

INSTITUTO DE ENGENHARIA NUCLEAR – IEN

ROQUE HUDSON DA SILVA MARTINS

**DESENVOLVIMENTO DE SISTEMA DE PROTEÇÃO
PARA REATOR NUCLEAR DE PESQUISA BASEADO
EM FIELD PROGRAMMABLE GATE ARRAY - FPGA**

Rio de Janeiro

2016

ROQUE HUDSON DA SILVA MARTINS

**DESENVOLVIMENTO DE SISTEMA DE PROTEÇÃO
PARA REATOR NUCLEAR DE PESQUISA BASEADO
EM FIELD PROGRAMMABLE GATE ARRAY - FPGA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia Nucleares do Instituto de Engenharia Nuclear da Comissão Nacional de Energia Nuclear como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Nuclear – Ênfase Acadêmico em Métodos Computacionais Aplicados.

Orientador: Prof. Dr. Paulo Victor Rodrigues de Carvalho

Rio de Janeiro

2016

MART MARTINS, Roque Hudson da Silva.

Desenvolvimento de sistema de proteção para reator nuclear de pesquisa baseado em Field Programmable Gate Array – FPGA / Roque Hudson da Silva Martins – Rio de Janeiro: CNEN/IEN, 2016.

x, 87 f. : il.; 31 cm

Orientador: Prof. Dr. Paulo Victor Rodrigues de Carvalho

Dissertação (Mestrado em Ciência e Tecnologia Nucleares Métodos Computacionais Aplicados) – Instituto de Engenharia Nuclear, PPGIEN, 2016.

1. Segurança de Reatores Nucleares.
2. Reatores de Pesquisa.
3. Sistema de Proteção de Reatores em FPGA.

**DESENVOLVIMENTO DE SISTEMA DE PROTEÇÃO PARA REATOR
NUCLEAR DE PESQUISA BASEADO EM FIELD PROGRAMMABLE
GATE ARRAY - FPGA**

ROQUE HUDSON DA SILVA MARTINS

DISSERTAÇÃO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E TECNOLOGIA NUCLEARES DO INSTITUTO DE ENGENHARIA NUCLEAR DA COMISSÃO NACIONAL DE ENERGIA NUCLEAR COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA NUCLEAR – ÉNFASE ACADÊMICO EM MÉTODOS COMPUTACIONAIS APLICADOS.

Examinada por:

Prof. Paulo Victor Rodrigues de Carvalho, D.Sc. (PPGIEN/CNEN)

Prof. Antônio Carlos de Abreu Mól, D.Sc. (PPGIEN)

Pedro Luiz da Cruz Saldanha, D. Sc. (CNEN)

RIO DE JANEIRO, RJ – BRASIL

JUNHO DE 2016

Dedico este trabalho em especial à Deus. À minha mãe Norma, pessoa que me mostrou o real valor da vida. Ao meu pai. À minha irmã Keytiúscia. Aos meus demais familiares e aos meus amigos.

AGRADECIMENTOS

Agradeço a Deus em primeiro lugar, por ter me proporcionado força, foco e motivação ao longo desta trajetória acadêmica; principalmente discernimento nos momentos mais difíceis e de desânimo.

A minha guerreira e amada mãe, Norma, que sempre esteve ao meu lado e não negou esforços para que eu pudesse estudar, por sempre colocar meu nome em suas orações, intercedendo junto a Deus, de forma que Ele pudesse me iluminar e guardar. Por me proporcionar estudo do jardim de infância à pós-graduação. Se aqui cheguei, foi graças a ela. O meu muito obrigado, MÃE!

Ao meu pai, José Martins, que também contribuiu de forma direta e indiretamente nos meus estudos.

À minha irmã Keytiúsica, que sempre esteve ao meu lado, me dando ânimo para que eu não desistisse e tendo paciência nos meus momentos de nervosismo.

Aos meus familiares, em especial aos meus tios Cássia e Hamilton, que foram como pais para mim, ajudando-me, guiando-me e aconselhando-me. Hoje, meu tio não está mais presente entre nós, mas sei que ele foi peça fundamental nesta conquista e estaria muito feliz. Muito obrigado, padrinhos.

Ao meu orientador Paulo Victor, pelos ensinamentos passados, pela paciência em me orientar, corrigir os meus erros e textos e pelas inúmeras dúvidas sanadas. Obrigado pelo apoio e encorajamento.

Ao Marcos Santana, por ajudar diretamente nos ensinamentos e apoio com materiais do tema da dissertação.

Aos professores do Programa de Pós-Graduação em Ciência e Tecnologia Nucleares, pelo conhecimento transmitido e pelo incentivo.

Aos funcionários da Secretaria Acadêmica, Dona Cláudia, Antônio Sérgio e Ana Paula, por sempre se colocarem a disposição para ajudar com eficiência e rapidez.

Aos demais funcionários do IEN e do restaurante que sempre me trataram com carinho e respeito.

Aos amigos da turma de mestrado de 2013, obrigado pelos momentos de alegria, desespero pelas provas difíceis e caronas até a COPPE. À Pâmela Nolasco, Caroline Mattos, Luciana Siqueira, Franciana Rosa, Ângelo Cnop, Mário Villela, Filipe Vieira, Filipe Almada, Alexander Debossam, Sérgio Costa, Anna Karla, Deise Diana, Luiz Gualberto, João Antunes, Mauro Lúcio e Jorge Rezende.

Em especial, à amiga Pâmela Nolasco, que juntos fazíamos o trajeto de quase 250 km entre Volta Redonda ao IEN, obrigado pela amizade, companhia e por suportar os meus momentos de cansaço e nervoso.

Ao professor e mestre Pedro Saldanha, este que foi peça fundamental para que eu chegasse até o mestrado, muito obrigado pela indicação e pelo auxílio prestado.

Aos professores do Centro Universitário de Barra Mansa - UBM, Hélio Camargo (in memoriam), Carlos Roberto e José Nilton, que me recomendaram a este mestrado.

Aos meus amigos verdadeiros, por estiveram ao meu lado, proporcionando força, suporte emocional e aguentando meus momentos de desespero e ausência. Não serei injusto de citar alguns e esquecer por ventura outros. Deixo aqui o meu abraço e agradecimento.

Aos meus alunos que sempre me incentivaram ao mestrado. Em especial a turma 9311.

Aos amigos da escola Jahyra, que me encorajaram a este mestrado.

À CNEN pelo suporte financeiro da bolsa de mestrado durante a realização da pós-graduação.

MUITO OBRIGADO!

“Nunca se deve complicar o que se pode ser feito
de maneira simples.”

Zilda Arns

RESUMO

Neste trabalho é realizado um estudo e apresentada uma proposta de implementação de um modelo de Sistema de Proteção para Reatores Nucleares de Pesquisa, através da utilização de dispositivo programável FPGA (Field Programmable Gate Array). Bem como são estudadas as lógicas de proteção de um desligamento automático (TRIP) de um reator, que garantem a segurança nesse tipo de sistema. A utilização desses novos mecanismos de controle e operação são desenvolvidos a fim de garantir que os limites do nível de segurança de uma usina não sejam superados, logo esses mecanismos podem acontecer de maneira isolada ou em conjunto de forma a garantir a segurança.

Para que essa implementação seja completa, são apresentados os principais conceitos e aspectos referentes a Sistema de Proteção, principalmente no que se refere à segurança dos reatores nucleares de pesquisa. Sendo exposto alguns termos utilizados na área de aplicação.

O sistema proposto nesta dissertação foi modelado através da linguagem de descrição de hardware VHDL (Very Speed Integrated Circuit), sendo utilizado o software ModelSim da Altera Software na programação das lógicas de desligamento automático (TRIP) do reator de pesquisa e na simulação de hipóteses de acionamento das mesmas.

Os resultados deste estudo apontam que para toda a aplicação de software em reatores nucleares como ferramenta de auxílio na segurança dos mesmos, passe por um teste de verificação e validação, de forma a se adequar a norma IEC 60880.

Este trabalho revela-se de grande importância, considerando-se que os sistemas de proteção de um reator nuclear constitui como um elemento básico na segurança dos mesmos.

Palavras-Chaves: reator nuclear, sistema de proteção, VHDL, FPGA, IEC 60880

ABSTRACT

This study presents a implementation purpose of a protection system for research nuclear reactors by using a programed device FPGA (Field Programmable Gate Array). As well as logic protection method involved on an automatic shutdown (TRIP) of a reactor, that ensure the security on such systems. These new control and operation mechanics are developed to guarantee that the security limits of a power plant are not exceeded, these mechanics can work isolated or in groups to safe guard the security levels.

For this implementation to be completed, there will be presented the main aspects and concepts referred to protection systems, mostly about research nuclear reactors, with some applications terms exposed.

The system proposed at this paper was developed following the VHDL (Very High Speed Integrated Circuits) hardware describing language, and the Modelsim software from Altera Software to program the automatic turning off routines, and hypothetical simulations for such.

The results show that for every software application for supporting nuclear reactors, like security devices, they have to meet the IEC 60880 criteria.

This paper have great importance, seeing that nuclear reactor security systems, are a basic element for ensure the reactor security.

Keywords: nuclear reactor, protection system, FPGA, VHDL

LISTA DE ILUSTRAÇÕES

FIGURA 1 - Reatores nucleares de pesquisa no mundo	22
FIGURA 2 - Sede do reator OPAL	23
FIGURA 3 - Estrutura interna do reator OPAL	24
FIGURA 4 - Interior do reator IEA – R1	25
FIGURA 5 - Prédio do reator IEA – R1	26
FIGURA 6 - Interior do reator TRIGA IPR –R1	27
FIGURA 7 - Prédio do reator ARGONAUTA	28
FIGURA 8 - Reator ARGONAUTA	29
FIGURA 9 - Prédio do reator IPEN/MB – 01	30
FIGURA 10 - Controlador Lógico Programável (CLP)	33
FIGURA 11 - Exemplo de configuração de sistema de controle de uma central nuclear	34
FIGURA 12 - Modelo estrutural de uma FPGA	40
FIGURA 13 - Etapas de um projeto de uma FPGA	41
FIGURA 14 - Exemplo de uma das lógicas de TRIP de um reator	49
FIGURA 15 - Processo de V & V de software	51
FIGURA 16 - Situação 1 para a Lógica de TRIP – Partida e Largo Alcance	54
FIGURA 17 - Situação 2 para a Lógica de TRIP – Partida e Largo Alcance	55
FIGURA 18 - Situação 1 para a Lógica de TRIP - Sistema de Resfriamento Primário.....	56
FIGURA 19 - Situação 2 para a Lógica de TRIP - Sistema de Resfriamento Primário	56

LISTA DE TABELAS

TABELA 1 - Situação de estado dos reatores de pesquisa pelo mundo	22
---	----

LISTA DE ABREVIATURAS E SIGLAS

FPGA	– Field Programmable Gate Array
TRIP	– Desligamento Automático
VHDL	– Very Speed Integrated Circuit
CO2	– Gás Carbônico
IAEA	– International Atomic Energy Agency
AIEA	– Agência Internacional de Energia Atómica
OPAL	– Open Pool Australian Light-water
ANSTO	– Organizaçāo Nuclear Australiana de Ciēncia e Tecnologia
INVAP	– INVestigaciones APlicadas
MW	– Mega Watts
RMB	– Reator Multipropósito Brasileiro
CNEN	– Comissāo Nacional de Energia Nuclear
IPEN	– Instituto de Pesquisas Energéticas e Nucleares
SP	– São Paulo
AAN	– Análise por Ativação com Nêutrons
TRIGA	– Treinamento, Pesquisa, Isótopos, General Atomics
CDTN	– Centro de Desenvolvimento da Tecnologia Nuclear
IPR	– Instituto de Pesquisas Radioativas
MG	– Minas Gerais
UFMG	– Universidade Federal de Minas Gerais
kW	– kilowatt
Argonaut	– Argonne National Laboratory Nuclear Assembly for University Trainning
W	– Watts
IEN	– Instituto de Engenharia Nuclear
DE	– Divisāo de Eletrônica
USP	– Universidade de São Paulo
FINEP	– Financiadora de Estudos e Projetos
CRT	– Cathode Ray Tube
CI	– Circuito Integrado
CLP	– Controlador Lógico Programável

IEC	– Comissão Eletrotécnica Internacional
IHM	– Interface Homem-Máquina
ASIC	– Application Specific Integrated Circuit
ARPANSA	– Australian Radiation Protection and Nuclear Safety Agency
E/S	– Entradas / Saídas
RAM	– Memória de Acesso Randômico
CLB	– Configuração de Blocos Lógicos
LUT	– Look-up tables
EEPROM	– Memória apenas de leitura programável eletricamente apagável
SRAM	– Memória Estática de Acesso Aleatório
HDL	– Linguagem de Descrição de Hardware

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO	17
1.1 CONTEXTUALIZAÇÃO	17
1.2 JUSTIFICATIVA	17
1.3 OBJETIVOS	18
1.3.1 OBJETIVO GERAL	18
1.3.2 OBJETIVO ESPECÍFICO	18
1.4 ESTRUTURA DA DISSERTAÇÃO	19
 CAPÍTULO 2 - FUNDAMENTAÇÃO TEÓRICA	20
2.1 REATOR NUCLEAR	20
2.1.1 REATOR DE PESQUISA	21
2.1.1.1 REATORES DE PESQUISA NO MUNDO	21
2.1.1.2 REATORES DE PESQUISA NO BRASIL	24
2.2 INSTRUMENTAÇÃO	31
2.3 ENGENHARIA DE SEGURANÇA	35
2.3.1 SISTEMA DE SEGURANÇA PASSIVO	36
2.3.2 SISTEMA DE SEGURANÇA ATIVO	36
2.3.3 SISTEMA DE PROTEÇÃO	36
2.3.4 SISTEMA DE SEGURANÇA DE UM REATOR DE PESQUISA MODERNO	37
2.4 SISTEMAS DIGITAIS EM REATORES DE PESQUISA	38
2.5 FPGA (FIELD-PROGRAMMABLE GATE ARRAYS)	39
2.5.1 FPGA EM REATORES NUCLEARES	42
2.6 VHDL – VERY SPEED INTEGRATED CIRCUIT	43
2.7 V & V (VERIFICAÇÃO E VALIDAÇÃO) – IEC 60880	43
 CAPÍTULO 3 – METODOLOGIA	45
3.1 INTRODUÇÃO	45
3.2 ETAPAS DA METODOLOGIA	45

CAPÍTULO 4 – PROJETO DO SISTEMA	50
4.1 ESTUDO DE CASO	50
4.1.1 NECESSIDADE DO TESTE DE V & V	52
4.1.2 TIPOS DE SOFTWARES	52
4.2 TESTES	53
4.3 IMPLEMENTAÇÃO	53
 CAPÍTULO 5 – CONCLUSÃO	57
 REFERÊNCIAS BIBLIOGRÁFICAS	59
 APÊNDICE A	62
 APÊNDICE B	63
 APÊNDICE C	77

CAPÍTULO 1 – INTRODUÇÃO

Este capítulo define o escopo do trabalho e o contextualiza, bem como introduz algumas definições sobre a segurança de reatores de pesquisa e a aplicação de sistemas digitais na segurança dos mesmos.

Estes conceitos são necessários para um melhor entendimento acerca desta dissertação.

1.1 CONTEXTUALIZAÇÃO

A segurança dos reatores em funcionamento ao redor do mundo, vêm sendo muito discutida pela IAEA – Agência Internacional de Energia Atômica. A grande motivação da discussão deste assunto foi devido ao acidente na Central Nuclear de Fukushima, no Japão em março de 2011.

O documento (IAEA, 2005), descreve que a análise de segurança aplicada a um reator de pesquisa deve contemplar todas as atividades realizadas para atingir o objetivo para o qual este foi projetado, construído ou modificado. Incluindo a construção, manutenção, testes e inspeção, troca de combustível e manuseio de material radioativo, operação de dispositivos experimentais, usando os sistemas do reator de pesquisa e outras atividades associadas.

A operação dos reatores de pesquisa segundo o documento (IAEA, 2006), é caracterizada por ações e intervenções humanas, sendo que a maioria apresenta pequeno potencial de perigo para o público em comparação com os reatores de potência, que podem representar um maior risco potencial aos seus operadores.

1.2 JUSTIFICATIVA

Com o desenvolvimento da tecnologia na área da instrumentação, diversos mecanismos e sistemas foram criados, de forma a trazerem benefícios e colaborarem com o homem. E essa evolução da instrumentação vem sendo utilizada na área nuclear e tem recebido uma atenção especial.

A criação de novos mecanismos, visa além de monitorar e comandar os processos a quem se destinam, reduzir os risco inerente de uma aplicação a um nível considerado aceitável.

Logo, acredita-se que o grande desafio de implementação de software em plantas nucleares está na dificuldade de realização do teste de V & V (Verificação e Validação) no processo tradicional de desenvolvimento de softwares. Pois é necessário que estes softwares estejam compatíveis com todas as especificações da norma IEC 60880.

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

O objetivo deste trabalho, é apresentar um método para o desenvolvimento de sistemas de proteção de reatores nucleares de pesquisa usando FPGA, mostrando que as diversas fases do processo tradicional de Verificação & Validação podem ser simplificadas. Para isso foram desenvolvidas e simuladas as lógicas de proteção de um sistema de proteção de um reator de pesquisa através da utilização de circuitos integrados FPGA (Field Programmable Gate Array), utilizando a modelagem com linguagem de descrição de hardware VHDL (VHSIC Hardware Description Language).

1.3.2 OBJETIVO ESPECÍFICO

Como objetivos específicos, têm-se:

- Desenvolvimento de um modelo de sistema de proteção para simular as lógicas de proteção do mesmo;
- Definição das lógicas de proteção através de blocos lógicos a serem programados em VHDL;
- Estudo de verificação e validação do software;
- Implementação em hardware através da modelagem em VHDL, com o programa ModelSim, das lógicas de proteção do TRIP de um reator;
- Simulação e verificação do modelo de sistema de proteção modelado.

1.4 ESTRUTURA DA DISSERTAÇÃO

A dissertação está organizada em cinco capítulos. No primeiro capítulo, consta a introdução, a justificativa e os objetivos.

O segundo capítulo apresenta uma fundamentação teórica realizada dos assuntos e conceitos essenciais, estudados ao longo do desenvolvimento da dissertação, onde são descritos os tipos de reatores, sistema de proteção, circuitos integrados FPGA, linguagem VHDL e Teste de V & V (Verificação e Validação).

No terceiro capítulo é descrita a metodologia utilizada na realização da dissertação.

O projeto do sistema é apresentado no quarto capítulo, no qual é descrito também um estudo de caso na proposta de verificação e validação do software, bem como os testes realizados.

O quinto e último capítulo apresenta a conclusão e os resultados obtidos.

CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda alguns conceitos fundamentais para o entendimento da dissertação, esclarecendo aspectos sobre reatores nucleares, em particular os reatores nucleares de potência. Assim, o leitor contextualizará o estudo a ser aplicado no desenvolvimento do projeto. O capítulo também inclui uma breve análise da engenharia de segurança de um reator, bem como os conceitos da instrumentação e sistemas de controle dos reatores nucleares. E por fim, uma abordagem sobre FPGA, VHDL e Teste de V & V (Verificação e Validação).

2.1 REATOR NUCLEAR

Os reatores nucleares usam como princípio básico a fissão do núcleo de determinados isótopos do Urânio e do Plutônio, através de nêutrons com energias determinadas, gerando a cada fissão uma grande quantidade de energia, produtos de fissão radioativos e nêutrons de altas energias. Com o efeito da fissão de vários núcleos, e com a consequente emissão de nêutrons a cada fissão, é gerada uma reação em cadeia que, controlada, pode ser utilizada para várias finalidades. (PERROTA, 1999).

Os reatores nucleares de pesquisa e de potência se diferenciam da seguinte forma, enquanto o reator nuclear de pesquisa serve para entender as propriedades físicas e químicas de materiais, utilizando o fluxo de nêutrons, ou seja, aproveitam as partículas e radiações geradas pelas reações nucleares. Já os reatores de potência, produz energia através do calor que é gerado pelo núcleo e a transforma em energia elétrica.

O informativo da (ABEN, 2013) apresenta diversas diferenças entre os reatores de pesquisa e os de potência, e uma delas é a segurança. Onde nos reatores de pesquisa, a piscina do reator é a própria blindagem e refrigerante. E esses tipos de reatores têm o seu funcionamento e projeto para obter o intenso fluxo de nêutrons necessários às diferentes aplicações em usinas nucleares. Estes, focam na eficiência termodinâmica para produzirem o máximo de eletricidade como resultado de seus processos, os equipamentos multipropósitos operam a baixas temperaturas, visando maximizar o uso dos nêutrons gerados pela fissão dos átomos de urânio.

Hoje o Brasil conta com seis reatores, sendo quatro de pesquisa e dois de potência (ANGRA I e II). Ressaltando que o número de reatores de potência aumentará em uma unidade, pois ANGRA III está em construção.

Este trabalho baseará sua linha de pesquisa utilizando um reator de pesquisa.

2.1.1 REATOR DE PESQUISA

Apresentam características físicas e de operação diferentes de um reator de potência. Por serem menores e simples, os reatores de potência apresentam uma potência máxima de 80 MW, operam com temperaturas mais baixas, necessitam de uma menor quantidade de combustível. E sua refrigeração (com água) acontece por meio de convecção natural ou forçada.

Os reatores de pesquisa são usualmente utilizados na:

- geração de nêutrons para diferentes propósitos científicos e sociais, podendo destacar a produção de radioisótopos para a medicina nuclear;
- produção de feixe de nêutrons;
- ativação neutrônica;
- radiografia de nêutrons (neutronografia);
- análise de ativação neutrônica .

2.1.1.1 REATORES DE PESQUISA NO MUNDO

No mundo, o número de reatores de pesquisa é de 774, de acordo com informações que podem ser obtidas no site da AIEA. Esta quantidade engloba os reatores que estão sendo construídos, os planejados, os que já foram descomissionados, os cancelados, os desligados, os que foram desligados temporariamente e os que estão em operação nos 56 países que são membros da AIEA, como pode ser mostrado na figura 1.

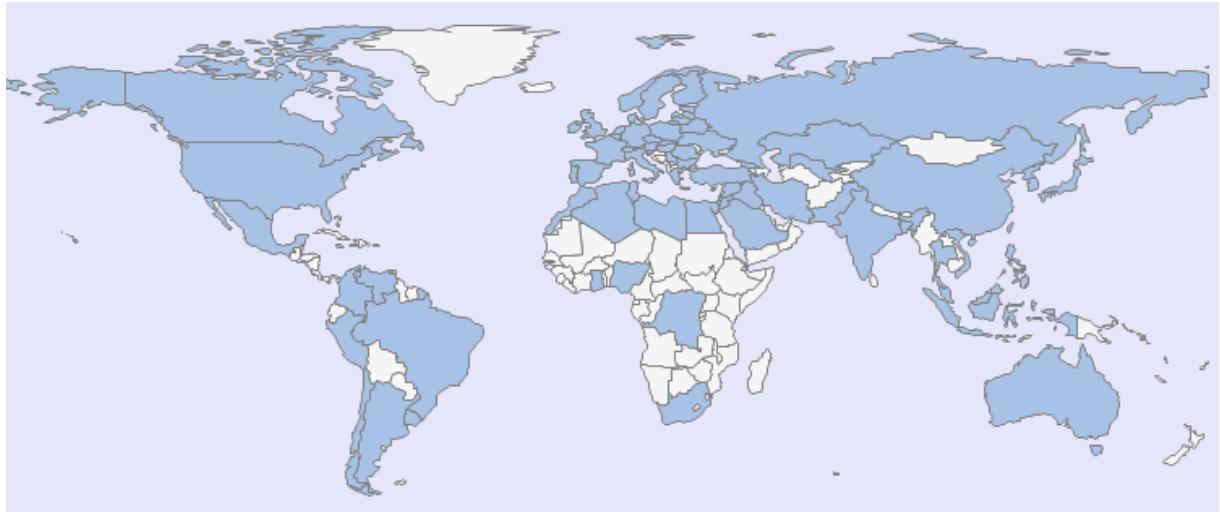


Figura 1 – Reatores Nucleares de Pesquisa no mundo <fonte: AIEA, 2015>

Atualmente existem 246 reatores de pesquisa em operação, a tabela 1 o estado e quantitativo de como os reatores de encontram.

SITUAÇÃO DE ESTADO	PAÍSES DESENVOLVIDOS	PAÍSES EM DESENVOLVIMENTO	TODOS OS PAÍSES
Em Operação	157	89	246
Desligado Temporariamente	13	6	19
Em Construção	4	3	7
Planejado	3	8	11
Desligado	118	21	139
Descomissionado	319	25	344
Cancelado	4	4	8
Total	618	156	774

Tabela 1: Situação de estado dos reatores nucleares de pesquisa pelo mundo
<FONTE: AIEA, 2016>

Reator OPAL (Open Pool Australian Lightwater)

Segundo informações retiradas do site da ANSTO (Australian Nuclear Science and Technology Organization). O OPAL é um reator de pesquisa localizado na Austrália, a apenas 40 km de Sydney, maior cidade australiana. Foi projetado pela empresa Argentina INVAP

(INVESTIGACIONES APLICADAS). O equipamento é considerado um exemplo de confiabilidade na indústria nuclear. Exerce um papel importante para a sociedade junto com uma série de outros reatores semelhantes ao redor do mundo. Funcionam como “fábricas de nêutrons”, na produção de radioisótopos que auxiliam na detecção e tratamento do câncer, bem como de feixes de nêutrons para a pesquisa de materiais fundamentais.

Inaugurado em abril de 2007, atingiu sua primeira criticalidade em 12 de agosto de 2006, antes mesmo de ser inaugurado. A figura 2 mostra o sede onde o OPAL se localiza.



Figura 2 – Sede do reator OPAL <fonte: ANSTO, 2015>

Estrutura Física do OPAL

O núcleo do reator OPAL é compacto, como se pode visualizar na figura 3, composto por 16 conjuntos de combustível, utilizam uma matriz 4x4, contendo cada conjunto 21 placas de combustíveis. E também 5 barras de controle, que controlam a alimentação do reator, facilitando assim o seu desligamento. Sendo que uma dessas barras é em forma de cruz e localizada no centro do interior do núcleo, enquanto as outras quatro placas dividem o núcleo em quatro quadrantes.

O combustível utilizado no reator OPAL é o urânio de baixo enriquecimento, produzindo uma potência máxima de 20 MW.

O conjunto do combustível (core) são arrefecidos por água leve desmineralizada (água comum) e são cercados por uma liga de zircônio “refletor” que contém agua pesada. O recipiente do refletor está posicionado na parte inferior de uma piscina de 13 metros de profundidade. Sendo o design desta piscina aberto, é mais fácil de ver e manipular itens dentro da mesma e a sua profundidade garante a proteção contra radiações eficaz do pessoal que trabalha acima da piscina.

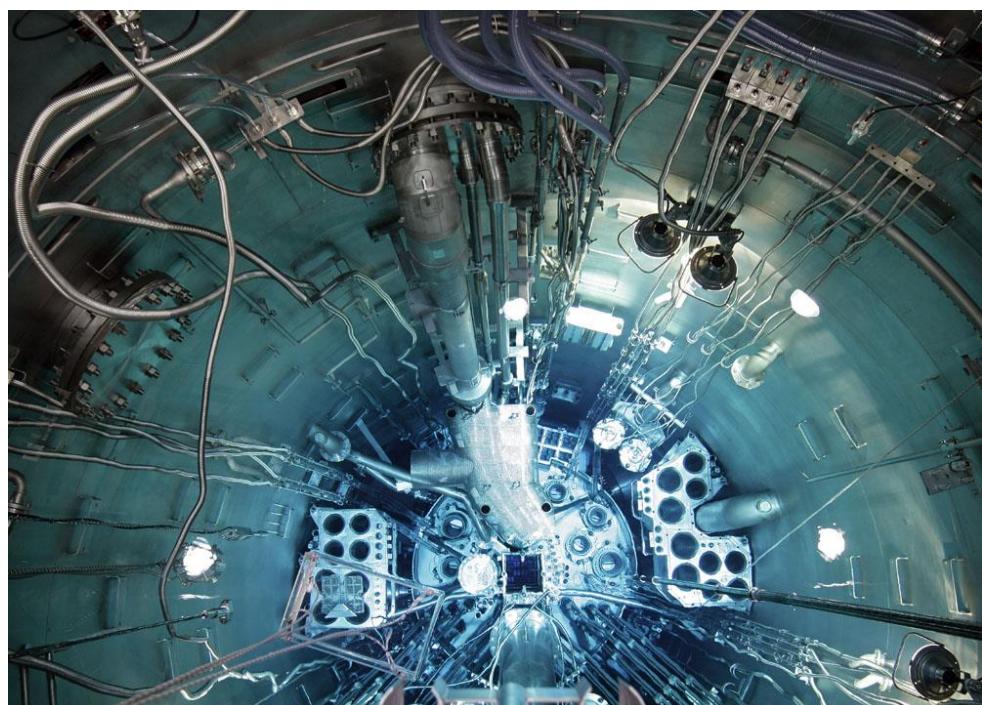


Figura 3 – Estrutura Interna do reator OPAL <fonte: ANSTO, 2015>

2.1.1.2 REATORES DE PESQUISA NO BRASIL

O Brasil possui hoje quatro reatores de pesquisas em operação, e mais um em planejamento, que é o Reator Multipropósito Brasileiro (RMB). Todos esses reatores são fiscalizados e certificados pela CNEN (Comissão Nacional de Energia Nuclear).

Reator IEA – R1

As informações contidas no site do IPEN (Instituto de Pesquisas Energéticas e Nucleares), informam que o reator IEA – R1 foi construído em 1956, dentro da Universidade

de São Paulo, no bairro do Butantã – SP. Atingiu a sua primeira criticalidade em 16 setembro de 1957, antes mesmo de ser inaugurado, que ocorreu em janeiro de 1958 pelo então presidente da época Juscelino Kubitscheck. É um reator de pesquisa tipo piscina, moderado e refrigerado a água leve e que utiliza elementos de berílio e de grafite como refletores.

Utilizando urânio de baixo enriquecimento (em torno de 20%) como combustível. Possui um núcleo com 20 elementos combustíveis. A estrutura interna do IEA – R1 é mostrado na figura 4.

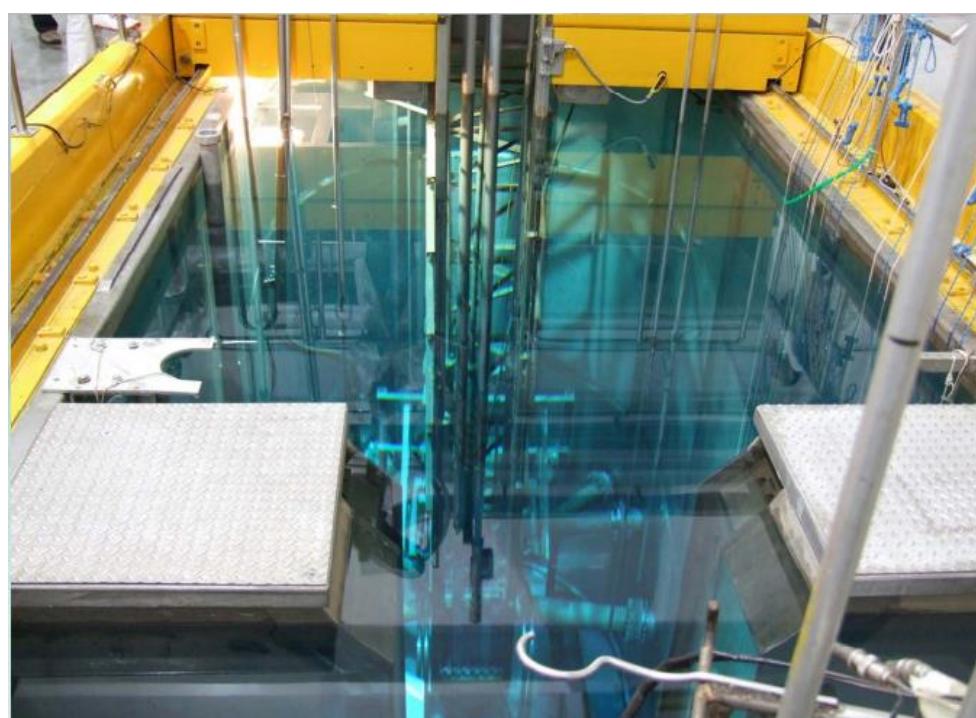


Figura 4 – Interior do reator IEA – R1 <fonte: SAXENA, 2007)

Hoje é o maior reator de pesquisa em operação no Brasil. A figura 5 mostra o prédio do reator. Projetado para trabalhar com uma potência máxima de 5 MW, trabalhou nas primeiras décadas, com uma potência de 2 MW, com 5 MW apenas em 1997 e hoje trabalha com uma potência de 4,5 MW.



Figura 5 – Prédio do Reator IEA – R1 <fonte: SAXENA, 2007>

Atualmente, o reator IEA-R1 é utilizado para algumas finalidades como:

- Produção de radioisótopos para uso em medicina nuclear;
- Produção de fontes radioativas para gamagrafia industrial e de radioisótopos para uso como traçadores em processos industriais;
- Pesquisas em Física Nuclear;
- Serviços de neutrongrafia;
- Irradiação de amostras para a realização de análises multielementares, utilizando a técnica AAN (Análise por Ativação com Nêutrons), em materiais geológicos, produtos;
- Treinamento de pessoal licenciado para operação de reatores.

Reator TRIGA IPR – R1

O reator do tipo TRIGA (Trainning, Research, Isotopes, General Atomic), foi comprado pelo então Instituto de Pesquisas Radioativas (IPR), que hoje se chama Centro de Desenvolvimento da Tecnologia Nuclear (CDTN/CNEN), e está localizado em Minas Gerais, na cidade de Belo Horizonte – MG, dentro do campus da Universidade Federal de Minas Gerais (UFMG).

Sua característica principal é a segurança intrínseca, relacionado à transferência de calor. Essas informações essas que foram retiradas do site da CDTN. A figura 6 demonstra a estrutura interna do reator.

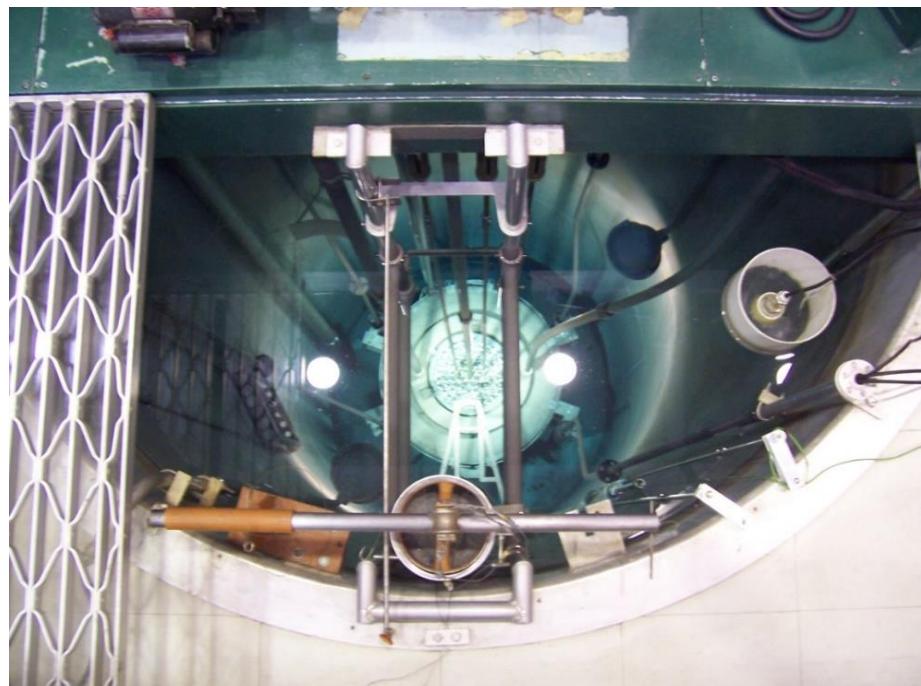


Figura 6 – Interior do reator TRIGA IPR – R1 <fonte: CDTN, 2014>

Atingiu sua primeira criticalidade em 06 de novembro de 1960, trabalha a uma potência máxima de 100 KW, mas já foram realizados testes que utilizavam uma potência de 250 KW. Utiliza combustível composto de liga metálica de urânio que contém de 8% a 8,5% em peso de urânio enriquecido a 20%. São bastantes usados na formação de operadores e em pesquisas experimentais em termo hidráulica, neutrônica, instrumentação e controle.

ARGONAUTA

Localizado no Instituto de Engenharia Nuclear, na Cidade Universitária – Ilha do Fundão no Rio de Janeiro, o ARGONAUTA (figura 7) é um reator do tipo Argonaut (Argonne National Laboratory Nuclear Assembly for University Trainning), construído por uma empresa brasileira, atingiu sua primeira criticalidade em 20 de fevereiro de 1965. Trabalha com uma potência máxima de 5 KW e uma operação contínua de 500 W, utilizando urânio enriquecido

a 20% em U-235. É utilizado atualmente para o ensino, pesquisa e treinamento de pessoal especializado em ciência e tecnologia nuclear.



Figura 7 – Prédio do Reator ARGONAUTA <fonte: IEN, 2012>

Após a inauguração do Reator Argonauta foi iniciada a construção dos prédios complementares ao do reator Argonauta, consolidando-se no final dos anos 60 as operações do Instituto de Engenharia Nuclear. Foram então instaladas, a DFN culminando com a construção de um Ciclotron e laboratórios anexos, a Divisão de Eletrônica (DE), a de Materiais e Metalurgia juntamente com a de Química Nuclear, completadas com a de Aplicações de Radioisótopos e de Proteção Radiológica. O reator ARGONAUTA pode ser visto na figura 8.

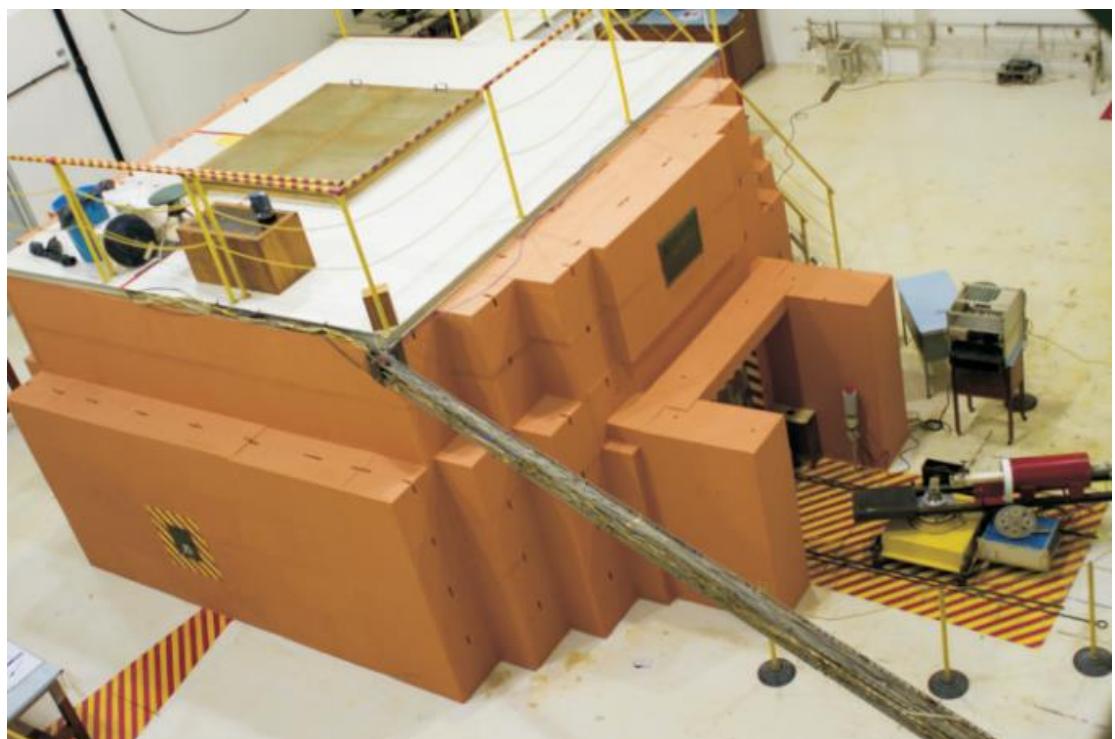


Figura 8 – Reator ARGONAUTA <fonte: IEN, 2012>

IPEN/MB-01

Construído no campus da Universidade de São Paulo (USP), o IPEN/MB-01 é um reator (figura 9) projetado por pesquisadores e engenheiros do IPEN-CNEN/SP. Atingiu sua primeira criticalidade em 9 de novembro de 1988. As varetas de combustíveis do reator são constituídas de aço inox, contendo em seu interior 52 pastilhas combustíveis de urânio enriquecidos a 4,3%. Trabalhando com uma potência máxima de 100 W.

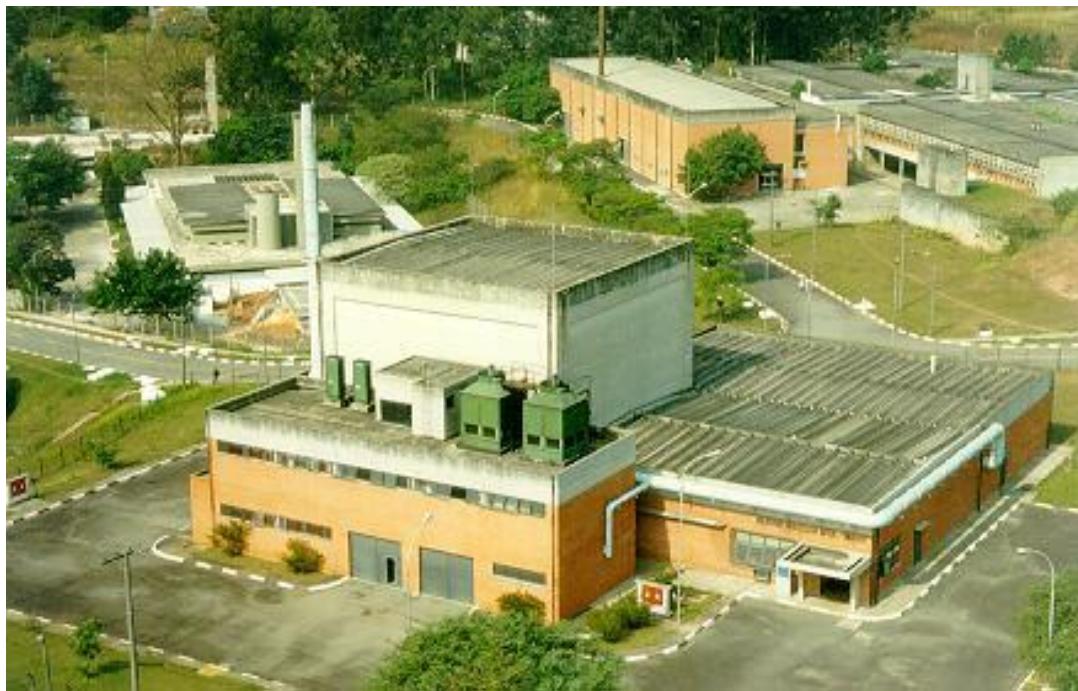


Figura 9 – Prédio do Reator IPEN/MB-01 <fonte: IPEN, 2015>

É uma instalação nuclear que permite a simulação de todas as características nucleares de um reator de grande porte em escala reduzida, sem que haja a necessidade de construir-se um complexo sistema de remoção de valor. Este tipo de reator é conhecido mundialmente como reator de potência zero ou de Unidade Crítica.

Apresentam como ferramenta básica, o fato de permitir que pesquisadores possam estudar não apenas por cálculos teóricos, mas também por medidas experimentais, o desempenho e as características do núcleo de um reator de potência, antes mesmo de sua efetiva instalação, simulando assim as condições de projeto na própria instalação.

RMB

De acordo com a notícia publicada no site da FINEP (Financiadora de Estudos e Projetos) o Reator Multipropósito Brasileiro (RMB), é um empreendimento a cargo da Comissão Nacional de Energia Nuclear (CNEN), que deverá entrar em funcionamento em 2018, no município de Iperó, no interior de São Paulo. É um reator nuclear de pesquisa multipropósito que tem como uma de suas finalidades a produção de radioisótopos, que são a base para os

radiofármacos utilizados na Medicina Nuclear e para produção de fontes radioativas usadas em aplicações na indústria, na agricultura e no meio ambiente.

2.2 INSTRUMENTAÇÃO

Instrumentação é qualquer dispositivo (instrumento), ou conjunto de dispositivos, utilizado com a finalidade de se medir, indicar, registrar ou controlar as variáveis de um processo. (STARLING, 2003). Ou seja, a instrumentação é a ciência que desenvolve e aplica técnicas de medição, indicação, registro e controle de processos de fabricação, visando a otimização e eficiência destes processos.

Há muitas definições para processo, (STARLING, 2003) duas delas são:

- “Equipamento ou meio físico que precisa ser controlado ou monitorado de forma a transformar a matéria-prima em um produto”.
- “Operação onde se varia pelo menos uma característica física ou química de um determinado material”.

Através do uso da instrumentação é possível obter algumas vantagens, como por exemplo:

- As informações de uma planta centralizadas em uma “sala de controle”;
- Manipulação de plantas mais complexas;
- Operação de plantas de alto risco com mais segurança;
- A segurança de pessoas e equipamentos.

O desenvolvimento tecnológico da instrumentação ao longo dos anos, proporcionou grandes evoluções nos métodos e formas de operação das usinas nucleares.

A instrumentação e sistemas de controle das mesmas, desempenham um papel fundamental, tendo as principais funções: o fornecimento de informações sobre a planta para a equipe de operação, de forma que a usina possa ser operada com segurança e eficiência; processamento dos comandos dos operadores da planta, fazendo com que a mesma se mantenha em um estado seguro ou traze-la de volta para um estado, após o início de uma das condições de acidentes ou eventos do projeto de base. (KHALLEP, 2014)

De acordo com (OLIVEIRA, 2005) a instrumentação de uma usina nuclear acontece basicamente da seguinte maneira, a sala de controle e os painéis locais de controle estão ligados aos sistemas da usina, que é composto de sensores, mecanismos automáticos e controle de acionamentos. Onde os sensores analógicos e binários informam o estado e a mudança de estado de um sistema, aos mecanismos automáticos de atuação, controle, proteção e monitoração, bem como ao pessoal da sala de controle. Os comandos, necessários para a manutenção da operação requerida de um sistema, chegam aos atuadores através de controles de acionamentos.

O histórico da evolução da instrumentação dos reatores nucleares pode ser resumido através das décadas, onde o processamento do sinais, lógicas de proteção, sequências de controle, e outros fatores foram sofrendo alterações, devido as tecnologias mais antigas se tornarem obsoletas e também por não cumprirem as exigências de regulamentação, que visam aumentar e melhorar a confiabilidade dos reatores, diminuindo assim possíveis acidentes e eventuais erros humanos.

No passado a instrumentação e controle das usinas nucleares acontecia por meio de dispositivos analógicos, com a evolução tecnológica, os sistemas analógicos começaram a ser substituídos gradualmente por sistemas digitais.

A evolução dos sistemas de controle utilizados em usinas nucleares e de reatores de pesquisa poder ser descrita abaixo resumidamente segundo (PINTO, 2010).

A instrumentação na década de 50 e 60 acontecia por meio de válvulas eletrônicas, onde as mesmas realizavam o monitoramento e controle dos sistemas. Por registradores gráficos que visualizam as variáveis e as armazenavam. Já os relés eletromagnéticos realizavam os acionamentos dos equipamentos e a lógica de controle. Sendo que na década de 60 iniciou-se a utilização de monitores de vídeo, os CRT (Cathode Ray Tube), para visualização da operação.

Já nos anos 70, com a evolução das tecnologias, deu-se início o uso das técnicas digitais, através do uso de circuitos integrados.

Um circuito Integrado (CI) é a construção de vários componentes de um circuito (transistores, diodos, resistores, etc) encapsulados em um mesmo invólucro. (MELLO, 1976). Logo os instrumentos de medidas começaram a se tornarem digitais, substituindo assim os sistemas de controle analógicos. A substituição analógica por digital, fez com que os instrumentos de medidas possuíssem mais recursos de memórias, fazendo com que os mesmos fossem programados.

A partir da década de 80, as transformações digitais dos sistemas de proteção e segurança se iniciam. A utilização do computador inicia-se, podendo tanto o hardware e

software atuar no controle dos processos de uma planta. Afim de que houvesse uma comunicação entre os computadores, os mesmos começaram a ser interligados em redes.

Ao final dos anos 90 e início dos anos 2000, o uso de relés para controle das plantas das usinas nucleares, começaram a ser substituídos por CLPs (Controladores Lógicos Programáveis).

Por definição da norma (IEC 61131-1), CLP (figura 10) é um sistema eletrônico digital, desenvolvido para uso em ambiente industrial, que usa uma memória programável para armazenamento interno de instruções do usuário, que realiza funções específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética, para controlar, através de entradas e saídas, vários tipos de máquinas e processos.



Figura 10 – Controlador Lógico Programável (CLP) <fonte: ROCKWELL, 2015>

Quando os relés estavam presentes nas plantas nucleares, os mesmos atuavam apenas no processo final. As mesas de controle faziam a composição de computadores e CLPs, onde era possível realizar o controle através de sistemas supervisórios.

Na atualidade, devido ao grande desenvolvimento tecnológico da eletrônica, quanto da informática, tanto hardware e software, possibilitam uma gama muito grande de recursos disponíveis afim de realizar o controle, monitoração, aquisição das variáveis que compõem um sistema.

E essa constante evolução tecnológica, fez com que os IHMs (Interface Homem-Máquina) também se aprimorassem, de forma a fornecer cada vez mais ao usuário, informações das variáveis com mais precisão e confiabilidade.

A IHM é um dispositivo utilizado para visualizar dados do processo que executem funções de monitoramento, controle de máquinas, processo e instalações industriais. E é através dela que o operador pode interagir com o sistema controlado, exercendo uma função extremamente importante, pois com as informações recebidas da IHM, ele pode realizar intervenções, visando sempre manter a segurança, o bom funcionamento e a disponibilidade de serviços. (PAREDE, 2011)

A figura 11 mostra um exemplo de como é a configuração da instrumentação de uma usina nuclear.

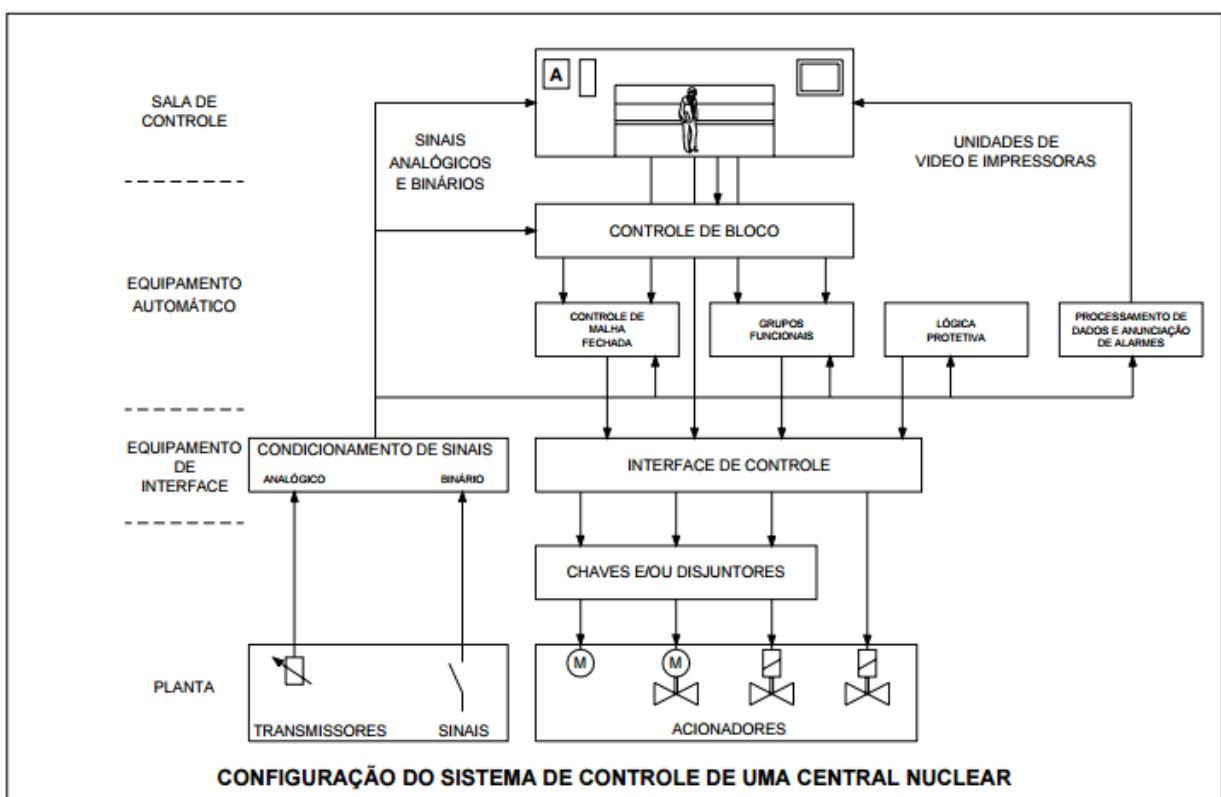


Figura 11 – Exemplo de configuração de sistema de controle de uma central nuclear
<fonte: SBUN, 2001>

A instrumentação e controle que são aplicáveis as usinas nucleares, pode ocorrer de dois tipos:

→ Sistemas baseados em microprocessadores

- *CLP – Controlador Lógico Programável*

→ Sistemas baseados em semicondutores

- *FPGA – Field Programmable Gate Array*
- *ASIC - Application Specific Integrated Circuit*

2.3 ENGENHARIA DE SEGURANÇA

Em projetos de usinas nucleares, um dos conjuntos de critérios gerais de projeto é a segurança, através dela é possível realizar a proteção de indivíduos, sociedade e meio ambiente contra os riscos radiológicos, atendendo assim que as usinas nucleares trabalhem em um alto grau de confiabilidade, visando uma operação contínua e totalmente segura. A engenharia de segurança de um reator pode ser atingida através da adoção de medidas que envolvem os diferentes níveis de projeto, construção, fabricação e operação do reator. Todos essas medidas de segurança são aplicadas na elaboração de um projeto de usina nuclear afim de assegurar a possibilidade de um controle em um eventual acidente ou falha.

- Prevenção - Afim de realizar o aumento da segurança, alguns itens como os requisitos de projeto, fabricação e operação são observados de maneira mais rigorosa.
- Ações Corretivas – Através da sala de controle é possível verificar as indicações e registros de falhas que estejam ocorrendo, e com isso possibilita-se que funções de corretivas de controle sejam atuadas em situações que estejam fugindo do padrão, retornando as condições normais de operação, não necessitando assim da intervenção de ações manuais.
- Controle de Acidentes – Nos sistemas, podem ocorrer “acidentes básicos de projeto”, que são eventos anormais improváveis. Assim, as centrais nucleares devem ser capazes de suportar e controlar.

Considerando a ocorrência de falhas, acidentes e erros, as usinas nucleares possuem dispositivos de segurança que são capazes de identificar a ocorrência dos mesmos, e aplicar medidas que possam controlá-los.

Um sistema de segurança de um reator é composto basicamente de um sistema de proteção, de sistemas de segurança ativos e passivos.

2.3.1 SISTEMA DE SEGURANÇA PASSIVO

Sistema de Segurança Passivos de acordo com o documento (ELETRONUCLEAR, 2005) são sistemas de segurança que não necessitam de ativação externa para serem utilizados. Estes sistemas são normalmente utilizados no âmbito do conceito de defesa de profundidade que consiste no estabelecimento de múltiplos níveis de proteção, através de envoltórios de contenção, barreiras protetoras de concreto e aço; e tanques pressurizados. Protegendo assim, o aumento da pressão interna, liberação de radiação e impactos externos (terremotos, maremotos, inundações e explosões).

2.3.2 SISTEMA DE SEGURANÇA ATIVO

Os sistemas de proteção ativos são aqueles que respondem a forma de operação elétrica e/ou mecânica em sistemas de comando, tem-se como exemplo a atuação de bombas de água de alta pressão.

2.3.3 SISTEMA DE PROTEÇÃO

O Sistema de Proteção de um reator nuclear é uma iniciativa de intervenção que realiza o controle e proteção das variáveis que constituem o processo da planta, de forma a prevenir acidentes. E caso haja alguma anormalidade, o sistema de proteção inicia suas ações de segurança necessárias a fim de evitar acidentes, como o superaquecimento que pode ocasionar a deterioração do núcleo, levando a contaminação do meio ambiente. Com isso, o Sistema de Proteção deve ser hábil em atuar de forma correta quando necessário, evitando atuações desnecessárias, desligando o reator e acionando os sistemas de segurança do mesmo.

Sistema de proteção segundo o documento (NUREG/CR – 5500) é um sistema de controle complexo que comprehende numerosos componentes eletrônicos e mecânicos, que possuem a capacidade de realizar um desligamento rápido de um reator nuclear, sendo ele de forma automática, quanto manual, de acordo com as condições de perturbação existentes na operação de um reator.

Através da utilização da instrumentação nas usinas nucleares, é possível que o sistema de proteção saiba por reconhecimento das variáveis físicas medidas com as variáveis de operação, quando há alguma condição fora do normal, que possa levar a interrupção ou dificuldade em remover o calor do núcleo do reator, bem como se existe a liberação de radiatividade para o meio ambiente.

Quando detectado algumas condições anormais, o sistema precisa ler e executar as ações necessárias para que se atinja os objetivos de proteção.

O desligamento rápido do reator (TRIP) deverá ser ativado pelo sistema de proteção, por um acionamento de forma manual ou automática, fazendo com que os alarmes sejam atuados, impedindo que as barras de controles sejam retiradas e controlando com prioridade os dispositivos de segurança. (COSTA, 2009). De forma que a alta disponibilidade seja cumprida, o TRIP é realizado através de 3 ações independentes e diversas entre si:

- Desligamento dos suprimentos de potência elétrica para as barras de controle;
- Desligamento de todo o suprimento de potência;
- Introdução por queda das barras de controle.

Os principais sinais que provocam o TRIP de um reator, são:

- Desligamento manual;
- Fluxo elevado de nêutrons;
- Pressão elevada;
- Alta taxa de aumento de potência;
- Sobrepotência e sobretemperatura.

2.3.4 SISTEMA DE SEGURANÇA DE UM REATOR DE PESQUISA MODERNO

A segurança inerente do reator é dada pela concepção da piscina aberta, por dispositivos de segurança passivos, bem como o coeficiente de reatividade negativa do núcleo do reator.

O formato de piscina aberta faz com que não seja necessário circuitos de refrigeração pressurizados, reduzindo assim as possibilidades de vazamentos ou outros tipos de falhas da tubulação.

O coeficiente de reatividade negativa assegura que a medida que o núcleo do reator aumenta a potência, a temperatura eleva, e assim o feedback inerente no combustível líquido de arrefecimento realiza a redução da reatividade do núcleo, diminuindo a energia do núcleo. Resultando numa energia gerada pelo núcleo do reator naturalmente auto-limitante.

Um reator de pesquisa moderno possui 2 sistemas de proteção independentes e redundantes. Onde cada um deles possui a capacidade de provocar o desligamento do mesmo, se necessário. Através do monitoramento de variáveis relevantes para a segurança do reatores.

Um dos sistemas de proteção é a inserção das barras de controle no núcleo do reator, estas que caem por gravidade. Já o outro sistema, esvazia o vaso refletor de sua água pesada, esta que também flui por gravidade para um tanque de armazenamento. Uma vantagem destes sistemas de proteção é que os mesmos não se faz uso de energia.

A construção deste tipo de reator atende à todas exigências estabelecidas pela ARPANSA e a AIEA, garantindo robustez e dentro das margens de segurança rígidas.

2.4 SISTEMAS DIGITAIS EM REATORES DE PESQUISA

Os sistemas digitais em reatores de pesquisa são de grande importância na segurança dos mesmos, de acordo com o documento (IAEA, 2015) reatores novos e antigos utilizam esses recursos nas funções de sistemas de controle de processos e sistemas de monitoramento, bem como em aplicações nos sistemas de proteção do reator.

É de suma importância que os sistemas digitais proporcionem uma confiabilidade, sendo está avaliada sistematicamente, documentada e revisada por uma engenharia de processo. Para a instalação de novos softwares, os mesmos devem ser avaliados e mostrar a experiência operando com software pré-existente.

Sendo assim, os modos de falhas de segurança dos dispositivos de segurança e o efeitos que os mesmos podem causar na instrumentação e controle do reator devem ser conhecidos e documentados, de forma a ser considerado em uma avaliação de risco do sistema.

Desde as falhas de softwares de natureza sistemáticas às não aleatórias, o emprego do modo de potencial de falha comum dos sistemas digitais devem ser sistematicamente considerados, baseando-se nos subsistemas de hardware redundantes através da utilização de cópias idênticas de softwares.

Dependendo da complexidade em que os dispositivos experimentais apresentarem no reator de pesquisa, é necessário que se considere ter sistemas digitais separados, tendo como premissa a instrumentação e controle para o reator e para os experimentos. Assim, cada sistema pode ser fornecido com seu próprio conjunto de requisitos e objetivos.

2.5 FPGA (FIELD-PROGRAMMABLE GATE ARRAYS)

As FPGAs de acordo com a National Instruments, são chips de silício reprogramáveis inventados em 1985, por Ross Freeman, cofundador da Xilinx.

Uma FPGA é um dispositivo que semicondutor que é composto por componentes de lógica programável e interconexões programáveis. A configuração dos componentes de lógica programável podem ser realizados de forma a duplicar a funcionalidade das portas lógicas básicas ou de funções simples e combinacionais mais complexas. As FPGAs podem ser programadas e reprogramadas na fase de desenvolvimento pelo cliente e/ou desenvolvedor, de acordo com o que se necessita, e podem ser configuradas para desempenhar uma função específica. (MEIXEDO, 2008)

O uso de FPGA pode ser evidenciado através de suas vantagens como o: paralelismo, tempo de mercado, velocidade de desempenho, flexibilidade, confiabilidade, baixo consumo de energia.

As FPGAs podem ser aplicadas em diversos segmentos, como:

- ASIC Prototipagem: modelagem de sistemas e verificação de software embarcado.
- Áudio: aplicações de imagem e vídeo.
- Industrial: automação industrial e aplicações de imagens industriais.
- Médico: Para aplicações de diagnóstico, monitoramento e terapia.
- Segurança: de controle de acesso aos sistemas de vigilância e segurança.

É constituído de um número finito de recursos de hardware predefinidos, com interconexões programáveis por softwares e implementadas em hardware. Possui também blocos de E/S que permitem que os circuitos acessem o mundo externo. (NATIONAL INSTRUMENTS, 2013)

A sua estrutura interna pode ser ilustrada na figura 12.

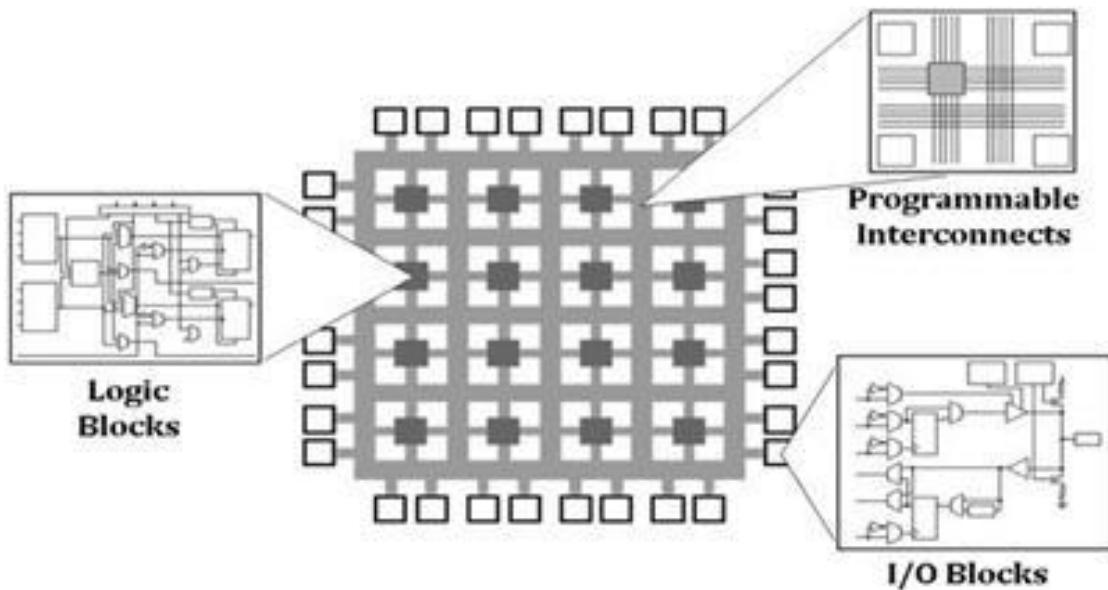


Figura 12 - Modelo estrutural de uma FPGA <fonte: NATIONAL INSTRUMENTS, 2013>

As especificações de um FPGA ne acordo com a (NATIONAL INSTRUMENTS, 2013) geralmente incluem o número de blocos lógicos configuráveis, número de blocos lógicos de funções fixas como multiplicadores e blocos de RAM (tamanho de memória). Entre todas as partes de um chip FPGA, estas geralmente são mais as importantes para uma aplicação.

Os blocos lógicos configuráveis (CLBs – configurable logic blocks) são a unidade lógica básica de uma FPGA. Algumas vezes referido como células lógicas, os CLBs são feitos de dois componentes básicos: flip-flops e lookup tables (LUTs). Várias famílias de FPGAs diferem na maneira com que os flip-flops e LUTs são agrupados, então é importante entender flip-flops e LUTs. (NATIONAL INSTRUMENTS, 2013)

Podemos citar algumas fabricantes de FPGA e de softwares.

- O primeiro fabricante de FPGAs foi Xilinx.
- Logic Rapid: conhecido com suas FPGAs para fins não-gerais programadas utilizando a tecnologia EEPROM.
- Actel (agora Microsemi): líder em programação usando de sinal misto e tecnologia antifuse.
- Altera: com um método de programação baseado em SRAM, tecnologia semelhante como Xilinx.

O ciclo de projeto em FPGA envolve várias etapas, que geralmente são automatizadas. Este processo acontece pelas seguintes etapas e ilustrado figura 13.

1. Especificação e entrada de projetos – etapa em que a entrada de projetos é realizada por meio de um diagrama lógico, fazendo o uso de portas lógicas e macroinstruções. Ou por meio de um editor de texto que use linguagem HDL (Hardware Description Language).
2. Síntese lógica e mapeamento da tecnologia – Consiste em duas fases distintas que é a otimização lógica para minimizar equações booleanas, e também o mapeamento da tecnologia para converter equações em células da biblioteca da tecnologia alvo.
3. Posicionamento e roteamento – é a representação textual de componentes lógicos a serem designados pelos componentes físicos de uma arquitetura FPGA.
4. Verificação e teste – a simulação é feita através de softwares que simulam as restrições e temporização, realizando assim uma verificação funcional.
5. Programação do FPGA – nesta etapa é gerado um arquivo de configuração, que deve ser carregado no dispositivo.

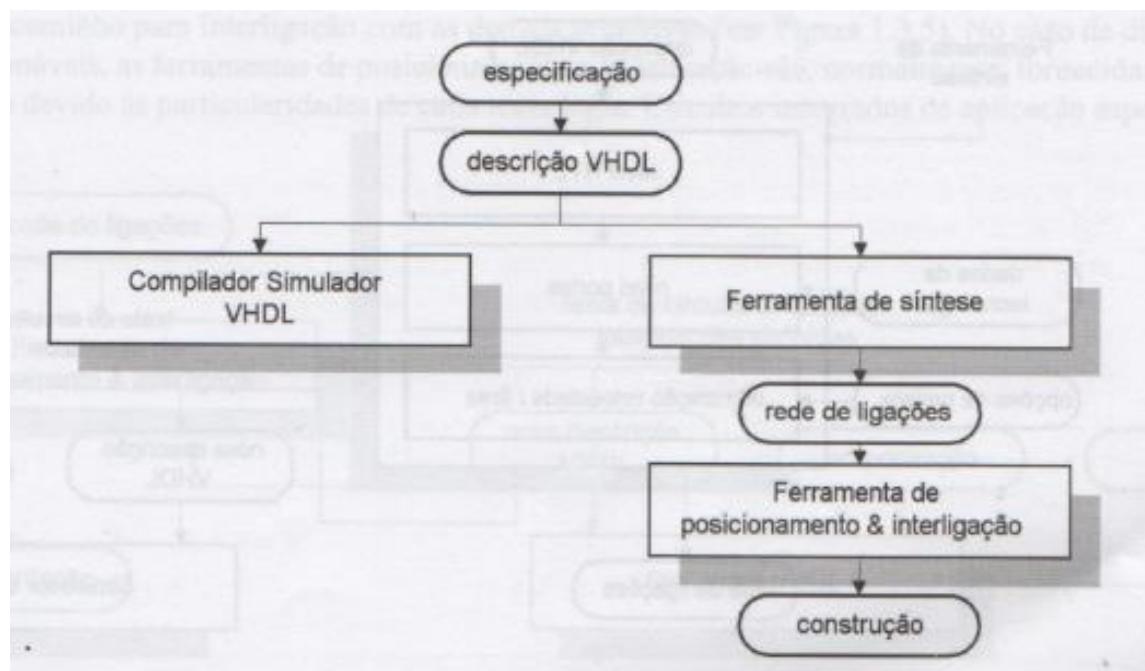


Figura 13 – Etapas de um projeto de uma FPGA <fonte: D'AMORE, 2012>

2.5.1 FPGA em Reatores Nucleares

O uso de FPGA em usinas nucleares segundo a publicação da (BRINC EDITION, 2012), ainda se encontra caminhando em passo lentos, existe um crescente número de aplicações da mesma para o controle e segurança dos reatores.

Em reatores nucleares, de acordo com o artigo (FARIAS et al, 2013) a utilização de FPGAs em centrais nucleares foram introduzidas pela primeira vez em sistemas que não estavam relacionados a segurança, onde nenhum requisito específico regulamentado para aplicações de desenvolvimento com FPGA deveria ser satisfeito. No entanto, para sistemas relacionados com a segurança, onde os processos são mais rigorosos e necessitam de um alto grau de confiabilidade, algumas regras são impostas pelos regulamentadores nucleares. Afim de garantir a confiabilidade e segurança dos sistemas.

A lógica para ser programado em FPGA deve ser descrita utilizando linguagens de design de hardware (HDLs). Porque o processo de desenvolvimento de FPGA é semelhante ao do software dos sistemas baseados em microprocessadores, onde o processo de desenvolvimento de software de segurança convencional possa ser aplicado.

Em sistemas baseados em computadores convencionais, uma separação pode ser estabelecida entre o hardware e partes de software. Mas com as FPGAs, Instrumentação e Controle os designers podem construir funções do aplicativo diretamente em um circuito integrado.

De acordo com a norma (IEC 62566), a alcançar a confiabilidade necessária para os sistemas de segurança de instrumentação e controle, o desenvolvimento de FPGA baseado em sistema, deve cumprir com processos rigorosos e requisitos técnicos impostos pela IEC.

As aplicações de funções de segurança implementadas pelos FPGAs são executados sem correr qualquer software ou sistemas operacionais, o que é uma vantagem. Isto reduz a vulnerabilidade do sistema digital e pode tornar mais fácil o processo de licenciamento. Uma segunda vantagem, FPGA baseia se em aplicações que têm mais resistência devido à portabilidade do código de HDL entre várias versões do FPGA-chips produzidos por diferentes fabricantes.

2.6 VHDL – Very Speed Integrated Circuit

VHDL segundo (BENFICA, 2009) é uma forma de se escrever, através de um programa, o comportamento de um circuito ou componente digital. É uma linguagem de descrição de hardware com ênfase em circuitos integrados de altíssima velocidade.

A linguagem VHDL é capaz de suportar projetos com múltiplos níveis de hierarquia de acordo com (D'AMORE, 2005), onde a descrição pode consistir na interligação de outras descrições menores, a um código que representa o comportamento esperado do circuito.

Através da especificação do projeto, uma descrição VHDL é gerada, onde está é submetida a um simulador e a uma ferramenta de síntese. Na simulação há a verificação da correspondência entre a especificação e o código. Enquanto na ferramenta de síntese há a inferência das estruturas necessárias para um circuito que corresponde à descrição. O resultado dessa etapa é um arquivo contendo uma rede de ligações de elementos básicos disponíveis na tecnologia do dispositivo empregado. Esse arquivo contendo a rede de ligações é a base de dados para a ferramenta que realiza o posicionamento e a interligação dos componentes. A saída da ferramenta de posicionamento e interligação é um arquivo que contém os dados necessários para a confecção do dispositivo. (D'AMORE, 2005)

2.7 V & V (VERIFICAÇÃO E VALIDAÇÃO) – IEC 60880

A introdução de softwares na realização de funções de segurança sempre causou uma grande preocupação, e na área nuclear não poderia ser diferente. Uma vez que os requisitos de segurança são mais exigentes. Para a utilização de novos métodos e critérios, é necessário que os mesmos sejam avaliados quanto a sua segurança e integridade.

Numa definição geral de acordo com (OBERKAMPF et al, 2007), os softwares são amplamente utilizados para resolver problemas e tomar decisões. Onde os usuários passam a acreditar e a trabalhar nos resultados apresentados por eles. Para isto, o software deve ser construído atendendo às especificações do projeto e o produto final deve servir às necessidades reais do usuário. A Verificação e a Validação, ou simplesmente V&V, têm respectivamente estes interesses.

A norma IEC 60880 foi desenvolvida para determinar a utilização de sistemas digitais, bem como sistemas multiprocessados nos sistemas de segurança de usinas nucleares. Logo, para um software ser aplicado em usinas nucleares, ele precisa atender alguns requisitos. Como:

- Defesa contra falhas de causa comum, causada pelo software;
- Ferramentas automatizadas para o desenvolvimento de software importantes para a segurança;
- Utilização de software pré-desenvolvidos.

CAPÍTULO 3 – METODOLOGIA

Este capítulo descreve sobre a metodologia utilizada no desenvolvimento desta dissertação. A proposta do presente trabalho é de um sistema de proteção baseado em FPGA utilizando linguagem VHDL. Para isso, o trabalho foi dividido em algumas etapas.

3.1 INTRODUÇÃO

Estudos preliminares da utilização de controladores digitais microprocessados em sistemas de proteção de um reator de pesquisa foram realizados, de forma a avaliar as possibilidades de implementação de uso dos mesmos. A partir destes estudos, uma proposta de trabalho foi realizada.

A concepção inicial da dissertação seria o desenvolvimento de um sistema de proteção usando CLP (Controlador Lógico Programável), mas devido ao mesmo necessitar que seu software fosse certificado e validado para uso nuclear, a proposta de utilizar FPGA (FIELD Programmable Gate Array) foi tomado forma, devido a possibilidade da mesma ser tratada quase como se fosse um hardware.

A seguir será descrita a metodologia para o projeto de sistemas de proteção baseado em FPGA para um reator de pesquisa para a produção de radioisótopos, semelhante ao reator OPAL da Austrália.

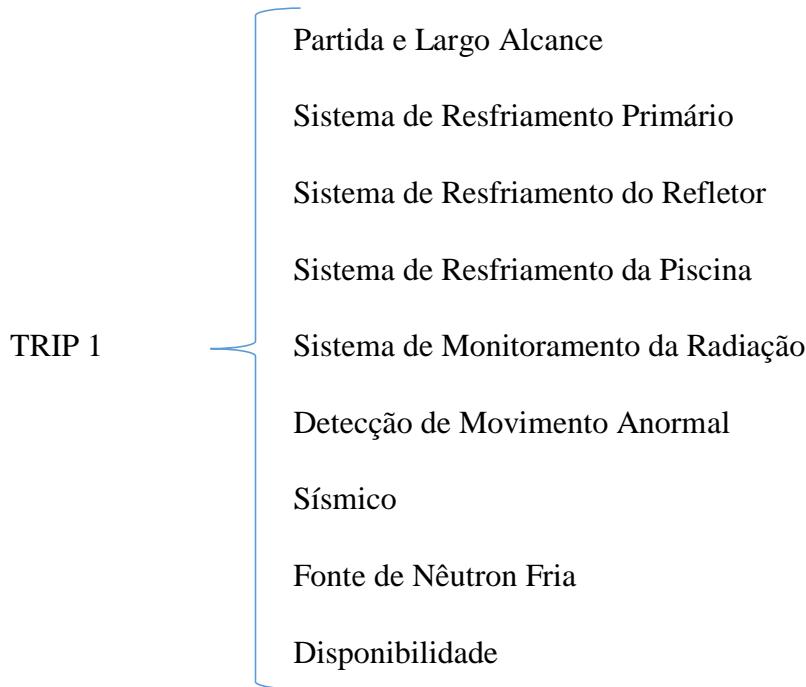
3.2 ETAPAS DA METODOLOGIA

O processo de desenvolvimento e implementação do projeto de sistema de proteção de um reator de pesquisa utilizando FPGA, pode ser dividido nas seguintes etapas:

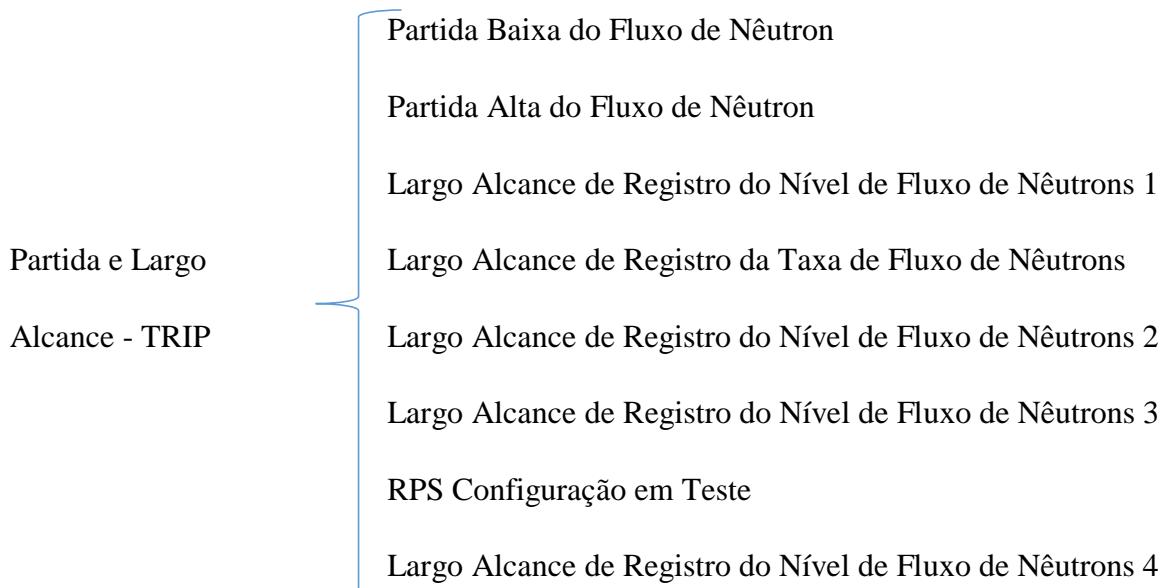
- 1^a Etapa - Descrição dos parâmetros da lógica de desligamento (TRIP);
- 2^a Etapa - Diagrama esquemático da lógica de desligamento (TRIP);
- 3^a Etapa - Modelagem das linguagens de alto nível da lógica de desligamento (TRIP).

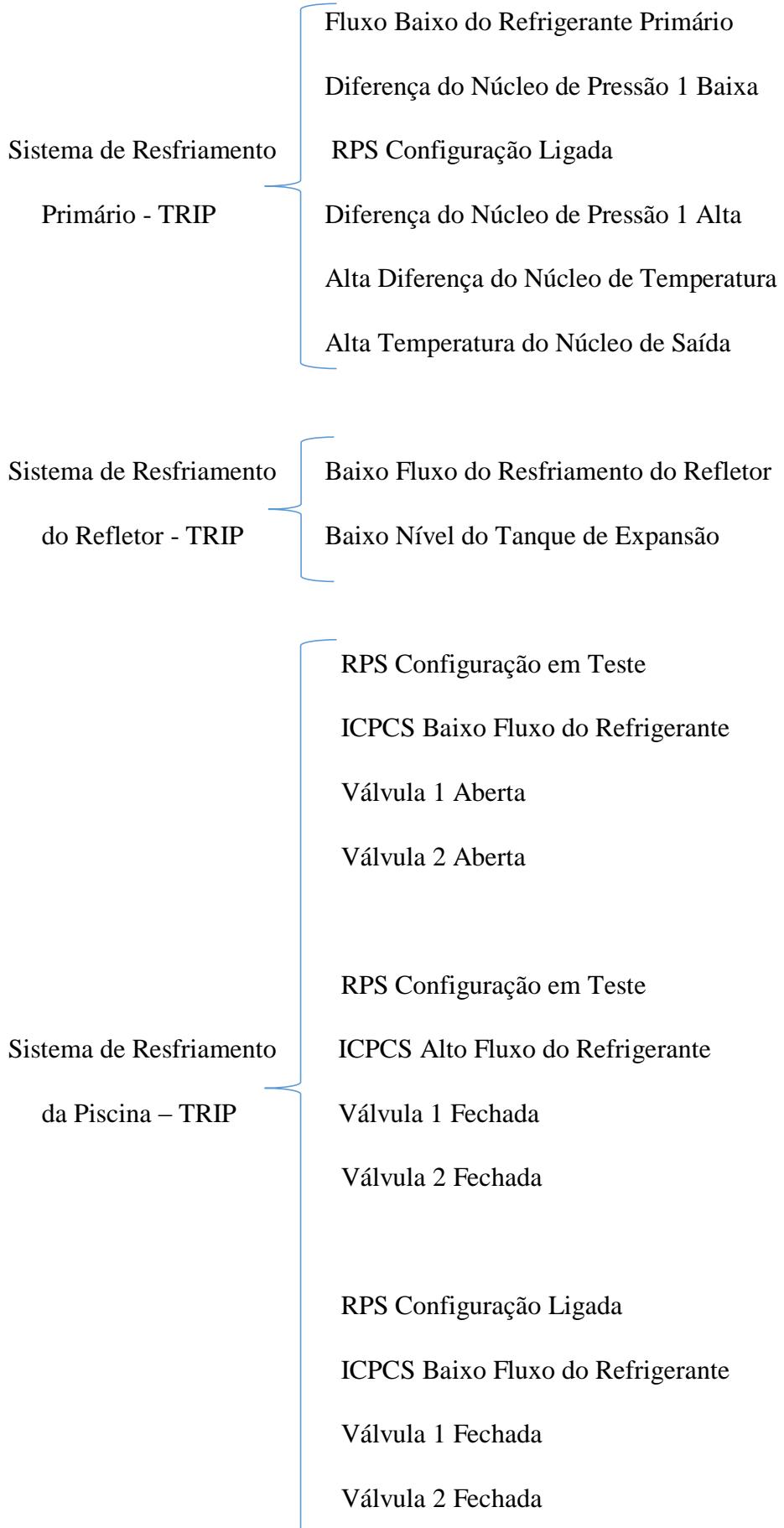
1^a Etapa

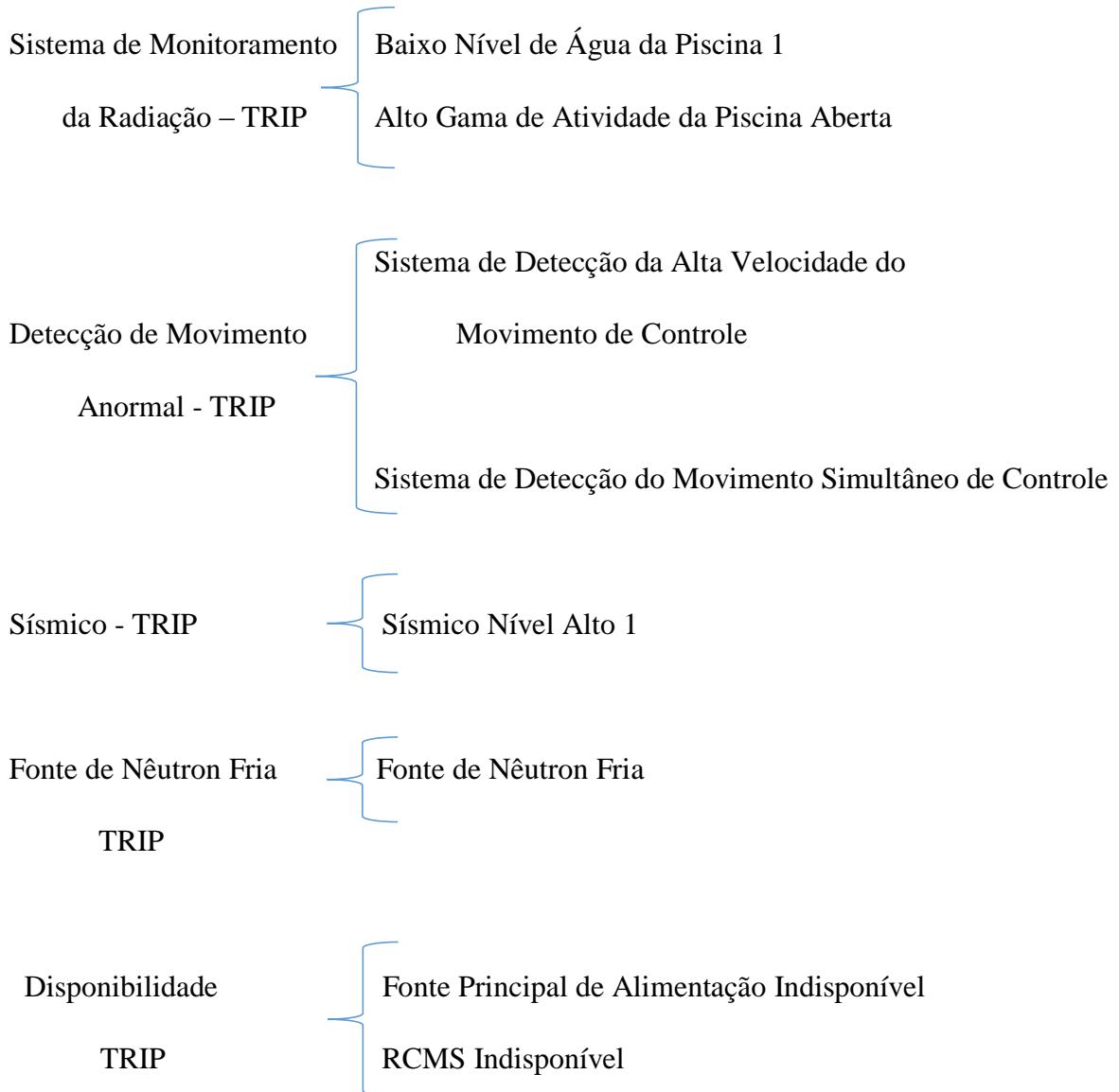
O sistema de TRIP de um reator de pesquisa é composto por um conjunto de 9 lógicas de proteção. As variáveis da lógica de desligamento automático do reator (TRIP) a serem avaliados em referência é a de um reator nuclear de pesquisa moderno. É através da habilitação de uma dessas lógicas que o TRIP do reator acontece.



Cada lógica de proteção é formada por um conjunto de variáveis, que são as entradas dos dados. Abaixo, pode-se ver cada uma dessas lógicas.







2^a Etapa

Nesta etapa as lógicas de TRIP de um reator de pesquisa foram analisadas, em seguida foram extraídas as expressões booleanas de cada uma dessas lógicas, a fim de realizar um diagrama esquemático, utilizando portas lógicas, da lógica de TRIP de um reator de pesquisa. Diagrama este que pode ser visto no Apêndice A.

3^a Etapa

De posse das expressões booleanas retiradas do circuito lógico do TRIP de um reator, esta etapa aconteceu pela modelagem das expressões no software ModelSim da Altera.

Utilizou-se a linguagem de descrição de hardware VHDL, que possui ênfase em circuitos integrados de altíssima velocidade, como já abordado nesta dissertação.

O desenvolvimento da interface das lógicas em VHDL foi realizada em blocos separados, de forma a se obter um melhor entendimento e verificar se o mesmo possuía erros. Quando todos os blocos lógicos foram modelados, realizou-se a união dos mesmos em uma única programação de forma a validar o circuito completo do sistema de lógica do TRIP do reator.

Uma das lógicas de um TRIP de um reator pode ser vista na figura 14, as demais se encontram no Apêndice B desta dissertação.

```
Library IEEE;
use IEEE.std_logic_1164.all;

Entity Start_up_Wide_Range is
Port(  SNFL : in bit;
        SNFH : in bit;
        WRLNFL1 : in bit;
        WRLNFR : in bit;
        WRLNFL2 : in bit;
        WRLNFL3 : in bit;
        RPSCT : in bit;
        WRLNFL4 : in bit;
        SWRTR : out bit);
End Start_up_Wide_Range;

Architecture logica of Start_up_Wide_Range is
Begin
    SWRTR <= (not WRLNFL1 and (SNFL or SNFH)) or
(WRLNFR and WRLNFL2) or (WRLNFL4 or (WRLNFL3 and RPSCT));
End logica;
```

Figura 14 – Exemplo de uma das lógicas de TRIP de um reator <fonte: própria, 2016>

CAPÍTULO 4 – PROJETO DO SISTEMA

Este capítulo descreve como a metodologia apresentada foi utilizada para desenvolver um sistema de proteção para um reator de pesquisa, utilizando FPGA. Vale ressaltar quer a maioria dos detalhes e dados apresentados são idênticos a outros reatores de pesquisa. Neste capítulo também, será abordado um estudo de caso no processo de V & V (Verificação e Validação) para aplicações de software na área nuclear. Bem como a apresentação de algumas hipóteses de simulações do sistema proposto.

4.1 ESTUDO DE CASO

O objetivo principal do teste de V & V (Verificação e Validação) consiste em assegurar que o software está rodando de forma correta, estabelecendo confiança. Logo, a qualidade do software é garantida de acordo com as conformidade dos requisitos funcionais e de desempenho, a documentação dos padrões de desenvolvimento e também as características esperadas do software com relação a corretitude e confiabilidade.

Basicamente a Verificação acontece de forma a garantir a consistência e corretitude do software fase a fase de seu ciclo. Já o processo de Validação visa garantir que o software atende aos requisitos do usuário. Testes de V & V são complementares, com objetivos e naturezas distintas, mas são complementares uma a outra, contribuindo para a detecção de erros, e por consequência aumentando a qualidade do produto.

O processo de V & V acontece por meio de:

- Inspeções de softwares – onde se realiza a aplicação de técnicas estáticas (sistema não precisa ser executado). Consiste em verificar as representações do sistema, documentação, diagrama e código-fonte.
- Teste de software – técnica dinâmica, onde a execução de uma implementação do software proposto é feita com dados de teste, e assim executar uma saída e comportamento operacional afim de verificar se o software está sendo executado conforme o esperado.

Inspeções de softwares e teste de softwares são técnicas que são executadas em conjunto, como mostrado na figura 15, e devem ocorrer durante e depois do desenvolvimento de aplicações de softwares.

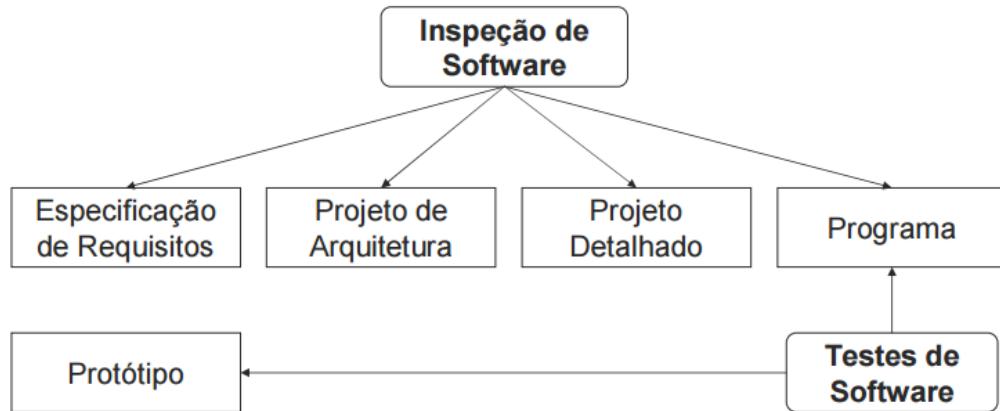


Figura 15: Processo de V & V de software <fonte: FIGUEIREDO, 2015>

Através do teste funcional, é possível verificar se as funcionalidades se comportam de acordo com as documentações de especificação funcional, não existindo diferenças entre os requisitos funcionais e o comportamento da aplicação. Com a atribuição de valores de entrada e saída é possível avaliar o comportamento do software.

Já no teste de desempenho, é possível verificar o tempo de resposta de uma determinada situação está de acordo com o tempo especificado nos requisitos definidos.

O teste de configuração verifica se a aplicação funciona como esperado em ambientes diferentes de softwares e hardwares.

No teste de integridade, verifica-se a resistência à falhas do objeto de teste, robustez, bem como a integridade dos dados armazenados.

A aplicação do processo de V & V pode ocasionar vantagens e desvantagens, como as listadas abaixo:

Vantagens

- Análise detalhada da especificação;
- Revela inconsistência ou omissões em potencial;
- Descoberta de diferentes defeitos em uma única inspeção;
- Inspeção de versões incompletas do sistema proposto.

Desvantagens

- Só podem ser utilizadas por pessoas especialmente treinadas;
- Problemas com requisitos podem ficar ocultos pela formalidade;
- De acordo com o aumento do sistema, os custos de verificação crescem.

4.1.1 NECESSIDADE DO TESTE DE V & V

Haja visto a possibilidade do programador cometer erros, estes erros resultará em falhas não detectadas no sistema. E assim, o sistema pode ser instalado com falhas.

- Erro – é o resultado de uma ação humana incorreta ou processo que produz um resultado involuntário.
- Falha – uma falha de software é o resultado de um erro no código de dados. Uma falha no sistema pode ser o resultado de um erro de design de hardware, uma falha de hardware ou uma falha de software e uma trajetória de sinal associado, resultando um uma insuficiênciа.

No desenvolvimento de um novo software, as falhas podem ser introduzidas em todas as fases do processo. O primeiro elemento de verificação é encontrar e corrigir os erros o mais rápido possível no ciclo de vida de um software. O segundo elemento de verificação é fornecer que todas as falhas de softwares que possa vir a atrapalhar o desempenho do sistema tenham sido retiradas. E também, para os sistemas de segurança, é preciso que os requisitos de sistemas que forem atendidos devem ser identificados.

A validação fornece a demonstração e garantia de que o software e o sistema cumpriam todos os requisitos especificados.

Quando bem sucedido, os teste de V & V resultam numa redução significativa no número de falhas do software, para um mínimo tolerável.

Depois de todos esses passos, é necessário que haja uma prova documental clara, afim de demonstrar que o software possui alta garantia de desempenho em serviço, tornando-o aceitável.

4.1.2 TIPOS DE SOFTWARES

- Software novo: todo software escrito especificamente para a aplicação;

- Software existente acessível: normalmente software de um aplicativo semelhante que é para ser reutilizado, e para o qual toda a documentação está disponível.
- Software proprietário existente: produto comercial ou software de outro aplicativo que atenda todos ou alguns dos requisitos do atual, mas para o qual pouca documentação está disponível;
- Softwares configuráveis: normalmente software que já existem, mas está configurado para aplicações específicas, usando dados ou uma entrada de uma específica linguagem.

4.2 TESTES

A utilização de FPGA se encaixa em softwares configuráveis. Logo a sua validação acontece em duas partes: a validação do software base e a validação da configuração de dados.

A validação de todas as funções contidas no FPGA foram realizadas com sucesso, possibilitando a simulação e testes de todas as lógicas implementadas no FPGA.

O presente trabalho foi revisado, compilado e simulado, como será mostrado a seguir.

4.3 IMPLEMENTAÇÃO

Com os diagramas de blocos das lógicas de TRIP de um reator, foi preciso realizar a conversão das mesmas em códigos VHDL, a fim de simulá-las no software ModelSim. A simulação foi realizada em modo interativo.

Na simulação, de forma a testar a validação funcional da modelagem, foi atribuído níveis lógicos nas entradas dos parâmetros e assim verificar se os níveis lógicos das saídas, ou seja, se cada segmento, estão de acordo com a lógica do projeto. Ressaltando que para realizar este procedimento de teste, é necessário que o mesmo seja definido, o que se denomina em geral *testbench*.

Das lógicas de TRIP de um reator apresentadas no presente trabalho, duas delas foram testadas e simuladas em separado, e em duas situações diferentes de atribuições de valores.

- Partida e Largo Alcance
- Sistema de Resfriamento Primário

A modelagem em VHDL dos testes (testbench) das lógicas de TRIP se encontram no Apêndice C.

TestBench: Partida e Largo Alcance

A simulação da situação 1 para a o TestBench da lógica de TRIP - Partida e Largo Alcance está representada na figura 16. Já a situação 2 está representada na figura 17.

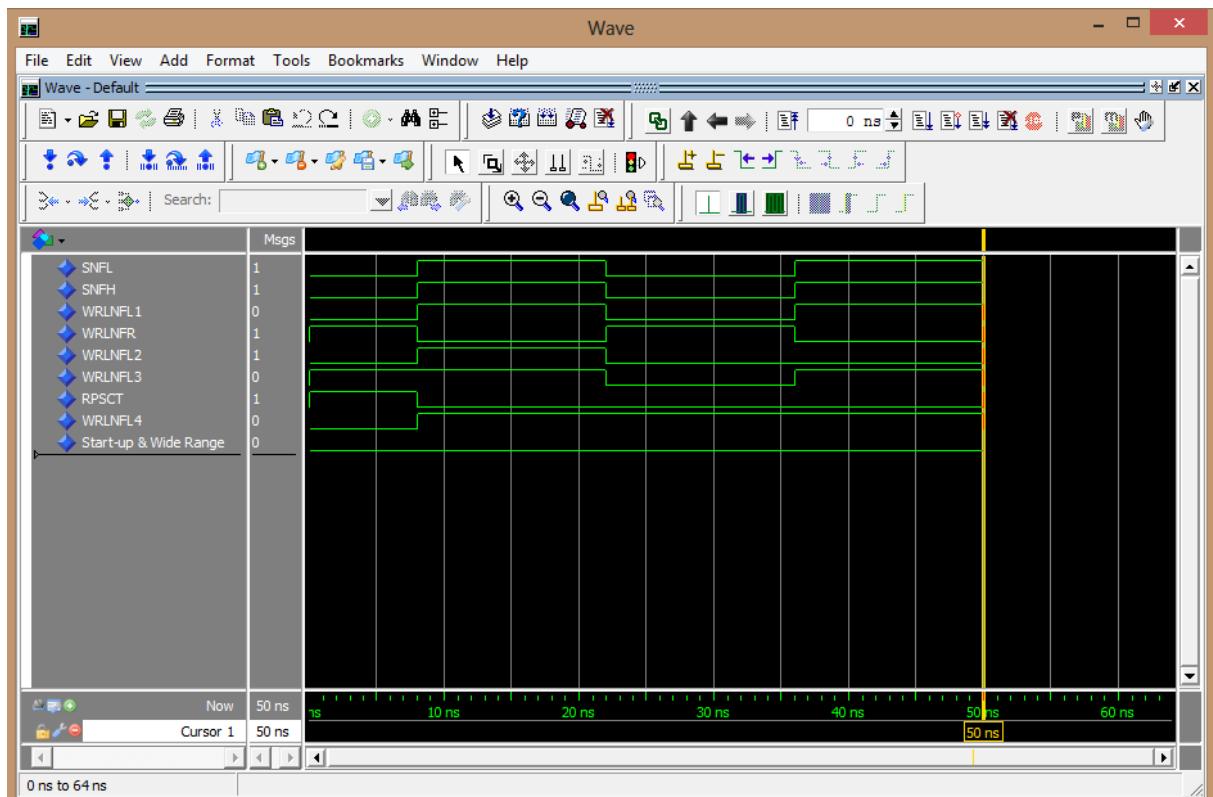


Figura 16: Situação 1 para a Lógica de TRIP - Partida e Largo Alcance
<fonte: Própria, 2016>

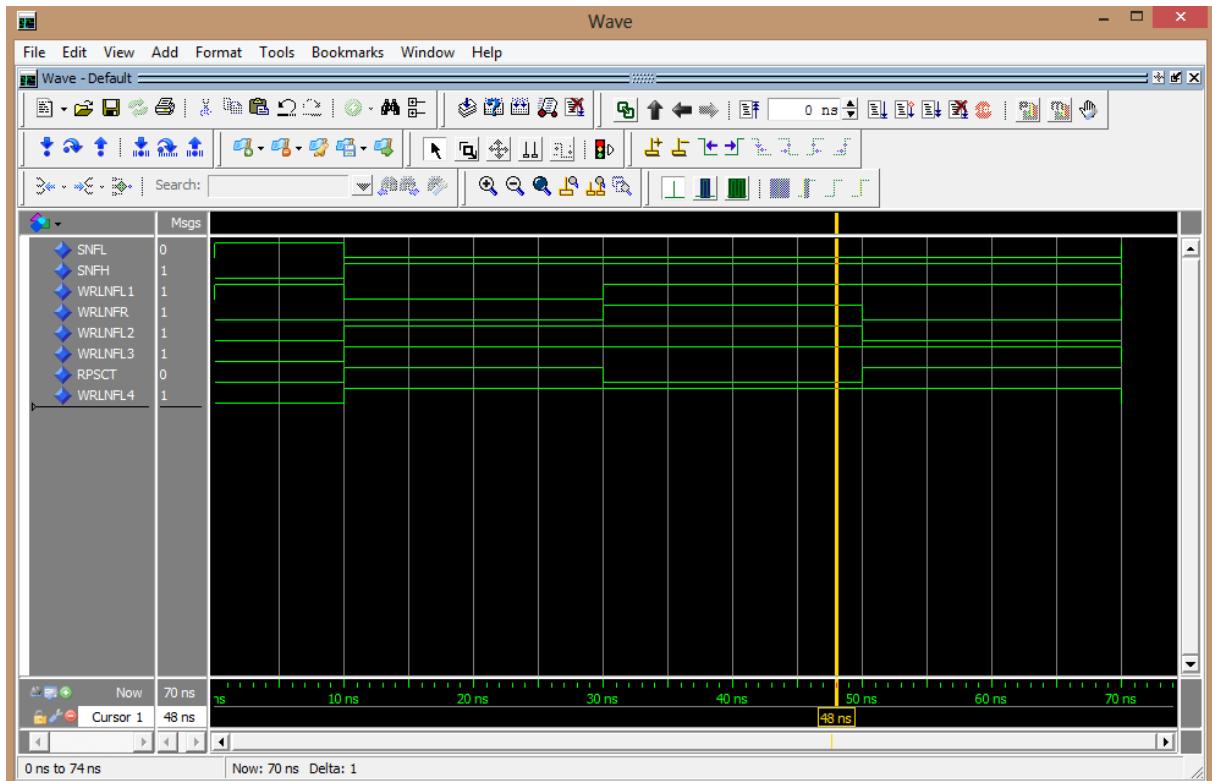


Figura 17: Situação 2 para a Lógica de TRIP - Partida e Largo Alcance
 <fonte: Própria, 2016>

TestBench: Sistema de Resfriamento Primário

A simulação da situação 1 para o TestBench da lógica de TRIP – Sistema de Resfriamento Primário está representada na figura 18. Já a situação 2 está representada na figura 19.

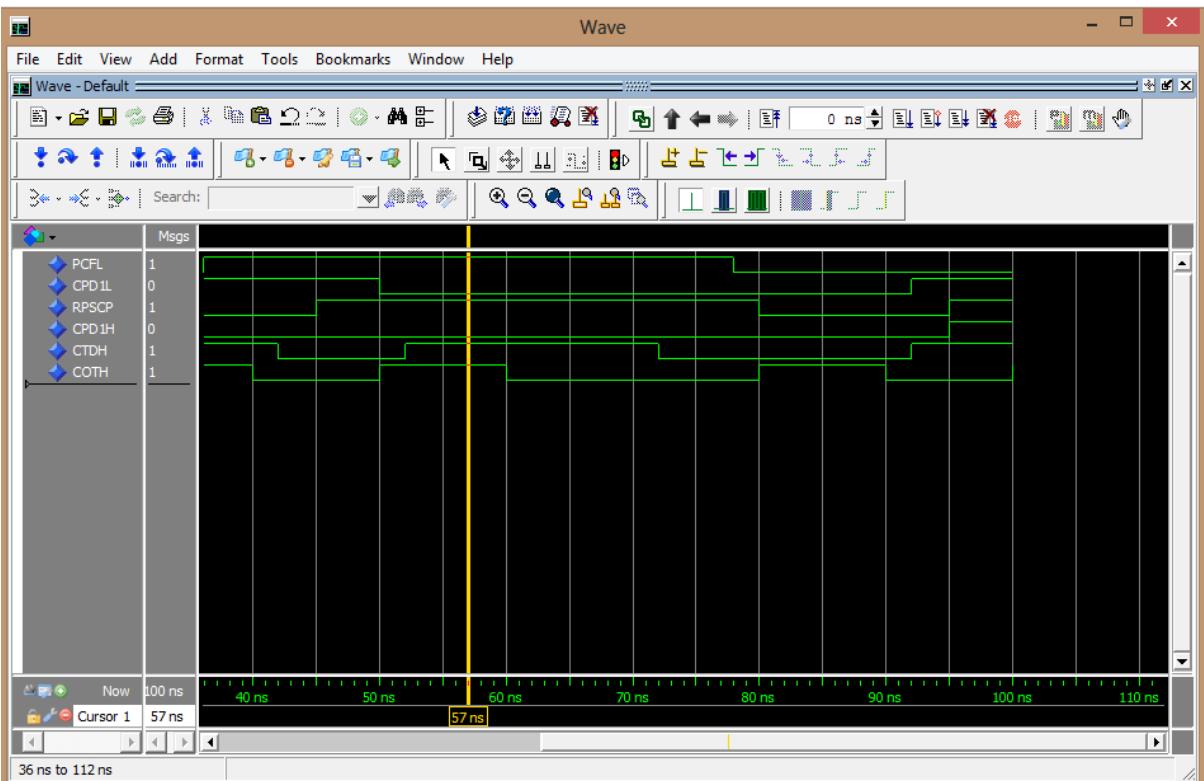


Figura 18: Situação 1 para a Lógica de TRIP – Sistema de Resfriamento Primário
<fonte: Própria, 2016>

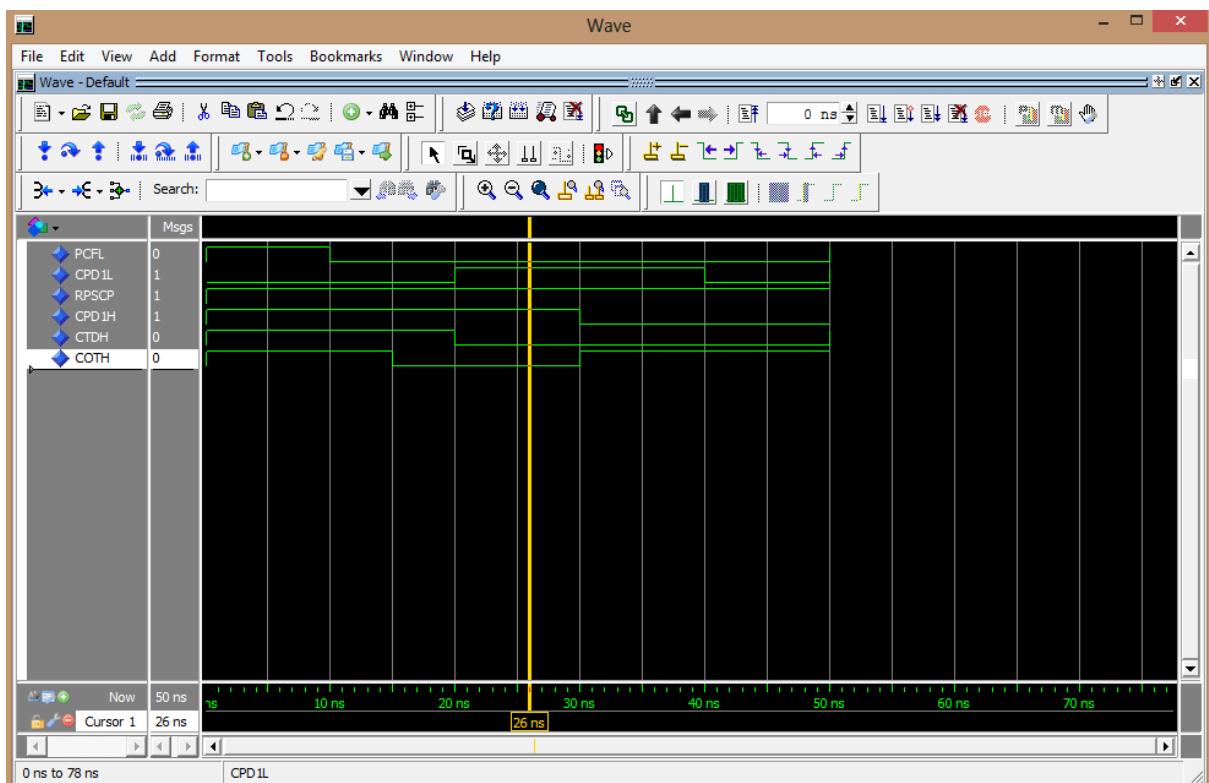


Figura 19: Situação 2 para a Lógica de TRIP - Sistema de Resfriamento Primário
<fonte: Própria, 2016>

CAPÍTULO 5 – CONCLUSÃO

O presente trabalho teve como objetivo apresentar uma proposta de um sistema de proteção baseado em FPGA utilizando linguagem de hardware VHDL. Devido a crescente evolução na produção de hardware e softwares para sistemas embarcados, a ideia deste trabalho surge como uma possibilidade de implementação de sistemas digitais micro controlados na área nuclear.

Para a implementação de qualquer software na área nuclear, a norma IEC 60880 deve ser tomada como referência para operação. A mesma estabelece requisitos que precisam ser atendidos e seguidos, evidenciando principalmente o processo de V & & (Validação de verificação) de software.

A escolha pela utilização do dispositivo programável FPGA ocorreu por ela apresentar um ciclo de projeto ser reduzido; fácil reconfiguração; não ter custos fixos de projeto e ser de prototipagem rápida. A não escolha do uso do CLP (Controlador Lógico Programável) na proposta do sistema de proteção para reator nuclear de potência, se deu pelo fato do mesmo apresentar um elevado custo de implementação, devido ao processo de verificação e validação.

Ao utilizar a linguagem VHDL na modelagem das FPGAs, permitiu-se o desenvolvimento de um projeto moderno e eficiente, que permitiu realizar a síntese dos circuitos e testes do circuito utilizando vários valores como parâmetros de maneira fácil e ágil. A possibilidade de se realizar modificações e atualização de forma rápida e simples foi observada, bem como a facilidade de se encontrar erros, os quais que podem ser causados pelo programador e também por falha do software.

O processo de modelagem da lógica de TRIP de um reator nuclear de pesquisa ocorreu com a análise do conjunto das lógicas que compõem a mesma. A partir das sequências lógicas, ocorreu a elaboração das mesmas no software ModelSim PE. Software que cumpriu os propósitos para a verificação de uma modelagem de um sistema de proteção, através das lógicas de TRIP de um reator.

Por fim, pelos resultados apresentados, esta proposta de dissertação atingiu a sua meta de servir como uma possibilidade de ferramenta de segurança na criação de um sistema de proteção de um reator de pesquisa.

Os testes do sistema proposto na dissertação, foram realizados por um a um, onde cada lógica de TRIP foi testada em separado, verificando sua funcionalidade. Quando todas

apresentaram um resultado satisfatório, o teste geral com todas as lógicas de TRIP foi realizado. De forma a avaliar condições que satisfizesse o lógica de TRIP.

Vale ressaltar que os testes realizados têm apenas como objetivo demonstrar uma prova de conceito, logo a modelagem proposta pode ser base de um possível desenvolvimento de trabalho futuro.

Como propostas para trabalhos futuros, de acordo com os resultados alcançados, que foi a modelagem, estudo de V & V, e as simulações realizadas, o sistema desenvolvido poderá futuramente:

- Ser implementado no hardware FPGA;
- Através dos hardwares de FPGAs programados, a criação de uma lógica de redundância, de forma a introduzir maior confiabilidade ao sistema proposto.

REFERÊNCIAS BIBLIOGRÁFICAS

ABEN – Associação Brasileira de Energia Nuclear. **Informativo da Associação Brasileira de Energia Nuclear.** Nº 40. pg 8-11. 2013

ANSTO - Australian Nuclear Science and Technology Organization. Reactor OPAL, 2015. Disponível em: <<http://www.ansto.gov.au/ResearchHub/Bragg/Facilities/OPALReactor>>. Acesso em: 14 dez. 2015.

BENFICA, Juliano D'Ornelas. **Linguagem de descrição de hardware VHDL.** Porto Alegre, 2009

BRINC EDITION. **Modern Power Systems.** Nº 8, 20012

COSTA, Rafael Gomes. **Sistema de auxílio para o direcionamento da atenção no diagnóstico de acidente em usinas nucleares baseado em inteligência artificial.** Rio de Janeiro, CNEN/IEN, 2009.

D'AMORE, Roberto. **VHDL – Descrição e Síntese de Circuitos Digitais – 2^a ED.** Editora LTC. 2012

ELETRONUCLEAR. **Subsídio Segurança Nuclear: Principais organizações envolvidas.** 2005

FARIAS, Marcos Santana; OLIVEIRA, Mauro Vitor; JAIME, Guilherme Dutra Gonzaga; ALMEIDA, José Carlos Soares; AUGUSTO, Silas Cordeiro. **Modeling the critical safety functions status tree of a NPP using FPGA.** INAC, 2013

FIGUEIREDO, Eduardo. **Verificação e Validação.** Universidade Federal de Minas Gerais – UFMG, 2015

GUERRINI, B.; PACI, Sandro. **Lessons of Nuclear Plants. Part II B: Advanced Reactors.** Universita di Pisa, 1999.

IAEA - INTERNATIONAL ATOMIC ENERGY AGENCY. Instrumentation and control systems and software important to safety for research reactors. — Vienna, 2015.

IAEA - INTERNATIONAL ATOMIC ENERGY AGENCY. Operational limits and conditions and operating procedures for research reactors. IAEA Safety Standards Series, DS261, Vienna, 2006.

IAEA - INTERNATIONAL ATOMIC ENERGY AGENCY. Safety of Research Reactors. Safety Standards Series. NS-R-4, Vienna, 2005.

IEC - International Electrotechnical Commission. IEC 61131-3. Controladores Programáveis. 1992

IEC – International Electrotechnical Commission. IEC 66880 Software for computers in the safety systems of nuclear power stations. Suiça, 1986

IEC - INTERNATIONAL ELECTROTECHNICAL COMMISSION. Nuclear power plants – Instrumentation and control important to safety – Development of HDL-programmed integrated circuits for systems performing category a functions. 2012

IEN – Instituto de Engenharia Nuclear. O Reator Argonauta do Instituto de Engenharia Nuclear 50 anos. Rio de Janeiro, 2012.

<http://www.ien.gov.br/noticias/noticias_arquivo/reator_exposicao_50anos.pdf>
Acesso em: 03/09/2015

KHALLEP, Muhammmad Tahir. Advancement in Intrumentation and Control Systems of Nuclear Power Plants and Reactor safety. Paquistão, 2014

MEIXEDO, João Miguel Ramos. Metodologias de projeto de baixo consumo para implementação em FPGA. FEUP – Faculdade de Engenharia da Universidade do Porto. Porto, 2008

MELLO, Hilton Andrade. Circuitos Integrados. Editora Edgard Blucher. São Paulo, 1976.

NATIONAL INSTRUMENTS. **Fundamentos da tecnologia FPGA**. 2013

< <http://www.ni.com/white-paper/6983/pt/>>

Acesso em: 22/02/2016

NRC - Nuclear Regulatory Commission. **NUREG/CR-5500. vol. 2**. Washington, USA, 1998.

OBERKAMPF, W.L., TRUCANO, T.G.. **Verification and validation benchmarks, Nuclear Eng. Design, Validation and Uncertainty Estimation Department**. Sandia National Laboratories, 2007.

OLIVEIRA, Carlos Henrique. **Avaliação da confiabilidade do sistema de limitação de Angra 2 para a função de desarme do reator em um cenário de perda total do sistema de água de alimentação principal**. COPPE, Rio de Janeiro, 2005

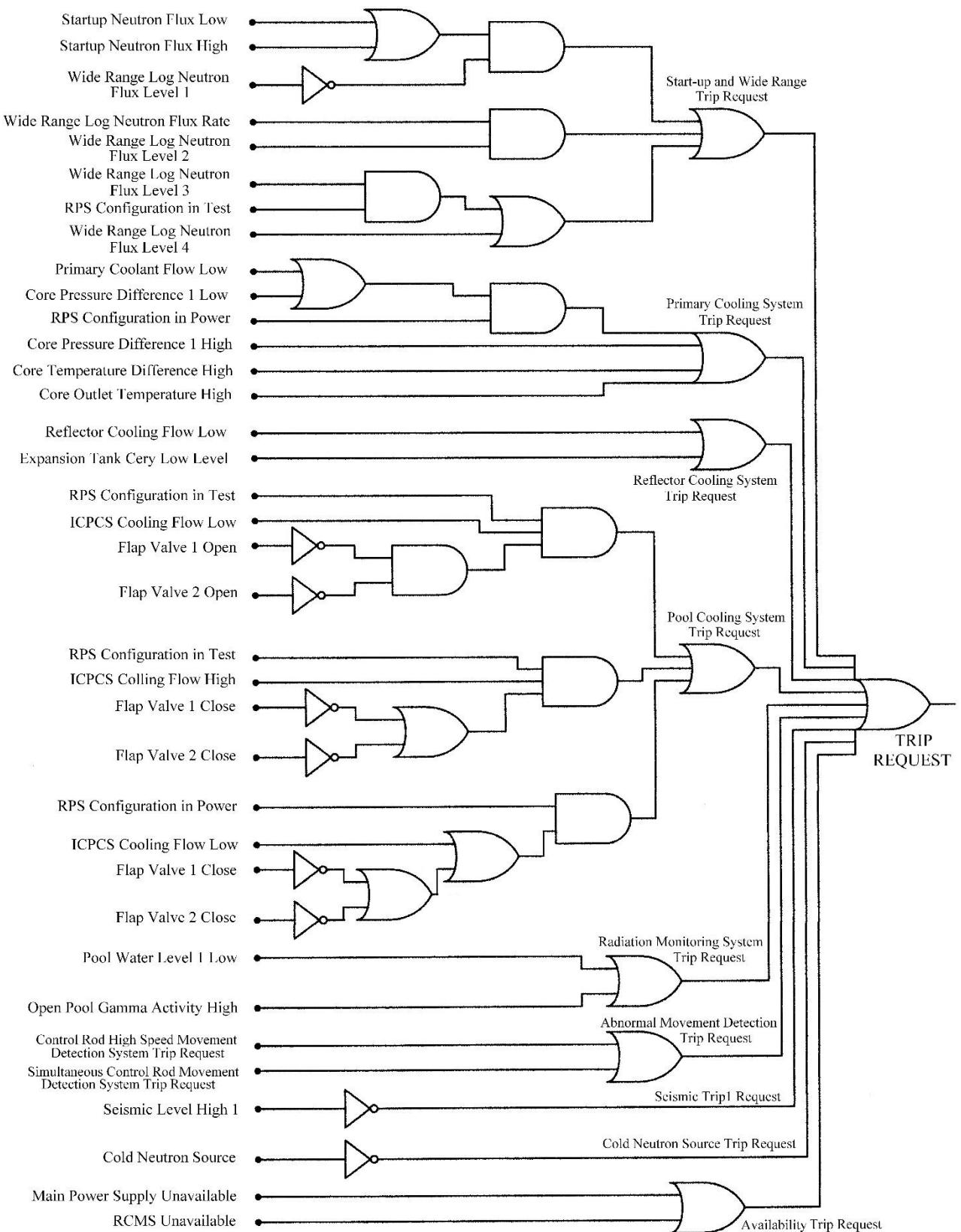
PAREDE, Ismael Moura; GOMES, Luiz Eduardo Lemes; HORTA, Edson. **Eletrônica: automação industrial**. São Paulo: Fundação Padre Anchieta, 2011 (Coleção Técnica Interativa. Série Eletrônica, v. 6)

PINTO, Antônio Juscelino Pinto. **Sistema digital para simulação de parâmetros neutrônicos do reator de pesquisa TRIGA IPR – R1**. Belo Horizonte, 2010

SBUN - Sistema Básico de Usinas Nucleares. **Eletrouclear**. Rio de Janeiro, 2001.

STARLING, Antônio N. **Controle e automação I: introdução a instrumentação industrial**. Belo Horizonte, 2003

APÊNDICE A: Diagrama Lógico da Lógica de TRIP de um Reator de Pesquisa



APÊNDICE B: Linguagem de Descrição de Hardware das lógicas de TRIP de um reator de pesquisa

Lógica: Partida e Largo Alcance

```
-- ****  
-- Circuito: * Porta OU (OR) de duas entradas:  
    -- SNFL - Entrada: Startup Neutron Flux Low  
    -- SNFH - Entrada: Startup Neutron Flux High  
    -- S1 - Saída: SNFL + SNFH  
-- * Porta E (AND) de duas entradas:  
    -- S1 - Entrada: SNFL + SNFH  
    -- WRLNFL1 - Entrada: Wide Range Log Neutron Flux Level 1  
    -- S2 - Saída: (SNFL + SNFH) . NOT WRLNFL1  
-- * Porta E (AND) de duas entradas:  
    -- WRLNFR - Entrada: Wide Range Log Neutron Flux Rate  
    -- WRLNFL2 - Entrada: Wide Range Log Neutron Flux Level 2  
    -- S3 - WRLNFR . WRLNFL2  
-- * Porta E (AND) de duas entradas:  
    -- WRLNFL3 - Entrada: Wide Range Log Neutron Flux Level 3  
    -- RPSCT - Entrada: RPS Configuration in Test  
    -- S4 - Saída: WRLNFL3 . RPSCT  
-- * Porta OU (OR) de duas entradas:  
    -- S4 - Entrada: WRLNFL3 . RPSCT  
    -- WRLNFL4 - Entrada: Wide Range Log Neutron Flux Level 4  
    -- S5 - Saída: (WRLNFL3 . RPSCT) + WRLNFL4  
-- * Porta OU (OR) de três entradas:  
    -- S2 - Entrada: (SNFL + SNFH) . NOT WRLNFL1  
    -- S3 - Entrada: WRLNFR . WRLNFL2  
    -- S5 - Entrada: (WRLNFL3 . RPSCT) + WRLNFL4  
    -- SWRTR - Saída: Start-up & Wide Range Trip Request  
-- ****
```

```
Library IEEE;
use IEEE.std_logic_1164.all;

Entity Start_up_Wide_Range is
Port(  SNFL : in bit;
        SNFH : in bit;
        WRLNFL1 : in bit;
        WRLNFR : in bit;
        WRLNFL2 : in bit;
        WRLNFL3 : in bit;
        RPSCT : in bit;
        WRLNFL4 : in bit;
        SWRTR : out bit);
End Start_up_Wide_Range;
```

Architecture logica of Start_up_Wide_Range is

```
Begin
    SWRTR <= (not WRLNFL1 and (SNFL or SNFH)) or (WRLNFR and WRLNFL2) or
(WRLNFL4 or (WRLNFL3 and RPSCT));
End logica;
```

Lógica: Sistema de Resfriamento Primário

-- Circuito: * Porta OU (OR) de duas entradas:
 -- PCFL - Entrada: Primary Coolant Flow Low
 -- CPD1L - Entrada: Core Pressure Difference 1 Low
 -- S1 - Saída: PCFL + CPD1L

-- * Porta E (AND) de duas entradas:
 -- S1 - Entrada: PCFL + CPD1L
 -- RPSCP - Entrada: RPS Configuration in Power
 -- S2 - Saída:(PCFL + CPD1L) . RPSCP

-- * Porta OU (OU) de quatro entradas:
 -- S2 - Entrada: RPSCP (PCFL + CPD1L)
 -- CPD1H - Entrada: Core Pressure Difference 1 High
 -- CTDH - Entrada: Core Pressure Difference High
 -- COTH - Entrada: Core Outlet Temperature High
 -- PRCSTR - Saída: Primary Cooling System Trip Request

-- *****

```
Library IEEE;  
use IEEE.std_logic_1164.all;
```

```
Entity Primary_Cooling_System is  
Port( PCFL : in bit;  
     CPD1L : in bit;  
     RPSCP : in bit;  
     CPD1H : in bit;  
     CTDH : in bit;  
     COTH : in bit;  
     PRCSTR : out bit);
```

```
End Primary_Cooling_System;
```

```
Architecture logica of Primary_Cooling_System is
```

Begin

PRCSTR <= ((PCFL or CPD1L) and RPSCP) or CPD1H or CTDH or COTH ;

End logica;

Lógica: Sistema de Resfriamento do Refletor

```
-- ****
-- Circuito: Porta OU (OR) de duas entradas:
-- RCFL - Entrada: Reflector Cooling Flow Low
-- ETVLL - Entrada: Expansion Tank Very Low Level
-- RCSTR - Saída: Reflector Cooling System Trip Request
-- ****
```

```
Library IEEE;
use IEEE.std_logic_1164.all;
```

```
Entity Reflector_Cooling_System is
Port( RCFL : in bit;
      ETVLL : in bit;
      RCSTR : out bit);
End Reflector_Cooling_System;
```

```
Architecture logica of Reflector_Cooling_System is
```

```
Begin
      RCSTR <= RCFL or ETVLL;
End logica;
```

Lógica: Sistema de Resfriamento da Piscina

- Circuito: Bloco 1
 - * Porta E (AND) de duas entradas:
 - FV1O - Entrada: Flap Valve 1 Open
 - FV2O - Entrada: Flap Valve 2 Open
 - S1 - Saída: NOT FV1O . NOT FV2O
 - * Porta E (AND) de três entradas:
 - RPSCT - Entrada: RPS Configuration in Test
 - ICPCSCFL - Entrada: ICPCS Cooling Flow Low
 - S1 - Entrada: NOT FV1O . NOT FV2O
 - S2 - Saída: RPSCT . ICPCSCFL . (NOT FV1O . NOT FV2O)
- Bloco 2
 - * Porta OU (OR) de duas entradas:
 - FV1C - Entrada: Flap Valve 1 Close
 - FV2C - Entrada: Flap Valve 2 Close
 - S3 - Saída: NOT FV1C + NOT FV2C
 - * Porta E (AND) de três entradas:
 - RPSCT - Entrada: RPS Configuration in Test
 - ICPCSCFH - Entrada: ICPCS Cooling Flow High
 - S3 - Entrada: NOT FV1C + NOT FV2C
 - S4 - Saída: RPSCT . ICPCSCFH . (NOT F + NOT FV2C)
- Bloco 3
 - * Porta OU (OR) de duas entradas:
 - FV1C - Entrada: Flap Valve 1 Close
 - FV2C - Entrada: Flap Valve 2 Close
 - S5 - Saída: NOT FV1C + NOT FV2C
 - * Porta OU (OR) de duas entradas:
 - ICPCSF - Entrada: ICPCS Cooling Flow Low
 - S5 - Entrada: NOT FV1C + NOT FV2C
 - S6 - Saída: (NOT FV1C + NOT FV2C) + ICPCSF
- * Porta E (AND) de duas entradas:
 - RPSCP - Entrada: RPS Configuration in Power
 - S6 - Entrada: (NOT FV1C + NOT FV2C) + ICPCSCFL

```

-- S7 - Saída: RPSCP . ((NOT FV1C + NOT FV2C) + ICPCSCFL)

-- * Porta OU (OR) de três entradas:
-- S2 - Entrada: RPSCT . ICPCSCFL . (NOT FV1O . NOT FV2O)
-- S4 - Entrada: RPSCT . ICPCSCFH . (NOT FV1C + NOT FV2C)
-- S7 - Entrada: RPSCP . ((NOT FV1C + NOT FV2C) + ICPCSCFL)
-- POCSTR - Saída: Pool Cooling System Trip Request

```

```

Library IEEE;
use IEEE.std_logic_1164.all;

```

```

Entity Pool_Cooling_System is
Port( RPSCT : in bit;
      ICPCSCFL : in bit;
      FV1O : in bit;
      FV2O : in bit;
      ICPCSCFH : in bit;
      FV1C : in bit;
      FV2C : in bit;
      RPSCP : in bit;
      POCSTR : out bit);
End Pool_Cooling_System;

```

```

Architecture logica of Pool_Cooling_System is

```

```

Begin
  POCSTR <= ((not FV1O and not FV2O) and RPSCT and ICPCSCFL) or ((not FV1C
or not FV2C) and RPSCT and ICPCSCFH) or (((not FV1C or not FV2C) or ICPCSCFL) and
RPSCP);
End logica;

```

Lógica: Sistema de Monitoramento da Radiação

```
-- ****
-- Circuito: Porta OU (OR) de duas entradas:
-- PWL1L - Entrada: Pool Water Level 1 Low
-- OPGAH - Entrada: Open Pool Gamma Activity High
-- RMSTR - Saída: Radiation Monitoring System Trip Request
-- ****
```

```
Library IEEE;
use IEEE.std_logic_1164.all;
```

```
Entity Radiation_Monitoring_System is
Port( PWL1L : in bit;
      OPGAH : in bit;
      RMSTR : out bit);
End Radiation_Monitoring_System;
```

```
Architecture logica of Radiation_Monitoring_System is
```

```
Begin
      RMSTR <= PWL1L or OPGAH;
End logica;
```

Lógica: Detecção de Movimento Anormal

```
-- ****
-- Circuito: Porta OU (OR) de duas entradas:
-- CRHSMDSTR - Entrada: Control Rod High Speed Movement Detection System Trip
Request
-- SCRMDSTR - Entrada: Simultaneous Control Rod Movement Detection System Trip
Request
-- AMDTR - Saída: Abnormal Movement Detection Trip Request
-- ****
```

```
Library IEEE;
use IEEE.std_logic_1164.all;
```

```
Entity Abnormal_Movement_Detection is
Port( CRHSMDSTR : in bit;
      SCRMDSTR : in bit;
      AMDTR : out bit);
End Abnormal_Movement_Detection;
```

```
Architecture logica of Abnormal_Movement_Detection is
```

```
Begin
      AMDTR <= CRHSMDSTR or SCRMDSTR;
End logica;
```

Lógica: Sísmico

```
-- ****  
-- Circuito: Apenas um sinal de bit  
-- SLH1 - Entrada: Seismic Level High 1  
-- STR - Saída: Seismic Trip Request  
-- ****
```

```
Library IEEE;  
use IEEE.std_logic_1164.all;
```

```
Entity Seismic is  
Port(  SLH1 : in bit;  
        STR : out bit);  
End Seismic;
```

```
Architecture logica of Seismic is
```

```
Begin  
    STR <= SLH1 ;  
  
End logica;
```

Lógica: Fonte de Nêutron Fria

```
-- ****
-- Circuito: Porta OU (OR) de uma entrada:
-- CNS - Entrada: Cold Neutron Source
-- CNTR - Saída: Cold Neutron Trip Request
-- ****
```

```
Library IEEE;
use IEEE.std_logic_1164.all;
```

```
Entity Trip is
Port( CNS : in bit;
      CNTR : out bit);
End Trip;
```

```
Architecture logica of Trip is
```

```
Begin
      CNTR <= CNS ;
End logica;
```

Lógica: Disponibilidade

```
-- ****
-- Circuito: Porta OU (OR) de duas entradas:
-- MPSU - Entrada: Main Power Supply Unavailable
-- RCMSU - Entrada: RCMS Unavailable
-- ATR - Saída: Availability Trip Request
-- ****
```

```
Library IEEE;
use IEEE.std_logic_1164.all;
```

Entity Availability is

```
Port(  MPSU : in bit;
        RCMSU : in bit;
        ATR : out bit);
```

End Availability;

Architecture logica of Availability is

```
Begin
    ATR <= MPSU or RCMSU;
End logica;
```

Lógica: TRIP Request

```
--*****  
-- Sistema Geral  
--*****
```

```
Library IEEE;  
use IEEE.std_logic_1164.all;
```

```
Entity TRIP_1 is  
Port(  
    SNFL : in bit;  
    SNFH : in bit;  
    WRLNFL1 : in bit;  
    WRLNFR : in bit;  
    WRLNFL2 : in bit;  
    WRLNFL3 : in bit;  
    RPSCT : in bit;  
    WRLNFL4 : in bit;  
    PCFL : in bit;  
    CPD1L : in bit;  
    RPSCP : in bit;  
    CPD1H : in bit;  
    CTDH : in bit;  
    COTH : in bit;  
    RCFL : in bit;  
    ETVLL : in bit;  
    FV1O : in bit;  
    FV2O : in bit;  
    ICPCSCFL : in bit;  
    ICPCSCFH : in bit;  
    FV1C : in bit;  
    FV2C : in bit;  
    PWL1L : in bit;  
    OPGAH : in bit;
```

```
CRHSMDSTR : in bit;
SCRMDSTR : in bit;
SLH1 : in bit;
CNS : in bit;
MPSU : in bit;
RCMSU : in bit;
TRIP_1_REQUEST : out bit);

End TRIP_1;
```

Architecture logica of TRIP_1 is

Begin

```
TRIP_1_REQUEST <= (((not WRLNFL1 and (SNFL or SNFH)) or (WRLNFR and
WRLNFL2) or (WRLNFL4 or (WRLNFL3 and RPSCT))) or (((PCFL or CPD1L) and RPSCP)
or CPD1H or CTDH or COTH) or (RCFL or ETVLL) or ((not FV1O and not FV2O) and
RPSCT and ICPCSCFL) or ((not FV1C or not FV2C) and RPSCT and ICPCSCFH) or (((not
FV1C or not FV2C) or ICPCSCFL) and RPSCP) or (PWL1L or OPGAH) or (CRHSMDSTR
or SCRMDSTR) or (SLH1) or (CNS) or (MPSU or RCMSU));
```

End logica;

APÊNDICE C: Testbench em Linguagem de Descrição de Hardware das lógicas de TRIP de um reator de pesquisa

Situação 1: Start_up_and_Wide_Range_Trip_Request

```
Entity TestBench1 is
end TestBench1;
```

```
Architecture Teste of TestBench1 is
```

```
component Start_up_and_Wide_Range_Trip_Request_1 is
Port(  SNFL : in bit;
        SNFH : in bit;
        WRLNFL1 : in bit;
        WRLNFR : in bit;
        WRLNFL2 : in bit;
        WRLNFL3 : in bit;
        RPSCT : in bit;
        WRLNFL4 : in bit;
        Start_up_and_Wide_Range_Trip_Request_1_request : out bit);
End component;
```

```
End component;
```

```
Signal SNFL_1, SNFH_1, WRLNFL1_1, WRLNFR_1, WRLNFL2_1, WRLNFL3_1,
RPSCT_1, WRLNFL4_1, Start_up_and_Wide_Range_Trip_Request_1_request_1: bit;
```

```
Begin
```

```
TripTeste: Start_up_and_Wide_Range_Trip_Request_1 port map (SNFL_1, SNFH_1,
        WRLNFL1_1, WRLNFR_1, WRLNFL2_1, WRLNFL3_1, RPSCT_1,
        WRLNFL4_1,     Start_up_and_Wide_Range_Trip_Request_1_request      =>
        Start_up_and_Wide_Range_Trip_Request_1_request_1);
```

SNFL_1 <= '0' after 0 ns,

'1' after 8 ns,

'0' after 22 ns,

'1' after 36 ns;

SNFH_1 <= '0' after 0 ns,

'1' after 8 ns,

'0' after 22 ns,

'1' after 36 ns;

WRLNFL1_1 <= '0' after 0 ns,

'1' after 8 ns,

'0' after 22 ns,

'1' after 36 ns,

'0' after 50 ns;

WRLNFR_1 <= '1' after 0 ns,

'0' after 8 ns,

'1' after 22 ns,

'0' after 36 ns,

'1' after 50 ns;

WRLNFL2_1 <= '0' after 0 ns,

'1' after 8 ns,

'0' after 22 ns,

'0' after 36 ns,

'1' after 50 ns;

WRLNFL3_1 <= '1' after 0 ns,

'1' after 8 ns,

'0' after 22 ns,

'1' after 36 ns,

'0' after 50 ns;

RPSCT_1 <= '1' after 0 ns,

'0' after 8 ns,

'0' after 22 ns,

'0' after 36 ns,

'1' after 50 ns;

WRLNFL4_1 <= '0' after 0 ns,

'1' after 8 ns,

'1' after 22 ns,

'1' after 36 ns,

'0' after 50 ns;

end Teste;

Situação 2: Start_up_and_Wide_Range_Trip_Request

```
Entity TestBench2 is
end TestBench2;
```

```
Architecture Teste of TestBench2 is
```

```
component Start_up_and_Wide_Range_Trip_Request_1 is
Port(  SNFL : in bit;
        SNFH : in bit;
        WRLNFL1 : in bit;
        WRLNFR : in bit;
        WRLNFL2 : in bit;
        WRLNFL3 : in bit;
        RPSCT : in bit;
        WRLNFL4 : in bit;
        Start_up_and_Wide_Range_Trip_Request_1_request : out bit);
```

```
End component;
```

```
Signal SNFL_1, SNFH_1, WRLNFL1_1, WRLNFR_1, WRLNFL2_1, WRLNFL3_1,
      RPSCT_1, WRLNFL4_1, Start_up_and_Wide_Range_Trip_Request_1_request_1: bit;
```

```
Begin
```

```
TripTeste: Start_up_and_Wide_Range_Trip_Request_1 port map (SNFL_1, SNFH_1,
        WRLNFL1_1, WRLNFR_1, WRLNFL2_1, WRLNFL3_1, RPSCT_1,
        WRLNFL4_1,      Start_up_and_Wide_Range_Trip_Request_1_request      =>
        Start_up_and_Wide_Range_Trip_Request_1_request_1);
```

```
SNFL_1 <= '1' after 0 ns,
      '0' after 10 ns,
      '0' after 30 ns,
```

'0' after 50 ns,
'1' after 70 ns;

SNFH_1 <= '0' after 0 ns,
'1' after 10 ns,
'1' after 30 ns,
'1' after 50 ns,
'0' after 70 ns;

WRLNFL1_1 <= '1' after 0 ns,
'0' after 10 ns,
'1' after 30 ns,
'1' after 50 ns,
'0' after 70 ns;

WRLNFR_1 <= '0' after 0 ns,
'0' after 10 ns,
'1' after 30 ns,
'0' after 50 ns,
'1' after 70 ns;

WRLNFL2_1 <= '0' after 0 ns,
'1' after 10 ns,
'1' after 30 ns,
'0' after 50 ns,
'0' after 70 ns;

WRLNFL3_1 <= '0' after 0 ns,
'1' after 10 ns,
'1' after 30 ns,
'1' after 50 ns,
'0' after 70 ns;

RPSCT_1 <= '0' after 0 ns,

'1' after 10 ns,

'0' after 30 ns,

'1' after 50 ns,

'1' after 70 ns;

WRLNFL4_1 <= '0' after 0 ns,

'1' after 10 ns,

'1' after 30 ns,

'1' after 50 ns,

'0' after 70 ns;

end Teste;

Situação 1: Primary Cooling System Trip Request

```
Entity TestBench3 is
end TestBench3;
```

```
Architecture Teste of TestBench3 is
```

```
component Primary_Cooling_System_Trip_Request_1 is
Port( PCFL : in bit;
      CPD1L : in bit;
      RPSCP : in bit;
      CPD1H : in bit;
      CTDH : in bit;
      COTH : in bit;
      Primary_Cooling_System_Trip_Request_1_request : out bit);
```

```
End component;
```

```
Signal PCFL_1, CPD1L_1, RPSCP_1, CPD1H_1,
      CTDH_1, COTH_1, Primary_Cooling_System_Trip_Request_1_request_1: bit;
```

```
Begin
```

```
TripTeste: Primary_Cooling_System_Trip_Request_1 port map (PCFL_1, CPD1L_1,
                                                       RPSCP_1,           CPD1H_1,           CTDH_1,           COTH_1,
                                                       Primary_Cooling_System_Trip_Request_1_request
                                                       => Primary_Cooling_System_Trip_Request_1_request_1);
```

```
PCFL_1 <=  '1' after 0 ns,
            '1' after 8 ns,
            '0' after 22 ns,
            '1' after 36 ns,
            '1' after 50 ns,
```

'1' after 64 ns,
'0' after 78 ns,
'0' after 92 ns;

CPD1L_1 <= '0' after 0 ns,
'1' after 8 ns,
'1' after 22 ns,
'1' after 36 ns,
'0' after 50 ns,
'0' after 64 ns,
'0' after 78 ns,
'1' after 92 ns;

RPSCP_1 <= '0' after 0 ns,
'0' after 10 ns,
'0' after 30 ns,
'1' after 45 ns,
'1' after 60 ns,
'1' after 75 ns,
'0' after 80 ns,
'1' after 95 ns;

CPD1H_1 <= '1' after 0 ns,
'1' after 5 ns,
'0' after 20 ns,
'0' after 35 ns,
'0' after 50 ns,
'0' after 65 ns,
'0' after 80 ns,
'1' after 95 ns;

CTDH_1 <= '0' after 0 ns,
'1' after 12 ns,

'0' after 22 ns,
'1' after 32 ns,
'0' after 42 ns,
'1' after 52 ns,
'0' after 72 ns,
'1' after 92 ns;

COTH_1 <= '1' after 0 ns,
'1' after 20 ns,
'0' after 40 ns,
'1' after 50 ns,
'0' after 60 ns,
'1' after 80 ns,
'0' after 90 ns,
'1' after 100 ns;

end Teste;

Situação 2: Primary Cooling System Trip Request

```
Entity TestBench4 is
end TestBench4;
```

```
Architecture Teste of TestBench4 is
```

```
component Primary_Cooling_System_Trip_Request_1 is
Port( PCFL : in bit;
      CPD1L : in bit;
      RPSCP : in bit;
      CPD1H : in bit;
      CTDH : in bit;
      COTH : in bit;
      Primary_Cooling_System_Trip_Request_1_request : out bit);
End component;
```

```
Signal PCFL_1, CPD1L_1, RPSCP_1, CPD1H_1,
      CTDH_1, COTH_1, Primary_Cooling_System_Trip_Request_1_request_1: bit;
```

```
Begin
```

```
TripTeste: Primary_Cooling_System_Trip_Request_1 port map (PCFL_1, CPD1L_1,
                                                       RPSCP_1,           CPD1H_1,           CTDH_1,           COTH_1,
                                                       Primary_Cooling_System_Trip_Request_1_request
                                                       => Primary_Cooling_System_Trip_Request_1_request_1);
```

```
PCFL_1 <=  '1' after 0 ns,
           '1' after 5 ns,
           '0' after 10 ns,
           '1' after 50 ns;
```

```
CPD1L_1 <=  '0' after 0 ns,  
              '1' after 20 ns,  
              '0' after 40 ns,  
              '1' after 50 ns;
```

```
RPSCP_1 <=  '1' after 0 ns,  
              '1' after 8 ns,  
              '1' after 25 ns,  
              '1' after 50 ns;
```

```
CPD1H_1 <=  '1' after 0 ns,  
              '1' after 5 ns,  
              '0' after 30 ns,  
              '0' after 50 ns;
```

```
CTDH_1 <=  '1' after 0 ns,  
              '0' after 20 ns,  
              '0' after 30 ns,  
              '1' after 50 ns;
```

```
COTH_1 <=  '1' after 0 ns,  
              '0' after 15 ns,  
              '1' after 30 ns,  
              '1' after 50 ns;
```

```
end Teste;
```