

**HARDWARE DEDICADO À PROTEÇÃO
DIFERENCIAL DE TRANSFORMADORES DE
POTÊNCIA**

Ivo Silveira dos Santos Filho

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**HARDWARE DEDICADO À PROTEÇÃO DIFERENCIAL DE
TRANSFORMADORES DE POTÊNCIA**

IVO SILVEIRA DOS SANTOS FILHO

ORIENTADOR: KLEBER MELO E SILVA

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGEE.DM - 649/2016

BRASÍLIA/DF: DEZEMBRO – 2016

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

HARDWARE DEDICADO À PROTEÇÃO DIFERENCIAL DE
TRANSFORMADORES DE POTÊNCIA

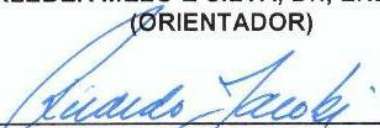
IVO SILVEIRA DOS SANTOS FILHO

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:



KLEBER MELO E SILVA, Dr., ENE/UNB
(ORIENTADOR)



RICARDO PEZZUOL JACOBI, Dr., CIC/UNB
(EXAMINADOR EXTERNO)



FELIPE VIGOLVINO LOPES, Dr., ENE/UNB
(EXAMINADOR INTENO)

Brasília, 12 de Dezembro de 2016.

FICHA CATALOGRÁFICA

SANTOS FILHO, IVO SILVEIRA

Hardware Dedicado à Proteção Diferencial de Transformadores de Potência [Distrito Federal] 2016.

xv, 111p., 210 x 297 mm (ENE/FT/UnB, Mestre, Dissertação de Mestrado, 2016)

Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Proteção de Transformadores

2. Proteção diferencial

3. ATP

4. FPGA

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

SANTOS FILHO, I. S. (2016). *Hardware Dedicado à Proteção Diferencial de Transformadores de Potência*. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGEE.DM-649/2016, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 111p.

CESSÃO DE DIREITOS

AUTOR: Ivo Silveira dos Santos Filho.

TÍTULO: *Hardware Dedicado à Proteção Diferencial de Transformadores de Potência*.

GRAU: Mestre

ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Ivo Silveira dos Santos Filho
Universidade de Brasília –UnB
Campus Darcy Ribeiro
Faculdade de Tecnologia – FT
Departamento de Engenharia Elétrica
Brasília – DF
CEP: 70910-900

Dedico este trabalho aos meus pais e irmãos,
à Carmem, ao Samuel e à Laura.

AGRADECIMENTOS

Ao Professor Kleber Melo, pelos ensinamentos e pela oportunidade e confiança depositada.

À Sra. Maria de Jesus, minha amada mãe, por todos sacrifícios e incentivos que dispensou para minha educação.

À Carmem, minha esposa, por todo companheirismo, doação, amor e carinho ofertados ao longo da jornada.

Ao Samuel e à Laura, por toda alegria e festa que trazem aos meus dias.

Aos meus leais familiares, por completarem meu time no conforto de sua amizade.

E a Deus, verdadeiro amigo e Pai, por ter me dado a sorte de dividir o caminho com pessoas tão especiais.

RESUMO

Apresentam-se nessa dissertação os resultados da primeira implementação em *hardware* dos algoritmos de proteção diferencial de transformadores pesquisados no âmbito do Laboratório de Proteção de Sistemas Elétricos – LAPSE da Universidade de Brasília.

Foram desenvolvidos dois dispositivos: um *hardware* microprocessado e um SOC (*System on a Chip*) em FPGA (*Field Programmable Gate Array*).

Por intermédio de casos simulados no software ATP (*Alternative Transient Program*), foram testados os comportamentos das funções diferenciais de fase (87T), de sequência negativa (87Q) e de falta à terra restrita (87REF), bem como o bloqueio por harmônicos capaz de detectar correntes de *inrush*.

Os resultados obtidos indicam desempenho satisfatório no que concerne à velocidade e robustez verificados em dispositivos disponíveis no mercado. Além disso, o trabalho desenvolvido apresenta metodologia de desenvolvimento de *hardware* que pode ser aplicada a outros algoritmos de proteção de sistemas de potência desenvolvidos no LAPSE, além da proteção diferencial de transformadores, permitindo a verificação de suas eficácias frente a situações reais ou simuladas.

PALAVRAS-CHAVE: Proteção de Transformadores, Proteção Diferencial, ATP, *Hardware* de Proteção Digital, FPGA.

ABSTRACT

The results of the first hardware implementation of the transformer differential protection algorithms researched in the scope of the Electrical Systems Protection Laboratory - LAPSE of the University of Brasilia are presented in this dissertation.

Two devices were developed: a microprocessor-based hardware and a FPGA (Field Programmable Gate Array) SOC (System on a Chip).

Through simulated cases in software ATP (Alternative Transient Program), the behavior of the phase differential function (87T), the negative sequence (87Q) and restricted earth fault (87REF) have been tested, as well as a blocking harmonic module able to detect inrush currents.

The results indicate satisfactory performance with regard to the speed and robustness verified in devices available in the market. In addition, the developed work presents methodology of hardware development that can be applied to other algorithms of protection of power systems developed in the LAPSE, besides the differential protection of transformers, allowing the verification of their efficacies through real or simulated situations.

INDEX TERMS: Transformer Protection, Differential Protection, ATP, Digital Protection Hardware, FPGA.

SUMÁRIO

CAPÍTULO 1.....	1
INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO DO TEMA	1
1.2. OBJETIVOS.....	2
1.3. ORGANIZAÇÃO DO TEXTO	2
CAPÍTULO 2.....	4
REVISÃO BIBLIOGRÁFICA.....	4
CAPÍTULO 3.....	7
FUNDAMENTAÇÃO DA PROTEÇÃO DIFERENCIAL DE TRANSFORMADORES DE POTÊNCIA	7
3.1. REQUISITOS DA PROTEÇÃO DIFERENCIAL	7
3.2. PROTEÇÃO DIFERENCIAL (87T).....	8
3.3. FATORES QUE INFLUENCIAM A PROTEÇÃO DIFERENCIAL DE TRANSFORMADORES	10
3.3.1 <i>Relação de Transformação</i>	10
3.3.2 <i>Defasagem Angular</i>	11
3.3.3 <i>Compensação da Corrente de Sequência Zero</i>	12
3.3.4 <i>Restrição e Bloqueio por Harmônicos</i>	12
3.4. PROTEÇÃO DIFERENCIAL DE SEQUÊNCIA NEGATIVA (87Q)	17
3.5. PROTEÇÃO DE FALTA À TERRA RESTRITA (87REF).....	18
3.6. ESTIMAÇÃO DE FASORES	19
CAPÍTULO 4.....	20
HARDWARE DE PROTEÇÃO DIFERENCIAL.....	20
4.1. DESENVOLVIMENTO NA PLATAFORMA RASPBERRY PI	20
4.1.1 <i>Modelagem Em Software das Funções de Proteção diferencial</i>	22
4.2. DESENVOLVIMENTO DO RELÉ DE PROTEÇÃO DIFERENCIAL DE TRANSFORMADOR EM FPGA	24
4.2.1 <i>Estrutura geral do sistema</i>	25
4.2.2 <i>Módulos Matemáticos</i>	27
4.2.3 <i>Módulos de Memórias RAM</i>	36
4.2.4 <i>Barramento de Acesso às Memórias RAM</i>	38
4.2.1 <i>Barramento de Acesso ao Módulo MSA</i>	39
4.2.2 <i>Barramento de Acesso ao Módulo SQRT</i>	41
4.2.3 <i>Módulo de Buffer</i>	41
4.2.4 <i>Módulo Filtro Cosseno</i>	43
4.2.1 <i>Módulo Compensador</i>	47
4.2.2 <i>Módulo F87T</i>	49
4.2.3 <i>Módulo F87Q</i>	52
4.2.4 <i>Módulo F87REF</i>	55
4.2.5 <i>Módulo de Bloqueio por Harmônicos</i>	57
4.2.6 <i>Módulo de Tratamento de TRIPS</i>	59
4.2.7 <i>Módulos de Extração de Dados</i>	60

4.2.8	<i>Módulo Maestro</i>	61
4.2.9	<i>Sistema Completo</i>	64
CAPÍTULO 5		68
APRESENTAÇÃO E ANÁLISE DOS RESULTADOS		68
5.1.	PLATAFORMA DE TESTES.....	68
5.2.	SISTEMA ANALISADO.....	69
5.3.	CONFIGURAÇÃO DO RELÉ DESENVOLVIDO.....	71
5.4.	ANÁLISE DOS CASOS	73
5.4.1	<i>Curto Monofásico Fase-Terra, dentro da Região Protegida, no lado de 230 kV</i>	76
5.4.2	<i>Curto Circuito Espira-Espira no Lado de 69 kV</i>	80
5.4.3	<i>Energização do Transformador</i>	83
5.4.4	<i>Energização do Transformador com Curto Circuito Interno Espira-Espira</i>	86
CAPÍTULO 6		89
CONCLUSÕES E PROPOSTAS DE TRABALHOS FUTUROS		89
REFERÊNCIAS BIBLIOGRÁFICAS		91

LISTA DE FIGURAS

Figura 3.1 - Condição Normal de Operação.	8
Figura 3.2 - Condição de Falta Interna ao Transformador.	8
Figura 3.3 – Plano de Operação - Restrição do Elemento de Proteção Diferencial Percentual de Fase.	10
Figura 3.4 - Grupos de Defasagem Angular [24].	11
Figura 3.5 - Fenômeno das Correntes de <i>Inrush</i> [27]	14
Figura 3.6 - Corrente de <i>Inrush</i> [27].	14
Figura 3.7 - Lógica de Bloqueio por Harmônicos Independente.	15
Figura 3.8 - Lógica de Bloqueio por Harmônicos Cruzado.	15
Figura 3.9 - Plano de Operação – Restrição, com Restrição por Harmônicos.	16
Figura 3.10 – Plano de Operação – Restrição do Elemento de Proteção Diferencial de Sequência Negativa.	17
Figura 3.11 - Lógica de Bloqueio por Harmônicos da Proteção Diferencial de Sequência Negativa (87Q).	18
Figura 4.1 - Modelo Conceitual no qual o Relé Desenvolvido se Insere.	21
Figura 4.2 - Diagrama descritivo do sistema modelado e implementado.	22
Figura 4.3 - Elemento Lógico dos FPGAs da Família Cyclone IV [38].	25
Figura 4.4 – Quartus II - Ambiente de Desenvolvimento para FPGAs do Fabricante Altera.	25
Figura 4.5 - Especificação Simplificada e Inicial do Sistema.	27
Figura 4.6 - Representação de Valores em Ponto Flutuante, padrão IEEE 754.	28
Figura 4.7 - Módulos Aritméticos para Implementação da Estimção de Fasores.	32
Figura 4.8 - Módulo MSA sintetizado no <i>hardware</i> do Relé de Proteção Diferencial de Transformadores.	34
Figura 4.9 - Diagrama de Tempo do Módulo Aritmético MSA.	35
Figura 4.10 - Módulo de Memória RAM.	36
Figura 4.11 - Diagrama de Tempo dos Módulos de Memória RAM.	37
Figura 4.12 - Esquemático do Barramento_RAM_0_1.	39
Figura 4.13 - Esquemático do Barramento_RAM_2.	39
Figura 4.14 - Esquemático do Barramento_MSA.	40
Figura 4.15 - Esquemático do Barramento_SQRT.	40
Figura 4.16 – Esquemático simplificado do módulo F5.	42
Figura 4.17 - Diagrama de Tempo da MSA ao Efetuar o Cálculo da Parte Real do Fasor. ...	44
Figura 4.18 - Blocos de Armazenamento dos Vetores Estimados.	44

Figura 4.19 - Vetor de Posições Atuais dos Fasores - VPAF.	45
Figura 4.20 - Endereços de Armazenamento dos Fasores nas Memórias RAM.....	45
Figura 4.21 - Diagrama de Tempo da MSA ao Efetuar o Cálculo da Parte Imaginária do Fasor.....	46
Figura 4.22 - Diagrama Simplificado Esquemático do Módulo FCOS.	47
Figura 4.23 - Diagrama de Tempo da MSA ao Efetuar o Cálculo da Parte Imaginária do Fasor Compensado da fase B.	48
Figura 4.24 - Diagrama Esquemático Simplificado do Módulo Compensador.	49
Figura 4.25 - Diagrama Esquemático Simplificado do Módulo F87T.....	51
Figura 4.26 - Diagrama Esquemático Simplificado do Módulo F87Q.	53
Figura 4.27 - Diagrama Esquemático Simplificado do Módulo F87REF.....	56
Figura 4.28 - Diagrama Esquemático Simplificado do Módulo BLOQH.	59
Figura 4.29 - Lógica de TRIP do Relé Desenvolvido no FPGA.....	60
Figura 4.30 - Diagrama Esquemático do Módulo Extrator de Dados.	61
Figura 4.31 - Diagrama Esquemático Simplificado do Módulo Maestro.	62
Figura 4.32 - Diagrama de Tempo do Módulo Maestro Controlando o Módulo FCOS.....	63
Figura 4.33 - Diagrama de Interconexões dos Módulos Desenvolvidos, no SOC do Relé de Proteção Diferencial de Transformadores.....	64
Figura 5.1 - Sistema Simulado no ATP Utilizado para Analisar o Desempenho do Relé de Proteção Desenvolvido no FPGA.	69
Figura 5.2 - Esquema de Seccionamento dos Enrolamentos de Transformador Simulado. ...	70
Figura 5.3 - Correntes do Lado de 230 kV do Transformador, Medida pelo TC - Falta Monofásica no Lado Primário.....	77
Figura 5.4 - Correntes do Lado de 69 kV do Transformador, Medida pelo TC - Falta Monofásica no Lado Primário.....	77
Figura 5.5 - Estimção Fasorial, no lado de 230 kV - Falta Monofásica no Lado Primário. .	78
Figura 5.6 - Estimção Fasorial, no lado de 69 kV - Falta Monofásica no Lado Primário.....	78
Figura 5.7 - Plano Operacional da Função 87T - Falta Monofásica no Lado Primário.....	78
Figura 5.8 - Plano Operacional da Função 87Q - Falta Monofásica no Lado Primário.	79
Figura 5.9 - Correntes de Operação e Restrição Calculadas para a Função 87REF - Falta Monofásica no Lado Primário.....	79
Figura 5.10 - Variáveis que Compõem a Lógica de TRIP - Falta Monofásica no Lado Primário.....	79
Figura 5.11 - Correntes no Enrolamento Primário, Medidas pelos TCs - Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.	81

Figura 5.12 - Correntes no Enrolamento Secundário, Medidas pelos TCs - Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.	81
Figura 5.13 - Plano Operacional da Função 87T – Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.	81
Figura 5.14 - Plano Operacional da Função 87Q – Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.	82
Figura 5.15 - Correntes de Operação e Restrição Calculadas para a Função 87REF - Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.	82
Figura 5.16 - Variáveis que Compõem a Lógica de TRIP - Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.	82
Figura 5.17 - Correntes de <i>Inrush</i> no Lado de 230 kV do Transformador, Durante a Energização do Transformador.	84
Figura 5.18 - Percentual da Componente de 2ª Harmônica em Relação à Fundamental, Durante a Energização do Transformador.	84
Figura 5.19 - Plano Operacional da Função 87T, Durante a Energização.	84
Figura 5.20 - Plano Operacional da Função 87Q, Durante a Energização.	85
Figura 5.21 - Correntes de Operação e Restrição Calculadas para a Função 87REF, Durante a Energização.	85
Figura 5.22 - Variáveis que Compõem a Lógica de TRIP, Durante a Energização.	85
Figura 5.23 - Correntes de <i>Inrush</i> no Lado Primário do Transformador, Durante a Energização Sob Falta.	87
Figura 5.24 - Plano Operacional da Função 87T, Durante a Energização Sob Falta.	87
Figura 5.25 - Plano Operacional da Função 87Q, Durante a Energização Sob Falta.	87
Figura 5.26 - Correntes de Operação e Restrição Calculadas para a Função 87REF, Durante a Energização Sob Falta.	88
Figura 5.27 - Variáveis que Compõem a Lógica de TRIP, Durante a Energização Sob Falta.	88

LISTA DE TABELAS

Tabela 3.1 - Comparação Entre o Bloqueia e a Restrição por Harmônicos.....	16
Tabela 4.1 - Dados Principais do FPGA Altera Cyclone IV, modelo EP4CE10F17C8N.....	24
Tabela 4.2 - Características do Módulo Somador, de propriedade intelectual da Altera ®...	29
Tabela 4.3 - Características do Módulo Multiplicador, de propriedade intelectual da Altera®.	30
Tabela 4.4 - Características do Módulo Raiz Quadrada (SQRT), de propriedade intelectual da Altera ®.....	31
Tabela 4.5 - Resultados de Possível Módulo de Estimação de Fasores, sintetizado conforme projeto indicado na Figura 4.7.....	33
Tabela 4.6 - Resultados da MSA sintetizada no FPGA adotado no projeto.	35
Tabela 4.7 - Características do Módulo Comparador, de propriedade intelectual da Altera ®.	36
Tabela 4.8 - Módulos de Memória RAM Instanciados.	37
Tabela 4.9 - Uso de Recursos do FPGA nos Barramentos de Acesso às Memórias RAM.....	38
Tabela 4.10 - Uso de Recursos do FPGA no Barramento de Acesso ao Módulo Aritmético MSA.....	40
Tabela 4.11 - Uso de Recursos do FPGA no Barramento de Acesso ao Módulo SQRT.....	41
Tabela 4.12 - Uso de Recursos do FPGA no módulo F5.	42
Tabela 4.13 - Uso de Recursos do FPGA no módulo FCOS.	46
Tabela 4.14 - Uso de Recursos do FPGA no módulo Compensador.	48
Tabela 4.15 - Uso dos Recursos do FPGA pelo Módulo F87T.....	51
Tabela 4.16 - Uso dos Recursos do FPGA pelo Módulo F87Q.	55
Tabela 4.17 - Uso dos Recursos do FPGA pelo Módulo F87REF.....	57
Tabela 4.18 - Uso dos Recursos do FPGA pelo Módulo BLOQH.....	59
Tabela 4.19 – Uso dos Recursos do FPGA pelo módulo TRIP87.	60
Tabela 4.20 - Uso dos Recursos do FPGA pelo Módulo de Extração de Dados da RAM_2.	61
Tabela 4.21 - Seletor de Barramento Adotado para Cada Módulo.	63
Tabela 4.22 - Uso dos Recursos do FPGA pelo Módulo Maestro.	63
Tabela 4.23 - Resumo do Uso do FPGA pelo Sistema Desenvolvido.	65
Tabela 4.24 - Comparação de Resultados Obtidos pela Estimação Fasorial Executada pelo Raspberry Pi e pelo <i>Hardware</i> Desenvolvido pelo FPGA.....	66

Tabela 5.1 - Impedâncias Equivalentes do Sistema Simulado no ATP.	70
Tabela 5.2 - Reatâncias dos Enrolamentos do Transformador Simulado.	70
Tabela 5.3 - Parametrização Adotada no Relé Desenvolvido para o Sistema Elétrico Analisado.....	71
Tabela 5.4 - Resumo das Atuações dos Elementos de Proteção nos Casos Analisados.	73

LISTA DE ALGORITIMOS

Algoritmo 4.1- Método C++ da Estimação Fasorial por Meio do Filtro Cosseno Modificado a ABB	31
Algoritmo 4.2 - Código C++ da Função F87T Implementada no FPGA.....	50
Algoritmo 4.3 - Código C++ da Função F87Q Implementada no FPGA.	52
Algoritmo 4.4 - Código C++ da Função F87REF Implementada no FPGA.....	55
Algoritmo 4.5 - Código C++ do Bloqueio por Harmônicos Implementada no FPGA.	58

GLOSSÁRIO

87	Proteção Diferencial
87T	Proteção Diferencial Percentual de Fase
87TA	Proteção Diferencial Percentual da fase A
87TB	Proteção Diferencial Percentual da fase B
87TC	Proteção Diferencial Percentual da fase C
87Q	Proteção Diferencial de Sequência Negativa
87REF	Proteção Diferencial de Falta à Terra Restrita
A/D	Conversor Analógico/Digital
ATP	<i>Alternative Transient Program</i>
BLOQH	Módulo que implementa o Bloqueio por Harmônicos
CC	Corrente Contínua
CA	Corrente Alternada
DC	<i>Direct Current</i>
F5	Módulo de atualização e <i>buffer</i> das amostras de corrente
F87T	Módulo que implementa a função 87T
F87Q	Módulo que implementa a função 87Q
F87REF	Módulo que implementa a função 87REF
FCOS	Módulo que implementa o Filtro Cosseno Modificado da ABB
FIR	<i>Finite Impulse Response</i>
FPGA	<i>Field Programmable Gate Array</i>
HVDC	<i>High Voltage Direct Current</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IIR	<i>Infinite Impulse Response</i>
LCD	<i>Liquid Cristal Display</i>
MSA	Módulo Multiplica-Soma-Acumula
RAM	<i>Random Access Memory</i>
ROM	<i>Read-only Memory</i>
SIN	Sistema Interligado Nacional
SNR	<i>Signal-Noise Ratio</i>
SOC	<i>System on a Chip</i>
SQRT	Módulo de cálculo da Raiz Quadrada
TC	Transformador de Corrente

TRIP	Comando de Abertura de Disjuntor (Desligamento)
TRIP87	Módulo que implementa os TRIPs do relé de proteção diferencial desenvolvido
VHDL	<i>VHSIC Hardware Description Language</i>
VHSIC	<i>Very High Speed Integrated Circuits</i>
VPAF	Vetor de Posições Atuais dos Fasores

INTRODUÇÃO

1.1. CONTEXTUALIZAÇÃO DO TEMA

Em busca de crescimento econômico sustentável, a sociedade brasileira tem empreendido constantes investimentos na expansão e modernização das instalações elétricas do Sistema Interligado Nacional (SIN). No entanto, isto acarreta o aumento da sua complexidade operacional, exigindo a utilização de modernos sistemas de proteção, a fim de garantir que fenômenos indesejáveis à segurança operativa do sistema elétrico sejam extintos rápida e apropriadamente. Assim, preserva-se a integridade dos equipamentos e evita-se o desencadeamento de outros defeitos que podem culminar em blecautes de grandes proporções, causando transtornos a grande parte da população brasileira.

A importância desse equipamento em sistemas de potência, justifica a necessidade de um sistema de proteção adequado para este tipo de equipamento. O reparo ou substituição de transformadores, em caso de falhas, implicam em tempo fora de serviço e custos elevados.

Os transformadores de potência são equipamentos fundamentais em qualquer sistema elétrico de potência. Eles são os responsáveis pela operação do sistema com a tensão mais conveniente dos pontos de vista técnico e econômico [1].

A referência [2] apresenta estatísticas de falhas, para três períodos, considerando os tipos que podem ocorrer nos transformadores. Tomando-se apenas faltas em enrolamentos e em comutadores de tap, casos em que a proteção diferencial deve atuar, tem-se:

- De 1955 a 1965 = 70% do total de falhas;
- De 1975 a 1982 = 76% do total de falhas;
- De 1983 a 1988 = 59% do total de falhas.

Em termos de dados brasileiros, indica-se por meio da referência [3] que no período do ano de 2008 a 2012, 30% dos transformadores dos SIN desligaram, com um tempo médio para retorno à operação de quase 17 horas.

Estas taxas indicadas justificam o estudo e a busca de algoritmos cada vez mais confiáveis de proteção para transformadores.

Dentre as diversas proteções de natureza elétrica utilizadas em transformadores, a função diferencial pode ser considerada como primária, sendo seu fundamento baseado na comparação da diferença entre as correntes primárias e secundárias com uma corrente de operação previamente definida.

Com o advento da tecnologia digital, os princípios fundamentais da proteção diferencial puderam ser implementados nos modernos relés digitais microprocessados, que vêm se mostrando rápidos e confiáveis quando comparados aos tradicionais relés eletromecânicos.

Além disso, pode-se afirmar que a tecnologia digital tem se tornado mais acessível, em termos de custos e praticidade de uso, o que abre portas para o início dos desenvolvimentos de dispositivos para testes dos algoritmos estudados e desenvolvidos no âmbito do LAPSE e da Universidade de Brasília, sendo este o principal fator motivador deste trabalho: desenvolver plataformas que possibilitem a execução de testes em *hardware* dos algoritmos proteção de sistemas de potência estudados e desenvolvidos no LAPSE.

1.2. OBJETIVOS

O objetivo da presente dissertação de mestrado é descrever o processo de implementação em *hardware* de funções de proteção diferencial de transformadores. Como objetivos específicos têm-se:

- Desenvolver *hardware* microprocessado que execute algoritmos capazes de desempenhar as principais funções de proteção diferencial de transformadores de potência;
- Desenvolver *hardware* em plataforma FPGA capaz de desempenhar as principais funções de proteção diferencial de transformadores de potência; e
- Avaliar o desempenho das funções de proteção diferenciais implementadas diante de diferentes situações.

1.3. ORGANIZAÇÃO DO TEXTO

Essa dissertação será apresentada de acordo com a seguinte estrutura:

- Capítulo 2: realiza-se uma revisão bibliográfica referente às técnicas de proteção de transformadores, em especial a diferencial. Também são mencionados trabalhos relacionados ao desenvolvimento de *hardware* de proteção de sistemas elétricos;
- Capítulo 3: são abordados os conceitos da proteção diferencial de transformadores de potência e estimação de fasores;

- Capítulo 4: são descritos os *hardwares* desenvolvidos;
- Capítulo 5: apresentam-se os resultados apresentados pelo *hardware* desenvolvido em FPGA, após a injeção das amostras de corrente obtidas por meio de simulações no ATP;
- Capítulo 6: são apresentadas as conclusões e propostas para trabalhos futuros.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

O uso de transformadores de potência remonta à última década do século XIX, época em que a única forma de proteção disponível eram os fusíveis [4]. Disjuntores com operação satisfatória foram produzidos logo depois, de tal forma que os relés de proteção começaram a ser usados nos transformadores [4]. No começo do século XX, relés com esquemas de proteção diferencial começaram a ser usados em grandes transformadores e, em 1938, a introdução da restrição por harmônicos permitiu que a proteção não operasse de forma errônea frente a correntes de excitação durante a energização [4].

Desde então, em busca de uma maior sensibilidade, seletividade e velocidade de operação da proteção diferencial de transformadores, diversos métodos vem sendo sugeridos e utilizados para evitar situações que podem causar a má operação dos relés. Essas situações geralmente estão relacionadas à [5]: saturação de transformadores de corrente (TCs), compensação de defasagem, comutação de tap sob carga, corrente de magnetização, corrente de *inrush* e energização solidária.

Em 1983, Phadke e Thorp [6] desenvolveram uma técnica que utiliza a relação corrente-fluxo do transformador para obter a função de restrição para o relé diferencial. Eles mostram que esta técnica requer menor esforço computacional quando comparada à técnica que utiliza a restrição por harmônicos.

Em 1993, Wiszniewski e Kasztenny [7] utilizam da lógica Fuzzy para implementar os critérios que determinam a atuação ou não do relé diferencial. De acordo com os autores, a aplicação dessa lógica em relés de proteção faz com que a velocidade e a seletividade dos mesmos aumentem consideravelmente. Isso ocorre porque a tomada de decisão de um relé diferencial convencional é lenta, levando pelo menos um ciclo completo de corrente para acontecer.

De acordo Gajic, Ivankovic, Filipovic-Grcic e Rubesa [8], a corrente de *inrush* e a sobre-excitação dos transformadores podem produzir falsas correntes diferenciais que podem levar a má operação dos relés. Várias técnicas já foram e são utilizadas para evitar a operação errônea do relé diferencial nessas condições: atraso intencional da operação, dessensibilização da função durante o período de *inrush*, reconhecimento do conteúdo harmônico da corrente, dentre outros. Os relés mais modernos usam a restrição e o bloqueio por harmônicas, mas eles podem não funcionar em casos em que o conteúdo harmônico da corrente de operação é muito baixo. Outra alternativa que distingue a

corrente de *inrush* de faltas internas é a de reconhecimento da forma de onda, que pode não identificar a condição de sobre-excitação. Observando tudo isso, os autores propõem um novo algoritmo, que combina métodos de bloqueio e restrição por harmônicos com a técnica de reconhecimento da forma de onda. Este novo método usa harmônicos pares para restrição e o quinto harmônico e a componente CC para bloqueio da operação do relé.

Um exemplo de avaliação da proteção diferencial de transformadores pode ser verificado na referência [9]. Os autores partem de simulações computacionais e atividades laboratoriais para verificar a operação de dois relés comerciais com função diferencial quando existem transformadores operando em paralelo. Nessa situação, a chamada energização solidária (ou *Sympathetic Inrush*) produz uma corrente que flui para os demais equipamentos e gera um fluxo CC que se sobrepõe ao fluxo CA normal de magnetização. Isso proporciona altas correntes de magnetização nos transformadores vizinhos. Simularam-se aproximadamente 2700 situações de operação, dentre as quais energização com e sem defeito, falta interna entre espiras e falta interna entre fases. Foi verificado, neste trabalho, que um dos relés atuou de forma indevida para o caso de falta interna entre espiras no transformador vizinho, demonstrando o quanto é importante testar exaustivamente as mais diversas situações para que possa obter uma avaliação significativa da atuação dos relés. Ainda de acordo com os autores, o motivo da operação inadequada de um dos relés pode ser a forma como cada fabricante utiliza as componentes harmônicas para realizar a restrição.

Em termos de aplicação de *hardware* reconfigurável, os autores da referência [10] apresentaram método para implementação de relés de proteção de equipamentos de alta potência, fazendo uso das características do dispositivo FPGA (*Field Programmable Gate Array*) relacionadas ao paralelismo e altas taxas de processamento matemático em 16 bits.

Os autores da referência [11] notificaram o sucesso obtido em uma das etapas cruciais do processamento dos sinais de um relé de proteção numérico: a estimação fasorial. Neste trabalho, a estimação de fasores se realizou por meio de algoritmos genéticos projetados em uma máquina de estados implantada em FPGA.

Zhang Guiqing et al. [12] desenvolveram um típico SOC (*System on a Chip*) em FPGA que implementa diversas funções de proteção de sistemas de potência (sobrecorrente, sobretensão, subtensão, subfrequência, sobrecarga, reacendimento de disjuntor etc). Além disso, o trabalho implementou filtros digitais para filtragem desde a componente fundamental até a 16ª ordem harmônica.

Os autores do trabalho [13] desenvolveram FPGA um SOC capaz de detectar distúrbios por meio de filtragens rápidas baseadas em filtros IIR (*Infinite Impulse Response*). No dispositivo

desenvolvido, as faltas são detectadas por intermédio de verificações de mudanças no nível instantâneo dos sinais de corrente e tensão em relação a valores armazenados de ciclos anteriores. Destaca-se que os módulos aritméticos foram desenvolvidos para execução de cálculos com números reais de ponto fixo, representados em 16 bits.

O trabalho desenvolvido por Khaled Shehata et al. [14] consistiu no desenvolvimento em FPGA e em um microcontrolador ATMega16PC03511 de elementos de proteção de sobrecorrente, perda de fase, rotor bloqueado, desequilíbrio de fases etc. A comparação dos resultados obtidos indicou similaridade de performance, no que tange ao atendimento dos requisitos de cada elemento de proteção, e menor esforço computacional exercido pelo FPGA para o desempenho das funções.

O trabalho desenvolvido por Vishal Kumar et al. [15] aborda a implementação em FPGA de uma plataforma híbrida em que compartilham o mesmo encapsulamento o *hardware* reconfigurável e um *soft processor* NIOS [16]. Os elementos de proteção abordados foram restritos à proteção de sobrecorrente seguindo o padrão IEEE C37.112-1996.

Por conta do paralelismo e altas taxas de processamento que os FPGAs possibilitam, o trabalho desenvolvido em [17] apresentou a implementação de um *hardware* reconfigurável para atuar como proteção por ondas viajantes de linhas de bipólos HVDC.

FUNDAMENTAÇÃO DA PROTEÇÃO DIFERENCIAL DE TRANSFORMADORES DE POTÊNCIA

Os transformadores de potência fazem parte da grande cadeia de produção, comercialização e consumo de energia elétrica na sociedade contemporânea. Devido a sua relevância para este processo e a seu elevado custo de instalação (algo em torno de R\$ 15 milhões, no caso dos de 230/69 kV[18]), faz-se necessária a implantação de aparato de proteção preciso e seguro, de modo a garantir a integridade e otimização do uso da vida útil do equipamento.

Neste cenário, a proteção diferencial de transformadores de potência é designada como umas das mais importantes no monitoramento e detecção de faltas internas ao equipamento.

3.1. REQUISITOS DA PROTEÇÃO DIFERENCIAL

Em termos de especificação, no mercado brasileiro estabeleceu-se, por meio dos Procedimentos de Rede [20], que as funções diferenciais (87) dos sistemas de proteção de transformadores devem utilizar os enrolamentos dos transformadores de corrente - TC localizados próximos aos disjuntores do transformador ou autotransformador, de forma a incluir em sua zona de proteção as ligações entre os disjuntores e o transformador. Com isso, as zonas de proteção das funções diferenciais devem se superpor com as zonas de proteção dos barramentos adjacentes.

Além disso, os regulamentos vigentes estabelecem que as funções diferenciais de transformadores devem atender aos seguintes requisitos [20]:

- Função diferencial percentual com atuação individual por fase;
- Número de circuitos de restrição igual ao número de transformadores de corrente da malha diferencial; e
- Restrição da atuação para correntes de magnetização (*inrush* e sobreexcitação) e desempenhos transitórios desiguais de transformadores de corrente.

Ao atender aos critérios estabelecidos, garante-se que a proteção diferencial seja capaz de detectar faltas nos terminais e buchas do transformador, nos enrolamentos, no tanque e nos acessórios do equipamento, no comutador de TAPs sob carga e faltas relacionadas à ruptura do óleo isolante.

3.2. PROTEÇÃO DIFERENCIAL (87T)

O princípio básico de atuação da proteção diferencial numérica consiste em, a partir da leitura das correntes nos lados primário e secundário do transformador, aplicar-se a Lei de Kirchhoff dos nós. Para este tipo de proteção, o transformador é tratado como um nó, no qual o somatório das correntes que chegam e partem deve ser nulo, em condições normais de operação. A soma das correntes que chegam ao enrolamento primário e das correntes que partem do enrolamento secundário (e terciário, se houver) é denominada corrente de operação I_{op} . A proteção diferencial de transformadores deve identificar e sinalizar a ocorrência de faltas dentro da região protegida, situação essa em que a corrente de operação I_{op} será diferente de zero.

A zona protegida é delimitada por TCs que fazem as medições dos valores de corrente nos terminais do transformador e as disponibilizam ao relé de proteção diferencial, conforme esquema simplificado apresentado na Figura 3.1 e Figura 3.2, que retratam, respectivamente, situação normal de operação e situação de falta interna ao elemento protegido.

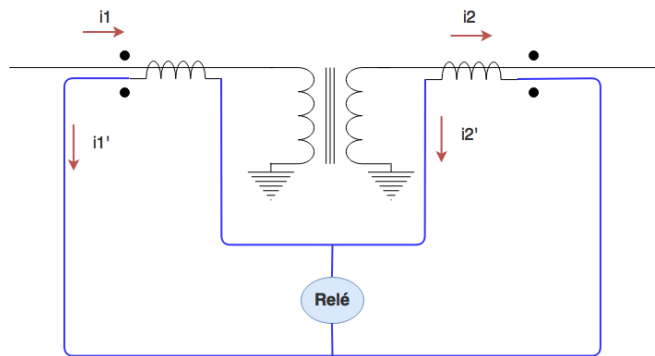


Figura 3.1 - Condição Normal de Operação.

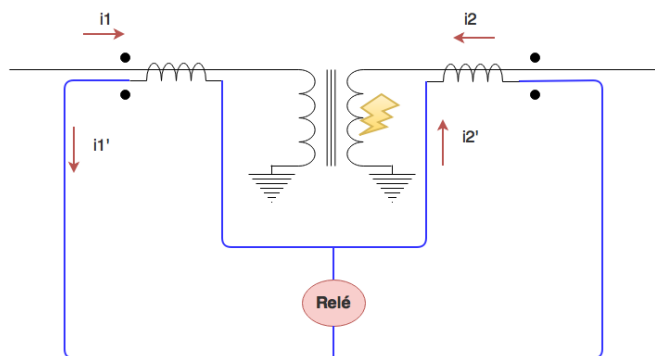


Figura 3.2 - Condição de Falta Interna ao Transformador.

A proteção diferencial de transformadores é sensivelmente dependente da qualidade da medição dos sinais de correntes que compõem a malha diferencial. Visto que existem diversos fatores que podem tornar imprecisas estas medições, opta-se normalmente por empregar a técnica de proteção diferencial

percentual, que se dá pela composição e comparação de correntes denominadas corrente de operação (I_{op}) e corrente de restrição (I_{res}).

A adoção de critérios baseados em uma corrente de restrição remonta ao princípio construtivo dos antigos relés eletromecânicos, em que bobinas de restrição eram dispostas de modo que, em condições normais, a força eletromecânica nelas gerada pelas correntes dos lados primário e secundário do transformador faziam forte oposição à força eletromecânica produzida pela bobina de operação, garantindo que o relé não fosse sensibilizado. Em situações de curto circuito, a força eletromecânica produzida na bobina de operação era de magnitude superior à produzida nas bobinas de restrição do relé, o que, de acordo com método construtivo, garantia a operação do relé e a indicação de TRIP [19].

As formulações indicadas nas Equações (3.1) e (3.2) são as que usualmente são adotadas em algoritmos de proteção diferencial de transformadores [21], empregados nos relés numéricos disponíveis no mercado. Ressalta-se que, no escopo do relé desenvolvido em FPGA, objeto dessa dissertação, adotou-se a segunda formulação da Equação (3.2) para o cálculo da corrente de restrição.

$$I_{op} = \left| \hat{I}_1 + \hat{I}_2 \right|, \quad (3.1)$$

$$I_{res} = \left\{ \begin{array}{l} k \cdot \left| \hat{I}_1 - \hat{I}_2 \right| \\ ou \\ k \cdot \left(\left| \hat{I}_1 \right| + \left| \hat{I}_2 \right| \right) \\ ou \\ \max \left(\left| \hat{I}_1 \right|, \left| \hat{I}_2 \right| \right) \end{array} \right\}, \quad (3.2)$$

em que \hat{I}_1 é o fasor da corrente medida pelo TC do lado primário do transformador e \hat{I}_2 , o fasor da corrente medida pelo TC do lado secundário.

A detecção de uma falta interna na região de proteção diferencial ocorre quando são atendidos conjuntamente os critérios apresentados nas Inequações (3.3) e (3.4), nas quais I_{pickup} é a corrente de sensibilização do relé diferencial, que leva em consideração em seu valor eventuais erros das medições dos TCs e das relações de transformação do transformador. A variável *SLOPE* define a inclinação da reta limiar que separa as regiões de operação e restrição do plano operacional, conforme exemplificado na **Figura 3.3**. O ajuste do *SLOPE* pode tornar a atuação da proteção diferencial percentual mais sensível ou restritivo, ao passo que o incremento de seu valor torna maior a área de restrição definida no plano.

$$I_{op} > I_{pickup} , \quad (3.3)$$

$$I_{op} > SLOPE \cdot I_{res} . \quad (3.4)$$

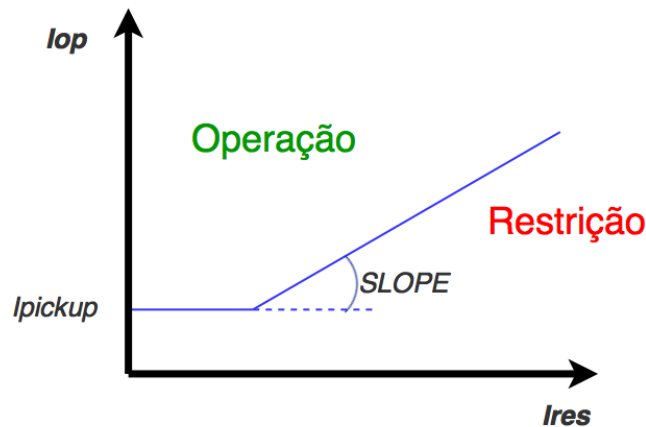


Figura 3.3 – Plano de Operação - Restrição do Elemento de Proteção Diferencial Percentual de Fase.

3.3. FATORES QUE INFLUENCIAM A PROTEÇÃO DIFERENCIAL DE TRANSFORMADORES

Tendo definido os critérios de detecção de faltas, faz-se necessário estabelecer métodos para adequação dos sinais de corrente que compõem a malha diferencial. Os tópicos que se seguem detalham os ajustes a serem efetuados nas correntes medidas para evitar que o relé de proteção diferencial de transformadores atue incorretamente devido a falsas correntes diferenciais.

Os ajustes e correções comumente adotados em proteção diferencial percentual numérica de transformadores resumem-se a:

- Correção das relações de transformação;
- Correção da defasagem angular;
- Compensação da sequência zero; e
- Restrição e/ou bloqueio por harmônicos.

3.3.1 Relação de Transformação

A compensação das relações de transformação tanto do transformador protegido quanto as dos transformadores de corrente são efetuadas por intermédio de grandezas de normalização (TAPs). O objetivo desta conversão consiste em padronizar todas as correntes para o sistema pu, cujas bases são as tensões e a potência nominais do transformador a ser protegido, por intermédio das Equações (3.5) e (3.6), [21] nas quais: $S_{nom,MVA}$ é a potência nominal, em MVA, do transformador; c_H e c_X são,

respectivamente e independentes entre si, constantes vinculadas ao esquema de ligação dos TCs primários e secundários (igual a 1, se o esquema for estrela, ou igual a $\sqrt{3}$, se o esquema for delta); $V_{H,nom,kV}$ e $V_{X,nom,kV}$ são respectivamente as tensões nominais, em kV, do lado primário e secundário do transformador; e RTC_H e RTC_X são as relações de transformação dos transformadores de corrente instalados nos lados primários e secundários do transformador.

$$TAP_H = \frac{1000 \cdot S_{nom,MVA} \cdot c_H}{\sqrt{3} \cdot V_{H,nom,kV} \cdot RTC_H}, \quad (3.5)$$

$$TAP_X = \frac{1000 \cdot S_{nom,MVA} \cdot c_X}{\sqrt{3} \cdot V_{X,nom,kV} \cdot RTC_X}. \quad (3.6)$$

3.3.2 Defasagem Angular

A depender da forma de conexão dos enrolamentos do transformador (estrela, delta e zig-zag), impõe-se uma defasagem angular entre as correntes primárias e secundárias medidas pelos TCs. Uma defasagem angular imposta às correntes do transformador pode implicar em má operação do relé de proteção diferencial, caso não haja uma prévia adequação dos valores de corrente. As diferentes combinações das conexões estão representas na Figura 3.4, as quais podem ser agrupadas em doze grupos de defasagem angular [22] [24].

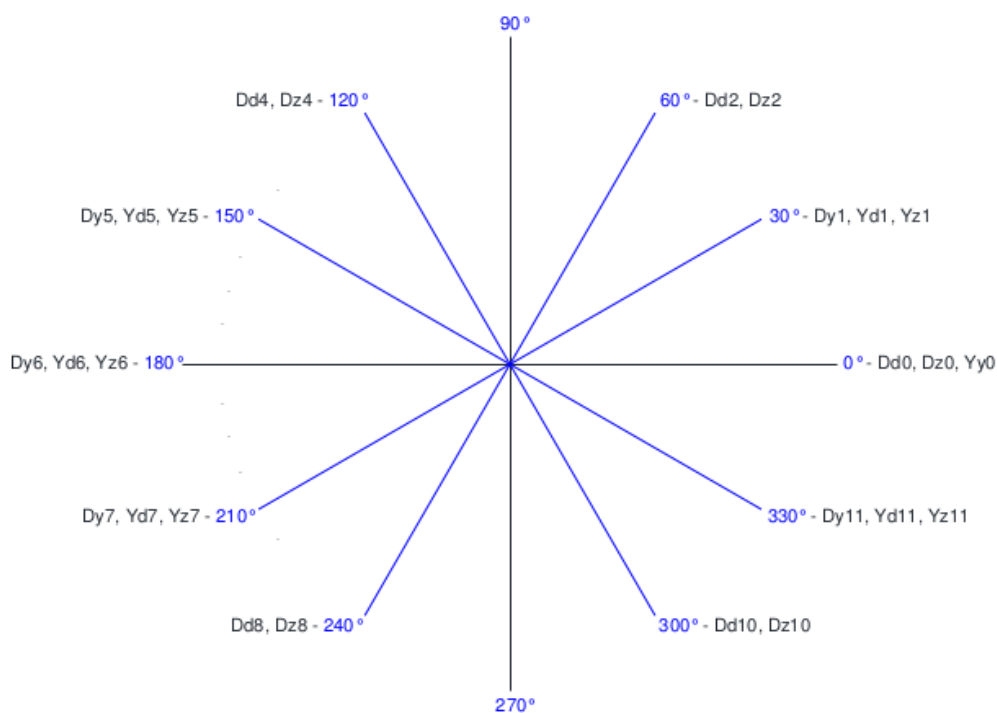


Figura 3.4 - Grupos de Defasagem Angular [24].

De acordo com o indicado em [23], a correção da defasagem angular aplicada em função do ângulo de defasamento entre as correntes do primário e as correntes do secundário são promovidas pela aplicação da Equação matricial (3.7), combinada com a Equação (3.8), para um dado ângulo de defasagem θ .

$$M_{\theta} = \begin{bmatrix} 1 + 2 \cos \theta & 1 + 2 \cos(\theta + 120^{\circ}) & 1 + 2 \cos(\theta - 120^{\circ}) \\ 1 + 2 \cos(\theta - 120^{\circ}) & 1 + 2 \cos \theta & 1 + 2 \cos(\theta + 120^{\circ}) \\ 1 + 2 \cos(\theta + 120^{\circ}) & 1 + 2 \cos(\theta - 120^{\circ}) & 1 + 2 \cos \theta \end{bmatrix}, \quad (3.7)$$

$$\begin{bmatrix} \hat{I}_{a_compensada_defasagem} \\ \hat{I}_{b_compensada_defasagem} \\ \hat{I}_{c_compensada_defasagem} \end{bmatrix} = M_{\theta} \cdot \begin{bmatrix} \hat{I}_a \\ \hat{I}_b \\ \hat{I}_c \end{bmatrix}. \quad (3.8)$$

3.3.3 Compensação da Corrente de Sequência Zero

Em transformadores com arranjo estrela-delta aterrados, durante curtos-circuitos que envolvem a terra, as correntes de sequência zero fluem apenas no lado estrela. Para o lado delta, apenas as componentes simétricas de sequência positiva e negativa são transferidas e de fato medidas pelos TCs. Essa diferença na composição das correntes medidas nos enrolamentos do transformador podem levar o relé de proteção diferencial a atuar indevidamente [25].

Neste sentido, a Equação (3.9) [25] promove compensação ao remover a corrente de sequência zero dos valores medidos de corrente no enrolamento do transformador cujo arranjo é estrela, permitindo que apenas as correntes compensadas possam ser comparadas adequadamente com as correntes do enrolamento delta do transformador.

$$\begin{bmatrix} \hat{I}_{a_compensada_seqZero} \\ \hat{I}_{b_compensada_seqZero} \\ \hat{I}_{c_compensada_seqZero} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} \hat{I}_a \\ \hat{I}_b \\ \hat{I}_c \end{bmatrix}. \quad (3.9)$$

3.3.4 Restrição e Bloqueio por Harmônicos

A presença de componentes harmônicas nos valores de corrente medidos pelos TCs pode interferir na precisão da detecção de faltas internas ao transformador ou até promover atuações indevidas da proteção diferencial percentual.

As correntes medidas em situações de saturação de TCs, sobreexcitação, energização de transformadores, entre outras, são contaminadas pela forte presença de componentes harmônicas, as quais podem chegar à ordem de 200% da corrente nominal do transformador [26].

A sobreexcitação ocorre principalmente em situações de sobretensões ou subfrequência, uma vez que o fluxo magnético no núcleo do transformador é diretamente proporcional à tensão aplicada e inversamente à frequência. A elevação do fluxo magnético no núcleo do transformador pode leva-lo à condição de saturação, o que implica em correntes dotadas principalmente de componentes harmônicas de 3ª e 5ª ordens [24].

Por outro lado, o processo de energização de transformadores merece grande atenção da proteção diferencial devido ao fenômeno que ocorre no ramo de magnetização do transformador: as correntes de *inrush*. As correntes de *inrush* são derivadas da combinação de fluxo magnético de regime transitório de energização com o fluxo remanescente no núcleo o transformador. O fluxo remanescente depende do valor fluxo no núcleo durante o desligamento do transformador. Conforme exemplificado na Figura 3.5, quando o transformador é energizado novamente, o fluxo remanescente soma-se com o fluxo de regime transitório de energização, o que pode levar fluxo total para a região de saturação da relação fluxo magnético-corrente de magnetização indicada na Figura 3.6. Desta forma, as correntes medidas pelos TCs durante o processo de energização do transformador (sem carga) são distorcidas e dotadas principalmente de componentes harmônicas de 2ª ordem. As correntes de *inrush* são tidas como um fenômeno estatístico, pois seu comportamento depende do instante de energização do transformador.

Os relés de proteção diferencial de transformadores empregam diversas técnicas para impedir que se desligue indevidamente o elemento protegido durante essas situações em que as correntes medidas pelos TCs são distorcidas. Usualmente estas técnicas consistem em filtrar as componentes de 2ª e 5ª harmônicas e comparar suas intensidades com as correntes fundamentais. O resultado destas comparações podem atuar bloqueando a atuação do relé de proteção diferencial ou restringindo sua atuação.

O bloqueio por harmônicos, para os casos discutidos, ocorre quando satisfeitas as condições indicadas pelas Inequações (3.10) e (3.11), nas quais \hat{I}_{2h} e \hat{I}_{5h} são, respectivamente, os fasores das correntes filtradas de 2ª e 5ª harmônicas e os fatores k_{2b} e k_{5b} são valores percentuais admissíveis das componentes harmônicas em relação à fundamental.

$$\frac{|\hat{I}_{2h}|}{I_{op}} > k_{2b}, \quad (3.10)$$

$$\frac{|\hat{I}_{5h}|}{I_{op}} > k_{5b}. \quad (3.11)$$

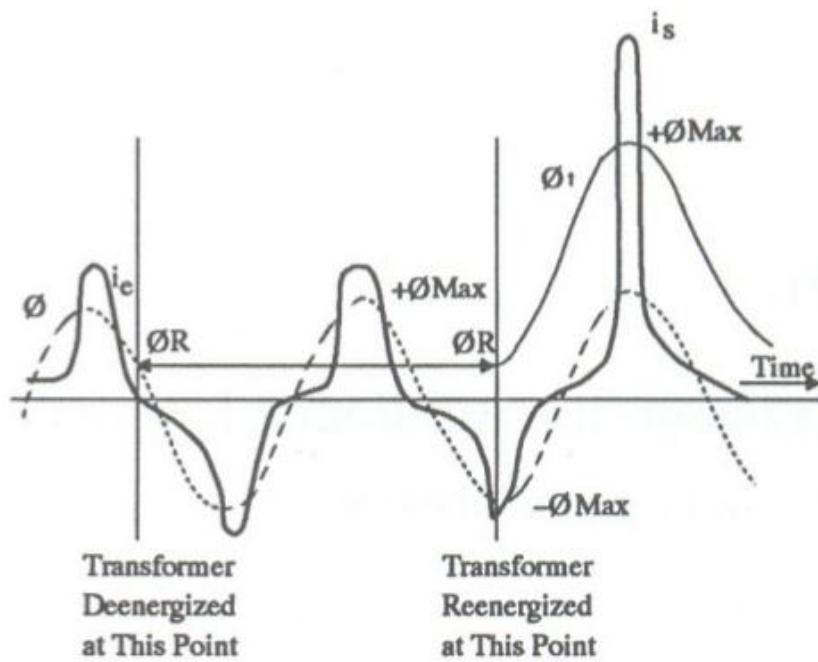


Figura 3.5 - Fenômeno das Correntes de *Inrush* [27]

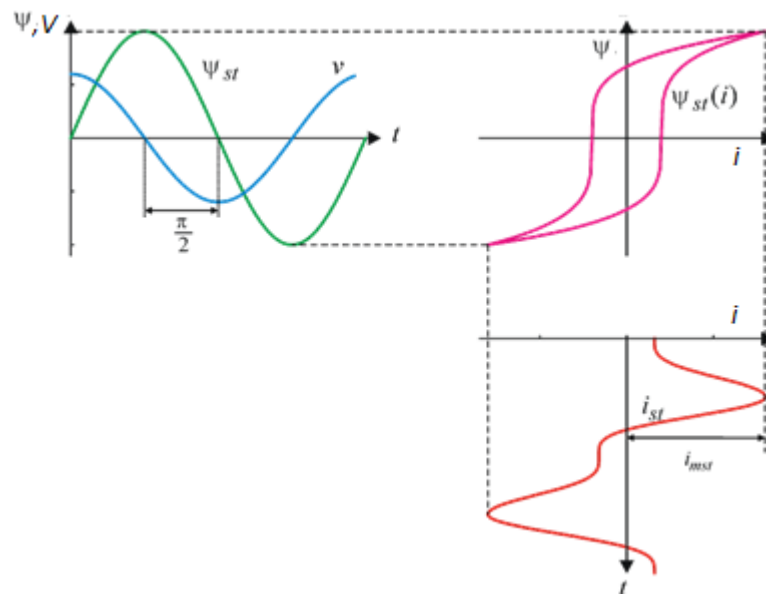


Figura 3.6 - Corrente de *Inrush* [27].

Usualmente, existem duas filosofias para a lógica de bloqueio por harmônicos: bloqueio independente e o bloqueio cruzado. Na lógica de bloqueio independente, a atuação da proteção diferencial percentual é inibida individualmente apenas nas fases em que os critérios das Inequações (3.10) e (3.11) forem atendidos, conforme indicado no diagrama lógico da **Figura 3.7**. A lógica de bloqueio cruzado, indicada no diagrama da **Figura 3.8**, tendo sido atendidas as condições indicadas nas Inequações (3.10) e (3.11) para pelo menos uma das fases, inibe os eventuais TRIPs provenientes de qualquer uma das fases.

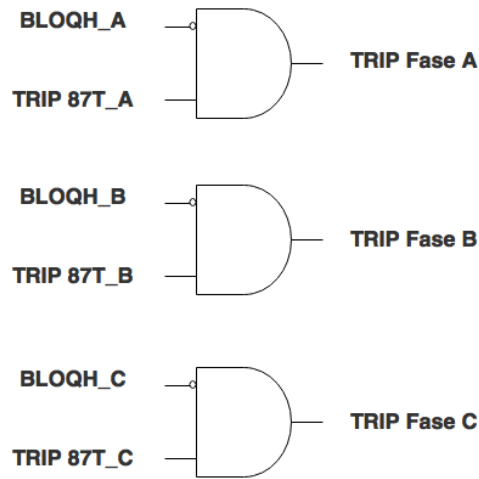


Figura 3.7 - Lógica de Bloqueio por Harmônicos Independente.

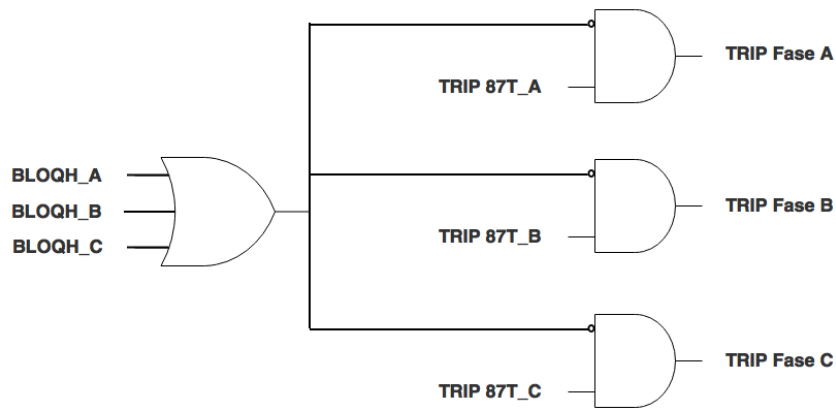


Figura 3.8 - Lógica de Bloqueio por Harmônicos Cruzado.

A técnica de restrição por harmônicos consiste em aumentar o valor do *SLOPE* quando detectado um alto percentual de componentes harmônicas na corrente medidas pelos TCs. Como resultado, a corrente de restrição passa a apresentar maior ponderação na Equação (3.4). Neste trabalho, adota-se a formulação para corrente de restrição com restrição por harmônico discutido e aprimorado em [24]:

$$I_{op} > SLOPE \cdot I_{res} + \frac{1}{k_{2r}} \cdot \left| \hat{I}_{2h} \right| + \frac{1}{k_{5r}} \cdot \left| \hat{I}_{5h} \right| , \quad (3.12)$$

em que os fatores k_{2r} e k_{5r} são constantes de proporcionalidade que relacionam a corrente fundamental às de 2ª e 5ª ordem, respectivamente.

A Equação (3.12) pode ser reformulada da seguinte forma [24]:

$$I_{op} > SLOPE \cdot I_{comp} , \quad (3.13)$$

em que:

$$I_{comp} = I_{res} + \frac{1}{SLOPE} \left(\frac{1}{k_{2r}} \cdot \left| \hat{I}_{2h} \right| + \frac{1}{k_{5r}} \cdot \left| \hat{I}_{5h} \right| \right) . \quad (3.14)$$

Neste caso, o plano operacional torna-se mais restritivo, conforme exemplificado na **Figura 3.9**.

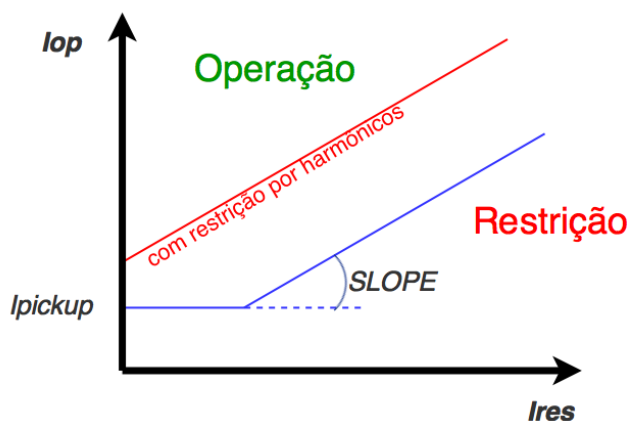


Figura 3.9 - Plano de Operação – Restrição, com Restrição por Harmônicos.

Além de um bom ajuste fatores k_{2b} , k_{5b} , k_{2r} e k_{5r} , e da inclinação *SLOPE* da reta que separa as regiões de operação e restrição no plano operacional, a definição da filosofia da lógica de bloqueio e/ou restrição por harmônicos impacta na qualidade da proteção diferencial aplicada ao transformador. Apresenta-se na **Tabela 3.1**[28] comparação entre os diferentes tratamentos aos harmônicos discutidos.

Tabela 3.1 - Comparação Entre o Bloqueio e a Restrição por Harmônicos.

	Bloqueio Independente	Bloqueio Cruzado	Restrição
Segurança para Falta Externas	Baixa	Moderada	Alta
Segurança para <i>Inrush</i>	Moderada	Alta	Alta
Segurança para Sobreexcitação	Baixa	Baixa	Alta
Confiabilidade para Falta Internas	Alta	Alta	Baixa
Confiabilidade para Falta Internas Durante Energização	Alta	Moderada	Alta
Velocidade para Falta Internas	Alta	Alta	Alta
Inclinação da curva	Bem definida	Bem definida	Depende dos harmônicos

Uma vez que para a maioria das situações relacionadas na Tabela 3.1 a filosofia do bloqueio cruzado tem se sobressai em relação à do bloqueio independente, considerando as gradações adotadas, a primeira filosofia foi adotada no âmbito do projeto do relé de proteção diferencial desenvolvido, combinada com a restrição por harmônicos na proteção diferencial percentual de fases.

3.4. PROTEÇÃO DIFERENCIAL DE SEQUÊNCIA NEGATIVA (87Q)

A proteção diferencial de sequência negativa identifica de forma rápida e precisa faltas que ocorrem entre espiras, inclusive em situações em que o transformador está com sua carga nominal e os efeitos deste tipo de curto circuito nas correntes de linha são mínimos e não sensibilizam os elementos de proteção diferencial percentual de fase.

As formulações indicadas nas Equações (3.15) e (3.16), muito similares às adotadas na proteção diferencial percentual de fases, são adotadas em algoritmos de proteção diferencial de sequência negativa.

$$I_{opQ} = \left| \hat{I}_{1Q} + \hat{I}_{2Q} \right| , \quad (3.15)$$

$$I_{resQ} = \max \left(\left| \hat{I}_{1Q} \right|, \left| \hat{I}_{2Q} \right| \right) , \quad (3.16)$$

em que \hat{I}_{1Q} é a corrente de sequência negativa calculada a partir das correntes medidas no enrolamento primário do transformador e \hat{I}_{2Q} , a corrente de sequência negativa calculada a partir das correntes medidas no enrolamento secundário.

A detecção de faltas pelo elemento de proteção diferencial de sequência negativa ocorre quando são atendidos conjuntamente os critérios indicados nas Inequações (3.17) e (3.18), nas quais $I_{pickupQ}$ é a corrente de sensibilização do elemento de proteção diferencial de sequência negativa. A variável $SLOPEQ$ define a inclinação da reta limiar que separa as regiões de operação e restrição do plano operacional, conforme exemplificado na Figura 3.10. O ajuste do $SLOPEQ$ pode tornar a atuação da proteção diferencial de sequência negativa mais sensível ou restritivo, ao passo que o incremento de seu valor torna maior a área de restrição definida no plano.

$$I_{opQ} > I_{pickupQ} , \quad (3.17)$$

$$I_{opQ} > SLOPEQ \cdot I_{resQ} . \quad (3.18)$$

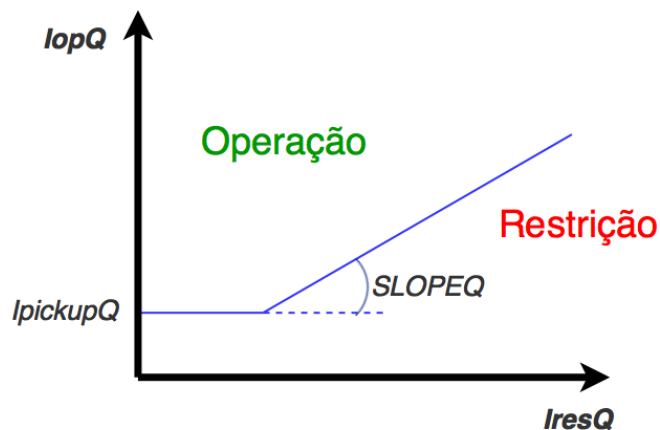


Figura 3.10 – Plano de Operação – Restrição do Elemento de Proteção Diferencial de Sequência Negativa.

Vale ressaltar que a proteção diferencial de sequência negativa é bastante sensível a desbalanços, desequilíbrios de corrente entre fases. Isso significa que, uma vez que as correntes de *inrush* são desequilibradas entre as fases, o elemento de proteção diferencial de sequência negativa certamente é sensibilizado. Logo, o bloqueio por harmônicos deve também inibir os TRIPs advindos desse elemento de proteção, de acordo com a lógica proposta na Figura 3.11.

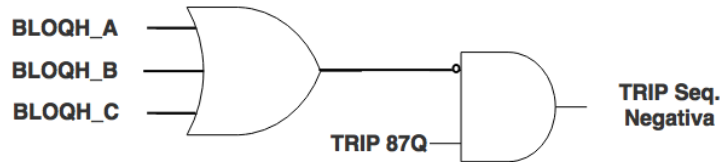


Figura 3.11 - Lógica de Bloqueio por Harmônicos da Proteção Diferencial de Sequência Negativa (87Q).

3.5. PROTEÇÃO DE FALTA À TERRA RESTRITA (87REF)

Devido à baixa sensibilidade dos elementos de proteção diferencial percentual de fases frente a curtos circuitos espira-terra que envolvam um pequeno percentual do enrolamento, muitos fabricantes de relés desenvolveram algoritmos para detecção de faltas à terra restrita para suprir tal deficiência e possibilitar que a totalidade do enrolamento seja protegida.

O elemento de proteção de falta à terra restrita atua verificando o diferencial entre a corrente de sequência zero e a corrente de neutro medida por TC específico conectado no neutro do enrolamento estrela do transformador. Desta forma, esse elemento de proteção atua com bastante segurança em situações de faltas que envolvam a terra, no lado do transformador ligado em estrela.

No âmbito do relé desenvolvido, foram empregadas as Equações (3.19) e (3.20) [29] para o cálculo das correntes de operação e restrição, nas quais \hat{I}_N é o fasor da corrente medida pelo TC de neutro, \hat{I}_0 é o fasor de corrente de sequência zero, calculado com base nos fasores das correntes de fase, e K_r é um fator de estabilização, cujos valores variam normalmente entre 2 e 4 [24].

$$I_{OPREF} = \left| \hat{I}_N \right| , \quad (3.19)$$

$$I_{RESREF} = K_r \cdot \left(\left| \hat{I}_N - 3\hat{I}_0 \right| - \left| \hat{I}_N + 3\hat{I}_0 \right| \right) . \quad (3.20)$$

A detecção de faltas pelo elemento de proteção de faltas à terra restrita ocorre quando são atendidos conjuntamente os critérios indicados nas Inequações (3.21) e (3.22), nas quais $I_{PickupREF}$ é a corrente de sensibilização do elemento de proteção diferencial de faltas à terra restrita. O ajuste do $SLOPEREF$ pode tornar a atuação da proteção de falta à terra restrita mais sensível ou restritivo.

$$I_{opREF} > I_{pickupREF}, \quad (3.21)$$

$$I_{opREF} > SLOPEREF \cdot I_{resREF}. \quad (3.22)$$

3.6. ESTIMAÇÃO DE FASORES

Como complemento à fundamentação teórica da proteção diferencial implementada no decorrer deste trabalho, faz-se necessário passar pelo estudo de filtros digitais empregados na estimação de fasores.

Boa parte dos relés numéricos da atualidade fazem uso de algoritmos de estimação de fasores da componente fundamental da tensão e/ou da corrente medida por transformadores de corrente - TCs e transformadores de potencial - TPs. Os fasores são estimados com base nas amostras digitais dos sinais [30]. Por manipular sinais digitais e tentarem extrair a frequência fundamental do sinal, esses algoritmos podem ser tratados como filtros digitais que recebem em sua entrada um sinal composto por uma frequência fundamental, harmônicas e a componente CC de decaimento exponencial.

A avaliação do desempenho do filtro, neste caso, consiste em averiguar sua capacidade de apresentar saídas precisas para a frequência para a qual foi ajustado e verificar qual a influência da componente CC de decaimento exponencial em seu comportamento.

Tradicionalmente, os filtros utilizados para estimação de fasores fazem uso de janela móvel que armazena uma quantidade fixa de amostras anteriores do sinal.

No âmbito deste trabalho, a estimação fasorial baseou-se no Filtro Cosseno Modificado desenvolvido pela ABB [33], cuja formulação é dada por:

$$Y_{Real}[k] = \frac{2}{N} \sum_{k=0}^{N-1} y[k] \cos\left(\frac{2\pi k}{N}\right), \quad (3.23)$$

$$Y_{Imag}[k] = \frac{Y_{an}[k-1] - Y_{an}[k] \cos\left(\frac{2\pi}{N}\right)}{\text{sen}\left(\frac{2\pi}{N}\right)}, \quad (3.24)$$

em que k é a k -ésima amostra do sinal $y(t)$ e $Y_{Real}[k]$ e $Y_{Imag}[k]$ são, respectivamente, as partes reais e imaginárias, do fasor estimado.

HARDWARE DE PROTEÇÃO DIFERENCIAL

Neste capítulo apresenta-se a arquitetura dos dispositivos voltados para a proteção diferencial de transformadores de potência desenvolvidos neste trabalho. Os desenvolvimentos concentraram-se exclusivamente no processamento numérico dos sinais de corrente já condicionados e digitalizados. Os circuitos de conversão analógico-digital, redução dos níveis das correntes provenientes dos secundários dos TCs e filtragem não fazem parte do escopo dos estudos realizados nesse mestrado.

O tópico 4.1 abordará o software desenvolvido para a plataforma Raspberry Pi, bem como suas possibilidades de integração a um sistema de condicionamento e amostragem de sinais de corrente provenientes de TCs conectados aos lados primário e secundário e terminal de neutro de um transformador de potência.

O tópico seguinte, 4.2, detalhará o SOC implementado em uma plataforma FPGA (*Field Programmable Gate Array*), o qual foi desenvolvido em dispositivo do fabricante Altera, modelo EP4CE10F17C8N.

4.1. DESENVOLVIMENTO NA PLATAFORMA RASPBERRY PI

Na primeira etapa do projeto do *Hardware* do Relé de Proteção Diferencial de Transformadores, implementou-se em linguagem C++ algoritmo para as funções de proteção diferencial discutidas no Capítulo 3, o qual foi compilado para execução na plataforma Raspberry Pi [34]. Os resultados obtidos nesta implementação foram apresentados por meio de artigo [42] apresentado no Simpósio Brasileiro de Sistemas Elétricos 2016 – SBSE.

Além da possibilidade de se testar a eficácia dos algoritmos de proteção diferencial, esta etapa do projeto é fundamental para que se possa ter um referencial para o desenvolvimento do *hardware* do relé de proteção diferencial de transformadores em FPGA.

A **Figura 4.1** apresenta modelo conceitual no qual pode ser inserida a plataforma Raspberry Pi executando os algoritmos desenvolvidos para as funções de proteção diferencial de transformadores de potência. De um modo geral, os algoritmos foram desenvolvidos para serem inseridos num ambiente que consiste na Plataforma Raspberry Pi, modelo B, associada a dois microcontroladores ATMEGA

328p, um display LCD de 16 colunas por 2 linhas e um par de relés de acionamento das bobinas de abertura dos disjuntores associados ao transformador protegido.

No escopo desse trabalho, os desenvolvimentos ficaram restritos ao desenvolvimento dos algoritmos de proteção diferencial de transformadores de potência. A etapa de condicionamento e digitalização dos sinais de correntes medidas pelos TCs poderá ser abordada em trabalhos futuros a fim de se obter uma versão completa de um relé de proteção diferencial.

O processador ARM1176JZF-S de 700 MHz, disponível na placa de prototipagem Raspberry Pi dispõe de 512 MB de memória RAM, portas USB, Ethernet e pinos de entrada e saída que operam em nível lógico de 3,3 volts (GPIOs), dentre outras interfaces. Ao operar na frequência de 700 MHz, este dispositivo oferece um desempenho equivalente a 41 MFLOPS (106 operações de ponto flutuante por segundo). O sistema operacional escolhido para o dispositivo é o *Raspbian* [35], versão variante da distribuição Linux Debian Wheezy, otimizada para operar com o conjunto de instruções ARMv6.

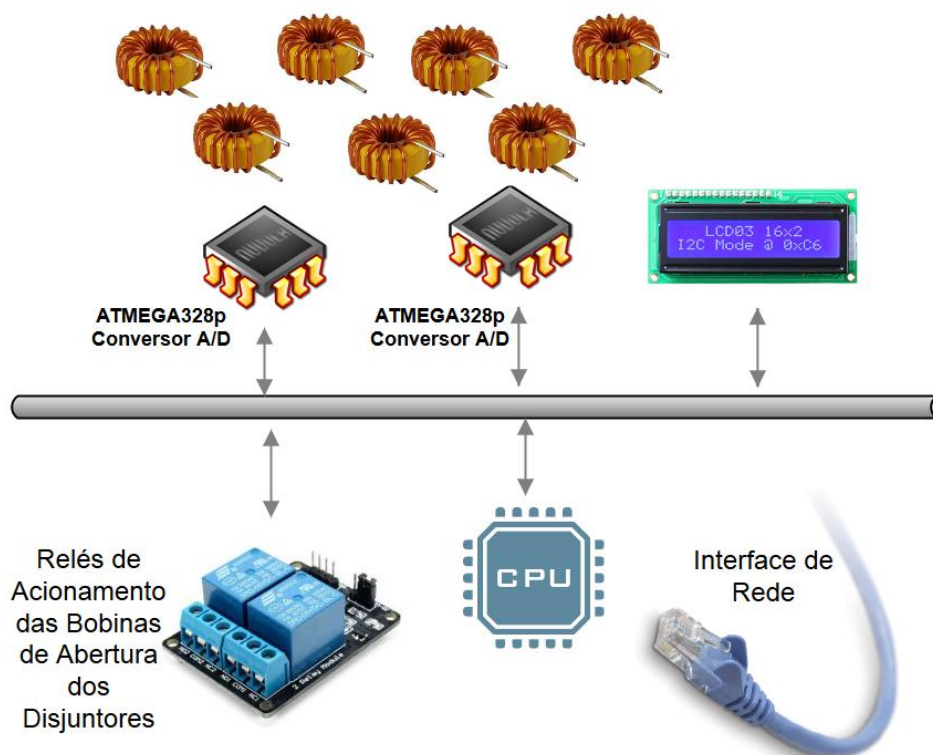


Figura 4.1- Modelo Conceitual no qual o Relé Desenvolvido se Insere.

Como método de entrada dos sinais de corrente, em alternativa à conversão A/D, são fornecidos arquivos gerados em simuladores diversos como, por exemplo, o ATP, via interface de rede do hardware.

O display LCD tem como função principal sinalizar as proteções atuadas, quando da detecção de faltas no transformador e, em regime permanente de operação, apresenta os valores das correntes nas fases e os valores das correntes de operação e restrição calculadas durante a execução do algoritmo.

Os relés de acionamento das bobinas de abertura, quando da detecção de falta interna à zona diferencial do transformador, recebem sinais em nível lógico de 5 Volts e fecham o circuito de TRIP dos disjuntores.

4.1.1 Modelagem em Software das Funções de Proteção Diferencial

A modelagem e o desenvolvimento do software de proteção diferencial de transformadores para o Raspberry Pi serviu como referencial para o posterior desenvolvimento do *hardware* em FPGA, detalhado no tópico 4.2 dessa dissertação. A **Figura 4.2** apresenta diagrama de blocos do núcleo do software desenvolvido em C++ que desempenha as funções de proteção diferencial de transformadores.

Para evitar atrasos provocados pelo escalador de processos do sistema operacional, o algoritmo foi desenvolvido e integrado ao ambiente Linux sob a forma de “*kernel module*”[36] para operar com chamadas de interrupção de alta prioridade do sistema operacional.

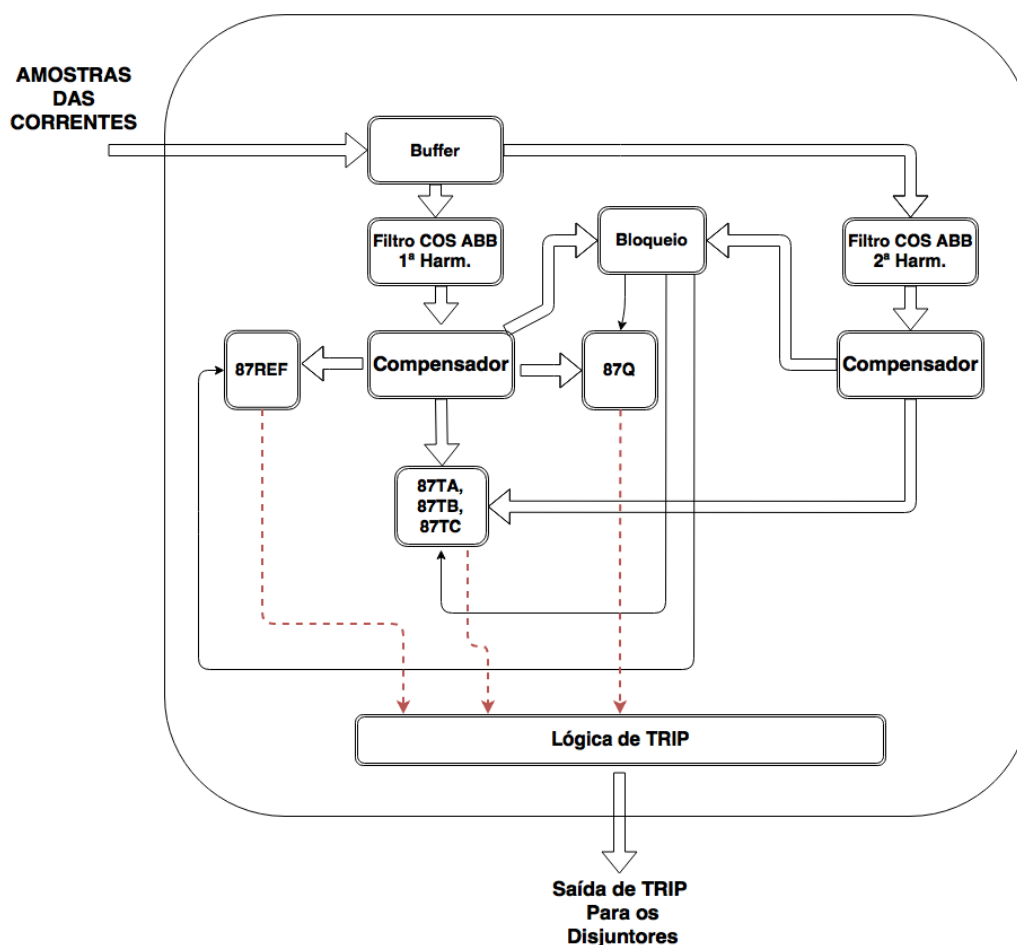


Figura 4.2 - Diagrama descritivo do sistema modelado e implementado.

Na sequência são detalhados os módulos implementados sob orientação a objetos, o que possibilitou o reaproveitamento de código durante a criação de instâncias de cada elemento de proteção diferencial.

4.1.1.1. Compensador

Os módulos compensadores foram projetados para empreender a normalização das amostras de corrente ou dos fasores de corrente para pu da corrente nominal do transformador. Além disso, os compensadores efetuam correções de defasagem angular entre os enrolamentos primário e secundário do transformador, bem como podem proceder à eliminação da corrente de sequência zero.

4.1.1.2. Buffer

O objetivo deste módulo é armazenar uma janela móvel de 16 (dezesesseis) amostras, no qual a amostra mais antiga é descartada. As amostras são disponibilizadas aos estimadores de fasores.

4.1.1.3. Filtro Cosseno ABB 1ª Harmônica e 2ª Harmônica

Estes módulos implementam o algoritmo do filtro cosseno modificado da ABB com base na janela de amostras obtida no módulo Buffer, a fim de realizar a estimação dos fasores na frequência fundamental e na frequência harmônica de segunda ordem.

Optou-se não abordar os filtros de quinta ordem, uma vez que se decidiu restringir as análises aos processos de energização de transformadores, os quais são dotados principalmente de harmônicos de 2ª ordem, deixando para trabalhos posteriores uma discussão e análise aprofundada de casos de sobreexcitação de transformadores.

4.1.1.4. 87TA, 87TB e 87TC

Estes módulos são responsáveis por desempenhar a função de proteção diferencial percentual por fases, com restrição por harmônicos.

4.1.1.5. 87Q e 87REF

Estes módulos são responsáveis por implementar as proteções diferenciais de sequência negativa, e de falta à terra restrita.

4.1.1.6. Bloqueio por Harmônicos

Este módulo detecta a presença de componentes harmônicas de 2ª ordem acima de um limiar estabelecido e envia sinais de bloqueio aos demais elementos de proteção diferencial que compõem o sistema.

4.2. DESENVOLVIMENTO DO RELÉ DE PROTEÇÃO DIFERENCIAL DE TRANSFORMADOR EM FPGA

Os FPGAs possuem como grande diferencial a possibilidade de reconfiguração de seu arranjo interno de elementos lógicos de modo a ser possível a execução de diferentes tarefas a cada nova configuração. Sua versatilidade facilita o desenvolvimento de protótipos, bem como a correção de eventuais falhas de projeto, após a entrega do produto ao mercado.

Adotou-se para este projeto o FPGA Cyclone IV, modelo EP4CE10F17C8N, do fabricante Altera. A **Tabela 4.1** resume as principais características do FPGA adotado.

Tabela 4.1 - Dados Principais do FPGA Altera Cyclone IV, modelo EP4CE10F17C8N.

Elementos Lógicos (EL)	10.320
Bits de memória embarcada	414.000
Multiplicadores 18 x 18 embarcados	23
Pinos de entrada e saída (IO)	179

Retrata-se na Figura 4.3 a estrutura de um Elemento Lógico, que é a menor unidade de lógica dos FPGAs da família Cyclone IV, contendo basicamente [38]:

- Uma *Look-Up Table* de quatro entradas, que permite implementar qualquer função de quatro variáveis;
- Um registrador, que pode ser configurado como Flip-flop do tipo D, T, JK ou SR; e
- Interconexões locais, de linha e coluna, que servem como meios de comunicação com outros ELs.

Durante o processo de desenvolvimento do relé na plataforma FPGA, fez-se uso do ambiente de desenvolvimento Quartus II, disponível em [37]. Este ambiente, cuja interface é ilustrada na Figura 4.4, permite que, a partir de uma linguagem de descrição de *hardware*, o projetista possa compilar projetos de dispositivos lógicos para que sejam capazes de serem programados no FPGA e apresentar respostas perante aos sinais apresentados em pinos pré-determinados.

Adotou-se neste projeto a linguagem de descrição de *hardware* VHDL - *VHSIC Hardware Description Language* (*VHSIC - Very High Speed Integrated Circuits*) para o desenvolvimento de todos os módulos componentes do *Hardware* do Relé de Proteção Diferencial de Transformadores.

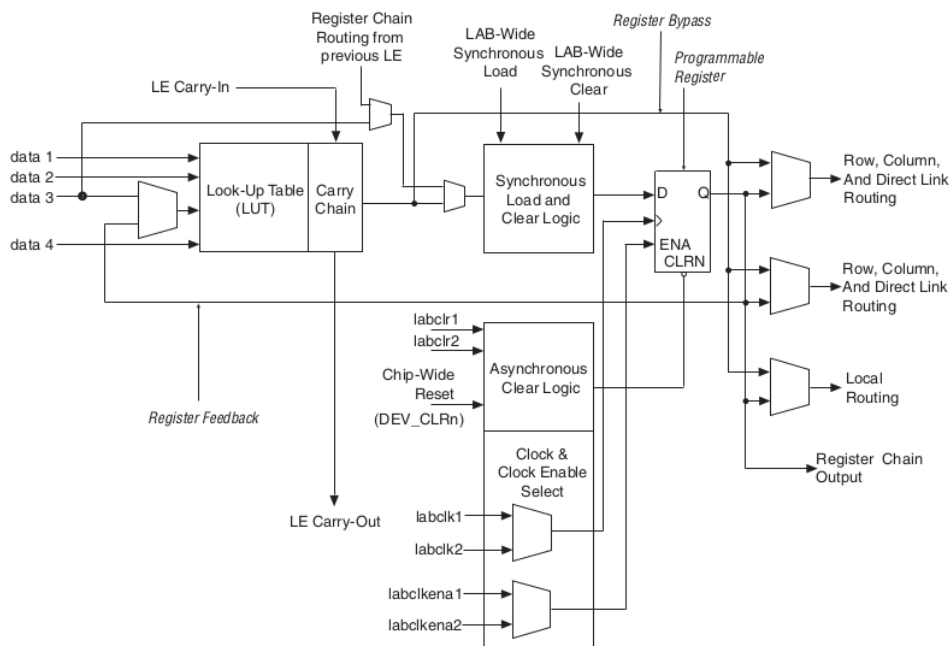


Figura 4.3 - Elemento Lógico dos FPGAs da Família Cyclone IV [38].

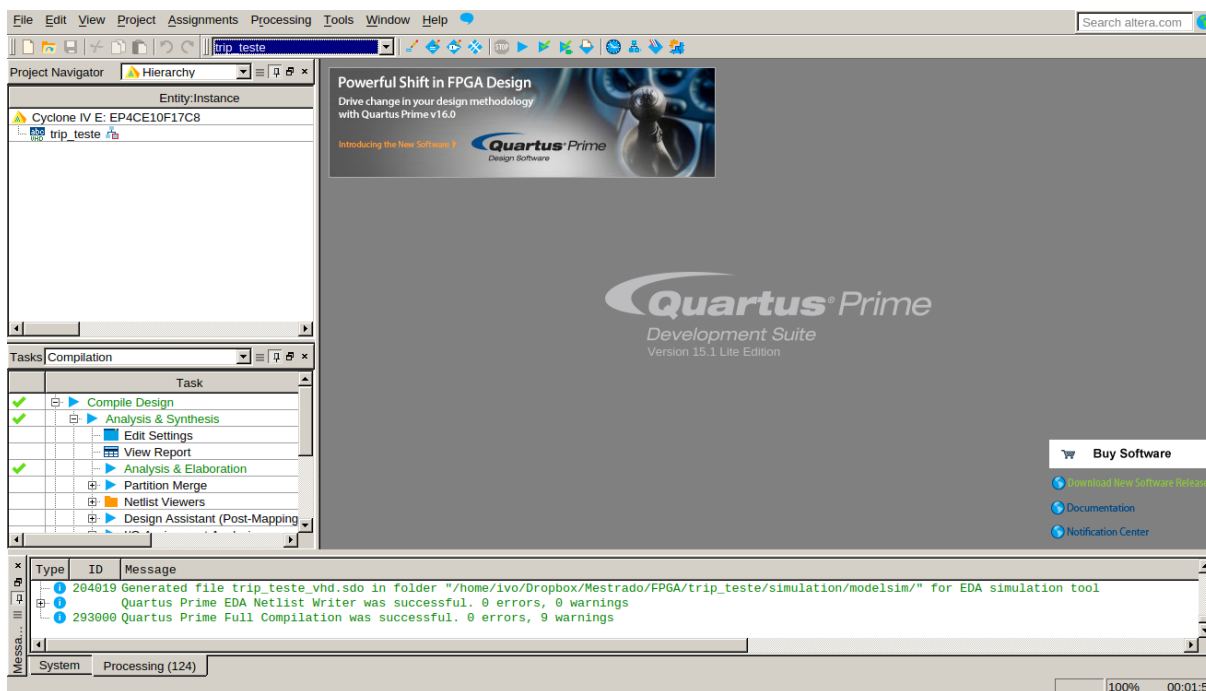


Figura 4.4 – Quartus II - Ambiente de Desenvolvimento para FPGAs do Fabricante Altera.

4.2.1 Estrutura geral do sistema

O sistema que implementa as funções do relé diferencial em FPGA foi projetado com base nas seguintes especificações:

- Distinção e tomada de decisões diante de situações de regime normal de operação, energização do transformador, curtos-circuitos durante a energização do transformador, faltas externas e internas, curtos-circuitos entre espiras, curtos-circuitos espira-terra;

- b) Estimação de fasores com base no filtro cosseno modificado da ABB;
- c) Execução da função de proteção diferencial percentual de fase (87T) com restrição de harmônicos;
- d) Execução da função de proteção diferencial de sequência negativa (87Q);
- e) Execução da função de proteção diferencial contra faltas restritas à terra (87REF);
- f) Bloqueio e restrição de atuação das funções de proteção diferenciais durante a energização do transformador;
- g) Execução das funções de proteção diferencial baseadas em fasores com base na taxa de amostragem de 16 amostras por ciclo;
- h) Sete canais de correntes amostradas, sendo três para as correntes primárias do transformador, outros três para as correntes secundárias do transformador e um canal para a corrente de neutro;
- i) Buffer circular de 16 posições para cada canal de corrente amostrada;
- j) Portas de entrada para os valores das amostras de corrente no padrão IEEE 754 para representação de número em ponto flutuante;
- k) Portas de saída de TRIP de cada uma das funções de proteção diferencial;
- l) Armazenamento interno dos valores das correntes de operação e de restrição calculadas em cada elemento de proteção diferencial;
- m) Portas de saída para consulta e avaliação dos valores calculados das correntes de operação e restrição em cada elemento de proteção diferencial. Os valores apresentados nesta porta de saída representam números em ponto flutuante no padrão IEEE 754;
- n) Execução em paralelo do maior número de funções de proteção diferencial;
- o) Execução sincronizada no *clock* do FPGA, de 50 MHz; e
- p) Tomada de decisões devem ocorrer em tempo inferior à periodicidade de chegada de cada nova amostra de corrente.

Em consonância com as especificações propostas, o sistema que implementa as funções de proteção diferencial pode assumir formatos que variam de acordo com a disponibilidade de recursos no FPGA. A Figura 4.5 apresenta proposta geral do sistema, na qual o elementos de proteção diferencial são dispostos de modo a operar em paralelo. Ressalta-se, contudo, que o formato final do sistema objeto

deste trabalho será apresentado no item 4.2.17 desta dissertação, no qual serão apresentadas as características do sistema que foi possível sintetizar no FPGA Cyclone IV, modelo EP4CE10F17C8N.

Uma vez definido o escopo do projeto, passamos ao detalhamento de módulos que são comuns e essenciais tanto aos elementos intermediários (que efetuam o condicionamento das amostras de corrente) quanto aos elementos de proteção diferencial (87T, 87Q e 87REF). Estes elementos comuns compreendem módulos matemáticos, que efetuam operações com números em ponto flutuante no padrão IEEE 754, comparadores e de acesso, leitura e escrita, na memória RAM.

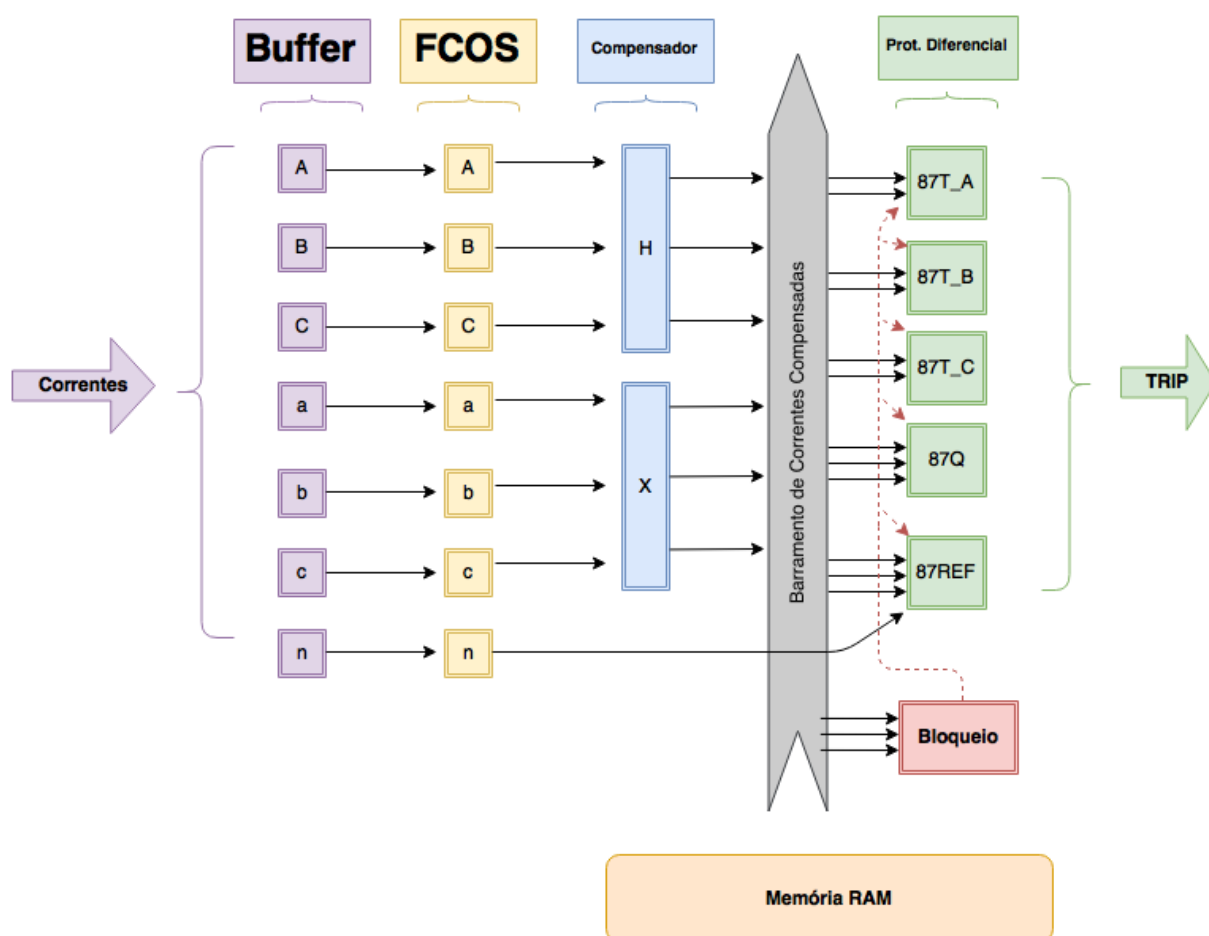


Figura 4.5- Especificação Simplificada e Inicial do Sistema.

4.2.2 Módulos Matemáticos

Para definição do escopo de quais módulos matemáticos serão necessários ao sistema, faz-se necessário analisar o algoritmo desenvolvido e implementado na plataforma *Raspberry Pi*. Com exceção dos módulos que executam a função de *buffer*, as atividades desempenhadas nos módulos do sistema efetuam operações matemáticas em ponto flutuante de adição, subtração, multiplicação e raiz quadrada. Uma rápida análise acerca dessas operações é realizada nos tópicos seguintes.

4.2.2.1. Representação Numérica de Ponto Flutuante no Padrão IEEE 754

O padrão adotado no projeto do *hardware* do Relé de Proteção Diferencial de Transformadores para representação numérica é o IEEE 754, que consiste no padrão adotado na grande maioria de dispositivos computacionais desde o ano de 1985 para representação similar à notação científica de números reais.

Comparada à possibilidade de adotar a formatação de ponto fixo para representação numérica de números reais, a adoção do padrão IEEE 754 implicou em maior complexidade do *hardware* desenvolvido, uma vez que as operações aritméticas dependem de maior número de iterações, adequações e arredondamentos dos operandos. Todavia, esta escolha nos permite representar um espectro mais largo de números e assegura um menor erro de arredondamento.

O padrão de representação em 32 bits adotado neste projeto permite representar números reais compreendidos entre $\pm 3,4 \times 10^{38}$ nas extremidades do universo representado, sendo que os valores mais próximos do zero são $\pm 1,2 \times 10^{-38}$ [43].

Há uma desigualdade no espaçamento entre os valores representados neste universo, sendo que o que o intervalo entre dois números é dez milhões de vezes menor que esses números [39]. Esse fato é importante para garantir um pequeno erro de arredondamento, uma vez que o espaçamento é largo entre números grandes e estreito entre números menores. E a relevância deste fato para o projeto do *hardware* do Relé de Proteção Diferencial de Transformadores reside na alta relação sinal-ruído (SNR) obtida durante os processos de arredondamento dos valores representados.

A **Figura 4.6** indica o modo de representação em 32 bits de números reais no padrão IEEE 754, o qual é composto por 1 bit, para representação do sinal, 8 bits em complemento de 2, para indicação do expoente, e 23 bits para representação da mantissa, na qual representa-se a parte fracionária da notação decimal.

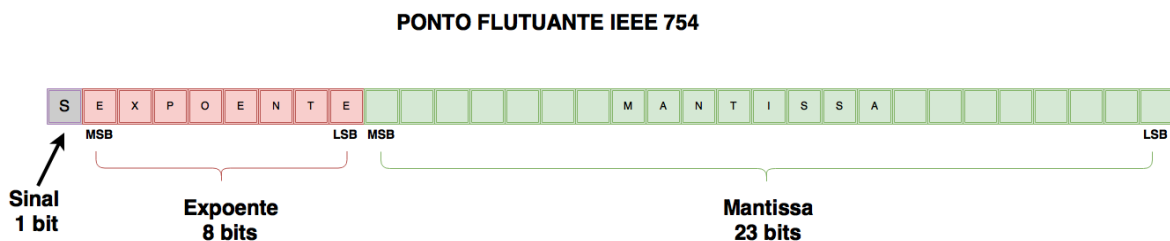


Figura 4.6 - Representação de Valores em Ponto Flutuante, padrão IEEE 754.

A relação indicada na Equação (4.1) permite converter qualquer sequência de bits no valor real por ela representado. Nesta relação, tem-se que S é valor do bit de sinal, M é a sequência fracionária de bits e E é a sequência de bits que representam o expoente.

$$v = (-1)^S \times M \times 2^{E-127} \quad (4.1)$$

4.2.2.2. Adição e Subtração de Números de Ponto Flutuante

Com base no formato estabelecido, o processo de soma de operandos em ponto flutuante consiste em 4 (quatro) etapas. Estas etapas consistem em:

- a) Verificar se os expoentes dos operandos são os mesmos;
- b) Caso sejam diferentes os expoentes, deve ser escolhido o operando com menor expoente para execução de sucessivos deslocamentos na mantissa, de modo a permitir que os operandos estejam com o mesmo expoente;
- c) Efetua-se a soma binária das mantissas dos operandos; e
- d) Por meio de sucessivos deslocamentos, efetua-se a normalização do resultado da operação. Normalmente, nesta etapa, ocorre um truncamento dos bits menos significativos resultantes na mantissa.

A subtração é realizada de forma similar, sendo necessário apenas inverter o bit de sinal de um dos operandos.

No projeto desenvolvido do *hardware* do Relé de Proteção Diferencial de Transformadores, adotou-se o módulo de soma e subtração cuja propriedade intelectual é da Altera®, o qual dispõe de otimizações tanto de desempenho quanto de área ocupada no FPGA. A **Tabela 4.2** apresenta o resumo das características deste módulo somador, após sua sintetização no FPGA Cyclone IV, modelo EP4CE10F17C8N.

Tabela 4.2 - Características do Módulo Somador, de propriedade intelectual da Altera®.

Operandos de 32 bits	2
Memória (bits)	36
Ciclos de <i>clock</i> por operação	7
Total de Elementos Lógicos Utilizados	744
Percentual de uso do FPGA adotado	7,21 %

4.2.2.3. Multiplicação de Números de Ponto Flutuante

O processo de multiplicação de operandos em ponto flutuante consiste em 3 etapas. Estas etapas consistem em:

- a) Somam-se os expoentes dos operandos;
- b) Multiplicam-se as mantissas dos operandos, normalmente por meio de métodos convencionais; e
- c) Verifica-se a ocorrência de *overflow*.

No projeto desenvolvido do *hardware* do Relé de Proteção Diferencial de Transformadores, adotou-se o módulo de multiplicação de propriedade intelectual da Altera®, o qual dispõe de otimizações tanto de desempenho quanto de área ocupada no FPGA. A Tabela 4.3 apresenta o resumo das características deste módulo multiplicador, após sua sintetização no FPGA Cyclone IV, modelo EP4CE10F17C8N.

Tabela 4.3 - Características do Módulo Multiplicador, de propriedade intelectual da Altera®.

Operandos de 32 bits	2
Multiplicadores embarcados 18 x 18	3
Multiplicadores embarcados 9 x 9	1
Ciclos de <i>clock</i> por operação	5
Total de Elementos Lógicos Utilizados	262
Percentual de uso do FPGA adotado	2,54 %

4.2.2.4. Raiz Quadrada de Números de Ponto Flutuante (Módulo SQRT)

O cálculo da raiz quadrada de um número real em ponto flutuante pode ser efetuado por meio de diversos métodos, tais como o algoritmo babilônico [40], a equação de Pell, o algoritmo de Goldschmidt, método de Newton-Raphson e CORDIC. Todos esses métodos têm como princípio básico o cálculo iterativo que se repete até uma provável convergência do resultado.

Tendo em vista que a Altera® disponibiliza em seu ambiente de desenvolvimento um módulo proprietário de cálculo de raiz quadrada de números ponto flutuante de 32 bits, resolveu-se no projeto do *hardware* do Relé de Proteção Diferencial de Transformadores, assim como nos módulos de soma e multiplicação, adotar esse módulo de propriedade intelectual do fabricante do FPGA, o qual denomina-

se módulo SQRT. A Tabela 4.4 apresenta o resumo das características deste módulo multiplicador, após sua sintetização no FPGA Cyclone IV, modelo EP4CE10F17C8N.

Tabela 4.4 - Características do Módulo Raiz Quadrada (SQRT), de propriedade intelectual da Altera ®.

Operandos de 32 bits	1
Memória (bits)	121
Ciclos de <i>clock</i> por operação	16
Total de Elementos Lógicos Utilizados	823
Percentual de uso do FPGA adotado	7,97 %

4.2.2.5. Módulo Multiplica-Soma-Acumula (MSA)

Uma vez verificadas as áreas necessárias para sintetização dos módulos aritméticos no FPGA Cyclone IV, modelo EP4CE10F17C8N, podemos proceder à análise da utilização destes módulos frente aos algoritmos que implementam a estimação fasorial, compensadores e funções específicas da proteção diferencial de transformadores.

Algoritmo 4.1- Método C++ da Estimação Fasorial por Meio do Filtro Cosseno Modificado a ABB

```

1. Complexo FCOS::run(float * janela, unsigned int harmonica) {
2.
3.     unsigned int N = (tamanho_janela-1);
4.     unsigned int m = harmonica;
5.     float d = (2 * pi)/N;
6.
7.     float *janela_velha = &janela[0];
8.     float *janela_nova = &janela[1];
9.
10.    float X_real_old = 0;
11.    float X_real_new = 0;
12.
13.    int i;
14.    for (i=1; i<= N; i++) {
15.
16.        X_real_old = X_real_old + (sqrt(2)/N) * janela_velha[i-1] * cos(m*(i)*d);
17.        X_real_new = X_real_new + (sqrt(2)/N) * janela_nova[i-1] * cos(m*(i)*d);
18.
19.    }
20.
21.    Complexo X(0,0);
22.    X.real = X_real_new;
23.    X.imaginario = (X_real_old - X_real_new * cos(m*d))/sin(m*d);
24.
25.    return X;
26.
27. }
```

Em uma primeira verificação, analisamos o **Algoritmo 4.1** que implementa o filtro cosseno modificado da ABB, executando os cálculos indicados pelas Equações (3.23) e (3.24). Observa-se que, para sintetização de um *hardware* que seja capaz de executar o referido algoritmo, no que tange ao

cálculo da apenas da parte real do fasor a ser estimado e tratando as 16 amostras ao mesmo tempo, são necessários 16 multiplicadores e 15 somadores, conforme indicado na **Figura 4.7**.

Percebe-se do **Algoritmo 4.1** que a estimação fasorial, baseada no filtro cosseno modificado da ABB, implementa um filtro de resposta finita ao impulso (FIR) [31], o qual consiste em uma sucessão de multiplicações e somas.

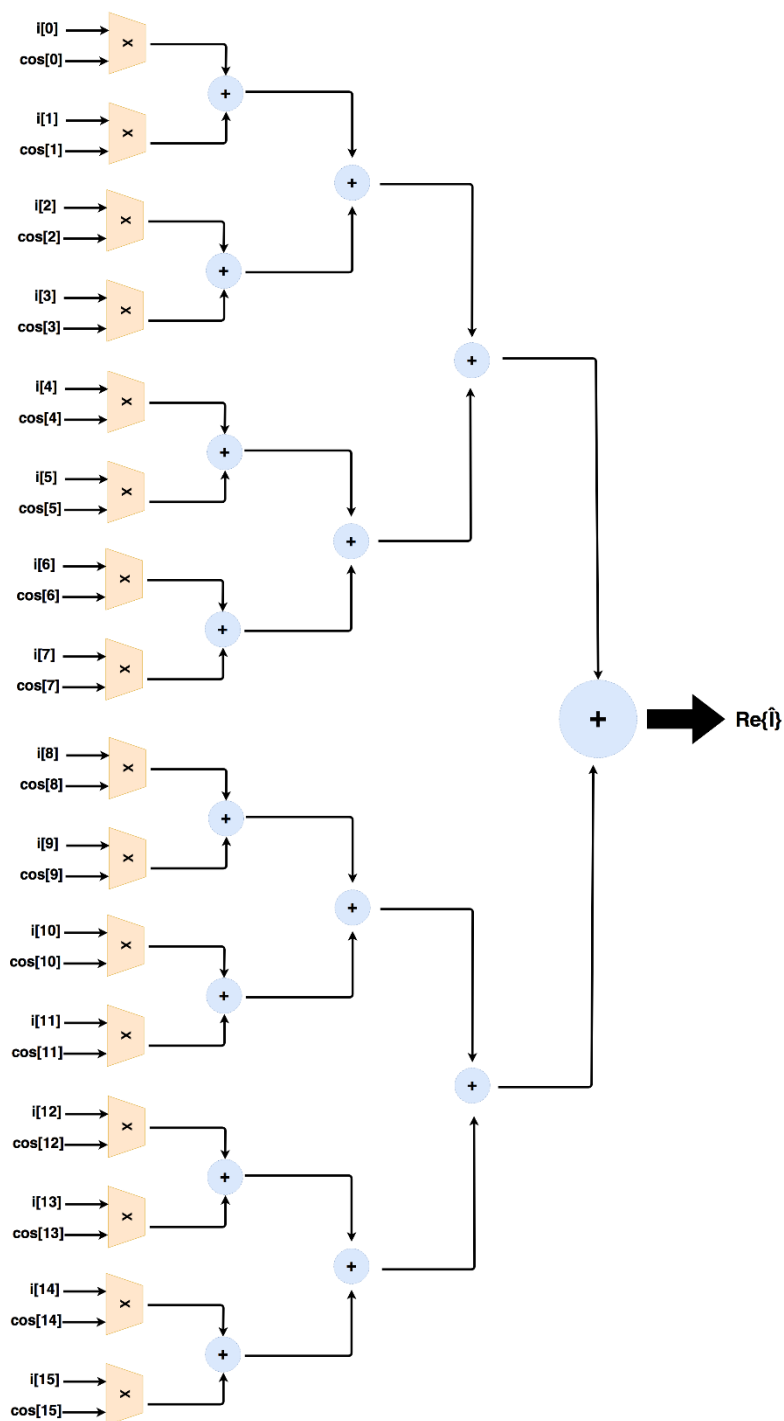


Figura 4.7 - Módulos Aritméticos para Implementação da Estimação de Fasores.

Caso fosse adotado o projeto indicado na Figura 4.7, e levando em conta os resultados da sintetização dos módulos aritméticos de soma e multiplicação indicados na Tabela 4.2 e na Tabela 4.3, o uso dos recursos do FPGA adotado neste projeto destinados à estimação de fasores de apenas uma fase consumiria 148,76% dos recursos disponíveis no modelo do FPGA adotado, o que torna inviável sintetizar esse projeto no Cyclone IV, modelo EP4CE10F17C8N. A **Tabela 4.5** apresenta resumo do uso dos recursos do FPGA, caso fosse adotada a proposta apresentada na Figura 4.7 para a estimação de fasores.

Tabela 4.5 - Resultados de Possível Módulo de Estimação de Fasores, sintetizado conforme projeto indicado na Figura 4.7.

Operandos de 32 bits	32
Memória (bits)	540
Multiplicadores embarcados 18 x 18	48
Multiplicadores embarcados 9 x 9	16
Ciclos de <i>clock</i> por operação	33
Total de Elementos Lógicos Utilizados	15.352
Percentual de uso do FPGA adotado	148,76 %

Desta forma, verificou-se a necessidade de desenvolver o projeto do *hardware* do Relé digital de Proteção Diferencial de Transformadores com uma topologia voltada ao compartilhamento de recursos do FPGA entre os módulos desenvolvidos, combinada com a performance mínima estabelecida. O que significa dizer que o *hardware* deverá tomar as decisões com base nas amostras de corrente em intervalo de tempo inferior ao intervalo entre amostras de corrente. Neste sentido, deve ser levado em consideração que: a) a frequência do *clock* do *hardware* é de 50 MHz; b) que cada ciclo de *clock* tem uma duração de 20 ns; e c) que o tempo de resposta do *hardware* deve ser inferior a 1,041 ms (intervalo entre amostras). Com isso, determina-se que relé desenvolvido no FPGA deve levar no máximo 52.083 ciclos de *clock* para executar todo o processamento das funções de proteção diferencial.

Uma vez definidas as condições de contorno, definiu-se no projeto em questão o escopo de um módulo capaz de executar o conjunto de operações multiplicação, soma e armazenar o resultado para operações futuras. Denominou-se este módulo com a sigla MSA (Multiplica Soma Acumula).

O funcionamento básico do módulo MSA consiste em efetuar a multiplicação dos dois operandos de 32 bits (ponto flutuante) em 5 ciclos de *clock*. O resultado da multiplicação é somado com o valor disponível em um registrador interno da MSA, em 7 ciclos de *clock*. O resultado dessa operação

é salvo no registrador interno da MSA, em 1 ciclo de *clock*. A MSA faz uso dos módulos de soma e multiplicação de propriedade intelectual da Altera®, já apresentados em itens anteriores dessa dissertação.

A **Figura 4.8** apresenta modelo simplificado do módulo MSA, indicando suas portas de entrada e saída. Observa-se que, além das portas destinadas aos operandos e ao resultado, a MSA dispõe de uma porta de reset, cuja função é reiniciar o módulo e zerar seu registrador interno, uma porta de *clock* que, em cada borda de subida, dispara os registradores para um novo passo de execução e uma porta de habilitação do *clock* (*clk_enable*), que, em nível lógico baixo, inabilita o módulo MSA para qualquer operação. A conclusão dos cálculos efetuados pela MSA é sinalizada durante um ciclo de *clock* por meio da porta de saída *result_ok*.

O diagrama de tempo do módulo MSA é apresentado na **Figura 4.9**. Neste diagrama, é possível observar a MSA operando multiplicação e soma de três mostras em um filtro FIR genérico.

A **Tabela 4.6** apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo aritmético MSA. Observa-se que esse módulo faz uso módico dos recursos do FPGA, o que deixa uma margem de 89,72% da área disponível para uso dos outros módulos desenvolvidos no *hardware* do relé. Destaca-se, também, que os valores indicados na referida tabela foram extraídos dos resultados apresentados pelo *software* Quartus II [37], utilizado para compilação, sintetização e programação do FPGA adotado.

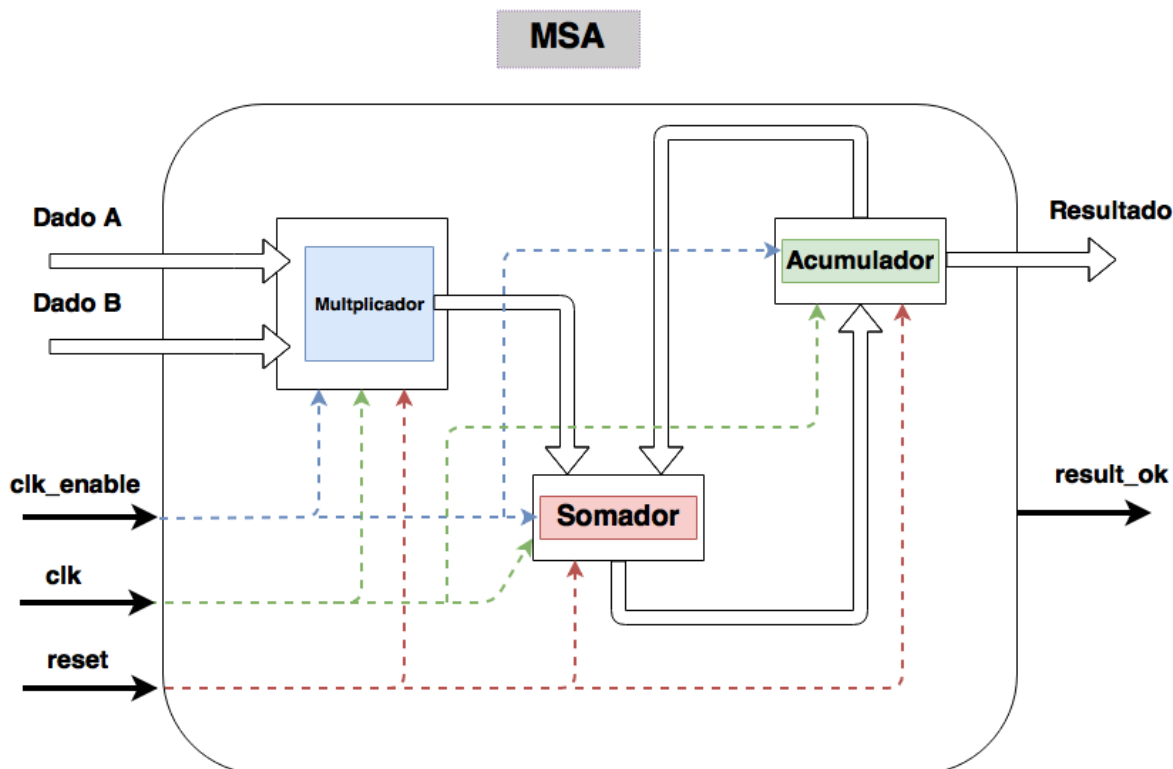


Figura 4.8 - Módulo MSA sintetizado no *hardware* do Relé de Proteção Diferencial de Transformadores.

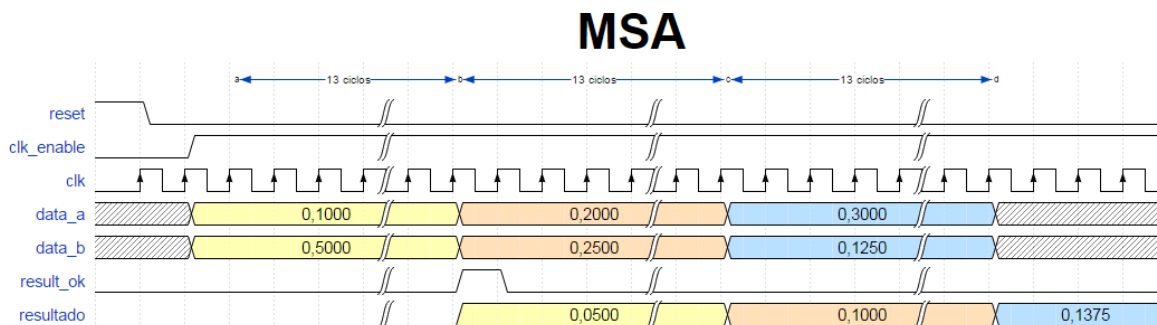


Figura 4.9- Diagrama de Tempo do Módulo Aritmético MSA.

Tabela 4.6 - Resultados da MSA sintetizada no FPGA adotado no projeto.

Operandos de 32 bits	2
Memória (bits)	36
Multiplicadores embarcados 18 x 18	3
Multiplicadores embarcados 9 x 9	1
Ciclos de <i>clock</i> por operação	13
Total de Elementos Lógicos Utilizados	1.061
Percentual de uso do FPGA adotado	10,28%

4.2.2.6. Módulo Comparador

O processo de comparação de valores expressos no formato IEEE 754 se torna um tanto complexo, pois os erros de arredondamento, inerentes à operação aritméticas de ponto flutuante, podem fazer com que números idênticos difiram em sua representação binária.

Tendo em vista que a Altera® disponibiliza em seu ambiente de desenvolvimento um módulo proprietário de comparação entre dois números ponto flutuante de 32 bits, resolveu-se no projeto do *hardware* do Relé de Proteção Diferencial de Transformadores, assim como nos módulos de soma, multiplicação e raiz quadrada, adotar esse módulo de propriedade intelectual do fabricante do FPGA. A **Tabela 4.7** apresenta o resumo das características deste módulo multiplicador, após sua sintetização no FPGA Cyclone IV, modelo EP4CE10F17C8N.

Uma vez que o módulo Comparador faz uso de um percentual muito baixo dos recursos do FPGA Cyclone IV, modelo EP4CE10F17C8N, adotou-se como opção de projeto não fazer o uso compartilhado deste módulo por meio de barramentos. Neste sentido, todos os módulos que necessitam de executar operações de comparação dispõem de um módulo comparador dedicado e incorporado, a exemplo dos módulos F87T, F87Q, F87REF e Bloqueio.

Tabela 4.7 - Características do Módulo Comparador, de propriedade intelectual da Altera ®.

Operandos de 32 bits	2
Ciclos de <i>clock</i> por operação	1
Total de Elementos Lógicos Utilizados	89
Percentual de uso do FPGA adotado	0,86%

4.2.3 Módulos de Memórias RAM

O FPGA Cyclone IV, modelo EP4CE10F17C8N, dispõe de 423.936 bits de memória embarcada, os quais podem ser agrupados em memórias do tipo ROM ou RAM. No contexto deste projeto, adotou-se o uso apenas de memórias do tipo RAM, separadas em 3 blocos, descritos na **Tabela 4.8**. Todos os blocos de memória possuem a mesma interface de acesso, leitura e escrita, dos dados. A **Figura 4.10** apresenta modelo simplificado de um módulo de memória RAM, indicando suas portas de entrada e saída.

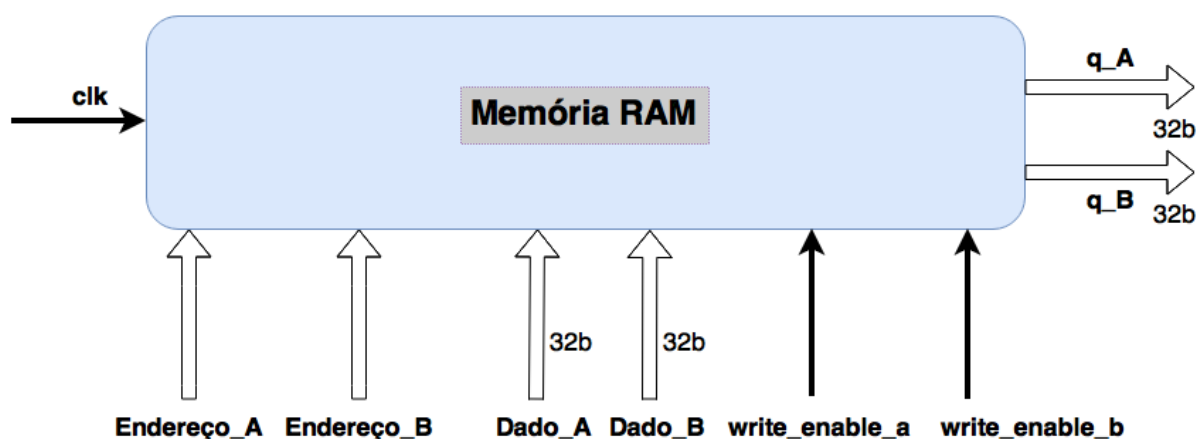


Figura 4.10 - Módulo de Memória RAM

A interface dos módulos de memória RAM permite acessar simultaneamente dois endereços internos da memória. É possível, dessa forma, realizar ao mesmo tempo duas operações de leitura, ou duas operações de escrita, ou uma operação de escrita combinada com uma de leitura. Outra característica importante reside no fato de a latência dos módulos de memória RAM ser de apenas um ciclo de *clock*, o que não implica em grandes atrasos aos módulos acessantes.

A **Figura 4.11** apresenta diagrama de tempo de um módulo memória no qual são executadas operações de escrita e leitura de determinados endereços. Observa-se que os sinais denominados *write_enable_a* e *write_enable_b*, quando em nível lógico 1, habilitam escrita dos dados nos endereços

especificados, enquanto que em nível lógico 0, habilita a leitura por meio das portas q_a e q_b dos valores armazenados nos endereços especificados.

Tabela 4.8 - Módulos de Memória RAM Instanciados.

Memória RAM	Espaço (em palavras de 32 bits)	Linhas de Endereçamento	Descrição
RAM_0	256	8	Destinada ao armazenamento das amostras de corrente, fasores estimados, coeficientes do filtro cosseno de 1º Harmônico (ordem fundamental) e coeficientes dos Compensadores.
RAM_1	256	8	Destinada ao armazenamento das amostras de corrente, fasores estimados, coeficientes do filtro cosseno de 2º Harmônico e coeficientes dos Compensadores.
RAM_2	12.415	14	Destinada ao armazenamento dos valores das correntes de operação e restrição, calculadas nos módulos que implementam o diferencial de fases (87T), diferencial de sequência negativa (87Q) e diferencial de faltas à terra restrita (87REF). São armazenados também os fasores das correntes primárias, secundárias e de neutro. Também são armazenados os valores binários de TRIP de cada um dos módulos de proteção F87T, F87Q, F87REF e de bloqueio emitidos pelo módulo de bloqueio por harmônicos.

Outro aspecto que vale ressaltar é o fato de as memórias RAM instanciadas não fazerem uso dos elementos lógicos disponíveis no FPGA.

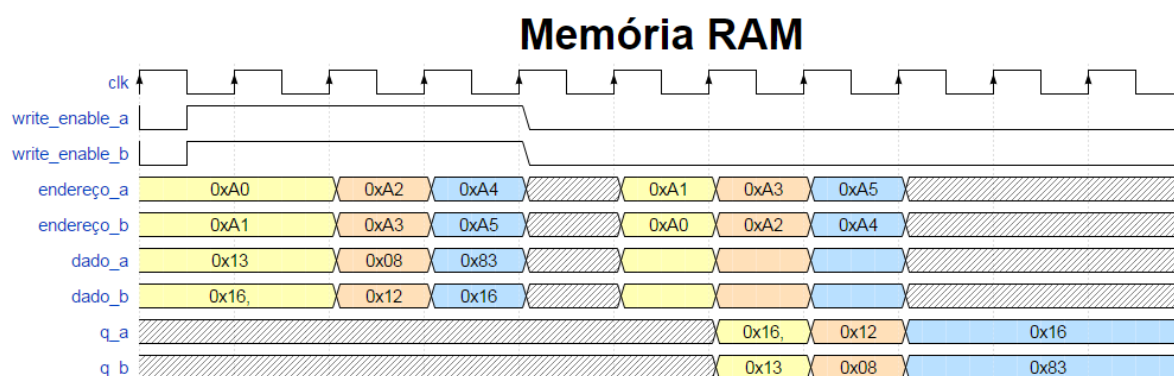


Figura 4.11 - Diagrama de Tempo dos Módulos de Memória RAM.

4.2.4 Barramento de Acesso às Memórias RAM

Uma vez que os módulos de memória são elementos de uso compartilhado entre os elementos de proteção diferencial desenvolvidos no *hardware*, vislumbra-se a necessidade de um módulo que implemente a função de barramento para controlar o fluxo de dados entre os módulos e as memórias RAM.

No contexto do projeto do *Hardware* do Relé de Proteção Diferencial de Transformadores, foram sintetizados dois barramentos de acesso às memórias RAM: a) o barramento de acesso às memórias RAM_0 e RAM_1, denominado BARRAMENTO_RAM_0_1; e b) o barramento de acesso à memória RAM_2, denominado BARRAMENTO_RAM_2.

Apresenta-se na **Figura 4.12** o modelo simplificado do BARRAMENTO_RAM_0_1, indicando suas portas de entrada e saída. O BARRAMENTO_RAM_2, é apresentado na **Figura 4.13**. Observa-se que o que difere os projetos dos dois barramentos é o conjunto de módulos acessantes e a linha seletora de memória de destino, presente apenas no barramento BARRAMENTO_RAM_0_1.

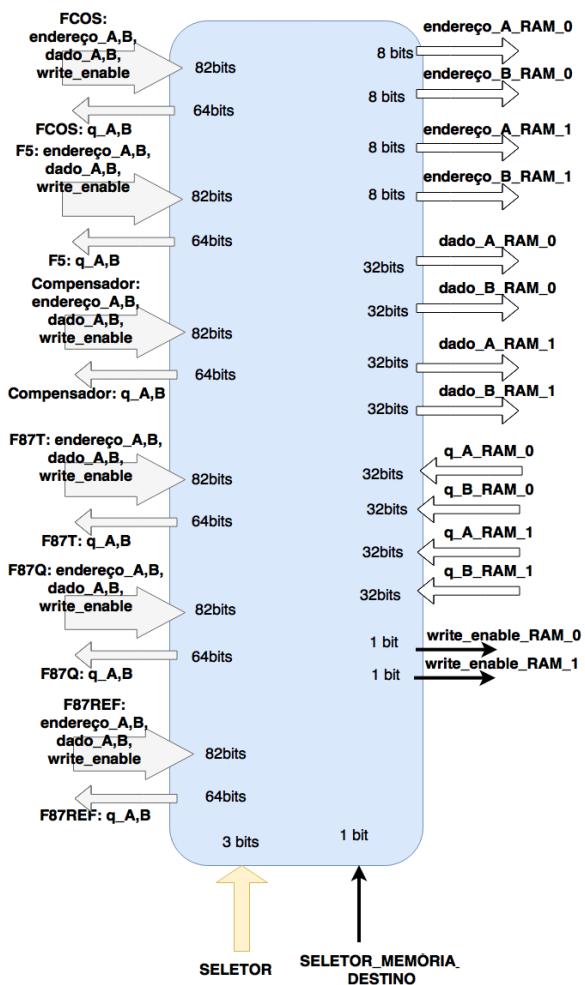
A implementação dos barramentos fez uso apenas de lógica combinacional, a qual não impôs atrasos significativos ao fluxo de dados. O controle do fluxo de dados é efetuado por meio das portas *seletor* (3 bits), nos dois barramentos, e *seletor_ram_destino* (1 bit), no BARRAMENTO_RAM_0_1. A necessidade de controlar o fluxo de dados é um dos fatos que deu origem ao módulo denominado MAESTRO, detalhado no tópico 4.2.16 dessa dissertação.

A **Tabela 4.9** apresenta resumo do uso dos recursos do FPGA, após a sintetização dos barramentos de acesso às memórias RAM.

Tabela 4.9 - Uso de Recursos do FPGA nos Barramentos de Acesso às Memórias RAM.

Barramento	BARRAMENTO_RAM_0_1	BARRAMENTO_RAM_2
Portas de 32 bits	36	16
Ciclos de <i>clock</i> por operação	-	-
Total de Elementos Lógicos Utilizados	470	210
Percentual de uso do FPGA adotado	4,55%	2,03%

Barramento_RAM_0_1



Barramento_RAM_2

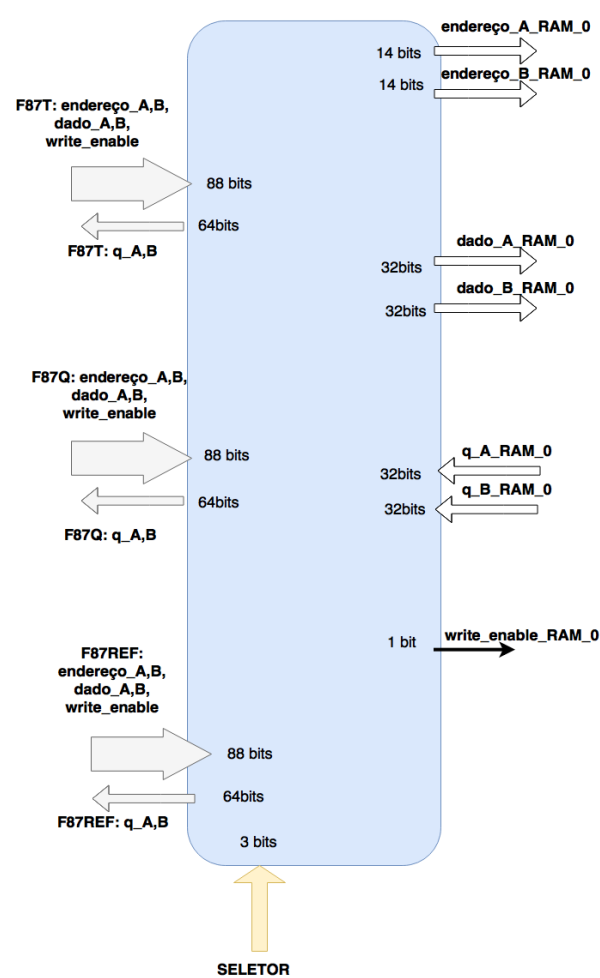


Figura 4.12 - Esquemático do Barramento_RAM_0_1.

Figura 4.13 - Esquemático do Barramento_RAM_2.

4.2.5 Barramento de Acesso ao Módulo MSA

De modo similar aos barramentos implementados para o compartilhamento das memórias RAM entre os módulos principais do sistema, faz necessário especificar um barramento para controle do fluxo de dados entre o módulo aritmético da MSA e os módulos. Este barramento é denominado BARRAMENTO_MSA. Apresenta-se na **Figura 4.14** o modelo simplificado do BARRAMENTO_MSA, indicando suas portas de entrada e saída.

Apresenta-se na **Tabela 4.10** resumo do uso dos recursos do FPGA, após a sintetização do barramento de acesso à MSA.

Tabela 4.10 - Uso de Recursos do FPGA no Barramento de Acesso ao Módulo Aritmético MSA.

Portas de 32 bits	15
Ciclos de <i>clock</i> por operação	-
Total de Elementos Lógicos Utilizados	793
Percentual de uso do FPGA adotado	7,68%

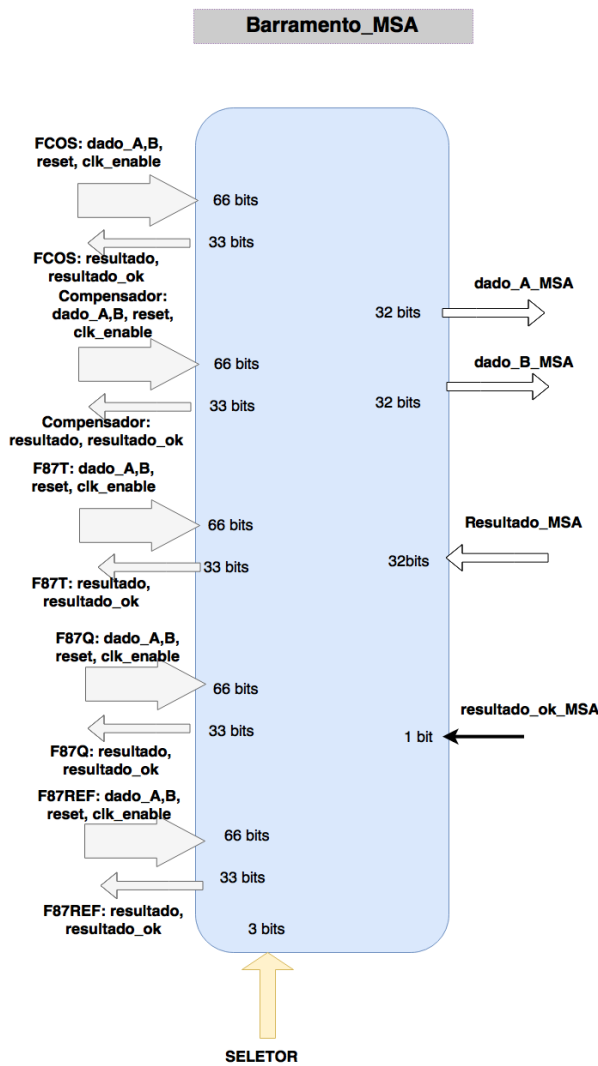


Figura 4.14 - Esquemático do Barramento_MSA.

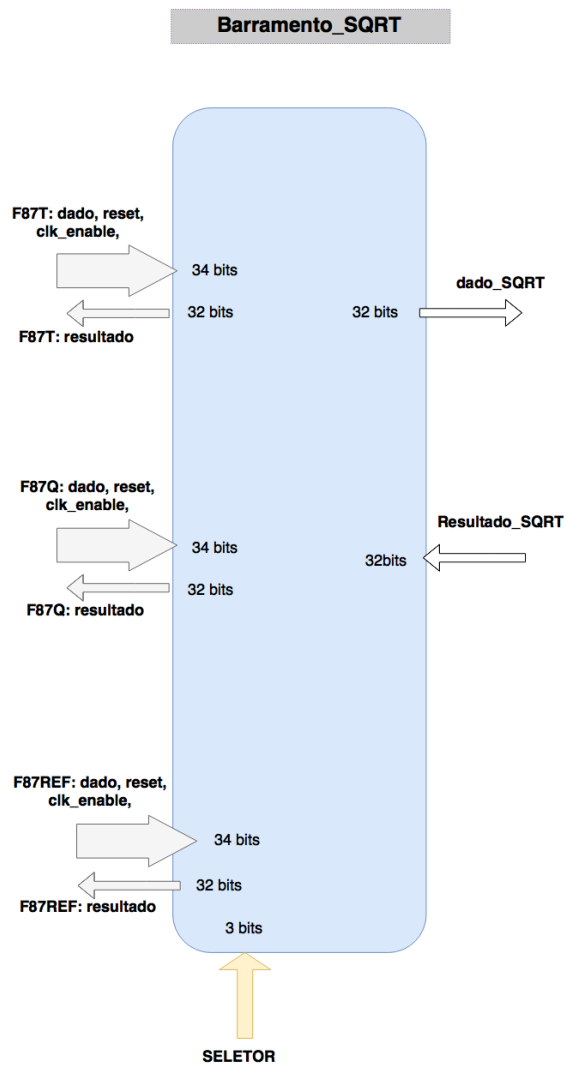


Figura 4.15 - Esquemático do Barramento_SQRT.

4.2.6 Barramento de Acesso ao Módulo SQR

O barramento que controla o acesso dos módulos principais ao módulo SQR foi projetado de modo similar aos barramentos implementados para o compartilhamento das memórias RAM e do módulo MSA. Este barramento é denominado BARRAMENTO_SQR. Apresenta-se na **Figura 4.15** o modelo simplificado do BARRAMENTO_SQR, indicando suas portas de entrada e saída.

A **Tabela 4.11** apresenta resumo do uso dos recursos do FPGA, após a sintetização do barramento de acesso ao módulo SQR.

Tabela 4.11 - Uso de Recursos do FPGA no Barramento de Acesso ao Módulo SQR.

Portas de 32 bits	6
Ciclos de <i>clock</i> por operação	-
Total de Elementos Lógicos Utilizados	34
Percentual de uso do FPGA adotado	0,33%

4.2.7 Módulo de *Buffer*

O módulo *Buffer* é responsável por captar as amostras de corrente, as quais podem ser geradas por um conversor A/D ou por qualquer outra espécie de dispositivo. No escopo do projeto do *Hardware* do Relé de Proteção Diferencial de Transformadores, o termo adotado para este módulo de entrada é F5. A topologia desenhada para este módulo de entrada e interface com o mundo real consiste em 7 canais independentes designados para as três fases de corrente do primário, as três fases de corrente do secundário do transformador e um canal designado para as correntes de neutro, caso disponível no transformador.

No projeto do *Hardware* do Relé de Proteção Diferencial de Transformadores, adotou-se a convenção *little-endian*, na qual os valores das amostras de corrente são recebidos no formato IEEE 754 em grupos de 8 bits. Vale lembrar que a taxa de amostragem adotada para as correntes apresentadas na entrada do *hardware* projetado é de 16 amostras por ciclo, o que corresponde a taxa de recepção de 3,84 kBytes por segundo, por canal de corrente. Em seguida, os dados (já agrupados em conjuntos de 32 bits) são disponibilizados ao módulo de estimação de fasores, por meio de escritas simultâneas em posições específicas nas memórias RAM_0 e RAM_1.

Por fim, as amostras de corrente são reorganizadas em posições pré-determinadas nas memórias RAM_0 e RAM_1. O módulo F5, a cada nova amostra gerada pelo submódulo de amostragem, salva nas memórias o novo valor de corrente, excluindo a amostra mais antiga. Uma vez concluídas as tarefas descritas, o módulo F5 sinaliza aos módulos de controle e de estimação fasorial que há uma nova amostra disponível em determinado canal de corrente.

O esquemático simplificado apresentado na **Figura 4.16** indica as portas de entrada e saída do módulo F5, bem como seus submódulos. A **Tabela 4.12** apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo F5.

Tabela 4.12 - Uso de Recursos do FPGA no módulo F5.

Portas de 32 bits	4
Ciclos de <i>clock</i> por operação	54
Total de Elementos Lógicos Utilizados	182
Percentual de uso do FPGA adotado	1,76%

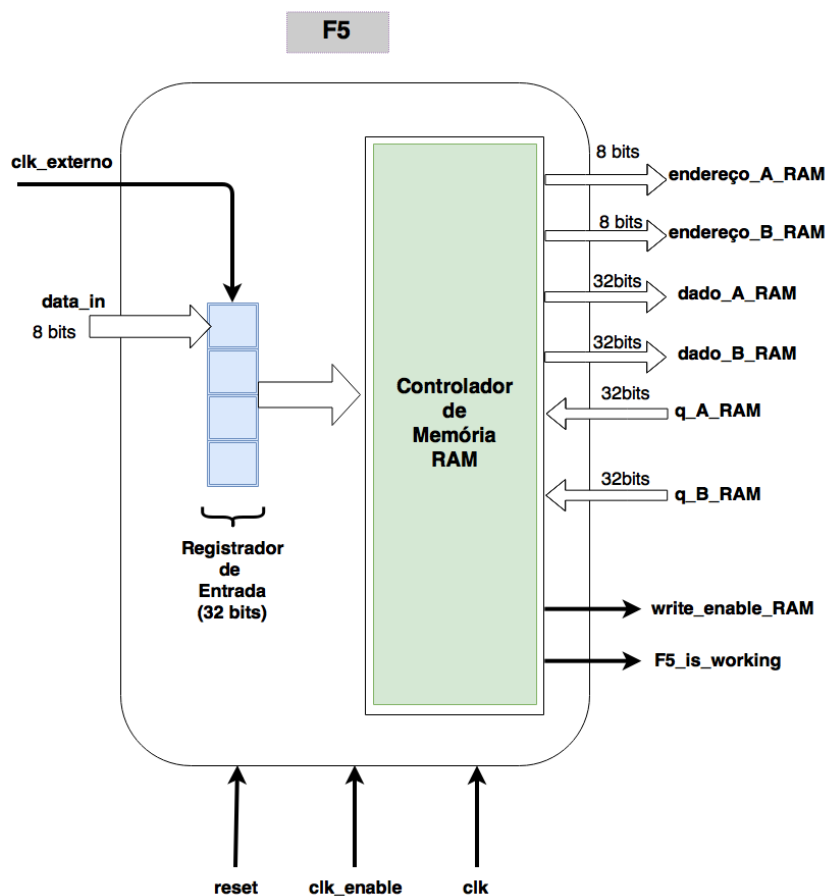


Figura 4.16 – Esquemático simplificado do módulo F5.

4.2.8 Módulo Filtro Cosseno

O processo de estimação de fasores é uma etapa essencial na detecção de faltas nos algoritmos de proteção diferencial implementados, uma vez que a maioria dos módulos faz uso das componentes simétricas da correntes que circulam pelo transformador. Neste sentido, o módulo desenvolvido no escopo deste projeto, denominado FCOS, adotou o **Algoritmo 4.1** como referência, que implementa o Filtro Cosseno Modificado da ABB.

A partir da discussão iniciada no tópico 4.2.2.5 dessa dissertação, verificou-se que é inviável sintetizar no FPGA adotado um módulo que execute todas operações do Filtro Cosseno em paralelo. O que nos levou a adotar uma solução de uso compartilhado dos recursos aritméticos, principalmente do módulo MSA.

Uma vez que o módulo MSA executa apenas uma operação *multiplica-soma-acumula* a cada 13 ciclos de *clock*, fez-se necessário implementar uma máquina de estados finitos para controlar as etapas de cálculo do filtro cosseno modificado da ABB, descritas no algoritmo de referência.

Basicamente, o módulo FCOS calcula a parte real do fasor estimado por meio da Equação (4.2):

$$RE\{\hat{I}\} = i[0]\cos[0] + i[1]\cos[1] + i[2]\cos[2] + \dots + i[15]\cos[15], \quad (4.2)$$

em que $i[0]$ corresponde à amostra de corrente mais antiga disponível no *buffer*, $i[15]$ é a amostra de corrente mais recente, $\cos[n]$ é o coeficiente do filtro cosseno modificado da ABB, calculado de acordo com a Equação (4.3).

$$\cos[n] = \frac{\sqrt{2}}{16} \cos\left(\frac{2\pi}{16} m \cdot n\right), \quad (4.3)$$

em que m é igual a 1, caso o filtro esteja sintonizado na frequência fundamental, ou igual a 2 no caso de a estimação fasorial ocorrer para a 2ª harmônica.

Para calcular a parte real do fasor a ser estimado, o módulo FCOS, a cada 13 ciclos de processamento, apresenta na entrada dos operandos da MSA uma amostra de corrente e um coeficiente do filtro, conforme indicado no diagrama de tempo da **Figura 4.17**. Tanto a amostra de corrente quanto o coeficiente do filtro são lidos de posições específicas de uma das memórias RAM_0 ou RAM_1. Ressalta-se que os coeficientes do filtro de primeira e segunda harmônica foram pré-calculados e salvos nas memórias antes do processo de síntese do *hardware*.

A decisão de qual memória será a origem dos dados a serem apresentados para a MSA depende para qual harmônica o filtro está sintonizado no momento. Se o filtro for sintonizado na frequência

fundamental, deve ser operada a memória RAM_0. Caso contrário, a memória RAM_1 será utilizada e a estimação dos fasores ocorrerá para a 2ª harmônica das amostras de corrente. Contudo, este processo de seleção da memória RAM de destino é transparente para o módulo FCOS, uma vez que é realizado exclusivamente pelo módulo de controle, denominado MAESTRO, tratado no tópico 4.2.16 dessa dissertação.

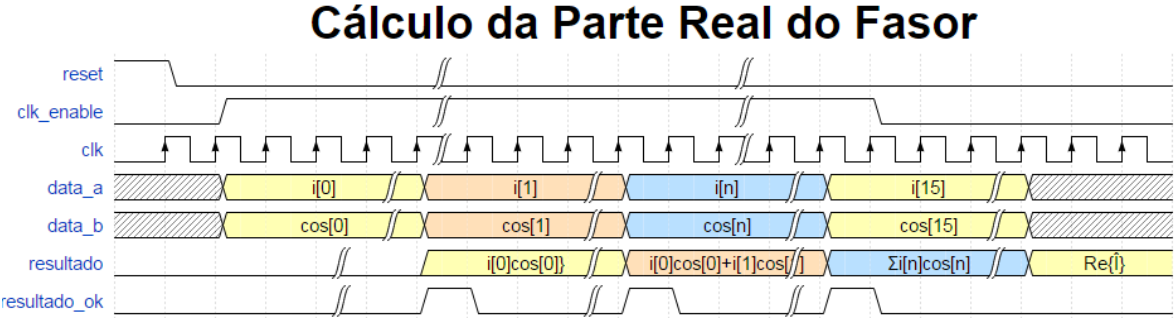


Figura 4.17 - Diagrama de Tempo da MSA ao Efetuar o Cálculo da Parte Real do Fasor.

Após a execução da operação *multiplica-soma-acumula* com as 16 amostras presentes nos endereços do *buffer* (administrado pelo módulo F5), o valor que se obtém na saída da MSA corresponde à parte real do fasor estimado, conforme indicado na Equação (4.2).

Por padrão, as partes reais e imaginárias dos fasores estimados são salvas em 4 posições de memória, agrupadas em dois blocos, conforme indicado na **Figura 4.18**. Os blocos são usados para armazenar o último fasor estimado e o fasor atual, para cada canal de corrente. Para definir qual bloco usar, o módulo FCOS efetua leitura do vetor VPAF (Figura 4.19), armazenado na posição 0x82. O bits armazenados no byte menos significativo deste vetor indicam qual bloco deverá ser utilizado pelo módulo FCOS. Uma vez calculada a parte real do fasor, e definido o bloco de armazenamento, o módulo FCOS salva este valor em uma posição pré-estabelecida na memória (RAM_0 ou RAM_1). A **Figura 4.20** apresenta os endereços de armazenamento reservados para os fasores estimados.

De acordo com o Algoritmo 4.1, a parte imaginária do fasor a ser estimada deve ser calculada conforme a Equação (4.4), que leva em consideração os valores das partes reais do fasor anterior e do fasor atual.

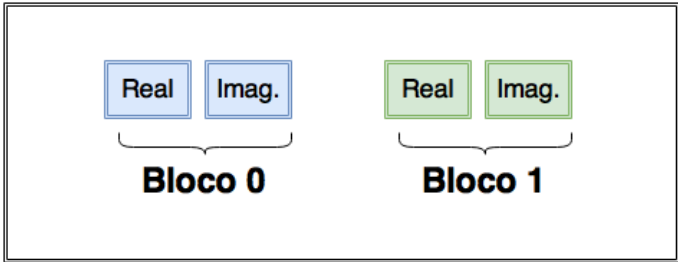


Figura 4.18 - Blocos de Armazenamento dos Vetores Estimados.

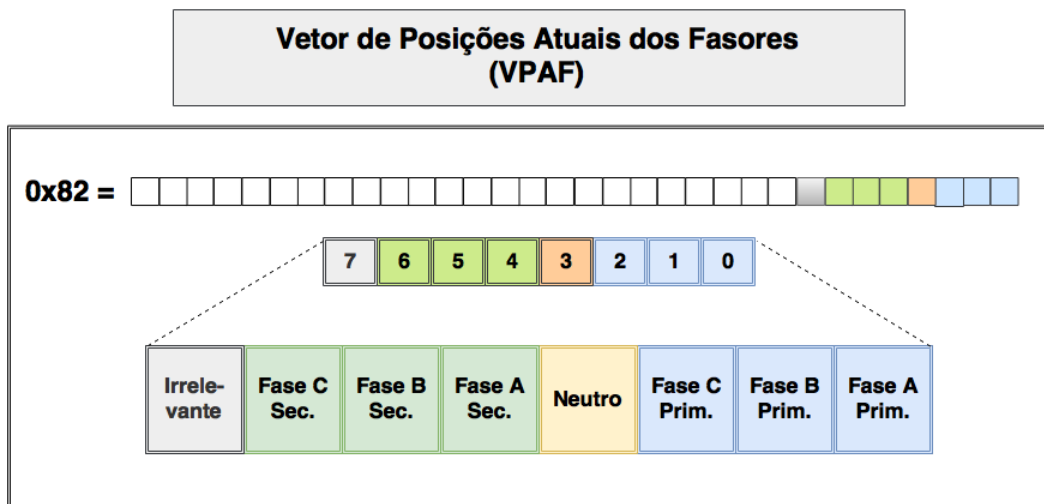


Figura 4.19 - Vetor de Posições Atuais dos Fasores - VPAF.

	Bloco 0	Bloco 1
Fase A Primário	0x90 0x91	0x92 0x93
Fase B Primário	0x94 0x95	0x96 0x97
Fase C Primário	0x98 0x99	0x9A 0x9B
NEUTRO	0x9C 0x9D	0x9E 0x9F
Fase A Secundário	0xA0 0xA1	0xA2 0xA3
Fase B Secundário	0xA4 0xA5	0xA6 0xA7
Fase C Secundário	0xA8 0xA9	0xAA 0xAB

Figura 4.20 - Endereços de Armazenamento dos Fasores nas Memórias RAM.

$$\text{Im} \{ \hat{I}_k \} = \text{Re} \{ \hat{I}_{k-1} \} - \text{Re} \{ \hat{I}_k \} \frac{\cos\left(\frac{2\pi}{16}\right)}{\text{sen}\left(\frac{2\pi}{16}\right)} \quad (4.4)$$

Desta forma, para o cálculo da parte imaginária do fasor, o módulo FCOS deve consultar o vetor VPAF a fim de localizar o valor da parte real do último fasor estimado e do fasor atual. Uma vez identificados os endereços de armazenamento, o FCOS realiza consulta a estes endereços na memória e prepara a MSA para efetuar o cálculo da Equação (4.4), conforme indicado no diagrama de tempo da Figura 4.21.

Uma vez calculada a parte imaginária do fasor, o módulo FCOS armazena o resultado na memória RAM designada.

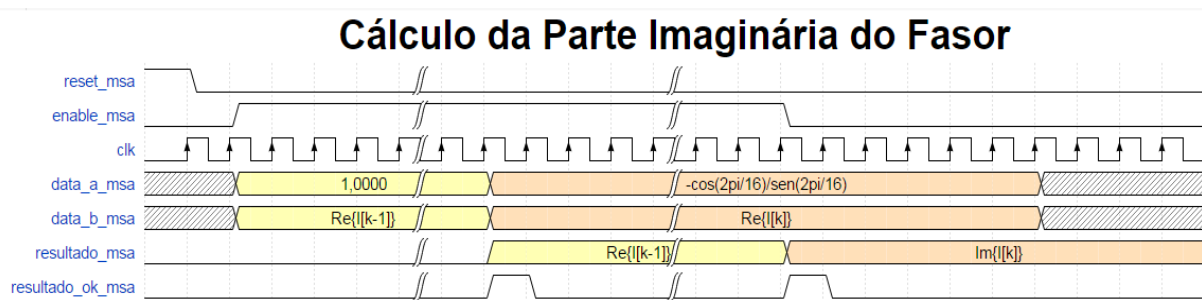


Figura 4.21 - Diagrama de Tempo da MSA ao Efetuar o Cálculo da Parte Imaginária do Fasor.

Como última tarefa do módulo FCOS, resta atualizar o vetor VPAF, invertendo o valor do bit relacionado com a fase para qual se estimou o fasor. Ao inverter o bit, deixa-se a mensagem para a próxima execução da estimação fasorial de qual bloco de memória deve ser utilizado.

A **Tabela 4.13** apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo FCOS e a **Figura 4.22** apresenta o diagrama esquemático simplificado do módulo FCOS, indicando suas interfaces. Observa-se que, por meio da porta *in_fase* de 3 (três) bits, o módulo controlador do *Hardware* do Relé de Proteção Diferencial de Transformadores indica ao módulo FCOS para qual canal de corrente (fase) a estimação fasorial será calculada. Com base nesta informação, o FCOS tem condições de determinar com quais endereços de memória trabalhar.

Tabela 4.13 - Uso de Recursos do FPGA no módulo FCOS.

Portas de 32 bits	7
Ciclos de <i>clock</i> por operação	367
Total de Elementos Lógicos Utilizados	179
Percentual de uso do FPGA adotado	1,73%

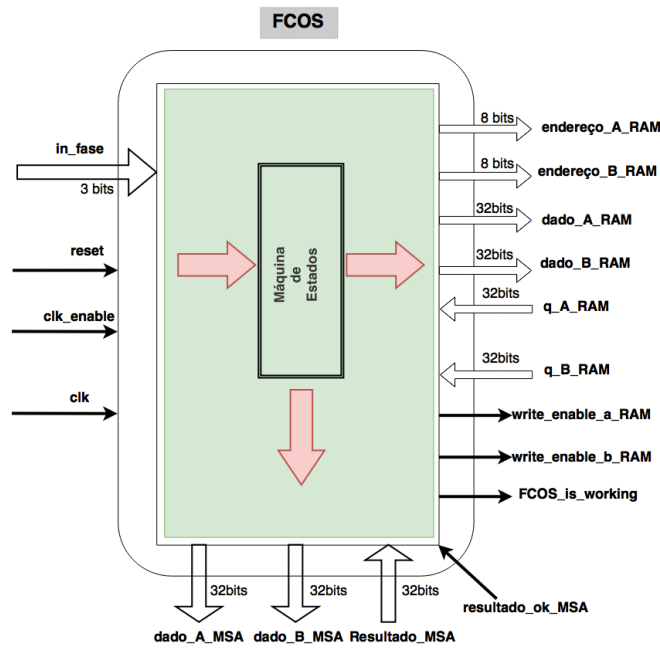


Figura 4.22 - Diagrama Simplificado Esquemático do Módulo FCOS.

4.2.9 Módulo Compensador

Conforme tratado nos tópicos 3.3.1, 3.3.2 e 3.3.3 dessa dissertação, o processo de compensação das correntes ocorre após a estimação dos fasores. O módulo Compensador desenhado para o *Hardware* do Relé de Proteção Diferencial de Transformadores, de modo análogo ao módulo FCOS, trabalha com dados salvos na memória RAM e efetua cálculos com auxílio do módulo aritmético MSA.

A principal função do módulo compensador é efetuar as multiplicações matriciais indicadas na Equações (4.5) e (4.6), onde os elementos A_H , B_H , C_H , (...), I_H , da matriz indicada na Equação (4.5) são os coeficientes previamente calculados (durante o processo de configuração do relé) para compensar as relações de transformação dos TCs, relação de transformação do transformador a ser protegido e eventuais correções de defasagem angular e remoção de componentes de sequência zero, no lado primário do transformador. O mesmo ocorre para os elementos A_X , B_X , C_X , (...), I_X , da matriz indicada na Equação (4.6), referentes ao lado secundário do transformador. Observa-se, ainda, que as Equações (4.5) e (4.6) guardam relação com as Equações (3.8), de compensação de defasagem angular, e (3.9), de compensação da corrente de sequência zero.

$$\begin{bmatrix} \hat{I}_{a_compH} \\ \hat{I}_{b_compH} \\ \hat{I}_{c_compH} \end{bmatrix} = \begin{bmatrix} A_H & B_H & C_H \\ D_H & E_H & F_H \\ G_H & H_H & I_H \end{bmatrix} \begin{bmatrix} \hat{I}_{a_H} \\ \hat{I}_{b_H} \\ \hat{I}_{c_H} \end{bmatrix}, \quad (4.5)$$

$$\begin{bmatrix} \hat{I}_{a_compX} \\ \hat{I}_{b_compX} \\ \hat{I}_{c_compX} \end{bmatrix} = \begin{bmatrix} A_x & B_x & C_x \\ D_x & E_x & F_x \\ G_x & H_x & I_x \end{bmatrix} \begin{bmatrix} \hat{I}_{a_x} \\ \hat{I}_{b_x} \\ \hat{I}_{c_x} \end{bmatrix}. \quad (4.6)$$

O módulo Compensador desempenha sua função ao fazer uso da operação *multiplica-soma-acumula*, executada a cada 13 ciclos de *clock* pelo módulo aritmético MSA.

Para calcular as correntes compensadas, inicialmente o módulo Compensador, a cada 13 ciclos de processamento, apresenta na entrada dos operandos da MSA a parte real do fasor da fase A e o elemento *A* da matriz de compensação. Concluída a primeira operação da MSA, em seguida, o Compensador apresenta nas entradas dos operandos da MSA a parte real do fasor da fase B e o elemento *B* da matriz de compensação. Efetuada esta segunda operação, o Compensador apresenta para a MSA a parte real do fasor da fase C e o elemento *C* da matriz de compensação. O resultado apresentado é salvo em posição específica da memória como sendo a parte real do fasor compensado da fase A.

Para o cálculo das partes reais e imaginárias dos fasores das demais fases, o processo é similar, devendo-se primeiramente aplicar um *reset* no módulo MSA para zerar seu registrador acumulador interno. A Figura 4.23 exemplifica, por meio de um diagrama de tempo, o modo de operação do módulo Compensador para o cálculo da parte imaginária do fasor compensado da fase B.

De modo idêntico ao projetado para o módulo FCOS, a decisão de qual RAM será utilizada no processo é efetuada pelo módulo MAESTRO, definido no tópico 4.2.16 dessa dissertação.

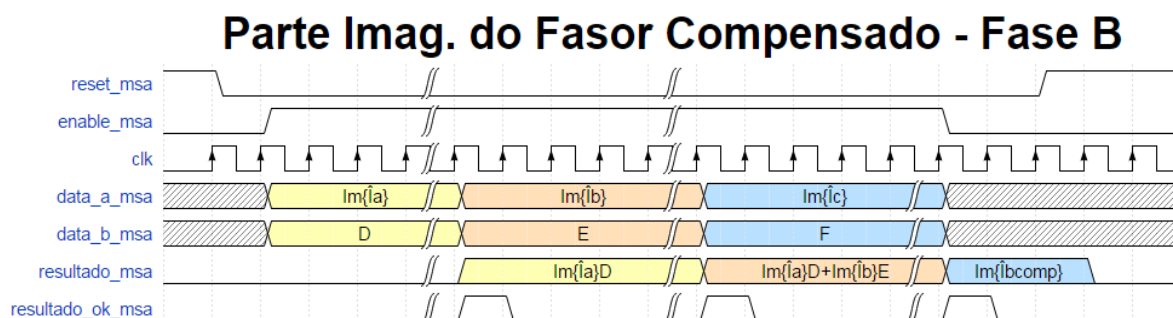


Figura 4.23 - Diagrama de Tempo da MSA ao Efetuar o Cálculo da Parte Imaginária do Fasor Compensado da fase B.

A **Tabela 4.14** apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo Compensador e a **Figura 4.24** apresenta o diagrama esquemático simplificado do módulo Compensador, indicando suas interfaces.

Tabela 4.14 - Uso de Recursos do FPGA no módulo Compensador.

Portas de 32 bits	7
Ciclos de <i>clock</i> por operação	139

Total de Elementos Lógicos Utilizados	287
Percentual de uso do FPGA adotado	2,78%

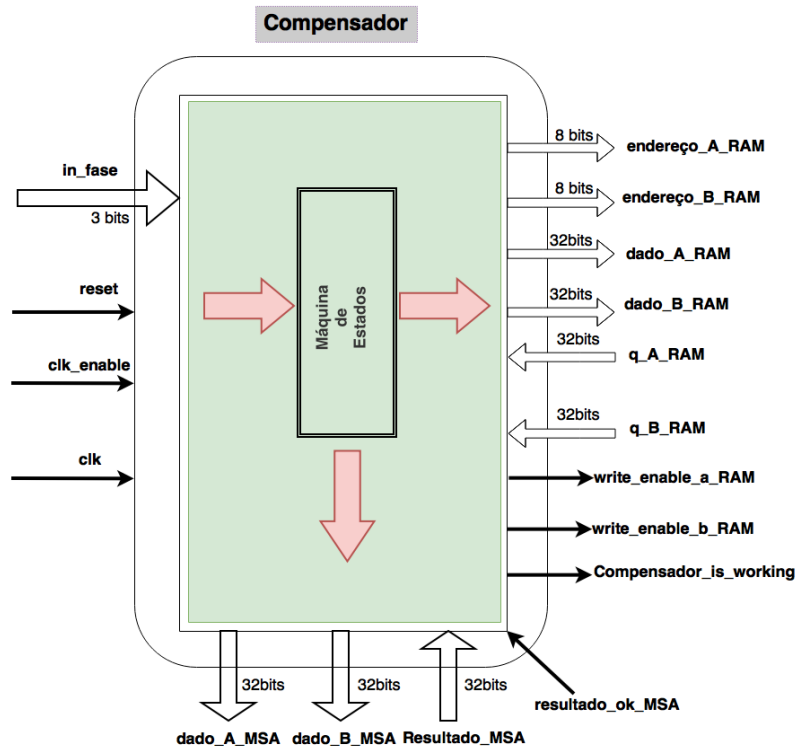


Figura 4.24 - Diagrama Esquemático Simplificado do Módulo Compensador.

4.2.10 Módulo F87T

O módulo denominado F87T tem como objetivo implementar a função diferencial percentual de uma fase do transformador com restrição de harmônicos, conforme discutido no tópico 3.2 dessa dissertação. Neste sentido, o módulo F87T desenvolvido para o FPGA adotou o **Algoritmo 4.2** como referência.

Pode-se observar do código de referência (**Algoritmo 4.2**) que o módulo F87T necessita dos três módulos aritméticos sintetizados no FPGA: a) o módulo MSA, para as operações de soma e multiplicação; b) o módulo SQRT, necessário para o cálculo dos módulos dos fasores das correntes (raiz quadrada da soma dos quadrados da parte real e imaginária); e c) o módulo comparador.

Uma vez constituída a estrutura de barramentos para acesso aos módulos aritméticos MSA e SQRT, e sendo seus acessos geridos pelo módulo MAESTRO, a execução das operações elencadas no algoritmo de referência do módulo F87T pode ser realizada de forma similar àquelas praticadas nos módulos FCOS e Compensador.

Algoritmo 4.2 - Código C++ da Função F87T Implementada no FPGA.

```
1. bool F87Tx::run(Complexo &IH, Complexo &IX, Complexo &IH2, Complexo &IX2) {
2.
3.     tempo();
4.     iop = (IH + IX).abs();
5.     irect = IH.abs() + IX.abs();
6.     irect2 = IH2.abs() + IX2.abs();
7.     irect = irect + (1/slope) * (100/k2) * irect2;
8.
9.     hist_iop[k_esima_amostra] = iop;
10.    hist_irect [k_esima_amostra] = irect2;
11.
12.    bool condicao1 = iop > pickup;
13.    bool condicao2 = iop > (slope*irect);
14.
15.    TRIP = TRIP || (condicao1 && condicao2);
16.
17.    log_trip[k_esima_amostra] = TRIP;
18.
19.    tempo();
20.    return TRIP;
21.
22. }
```

Contudo, uma vez que há disponibilidade de recursos no FPGA e a função F87T é função de proteção intrínseca que é solicitada na maioria dos eventos e faltas aos quais um transformador de potência é submetido, adotou-se a estratégia de sintetizar um módulo MSA e um módulo SQRT exclusivos e embarcados no módulo F87T, o que permite que esta função de proteção possa operar efetivamente em paralelo com as demais funções de proteção presentes no relé desenvolvido. Com base nisso, a **Figura 4.25** apresenta o diagrama esquemático simplificado do módulo F87T, com algumas de suas conexões internas e interfaces de entrada e saída.

De modo similar aos demais módulos, fez-se necessário implementar uma máquina de estados finitos para controlar as etapas de cálculo do função F87T, descritas no algoritmo de referência.

A primeira rodada de cálculos envolve a obtenção dos valores dos módulos dos fasores das correntes primárias e secundárias estimados para frequência fundamental e de 2ª harmônica, que subsidiam os cálculos das correntes de operação *Iop*, e de restrição *Ires* e *Ires2*.

Na sequência, a máquina de estados do módulo F87T faz uso dos valores *slope_invertido* e *k2_100*, pré-calculados durante o processo de configuração do relé, para obtenção do valor da corrente *Icomp*.

Por fim, o módulo F87T procede às comparações, por meio do módulo Comparador: a) entre a corrente de operação *Iop* e valor *Ipickup* apresentado na interface de entrada do módulo; e b) entre a corrente de operação *Iop* e *Icomp*, multiplicada pelo valor de *SLOPE*, também apresentado na interface de entrada do módulo.

De acordo com o resultado da comparação, a porta *saída_f87t* pode ser acionada em caso de detecção de TRIP pelo módulo F87T.

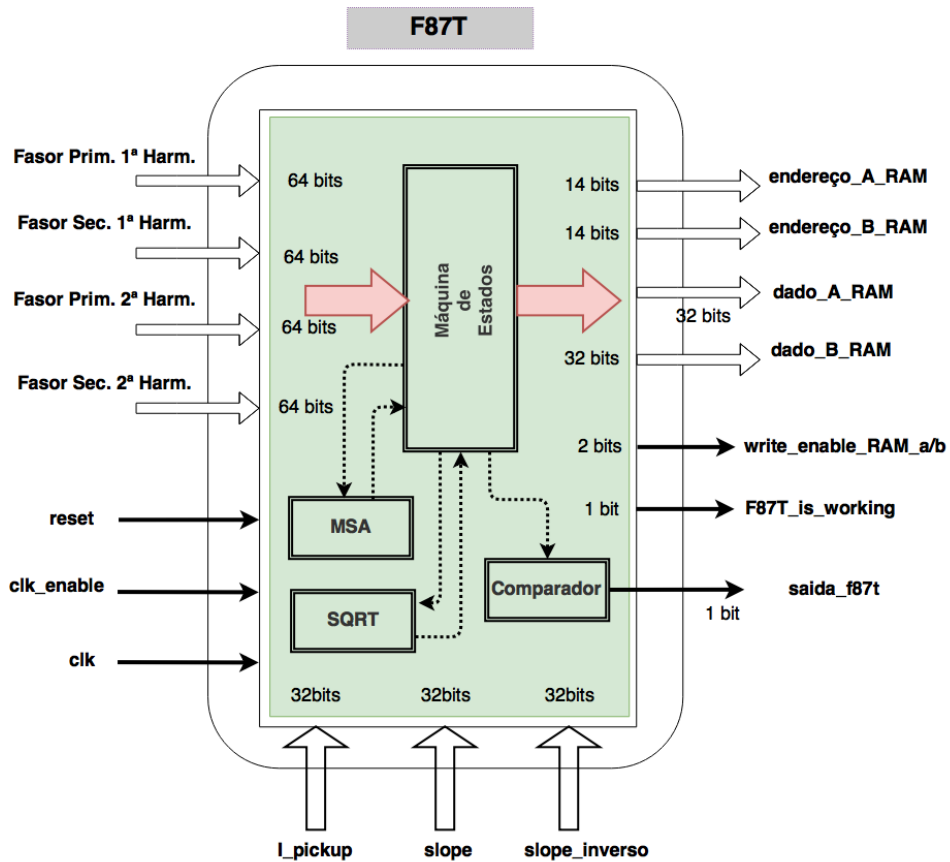


Figura 4.25 - Diagrama Esquemático Simplificado do Módulo F87T.

Os valores calculados das correntes I_{op} e I_{comp} são salvos na memória RAM_2, os quais fundamentam as análises realizadas no Capítulo 5.

Apresenta-se na **Tabela 4.15** resumo do uso dos recursos do FPGA, após a sintetização do módulo F87T.

Tabela 4.15 - Uso dos Recursos do FPGA pelo Módulo F87T.

Operandos de 32 bits	13
Memória (bits)	157
Multiplicadores embarcados 18 x 18	3
Ciclos de <i>clock</i> por operação	469
Total de Elementos Lógicos Utilizados	2.928
Percentual de uso do FPGA adotado	28,37%

4.2.11 Módulo F87Q

O módulo denominado F87Q tem como objetivo implementar a função diferencial de sequência negativa. Este módulo opera de modo similar ao módulo F87T, com passos de cálculos adicionais relacionados à obtenção dos valores das componentes de sequência negativa, obtidas com base nos fasores estimados e compensados pelos módulos FCOS e Compensador.

Neste sentido, o módulo F87TQ desenvolvido para o FPGA adotou o **Algoritmo 4.3** como referência.

Algoritmo 4.3 - Código C++ da Função F87Q Implementada no FPGA.

```
1. bool F87Q::run(Complexo &IaH, Complexo &IbH, Complexo &IcH, Complexo &IaX,  
2.               Complexo &IbX, Complexo &IcX) {  
3.  
4.     tempo();  
5.     Complexo IQH = (IaH + (a2 * IbH) + (a * IcH)) * (1.0f/3.0f);  
6.     Complexo IQX = (IaX + (a2 * IbX) + (a * IcX)) * (1.0f/3.0f);  
7.  
8.     iop = (IQH + IQX).abs();  
9.     irect = (IQH.abs() > IQX.abs()) ? IQH.abs() : IQX.abs();  
10.  
11.    bool condicao1 = iop > pickup;  
12.    bool condicao2 = iop > (slope*irect);  
13.  
14.  
15.    TRIP = TRIP || (condicao1 && condicao2 && ENABLE);  
16.  
17.  
18.    log_trip[k_esima_amostra] = TRIP;  
19.  
20.    tempo();  
21.    return TRIP;  
22. }
```

Pode-se observar do código de referência (**Algoritmo 4.3**) que o módulo F87Q necessita dos três módulos aritméticos sintetizados no FPGA: a) o módulo MSA, para as operações de soma e multiplicação; b) o módulo SQRT, necessário para o cálculo dos módulos dos fasores das correntes; e c) o módulo comparador.

Diferentemente da estratégia adotada para o módulo F87T, o módulo F87Q compartilha os módulos aritméticos MSA e SQRT por meio de seus respectivos barramentos. O Comparador é o único módulo aritmético embarcado no F87Q. A **Figura 4.26** apresenta o diagrama esquemático simplificado do módulo F87Q.

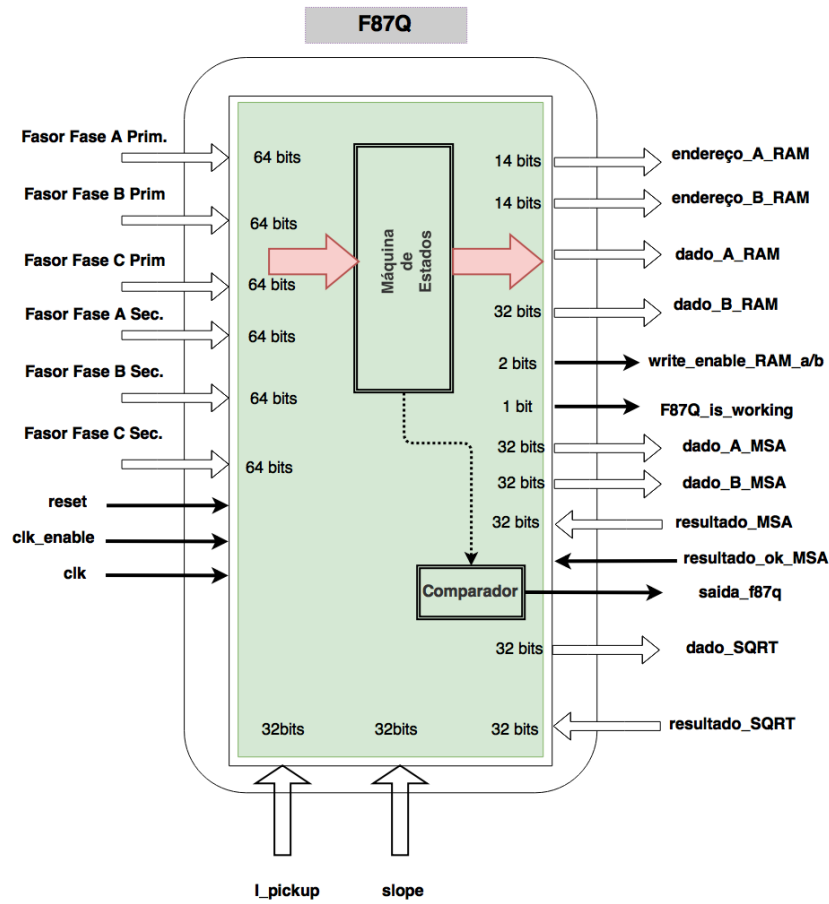


Figura 4.26 - Diagrama Esquemático Simplificado do Módulo F87Q.

Para operar os módulos aritméticos e memória RAM envolvidos nas operações indicadas no algoritmo de referência, foi constituída uma máquina de estados finitos para executar todas as etapas de cálculo da função F87Q.

A primeira rodada de cálculos da máquina de estados consiste em obter os valores dos fasores das correntes de sequência negativa primária e secundária, de acordo com as Equações (4.7), (4.8), (4.9) e (4.10), as quais foram obtidas a partir da expansão da relação matricial que converte fasores de fases em suas componentes simétricas.

$$\operatorname{Re}\{\hat{I}_{QH}\} = \frac{1}{3} \begin{bmatrix} \operatorname{Re}\{\hat{I}_{a_prim}\} \\ -0,5 \cdot \operatorname{Re}\{\hat{I}_{b_prim}\} \\ +0,8667 \cdot \operatorname{Im}\{\hat{I}_{b_prim}\} \\ -0,5 \cdot \operatorname{Re}\{\hat{I}_{c_prim}\} \\ -0,8667 \cdot \operatorname{Im}\{\hat{I}_{c_prim}\} \end{bmatrix}, \quad (4.7)$$

$$\text{Im} \{ \hat{I}_{QH} \} = \frac{1}{3} \begin{bmatrix} \text{Im} \{ \hat{I}_{a_prim} \} \\ -0,8667 \cdot \text{Re} \{ \hat{I}_{b_prim} \} \\ -0,5 \cdot \text{Im} \{ \hat{I}_{b_prim} \} \\ +0,8867 \cdot \text{Re} \{ \hat{I}_{c_prim} \} \\ -0,5 \cdot \text{Im} \{ \hat{I}_{c_prim} \} \end{bmatrix}, \quad (4.8)$$

$$\text{Re} \{ \hat{I}_{QX} \} = \frac{1}{3} \begin{bmatrix} \text{Re} \{ \hat{I}_{a_sec} \} \\ -0,5 \cdot \text{Re} \{ \hat{I}_{b_sec} \} \\ +0,8667 \cdot \text{Im} \{ \hat{I}_{b_sec} \} \\ -0,5 \cdot \text{Re} \{ \hat{I}_{c_sec} \} \\ -0,8667 \cdot \text{Im} \{ \hat{I}_{c_sec} \} \end{bmatrix}, \quad (4.9)$$

$$\text{Im} \{ \hat{I}_{QX} \} = \frac{1}{3} \begin{bmatrix} \text{Im} \{ \hat{I}_{a_sec} \} \\ -0,8667 \cdot \text{Re} \{ \hat{I}_{b_sec} \} \\ -0,5 \cdot \text{Im} \{ \hat{I}_{b_sec} \} \\ +0,8867 \cdot \text{Re} \{ \hat{I}_{c_sec} \} \\ -0,5 \cdot \text{Im} \{ \hat{I}_{c_sec} \} \end{bmatrix}. \quad (4.10)$$

Na sequência, a máquina de estados do módulo F87Q trabalha com os módulos aritméticos MSA, SQRT e Comparador na obtenção dos valores dos módulos dos fasores das correntes de sequência negativa primária e secundária, obtidos na etapa anterior, os quais subsidiam os cálculos das correntes de operação $IopQ$, e de restrição $IresQ$.

Por fim, o módulo F87Q procede às comparações, por meio do módulo Comparador: a) entre a corrente de operação $IopQ$ e valor $IpickupQ$ apresentado na interface de entrada do módulo; e b) entre a corrente de operação $IopQ$ e $IresQ$, multiplicada pelo valor de $SLOPEQ$, também apresentado na interface de entrada do módulo.

De acordo com o resultado da comparação, a porta $saída_f87q$ pode ser acionada em caso de detecção de TRIP pelo módulo F87Q.

Os valores calculados das correntes $IopQ$ e $IresQ$ são salvos na memória RAM_2, os quais fundamentam as análises realizadas no Capítulo 5.

A **Tabela 4.16** apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo F87Q.

Tabela 4.16 - Uso dos Recursos do FPGA pelo Módulo F87Q.

Operandos de 32 bits	20
Ciclos de <i>clock</i> por operação	604
Total de Elementos Lógicos Utilizados	770
Percentual de uso do FPGA adotado	7,46%

4.2.12 Módulo F87REF

O módulo denominado F87REF tem como objetivo implementar a função diferencial de faltas à terra restrita. Este módulo opera de modo similar ao módulo F87T, com passos de cálculos adicionais relacionados à obtenção dos valores das componentes de sequência zero, obtidas com base nos fasores estimados e compensados pelos módulos FCOS e Compensador.

Neste sentido, o módulo F87REF desenvolvido para o FPGA adotou o **Algoritmo 4.4** como referência.

Algoritmo 4.4 - Código C++ da Função F87REF Implementada no FPGA.

```
1.  bool F87REF::run(Complexo &In, Complexo &Ia, Complexo &Ib, Complexo &Ic) {
2.
3.     tempo();
4.     Complexo Inn = In * (1.0/TAP);
5.     Complexo I0 = (Ia + Ib + Ic) * (1.0f/TAP/3.0f);
6.
7.     iop = Inn.abs();
8.     irest = Kr * ( (Inn + ( I0 * (-
9.     3.0f))).abs() - (Inn + (I0 * 3.0f)).abs() );
10.
11.    bool condicao1 = iop > pickup;
12.    bool condicao2 = iop > (slope*irest);
13.    TRIP = TRIP || (condicao1 && condicao2 && ENABLE);
14.
15.    log_trip[k_esima_amostra] = TRIP;
16.
17.    tempo();
18.    return TRIP;
19.
20. }
```

Pode-se observar do código de referência (**Algoritmo 4.4**) que o módulo F87REF necessita dos três módulos aritméticos sintetizados no FPGA: a) o módulo MSA, para as operações de soma e multiplicação; b) o módulo SQRT, necessário para o cálculo dos módulos dos fasores das correntes; e c) o módulo comparador.

De forma similar ao módulo F87Q, o módulo F87REF compartilha os módulos aritméticos MSA e SQRT por meio de seus respectivos barramentos. O Comparador é o único módulo aritmético

embarcado no F87REF. A **Figura 4.27** apresenta o diagrama esquemático simplificado do módulo F87REF.

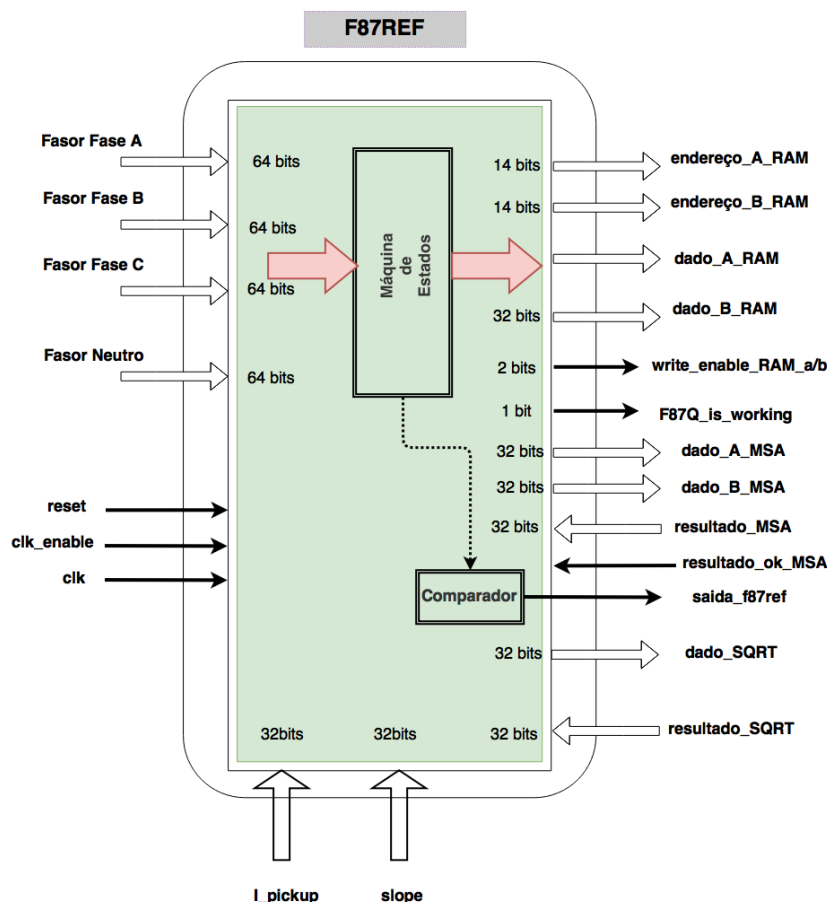


Figura 4.27 - Diagrama Esquemático Simplificado do Módulo F87REF.

Para operar os módulos aritméticos e memória RAM envolvidos nas operações indicadas no algoritmo de referência, foi constituída uma máquina de estados finitos para executar todas as etapas de cálculo da função F87REF.

A primeira rodada de cálculos da máquina de estados consiste em obter o valor do fasor da corrente de sequência zero, de acordo com as Equações (4.11) e (4.12), as quais foram obtidas a partir da expansão da relação matricial que converte fasores de fases em suas componentes simétricas, levando em consideração que o lado primário do transformador tem a configuração estrela aterrada.

$$\operatorname{Re} \{ \hat{I}_0 \} = \frac{3}{TAPH} \begin{bmatrix} \operatorname{Re} \{ \hat{I}_{a_prim} \} \\ + \operatorname{Re} \{ \hat{I}_{b_prim} \} \\ + \operatorname{Re} \{ \hat{I}_{c_prim} \} \end{bmatrix}, \quad (4.11)$$

$$\operatorname{Im} \{ \hat{I}_0 \} = \frac{3}{TAPH} \begin{bmatrix} \operatorname{Im} \{ \hat{I}_{a_prim} \} \\ + \operatorname{Im} \{ \hat{I}_{b_prim} \} \\ + \operatorname{Im} \{ \hat{I}_{c_prim} \} \end{bmatrix}. \quad (4.12)$$

Na sequência, a máquina de estados do módulo trabalha com os módulos aritméticos MSA e SQRT na obtenção do valor da corrente de operação I_{opREF} , por meio do cálculo do módulo do fasor da corrente de neutro. A corrente de restrição I_{resREF} é obtida por meio do cálculo indicado na Equação (3.20).

Por fim, o módulo F87REF procede às comparações, por meio do módulo Comparador: a) entre a corrente de operação I_{opREF} e valor $I_{pickupREF}$ apresentado na interface de entrada do módulo; e b) entre a corrente de operação I_{opREF} e I_{resREF} , multiplicada pelo valor de $SLOPEREF$, também apresentado na interface de entrada do módulo.

De acordo com o resultado da comparação, a porta $saída_{f87ref}$ pode ser acionada em caso de detecção de TRIP pelo módulo F87REF.

Os valores calculados das correntes I_{opREF} e I_{resREF} são salvos na memória RAM_2, os quais fundamentam as análises realizadas no Capítulo 5.

A **Tabela 4.17** apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo F87REF.

Tabela 4.17 - Uso dos Recursos do FPGA pelo Módulo F87REF.

Operandos de 32 bits	16
Ciclos de <i>clock</i> por operação	487
Total de Elementos Lógicos Utilizados	669
Percentual de uso do FPGA adotado	6,48%

4.2.13 Módulo de Bloqueio por Harmônicos

A função do módulo de bloqueio por harmônicos, denominado BLOQH, é identificar e sinalizar ao módulo de tratamento de TRIPs a considerável presença de harmônicos nos sinais de corrente do transformador protegido, de modo a evitar atuações indevidas de módulos mais sensíveis, como o F87Q e o F87REF.

O módulo BLOQH desenvolvido para o FPGA adotou o **Algoritmo 4.5** como referência. A atuação do módulo ocorre fase a fase, o que permite flexibilizar o projeto quanto ao uso de um esquema de bloqueio cruzado ou individualizado por fase.

Algoritmo 4.5 - Código C++ do Bloqueio por Harmônicos Implementada no FPGA.

```
1. bool Bloqueio_Harmonicos::run (Complexo &IH, Complexo &IX, Complexo &IH2) {  
2.  
3.     tempo();  
4.     float iop = (IH + IX).abs();  
5.  
6.     float percentual = IH2.abs()/iop;  
7.  
8.     bool condicao1 = percentual >= k2;  
9.  
10.    BLOQUEIO_ON = BLOQUEIO_ON || (condicao1);  
11.  
12.    log_trip[k_esima_amostra] = BLOQUEIO_ON;  
13.  
14.    tempo();  
15.    return BLOQUEIO_ON;  
16. }  
17.
```

Observa-se no código de referência (**Algoritmo 4.5**) que o módulo BLOQH necessita dos três módulos aritméticos sintetizados no FPGA: a) o módulo MSA, para as operações de soma e multiplicação; b) o módulo SQRT, necessário para o cálculo dos módulos dos fasores das correntes; e c) o módulo comparador. A operação de divisão indicada na linha 6 do algoritmo de referência foi substituída por uma multiplicação, conforme indicado na Equação (4.13), o que nos permite fazer uso de funções já disponíveis no *hardware* desenvolvido.

$$|I_{2H}| \geq k_{2b} \cdot I_{op} \quad (4.13)$$

De forma similar à maioria dos módulos sintetizados no FPGA, o módulo BLOQH compartilha os módulos aritméticos MSA e SQRT por meio de seus respectivos barramentos. O Comparador é o único módulo aritmético embarcado no BLOQH. A **Figura 4.28** apresenta o diagrama esquemático simplificado do módulo BLOQH.

Para operar os módulos aritméticos e memória RAM envolvidos nas operações indicadas no algoritmo de referência, foi constituída uma máquina de estados finitos para executar todas as etapas de cálculo da função BLOQH.

A primeira rodada de cálculos da máquina de estados consiste em obter o valor da corrente de operação I_{op} , com base no valor do módulo da soma dos fasores das correntes primárias e secundárias. Uma vez calculado o valor da corrente de operação, este é multiplicado, com o módulo MSA, pelo fator de bloqueio por harmônicos k_{2b} .

Na sequência, o módulo BLOQH calcula o módulo do fasor da corrente de 2ª harmônica, apresentada em sua interface de entrada. Por fim, por meio do módulo Comparador, verifica-se se é atendida a condição de bloqueio por harmônicos indicado na Equação (4.13). De acordo com o resultado da comparação, a porta *saída_bloqh* pode ser acionada.

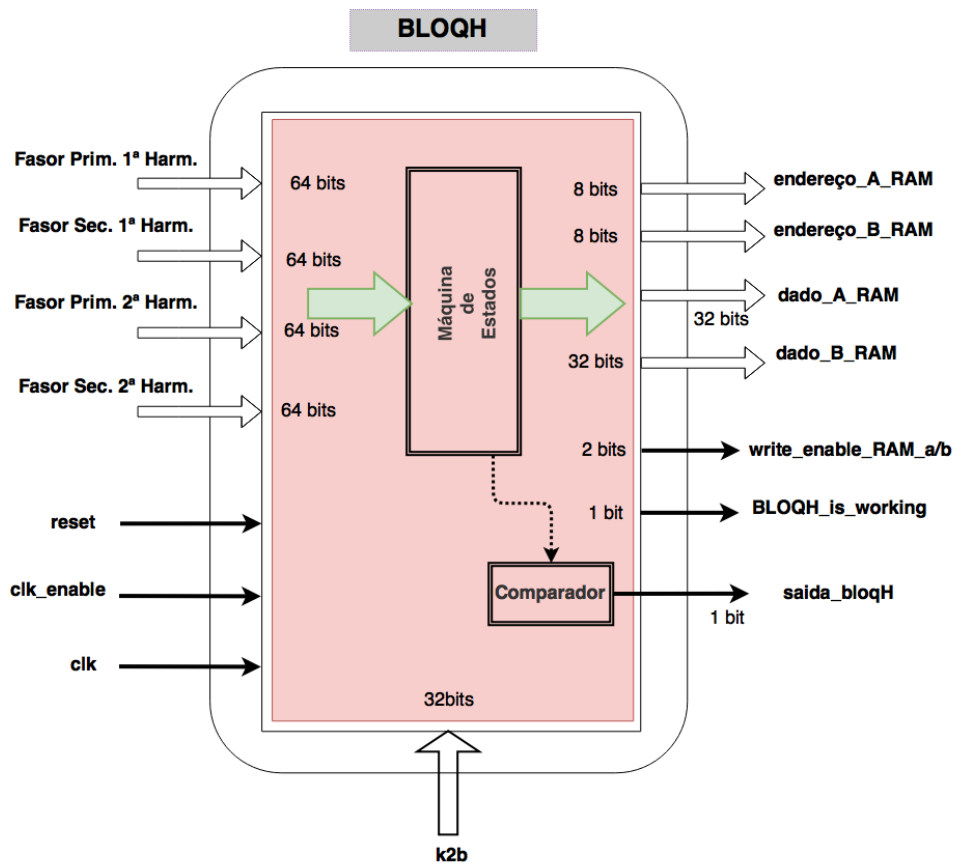


Figura 4.28 - Diagrama Esquemático Simplificado do Módulo BLOQH.

A Tabela 4.18 apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo BLOQH.

Tabela 4.18 - Uso dos Recursos do FPGA pelo Módulo BLOQH.

Operandos de 32 bits	11
Ciclos de <i>clock</i> por operação	236
Total de Elementos Lógicos Utilizados	642
Percentual de uso do FPGA adotado	6,22%

4.2.14 Módulo de Tratamento de TRIPS

O tratamento de todos os TRIPs oriundos dos módulos F87T, F87Q e F87REF é executado pelo módulo denominado TRIP87. Ao compatibilizar os sinais de bloqueio de cada fase, o módulo TRIP87 foi sintetizado com a lógica combinacional indicada na Figura 4.29.

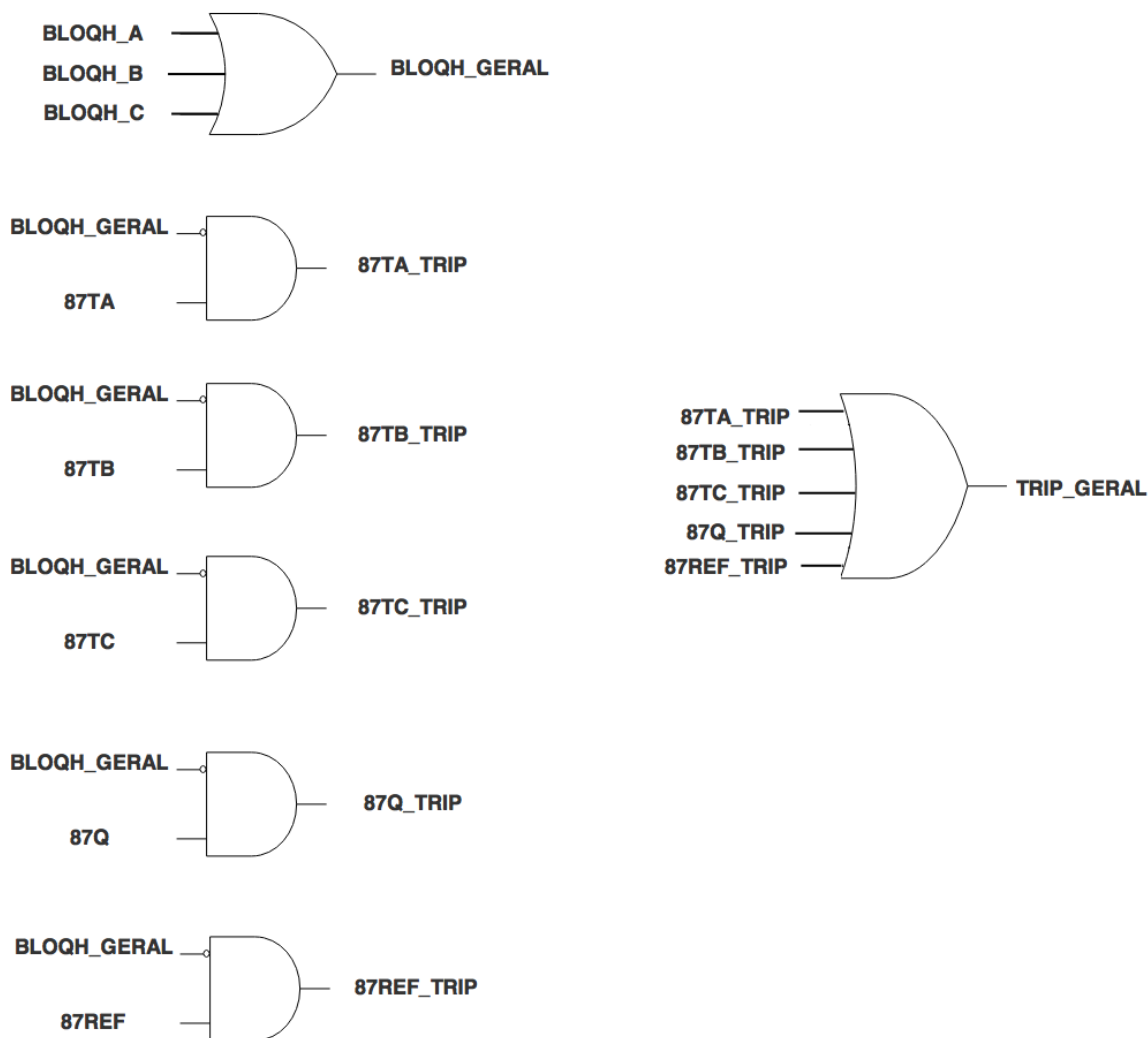


Figura 4.29 - Lógica de TRIP do Relé Desenvolvido no FPGA.

Observa-se que foi adotada a filosofia de bloqueio cruzado no *hardware* desenvolvido.

A **Tabela 4.19** apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo BLOQH.

Tabela 4.19 – Uso dos Recursos do FPGA pelo módulo TRIP87.

Ciclos de <i>clock</i> por operação	0
Total de Elementos Lógicos Utilizados	8
Percentual de uso do FPGA adotado	0,08%

4.2.15 Módulos de Extração de Dados

O módulo de extração de dados sintetizado no FPGA tem como objetivo acessar diretamente a memória RAM_2 e disponibilizar nos pinos do FPGA os valores armazenados para que sejam copiados em forma sequencial e analisados por um dispositivo externo.

A **Figura 4.30** apresenta o diagrama esquemático simplificado do módulo de extração dos dados do FPGA e suas interfaces. A **Tabela 4.20** apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo.

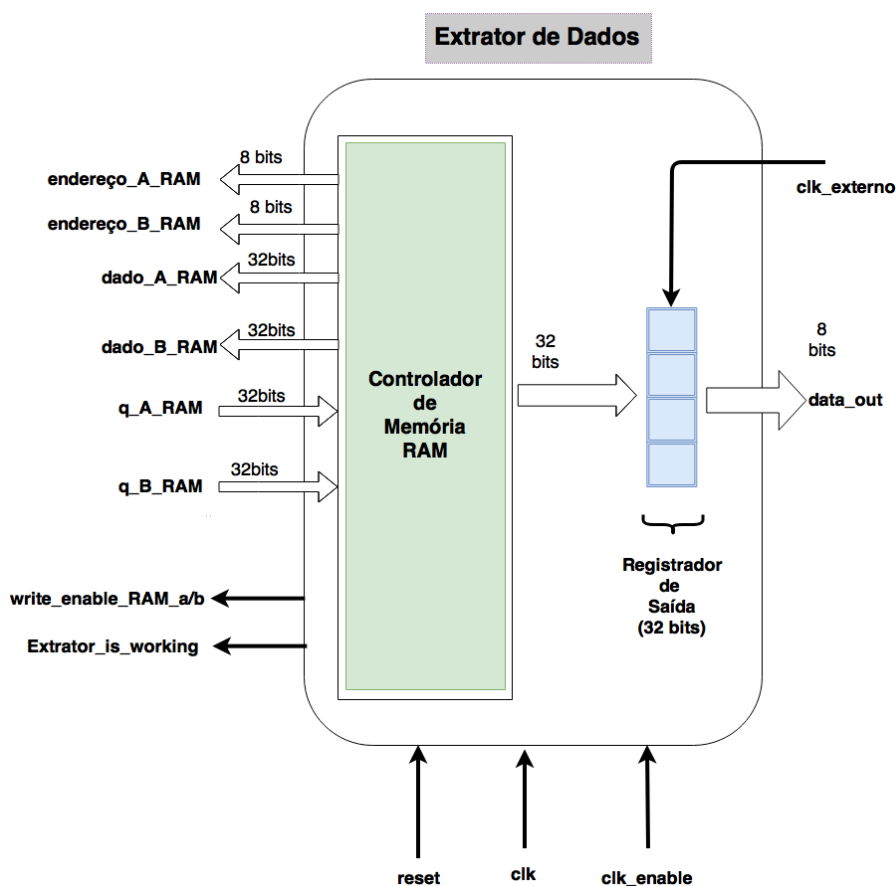


Figura 4.30 - Diagrama Esquemático do Módulo Extrator de Dados.

Tabela 4.20 - Uso dos Recursos do FPGA pelo Módulo de Extração de Dados da RAM_2.

Operandos de 32 bits	4
Ciclos de <i>clock</i> por operação	6
Total de Elementos Lógicos Utilizados	64
Percentual de uso do FPGA adotado	0,62%

4.2.16 Módulo Maestro

Uma vez que o sistema desenvolvido adotou a topologia de compartilhamento de recursos aritméticos e de memória para implementar as funções de proteção diferencial de transformadores, fez-se necessário desenvolver um módulo cuja função é sincronizar o uso dos recursos, bem como garantir que todas as funções de proteção diferencial apresentem resultados em menos de 52.083 ciclos de *clock*. A este módulo atribui-se o nome Maestro.

A **Figura 4.31** apresenta o diagrama esquemático simplificado do Maestro com todas suas interfaces indicadas. Uma vez que o Maestro se comunica diretamente com quase todos elementos do sistema, o número de interfaces chega a 781 bits de conexão e controle.

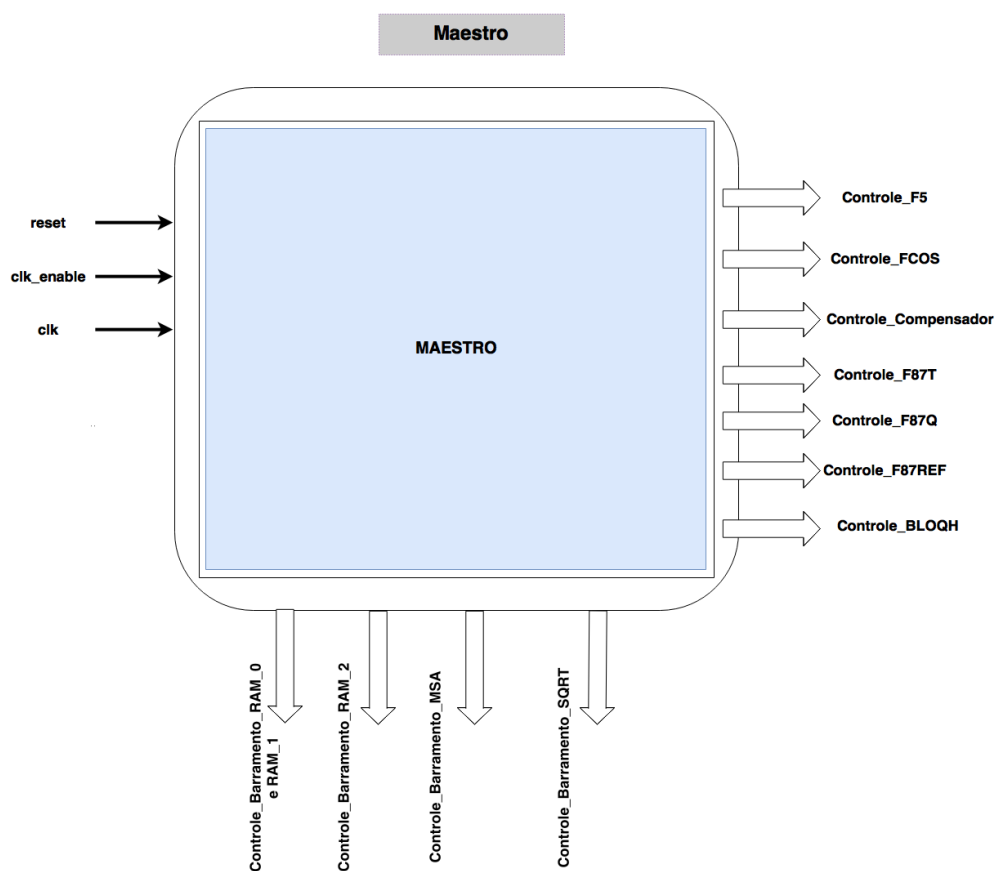


Figura 4.31 - Diagrama Esquemático Simplificado do Módulo Maestro.

No cerne do Maestro, está uma máquina de estados finitos que decide quando cada módulo controlado será ativado e lhes dispõe os recursos aritméticos e de memória para a execução de suas atividades específicas.

Cada módulo é controlado basicamente pelas portas *reset* e *clk_enable* que permitem, respectivamente, reiniciar o módulo controlado e suspender a execução da atividade executada. Normalmente, quando o Maestro inicia a atividade de um módulo, é aguardada a sinalização de tarefa concluída por aquele módulo controlado. A **Figura 4.32** apresenta o diagrama de tempo do controle efetuado sobre o módulo de estimação de fasores, o FCOS, a título de exemplo.

O controle de acesso dos módulos às memórias e aos módulos aritméticos MSA e SQRT é realizado por meio das linhas seletoras (de 3 bits) dos 4 (quatro) barramentos disponíveis no sistema. Os valores apresentados nos seletores dos barramentos são padronizados e únicos para cada módulo, conforme indicado na **Tabela 4.21**.

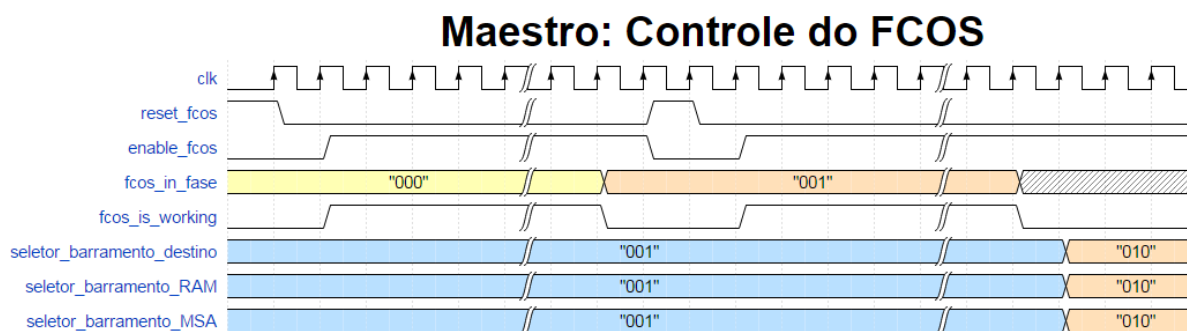


Figura 4.32 - Diagrama de Tempo do Módulo Maestro Controlando o Módulo FCOS.

Tabela 4.21 - Seletor de Barramento Adotado para Cada Módulo.

Módulo	Seletor do Barramento
F5	“000” (0x00)
FCOS	“001” (0x01)
Extração de Dados	“010” (0x02)
Compensador	“011” (0x03)
F87T	“100” (0x04)
F87Q	“101” (0x05)
F87REF	“110” (0x06)
BLOQH	“111” (0x07)

O módulo Maestro também é responsável por fornecer aos módulos F87T, F87Q, F87REF e BLOQH os valores dos fasores de corrente calculados pelos módulos FCOS e Compensador. Para a consecução de tal finalidade, internamente ao Maestro, as partes reais e imaginárias dos fasores de corrente são armazenadas em registradores de 32 bits e convenientemente apresentados aos módulos controlados.

A **Tabela 4.22** apresenta resumo do uso dos recursos do FPGA, após a sintetização do módulo.

Tabela 4.22 - Uso dos Recursos do FPGA pelo Módulo Maestro.

Ciclos de <i>clock</i> por operação	8043
Total de Elementos Lógicos Utilizados	648
Percentual de uso do FPGA adotado	6,30%

4.2.17 Sistema Completo

A **Figura 4.33** apresenta o diagrama do SOC desenvolvido dentro das especificações estabelecidas e das limitações de recursos (elementos lógicos) disponíveis no FPGA. A sintetização no FPGA de todos módulos que integram o *Hardware* do Relé de Proteção Diferencial de Transformadores consumiu 94,65% dos Elementos Lógicos disponíveis, conforme resumido na **Tabela 4.23**.

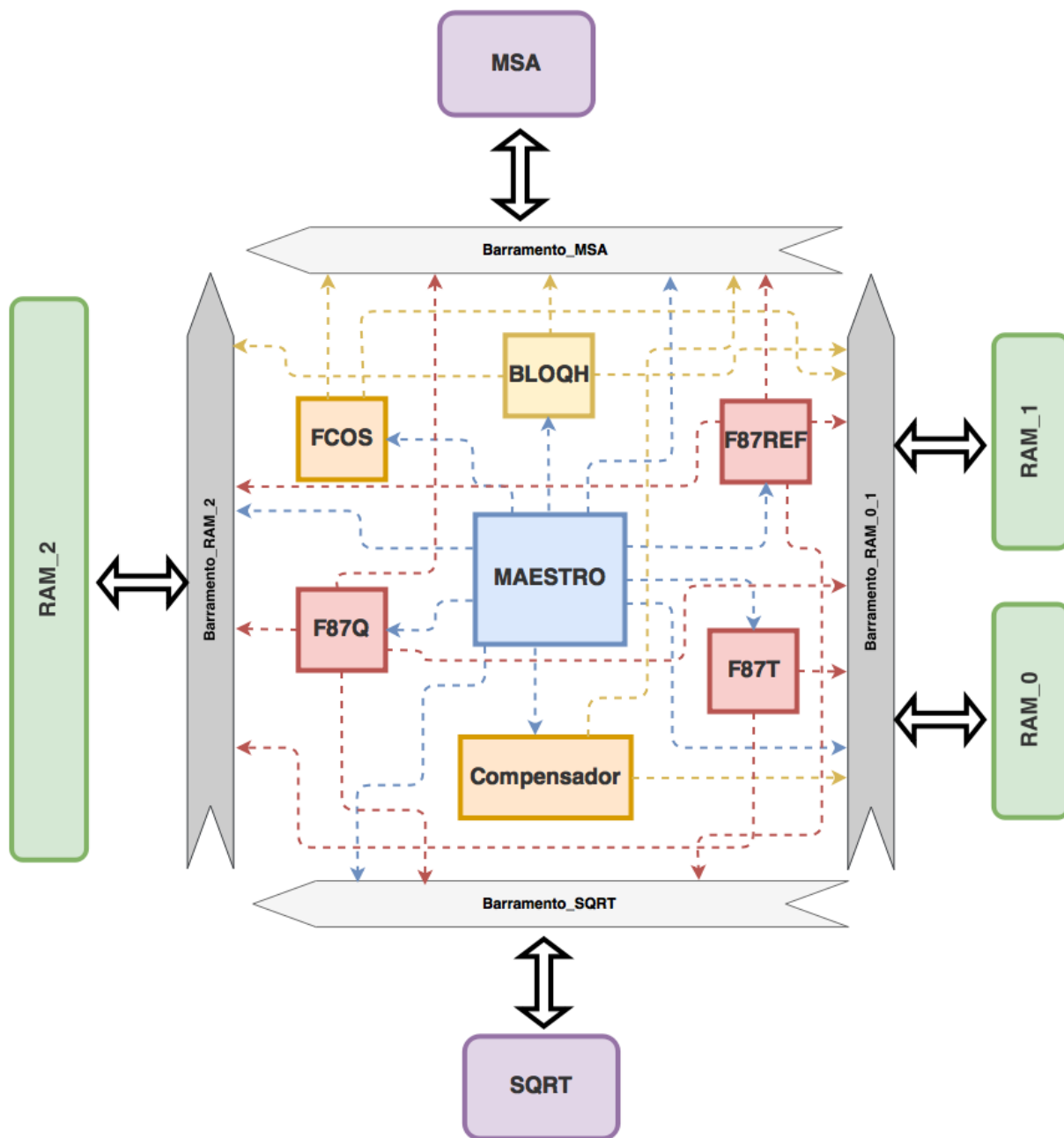


Figura 4.33 - Diagrama de Interconexões dos Módulos Desenvolvidos, no SOC do Relé de Proteção Diferencial de Transformadores.

Tabela 4.23 - Resumo do Uso do FPGA pelo Sistema Desenvolvido.

Módulo	Elementos Lógicos	Multiplicador 18x18	Multiplicador 9x9	Memória (em bits)
Maestro	648	0	0	0
Extrator de Dados	64	0	0	0
Trips	8	0	0	0
BLOQH	642	0	0	0
F87REF	669	0	0	0
F87Q	770	0	0	0
F87T	2.928	3	1	157
Compensador	287	0	0	0
FCOS	179	0	0	0
F5	182	0	0	0
Barramento_SQRT	34	0	0	0
Barramento_MSA	793	0	0	0
Barramento_RAM_0_1	470	0	0	0
Barramento_RAM_2	210	0	0	0
RAM_0	0	0	0	8.192
RAM_1	0	0	0	8.192
RAM_2	0	0	0	397.280
MSA	1.061	3	1	36
SQRT	823	0	0	121
<u>Total</u>	<u>9.768</u>	<u>6</u>	<u>2</u>	<u>413.978</u>
<i>Percentual</i>	<i>94,65%</i>	<i>26,09%</i>	<i>4,35%</i>	<i>99,99%</i>

Observa-se que, dentro da filosofia de compartilhamento de recursos, o SOC desenvolvido é comandado pela máquina de estados implementada no módulo Maestro. Conforme já indicado, o tempo necessário para o processamento de uma amostra de corrente de cada um dos canais (3 fases primárias, 3 fases secundárias e 1 neutro) é de 8.043 ciclos de *clock*, o que corresponde a apenas 15,44% do tempo disponível para a tomada de decisão entre as atualizações das amostras de corrente.

Em termos de qualidade dos resultados obtidos por cada um dos módulos desenvolvidos, deve ser ressaltado que no projeto do sistema no FPGA, tomou-se como referência o código C++ implementado no Raspberry Pi. Desta forma, os resultados numéricos obtidos em ambas plataformas

foram equivalentes, conforme exemplificado pela **Tabela 4.24**, na qual pode ser verificado que os valores obtidos nas duas plataformas diferem a partir da quinta casa decimal.

Tabela 4.24 - Comparação de Resultados Obtidos pela Estimação Fatorial Executada pelo Raspberry Pi e pelo *Hardware* Desenvolvido pelo FPGA.

la_Raspberry_pi	la_FPGA	lb_Raspberry_pi	lb_FPGA	lc_Raspberry_pi	lc_FPGA
-1.1268+0.30432i	-1.1268+0.30429i	0.29246-1.1303i	0.29246-1.1303i	0.83406+0.82631i	0.83407+0.82629i
-0.91939+0.72483i	-0.91934+0.72486i	-0.16348-1.1589i	-0.16343-1.1589i	1.0827+0.43428i	1.0827+0.43431i
-0.57413+1.0164i	-0.57411+1.0163i	-0.59342-1.0054i	-0.59341-1.0055i	1.1674-0.010669i	1.1675-0.010757i
-0.14118+1.1594i	-0.14122+1.1595i	-0.9331-0.702i	-0.93314-0.70198i	1.0743-0.45713i	1.0742-0.4571i
0.31327+1.1252i	0.31323+1.1253i	-1.1306-0.2912i	-1.1306-0.29113i	0.81746-0.83371i	0.81742-0.83363i
0.72+0.91962i	0.71997+0.91956i	-1.1559+0.16389i	-1.1559+0.16382i	0.43611-1.0833i	0.43608-1.0833i
1.017+0.57388i	1.017+0.57394i	-1.0051+0.59398i	-1.0051+0.59405i	-0.011677-1.1678i	-0.011696-1.1677i
1.1591+0.1407i	1.1591+0.14078i	-0.70116+0.9336i	-0.70118+0.93369i	-0.45769-1.0744i	-0.45772-1.0744i
1.1246-0.31386i	1.1247-0.31384i	-0.29044+1.131i	-0.29034+1.131i	-0.83398-0.81741i	-0.83388-0.81739i
0.91882-0.72052i	0.91891-0.72044i	0.16452+1.1562i	0.16462+1.1562i	-1.0832-0.43578i	-1.0831-0.4357i
0.57306-1.0175i	0.57307-1.0175i	0.59443+1.0052i	0.59444+1.0052i	-1.1674+0.01219i	-1.1674+0.012229i
0.14-1.1595i	0.14004-1.1595i	0.93379+0.70104i	0.93383+0.70101i	-1.0738+0.45827i	-1.0737+0.45825i
-0.3144-1.1248i	-0.31437-1.1248i	1.1309+0.29011i	1.1309+0.29014i	-0.81658+0.83448i	-0.81655+0.83451i
-0.7209-0.91885i	-0.72093-0.91884i	1.1557-0.16501i	1.1557-0.165i	-0.43499+1.0836i	-0.43503+1.0837i
-1.0176-0.57288i	-1.0177-0.57283i	1.0045-0.59498i	1.0044-0.59494i	0.012858+1.1677i	0.012786+1.1678i
-1.1593-0.13964i	-1.1592-0.13971i	0.70024-0.93433i	0.70032-0.9344i	0.45876+1.0739i	0.45884+1.0739i
-1.1244+0.31489i	-1.1244+0.31496i	0.28931-1.1313i	0.2893-1.1313i	0.83477+0.81653i	0.83475+0.8166i
-0.91815+0.72147i	-0.91816+0.72145i	-0.16569-1.156i	-0.1657-1.156i	1.0836+0.43475i	1.0836+0.43472i
-0.57207+1.0181i	-0.57214+1.0182i	-0.59545-1.0046i	-0.59551-1.0045i	1.1674-0.013283i	1.1673-0.013213i
-0.13885+1.1597i	-0.13876+1.1596i	-0.93451-0.70011i	-0.93442-0.70016i	1.0734-0.45929i	1.0734-0.45934i
0.31553+1.1246i	0.31546+1.1246i	-1.1312-0.28893i	-1.1313-0.28896i	0.81578-0.83534i	0.81571-0.83538i
0.72185+0.91817i	0.72193+0.91818i	-1.1555+0.16621i	-1.1555+0.16622i	0.43393-1.0841i	0.43401-1.0841i
1.0182+0.57184i	1.0182+0.57182i	-1.0039+0.59601i	-1.0039+0.59599i	-0.014026-1.1678i	-0.014048-1.1678i
1.1594+0.13838i	1.1594+0.13838i	-0.69928+0.93502i	-0.69929+0.93502i	-0.45985-1.0735i	-0.45985-1.0735i
1.124-0.31611i	1.124-0.31606i	-0.28817+1.1316i	-0.28813+1.1317i	-0.83562-0.81572i	-0.83558-0.81568i
0.91738-0.72237i	0.91747-0.72239i	0.16685+1.1558i	0.16695+1.1558i	-1.0841-0.4336i	-1.084-0.43363i
0.57102-1.0187i	0.57107-1.0187i	0.59645+1.004i	0.59651+1.0039i	-1.1674+0.014529i	-1.1673+0.014475i
0.13767-1.1598i	0.1376-1.1598i	0.9352+0.69916i	0.93513+0.69916i	-1.0728+0.46042i	-1.0729+0.46042i
-0.31666-1.1242i	-0.31672-1.1243i	1.1315+0.28783i	1.1314+0.28779i	-0.8149+0.83613i	-0.81496+0.83609i
-0.72275-0.9174i	-0.72269-0.91737i	1.1554-0.16734i	1.1554-0.16731i	-0.43282+1.0845i	-0.43276+1.0846i
-1.0187-0.57083i	-1.0188-0.57091i	1.0033-0.59701i	1.0032-0.59709i	0.015205+1.1677i	0.015156+1.1676i
-1.1595-0.1373i	-1.1595-0.13732i	0.69836-0.93574i	0.69844-0.93576i	0.46092+1.073i	0.461+1.073i
-1.1237+0.31716i	-1.1238+0.31721i	0.28704-1.1319i	0.28699-1.1319i	0.83641+0.81485i	0.83637+0.8149i
-0.91669+0.72331i	-0.91676+0.72323i	-0.16801-1.1557i	-0.16808-1.1558i	1.0845+0.43256i	1.0844+0.43247i
-0.57002+1.0193i	-0.57+1.0192i	-0.59747-1.0034i	-0.59745-1.0034i	1.1674-0.01564i	1.1674-0.015704i
-0.13651+1.16i	-0.13642+1.16i	-0.93591-0.69823i	-0.93582-0.69822i	1.0724-0.46145i	1.0725-0.46144i
0.31779+1.1239i	0.31774+1.124i	-1.1318-0.28666i	-1.1318-0.28663i	0.81409-0.83698i	0.81404-0.83695i
0.72369+0.91671i	0.72373+0.91674i	-1.1552+0.16853i	-1.1552+0.16857i	0.43174-1.085i	0.43179-1.085i
1.0193+0.56979i	1.0193+0.56979i	-1.0027+0.59802i	-1.0027+0.59803i	-0.016374-1.1677i	-0.016367-1.1677i
1.1597+0.13606i	1.1596+0.13607i	-0.6974+0.93643i	-0.69744+0.93644i	-0.462-1.0726i	-0.46204-1.0726i
1.1234-0.31836i	1.1233-0.31839i	-0.28589+1.1322i	-0.28596+1.1322i	-0.83725-0.81404i	-0.83732-0.81407i
0.91593-0.72422i	0.91597-0.72431i	0.16918+1.1555i	0.16922+1.1554i	-1.0849-0.43143i	-1.0849-0.43151i
0.56897-1.0198i	0.56892-1.0199i	0.59847+1.0028i	0.59842+1.0027i	-1.1673+0.016869i	-1.1674+0.01677i
0.13534-1.1601i	0.13531-1.16i	0.93661+0.69728i	0.93658+0.69732i	-1.0719+0.46257i	-1.0719+0.46261i
-0.31892-1.1236i	-0.31894-1.1235i	1.1321+0.28555i	1.132+0.28559i	-0.81322+0.83777i	-0.81325+0.83781i
-0.72459-0.91595i	-0.72463-0.91601i	1.155-0.16967i	1.155-0.16973i	-0.43064+1.0854i	-0.43067+1.0853i
-1.0199-0.56878i	-1.0198-0.56887i	1.0021-0.59903i	1.0021-0.59912i	0.017552+1.1677i	0.017608+1.1676i

la_Raspberry_pi	la_FPGA	lb_Raspberry_pi	lb_FPGA	lc_Raspberry_pi	lc_FPGA
-1.1598-0.13495i	-1.1598-0.13489i	0.69647-0.93714i	0.69649-0.93708i	0.46307+1.0721i	0.46309+1.0721i
-1.1231+0.31943i	-1.1231+0.31944i	0.28476-1.1325i	0.28476-1.1325i	0.83805+0.81317i	0.83805+0.81318i
-0.91523+0.72515i	-0.91526+0.72509i	-0.17034-1.1553i	-0.17037-1.1554i	1.0854+0.43037i	1.0853+0.4303i
-0.56797+1.0204i	-0.56802+1.0203i	-0.59949-1.0022i	-0.59954-1.0023i	1.1673-0.017995i	1.1673-0.018083i
-0.13418+1.1602i	-0.13417+1.1602i	-0.93732-0.69634i	-0.9373-0.69637i	1.0715-0.46361i	1.0715-0.46364i
0.32005+1.1233i	0.32007+1.1233i	-1.1323-0.28438i	-1.1323-0.2844i	0.8124-0.83861i	0.81242-0.83862i
0.72553+0.91525i	0.7256+0.91526i	-1.1549+0.17086i	-1.1548+0.17087i	0.42956-1.0859i	0.42963-1.0858i
1.0205+0.56774i	1.0205+0.56767i	-1.0015+0.60004i	-1.0015+0.59997i	-0.018722-1.1677i	-0.018711-1.1678i
1.1599+0.13373i	1.16+0.13371i	-0.69552+0.93783i	-0.69545+0.93781i	-0.46415-1.0716i	-0.46409-1.0717i
1.1227-0.32061i	1.1227-0.32063i	-0.28362+1.1328i	-0.28364+1.1327i	-0.83888-0.81235i	-0.8389-0.81237i
0.91447-0.72606i	0.91456-0.726i	0.1715+1.1552i	0.17158+1.1552i	-1.0858-0.42924i	-1.0857-0.42919i
0.56693-1.021i	0.56694-1.021i	0.60049+1.0016i	0.60051+1.0015i	-1.1673+0.019211i	-1.1673+0.019173i
0.13301-1.1603i	0.13308-1.1604i	0.93801+0.6954i	0.93808+0.6953i	-1.071+0.46472i	-1.0709+0.46463i
-0.32102-1.1226i	-0.32103-1.1227i	1.1326+0.28309i	1.1326+0.28301i	-0.81161+0.83922i	-0.81161+0.83914i
-0.71817-0.89495i	-0.7181-0.89499i	1.1506-0.18169i	1.1507-0.18174i	-0.43258+1.0765i	-0.43251+1.0765i
-0.99391-0.52284i	-0.99393-0.52278i	0.98732-0.62309i	0.9873-0.62303i	0.0062559+1.1455i	0.0062351+1.1455i
-1.1072-0.075834i	-1.1072-0.075791i	0.66773-0.96794i	0.66775-0.9679i	0.4383+1.0418i	0.43832+1.0418i
-1.0428+0.37576i	-1.0427+0.3758i	0.24155-1.1617i	0.24162-1.1617i	0.79871+0.78292i	0.79878+0.78296i
-0.813+0.7622i	-0.81305+0.76216i	-0.22464-1.1735i	-0.2247-1.1736i	1.035+0.41162i	1.035+0.41158i
-0.45618+1.0232i	-0.45619+1.0232i	-0.65773-1.0009i	-0.65774-1.0009i	1.1134-0.016745i	1.1133-0.016747i
-0.0319+1.115i	-0.031936+1.115i	-0.98845-0.66761i	-0.98848-0.66762i	1.0236-0.43819i	1.0235-0.43821i
0.39018+1.0253i	0.39015+1.0253i	-1.1642-0.2277i	-1.1642-0.22769i	0.77982-0.79211i	0.77978-0.7921i
0.74186+0.77142i	0.74191+0.77143i	-1.1579+0.24671i	-1.1579+0.24672i	0.42135-1.0205i	0.42139-1.0205i

APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Uma vez implementadas e testadas no FPGA e no Raspberry Pi as funções de proteção diferenciais de fase (87T), de sequência negativa (87Q) e de faltas à terra restrita (87REF), apresentam-se neste capítulo alguns resultados obtidos com base em casos simulados no ATP (*Alternative Transients Program*), que é um software especializado em simulação digital de fenômenos transitórios de natureza eletromagnética em sistemas elétricos de potência.

5.1. PLATAFORMA DE TESTES

A metodologia adotada para efetuar os testes consistiu em injetar as amostras dos sinais de corrente no FPGA, de modo que em tempo real o *hardware* desenvolvido efetuasse os cálculos necessários e tomasse as decisões referentes às sinalizações de TRIP e bloqueio.

O passo a passo dos testes resume-se nos seguintes pontos:

- a) Simular o caso no ATP.
 - i. Nesta etapa, através da biblioteca MODELS desenvolvida em [24], são salvos em um arquivo de texto, a uma taxa de 16 (dezesesseis) amostras por ciclo, os valores de corrente do secundário dos TCs.
- b) Conversão do arquivo texto no formato ponto flutuante IEEE 754.
 - i. No âmbito deste projeto, desenvolveu-se uma rotina C++ para conversão dos valores das amostras de corrente para o formato IEEE 754, adotado no *hardware* desenvolvido.
 - ii. Especificamente, essa rotina salva cada amostra de corrente em 4 (quatro) bytes sequenciais na seguinte ordem: 1) corrente da fase A primária; 2) corrente da fase B primária; 3) corrente da fase C primária; 4) corrente de neutro; 5) corrente da fase A secundária; 6) corrente da fase B secundária; e 7) corrente da fase C secundária.

- c) Injeção das amostras de corrente no sistema desenvolvido para o FPGA.
 - i. Adotou-se um processador ATMEL baseado na plataforma *Arduino* para a leitura de um cartão de memória SD, contendo os arquivos, no formato ponto flutuante IEEE 754, com as amostras de corrente geradas pelas simulações.
 - ii. A leitura destes arquivos é realizada byte a byte, os quais são disponibilizados nas 8 portas de saída do *arduino* conectadas à porta de entrada do sistema sintetizado no FPGA.
- d) Extração dos dados de análise do sistema sintetizado no FPGA.
 - i. Outro *Arduino* é conectado à porta de saída do relé implantado no FPGA para extração dos dados produzidos pelos módulos de proteção 87T, 87Q, 87REF e Bloqueio, salvos na memória RAM_2.
 - ii. Um arquivo para cada caso simulado é salvo em um cartão de memória SD para posterior análise.
 - iii. A cada caso tratado pelo sistema sintetizado no FPGA, aplica-se o comando de *reset* no relé para apagar os dados salvos na memória RAM_2 e prepará-lo para tratar um novo caso.

5.2. SISTEMA ANALISADO

Propõe-se o sistema elétrico indicado na **Figura 5.1** para análise dos casos e observação da resposta do relé sintetizado no FPGA. Esse sistema é composto um transformador 230/69 kV, arranjo estrela aterrada no lado primário (230 kV) e delta no lado secundário (69 kV). Os sistemas de potência conectados às barras de 69 kV e de 230 kV são representados por fontes de tensão e suas respectivas impedâncias de Thévenin. A **Tabela 5.1** apresenta os valores das impedâncias equivalentes de sequência zero e positiva adotados no sistema elétrico simulado.

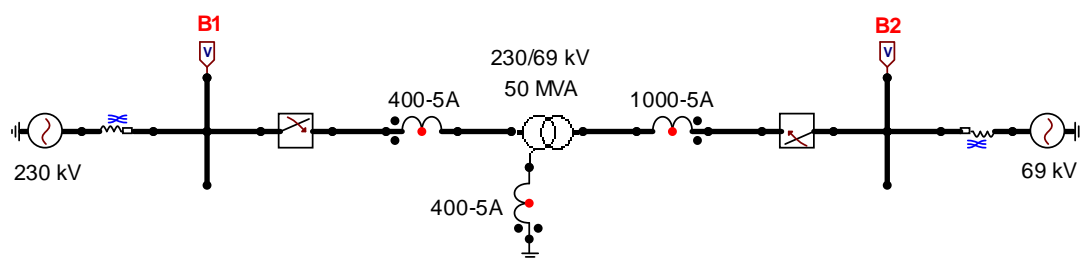


Figura 5.1 - Sistema Simulado no ATP Utilizado para Analisar o Desempenho do Relé de Proteção Desenvolvido no FPGA.

Tabela 5.1 - Impedâncias Equivalentes do Sistema Simulado no ATP.

	Barra de 230 kV	Barra de 69 kV
Z_0	$3 + j 27,90 \Omega$	$3,635 + j 35,707 \Omega$
Z_1	$9 + j 46,92 \Omega$	$3,635 + j 35,707 \Omega$

Os TCs utilizados no sistema elétrico simulado são do tipo C400 400-5 A, no lado de 69 kV e no neutro do transformador, e C400 1000-5 A, no lado de 69 kV, tendo sido modelados conforme proposto pelo IEEE *Power System Relaying Committee* [41].

A modelagem do transformador 230/69 kV no ATP seguiu a mesma linha adotada no trabalho desenvolvido por [24], no qual modelou-se o transformador de potência pelo componente *Saturable Transformer*, sendo que seu ciclo de histerese foi representado pelo elemento 96 do ATP. A **Tabela 5.2** indica os valores da reatância do primário e do secundário adotados nas simulações dos casos.

Tabela 5.2 - Reatâncias dos Enrolamentos do Transformador Simulado.

Enrolamento	Reatância
Primário	12,54 Ω
Secundário	38,035 Ω

Conforme apresentado na **Figura 5.2**, no modelo de transformador adotado, há a possibilidade de seccionar em três segmentos os enrolamentos primário e secundário para que se possam aplicar, nos casos simulados, curtos-circuitos internos de naturezas espira-espira e espira-terra.

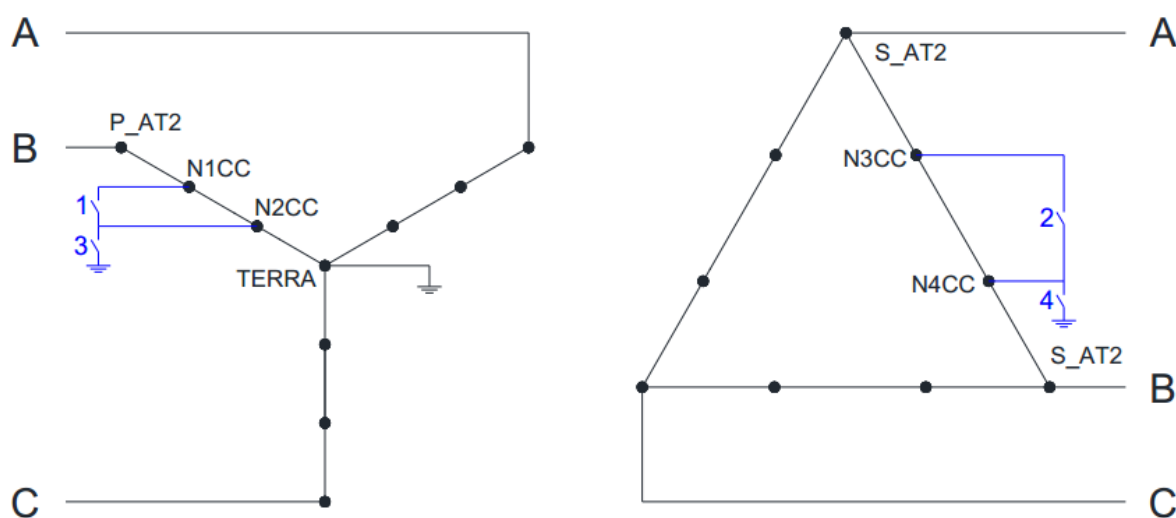


Figura 5.2 - Esquema de Seccionamento dos Enrolamentos de Transformador Simulado.

5.3. CONFIGURAÇÃO DO RELÉ DESENVOLVIDO

Durante a fase de testes do *Hardware* de Proteção Diferencial de Transformadores de Potência, foi utilizada a configuração indicada na **Tabela 5.3**.

Tabela 5.3 - Parametrização Adotada no Relé Desenvolvido para o Sistema Elétrico Analisado.

Parâmetro	Valor	Parâmetro	Valor	Descrição
Fator multiplicativo de corrente primária	0,6374	Fator multiplicativo de corrente secundária	0,2760	Valores de normalização (TAP), calculados com base nas Equações (3.5) e (3.6).
Fator multiplicativo de corrente de neutro			0,6374	Valor de normalização (TAP) da corrente de neutro.
Defasagem Angular Primária	0°	Defasagem Angular Secundária	30°	Parâmetro configurável de acordo com o esquema de ligação do transformador protegido.

Parâmetro	Valor	Parâmetro	Valor	Descrição
Dedução de Corrente de Sequência Zero no Primário	Sim	Dedução de Corrente de Sequência Zero no Secundário	Não	Determina o uso da matriz de supressão da corrente de sequência zero no cálculo da compensação das correntes.
Corrente Pick-Up Função 87T			1,0 pu	Ajuste da corrente de Pick-Up do Elemento 87T.
Corrente Pick-Up Função 87Q	0,02 pu	Corrente Pick-Up Função 87REF	0,2 pu	Ajuste da corrente de Pick-Up dos Elementos 87Q e 87REF.
SLOPE da Função 87T	0,4	SLOPE_INVERSO da Função 87T	2,5	Ajuste do <i>SLOPE</i> do Elemento 87T.
SLOPE da Função 87Q	0,99	SLOPE da Função 87REF	1,0	Ajuste do <i>SLOPE</i> dos Elementos 87Q e 87REF.

Parâmetro	Valor	Parâmetro	Valor	Descrição
Percentual de Restrição da 2ª Harmônica	25%	Percentual de Bloqueio de 2ª Harmônica	25%	Ajuste do percentual de restrição por harmônicos para o elemento 87T e do percentual de bloqueio para o elemento BLOQH.
Kr (Fator de Estabilização da Corrente de Restrição do Módulo 87REF)		2,0		Fator de sensibilidade do elemento 87REF [24]

Os ajustes adotados para as funções de proteção diferencial estão de acordo com aqueles que são tipicamente utilizados em subestações para proteção de transformadores reais. No relé desenvolvido, os ajustes definidos são previamente carregados na memória RAM_0 e podem ser facilmente reajustados em caso de necessidade de adequar os ajustes indicados na **Tabela 5.3**.

5.4. ANÁLISE DOS CASOS

A **Tabela 5.4** apresenta os principais casos simulados (no ATP) e o que foi verificado quanto à atuação ou não das funções implementadas em *hardware*. As faltas monofásicas foram simuladas envolvendo a fase A e a terra. As faltas bifásicas envolveram as fases A e B, não havendo envolvimento da terra. As faltas trifásicas também não envolveram a terra.

Tabela 5.4 - Resumo das Atuações dos Elementos de Proteção nos Casos Analisados.

Caso	87T	87Q	87REF
Regime Permanente	-	-	-
Curto Monofásico Externo na Barra de 230 kV	-	-	-

Caso	87T	87Q	87REF
Curto Monofásico Externo na Barra de 69 kV	-	-	-
Curto Monofásico dentro da Região Protegida, lado Primário	A		
	B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	C		
Curto Monofásico dentro da Região Protegida, lado Secundário	A		
	-	<input checked="" type="checkbox"/>	-
	C		
Curto Bifásico Externo na Barra de 230 kV	-	-	-
Curto Bifásico Externo na Barra de 69 kV	-	-	-
Curto Bifásico Interno na Barra de 230 kV	A		
	B	<input checked="" type="checkbox"/>	-
	-		
Curto Bifásico Interno na Barra de 69 kV	A		
	B	<input checked="" type="checkbox"/>	-
	C		
Curto Trifásico Externo na Barra de 230 kV	-	-	-
Curto Trifásico Externo na Barra de 69 kV			
Curto Trifásico Interno na Barra de 230 kV	A		
	B	<input checked="" type="checkbox"/>	-
	C		
Curto Trifásico Interno na Barra de 69 kV	A		
	B	<input checked="" type="checkbox"/>	-
	C		
Curto interno Espira-Terra, em 0,5% do enrolamento, no lado Estrela	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Caso	87T	87Q	87REF
Curto interno Espira-Terra, em 1% do enrolamento, no lado Estrela	A - -	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Curto interno Espira-Terra, em 50% do enrolamento, no lado Estrela	A B C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Curto interno Espira-Espira, em 1% do enrolamento, no lado Estrela	-	<input checked="" type="checkbox"/>	-
Curto interno Espira-Espira, em 50% do enrolamento, no lado Estrela	A B C	<input checked="" type="checkbox"/>	-
Curto interno Espira-Terra, em 1% do enrolamento, no lado Delta	A B -	<input checked="" type="checkbox"/>	-
Curto interno Espira-Terra, em 50% do enrolamento, no lado Delta	-	<input checked="" type="checkbox"/>	-
Curto interno Espira-Espira, em 1% do enrolamento, no lado Delta	-	-	-
Curto interno Espira-Espira, em 50% do enrolamento, no lado Delta	A - -	<input checked="" type="checkbox"/>	-
Energização pelo lado de 230 kV	-	<input checked="" type="checkbox"/>	-
Energização pelo lado de 69 kV	-	<input checked="" type="checkbox"/>	-

Ressalta-se que optou-se por aplicar, nos casos simulados no ATP, curtos circuitos francos externos ao transformador, dentro e fora da zona protegida. Ou seja, são nulas as impedâncias das faltas monofásicas, bifásicas e trifásicas relacionadas na **Tabela 5.4**, o que aumenta a probabilidade de sensibilização do relé desenvolvido parametrizado de acordo com a **Tabela 5.3**.

De um modo geral, as funções de proteção diferencial de fase (87TA, 87TB e 87TC) atuaram adequadamente quando ocorreram faltas dentro da região protegida, internas e externas ao transformador.

A função de proteção diferencial de falta restrita à terra (87REF), de acordo com as expectativas, atuou em todas as faltas que envolveram a terra, exceto aquelas ocorridas no lado de 69 kV do transformador, cuja ligação é em delta. Destaca-se a principal característica desse elemento de proteção diferencial que a capacidade de identificar curtos circuitos espira-terra em toda totalidade do enrolamento primário do transformador.

Especial atenção deve ser dedicada ao comportamento do elemento de proteção diferencial de sequência negativa (87Q). Observa-se que sua atuação é bastante sensível aos menores desequilíbrios de corrente entre as fases, o que justifica sua atuação em faltas monofásicas e bifásicas dentro da região protegida, assim como para curto circuitos espira-espira e espira-terra, internos ao transformador. Por esta mesma razão, notou-se atuação indevida do elemento 87Q frente às faltas trifásicas. Justificam-se essas situações indesejadas pela falsa identificação de correntes de sequência negativa pela estimação fasorial durante as primeiras amostras do transitório. Por essa razão, costuma-se incorporar um atraso intencional de um ou dois ciclos na atuação do elemento de proteção diferencial de sequência negativa.

Nos próximos subitens, são discutidos, dentre os casos indicados na **Tabela 5.4**, aqueles cujos comportamentos transitórios apresentam aspectos a serem destacados.

5.4.1 Curto Monofásico Fase-Terra, dentro da Região Protegida, no lado de 230 kV (Lado Y)

Apresentam-se, respectivamente, na **Figura 5.3**, **Figura 5.4**, **Figura 5.5** e **Figura 5.6** as correntes medidas pelos TCs dos lados primário e secundário do transformador, bem como seus fasores estimados resultantes do caso simulado no ATP de uma falta monofásica, fase-terra, ocorrida no lado primário do transformador no instante de 80 ms.

Observa-se na **Figura 5.5** que a estimação de fasores implementada no *hardware* de proteção diferencial é prejudicada, neste caso, pela componente CC de decaimento exponencial embutida na resposta transitória ao curto circuito associada com a saturação do TC da fase A.

Conforme indicam os planos operacionais e gráfico das correntes de operação e restrição na **Figura 5.7**, **Figura 5.8** e **Figura 5.9**, as unidades de proteção diferencial implementadas no FPGA (87TA, 87TB, 87TC, 87TQ e 87REF) atuaram para a falta monofásica dentro da região protegida de 230 kV. A sinalização das variáveis que compõem a lógica de TRIP principal do relé implementado, apresentadas na **Figura 5.10**, indica que o bloqueio por harmônicos foi sensibilizado durante cerca de um intervalo de ciclo durante o transitório provocado pelo curto circuito, o que inibiu a sinalização de TRIP indicada primeiramente pelos elementos 87Q e 87REF. O sinal de TRIP principal, denominado TRIP87, foi acionado 11,7 ms após a ocorrência do curto circuito monofásico.

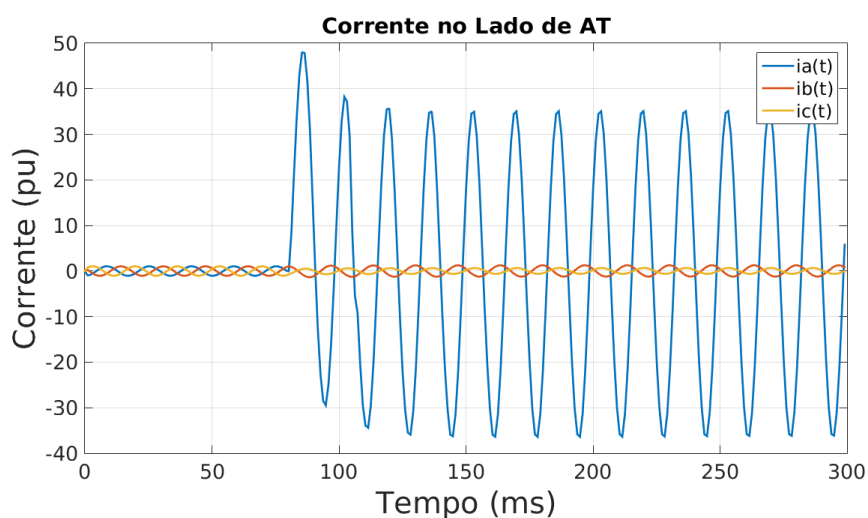


Figura 5.3 - Correntes do Lado de 230 kV do Transformador, Medida pelo TC - Falta Monofásica no Lado Primário.

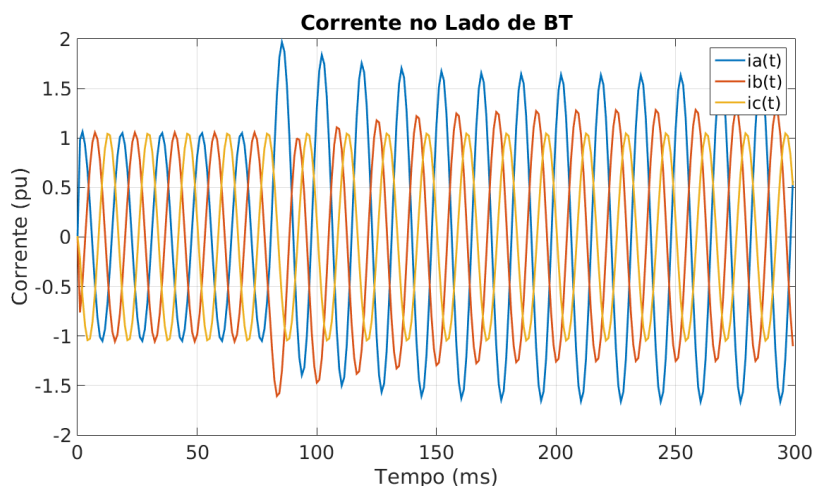


Figura 5.4 - Correntes do Lado de 69 kV do Transformador, Medida pelo TC - Falta Monofásica no Lado Primário.

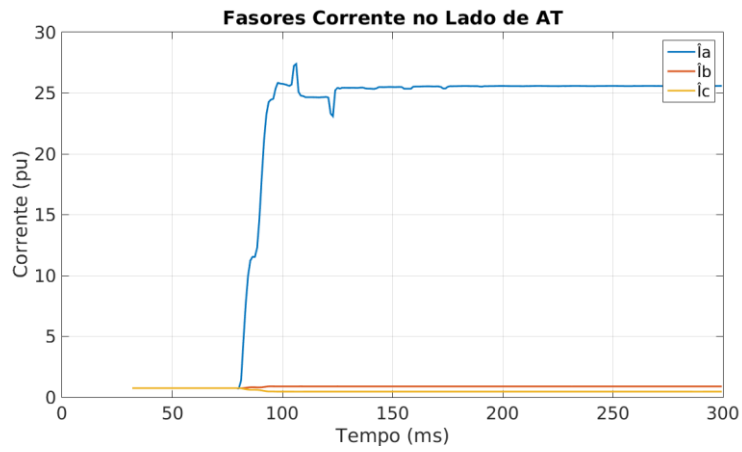


Figura 5.5 - Estimação Fasorial, no lado de 230 kV - Falta Monofásica no Lado Primário.

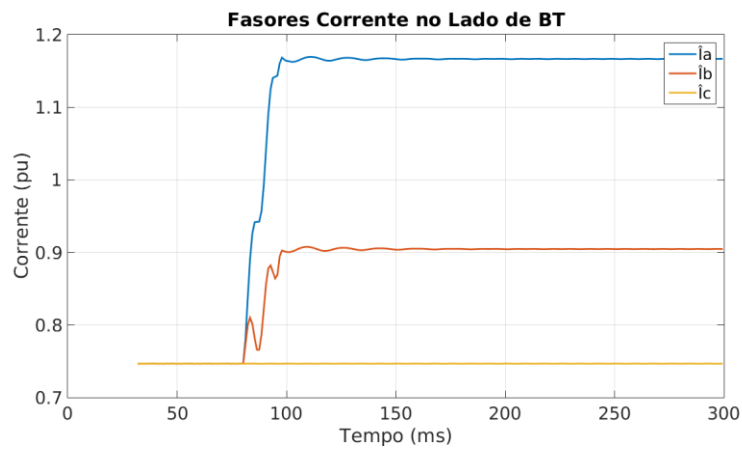


Figura 5.6 - Estimação Fasorial, no lado de 69 kV - Falta Monofásica no Lado Primário.

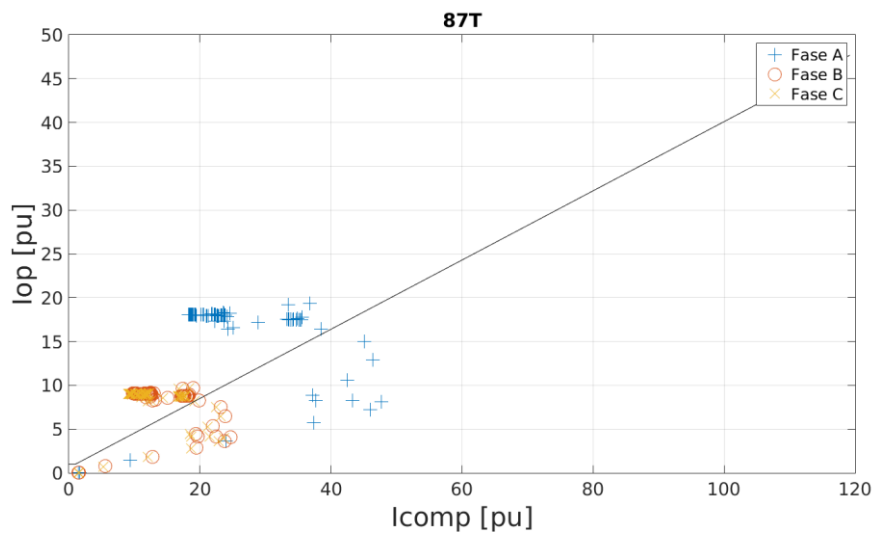


Figura 5.7 - Plano Operacional da Função 87T - Falta Monofásica no Lado Primário.

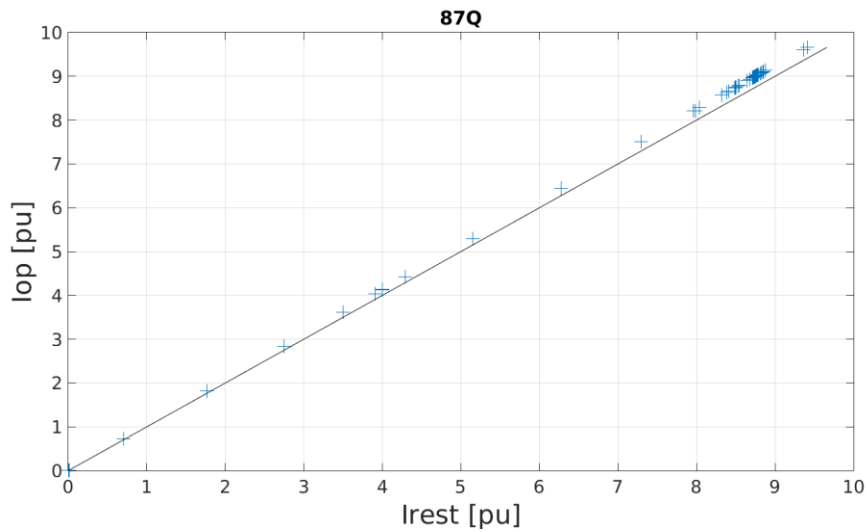


Figura 5.8 - Plano Operacional da Função 87Q - Falta Monofásica no Lado Primário.

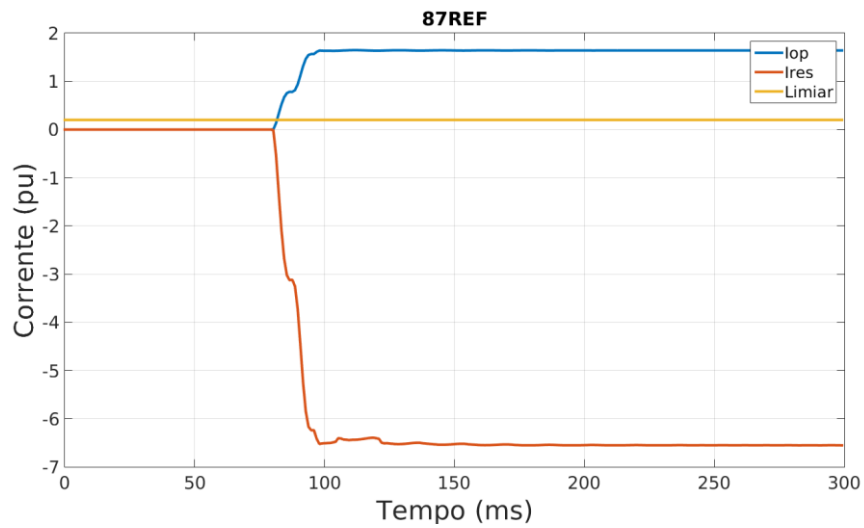


Figura 5.9 - Correntes de Operação e Restrição Calculadas para a Função 87REF - Falta Monofásica no Lado Primário.

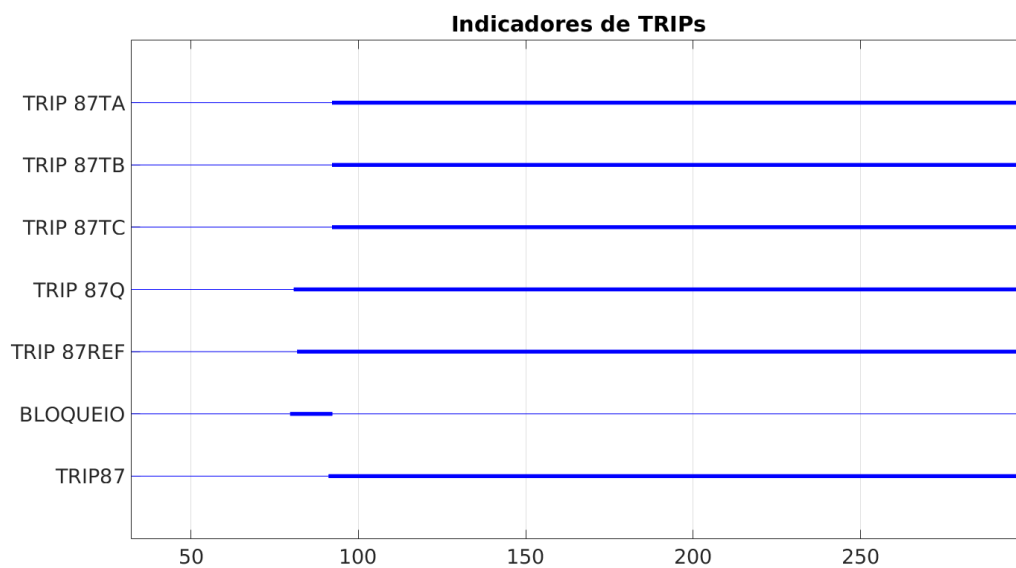


Figura 5.10 - Variáveis que Compõem a Lógica de TRIP - Falta Monofásica no Lado Primário.

5.4.2 Curto Circuito Espira-Espira no Lado de 69 kV (Lado Δ)

Com o intuito de averiguar a detecção de curto circuitos entre espiras pelo relé desenvolvido no FPGA, simulou-se no ATP uma falta espira-espira, ocorrida em $t = 80$ ms, envolvendo 30% do enrolamento delta, no lado de 69 kV do transformador. A **Figura 5.11** e a **Figura 5.12** apresentam as correntes medidas pelos TCs dos lados primário e secundário do transformador.

Observa-se por intermédio da **Figura 5.13**, **Figura 5.14** e **Figura 5.15** o comportamento das correntes de operação e restrição dos elementos 87T, 87Q e 87REF desenvolvidos para o FPGA, no qual verifica-se que apenas o elemento de sequência negativa é sensibilizado durante o curto circuito espira-espira envolvendo 30% do enrolamento de delta do transformador.

Os elementos de fase 87TA, 87TB e 87TC não chegam a ser sensibilizados devido à baixa variação das correntes nos lados primário e secundário do transformador. Esta pequena variação sequer atingiu o valor de *pick-up* (1,0 pu), uma vez que sua magnitude é da ordem das flutuações de carga que ocorrem naturalmente ao longo do dia.

O elemento de falta restrita à terra, 87REF, naturalmente não identifica a falta pelo fato de o curto circuito não envolver a terra e estar do lado delta do transformador.

O comportamento das variáveis lógicas do relé desenvolvido no FPGA, apresentado na **Figura 5.16**, indica que durante as primeiras amostras do transitório o sinal de bloqueio é ativado, impedindo a ativação o TRIP principal do relé, TRIP87, por intermédio do sinal de TRIP enviado pelo módulo 87Q. O módulo BLOQH ativou o bloqueio a partir do instante $t = 82,32$ ms até $t = 86,49$ ms. Já o módulo 87Q também foi sensibilizado a partir do instante $t = 82,32$ ms. Adotou-se no projeto do *hardware* do relé de proteção diferencial de transformadores de potência implementado em FPGA estratégia de acionamento e execução do módulo de bloqueio por harmônicos (BLOQH) anterior ao acionamento dos módulos de proteção diferencial 87T, 87Q e 87REF. Desta forma, sempre em que o *hardware* se deparar com situações como esta, o bloqueio prevalecerá frente aos demais sinais de TRIP concorrentes.

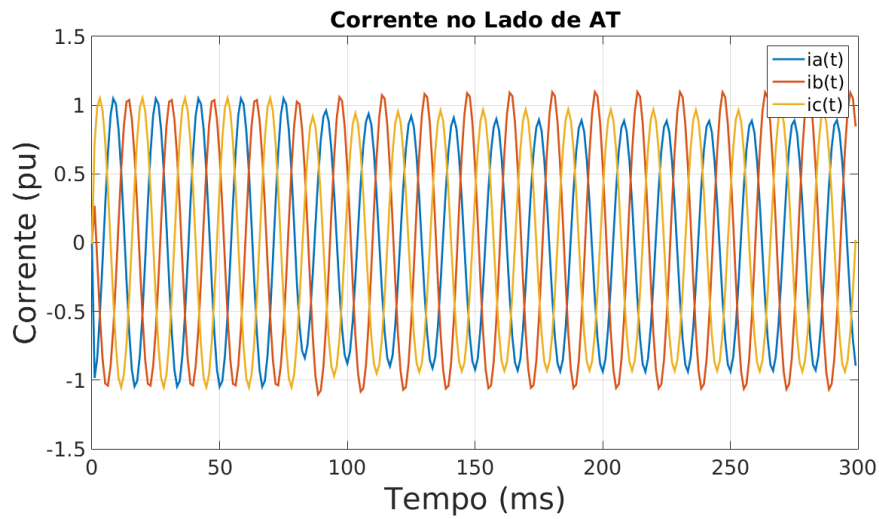


Figura 5.11 - Correntes no Enrolamento Primário, Medidas pelos TCs - Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.

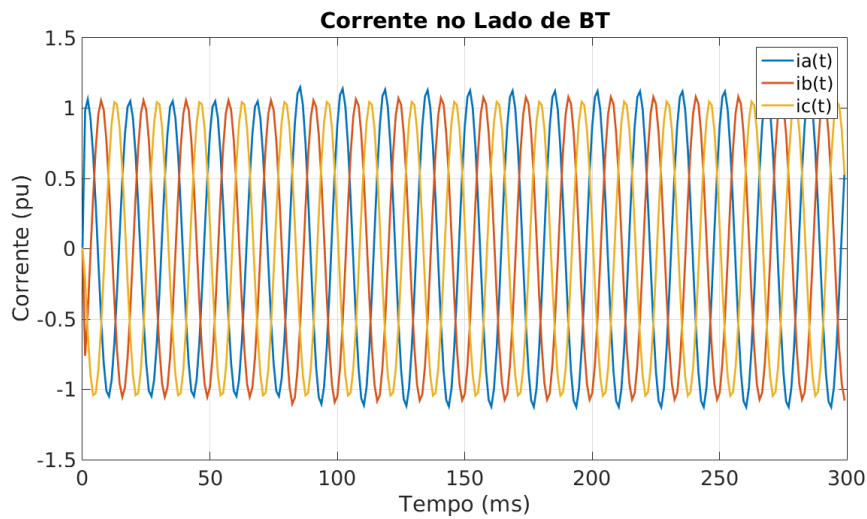


Figura 5.12 - Correntes no Enrolamento Secundário, Medidas pelos TCs - Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.

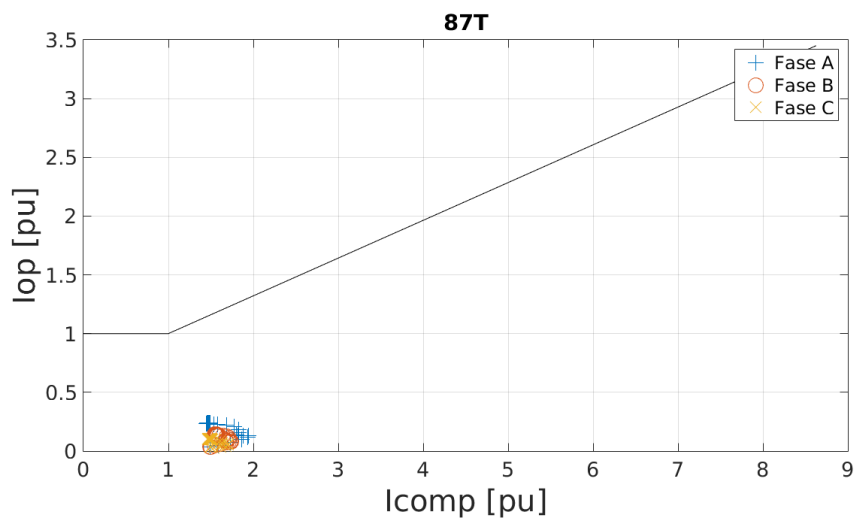


Figura 5.13 - Plano Operacional da Função 87T – Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.

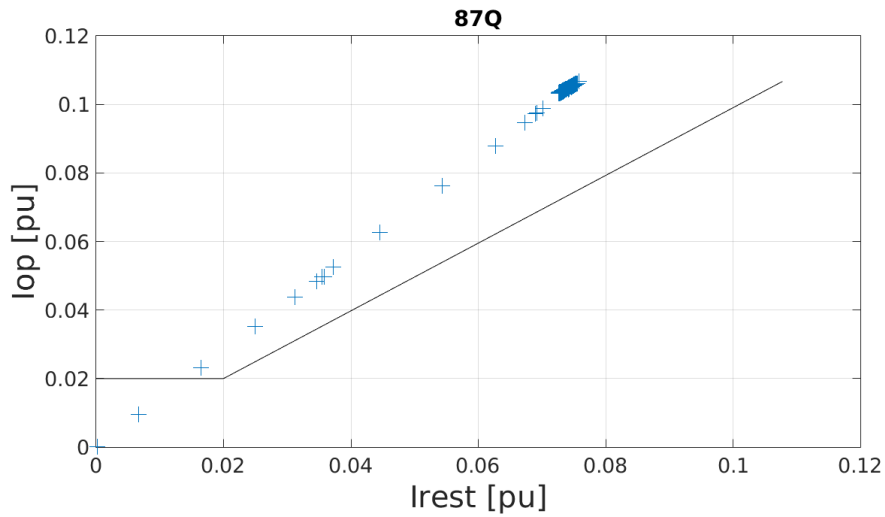


Figura 5.14 - Plano Operacional da Função 87Q – Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.

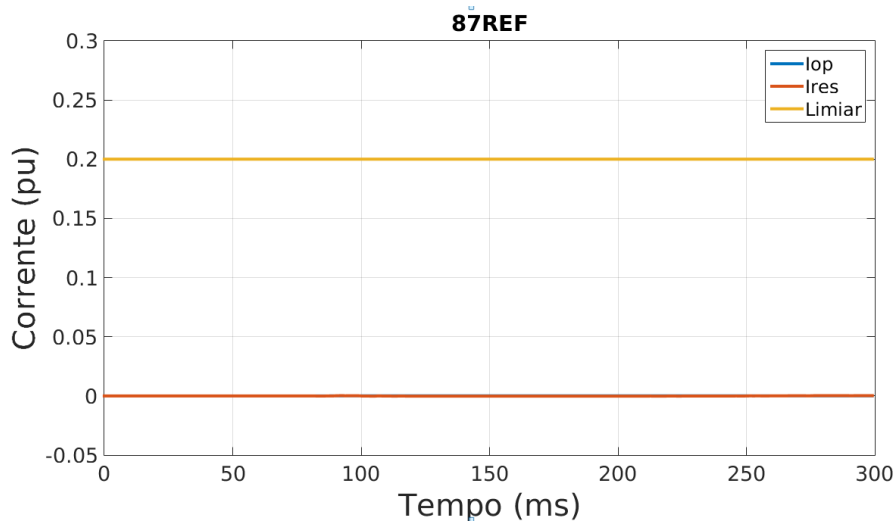


Figura 5.15 - Correntes de Operação e Restrição Calculadas para a Função 87REF - Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.

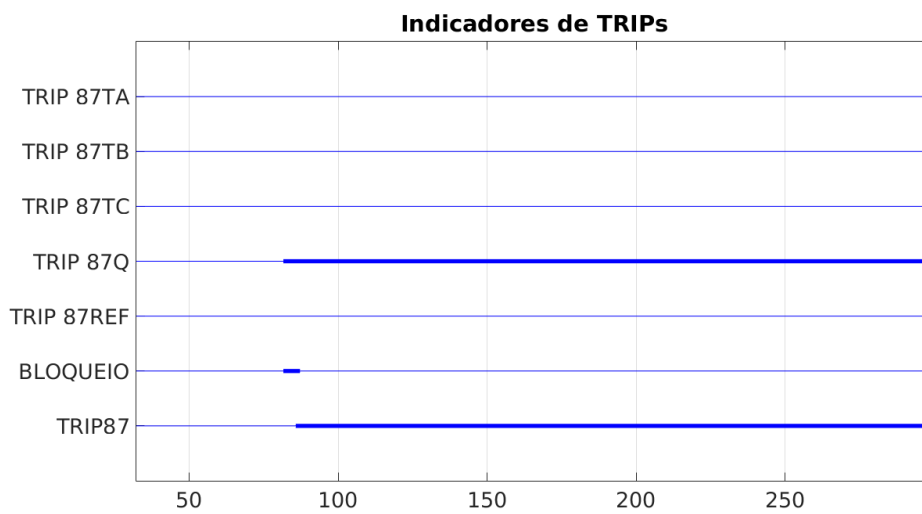


Figura 5.16 - Variáveis que Compõem a Lógica de TRIP - Caso de Curto Circuito Espira-Espira (30%) no lado de 69 kV.

5.4.3 Energização do Transformador

A energização do transformador é uma das situações que merecem atenção no projeto e parametrização dos relés de proteção diferencial. Devido ao fenômeno das correntes de *inrush*, a função de proteção diferencial de sequência negativa, que é mais sensível a desbalanços entre as correntes medidas nas fases do transformador, pode atuar indevidamente.

Para tanto, foi simulado no ATP um caso de energização em 80 ms com o fechamento apenas dos disjuntores instalados no lado de 230 kV do transformador. As correntes medidas pelos TCs no lado primário são apresentadas na **Figura 5.17**, na qual pode ser observado, em conjunto com a **Figura 5.18**, a forte presença de componentes de 2ª harmônica nessas correntes. Conforme pode ser observado na **Figura 5.19**, **Figura 5.20** e **Figura 5.21**, os gráficos que relacionam as correntes de operação e restrição dos elementos de proteção diferencial 87T, 87Q e 87REF indicam que apenas o elemento 87Q foi sensibilizado durante o processo de energização do transformador pelo lado de 230 kV. Contudo, devido à forte presença da componente de 2ª harmônica detectada pelo elemento de bloqueio (BLOQH), implementado no FPGA, o sinal de TRIP principal, TRIP87, corretamente não foi ativado durante a energização.

Um aspecto que merece destaque é o comportamento do elemento de proteção diferencial de fase 87T que foi desenvolvido com a filosofia de restrição de harmônicos discutida por [24]. Em virtude da presença do 2º harmônico nas correntes de *inrush*, observa-se na **Figura 5.19** que os pontos no plano operacional ficam mais dispersos na área de restrição.

A **Figura 5.22** apresenta o comportamento das variáveis lógicas do relé desenvolvido no FPGA. O sinal de bloqueio por harmônicos foi produzido pelo módulo BLOQH em 0,23 ms após o fechamento dos disjuntores do primário do transformador. Já o sinal de TRIP do módulo 87Q ocorreu apenas 4,4 ms após a energização. Desta forma, o sinal de TRIP principal, TRIP87, corretamente ficou inibido durante o processo de energização do transformador pelo lado de 230 kV.

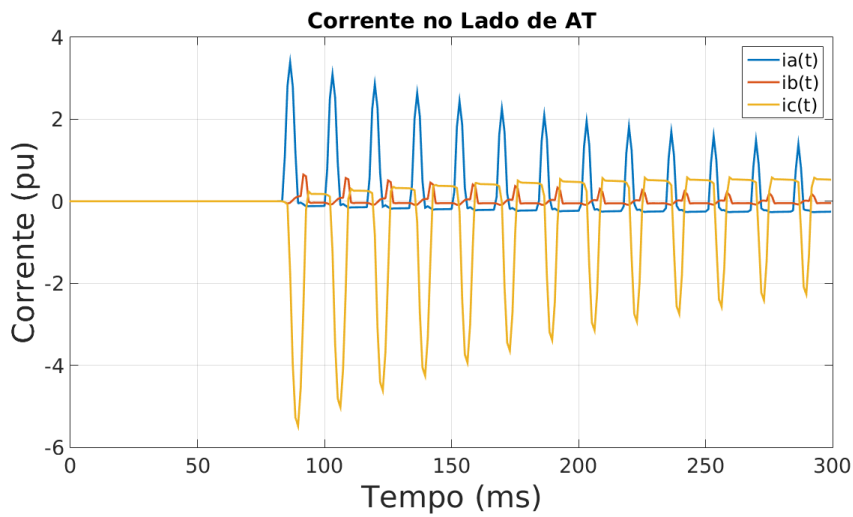


Figura 5.17 - Correntes de *Inrush* no Lado de 230 kV do Transformador, Durante a Energização do Transformador.

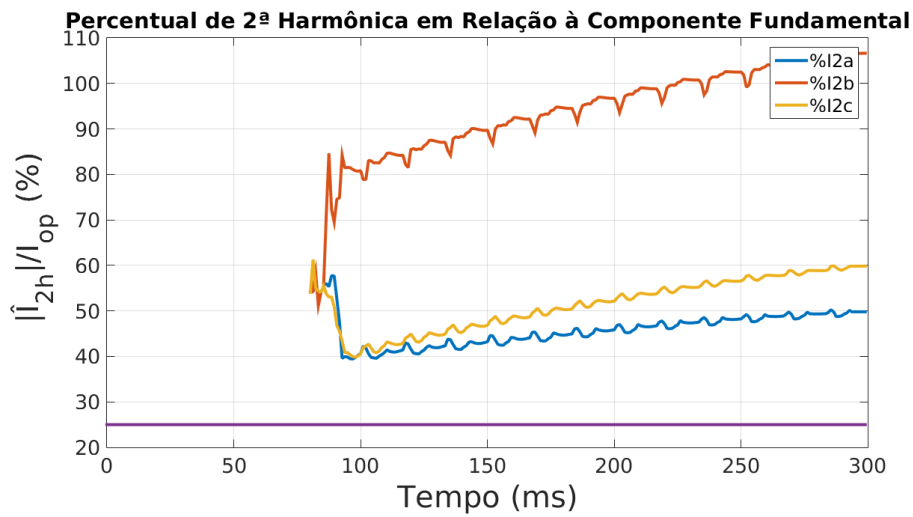


Figura 5.18 - Percentual da Componente de 2ª Harmônica em Relação à Fundamental, Durante a Energização do Transformador.

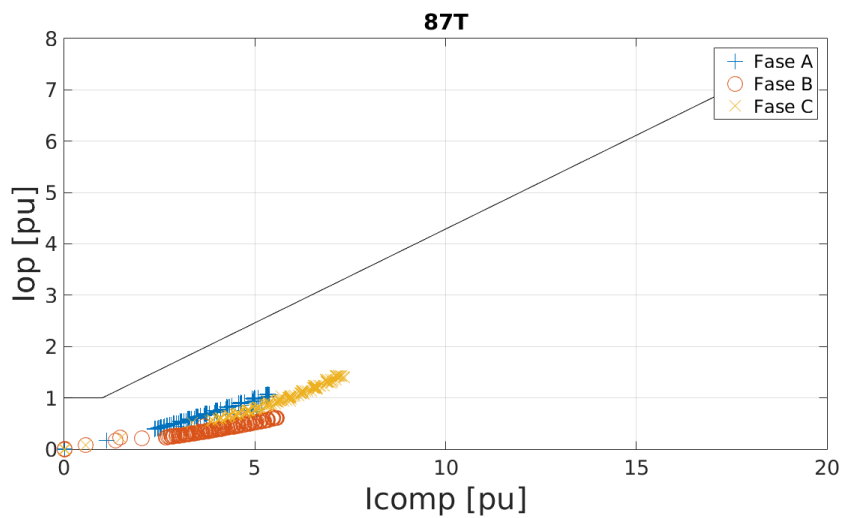


Figura 5.19 - Plano Operacional da Função 87T, Durante a Energização.

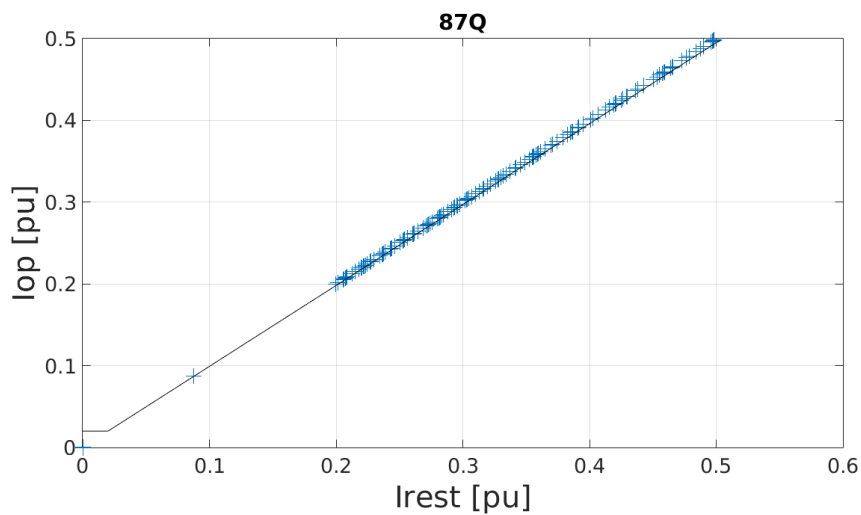


Figura 5.20 - Plano Operacional da Função 87Q, Durante a Energização.

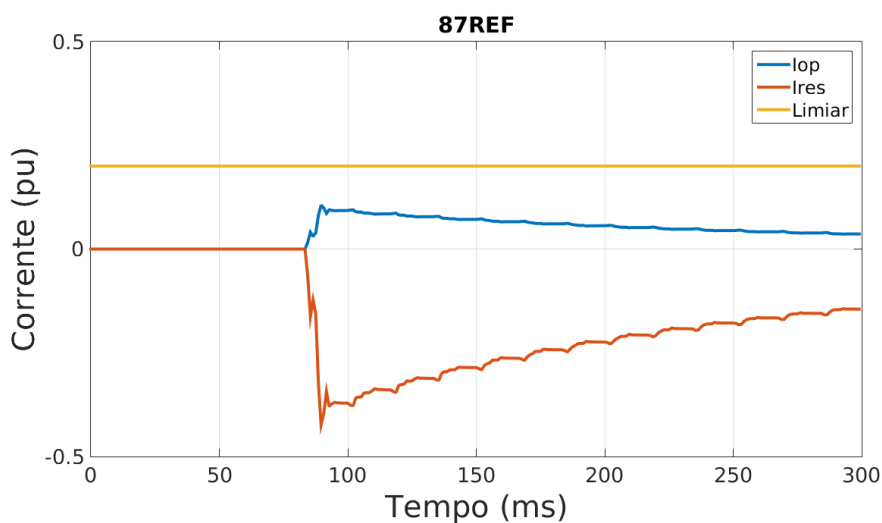


Figura 5.21 - Correntes de Operação e Restrição Calculadas para a Função 87REF, Durante a Energização.

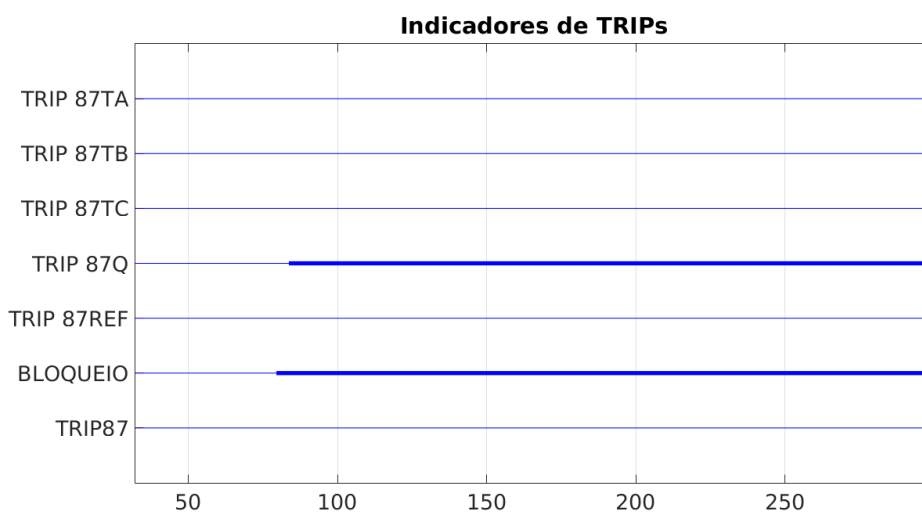


Figura 5.22 - Variáveis que Compõem a Lógica de TRIP, Durante a Energização.

5.4.4 Energização do Transformador com Curto Circuito Interno Espira-Espira, no Lado Y

Historicamente, a filosofia dos esquemas de proteção no setor elétrico brasileiro considera que a atuação da proteção diferencial de transformadores deve ativar o relé de bloqueio (86) dos disjuntores do equipamento, até que sejam efetuadas inspeções em campo pelas equipes locais da subestação e que se averigüe a possibilidade ou não de desabilitar o bloqueio dos disjuntores do transformador.

Muitas vezes, após uma rápida verificação externa da ausência de danos físicos ao equipamento e adjacências, incêndios ou explosões, as equipes técnicas desabilitam o relé de bloqueio e autorizam à operação do sistema a execução das manobras para a tentativa de energização do transformador.

Dentro deste contexto, o caso simulado no ATP contemplou a tentativa de energização pelo lado de 230 kV, no instante $t = 80$ ms, do transformador com um curto circuito espira-espira pré-existente envolvendo 30% das espiras a fim de observar o tempo de atuação das proteções diferenciais implementadas no FPGA.

A **Figura 5.23** apresenta as correntes de *inrush* verificadas no lado primário do transformador. Já a **Figura 5.24**, a **Figura 5.25** e a **Figura 5.26** apresentam, respectivamente, o comportamento dos elementos 87T, 87Q e 87REF frente ao caso simulado. Conforme o comportamento das variáveis lógicas do relé implementado, indicado na **Figura 5.27**, os elementos de proteção diferencial de fase, 87TA, 87TB e 87TC detectaram a falta em 12,74 ms, 13,78 ms e 9,61 ms, respectivamente. Já o elemento de proteção diferencial de sequência negativa, 87Q, detectou o curto interno em apenas 1,28 ms. Contudo, apesar da rápida detecção do defeito pelo elemento 87Q, o sinal de TRIP do relé implementado no FPGA ficou inibido devido à atuação do elemento de bloqueio por harmônicos desde o instante 0,23 ms, após a energização, até o instante 11,7 ms, momento em que o sinal de TRIP principal do relé desenvolvido é de fato ativado. Entende-se que a resposta do relé foi adequada à situação e, mesmo tendo ficado bloqueado por harmônicos por determinado momento, levou apenas 70% de um ciclo para atuar frente à situação de um curto-circuito interno pré-existente.

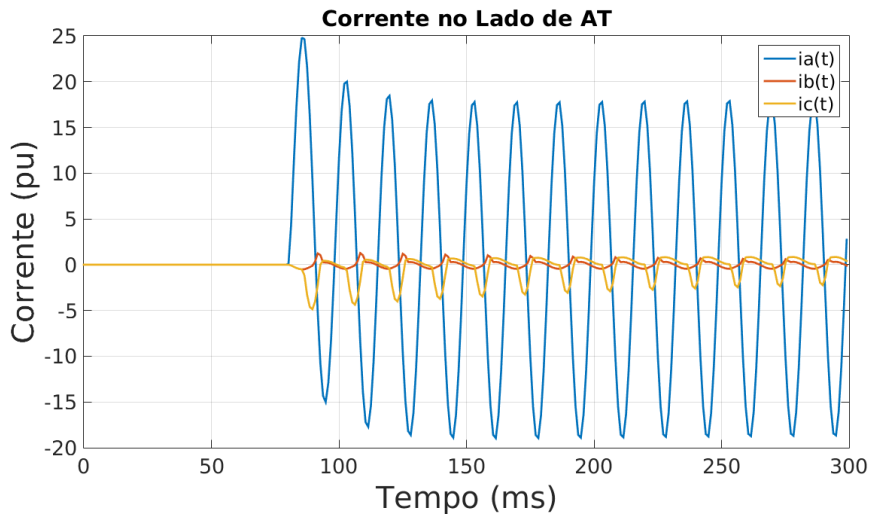


Figura 5.23 - Correntes de *Inrush* no Lado Primário do Transformador, Durante a Energização Sob Falta.

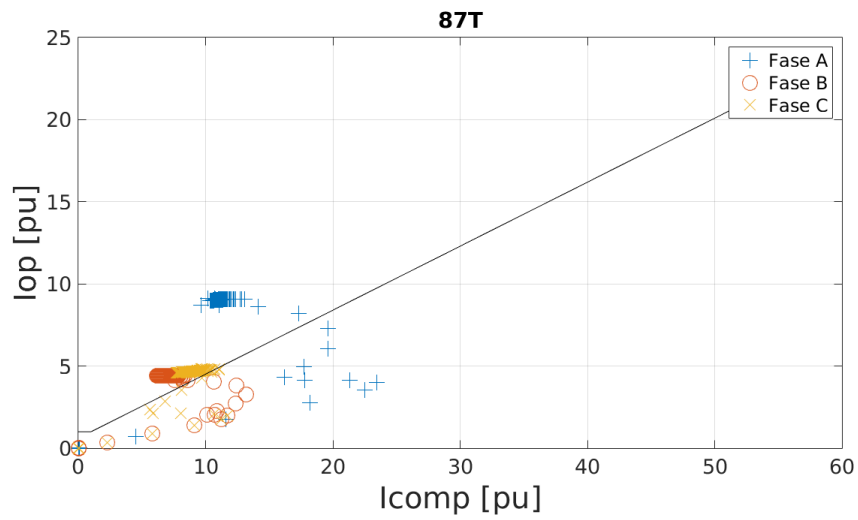


Figura 5.24 - Plano Operacional da Função 87T, Durante a Energização Sob Falta.

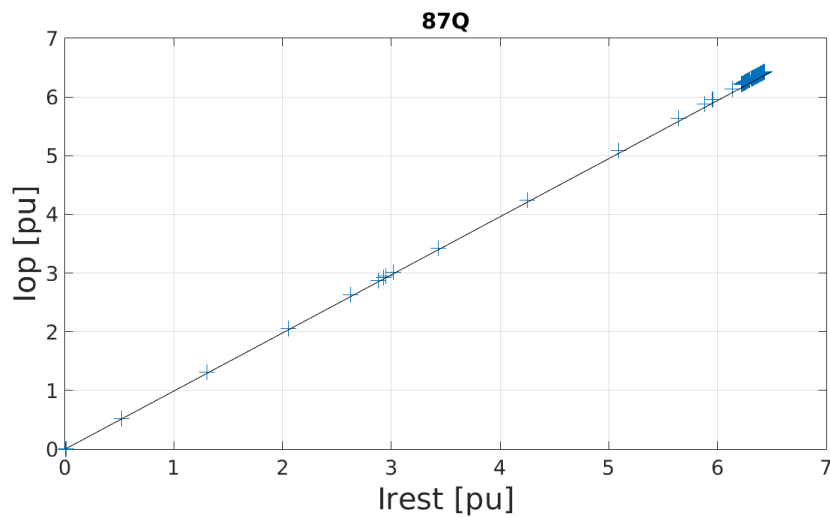


Figura 5.25 - Plano Operacional da Função 87Q, Durante a Energização Sob Falta.

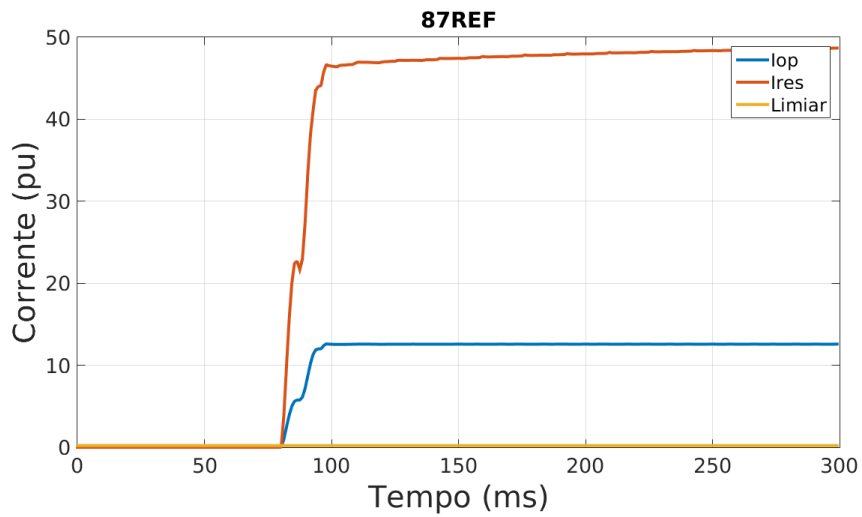


Figura 5.26 - Correntes de Operação e Restrição Calculadas para a Função 87REF, Durante a Energização Sob Falta.

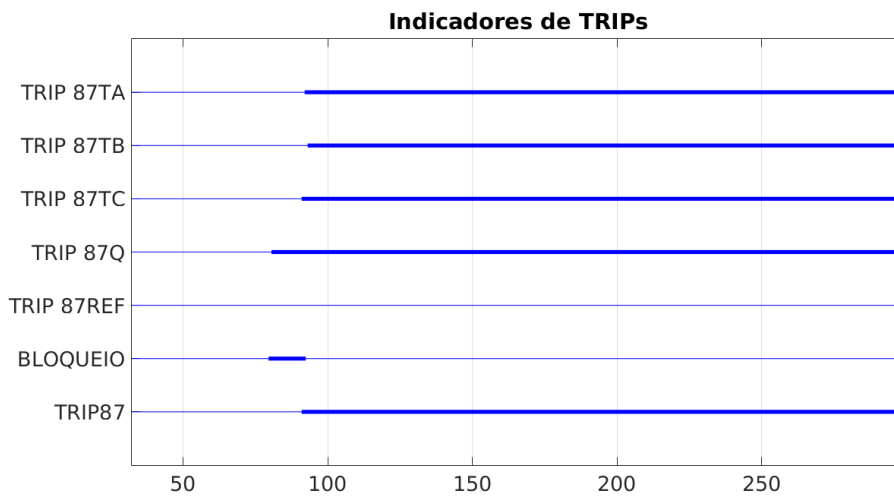


Figura 5.27 - Variáveis que Compõem a Lógica de TRIP, Durante a Energização Sob Falta.

CONCLUSÕES E PROPOSTAS DE TRABALHOS FUTUROS

O trabalho apresentado nessa dissertação consistiu no desenvolvimento de um protótipo de *hardware* dedicado às funções de proteção diferencial de transformadores (87T, 87Q e 87REF), bem como o bloqueio e a restrição por harmônicos.

Desenvolveram-se as referidas funções para serem executadas pelo processador ARM1176JZF-S de 700 MHz, disponível no Raspberry Pi, cujas implementações em C++ serviram como código de referência para o desenvolvimento do *hardware* em FPGA, por intermédio da linguagem VHDL.

Os resultados numéricos obtidos em ambas propostas foram compatíveis, conforme pode ser verificado na **Tabela 4.24**.

Os requisitos estabelecidos para o *hardware* desenvolvido em FPGA foram atingidos, uma vez que o Relé de Proteção Diferencial de Transformadores, com o uso de 94,65% da área disponível no dispositivo Cyclone IV, modelo EP4CE10F17C8N, do fabricante Altera, é capaz de processar e tomar decisões com base e sete amostras simultâneas de correntes (sendo 3 fases primárias, 3 secundárias e um neutro) em 8.403 ciclos de *clock*, ou seja, em 168,06 μ s.

Estes resultados levam a crer que o esforço computacional do SOC desenvolvido no FPGA para as funções de proteção diferencial é bastante inferior ao observado nas funções implementadas em C++ que rodaram no microprocessador do Raspberry Pi, uma vez que, nesta plataforma rodando com um *clock* de 700 MHz, as funções de proteção diferencial levaram conjuntamente em média 30 μ s para processar a atualização das amostras de cada fase. Caso o *clock* do Raspberry Pi fosse reduzido ao mesmo *clock* do FPGA adotado, 50 MHz, o tempo estimado de execução das funções de proteção diferencial seria quatorze vezes maior, ou seja, 420 μ s.

Nestes termos de comparação, o FPGA acelerou o algoritmo de proteção diferencial em 2,5 vezes.

Com relação ao desempenho das funções de proteção diferencial implementadas, pôde-se observar seus comportamentos frente aos distúrbios simulados no sistema de potência proposto no Capítulo 5 dessa dissertação. De um modo geral, pôde-se verificar o comportamento de todas as funções de proteção diferencial e do bloqueio por harmônicos, destacando-se o fato de o elemento 87Q, quando

comparado aos demais elementos de proteção diferencial implementados, necessitar de um número menor de amostras, durante o transitório de curto-circuito, para ser sensibilizado.

Uma vez testado e detalhado nessa dissertação o *hardware* do relé de proteção diferencial de transformadores de potência desenvolvido em FPGA, sugere-se batizá-lo. Por ser um projeto pioneiro no âmbito do LAPSE, entende-se que a denominação mais adequada é **87T-LAPSE-1**.

Em termos de propostas para futuros trabalhos, sugere-se:

- Adequar o relé desenvolvido no FPGA para modelos de FPGA com maior número de elementos lógicos, de forma que seja possível tornar paralela, ao menos, a execução das funções de proteção diferencial F87T (uma por fase), F87Q, F87REF e o bloqueio por harmônicos.
- Aperfeiçoar o módulo aritmético MSA de modo que possa opcionalmente executar apenas operações de soma ou de multiplicação.
- Implementar módulo instantâneo de detecção de faltas externas, que possibilitará bloquear mais rapidamente a atuação dos elementos de proteção diferencial mais sensíveis a possíveis saturações de TCs.
- Agregar outras funções de proteção de transformadores de potência, como sobrecorrente, sobretensão, direcional de neutro, sobreexcitação etc.
- Desenvolver os circuitos eletrônicos de acondicionamento das correntes elétricas provenientes dos TCs.
- Testar o desempenho tanto do *hardware* quanto dos algoritmos de proteção diferencial com faltas reais ocorridas no sistema elétrico brasileiro.
- Testar, integrado a simuladores de tempo real, o desempenho tanto do *hardware* quanto dos algoritmos de proteção diferencial.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Y. G. Paithankar e S. R. Bhide. *Fundamentals of Power System Protection*. Prentice-Hall of India Pvt. Ltd, 2004.
- [2] *IEEE Guide for Protective Relay Applications to Power Transformers*. IEEE Std. C37.91-2000.
- [3] Operador Nacional do Sistema Elétrico – ONS. *Análise Estatística dos Desligamentos Forçados de Componentes do Sistema Elétrico Brasileiro Referente ao Ano de 2012*. 2013.
- [4] C. Christopoulos e A. Wright. *Electrical Power System Protection*. 2ª Edição, Kluwer Academic Publishers, 2010.
- [5] *Transformer Protection Application Guide*. Basler Electric Company, 2003.
- [6] A. G. Phadke e J. S. Thorp. *A New Computer-Based Flux-Restrained Current-Differential Relay for Power Transformer Protection*. IEEE Transactions on Power Apparatus and Systems, 1983.
- [7] A. Wiszniewski e B. Kasztenny. *Fuzzy set approach to transformer differential relay*. Fifth International Conference on Developments in Power System Protection, 1993.
- [8] Z. Gajic, I. Ivankovic, B. Filipovic-Grcic e R. Rubesa. *New General Method for Differential Protection of Phase Shifting Transformers*. International Conference on Advanced Power System Automation and Protection, Coréia do Sul, 2007.
- [9] D. Barbosa, U. C. Netto, H. M. G. C. Branco, D. V. Coury e M. Oleskovicz. *Impacto do paralelismo na proteção diferencial de transformadores de potência*. XX SNPTEE, Recife, 2009.
- [10] J. Ruan, J. Lin. *The application of FPGA in the Field of Relay Protection*. International Journal of Information Technology, Vol. 11, 2005.
- [11] D.V. Courya, R.P.M. Silvaa, A.C.B. Delbemb, M.V.G. Casseb. *Programmable logic design of a compact Genetic Algorithm for phasorestimation in real-time*. Electric Power Systems Research 107, 2014.
- [12] Zhang Guiqing, Feng Tao, Bang Hang, Wang Jianhua, XuHong, Geng Yingsan et. al. *The implementation of digital protection in power system using FPGA*. ASIC, 2001. Proceedings. 4th International Conference on.
- [13] Chunlin Li, Francis Dawson. *A system-on-a-chip architecture for power signal processing*. Power Engineering Society General Meeting, 2003, IEEE.

- [14] Khaled Shehata , Ahmed Bahaa, Karim Morad, Ahmed Shard. *Design and Implementation of FPGA Based and Microcontroller Based Current Relay*. 2003, IEEE.
- [15] Vishal Kumar, Sanjay Prabhu, I. Gupta, H. O. Gupta. *SOPC Design for Implementation of Overcurrent Relay*. 2006, IEEE.
- [16] *Nios II (Gen2) Processor Reference Handbook*. 2016, Altera.
- [17] Xiaolei Liu, Ahmed H. Osman, Om P. Malik. *Real-Time Implementation of a Hybrid Protection Scheme for Bipolar HVDC Line Using FPGA*. 2011, IEEE Transactions on Power Delivery, vol. 26, no. 1.
- [18] *Banco de Preços de Referência ANEEL*, homologado pela Resolução Homologatória nº 758, de 06 de janeiro de 2009.
- [19] Kindermann, Geraldo. *Proteção de Sistema Elétricos de Potência Volume 2*. UFSC.
- [20] *Procedimentos de Rede – Submódulo 2.6, versão 2.0*. Resolução Normativa ANEEL nº 461/11.
- [21] H. J. A. Ferrer e E. O. Schweitzer. *Modern Solutions for Protection, Control, and Monitoring of Electric Power Systems*. Schweitzer Engineering Laboratories, Inc., 2010.
- [22] *IEEE Guide for Protecting Power Transformers*. IEEE Standard C37.91-2008, Jan. 2008.
- [23] Z. Gajić. *Differential Protection for Arbitrary Three-Phase Power Transformers*. Department of Industrial Electrical Engineering and Automation, Lund University, 2008.
- [24] TAVARES, K. A. (2013). *Modelagem e Simulação da Proteção Diferencial de Transformadores de Potência no ATP*. Dissertação de Mestrado, Universidade de Brasília, Brasília - DF.
- [25] WG B5.05 *Report: Modern Techniques for Protecting, Controlling and Monitoring Power Transformers*. Cigré, Jun. 2011.
- [26] J. L. Blackburn e T. J. Domin. *Protective Relaying Principles and Applications*. Boca Raton, USA: CRC Press, 3a Ed., 2007.
- [27] <http://www.slideshare.net/ashwinfcc/transformer-design-and-protection>. Consulta em 02/11/2016.
- [28] K. Behrendt, N. Fischer e C. Labuschagne. *Considerations for using harmonic blocking and harmonic restraint techniques on transformer differential relays*. SEL, 2006.
- [29] B. N. Taj, A. Mahmoudi e S. Kahourzade. *Comparison of Low-Impedance Restricted Earth Fault Protection in Power Transformer Numerical Relay*. Australian Journal of Basic and Applied Sciences, 2011.

- [30] A. G. Phadke and J. S. Thorp, *Computer Relaying for Power Systems, 2nd ed.* West Sussex, UK: John Wiley & Sons Inc, 2009. <http://docs.exdat.com/docs/index-44453.html>
- [31] OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-Time Signal Processing*. New Jersey, USA. Prentice-Hall, 1989.
- [32] SCHWEITZER, E. O.; HOU, D. *Filtering for protective relays*. Annual Western Protective Relay Conference, Spokane, WA, Oct. 1993.
- [33] HART, D. G.; NOVOSEL, D.; SMITH, R. A. *Modified Cosine Filters*. Nov. 2000. U.S. Patent 6,154,687.
- [34] Raspberry Pi. Disponível em: www.raspberrypi.org
- [35] Raspbian. Disponível em <https://www.raspbian.org>
- [36] J. Corbet, A. Rubini, G. Kroah-Hartman (2005). *Linux Device Drivers*.
- [37] www.altera.com
- [38] Cyclone IV Device Handbook, Volume 1, 2014. Altera.
- [39] Steven W. Smith, Ph.D. *The Scientist and Engineer's Guide to Digital Signal Processing*. 2002.
- [40] Heath, Thomas (1921). *A History of Greek Mathematics, Vol. 2*. Oxford: Clarendon Press. pp. 323–324.
- [41] *EMTP Reference Models for Transmission Line Relay Testing*. IEEE PES/PSRC Special Publication, 2004.
- [42] Filho, I. S. S., Silva, K. M. *Implementação em Hardware de Funções de Proteção Aplicadas a Transformadores de Potência*. Simpósio Brasileiro de Sistemas Elétricos, 2016.
- [43] Smith, S. W. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.