

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**“Detecção de Falhas em Sistema de Distribuição de Energia
Elétrica Usando Dispositivos Programáveis ”**

FABIANO ALVES DE SOUZA

Orientador: Profa. Dra. Suely C. A. Mantovani

Dissertação apresentada à Faculdade de
Engenharia - UNESP – Campus de Ilha
Solteira, para obtenção do título de
Mestre em Engenharia Elétrica.

Área de Conhecimento: Automação.

Ilha Solteira – SP
Setembro/2008

FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação
Serviço Técnico de Biblioteca e Documentação da UNESP - Ilha Solteira.

S729d	<p>Souza, Fabiano Alves de. Detecção de faltas em sistemas de distribuição de energia elétrica usando dispositivos programáveis / Fabiano Alves de Souza. -- Ilha Solteira : [s.n.], 2008. 120 f.</p> <p>Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação</p> <p>Orientador: Suely Cunha Amaro Mantovani Bibliografia: p. 92-95</p> <p>1. Detecção de faltas. 2. Energia elétrica – Distribuição. 3. VHDL (Linguagem descritiva de hardware). 4. Sistemas reconfiguráveis.</p>
-------	--

CERTIFICADO DE APROVAÇÃO

TÍTULO: DETECÇÃO DE FALTAS EM SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA
USANDO DISPOSITIVOS PROGRAMÁVEIS

AUTOR: FABIANO ALVES DE SOUZA

ORIENTADORA: Profa. Dra. SUELY CUNHA AMARO MANTOVANI

Aprovado como parte das exigências para obtenção do Título de MESTRE em ENGENHARIA ELÉTRICA pela Comissão Examinadora:

Profa. Dra. SUELY CUNHA AMARO MANTOVANI

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

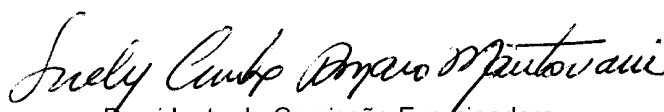
Prof. Dr. NOBUO OKI

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. LUIS GUSTAVO WESZ DA SILVA

Escola de Engenharia Elétrica / Universidade Federal de Goiás

Data da realização: 08 de setembro de 2008.



Presidente da Comissão Examinadora
Profa. Dra. SUELY CUNHA AMARO MANTOVANI

*Aos amigos Adriano Cardoso, Carlos Antunes Febres
A minha orientadora Suely Cunha Amaro Mantovani
Ofereço*

Agradecimentos

Primeiramente quero agradecer a Deus pela força que Ele me proporcionou nesta batalha.

Aos meus pais Orlando e Gisélia pela força, pelo apoio, pelo incentivo em muitas horas e o principal que obtive deles: Oração.

A minha irmã Fabrícia pelas horas em que eu a ocupei e pelo incentivo que sempre recebi dela.

A minha namorada Luiza pelas horas de auxilio e incentivo. “Te amo”.

A minha orientadora Suely, por ter me agüentado todos estes momentos e pela oportunidade de ter trabalhado comigo neste projeto.

Aos amigos Marcelo Sanches, Elcio Alteris, Wesley Pontes, Renato de Aguiar (Soneca), CRA, Marcelo (Pardal) e ao Carlos (Peruano).

Ao Departamento de Engenharia Elétrica pelo apoio financeiro na divulgação de trabalhos.

Aos professores Prof. Dr. Nobuo Oki, Prof. Dr. Luis Gustavo Wesz da Silva por aceitar fazer parte da banca examinadora.

Aos professores supelentes Dr. Jozue Vieira Filho, Dr. Marco Aparecido Queiroz Duarte por estar dispostos a fazerem parte da banca.

Ao pessoal da seção da Pós Graduação, Onilda Naves de Oliveira Akasaki, Adelaide Amaral dos Santos, entre muitos outros funcionários da FEIS/UNESP.

*“O coração alegre serve de bom remédio, mas o espírito
abatido virá secar os ossos”
Provérbios 17:22*

Resumo

Atualmente as empresas do setor elétrico deparam-se cada vez mais com as exigências do mercado energético sendo obrigadas a assegurarem aos seus clientes bons níveis de continuidade e confiabilidade no serviço de fornecimento da energia elétrica e também atender os índices de continuidade do serviço estabelecidos pela agência reguladora do setor elétrico (ANEEL – Agência Nacional de Energia Elétrica). Para alcançar estes objetivos além de investir na otimização dos seus sistemas de transmissão e distribuição, as empresas responsáveis têm investido na automação de suas operações, buscando alternativas que reduzam os tempos de interrupção por faltas permanentes nos sistemas de potência. Através de informações disponíveis em uma subestação, é possível estabelecer um procedimento para determinar e classificar condições de faltas, localizando o elemento de proteção acionado, e assim fornecer o apoio à tomada de decisão no ambiente de subestações de sistemas de distribuição de energia elétrica. Neste trabalho é proposta uma metodologia que fornece respostas rápidas (controle *on line*), para detecção e classificação de faltas em sistemas de distribuição de energia elétrica através de informações analógicas disponíveis em uma subestação, tais como amostras de sinais de tensões e correntes na saída dos alimentadores, com uma arquitetura reconfigurável paralela que usa dispositivos lógicos programáveis (Programmable Logic Devices – PLDs) -FPGAs e a linguagem de descrição de hardware – VHDL (Very High Speed Integrated Circuit – VHSIC). Para validar o sistema proposto, foram gerados dados de forma aleatória, compatíveis com informações fornecidas em tempo real pelo sistema SCADA (*supervisory control and data-acquisition*) de uma subestação real. Os resultados obtidos com as simulações realizadas, mostram que a metodologia proposta, apresenta resultados satisfatórios, e tempos de respostas razoáveis.

Palavras-chaves – Detecção de Faltas, Sistema de Distribuição, Linguagem de Descrição de Hardware , Sistemas Reconfiguráveis.

Abstract

Currently companies of the energy industry is facing increasingly with the requirements of the energy market are obliged to ensure their customers good levels of continuity in service and reliability of supply of electric energy and also meet the rates of continuity of service established by the agency regulator of the energy industry (ANEEL - National Electric Energy Agency). To achieve these goals than to invest in optimization of its transmission and distribution systems, the companies responsible have invested in automation of its operations, seeking alternatives that reduce the time of interruption by failures in the systems of permanent power. Through information available in a substation, it is possible to establish a procedure for identifying and classifying conditions of absence, finding the element of protection driven, and thus provide support for decision-making within the environment of substations to distribution systems for power. This work is proposed a methodology that provides quick answers (control online), for detection and classification of faults in distribution systems of electric energy through analog information available on a substation, such as samples for signs of tensions and currents in the output of feeders, with an architecture that uses parallel reconfigurable programmable logic devices (Programables Logics Devices - PLDs)-FPGAs and the language of description of hardware - VDHL (Very High Speed Circuit Integrated - VHSIC). To validate the proposed system, data were generated at random, consistent with information provided by the system in real time SCADA (supervisory control and data-acquisition) of a real substation. The results obtained with the simulations conducted, show that the proposed methodology, presents satisfactory results, and times of reasonable answers.

Keywords – Fault Detection, Distribution Systems, Language of Description of Hardware, Reconfigurable Systems.

Listas de Figuras

Figura 1.1.	Diagrama resumido de um sistema elétrico de potência.....	19
Figura 2.1.	Diagrama esquemático de um sistema elétrico de potência.....	24
Figura 2.2.	Esquema de seletividade da proteção	28
Figura 2.3.	Zonas de proteção principal	29
Figura 2.4.	Zona de proteção de retaguarda – alimentadores.....	30
Figura 2.5.	Ligações - TPs/TCs e IEDs.....	34
Figura 2.6.	Sistema SCADA típico	35
Figura 3.1.	Modelo de Santos para detectar faltas em sistema de potência usando Redes de Petri Coloridas.....	39
Figura 3.2.	Sistema de detecção de faltas utilizando redes de Petri Hierárquica.....	41
Figura 3.3.	Diagrama de blocos do sistema de detecção.....	42
Figura 3.4.	Diagrama unifilar do alimentador simulado	42
Figura 3.5.	Testes em bancada com sensores aplicados nas três fases.....	43
Figura 3.6.	Diagrama de blocos simplificado.....	44
Figura 3.7.	Diagrama de Blocos com a localização de faltas usando Redes Neurais	45
Figura 3.8.	Emulação Eletrônica baseada no conceito de DNA	47
Figura 4.1.	Estrutura de uma PROM típica	52
Figura 4.2.	PLA básico (AND-OR) programáveis nos dois planos	53
Figura 4.3.	Diagrama bloco do dispositivo da família MAX (Multiple Array Matrix) MAX7000	55
Figura 4.4.	Arquitetura de um FPGA	57
Figura 4.5.	Fluxo de um projeto	63
Figura 4.6.	Evolução das famílias de PLDS da ALTERA	65
Figura 4.7.	Comparação FPGA convencional x FPGA Stratix III.....	66
Figura 4.8.	Arquitetura do Stratix III E EP3SE50.....	66

Figura 5.1.	Tela de supervisão de uma Subestação real	73
Figura 6.1.	Diagrama de Fluxo do Sistema	79
Figura 6.2.	Arquitetura do Sistema de Detecção de Faltas	80
Figura 6.3.	Circuito que compara os registradores de faltas e pré-faltas	81
Figura 6.4.	Simulação do controlador	82
Figura 6.5.	(a) Esquemático ilustrando a função do registrador paralelo/paralelo.....	83
	(b) Símbolo para o Registrador– Quartus II Altera	83
Figura 6.6.	Simulação do Registrador Paralelo/Paralelo.....	83
Figura 6.7.	Simulação do Conversor 64,16X80	84
Figura 6.8.	Esquemático das Memórias ROMs.....	84
Figura 6.9.	Simulação da Memória Rom	85
Figura 6.10.	Simulação para o comparador mostrando uma falta ocorrida	85
Figura 6.11.	Bloco Esquemático topo do projeto hierárquico.....	86
Figura 6.12.	Diagrama esquemático para o Sistema de Detecção de Faltas	87
Figura 6.13.	Simulação 1, completa do Sistema proposto	88
Figura 6.14.	Simulação 2, completa do Sistema proposto	88

Listas de Tabelas

Tabela 4.1.	Principais tipos de chaves e tecnologia de fabricação	59
Tabela 5.1.	Probabilidade entre os Tipos de Faltas e Composição das mesmas	69
Tabela 5.2.	Sinais Analógicos obtidos no alimentador C1 de uma SE real.....	71
Tabela 5.3.	Sinais Digitais	72
Tabela 5.4.	Sinais analógicos em C1 e seus valores correspondentes.....	74

Listas de Abreviaturas e Siglas

AHDL	Altera- High Speed Integrated Circuit
A/D	Analógico/Digital
B-AT	Barramento de Alta
CLBs	Configurable Logic Block
CI	Circuito Integrado
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Complex Programmable Logic Device
DMS	Distribution Management System
EDL	Entrada de Linha
EEPROM	Electrical Erasable PROM
EPROM	Erasable PROM
FPAA	Field Programmable Analog Array
FPGA	Field-Programmable Gate Array
FPLA	Field Programmable Logic Array
HDL	Hardware Description Language
VHDL	Very High Speed Integrated Circuit – VHSIC
IEDs	Intelligent Electronic Devices
INPI	Instituto Nacional da Propriedade Industrial
IOBs	Input/Output Blocks
I/O	Input e Output
PLA	Programmable Logics Array
PLD	Programmable Logics Devices
RAM	Random Access Memory
ROM	Read Only Memory
SCADA	Supervisory Control and Data Acquisition
SDF	Sistema de Detecção de Falta
SE	Subestação
SDL	Saída de Linha
TC	Transformadores de Corrente
TP	Transformadores de Potência

UTR

Unidades Terminais Remotas

Sumário

Capítulo 1: Introdução	17
Capítulo 2: Subestação de Distribuição	23
2.1. Conceitos Básicos de uma Subestação (SE)	23
2.2. Subestação de Distribuição - Visão Geral	25
2.3. Sistema de Proteção	27
2.4. Automação de Subestações	30
2.4.1-Sistema SCADA	32
2.5. Conclusão	36
Capítulo 3: Revisão Bibliográfica em Detecção de Faltas	37
3.1. Trabalhos Publicados para a Detecção de Faltas em SE	37
3.2. Conclusão	47
Capítulo 4: Dispositivos Lógicos Programáveis	49
4.1. Evolução Tecnológica	49
4.2. Dispositivos Lógicos Programáveis (PLDs- Programmable Logic Device)	51
4.2.1.Dispositivos PROM	51
4.2.2.Dispositivos PLA	52
4.2.3. Dispositivos CPLDs	53
4.2.4. Dispositivos FPGAs	55
4.3. Considerações no uso de PLDs	57
4.3.1. Arquitetura em Hierarquia de Projeto	57
4.3.2. Paralelismo	58
4.4. Características comuns aos Dispositivos Lógicos Programáveis	58
4.4.1. Reconfiguração	58
4.4.2. Tecnologia de Programação	59
4.4.3. Processo de realização de um projeto	59
4.5. Linguagens de Descrição de Hardware	60
4.5.1. Linguagem VHDL (Very High Speed Integrated Circuit Hardware Description Language)	61
4.6. Principais Fabricantes dos PLDs	64
4.7. Dispositivo Stratix III	65
4.8. Conclusão	67
Capítulo 5: Metodologia Proposta para Detecção de Faltas	68
5.1. Faltas em Sistemas Elétricos	68
5.2. Tipos de Arquiteturas	75
5.2.1. Arquitetura Pipeline	75
5.2.2. Arquitetura Superescalar	75
5.3. Conclusão	75
Capítulo 6: Arquitetura do Sistema de Detecção de Faltas	77
6.1. Arquitetura	78
6.2. Controlador	82
6.3. Registrador Paralelo/Paralelo	82

6.4. Conversor 64,16 x 80	83
6.5. Memórias	84
6.6. Comparador	85
6.7. Arquitetura Completa	86
6.8. Conclusão	89
Capítulo 7: Conclusões Gerais e Sugestões para Trabalhos Futuros.....	90
7.1. Conclusões Gerais	90
7.2. Sugestões para Trabalho Futuro	91
Referências	92
Apêndice A: Decodificador	96
Apêndice B: Conversor	98
Apêndice C: Registrador Paralelo/Paralelo	100
Apêndice D: Memórias Roms	103
Apêndice E: Comparador	113
Apêndice F: Conversor 64,16x80.....	115
Apêndice G: Controlador.....	118
Anexo I: Artigo Publicado	120

Capítulo 1

Introdução

Nos últimos anos, devido ao crescimento da demanda de energia elétrica, as operações dos sistemas de potência tornaram-se muito complexas. Este fato é notório no número de interligações entre os sistemas de proteção.

Os sistemas de transmissão e distribuição de energia elétrica têm apresentado um grande crescimento em vista do aumento dos núcleos populacionais e principalmente devido ao desenvolvimento Industrial. Este crescimento tem maior reflexo no sistema de distribuição, pois são eles os circuitos terminais de fornecimento de energia. Portanto, o grande desafio às concessionárias de distribuição hoje em dia é fornecer energia elétrica aos seus consumidores de maneira sustentável e confiável.

A confiabilidade e a sustentabilidade do fornecimento de energia relacionados a qualidade do produto, são fatores que atualmente estão sendo atingidos pelos projetos de um Sistema de Proteção Robusto e o Investimento e Caracterização de um Sistema de Automação

capaz de controlar em tempo real, grandes quantidades de informações (JARDINI, 1996, CAMINHA, 2004).

Um sistema de proteção para ser considerado robusto, deve ter características como seletividade, rapidez, sensibilidade, segurança e economia (CAMINHA, 2004). Um sistema de automação atualmente exige a instalação de equipamentos de aquisição de dados e dispositivos (sensores, relés) remotamente controlados, com capacidade de processar as informações advindas do sistema de aquisição e suporte de funções de decisão.

Um sistema de Supervisão e Controle muito conhecido atualmente é o sistema SCADA (*Supervisory Control and Data Acquisition*). O sistema SCADA conecta a subestação, onde estão os equipamentos de medição, monitoramento e controle digitais ao centro de operação, onde são coletados, armazenados, apresentados e processados os dados. Este sistema foi desenvolvido pela DMS (*Distribution Management System*)

As tradicionais práticas de diagnóstico de defeitos se baseiam no julgamento dos operadores, levando em conta suas experiências em análise e planejamento de ações corretivas. Em subestações automatizadas, durante a ocorrência de uma falta, os sinais de alarmes recebidos indicando anormalidade, caracterizam-se pelo *status* da atuação dos relés e dos disjuntores. Na sequência e em tempo mínimo, os operadores devem estimar a situação do defeito (detecção, classificação e localização). Em vista dos complexos sistemas atuais de distribuição, realizar a automação de forma eficiente é considerada uma tarefa difícil, se os procedimentos adotados baseiam-se somente nos operadores humanos.

Por isso, as metodologias que utilizam a inteligência artificial (Sistemas Especialistas (SANTOS et al, 2007), Redes Neurais (MORETO, 2005), Lógica Nebulosa (DECANINI, 2008) são procedimentos disponíveis atualmente, capazes de extrair conhecimentos e bastante aplicados a processos complexos, como é o caso da automação de subestações.

Estas técnicas têm sido usadas para auxiliar os operadores na execução de rotinas de forma segura, rápida e eficiente e no planejamento de ações corretivas. Estes recursos têm proporcionados inúmeras alternativas para realizar a automação de subestações de forma plena e eficiente.

Destacam-se também entre as mudanças que atingem o setor elétrico, as evoluções dos dispositivos tecnológicos que fazem parte da distribuição de energia e possibilitam mais segurança e eficiência.

Por outro lado, o benefício e a comodidade proporcionados pela energia elétrica tornam a população dependente dos serviços prestados pelas empresas de distribuição de energia

elétrica, de forma que a qualidade nestes serviços é uma exigência atual do consumidor (JARDINI, 2006).

A função de um sistema elétrico de potência é disponibilizar energia elétrica de uma forma segura, confiável, baixo custo e de qualidade. Este sistema é composto de usinas geradoras, linhas de transmissão e subestações. A figura 1.1 apresenta um diagrama resumido de um sistema elétrico de potência.

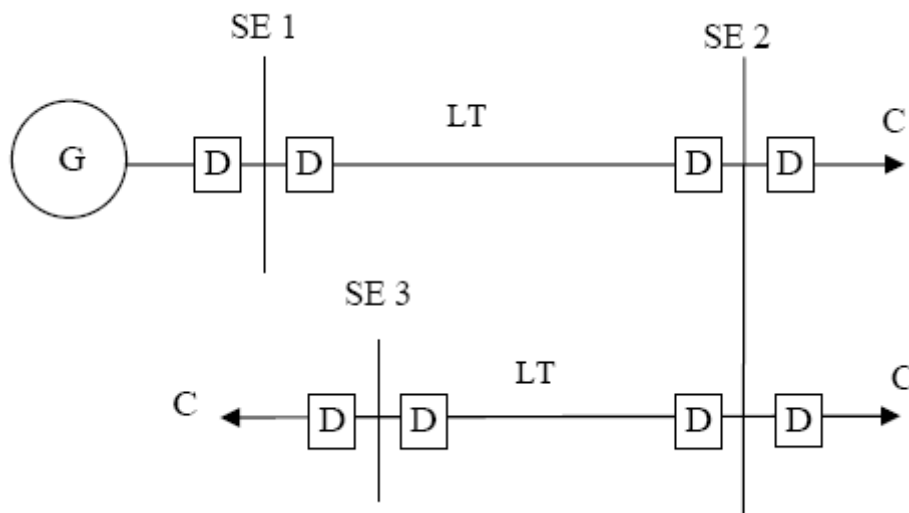


Figura 1.1- Diagrama resumido de um Sistema Elétrico de Potência (SAMPAIO, 2002).

Legenda:

G - Geração

D - Equipamento de Disjunção

SE1 - Subestação Elevadora

SE2 - Subestação Distribuidora

SE3 - Subestação Distribuidora

LT - Linha de Transmissão

C - Carga ou Consumidor

Para gerar e fornecer energia elétrica dentro dos padrões de qualidade e segurança, um sistema de geração, transmissão e distribuição de energia elétrica deve dispor de sistemas altamente confiáveis que permitem a redução de tempo para se executar proteções, causadas por descargas atmosféricas, falhas de equipamentos, colisão, dentre outras.

Embora os sistemas modernos de supervisão e controle, que constituem os sistemas de proteção de uma subestação, possuam dispositivos tecnológicos avançados, eles não estão livres de sofrerem danos pela natureza, causando as falhas.

O equipamento de proteção em um sistema de distribuição de energia tem como finalidades não somente evitar danos permanentes aos equipamentos, mas também, diminuir o tempo de interrupção do fornecimento de energia causado por falhas.

A detecção de faltas em um sistema de distribuição de energia pode ser realizada de duas formas, conforme propõe (SAMPAIO, 2002):

- **Forma local** - a detecção é executada internamente em uma SE.
- **Forma centralizada**- a detecção de faltas é executada dentro de um sistema de controle que contam com equipamentos do sistema SCADA central.

Neste trabalho está sendo proposto, um circuito ou arquitetura que atue próximo a um sistema de aquisição de dados (em campo), atuando de forma rápida na detecção e diagnóstico de faltas, usando para isso dispositivos lógicos programáveis - PLDs (*Programmable Logic Devices*) e a uma linguagem que permite a descrição de hardware como a VHDL (*Very High Speed Integrated Circuit – VHSIC*) para construir uma arquitetura reconfigurável, tendo em vista, reduzir não somente o tempo de atuação, mas também custos e o tamanho do projeto .

Objetivos

Este trabalho tem como objetivos:

- Uma revisão de literatura, mostrando alguns trabalhos e sua metodologia sobre o diagnóstico de faltas em Sistemas Elétricos de Distribuição de Energia;
- Aplicação de dispositivos lógicos programáveis e a tecnologia dos circuitos reconfiguráveis, e o uso de uma linguagem de descrição de hardware.
- Propor o desenvolvimento de uma arquitetura de um sistema digital capaz de detectar e classificar faltas em um Sistema de Potência;

Justificativa

Grandes investimentos estão sendo aplicados na modernização e eficiência dos sistemas de energia elétrica, em especial, nas subestações agregadas aos sistemas de distribuição, para a automação de suas operações. A tecnologia da informação com suas técnicas de processamento e análise de sinais, aliados à tecnologia digital oferecem hoje, dispositivos semicondutores de alta velocidade e grandes recursos. A partir da aquisição de dados, podem-se desenvolver sistemas integrados que façam o processamento e a análise de dados para realizar o controle e a tomada de decisão no ambiente de subestações. Somam-se a esses

fatores, as exigências impostas pelos órgãos reguladores e a abertura do mercado de energia, fazendo com que as empresas e os profissionais de engenharia necessitem cada vez mais de ferramentas efetivamente eficientes, capazes de analisar as informações e apresentar diagnósticos claros, com rapidez e precisão para auxiliar os profissionais no exercício da função.

Uma subestação está sujeita as várias perturbações que podem ser causadas por descargas atmosféricas, rompimento de cadeias de isoladores, acidentes, incêndios, queimadas, etc., gerando faltas do tipo curto-circuito (monofásicos, bifásicos e trifásicos). Para tal, existem dispositivos capazes de detectar e disparar sinais para interromper a linha de transmissão em que houve esta falta. Estes dispositivos são conhecidos como equipamentos de proteção e são responsáveis pela detecção e eliminação de faltas ocorridas, e devem operar no menor tempo possível, evitando que a integridade física do sistema seja comprometida devido a estas faltas.

Diante dessa necessidade e com a evolução dos equipamentos para aquisição, processamento e transmissão de sinais elétricos, várias técnicas e algoritmos para localização de faltas em subestações de distribuição de energia elétrica têm sido propostas (SAMPAIO, 2002, MORETO, 2005, DECANINI, 2008). Dentre as principais dificuldades encontradas pela grande maioria das técnicas empregadas na localização das faltas, está a topologia da rede que geralmente é altamente ramificada, as variações nas impedâncias da rede devido à reconfiguração, a existência ou não de co-geração no alimentador considerado, a distribuição e os níveis de cargas na rede que se refletem diretamente nas correntes e tensões pré-falta, as seções da rede com condutores e o conhecimento exato da impedância do sistema que se encontra atrás da subestação (JARDINI, 1996, CAMINHA, 2004).

Visando a qualidade dos serviços prestados, as empresas de geração e distribuição de energia elétrica, buscam alternativas que reduzam os tempos de interrupção dos serviços de fornecimento de energia devido às faltas permanentes nos sistemas de potência. Através de informações disponíveis em uma subestação, é possível estabelecer um procedimento para determinar e classificar condições de faltas, localizando o elemento de proteção acionado, fornecendo o apoio à tomada de decisão no ambiente de subestações de sistemas de distribuição de energia elétrica.

A implementação desta metodologia só é possível, devido aos modernos sistemas de aquisição e processamento de sinais elétricos nas subestações de distribuição, onde existem medidores equipados com unidades terminais remotas (UTR) (MC DONALD, 2003), o conhecimento das condições topológicas da rede em associação com modelos e técnicas de

análise de circuitos elétricos. Com isso, pode-se obter um sistema rápido e eficiente para localização de faltas e contribuir para que as concessionárias trabalhem com bons indicadores de qualidade e continuidade de serviço.

Metodologia

Para o estudo e desenvolvimento deste sistema, este trabalho considera o embasamento conceitual, o desenvolvimento tecnológico e as principais pesquisas na área.

Este sistema foi baseado no artigo de Delgado (PERLAZA; DELGADO, 2005) onde em sua pesquisa, foi utilizada a teoria de um *chip de DNA* para detectar faltas em um sistema de potência.

Para o entendimento teórico foram realizadas pesquisas, compreendendo a importância de dispositivos de proteção em um sistema de distribuição de energia elétrica. Primeiramente, foram estudados conceitos relacionados a uma subestação e sistemas digitais. Em seguida, estudos foram realizados para desenvolver um dispositivo que permitiria a criação de diversos componentes computacionais. Por fim, foram simulados exemplos e desenvolvida uma arquitetura para atingir os objetivos propostos inicialmente.

Organização do texto

O trabalho está organizado em 7 capítulos. No capítulo 2 apresentam-se os conceitos que regem uma subestação e seus principais equipamentos utilizados para proteção e distribuição de energia elétrica. No capítulo 3 apresenta-se a revisão de literatura sobre o diagnóstico de faltas. No capítulo 4 são detalhadas as tecnologias reconfiguráveis, onde aborda-se principalmente os dispositivos CPLDs e FPGAs, de forma a constituir o embasamento necessário para compor a plataforma de aplicação para este trabalho. No capítulo 5, trata-se da metodologia proposta para a detecção de faltas e sinais de uma SE. Também neste capítulo descrevem-se os parâmetros que servirão para o entendimento da arquitetura proposta. Apresenta-se no capítulo 6 a arquitetura final do sistema com algumas simulações. Finalmente, as conclusões finais e sugestões para futuros trabalhos são apresentadas no capítulo 7, seguido das Referências Bibliográficas.

Capítulo 2

Subestação de Distribuição

Neste capítulo apresentam-se os conceitos básicos de proteção e automação de subestações de distribuição de energia elétrica, tema abordado nesta dissertação e importante para o entendimento deste trabalho.

2.1 Conceitos Básicos de uma Subestação (SE)

Os sistemas elétricos de potência, de modo geral, podem ser divididos basicamente em: geração, transmissão, subtransmissão e distribuição. Em cada parte do sistema elétrico, há a necessidade de se reduzir e elevar, de maneira conveniente, os níveis de tensão de modo que o transporte de energia elétrica seja econômico e, portanto viável. As subestações de energia elétrica interconectam de maneira confiável e segura o sistema elétrico considerando os diversos níveis de tensão ao longo de seu percurso. Na Figura 2.1 ilustra-se um sistema elétrico de potência típico.

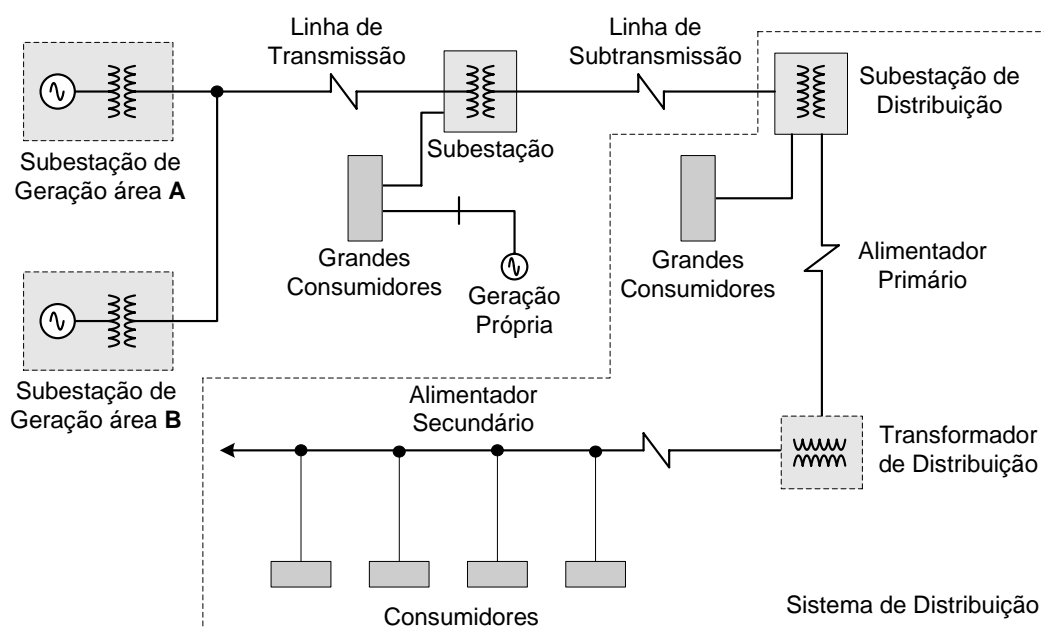


Figura 2.1- Diagrama esquemático de um sistema elétrico de potência.

Algumas tensões típicas em cada parte do sistema elétrico são:

- a) Geração (hídrica, térmica, nuclear, diesel, fontes não convencionais): 6,6; 10,5; 11; 13,8; 15,75; 21 e 33 kV;
- b) Interconexão entre áreas de geração: 220, 400, 500, 750, 765, 800 kV etc.;
- c) Subtransmissão em alta tensão: 33, 66, 110, 132, 220 kV;
- d) Alimentadores primários: 3,3; 6,6; 11,2; 13,8; 15; 22; 33 e 66 kV;
- e) Alimentadores secundários: 127; 220; 230; 240; 380; 400 V.

O sistema elétrico da Figura 2.1 é composto de quatro subestações de energia elétrica: duas subestações de geração A e B, uma subestação de subtransmissão e uma subestação de distribuição.

A confiabilidade e a sustentabilidade do fornecimento de energia elétrica estão relacionadas com a qualidade e disponibilidade da energia nos pontos de consumo. Deve-se ter em mente que o projeto do arranjo da subestação varia muito de concessionária para concessionária, da região em que será construída a subestação, que tipo de carga se destina, o quanto é importante essa carga e principalmente quanto à concessionária está disposta a investir no arranjo de sua subestação etc.

2.2- Subestação de Distribuição - Visão Geral

São vários os equipamentos existentes em uma subestação de distribuição, tais como:

- Barramentos;
- Linhas e alimentadores;
- Equipamentos de manobra e disjunção: disjuntores, chaves seccionadoras;
- Equipamentos de transformação: transformadores de potência, transformadores de instrumentos – de potencial (TP) e de corrente (TC), e de serviço;
- Equipamentos de proteção: relés (primário, retaguarda e auxiliar), pára-raios e malha de terra;
- Equipamentos de compensação: reatores e capacitores.

As subestações são compostas por conjuntos de elementos, com funções específicas no sistema elétrico, denominados vãos (*bays*) que permitem a composição da subestação em módulos. As subestações de distribuição, usualmente, são compostas pelos seguintes vãos: entrada de linha (EDL); saída de linha (SDL); barramentos de alta (B-AT) e média (B-MT) tensão; vão de transformação (TF); banco de capacitor ou vão de regulação (BC) e saída de alimentador (AL).

Em uma subestação os serviços auxiliares são de grande importância para sua operação adequada e contínua. Os serviços auxiliares são do tipo:

- Serviços auxiliares de corrente alternada:
 - Fonte: transformador de serviços auxiliares – tipicamente :13.800/380-220 V;
 - Cargas: casa de comando, iluminação/tomada do pátio, retificador, etc;
- Serviços auxiliares de corrente contínua:
 - Fonte: retificador/carregador e banco de bateria – tipicamente: 125 Vcc;
 - Cargas: componentes do sistema digital (relés, etc.), funcionais dos equipamentos, motores dos equipamentos, iluminação de emergência.

As funções em uma subestação são:

- Monitoração de "*status*" de equipamentos;
- Medição;
- Proteção de linha, transformadores, barra, reator, etc;
- Supervisão das proteções;
- Religamento automático;

- Localização de falta na linha;
- Telecomandos;
- Proteção de falha de disjuntor;
- Intertravamentos;
- Monitoração de sobrecarga em transformadores;
- Controle de tensão;
- Fluxo de reativos;
- Corte seletivo de cargas;
- Sincronização;
- Alarmes em geral;
- Registro de seqüência de eventos;
- Oscilografia;
- Interface Homem-Máquina (IHM);
- Impressão de relatórios;
- Interface com os Centros de Operação de Distribuição (CODs);
- Auto-diagnóstico.

Todos os vão da subestação devem possuir dispositivos de proteção (relés) e equipamentos de manobra e disjunção com a finalidade de limitar os impactos proporcionados por faltas (curtos-circuitos) no sistema elétrico. Uma falta pode ser temporária ou permanente. As faltas temporárias são aquelas em que havendo a operação de um equipamento de proteção desaparece a causa do defeito e o sistema funciona normalmente depois de religado (SATO, 2003).

Aproximadamente cerca de 80% das faltas que ocorrem nos sistemas de distribuição são **faltas temporárias** (CPFL, 2003) que tem como causas mais comuns, descargas atmosféricas, contatos momentâneos entre condutores, abertura de arco elétrico e materiais sem isolamento adequada.

As **faltas permanentes** são aquelas em que é necessária a intervenção do homem para que se corrija o defeito causador da interrupção antes de se religar o equipamento operado. Dentre as várias causas de faltas permanentes têm-se a quebra de um cabo, colisão de um veículo com um poste etc.

Eventualmente, uma falta do tipo temporária pode se transformar em uma falta do tipo permanente caso não haja uma operação adequada dos equipamentos de proteção.

2.3- Sistema de Proteção

A principal finalidade do sistema de proteção é proteger o sistema elétrico contra possíveis faltas, permanentes ou temporárias.

Para que o sistema de proteção cumpra adequadamente com a sua finalidade ele deve:

- Isolar a menor parte possível do sistema no caso de alguma falta, ou isolar o curto-circuito tão próximo quanto possível de sua origem;
- Efetuar o isolamento do curto-circuito em um tempo mínimo a fim de reduzir os danos aos condutores e equipamentos.

Para conseguir alcançar os objetivos para os quais a proteção é projetada um sistema de proteção deve apresentar (ACKERMAN, 2006, ANDERSON, 1999, CPFL, 2003):

a) **Seletividade:** a proteção deve somente isolar a parte do sistema atingida pelo defeito/falta, mantendo a continuidade do serviço das demais partes do sistema. A seletividade determina a coordenação da proteção. Esta pode ser definida como o ato ou efeito de dispor dois ou mais equipamentos de proteção em série segundo certa ordem, de forma a atuarem em uma seqüência de operação preestabelecida. O objetivo da coordenação então é evitar que faltas temporárias causem a operação de dispositivos de proteção que não tenham religamentos automáticos e que, no caso de defeitos permanentes, a menor quantidade possível do sistema fique desligada. O estudo da coordenação e da seletividade é feito pela superposição das curvas características tempo versus corrente dos diversos equipamentos em um gráfico bi-log, com o objetivo de definir as temporizações mais adequadas para cada equipamento.

Premissas da seletividade:

- Solicitação de todas as proteções situadas entre a fonte e o ponto de defeito;
- Somente a proteção mais próxima ao ponto de defeito deve atuar, isolando completamente o componente defeituoso e desligando a menor porção do sistema elétrico. Na Figura 2.2 ilustra-se como deve operar um esquema de proteção seletivo.

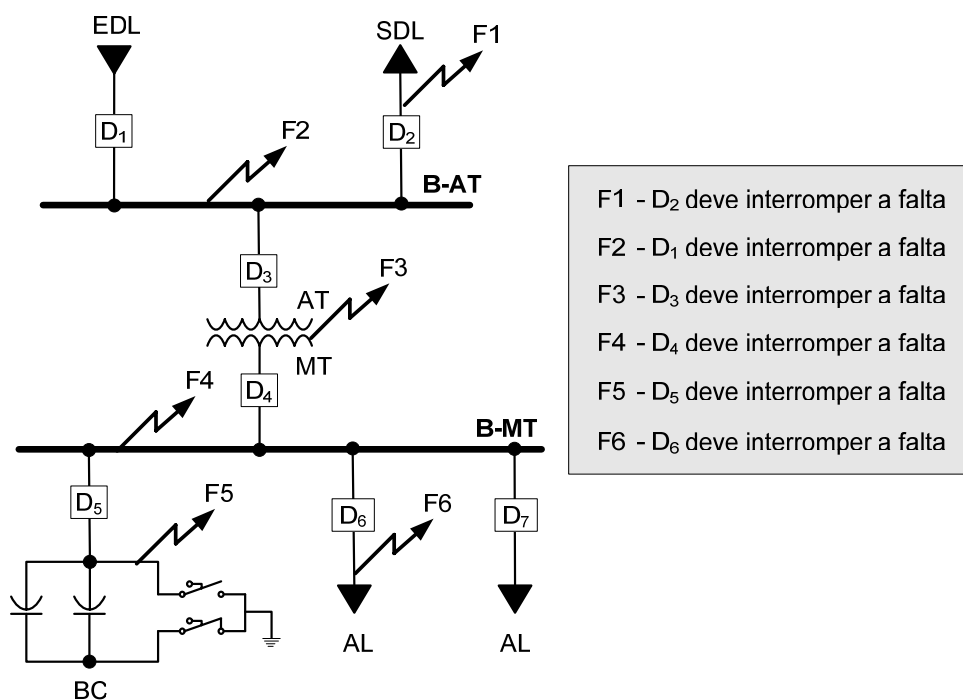


Figura 2.2-Esquema de seletividade da proteção.

b) **Rapidez e velocidade:** capacidade de resposta no menor tempo possível de modo a:

- Assegurar a continuidade do suprimento e a manutenção de condições normais de operação nas partes não afetadas do sistema;
- Auxiliar na manutenção da estabilidade do sistema pela remoção do distúrbio antes que este se espalhe e conduza a uma perda de sincronismo e conseqüentemente ao colapso do sistema de potência;
- Evitar ou reduzir a extensão dos danos ao sistema.

c) **Sensibilidade:** capacidade do sistema de proteção de identificar uma condição anormal que excede um valor limite ou de *pickup* para a qual inicia uma ação de proteção quando as quantidades sentidas excedem o valor limite. A sensibilidade refere-se ao nível mínimo de operação - corrente, tensão, potência, etc. - de relés ou de esquemas de proteção. É a capacidade de resposta dentro de uma faixa esperada de ajuste, ou seja, é a capacidade da proteção responder às anormalidades nas condições de operação, e aos curtos-circuitos para os quais foi projetada.

d) **Confiabilidade:** probabilidade de um componente, equipamento ou sistema funcionar corretamente quando sua atuação for requerida. A confiabilidade baseia-se na certeza

de uma operação correta mediante a ocorrência de uma falta – os dispositivos de proteção devem operar na presença das faltas que estão dentro da zona de proteção; e na *segurança* que mede o grau de certeza de não haver operação indesejada – estes dispositivos não devem operar desnecessariamente para faltas fora da zona de proteção ou na ausência de faltas no sistema.

O sistema de proteção deve ser seguro, ou seja, em caso de defeito ou condição anormal, a proteção nunca deve falhar ou realizar uma operação indevida. A operação incorreta ou intempestiva de um dispositivo pode ser atribuída a: projeto incorreto do sistema de proteção, ajuste incorreto do relé, testes incorretos com os relés, instalação incorreta ou degradação em serviço.

Um sistema de proteção seguro implica em apresentar uma segunda proteção, ou proteção de retaguarda (*backup*) no caso de falha da proteção principal. Na Figura 2.3 ilustram-se as zonas de proteção principal de uma subestação.

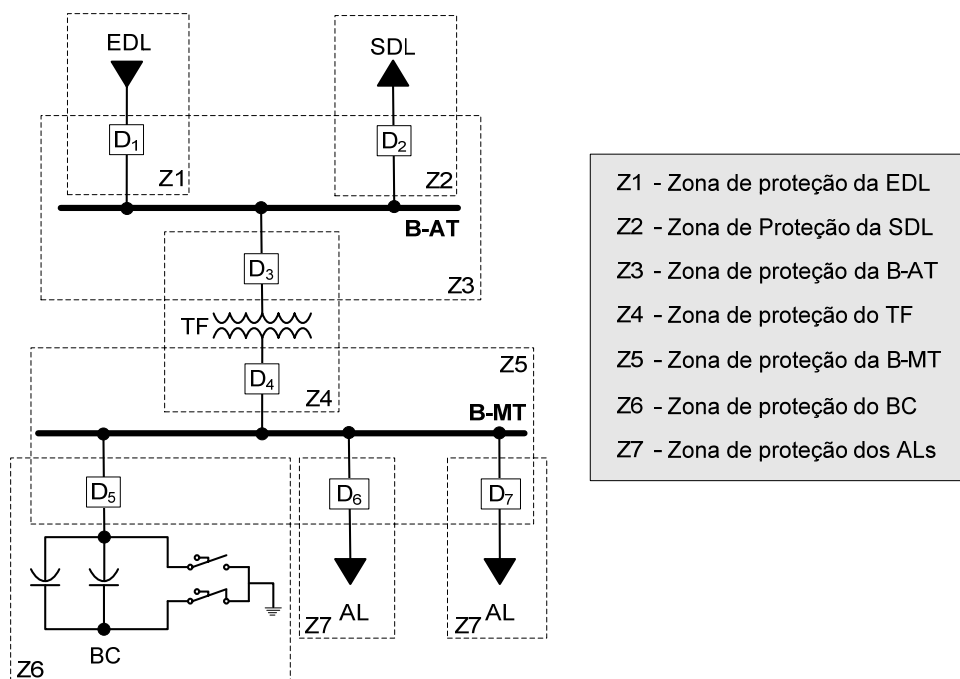


Figura 2.3- Zonas de proteção principal.

A zona de proteção principal é responsável pela proteção principal da seção delimitada por ela. Deste modo, caso ocorra uma falta em uma determinada seção, o dispositivo de proteção responsável pela zona principal será o primeiro a sentir a corrente de curto-circuito e tentar eliminar a falta. Caso a proteção principal falhe, ou seja, estes dispositivos falhem a

proteção secundária ou de retaguarda deve atuar. Na Figura 2.4 é ilustrada a zona de proteção de retaguarda dos alimentadores.

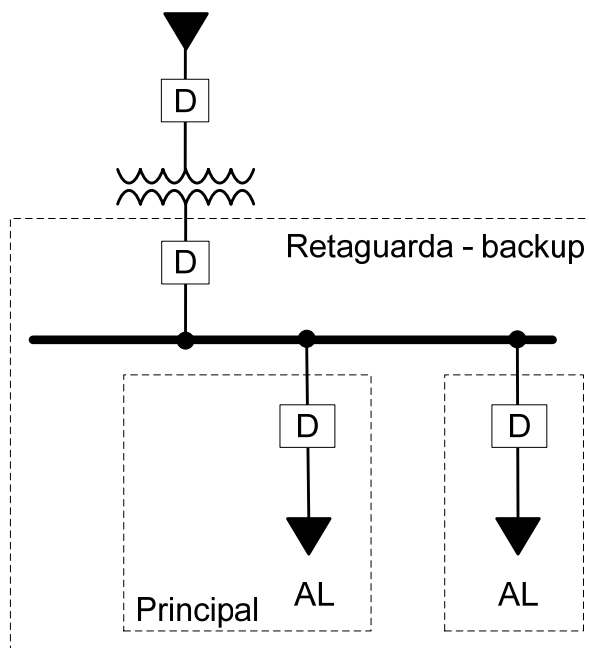


Figura 2.4- Zona de proteção de retaguarda – alimentadores.

e) **Economia:** o sistema de proteção deve ter sua implantação viável economicamente, evitando-se um número excessivo de dispositivos de proteção.

Quando todos ou parte desses requisitos são alcançados a proteção traz benefícios para o sistema elétrico, como: menores danos aos materiais e equipamentos (alimentadores, transformadores, instalações etc); menores custos de manutenção (combustíveis, transportes, mão de obra etc); maior vida útil dos materiais e equipamentos; menores números de desligamentos e consumidores atingidos por faltas no sistema; maior segurança e melhoria da imagem da empresa junto aos consumidores.

2.4 -Automação de Subestações

A **confiabilidade e a sustentabilidade** do fornecimento da energia estão relacionadas com a qualidade do produto determinada, principalmente, através da disponibilidade da energia elétrica nos pontos de consumo. Essas duas premissas são alcançadas através de:

a) Projeto e implementação coerente de um sistema de proteção robusto;

b) Investimento e caracterização de um sistema de automação capaz de manipular grande quantidade de informação de forma segura e confiável em tempo real, advinda do sistema de proteção e controle (ACKERMAN,2006, JARDINI, 1996, McDONALD, 2003).

A caracterização de um sistema de automação adequado é conseguida, basicamente, através da (ACKERMAN,2006, LIYANAGE, 2008, McDONALD, 2003).:

- Instalação de equipamentos de aquisição de dados e dispositivos remotamente controlados;
- Capacidade de processamento de informação e suporte de funções de decisão.

A *instalação de equipamentos de aquisição de dados* é traduzida pela implementação de um sistema SCADA (*Supervisory Control and Data Acquisition*). O sistema SCADA pertence ao DMS (*Distribution Management System*) e tem a função de conectar dois ambientes distintivamente diferentes: a subestação, onde ficam os equipamentos de medida, monitoramento e controle digitais e o centro de operação, onde são coletados, armazenados, mostrados e processados os dados (GAUSHELL ; BLOCK, 1993).

A *instalação de dispositivos remotamente controlados* ou IEDs (*Intelligent Electronic Devices*) alocados de forma independente, ou seja, conectados diretamente no sistema SCADA ou controlados através de RTUs (*Remote Terminal Unit*), ajudam a reduzir o tempo de chaveamento e os custos em relação à operação manual. Poucas operações manuais também contribuem para redução de possíveis erros humanos. Também, as instalações de equipamentos de aquisição de dados e dispositivos remotamente controlados permitem um monitoramento mais efetivo da rede.

A *capacidade e eficiência do processamento de informação* são alcançadas através da perfeita integração entre os dispositivos presentes na subestação. Essa integração é consolidada através da arquitetura da rede de comunicação empregada e da tradução dos protocolos utilizados por cada dispositivo conectado à rede.

As *funções de decisão* (ACKERMAN, 2006, McDONALD, 2003) estão relacionadas à implementação de softwares nas subestações de maneira a auxiliar os engenheiros e operadores do sistema. Estes softwares são desenvolvidos com o objetivo de realizar de maneira total ou parcial várias funções, entre elas, o enfoque que está sendo proposto, a **detecção de faltas**.

A detecção de faltas está relacionada com a detecção de defeitos nos componentes do sistema elétrico e consiste da *análise e interpretação dos alarmes* devido a esses defeitos. A

detecção pode ser efetuada de forma local ou centralizada. Detecção de faltas efetuada de forma local está limitada a ambientes como subestações e usinas geradoras de energia elétrica e tem como objetivo fornecer diagnóstico de faltas aos operadores e centros de controle destas unidades específicas.

2.4.1-Sistema SCADA

Os sistemas de automação tipicamente têm uma ou mais conexões para o exterior do sistema. Conexões de comunicações comuns incluem centros de operação, escritórios de manutenção e centros de engenharia. A maioria dos sistemas de automação conecta-se através de um sistema SCADA tradicional servindo às necessidades de operação em tempo real da subestação que é controlada através dos centros de operações por meio de uma rede de comunicação (ACKERMAN,2006).

De fato, o sistema SCADA básico interconecta dois ambientes distintos: a subestação e o centro de operações. Interfaces para equipamentos da subestação e fontes de conversões e comunicações completam o sistema. O ponto terminal para um sistema SCADA tradicional é uma RTU onde ocorre a interface entre a comunicação e os equipamentos da subestação.

As RTUs coletam medidas do sistema elétrico, transmitem essas medidas para um centro de operações, onde o SCADA central apresenta-as aos operadores através de uma IHM (Interface Homem-Máquina). A IHM permite ao operador monitorar e controlar todos os parâmetros do sistema elétrico em tempo real. Isso é feito utilizando programas dedicados implementados em plataformas gráficas, tipo Windows ou Linux. .

De maneira geral uma IHM inclui as seguintes funções principais:

- *Controle de acesso:* diferentes níveis de acesso podem ser definidos para diferentes grupos de funcionários. Por exemplo, os operadores devem ter completo acesso ao display e as funções de controles enquanto o pessoal da manutenção tem acesso restrito.
- *Mapeamento:* mapa gráfico em duas dimensões de representação do sistema de potência dividido em várias camadas. Essas camadas podem conter diferentes representações do sistema. Por exemplo, o nível 1 mostra o sistema elétrico inteiro, o nível 2 a subestação, o nível 3 um resumo do estado dos alimentadores principais do sistema etc.
- *Displays tabulares:* lista os dados referentes aos dispositivos da subestação. Por exemplo, um display tabular pode listar todos as RTUs da subestação e seus estados em/fora de serviço, valores de potência, relações de TCs e TPs etc.

- *Displays de tendência:* mostra graficamente a variação no tempo dos dados coletados no sistema. Os dados podem ser escolhidos pelo operador.
- *Display de alarmes:* funciona equivalente a um anunciador de alarmes, mostrando todos os alarmes informados pelo sistema, sejam alarmes da proteção, falha de comunicação, abertura e fechamento de disjuntores etc.

Geralmente, essas medidas são fluxos de potência ativa e reativa (watt e var), tensões e correntes. Entretanto, outras medidas do tipo: níveis de tanques de óleo, pressão, posições do tap e ventilação dos transformadores, são comuns. Estas medidas possuem características analógicas e, portanto, são atualizados periodicamente de tal modo que o operador possa ser assegurado de que o dado que aparece na tela do monitor é real.

O sistema SCADA central monitora a entrada de fluxo de variáveis analógicas e bandeirola (destaca) com *warnings* e alarmes os valores de grandezas elétricas que estão fora dos limites preestabelecidos para alertar o operador do sistema. Este sistema quase sempre fornece meios para os operadores do sistema controlarem os equipamentos da subestação. Isto inclui: disjuntores, chaves seccionadoras, ventilação e taps de transformadores.

As medidas constituem uma ponte que leva grandezas do sistema físico para a tela dos monitores nos centros de operações. Assim os sistemas de medidas transformam o mundo físico em um mundo digital e constituem um aspecto crítico do sistema SCADA e do sistema de automação.

A principal função do sistema SCADA ou sistema de automação é medir as atividades dos sistemas de energia, processando as medidas e relatando os dados aos centros de operação. A fonte dos dados medidos são TPs e TCs.

Os valores de tensão e corrente medidos pelos TPs e TCs, respectivamente, são injetados em dispositivos conhecidos como IEDs. Estes possuem recursos valiosos capazes de disponibilizar informações importantes sobre o sistema, seja ela operacional ou não operacional, necessárias a muitos grupos de usuários dentro da concessionária de energia em um ambiente integrado. O desafio das concessionárias é então determinar um padrão de integração que possa vir de encontro com suas necessidades específicas.

IEDs são definidos como qualquer dispositivo incorporando um ou mais processadores com capacidade de receber ou transmitir dados/controles de/ou para uma fonte externa (medidores multifuncionais eletrônicos, relés digitais, CLPs - Controladores Lógico

Programáveis etc) (McDONALD, 2003). Na Figura 2.5 ilustra-se um esquema de ligação de TPs/TCs e IEDs.

Com a ascensão do uso de IEDs e o desenvolvimento de RTUs mais modernas a tendência destas é não mais ser dispositivos de medição. Em sistemas de automação atuais as RTUs podem executar a função de coleta de dados de IEDs.

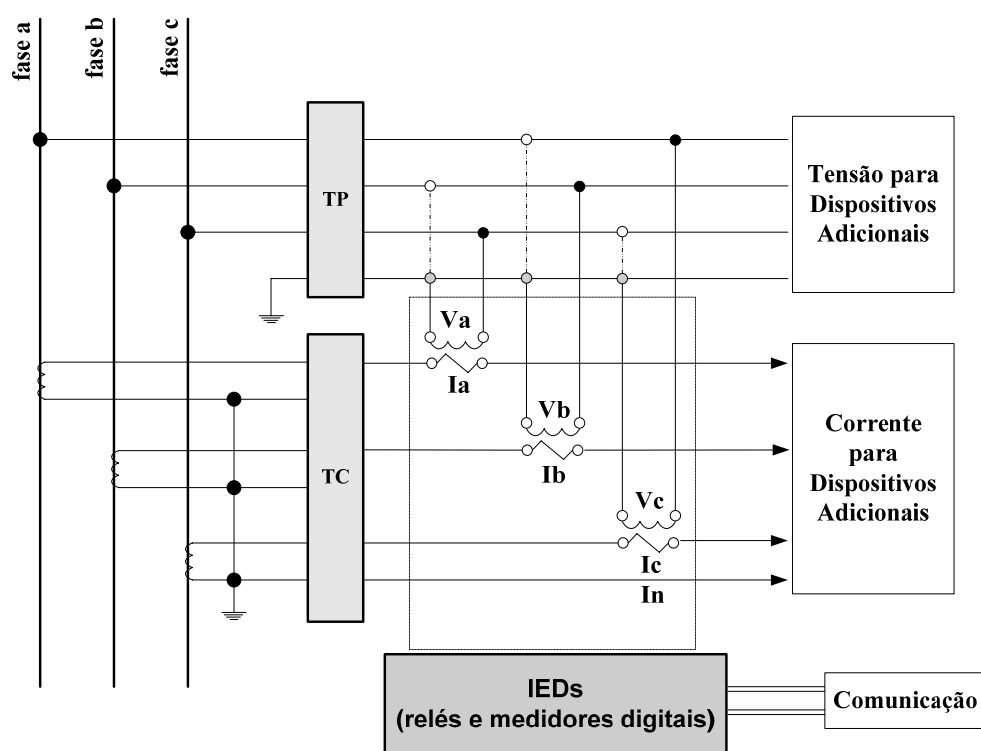


Figura 2.5- Ligações - TPs/TCs e IEDs (ACKERMAN,2006) e (McDONALD, 2003).

Do mesmo modo que a RTU conecta-se diretamente ao sistema SCADA, os IEDs também o podem fazer. Deste modo, os comandos de controle dos centros de operação podem passar através da RTU ou diretamente para a rede da subestação até alcançar os IEDs configurados para controlar um determinado dispositivo. Usualmente, distribuindo o controle e o monitoramento através de múltiplos dispositivos, requer da concessionária de energia condições de trabalho que exigem cooperação entre operação da subestação, manutenção e proteção.

A arquitetura dos Sistemas SCADA modernos têm tanto software como hardware com arquiteturas distribuídas. Nestas o processamento de dados é distribuído entre vários

computadores e servidores que se comunicam entre si através de uma rede LAN (Local Area Network) dedicada (ACKERMAN,2006).

Outra característica importante dos sistemas SCADA modernos é sua arquitetura aberta. Esta característica permite interconectividade com outros sistemas. Sistemas abertos são importantes porque permitem a possibilidade de interfaceamento com produtos de outros vendedores. Apesar do fato da maioria dos vendedores oferecerem sistemas abertos às concessionárias, eles ainda desenvolvem suas próprias IPAs (Interface de Programação de Aplicação) instaladas no computador central que contém a IHM. A IPA possui módulos dedicados de programas capazes de se comunicar utilizando objetos comuns e mecanismos de troca de dados. Na Figura 2.6 mostra-se a arquitetura distribuída de um sistema SCADA típico.

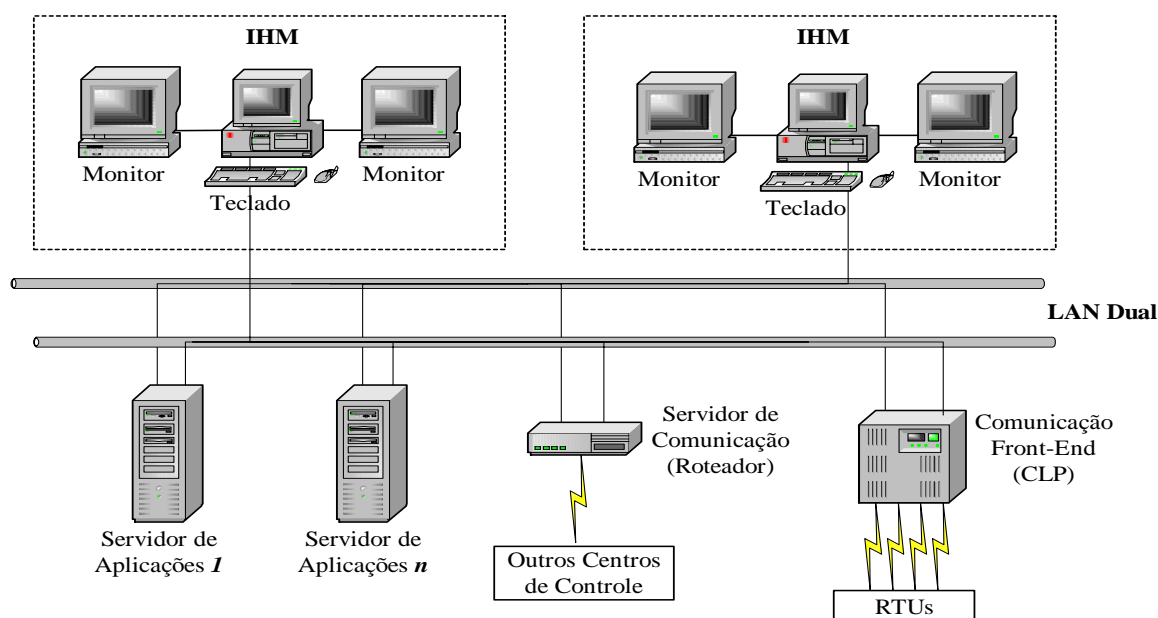


Figura 2.6-Sistema SCADA típico (ACKERMAN,2006).

Os servidores de aplicações têm a característica de possuir grande capacidade de processamento. Assim eles podem comportar os seguintes subsistemas:

- *Núcleo SCADA*: servidor usado principalmente para funções de processamento de dados.
- *Base de Dados*: suporta a base de dados de todo o sistema.

- *Aplicações avançadas:* suporta todas as aplicações do DMS. A principal característica desse servidor é a alta capacidade de processamento.
- *Base de dados histórica:* suporta a base de dados que contém todos os dados históricos. As informações provenientes desses dados podem ser usadas para estudos futuros e treinamento dos operadores.
- *Configuração e administração:* controle, gerenciamento e manutenção do sistema SCADA. Este servidor pode gerenciar o modo de operação de cada servidor e controlar funções como *backup* do sistema.

A comunicação *front-end*, geralmente concebida utilizando CLPs tem a função de aquisição de dados das RTUs e dispositivos de campo, como relés, chaves seccionadoras, contadores, disjuntores etc. Os CLPs executam funções tais como automatização de chaveamentos, intertravamentos e segurança.

O servidor de comunicação ou roteador possibilita a troca de dados ou arquivos em tempo real com outros centros de controle.

Além das considerações anteriores, deve-se atentar para o fato de que o sistema mostrado na Figura 2.6 possui uma rede LAN dual composta de duas redes funcionando paralelamente. Este tipo de configuração é utilizado para assegurar ao sistema SCADA alta disponibilidade, operação contínua e flexibilidade na manutenção. Assim se um computador ou uma rede falhar todas as comunicações serão chaveadas para o computador e a rede que estão em correto funcionamento.

Conclusão

Neste capítulo foram apresentadas teorias relacionadas aos projetos de controle e proteção de subestações de distribuição de energia elétrica e assuntos relacionados a automação de um sistema de subestações de distribuição de energia elétrica, nos quais fornecem toda a base para a compreensão e motivação do desenvolvimento de um sistema para detecção de faltas proposto neste trabalho.

Capítulo 3

Revisão Bibliográfica em Detecção de Faltas

Ao longo do tempo, devido às diversas perturbações a que estão sujeitas as subestações, tem havido a necessidade de mudanças nos sistemas de distribuição de energia, para que tornem-se cada vez mais eficientes. Com este objetivo, são vários os trabalhos que estudam procedimentos eficazes, para a detecção de faltas em tempo mínimo, para que não causem danos a equipamentos e aos usuários do sistema.

Neste capítulo, comenta-se alguns exemplos de sistemas que tem sido propostos para solucionar estes problemas, melhorando o desempenho dos sistemas de distribuição de energia. Inclusive o artigo no qual se baseia este trabalho.

3.1 Trabalhos Publicados para a Detecção de Faltas em SE

São vários os trabalhos publicados com proposta de minimizar os tempos de interrupção dos serviços por faltas permanentes nos sistemas de potência. Diferem principalmente, na metodologia empregada em cada um, como Rede de Petri Colorida, propõe, portanto uma

solução por software, outros trabalham com Inteligência Artificial, Redes Neurais e Lógica Nebulosa, também soluções por software, todos se utilizam de informações e recursos de sistemas de aquisição de dados, disponibilizados na maioria das concessionárias de geração e distribuição de energia elétrica, atualmente. E ainda algumas soluções por hardware como as de Delgado et al (PERLAZA; DELGADO, 2003).

Santos et al (SANTOS et al, 2007), publicou recentemente o trabalho sob o título, Sistema de Diagnóstico de Falhas para Sistemas Elétricos baseado em Redes de Petri Colorida. Neste trabalho é proposto um *Sistema de Diagnóstico de Falhas baseado em Redes de Petri Coloridas*. O objetivo deste sistema é o de simplificar o trabalho do operador dos sistemas de potência, apresentando um diagnóstico rápido e preciso para eventos ocorridos em subestações e rede elétrica. O sistema de diagnóstico de falhas possui três camadas, uma interface de entrada que filtra os dados advindos do SCADA e os converte em marcação inicial para a rede de Petri colorida localizada na segunda camada. A saída da rede de Petri é então interpretada na terceira camada, resultando em um sucinto relatório que contém a localização da falta, o tipo de falta, a(s) fase(s) em falta, e se houve falha de relés e disjuntores no processo de eliminação da falta.

Neste sistema foram utilizados dados de uma ocorrência real das SEs de Cariré e Araras, desde as informações do SCADA até a emissão do diagnóstico, como forma de demonstrar o funcionamento correto. Mostra-se na figura 3.1, o modelo da rede de Petri desenvolvida por Santos.

Sampaio propôs uma metodologia distribuída para implementação de Sistemas de Diagnóstico de Falta (SDF) para uma Subestação (SE), baseado em Redes de Petri Coloridas Hierárquicas (RPC-H). O SDF proposto visou filtrar as informações que chegam aos operadores do sistema elétrico de potência, tornando a tarefa de diagnóstico de falta menos estressante, mais rápida e menos susceptível a erros. O modelo desenvolvido foi aplicado na SE digitalizada de Beberibe que é parte integrante do sistema elétrico de potência da Companhia Energética do Ceará – COELCE. Mostra-se na figura 3.2 uma parte do sistema em Redes de Petri Hierárquica.

Este sistema, segundo Sampaio, permite informar ocorrências de faltas mais rápidas e precisas. Com essa rapidez os clientes contam com um melhor atendimento e maior satisfação em obter uma energia segura constantemente e torna mais fácil executar a tarefa de diagnóstico de faltas pelos operadores.

Sampaio conclui seu trabalho, afirmando ser esta técnica realmente adequada para a modelagem, análise e validação de sistemas de diagnósticos de falta, baseado nos testes realizados na SE BBR¹ que apresentou diagnósticos corretos em todas as faltas que foram simuladas.

¹ Código da SE de Beberibe

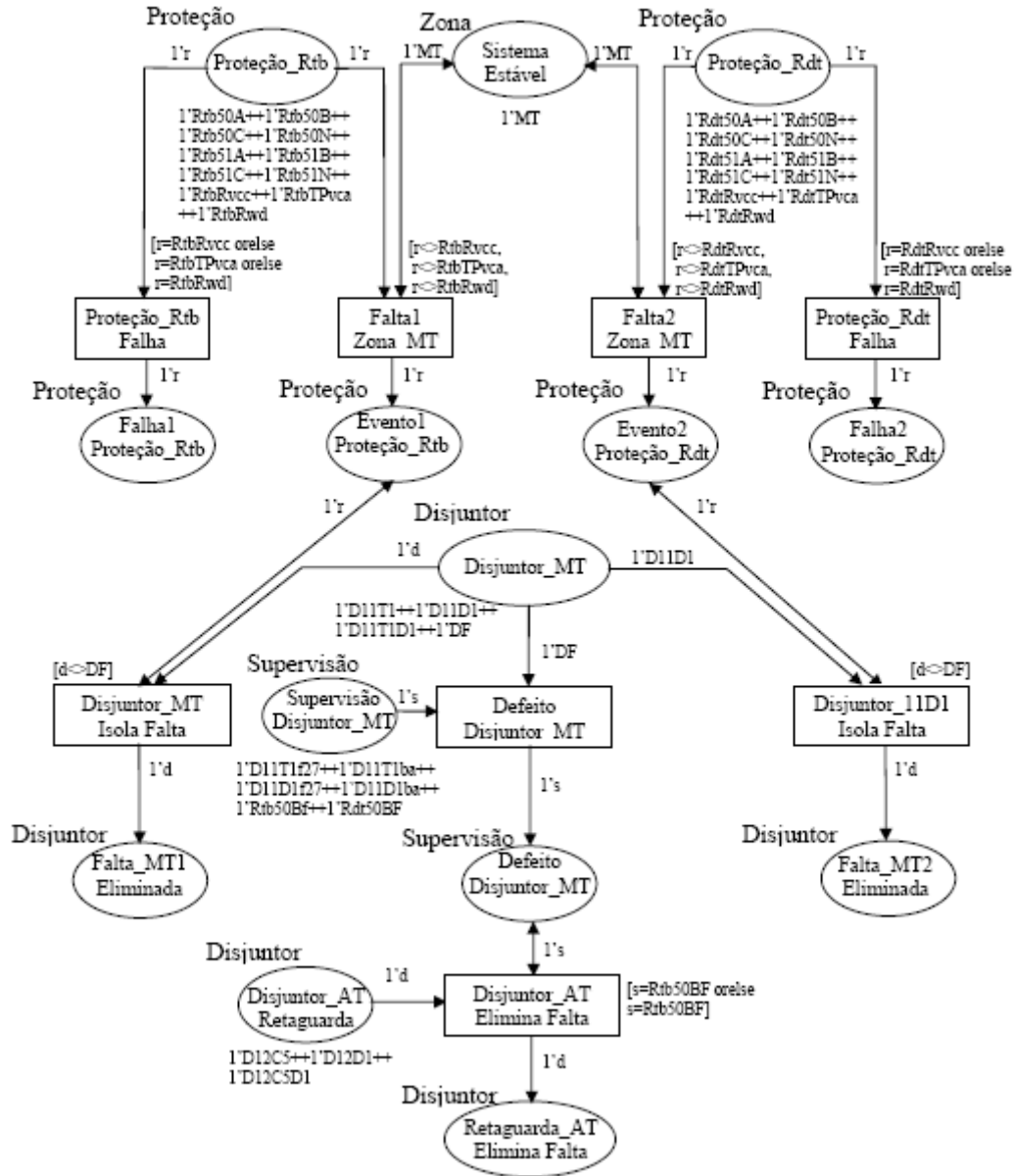


Figura 3.2 – Sistema de Detecção de Falhas utilizando Redes de Petri Hierárquica (SAMPAIO, 2002).

Usando Lógica Nebulosa ou *Fuzzy*, Decanini (DECANINI, 2008) desenvolveu um modelo para a *Detecção e classificação de falhas de Curto-circuito em Sistemas de Distribuição de Energia Elétrica*.

Decanini desenvolveu algoritmos inteligentes modulares, visando executar o diagnóstico de defeitos de forma robusta e rápida, sem a necessidade do operador conhecer as estruturas a serem usadas. Os módulos terão a capacidade de detectar e classificar falhas de curtos-circuitos, fornecendo o tipo e o nível de impedância do defeito. Para isso, usa as

informações de corrente elétrica nas três fases. As entradas são os dados de correntes elétricas trifásicas pré-processadas que indicam na saída, o diagnóstico de falta. Este pré-processamento consiste em gerar grandezas normalizadas e expressas de forma relativa entre as três fases. Todo este procedimento visa a extração das características dos defeitos de curto circuito.

Foram monitorados neste projeto os pontos mais relevantes em uma SE tais como , o alimentador, o transformador , e ramais . Resumindo este projeto em diagrama de blocos tem-se a figura 3.3. Na figura 3.4 mostra-se o diagrama unifilar que foi simulado.

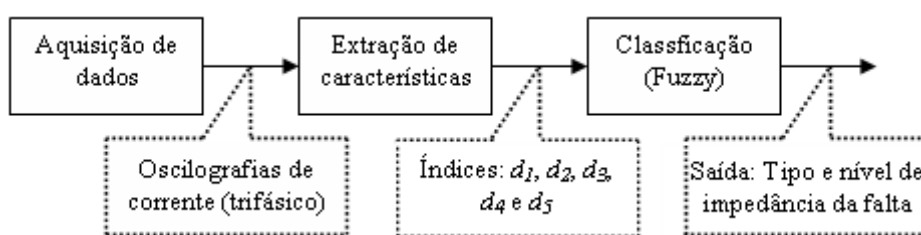


Figura 3.3 – Diagrama de Blocos do Sistema de Detecção (DECANINI, 2008).

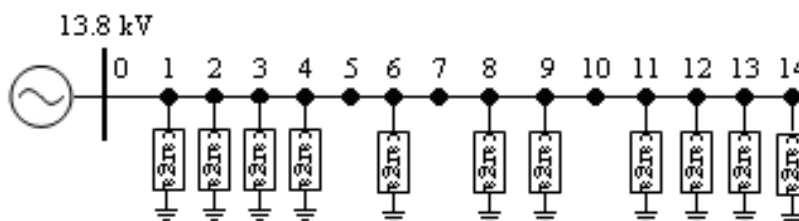


Figura 3.4 – Diagrama unifilar do alimentador simulado (DECANINI, 2008).

Este sistema utilizou o software ATP (*Alternative Transients Program*) para simular as oscilações das três fases e obter os dados necessários para a avaliação de seu projeto. Dos resultados obtidos, Decanini concluiu que, o seu trabalho aplicando a lógica *fuzzy* na detecção de faltas, teve pleno êxito e grande rapidez nas respostas.

Em Neto, foi desenvolvido um *Sistema para Detecção de Faltas de Alta impedância e de Rompimento de Condutores em Redes de Distribuição de Energia Elétrica* (PENTEADO NETO, 2005), para auxiliar o Sistema de Distribuição e a tomada de decisões. Neste trabalho

foram desenvolvidos dois sistemas para detecção do rompimento dos condutores. O primeiro sistema foi baseado na aplicação de sensores de corrente elétrica e o outro na aplicação da técnica de extensometria. O segundo sistema é um sistema inédito e recebeu inclusive uma patente junto ao INPI.

Neste projeto foi desenvolvida uma rede experimental de distribuição de energia onde as características físicas e elétricas são próximas de uma rede convencional. Observa-se na figura 3.5 os sensores aplicados nas três fases.

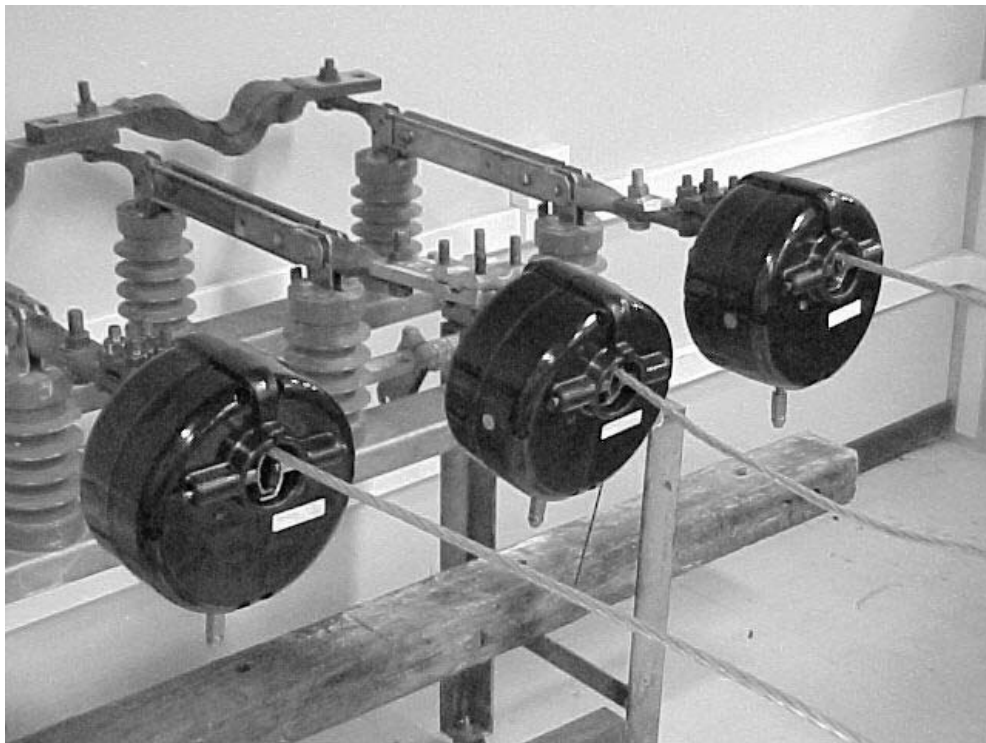


Figura 3.5 – Testes em bancada com sensores aplicados nas três fases (PENTEADO NETO, 2005).

O sistema proposto por Neto fundamenta-se no fato de que, na ocorrência do rompimento de um condutor instalado em uma rede de distribuição de energia, a tração mecânica no condutor rompido irá se alterar. Esta alteração é detectada e comunicada remotamente permitindo ações por parte do operador da rede.

Para detectar a alteração da carga mecânica, foram desenvolvidos os sensores, fundamentados na técnica de extensometria.

Na conclusão de seu trabalho, Neto mostra que este sistema apresenta a sensibilidade suficiente para detecção da mudança da tensão mecânica do cabo ao ser rompido.

O trabalho de Moreto (MORETO, 2005) trata da **Localização de Faltas de Alta Impedância em Sistemas de Distribuição de Energia Elétrica**, onde se baseia na metodologia de Redes Neurais Artificiais. O sistema desenvolvido é capaz de obter uma estimativa precisa da localização tanto de faltas sólidas e lineares, quanto de faltas de alta impedância. O algoritmo implementado para este sistema está baseado em Redes Neurais (RNAs), e pode ser adicionado como uma função extra de um relé de proteção digital. O diagrama de blocos do sistema proposto por Moreto pode ser visto na figura 3.6. A figura 3.7 permite observar o diagrama de blocos e a localização de faltas usando redes neurais.

Ainda segundo Moreto, o sistema desenvolvido por ele apresenta resultados satisfatórios, pois fornece uma precisa estimativa da localização de faltas de alta impedância em alimentadores de distribuição.

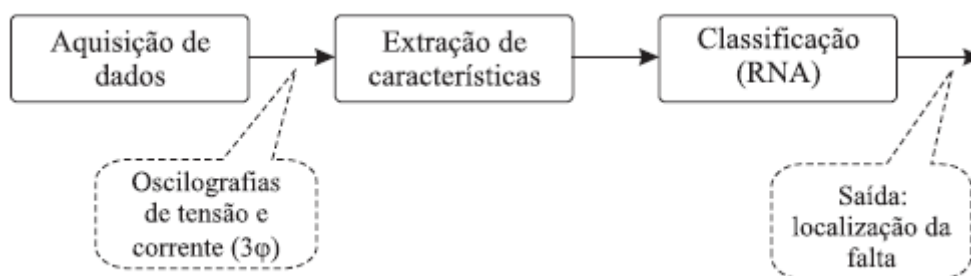


Figura 3.6 – Diagrama de Blocos Simplificado (MORETO, 2005).

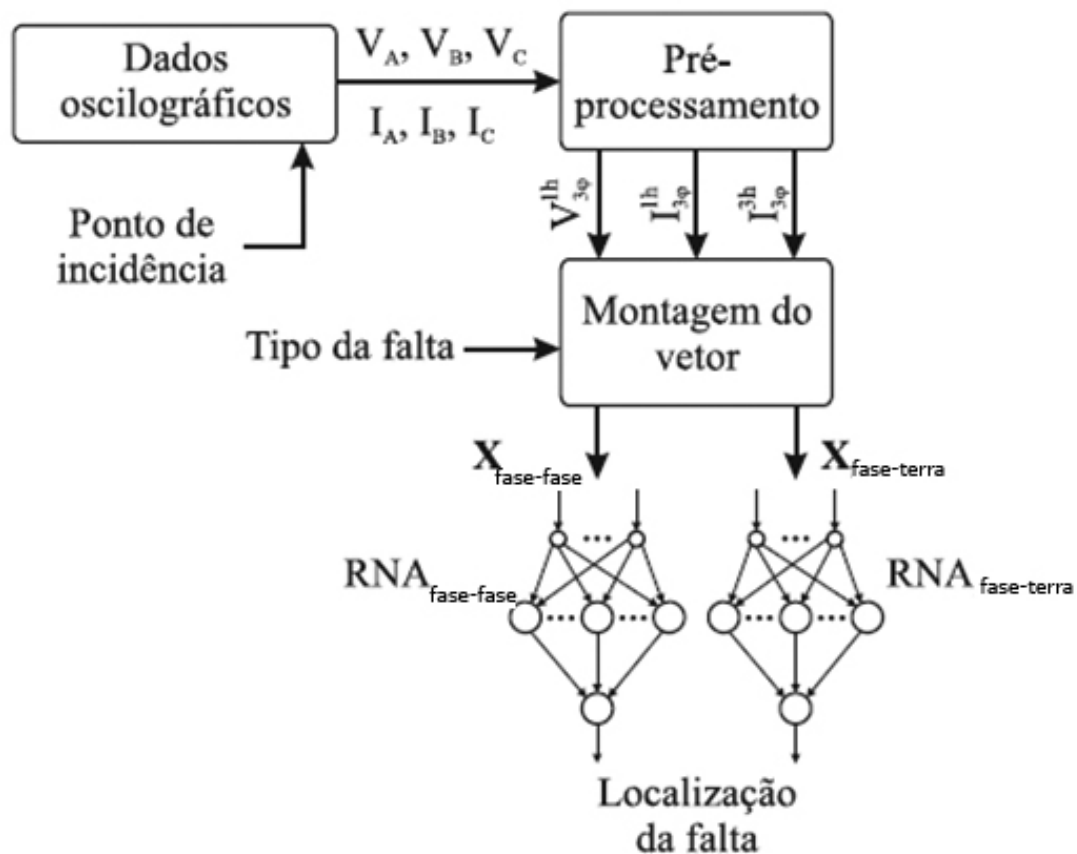


Figura 3.7 – Diagrama de Blocos com a localização de faltas usando Redes Neurais (MORETO, 2005).

Finalmente, apresenta-se como último trabalho nesta revisão de literatura a *Deteccção de falhas em Sistema de Potência com Chip DNA em um FPGA*, (PERLAZA ; DELGADO, 2003), artigo do qual originou o presente trabalho sobre a detecção de faltas em Sistemas de Distribuição de Energia Elétrica.

O sistema de detecção de falhas proposto por Delgado e Perlaza, tem como principal diferença entre todos os propostos , uma solução predominantemente realizada em hardware , e para isso baseia-se no conceito da biologia celular , ou seja o DNA. Este conceito foi usado para emular um chip de DNA eletronicamente, em um dispositivo lógico programável, por exemplo, FPGA.

No trabalho são usados dados da SE Tunal localizada na cidade de Bogotá – Colômbia. Com o objetivo de enfocar o problema da detecção de faltas, Delgado trata dois grupos de sinais que permitem conhecer o estado do sistema de potência, são os sinais analógicos e os sinais digitais. Se baseia nos sinais analógicos de corrente trifásica, tensão trifásica, corrente

do neutro e tensão residual, obtidos respectivamente dos transformadores de corrente e tensão, computando um total de 8 sinais para serem tratados e analisados. Os sinais digitais são obtidos dos relés de proteção que podem ser analógicos ou digitais, conectados por meio de transdutores de tensão e/ou corrente ao sistema de potência para detectar condições não desejadas dentro de uma área específica; dos interruptores e dos sinais de tele proteção que somados quantificam um total de 16 sinais .

Usa conversor A/D com resolução de 8 bits. Para cada conversão realizada tem-se um total de 64bits (8x8) de informação, que juntos com os 16 sinais digitais formam um registro paralelo de pré-faltas de 80bits.

São constituídas informações de possíveis faltas individuais e combinadas para formar um banco de dados de 300 faltas. A comparação entre o registro de 80 bits e um registro de faltas é feito através da porta lógica XOR, conforme figura 3.8, e uma porta lógica AND que detecta quando as informações são complementares, ou seja , acionando um alarme ou led avisando da ocorrência de uma falta e a sua localização. É um projeto que possui restrições e tem sua validade comprovada por simulações.

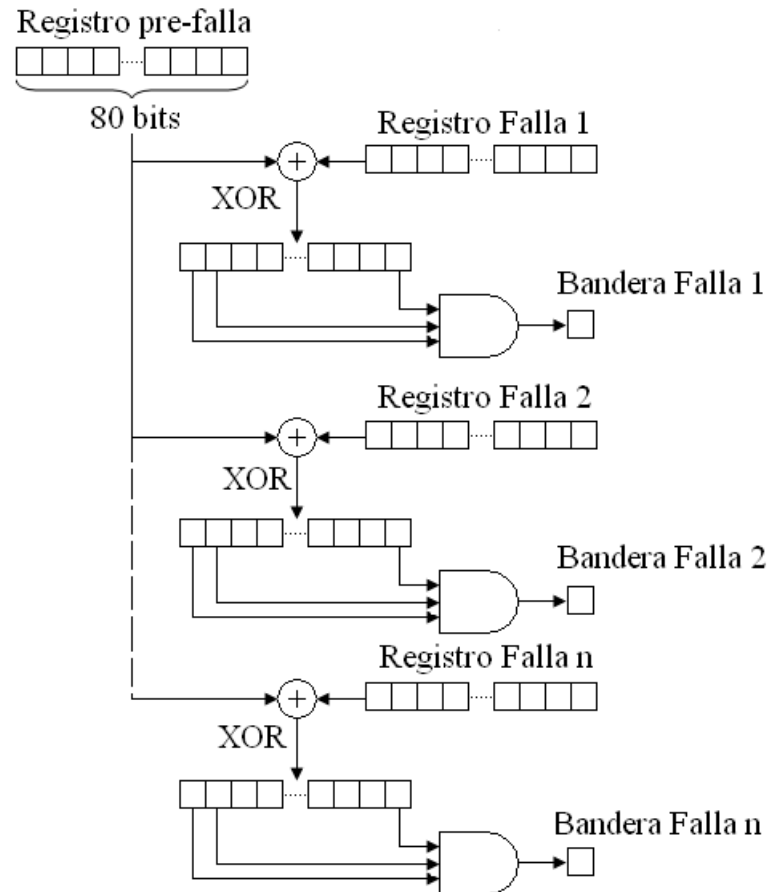


Figura 3.8 – Emulação Eletrônica baseada no conceito de DNA (PERLAZA ; DELGADO 2003).

No projeto foi utilizado o FPGA xc2s50-6tq144 da família Spartan II de XILINX. Como esta família possui poucos pinos de I/O, foi feita uma codificação das 'bandeiras' em um vetor de 9bits, para cobrir as 300 comparações a que se propôs.

Para Delgado, o trabalho apresenta vantagens quando se trata de detecção de faltas, pois o sistema apresenta um processamento paralelo e pode manusear um grande número de variáveis. As desvantagens apontadas pelo autor são o alto custo da tecnologia que tem baixado conforme a demanda, rigorosas condições de funcionamento e grandes tempos de operação do sistema.

3.2 Conclusão

Ao longo do tempo, as pesquisas desenvolvidas para diagnosticar faltas em sistemas de distribuição de energia vêm crescendo rapidamente. Foram citados apenas alguns exemplos de trabalhos desenvolvidos e publicados nos quais buscam um único objetivo, ou seja,

proporcionar tomadas de decisões rápidas que são realizadas pelos operadores e engenheiros, assegurando aos usuários do sistema uma alta confiabilidade.

Capítulo 4

Dispositivos Lógicos Programáveis – Tecnologia

Este capítulo apresenta uma descrição sobre a evolução, em termos de tecnologia, dos dispositivos lógicos programáveis, tais como os reconfiguráveis PLAs, FPGAs e CPLDs. Serão abordados blocos funcionais e características mais comuns, arquitetura dos blocos, tecnologia usada em sua programação, ferramentas de auxílio de projeto, alguns fabricantes e áreas de aplicação, que são informações importantes para situar e fornecer subsídios do cenário dos dispositivos programáveis. Complementando este capítulo, apresenta-se as principais características do dispositivo programável que está sendo utilizado neste trabalho, *Stratix III E EP3SE50* da Altera.

4.1- Evolução Tecnológica

Tanto em termos de *software* quanto de *hardware*, a lógica reconfigurável trouxe novos paradigmas aos modelos computacionais atuais. A busca de tecnologias pelas empresas visando aumentar a produção e a qualidade, diminuindo o tempo e o custo final do produto

para o mercado de eletro-eletrônicos em contínua expansão, são alguns dos fatores que mudaram o cenário dos ambientes de projeto de sistemas digitais e como consequência mudaram o perfil dos profissionais que trabalham nesta área. Atualmente, os projetistas de sistemas computacionais dedicados, devem possuir conhecimentos múltiplos (de *hardware* e *software*), que englobam desde a arquitetura de computadores até o desempenho de algoritmos de processamento digital de sinais.

No mercado de componentes atual, encontram-se dispositivos programáveis com alta densidade e custos relativamente baixos para os padrões industriais. Empresas que fabricam estes dispositivos estão desenvolvendo ou comprando processadores e incorporando-os a outros periféricos, criando sistemas em uma única pastilha (*system-on-chip*). Outras estão eliminando o processador e o software juntos, criando alternativas baseadas apenas em hardware. 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.

Para a realização de um projeto, tomando-se como base as suas especificações, determina-se o dispositivo lógico programável mais adequado, em função do número de portas, número de entradas e saídas e quantidades de memória, necessárias para a sua implementação. Atualmente, há disponibilidade de dispositivos que permitem a implementação de memórias, multiplexadores, máquinas de estados, etc., em um único componente. Os circuitos podem ser modelados e remodelados através de entradas de esquemático e/ou uma linguagem de descrição de hardware, como o VHDL.

São considerados circuitos integrados programáveis todo o dispositivo de propósito geral que pode ter sua função configurada mediante um programa. Depois de fabricados em um estado genérico, o usuário pode adaptá-lo em uma determinada função

A tecnologia dos circuitos lógicos programáveis começou com os dispositivos microprocessadores e memórias. O primeiro dispositivo lógico programável largamente utilizado foi a memória PROM (Programmable Read - Only Memory), que é um dispositivo programável somente uma única vez. Depois disso, tecnologias mais modernas disponibilizaram memórias do tipo EPROM, EEPROM, Flash EEPROM que permitem ao usuário apagar e reprogramar mais de uma vez. As tecnologias EEPROM e flash possuem as mesmas vantagens relacionadas à facilidade de programação e re-programação, normalmente incluindo suporte à gravação "*in-circuit*". Neste ponto, estes dispositivos assemelham-se bastante aos microcontroladores, alguns suportando inclusive as interfaces JTAG (*Joint Test*

Action Group - Padrão IEEE 1149.1 para um interface de 4 pinos para testes de circuitos integrados).

Tecnologias mais recentes e baseadas em uma PROM, como os dispositivos lógicos programáveis (*Programmable Logic Devices* – PLDs), trouxeram um avanço importante para o desenvolvimento de sistemas eletrônicos digitais, detalhados a seguir (MANTOVANI; OLIVEIRA, 2003).

4.2-Dispositivos Lógicos Programáveis (PLDs- *Programmable Logic Device*)

Os dispositivos lógicos programáveis- PLDs, foram construídos originalmente para implementar pequenos circuitos lógicos, como as funções lógicas de 2 níveis. Possuem internamente um conjunto de portas AND-OR que geram termos produtos da entrada do sistema para o conjunto de portas OR, que são as saídas do sistema (VÉSTIA, 2006). Entre eles destacam-se PROMs , PLAs , FPGAs e CPLDs.

4.2.1-Dispositivos PROM

Na PROM **Programmable ROM**, primeiro dispositivo programável pelo usuário, as entradas são as linhas de endereço, e as saídas são as linhas de dados. As PROMS funcionavam como look-tables, sendo capazes de implementar qualquer tipo de função combinacional. Apresenta dois planos: plano de ANDs fixos e um plano de ORs programáveis. São não voláteis, mas não podem ser reprogramáveis. Uma estrutura típica para uma PROM constituída de 8 palavras de 3 bits (onde “×” representa uma conexão por fusível). é mostrada na figura 4.1.

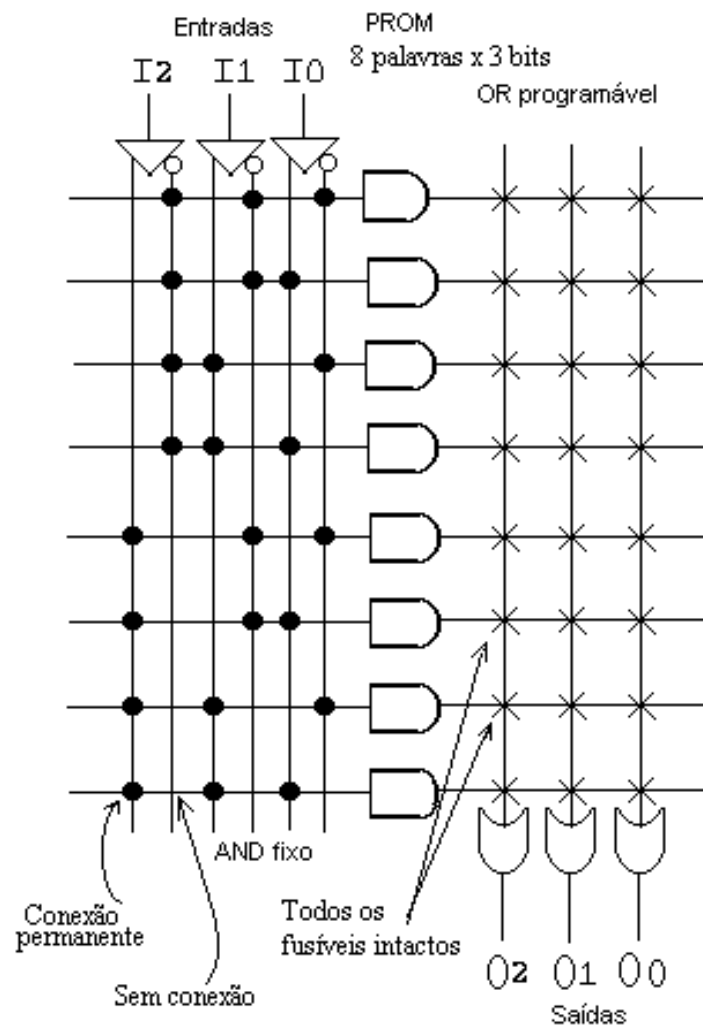


Figura 4.1 Estrutura de uma PROM típica

4.2.2-Dispositivos PLA

A estrutura **PLA** é similar às PROM's, consiste de dois planos AND-OR. Contudo, os dois planos são programáveis na PLA, permitindo que o usuário possa configurar a combinação das variáveis de entrada que irão formar cada termo produto; e no plano OR, qualquer soma com estes termos produtos. Pode ser encontrada nas versões mascarável (programada pelo fabricante) ou programadas pelo usuário (programáveis em campo, ou FPLAs). A PLA mostrada na figura 4.2 tem 3 entradas I_2 , I_1 , I_0 , fornecendo 8 produtos canônicos e 3 saídas O_2 , O_1 , O_0 que possibilitam realizar três funções. Sua programação é feita através dos pontos com símbolo "x" que representam uma conexão de fusível (nível lógico '1'), e quando "queimados" sem conexão (nível lógico '0').

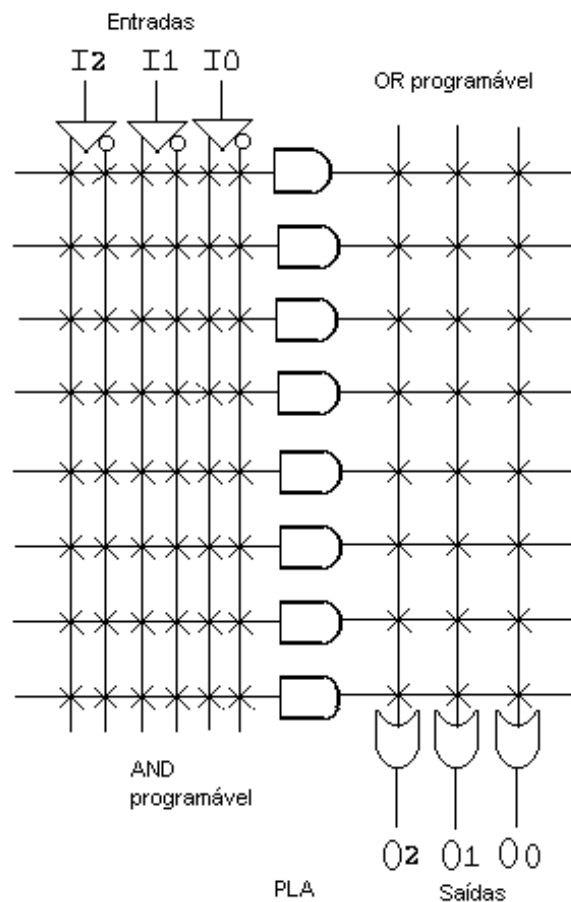


Figura 4.2 - PLA básico (AND-OR) programáveis nos dois planos.

Os módulos programáveis PLDs tornaram-se muito populares, mas apesar das vantagens decorrentes de sua flexibilidade, não substituíram totalmente os módulos padrão de função fixa em todas as aplicações, por serem caros e mais lentos.

Os PLDs em geral, apresentavam duas grandes limitações que eram, a impossibilidade de realizar funções multiníveis e de compartilhar produtos em diferentes funções. Em ambos os casos, isto era devido às características fixas das interconexões, onde é possível programar funções, mas não as interconexões entre funções. Tentando superar estas limitações foram projetados os dispositivos denominados FPGAs (*Field Programmable Gate Arrays*) e CPLDs (*Complex Programmable Logic Device*).

4.2.3-Dispositivos CPLDs

Com a densidade dos chips aumentando a cada dia os fabricantes passaram também a produzir componentes com características semelhantes às PLDs mas cada vez maiores em densidade. Surgiram os CPLD (*Complex Programmable Logic Devices*). Basicamente, um CPLD é um conjunto de múltiplos PLDs e um barramento configurável de interconexão em

um único chip. A densidade destes componentes permite o desenvolvimento de circuitos muito mais complexos, pois um único *chip* CPLD, substitui facilmente centenas de componentes da série 74xx (portas lógicas).

Nos CPLDS uma célula programável realiza funções lógicas universais de n variáveis, tais como multiplex, memória, conjunto AND-OR, etc. Uma rede de interconexões seleciona as entradas das células programáveis sob o controle de variáveis externas e as próprias saídas das células programáveis. Embora apresentem diferenças conforme o fabricante, as características básicas são:

- Função lógica no LAB (Logic Array Block) são programáveis;
- Um LAB é composto de 16 células e 32 expansores portas ANDs;
- Células constituídas de flip-flops tendo como entrada a soma de três produtos ;
- A saída de uma célula pode ser enviada para o pino de saída ou roteadas pela PIA (*Programmable Interconnect Array*) para outra célula.

Um exemplo de uma arquitetura CPLD é visto na figura 4.3. Nestes dispositivos a programabilidade das interconexões é limitada. Na verdade algumas interconexões são roteadas durante a fabricação. Contudo, esta limitação apresenta-se como uma vantagem, pois a temporização é completamente previsível. Por outro lado, a limitação da programabilidade é um dos aspectos de restrição no seu uso.

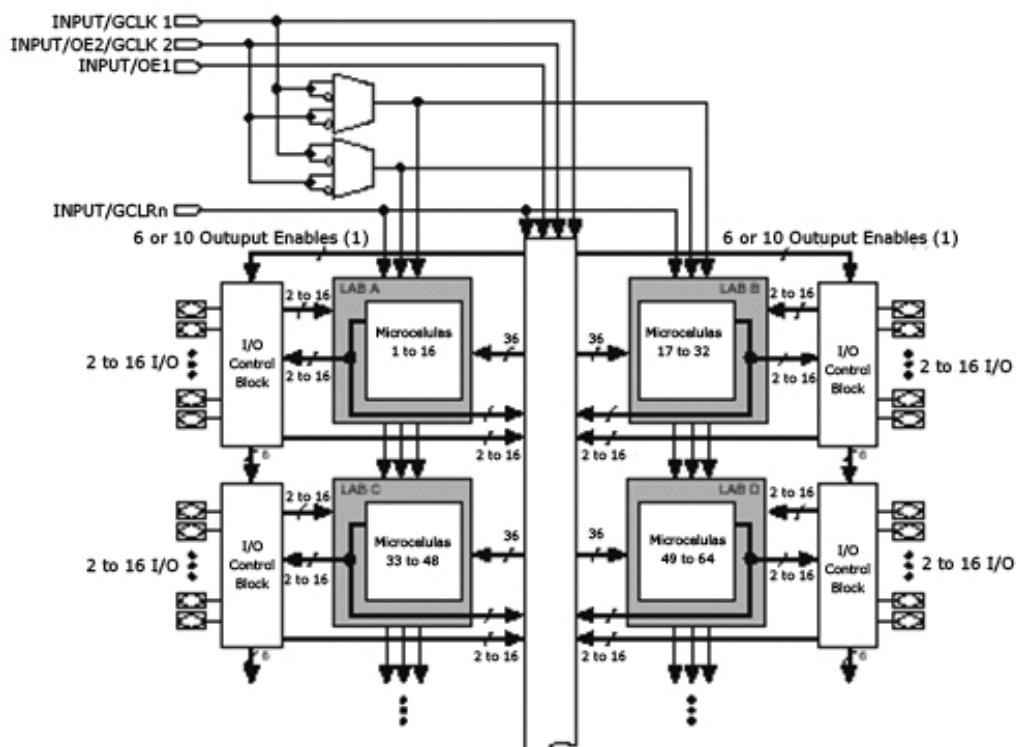


Figura 4.3 - Diagrama bloco do dispositivo da família MAX (Multiple Array Matrix) - MAX7000 – (ALTERA, 2006)

Devido ao fato dos CPLDs serem mais robustos que os PLDs simples, o seu potencial de uso é muito mais versátil. É possível utilizá-los tanto para simples aplicações como decodificação de endereços, como para lógicas de controle de alta performance ou complexas máquinas de estados finito. Nas aplicações que exigem lógica de alta performance, tradicionalmente os CPLDs são escolhidas ao invés dos FPGAs por serem menos flexíveis internamente, e possuírem um atraso (*delay*) interno usualmente menor (medido em nanosegundos). O atraso de propagação de um CPLD é mais previsível (não depende do roteamento escolhido), ao contrário do que acontece no FPGA.

4.2.4-Dispositivos FPGAs

Estes dispositivos são os mais importantes dispositivos lógicos programáveis, atualmente. *Field Programmable Gate Arrays* - FPGAs, ou matriz de portas programáveis em campo (ou pelo usuário) apresenta uma tecnologia introduzida pelo fabricante Xilinx (1984), que permite aos projetistas implementar circuitos e arquiteturas mais complexos.

Permitem a implementação de portas lógicas básicas e a interconexão entre essas portas para circuitos multiníveis. O aprimoramento desta tecnologia disponibilizou uma grande quantidade de dispositivos que são reprogramáveis pelo usuário e tornam possíveis teorias como a dos circuitos evolutivos.

Os FPGA's apresentam internamente circuitos multiníveis, possibilitando projetar circuitos complexos em um único dispositivo e ao contrário dos simples PLDs, dificultando a previsão de atrasos no processamento.

São circuitos programáveis compostos por um conjunto de células lógicas alocadas em forma de uma matriz. Algumas arquiteturas apresentam flip-flops e/ou registradores para gerar aplicações em lógica sequencial.

Existe um conjunto de blocos de entrada/saída que pode ser configurado como entrada, saída e bidirecional. As saídas (tri-state) e os registradores podem armazenar dados de entrada ou de saída e todos os blocos lógicos podem ser conectados para implementar qualquer função lógica desejada. Cada bloco lógico está conectado a um número determinado de matrizes de conexões programáveis, que, por sua vez, estão ligadas a um número de matrizes

de chaveamento programáveis. Em geral, a funcionalidade dos blocos assim como seu roteamento são configuráveis via software.

Programando as conexões apropriadas, cada bloco pode fornecer uma variedade de funções lógicas combinacionais e/ou sequenciais. As matrizes de conexões programáveis são usadas para estabelecer ligações entre entradas e saídas dos blocos lógicos, enquanto as matrizes de chaveamento programáveis são empregadas para rotear os sinais entre as várias matrizes de conexões.

Os FPGAs proporcionam um ambiente de trabalho simplificado e de baixo custo, tornando possíveis operar com um número ilimitado de circuitos através da configuração do próprio dispositivo. Como são reprogramáveis são aplicáveis em projeto de protótipos, economizando tempo e custo, isto devido à agilidade e fácil processo de desenvolvimento, simulação, teste, depuração e alteração do projeto.

Para ilustrar a arquitetura de um FPGA mostra-se na figura 4.4 seus blocos interconectores (blocos de comutação e blocos de conexão), trilhas e macrocélulas (blocos lógicos).

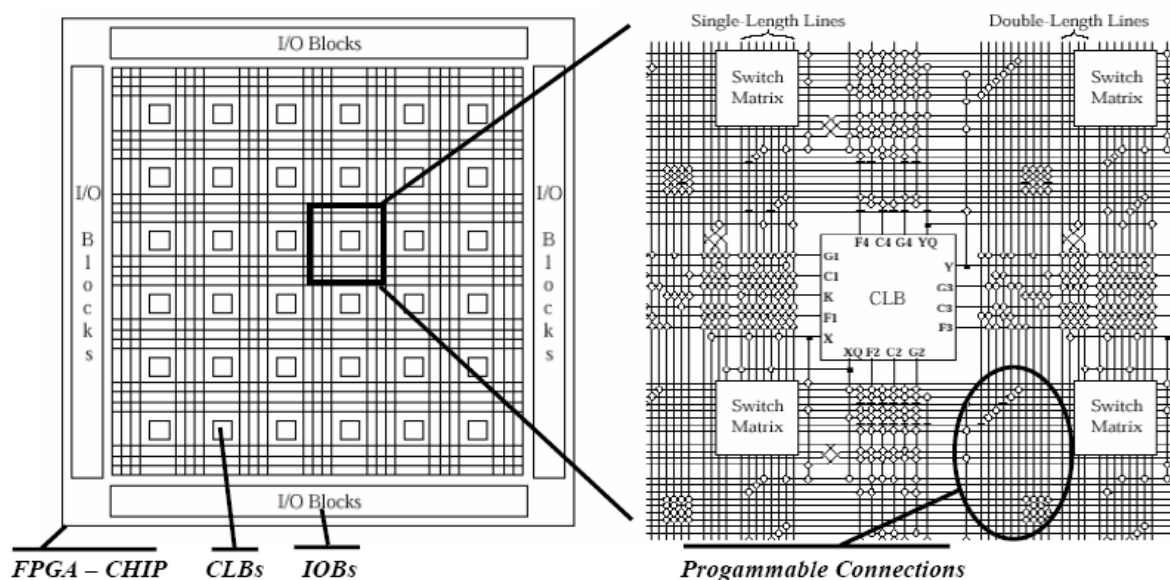


Figura 4.4 - Arquitetura de um FPGA (NASCIMENTO et al, 2008).

Na figura 4.4, na região periférica do chip, temos os blocos de entrada e saída (IOBs) responsáveis pela ligação dos pinos do chip com a matriz de CLBs. Os círculos pequenos sobre as linhas representam os pontos de conexão programáveis. Os blocos nas extremidades são chamados de *Switch Matrixs*. Estes blocos permitem mudar a direção de roteamento dos

sinais entre linhas horizontais e verticais. Na programação define-se a funcionalidade dos CLBs e IOBs e pontos de conexão das linhas de roteamento (NASCIMENTO et al, 2008).

4.3- Considerações no uso de PLDs

Anteriormente citadas, os PLDs fornecem inúmeras vantagens no seu uso. Podemos detalhar algumas delas que são importantes, pois proporcionam novos conceitos e habilidades mais robustas. Uma dessas vantagens é o alto grau de paralelismo, ou seja, permite lidar com projetos digitais em ambientes que trabalham com arquitetura em hierarquia. Cada projeto compilado e validado, pode ser interpretado como um símbolo ou como um componente padrão - *standard* que é incluído e reutilizado quando necessário, em outros projetos.

4.3.1-Arquitetura em Hierarquia de Projeto

Muitos projetos de circuitos digitais simples, podem ser desenvolvidos com técnicas de projeto usando componentes padrões. Entretanto, quando o projeto é complexo, pode haver uma enorme dificuldade pelo projetista, de interpretação, implementação, alocação, alterações e correção de alguns parâmetros do projeto.

O uso de ambientes de projeto que trabalhem com arquitetura em hierarquia, torna-se uma alternativa importante para implementar projetos digitais cada vez maiores e complexos . Basta partir dos sub-projetos e agregá-los no final para compor um projeto global .

4.3.2-Paralelismo

O alto grau de paralelismo marca a diferença entre as arquiteturas de circuitos baseadas em microprocessadores e os componentes lógicos programáveis. 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, podem ser inseridos várias etapas de um projeto, e possivelmente com frequências de trabalho diferentes (permitem clock elevados). Os circuitos de um projeto em lógica programável são facilmente configurados e o tamanho de barramentos podem ser adaptados para cada especificação de projeto. Fácil roteamento dos pinos de entrada e saída que são configurados pelo projetista na fase de compilação. Em único componente FPGA podem ser implementados, memórias, registradores, contadores, multiplexadores, demultiplexadores, máquinas de estados etc, formando um sistema completo em um único CI.

4.4 - Características comuns aos Dispositivos Lógicos Programáveis

Os dispositivos programáveis tem algumas características que agregam vários circuitos integrados e são detalhadas a seguir.

4.4.1- Reconfiguração

Os Dispositivos ditos Reconfiguráveis são circuitos integrados-CI, cujas as conexões internas podem ser programadas pelo usuário, isto é, a programação se dá a nível de hardware. FPGAs e FPAAs constituem o estado da arte da tecnologia dos dispositivos reconfiguráveis, referidos como dispositivos digitais e analógicos, respectivamente. Estes dispositivos trazem em seus blocos de construção duas importantes propriedades: auto-adaptação e características de reparação, através da reconfiguração automática, características exigidas na evolução de circuitos (MARTINS, 2008).

4.4.2 -Tecnologia de Programação

A tecnologia de programação caracteriza os PLDs em termos de **programabilidade** e **volatibilidade**. A programabilidade de um PLD se refere à capacidade de alteração do conteúdo funcional do dispositivo após sua primeira programação. É classificada de acordo com a tecnologia utilizada para implementar o conjunto de chaves usadas na configuração do PLD. Este conjunto de chaves, determina se os dispositivos podem ser reprogramáveis ou não. Adicionalmente, um circuito pode permanecer configurado ou não, conforme a propriedade de volatibilidade das chaves. Na tabela 4.1, apresenta-se as principais tecnologias utilizadas para implementar as chaves.

Tabela 4.1- Principais tipos de chaves e tecnologia de fabricação

Tipo de chave	Dispositivo	Tecnologia	Reprogramáveis	Voláteis
Fusível	PLAs	Bipolar	Não	Não
EPROM	FPGAs	UVC MOS	Sim	Não
EEPROM	FPGAs	EECMOS	Sim	Não
SRAM	FPGAs	CMOS	Sim	Sim
Anti-fusível	FPGAs (Actel)	CMOS+	Não	Não

4.4.3- Processo de realização de um projeto

O processo de projeto pode ser dividido em dois tipos de atividades em uma Síntese ou uma Análise. O processo de síntese agrega informações de uma descrição, que produzirá um projeto ou um circuito final. Na análise, parte-se do circuito para estudar o desempenho e as tomadas de decisão. A correção e a otimização do sistema a ser construído, são conceitos que devem ser considerados para a realização de projetos. Uma descrição final é correta se ela atende a todos os requisitos do problema inicial e pode ser fabricada. O projeto é considerado ótimo se 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 esbarra no compromisso entre custo e desempenho.

Os critérios de otimização para sistemas digitais podem ser resumidos em, espaço (o menor possível), tempo (o mais rápido possível), energia (consumir o mínimo de energia por unidade de tempo). Resumindo, baseado em uma descrição abstrata o projetista deve ser capaz de uma descrição detalhada de forma a atingir um projeto ótimo e mantenha a funcionalidade e desempenho a um custo mínimo.

4.5 - Linguagens de Descrição de Hardware

A linguagem de descrição de hardware ou *Hardware Description Language* (HDL) é um dos recursos disponíveis importantes, que tem facilitado o projeto e síntese de circuitos digitais, principalmente os que se utilizam de PLDs. Uma linguagem de descrição de hardware é própria para modelar a estrutura e/ou o comportamento de um circuito. É estruturada para facilitar a descrição abstrata do comportamento do circuito para propósitos de especificação (MANTOVANI; OLIVEIRA, 2003).

O comportamento pode ser modelado e representado em vários níveis de abstração durante o projeto, este fato, facilita e agiliza o trabalho e rapidez de desenvolvimento de muitos projetos. É possível modelar uma estrutura em qualquer linguagem de descrição de *hardware*, independente do comportamento do circuito. São exemplos de linguagem HDL: VHDL, VERILOG, AHDL (Altera), entre outras, desenvolvidas por inúmeros fabricantes.

A importância da utilização de linguagens de descrição de hardware manifesta-se principalmente na documentação do sistema, na simulação em diversos níveis, simplificando a migração tecnológica e reutilização dos recursos, com a construção de bibliotecas ou módulos.

O alto nível de abstração de uma linguagem de descrição de hardware, facilita ao projetista o desenvolvimento de projetos sem grandes conhecimentos em circuitos digitais .

Porém é necessário, um bom conhecimento de software e hardware de um sistema visando a criação de sistemas otimizados.

O projeto em linguagem de descrição de hardware facilita modificações, correções de erros e ampliações pelo projetista, com apenas algumas mudanças nas linhas de comando.

O uso de linguagens HDL, está ligado intrinsecamente ao uso de tecnologias dos dispositivos PLDs digitais, possibilitando a implementação de vários tipos de modelamento de circuitos. A descrição de um circuito não está relacionada a uma ou a outra tecnologia, descreve um projeto digital 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), indistintamente.

Deve conter o comportamento de sua estrutura física. Deve ser estruturada e descrita de forma tal que o interpretador (compilador) gere de forma única o circuito almejado. Independente do tipo de tecnologia empregada, a descrição de hardware permite lidar com circuitos complexos, e 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.

A escolha de uma linguagem de descrição de hardware entre as diversas que existem, 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 apresentar modelamento diferente para um mesmo circuito. Pode apresentar diferenças entre os resultados na compilação, tais como, tamanho do circuito sintetizado, alocação de memória na fase de compilação, tempo de compilação, etc.

Algumas linguagens de descrição de hardware apresentam a característica de **portabilidade**, ou seja, uma descrição comportamental ou estrutural pode, ser implementada em qualquer tecnologia. 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.

4.5.1-Linguagem VHDL (Very High Speed Integrated Circuit Hardware Description Language)

A Linguagem **VHDL** permite a transferência de projetos para qualquer tecnologia em construção de hardware existente, ou que ainda será desenvolvida, pois firmou-se como um padrão internacional para componentes digitais. Por isso é aceita por toda ferramenta comercial de síntese de circuitos. Esta linguagem fornece uma variedade de construções que permitem modelar o *hardware* em um nível elevado de abstração (PERRY, 2002).

Foi criada pelo Departamento de Defesa dos Estados Unidos (DoD), em 1980, para descrever a função e a estrutura de circuitos digitais aplicados ao projeto de circuitos integrados de aplicação específica ASICs. O manual de referência da linguagem, publicado em dezembro de 1997, incluía bibliotecas para FPGA em VHDL e depois desta data, foi amplamente utilizada na descrição e síntese de sistemas digitais, antes disso o projeto de circuitos era criado através de diagramas esquemáticos. Esta linguagem veio somar-se ao rápido avanço tecnológico alcançado pelas indústrias de circuitos integrados, tendo como ápice a tecnologia de alta velocidade VHSIC (*Very High Speed Integrated Circuit*), que permite uma maior integração e consequentemente uma maior complexidade de circuitos contidos em uma mesma pastilha.

Uma linguagem como essa, independente do formato original do circuito, pode servir como uma descrição e documentação eficientes do circuito, possibilitando os mais diferentes fornecedores e participantes a entender o funcionamento das outras partes, padronizando a comunicação. Facilita o projeto, pois a partir de uma descrição textual, um algoritmo, desenvolve o circuito, sem necessidade de especificar explicitamente as ligações entre componentes e ainda é utilizada para as tarefas de documentação, descrição, síntese, simulação, teste, verificação formal e ainda compilação de software, em alguns casos (PONTE ; MADEIRA, 2006, CASILLO, 2005).

A linguagem VHDL, bem como outras linguagens seguem um fluxo de projeto bem definido, composto de sete etapas, como apresenta a Figura 4.5:

Especificação de Requisitos, Modelamento, Síntese de Alto Nível, Mapeamento Tecnológico, Implementação e/ou Fabricação, Testes e Simulação.

O tempo e o custo de cada etapa dentro de um projeto é bastante variável, dependendo da tecnologia utilizada para implementar o sistema

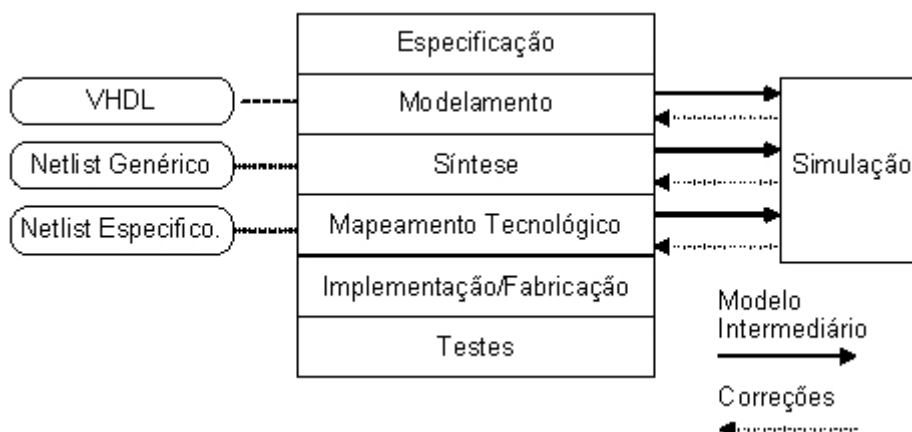


Figura 4.5 – Fluxo de um projeto (CARDOSO, 2003).

Durante a etapa de **Especificação de Requisitos**, o projetista faz um estudo e levantamento de todos os requisitos desejáveis e características do sistema que definem o seu funcionamento. Características tais como, atraso máximo permitido para as saídas, frequência máxima de operação, consumo de potência, custo, temperatura de operação, tensão de alimentação. Esta fase é de extrema importância porque evita-se a ocorrência de erros futuros e a perda de tempo na realização do projeto.

No **Modelamento** é iniciado o projeto, com base nas especificações da etapa anterior. Nesta fase o projetista descreve os modelos que representam o circuito. Os modelos sempre que possível, de maneira a não afetar o desempenho e a portabilidade, devem seguir o padrão estabelecido na linguagem, e não as extensões oferecidas pelos desenvolvedores das ferramentas de síntese. Finalmente, documenta-se adequadamente os modelos, incluindo nome do autor, datas de manutenção, comentários e explicações relevantes.

A **Síntese** de alto nível está para o hardware assim como a compilação está para o software. Na síntese, o modelo descrito será convertido para estruturas de dados representando as conexões, blocos, componentes e portas lógicas. Esta etapa é automática e dependente da ferramenta de software utilizada. Após a síntese ainda não está definido o circuito a ser implementado, a especificação intermediária que é resultante é ainda bastante genérica e pode ser direcionada para uma de muitas possibilidades de tecnologias de implementação.

No **Mapeamento Tecnológico** o circuito está definido dentro da tecnologia em que será implementado. Fazendo uma analogia com o software, essa etapa corresponderia à geração de código executável que ocorre ao final da compilação de um código fonte. Só é possível entender essa etapa adequadamente conhecendo-se as diferentes tecnologias disponíveis no mercado, como *full custom*, *gate array*, FPGAs, entre outros. O projetista pouco consegue influir no mapeamento, especificando apenas os parâmetros de otimização desejados.

Na etapa de **Implementação/Fabricação** são criados os primeiros protótipos, avaliadas as condições finais, detalhes de produção entre outros detalhes de implementação final. Em seguida à fabricação, os circuitos são testados para que possam ser entregues ao usuário com garantia de isenção de falhas.

A **Simulação** é uma etapa auxiliar, mas de grande relevância no ciclo de vida do projeto, pois avalia o comportamento do circuito e valida o modelo produzido até aquele momento. Durante a simulação, são apresentados amostras de entradas possíveis ao modelo do circuito, e os valores das saídas, memórias e nós internos do circuito são analisados a fim de comparar com o esperado na especificação. A Simulação gera uma realimentação para os processos de modelamento, síntese e mapeamento, evitando a propagação de erros para etapas posteriores. Muitos dos problemas encontrados na simulação não estão necessariamente ligados a erros no projeto, mas ao não preenchimento dos requisitos necessários, principalmente no que se refere aos tempos do circuito (atraso, setup/hold, frequência de operação).

4.6-Principais Fabricantes dos PLDs

Os principais fabricantes de dispositivos lógicos programáveis atualmente são: Xilinx, Altera, Atmel e Actel. Os fabricantes de Dispositivos Lógicos Programáveis procuram desenvolver a cada instante outros dispositivos, melhorando o desempenho e aumentando as pinagens, visando a criação de dispositivos que sejam capazes de configurar e reconfigurar sistemas digitais cada vez mais complexos e maiores, aumentando suas participações e lucros no mercado de dispositivos altamente rentável.

Dentre esses fabricantes encontram-se a Altera, que desenvolveu várias famílias de dispositivos como a MAX, FLEX, APEX, STRATIX e etc. Junto com estes dispositivos a Altera desenvolve editores para a sintetização e programação de um sistema digital para uso dos seus componentes, como os ambiente de projeto MaxPlus e Quartus II. O ambiente de projeto fornece muitas ferramentas de síntese, otimização, compilação, simulação e

programação para que o projetista possa criar um projeto otimizado e eficiente e editores diferenciados em gráficos, texto e formas de onda.

Na figura 4.6 é ilustrado a evolução dos componentes PLDs Altera.

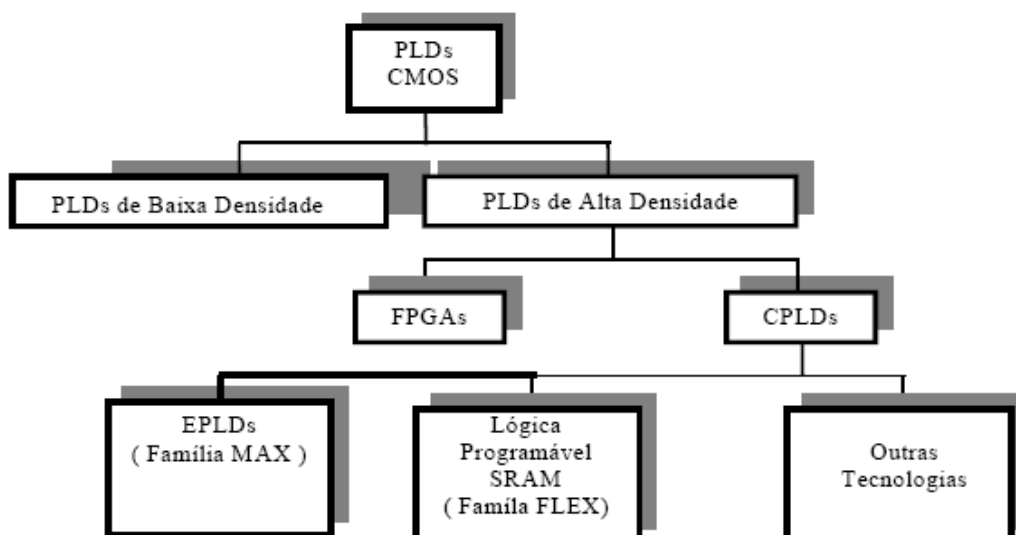


Figura 4.6- Evolução das famílias de PLDS da ALTERA.

O fabricante Xilinx apresenta uma boa aceitação de mercado de seus componentes com suas famílias Virtex, Spartan, CoolRunner e um ambiente de desenvolvimento de projeto bastante amigável, próprio.

4.7- Dispositivo Stratix III

O dispositivo FPGAs Stratix III foi utilizado neste projeto por ser um dispositivo que possibilitou a realização da arquitetura que está sendo proposta, em termos de quantidades de I/O (input/output) 488 pinos no total, portas lógicas e 5.455,872 blocos de memórias em bits. Este dispositivo pertence a família de componentes do fabricante Altera e é baseado em elementos reconfiguráveis SRAM CMOS. A escolha pelo componente da Altera deve-se ao fato da disponibilidade do software.

Este dispositivo oferece núcleo de tensão selecionáveis, na faixa de 0,9 a 1.1V necessários para o tipo de sistema que irá programar, apresentando um alto desempenho e exigindo um mínimo de consumo.

Na figura 4.7 faz-se a comparação entre os blocos de programação de um FPGA convencional e o Stratix III.

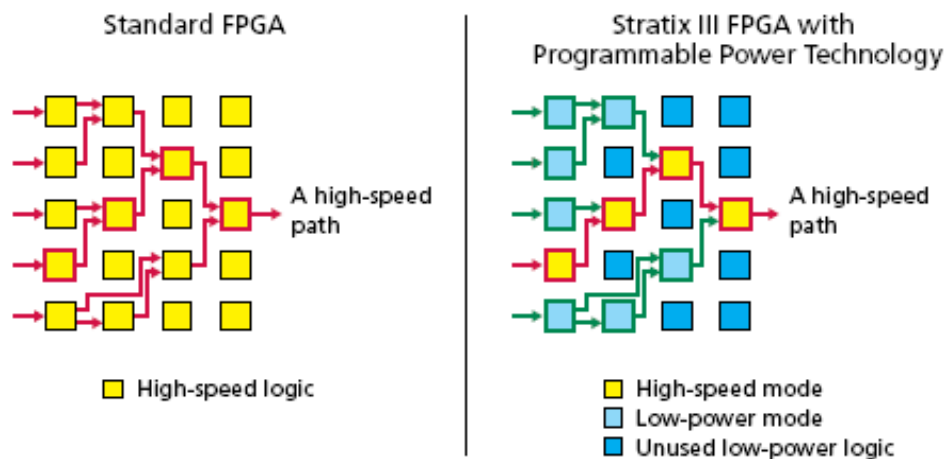


Figura 4.7 – Comparação FPGA convencional x FPGA Stratix III (ALTERA, 2008).

O FPGA convencional possui modo de alta velocidade na parte de comunicação entre os blocos de programação e velocidade rápida entre os caminhos de cada bloco. Para o Stratix III soma-se a estas características, a baixa potência.

Mostra-se na figura 4.8 a arquitetura do bloco do dispositivo Stratix III utilizado neste projeto (*Stratix III E EP3SE50 floorplan*) e logo abaixo as características deste dispositivo.

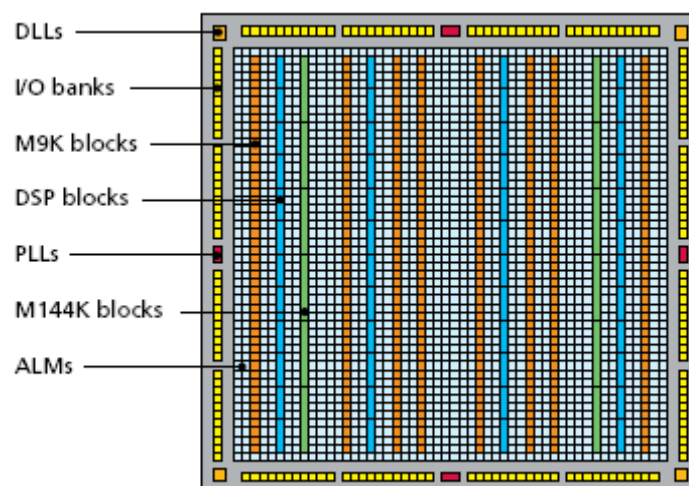


Figura 4.8 – Arquitetura do Stratix III E EP3SE50 (ALTERA, 2008).

Esta arquitetura suporta um bom desempenho no processamento de sinal digital pois, possui o dobro de blocos DSP comparado a outros tipos de dispositivos convencionais e/ou antecessores. Apresenta mais blocos multiplexadores/acumuladores além de ter o mais alto desempenho e baixo custo.

4.8 - Conclusão

Neste capítulo foram apresentados teorias sobre a tecnologia de dispositivos programáveis, visando fornecer subsídios para o entendimento destes dispositivos e para a escolha de um determinado componente que satisfaça as especificações de projeto. Descreve-se as principais características, tipos, tecnologia de programação, fabricantes, ferramentas de trabalho que mudaram a concepção de projeto, permitindo ao projetista desenvolver sistemas digitais, em forma gráfica ou em linguagem texto de um modo mais fácil e rápido, menor custo e diminuído a probabilidade de erros. Finalmente, faz-se uma breve descrição do componente usado na simulação do projeto proposto.

Capítulo 5

Metodologia Proposta para Detecção de Faltas

Neste capítulo apresenta-se a metodologia proposta neste trabalho para detecção de faltas em alimentadores de distribuição. Compõem também este capítulo a descrição dos parâmetros e a técnica usada no projeto para realização de uma arquitetura.

5.1-Faltas em Sistemas Elétricos

Os sistemas de distribuição de energia elétrica estão susceptíveis às faltas que podem ser temporárias ou permanentes. As faltas temporárias são aquelas cuja duração é limitada ao período necessário para restabelecer o serviço através de operação automática do equipamento de proteção que desligou o circuito ou parte dele. As faltas permanentes são todas as interrupções não classificadas como temporárias ou programadas. Um dos tipos de faltas mais comuns em um sistema de energia elétricas é o curto-circuito. Este tipo de falta causa

sobrecorrente e desbalanço na tensão. Na Tabela-5.1 (GIGUER, 1988) apresentam-se dados estatísticos entre os tipos de faltas e a composição das mesmas.

Tabela–5.1: Probabilidade entre os Tipos de Faltas e Composição das mesmas

Faltas	%	Permanentes(%)	Temporárias(%)
Trifásicas	2	95	5
Bifásicas	11	70	30
Fase-Terra	79	20	80
Outros	8	-	-

Essas faltas no sistema são normalmente provocadas pela ação de descargas atmosféricas, contatos de árvores e animais às partes vivas do sistema, falhas de equipamento e erro humano. Classificam-se como faltas simultâneas quando ocorrem duas ou mais faltas no mesmo instante de tempo. Estas ocorrências são resultados de eventos como descarga atmosférica e manipulações erradas de equipamentos pelos operários. Este tipo de falta é um evento que tem baixa probabilidade que ocorra, pois é muito difícil que ocorram duas faltas no mesmo instante de tempo.

O diagnóstico de alarmes ou diagnóstico de faltas consiste na análise e interpretação de alarmes a partir dos relatórios de alarmes fornecidos pelo sistema SCADA de modo a identificar (qual componente apresenta defeito), localizar (qual a localização no sistema) e diagnosticar (saber qual tipo de defeito) componentes no sistema de distribuição que apresentem qualquer defeito. Esse diagnóstico pode ser efetuado de forma local ou centralizado. Neste trabalho considera-se o diagnóstico efetuado de forma local e limitado ao ambiente das subestações de distribuição de energia elétrica e tem como objetivo fornecer diagnóstico de faltas aos operadores de centros de controle destas unidades específicas (CODs).

Através de dispositivos de proteção e controle que registram todos os eventos no sistema SCADA, pode-se monitorar faltas e registros de eventos em uma subestação, que permite identificar sobre o tipo de faltas no sistema ou falhas ocorridas em algum componente do sistema. Para os registros desses eventos, normalmente utilizam-se registradores digitais de faltas por estes apresentarem multicanais. Um microprocessador pode gerenciar os sinais do

registrador e as unidades de aquisição de dados, análises ou unidades mestres em um centro de controle remoto e armazená-los em uma memória de pré-faltas.

Visando a realização de uma arquitetura para o tratamento dos dados provenientes do sistema de aquisição e o desenvolvimento do sistema de diagnóstico de faltas, consideram-se que os eventos estão registrados num vetor composto de 24 posições. As informações com os sinais analógicos e digitais usadas para efetuar o diagnóstico encontram-se nas Tabelas 5.2 e 5.3, (PERLAZA; DELGADO, 2005).

Na Tabela 5.2, os sinais de corrente e de tensão nas fases (R, S, T) são obtidos dos transformadores de correntes e de potencial. Os valores de corrente e tensão de linha quando necessários são obtidos usando-se as relações de linha e de fase. O sinal residual é obtido do enrolamento conectado em delta aberto do transformador e é utilizado para alimentar os equipamentos de proteção de medida (GIGUER, 1988). A tabela 5.2 mostra os dispositivos que estão sendo utilizados na medição das grandezas com seus respectivos códigos. Estes sinais analógicos são apenas para as fases do Circuito 1 (Fig. 5.1). Cada sinal analógico passa por um conversor A/D, com amostragem a cada 30 segundos. Para o exemplo de uma subestação real como é caso da figura 5.1, observa-se 12 circuitos onde cada circuito possui as grandezas físicas mostrada na tabela 5.2.

Tabela 5.2 – Sinais Analógicos obtidos no alimentador C1 de uma SE real (fig.5.1)

Sinal	Descrição	Código
1	Corrente da fase R	ElmVAN00161
2	Corrente da fase S	ElmVAN00162
3	Corrente da fase T	ElmVAN00163
4	Corrente trifásica	ElmVAN00164
5	Tensão da fase RN	ElmVAN00165
6	Tensão da fase SN	ElmVAN00166
7	Tensão da fase TN	ElmVAN00167
8	Tensão de linha RS	ElmVAN00168
9	Tensão de linha ST	ElmVAN00169
11	Tensão de linha Trifásica	ElmVAN00171
10	Tensão de linha TR	ElmVAN00170
12	Potência Ativa	ElmVAN00172
13	Potência Reativa	ElmVAN00173

Na tabela 5.3 apresentam-se os sinais digitais que registram os estados de proteção dos disjuntores e sinais de tele proteção dos sistemas de potência que podem ser monitorados a partir da subestação de distribuição.

Tabela 5.3. Sinais Digitais (PERLAZA e DELGADO, 2005)

Sinal	Descrição
9	Posição do polo A do disjuntor.
10	Posição do polo B do disjuntor.
11	Posição do polo C do disjuntor
12	Partida de proteção principal 1
13	Partida de proteção secundária ou de retaguarda 2
14	Disparo proteção principal 1
15	Disparo proteção secundária ou de retaguarda 2
16	Envia sinal de onda portadora - de proteção com canal piloto.
17	Recebe sinal de onda portadora - de proteção com canal piloto.
18	Ordem de religamento monofásico
19	Ordem de religamento trifásico
20	Partida do relé direcional de sobrecorrente 67NB
21	Disparo do relé direcional de sobrecorrente 67NB
22	Disparo do relé de sobrecorrente instantâneo 50BF
23	Disparo de relé de sobretensão (59).
24	Oscilação de potência

O estado operacional do disjuntor (9-11, aberto-A, ou fechado-F) é o elemento mais importante a ser monitorado em uma subestação, uma vez que é o disjuntor que recebe as ordens enviadas pelos relés de proteção ou do operador para isolação de linhas, barramentos e transformadores. O disjuntor possui três pólos conectados a cada uma das fases do sistema (R, S, T), sendo necessário monitorar a operação de cada um destes pólos, visto que o disjuntor pode apresentar abertura monopolar devido às faltas monofásicas que são as mais frequentes. Nas partidas das proteções (12 e 13), deve-se contar com dois tipos de proteção: a proteção principal e a proteção de retaguarda ou secundária que podem ou não ter o mesmo

princípio de funcionamento. Nos disparos das proteções (14, 15), ao registrar estes sinais, obtêm-se os tempos de operação das proteções, e com isto é verificado se está devidamente correta à operação de detecção de ocorrência de faltas.

Quando há imprevistos como uma operação incorreta, deve haver registros das ordens do religamento do disjuntor e assim, determinar a causa do mau funcionamento ou a não funcionalidade do equipamento de proteção. Esta função é a operação de religamento (18, 19) que a executa. Os relés de proteção 67NB têm como finalidade comparar a direção do fluxo de potência em caso de faltas, em ambos os extremos de uma linha de transmissão. Quando há uma medida, e esta indica que as correntes circulam vazias em uma área protegida é porque houve uma falha interna em algum equipamento do sistema, operando um relé instantaneamente. E quando há um caso de que as mesmas correntes circulam fora da área protegida é porque houve uma falha externa e com isso não haverá disparos dos relés. Os relés 50BF (22) possuem uma unidade de sobrecorrente e uma de temporização.

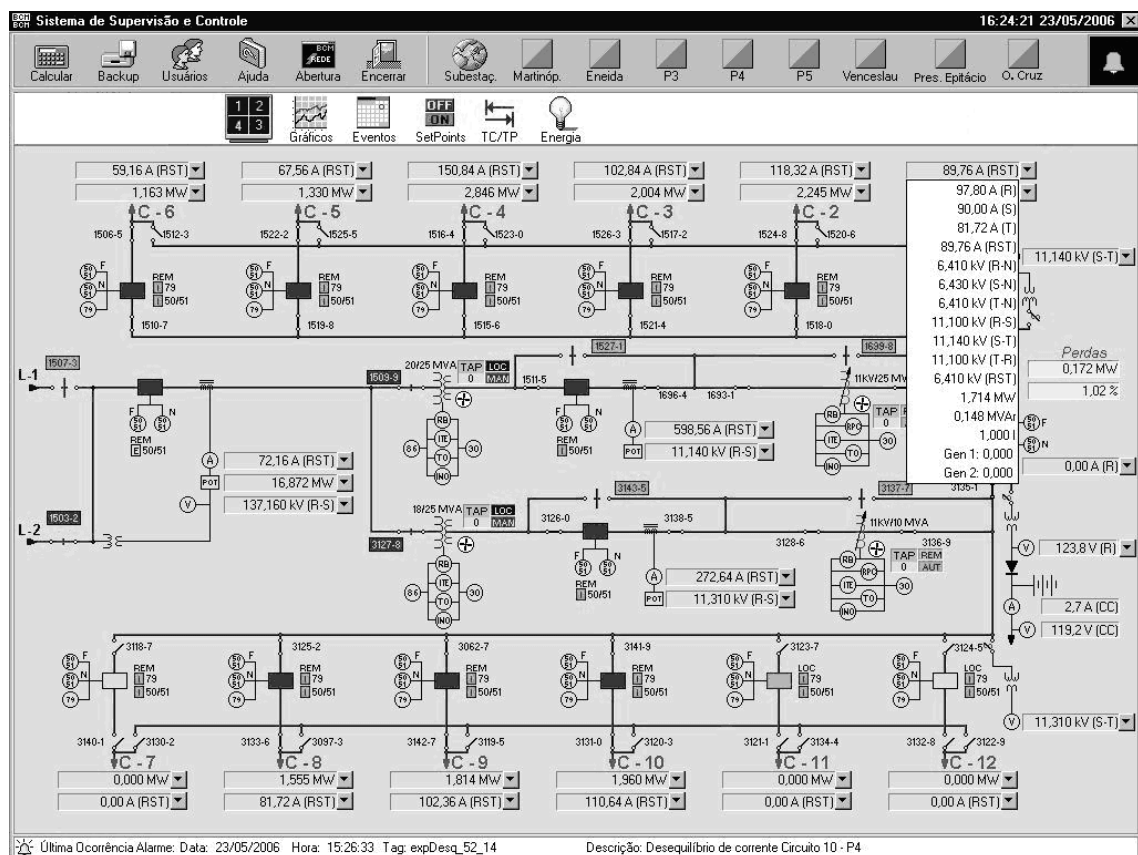


Figura 5.1 – Tela de supervisão de uma Subestação real

Com base nos sinais das tabelas 5.2 e 5.3 foi possível desenvolver um sistema capaz de detectar faltas usando uma arquitetura para ser implementada em um dispositivo lógico programável - FPGA.

Mostra-se na tabela 5.4 o tratamento dado as informações provenientes de um sistema de aquisição de uma subestação real e o registro de seu banco de dados, armazenados a intervalos de 30 segundos. Os dados coletados na base decimal são convertidos para a base binária e depois normalizados para serem expressos com uma resolução de 8 bits (resolução do conversor A/D).

Tabela 5.4 – Sinais analógicos em C1 e seus valores correspondentes

Descrição	Valores Analógicos	Valores normalizados (100/10000)	Valores Binários
Corrente da fase R	97,8A	0,978	00000001
Corrente da fase S	90A	0,90	00000001
Corrente da fase T	81,72A	0,8172	00000001
Corrente trifásica	89,76A	0,8976	00000001
Tensão da fase R	6,41KV	64,1	01000000
Tensão da fase S	6,43 KV	64,3	01000000
Tensão da fase T	6,41 KV	64,1	01000000
Tensão de linha R	11,1 KV	111	01101111
Tensão de linha S	11,14 KV	111,4	01101111
Tensão de linha T	11,1 KV	111	01101111
Tensão de linha Trifásico	6,41 KV	64,1	01000000
Potência Ativa	1,71MV	171	10101011
Potência Reativa	0,15MVA _r	15	00001111

5.2- Tipos de Arquiteturas

Para o desenvolvimento de uma arquitetura em sistemas digitais, existem vários conceitos que podem ser aplicados de forma a melhorar o processamento dos dados em termos de velocidade e otimização do projeto.

Com esse objetivo foram pesquisados alguns tipos dos quais podemos citar as arquiteturas avançadas paralelas *pipeline*, arquitetura superescalar, dentre outras. Descreve-se a seguir algumas arquiteturas e vantagens e desvantagens no seu uso.

5.2.1 -Arquitetura Pipeline

A arquitetura *pipeline* é um tipo de arquitetura em que dois ou mais programas podem ser executados de forma coordenada de modo que a saída de cada um é redirecionada como entrada do próximo. Assim, o conjunto dos programas que são executados desta forma passa a se comportar como um novo programa, com a entrada direcionada ao primeiro programa e a saída vindo do último. Desta forma combina-se a ação de vários executáveis binários de forma fácil e flexível, ao invés de modificar e recompilar os respectivos programas, ou mesmo escrever novos programas para as tarefas que se queira realizar. Esta arquitetura aumenta a velocidade de processamento (ANDRÉ, 2008).

5.2.2 Arquitetura Superescalar

Uma arquitetura superescalar é aquela que permite ter várias instruções nas quais cada instrução pode ser iniciada simultaneamente e executadas independentemente (ANDRÉ, 2008).

Esta arquitetura inclui todas as características do pipeline, com a diferença que possui instruções que podem estar executando no mesmo estágio do pipeline, ou seja, em linhas pipeline diferentes. Uma característica necessária da arquitetura superescalar é de iniciar múltiplas instruções no mesmo ciclo de relógio.

No projeto foi utilizada a arquitetura do tipo superescalar com o tratamento de dados na forma paralela, visando aumentar a velocidade de processamento dos dados.

5.3. Conclusão

Baseados em dados disponíveis de uma SE real, mostra-se uma metodologia para desenvolver o projeto do sistema para detecção de faltas na forma de hardware, usando informações provenientes da saída de alimentadores em subestações de distribuição. Na

realização de uma arquitetura em FPGA, procurou-se a forma mais adequada e uma arquitetura de circuito que otimizasse as respostas dos sinais e a área de utilização no dispositivo.

Capítulo 6

Arquitetura do Sistema de Detecção de Faltas

Depois de realizado o estudo sobre os principais parâmetros necessários para atuar de maneira eficiente, via hardware, na detecção de faltas no sistema de uma subestação, implementa-se uma proposta de circuito ou arquitetura usando software de desenvolvimento da ALTERA, Quartus versão 7.2 sp2 Web Edition e a linguagem de descrição de hardware VHDL, visando a aplicação em um dispositivo lógico programável como os CPLDs StratixIII (EP3SE50F780C2) do fabricante ALTERA. Neste capítulo, apresenta-se uma arquitetura e o resultado das simulações para mostrar o desempenho do circuito projetado.

6.1 Arquitetura

A arquitetura proposta foi originalmente projetada a partir da idéia do sistema de detecção de faltas desenvolvido em (PERLAZA; DELGADO, 2003). Em Delgado, o projeto consiste de um sistema que usa o conceito de DNA na comparação de dados, por isso, foi denominado de **chip de DNA**. Neste projeto optou-se por usar portas XNOR para fazer a comparação, observando as igualdades bit a bit entre os bancos de padrões de pré-faltas e as informações coletadas em tempo real.

Usa-se a linguagem VHDL para implementar registradores para armazenamento temporário dos dados e memórias do tipo ROM para guardar o banco de dados de faltas. A configuração usada permite a otimização de pinos e o aumento de registradores, caso seja necessário um aumento na capacidade de tratamento de dados.

Com base nos parâmetros anteriormente mencionados e nas especificações do projeto, implementa-se na forma de diagrama de fluxo mostrado na figura 6.1, o sistema de detecção e diagnóstico de faltas.

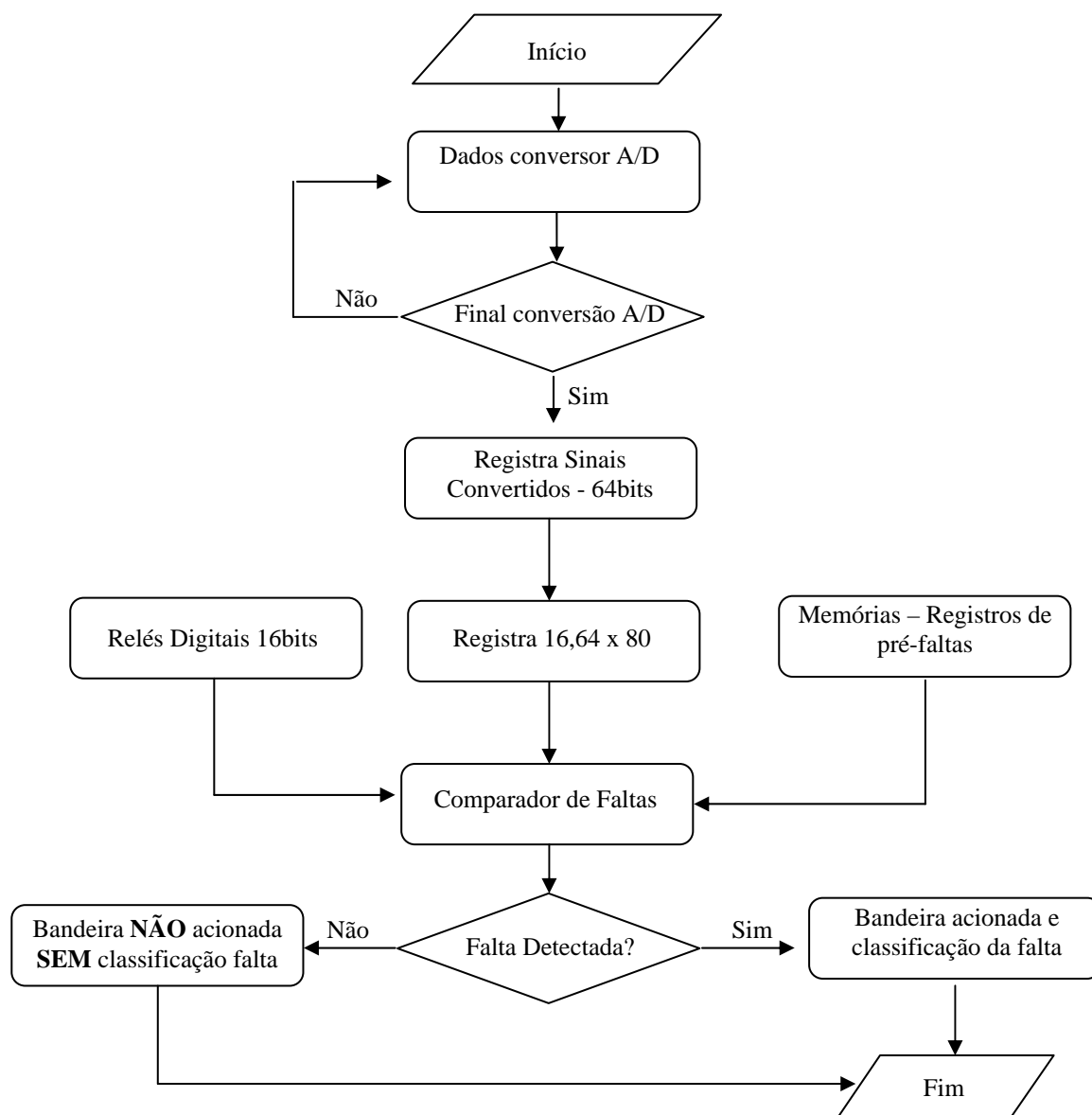


Figura 6. 1- Diagrama de Fluxo do Sistema

Com base no diagrama de fluxo foi desenvolvida a arquitetura da figura 6.2 de forma hierárquica, onde cada bloco foi realizado em linguagem de descrição de hardware VHDL e agrupados no final de forma gráfica para compor o projeto global.

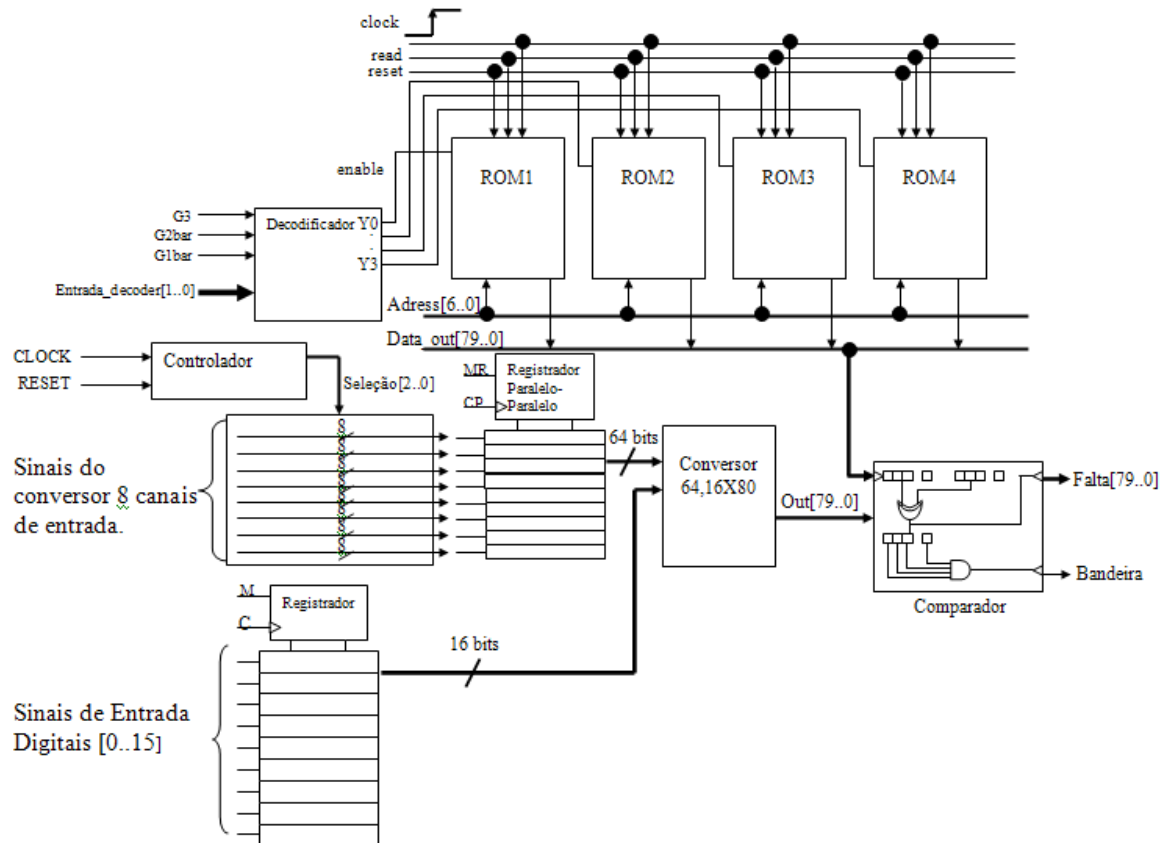


Figura 6.2- Arquitetura do Sistema de Detecção de Falta

Neste circuito faz-se o controle da aquisição de dados, através do conversor, depois armazena-se em um só vetor, um registrador de 80 bits, de forma a proceder na seqüência a comparação através de portas XNOR e AND com um banco de informações de padrões de pré-faltas armazenados em memórias ROMs. Se for verificada alguma igualdade é acionado um alarme luminoso (acendimento de LEDs), alertando ao operador a existência de falta.

Baseado em um conversor A/D de 8 canais, com uma resolução de oito bits, amostragem de 1ms, o sistema deve controlar o início e fim de conversão. A medida que faz a varredura dos canais e a conversão de cada canal, o resultado vai sendo registrado em um registrador de 8x64 bits. Esta informação tratada na forma paralela, é anexada aos 16 sinais digitais, resultando um string de 80 bits que representa um registro de eventos de pré-faltas.

O registro de pré-faltas é comparado em forma paralela, através de uma função lógica XNOR e uma porta AND (figura 6.3), com um string de dados contidos nas memórias

ROMs, que representam as informações das faltas individuais e de todas as possíveis combinações de duas faltas do sistema de potência. A função lógica AND detecta a sequência de '1s' que pode acusar uma falta.

Baseado no artigo de Delgado e Sepúlveda (PERLAZA; DELGADO, 2003, SEPÚLVEDA *et al.*, 2004) foram considerados 300 diferentes tipos de faltas, isto é, 24 faltas individuais e 276 possíveis combinações de duas faltas. Terminada a comparação transfere-se a informação a unidade mestra do sistema de potência, via sistema de comunicações para realizar a análise de faltas correspondentes.

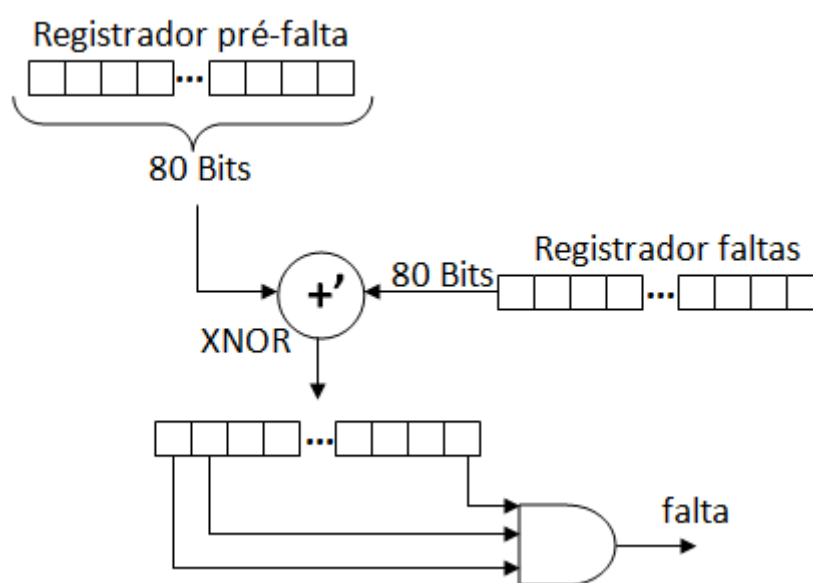


Figura 6.3 - Circuito que compara os registradores de faltas e pré-faltas

Para a realização das simulações e a validação do projeto, foram usados alguns dados reais de uma SE (P4 de Presidente Prudente do estado de São Paulo). Portanto, apresenta-se a seguir os detalhes de cada bloco da arquitetura proposta (figura 6.2) e as suas simulações.

6.2-Controlador

O controlador foi implementado para gerenciar as várias funções do sistema. Para implementar este controlador baseou-se nas características de funcionamento de um conversor (A/D0808) comercial, já descritas suas características anteriormente.

O controlador foi desenvolvido em linguagem VHDL que libera a cada sinal de *clock* (transição positiva) um canal do conversor. Este processo é repetido até que todos os 8 canais sejam habilitados para um registrador paralelo/paralelo. Para validar este funcionamento, mostra-se uma simulação, figura 6.4, com controle de seleção dos canais para o conversor A/D. Considera-se nesta simulação o sinal de reset em nível alto (reset em nível baixo, o controlador permanece no estado inicial).

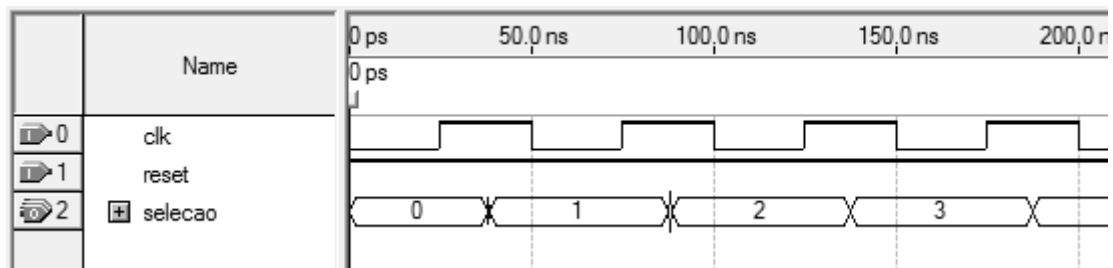


Figura 6.4- Simulação do controlador

6.3- Registrador Paralelo/Paralelo

Este registrador registra os 8 sinais analógicos provenientes do conversor A/D de modo paralelo/paralelo. Após a leitura dos 8 canais (8 pulsos de *clock*), este registrador terá uma informação de 64 bits. Estes 64 bits irão compor no próximo bloco de circuito, junto aos sinais dos relés, os 80 bits que serão processados pelo comparador.

A figura 6.5a, ilustra o armazenamento de cada célula (8 bits). Na figura 6.5b mostra-se o símbolo do bloco que representa uma entrada de programa realizado na linguagem de descrição de hardware VHDL, gerado pelo software de desenvolvimento Quartus II da Altera. Na figura 6.6 apresenta-se uma simulação deste bloco.

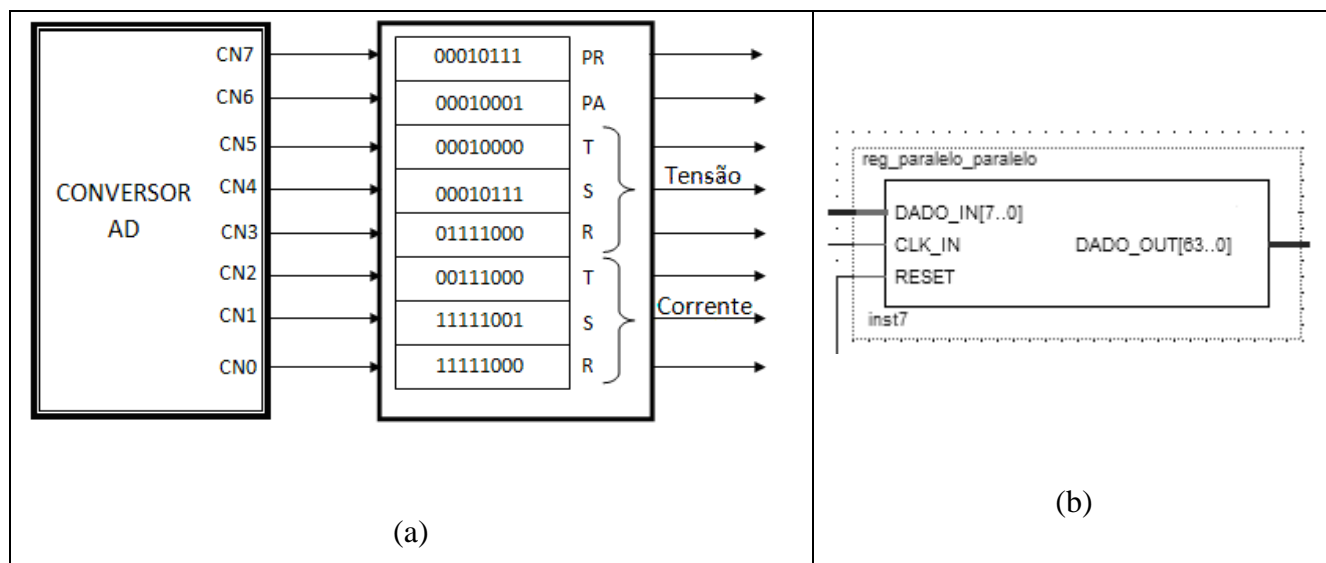


Figura 6.5- (a) Esquemático ilustrando a função do registrador paralelo/paralelo. (b) Símbolo para o Registrador– Quartus II Altera.

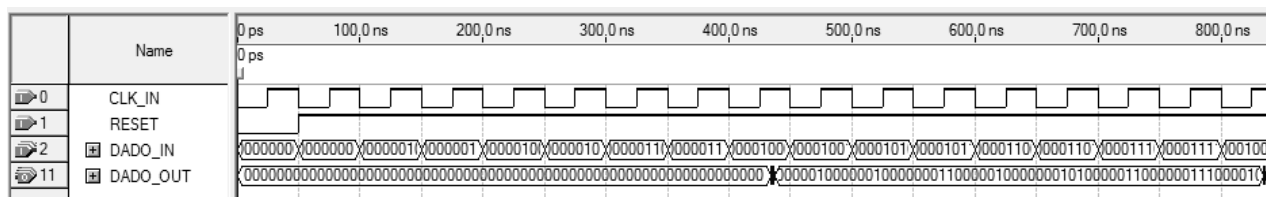
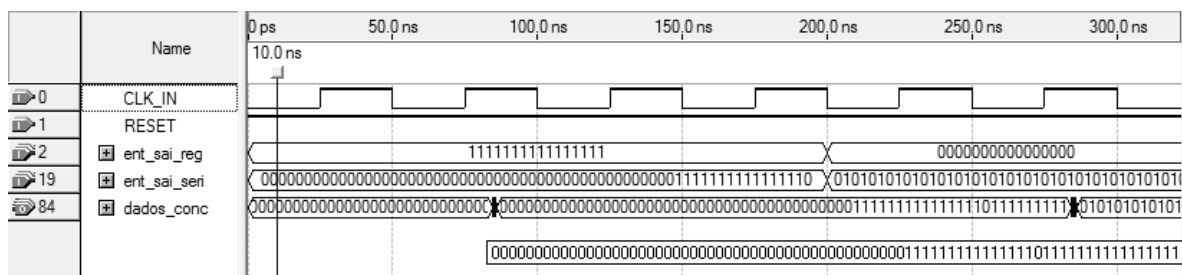


Figura 6.6- Simulação do Registrador Paralelo/Paralelo

Na simulação do registrador (figura 6.6) os dados **DADO_IN** provenientes do conversor são armazenados a cada subida do ciclo de *clock*. Após a varredura dos 8 canais, um string de 64 bits é mostrado na saída **DADO_OUT**, conforme se observa na simulação. Nesta simulação foi usado um ciclo mínimo de $50ns$ para que a simulação não apresente sinais espúrios.

6.4-Convertor 64,16 x 80

Neste bloco são processados os dados do registrador paralelo/paralelo, 64 bits e um string de 16 bits conforme arquitetura na figura 6.2, que são as informações dos relés digitais formando um string único de 80 bits, que serão enviados para a entrada do comparador. É mostrada na figura 6.7, uma simulação deste conversor.



A **ent_sai_reg** representa os sinais do **registrador digital**, 16 bits, a **ent_sai_seri** são os sinais do **registrador paralelo/paralelo**. Na saída, **dados_conc**, observa-se os dados formando um único string (80bits). Nesta simulação o sinal de *clock* é de 50ns.

6.5-Memórias

Para compor o banco de 300 faltas foram usadas 4 memórias de 128 posições cada, projetas na linguagem de descrição de hardware. Junto com este bloco de memórias implementa-se um decodificador (em nível alto), cada memória existente no circuito. Conforme a memória habilitada e o endereço solicitado, os dados vão sendo liberados para o comparador. Mostra-se na figura 6.8 o esquemático das memórias e na figura 6.9 uma simulação onde a entrada **a[2..0]** é a entrada de seleção.

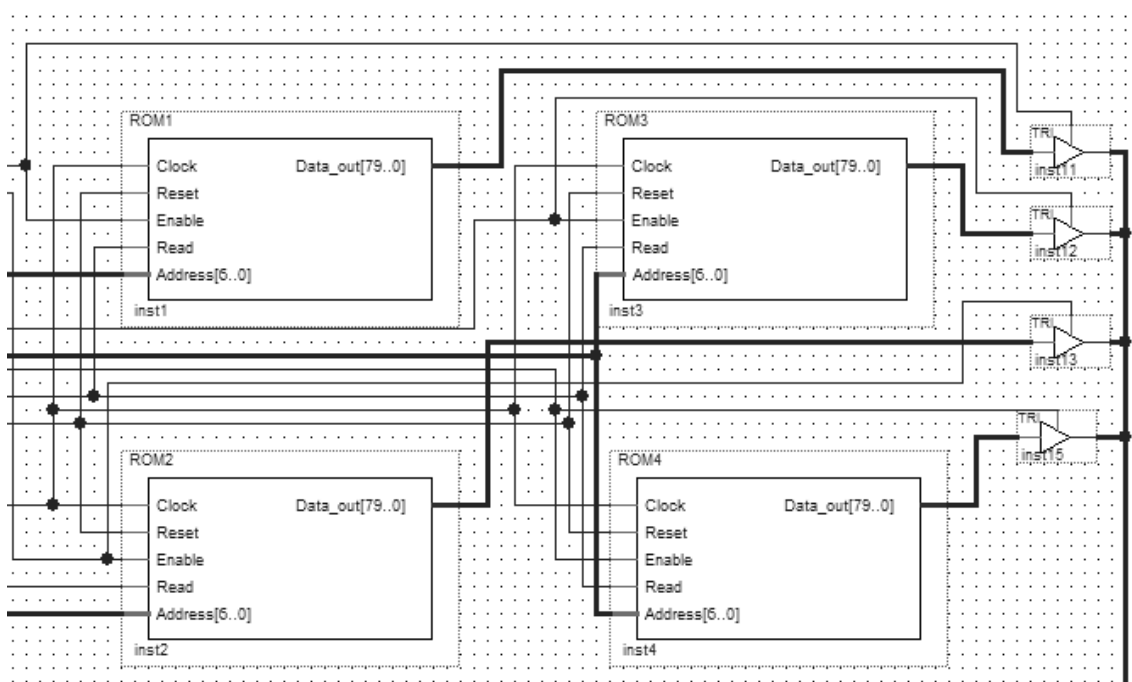


Figura 6.8-Esquemático das Memórias ROMs.

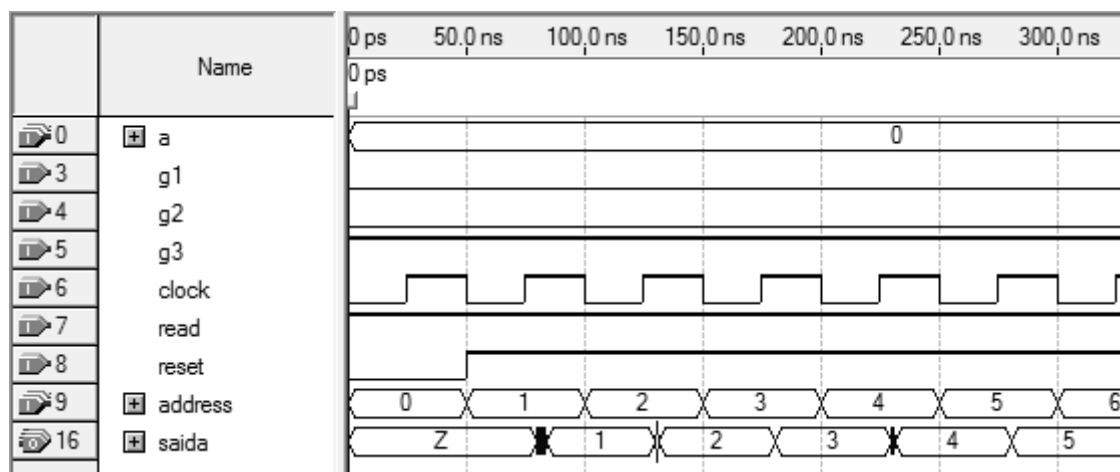


Figura 6.9- Simulação da Memória Rom.

Nesta simulação para $a=0$, é selecionada a ROM1, sendo que as entradas de controle devem ser colocadas $g1=g2=0$ e $g3=1$. Para **read** em nível alto, observa-se a **saída** de dados, para alguns endereços no barramento de endereços, onde em cada endereço têm-se armazenado um evento de falta. Estas memórias respondem a frequência de operação de no máximo 25MHz.

6.6-Comparador

O comparador realiza as comparações de 80bits entre os dados das memórias e os dados vindos do conversor 64,16x80, usando portas XNOR e AND, que detectam as igualdades binárias.

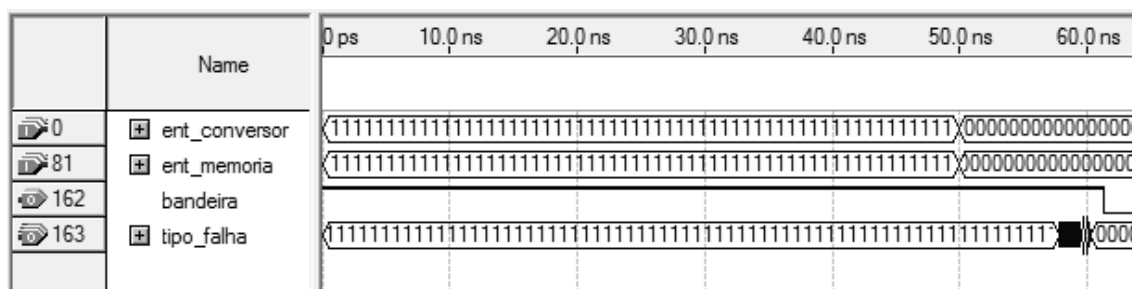


Fig. 6.10- Simulação para o comparador mostrando uma falta ocorrida.

Observa-se nesta simulação (figura 6.10) duas entradas iguais onde a primeira consiste em uma entrada de dados vindo do conversor e a outra das memórias, que irão ser comparadas, acionando o alarme indicando uma falta.

6.7 Arquitetura Completa

Neste projeto está sendo usado na compilação e simulação o *Device* Stratix III do ambiente Quartus II –Altera – EP3SE50F780C2 que disponibiliza 488 pinos de entradas e saídas (I/O). Para o projeto em questão, estão sendo utilizados apenas 69% (339 pinos de I/O), por enquanto. Na figura 6.11 apresenta-se o bloco nível de topo do projeto hierárquico com as entradas e saídas globais, simulado no *software* QuartusII. Na figura 6.12 é mostrado o diagrama de bloco que compõe o projeto hierarquico. Para todas as simulações a seguir, o clock utilizado foi de 20MHz, para que o circuito apresente um bom desempenho (figuras 6.13 e 6.14).

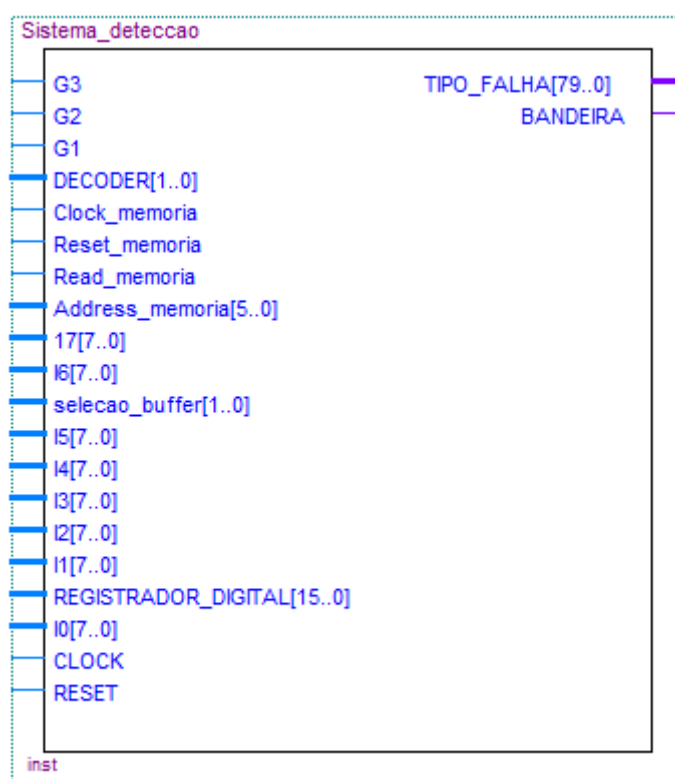


Figura 6.11 - Bloco esquemático topo do projeto hierárquico

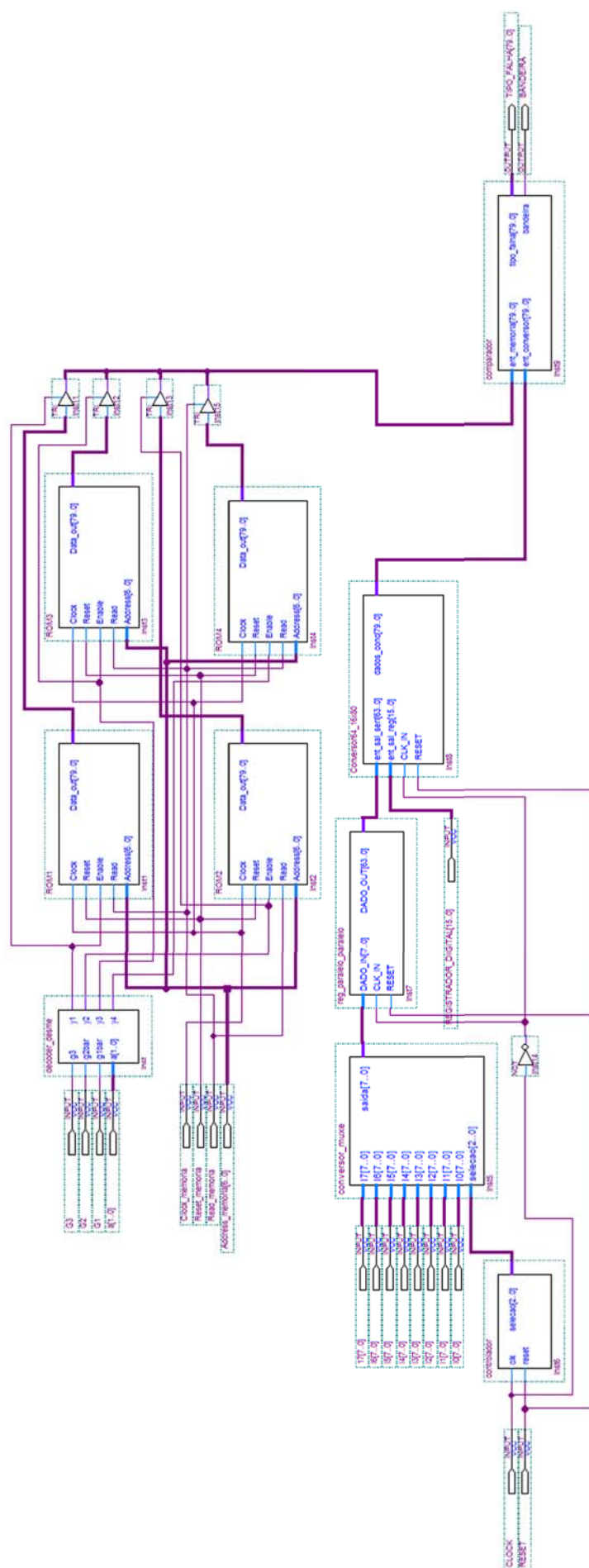


Figura 6.12- Diagrama esquemático para o Sistema de Detecção de Faltas

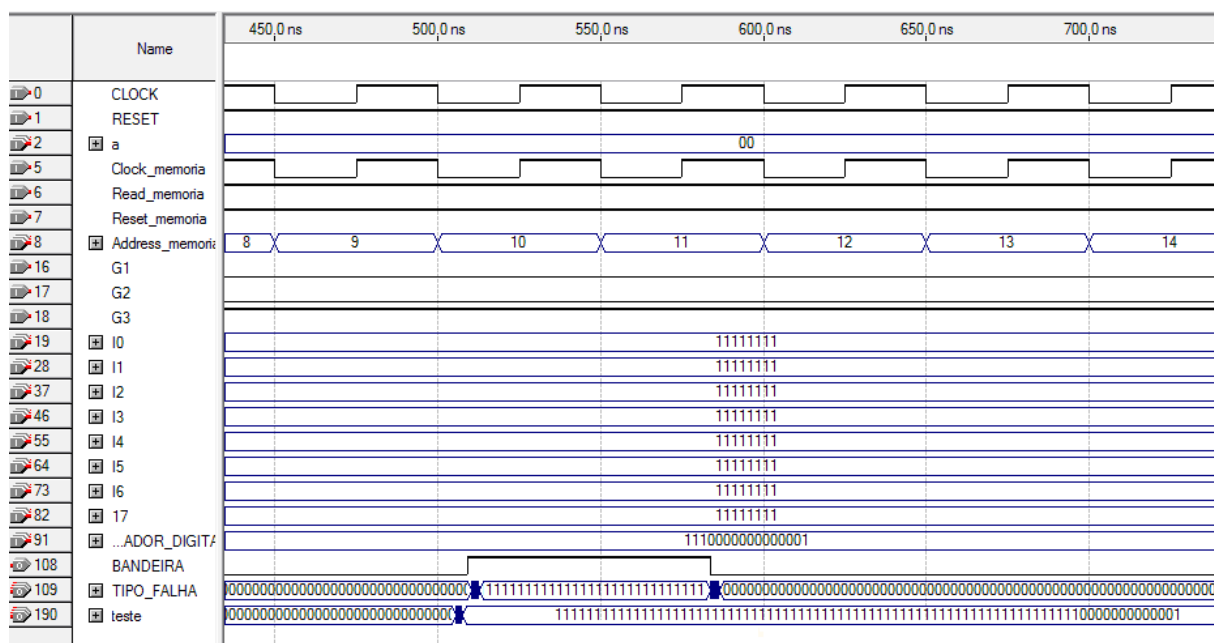


Figura 6.13- Simulação completa(1) do Sistema proposto.

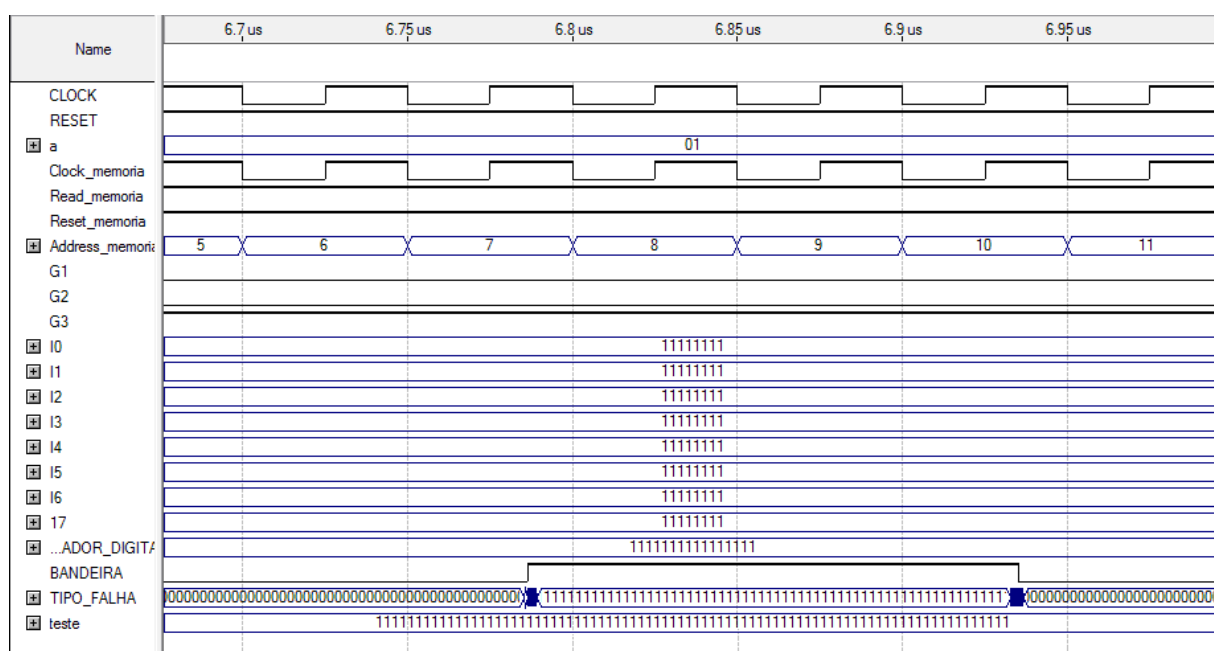


Figura 6.14- Simulação completa (2) do Sistema proposto.

Pelo resultado da simulação da fig. 6.13 observa-se que, para uma memória selecionada no caso MEM1 onde **a= '00'**, a entrada **Address_memoria** está selecionando o endereço 10 o dado armazenado correspondente de falta é mostrado na saída **TIPO_FALHA**, que está sendo comparado às informações de **I₀₋₇** juntamente com as informações do **Registrador Digital**, que formam o registrador de pré- falta . Desta comparação verifica-se através do sinal de saída **bandeira** em nível alto , que houve uma igualdade com os sinais armazenados na

memória. O mesmo procedimento é visto em 6.14 quando é selecionada a memória 2, onde **a= '01'** quando **Adress_memoria= 7**.

Conclusão

Neste capítulo foram apresentados a arquitetura proposta para a detecção de faltas e os resultados finais da simulação do sistema. Para entender o funcionamento mostra-se um fluxograma que explica o algoritmo do circuito para ter seu objetivo realizado. Os resultados foram apresentados em simulações realizadas em cada bloco que compõe o circuito, permitindo uma análise passo a passo de todo o processamento. O ambiente de desenvolvimento é o Quartus II (Altera).

Capítulo 7

Conclusões Gerais e Sugestões para Trabalhos Futuros

7.1-Conclusões Gerais

Baseado em possíveis danos que uma SE possa sofrer e a necessidade de ferramentas para tomada de decisão rápida, foi proposta nesta dissertação o desenvolvimento de um circuito capaz de detectar faltas e classificá-las dentro de um sistema de distribuição de energia em um tempo otimizado.

Buscou-se estabelecer procedimentos que permitissem realizar tal tarefa de forma que os resultados fossem satisfatórios e a sua execução se realizasse de forma rápida e eficiente, conseqüentemente, obtendo-se respostas também rápidas.

A metodologia usada neste trabalho tornou-se possível devido aos recursos e a capacidade dos dispositivos programáveis que aumentam a cada dia, e ao uso de uma linguagem de descrição de hardware. Os dispositivos FPGAs apresentam uma performance muito boa, dando ao projeto um entendimento amplo e uma fácil atualização.

Pelos resultados mostrados, considera-se que o sistema apresenta um tempo de resposta rápido na indicação de ocorrência de um evento de falta, o que permite avaliar que a metodologia empregada pode ser eficaz se implementada em uma SE real. Para um pulso de *clock* de 50ns, obteve-se na simulação do processo completo um ciclo de 32 μ s.

7.2-Sugestões Para Trabalhos Futuros

Visando aperfeiçoar os resultados obtidos do sistema em questão, sugerem-se alguns trabalhos futuros:

- 1º. Usar outros dispositivos com mais recursos para proceder novas simulações e testes;
- 2º. Simulação completa utilizando o software ATP para simular faltas juntamente com o software MATLAB, onde o MATLAB interpretará o sistema unifilar desenvolvido no software ATP, assim podendo trabalhar com dados mais reais simultaneamente;
- 3º. Inserir mais conversores para tornar as medidas mais rápidas;
- 4º. Utilizar um display para visualização do alerta, datas, e informação por extenso da falta ocorrida;
- 5º. Realizações de testes em uma SE real, adquirindo-se novos parâmetros.

Referências

ACKERMAN, W. J. Fundamentals of automation systems & current trends in substations. **Substation automation tutorial sponsored by IEEE Power Engineering Society**, São Paulo, v.1, n.1, p.15, 2006.

ACKERMAN, W. J. Substation and the EMS. **IEEE Power Engineering Society**, New York, v.1, n.1, p. 274 -279, 1999.

ANDERSON, P. M. **Power system protection**. Hardcover: IEEE Press Series on Power Engineering, 1999. v.1, p.10.

ANDRÉ, A. S. **Pipeline**: arquiteturas de computadores. [s.l.:s.n.], 2008. p.6

CAMINHA, A. C. **Introdução à proteção dos sistemas elétricos**. 9.ed. São Paulo: Editora Egard Blüher, 2004. p.224.

CARDOSO, A. S. **Desenvolvimento de um gerador de sinais para estimular neuromuscular utilizando tecnologia FPGA**. 2003. 136.f. Dissertação (Engenharia Elétrica) – Faculdade de Engenharia, Universidade Estadual Paulista , Ilha Solteira, 2003.

COORPOTATION ALTERA. **Interfacing to external memory with Altera FPGAs**. San Jose: [s.n. 2004?]. Disponível em: <<http://www.altera.com>>. Acessado em: 28 ag. 2006.

COORPOTATION ALTERA. **Stratix III**. San Jose: [s.n. 2004?]. Disponível em: <<http://www.altera.com/literature/br/br-stratixIII.pdf>>. Acessado em: 10 ag. 2008.

CASILLO, L. A. **Projeto e implementação em FPGA de um processador com conjunto de instrução reconfigurável utilizando FPGA**. 2005. 120.f. Dissertação (Sistemas e Computação) – Faculdade de Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, Natal 2005.

COMPANHIA PAULISTA DE FORÇA E LUZ- CPFL. **Norma técnica 2912**: proteção de redes áreas de distribuição – sobrecorrente. [s.l.], 2003. p.100.

DECANINI, J. G. M. S. **Detecção e classificação de faltas de curto-circuito em sistemas de distribuição de energia elétrica usando lógica nebulosa**. 2008. 110.f. Dissertação (Engenharia Elétrica) – Faculdade de Engenharia, Universidade Estadual Paulista, Ilha Solteira, São Paulo, 2008.

DY-LIACCO, T. E. Real-time computer control of power systems. **Proceedings of the IEEE**, New York , v.62, n. 7, p. 884-891, 1974.

GAUSHELL, D. J.; BLOCK, W. R. SCADA communication techniques and standards. **IEEE Computer Applications in Power**, New York, v.1, n.1, p.1-7, 1993.

GIGUER, S. **Proteção de sistemas de distribuição**. [s.l.]: Sagra, 1988. p.334.

JARDINI, J. A. **Sistemas digitais para automação da geração, transmissão e Distribuição de energia elétrica**. São Paulo: [s.n.], 1996. p.15

[LIYANAGE, K.M.](#); [LIU, C.C.](#); [MUCHLINSKI, S.I.](#); [EIDE, A.](#) Intelligent systems for distribution automation. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS , 1994, San Antonio. **Proceedings...** San Antonio: [s.n.], 1994. v.2, p.1103.

MANTOVANI, S. C. A.; OLIVEIRA, J. R. **Estudos de circuito evolutivo aplicados ao reconhecimento de voz**. 2003. 122.f. Tese (Doutorado)- Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campina, Campinas, 2003.

MARTINS, C. A. P. S. et al **Computação reconfigurável**: conceitos, tendências e aplicações. Minas Gerais: Pontifícia Universidade Católica de Minas Gerais – PUC , 2008. p. 1-64. (Informática Laboratório de Sistemas Digitais e Computadores).

MORETO, M. **Localização de faltas de alta impedância em sistemas de distribuição de energia**: uma metodologia baseada em redes neurais artificiais. 2005. 126.f. Dissertação

(Mestrado Automação e Instrumentação Eletro-Eletrônica)– Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005.

McDONALD, J. D. **Electric power substations engineering**. Boca Raton : CRC Press LLC, 2003. p.8.

PENTEADO NETO, R. A. **Sistemas para detecção de falta de alta impedância e de rompimento de condutores em redes de distribuição de energia elétrica**. 2005. 114 f. Tese (Engenharia e Ciência dos Materiais) – Faculdade de Engenharia, Universidade Federal do Paraná, Curitiba, 2005.

NASCIMENTO, P. S. B. et al. **Sistemas reconfiguráveis: novo paradigma para o desenvolvimento de aplicações de computação massiva de dados**. Recife: Centro de Informática – UFPE, 2008. p.1-17.

NATIONAL SEMICONDUCTOR. **ADC0808/ADC0809: 8 bit up Compatible A/D Converters with 8 – Channle Multiplexer**. [s.l.: s.n. 2002?]. Disponível em: <<http://www.national.com/mpf/DC/ADC0808.html>>. Acessado em 05 de setembro de 2002.

PERLAZA, D.; DELGADO, A. **Detección de fallas em sistemas de potência com chip ADN em FPGA**. Bogotá: Departamento de Ingeniería Eléctrica y Electrônica-Universidad Nacional de Colômbia, 2005. p.7.

PERRY, D. L. **VHDL: programmy by example**. New York : [s.n.]: 2002. p.12

PONTE, D. F.; MADEIRA, C. S. Aplicação da linguagem descritiva de hardware no ensino de circuitos digitais. In: CONGRESSO DE PESQUISA E INOVAÇÃO DA REDE NORTE NORDESTE DE EDUCAÇÃO TECNOLÓGICA,1, 2006, Natal. **Anais...** Natal: CEFET-RN. p.1-12. 1 CD-ROM.

SAMPAIO, R. F. **Sistema de diagnóstico de faltas para subestações baseados em redes de petri coloridas**. 2004. 143 f. Dissertação (Engenharia Elétrica) –Faculdade de Engenharia Elétrica , Universidade Federal do Ceará, Fortaleza, 2002.

SEPÚLVEDA, M.; CAMARGO, C.; DELGADO, A. Implementación de chip ADN em FPGA. In: WORKSHOP IBERCHIP, 10, 2004, Cartagena. **Proceedings...** Bogotá: UNIANDES, 2004. p.7.

SANTOS, F. G. et al. **Sistema de diagnóstico de faltas para sistemas elétricos baseado em redes de petri colorida.** [s.l.: s.n.], 2007. p.5.

SOUZA, F. A.; OKI, N.; MANTOVANI, S. C. A. Aplicando FPGA em diagnósticos de faltas em subestações de distribuição de energia elétrica. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND APPLICATIONS - DINCON, 7, 2008, Presidente Prudente. **Proceedings...** Presidente Prudente: [s.n], 2008. 1 CD/ROM.

SATO, F. **Proteção de sistemas de energia elétrica.** Campinas: Engenharia Elétrica e da Computação-UNICAMP. 2003. v2, p.6.

Apêndice A.

Decodificador

Apresenta-se neste apêndice o código do decodificador (1x4) das 4 memórias, ou seja, a seleção de 0 a 3, programado na linguagem de descrição de hardware, VHDL no ambiente de projeto QUARTUS II.

Decoder_desme.vhd

```
library ieee;
use ieee.std_logic_1164.all;
USE IEEE.std_logic_unsigned.ALL;

entity decoder_desme is
  port(
    g3,g2bar,g1bar : in bit;          -- entradas de habilita
    a : in bit_vector(1 downto 0);    --entradas binária de seleção
    y1,y2,y3,y4 : out bit
  );--saídas do decoder
end decoder_desme;

architecture tabela of decoder_desme is
  signal habilita : bit_vector(4 downto 0); --combina enables

begin
  habilita <= g3 & g2bar & g1bar & a;
process(habilita)
```

```
begin
case habilita is
  when "10000" =>
    y4<='0';y3<='0';y2<='0';y1<='1';
    when "10001" =>
      y4<='0';y3<='0';y2<='1';y1<='0';
      when "10010" =>
        y4<='0';y3<='1';y2<='0';y1<='0';
        when "10011" =>
          y4<='1';y3<='0';y2<='0';y1<='0';
  when others =>
    y4<='0';y3<='0';y2<='0';y1<='0';

end case;
end process;
end tabela;
```


Apêndice B

Conversor

O código em VHDL do conversor baseia-se em um conversor A/D com MUX, para simular sua operação e testar a unidade de controle do projeto. As saídas deste bloco são multiplexadas e cada pulso de clock libera um dado de 8bits.

Conversor_mux.vhd

```
library ieee;
use ieee.std_logic_1164.all;

entity conversor_mux is
port(
    I7:    in std_logic_vector(7 downto 0);
    I6:    in std_logic_vector(7 downto 0);
    I5:    in std_logic_vector(7 downto 0);
    I4:    in std_logic_vector(7 downto 0);
    I3:    in std_logic_vector(7 downto 0);
    I2:    in std_logic_vector(7 downto 0);
    I1:    in std_logic_vector(7 downto 0);
    I0:    in std_logic_vector(7 downto 0);
    Selecao: in std_logic_vector(2 downto 0);
    saida:  out std_logic_vector(7 downto 0)
);
```

```
end conversor_muxe;
```

```
architecture ok of conversor_muxe is
```

```
begin
```

```
  process(I7,I6,I5,I4,I3,I2,I1,I0,Selecao)
```

```
  begin
```

```
    -- use case statement
```

```
    case Selecao is
```

```
      when "000" =>      saida <= I7;
```

```
      when "001" =>      saida <= I6;
```

```
      when "010" =>      saida <= I5;
```

```
      when "011" =>      saida <= I4;
```

```
        when "100" => saida <= I3;
```

```
        when "101" => saida <= I2;
```

```
        when "110" => saida <= I1;
```

```
        when "111" => saida <= I0;
```

```
      when others =>      saida <= "ZZZZZZZZ";
```

```
    end case;
```

```
  end process;
```

```
end ok;
```

Apêndice C.

Registrador Paralelo/ Paralelo 8/64

O código a seguir , em VHDL , registra os dados vindo do conversor A/D. Possui um canal de entrada paralelo com 8 bits de informação, que armazena a cada ciclo de clock um string de 8 bits, completando 8 sinais de 8 bits cada. Após os 8 ciclos de clock, os dados são enviados para a saída em um string contendo 64 bits de informações. Para reiniciar basta colocar o **RESET** em nível baixo.

Reg_paralelo_paralelo.vhd

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.std_logic_arith.ALL;
USE IEEE.std_logic_signed.ALL;
USE IEEE.std_logic_unsigned.ALL;

ENTITY reg_paralelo_paralelo IS
    PORT(
        DADO_IN      :    IN      STD_LOGIC_VECTOR(7 DOWNT0 0);
        CLK_IN       :    IN      STD_LOGIC;
        RESET        :    IN      STD_LOGIC;
        DADO_OUT      :    OUT     STD_LOGIC_VECTOR(63 DOWNT0 0) );
```

```
END reg_paralelo_paralelo;
```

```
ARCHITECTURE a OF reg_paralelo_paralelo IS
```

```
SIGNAL      AUX      :      STD_LOGIC_VECTOR(55 DOWNT0 0);
```

```
SIGNAL D0,D1,D2,D3,D4,D5,D6,D7 : STD_LOGIC_VECTOR (7 DOWNT0 0);
```

```
SIGNAL CONTADOR : NATURAL RANGE 0 TO 7;
```

```
BEGIN
```

```
PROCESS (CLK_IN,RESET)
```

```
BEGIN
```

```
D0 <="00000000";D1 <="00000000";D2 <="00000000";D3 <="00000000"; D4 <="00000000"; D5  
<="00000000";D6 <="00000000";D7 <="00000000";
```

```
IF reset = '0' THEN
```

```
    DADO_OUT <= (OTHERS => '0');
```

```
    AUX <= (OTHERS => '0');
```

```
    CONTADOR <= 0;
```

```
    --ENABLE_CONT<='0';
```

```
ELSIF CLK_IN'EVENT AND CLK_IN = '1' THEN
```

```
CASE CONTADOR IS
```

```
    WHEN 0 =>
```

```
        AUX(55 DOWNT0 48) <= DADO_IN(7 DOWNT0 0);      --Em um string de 64 ele  
armazena nos ultimos bits sendo do mais para o menos significativo
```

```
        --REGISTRO <= DADO_IN & D1 & D2 & D3 & D4 & D5 & D6 & D7;
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 1 =>
```

```
        AUX(47 DOWNT0 40) <= DADO_IN(7 DOWNT0 0);
```

```
        --REGISTRO <= D0 & DADO_IN & D2 & D3 & D4 & D5 & D6 & D7;
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 2 =>
```

```
        AUX(39 DOWNT0 32) <= DADO_IN(7 DOWNT0 0);
```

```
        --REGISTRO <= D0 & D1 & DADO_IN & D3 & D4 & D5 & D6 & D7;
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 3 =>
```

```
        AUX(31 DOWNT0 24) <= DADO_IN(7 DOWNT0 0);
```

```
        --REGISTRO <= D0 & D1 & D2 & DADO_IN & D4 & D5 & D6 & D7;
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 4 =>
```

```
        AUX(23 DOWNT0 16) <= DADO_IN(7 DOWNT0 0);
```

```
        --REGISTRO <= D0 & D1 & D2 & D3 & DADO_IN & D5 & D6 & D7;
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 5 =>
```

```
        AUX(15 DOWNT0 8) <= DADO_IN(7 DOWNT0 0);
```

```
        --REGISTRO <= D0 & D1 & D2 & D3 & D4 & DADO_IN & D6 & D7;
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 6 =>
```

```
        AUX(7 DOWNT0 0) <= DADO_IN(7 DOWNT0 0);
```

```
        --REGISTRO <= D0 & D1 & D2 & D3 & D4 & D5 & DADO_IN & D7;
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 7 =>
```

```
        DADO_OUT <= AUX&DADO_IN(7 DOWNT0 0);
```

```
        --REGISTRO <= D0 & D1 & D2 & D3 & D4 & D5 & D6 & DADO_IN;
```

```
        CONTADOR <= 0;
```

```
        --ENABLE_CONT<='1';
```

```
    WHEN OTHERS =>
```

```
END CASE;  
END IF;
```

```
END PROCESS ;
```

```
END a;
```

Apêndice D.

Memórias ROM

O código das memórias ROMs - Ready Only Memory 128X80 a seguir, permite a leitura dos dados que constituem nas possíveis faltas que podem ocorrer dentro de uma SE. Para armazenar previamente esses dados na ROM pode ser utilizado os recursos do software da ALTERA, que aceita um arquivo em texto .mif – Memory Initial File, que é carregado no programa quando da sua compilação. Atendendo a capacidade, a seleção dos endereços é realizada através de 7 bits.

Memória ROM 1

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity ROM1 is
port(
    Clock   : in std_logic;
    Reset   : in std_logic;
    Enable  : in std_logic;
    Read    : in std_logic;
    Address : in std_logic_vector(6 downto 0);
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
begin
    process(Clock, Reset, Read, Address)
    begin
        if( Reset = '0' ) then
            --if (enable='0') then
                Data_out <=
                    "////////////////////////////////////";
            elsif( Clock'event and Clock = '1' ) then
                if Enable = '1' then
                    if( Read = '1' ) then
                        Data_out <= Content(conv_integer(Address));
                    else
                        Data_out <=
                            "////////////////////////////////////";
                    end if;
                end if;
            end if;
        --end if;
    end process;
end Behav;
```

Apêndice E

Comparador

O programa em VHDL compara a cada ciclo de clock (20M Hz), os sinais reais de falhas (80 bits) com as informações de pré-falhas (80 bits) armazenadas nas memórias ROMs. A saída (**tipo_falha**) mostra o string que acusa uma falha e a saída **bandeira** acende um led informando.

Comparador.vhd

```
LIBRARY IEEE;
use ieee.std_logic_1164.all;
USE IEEE.std_logic_signed.ALL;
USE IEEE.std_logic_unsigned.ALL;

entity comparador is
    port(
        ent_memoria: IN std_logic_vector (79 downto 0);
        ent_conversor: IN std_logic_vector (79 downto 0);
        tipo_falha: OUT std_logic_vector (79 downto 0);
        bandeira:buffer std_logic);

end comparador;

Architecture func of comparador is
```



```
signal dna2: std_logic_vector(79 downto 0);  
signal band2: std_logic;  
begin  
process (dna2,band2,ent_memoria,ent_conversor)  
  
begin  
dna2 <=not (ent_memoria xor ent_conversor); -- Nesta linha é simulada a porta lógica XNOR  
band2<=(  
dna2(0)and dna2(1)and dna2(2)and dna2(3)and dna2(4)and dna2(5)and dna2(6)and dna2(7)and dna2(8)and  
dna2(9)and dna2(10)and dna2(11) and dna2(12)and dna2(13)and dna2(14)and dna2(15)and dna2(16)and  
dna2(17)and Ana2(18)and dna2(19)and dna2(20)and dna2(21)and dna2(22)and dna2(23)and dna2(24)and  
dna2(25)and dna2(26)and dna2(27)and dna2(28)and dna2(29)and dna2(30)and dna2(31)and dna2(32)and  
dna2(33)and dna2(34)and dna2(35)and dna2(36)and dna2(37)and dna2(38)and dna2(39)and dna2(40)and  
dna2(41)and dna2(42)and dna2(43)and dna2(44)and dna2(45)and dna2(46)and dna2(47)and dna2(48)and  
dna2(49)and dna2(50)and dna2(51)and dna2(52)and dna2(53)and dna2(54)and dna2(55)and dna2(56)and  
dna2(57)and dna2(58)and dna2(59)and dna2(60)and dna2(61)and dna2(62)and dna2(63)and dna2(64)and  
dna2(65)and dna2(66)and dna2(67)and dna2(68)and dna2(69)and dna2(70)and dna2(71)and dna2(72)and  
dna2(73)and dna2(74)and dna2(75)and dna2(76)and dna2(77)and dna2(78)and dna2(79)); -- A porta AND é  
utilizada para acionar a bandeira.  
  
if (band2='1') then  
    bandeira<=band2;  
    tipo_falha<=ent_memoria;  
else  
    bandeira<='0';  
  
    tipo_falha<="0000000000000000000000000000000000000000000000000000000000000000"  
00000000";  
end if;  
end process;  
end func;
```

Apêndice F

Conversor 64_16x80

Este código de programa tem a função de registrar os 64 bits da saída do registrador paralelo_ paralelo e os 16bits (sinais dos relés de proteção digital) em um único string de 80 bits.

Conversor64_16x80.vhd

```

LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.std_logic_arith.ALL;
USE IEEE.std_logic_signed.ALL;
USE IEEE.std_logic_unsigned.ALL;

ENTITY Conversor64_16x80 IS
    PORT(
        ent_sai_seri      :IN      STD_LOGIC_VECTOR(63 DOWNTO 0);
        ent_sai_reg       :IN      STD_LOGIC_VECTOR(15 DOWNTO 0); -- Dados de Entrada
        CLK_IN            :IN      STD_LOGIC;
        RESET             :IN      STD_LOGIC;
        dados_conc        :OUT     STD_LOGIC_VECTOR(79 DOWNTO 0) Dados de saída de 64
        bits concatenados
    );

```

```
END Conversor64_16x80;
```

```
ARCHITECTURE a OF Conversor64_16x80 IS
```

```
SIGNAL      AUX      :      STD_LOGIC_VECTOR(79 DOWNTO 0);
```

```
SIGNAL CONTADOR : NATURAL RANGE 0 TO 8;
```

```
BEGIN
```

```
PROCESS (CLK_IN,RESET)
```

```
BEGIN
```

```
IF reset = '0' THEN
```

```
    dados_conc <= (OTHERS => '0');
```

```
    AUX <= (OTHERS => '0');
```

```
    CONTADOR <= 0;
```

```
ELSIF CLK_IN'EVENT AND CLK_IN = '1' THEN
```

```
CASE CONTADOR IS
```

```
    WHEN 0 =>
```

```
        AUX(79 DOWNTO 0) <= ent_sai_seri(63 DOWNTO 0)& ent_sai_reg(15 DOWNTO 0);
```

```
--Em um string de 64 ele armazena nos ultimos bits sendo do mais para o menos significativo
```

```
        dados_conc <= AUX(79 DOWNTO 0);
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 1 =>
```

```
        AUX(79 DOWNTO 0) <= ent_sai_seri(63 DOWNTO 0)& ent_sai_reg(15 DOWNTO 0);
```

```
        dados_conc <= AUX(79 DOWNTO 0);
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 2 =>
```

```
        AUX(79 DOWNTO 0) <= ent_sai_seri(63 DOWNTO 0)& ent_sai_reg(15 DOWNTO 0);
```

```
        dados_conc <= AUX(79 DOWNTO 0);
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 3 =>
```

```
        AUX(79 DOWNTO 0) <= ent_sai_seri(63 DOWNTO 0)& ent_sai_reg(15 DOWNTO 0);
```

```
        dados_conc <= AUX(79 DOWNTO 0);
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 4 =>
```

```
        AUX(79 DOWNTO 0) <= ent_sai_seri(63 DOWNTO 0)& ent_sai_reg(15 DOWNTO 0);
```

```
        dados_conc <= AUX(79 DOWNTO 0);
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 5 =>
```

```
        AUX(79 DOWNTO 0) <= ent_sai_seri(63 DOWNTO 0)& ent_sai_reg(15 DOWNTO 0);
```

```
        dados_conc <= AUX(79 DOWNTO 0);
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 6 =>
```

```
        AUX(79 DOWNTO 0) <= ent_sai_seri(63 DOWNTO 0)& ent_sai_reg(15 DOWNTO 0);
```

```
        dados_conc <= AUX(79 DOWNTO 0);
```

```
        CONTADOR <= CONTADOR + 1;
```

```
    WHEN 7 =>
```

```
        AUX(79 DOWNTO 0) <= (ent_sai_seri(63 DOWNTO 0)& ent_sai_reg(15 DOWNTO 0));
```

```
--Em um string de 64 ele armazena nos ultimos bits sendo do mais para o menos significativo
```

```
        dados_conc <= AUX(79 DOWNTO 0);
```

```
        CONTADOR <=0;
```

```
--WHEN 8 =>
```

```
        --dados_conc <= AUX(79 DOWNTO 0);
```

```
        --CONTADOR <= 0;
```

```
    WHEN OTHERS =>
```

```
        END CASE;  
    END IF;  
    END PROCESS ;  
END a;
```

Apêndice G

Controlador

Este programa faz a seleção dos canais do conversor implementado ambos em linguagem VHDL.

controlador.vhd

```
library ieee;
use ieee.std_logic_1164.all;
USE IEEE.std_logic_unsigned.ALL;

entity controlador is
port(
    clk,reset: in std_logic;
    selecao: out std_logic_vector(2 downto 0));
end controlador;

architecture cont of controlador is
    type etapas is (e0,e1,e2,e3,e4,e5,e6,e7);
    signal pr_etapa,siga_etapa: etapas;

begin

    ----- Processo das Etapas -> Inicial -----
    acesso:process(reset,clk)

        begin
            if (reset='0') then                -- reseta as etapas iniciando pela etapa1
                pr_etapa<=e0;                  -- etapa inicial caso seja resetado
            end if;
        end process;
    end architecture;
```

```

        elsif (clk'EVENT and clk='1') then -- cada evento do clock
            pr_etapa<=siga_etapa; --siga_etapa;
            end if;
    end process acesso;

    etapas_i: process(pr_etapa,siga_etapa)
    begin

        case pr_etapa is

            ----- etapa 1 -----
                when e0 =>
                    selecao<="000";
                    siga_etapa<=e1;
            ----- etapa 2 -----
                when e1 =>
                    selecao<="001";
                    siga_etapa<=e2;
            ----- etapa 3 -----
                when e2 =>
                    selecao<="010";
                    siga_etapa<=e3;
            ----- etapa 4 -----
                when e3 =>
                    selecao<="011";
                    siga_etapa<=e4;
            ----- etapa 5 -----
                when e4 =>
                    selecao<="100";
                    siga_etapa<=e5;
            ----- etapa 6 -----
                when e5 =>
                    selecao<="101";
                    siga_etapa<=e6;

            ----- etapa 7 -----
                when e6 =>
                    selecao<="110";
                    siga_etapa<=e7;
            ----- etapa 8 -----
                when e7 =>
                    selecao<="111";
                    siga_etapa<=e0;

            ----- Finalizando processo -----

                when OTHERS =>
                    end case;
        end process etapas_i;
    end cont;

```

Anexo I.

Artigo Publicado

SOUZA, F. A; OKI, N.; MANTOVANI, S. C. A. Aplicando FPGA em diagnósticos de faltas em subestações de distribuição de energia elétrica. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND APPLICATIONS DINCON, 7, 2008, Presidente Prudente. *Proceedings* DINCON Presidente Prudente, 2008, p. 6 CD/ROM.