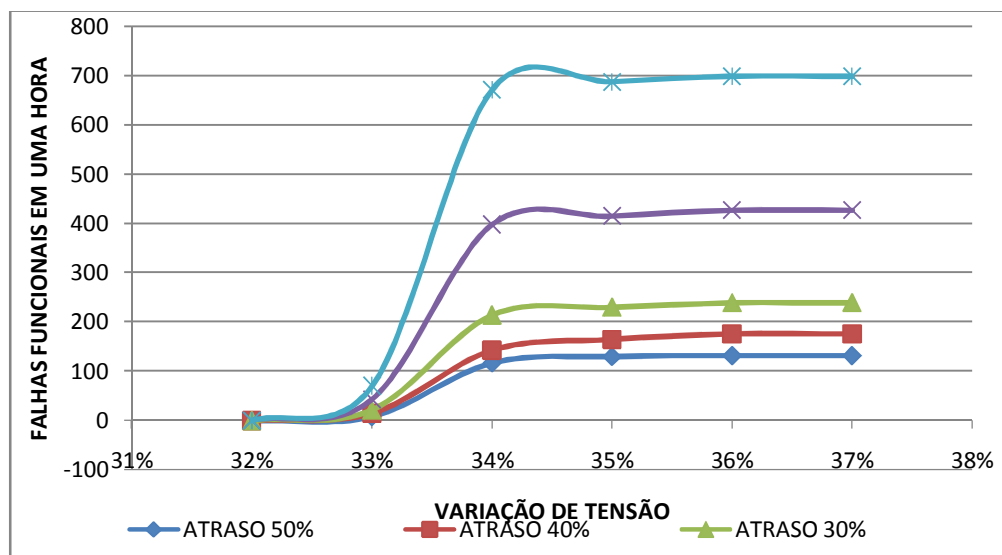


Figura 63 – Confiabilidade do processador DLX em função de diferentes valores para a lógica de atraso e quedas de tensão de alimentação.

Podemos observar através da figura 63, que o gráfico apresenta uma forma logarítmica, com um número crescente de falhas para valores de queda de tensão a partir de 32%, mas praticamente estável para valores acima de 35%, para qualquer tipo de atraso implementado na lógica dos controladores de *handshaking*. A tabela 16 apresenta os dados recolhidos nessa etapa de ensaios.

Tabela 16 – Falhas funcionais com diferentes atrasos e variações.

Número médio de falhas por hora (<i>failure rate</i> , λ)						
Atraso	Dip 32%	Dip 33%	Dip 34%	Dip 35%	Dip 36%	Dip 37%
50%	0	8	117	125	130	130
40%	0	15	143	165	176	176
30%	0	21	214	230	239	239
20%	0	43	398	415	422	422
10%	0	70	671	687	698	698

Dos resultados apresentados na tabela 16, é possível concluir-se que essa proposta pode ser utilizada como uma ferramenta de apoio no projeto da lógica de atraso dos controladores de *handshaking* de circuitos assíncronos, no que diz respeito à estimativa da taxa de falha (*failure rate*) que este circuito apresentará quando operando em ambientes com diferentes níveis de ruído. Em outras palavras, este trabalho apresenta uma metodologia para o projeto da lógica de atraso dos controladores de *handshaking* de circuitos assíncronos tendo como alvo uma determinada taxa de falha e um determinado tipo de ruído ambiente.

Por exemplo, temos que projetar um processador assíncrono para operar em um ambiente exposto a emissões eletromagnéticas que injetam ruído de até 34% nas linhas de alimentação do sistema eletrônico, ao mesmo tempo em que este circuito não poderá apresentar taxa de falha superior a 150 (150 falhas por hora). Neste cenário, o atraso que deve ser acrescido à lógica de controle de *handshaking* não deve ser inferior a 40%.

8.6 CONCLUSÕES DO CAPÍTULO

Este capítulo apresentou uma proposta de alteração nos atrasos dos *pipelines* de circuitos assíncronos para circuitos assíncronos que utilizam a lógica de atraso. Pela estrutura de relógios locais, foram adicionados atrasos proporcionais em cada bloco do circuito. Posteriormente, foram apresentados os resultados obtidos em diversas quantidades de atrasos inseridos.

Os resultados mostraram-se consideravelmente significativos no que diz respeito à confiabilidade e a robustez do circuito, principalmente em ambientes que apresentam alto índice de ruído nas linhas de alimentação do circuito. Neste contexto, observou-se

que é possível aumentar a confiabilidade de circuitos assíncronos às quedas de tensão de alimentação, nitidamente aquelas superiores a 32%, momento esse em que este tipo de circuito começa a apresentar falhas funcionais.

Os resultados obtidos foram positivos, já que se comparado ao original em uma queda na tensão de alimentação com uma amplitude em torno de 40%, o circuito modificado chega a apresentar 85% menos falhas funcionais.

Finalmente, cabe comentar que os resultados apresentados neste capítulo demonstram a viabilidade de se utilizar este trabalho como uma ferramenta de auxílio para o projeto da lógica de atraso dos controladores de *handshaking* de circuitos assíncronos, tendo como alvo uma determinada taxa de falha e um determinado nível de ruído ambiente.

9 CONCLUSÃO

Atualmente, a robustez e a confiabilidade de circuitos integrados e sistemas computacionais são critérios cada vez mais almeçados pela sociedade de consumo. Surge então, a necessidade de novas metodologias que aumentem esses dois fatores durante a fase de projeto destes circuitos e/ou sistemas.

Esse trabalho apresentou uma comparação entre um circuito síncrono e a sua contraparte, assíncrona, no que tange à Compatibilidade Eletromagnética (EMC). Em seguida, apresentou-se uma proposta para modificar os atrasos relacionados à lógica de controle para *handshake* entre os estágios de um circuito pipeline assíncrono.

Circuitos assíncronos em geral se mostram uma escolha interessante no que diz respeito à interferência eletromagnética, comparados a circuitos síncronos, mas a deficiência que esse tipo de paradigma impõe como a não existência de ferramentas de CAD e projetistas não familiarizados com este paradigma, dentre outros, restringe muito o seu uso.

Por outro lado, a técnica de dessincronização, conforme visto na literatura, abre um leque de oportunidades para explorar circuitos assíncronos, pois não há a necessidade de ferramentas apropriadas a esse tipo de circuito e não se necessita de projetistas acostumados com o paradigma assíncrono. Experimentos práticos realizados com essa técnica de dessincronização se apresentaram de forma mais robusta e confiável no que tange à compatibilidade eletromagnética quando comparados com circuitos assíncronos convencionais (não projetados de acordo com a técnica de dessincronização).

Apesar das vantagens apresentadas por esta técnica de dessincronização para circuitos operando em ambientes com ruídos extremos, no qual circuitos síncronos apresentam uma baixa confiabilidade, a partir de um determinado nível de ruído, os circuitos assíncronos também começam a apresentar falhas funcionais. Com isso, o presente trabalho teve o intuito de apresentar uma nova proposta que aumenta ainda mais a confiabilidade de sistemas baseados em circuitos assíncronos. Esta técnica proposta está fundamentada na modificação progressiva dos atrasos da lógica que controla os sinais de *handshaking* entre os estágios de circuitos pipeline assíncronos.

Experimentos práticos confirmaram positivamente o uso da técnica proposta para aumento da confiabilidade de circuitos assíncronos à interferência eletromagnética conduzida. A proposta sugere a utilização de diferentes acréscimos de atrasos na lógica que controla os sinais de *handshaking* entre os estágios de circuitos pipeline assíncronos, de forma a garantir que estes circuitos possam atender às diversas exigências do standard internacional IEC 61.0004-29.

9.1 TRABALHOS FUTUROS

Como sugestões a trabalhos futuros:

- I. Implementar a técnica de dessincronização em um processador de maior complexidade. Uma sugestão de processador seria o PLASMA, muito utilizado em nível acadêmico e que possui uma maior complexidade que o processador DLX ASPIDA. Com isso realizar ensaios de interferência eletromagnética semelhantes às apresentadas nesse trabalho e com modificações em seus atrasos para a análise do desempenho frente a isso.
- II. Uma comparação entre a proposta em questão, com uma implementação que faz a substituição dos atrasos que simulam relógios globais, juntamente com um processo de detecção e conclusão nos blocos funcionais do processador DLX.
- III. A implementação de um circuito que irá detectar interferências nas linhas de alimentação do processador e, com isso, selecionar o atraso que melhor se encaixa nesse tipo de ruído para atingir uma melhor confiabilidade e robustez dos resultados automaticamente tornando o projeto dinâmico. O circuito manteria o aumento de área do maior atraso, mas o desempenho se adequaria com a necessidade para atender ao *Standard*, tornando o circuito mais robusto e perdendo apenas o desempenho necessário.

10 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] WOLF, W. Embeddedisthe new paradigms*Computer, IEEE Computer Society*, Los Alamitos, CA, USA, v. 37, n. 3, p. 99_101, 2004. ISSN 0018-9162.
- [2] SEMIÃO, J. ET AL. *Power-supply instability aware clock signal modulation for digital integrated circuits*. In: EMC Europe 08. Hamburg, Germany: [s.n], 2008.
- [3] MORAES, M. L. Validação de Uma Técnica Para o Aumento da Robustez de SoC's a Flutuações de Tensão no Barramento de Alimentação. Dissertação (Dissertação de Mestrado) Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2008. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.
- [4] MIREMADI, G., TORIN, J., “*Evaluating Processor-Behavior and Three Error-Detection Mechanisms Using Physical Fault-Injection*”. IEEE Transactions on Reliability, Vol. 44, No. 3, September 1995.
- [5] TAUBIN, A.; CORTADELLA, J.; LAVAGNO, L.; KONDRATYEV, A.; PEETERS, A. *Design Automation of Real-Life Asynchronous Devices and Systems*. Now Publisher Inc, 2007.
- [6] SPARSO, J. and FURBER, S. (Eds). “*Principles of asynchronous circuit design - A systems perspective*”. London: Springer, 2002, 360p.
- [7] HSUEH, M. TSAY T., IYER, R. , “*Fault Injection Techniques and Tools*”, IEEE Computer, Vol. 30, No. 4, pp. 75-82, 1997
- [8] IEC *International Electrotechnical Commission. Electromagnetic Compatibility (EMC) - Part 4-29: Testing and Measurement Techniques - Voltage Dips, Short Interruptions and Voltage Variations on d.c. Input Power Port Immunity Tests (61.000-4-29)*. Geneva, Switzerland, 2000. P37, Norma Técnica.
- [9] IEC *International Electrotechnical Commission. Electromagnetic Compatibility (EMC) - Part 4-17: Testing and Measurement Techniques – Ripple on d.c. Input Power Port Immunity Test (61.000-4-17)*. Geneva, Switzerland, 2002. P27, Norma Técnica.
- [10] BOUESSE, F.; NINON, N.; SICARD, G.; RENAUDIN, M.; BOYER, A.; SICARD E. *Asynchronous logic Vs Synchronous logic: Concrete Results on Electromagnetic Emissions and Conducted Susceptibility*. EMC Compo 2007: 5th International Workshop on Electromagnetic Compatibility of Integrated Circuits, Turin, Italy, 2007.
- [11] ANDRIKPS, N. et al. A fully-automated Desynchronization flow for synchronous circuits. In: Proc. 44th ACM/IEEE Design Automation Conference DAC `07. [S.l.: s.n.], 2007. P. 982-985.
- [12] CORTADELLA, J. et al. *Desynchronization: Synthesis of asynchronous circuits from Synchronous specifications*. IEEE Transactions on Computer-aided Design of integrated circuits and Systems, v.25, n.10, p. 1904-1921, 2006.
- [13] CRISTÓFOLI, L. F.; HENGLES, A.; BENFICA, J.; BOLZANI, L.; VARGAS, F.: Atienza. A.; Silva, F. *On the Comparison of Synchronous versus Asynchronous Circuits under the Scope of Conducted Power-Supply Noise*. 2010 Asia-Pacific International Symposium on Electromagnetic Compatibility & technical Exhibition on EMC RF/Measurements and Instrumentation. April 10-12, 2010. Beijing, China. (www.apemc2010.org).
- [14] CRISTÓFOLI, L. F.; HENGLES, A.; BENFICA, J., BOLZANI, L.; VARGAS, F.: Atienza. A.; Silva, F. *Synchronous versus Asynchronous Circuits Reliability under*

- Radiated Eletromagnetic Disturbance*. 20th International Wroclaw Symposium on EMC. September 13-17, 2010.
- [15] PRADHAN, DHIRAJ K. *Fault-Tolerant Computer System Design*. s.l. Prentice Hall, 1996.
- [16] LAPRIE, J. *Dependable computing and fault-tolerance: Concepts and terminology*. In: 15th IEEE Int. Symp. on Fault-Tolerant Computing. New York: IEEE, 1985. s.n., p. 211.
- [17] ANDERSON T. E LEE, P. *Fault Tolerance - Principles and Practice*. [S.l.]: Prentice- Hall, 1981.
- [18] PRADHAN, D. K. *Fault-Tolerant Computer System Design*. [S.l.]: Prentice Hall, 1996.
- [19] BARDELL, P. H. *Built in Test for VLSI: Pseudorandom Techniques*. New York : s.n., 1987.
- [20] STROUND, E. C. *A Designer's Guide to Built-In Self-Test*. Boston : Kluwer Academic Publishers, 2002. pp. 15-27.
- [21] CORTNER, J. M. *Digital Test Engineering*. United States of America : Wiley-Interscience, 1987. pp. 1-27.
- [22] TEMMELTSHAMMER, P., A. STEININGER, "Power Supply Induced Common Cause Faults-Experimental Assessment Of Potential Countermeasures", Dependable Systems & Networks, 2009. DSN '09. IEEE/IFIP International Conference, pp. 449-457, June 29 2009-July 2 2009.
- [23] LAPRIE, J. C. *Dependability: From concepts to limits*. In: *Proceedings of the IFIP International Workshop on Dependable Computing and its Applications* Johannesburg, South Africa: [s.n.], 1998. p. 108-126.
- [24] G. MIREMADI, J. TORIN. "Evaluating Processor-Behavior and Three Error-Detection Mechanisms Using Physical Fault-Injection". IEEE Transactions on Reliability, Vol. 44, No. 3, September 1995.
- [25] J. KARLSSON, U. GUNNEFLO, P. LIDEN, J. TORIN, "Two Fault Injection Techniques for Test of Fault Handling Mechanisms", Test Conference, 1991, Proceedings, International, pp. 140, 26-30 Oct 1991.
- [26] BOSCH, R. *Eletromagnetic Compatibility – Definition and Need*. S. India: Sungaya, 2004
- [27] IEC *International Electrotechnical Commission. International Electrotechnical Vocabulary – Part 161*. Geneva, Switzerland, 1990. P 66, Norma.
- [28] WESTON, D. *Electromagnetic Compatibility: Principles and Applications*. 2a. ed. New York, USA: Marcel Dekker, Inc, 2001. 1-17 p. ISBN 0-8247-8889-3.
- [29] SOUZA, A. M. Eletromagnetismo, Notas de Aula. Dezembro 2007. Online. Disponível em: <http://cict.inatel.br>
- [30] KRAUSS, J. D.; FLEISCH, D. *Electromagnetics With Aplications*. [S.l.]: Mc Graw-Hill, 1999.
- [31] ZEVZIKOVAS, M. Efeitos da Interferência Eletromagnética Conduzida em Equipamentos Eletro médicos no Ambiente Hospitalar. Dissertação (Mestrado) Universidade Estadual de Campinas - UNICAMPI, São Paulo, 2004. P. 86, Programa de Pós- Graduação em Engenharia Elétrica e de Computação - Dissertação de Mestrado.
- [32] NOURANI M, A. A. *Signal integrity: Fault modeling and testing in high-speed SoC's*. *Journal of Electronic Testing*, v. 18, n. 4-5, p. 539_554, Agosto 2002.
- [32] FANG PENG, T. J. C. J.; HU, C. *Design in hot-carrier reliability for high performance logic applications*. In: *Custom Integrated Circuits Conference*. Santa Clara, CA, USA: IEEE, 1998. Custom Integrated Circuits Conference, p. 525_531. ISBN: 0-7803-4292-5.

- [33] LEBLEBICI, Y. *Design considerations for cmos digital circuits with improved hotcarrier reliability*. Journal of Solid-State Circuits, v. 31, p. 1014_1024, Julho 1996.
- [34] ZHAO SHIYOU, R. K. *Estimation of switching noise on power supply lines in deep sub-micron Cmos circuits*. In: 13th International Conference on VLSI Design. Los Alamitos, CA, USA: IEEE Computer Society, 2000. p. 168. ISSN: 1063-9667.
- [35] VARGAS, F. et al. *Emi - induced soft error rate for cots microprocessor*. In: 5th IEEE Latin American Test Workshop. [S.l.: s.n.], 2004. p. 169_172.
- [36] CHEN, H. H.; WANG, L. K. *Design for signal integrity: The new paradigm for deep submicron VLSI design*. In: Proc. Technical Papers VLSI Technology, Systems, and Applications 1997 International Symposium on. [S.l.: s.n.], 1997. p. 329_333.
- [37] CHO, D. H. et al. *Interconnect capacitance, crosstalk, and signal delay for 0.35 & cmos technology*. In: Proc. International Electron Devices Meeting. [S.l.: s.n.], 1996. p. 619_622.
- [38] RODRIGUEZ-IRAGO, M. et al. *Dynamic fault test and diagnosis in digital systems using multiple clock schemes and multi-vdd test*. In: 11th IEEE International On-Line Testing Symposium - IOLTS. Los Alamitos, CA, USA: IEEE Computer Society, 2005. p. 281-286. ISSN: 1530-1591.
- [39] IEC *International Electrotechnical Commission. Electromagnetic Compatibility (EMC) - Part 4-17: Testing and Measurement Techniques – Ripple on d.c. Input Power Port Immunity Test (61.000-4-17)*. Geneva, Switzerland, 2002. P27, Norma Técnica.
- [40] *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.
- [41] MARTIN, G.; CHANG, H. SHANHAI, *System on Chip Design*. China: IEEE, 2001. *International Symposium on Integrated Circuits, Devices & Systems - ASIC'01*. pp. 12-17. ISBN: 0-7803-6677-8.
- [42] REIS, AUGUSTO, R. L.. *Concepção de Circuitos Integrados*. Porto Alegre: Sagra Luzzatto, 2000. p. 252. ISBN 8524106255.
- [43] BERGAMASCHI, R. A.; LEE, W. R.;. *Designing Systems-On-Chip Using Cores*. Los Angeles, California, United States : ACM, 2000. Design Automation Conference - DAC'00. pp. 420-425. ISBN: 1-58113-187-9.
- [44] BERGAMASCHI, R. A.; BHATTACHAYA, S.; WAGNER, R.; FELLEENZ, C.; MUHLADA, M.; WHITE, F.; DAVEAU, J. M.; Lee, W. R.; *Automating the Design of SoC's Using Cores*. 2001. IEEE Design & Test of Computers. Vol. 18(5), pp. 32 - 45.
- [45] MARTIN, G.; CHANG, H. SHANGHAI, China : IEEE, 2001. *System on Chip Design*. International Symposium on Integrated Circuits, Devices & Systems - ASIC'01. pp. 12 17. ISBN: 0-7803-6677-8.
- [46] JUNEID, Z.; TORKI, K.; MARTINEZ, S.; NICOLESCU, G.; COURTOIS, B.; JERRAYA, A.; *Global Modeling and Simulation of System-on-Chip Embedding MEMS Devices*. Shanghai, China:IEEE, 2001. International Symposium on Integrated Circuits, Devices & Systems - ASIC'01. pp. 666 - 669.
- [47] BENINI, L.; De MICHELI, G. Montréal, *Powering Networks on Chip: Energy-Efficient and Reliable Interconnect Design for SoC's*. P.Q., Canada : ACM, 2001. International Symposium on System Synthesis. pp. 33 - 38. ISBN:1-58113-418-5.
- [48]. RUNNER, S.; SANAKA, V.; Yu, E;. *Building an Infrastructure for IP Reuse*. [Online] 15 de Maio de 2000. <http://www.eetimes.com>.
- [49] The National Technology Roadmap for Semiconductors. [Online] 1999. <http://public.itrs.net>.

- [50] SALLOUM C. E., STEININGER A., TUMMELTSHAMMER P., *Recovery mechanisms for dual core architectures*. In DFT '06: Proceedings of the 21 st IEEE International Symposium on Defect and Fault – Tolerance in VLSI Systems, pages 380-388. IEEE computer Society, 2006.
- [51] BERGAMASCHI, R.; COHN, J. The a to z of SoC's. In: *International Conference on Computer Aided Design - ICCAD*. [S.l.]: IEEE / ACM, 2002. p. 791-798. ISSN: 1092-3152.
- [52] BERGAMASCHI, R. A.; LEE, W. R. *Designing systems-on-chip using cores*. In: *Design Automation Conference - DAC'00*. Los Angeles, California, United States: ACM, 2000. p. 420-425. ISBN: 1-58113-187-9.
- [53] *The National Technology Roadmap for Semiconductors*. [Online] 1999. <http://public.itrs.net>.
- [54] OST, L. C. *Redes Intra-Chip Parametrizáveis com Interface Padrão para Síntese em Hardware*. 134 p. Dissertação (Dissertação de Mestrado) _ Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2004. Faculdade de Informática, Programa de Pós-Graduação em Ciência da Computação.
- [55] GUPTA, R. K.; ZORIAN, Y. *Introducing core-based system design*. In: *IEEE Design & Test of Computers*. [S.l.: s.n.], 1997. v. 14, n. 4, p. 15-25.
- [56] KUMAR, S. *Networks on chip*. In: [S.l.]: *Kluwer Academic Publishers*, 2003. cap. 5 - *On packet switched networks for on-chip communication*, p. 85_106.
- [57] THOMAS D. E MOORBY, P. *The Verilog Hardware Description Language*. [S.l.]: Kluwer Academic Publishers, 1991. 376 p.
- [58] PERRY, D. *VHDL*. 3rd edition. ed. New York, USA: McGraw-Hill, 1998. 493 p.
- [59] SCHILDT, H. C *Completo e Total*. 3 ed. ed. São Paulo: [s.n.], 1997. 827 p.
- [60] SOCIETY, I. C. *IEEE Standard SystemC Language Reference Manual*. New York, USA: IEEE Computer Society, 2006. 423 p. ISBN 0-7381-4870-9 SS95505. Disponível em: <<http://standards.ieee.org/getieee/1666/download/1666-2005.pdf>>. 135
- [61] GAJSKI, D. et al. *Specification Language and Methodology*. Norwell, MA: Kluwer Academic Publishers, 2000. 336 p.
- [62] MADISETTI, V. K.; L, S. *Interface design for core-based systems*. In: *IEEE Design & Test of Computers*. [S.l.: s.n.], 1997. v. 14, n. 4, p. 45_51.
- [63] 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.
- [64] International Sematech. *International Technology Roadmap for Semiconductors Update*. [Online] 2002. <http://www.itrs.net/>.
- [65] 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.
- [66] OST, L. C. *Redes Intra-Chip Parametrizáveis com Interface Padrão para Síntese em Hardware*. 134 p. Dissertação (Dissertação de Mestrado) _ Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2004. Faculdade de Informática, Programa de Pós-Graduação em Ciência da Computação.
- [67] GUERRIER, A. G. P. *A generic architecture for on-chip packet-switched interconnections*. In: *Design Automation and Test in Europe - DATE'00*. Paris, França: [s.n.], 2000. p. 250_256. ISBN: 0-7695-0537-6.
- [68] HWANG, K. *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. New York : McGraw-Hill, 1993. p. 771.

- [69] BENINI, L.; MICHELI, G. D. *Powering networks on chip: Energy-efficient and reliable interconnect design for SoC's*. In: *International Symposium on System Synthesis*. Montréal, P.Q., Canada: ACM, 2001. p. 33-38. ISBN: 1 58113-418-5.
- [70] BENINI L., D. M. G. *Networks on chips: a new soc paradigm*. *Computer*, v. 35, n. 1, p. 70_78, Jan. 2002.
- [71] KUMAR, S. et al. *A network on chip architecture and design methodology*. In: *Computer Society Annual Symposium on VLSI - ISVLSI'02*. [S.l.: s.n.], 2002. p. 105-112.
- [72] GUERRIER, A. G. P. *A generic architecture for on-chip packet switched interconnections*. In: *Design Automation and Test in Europe - DATE'00*. Paris, França: [s.n.], 2000. p. 250-256. ISBN: 0-7695-0537-6.
- [73] IBM. *The Core Connect Bus Architecture*. Dezembro 2007. Online. Disponível em: <<http://www.ibm.com/chips/products/coreconnect>>.
- [74] Xilinx Inc. *Designing Custom OPB Slave Peripherals for MicroBlaze*. Fevereiro 2002. Online. Disponível em: <http://www.xilinx.com/ipcenter/processor_central/microblaze/doc/opb_tutorial.pdf>.
- [75] ARM Corp. *AMBA 2.0 Specification*. 2007. Online. Disponível em: <<http://www.arm.com/products/solutions/AMBAHomePage.html>>.
- [76] ALTERA. *Avalon Bus Specification - Reference Manual*. Dezembro 2007. Online. Disponível em: <http://www.altera.com/products/software/products/sopc/avalon/nioavalon_features.html>.
- [77] MORAES, F. G. et al. *Nocgen - Uma ferramenta para geração de redes intra-chip baseada na infra-estrutura Hermes*. In: *X Workshop IBERCHIP*. Cartagena, Colômbia: [s.n.], 2004. v. 1, p. 210_216.
- [78] MORAES, F. G.; CALAZANS, N. L. V.; MELLO, A. V.; MOLLER, L. H.; OST, L. C. "*Hermes: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip*". *Integration the VLSI Journal*, 38(1), pp. 69-93, Oct. 2004.
- [79] BJERREGAARD, T.; MAHADEVAN, S. "*A Survey of Research and Practices of Network-on-Chip*". *ACM Computing Surveys*, 38(1), pp. 1-51, 2006.
- [80] ROSTISLAV, D.; VISHNYAKOV, V.; FRIENDMAN, E.; GINOSAR, R. "*An asynchronous router for multiple service levels networks on chip*". In: *IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC05)*, pp. 44-53, March 2005.
- [81] BEIGNÉ, E.; CLERMIDY, F.; VIVET, P.; CLOUARD, A.; RENAUDIN, M.; "*An Asynchronous NoC Architecture Providing Low Latency Service and its Multi-level Design Framework*". In: *IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC05)*, pp. 54-63, 2005.
- [82] Xilinx Inc. *Xilinx Data Book*. Outubro 2010. Online. Disponível em: <<http://www.xilinx.com/support/documentation/index.htm>>.
- [83] ZEIDMAN, B. *Introduction to CPLD and FPGA Design*. 2001. Online. Disponível em: <<http://www.scribd.com/doc/11572884/Introduction-to-CPLD-and-FPGADesign>>
- [84] FANG, D.; TEIFEL, J.; MANOHAR, R. "*A High-Performance Asynchronous FPGA: Test Results*". In: *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM05)*, pp. 271- 272, April 2005.
- [85] PAYNE, R. "*Asynchronous FPGA architectures*". *IEE Proceedings - Computers and Digital Techniques*, 143(5), pp. 282- 286, September 1996.
- [86] TEIFEL, J.; MANOHAR, R., "*An asynchronous dataflow FPGA architecture*". *IEEE Transactions on Computers*, pp. 1376-1392, November 2004.
- [87] TOROK, D. L. *Projeto Visando a Prototipação do Protocolo de Acesso ao Meio em Redes Ethernet*. Dissertação (Dissertação de Mestrado) _ Pontifícia Universidade

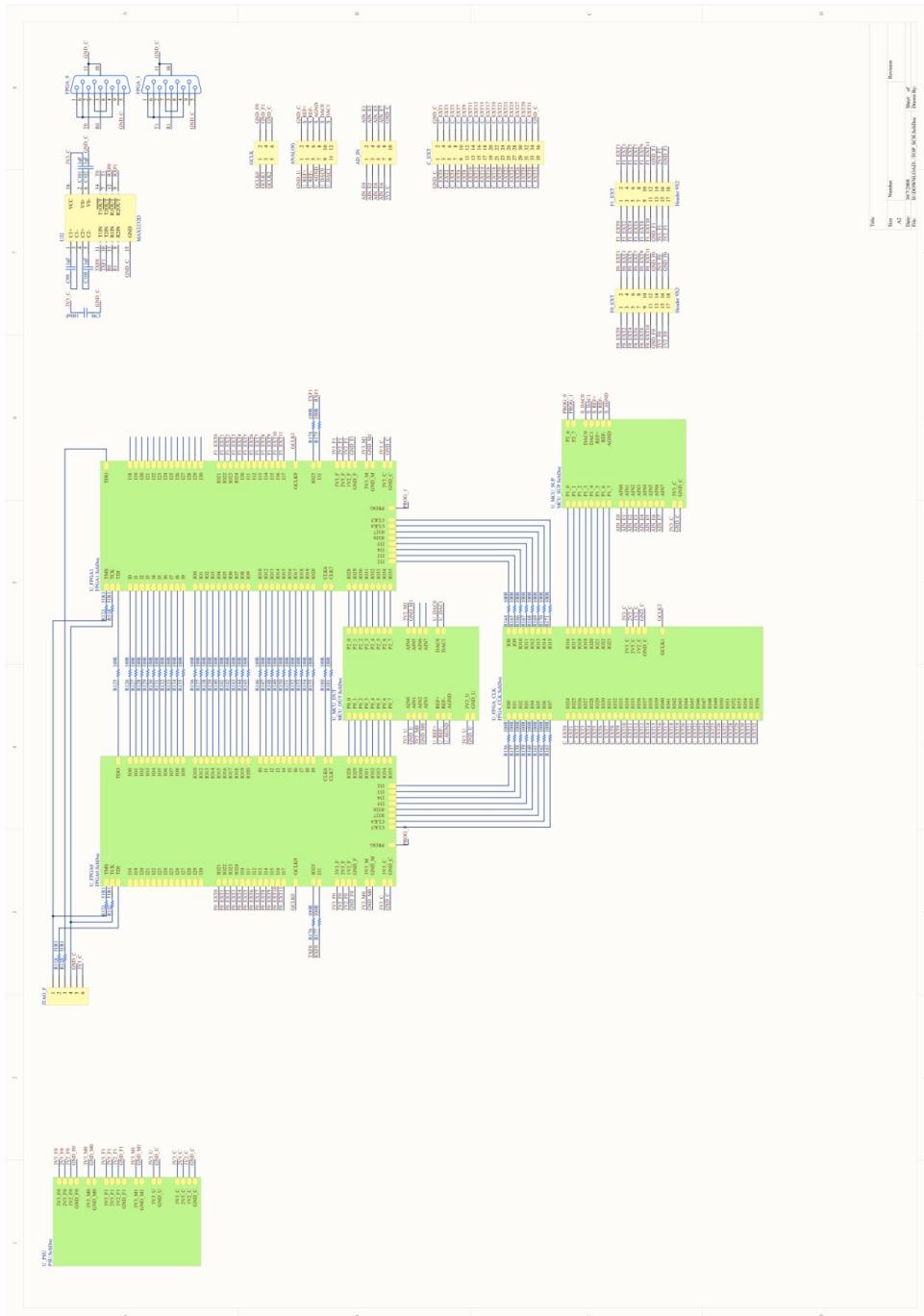
- Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2001. Faculdade de Informática, Programa de Pós-Graduação em Ciência da Computação.
- [88] ROSE, J.; GAMAL, A. E.; SANGIOVANNI-VINCENTELLI, A. *Architecture of Field programmable*. In: Proceedings of the IEEE. [S.l.: s.n.], 1993. Vol. 81, p. 1013-1029.
- [89] ARAGÃO, A. C. O. S. Uma Arquitetura Sistólica para Solução de Sistemas Lineares Implementada com Circuitos FPGA's. Dissertação (Dissertação de Mestrado) _ Universidade de São Paulo - USP, 1998. Instituto de Computação e Matemática Computacional.
- [90] BENFICA, J. D. Plataforma para Desenvolvimento de SoC's (System-on-Chip) Robusto à Interferência Eletromagnética. Dissertação (Dissertação de Mestrado) _ Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2007. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.
- [91] BERKEL, C. H. V.; JOSEPHS, M. B.; NOWICK, S. M. *Applications of asynchronous circuits*. Proceedings of the IEEE, v. 87, n. 2, p. 223_233, Feb. 1999.
- [92] LAVAGNO, L.; SANGIOVANNI-VINCENTELLI, L. *Algorithms for Synthesis and Testing of Asynchronous Circuits*. [S.l.]: Kluwer Academic Publishers, 1993. (Kluwer Academic Series in Engineering and Computer Science).
- [93] CALAZANS, N. L. V. "Automated Logic Design of Sequential Digital Circuits". Rio de Janeiro: Imprinta, 1998. 342p.
- [94] WESTE, N.; ESHRAGHIAN, K. "Principles of CMOS VLSI Design". Boston: Addison- Wesley, 2nd edition, 1994, 735 p.
- [95] SUTHERLAND, I. *Micropipelines*. In: *Communication of the ACM*. [S.l. s.n.], 1989. v. 32, n. 6.
- [95] MARTIN, A. J.; NYSTROM, M. "Asynchronous techniques for system-on-chip design". Proceedings of the IEEE, 94(6), pp. 1089-1120, June 2006.
- [96] MULLER, D.; BARTKY, W. *A theory of asynchronous circuits*. In: Proc. Int. Symp. Theory of Switching. [S.l.]: Harvard Univ. Press, 1959. p. 204-243.
- [97] RENAUDIN, M.; VIVET, P.; ROBIN, F. ASPRO-216: A standard-cell QDI 16-bit RISC asynchronous microprocessor. In: *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*. [S.l.: s.n.], 1998. p. 22_31.
- [98] MARTIN, A. J. et al. *The design of an asynchronous MIPS R3000 microprocessor*. In: *Advanced Research in VLSI*. [S.l.: s.n.], 1997. p. 164-181.
- [99] DUC, A.V.D. et al. *Tast cad tools: Tutorial*. In: Int. Symp. on Advanced Research in Asynchronous Circuits and Systems ASYNC'02. Manchester, UK: [s.n.], 2002. (TIMA internal report).
- [100] MARTIN, A. J.; NYSTRÖM, M. *Cast: Caltech asynchronous synthesis tools*. In: Proc. of Fourth Asynchronous Circuit Design Working Group Workshop. Turku, Finland: [s.n.], 2004.
- [101] BARDSLEY, A.; EDWARDS, D. A. *The Balsa asynchronous circuit synthesis system*. In: Forum on Design Languages. Tübingen, Germany: [s.n.], 2000.
- [102] ANDRIKOS, N. A Fully-Automated Desynchronization Flow for Synchronous Circuits. Dissertação (Master's Thesis) University of Crete, Heraklion, Greece, February 2006. P. 45.
- [103] T. MURATA, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541-574, April 1989.
- [104] PRESTES, D. Plataforma para Injeção de Ruído eletromagnético Conduzido em Circuitos Integrados Dissertação (Dissertação de Mestrado) _ Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2010. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.

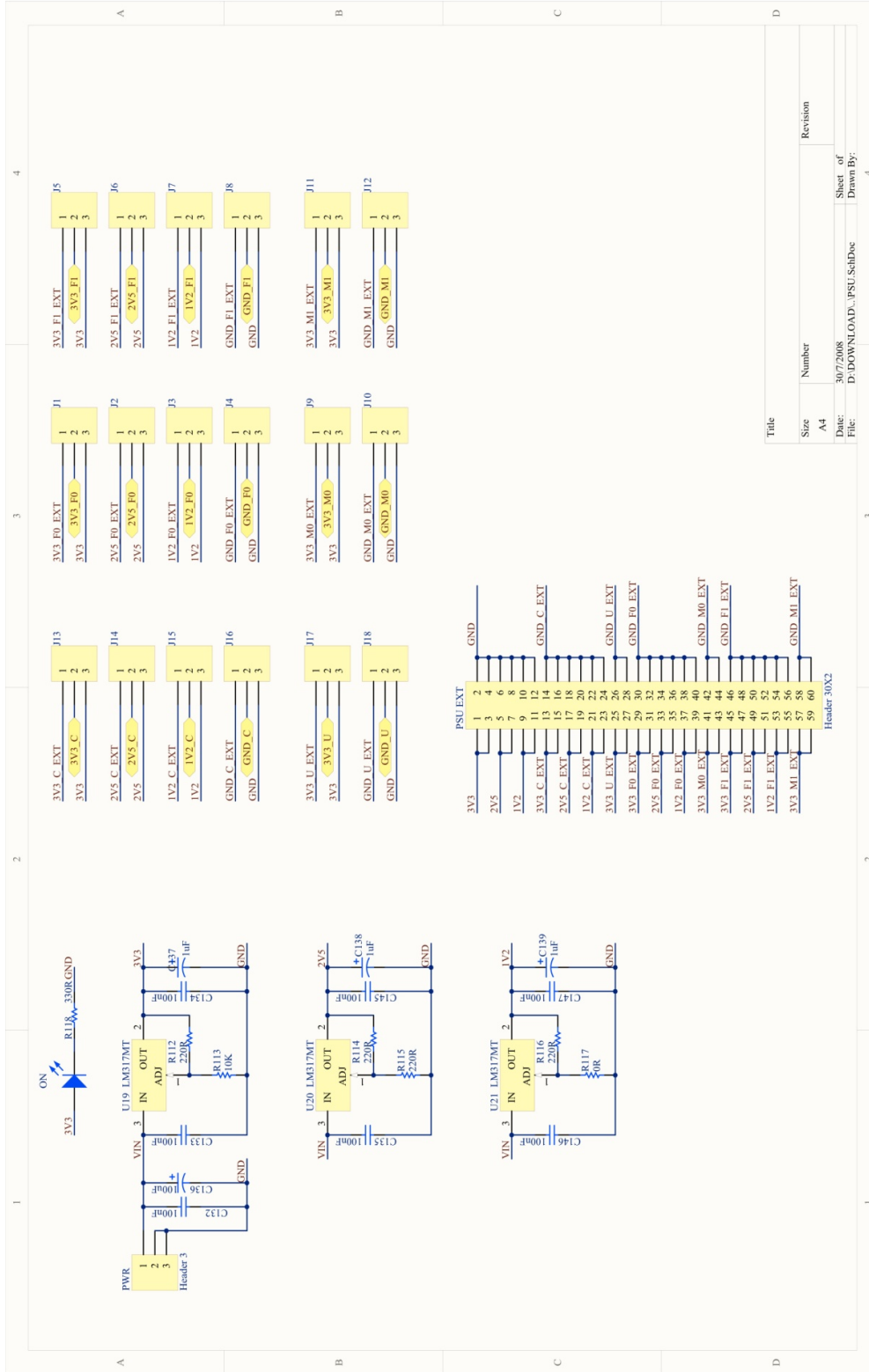
- [105] SOTIRIOU, C. et al. *Asynchronous Open-Source DLX Processor (ASPIDA)*. 2004. Online. *Asynchronous DLX Demo at ASYNC 2004 Conference*. Disponível em: <<http://www.ics.forth.gr/carv/async/demo/>>. SOTIRIOU, C. ASPIDA sync/async DLX Core: DLX ISA CPU with SYNCHRONOUS and ASYNCHRONOUS Implementations. Sep 2005. Online. Disponível em: <<http://www.opencores.org/Projects.cgi/web/aspida/overview>>.
- [106] HENNESY, J.; PATTERSON, D. *Computer Architecture: A Quantitative Approach*. San Mateo, CA: Morgan Kaufmann, 1990.
- [107] MIPS Technologies, Inc. MIPS Technologies. 2010. Online. Disponível em: <<http://www.mips.com/>>.
- [108] I. BLUNNO, J. CORTADELLA, A. Kondratyev, *Handshake protocols for Desynchronization* Disponível em: <http://www.lsi.upc.edu/~jordicf/gavina/BIB/files/async04.pdf>
- [109] LOPES, D. C. Estimação da Robustez de Sistemas Eletrônicos Via Injeção de Falhas por Interferências Eletromagnéticas. 97 p. Dissertação (Mestrado) _ Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2005. Programa de Pós-Graduação em Engenharia Elétrica, Faculdade de Engenharia - Dissertação de Mestrado.
- [110] ROCHA, C. A. Monitoramento do Fluxo de Controle de Processadores Embarcados Baseado em *Profiling* de Software. Dissertação (Dissertação de Mestrado) Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, Porto Alegre, 2007. Faculdade de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica.
- [111] Murata, T., Petri Nets: *Properties, analysis and applications*. *Proceedings of the IEEE*, pages 541–580, Apr. 1989.

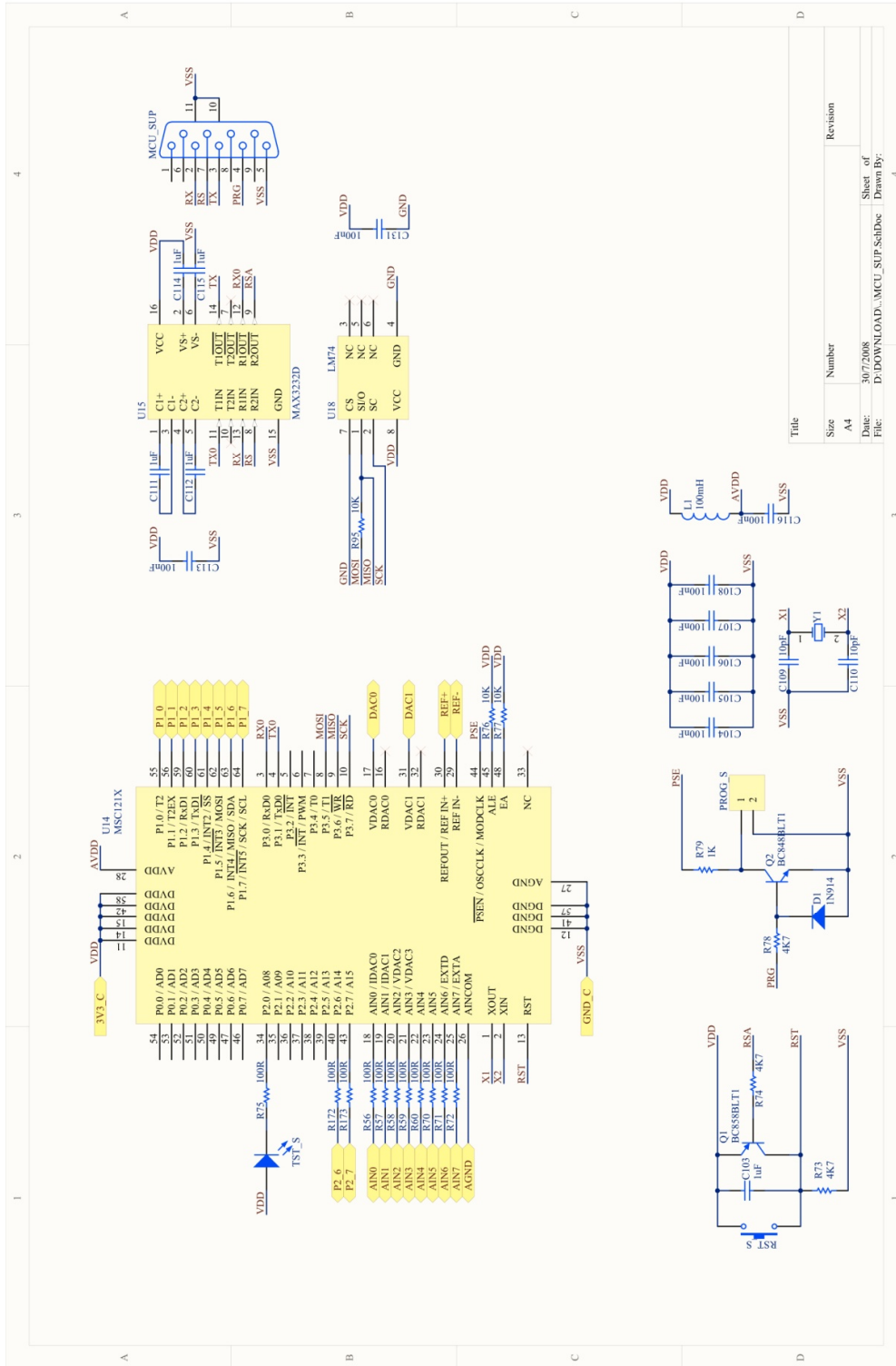
11 ANEXOS

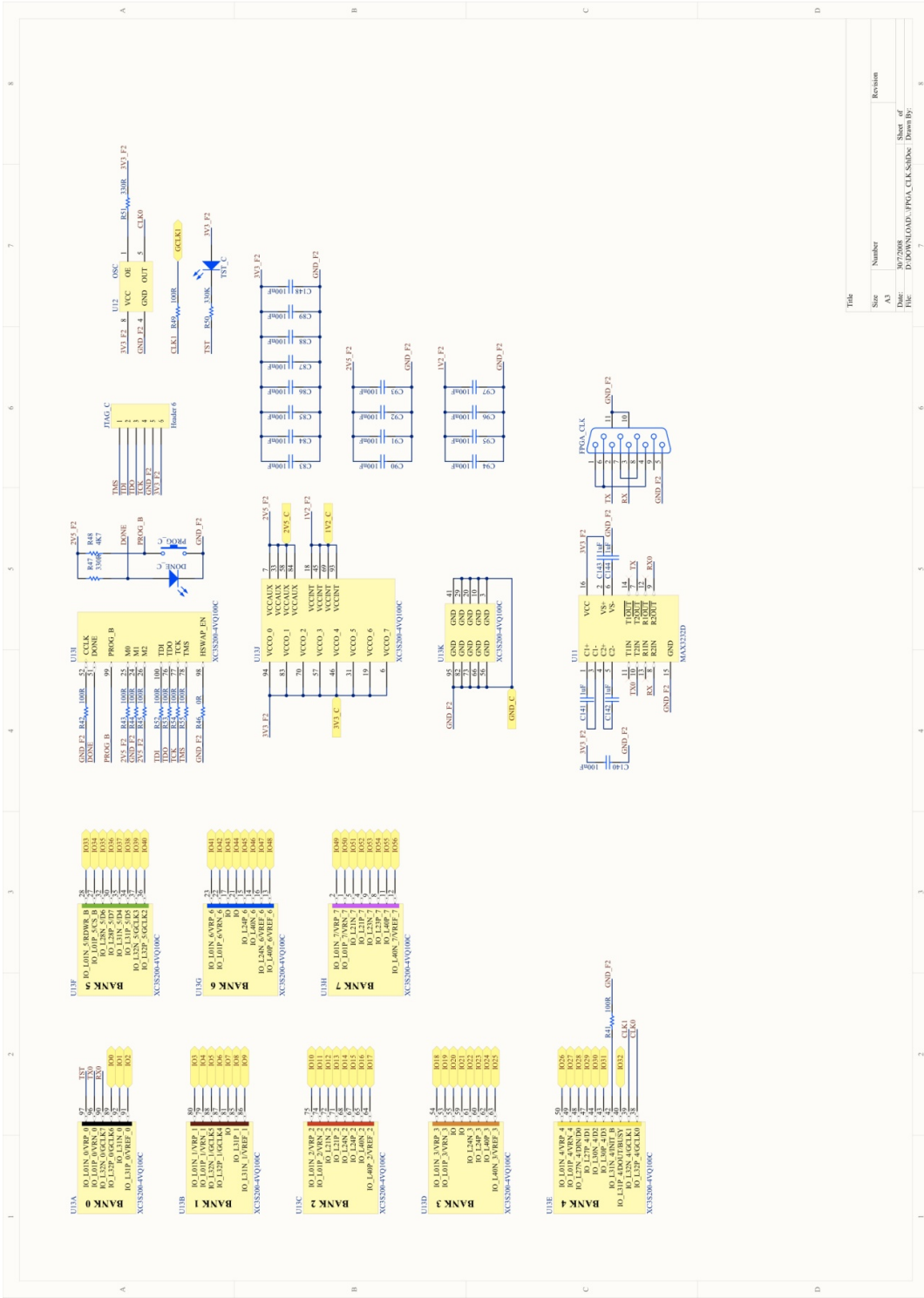
Nas páginas seguintes são apresentados os diagramas esquemáticos da plataforma de ensaios utilizados nesse trabalho.

11.1 PLATAFORMA DE ENSAIOS

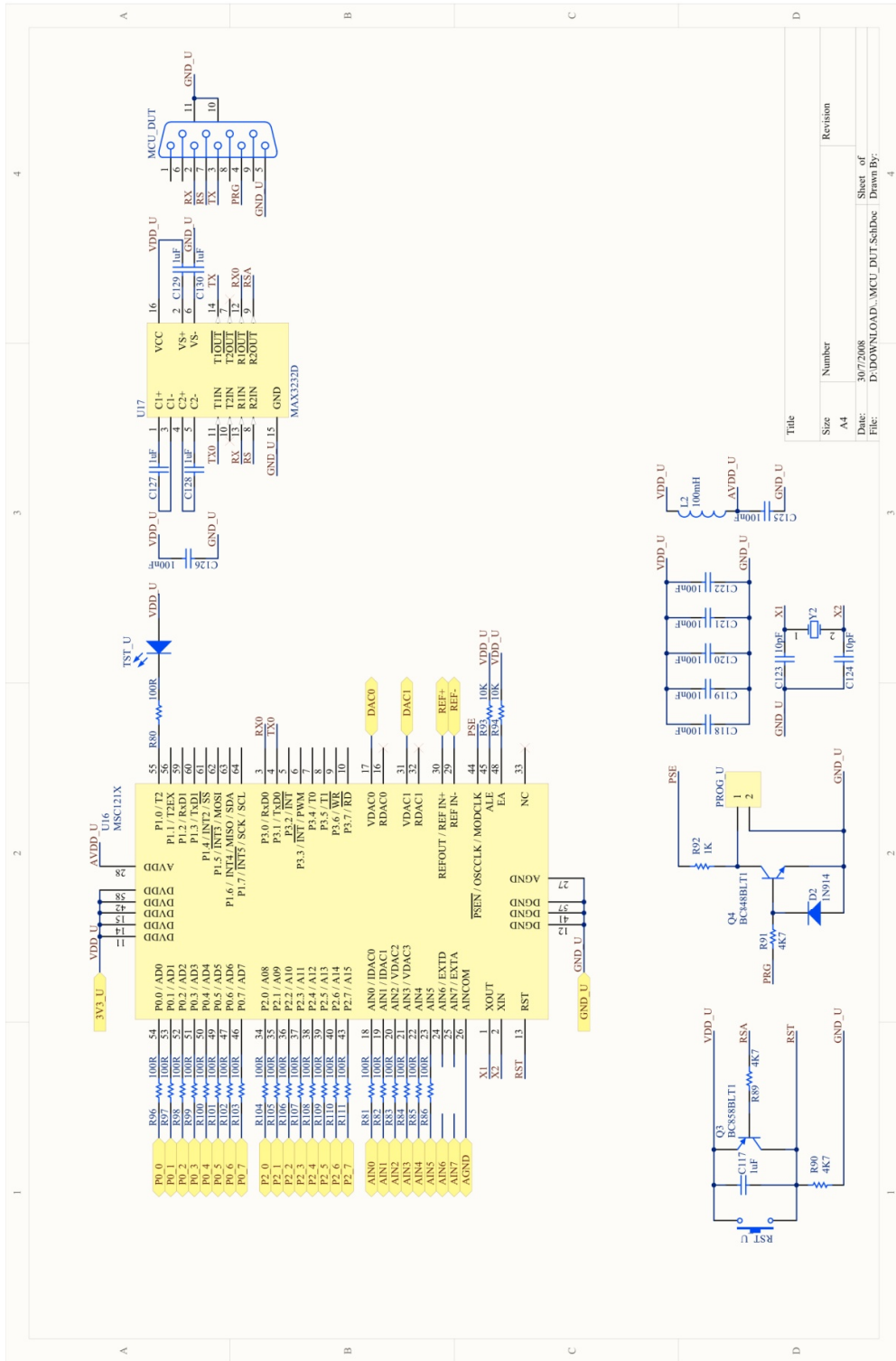






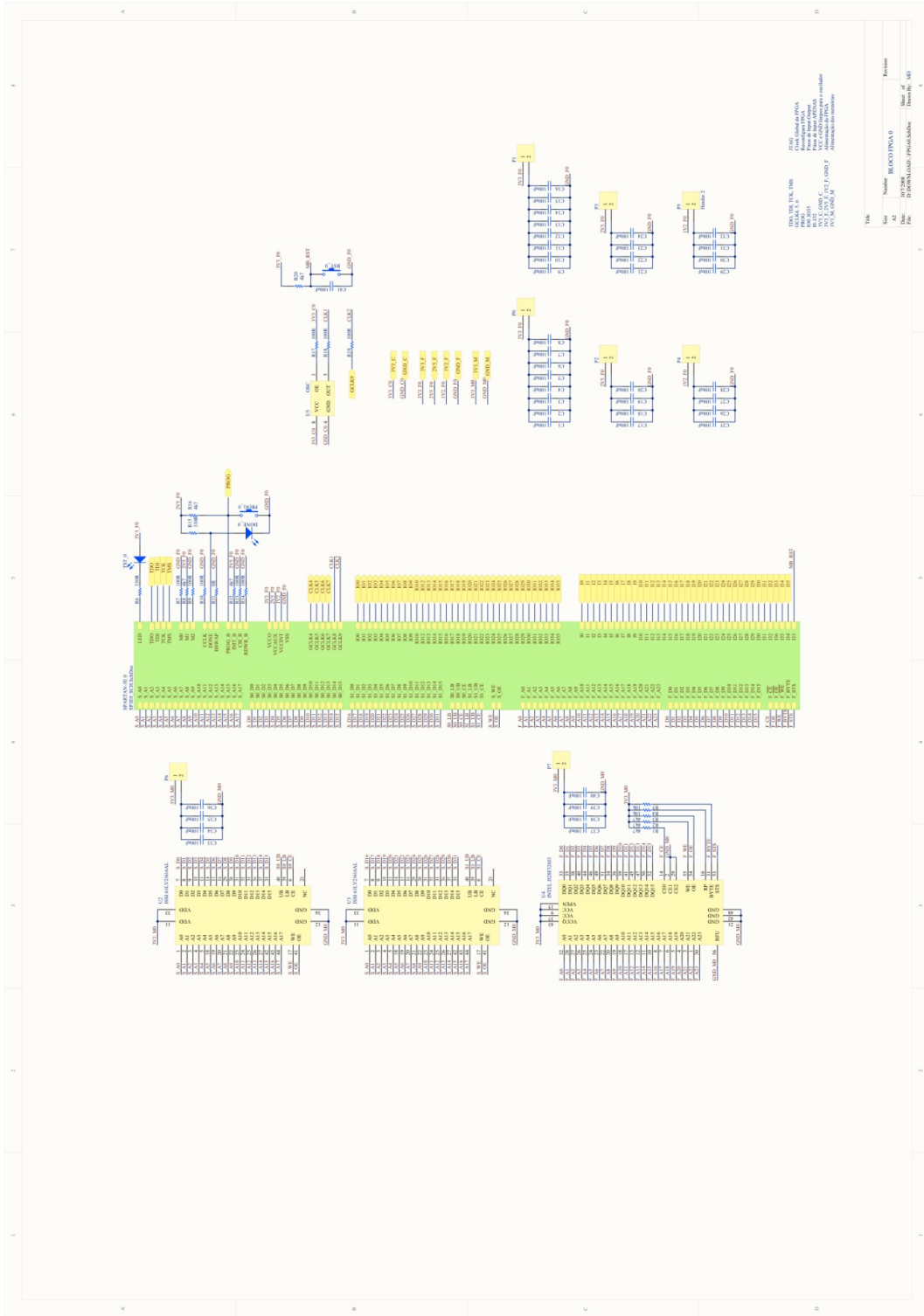


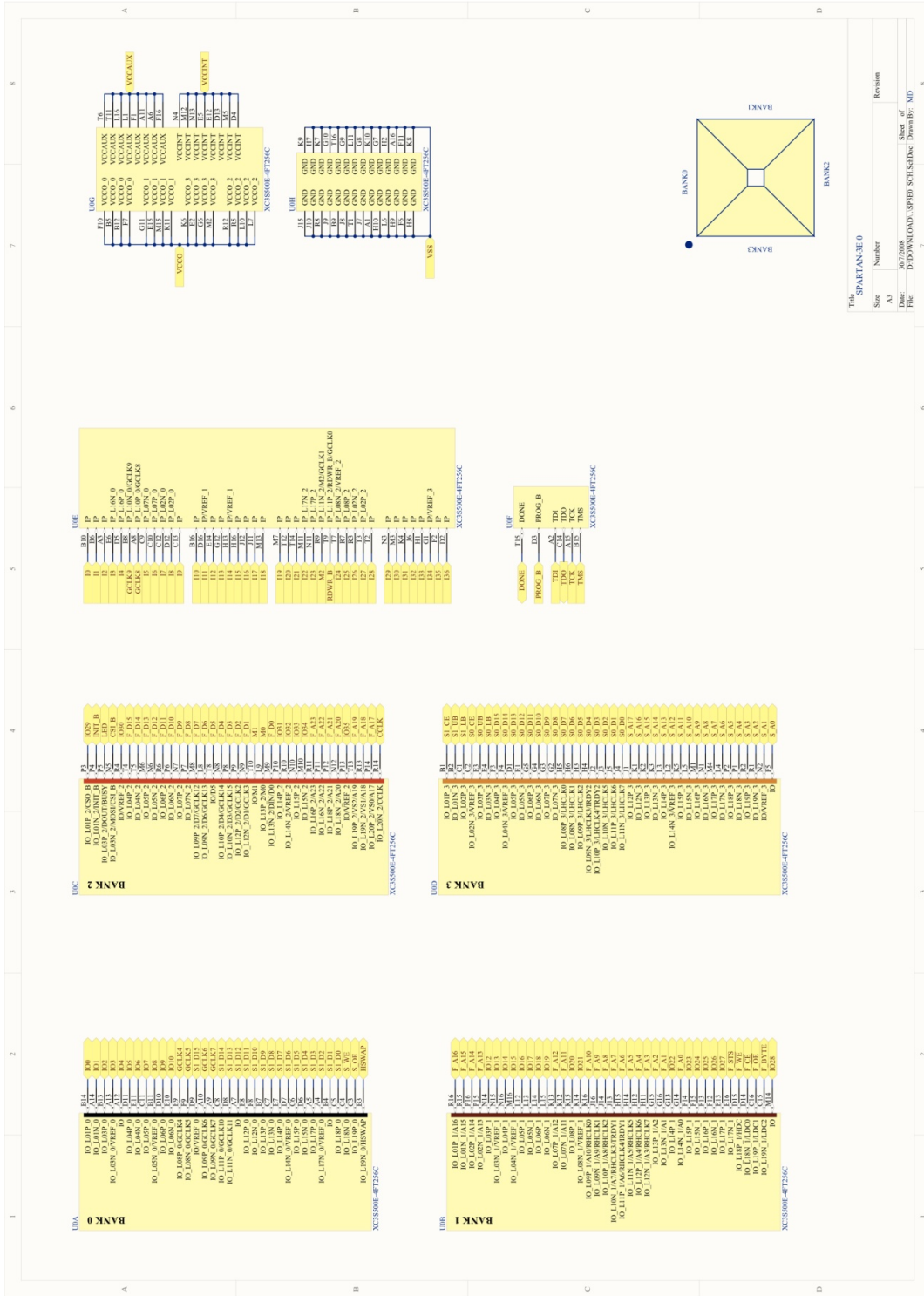
Size	Number	Revision
A1	7	1
Title: D:\DOWNLOAD\JPG\CLK_SchDoc_Dispay		
Sheet: 7		
Part: 121		



Title	Size	Number	Revision
	A4	30/07/2008	
File: D:\DOWNLOAD\MCU_DUT_SchDoc Sheet: 4 of 4			

Sheet of 4
 Drawn By: D:\DOWNLOAD\MCU_DUT_SchDoc



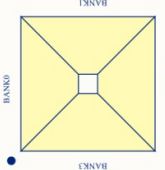


Title: SPARTAN-3E0

Size	Number	Revision
A1	1	1

Date: 08/25/2008

File: D:\DOWNLOADS\SP3E0_SCH\Sheet01.Dwg By: MD



11.2 PUBLICAÇÕES CIENTÍFICAS REFERENTES AO TRABALHO

CRISTÓFOLI, L. F.; HENGLES, A.; BENFICA, J.; BOLZANI, L.; VARGAS, F.; Atienza, A.; Silva, F. *On the Comparison of Synchronous versus Asynchronous Circuits under the Scope of Conducted Power-Supply Noise*. 2010 Asia-Pacific International Symposium on Electromagnetic Compatibility & technical Exhibition on EMC RF/Measurements and Instrumentation. April 10-12, 2010. Beijing, China. (www.apemc2010.org).

CRISTÓFOLI, L. F.; HENGLES, A.; BENFICA, J.; BOLZANI, L.; VARGAS, F.; Atienza, A.; Silva, F. *Synchronous versus Asynchronous Circuits Reliability under Radiated Eletromagnetic Disturbance*. 20th International Wroclaw Symposium on EMC. September 13-17, 2010.