



UNIVERSIDADE ESTADUAL PAULISTA

Programa de Pós-Graduação em Engenharia Elétrica

**Desenvolvimento de um Gerador de
Sinais para Estimulador Neuromuscular
Utilizando Tecnologia FPGA**

Adriano dos Santos Cardoso

1210001488



Ilha Solteira - SP



UNIVERSIDADE ESTADUAL PAULISTA
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Desenvolvimento de um Gerador de Sinais para Estimulador Neuromuscular Utilizando Tecnologia FPGA

Dissertação submetida à Faculdade de
Engenharia de Ilha Solteira - UNESP como
parte dos requisitos necessários para a
obtenção do título de Mestre em
Engenharia Elétrica

Candidato: Adriano dos Santos Cardoso

Orientador: Eng. Eletr. Alexandre César Rodrigues da Silva

1210001488



Ilha Solteira – S. P
Novembro – 2003.

30400007
sys 221288
sys 56299

Proc. 063/04 - NPD 64

UNESP - "CAMPUS DE ILHA SOLTEIRA"	
SERVIÇO TÉC. DE BIBLIOTECA E DOCUMENTAÇÃO	
DATA DE CHEGADA	DATA DE TOMBO
<i>04.05.04</i>	<i>31.05.04</i>
REGISTRADO POR	TOMBO
<i>Ailza</i>	<i>Te. 1488</i>
AQUISIÇÃO	CLASSIFICAÇÃO
<i>Ilha Solteira</i> <i>aut</i> <i>R\$ 10,00</i>	<i>C268d</i>

FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação/Serviço Técnico de Biblioteca e Documentação da FEIS/UNESP

Cardoso, Adriano dos Santos

C268d

Desenvolvimento de um gerador de sinais para estimulador neuromuscular utilizando tecnologia FPGA / Adriano dos Santos Cardoso . -- Ilha Solteira : [s.n.], 2003
136 p. : il.

Dissertação (mestrado) – Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2003

Orientador : Alexandre César Rodrigues da Silva

Bibliografia: p. 134-136

1. VHDL (Linguagem descritiva de hardware) 2. Circuitos integrados – Integração em larga escala. 3. Hardware linguagens descritivas. 4. Eletroterapia.

DESENVOLVIMENTO DE UM GERADOR DE SINAIS PARA ESTIMULADOR NEUROMUSCULAR UTILIZANDO TECNOLOGIA FPGA

Adriano dos Santos Cardoso

DISSERTAÇÃO SUBMETIDA À FACULDADE DE ENGENHARIA - CAMPUS DE
ILHA SOLTEIRA – UNESP – COMO PARTE DOS REQUISITOS NECESSÁRIOS
PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA ELÉTRICA

A. A. Carvalho

Prof. Dr. Aparecido Augusto de Carvalho
Coordenador do Programa de Pós-Graduação
em Engenharia Elétrica

COMISSÃO EXAMINADORA:

[Signature]

Prof. Dr. Alexandre César Rodrigues da Silva - orientador

[Signature]

Prof. Dr. Edward David Moreno Ordonez

A. A. Carvalho

Prof. Dr. Aparecido Augusto de Carvalho

Ilha Solteira-SP, novembro de 2003.

Agradecimentos

A minha família que nunca deixou de confiar e apoiar o meu trabalho.

Aos companheiros de laboratório Edson Baptista e Silvano Renato Rossi pelas inúmeras sugestões e também pela paciência.

À Tatiane Tobal dos Reis pela atenção dedicada.

À Clara Isabel Guimarães pela discussões e atenção despendida.

À Eliane da Silva pela atenção e auxílio nas correções de texto.

Aos funcionários do Departamento de Engenharia Elétrica pelas inúmeros auxílios e presteza nos atendimentos.

Aos Prof. Dr. Aparecido Augusto de Carvalho, Claudio Kitano, Edvaldo Assunção que trouxeram auxílio em discussões, Estudos Especiais e na elaboração dos textos.

Ao Eng. Eletr. Alexandre César Rodrigues da Silva pela atenção, paciência e comprometimento indispensáveis para a conclusão deste trabalho.

Ao CNPq pelo apoio financeiro.

A Deus.



RESUMO

“Tudo que pedires com Fé, em
oração vós o recebereis.”

MATEUS 21, 22

RESUMO

As ferramentas automáticas de síntese apresentam-se como poderosas ferramentas na geração de descrição de circuitos de forma rápida e correta. Além disso a alta abstração de suas linguagens permite que circuitos sejam gerados sem discriminar qual tecnologia alvo será aplicada. Este trabalho apresenta o desenvolvimento de ferramentas automáticas de síntese que tem como função a descrição de circuitos para a geração de formas de onda. Estas formas de onda são aplicáveis nos mais diversos sistemas, podendo ser utilizados em aparelhos neuromusculares. Para uma melhor análise dos resultados de simulação das formas de onda implementáveis foi desenvolvida a ferramenta TBL2M, que a partir do arquivo de simulação do ambiente de projeto Altera Max + Plus II gera-se um arquivo contendo vetores cuja visualização torna-se possível com o emprego de comandos de plotagem do MatLab.

Conceitos sobre eletroterapia foram estudados para o entendimento das formas de onda e dos equipamentos eletroterapeúticos disponíveis atualmente. Diversas formas de onda utilizadas na eletroterapia e alguns protocolos foram implementados utilizando as ferramentas de síntese.

Variações nos modelos de descrição das ferramentas de síntese e opções nos parâmetros de entrada permitiram a implementação de sinais em diferentes linguagens de descrição de hardware (AHDL e VHDL) e diferentes configurações de resolução e definição. Projetos utilizando essas descrições de hardware mostraram se eficientes como forma de geração de sinais no espectro de frequência de sinais de eletroterapia.

Palavras chave: Eletroterapia, FPGA, HDL e VHDL.

ABSTRACT

Automatic synthesis tools introduces as powerful tools in the generation of description of circuits in a fast and correct way. Besides the high abstraction of its languages allows that circuits are generated without to discriminate which target technology will be applied. This work introduce the development of automatic tools of synthesis that has as task the description of circuits to waveforms generation. These waveforms are applicable in the most different systems that could be used in neuromuscular devices. For a better analysis of the results of simulation of waveforms implemented, the tool TBL2M it was developed, that starting from the file of simulation of the design environment generates a file of visualization of waveform.

Concepts on electrotherapy were studied for the understanding of waveforms and equipments available in electrotherapy. A lot of waveforms used in the electrotherapy and some protocols were implemented using these circuits.

Variations in the description models of the synthesis tools and its options in the input parameters allowed the development of signs in different languages of hardware description (AHDL and VHDL) and different resolutions and definitions configurations.

Projects using those HDLs was shown an efficient form of generation of signs in the spectrum of frequency range of electrotherapy.

Key words: Electrotherapy, FPGA, HDL e VHDL.

LISTA DE FIGURAS

Figura 1: Aparelhos de Eletroterapia: (a) - Physiotonus II (analógico) e (b) - Dualpex 961 (digital).	30
Figura 2: Rampa de Subida e Rampa de Descida em Burst - (Subida lenta – Descida lenta).	32
Figura 3: Rampa de Subida e Rampa de Descida em Burst - (Subida Rápida – Descida lenta).	32
Figura 4: Rampa de Subida e Rampa de Descida em Burst - (Subida lenta – Descida Abrupta).	33
Figura 5: Rampa de Subida e Rampa de Descida em Burst - (Subida Abrupta – Descida lenta).	33
Figura 6: Eletrodos Comerciais: (a) -Oval e (b) - Retangular.	35
Figura 7: Configurações de Eletrodos Bipolares.	38
Figura 8: Configurações de Eletrodos Quadripolares.	38
Figura 9: Modulação de Corrente – Modulação em Amplitude.	42
Figura 10: Modulação de Corrente – Modulação de Duração do Pulso.	43
Figura 11: Modulação de Corrente – Modulação PPM.	43
Figura 12: Controles da Formas de Ondas por Burst.	44
Figura 13: Resultado da Ativação das Unidades Motoras por Contração Voluntária.	46
Figura 14: Eletrodos Aplicados à Pele – Corrente Circulante.	47
Figura 15: Níveis de Estimulação em Relação a intensidade corrente - Duração do Pulso.	49
Figura 16: Estimulo Aplicado (A) e o Desempenho do Músculo (B).	51
Figura 17: Resposta Muscular em Relação ao Aumento de Frequência.	52
Figura 18: Elementos primitivos de cada nível de abstração de Gajski e Kuhn (Diagrama Y).	61
Figura 19: Evolução dos Componentes de Lógica Programável.	65
Figura 20: Arquitetura de um FPGA.	66
Figura 21: Arquivo AHDL de descrição da equação seno com $n_i = 12$ bits e $n_o = 8$ bits.	84
Figura 22: Arquivo VHDL de descrição da equação seno com $n_i = 8$ bits e $n_o = 8$ bits.	85
Figura 23: Implementação de Equações usando a F2HDL.	90
Figura 24: Ambiente de Programação – Simulação do decodificador seno ($n_i = 8$ e $n_o = 8$).	91

Figura 25: Diagrama Y para as etapas de síntese do projeto.....	93
Figura 26: Esquema de Processo de Geração de Formas de Onda em FPGA.....	95
Figura 27: Geração de Sinal Analógico utilizando Decodificador.....	98
Figura 28: <i>Projeto_1</i> – Geração de Sinal Analógico Usando FPGA.	99
Figura 29: Arquitetura do Projeto Seleção de Formas de Onda.	101
Figura 30: Hierarquia do Projeto <i>Proje_2</i>	102
Figura 31: Projeto de seleção de formas de onda implementado.....	102
Figura 32: Símbolo do Projeto <i>Proje_2</i>	103
Figura 33: Projeto para Protocolo de Estimulação – Controle de Frequência.....	104
Figura 34: Visualização no MatLab da simulação da função seno ($n_i = 8$ bits, $n_o = 8$ bits).	114
Figura 35: Visualização da forma de onda seno de 8 bits em osciloscópio.	115
Figura 36: Visualização no MatLab da simulação da função gaus ($n_i = 8$ bits, $n_o = 8$ bits).	116
Figura 37: Visualização da forma de onda gaussiana de 8 bits em osciloscópio.	116
Figura 38: Visualização no MatLab da simulação da função sigmoide ($n_i = 8$ bits, $n_o = 8$ bits).	117
Figura 39: Visualização da forma de onda sigmóide de 8 bits 56Hz em osciloscópio.	118
Figura 40: Simulação no MatLab da Geração Seno em três canais.	119
Figura 41: Simulação no MatLab da forma de onda Farádica 1.....	120
Figura 42: Simulação no MatLab da forma de onda Farádica 2.....	120
Figura 43: Simulação da Onda Farádica_2 com 1280 passos.	121
Figura 44: Visualização da forma de onda Farádica 1 de 8 bits em 0.125 Hz.....	122
Figura 45 : Visualização da forma de onda Farádica de 8 bits em 2637 Hz.	122
Figura 46 : Visualização da forma de onda Farádica 1 de 8 bits em 204,6 Hz.	123
Figura 47: Simulação de Escolha de Formas de Ond	124
Figura 48: Visualização do protocolo de estimulação Farádica controlado pelo sinal enable.	125
Figura 49: Visualização do protocolo de estimulação Farádica controlado pelo sinal selecao.	125
Figura 50: Visualização da seleção de formas de onda Farádica-Sigmóide em 8 bits (56 Hz).	126
Figura 51: Visualização da seleção de formas de onda Farádica-Senóide em 8 bits (56 Hz).	126
Figura 52: Modulação PPM.....	127

Figura 53: Onda Seno Modulada em PPM..... 128

Figura 54: Onda Farádica Modulada PPM..... 128

LISTA DE TABELAS

Tabela 1: Valores de Entrada para F2HDL.....	89
Tabela 2: Variação da quantidade de células utilizadas no decodificador seno com $n_i = 8$. Família MAX 7000.....	107
Tabela 3: Variação da quantidade de células utilizadas no decodificador seno com $n_i = 8$ Família FLEX 10K20.....	107
Tabela 4: Variação da quantidade de células utilizadas no decodificador sem o com $n_o = 8$. Família FLEX 10K20.....	108
Tabela 5: Resultados de comparação de simulações entre linguagens de descrição de hardware – Família MAX 7000.....	110
Tabela 6: Resultados de comparação de simulações entre linguagens de descrição de hardware – Família FLEX 10K.....	111
Tabela 7: Comparação entre linguagens de descrição: Quantidade de Células – Família FLEX 10K.....	112

SUMÁRIO

RESUMO.....	6
ABSTRACT	7
LISTA DE FIGURAS.....	8
LISTA DE TABELAS	11
SUMÁRIO.....	12
DEFINIÇÃO DE TERMOS.....	17
ELETROTERAPIA.....	20
1.1 HISTÓRICO.....	20
INSTRUMENTAÇÃO PARA A ELETROTERAPIA	27
2.1 CARACTERÍSTICAS DE PROJETO DE ESTIMULADORES PARA APLICAÇÕES ELETROTERAPÊUTICAS.....	27
2.2 PROJETO DE ESTIMULADORES DE CONTROLE ANALÓGICO VERSUS DE CONTROLE DIGITAL	28
2.3 SELEÇÃO DAS FORMAS DE ONDA.....	30
2.4 CONTROLES DE RAMPA DE SUBIDA E RAMPA DE DESCIDA	31
2.5 SISTEMAS DE ELETRODOS PARA ELETROTERAPIA.....	31
2.5.1 Tipos de Eletrodos de Superfície.....	34
2.6 ESPECIFICAÇÃO DO TAMANHO E DA FORMA DOS ELETRODOS.....	35
2.7 DESIGNAÇÃO DA LOCALIZAÇÃO DO ELETRODO NAS APLICAÇÕES TERAPÊUTICAS.....	36
2.8 TIPOS DE ESTIMULADORES.....	39
2.8.1 Estimuladores de Corrente versus Tensão	39
2.9 CLASSIFICAÇÃO DAS CORRENTES EM ELETROTERAPIA.....	40

2.9.1 Designações e nomenclaturas das Correntes.....	40
2.9.2 Tipos de correntes Eletroterapêuticas.....	40
2.9.3 Corrente contínua.....	41
2.9.4 Corrente pulsada.....	41
2.9.5 Equilíbrio de carga em formas de onda bifásicas.....	42
2.10 MODULAÇÕES DE CORRENTE.....	42
2.10.1 Corrente Interferencial.....	43
2.10.2 Trens de Pulsos, Burst.....	44
FISIOLOGIA DO NERVO E DO MÚSCULO.....	45
3.1 ATIVAÇÃO DE TECIDOS EXCITÁVEIS COM ESTIMULAÇÃO ELÉTRICA.....	47
3.2 RESPOSTA CLÍNICAS PARA A ESTIMULAÇÃO NERVOSA E MUSCULAR.....	48
3.2.1 Estimulação em Nível Sensitivo.....	48
3.2.2 Estimulação de Nível Motor.....	49
3.3. RESPOSTAS MUSCULARES SEGUNDO A INTENSIDADE DO PULSO.....	50
3.3.1 Em nível de fibra muscular: lei do tudo ou nada.....	50
3.4. RESPOSTAS MUSCULARES SEGUNDO A FREQUÊNCIA DO PULSO.....	51
3.4.1. Tetanização.....	51
3.5. ESTIMULAÇÃO ELÉTRICA DO MÚSCULO: TÉCNICAS E APLICAÇÕES.....	52
3.5.1. Plasticidade muscular em resposta à estimulação elétrica.....	53
3.5.2 Adaptação Para a Atividade Prolongada com Nível de Força Baixo.....	53
3.5.3 Adaptação Para a Atividade Prolongada com Nível de Força Alto.....	53
3.6. SELEÇÃO DE FORMA DE ONDA DA CORRENTE EM EENM.....	54
3.6.1. Preferência do Indivíduo e Conforto.....	54
3.7. ELETROESTIMULAÇÃO INTEGRADA.....	55
3.8. RISCOS E CONTRA-INDICAÇÕES.....	55
SISTEMAS DIGITAIS.....	57
4.1 O PROCESSO DE PROJETO DE SISTEMAS DIGITAIS.....	58



4.1.1 Um modelo para o processo de projeto – O diagrama Y.....	59
4.1.2 Elementos de Sistemas Digitais.....	62
4.1.2.1. Uso de Lógica Programável	63
4.1.2.2 Características no uso de FPGA.....	66
4.1.3 Programação em FPGA, Ambiente e Ferramentas de Projeto	69
4.1.3.1 Editor Gráfico	69
4.1.3.2 Editor de Texto.....	70
4.1.3.3 Compilador	70
4.1.3.4 Simulador.....	71
4.1.3.5 Editor de Planta Baixa.....	72
4.1.3.6 Programador (dos componentes FPGAs)	72
4.2 LINGUAGENS DE DESCRIÇÃO DE HARDWARE	72
4.2.1 Aplicação.....	72
4.2.2 Descrição	73
4.2.3 Portabilidade:	74
4.3 EXEMPLOS DE LINGUAGENS DE DESCRIÇÃO DE HARDWARE.....	75
4.3.1 A Linguagem AHDL (Altera Hardware Description Language).....	75
4.3.2 A Linguagem VHDL (Very High Speed Integrated Circuit Hardware Description Language)	76
DESENVOLVIMENTO DE FERRAMENTAS DE SÍNTESE AUTOMÁTICA.....	78
5.1 OBJETIVOS	79
5.2 PROGRAMAÇÃO EM LINGUAGEM C.....	80
5.3 FERRAMENTAS DE SÍNTESE	80
5.3.1 Ferramenta E2HDL	81
5.3.1.1 Desenvolvimento da Ferramenta de Síntese.....	81
5.3.1.2 Exemplo 1 : Função Seno implementada em AHDL	83
5.3.1.3 Exemplo 2: Função Seno implementada em VHDL	84
5.3.2 Geração de Seno em Três Canais.....	85
5.3.3 Ferramenta - Análise em Série de Fourier.....	86

5.3.3.1 Engenharia Reversa.....	86
5.3.3.2 Ferramenta F2HDL - Da Análise em Série de Fourier à HDL.....	87
5.3.3.2.1 Desenvolvimento da Ferramenta de Síntese	87
5.3.3.2.2 Exemplo 3: Emprego da F2HDL	89
5.4 FERRAMENTA TBL2M.....	90
5.5 E2HDL E F2HDL NO DIAGRAMA DE GAJSKI	92
5.6 ETAPAS DE VALIDAÇÃO DOS DECODIFICADORES-FUNÇÃO	93
PROJETOS DE SISTEMAS DIGITAIS PARA ESTIMULADORES NEUROMUSCULARES.....	96
6.1 IMPLEMENTAÇÃO DE FORMAS DE ONDAS ANALÓGICAS UTILIZANDO RECURSOS DIGITAIS	96
6.1.1 O Contador.....	97
6.1.2 Decodificador-Função	97
6.1.3 O conversor D/A.....	98
6.2 SELEÇÃO DE FORMAS DE ONDA.....	99
6.2.1 Objetivos do Projeto seletor de formas de onda	100
6.2.2 Desenvolvimento do Projeto.....	100
6.3 PROTOCOLO DE ESTIMULAÇÃO – CONTROLE DE DURAÇÃO DOS PULSOS.....	103
6.3.1 Objetivos do Projeto.....	103
6.3.2 Desenvolvimento do Projeto.....	104
RESULTADOS	106
7.1 RESULTADOS DE COMPARAÇÕES DE PARÂMETROS DOS DECODIFICADORES, FAMÍLIAS DE COMPONENTES E LINGUAGENS DE DESCRIÇÃO DE CIRCUITOS.....	106
7.1.1 Comparações de parâmetros de decodificadores e famílias de componentes.....	106
7.1.2 Comparações entre Linguagens de Descrição de Hardware	109
7.2 VISUALIZAÇÕES E IMPLEMENTAÇÕES DAS FORMAS DE ONDA GERADAS A PARTIR DOS DECODIFICADORES.....	112
7.2.1 Função Seno.....	113
7.2.2 Função Gaussiana.....	115

7.2.3 Função Sigmóide.....	117
7.2.4 Função Seno: Três canais.....	118
7.2.5 Função Farádica.....	119
7.2.6 Implementação de Seleção de Formas de ondas.....	123
7.2.7 simulações de Controle de Largura de Pulso.....	127
DISCUSSÕES E CONCLUSÕES.....	130
TRABALHOS FUTUROS	133
REFERÊNCIAS.....	134

Apêndice A – Montagem do Gerador de Sinais

DEFINIÇÃO DE TERMOS

AHDL (*Altera Hardware Description Language*): Linguagem de Descrição de Hardware da Altera, linguagem desenvolvida pela empresa Altera e é base no desenvolvimento das megafunções. Apresenta grande versatilidade e grande número de comandos.

Ambiente de Programação (Ambiente de Projeto): Conjunto de programas necessários para especificação, simulação e compilação dos circuitos digitais. Nele, são também realizadas as verificações de sintaxe e síntese das HDLs bem como sua configuração para gravação.

ASIC (*Application Specific Integrated Circuit*): Circuito Integrado de aplicação específica. Nome que se dá para os circuitos integrados com grande integração e aplicações bastante específicas. Este nome engloba várias tecnologias como: FPGA, CMOS, BiCMOS, EPROM, EEPROM entre outras.

Célula Lógica (Macro célula): Termo genérico dos blocos de construção dos componentes FPGAs.

Compilação: Fase de projeto que analisa os possíveis erros, extrai as informações necessárias para a criação de arquivo necessário para uma minimização e otimização dos circuitos do projeto.

CPLD (*Complex Programmable Logic Device*): Tecnologia de componentes lógicos programáveis (ver FPGA).

Ferramenta de Síntese: As ferramentas de síntese são programas aplicativos que atuam sobre uma especificação (por exemplo HDL ou esquemático) adicionando ou alterando algum tipo de

especificação. Um exemplo para tal é a minimização de funções booleanas que tem por finalidade a redução do tamanho ou custo do circuito a ser implementado.

FES (*Funcitonal Electrical Stimulation*): Estimulação Elétrica funcional – A técnica fisioterápica que utiliza eletroestimuladores e eletrodos sobre a pele no tratamento e estimulação de músculos e nervos.

FPGA (*Field Programmable Gate Array*): Tecnologia de componentes que tem a propriedade de configuração (e reconfiguração) da sua lógica interna pela sua arquitetura em matriz de elementos lógicos, e para isso, possui também blocos interconectores que fazem a ligação entre os blocos lógicos, configurando assim o circuito. A configuração é feita por meio de circuitos programadores.

Megafunção: Um sistema complexo e de alto nível de descrição de hardware que geralmente possui parâmetros de configuração. O ambiente gráfico, com a representação em forma de blocos funcionais, permite a integração de uma ou mais megafunções com outros projetos.

Simulação: Fase de projeto que, através de modelos, faz a análise lógica e a temporização das variáveis do projeto.

Validação: Parte do projeto que estabelece os acertos, correções e mudança de parâmetros necessários para um bom e adequado funcionamento dos circuitos.

VLSI (*Very large Scale Itegrated*): Nome dado a várias famílias de circuitos integrados que possuem uma capacidade de integração tal que, em um único circuito integrado possa conter pelo menos 1000000 de componentes.

VHDL (*Very High Speed Integrated Circuit Hardware Description Language*): Linguagem de Descrição de Hardware que foi adotado como padrão IEEE. Permite a descrição de um circuito nas formas Comportamental e RTL (funcional) e estrutural.

Capítulo 1

ELETROTERAPIA

A eletroterapia é atualmente apresentada como uma entre as alternativas terapêuticas. Atuando como fonte de estímulos musculares para o indivíduo e apresentando eficiente alternativa para tratamento de dores musculares, fortalecimento muscular, condicionamento pós-traumático e lesões musculares. Muitos trabalhos sobre eletroterapia e suas implicações foram desenvolvidos de forma a melhorar o conforto do indivíduo e equipamentos de diferentes tecnologias foram empregados de forma a tornar o processo de estimulação mais adequado em vários aspectos.

1.1 HISTÓRICO

O uso da eletricidade como recurso terapêutico em medicina é antigo. Os romanos usavam as descargas elétricas do peixe torpedo para o tratamento da gota e o alívio de dores de cabeça [1]. Em 1786, Luigi Galvani fez experiências com pernas de rã. Unindo-se dois metais diferentes de um lado e tocando com as outras extremidades em certos pontos do corpo da rã, as pernas agitavam-se bruscamente como se o animal estivesse vivo [2].



Em 1867, Duchenne de Boulogne estudou a fisiologia dos movimentos com a ajuda da faradizaçãoⁱ, introduzindo o uso de corrente induzida no diagnóstico e na terapêutica dos músculos [3].

Como o passar do tempo, inúmeros aparelhos foram produzidos visando aplicações médicas. Em 1962, Kantrowitz [1] implantou eletrodos em um paciente paraplégico e conseguiu condições de ortostatismoⁱⁱ.

Liberson [1] idealizou um gerador de pulsos elétricos cujos eletrodos estimulavam o nervo pioneiro, promovendo a dorsiflexão do pé durante a fase da passagem, permitindo desta forma uma marcha mais eficiente. Esta idéia abriu um novo campo na medicina de reabilitação e foi denominada por Vadovnik como *Functional Electrical Stimulation* (FES), ou Estimulação Elétrica Funcional (EEF), que assim a descreveu: “O objetivo da EEF é promover contração em músculos privados de controle nervoso, tendo como resultado um movimento funcional” [1].

A eletroterapia usada no controle da dor foi introduzida a partir da “teoria da comporta”, proposta por Melzack e Wall em 1965 [1], em que foram estudados mecanismos inibidores e acionadores das células neuro sensitivas.

Muitos estudos sobre EEF estão direcionados para o restabelecimento e cura da função motora de membros prejudicados. A demanda pelo uso clínico de estímulos elétricos funcionais para membros tem aumentado gradualmente. Muitos equipamentos de EEF satisfazem as necessidades do paciente, dos familiares e pesquisas nas áreas médicas [4]. Komiyama et al. [5] desenvolveram um sistema para EEF híbridoⁱⁱⁱ com eletrodos de superfícies em combinação com

ⁱ Tipo de corrente obtido pela indução eletromagnética.

ⁱⁱ Ficar em pé, ereto.

ⁱⁱⁱ Conjunto que reúne andador e ortose para membros inferiores.

um sistema denominado HKAFO (Hipknee-Ankle-Foot-Orthotic) para o controle dos passos de pacientes paraplégicos, analisando o posicionamento e tamanho dos eletrodos para um sistema eficiente.

Kagaya et al. [6] desenvolveram um sistema para EEF híbrido com eletrodos. O sistema consiste em um estimulador com 18 canais de saídas para um melhor controle da fase de marcha.

Huifang-Dou et al. [7] desenvolveram um sistema para EEF flexível e com múltiplos canais que foi utilizado para estímulo neuromuscular funcional na restauração do movimento de membros superiores e inferiores de pacientes paralisados. Esse equipamento gera diferentes pulsos de correntes em 16 canais independentes. É rotineiramente utilizado para produzir pulsos retangulares - formas de pulsos bifásicos^{iv} tendo duração do pulso na faixa de 5 ms a 1000 ms com resolução de 1 ms. A amplitude pode ser modulada de -100 a +100 mA com resolução máxima de 0,4 mA. Resultados experimentais mostraram que o estimulador teve um bom desempenho.

Gudnason G. et al. [8] descrevem um CI (circuito integrado) de um estimulador neural com vários canais para ser utilizado em EEF. O CI descrito executa todo o processamento de sinais necessário na implementação de um estimulador neural. Podem ser gerados pulsos de correntes com cargas balanceadas e possui 4 canais de saídas de forma que pode ser empregado em eletrodo com várias conexões para o nervos. O propósito desse estimulador elétrico funcional é a restauração de várias funções motoras do corpo em pacientes que tenham perdido seu movimento devido a lesões ou enfermidades motoras.

^{iv} Nomenclatura utilizada na eletroterapia para designar corrente alternada.

Feng W. et al. [9] trabalharam com máquinas de estados finitos para flexibilizar o projeto de um controlador de estímulos para equipamento EEF. Tal iniciativa possibilitou a implementação de diferentes estágios de controles sob diferentes estados. O controlador desenvolvido é baseado em um microcontrolador 68332 da Motorola. Os estados, as ações e as regras para a transição entre estados são definidos em um arquivo texto que é editado pelo usuário e carregado em um estimulador para EEF portátil, através da porta serial RS-232 de um PC.

Em 1991, K. Kaczmarek propôs um estimulador de 16 canais microcontrolado. O sistema consiste em gerar rajadas de trens de pulsos bifásicos de 1 a 100, frequência de 0.1 a 25 KHz, duração de 2 a 1000 μ s e corrente de 0 a 50 mA [10]. O estimulador apresenta uma grande flexibilidade nos parâmetros da forma de onda gerando pulsos retangulares simétricos (bifásicos) que podem provocar desconforto ao indivíduo.

O projeto de um estimulador de baixo custo apresentado por De Lima J.[11] apresenta um gerador de correntes com as mesmas características apresentadas por Kaczmarek e com circuitos desenvolvidos a partir de um sinal de alta voltagem (± 100 V) e com um conversor tensão-corrente.

A seletividade é um parâmetro muito importante na EEF.

A seletividade no diâmetro das fibra refere-se a habilidade de ativar um grupo de nervos contendo um diâmetro comum sem ativar fibra de nervos com diâmetros diferentes. A seletividade espacial refere-se a habilidade em ativar fibras de nervos em uma região localizada sem ativar as fibras de nervos na região vizinha.

A função do estímulo aplicado é ou a despolarização da membrana do nervo ou a ação de contração da fibra muscular. Devido ao aspecto de ação pontual, deve-se escolher um

estimulador que afete seletiva e quantitativamente as áreas idealizadas. Estudos apontaram diferentes formas de se abordarem os problemas de seletividade. A escolha da forma de onda tem um papel fundamental neste quesito.

Um grande número de diferentes formas de ondas tem sido empregado em diferentes aplicações. No trabalho de Warren M. Grill e J. Thomas Mortimer [12] é feita uma revisão sobre a geração de formas de ondas para estimuladores elétricos em sistemas nervosos. Foram considerados dois tipos de seletividade – amplitude e frequência.

Outro importante aspecto da estimulação elétrica é a tolerância do indivíduo ao procedimento, o que pode afetar o potencial de resultado do tratamento. O limite de tolerância à estimulação pode ocasionar procedimentos ineficazes de estimulação elétrica. Delitto A; Rose S.J, [13] avaliaram o conforto do indivíduo em resposta à estimulação elétrica do grupo muscular quadríceps em indivíduos normais, não encontrando nenhuma forma de onda que pudesse ser considerada a mais confortável na evocação de níveis comparáveis de contração, ou seja, o estimulador deve ser tal que se possa modificá-lo a forma de onda de acordo com o mais confortável para cada indivíduo.

Um estudo de Bowman e Baker [14] sugere que a maioria dos indivíduos preferem a forma de onda bifásica simétrica ao invés da onda bifásica assimétrica, devido a uma melhora no conforto durante a estimulação.

Estudos de Crépon sugerem que o estimulador para a aplicação da estimulação elétrica deve ser projetado para permitir considerável flexibilidade no ajuste dos parâmetros da corrente [3].

De modo geral, os trabalhos sugerem que para um estimulador elétrico adequado exista uma flexibilização nos parâmetros do estimulador, número de canais de acordo com os grupos musculares excitáveis e também uma escolha de uma forma de onda adequada e confortável.

Apresenta-se no capítulo 2 a evolução dos equipamentos eletroterapêuticos empregados comercialmente e suas características.

No capítulo 3 é mostrado a fisiologia do nervo e do músculo para facilitar o entendimento do funcionamento dos mecanismos da estimulação elétrica.

O capítulo 4 traz um breve histórico da evolução dos componentes digitais e características das ferramentas de projeto atualmente disponíveis para desenvolvimento de circuitos usando componentes lógicos programáveis.

Algumas ferramentas de síntese foram especialmente desenvolvidas para gerar descrições de hardware denominadas E2HDL [15], [16] e [17] e F2HDL [18], gerando diversos tipos de descrição de hardware denominados decodificadores-função.

No capítulo 5 é apresentado a descrição e o desenvolvimento de ferramentas de síntese automática necessárias para a implementação, em hardware, de equações algébricas para o desenvolvimento de um gerador de formas de ondas. Apresenta-se também a ferramenta TBL2M [16], que auxilia no entendimento e visualização dos arquivos de simulação gerados a partir das descrições de hardware, tendo como arquivo de entrada um arquivo de simulação de forma de onda e gerando como resultado um vetor que pode ser plotado no software MatLab.

O capítulo 6 mostra o desenvolvimento de projetos de sistemas digitais para estimuladores neuromusculares. Formas de onda foram geradas usando somente recursos digitais. As formas de onda desenvolvidas, usando os decodificadores-função apresentados no capítulo 5. Os projetos

permitem que uma dentre diversas formas de onda seja selecionada. A seleção de forma de onda fornece a escolha de um estímulo mais adequado a um paciente em diversas aplicações.

Para avaliar as ferramentas e seus resultados foram feitas simulações dos arquivos de descrição de hardware gerados e usando a ferramenta TBL2M, gerados os gráficos das descrições correspondentes. As formas de onda senoidal, sigmóide, gaussiana e farádica foram implementadas fisicamente usando FPGAs e um conversor de 8 bits. O capítulo 7 mostra esses resultados.

As descrições de hardware foram desenvolvidas partindo de duas linguagens AHDL e VHDL. Permitindo assim, que estas linguagens fossem comparadas quanto a alguns critérios de desempenho tais como quantidade de células alocadas para cada projeto, tempo de compilação e memória alocada durante a etapa de compilação.



Capítulo 2

INSTRUMENTAÇÃO PARA A ELETROTERAPIA

Uma grande variedade de aparelhos está disponível comercialmente para aplicações fisioterapêuticas e eletroterapêuticas. A instrumentação nestas áreas está se proliferando rapidamente devido aos avanços da engenharia, aos desenvolvimentos na pesquisa de eletroterapia e a um alcance ampliado dos problemas tratados pela eletroterapia. Um entendimento claro dos componentes e dos princípios de operação dos aparelhos de estimulação elétrica é essencial para aplicações clínicas eficazes e seguras.

2.1 CARACTERÍSTICAS DE PROJETO DE ESTIMULADORES PARA APLICAÇÕES ELETROTERAPÊUTICAS

Os aparelhos eletroterapêuticos são usados para induzir correntes elétricas nos tecidos corporais. Quando a eletroterapia for determinada para ser uma intervenção terapêutica apropriada, a primeira questão enfrentada pelo terapeuta é a escolha do estimulador que produzirá

o melhor resultado [19]. A escolha de um estimulador para uma aplicação específica é determinada por vários fatores relacionados com o seu respectivo projeto que devem ser observados como segue:

O estimulador deve produzir os parâmetros de estimulação elétrica requeridos para realizar o efeito desejado. A forma de onda, o valor da frequência de saída, a temporização da corrente no canal de saída etc. devem ser suficientes e adequados para a aplicação especificada;

O estimulador deve ter amplitude de saída máxima suficiente para atingir o efeito desejado;

O estimulador precisa ter um número suficiente de canais de saída para a aplicação;

As características de segurança devem estar incluídas no projeto do estimulador.

Existem também características de projeto que devem ser escolhidas de acordo com os protocolos de estimulação e portabilidade do aparelho:

O estimulador deve possuir controles apropriados para fazer quaisquer ajustes necessários dos parâmetros de estimulação durante o tratamento;

Os parâmetros de estimulação podem ser ajustados enquanto a estimulação estiver sendo fornecida;

O estimulador deve possuir bateria elétrica para não necessitar conexão à rede elétrica.

2.2 PROJETO DE ESTIMULADORES DE CONTROLE ANALÓGICO VERSUS DE CONTROLE DIGITAL

Estimuladores de controle analógico são aqueles que permitem o ajuste dos parâmetros de estimulação principalmente através do uso de botões rotatórios ou mostradores. Já que esses controles de estimuladores são de natureza mecânica (por exemplo potenciômetros), os

estimuladores de controle analógico também são chamados de “controlados por hardware”. Os estimuladores de controle digital, por outro lado, são aqueles que permitem ajuste através do uso de botões ou de chaves seletoras. Em geral, os aparelhos com projeto do controle digital incluem apenas alguma forma de display visual (por exemplo um LCD), que permite que o usuário selecione a função a partir de um menu de parâmetros de estimulação disponíveis. Os estimuladores com projeto digital atuais geralmente são baseados em microprocessadores e também são chamados “controlados por software”.

A vantagem principal dos controles analógicos é a capacidade de ajustar continuamente um parâmetro de tratamento, permitindo dessa forma que o usuário instruído faça ajustes finos nas características de estimulação enquanto o tratamento está ocorrendo. Em contraste, os controladores digitais muitas vezes só permitem ajustes incrementais dos parâmetros. Essa desvantagem pode ser minimizada se o tamanho dos aumentos de um ajuste para o próximo for o menor possível. Por exemplo, o ajuste de frequência em incrementos de 1 pps (pulso por segundo) ao contrário de incrementos de 10 pps, aumentaria a capacidade do usuário em controlar precisamente o padrão de estimulação para uma variedade de aplicações. Além disso, a modificação das características de estimulação com aparelhos digitais muitas vezes requer que o usuário percorra as opções do menu. Essa abordagem de controle é consumidora de tempo se as mudanças são desejadas durante o tratamento. A Figura 1, mostra aparelhos comerciais de eletroterapia com controle analógico e digital.

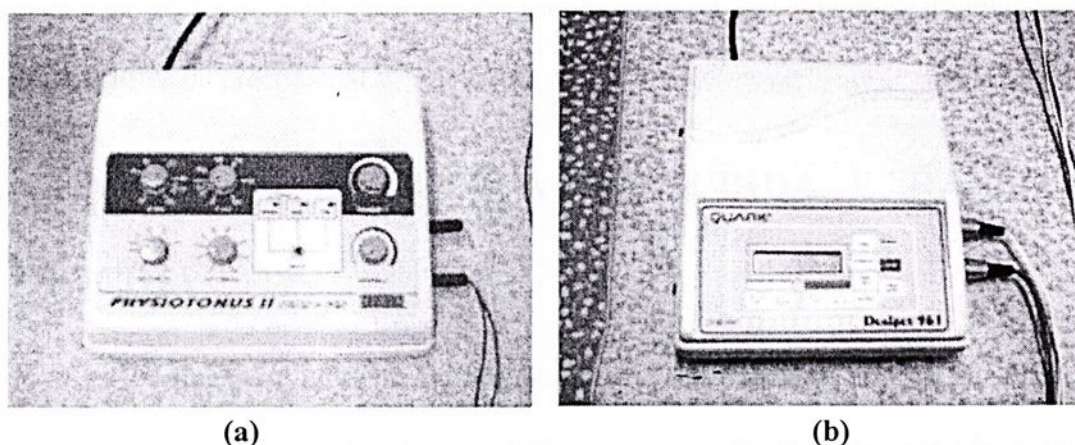


Figura 1: Aparelhos de Eletroterapia: (a) - Physiotonus II (análogo) e (b) - Dualpex 961 (digital).

2.3 SELEÇÃO DAS FORMAS DE ONDA

A proliferação de aparelhos de estimulação disponíveis comercialmente tem sido acompanhada pelo desenvolvimento de vários tipos de formas de onda de corrente (ou tensão) que podem ser usadas na eletroterapia clínica. Muitos aparelhos projetados antes do início dos anos 80 possuíam geradores de forma de onda capazes de produzir somente um tipo de forma de onda a partir de um único aparelho. Hoje, muitos estimuladores disponíveis comercialmente são capazes de produzir várias formas de onda distintas. Um dos primeiros passos para aplicar a estimulação clinicamente, é a seleção de uma forma de onda adequada. Isso é conseguido em alguns estimuladores por um controle genericamente chamado de seletor de forma de onda.

Os seletores de forma de onda podem ser rotatórios, sensíveis à pressão ou outros tipos de controles rotulados com pequenos diagramas das formas de onda disponíveis.

Em alguns estimuladores baseados em microprocessadores, as formas de onda podem ser selecionadas a partir de um menu na tela do display. A escolha da forma de onda mais

apropriada para um determinado resultado deve ser baseada em descobertas ou recomendações da literatura científica ou profissional.

2.4 CONTROLES DE RAMPA DE SUBIDA E RAMPA DE DESCIDA

A fim de aumentar ou diminuir automaticamente a amplitude de corrente da aplicação do sinal de estimulação, os controles de modulação de rampa são incluídos em vários tipos de estimuladores disponíveis comercialmente. Esses controles permitem ao fisioterapeuta ajustar o tempo de subida, de permanência e de descida do sinal que, geralmente, é um trem de pulso (Burst). Esse tipo de ajuste é necessário para controle de marcha, a medida que com o aumento da amplitude do sinal e com o devido controle entre os canais, pode-se gerar uma seqüência de estimulação para vários grupos musculares envolvidos.

As Figuras 2, 3, 4, e 5, apresentam diferentes ajustes permitidos e as respectivas formas de ondas selecionadas. Os diferentes ajustes permitem a evocação de diferentes níveis de força para diferentes grupos musculares em uma seqüência adequada para o treino de marcha em um indivíduo.

2.5 SISTEMAS DE ELETRODOS PARA ELETROTHERAPIA

Um eletrodo é um material condutor que serve como interface entre um estimulador e os tecidos do paciente. Os eletrodos são conectados aos estimuladores por meio de fios condutores. Em muitas aplicações, os eletrodos são presos à pele (eletrodos de superfície). Em outras aplicações, eles foram designados para serem implantados perto dos tecidos, tais como o nervo periférico ou osso (eletrodos invasivos), ou em cavidades corporais (eletrodos internos).



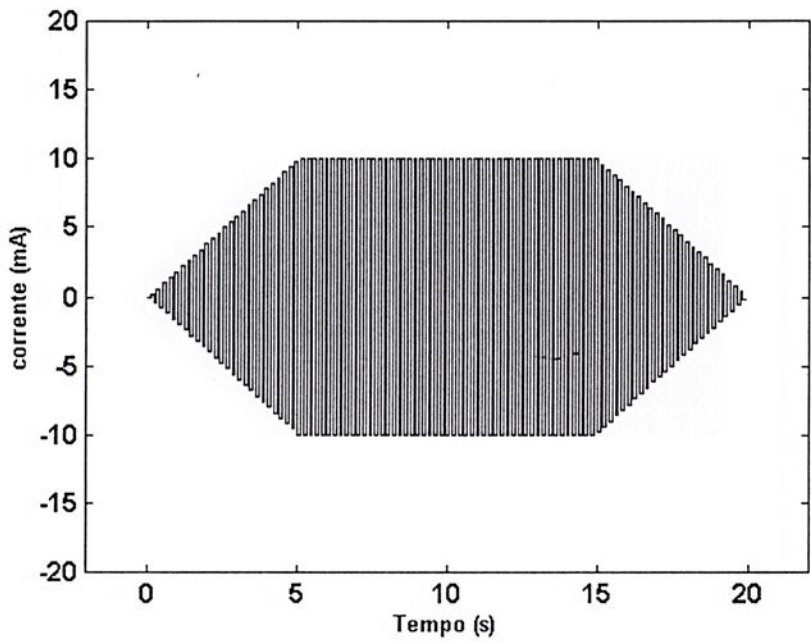


Figura 2: Rampa de Subida e Rampa de Descida em Burst - (Subida lenta – Descida lenta).

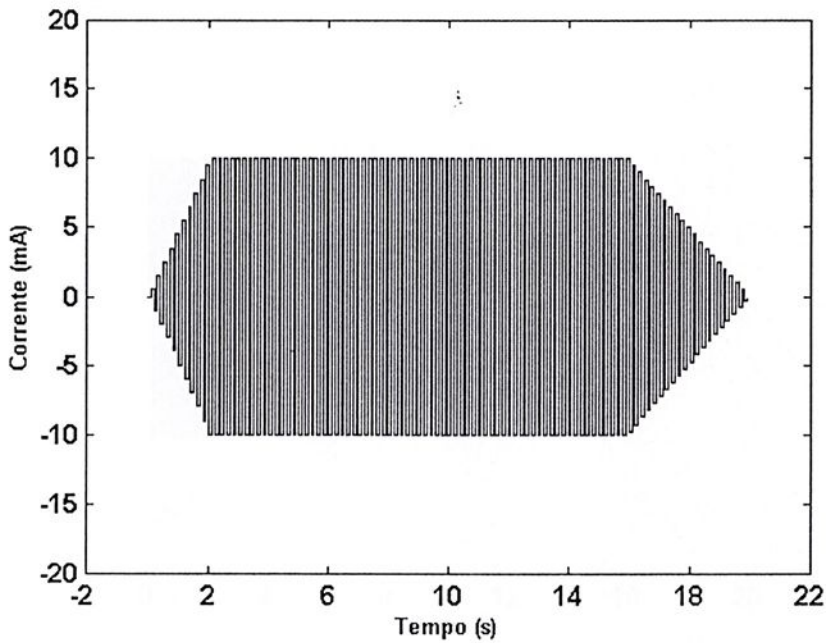


Figura 3: Rampa de Subida e Rampa de Descida em Burst - (Subida Rápida – Descida lenta).

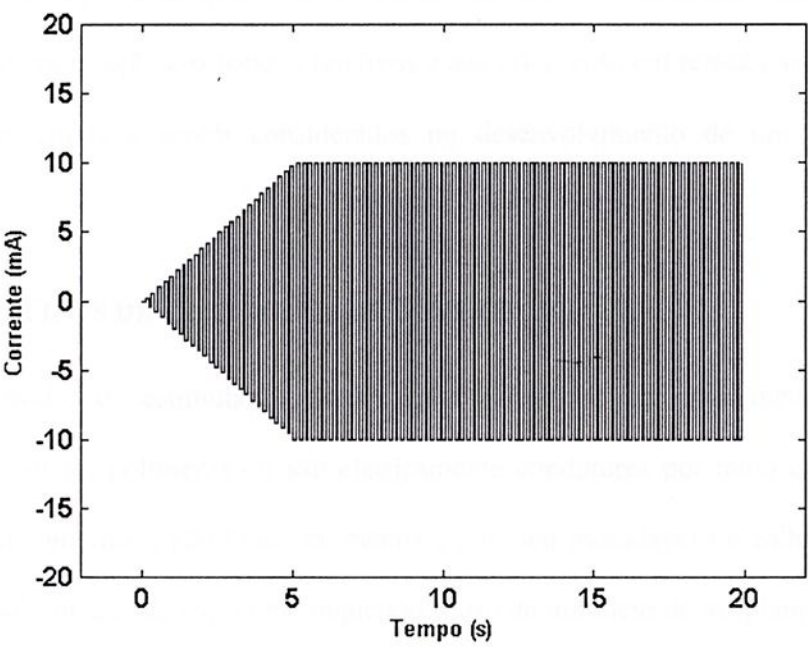


Figura 4: Rampa de Subida e Rampa de Descida em Burst - (Subida lenta – Descida Abrupta).

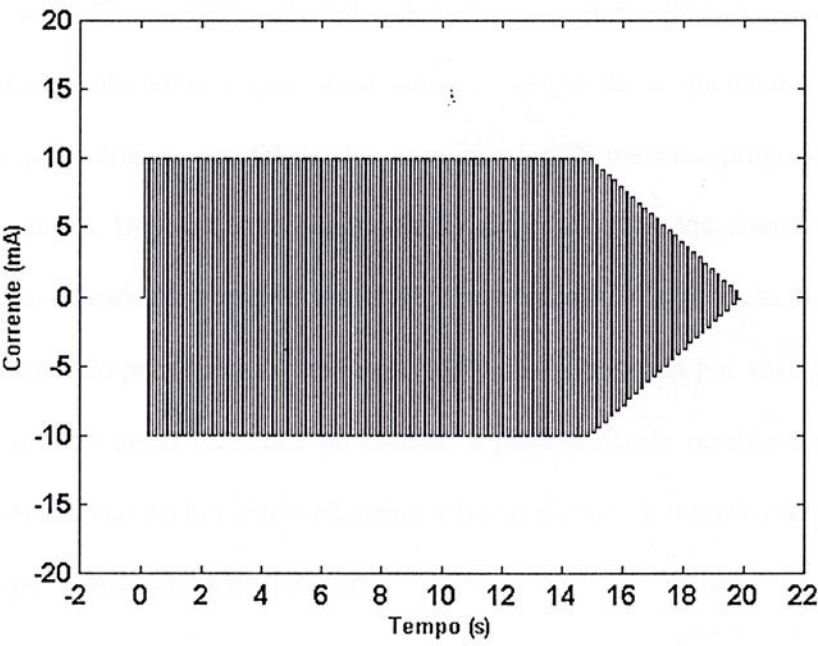


Figura 5: Rampa de Subida e Rampa de Descida em Burst - (Subida Abrupta – Descida lenta).

O material a partir do qual os eletrodos são fabricados, o tamanho e formato do eletrodo, sua localização em relação aos tecidos relativos e sua orientação em relação aos outros eletrodos, são itens importantes a serem considerados no desenvolvimento de um plano terapêutico cauteloso.

2.5.1 TIPOS DE ELETRODOS DE SUPERFÍCIE

Os eletrodos de estimulação usados em eletroterapia são, geralmente, feitos de uma borracha com silício, polímeros ou são eletricamente condutores por meio de impregnação de carbono. Alguns eletrodos são feitos de metais como aço inoxidável ou folha de alumínio. Os eletrodos de estimulação de superfície requerem o uso de um meio de acoplamento para fornecer um caminho de baixa resistência elétrica para facilitar o fluxo de corrente no tecido muscular. Para eletrodos de borracha condutora flexível, o meio de acoplamento pode ser uma pomada, gel, ou um líquido eletrolítico. Alguns eletrodos disponíveis comercialmente são cobertos com um polímero condutor auto-adesivo que serve como o agente de acoplamento. Os eletrodos de borracha e os poliméricos, são fabricados para manter as mesmas propriedades condutoras durante muitos meses. Depois de um período de tempo, esses eletrodos, começam a se deteriorar e não fornecerão fluxo de corrente uniforme sobre toda sua superfície. Essa mudança na capacidade do eletrodo para conduzir uniformemente é acompanhada por variações na densidade de corrente fornecida pelos eletrodos no contato à pele, podendo resultar em altos níveis de estimulação desconfortáveis em locais pequenos (locais quentes), muitas vezes percebidos pelo paciente como uma sensação de queimadura.



2.6 ESPECIFICAÇÃO DO TAMANHO E DA FORMA DOS ELETRODOS

A área de contato requerida dos eletrodos de estimulação depende em parte da área de tecidos excitáveis a serem estimulados. Um eletrodo que é muito grande ou escolhido de forma errada pode fazer com que a corrente se espalhe para estruturas excitáveis que não o nervo ou músculo de interesse. Os eletrodos estão disponíveis em uma grande variedade de tamanhos e formas a fim de abranger o rol de procedimentos eletroterapêuticos na prática contemporânea. A Figura 6, apresenta dois modelos muito utilizados de eletrodos de superfície.

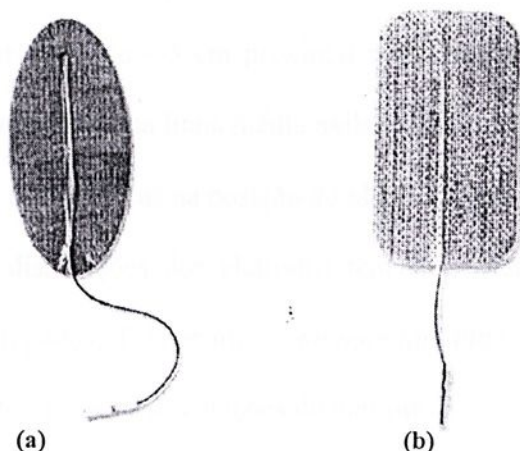


Figura 6: Eletrodos Comerciais: (a) -Oval e (b) - Retangular.

A densidade de corrente (quantidade de corrente por unidade de área) é inversamente proporcional à área de contato do eletrodo, assim, para uma mesma corrente gerada pelo estimulador, à medida que a área do eletrodo aumenta, a densidade de corrente aplicada diminui e vice-versa. Se um eletrodo pequeno e um eletrodo grande são usados em uma única aplicação a estimulação será percebida com maior intensidade no eletrodo pequeno.

Na documentação de procedimentos eletroterapêuticos, é recomendado que o tipo, o tamanho e a forma dos eletrodos sejam claramente especificados e que as mudanças feitas em qualquer uma dessas características do eletrodo sejam também anotadas.

2.7 DESIGNAÇÃO DA LOCALIZAÇÃO DO ELETRODO NAS APLICAÇÕES TERAPÊUTICAS

A colocação de eletrodos na eletroterapia é decisiva para atingir benefícios. Por essa razão, a localização exata deve ser claramente registrada. Para descrever com precisão o local do eletrodo, deve-se medir a distância a partir do centro dos eletrodos para saber os sinais ou estruturas anatômicas (por exemplo - 5 cm proximal a borda superior da patela sobre o reto femoral e 10 cm distal à prega axilar na linha média axilar). Essas medidas devem ser registradas após tratamentos iniciais, e as mudanças na posição do eletrodo devem ser documentadas.

Historicamente, as disposições dos eletrodos têm sido descritas em relação à posição relativa dos eletrodos empregados. Três termos – *monopolar*, *bipolar* e *quadripolar* – têm sido usados regularmente para descrever as orientações do eletrodo.

Na orientação de eletrodo monopolar simples, um único eletrodo é colocado sobre a área de ação ou sobre o tecido onde é desejado o maior efeito. O segundo eletrodo é colocado em uma distância bem longe da área de ação a fim de fechar o circuito elétrico. O eletrodo colocado na área de ação é as vezes chamado de eletrodo de estimulação ou ativo, e o segundo eletrodo (colocado longe da área de ação) é chamado de eletrodo de dispersão ou de referência.

Em algumas situações, o eletrodo único colocado na área de ação é substituído por dois eletrodos conectados ao mesmo canal. Até mesmo essa configuração de colocação do eletrodo



usa dois eletrodos na área de ação, ambos são conectados a um conector de saída única e, consequentemente, esse padrão de colocação é também chamado de monopolar.

Na orientação de eletrodo bipolar simples, ambos os eletrodos de superfície de um canal de estimulador são colocados sobre a área de ação. Na Figura 7(a), é mostrado a estimulação do deltóide anterior e o bíceps braquial. Na Figura 7(b), tem-se a estimulação dos músculos flexores. Esse padrão de colocação de eletrodo fornece uma corrente que é mais limitada aos tecidos excitáveis de interesse.

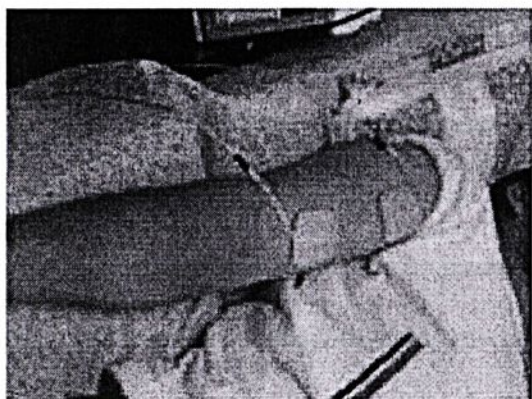
Os eletrodos usados na técnica bipolar geralmente são iguais em tamanho. Assim, a habilidade relativa de cada eletrodo para ativar o nervo ou o músculo será igual quando as formas de onda bifásica simétricas forem aplicadas. Em algumas ocasiões, a técnica bipolar pode requerer que um dos dois eletrodos seja menor a fim de obter a resposta desejada, neste caso, o eletrodo menor será relativamente mais efetivo na ativação do tecido excitável.

Existem outras configurações bipolares de colocação de eletrodos que usam mais de dois eletrodos que se originam de um único canal.

Na configuração de eletrodo quadripolar, dois eletrodos de dois circuitos de estimulação separados são posicionados na área de ação principal.

Duas configurações quadripolares de colocação dos eletrodos são mostradas na Figura 8.



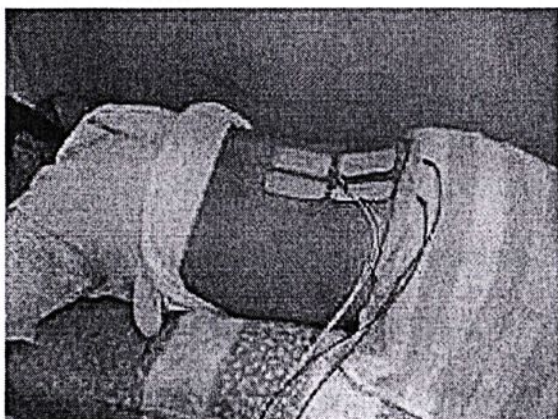


(a)

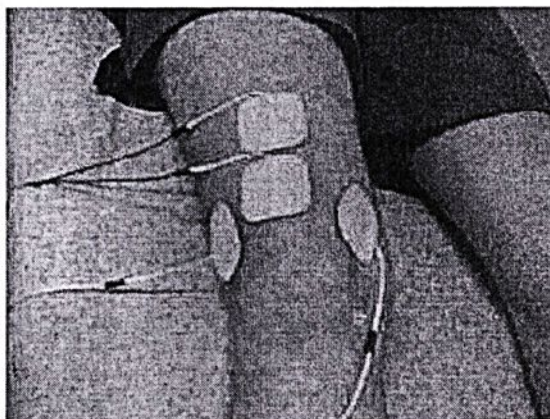


(b)

Figura 7: Configurações de Eletrodos Bipolares.



(a)



(b)

Figura 8: Configurações de Eletrodos Quadripolares.

Na Figura 8(a), dois eletrodos de cada um dos canais de estimulação são colocados na região de ação (parte inferior das costas) de uma maneira que não exista interferência das correntes de cada canal.

Na Figura 8(b), os quatro eletrodos são posicionados de tal forma que as correntes produzidas por cada canal interajam ou interfiram uma com a outra. Esse tipo de colocação de eletrodo tem sido usado para a técnica de corrente interferencial.

2.8 TIPOS DE ESTIMULADORES

Os estimuladores elétricos para aplicações diagnósticas e terapêuticas podem ser genericamente descritos por sua fonte de energia externa (bateria, rede elétrica) e pela natureza do estímulo elétrico que é aplicado ao paciente (corrente constante, tensão constante). Em outros casos, os estimuladores disponíveis comercialmente são geralmente categorizados baseados nas características da forma de onda do estímulo ou de algumas características relacionadas com a técnica de estimulação associada com o estimulador determinado, por exemplo, estimulador de corrente Farádica.

2.8.1 ESTIMULADORES DE CORRENTE VERSUS TENSÃO

Os aparelhos chamados de corrente, são aqueles que têm como princípio a geração de uma forma de onda de corrente (constante), sem fixar no entanto a tensão aplicada no corpo decorrente da relação com a impedância do “corpo” estimulado. Por outro lado, os geradores de tensão, geram uma forma de onda de tensão (constante) que tem um limitante de corrente, devido aos valores de impedância do “corpo” a ser estimulado.

Cada tipo de estimulador tem vantagens e desvantagens potenciais em aplicações clínicas. Durante o tratamento, em um determinado ajuste de amplitude, os estimuladores de corrente constante reduzirão automaticamente a tensão quando o contato do eletrodo melhora ou quando a transmissão elétrica aumenta, mantendo assim, o nível de estimulação desejado. Em casos em que o contato do eletrodo ou a transmissão é reduzida, os estimuladores de corrente constante aumentarão automaticamente a tensão, que pode resultar em queimaduras na pele à medida que a densidade de corrente aumenta para níveis muito altos. Por esta razão, estimuladores de corrente constante, bem projetados, muitas vezes apresentam um limite de



tensão que não pode ser excedido independentemente de um aumento da impedância na interface entre o eletrodo e tecido.

Os estimuladores de tensão constante têm vantagens e desvantagens parecidas às dos aparelhos de corrente constante. Em casos em que o contato do eletrodo ou a transmissão elétrica é reduzida, os estimuladores de tensão constante reduzirão automaticamente a corrente produzida e desse modo diminui a chance de queimaduras da pele, resultantes da densidade de corrente aumentada. Inversamente, se o contato entre o eletrodo e o tecido melhora, a transmissão de cargas aumenta no aparelho de tensão constante, as correntes fornecidas podem aumentar significativamente, resultando em um nível alto de estimulação indesejado.

2.9 CLASSIFICAÇÃO DAS CORRENTES EM ELETROTHERAPIA

2.9.1 DESIGNAÇÕES E NOMENCLATURAS DAS CORRENTES

As correntes elétricas têm sido usadas como propostas terapêuticas por centenas de anos. Com o desenvolvimento de diferentes formas de geradores elétricos durante o século XX, os tipos de correntes elétricas empregados nas aplicações terapêuticas proliferaram. A introdução no mercado de vários tipos diferentes de estimuladores - que produzem diferentes formas de correntes elétricas - tem sido acompanhada por muita confusão devido à falta de padronização e nomenclatura adequada em relação às características das correntes geradas. Antes de 1990, nenhum sistema tinha sido desenvolvido para padronizar descrições de correntes elétricas usadas na eletroterapia.

2.9.2 TIPOS DE CORRENTES ELETROTHERAPÊUTICAS

As correntes elétricas usadas na eletroterapia clínica atualmente podem ser divididas nos



seguintes tipos: corrente contínua, corrente pulsada (pulsátil) uni e bidirecional.

2.9.2.1 Corrente contínua

O fluxo unidirecional contínuo de partículas carregadas é definido como corrente contínua (CC). Quando uma corrente contínua e constante é aplicada em estimulação, é denominada “Corrente Galvânica”. A corrente contínua pode ser facilmente produzida por uma tensão de magnitude fixa aplicada a um condutor com uma resistência fixa. Esse tipo de estimulação tem importância histórica devido aos trabalhos realizados por Luigi Galvani em 1786 mas não tem muitas aplicações em estimulação elétrica hoje, devido a constatação do perigo de queimaduras químicas mesmo quando é uma “Corrente Galvânica Interrompida” por consequência do acúmulo de cargas (íons) nas regiões estimuladas.

Uma solução apresentada na literatura para o problema do acúmulo de cargas é a geração de formas de onda que apresentem pulsos bifásicos, gerando assim, formas de onda que gerem estímulos com fluxo bidirecional de cargas. Desta forma, em um ciclo o estímulo é aplicado e no outro, existe uma dissolução das cargas acumuladas.

2.9.2.2 Corrente pulsada

A corrente pulsada (ou interrompida) é definida como o fluxo uni ou bidirecional de partículas carregadas que periodicamente cessam por um período de tempo definido. Esse termo, descreve a forma de onda de corrente mais comumente usada em aplicações clínicas de estimulação elétrica.



2.9.5 EQUILÍBRIO DE CARGA EM FORMAS DE ONDA BIFÁSICAS

Se uma forma de onda, tem a “área” do ciclo positivo igual a do ciclo negativo, essa função é dita como equilibrada. Em outras palavras, esta função, deve apresentar valor médio nulo no termo DC da série de Fourier.

2.10 MODULAÇÕES DE CORRENTE

No uso de estimulação elétrica para tratamento de problemas do paciente, os parâmetros das correntes relacionados com a amplitude e frequência são muitas vezes variados de uma maneira independente, ou seja, somente um parâmetro pode ser alterado de cada vez. As diferentes formas de modulação de corrente podem ser seqüenciais, intermitentes ou variáveis. Várias das características quantitativas da corrente são moduladas nas aplicações clínicas selecionadas. Variações na amplitude máxima de uma série de pulsos são chamadas de modulações em amplitude, conforme mostrado na Figura 9. Mudanças reguladas no tempo de duração de cada pulso, ou a “largura de cada pulso” é chamada de modulação de duração do pulso ou, na nomenclatura utilizada na engenharia elétrica, modulação PWM (Pulse Width Modulation), conforme ilustra a Figura 10.

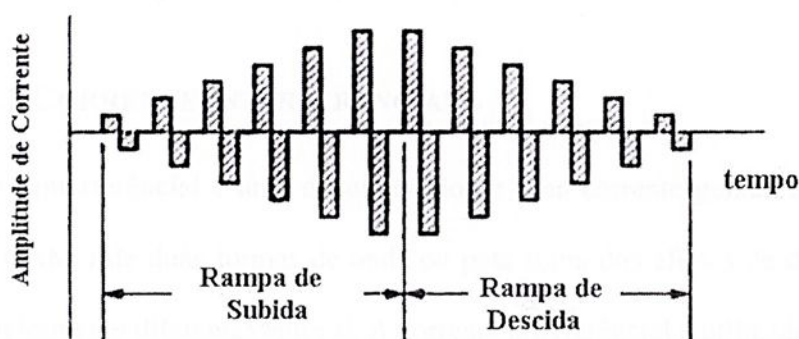


Figura 9: Modulação de Corrente – Modulação em Amplitude.

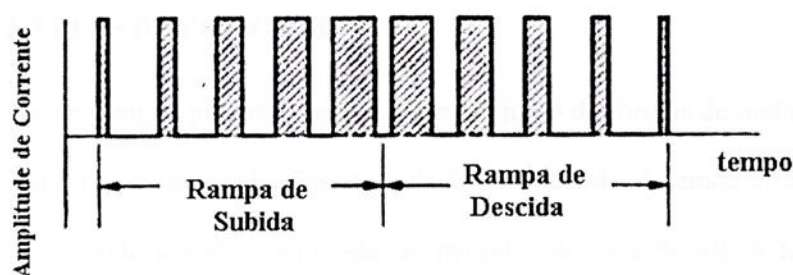


Figura 10: Modulação de Corrente – Modulação de Duração do Pulso.

Outra técnica utilizada de modulação, é a variação do intervalo de tempo entre os pulsos, sendo que os pulsos mantêm a mesma “largura”. Essa forma de modulação leva o nome de modulação PPM (Pulse Position Multiplex) na engenharia ou Wobulação nos termos de fisioterapia, mostrado na Figura 11.

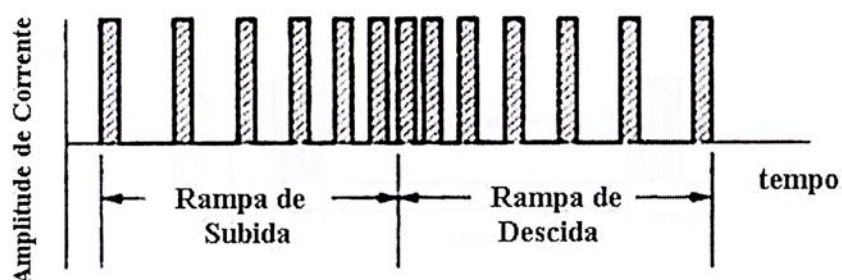


Figura 11: Modulação de Corrente – Modulação PPM.

2.10.1 CORRENTE INTERFERENCIAL

Corrente Interferencial é uma denominação de uma corrente gerada ou pela modulação em amplitude (AM) de duas formas de onda ou pela soma dos efeitos de duas correntes com frequências ligeiramente diferentes entre si. A corrente interferencial é utilizada em tratamento de dores musculares e lesões geralmente utilizando no tratamento, quatro ou mais eletrodos.

2.10.2 TRENS DE PULSOS, BURST

Um pacote de trem de pulsos, representa um conjunto de formas de onda iguais, repetidas no tempo e seguidas de um intervalo. Esse tipo de forma de onda, é gerado através de uma série contínua de pulsos, sendo a saída controlada em períodos de *on* e de *off*. A forma de onda do pulso aplicado e a relação entre os períodos é definida no protocolo de estimulação de acordo com as necessidades do paciente. O protocolo tem que ser adequado para evitar que a fadiga muscular ocorra de forma precoce. A Figura 12, apresenta protocolos de estimulação utilizando pulsos monofásicos retangulares.

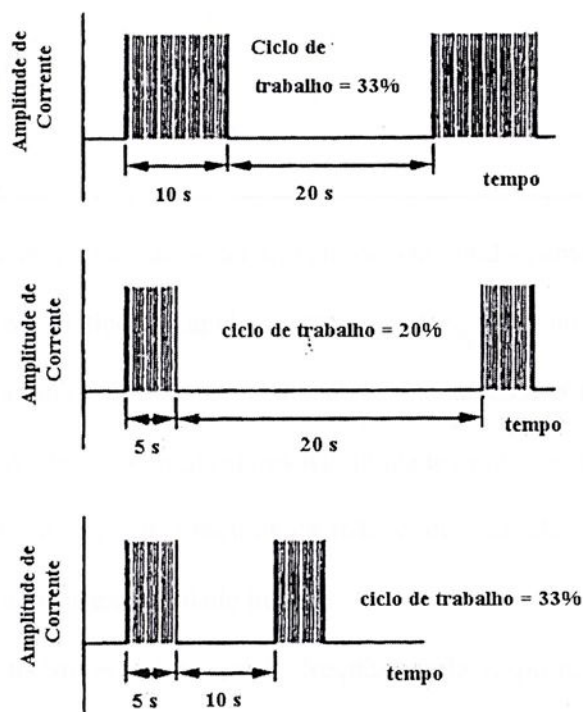


Figura 12: Controles da Formas de Ondas por Burst.

Para que se possa entender o comportamento dos nervos e do músculo sob as condições de estimulação, apresenta-se no capítulo 3 um resumo da fisiologia do nervo e do músculo

Capítulo 3

FISIOLOGIA DO NERVO E DO MÚSCULO

O músculo esquelético não deve ser tratado ou entendido como uma estrutura única e uniforme. Fibras de diferentes tipos (rápidas, lentas, grossas, finas etc.) em conjunto com o motoneurônio e com o axônio, formam o elemento funcional básico do sistema muscular, a unidade motora. O número de fibras musculares nas unidades motoras varia conforme a região muscular, variando de 200 a 300 em músculos da mão e até mais de 1000 fibras por unidade motora em músculos grandes da extremidade inferior.

Devido as características diferentes de frequência de resposta de uma solicitação de força de cada unidade motora de um músculo, durante a contração muscular voluntária, as unidades apresentam um comportamento assíncrono de resposta. Este fato, apresentado na Figura 13, mostra como pode ser gradativo e suave o aumento de força total resultante. Nota-se as diferenças de frequências de respostas individuais de cada fibra na solicitação voluntária de força.

A resposta a uma estimulação elétrica neuromuscular é muito diferente da ação de uma contração voluntária.

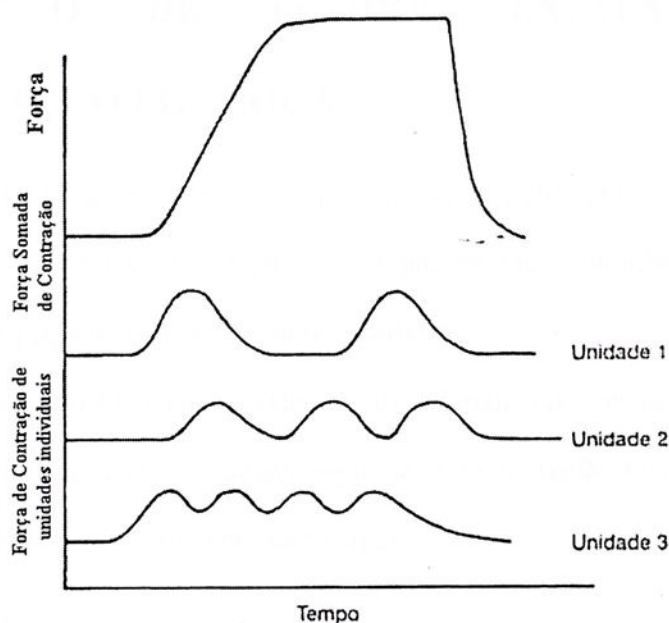


Figura 13: Resultado da Ativação das Unidades Motoras por Contração Voluntária.

A estimulação elétrica neuromuscular, produz ação potencial em nervos e músculos indistintamente. Acionada pelas cargas do potencial elétrico do estímulo as fibras apresentam resposta síncrona, de acordo com a frequência do estímulo. A ação elétrica aciona fibras nervosas periféricas sensitivas (sensação formigamento) em baixa frequência e baixa amplitude de corrente (corrente menor que 20 mA e frequência menor que 80 Hz) e sensação de dor para corrente de alta amplitude (corrente maior que 100 mA).

Potenciais elétricos de alta amplitude (acima de 50 mA) podem, acionando alguns tipos de fibras nervosas periféricas, ajudar a liberação neurotransmissores que atuam de forma a diminuir a sensação de dor. Este efeito, quando bem administrado, contribui no controle da dor e na diminuição farmacológica.

3.1 ATIVAÇÃO DE TECIDOS EXCITÁVEIS COM ESTIMULAÇÃO ELÉTRICA

A circulação de corrente por entre os tecidos, promove a ativação e excitação de nervos e músculos. A Figura 14 mostra dois eletrodos de um circuito estimulador aplicado à pele diretamente sobre um pequeno ramo de um nervo periférico.

Quando a estimulação é iniciada, um eletrodo transitoriamente contém um excesso de carga negativa e o outro está deficiente em cargas negativas. Íons na região migrarão em direção ou para longe desses eletrodos de acordo com suas cargas.

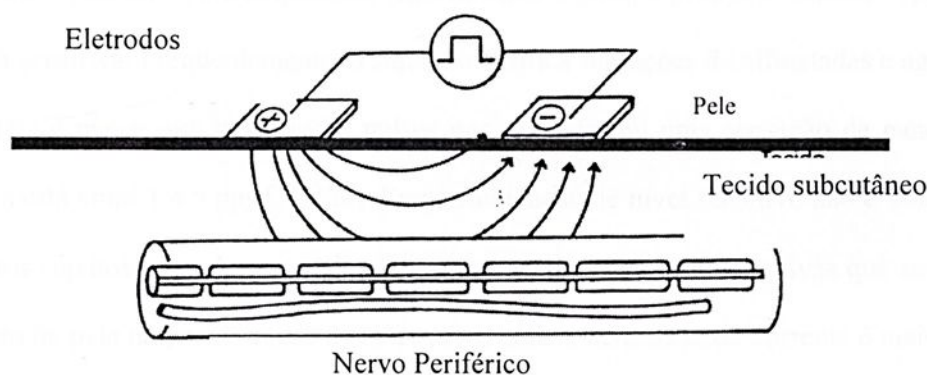


Figura 14: Eletrodos Aplicados à Pele – Corrente Circulante

O padrão de corrente circulante pelo movimento desses íons é representado pelas linhas conectando os eletrodos. Alguns dos movimentos iônicos ocorrem no fluido extracelular, e alguma das correntes passam através da membrana do nervo. O efeito de rede dessa corrente é uma fraca despolarização da membrana do nervo. Se a breve corrente circulante através da membrana for muito baixa, a mudança no potencial de membrana rapidamente retornará ao potencial de membrana de repouso. Se a corrente circulante através da membrana for grande o

suficiente, um potencial de ação será evocado e propagado ao longo da membrana. Esse potencial de membrana evocado é idêntico ao produzido ao longo da membrana da fibra em resposta à ativação fisiológica.

3.2 RESPOSTA CLÍNICAS PARA A ESTIMULAÇÃO NERVOSA E MUSCULAR

3.2.1 ESTIMULAÇÃO EM NÍVEL SENSITIVO

Quando eletrodos são colocados na superfície da pele e a amplitude da estimulação elétrica é gradualmente aumentada, três formas gerais de resposta são consistentemente notadas em indivíduos normais. Para amplitudes relativamente baixas, a primeira resposta reportada pelo sujeito é a sensitiva. Frequentemente, o sujeito identifica sensações de alfinetadas e agulhadas, se a frequência é maior que uns poucos pulsos por segundo, ou uma sensação de pancadas, se a frequência está entre 1 a 5 pps [3], [20]. Essa estimulação de nível sensitivo não é desconfortável para muitos sujeitos e resulta da excitação daquelas fibras nervosas sensitivas que se encontram na ou perto da pele na proximidade dos eletrodos, onde a densidade de corrente é maior. Quando a amplitude da estimulação é gradualmente aumentada, a sensação torna-se forte e com frequência expande-se para a região entre os eletrodos e é mais profunda nos tecidos. Como pode ser notado na Figura 15, a estimulação de nível sensitivo (que é o objetivo da estimulação elétrica), pode ser alcançada pelo uso de muitas diferentes combinações de amplitude de estímulos e de duração do pulso.

Se a estimulação de nível sensitivo para frequências maiores do que aproximadamente 15 pps é mantida por períodos prolongados de tempo, o sujeito geralmente notará uma diminuição

gradual na habilidade para sentir a estimulação; esse fenômeno é chamado de adaptação, ou acomodação do músculo. A acomodação pode ser reduzida pela interrupção intermitente da estimulação, ou pela variação de sua amplitude, ou de sua frequência, ou ainda pela mudança da forma de onda da corrente do estímulo [20].

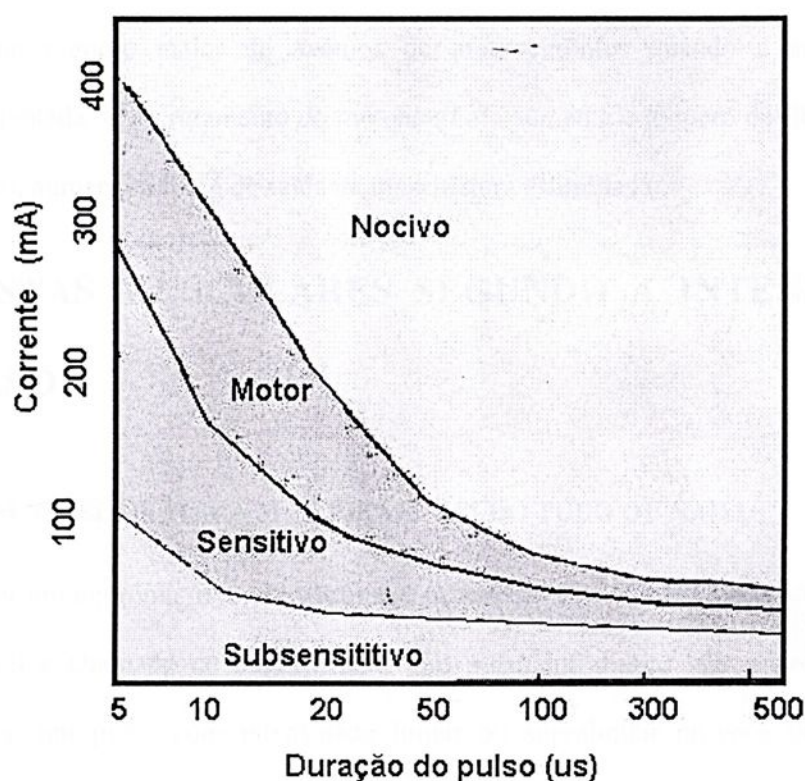


Figura 15: Níveis de Estimulação em Relação a intensidade corrente - Duração do Pulso.

3.2.2. ESTIMULAÇÃO DE NÍVEL MOTOR

Quando a intensidade de estimulação é gradualmente aumentada, a sensação de formigamento sentida pelo indivíduo aumenta tanto quanto progressivamente maior for o número de fibras nervosas sensitivas recrutadas. Para níveis baixos de estimulação motora, a contração

somente pode ser detectada pela palpação da musculatura inervada. Quando a amplitude (ou duração do pulso) é aumentada, a contração logo se torna forte o suficiente para produzir visível movimento articular.

A forma e o comprimento da contração muscular produzida pela estimulação elétrica são determinados pelos mesmos dois processos envolvidos na contração voluntária. Há estimulação progressiva de um número maior de axônios de motoneurônios quando a amplitude da estimulação é aumentada. O recrutamento do motoneurônio aumenta o número de fibras motoras ativadas, e portanto, aumenta a força de saída da musculatura estimulada.

3.3. RESPOSTAS MUSCULARES SEGUNDO A INTENSIDADE DO PULSO

3.3.1 EM NÍVEL DE FIBRA MUSCULAR: LEI DO TUDO OU NADA

Para ativar um neurônio ou motoneurônio é necessário que o pulso tenha um certo nível mínimo. Esse nível é chamado de pulso limiar. Um pulso infralimiar não provoca resposta muscular. E ainda, um pulso com intensidade limiar ou supralimiar provoca uma resposta tendendo à máxima. Não se observa progressividade crescente de resposta.

Os estímulos de infralimiais a supralimiais provocam uma contração progressivamente crescente, excitando um mínimo crescente de unidades motoras. Um estímulo eficaz máximo dá uma contração muscular máxima. A Figura 16 apresenta o desempenho da força apresentada segundo uma função progressiva de solicitação de força [3].



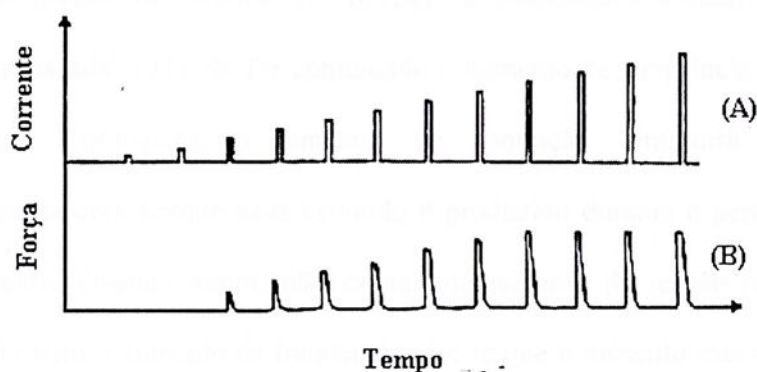


Figura 16: Estimulo Aplicado (A) e o Desempenho do Músculo (B).

3.4. RESPOSTAS MUSCULARES SEGUNDO A FREQUÊNCIA DO PULSO

3.4.1. TETANIZAÇÃO

Quando o músculo é estimulado em frequências progressivamente maiores, atinge-se por fim uma frequência em que as contrações sucessivas se fundem e não podem ser distinguidas umas das outras. Esse estado denomina-se tétano, e a frequência mais baixa onde isso ocorre é conhecida como frequência crítica.

O tétano resulta em parte das propriedades viscosas do músculo e, em parte, da natureza do próprio processo contrátil. Depois que a frequência crítica para o tétano é atingida, aumentos subsequentes na frequência de estimulação aumentam a força de contração apenas em pequena porcentagem [20].

Se a frequência dos estímulos é inferior a 8 Hz, o músculo responde por contrações isoladas e sucessivas que são chamadas de fibrilações elementares. Se a frequência dos estímulos é aumentada acima de 10 Hz, observa-se uma tremulação; cada fibrilação elementar é iniciada

antes que a fibrilação precedente termine. As fibrilações elementares somam-se e a tensão muscular aumenta “em escada” [21]. Se for continuado o aumento de freqüência dos estímulos além das freqüências fisiológicas, o aumento de contração diminuirá e depois e consequentemente desaparecerá porque cada estímulo é produzido durante o período refratário induzido pelo precedente. Quando assim, não existe um aumento da tensão na resposta da intensidade do estímulo com o aumento de freqüência, diz-se que o músculo está tetanizado [3].

A Figura 17 representa um gráfico da resposta muscular em relação a freqüência do estímulo.

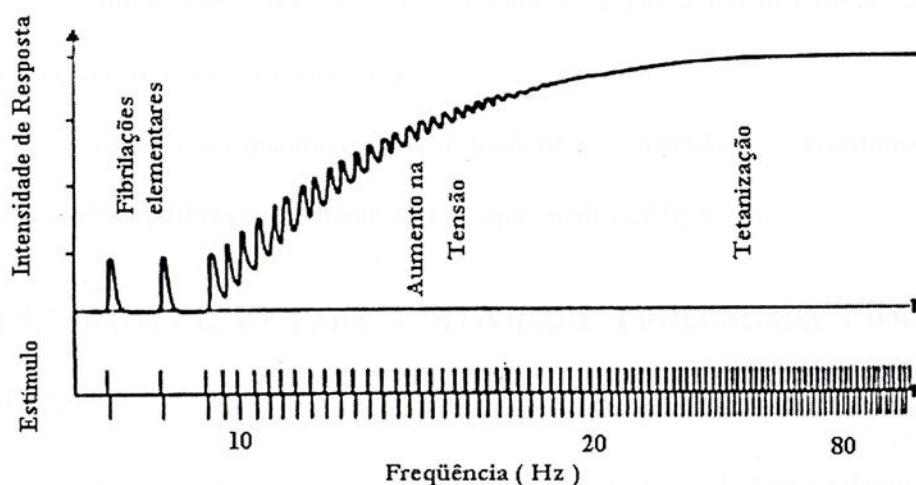


Figura 17: Resposta Muscular em Relação ao Aumento de Freqüência.

3.5. ESTIMULAÇÃO ELÉTRICA DO MÚSCULO: TÉCNICAS E APLICAÇÕES

Um fator que aumentou o interesse na EENM (Estimulação Elétrica Neuro Muscular) foram os relatórios na metade dos anos 70 sobre a eficácia dos programas de treinamento de EENM para promover o desenvolvimento de força em atletas de elite e indivíduos saudáveis.

Os trabalhos contemporâneos sobre os efeitos da EENM sobre o músculo esquelético focalizam não somente seus efeitos no músculo saudável, mas também sua utilidade no tratamento de músculos fracos.

3.5.1. PLASTICIDADE MUSCULAR EM RESPOSTA À ESTIMULAÇÃO ELÉTRICA

As características do músculo esquelético não são imutáveis [22]. Em resposta às mudanças no uso do músculo, suas características estruturais, bioquímicas e fisiológicas adaptam-se para satisfazer mais apropriadamente as demandas impostas. Fibras rápidas podem se tornar lentas ou fibras lentas podem se tornar rápidas, dependendo dos níveis de atividade e tempo de duração das atividades solicitadas.

Características como quantidades de mitocôndrias, capacidade de consumo de ATP etc., diferenciam células de fibras musculares e suas capacidades de resposta.

3.5.2 ADAPTAÇÃO PARA A ATIVIDADE PROLONGADA COM NÍVEL DE FORÇA BAIXO

Os músculos que são requeridos para trabalhar em níveis de força relativamente baixos por longos períodos de tempo nas atividades de vida diária, desenvolvem uma capacidade aumentada em fornecer ATP dos depósitos de energia para reabastecer a contração muscular repetitiva. Essas mudanças caracterizam a resposta muscular adaptativa não somente para programas de treinamento de resistência voluntária, mas também para as mudanças no uso causadas pela estimulação elétrica crônica de nível baixo do músculo.

3.5.3 ADAPTAÇÃO PARA A ATIVIDADE PROLONGADA COM NÍVEL DE FORÇA ALTO



Quando o músculo é ativado para que a maioria das fibras musculares seja recrutada, altas forças são geradas. As forças dessas contrações requerem alta energia; como resultado, os depósitos de energia usados para reabastecer essas contrações são rapidamente esgotados. Consequentemente, essa forma de contração pode ser experimentada somente por períodos de tempo muito curtos antes que a fadiga comece a se manifestar. Apesar do início rápido da fadiga esse padrão de contração muscular está associado com mudanças no músculo que levam a aumentos da força muscular.

3.6. SELEÇÃO DE FORMA DE ONDA DA CORRENTE EM EENM

A escolha da forma de onda empregada em EENM é influenciada pela síntese dos resultados de diversos estudos que, monitorando a força produzida em resposta à estimulação, a CMV, e o nível confortável do indivíduo durante a estimulação descreveram as preferências entre várias formas de onda. Os estudos sobre a escolha de forma de onda, apesar de não fornecer a resposta final dessa questão, apresentam um caminho para respondê-la. Procedimentos para a relativa tolerância à EENM podem ser avaliados determinando a preferência do sujeito, ou conforto relativo, ou determinando a força de contração muscular em resposta a níveis máximos toleráveis de estimulação.

3.6.1. PREFERÊNCIA DO INDIVÍDUO E CONFORTO

Um dos mais importantes aspectos da estimulação é a tolerância do indivíduo ao procedimento, o que pode afetar o potencial de resultado do tratamento. O limite de tolerância à estimulação pode apresentar procedimentos ineficazes de EENM. Delitto e Rose [13] avaliaram o



conforto do indivíduo em resposta à estimulação elétrica utilizando três diferentes formas de onda bifásica simétrica (senoidal, triangular e retangular). Esses três tipos de forma de onda foram usados para atividade elétrica do grupo muscular quadríceps em indivíduos normais. A amplitude de estimulação foi gradualmente aumentada durante as sessões de estimulação para amplitudes que evocariam magnitudes similares ao torque de extensão do joelho; indivíduos foram, então, questionados a indicar seus níveis de conforto numa escala visual analógica. Nenhuma forma de onda foi encontrada para ser a mais confortável na evocação de níveis compatíveis de contração. Num estudo similar, Bowman e Baker [14], compararam formas de onda bifásica simétrica e assimétrica aplicadas para evocar contrações de músculos quadríceps em aproximadamente 20% do nível voluntário máximo. Seus resultados demonstram que a maioria dos indivíduos preferiu a forma de onda bifásica simétrica ao invés da forma de onda assimétrica na tensão constante e nos estimuladores com corrente constante.

3.7. ELETROESTIMULAÇÃO INTEGRADA

A eletroestimulação elétrica excitomotora somente tem valor terapêutico quando é integrada em um tratamento de reeducação ativa à qual é concomitante e não substituída. Assim, o uso desta, deve ser acompanhado de outras alternativas sempre que possível. As correntes excitomotoras devem então ajudar a contração voluntária e não substituí-la [1] .

3.8. RISCOS E CONTRA-INDICAÇÕES

O emprego da EENM deve ser usado com algumas cautelas. Em tratamentos com correntes desequilibradas (por exemplo: contínuas ou contínuas interrompidas) pode haver uma concentração muito grande de cargas o que, causa vermelhidão e eventualmente, danos à pele.

Correntes com amplitudes maiores que 100 mA (consideradas nocivas nas estimulações

de fibras), podem atuar no tratamento de síndromes dolorosas em pontos nervosos, mas deve-se utilizar pulsos de curta duração para que não haja um grande aquecimento de tais áreas.

EENM não deve ser empregada, ou deve ser usada com cautela, em pacientes com os seguintes problemas:

Marcapassos;

Indivíduos hipertensos e hipotensos; EENM pode afetar as respostas autônomas desses pacientes;

Áreas de excesso de tecido adiposo em pacientes obesos; esses indivíduos podem necessitar de níveis elevados de estímulos, o que poderá acarretar alterações autônomas;

Tecido neoplástico;

Áreas desvitalizada, por exemplo, em seguida a uma terapia por aplicação de raios-X profundos;

Pacientes incapazes de entender a natureza da intervenção, ou de fornecer um *feedback* acerca do tratamento;

E também, o tratamento não deve ser aplicado nas seguintes áreas:

Seio carotídeo;

Região torácica;

Nervo frênico;

Tronco de uma mulher grávida;

Proximidades dos olhos.

No próximo capítulo é apresentado a evolução dos sistemas digitais que permitem o desenvolvimento de sistemas e ferramentas para otimização e melhoria das soluções digitais de projetos.



Capítulo 4

SISTEMAS DIGITAIS

Com a mudança dos padrões de consumo da sociedade, houve a necessidade da evolução dos processos de produção tecnológicos.

O mercado demanda por circuitos cada vez mais eficientes e redução no tempo de desenvolvimento; a capacidade de integração de circuitos avançou muito significativamente nos últimos anos. Circuitos desenvolvidos nos anos 60 não apresentavam mais de que algumas centenas de transistores, e em contrapartida, no final dos anos 80, circuitos podiam ser fabricados com mais de um milhão destes.

Ferramentas CAD (*Computer Aided Design*) baseadas em captura de esquemático apresentam uma dificuldade enorme de implementar circuitos com centenas de elementos. Deste fato, surge a necessidade do estudo e desenvolvimento de sistemas capazes de implementar circuitos e sistemas digitais com um maior número de elementos.

O uso de ferramentas computacionais é indispensável na fase de projeto quando o sistema se utiliza de muitos elementos. Várias ferramentas computacionais foram desenvolvidas nos últimos anos para auxiliar no processo de síntese de circuitos digitais, tais como:



simuladores, compiladores, ferramentas para melhor alocação de estado, minimização booleana etc., e contribuem para um rápido e eficiente desenvolvimento e prototipagem agregando, assim, maior valor ao projeto.

Para o desenvolvimento de um projeto digital é necessário o entendimento das etapas de projeto, critérios de otimização, etapa de síntese e os tipos de elementos de um sistema digital.

4.1 O PROCESSO DE PROJETO DE SISTEMAS DIGITAIS

O processo de projeto pode ser definido, de forma simplificada, como a transformação de uma descrição inicial do sistema (freqüentemente denomina especificação), em uma descrição final, também chamada de projeto final ou projeto detalhado. A diferença fundamental entre a descrição inicial e a descrição final está no fato desta última conter todas as informações necessárias à fabricação do sistema, ao contrário da primeira.

Cabe ressaltar que, em geral, a descrição final que atende aos requisitos da descrição inicial não é única.

O processo de projeto pode ser então dividido em dois tipos de atividades, as de síntese onde se agrega informação a uma descrição, produzindo uma descrição mais próxima da descrição final, e as de análise, onde se verifica o acerto das tomadas de decisão durante uma etapa de síntese.

Dois conceitos a serem considerados durante o processo de projeto são os de correção e otimização do sistema a construir. Uma descrição final é correta se ela atende a todos os requisitos de descrição inicial e pode ser fabricada. Uma descrição final é ótima se ela é correta e possui custo mais baixo e melhor desempenho que qualquer outra solução correta. O processo de projeto deve ser guiado pela busca de uma das soluções ótimas. Contudo, na maior parte dos



casos, a solução ótima não existe, pois algum compromisso entre custo e desempenho deve ser obtido.

Os critérios de otimização para sistemas digitais podem ser resumidos em:

- Espaço – o sistema digital deve ser o menor possível;
- Tempo – o sistema digital deve ser o mais rápido possível;
- Energia – o sistema digital deve consumir o mínimo de energia por unidade de tempo.

Como estes critérios são muitas vezes conflitantes entre si, a cada projeto deve-se favorecer aqueles que concorram mais facilmente à satisfação dos requisitos particulares da descrição inicial. Por exemplo, durante o projeto de um telefone celular deve-se favorecer os critérios de espaço e energia, ficando o de tempo em segundo plano. Por outro lado, o projeto de um super-computador deve favorecer em primeiro lugar o critério tempo, ficando os outros em um plano secundário.

A partir de uma descrição abstrata o projetista deve ser capaz de uma descrição detalhada de forma tal que o projeto seja, então, ótimo e mantenha a funcionalidade e desempenho a um custo mínimo.

4.1.1 UM MODELO PARA O PROCESSO DE PROJETO – O DIAGRAMA Y

Em termos de hierarquia, um projeto digital pode ser dividido em quatro níveis ascendentes de abstração:

- nível elétrico (Dispositivo)- onde os modelos dos componentes provêm da teoria de circuitos elétricos e eletrônicos, e ou da física de semicondutores, isto é, transistores, resistores, capacitores, equações diferenciais, diagramas elétricos etc.;



- nível lógico - onde o sistema digital é descrito a partir de noções elementares da teoria de circuitos digitais: portas lógicas, Flip-Flops, equações Booleanas, tabelas e grafos de transição etc.;
- nível arquitetural (RTL) - onde os modelos de base são os componentes mais complexos da teoria de circuitos digitais e as formas de descrevê-los: registradores, decodificadores, ULAs, multiplexadores, linguagens de transferência entre registradores, grafos de fluxo de dados e de controle etc;
- nível sistêmico (Arquitetura) - aqui, o sistema digital é descrito como um conjunto de algoritmos e módulos capazes de executar aqueles que podem ou não ter um mapeamento direto para um *hardware* existente: processadores, memórias, componentes abstratos implementados no *software* básico do sistema, tais como um pacote de ponto flutuante em um sistema sem coprocessador aritmético, ou monitores de sistemas operacionais.

Pode-se, ainda, decompor as informações de um sistema digital em três diferentes categorias:

- **Físico** - contém informações geométrica sobre os componentes, módulos e sobre a disposição espacial destes no sistema a ser fabricado;
- **Estrutural** - contém informações sobre como interconectar blocos de base de comportamento conhecido para realizar determinada funcionalidade, sem se ocupar com a disposição física destes no sistema;
- **Comportamental** - carrega informações sobre o comportamento do sistema, sem se envolver em como tal comportamento pode ser obtido, seja do ponto de vista físico, seja do ponto de vista estrutural.

A combinação de níveis de abstração e domínios de descrição provê um modelo ao mesmo tempo simples e poderoso para interpretar o processo de projeto de sistemas digitais. Uma forma de representação de tal modelo bidimensional foi proposta pela primeira vez por Gajski e Kuhn [23], o chamado Diagrama Y, uma amostra do qual aparece na Figura 18.

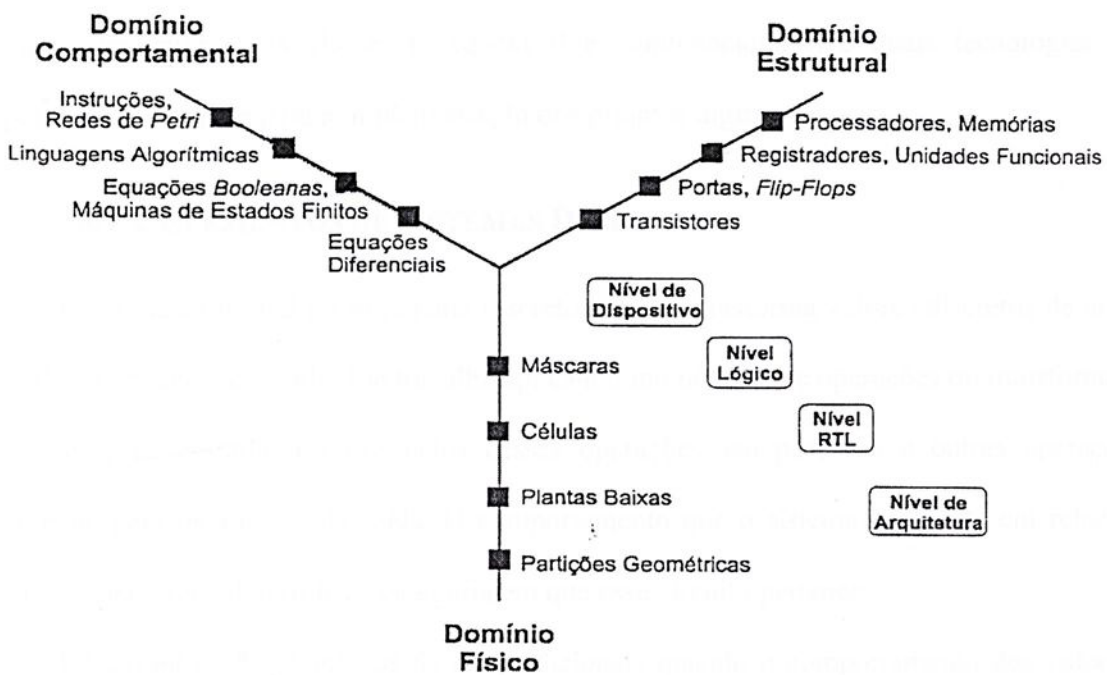


Figura 18: Elementos primitivos de cada nível de abstração de Gajski e Kuhn (Diagrama Y).

Neste diagrama, os níveis de abstração (Nível de Dispositivo, Nível Lógico, Nível RTL, Nível de Arquitetura) são identificados pelas suas distâncias do centro do diagrama, enquanto que segmentos de reta radiais correspondem a domínios de descrição. Por exemplo, um diagrama de esquemáticos de portas lógicas TTL é uma descrição estrutural lógica, estando portanto localizado na intercessão do eixo rotulado Domínio Estrutural com o nível lógico de abstração.

Em outra perspectiva, grafos orientados - levam de um elemento primitivo à outro elemento primitivo - representam transformações. Estas transformações são ferramentas de síntese, ferramentas de minimização, ferramentas de otimização, ferramentas de alocação de estado etc.; assim, são ferramentas que agregam otimização e rapidez de desenvolvimento ao projeto.

Para o desenvolvimento de sistemas digitais é necessário o entendimento das estruturas de seus elementos, suas classes (seqüencial e combinacional) e quais tecnologias estão disponíveis atualmente para a implementação dos projetos digitais.

4.1.2 ELEMENTOS DE SISTEMAS DIGITAIS

Um sistema digital é um sistema discreto, isto é, transforma valores discretos de entrada em valores discretos de saída. Ele trabalha aplicando um número de operações ou transformações nos valores de entrada. Os resultados dessas operações são passados a outras operações e finalmente para os valores de saída. O comportamento que o sistema apresenta em relação as variações de entrada determina a categoria em que esse circuito pertence.

Os circuitos são chamados de combinacionais quando o comportamento dos valores de saída, dependem exclusivamente dos valores atuais de entrada. Esses, são constituídos puramente de portas lógicas dos tipos AND, OR e NOT.

Os circuitos ditos seqüenciais, são circuitos que o comportamento de saída depende não somente dos valores atuais de entrada, mas também de valores e estados anteriores que permaneceram armazenados. Além de possuírem elementos lógicos, apresentam necessariamente algum dos tipos de elementos de memória, geralmente Flip-Flops.

A classe de sistema digital a ser utilizada depende do projeto e das disponibilidades de tecnologia.



Entre as principais tecnologias de sistemas digitais destacou-se por muito tempo o uso de componentes discretos, onde cada componente contém um sistema digital que pode ser sequencial ou combinacional e que não pode ser alterada. Entre os sistemas digitais encontrados que utilizam componentes discretos pode-se citar as memórias, registradores, decodificadores, processadores etc.

Com a evolução dos processos industriais e o surgimento da necessidade de componentes cada vez mais complexos e mais rápidos, surgem as tecnologias de Lógica Programável. Nelas, os componentes podem ser programados de acordo com as especificações do projeto, podendo ser alterada ou configurada sempre que necessário (algumas tecnologias permitem a configuração apenas uma vez).

A facilidade de programação permite que os sistemas que utilizam Lógica Programável possam ser utilizados na confecção de protótipos em um tempo de desenvolvimento muito curto. Nos próximos tópicos são apresentados a evolução dos sistemas de lógica programável e suas vantagens.

4.1.2.1. Uso de Lógica Programável

Elementos digitais, em termos de arquitetura, podem ser dispostos no CI em disposições de vetores e ainda permitir a configuração destes posteriormente (externamente). A evolução desses sistemas permitem que hoje possa, em um único CI, estabelecer a configuração de um sistema completo que contenha processador, memórias, registradores tudo configurável pelo usuário (tecnologia SOC, *system-on-a-chip*).

Os primeiros circuitos lógico programáveis foram introduzidos no mercado pela Philips nos anos 70. Os PLAs (*Programmable Logic Arrays*) consistem de dois níveis de portas lógicas, ambos programáveis formando um plano *wired-OR*. Apesar da inovação, este tipo de

componente apresentava grandes desvantagens tais como alto custo de implementação e atrasos de propagação muito significativos [24].

Os PLDs (*Programmable Logic Device*) são os componentes lógicos da segunda geração surgiu nos anos 80. São componentes lógicos programáveis de vários tipos de especificações de densidade e tecnologia empregada, constituídos por macrocélulas, blocos interconectores, canais de roteamento e trilhas.

Os blocos interconectores permitem fazer no processo de roteamento uma adequada ligação entre os blocos lógicos. Permitem também, quando necessário, a configuração de quais os pinos de *I/O* serão utilizados facilitando conexões com outros dispositivos.

O que difere as famílias de PLDs são a quantidade de macrocélulas implementadas (densidade), tamanho do circuito da macrocélula, grau de configurabilidade da macrocélula, desempenho em relação à atrasos entre as trilhas de roteamento, frequência máxima de trabalho do CI etc.

Com a evolução destes, surgiram PLDs de alta densidade. Entre os principais PLDs de alta densidade, destacam-se os FPGAs e os CPLDs (*Complex Programmable Logic Device*).

Na Figura 19 mostra-se, de forma simplificada, a evolução dos componentes de lógica programável.

Existe uma certa indefinição quanto as nomenclaturas de FPGAs e CPLDs. Em muitos artigos são apresentados componentes como FPGAs que estruturalmente são CPLDs. Este fato se deve a popularização do termo FPGA como sinônimo de componente de lógica programável.



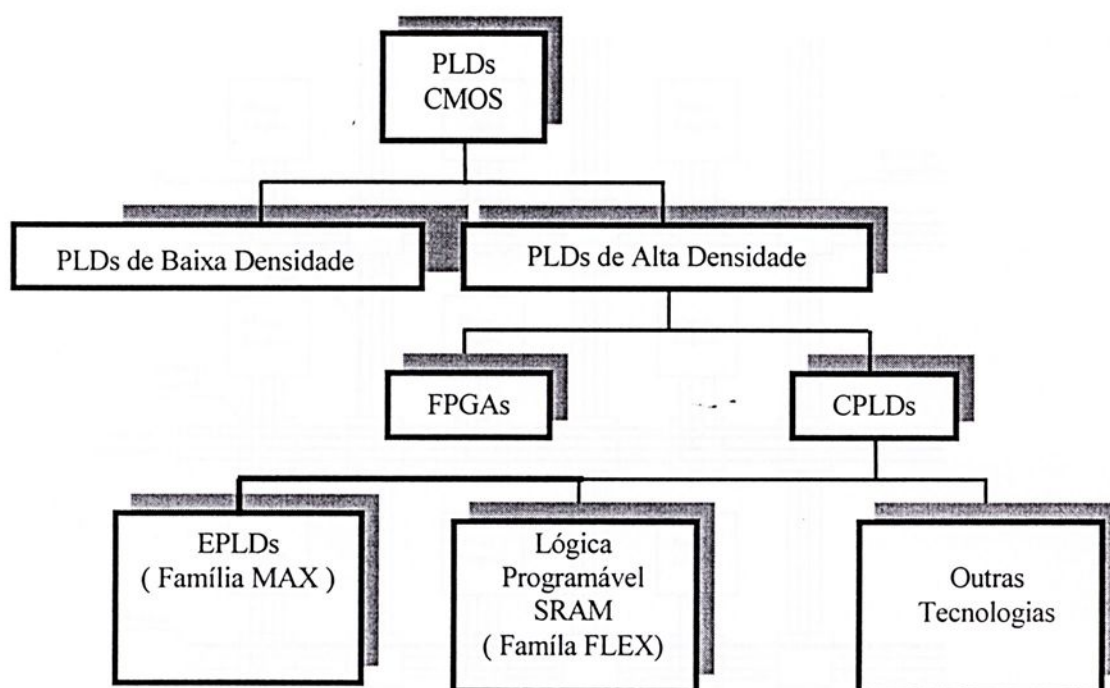


Figura 19: Evolução dos Componentes de Lógica Programável.

A diferença entre os FPGAs e os CPLDs é que estes últimos permitem a configuração dos blocos lógicos em níveis mais profundos, permitindo a configuração de Flip-Flops contidos nas macrocélulas. Outro fator é que em um CPLD o atraso de propagação é mais previsível (não depende do roteamento escolhido), ao contrário do que acontece no FPGA.

A Figura 20 apresenta de forma simplificada a arquitetura de um FPGA, mostrando seus blocos interconectores (blocos de comutação e blocos de conexão), trilhas e macrocélulas (blocos lógicos).

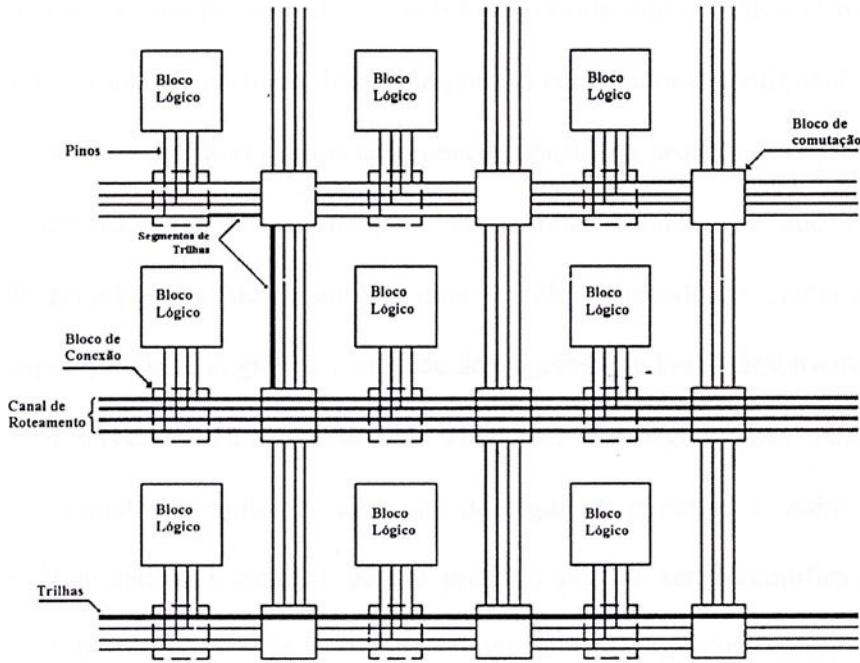


Figura 20: Arquitetura de um FPGA.

4.1.2.2 Características no uso de FPGA

A utilização dos componentes lógico programáveis é decorrente de algumas vantagens dentre as quais pode-se citar:

1. Arquitetura em Hierarquia de Projeto:

Muitos projetos de circuitos digitais simples podem ser desenvolvidos com técnicas de projeto usando componentes *standard*. Entretanto, quando o projeto é complexo, existe uma dificuldade enorme de interpretação, implementação, alocação e de alterações pelo projetista em melhorar ou corrigir alguns parâmetros do projeto.

Uma alternativa em lidar com projetos digitais cada vez maiores é o uso de ambientes de projeto de trabalho com arquitetura em hierarquia. Neste, cada projeto compilado e validado pode ser interpretado como um símbolo ou como um componente *standard* que é incluído para

ser reutilizado em novos projetos. Essa facilidade permite que o projetista possa criar projetos cada vez mais complexos partindo de sub-projetos já conhecidos e configurados segundo as suas próprias restrições. E, uma vez necessária uma alteração no projeto, pode-se alterar somente o sub-projeto que necessite de modificações. Outro fato importante é que, nas bibliotecas do ambiente de projeto além de possuírem uma grande variedade de componentes *standard* é comum a disposição de uma grande variedade de projetos funcionais desenvolvidos pela empresa *vendor* ou por terceiros. Tais projetos são chamados de megafunções. Além de otimizados, alguns desses projetos permitem a alteração de alguns parâmetros e assim são chamados de projetos parametrizados. Exemplos desses projetos podem ser: decodificadores, contadores, projetos com implementações de funções matemáticas como dividir, somar, multiplicar valores de dois ou mais barramentos etc. E por serem desenvolvidos e otimizados pela empresa *vendor*, são altamente recomendados na utilização em projetos.

2. Paralelismo:

Uma grande diferença entre as arquiteturas de circuitos baseadas em microprocessadores e os componentes lógico programáveis, é a facilidade de implementação de um alto grau de paralelismo. Dividir tarefas para vários microprocessadores e gerenciar os resultados obtidos em cada um deles geralmente é uma tarefa que demanda excelentes softwares e muitos barramentos que devem ser construídos. Em um único componente FPGA pode ser inseridos várias etapas de um projeto, e possivelmente com frequências de trabalho diferentes. Os circuitos de um projeto em lógica programável são facilmente configurados e tamanhos de barramentos podem ser adaptados para cada especificação de projeto.

3. Relógio (*Clock*) elevado:

As famílias de componentes FPGAs apresentam como desempenho de *clock* de trabalho



dezenas de MHz. Componentes FPGAs apresentam uma alternativa para projetos onde o processamento necessita de grande rapidez.

4. Grande número de I/O (E/S):

Pode-se utilizar FPGAs com grande número de pinos de I/O para implementar sistemas com muitas entradas e saídas facilitando conexões a diversos dispositivos periféricos ao mesmo tempo. Este fato facilita a implementação de sistemas de controle ou sistemas de processamento paralelo.

5. Grande número de portas lógicas implementáveis:

A evolução tecnológica dos FPGAs permite hoje em dia uma implementação de dezenas de milhares de portas lógicas em um único componente. E apesar das ferramentas CAD que compilam e criam os circuitos implementados em FPGA não serem devidamente eficientes criando circuitos não minimizados, a grande quantidade de portas lógicas disponíveis para configuração permite assim a implementação de circuitos bastante complexos.

6. Fácil Roteamento:

Todo o roteamento dos pinos de entrada e saída podem ser configurados pelo projetista na fase de compilação.

7. Possibilidade de Sistemas-em-um-Chip:

Memórias, registradores, contadores, multiplexadores, demultiplexadores, máquinas de estados etc., podem ser implementados em um único componente FPGA. O que torna o FPGA um sistema completo em um CI.

8. Ferramentas de projeto:

Os ambientes de programação dos componentes FPGA apresentam grande variedade de ferramentas de minimização, alocação, compilação, simulação etc. Pode-se desenvolver sistemas



digitais baseados na idéia de componentes *standard* fazendo-se a alocação componente por componente. Mas, com o surgimento das linguagens de descrição de hardware abre-se a possibilidade da descrição de circuitos em níveis mais altos de abstração.

Em uma linguagem de descrição de hardware arquivos de texto descrevem circuitos através de comandos reservados. Circuitos podem ser modelados e remodelados criando ou alterando poucos parâmetros no código fonte do arquivo descrito. Este fato, facilita e agiliza o trabalho e rapidez de desenvolvimento de muitos projetos.

4.1.3 PROGRAMAÇÃO EM FPGA, AMBIENTE E FERRAMENTAS DE PROJETO

O ambiente de programação fornece para o projetista muitas ferramentas de síntese, otimização, compilação, de verificações (etapa de validação) e programação para que o projetista possa criar um projeto otimizado e eficiente. Entre as principais ferramentas de projeto do ambiente utilizado MAX + Plus II da Altera destacamos:

4.1.3.1 Editor Gráfico

Neste editor, cada projeto ou sistema pode ser entendido como um único componente através da representação por símbolo. O editor permite a fácil substituição de uma parte do sistema por causa desta representação. O uso de representação por símbolo permite também o uso de vários sub-sistemas – que podem ser confeccionados sob diferentes métodos ou estilos.



4.1.3.2 Editor de Texto

O editor de texto permite a edição de forma rápida dos textos de descrição de hardware. Escolhida a HDL, mostra em cores diferentes, comentários, comandos reservados ou quando existem erros sintáticos.

4.1.3.3 Compilador

O compilador é o programa inserido no ambiente de projeto que faz a compilação.

A compilação é uma série de procedimentos realizados pelo ambiente para analisar, minimizar, corrigir e apontar falhas no projeto desenvolvido.

O primeiro estágio de compilação faz a verificação sintática e semântica do arquivo de uma descrição de hardware. Caso haja algum erro, o processo de compilação é interrompido. O segundo passo da compilação é a criação de um arquivo que possa representar no nível de equações o circuito almejado, sem levar em consideração qual componente será configurado e tampouco o tipo de roteamento utilizado. O arquivo que representa a descrição das equações do circuito é chamado de Netlist. A partir deste, é avaliado se o componente escolhido e suas configurações permitem um roteamento implementável.

Deve-se ressaltar que a fase do processo de criação do arquivo Netlist é a mais crítica quanto a problemas na compilação, pois muitas vezes opções de síntese e de minimização ou escolha de família de componente podem inviabilizar a continuidade da síntese do projeto.

Para garantir uma rápida configuração pelo compilador, permite-se opção de escolha automática de componentes para que o ambiente de programação faça as escolhas mais adequadas. Devido às diferenças entre arquiteturas dos componentes lógicos programáveis, alguns projetos podem se adequar melhor ou pior em um tipo de família de componente.

Em comparação com a família CPLD (*MAX logic*), a família SRAM (*FLEX logic*) apresenta um maior número de portas lógicas utilizáveis e número de pinos de *I/O*. Já a família MAX permite uma maior configurabilidade em cada bloco lógico e, apresentando também um menor tempo de atraso de propagação entre seus pinos. Por essas características, quando é implementado um sistema que necessita de poucos blocos lógicos, porém precisa de alta frequência de trabalho, são recomendadas famílias MAX. - Por outro lado, quando o sistema necessita de um grande número de portas lógicas ou utilização de muitos barramentos, a escolha são os componentes FLEX.

As famílias de componentes podem ser também escolhidas devido ao custo de implementação. A cada ano novas tecnologias surgem permitindo o aumento da quantidade de células lógicas implementadas em um único componente e com uma melhor relação custo de implementação – benefício de um projeto.

O passo seguinte da compilação é a fase de simulação onde as características iniciais de projeto podem ser confrontadas com as temporizações dos sistemas implementáveis.

4.1.3.4 Simulador

A simulação é a fase de projeto que permite ao projetista uma análise do comportamento do circuito compilado. Deve-se na simulação escolher todas as variáveis relevantes e usar temporizações o mais perto das condições a que se propõe o sistema. Atrasos de propagação devem ser analisados e também os *spikes* (pulsos indesejáveis originados por atrasos de propagação). Quando os atrasos de propagação forem maiores que o permitido, deve-se escolher famílias de componentes lógicos mais rápidos. E quando a quantidade de spikes for grande, pode-se trocar de família de componente ou escolher outras opções de síntese de compilação (que



priorizem a velocidade em detrimento da área) e desta forma o sistema deve ser recompilado e reavaliado.

4.1.3.5 Editor de Planta Baixa

Em muitos projetos existe a necessidade de que o roteamento do componente FPGA seja configurado pelo projetista para uma melhor interface com outros dispositivos. Esse editor permite o gerenciamento do roteamento pelo projetista, permitindo a escolha de quais pinos do FPGA serão utilizados em I/O.

4.1.3.6 Programador (dos componentes FPGAs)

Esta ferramenta permite a implementação do circuito proposto em um componente FPGA. Desta forma, cria-se um protótipo do sistema que foi proposto e descrito nas entradas do ambiente.

Para utilizar os recursos dos componentes configuráveis foram desenvolvidas linguagens de descrição de hardware. E estas, definem em um alto nível de abstração como descrever um projeto. O próximo tópico descreve as principais características de algumas das linguagens.

4.2 LINGUAGENS DE DESCRIÇÃO DE HARDWARE

4.2.1 APLICAÇÃO

Desenvolver projetos em linguagem de descrição de hardware permite inúmeras vantagens. Pode-se dizer que o projetista não necessita possuir conhecimentos profundos em circuitos e sistemas digitais para desenvolver o seu projeto – devido ao alto nível de abstração de uma linguagem de descrição de hardware. Mas é necessário um conhecimento profundo de software e hardware de um sistema para a criação de sistemas otimizados.



O projeto em linguagem de descrição de hardware facilita modificações e correções de erros, podendo assim, o projetista ampliar facilmente o seu projeto com apenas algumas modificações em algumas linhas de comando.

Os usos de linguagens de descrição de hardware, estava ligado intrinsecamente com o uso de tecnologias digitais, ou seja, a finalidade de projetos descritos em hardware é desenvolver projetos com tecnologias digitais. Vários tipos de modelamento de circuitos podem ser utilizados na implementação de sistemas digitais. Atualmente existem também HDLs para circuitos analógicos.

Um fato que deve ser ressaltado no uso de linguagens de descrição de hardware é que uma descrição de um circuito realizada não esta relacionada a uma ou a outra tecnologia mas, descreve de maneira indistinta um projeto digital seja esta descrição de forma Comportamental (Instruções, redes de Petri, Linguagens algorítmicas, Equações Booleanas, Máquinas de Estados Finitos e Equações Diferenciais) ou Estrutural (Processadores, Memórias, Registradores, Unidades Funcionais, Portas Lógicas, Flip-Flops e Transístores).

4.2.2 DESCRIÇÃO

Uma linguagem de descrição de hardware é um texto de descrição de circuitos digitais. Deve conter ou elementos de sua estrutura física ou de seu comportamento. Ela deve ser estruturada e descrita de forma tal que o interpretador (compilador) gere de forma única o circuito almejado. Por não entrar em detalhes de qual o tipo de tecnologia empregada, a descrição de hardware permite lidar-se com o aumento da complexidade dos circuitos, pois permite que se trabalhe com uma descrição mais abstrata dos componentes e que os mesmos sejam agrupados hierarquicamente com bastante facilidade. Entre os principais itens de uma descrição pode-se citar:

- Bibliotecas (*Package*);
- Parâmetros e definições;
- Declaração de variáveis de entrada e de saída;
- Corpo;
- Declarações de variáveis internas;
- Estrutura do corpo (processos, tabelas verdade, máquinas de estado, lógica booleana etc.);
- Finalizações.

Existem diversas linguagens de descrição de hardware tais como VHDL, Verilog HDL, ABEL, AHDL e Randel C. A escolha da linguagem pode se dar pela familiaridade do projetista, entre outros fatores. O tempo de compilação pode ser bastante diferente no uso de diferentes descrições de hardware obtidas em diferentes linguagens. Durante a fase de compilação, a alocação de memória usada na geração dos arquivos de Netlist depende da linguagem de descrição utilizada. Em outras palavras, cada linguagem de descrição de hardware - por apresentarem modelamentos diferentes de um mesmo circuito - pode haver diferenças entre os resultados de compilações tais como tamanho do circuito sintetizado, alocação de memória na fase de compilação, tempo de compilação etc.

4.2.3 PORTABILIDADE:

Uma descrição de hardware não está ligada a tecnologia à ser implementada (família do componente) e sim ao projeto. Uma descrição comportamental ou estrutural pode assim, ser implementado em qualquer tecnologia alvo. Além disso, ambientes de descrição de hardware apresentam compatibilidade entre os arquivos de descrição – o que permite a troca de arquivos em diferentes ambientes de programação.



4.3 EXEMPLOS DE LINGUAGENS DE DESCRIÇÃO DE HARDWARE

4.3.1 A LINGUAGEM AHDL (ALTERA HARDWARE DESCRIPTION LANGUAGE)

A AHDL [25] é uma linguagem de alto nível, modular e completamente integrada no ambiente de programação utilizado no desenvolvimento deste trabalho (Max + Plus II). Idealizada pela empresa Altera Corp., apresenta facilidade de implementação de sistemas baseados em componentes das famílias Classic, MAX 3000, MAX 7000, Flash e FLEX.

Comandos permitem o rápido desenvolvimento de projetos baseados em contadores síncronos, máquinas de estado finito do modelo de Mealy e de Moore, funções lógicas, equações booleanas e funções parametrizáveis. Permite a instanciação de outros projetos ou componentes ou ainda projetos de Megafunções fornecidas pela ambiente de programação. Parâmetros podem ser definidos ou alterados sem a necessidade de reescrever todo arquivo de descrição. Uma grande habilidade da linguagem é o uso de tabelas estruturadas para descrever equações booleanas. Com estas, pode-se descrever rapidamente e de forma clara, decodificadores com barramentos de entrada e saída de tamanhos diversos.

Como a linguagem foi desenvolvida pela Altera Corp., os arquivos Netlist são mais fáceis de serem gerados pois existe uma maior compatibilidade entre a linguagem de descrição de hardware e a ferramenta de compilação.

4.3.2 A LINGUAGEM VHDL (VERY HIGH SPEED INTEGRATED CIRCUIT HARDWARE DESCRIPTION LANGUAGE)

A linguagem VHDL [26] é um dos padrões industriais em linguagens de descrição de hardware que descreve as entradas e saídas, comportamentos e funções dos circuitos. Esta linguagem é definida por dois padrões: IEEE Std 1076-1987 (Chamado “VHDL 1987”) e Std 1076-1993 (Chamado “VHDL 1993”).

Esses dois padrões (1987 e 1993) são completamente integrados no ambiente (MAX + Plus II). Assim, pode-se compilar e simular os arquivos de projeto VHDL e usá-los para programar componentes FPGA.

O VHDL é uma linguagem genérica para descrever sistemas digitais de hardware. Ela não faz alusão à cerca da tecnologia ou da metodologia usada para descrever os sistema. Ao invés disso, uma abstração dos sistemas digitais de hardware é usada como um dos principais fundamentos da linguagem. Essa abstração inclui comportamento, temporização e características estruturais dos sistemas digitais. Essas características integram uma linguagem singular que permite uma ampla faixa de opções para descrição e esquematização.

Na linguagem VHDL, bibliotecas declaradas definem o comportamento das operações utilizadas pelos seus comandos. Pode-se também desenvolver bibliotecas especificando quais tipos de comportamentos as variáveis podem obter em relação a bit de paridade, soma booleana, soma aritmética etc., e ainda, a utilização de bibliotecas facilita a implementação de sistemas digitais baseados em microprocessadores.

No ambiente de projeto atualmente, quando desenvolvido um sistema (ou sub-sistema) e criado um símbolo, este pode interagir com outros projetos desenvolvidos pela mesma ou por



outra linguagem de descrição, permitindo ao projetista uma escolha de qual parte de cada projeto será desenvolvida por cada linguagem.

O capítulo 5 mostra o desenvolvimento das ferramentas de síntese. Estas ferramentas têm como arquivo de saída as descrições em hardware dos decodificadores-função que representam funções algébricas.

Capítulo 5

DESENVOLVIMENTO DE FERRAMENTAS DE SÍNTESE AUTOMÁTICA

Métodos de síntese automatizada existem desde o início dos estudos sobre lógica booleana e datam das décadas de 50 e 60, mas seu impacto na prática de projeto de sistemas digitais não foi então significativo, tendo permanecido como trabalho de relevância puramente teórica até meados da década de 70.

A necessidade de manipular quantidades crescentes de informação durante o projeto de sistemas digitais exigia a consideração constante de novas ferramentas. Essas deveriam transcender as atividades de facilitar a captura e de exercitar descrições, passando a ser capazes de gerar novas descrições de forma automatizada e corretas por construção, baseado nos mesmos requisitos manipulados pelos projetistas. O estilo de projeto estabelecido por estas novas ferramentas é centrado não apenas no talento criativo do ser humano, mas também no fato deste

talento poder ser capturado por ferramentas computacionais e usado para guiar o processo de projeto. A ferramenta central deve possuir embutido um conjunto de modelos de síntese, e ser capaz de gerar uma descrição correta por construção, bem como uma avaliação do desempenho desta descrição. O laço de realimentação é fechado pelo projetista, que julga os resultados da síntese e aceita a descrição, ou a rejeita, escolhendo um novo modelo de síntese para ser usado, ou, em outras palavras, novos parâmetros para a síntese do circuito proposto.

As ferramentas de simulação podem ainda ser empregadas, visando fornecer mais detalhes sobre a implementação. Frequentemente, o sintetizador gera uma descrição a partir dos requisitos, e refina esta descrição através de técnicas de otimização. Somente a descrição resultante deste processo é fornecida ao projetista.

De fato, ferramentas computacionais apresentam-se como eficientes mecanismos de projeto e otimização de circuitos microeletrônicos. As técnicas e ferramentas de síntese, apresentam um ganho muito relevante no tempo de desenvolvimento de projetos além de automatizar as tarefas antes realizadas manualmente.

5.1 OBJETIVOS DO DESENVOLVIMENTO DE FERRAMENTAS DE SÍNTESE

Implementou-se, neste trabalho ferramentas de síntese que fazem o papel da descrição de hardware de decodificadores de forma automatizada e correta. Para isto, devem ser estudadas modelos de descrição de hardware de diferentes linguagens. Os decodificadores gerados apresentam o modelamento em termos de tabela da descrição de uma equação algébrica.

As ferramentas criam arquivos de descrição de hardware e tem como entrada os números de bits dos barramentos de entrada e saída do decodificador.

5.2 PROGRAMAÇÃO EM LINGUAGEM C

Para o desenvolvimento das ferramentas de síntese automática foi necessária a utilização de uma linguagem de programação que possua as seguintes características:

- Apresentar como resultado a geração de um arquivo executável de pequeno tamanho;
- Criar com facilidade arquivos de texto ASCII e permitir a mudança no nome da extensão do arquivo;
- Aceitar como entradas destes arquivos nome e dados de descrição da equação à ser implementada;
- Ocupar pouca memória alocável no microcomputador.

Escolheu-se a linguagem C para o desenvolvimento de ferramentas de síntese automática por seguir estes requisitos, e assim, permitir a criação de arquivos de descrição de hardware com extensões .TDF (*text design file* para AHDL) e .VHD (*VHDL design file* para VHDL).

5.3 FERRAMENTAS DE SÍNTESE

Neste capítulo são apresentadas as ferramentas de síntese automática desenvolvidas para a geração de decodificadores através da modelagem em linguagem de descrição de hardware em AHDL e VHDL.

O processo de geração de formas de ondas a partir de descrições de hardware será apresentado no capítulo 6. Os resultados destas descrições e suas implementações são apresentados no capítulo 7.



5.3.1 FERRAMENTA E2HDL

Utilizando-se de um programa em Linguagem C, foi desenvolvida uma ferramenta de síntese para projetos de sistemas digitais denominada E2HDL (Equação para Linguagem de Descrição de Hardware). Dada uma equação que seja limitada em amplitude em um intervalo $[a, b]$, pode-se implementar essa equação em um componente FPGA, usando assim, somente recursos digitais.

Com a especificação dos parâmetros da equação a ferramenta desenvolvida gera um modelo em linguagem de descrição de hardware, podendo ser a AHDL ou a VHDL, que implementa a equação desejada [15], [16] e [17].

5.3.1.1 Desenvolvimento da Ferramenta de Síntese

Esta ferramenta de síntese automática descreve uma forma de onda através de uma representação tabular. Gerando-se um texto, que descreve a cada passo (acréscimo do tempo), qual o valor de saída do sistema. Para gerar essa tabela, deve-se transformar o sinal $f(t)$ em uma função discretizada chamada de $X(n)$. Essa tabela possui como parâmetros:

n_i – barramento de entrada do decodificador;

n_o – barramento de saída do decodificador (ou resolução).

A representação de uma função $f(t)$, limitada em amplitude no intervalo $[a, b]$, necessita que, exista um $f(t_1) = A_{\min}$ e um $f(t_2) = A_{\max}$, os respectivos valores mínimos e máximos da função no intervalo $[a, b]$. Para que os recursos do sistema digital sejam bem aproveitados, deve-se utilizar uma transformação tal que o maior valor de $X(n)$ seja igual ao maior valor de escala do barramento de saída e que o valor mínimo para $X(n)$ seja zero, pois

nessa implementação os valores do barramento de saída são de 0 a $(2^{n_o} - 1)$, sendo n_o o número de bits do barramento de saída. Em (1), é mostrado o equacionamento da função $X(n)$.

$$X(n) = \frac{(2^{n_o} - 1)}{A_{\max} - A_{\min}} f\left(a + (b - a) \frac{n}{(2^{n_i} - 1)}\right) + \frac{(2^{n_o} - 1)}{A_{\max} - A_{\min}} \quad (1)$$

Sendo:

$n \Rightarrow$ número inteiro;

$n_i \Rightarrow$ o número de bits de entrada;

$n_o \Rightarrow$ o número de bits de saída;

$a, b \Rightarrow$ números arbitrários que definem o intervalo (período) da função.

Além disso, tem-se as seguintes restrições:

$$0 \leq n \leq (2^{n_i} - 1);$$

$$0 \leq X(n) \leq (2^{n_o} - 1).$$

Com os valores de $X(n)$ cria-se uma tabela para cada um dos valores de $0 \leq n \leq (2^{n_i} - 1)$. A tabela gerada apresenta uma descrição comportamental através de uma tabela verdade.

Acrescentando a esta descrição comportamental, os cabeçalhos, nome do arquivos etc., gera-se um arquivo de descrição de hardware que pode ser implementado em diferentes linguagens.

Nos próximos itens são mostrados um exemplo de descrição de hardware em diferentes linguagens e modelamentos. Em AHDL (Linguagem de Descrição de Hardware da Altera) para o seno com 4096 passos ($n_i = 12$ bits) e resolução de 8 bits ($n_o = 8$). E, em VHDL para a função seno com 256 passos ($n_i = 8$ bits) e resolução de 8 bits ($n_o = 8$). A variação do parâmetro n_i

nestes casos é meramente ilustrativa. Comparações entre linguagens, parâmetros dos decodificadores (n_i e n_o) e famílias de componentes utilizados são devidamente exploradas no capítulo 7.

5.3.1.2 Exemplo 1 : Função Seno implementada em AHDL

Os valores máximo e mínimo da função $f(t) = \text{sen}(2\pi \cdot t)$, onde $a = 0 \leq t \leq 2\pi = b$ são:

$A_{\max} = 1$ e $A_{\min} = -1$. Em (2), tem-se para $n_i = 12$ e $n_o = 8$ bits a expressão para $X(n)$:

$$X(n) = \frac{(255)}{2} \text{sen}\left(\frac{2\pi \cdot n}{4095}\right) + \frac{1}{2}(255) \quad (2)$$

Sendo:

$n \Rightarrow$ número inteiro;

Com as seguintes restrições:

$$0 \leq n \leq 4095;$$

$$0 \leq X(n) \leq 255.$$

A função $X(n)$ descreve o comportamento do decodificador-função que implementa a função $f(t) = \text{sen}(2\pi \cdot t)$. A Figura 21, apresenta partes do código do arquivo de descrição de hardware seno com $n_i = 12$ e $n_o = 8$ bits gerado pela ferramenta E2HDL.


```

SUBDESIGN          12_8
(
    i[11..0]        :INPUT; % Barramento de Entrada%
    oa[7..0]         :OUTPUT;% Barramento de Saída %
)
BEGIN
    TABLE
        %n %                % X(n) %
        i[11..0]            =>        oa[7..0] ;
        H"0"                =>        H"80" ;
        H"1"                =>        H"80" ;
        H"2"                =>        H"80" ;
        . . .
        H"FFD"              =>        H"7F" ;
        H"FFE"              =>        H"7F" ;
        H"FFF"              =>        H"7F" ;
    END TABLE;
END;

```

Figura 21: Arquivo AHDL de descrição da equação seno com $n_i = 12$ bits e $n_o = 8$ bits.

5.3.1.3 Exemplo 2: Função Seno implementada em VHDL

Os valores máximo e mínimo da função $f(t) = \sin(2\pi \cdot t)$, onde $a = 0 \leq t \leq 2\pi = b$ são:

$A_{\max} = 1$ e $A_{\min} = -1$. E em (3), tem-se para $n_i = 8$ e $n_o = 8$ bits, a expressão para $X(n)$:

$$X(n) = \frac{(255)}{2} \sin\left(\frac{2\pi \cdot n}{255}\right) + \frac{1}{2}(255) \quad (3)$$

Sendo:

$n \Rightarrow$ número inteiro;

E com as seguintes restrições:

$$0 \leq n \leq 4095;$$

$$0 \leq X(n) \leq 255.$$

A Figura 22, apresenta partes do código do arquivo de descrição de hardware modelado em VHDL para a função seno com $n_i = 8$ e $n_o = 8$ bits gerado pela ferramenta E2HDL.

```

ENTITY      seno8_8  IS
PORT (
  i  : IN INTEGER RANGE 0 TO 255; % entradas %
  o  : OUT INTEGER RANGE 0 TO 255 % saídas  %
);
END seno8_8;
ARCHITECTURE PRINCIPAL OF seno8_8 IS

BEGIN
  WITH i SELECT
o <= 128 WHEN 0,
    131 WHEN 1,
    134 WHEN 2,
    137 WHEN 3,
    .
    .
    .
    118 WHEN 253,
    121 WHEN 254,
    124 WHEN 255;
END PRINCIPAL ;

```

Figura 22: Arquivo VHDL de descrição da equação seno com $n_i = 8$ bits e $n_o = 8$ bits.

5.3.2 GERAÇÃO DE SENO EM TRÊS CANAIS

Existem alguns projetos de estimuladores neuromusculares que necessitam mais de um canal para que músculos diferentes possam ser estimulados ao mesmo tempo. Desta forma, existe a necessidade da implementação de decodificadores-função que apresentem mais de um canal de saída. Partindo da ferramenta de síntese E2HDL foi desenvolvida uma versão modificada da ferramenta que gera a descrição de hardware de decodificadores com três barramentos de sinais de saída [15]. As formas de onda implementadas foram: $f_2(t) = \text{sen}(2\pi \cdot t + \frac{4\pi}{3})$, $f_1(t) = \text{sen}(2\pi \cdot t + \frac{2\pi}{3})$ e $f_0(t) = \text{sen}(2\pi \cdot t)$. Se necessário, pode-se implementar outros tipos de formas de onda com outros ângulos de defasagem entre os sinais. No capítulo 7, é apresentado um exemplo do uso da ferramenta descrita. O circuito proposto necessita de 138 células (86% do componente EPM7160ELC84-10 da família MAX).

As linguagens de descrição de hardware, apresentam modelagens diferentes para implementar os mesmos circuitos. A linguagem AHDL apresenta, como mostrado na Figura 21, um modelo bem claro de uma tabela de valores para geração de funções booleanas. Em AHDL não é necessário completar todos os valores da tabela para que o circuito possa ser compilado. Já em VHDL todos os valores devem ser atribuídos para que haja a compilação da descrição – como mostrado na Figura 22. Nos modelos desenvolvidos em VHDL é necessário o uso de comandos que implicam em maior dificuldade (mais tempo e mais memória alocada) de criação do arquivo Netlist (processo de compilação) quando os parâmetros de $n_i = n_o = 8$ bits em relação à AHDL. Esse fato é verificado nas tabelas apresentadas no capítulo 7.

5.3.3 FERRAMENTA - ANÁLISE EM SÉRIE DE FOURIER

5.3.3.1 Engenharia Reversa

Algumas formas de onda, podem ser facilmente descritas por meio de equações e de modo particular, com os coeficientes de um polinômio. Outras porém, podem apresentar uma forma um tanto complexa para esse tipo de modelagem, ou ainda, pode-se ter apenas a forma visual (ou capturada por Osciloscópio Digital) para a possível reprodução. Deste fato, surgiu a necessidade de uma método de geração de formas de onda partindo da análise espectral. Esta análise pode-se ser facilmente obtida usando programas que fazem análises de formas de ondas capturadas por osciloscópios digitais.

Uma vez capturada a forma de onda que se quer reproduzir, pode-se efetuar uma análise, em que os resultados serão a entrada para uma ferramenta de síntese que, através da superposição dos coeficientes e dos respectivos pesos multiplicados por funções cosenos, podem reproduzir a forma sob análise conforme o tipo de refinamento necessário. Quanto maior for a quantidade dos

coeficientes harmônicos, mais fiel será a reconstituição de um sinal. Observa-se porém que, para uma reconstituição por este método, um sinal com coeficientes harmônicos de alta ordem significativos, a quantidade de bits de n_i pode aumentar para valores maiores que 12 (ou seja, 4096 amostras). O que dificultaria a implementação em FPGA pois necessitaria de uma quantidade muito grande de células para representar a forma de onda- como mostrado no capítulo 7.

5.3.3.2 Ferramenta F2HDL - Da Análise em Série de Fourier à HDL

Sinais analógicos ou digitais podem ser descritos por funções. Dessa forma a expansão em série é um método válido para a descrição desses tipos de sinais. A série de Fourier é, provavelmente, a mais utilizada dentre todas as formas de expansões em séries de funções ortogonais.

Foi desenvolvida uma ferramenta de síntese denominada F2HDL (Fourier para HDL), onde os coeficientes da Série de Fourier, obtidos de um sinal analógico, a partir de uma análise espectral, é usada como entrada da ferramenta. Desta forma pode-se reproduzir um sinal obtido implementado em FPGA [18].

5.3.3.2.1 Desenvolvimento da Ferramenta de Síntese

Dado uma função $f(t)$ que seja limitada em amplitude e em frequência, define-se um intervalo $[a, b]$, no qual, pode-se determinar os coeficientes da série de Fourier conforme (4):

$$f(t) \approx g(t) = A_1 \cos(1 \cdot 2\pi + \delta_1) + A_2 \cos(2 \cdot 2\pi + \delta_2) + A_3 \cos(3 \cdot 2\pi + \delta_3) + \dots \quad (4)$$

Sendo:

$A_1, A_2, A_3 \dots \Rightarrow$ coeficientes da série.

$\delta_1, \delta_2, \delta_3 \dots \Rightarrow$ os respectivos ângulos dos coeficientes A_1, A_2, A_3 .

Com a seguinte restrição:

$$-\pi \leq \delta_i \leq \pi, \text{ onde } i \geq 1.$$

A função $g(t)$ não leva em conta o coeficiente a_0 da série de Fourier porque na implementação de $H(n)$, assim como em $X(n)$, otimiza o uso de recursos digitais, desconsiderando-se assim o *offset* da função $f(t)$. Assim, a função $g(t)$ é uma aproximação de $f(t)$, desconsiderando o componente DC.

Se considerássemos a componente DC na forma de onda, para valores de amplitude do *offset* igual a forma de onda, haveria um aumento de 20% de células requeridas para o caso de uma função seno de $n_i = n_o = 8$ bits.

Para gerar uma descrição de hardware de uma função $g(t)$, deve-se criar uma função $H(n)$ que leva em conta os valores:

$A_{\max}$ e $A_{\min}$, os respectivos valores máximos e mínimos de $g(t)$ no intervalo $[a, b]$;

n_i – barramento de entrada do decodificador;

n_o – barramento de saída do decodificador (ou resolução).

A ferramenta de síntese por meio de laços, incrementando o valor de n , encontra os valores de $A_{\max}$ e $A_{\min}$. E desta forma, tem-se a Equação (5) como:

$$H(n) = \frac{(2^{n_o} - 1)}{A_{\max} - A_{\min}} \left\{ A_1 \left(\frac{\cos(1 \cdot 2\pi n + \delta_1)}{(2^{n_i} - 1)} \right) + A_2 \left(\frac{\cos(2 \cdot 2\pi n + \delta_2)}{(2^{n_i} - 1)} \right) + \dots \right\} + \frac{(2^{n_o} - 1)}{A_{\max} - A_{\min}} \quad (5)$$

Sendo:

$n \Rightarrow$ número inteiro;

$n_i \Rightarrow$ o número de bits de entrada;

$n_o \Rightarrow$ o número de bits de saída.

Com as seguintes restrições:

$$0 \leq n \leq (2^{n_i} - 1);$$

$$0 \leq H(n) \leq (2^{n_o} - 1).$$

5.3.3.2 Exemplo 3: Emprego da F2HDL

A partir dos valores de amplitude e ângulo de defasagem, pode-se sintetizar (ou representar) muitos tipos de sinais analógicos. A Tabela 1, apresenta valores usados para a implementação de um sinal analógico em FPGA usando linguagem de descrição de hardware. O sinal gerado tem sua simulação apresentada na Figura 23.

Os valores de n_i e n_o determinam a resolução e amostragem da forma de onda. A Figura 23 mostra uma forma de onda modelada com $n_i = 7$ bits e $n_o = 8$ bits. Assim, necessita-se de 128 passos para que a forma de onda complete um ciclo e a variação mínima de amplitude do sinal é 1/255 da amplitude máxima.

Tabela 1: Valores de Entrada para F2HDL

N. da Harmônica	Amplitude (p.u)	Defasagem (rad)
1	1	0
2	1	0.7

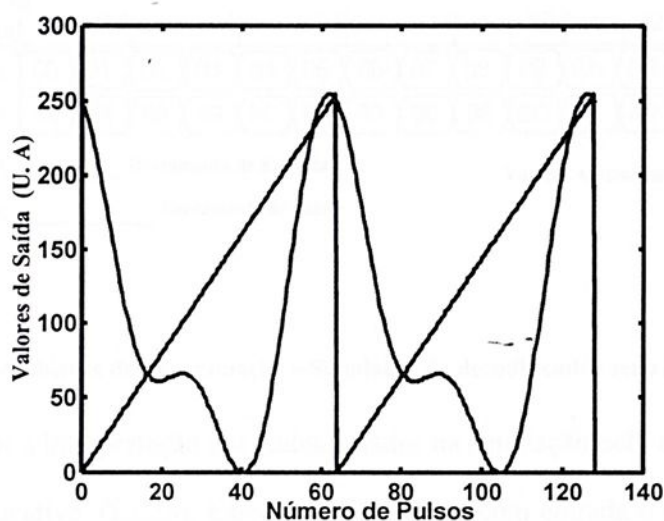


Figura 23: Implementação de Equações usando a F2HDL.

5.4 FERRAMENTA TBL2M

Quando uma descrição de hardware é compilada e simulada, o ambiente de projeto Max + Plus II gera vários arquivos de relatório para o projetista analisar. O arquivo .SCF (*stimulator channel file*) é um arquivo de simulação gráfica que contém a representação em forma de onda dos valores dos vetores de entrada da simulação. A interpretação desses vetores nem sempre é uma tarefa fácil. Quando se necessita de uma correlação entre a base de tempo e os diferentes barramentos de saída, pode-se encontrar dificuldades na interpretação de todos esses dados, pela própria disposição que o programa oferece. A Figura 24 mostra um exemplo de simulação de um decodificador seno com dois barramentos de dados $n_i = 8$ bits e $n_o = 8$ bits.



Figura 24: Ambiente de Programação – Simulação do decodificador seno ($n_i = 8$ e $n_o = 8$).

Para facilitar a interpretação dos dados gerados na simulação pelo ambiente de projeto, foi desenvolvido o aplicativo TBL2M. Esse aplicativo, tendo com entrada o arquivo .TBL (que é a representação em formato ASCII do .SCF), gera-se um arquivo no formato M (MatLab File) que permite, a interpretação gráfica de vários barramentos de saída ao mesmo tempo.

Para a criação do arquivo M, obtendo dados do arquivo TBL, deve-se criar em uma coluna os valores do tempo e nas próximas colunas os resultados dos respectivos barramentos de saída. Essas colunas formam uma matriz de dados onde a primeira coluna é a evolução do tempo. Assim, usando os comandos de representação gráfica de funções (*plot*, *stairs*, *stem* etc.) do MatLab, pode-se gerar um gráfico da evolução do tempo dos resultados das simulações.

Pode-se, utilizando este aplicativo, obter a visualização de uma série de simulações com diferentes parâmetros de barramentos de entrada e saída. O que facilita, em muito, a implementação ou correção de sistemas mais complexos.

Decodificadores com vários parâmetros (com $8 \geq n_i \geq 12$ e $8 \geq n_o \geq 12$) foram utilizados para verificar o funcionamento do aplicativo, que se mostrou eficiente em todos eles. No tópico 7.1 é mostrado em detalhes tabelas com os resultados de comparações.

Outra utilidade apresentada pelo aplicativo é a visualização das formas de ondas desenvolvidas nos programas E2HDL e F2HDL, sem a necessidade de uma implementação prática prévia. As formas de onda visualizadas são mostradas no tópico 7.2.

5.5 E2HDL E F2HDL NO DIAGRAMA DE GAJSKI

As ferramentas de síntese E2HDL e F2HDL geram, a partir de uma descrição da forma de onda, uma descrição em termos de equações booleanas. O que significa deslocar-se para um nível mais alto de abstração, ou seja mais distante do centro do Diagrama Y apresentado na Figura 18. O ambiente de programação gera, a partir de uma descrição booleana, o circuito em termos de portas lógicas. Faz-se assim, uma descrição estrutural do circuito. E por fim, a partir da descrição estrutural, o ambiente gera a descrição física (implementável) no componente FPGA (Domínio Físico). A Figura 25 apresenta no Diagrama Y do processo de síntese. A ferramenta de síntese leva a descrição do nível de Equação para Linguagem Algorítmica. O processo compilação (geração do arquivo Netlist) cria uma descrição no nível de Portas Lógicas e Flip-Flops. E fechando o processo, o configurador de FPGA leva ao nível de Planta Baixa no Domínio Físico.



4. Criação do Arquivo de Simulação TBL;
5. Execução do Programa TBL2M;
6. Visualização da forma de onda proposta no MatLab;
7. Validação do Circuito.

A Figura 26 representa o diagrama das etapas de validação dos circuitos decodificadores projetados. Uma ferramenta de síntese automática gera o arquivo de descrição de hardware (HDL). Utilizando a HDL como entrada do ambiente de programação, gera-se os arquivos de simulação (.SCF e .TBL), relatório (RTP) entre outros. Usando o arquivo .TBL como entrada do aplicativo TBL2M, gera-se um arquivo de entrada do MatLab que gera a visualização gráfica da forma de onda da equação sintetizada pela ferramenta de síntese.

Com a visualização da forma de onda, o projetista pode validar (aceitar) ou não os resultados e propor uma mudança nos parâmetros de entrada das ferramentas de síntese, mudar opções de minimização do circuito, etc., até que o circuito esteja de acordo com a descrição inicial.

O projeto de um estimulador pode conter mais de uma forma de onda. Devido a essa necessidade propõe-se um projeto que permite a seleção entre várias formas de onda. Assim, neste são inclusos vários decodificadores (cada decodificador fazendo o papel de uma forma de onda) que através de uma seleção gerando o sinal analógico correspondente.



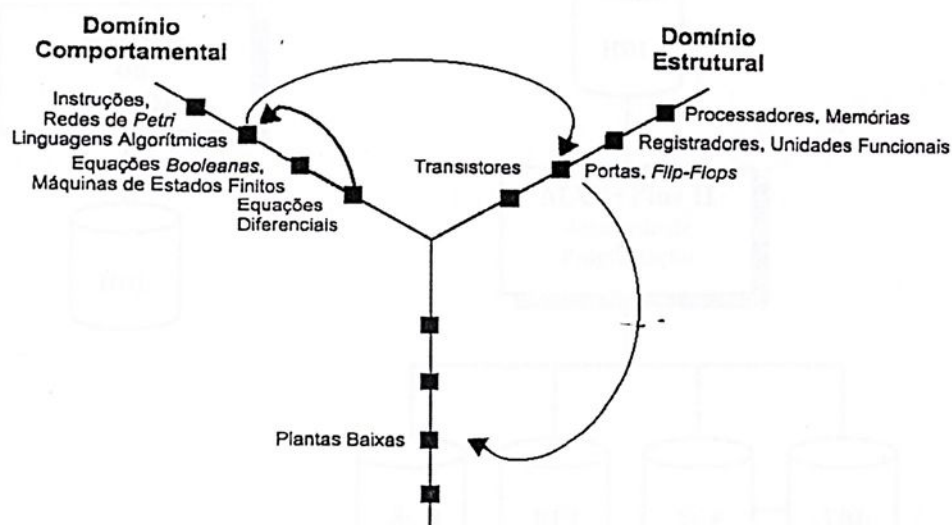


Figura 25: Diagrama Y para as etapas de síntese do projeto.

5.6 ETAPAS DE VALIDAÇÃO DOS DECODIFICADORES-FUNÇÃO

Uma etapa importante da geração de sinais analógicos usando recursos digitais é a validação dos decodificadores gerados a partir das ferramentas de síntese automática. A validação é a etapa de análise do projeto que garante que a descrição final do circuito está de acordo com a descrição inicial (que é a entrada das ferramentas de síntese). Esta análise tem os seguintes passos:

1. Descrição da forma de onda (análise espectral ou equação);
2. Utilização de uma das ferramentas de síntese para gerar o arquivo HDL (ou AHDL ou VHDL);
3. Compilação, Simulação;

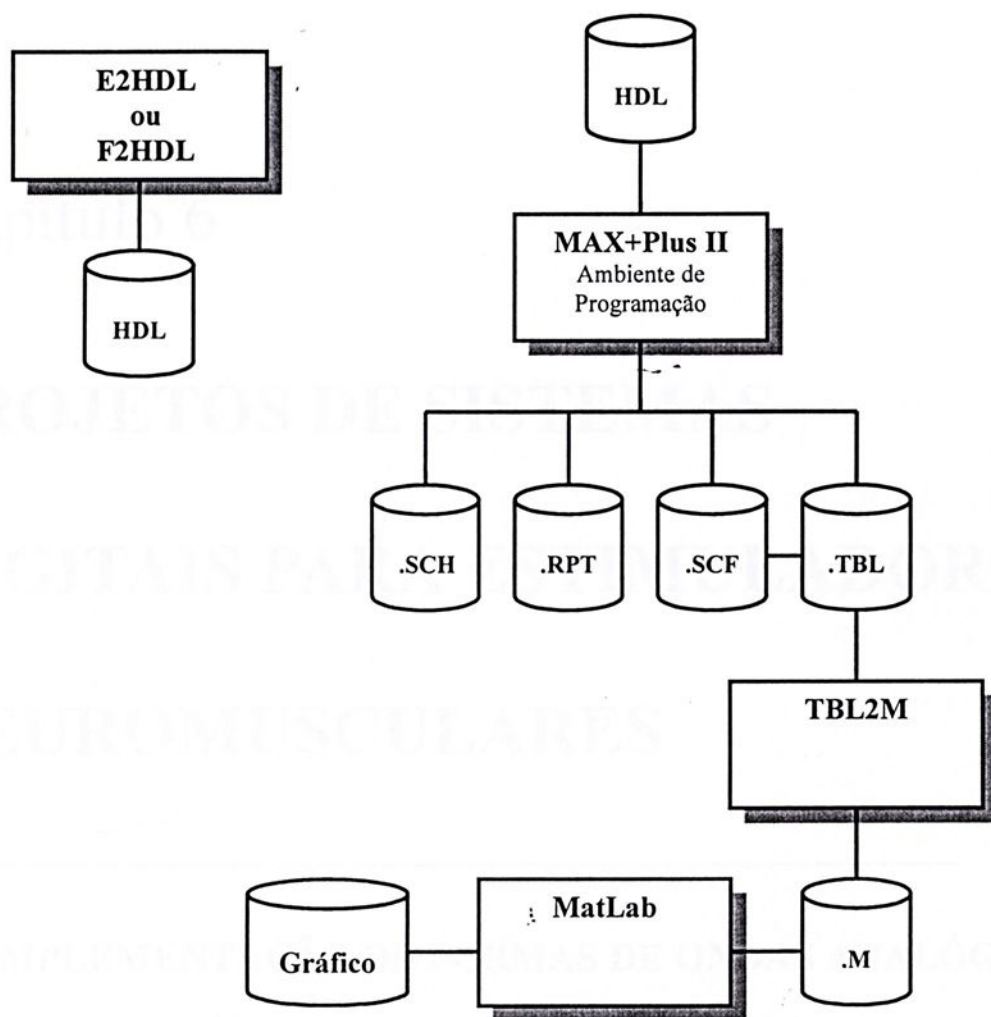


Figura 26: Esquema de Processo de Geração de Formas de Onda em FPGA.

No capítulo 6, é apresentado o processo de implementação de formas de onda analógicas utilizando recursos digitais. Também no capítulo 6, é mostrado o projeto de seleção de formas de onda.

Capítulo 6

PROJETOS DE SISTEMAS

DIGITAIS PARA ESTIMULADORES

NEUROMUSCULARES

6.1 IMPLEMENTAÇÃO DE FORMAS DE ONDAS ANALÓGICAS UTILIZANDO RECURSOS DIGITAIS

Para facilitar o desenvolvimento de projetos digitais, pode-se desenvolver em separado estágios puramente seqüenciais e puramente combinacionais. Na implementação de formas de onda utilizando recursos digitais, desenvolveu-se um contador (cujo comportamento é seqüencial e síncrono) onde a cada novo pulso na entrada do *clock*, acrescenta-se uma unidade no valor do barramento de saída. Desenvolveu-se também, decodificadores, que são circuitos puramente combinacionais, cujo valor do barramento de saída depende somente do valor atual no barramento de entrada. E, por fim, utiliza-se um conversor D/A, acoplado no barramento de saída

do decodificador-função, ponderando os valores do barramento de saída do decodificador, constituindo-se assim um sinal analógico.

6.1.1 O CONTADOR

O intervalo de valores possíveis de um contador, depende da quantidade de Flip-Flops que se está trabalhando, ou, de outra forma, do número de bits. Assim, um contador de $n_i = 8$ bits, por exemplo, apresenta em função da contagem, valores em sua saída entre 0 e 255, ou seja, de 0 a $2^{n_i} - 1$. Desta forma, a cada pulso de relógio a contagem aumenta em uma unidade percorrendo todos os valores de forma crescente. Se a saída do contador é um barramento de dados, esse barramento deve representar os valores atuais da contagem. O valor da frequência do barramento de saída do contador depende do número de bits e da frequência do relógio conforme (6):

$$f_{\text{signal}} = \frac{f_{\text{clock}}}{2^{n_i}} \quad (6)$$

Sendo:

n_i = número de bits do contador;

f_{clock} = frequência do relógio (entrada);

f_{signal} = frequência do barramento (saída).

Para diminuir a quantidade de *spikes* e atrasos de propagação de modo geral, recomenda-se o uso de contadores síncronos, onde todos os Flip-Flops do contador são acionados ao mesmo tempo pela entrada de *clock*.

6.1.2 DECODIFICADOR-FUNÇÃO

O Decodificador-Função é um circuito puramente combinacional. O valor da saída, depende exclusivamente do valor alocado na entrada. O seu comportamento é de uma tabela de valores, e assim, para cada valor apresentado em seu barramento de entrada existe um único valor

correspondente na saída. Os valores de n_i e n_o , definem os tamanhos dos barramentos de entrada e saída. Esses valores são especificados na etapa de síntese, quando são utilizadas as ferramentas automática de síntese desenvolvidas.

6.1.3 O CONVERSOR D/A

A especificação do conversor D/A é um fator importante. Ele deve ser escolhido de acordo com a quantidade de bits do decodificador-função implementado em FPGA (como mostrado no Apêndice A). Além disso, os parâmetros de frequência de trabalho devem satisfazer o circuito. Assim, a frequência de trabalho do conversor deve ser de pelo menos igual a f_{signal} da Equação (6).

Acoplando-se um contador ao decodificador-função e este último em um conversor D/A, tem-se a geração de uma função analógica.

A Figura 27, representa o diagrama de blocos da geração de uma função analógica $y(t)$ utilizando recursos digitais.

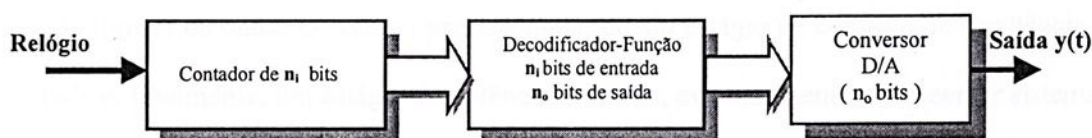


Figura 27: Geração de Sinal Analógico utilizando Decodificador.

A Figura 28 apresenta o exemplo intitulado *Projeto_I* em sua instanciação no ambiente gráfico MAX +Plus II da Altera.

O *Projeto_I* apresenta um exemplo de um projeto de geração de sinal analógico utilizando recursos digitais implementado no ambiente de projeto MAX + Plus II da Altera. Com

a utilização do decodificador-função *seno* no projeto, implementa-se a função

$$y(t) = \sin\left(2\pi \cdot \frac{t}{2^{n_i}}\right).$$

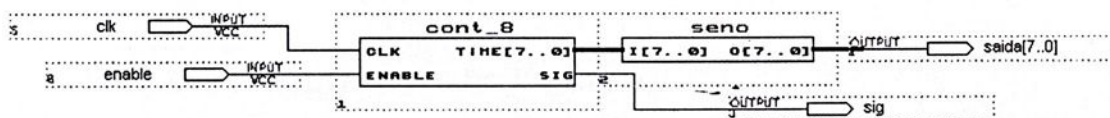


Figura 28: Projeto_1 – Geração de Sinal Analógico Usando FPGA.

Cada decodificador-função implementa uma forma de onda única. Para a obtenção de outras formas de onda é necessário somente a substituição do decodificador-função obtido no uso das ferramentas de síntese E2HDL ou F2HDL.

6.2 SELEÇÃO DE FORMAS DE ONDA

O núcleo de um estimulador elétrico neuromuscular é constituído pelos circuitos que geram as formas de onda. O sistema precisa ainda, de um estágio de controle de frequência e/ou amplitude e, finalmente, um estágio de potência. Pode-se, eventualmente, acrescentar sistemas de acoplamento (entre os estágios). É possível ainda acrescentar um conversor tensão-corrente se o sinal do estímulo a ser implementado tiver como característica a amplitude da forma de onda definida em corrente.

O estágio de potência do estimulador, assim como suas características serão definidas em trabalhos futuros.

6.2.1 OBJETIVOS DO PROJETO SELETOR DE FORMAS DE ONDA

O objetivo do projeto é a implementação de um sistema em FPGA que possa fazer uma opção entre os decodificadores gerados, e assim, gerar formas de onda com diferentes tipos de especificações.

6.2.2 DESENVOLVIMENTO DO PROJETO

O projeto do gerador de forma de onda para o estimulador, necessita a implementação de vários decodificadores-função. Assim, é necessário que haja no projeto um barramento que faça a seleção de qual decodificador estará gerando o sinal $y(t)$, que é a saída do estimulador.

Um sinal externo (ou entrada) necessário é o de *clock* que dita em que frequência será a resposta de $y(t)$. Barramentos devem ligar os decodificadores ao contador à chave seletora e finalmente ao conversor D/A.

A Figura 29 representa a arquitetura do projeto Seleção de Formas de Onda implementado.

Em um ambiente gráfico de projeto, componentes e sistemas são representados por símbolos facilitando o gerenciamento de projetos com vários sub-sistemas. Neste sentido, pode-se estanciar (neste caso significando representar) cada sub-projeto facilitando a conexão (ou ligação) entre eles. Para agrupar todos os projetos menores (decodificadores, contador e chave) em um único projeto, utiliza-se, desta forma, o ambiente gráfico do ambiente de projeto do MAX + Plus II. Assim, o projeto criado foi denominado *Proje_2*.

O uso de projetos que representam vários outros é chamado Arquitetura em Hierarquia de Projeto. E o projeto que representa todo o conjunto de projetos é chamado de topo de hierarquia.

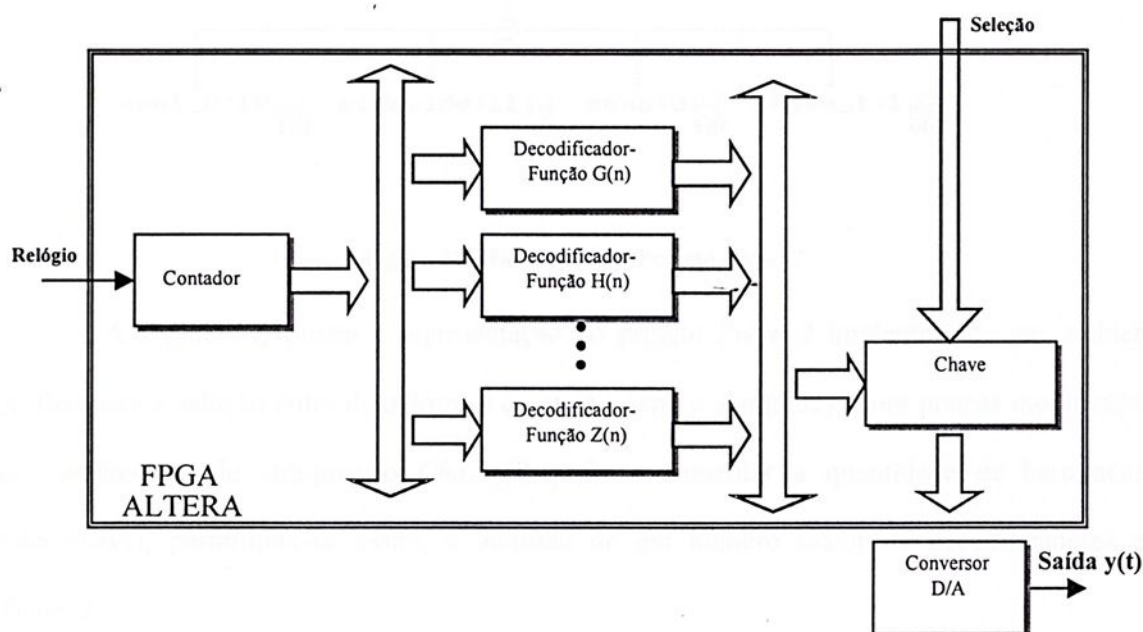


Figura 29: Arquitetura do Projeto Seleção de Formas de Onda.

A Figura 30 apresenta a Hierarquia de Projeto de *Proje_2* e seus respectivos arquivos. Os arquivos que dão origem ao projeto topo de hierarquia são mostrados na parte inferior (*cont_8*, *sigmoide*, *seno* e *chave_t*) e possuem extensões do tipo VHD (VHDL *design file*) ou TDF (AHDL, *text design file*), e ao lado esquerdo do nome do topo de hierarquia (*Proje_2*) são mostrados os arquivos gerados nas compilações e simulações do projeto como um todo. A extensão GDF (*graphic design file*) do arquivo *Proje_2* é devido o projeto ter sido gerado utilizando o ambiente gráfico fazendo a instanciação dos projetos menores.

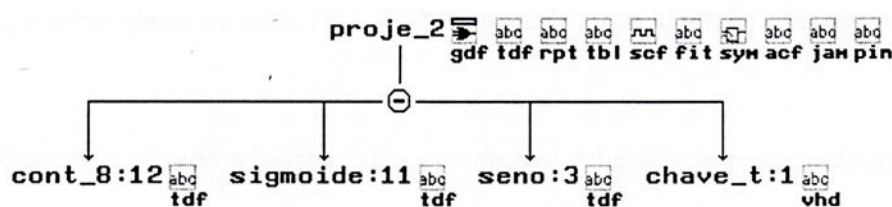


Figura 30: Hierarquia do Projeto *Proje_2*.

A Figura 31, ilustra a representação do projeto *Proje_2* implementado em ambiente gráfico para a seleção entre duas formas de onda (seno e sigmóide). Com poucas modificações nos parâmetros do sub-projeto *Chave_T*, pode-se aumentar a quantidade de barramentos selecionável, permitindo-se assim, a inclusão de um número maior de decodificadores em *Proje_2*.

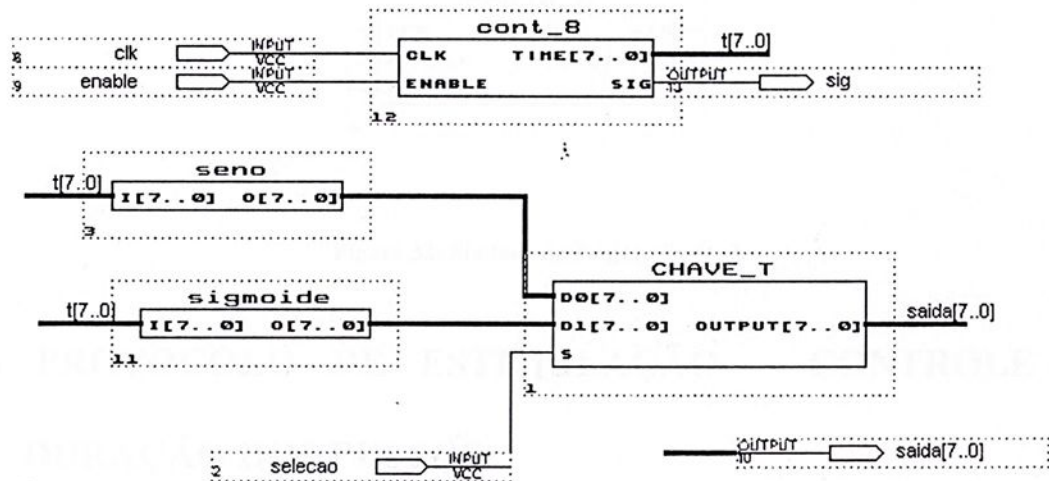


Figura 31: Projeto de seleção de formas de onda implementado.

A Figura 32 apresenta o símbolo gerado como representação de *Proje_2*. O *Proje_2* apresenta os sinais de entrada *clk* (relógio), *enable* (habilita) e *selecao* (selecionador de forma

de onda), e como sinais de saída **sig** (sinal de controle) e **saida[7..0]** (barramento do sinal de saída).

Quando é ativado o sinal **enable** a contagem é habilitada permitindo um aumento da contagem interna se o sinal de **clk** estiver pulsando. O valor aplicado em **selecao** determina qual decodificador está selecionado. Conforme aumenta-se o valor da contagem interna o barramento **saida[7..0]** apresenta o valor decodificado do decodificador.

Por fim, o barramento **saida[7..0]** é acoplado no conversor D/A fazendo a soma ponderada do barramento e convertendo o sinal à analógico. O sinal **sig** é um pulso a cada ciclo completo do contador. Este sinal poderá ser utilizado em trabalhos futuros com o objetivo de auxiliar no controle de pulsos a ser desenvolvido.

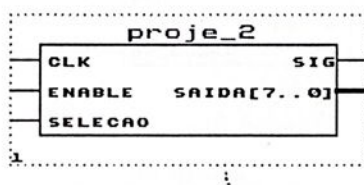


Figura 32: Símbolo do Projeto *Proje_2*.

6.3 PROTOCOLO DE ESTIMULAÇÃO – CONTROLE DE DURAÇÃO DOS PULSOS

6.3.1 OBJETIVOS DO PROJETO

Com o controle de frequência de saída de um sinal estimulador, pode-se controlar o nível de solicitação de força aplicada sobre grupos musculares, e desta forma, estimular mais adequadamente o paciente.

6.3.2 DESENVOLVIMENTO DO PROJETO

Pode-se, a partir de um circuito divisor de frequência, criar um sistema de controle de duração de um sinal. Partindo da idéia de um contador, este divisor, deve possuir como entrada sinal de relógio (clock), habilita (enable) e um sinal para definir a direção - aumentar ou diminuir - o valor de contagem; e ter como saída, um sinal com frequência múltipla em relação ao relógio. A Figura 33 mostra a instanciação do sistema desenvolvido no ambiente de projeto. O projeto é desenvolvido em dois estágios- counter e compara. O sinal do pino **pulsos** controla o sinal de *clock* de *Proje_2*.

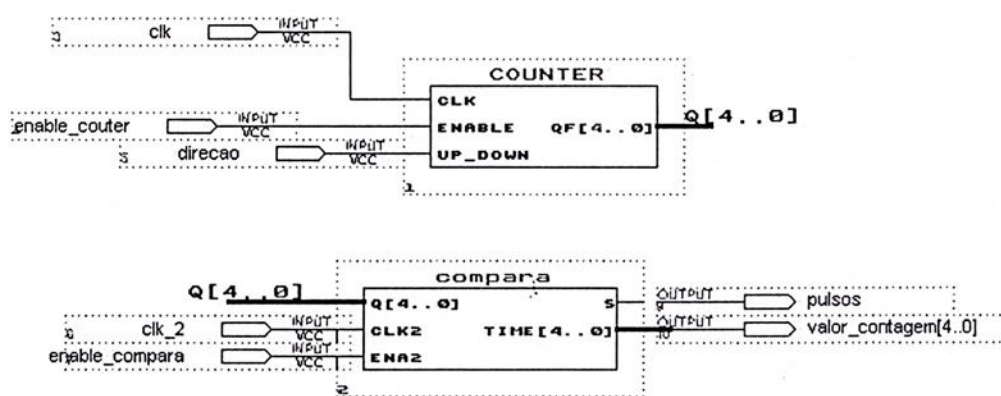


Figura 33: Projeto para Protocolo de Estimulação – Controle de Frequência.

Atuando em **up_down** controla-se a duração dos pulsos da forma de onda implementada. A geração dos protocolos será apresentada no capítulo 7.

O capítulo 7 mostra o estudo de comparação entre os decodificadores com diferentes parâmetros, comparação de resultados entre as linguagem de descrição de hardware e as comparações de componentes FPGA em relação a quantidade de células necessárias na implementação dos circuitos.

Também são apresentados as visualizações geradas a partir da ferramenta TBL2M e as implementações das respectivas formas de onda nos componentes FPGAs.

Capítulo 7

RESULTADOS

7.1 RESULTADOS DE COMPARAÇÕES DE PARÂMETROS DOS DECODIFICADORES, FAMÍLIAS DE COMPONENTES E LINGUAGENS DE DESCRIÇÃO DE CIRCUITOS

7.1.1 COMPARAÇÕES DE PARÂMETROS DE DECODIFICADORES E FAMÍLIAS DE COMPONENTES

Os decodificadores apresentam dois tipos de parâmetros de descrição: $n_i \Rightarrow$ quantidade de bits de entrada e $n_o \Rightarrow$ quantidade de bits de saída.

Para analisar a influência desses parâmetros no tamanho (número de células utilizadas) de um circuito, escolheu-se o decodificador-função seno para ser avaliado. Esta escolha se deve pelo fato dos decodificadores “seno” apresentarem poucos valores nulos no barramento de saída.

Na Tabela 2 , é apresentada a quantidade de células utilizadas aumentando o valor de n_o usando a família de componentes FPGA MAX 7000S, e na Tabela 3 para o FLEX 10K.

Tabela 2: Variação da quantidade de células utilizadas no decodificador seno com $n_i = 8$. Família MAX 7000.

Família MAX 7000		
Decodificador	Quantidade de Células utilizadas	Espaço ocupado no FPGA
seno ($n_i = 8, n_o = 8$)	42	33
seno ($n_i = 8, n_o = 9$)	45	35
seno ($n_i = 8, n_o = 10$)	53	41
seno ($n_i = 8, n_o = 11$)	65	50
seno ($n_i = 8, n_o = 12$)	78	61

Tabela 3: Variação da quantidade de células utilizadas no decodificador seno com $n_i = 8$ Família FLEX 10K20.

Família FLEX 10K20		
Decodificador	Quantidade de Células utilizadas	Espaço ocupado no FPGA (%)
seno ($n_i = 8, n_o = 8$)	258	22
seno ($n_i = 8, n_o = 9$)	305	26
seno ($n_i = 8, n_o = 10$)	299	26
seno ($n_i = 8, n_o = 11$)	342	30
seno ($n_i = 8, n_o = 12$)	334	29

A quantidade de células necessárias pode aumentar muito com a quantidade de bits de n_i . Assim, nas simulações variando este parâmetro utilizou-se os componentes da família FLEX 10K por apresentarem maior quantidade de portas lógicas programáveis. A Tabela 4 mostra a quantidade de células necessárias na implementação dos decodificadores.

Tabela 4: Variação da quantidade de células utilizadas no decodificador sem o com $n_o = 8$. Família FLEX 10K20.

Família FLEX 10K20 de componentes		
Decodificador	Quantidade de Células Utilizadas	Espaço ocupado no FPGA (%)
seno ($n_i = 8, n_o = 8$)	258	22
seno ($n_i = 9, n_o = 8$)	512	44
seno ($n_i = 10, n_o = 8$)	849	74
seno ($n_i = 11, n_o = 8$)	1121	97
seno ($n_i = 12, n_o = 8$)	2275	197

Confrontando-se os valores das Tabela 2 e Tabela 3, verifica-se que as células da família MAX mostram maior capacidade de implementação do que os componentes da família FLEX. Este fato se deve pela arquitetura dos componentes da família MAX demonstrarem melhor performance (configurabilidade) e também por cada célula possuir maior capacidade de implementação do que na tecnologia FLEX. Outro fato verificado, é que o aumento na resolução (n_o) do decodificador aumenta a necessidade de células de forma linear nos componentes MAX, porém pouco significativa nos componentes FLEX.

A cada aumento de uma unidade no parâmetro n_i significa que é necessário o dobro de linhas de código de comando para se descrever os decodificadores. Assim para $n_i = 8$ necessita-se de 256 linhas e para $n_i = 12$, de 4096. Este aumento exponencial de descrição influencia também a quantidade de células necessárias.

A Tabela 4, mostra como é o aumento da quantidade de células para implementar um circuito em relação ao parâmetro n_i . O aumento na quantidade de células está ligado com o aumento da complexidade das funções booleanas que representam o circuito do decodificador.

7.1.2 COMPARAÇÕES ENTRE LINGUAGENS DE DESCRIÇÃO DE HARDWARE

Compiladores de ambientes de projeto podem apresentar desempenho melhor em relação à algumas linguagens de descrição de hardware por estas permitirem um modelo de descrição mais “adaptado” ao processo de síntese. Desta forma, quando são confrontadas linguagens de descrição pretende-se analisar a capacidade do ambiente em otimizar recursos (tamanho do circuito, tempo de compilação, valor de pico de memória alocada etc.).

Comparações entre linguagens de descrição de hardware foram feitas confrontado parâmetros com o objetivo de verificar qual delas é mais viável dos pontos de vista de otimização de tempo e custos de implementação. Utilizou-se decodificadores gerados a partir de linguagens de descrição de hardware diferentes (AHDL e VHDL). O computador utilizado para as compilações foi um PC 233 MHz com 96 Mbytes de memória RAM.

A Tabela 5, apresenta a comparação de resultados de compilações de dois decodificadores em relação ao tempo, pico de memória alocada e quantidade de células necessárias para implementação nas linguagens AHDL e VHDL.



Tabela 5: Resultados de comparação de simulações entre linguagens de descrição de hardware – Família MAX 7000.

Família MAX 7000 de componentes				
Parâmetro	Linguagem AHDL		Linguagem VHDL	
	seno (n _i , n _o)		seno (n _i , n _o)	
	(n _i = 8, n _o = 8)	(n _i = 9, n _o = 8)	(n _i = 8, n _o = 8)	(n _i = 9, n _o = 8)
Decodificador				
Pico de memória (kbytes)	8,619	15,287	24,571	52,70
Número de células usadas	33	193	34	66
Tempo de compilação (min)	0.9	3.7	16.6	99.2

Nota-se na Tabela 5 que, tanto em quantidade de memória e tempo de compilação, a linguagem AHDL demonstrou um melhor desempenho. E quanto ao tamanho dos circuitos, para o decodificador seno (n_i = 8, n_o = 8), as linguagens obtiveram praticamente o mesmo desempenho. O decodificador seno (n_i = 9, n_o = 8) apresentou uma maior dificuldade de compilação em relação ao seno (n_i = 8, n_o = 8). Os tempos de compilação e a quantidade de memória alocada foram bem maiores. Este fato se deve possivelmente à dificuldade do compilador fazer a extração do arquivo de síntese lógica para o componente da família MAX, principalmente na linguagem VHDL.

A Tabela 6, apresenta os resultados de compilação de seno (n_i = 9, n_o = 8) para família FLEX, da mesma forma que na Tabela 5 foi para a família MAX. Comparando-se as tabelas,

verifica-se que no componente da família FLEX, diminuiu-se o tempo para gerar o decodificador seno ($n_i = 9, n_o = 8$) em ambas as linguagens. A quantidade de memória alocada em VHDL diminui bastante e em AHDL a quantidade de memória alocada não variou muito.

Tabela 6: Resultados de comparação de simulações entre linguagens de descrição de hardware – Família FLEX 10K.

Família FLEX 10K de componentes		
Parâmetro	Linguagem AHDL	Linguagem VHDL
Decodificador	seno ($n_i = 9, n_o = 8$)	seno ($n_i = 9, n_o = 8$)
Pico de memória (kbyte)	19,308	26,586
Número de células usadas	512	1024
Tempo de compilação (min)	2,8	3,3

Comparando-se as linguagens AHDL e VHDL supõe-se, devido à observação dos resultados, que para circuitos de decodificadores pequenos, os modelamentos em AHDL são mais eficientes tanto em compilação quanto na quantidade de células necessárias para implementação. Todos os circuitos modelados em AHDL obtiveram menor tempo de compilação em relação aos modelos em VHDL. Mas para circuitos muito complexos, a linguagem VHDL apresenta circuitos com menor quantidade de células conforme é mostrado na **Erro! A origem da referência não foi encontrada..** Apesar de apresentar circuitos com menor quantidade de células é bastante demorada, do ponto de vista de tempo de compilação, tendo em vista que o decodificador seno ($n_i = 12, n_o = 8$) demora 991 minutos para compilar.

Tabela 7: Comparação entre linguagens de descrição: Quantidade de Células - Família FLEX 10K.

Família FLEX 10K20 de componentes				
Decodificador	AHDL	Espaço (%)	VHDL	Espaço (%)
seno ($n_i = 8, n_o = 8$)	258	22	580	50
seno ($n_i = 9, n_o = 8$)	512	44	1024	89
seno ($n_i = 10, n_o = 8$)	849	74	1213	105
seno ($n_i = 11, n_o = 8$)	1121	97	1482	129
seno ($n_i = 12, n_o = 8$)	2275	197	1639	147

7.2 VISUALIZAÇÕES E IMPLEMENTAÇÕES DAS FORMAS DE ONDA GERADAS A PARTIR DOS DECODIFICADORES

As ferramentas automáticas de síntese introduzidas no capítulo 5, têm o objetivo de criar sistemas de geração de formas de ondas utilizando recursos digitais conforme descrito no capítulo 6. Os arquivos gerados pelas ferramentas automáticas de síntese foram compilados e simulados.

Utilizando a ferramenta TBL2M (apresentado no capítulo 5), gerou-se uma visualização das formas de onda propostas a partir dos resultados das simulações do ambiente de projeto MAX + Plus II, e também, foram implementados no EPM7128SLC84-7 (componente da família MAX da Altera), existente no módulo do kit didático da Altera.

A captura do sinal implementado foi permitida pelo uso de osciloscópio digital (Tektronix) e do programa WaveStar. Foram realizadas várias medidas para diferentes

freqüências de entrada, e verificou-se um bom funcionamento para valores com até 2 MHz de freqüência (*clock* de entrada e não da forma de onda).

Todos os arquivos de forma de onda foram implementados com a configuração $n_i = 8$ bits (relativo a quantidade de pulsos necessários para completar um ciclo) e $n_o = 8$ bits (resolução da forma de onda).

As formas de onda implementadas nos componentes da família MAX foram Senóide, Gaussiana , Sigmóide e Farádica. Também foram implementadas protocolos de estimulações (seqüências de pulsos) e seleções de formas de onda. A forma de onda Senóide-Trifásica foram analisadas somente pelas visualizações de suas simulações.

7.2.1 FUNÇÃO SENO

A Figura 34, mostra a visualização do arquivo de simulação do decodificador seno ($n_i = 8$, $n_o = 8$) com o uso da ferramenta TBL2M. Na Figura 34, nota-se que são necessários $2^{n_i} = 256$ pulsos (de 0 até 255) para que a função complete um ciclo. O valor inicial da função é 128 unidades. Esse valor corresponde ao *offset* necessário para que a função implementada tenha como valor máximo $(2^{n_o} - 1) = 255$ e valor mínimo igual zero. Também é mostrado na figura, uma rampa que representa a evolução do tempo.

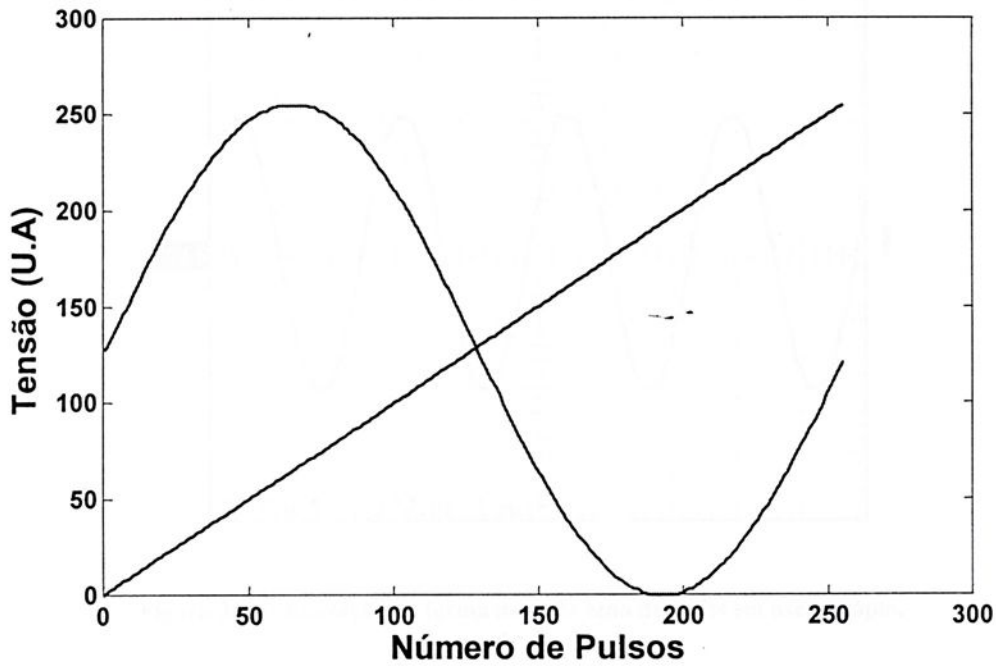


Figura 34: Visualização no MatLab da simulação da função seno ($n_i = 8$ bits, $n_o = 8$ bits).

A Figura 35, apresenta a implementação da forma de onda seno (com $n_i = 8$ bits e $n_o = 8$ bits). Nota-se que não existe offset devido a escolha de um conversor D/A que utiliza o princípio de escala completa.

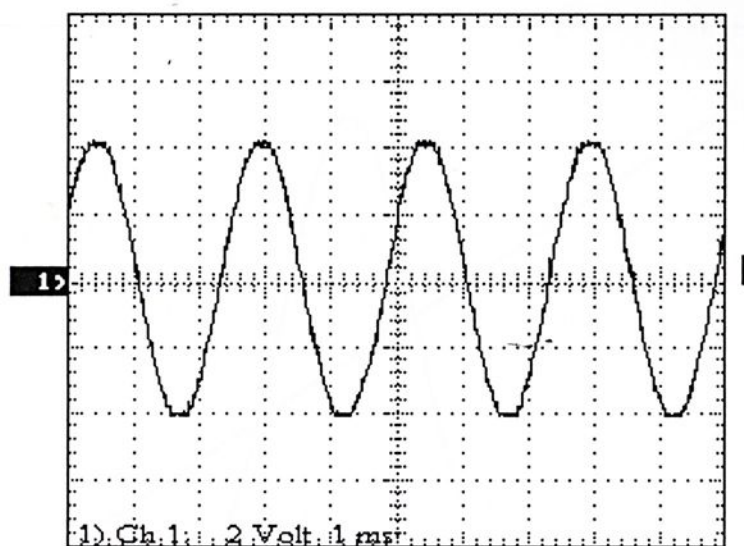


Figura 35: Visualização da forma de onda seno de 8 bits em osciloscópio.

7.2.2 FUNÇÃO GAUSSIANA

A Figura 36, apresenta a implementação da forma de onda gaussiana ($n_i = 8$ bits, $n_o = 8$ bits). A função gaussiana possui alguns parâmetros tais como abertura e descolamento, e por este motivo, alguns valores de parâmetros da equação foram definidos na etapa de descrição inicial.

A Figura 37, mostra a visualização da forma de onda gaussiana ($n_i = 8$ bits, $n_o = 8$ bits) em osciloscópio.

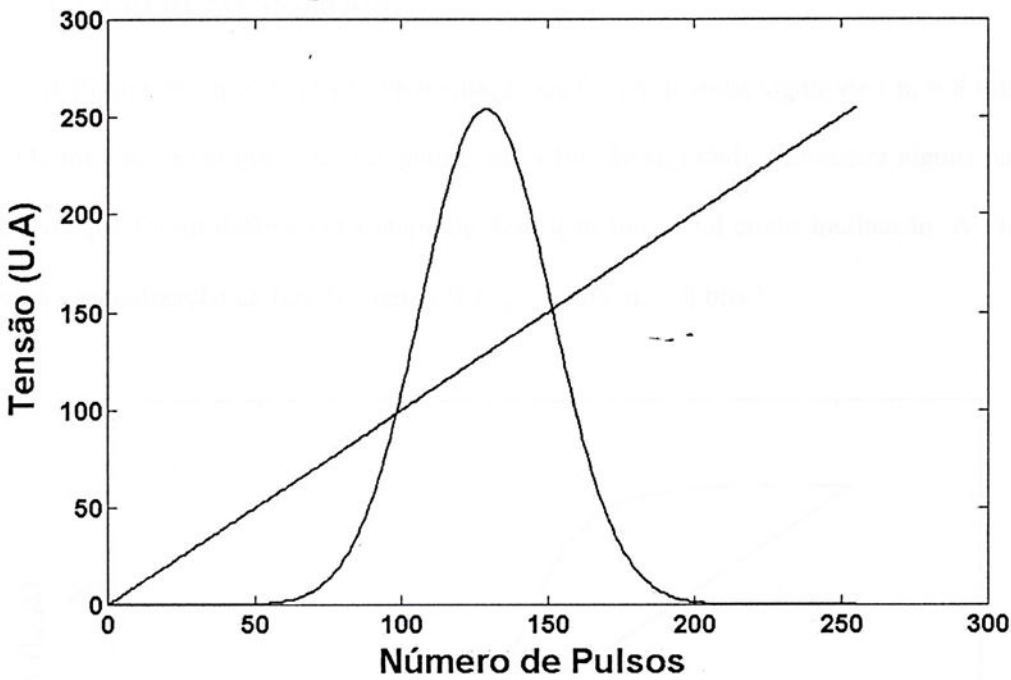


Figura 36: Visualização no MatLab da simulação da função gaus ($n_i = 8$ bits, $n_o = 8$ bits).

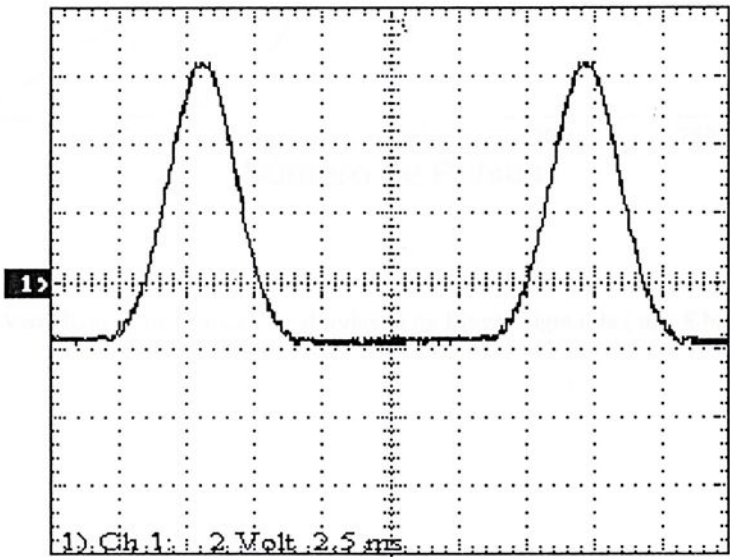


Figura 37: Visualização da forma de onda gaussiana de 8 bits em osciloscópio.

7.2.3 FUNÇÃO SIGMÓIDE

A Figura 38, apresenta a implementação da forma de onda sigmóide ($n_i = 8$ bits, $n_o = 8$ bits). Do mesmo modo que a função gaussiana, a função sigmóide demonstra alguns valores de parâmetros que foram definidos na etapa de descrição inicial tal como inclinação. A Figura 39, apresenta a visualização da função sigmóide ($n_i = 8$ bits, $n_o = 8$ bits).

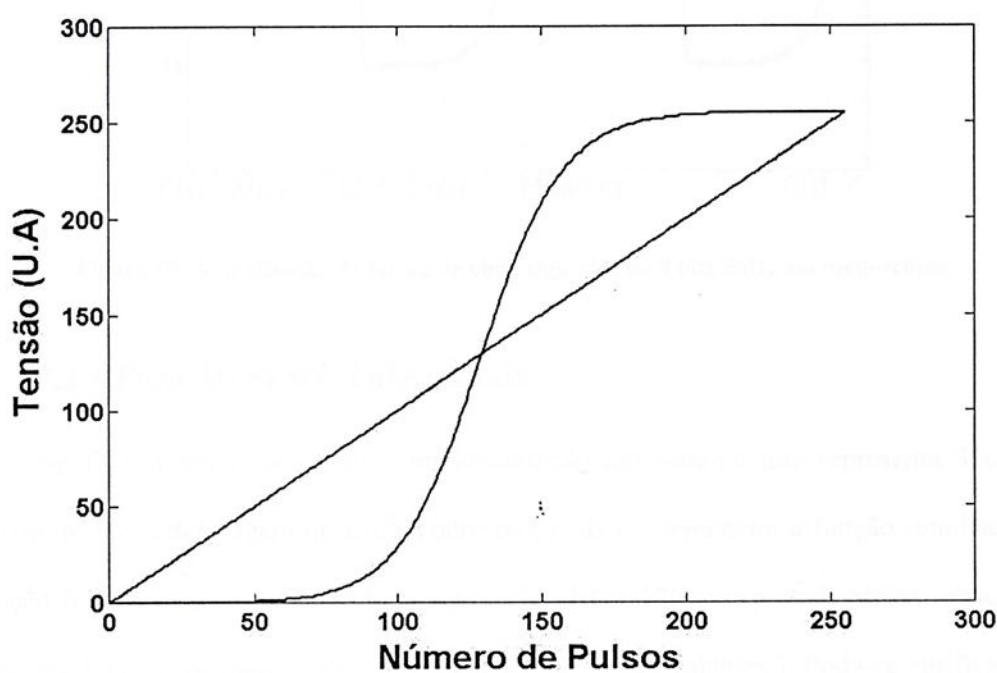


Figura 38: Visualização no MatLab da simulação da função sigmoide ($n_i = 8$ bits, $n_o = 8$ bits).

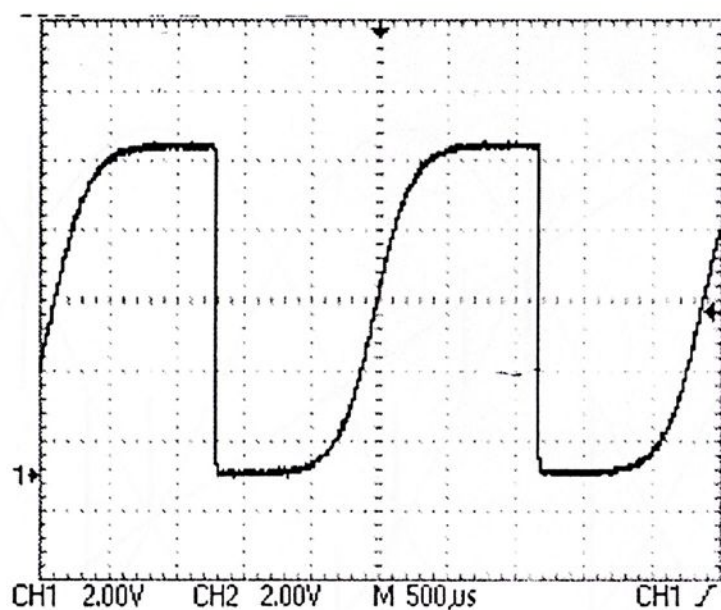


Figura 39: Visualização da forma de onda sigmóide de 8 bits 56Hz em osciloscópio.

7.2.4 FUNÇÃO SENO: TRÊS CANAIS

Na Figura 40, é mostrado a implementação um sistema que representa 3 canais de estimuladores com defasagem de 120° . Todos os canais implementam a função senoidal [17]. A simulação foi feita com a escolha do componente FLEX 10K por possuir maior capacidade de implementação (maior número de componentes lógicos equivalentes). Pode-se verificar porém, que na Figura 40, atrasos de propagação provocar *spikes*. Esse tipo de problema pode ser diminuído se a família de componentes par implementação da forma de onda demonstrar maior capacidade de implementação (utilizando opções de síntese do projeto que priorizem a frequência em detrimento da área ocupada), ou menores atrasos de propagação entre as portas lógicas.

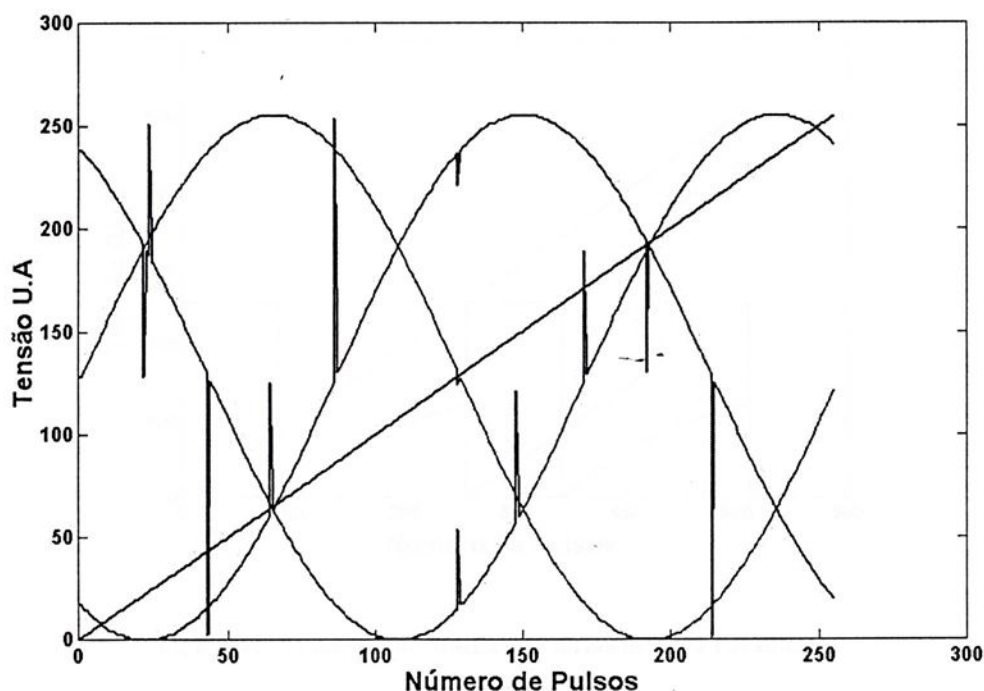


Figura 40: Simulação no MatLab da Geração Seno em três canais.

7.2.5 FUNÇÃO FARÁDICA

Outra forma de onda compilada e simulada foi a função Farádica. Este, é um tipo de estímulo muito comumente encontrado nos eletroestimuladores comerciais [1]. Suas características principais são de um pulso com forte ascendência no ciclo positivo e uma ascendência exponencial no ciclo negativo. Outro ponto importante é o equilíbrio dos ciclos, assim, a integral de área da função deve ser nula. A Figura 41 e a Figura 42 apresentam duas formas de onda Farádicas simuladas com diferentes parâmetros de descrição da etapa de especificação.

A Figura 41 apresenta a forma de onda Farádica 1 com ($n_i = 8$ bits e $n_o = 8$ bits). Na Figura 42 é mostrada a forma de onda Farádica 2 com ($n_i = 8$ bits e $n_o = 8$ bits).

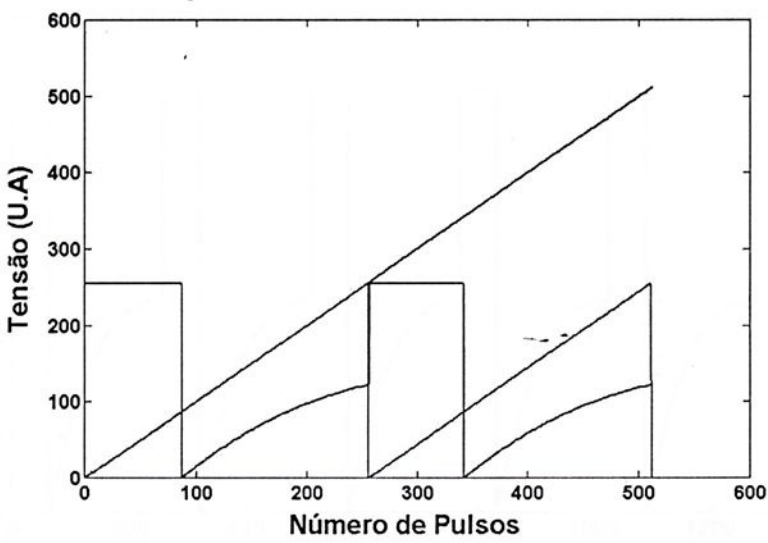


Figura 41: Simulação no MatLab da forma de onda Farádica 1

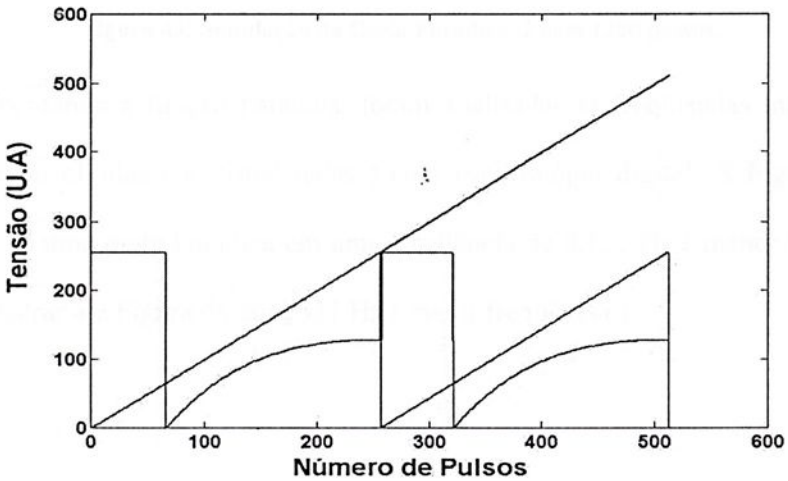


Figura 42: Simulação no MatLab da forma de onda Farádica 2.

A Figura 43, apresenta a Simulação da Onda Farádica 2 com o maior intervalo de tempo (maior quantidade de pulsos).

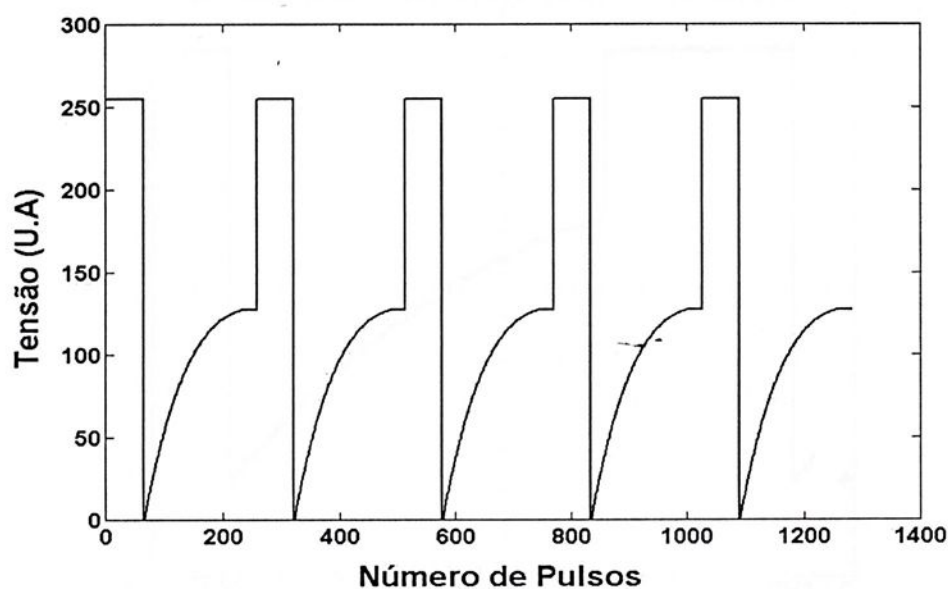


Figura 43: Simulação da Onda Farádica_2 com 1280 passos.

Implementando a função Farádica, foram analisadas as frequências máxima e mínima possíveis de serem obtidas (e visualizadas) com osciloscópio digital. A Figura 44 mostra a implementação de uma onda Farádica em uma frequência de 0.125 Hz (menor frequência) que foi possível amostrar a e Figura 45 em 2637 Hz (maior frequência).

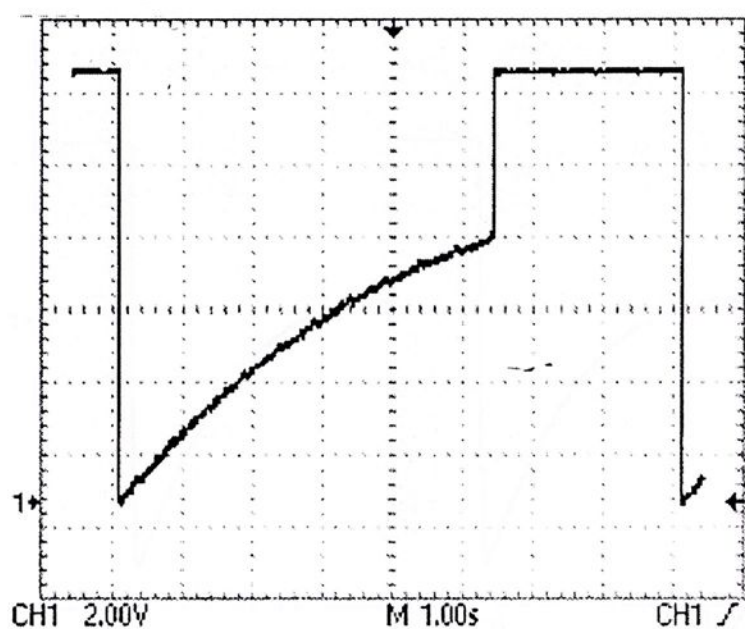


Figura 44: Visualização da forma de onda Farádica 1 de 8 bits em 0.125 Hz

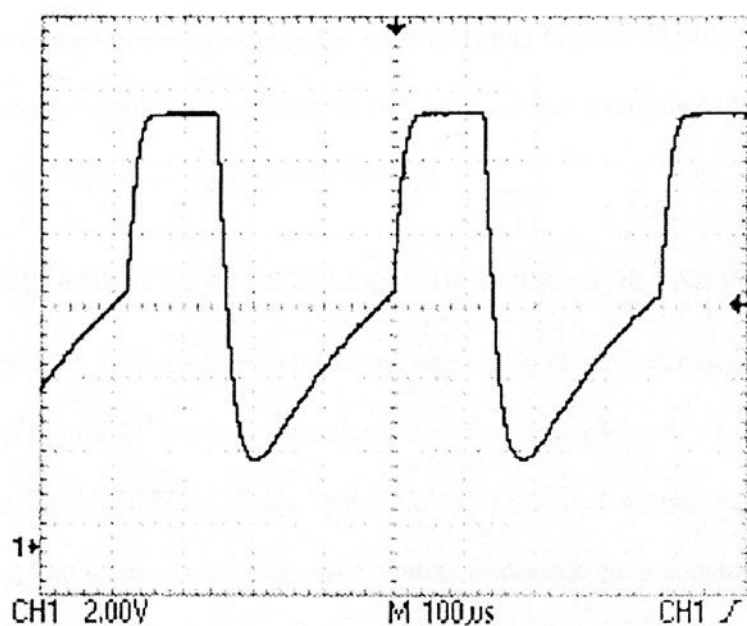


Figura 45 : Visualização da forma de onda Farádica de 8 bits em 2637 Hz.

Por fim, a Figura 46 mostra uma forma de onda em uma frequência típica de estimulador neuromuscular.

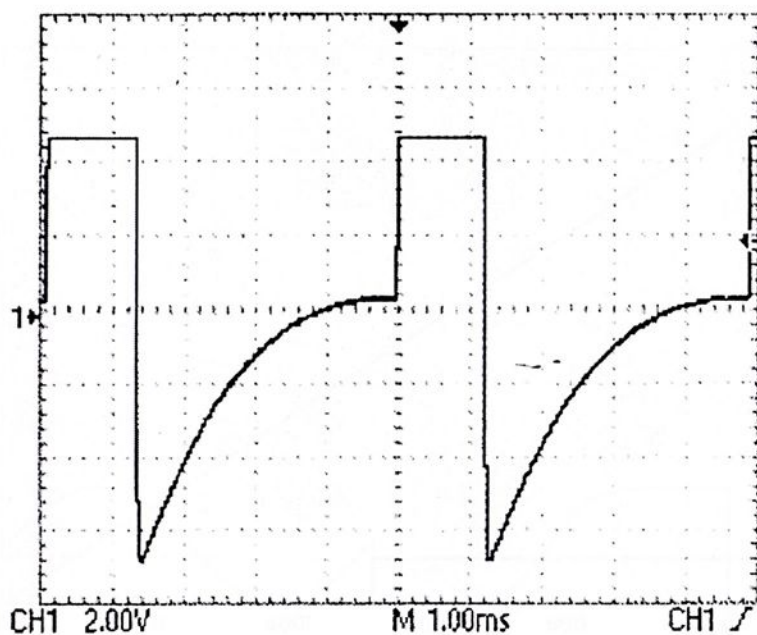


Figura 46 : Visualização da forma de onda Farádica 1 de 8 bits em 204,6 Hz.

Supõe-se que as pequenas oscilações mostradas nas Figuras 38, 40, 42, 47, 48 e 49 são devidas aos atrasos provocados pelo conversor D/A uma vez que a família MAX de componentes FPGA trabalha com frequências superiores a 50 MHz.

7.2.6 IMPLEMENTAÇÃO DE SELEÇÃO DE FORMAS DE ONDAS

As Figura 29 e Figura 31 apresentado no capítulo 6 fazem referência a uma seleção de formas de onda. A Figura 47, introduz o resultado de uma simulação com a escolha de formas de onda entre seno e sigmóide. Nota-se que, quando o valor do sinal seleção vai de zero para um, acontece a mudança da forma de onda de saída. Outras formas de onda podem ser inseridas neste processo, fazendo, assim, uma escolha mais adequada de qual tipo de sinal estimulador estará sendo aplicado ao paciente.

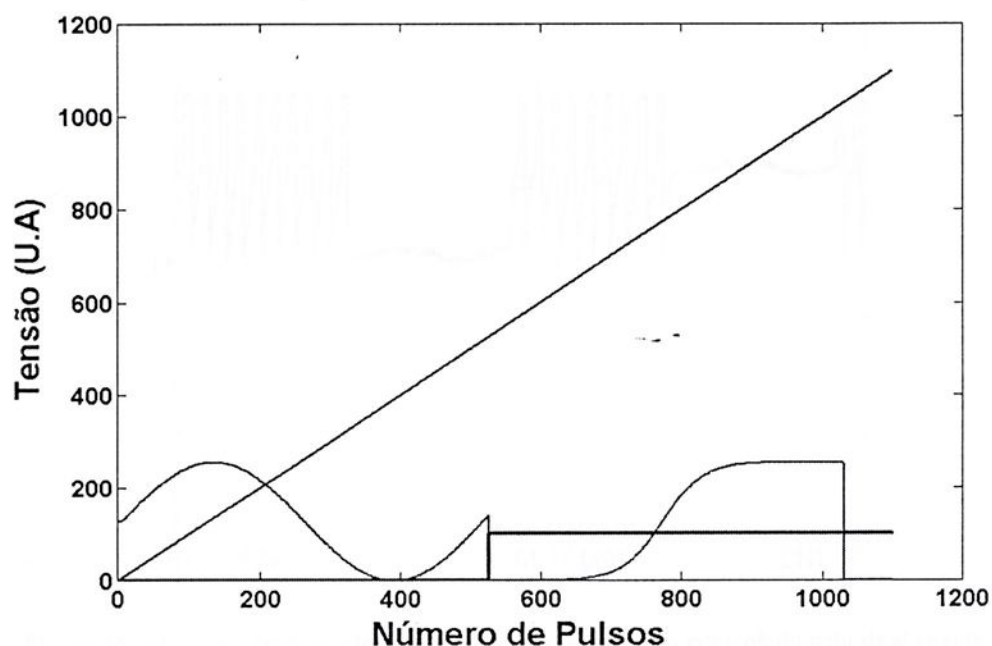


Figura 47: Simulação de Escolha de Formas de Ond

Além de permitir a seleção entre dois (ou mais) tipos de forma de onda, pode-se fazer a escolha entre uma forma de onda e um valor nulo representando assim um intervalo de tempo sem estímulo aplicado. Ou ainda de outra forma, permite-se controlar o pino **enable** (habilita contagem) para que haja um controle de tempos ativo e inativo de contagem.

A Figura 48 demonstra uma das formas de controlar o período ativo da onda Farádica utilizando-se de um sinal de **enable** cerca de 2560 vezes mais lento que o sinal de *clock*. E a Figura 49 mostra um protocolo de estimulação usando o controle **selecao** com 1280 vezes mais lento que o sinal de *clock*.

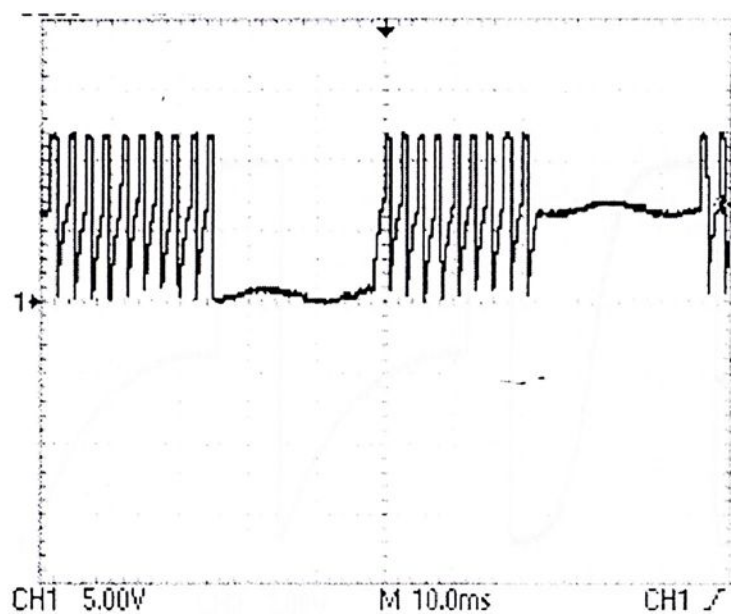


Figura 48: Visualização do protocolo de estimulação Farádica controlado pelo sinal enable.

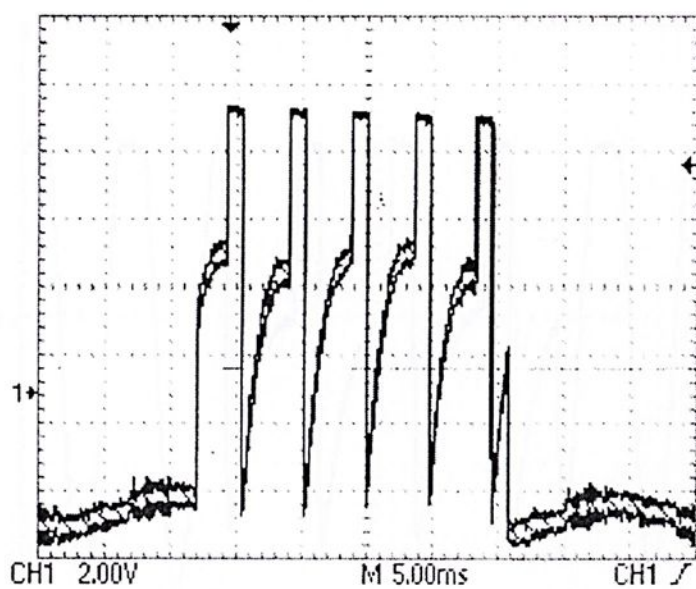


Figura 49: Visualização do protocolo de estimulação Farádica controlado pelo sinal selecao.

Utilizando uma seleção de formas de onda diferentes, pode-se obter a Figura 50 (Farádica-Senoidal) e a Figura 51.

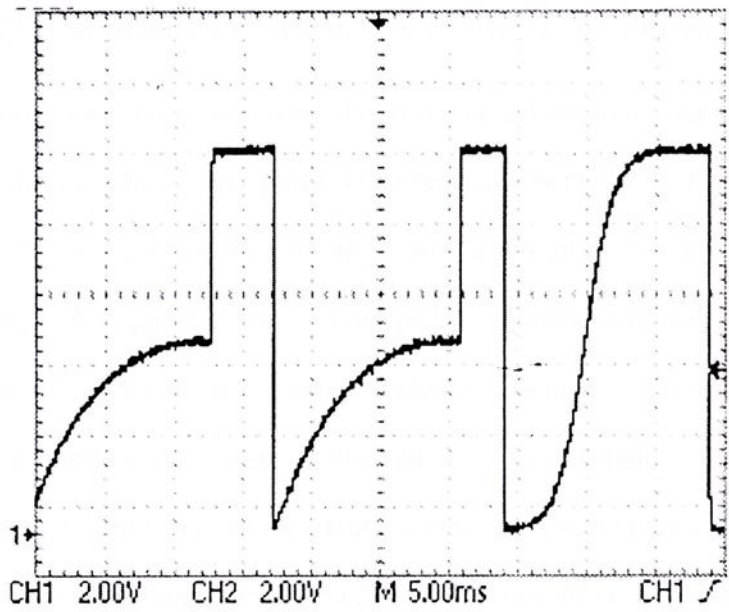


Figura 50: Visualização da seleção de formas de onda Farádica-Sigmóide em 8 bits (56 Hz).

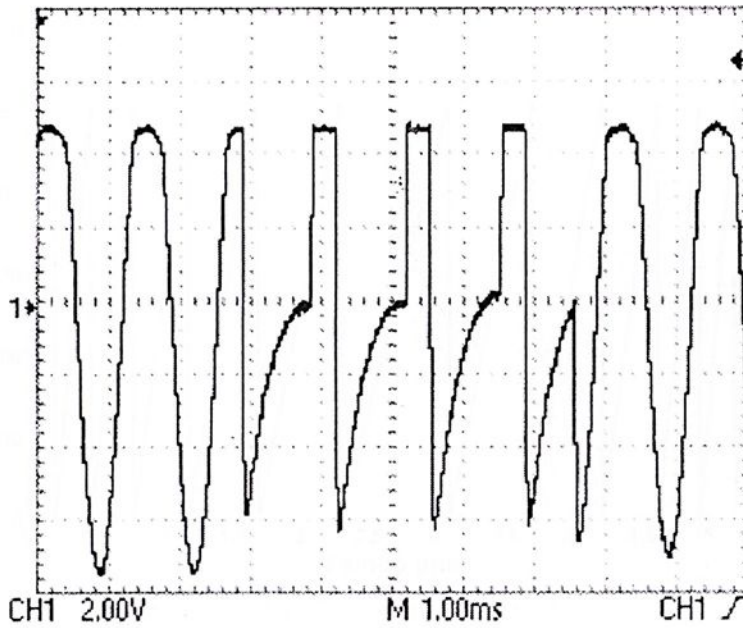


Figura 51: Visualização da seleção de formas de onda Farádica-Senóide em 8 bits (56 Hz).

Inúmeras outras formas de onda podem ser geradas a partir de outros modelamentos de sinais, e escolhendo outros parâmetros de controle dos sinais **enable** e **selecao**.

7.2.7 SIMULAÇÕES DE CONTROLE DE LARGURA DE PULSO

Um outro tipo de controle possível sobre o sinal é a duração do seu período, ou de outra forma, do tempo de duração do seu pulso. O sistema apresentado na Figura 33, tem como objetivo, a partir de um sinal de direção **up_down**, determinar se a duração do pulso vai aumentar ou diminuir. A Figura 52, ilustra como podem ser obtidos os valores de contagem na geração dos sinais. A partir de um controle externo (manual), gera-se pulsos que vão aumentando e depois diminuindo o tempo de duração de um sinal gerado.

A Figura 53 exemplifica como ficariam a implementação do protocolo gerado na Figura 52 para a onda Senóide. No exemplo da Figura 54 é mostrado como é implementação com a onda Farádica.

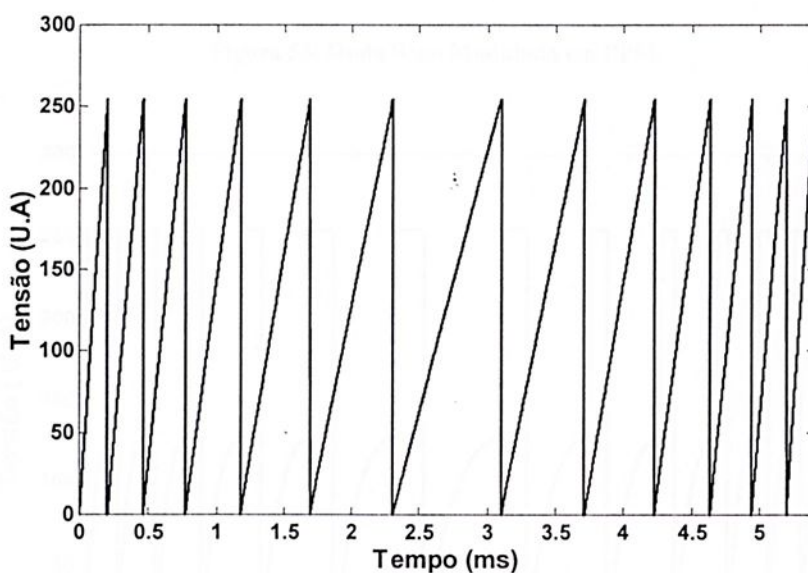


Figura 52: Modulação PPM.

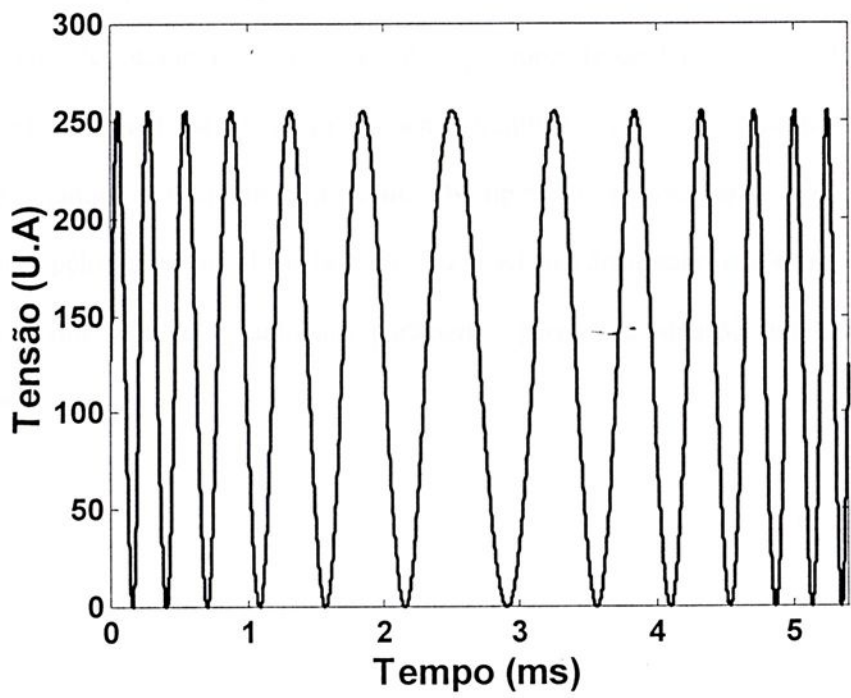


Figura 53: Onda Seno Modulada em PPM.

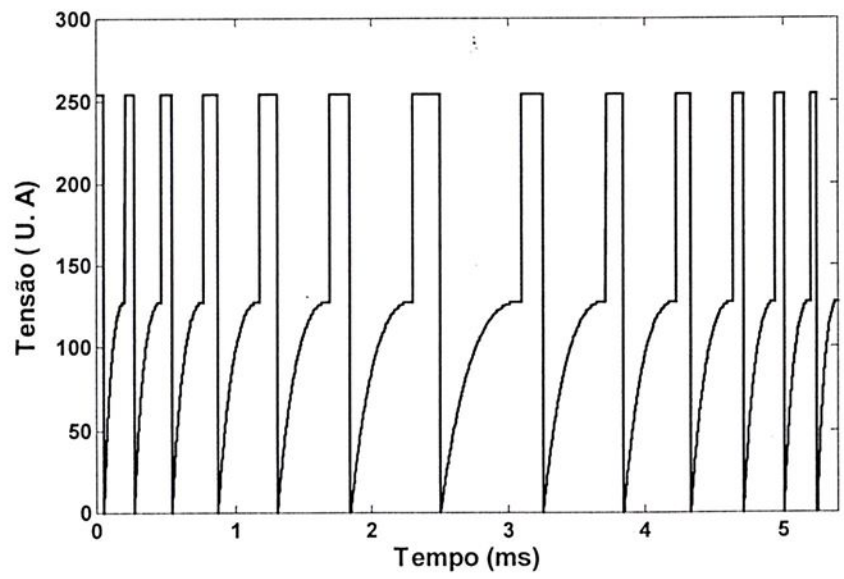


Figura 54: Onda Farádica Modulada PPM.

As diferenças entre os diferentes protocolos de estimulação se deve a diferentes aplicações em eletroterapia. O protocolo do tipo controle de largura de pulso, por exemplo, determina maior solicitação de força quando a frequência do sinal aplicado aumenta e menor solicitação quando a frequência diminui. Os tipos de protocolos e seus parâmetros são determinados pelo responsável médico que faz a adaptação a cada paciente. Desta forma, um aparelho bastante flexível quanto aos parâmetros permite a atuação em diversas síndromes musculares.

Capítulo 8

DISCUSSÕES E CONCLUSÕES

Foi apresentado neste trabalho um estudo sobre eletroterapia e os princípios dos estimuladores. Concluiu-se que o estimulador para ser eficiente deve apresentar como parâmetros: Escolha do nível de estimulação, forma de onda do sinal aplicado e frequência de estimulador. Foram mostrados estímulos modulados de diferentes formas. Estudou-se também os aspectos construtivos dos equipamentos e aspectos de segurança ao paciente.

Foram desenvolvidas ferramentas de síntese automática para geração de projetos em linguagens de descrição de hardware. Foram testadas também, usando diferentes tipos de descrição inicial e foram implementados decodificadores com diferentes parâmetros.

As ferramentas de síntese automática apresentaram-se como poderosas ferramentas para a geração de circuitos digitais.

A utilização da ferramenta E2HDL demonstrou uma maneira rápida de desenvolver uma descrição de hardware baseada na idéia de descrição de forma de onda.

A ferramenta F2HDL apresenta uma nova maneira de se obter uma forma de onda. Através da engenharia reversa, pode-se capturar um sinal periódico qualquer, fazer a análise

espectral e, a partir daí, gerar o circuito digital em que poderia ser implementado em diferentes tecnologias.

O aplicativo TBL2M, demonstra como uma poderosa ferramenta para a verificação e validação de circuitos digitais principalmente decodificadores. Outro aspecto demonstrado pelo aplicativo é a possibilidade de uma análise gráfica de circuitos com diferentes quantidades de bits podendo até gerar gráficos de sinais que não possuam conversores D/A com a quantidade correspondente de bits.

Foram comparadas as linguagens AHDL e VHDL na síntese dos decodificadores implementados e verificou-se diferentes comportamentos quanto as compilações no ambiente MAX + Plus II.

Na linguagem AHDL foi alocado menos memória do que em VHDL para a mesma forma de onda. Este fato sugere que a linguagem AHDL tem um melhor desempenho no ambiente em termos de facilidade de geração de uma descrição booleana do circuito.

Nos componentes FPGA da família FLEX 10K, os circuitos simulados desempenharam grande geração de *spikes* na implementação de circuitos maiores, como a geração trifásica (três canais independentes) e nos decodificadores com valores de n_i maiores que 9 bits. As possíveis soluções para este problema é o uso de componentes com maior capacidade de implementação ou menor tempo de atraso de propagação entre as portas lógicas. Outro tipo de solução poderia ser a implementação de filtros passa-baixa retirando assim os ruídos de alta frequência.

Foi verificado que o aumento na resolução de uma forma de onda, faz que o circuito praticamente dobre de tamanho em termos da quantidade de células. Desta forma, foi escolhida como 8 bits a resolução mais adequada para implementação de circuitos específicos a estimuladores neuromusculares.

O modelamento de circuitos em AHDL permitiu a implementação de circuitos geradores de formas de onda de forma rápida e eficiente e com a conveniência de serem corretas pelo uso de ferramentas automáticas.

Foram implementadas em um único FPGA (EPM7000S) várias formas de onda. Este fato viabiliza o desenvolvimento de um estimulador com grande redução no tamanho e peso dos componentes do circuito.

Na implementação do Projeto Seleção de Formas de Onda, verificou-se que a utilização de um ambiente de projeto que trabalhe em arquitetura em hierarquia, facilita a manutenção do projeto digital, podendo assim, alterar, incluir, ou subtrair sub-projetos sem apresentar dificuldades.

O projeto de Seleção de Formas de Onda, mostrou-se funcional e permite que se faça um projeto em um único componente FPGA da família MAX, permitindo a implementação de até 3 decodificadores-função.

Utilizando dos controles de seleção de forma de onda e do sinal de habilita do contador de forma de onda, pode-se gerar inúmeras formas de onda, fazendo seleções entre os sinais implementados e gerando-se assim protocolos de sinais bastante variados.

As frequências de trabalho realizadas nas medições e simulações, permitem a implementação física das formas de onda, uma vez que o objetivo é a aplicação em sistemas de Estimuladores Neuromusculares onde a frequência de trabalho é sempre menor que 10 KHz.



TRABALHOS FUTUROS

Uma continuação a ser desenvolvida no sistema gerador de formas de onda é a inserção de um estágio de potência, para assim, criar um estimulador neuromuscular.

Pode-se implementar os mesmos sistemas, mas adicionando na saída do conversor D/A filtros passa-baixa, de forma que se elimine parte dos ruídos de alta frequência.

Outro desenvolvimento a ser proposto é a implementação de outros sistemas de protocolos (ou controle) de forma a gerenciar amplitude de corrente de saída.

Usando outras plataformas de síntese, pode-se analisar outras tecnologias de componentes e também o uso de outras linguagens de descrição de hardware.

Pretende-se desenvolver sistemas ASIC que gerem as mesmas formas de onda, mas com redução no consumo de corrente, para que o sistema como um todo possa ser mais compacto.



Capítulo 9

REFERÊNCIAS

-
- [1] LIANZA, S., “Estimulação Elétrica Funcional – FES e Reabilitação”. São Paulo: Editora Atheneu, 1993.
- [2] MUSEU DE ELETROFISIOLOGIA, Eletrofisiologia em: <http://geocities.yahoo.com.br/jcc5001pt/museuelectrofisiologia.htm>, acesso em 07 de jan. 2002.
- [3] CRÉPON, F., “Eletrofisiologia e Reeducação Funcional”, Editora Lovise, 1996.
- [4] HANDA Y., et al. “Current Topics in Clinical Functional Electrical Stimulation in Japan”, J. Electromyogr. Kinesiol., Vol. 7, N0. 4, pp 269-274, 1997.
- [5] KOMIYAMA, C.; SAITO, E. et al “Hybrid FES Assistive System Using a New HKAFO and Surface FES”, Proc. 2nd Int. FES Symp. Sendai, pp. 25-26, 1995.
- [6] KAGAYA H., et al. “An Electrical Knee Lock System for Functional Electrical Stimulation”, Arch. Phys Med. Rehabil, 77, pp 687-694, 1990.

-
- [7] HUIFANG DOU, ASRES A., ZHAOYING AHOU, SENCUN ZHU, "Stimulator for Real Time Control of Paralyzed Muscles During Functional Electrical Stimulation", Proc. Of the Third IEEE International Conference on Electronics Circuits and Systems, pp. 1096-1099, 1996.
- [8] GUDNASON G., BRUUN E., HAUGLAND M., "A chip for an Implantable Neural Stimulator:", Proc. 16th NORCHUO Seminar, Denmark, pp.308-315, 1998.
- [9] FENG W., ANDREWS B. J., "Finite State Machine Provides a Flexible Framework to Design Controllers for Functional Electrical Stimulation", Proc. For the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2578-2581, 1998.
- [10] K. Kaczmarek; K. Kramer; J. Webster; R. Radwin "A 16-channel 8-parameter waveform electrotactile stimulation system" IEEE Trans. Biomed. Eng., vol 38, Out, 1991.
- [11] De Lima, J.; Cordeiro, A. S. "A low-cost neuroestimulator with accurate pulsed-current control" IEEE Trans. Bio. Eng, Vol 49, Mai. 2002.
- [12] WARREN M. GRILL AND THOMAS J. MORTIMER, "Stimulus Waveforms for Selective Neural Stimulation", IEEE Engineering in Medicine and Biology Magazine, July-August, pp. 375-385, 1995.
- [13] DELITTO, A; ROSE, S. J. "Comparative Confort of three waveforms used in electrically eliciting quadriceps" Physical Therapy, nov 1986.
- [14] BOWMAN B. R; BAKER, L. L. "Effects of waveform parameters on comfort during transcutaneous neuromuscular electrical stimulation" Ann Biomed Eng 1985;13: 59-74.
- [15] A. C.R. SILVA; A. S. CARDOSO, "A new Environment to generation of analog waveform using AHDL". In: SBMICRO 2000 - XV International Conference on Microelectronics and Packaging, Manaus -AM -Brasil, 2000.

-
- [16] CARDOSO, A. S.; SILVA, A. C. R.; CARVALHO, A. A., "E2HDL: Uma Ferramenta de Síntese para Implementar Equações Algébricas em FPGA", In: 3rd International Conference on Engineering and Computer Education, Santos – SP - Brasil, 2003.
- [17] A. C.R. SILVA; A. S. CARDOSO, "An Environment to Aid the Synthesis of Threephase Analogue Waveform Using AHDL". In: SBCCI 2001- 14th Symposium On Integrated Circuits And Systems Design, Pirinópolis, GO- Brasil, 2001.
- [18] CARDOSO, A. S.; SILVA, A. C. R.; CARVALHO, A. A., "Da Análise em Série de Fourier ao FPGA", In:3rd International Conference on Engineering and Computer Education, Santos – SP – Brazil, 2003.
- [19] ROBINSON, A. J.; SNYDER-MACLER, L. "Eletrofisiologia Clínica – Eletroterapia e Teste eletrofisiológico." São Paulo: Editora Artmed, 2001.
- [20] M. D. GUYTON, C. ARTHUR, "Tratado da Fisiologia Médica", Editora Guanabara – Sexta Edição.
- [21] KITCHEN S., "Eletroterapia de Clayton", Editora Manole – Décima edição, São Paulo, 1998.
- [22] ROSE S. J., Routhstein J.M., Muscle Mutability. Part 1, "General concepts and adaptations to altered patterns of use". Phys Ther 1982; 62: 1751-1830.
- [23] GAJSKI, D. D ; KUHN, R. H. "New VLSI Tools". IEEE Computer, New York, 16 (12): 11-14, Dez. 1993.
- [24] UNIV. FEDERAL DO RIO GRANDE DO SUL em : http://www.inf.ufrgs.br/inf118/epld_fpga.pdf , acesso em 10/06/2003.
- [25] ALTERA, "MAX + Plus II - AHDL" Versão 6, Nov. 1995.
- [26] ALTERA, "MAX + Plus II - VHDL", Versão 6, Nov. 1995

Apêndice A

MONTAGEM DO GERADOR DE SINAIS

O gerador de sinais implementado em FPGA é constituído por três circuitos distintos: O Gerador de Pulsos, O Decodificador e Conversor Digital Analógico (D/A).

O Decodificador-função

O Decoficador implementado em FPGA, possui dois parâmetros n_i e n_o . O valor de n_i é o tamanho do barramento de entrada e define a amostragem do sinal. Quanto maior o valor de n_i mais pulsos de clock serão necessários para que o sinal percorra um período. O valor de n_o define a resolução do sinal. Para cada valor no barramento de entrada existe um valor correspondente de saída definindo assim, uma tabela de valores. De outra forma, existe uma relação dos valores de entrada e saída implementando desta forma uma função em uma variável. O valor de n_o escolhido deve ser o mesmo da resolução do conversor D/A que é de 8 bits.



O Conversor D/A

A escolha do conversor D/A é um item importante na geração de sinais analógicos. A quantidade de bits define a resolução que o sinal gerado possui. O componente utilizado neste projeto foi o DAC –08CN – Philips que possui as seguintes características:

Interface direta com as tecnologias TTL, CMOS, HTL, PMOS etc.

Escala completa.

Baixo consumo de energia – 37 mW em 5V.

Rápido estabelecimento do sinal de saída – 70 ns.

Alto valor de alimentação –10 até 18 V.

A interface direta permite a conexão a dispositivos FPGA sem a necessidade de buffers ou qualquer outro tipo de acoplador de sinal.

A escala completa permite que se considere todos os valores de contagem como sendo positivos conforme previsto no projeto inicial de geração de formas onda.

Contador – Gerador de Pulsos

O gerador de pulsos cria pulsos retangulares cuja frequência define a frequência do sinal implementado. Tem como entrada um sinal de clock e como saída um barramento de dados do tamanho da entrada do decodificador-função (n_i). Para que o sinal do contador seja mais estável, prefere-se o uso de sistemas síncronos de forma a diminuir os atrasos de propagação.

A Figura A.1 representa de forma simplificada um diagrama de um gerador proposto.



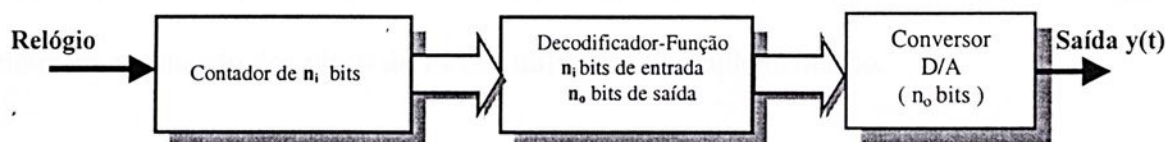


Figura A.1: Geração de Sinal Analógico utilizando Decodificador.

CONFIGURAÇÕES E OPÇÕES

Configuração do Roteamento do FPGA

Para o a criação de um leiaute adequado é necessário que o dispositivo FPGA tenha opções de configuração de quais pinos de I/O serão utilizados.

Devido ao uso de um conversor D/A que dispõe de uma configuração de 16 pinos (DIP – Dual in-line Package) escolhe-se no FPGA duas seqüências onde possa-se seleccionar pinos para I/O.

Desta forma no FPGA, foi configurados os pinos 33, 35, 37 e 39 (mais significativos) e chamados de B+. Os pinos 54, 56, 58 e 60 (menos significativos) de B-.

O barramento B+ é conectado aos pinos A1, A2, A3 e A4 do conversor D/A. E o barramento B- aos pinos A5, A6, A7 e A8.

Para não haver flutuações de tensão, é necessário conectar os pinos de Terra entre o conversor D/A e o componente FPGA.

São ainda configurados também, de acordo com as necessidades de projeto, o posicionamento dos pinos **selecao** e **clock** que são entradas do componente FPGA. A Figura A.2 demonstra a conexão dos pinos do FPGA utilizados na implementação.

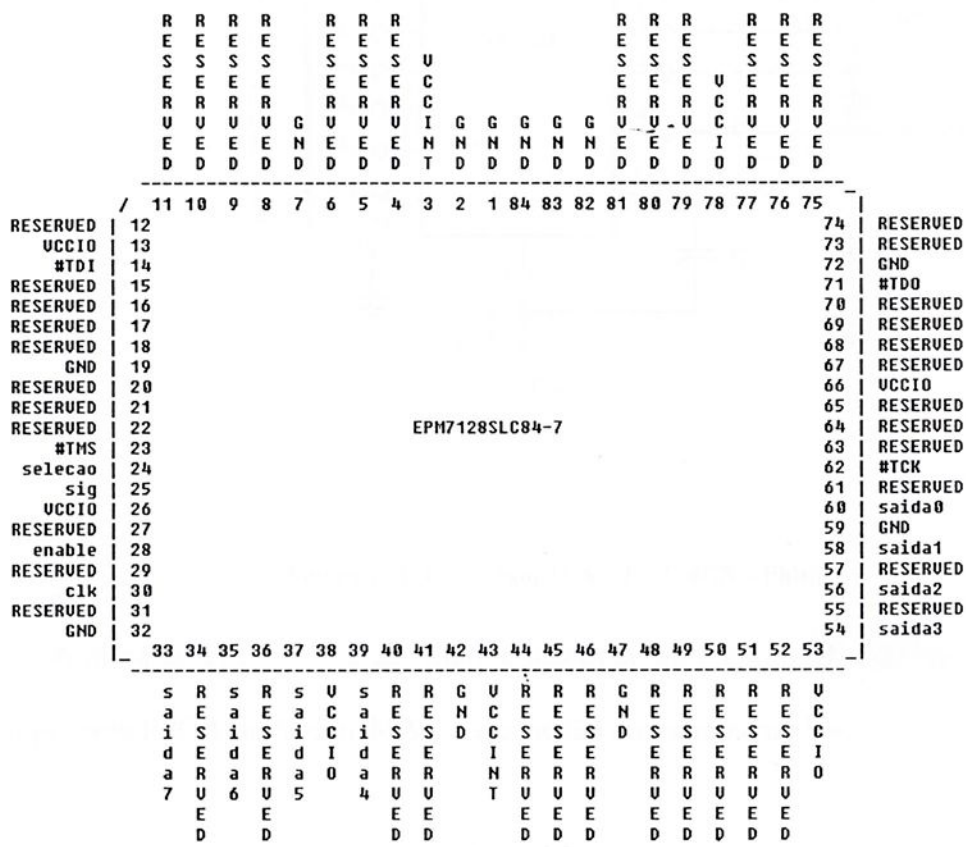


Figura A.2: Planta Baixa do FPGA configurado.

Para o funcionamento do conversor foram seguidos os valores dos resistores, capacitores e tensão de alimentação conforme descrito no manual. A Figura A.3 mostra o diagrama de montagem do circuito de funcionamento do conversor.

O valor do resistor R_L pode necessitar ser diminuído para valores menores que $2\text{ K}\Omega$ caso haja saturação na tensão de saída (V_o).

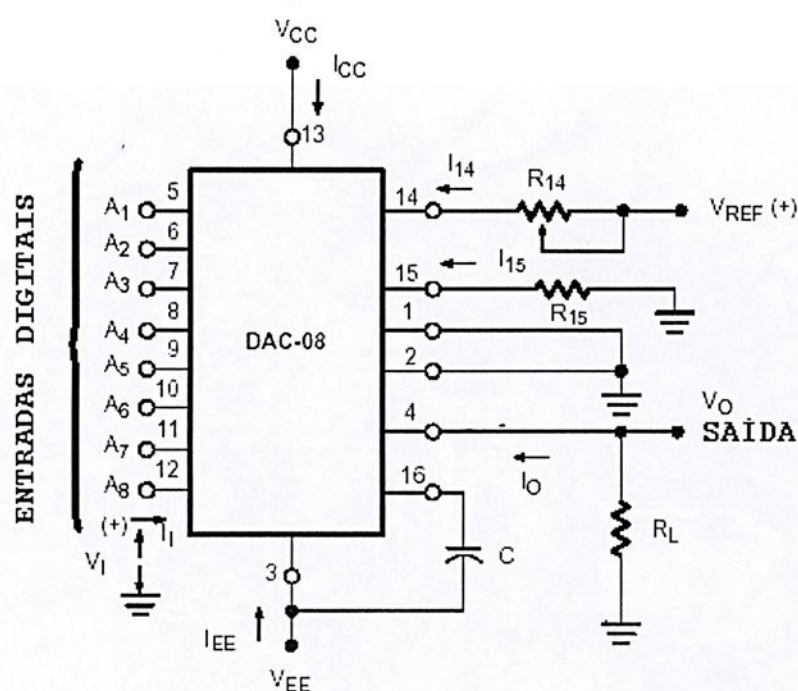


Figura A.3: Conversor D/A – DAC 8CN - Philips.

A placa do Kit didático da Altera é mostrado na Figura A.4. Pode-se notar a utilização do componente FPGA da família MAX e a conexão do barramento B+.

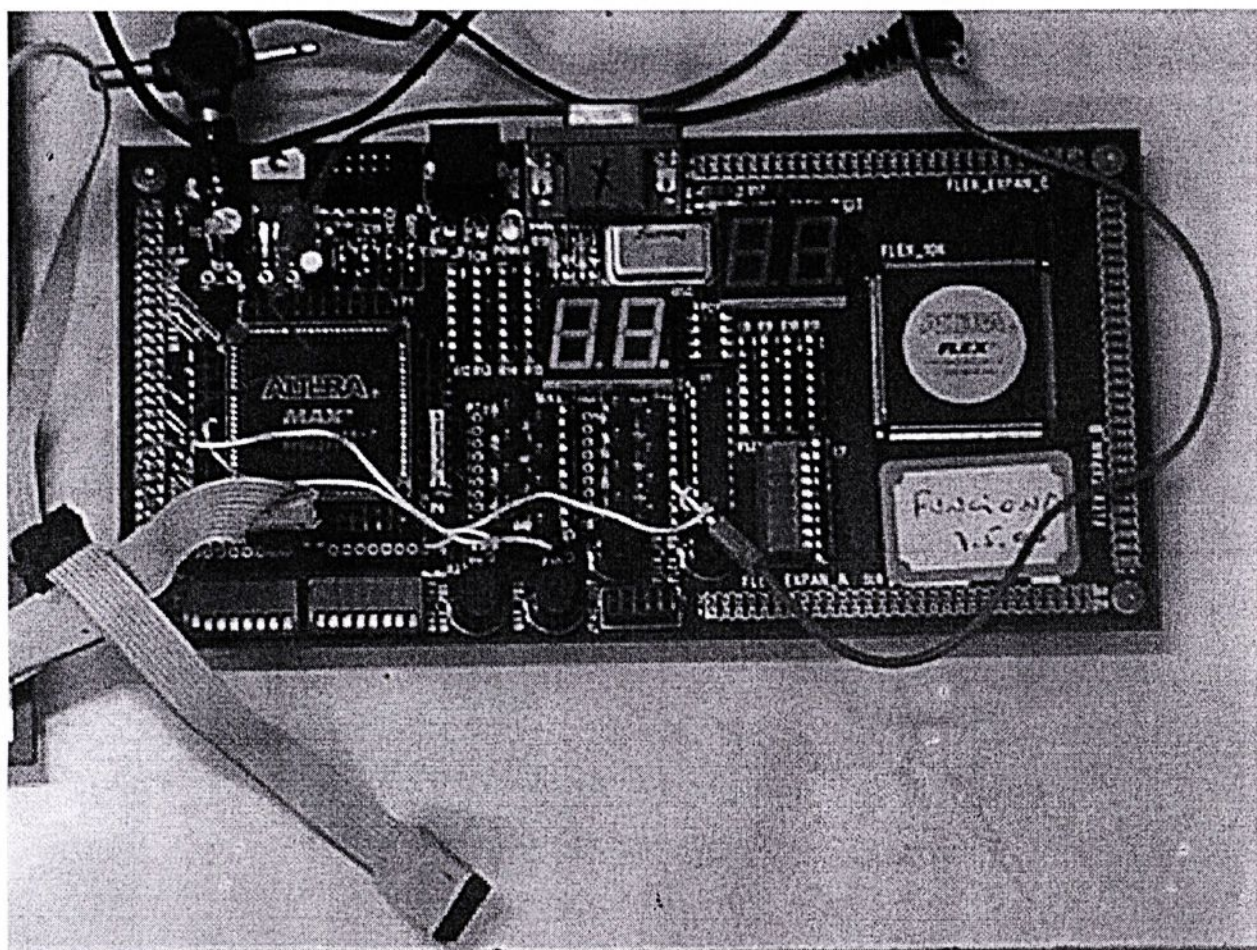


Figura A.4: Modulo Utilizado.



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Câmpus de Ilha Solteira
Programa de Pós-Graduação em Engenharia Elétrica
Av. Brasil Centro, 56
15385-000 Ilha Solteira - SP
www.dee.feis.unesp.br

