



HOMERO LUZ GUIMARÃES

***UMA ARQUITETURA DE PROCESSAMENTO PARALELO PARA IMPLEMENTAÇÃO
DE UM TRIGGER NÍVEL ZERO PARA INSTRUMENTAÇÃO NUCLEAR***

**CAMPINAS
2013**



UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

**UMA ARQUITETURA DE PROCESSAMENTO PARALELO PARA IMPLEMENTAÇÃO
DE UM TRIGGER NÍVEL ZERO PARA INSTRUMENTAÇÃO NUCLEAR**

Autor: Homero Luz Guimarães

Orientador: Prof. Dr. José Antonio Siqueira Dias

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Doutor em Engenharia Elétrica, na área de concentração: Eletrônica, Microeletrônica e Optoeletrônica

Este exemplar corresponde à versão final da Tese defendida pelo aluno Homero Luz Guimarães e orientada pelo Prof. Dr. José Antonio Siqueira Dias

Campinas

2013

iii

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

G947a Guimarães, Homero Luz
Uma arquitetura de processamento paralelo para implementação de um trigger nível zero para instrumentação nuclear / Homero Luz Guimarães. -- Campinas, SP: [s.n.], 2013.

Orientador: José Antonio Siqueira Dias.
Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Detectores de partículas sensíveis a posição. 2. Detectores nucleares. 3. Circuitos integrados. 4. Circuitos integrados lineares. 5. Circuitos integrados digitais. I. Dias, José Antonio Siqueira. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: A parallel processing architecture for the implementation of a level zero trigger for nuclear instrumentation

Palavras-chave em Inglês: Particle detectors position sensitive, Nuclear detectors, Integrated circuits, Linear integrated circuits, Digital integrated circuits

Área de concentração: Eletrônica, Microeletrônica e Optoeletrônica

Titulação: Doutor em Engenharia Elétrica

Banca examinadora: Saulo Finco, Rogério Lara Leite, Elnatan Chagas Ferreira, Wilmar Bueno de Moraes

Data da defesa: 28-01-2013

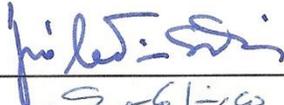
Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

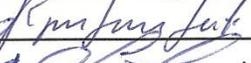
Candidato: Homero Luz Guimarães

Data da Defesa: 28 de janeiro de 2013

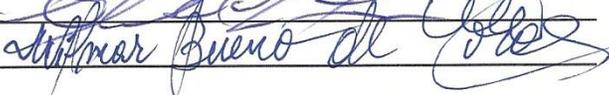
Título da Tese: "Uma Arquitetura de Processamento Paralelo para Implementação de um Trigger Nível Zero para Instrumentação Nuclear"

Prof. Dr. José Antonio Siqueira Dias (Presidente):  _____

Dr. Saulo Finco:  _____

Dr. Rogério Lara Leite:  _____

Prof. Dr. Elnatan Chagas Ferreira:  _____

Prof. Dr. Wilmar Bueno de Moraes:  _____

Agradecimentos

Agradeço primeiramente a Deus que me proporcionou vida, oportunidade, perseverança, condições e forças para concluir esta dissertação.

Agradeço a minha Família, a minha esposa Noemi, pelo apoio e incentivo durante todo o período de elaboração deste trabalho.

Agradeço especialmente ao Professor José A. Siqueira Dias pela orientação e todo o suporte necessário durante esta pesquisa.

Agradeço ao grupo de instrumentação Dzero do Fermilab (Batavia, EUA) e, em especial aos Dr. Marvin Johnson, Dr. Jose Martin, Dr. Mario Vaz, Dr. Fred Borcharding e Robert Angstad pelo suporte dado durante o início desta dissertação.

Agradecimentos especiais também ao DH-CTI, que com seu apoio em termos de recursos computacionais e dados relativos ao processo CMOS tornaram a simulação deste trabalho possível.

Agradeço também a colaboração Dzero no Fermilab pelas figuras usadas para descrever tanto o acelerador como a estrutura do detector Dzero usadas em alguns capítulos desta tese.

Também quero agradecer a todas as pessoas que não sendo mencionadas aqui, direta ou indiretamente me auxiliaram durante a pesquisa e elaboração deste trabalho.

“Viva como se você fosse morrer amanhã,
aprenda como se você fosse viver para sempre”
Mahatma Gandhi

Resumo

Os experimentos em Física de alta energia tem se beneficiado enormemente do progresso alcançado na área de Microeletrônica, pois isto tem proporcionado a criação de detectores mais acurados e circuitos de processamento de sinais analógico/digitais cada vez mais rápido e precisos. A redução no comprimento mínimo de canal dos processos CMOS além de proporcionar maior velocidade e precisão também reduz a área usada por cada canal, o que permite a implementação de mais canais numa mesma pastilha. Com um numero maior de canais por pastilha, com um mesmo numero de chips podemos programar um numero maior de canais do que anteriormente possível e com isso os físicos podem realizar uma reconstrução da trajetória de maneira mais precisa.

Este Trabalho descreve uma proposta para o Trigger de nível zero baseando-se nas especificações disponíveis do Experimento Dzero no Fermi National Accelerator Laboratory (FERMILAB).

Este trabalho descreve o projeto e implementação de um front-end analógico que detecta a carga provida pelo VLPC (detector luminoso usado no Dzero) seguida por um comparador de alta velocidade que fornece um nível logico para um processador digital.

O processador digital por sua vez usa uma arquitetura de processadores paralelos que, comunicando-se entre si são capazes de estimar a trajetória de partículas baseando-se em dados iniciais programados a partir de simulações do detector feitas em computadores pelos Físicos.

Tanto o bloco analógico quanto o processador digital foram implementados usando-se o processo CMOS90 da IBM.

Abstract

The experiments in high-energy physics has benefited greatly from the progress made in the area of Microelectronics, since it has provided the creation of more accurate detectors and analog / digital signal processing circuits that are increasingly fast and accurate. The reduction in the minimum length of the channel in modern CMOS processes while providing greater speed and precision also reduces the area used by each channel, which enables the implementation of more channels on the same chip. With a larger number of channels per chip, we can with the same number of chips implement a larger number of channels than previously possible and with that physicists can perform a reconstruction of the trajectory more accurately.

This work describes a proposal for a Trigger level zero based on the available specifications of the DZero experiment at the Fermi National Accelerator Laboratory (FERMILAB).

In the following pages the design and implementation of an analog front-end that detects the charge provided by the VLPC detector followed by a high-speed comparator that provides a logical level to a digital processor are described.

The digital processor in turn uses an architecture of parallel processors that communicate with each other are able in order to estimate the trajectory of particles based on initial data loaded in RAM based on simulations of the detector geometry made by physicists. Both the analog block and the digital processor are implemented using the IBM CMOS90 process.

Sumário

Agradecimentos	vii
Resumo	ix
Abstract	xi
Lista de Figuras	xv
Lista de Tabelas	xix
Capítulo 1: O acelerador Tevatron e o experimento Dzero	1
1.1- O Fermilab e o acelerador Tevatron.....	1
1.2 – Descrição do detector e sua função.....	5
1.3 – Descrição do Detector Dzero	9
Capitulo 2 – O sistema de trigger e o sensor CFT.	29
2.1- Princípio básico de um sistema de trigger	29
2.2- O sistema de Trigger do Detector Dzero	30
2.3- Uma visão detalhada do sensor de Fibras Cintilantes.	36
2.4 – Esquemas de distribuição do trigger atual	39
2.5 – Metodologia usada no projeto do trigger atual	50
Capitulo 3 - Estudo e proposta de nova arquitetura	55
3.1 - Estudo e simulação	55
3.2 - Proposta de Arquitetura	60
3.3 - Algoritmos usados no LDE	73
3.4 – Protocolo de comunicação	77
3.4.1 -Algoritmo do LDE:.....	77
Capitulo 4 – Código VHDL e simulação Digital.....	93
4.1-Codificação em VHDL da parte digital	93
4.2- Circuitos de teste e Resultados das Simulações digitais.....	97
Capitulo 5 – Circuitos Analógicos	109
5.1-Comparador modo corrente	109
5.2- Projeto do front-end Analógico	114
5.3-Circuito do pré-amplificador de Carga.....	117
5.4 – Comparador CMOS	121
5.5 - Simulando o front-end analógico Completo.....	128
Capitulo 6 - Conclusão.....	131
Referencias Bibliográficas	133

Lista de Figuras

- Figura 1.1.1 – O Tevatron
- Figura 1.1.2 – Vista aérea do Tevatron
- Figura 1.1.3 – Os vários estágios do Acelerador Tevatron
- Figura 1.2.1 – O detector CDF
- Figura 1.2.2 – O Detector Dzero
- Figura 1.3.1 – O detector Dzero e seus subsistemas de sensores
- Figura 1.3.2 – Detectores centrais do Dzero
- Figura 1.3.3 – Detector de Vértice (Silicon Tracker)
- Figura 1.3.4(a) – Visão interna do detector de vértice (cilindros e discos)
- Figura 1.3.4(b) – Visão externa de uma seção do detector de vértice (cilindros e discos perpendiculares)
- Figura 1.3.5 – Layout do detector de silício
- Figura 1.3.6 – Vista lateral de uma das placas do detector de silício mostrando o C.I. de leitura montado sobre esta.
- Figura 1.3.7 – Ilustração do efeito combinado do detector em barril e em disco para elevar a resolução de detector central de silício
- Figura 1.3.8 – Foto mostrando o detector central de silício durante sua montagem
- Figura 1.3.9 – Foto do CFT durante montagem
- Figura 1.3.10 – Vista em corte do preshower central e das fibras
- Figura 1.3.11 – Vista em corte do FPS
- Figura 1.3.12- Uma célula do sensor do calorímetro
- Figura 1.3.13 – Sensores de Calorímetro e sua segmentação
- Figura 1.3.14 – Vista do Calorímetro durante a instalação do CFT
- Figura 1.3.15 – Sensores de Muons
- Figura 1.3.16 – Vista perpendicular da Câmara de fios usada nos sensores de Muons
- Figura 1.3.17 – Pad de catodo usado na câmara de Muons
- Figura 2.1.1 – Conceitos de um sistema de trigger
- Figura 2.2.1 – Timing das colisões
- Figura 2.2.2 – Eletrônica de Leitura dos sensores centrais
- Figura 2.2.3 – Diagrama simplificado do Pré-amplificador CMOS usado no CFT
- Figura 2.2.4 – Diagrama em blocos do SVXII
- Figura 2.2.5 – Triggering proposto para o Dzero
- Figura 2.3.1 – Fibras cintilantes
- Figura 2.3.2 – Visão em corte de uma Superlayer
- Figura 2.3.3 – Um setor do CFT
- Figura 2.4.1 – Calculo de DL
- Figura 2.4.2 – Formando SL's
- Figura 2.4.3 – Uma colisão típica
- Figura 2.4.4 – Arquitetura proposta para a versão 1
- Figura 2.4.5 – Nova distribuição das fibras
- Figura 2.4.6 – trajetória de uma partícula de 3 GeV

Figura 2.4.7 – esquema de transferência de dados entre os setores
Figura 2.4.8 – Barramento interno da placa
Figura 2.5.1 – Arquivo texto proveniente da simulação Monte Carlo
Figura 2.5.2 – Numeração das fibras
Figura 2.5.3 – Arquitetura do trigger 1 (proposta corrente)
Figura 3.2.1 – Bifurcação devido a trilhas falsas
Figura 3.2.2 – Arquitetura proposta (1 chip)
Figura 3.2.3 – Arquitetura do LDE
Figura 3.2.4 – Comparador de modo Corrente (porta AND)
Figura 3.2.5 – Estrutura da RAM estática
Figura 3.2.6 – Estrutura Interna dos registros
Figura 3.2.7 – associando vários LDEs por camada formando uma cascata
Figura 3.2.8 – associando uma cascata para cada camada
Figura 3.3.1 – Visões detalhada do CI
Figura 3.3.2 – Visão detalhada de cada um dos LDE
Figura 3.3.3 – Comparador de corrente
Figura 3.3.4 – Visão detalhada da tabela Look-up
Figura 4.1.1 –Esquemático de topo do LDE_CONTROLLER
Figura 4.1.2 –Esquemático de topo do LDE
Figura 4.2.1 – Programação de um LDE
Figura 4.2.2 - Operação do LDE
Figura 4.2.3 – Reset e programação de um LDE
Figura 4.2.4 – Programação de um LDE
Figura 4.2.5 – Programação dos 36 LDEs de um chip
Figura 4.2.6 – Resposta a um requerimento de um LDE extra
Figura 4.2.7 – Resposta a um requerimento de um LDE extra no modo escravo
Figura 4.2.8 – Operação normal com dois chips
Figura 4.2.9 – Programação dos LDEs com dois chips
Figura 4.2.10 – Resposta a requisições de LDEs extras com dois chips
Figura 4.2.11 – Resposta a requisições de leitura com dois chips
Figura 4.2.12 – Resposta a requisições de leitura com dois chips – foco numa única trilha
Figura 4.2.13 – Resposta Completa
Figura 5.1.1 – Modelo do comparador modo corrente
Figura 5.1.2 – Esquemático do comparador modo corrente
Figura 5.1.3 – Resposta do comparador modo corrente passos 1 e 2
Figura 5.1.4 – Resposta do comparador modo corrente passo 3
Figura 5.1.5 – Esquemático final do comparador modo corrente
Figura 5.2.1 – Front-end e sua integração com o VLPC
Figura 5.2.2 – Caracterização de um NMOS
Figura 5.2.3 – Caracterização G_m/I_d e G_{ds}/I_d como função de V_{gs} e L
Figura 5.3.1 – Circuito Pré-amplificador de Carga
Figura 5.3.2 – Margem de phase do loop
Figura 5.3.3 – Resposta AC em malha aberta
Figura 5.3.4 – Resposta a um pulso de carga
Figura 5.3.5 – Resposta a um pulso de carga incluindo ruído dos dispositivos
Figura 5.4.1 – Primeiro estágio do Comparador (pré-amplificador)

Figura 5.4.2 – Testbench do Pré-amplificador
Figura 5.4.3 – Simulando o Pré-amplificador
Figura 5.4.4– Atraso no Pré-amplificador
Figura 5.4.5– Latch Analógico
Figura 5.4.6– Reduzindo o efeito Kickback
Figura 5.4.7– Reduzindo o efeito Kickback (Simulação)
Figura 5.4.8– Pre-amp e latch analógica em cascata
Figura 5.4.9– Aplicando o degrau mínimo na entrada do Pre-amp + Latch analógico
Figura 5.4.10– Comparador mostrando os seus 3 estágios
Figura 5.4.11– Comparador detectando mínima degrau
Figura 5.5.1– Simulando o front-end analógico completo
Figura 5.5.2– Comparador detectando mínima carga
Figura 5.5.3– Reset do comparador

Lista de Tabelas

- Tabela 2.4.1 – Comparacao entre Ancoras na Camada A e D
- Tabela 2.4.2 – Numero de fibras necessárias por Camada
- Tabela 3.1 – Fibras atingidas conforme o modelo em C++
- Tabela 3.2 – Parte das trilhas resultantes como função das camadas
- Tabela 3.3 – Máximo numero de trilhas por combinação (parte)
- Tabela 5.3.1 – Dimensões dos transistores do Pré-amplificador

Capítulo 1: O acelerador Tevatron e o experimento Dzero

1.1- O Fermilab e o acelerador Tevatron

Os Físicos usam aceleradores com energias cada vez maiores para investigar as camadas mais profundas dentro do núcleo atômico. As altas energias do feixe possibilitam a produção de partículas mais pesadas e raras devido a maior desintegração do núcleo.

O Fermilab (Fermi National Accelerator Laboratory) é onde está localizado o Tevatron – um dos mais poderosos aceleradores de partículas da atualidade. O Tevatron provê feixes de partículas para os experimentos no Fermilab de três formas distintas: No modo de alvo-fixado, o Tevatron acelera prótons até 800 GeV e os envia num feixe para a área de experimentos de alvo fixo onde este incide num alvo feito de um material especial. As partículas resultantes da colisão são analisadas por detectores ao redor do alvo. No segundo modo, Anti-prótons são refrigerados até quase o zero absoluto e armazenados num anel sob a forma de feixe. Este feixe é usado em experimentos com alvos gasosos (plasma).

O terceiro modo é o mais importante e mais usado: Os anti-prótons armazenados no anel são injetados no anel principal e acelerados até 800 GeV. Ao mesmo tempo prótons são acelerados na direção contrária no mesmo anel. Quando a máxima energia é atingida os dois feixes são colimados para causar colisão entre as duas partículas. Esta colisão é calculada para ocorrer dentro de dispositivos chamados detectores. Os detectores separam e analisam as partículas que resultam da colisão dos dois feixes. Esta colisão tem uma energia de 1.8 TeV no seu centro de massa, vindo daí o nome Tevatron. Atualmente esta é a segunda maior energia obtida por qualquer acelerador de partículas no mundo, atrás apenas do Large Hadron Collider (LHC) do CERN (o qual gera 7 TeV) (1).

No Fermilab existem dois detectores posicionados ao longo do Anel: os experimentos Dzero e CDF.

O Tevatron é um acelerador em Anel composto de magnetos supercondutores. Este anel é preenchido com pacotes de prótons e anti-prótons, os quais circulam em sentido contrário. Nas áreas experimentais D0 e B0 (as posições no diâmetro do anel do Tevatron são identificadas por uma letra indicando a seção e um número indicando a ordem dentro da seção) os dois feixes são direcionados de modo a colidir entre si. O processo de preenchimento do anel é extremamente delicado, um sucinto resumo segue abaixo. Informações mais detalhadas são fornecidas nas referências (2), (3), (4), (5) e (6).

A figura 1.1.1 mostra um diagrama do acelerador Tevatron com seus dois detectores.

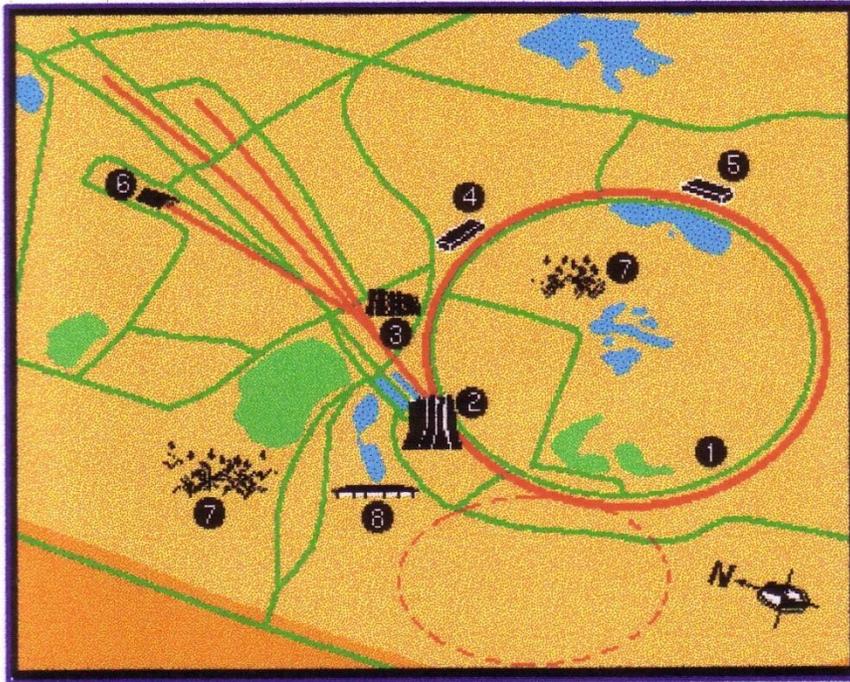


Figura 1.1.1 – O Tevatron

Na figura 1.1.1 os números representam os seguintes estágios:

- 1- Anel Principal
- 2- Wilson Hall – Centro de Controle do acelerador e sede do Fermilab
- 3- Feynman Computing Center – Centro de Computação
- 4- Detector CDF
- 5- Detector Dzero
- 6- Área Experimental de alvo fixo
- 7- Lagos de refrigeração do anel principal
- 8- Injetor Principal (Main Injector) – Anel em pontilhado

A Figura 1.1.2 mostra uma vista aérea do Fermilab que dá uma ideia do tamanho do acelerador e das suas partes:

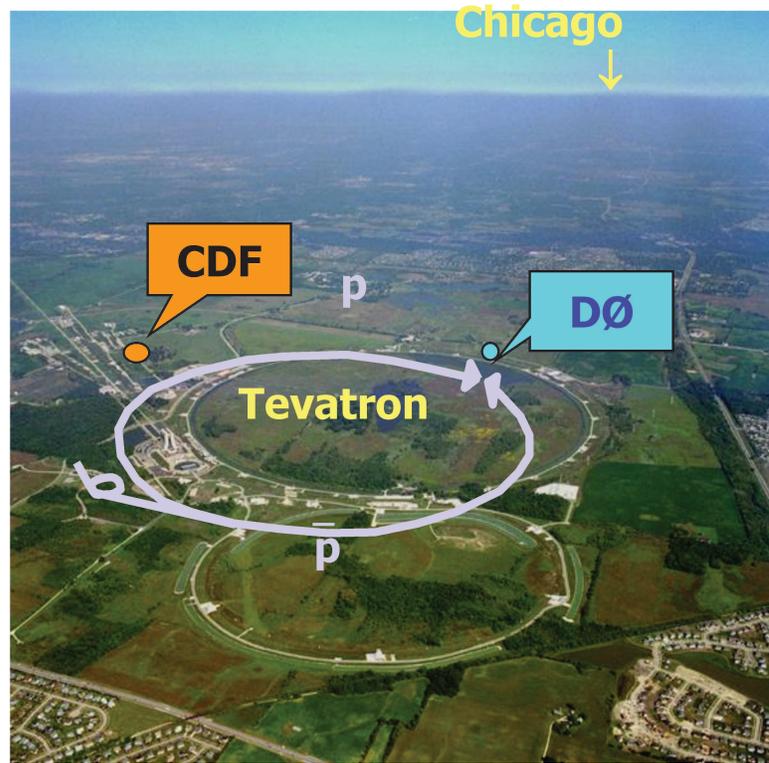


Figura 1.1.2 – Vista aérea do Tevatron

No modo de colisão o Tevatron acelera Prótons e anti-prótons em sentidos contrários num Anel subterrâneo de 6.4 km de diâmetro. Quando prótons e anti-prótons colidem numa velocidade próxima da velocidade da luz, uma energia tão grande quanto ao do “Big Bang” é produzida por um trilionésimo de segundo. Parte da energia envolvida na colisão é convertida em matéria (de acordo com o principio da relatividade formulado por Einstein) produzindo jatos de partículas em todas as direções. São estes jatos de partículas que os detectores devem mapear e analisar em busca de padrões que indiquem a descoberta de uma nova partícula e/o fenômenos que confirmem as predições e modelos dos físicos teóricos.

A figura 1.1.3 mostra de maneira mais simplificada os vários estágios que compõem o Tevatron.

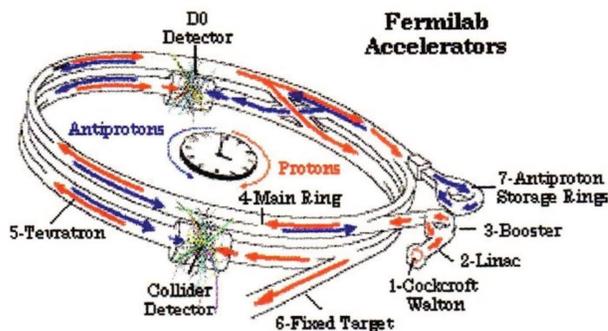


Figura 1.1.3 – Os vários estágios do Acelerador Tevatron

Inicialmente o feixe de partículas se origina a partir de Íons negativos de Hidrogênio gerados pelo acelerador eletrostático de Cockcroft-Walton. Este pré-acelerador opera num modo pulsado com uma frequência de 15 Hz. Dentro deste dispositivo, um elétron extra é acrescentado a cada átomo de hidrogênio num feixe. Os íons resultantes se constituem de dois elétrons e um próton são atraídos por um elevado potencial positivo e acelerados até uma energia de 750.000 eV. Os íons são agrupados em pacotes e transportados para o início do LINAC.

Os íons a seguir são acelerados pelo LINAC que é um pequeno acelerador linear de cerca de 150 metros de comprimento e cinco estágios de aceleração. O LINAC consiste de uma série de pequenos tubos (chamados de “drift tubes”) espaçados de maneira crescente. Um campo elétrico é aplicado aos tubos de tal forma que a polaridade é periodicamente invertida. As partículas ao cruzarem o tubo estão protegidas quando o campo elétrico tenderia a desacelera-las e emergem na extremidade entre os tubos justamente quando o campo elétrico está na direção de acelera-las ainda mais. Cada tubo corresponde a um estágio de aceleração e quando os íons emergem no último estágio eles estão acelerados com uma energia de 400 MeV. Quando o feixe de íons negativo de hidrogênio atinge o terceiro estágio (o Booster Circular) os íons passam através de um fino filme de carbono e tem os seus dois elétrons removidos e se transforma num feixe de prótons. O Booster é essencialmente um Síncrotron de cerca de 150 metros de diâmetro. Um Síncrotron é um acelerador circular que usa Magnetos para encurvar partículas carregadas eletricamente num caminho circular de tal maneira que a partícula seja submetida repetidamente em cada revolução a campos elétricos que tendem a acelera-las mais e mais. Este acréscimo de energia se dá pela passagem do feixe por Cavidades Ressonantes de Micro-ondas, quando isto ocorre o campo gerado pelos magnetos deve ser elevado de maneira síncrona de modo a manter as partículas girando na mesma órbita.

Os prótons circulam no Booster cerca de 20.000 vezes e tem a sua energia elevada até cerca de oito GeV. O Booster normalmente completa 12 ciclos numa sucessão rápida de forma a carregar 12 pacotes de prótons no Main injector. O Main Injector é um acelerador em anel (síncrotron) localizado no mesmo túnel que o Tevatron e tem cerca de 6 Km de raio composto de magnetos convencionais. O Main Injector está localizado num plano perfeitamente horizontal, porém nas áreas experimentais do Dzero e CDF ele é desviado ligeiramente para cima de modo a abrir espaço para os detectores O main Injector eleva a energia do feixe até 150 GeV. Os prótons são a seguir injetados no anel do Tevatron.

O Tevatron contém cerca de 1000 magnetos supercondutores que colimam e curvam o feixe de partículas. Os magnetos supercondutores operam refrigerados por hélio líquido. Estes magnetos produzem um campo magnético extremamente intenso usando apenas uma fração da energia requerida por magnetos convencionais. Como mencionado anteriormente, o Injetor principal está no mesmo túnel do Tevatron conforme é mostrado na figura 1.1.3.

No modo de colisão o Tevatron é preenchido com 12 pacotes de prótons e 12 pacotes de anti-prótons circulando em sentido contrário e orientados a colidir nos experimentos Dzero e CDF. Os feixes são colimados de forma a colidir por cerca de 20 Horas contínuas, após o que o acelerador é esvaziado de pacotes e preenchido por um novo grupo de recém - produzido pacotes de prótons e anti-prótons e todo o ciclo se repete.

Como já foi dito anteriormente, no modo colisão nós precisamos de prótons e anti-prótons. Cada anti-prótons tem a mesma massa de um próton, mas tem uma carga de polaridade oposta. Para produzir os anti-prótons necessários um feixe de prótons é primeiro acelerado até 120 GeV no Main Injector e depois extraído e focado num alvo de um material conveniente (níquel). A colisão do feixe com o material do alvo produz uma gama variada de partículas incluindo anti-prótons. Imediatamente após o alvo temos lentes de lítio, uma peça cilíndrica de lítio através da qual uma corrente circula. Isto cria um campo magnético azimutal o qual atua de forma a colimar partículas de carga negativa passando pelo seu interior. Após as lentes temos um magneto que desvia as partículas de carga negativa com energia maior do que oito GeV para um outro anel (“Debuncher ring”) onde eles são reduzidos em tamanho por um processo chamado congelamento estocástico. Depois disto eles são transferidos e armazenados em um pequeno anel em forma de triângulo que forma o Anel de Armazenagem de Anti-prótons onde o processo de congelamento estocástico continua criando um núcleo extremamente denso ao longo do anel do acumulador. Finalmente, quando um número suficiente de anti-prótons são produzidos (tipicamente da ordem de $50-150 \times 10^{10}$ partículas), os anti-prótons são injetados no anel do Main Injector e são acelerados simultaneamente aos prótons que circulam no sentido contrário. A seguir os anti-prótons são injetados no Tevatron onde eles sofrem a aceleração final até os 1.8 Tev.

Os prótons e anti-prótons, que são cerca de 100.000 vezes menores que um átomo são colimados por campos Magnéticos criados por solenoides supercondutores em dois feixes isolados. Estes dois feixes são mantidos isolados ao redor do anel com exceção da região dos dois detectores (experimentos) onde eles são forçados a se cruzarem sob o controle da sala de operações do experimento.

1.2 – Descrição do detector e sua função

Feixes de prótons e anti-prótons colidem em velocidades próximas da luz a cada 396 ns dentro dos detectores. Esta colisão dura cerca de um trilionésimo de segundo. Nossa compreensão do que acontece no instante da colisão é obtida indiretamente pelos debris que são gerados a partir desta. A cada 396 ns Prótons e anti-prótons se desintegram em um jato de partículas secundárias: quarks, elétrons, muons, neutrinos, B mésons e W bósons são algumas delas. Existem mais que uma centena de possibilidades. A tarefa do detector é monitorar o maior número de colisões possíveis, reconhecer e armazenar as partículas originadas e preservar os dados para um estudo posterior mais detalhado.

Pela análise dos dados armazenados pelo detector, os físicos podem fazer descobertas fundamentais a respeito da natureza da matéria e energia. No Anel do Tevatron existem dois detectores: o detector CDF e o detector Dzero. A figura 1.2.1 mostra uma visão do detector CDF e a figura 1.2.2 mostra parte do Dzero.

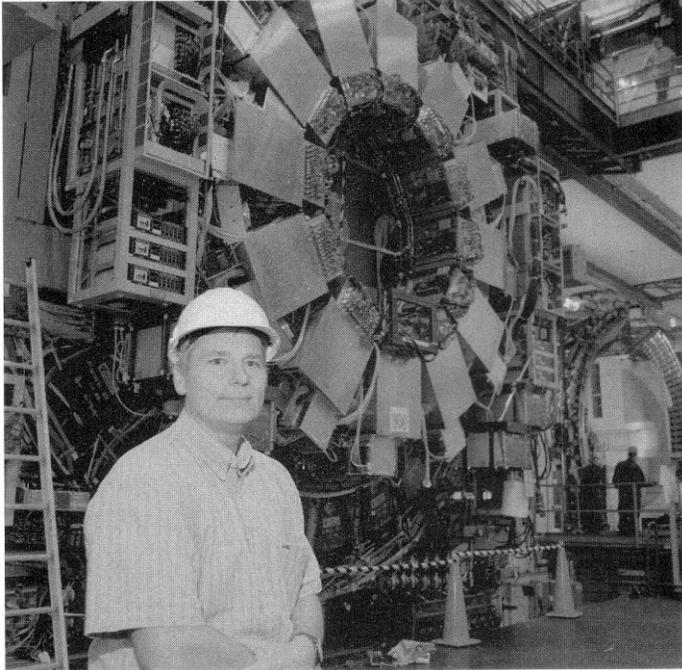


Figura 1.2.1- O detector CDF



Figura 1.2.2-O Detector Dzero

A primeira função do detector é decidir quais das milhares de colisões que acontecem a cada segundo merecem ser registradas, pois seria inviável registrar todas as colisões. Circuitos eletrônicos são programados para filtrar as colisões mais comuns e salvar as mais interessantes para análises posteriores. O detector deve registrar o máximo possível de informações a respeito de cada partícula gerada por cada colisão: a trajetória da partícula, sua energia e a sua carga. Devido ao fato de cada partícula que é gerada numa colisão ter um comportamento próprio, o detector pode usar esta informação para identificar a assinatura eletrônica da partícula e determinar se determinada partícula estava presente ou não em uma determinada colisão. Para criar uma visão completa da colisão o detector é composto de várias camadas de diferentes tipos de materiais envolvendo o ponto central da colisão.

Cada camada possui sensores que são projetados para coletar e transmitir um tipo diferente de informação a respeito das partículas e sua interação com o material da camada. Os dados fornecidos pelos sensores são analisados por processadores de modo a traçar as trajetórias das partículas assim como a sua carga e energia. Ao se juntar todas as informações de todas as camadas o detector constrói uma espécie de “foto” instantânea da colisão.

Um detector deve conter um número suficiente de camadas de diferentes materiais de modo que todas as partículas geradas pelas colisões interajam ao menos com uma das camadas e deve ter um número suficiente de canais de aquisição de dados de modo a coletar todas as informações de modo extremamente rápido e envia-las para processamento.

Cada um dos dois detectores do Fermilab pesa cerca de 5.000 Toneladas e tem uma altura média de um prédio de três andares. Durante a operação do Tevatron no modo colisão (chamada de “runs” pelos físicos), a sala de operação do experimento opera 24 horas por dia/7 dias por semana com turnos de 8 horas. Ao fim do “run” quando todos os dados foram coletados, filtrados e armazenados, o detector é removido do túnel do acelerador e movido para o hall de manutenção (“Assembly Hall”) para se fazer quaisquer upgrades que se façam necessários para o próximo “run” do Tevatron. A partir deste ponto os dados coletados são analisados pelos físicos em busca de novas partículas e interações. Uma vez que cada partícula gerada por uma colisão tem um comportamento único, o detector usa esta característica para verificar a presença de uma determinada partícula em uma dada colisão. Por exemplo, se por acaso acharmos assinaturas de um elétron com alta-energia, um Muon, um Neutrino e dois jatos de partículas numa colisão, isto é um forte indício que um Top Quark emergiu da colisão e depois decaiu nestas partículas de maior vida média. Como os físicos conseguem prever isto? Eles essencialmente usam um programa chamado “Geant” (7) onde eles entram com as dimensões físicas do detector e suas camadas de materiais detectores, as partículas que vão interagir no seu núcleo (no caso do Tevatron Prótons e anti-prótons) e o software fornece as trajetórias de todas as partículas possíveis de serem geradas. Uma vez que o número de possibilidades é muito elevado, os físicos limitam este numero colocando condições de contorno, por exemplo: ao se especificar que uma das partículas geradas seja um Top Quark, o “Geant” nos fornece o mapa das trajetórias das partículas que o detector deve registrar.

Como veremos adiante no texto esta mesma informação obtida do Geant será usada na programação dos padrões que o Circuito Integrado, objeto desta dissertação, deve detectar.

Mecanicamente um detector básico é composto das seguintes partes: o tubo metálico com o feixe do tevatron passando no centro geométrico do detector. Ao redor deste tubo existem câmaras cheias de gás usadas para detectar as partículas de alta energia que se afastam do ponto de colisão. Estes são os chamados detectores de traço (“tracking detectors”). Também ao redor deste tubo um campo magnético extremamente intenso é criado de forma a interagir com as partículas carregadas e encurvar as suas trajetórias. À medida que estas partículas passam pelo gás elas arrancam elétrons das moléculas de gás as ionizando. Estes elétrons migram na direção de fios cuidadosamente espaçados bem próximos um do outro (de modo a determinar a posição da partícula) e que são polarizados a um alto potencial positivo. A corrente produzida em cada um dos fios fornece-nos uma ideia de onde a partícula passou. A posição da partícula pode ser medida com uma precisão de fração de milímetros. Mudanças na posição e direção de uma partícula à medida que ela cruza o campo magnético nos dão uma medida de sua carga e momento.

Ao deixar estas camadas mais internas do detector (“tracking chambers”), as partículas encontram o calorímetro. No calorímetro eles perdem a sua energia numa sucessão de choques com várias camadas de materiais densos e absorventes intercalados de camadas de material cintilante situado entre elas. À medida que estas partículas cruzam camada após camada destes materiais elas criam chuveiros de partículas incluindo fótons assim como elétrons deixando um traço indicativo de sua energia e trajetória.

A maioria das partículas termina sua trajetória em algum ponto dentro do calorímetro ao perderem toda a sua energia nas colisões. Porém existem partículas, como o Muon, que não só cruzam as câmaras de detecção internas (“Tracking Chambers”), o calorímetro e finalmente deixam o seu traço ao cruzar as “Tracking chambers” localizadas no exterior do detector chamada de Câmaras de Muon (“Muon Chambers”). Nestas Câmaras os Muons deixam os seus traços antes de prosseguir a sua jornada para o exterior do detector e mesmo do prédio do detector. Outra espécie de partícula chamada de Neutrino simplesmente é capaz de cruzar todo o universo e facilmente cruza todas as camadas do detector sem sequer ser detectada. Uma vez que o Neutrino não pode ser detectado se assume que sua trajetória passa pela seção do detector diametralmente oposta a região onde trilhas de partículas são detectadas. Uma vez que a energia total é conservada numa colisão a energia do Neutrino pode ser calculada como a energia faltando após uma colisão próton-antipróton.

Obviamente com colisões ocorrendo a cada 396 ns seria impossível para um computador gravar os resultados de todos os detectores para todas as colisões. Circuitos programáveis especiais chamados triggers filtram os eventos não-importantes e somente registram os eventos mais interessantes para análise. Como veremos em capítulos posteriores existem vários níveis de trigger que operam em cascata de forma a filtrar de forma cada vez mais rigorosa em relação ao estágio anterior.

Uma vez que cada colisão produz dezenas de partículas, um detector típico monitora a atividade de milhões de partículas a cada segundo durante 24 Horas por dia. Destas Dezenas de bilhões de dados monitorados, somente um punhado irá servir para indicar a

existência de uma partícula nova. Assim uma rodada do Tevatron tipicamente dura alguns anos, durante os quais os detectores operam 24 horas por dia em um esquema de turnos de equipes de pesquisadores e técnicos de manutenção, de forma a não se perder nenhum evento.

1.3 – Descrição do Detector Dzero

Nesta seção descreveremos detalhes do detector Dzero como especificado após o upgrade previsto. A figura 1.3.1 mostra uma visão geral do detector com os vários sistemas de detectores indicados:

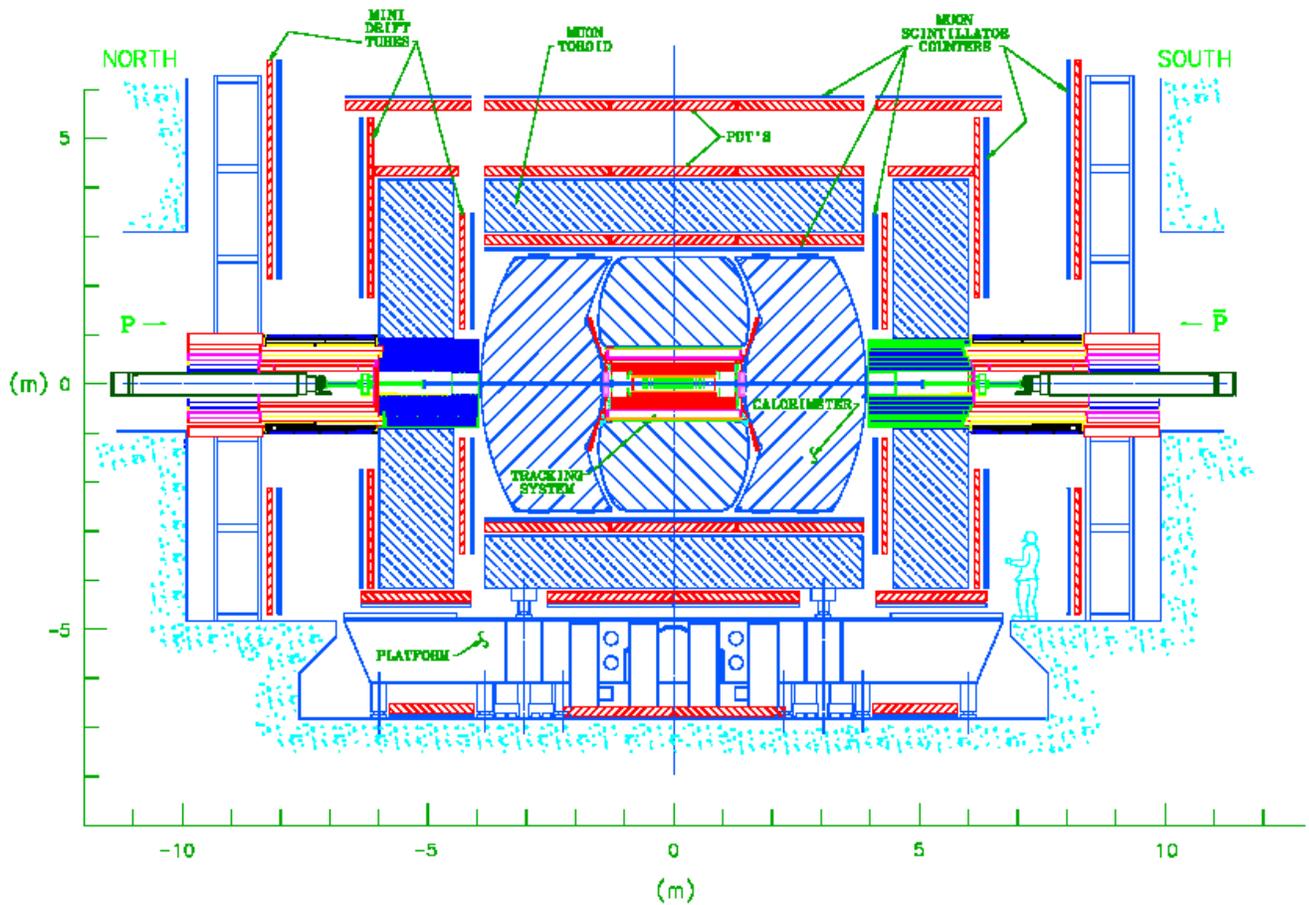


Figura 1.3.1 – O detector Dzero e seus subsistemas de sensores

Um dos principais detectores que sofreu upgrade foi o detector central (“Inner tracking”). Este upgrade foi requerido por dois motivos: primeiramente devido ao esperado dano causado pela elevada dose de radiação prevista com a entrada em operação do Main Injector e em segundo lugar para aperfeiçoar as capacidades de detecção do detector Dzero.

O Inner detector consiste de um detector estado-sólido de vértice (Silicon Vertex detector) envolto por quatro supercamadas de detectores de fibras cintilante (scintillating fiber detector). A razão do termo supercamadas será visto mais adiante quando discutirmos o detector de fibras cintilantes em mais detalhe. O Inner detector está localizado dentro de um solenoide supercondutor que gera um campo magnético de cerca de dois Teslas. Um detector preshower (preshower detector) também baseado em fibras cintilantes se encontra localizado na periferia externa do solenoide porem dentro do calorímetro. O preshower provê identificação de elétrons deixando a região central de colisão. Uma visão mais detalhada dos sensores do Dzero é vista na figura 1.3.2.

Na parte frontal do detector temos um detector também baseado em fibras cintilantes e discos de silício, chamado de preshower frontal (Forward preshower) que tem como função detectar o pequeno número de partículas que escapa do detector a ângulos pequenos.

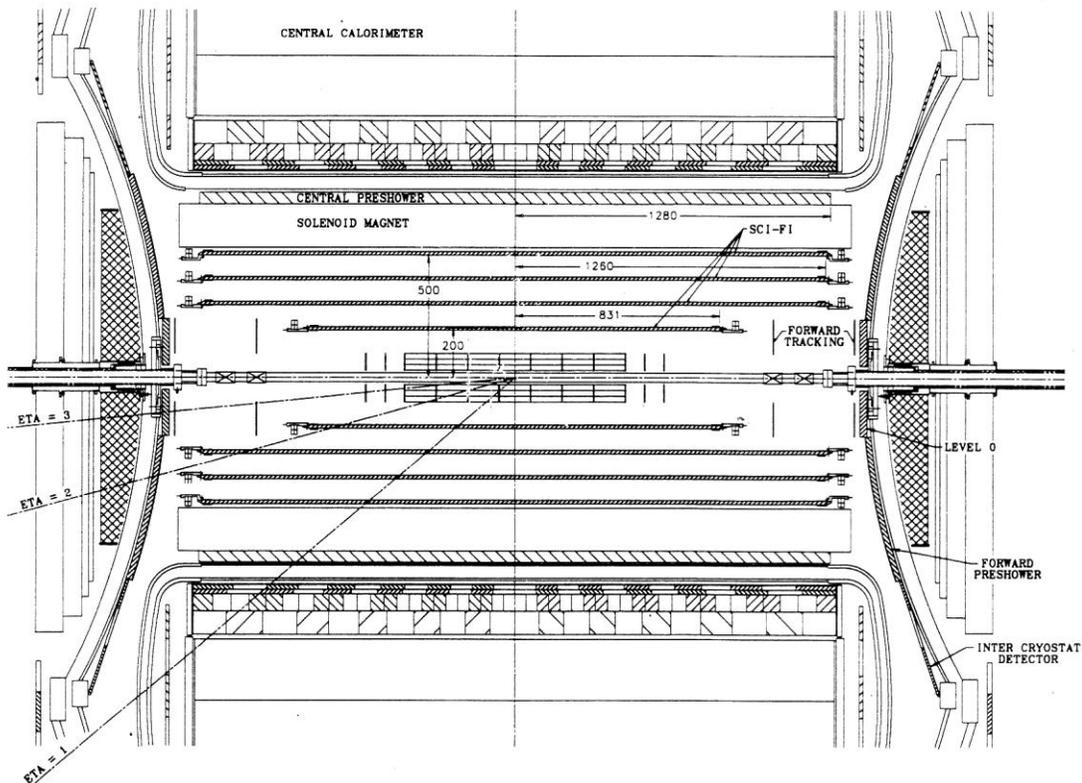


Figura 1.3.2 – Detectores centrais do Dzero

A seguir descrevemos em detalhes cada um dos principais subsistemas de detectores:

1) O Magneto Solenoide supercondutor:

O momento das partículas que tem carga elétrica é determinado pela sua curvatura na presença do campo magnético de dois Teslas gerado por um solenoide supercondutor de cerca de 2m de comprimento. O Solenoide supercondutor, construído sob encomenda pela Toshiba no Japão, é essencialmente uma bobina de duas camadas com um raio médio de 60 cm e possui uma energia armazenada de cinco Milhões de Joules. Próximo do detector está uma unidade criogênica que provê o Hélio líquido necessário para a refrigeração tanto do solenoide como dos detectores ópticos de estado sólido associado às fibras cintilantes (chamados de VLPC).

2) O detector de vértice:

Com o intervalo entre pacotes reduzido para 396 ns, não se pode usar mais as câmaras de fios descritas na seção 1.2, pois o tempo de deriva de elétrons até atingir os fios é muito elevado se comparado ao intervalo entre pacotes. O detector de vértice representa o detector de mais alta resolução do detector e é o primeiro detector a ser cruzado pelas partículas depois da colisão. No Dzero, se optou por se usar uma série de sensores de silício baseados em junções PN reversamente polarizadas. As partículas ao cruzarem as zonas de depleção dos diodos induzem correntes que podem ser medidas. Wafers inteiros são difundidos com milhares destes diodos distribuídos sobre o wafer e cortados em forma de disco e em forma de retângulos de maneira a montar a estrutura mostrada na figura 1.3.3:

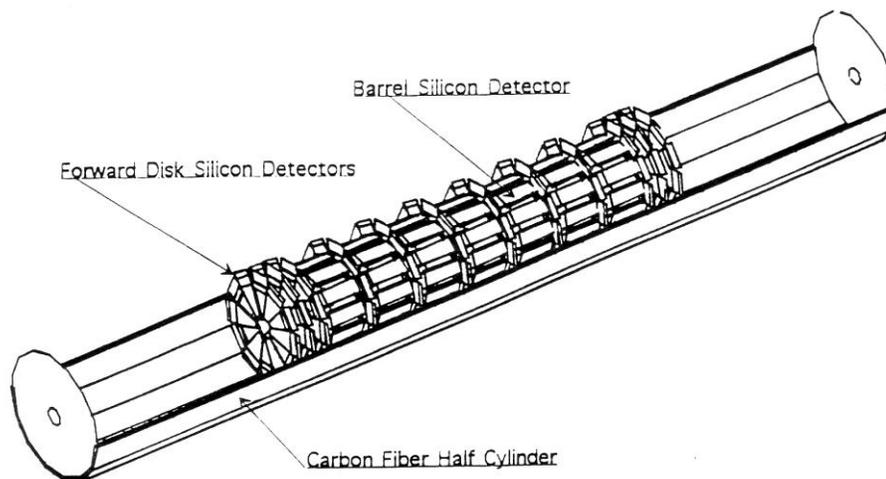


Figura 1.3.3 – Detector de Vértice (Silicon Tracker)

A estrutura do detector de vértice consiste de vários cilindros concêntricos ao longo da direção do feixe os quais são intercalados periodicamente por discos perpendiculares ao feixe conforme mostrado na figura 1.3.4. Existem quatro camadas de detectores de silício cada camada sendo formada por uma série de pedaços retangulares de wafers contendo um conjunto de detectores montados de forma a criar um barril concêntrico ao feixe. Outros wafers em forma de disco, com detectores posicionados de modo diferente são posicionados de forma perpendicular ao feixe a intervalos determinados de forma a detectar partículas com baixo ângulo. Em tal configuração a separação entre discos deve ser pequena de modo a minimizar erros na extrapolação das trajetórias, porém cada plano de discos introduz uma zona morta entre os barris o qual reduz a eficiência do detector o que implica num compromisso entre a distância entre discos e a eficiência total do detector. O Detector Dzero usa a seguinte geometria:

- Barris concêntricos divididos em sete partições ao longo do feixe cada uma com 12 cm
- Quatro camadas de detectores por segmento
 - A camada mais interna (camada um) e a camada três contem detectores no sentido axial.
 - Camadas dois e quatro contem dois detectores cada (um axial e outro deslocado de 2 graus de inclinação)
- 12 discos de pequeno diâmetro e com detectores a $+30/-30$ graus de inclinação – Discos F
- 4 discos de grande diâmetro com sensores em ambos os lados e com detectores a $+15/-15$ graus de inclinação estéreo – Disco H

A figura 1.3.4 mostra o layout do detector. Cada segmento do barril tem 12 cm e são separados por espaços de oito mm contendo os discos F. Os discos H e os discos F restantes (3) são localizados nos extremos do detector.

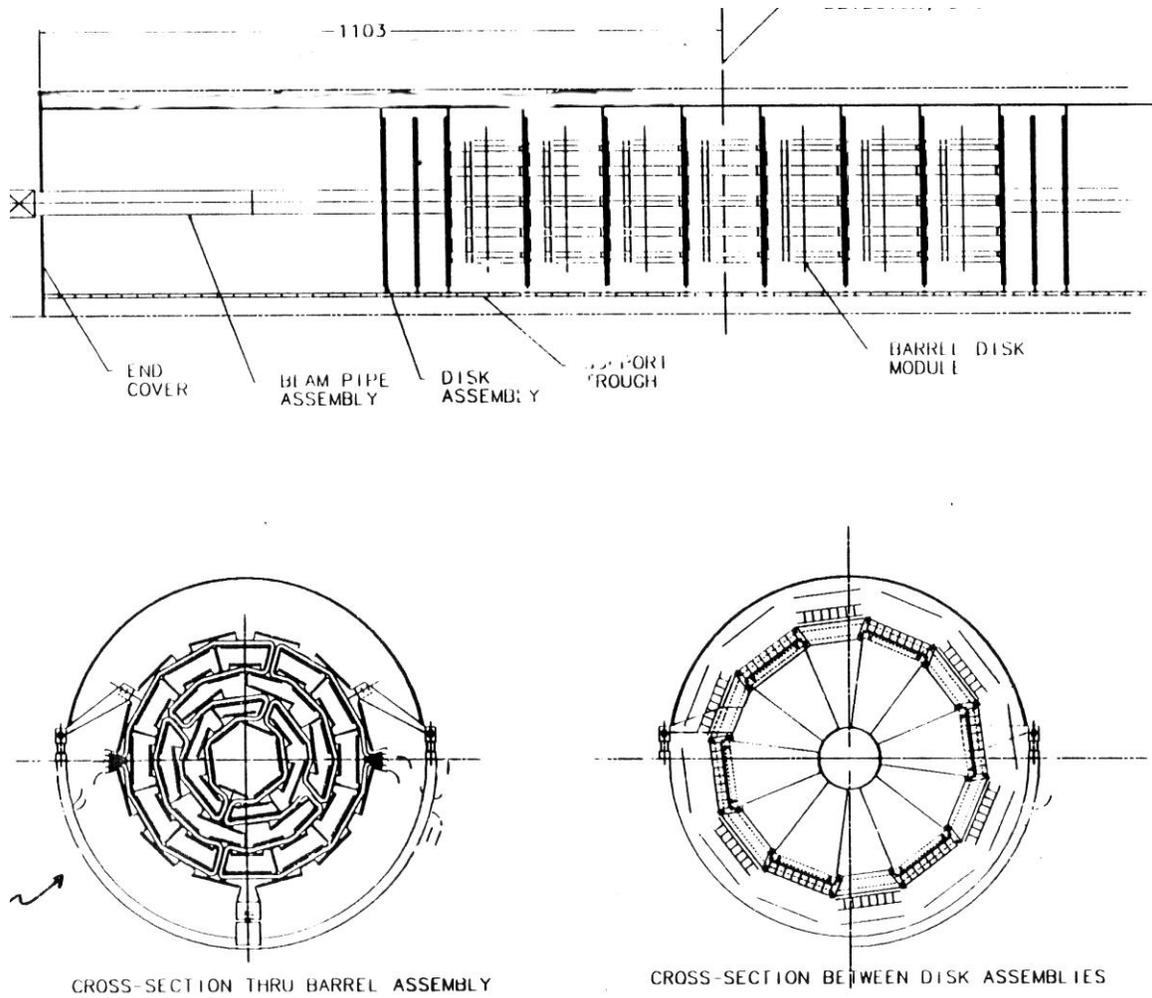


Figura 1.3.4(a) – Visão interna do detector de vértice (cilindros e discos)

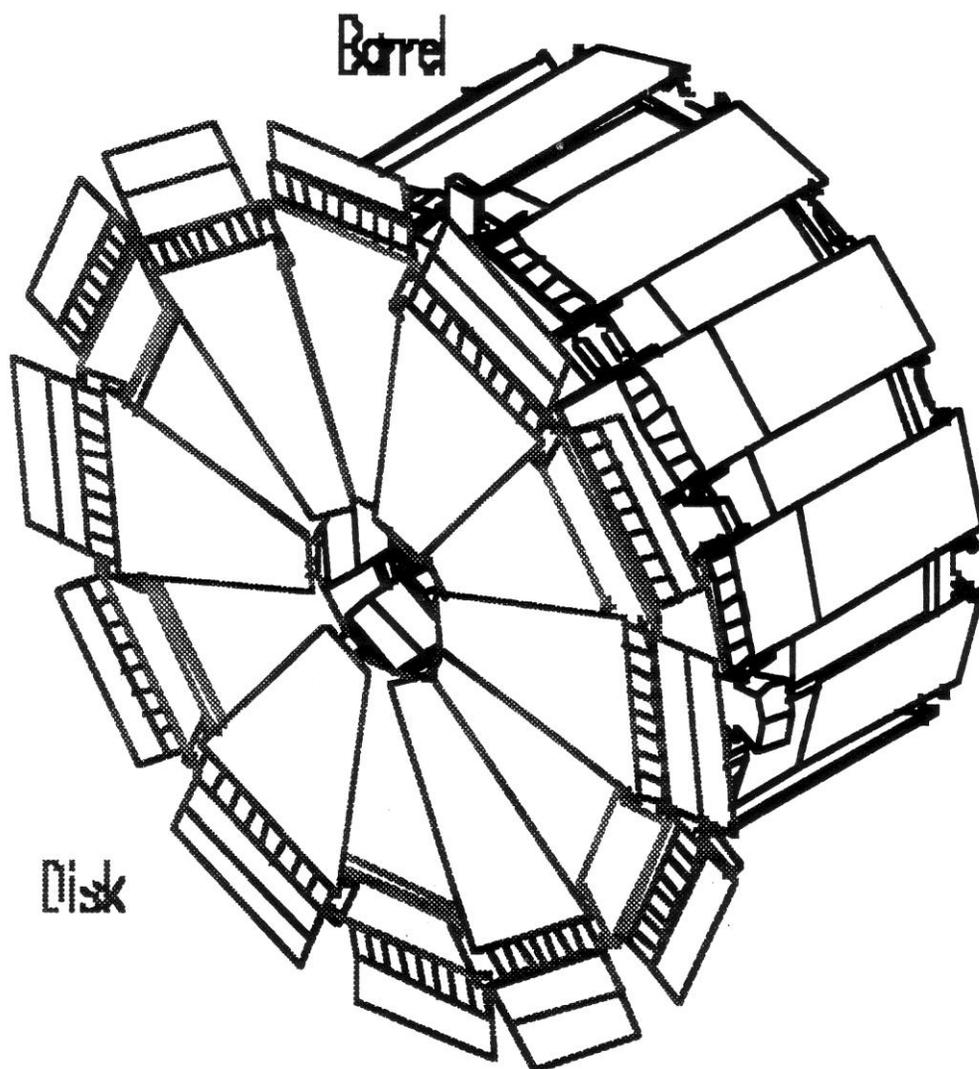


Figura 1.3.4(b) – Visão externa de uma seção do detector de vértice (cilindros e discos perpendiculares)

Os barris são baseados num detector de silício do tipo microstrip com linhas de sensores separadas uma da outra 50 μm e com cerca de 300 μm de profundidade, o que no geral nos dá uma precisão de detecção de cerca de 10 μm . O elemento básico é um diodo PIN reversamente polarizado (8). Os detectores são acoplados via capacitores (AC coupling) com cada linha de detectores possuindo seu próprio capacitor de acoplamento e resistor de polarização on-chip conforme mostra o layout do CI do detector na figura 1.3.5(a). A figura 1.3.5(b) mostra um corte transversal de uma das placas de silício, mostrando que elas são montadas duas as duas e com um ligeiro desalinhamento entre as linhas de sensores dos dois lados de forma a traçar a trajetória da partícula cruzando os dois lados do sensor. O SVXII é um circuito integrado desenvolvido pelo grupo do Dzero para efetuar a leitura dos detectores de silício. Este circuito integrado é montado sobre um circuito integrado de alta-densidade (HDI) usando kapton como substrato. Substrato este

que por sua vez é laminado sobre uma base de berílio de cerca de 300um e colado à superfície de cada uma das placas que compõem o cilindro do detector. Uma das extremidades do HDI consiste de um cabo flexível o qual faz a conexão com a eletrônica de processamento localizada externamente ao detector na plataforma-base do Dzero. A figura 1.3.6 mostra a montagem de uma das placas onde também se nota que sobre o substrato de Berílio, vemos duas placas de silício montadas (uma de cada lado) assim como dissipadores de calor (heat spreaders) que ajudam a remover calor do sistema.

A figura 1.3.5 mostra o layout do sensor e podemos ver que existem várias linhas conectadas via um resistor de Poly-silício a VDD. Estas linhas provêm a polarização reversa para os Diodos PIN. As linhas entre as linhas de polarização fazem conexão com as difusões P+ que formam um dos lados do diodo. Os resistores de poly-silício de alto valor (2 M Ω) servem como resistor de carga para o diodo indicando em qual das linhas uma partícula passou. O capacitor de acoplamento também é integrado na mesma pastilha. O anel de guarda (guard ring) é mantido no potencial de terra de forma a drenar qualquer corrente de fuga que ocorra na periferia do detector. Na Figura 1.3.5 (b) temos uma visão em corte do sensor mostrando as duas faces do detector, cada face tendo as linhas de detecção orientadas a 90 graus entre si. Esta figura mostra também uma partícula cruzando o sensor à aproximadamente 45 graus. O sinal é detectado por dois sensores neste caso. À medida que este angulo é reduzido (a partícula passa quase paralela ao plano do sensor) mais detectores passam a detectar a partícula e o detector perde resolução. A solução para este problema usada no Dzero foi o uso alternado de discos perpendiculares à trajetória. Estes discos apresentam uma alta resolução neste caso conforme ilustrado na figura 1.3.7.

Cada um dos SVXII dissipa cerca de 3 mW de calor. Se considerarmos que cada placa do detector central de silício tem cerca de 1280 canais isto equivale à produção de 4 W de potencia por placa. Se multiplicarmos por 1000, que é o número de placas que constituem o detector central, nós chegamos a 4KW de potência dissipada. Isto representa uma respeitável carga térmica gerada num volume relativamente pequeno (cerca de 20 cm de diâmetro e 1.5m de comprimento). Daí a necessidade dos dissipadores de calor (heat spreaders) embutidos entre os dois lados do detector de silício conforme mostrado na figura 1.3.6 para remover o calor gerado pelos chips. Na extremidade da placa o calor é transferido dos dissipadores pela circulação de liquido refrigerante através de mini tubos. Para maiores informações sobre este detector (9) é uma excelente referencia.

A eletrônica de leitura deste detector (o chip SVXII) será discutida em mais detalhes no capítulo dois quando abordarmos o detector de fibras cintilantes e o novo sistema de trigger proposto nesta dissertação. A figura 1.3.8 mostra uma foto do detector de silício durante a sua instalação no Dzero.

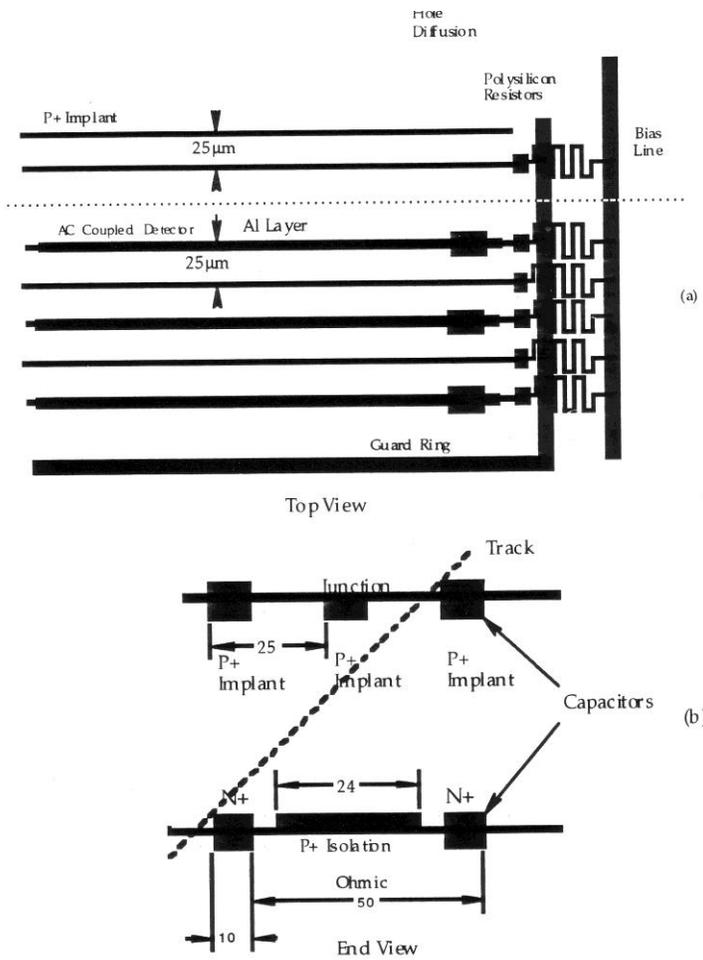


Figura 1.3.5 – Layout do detector de silício

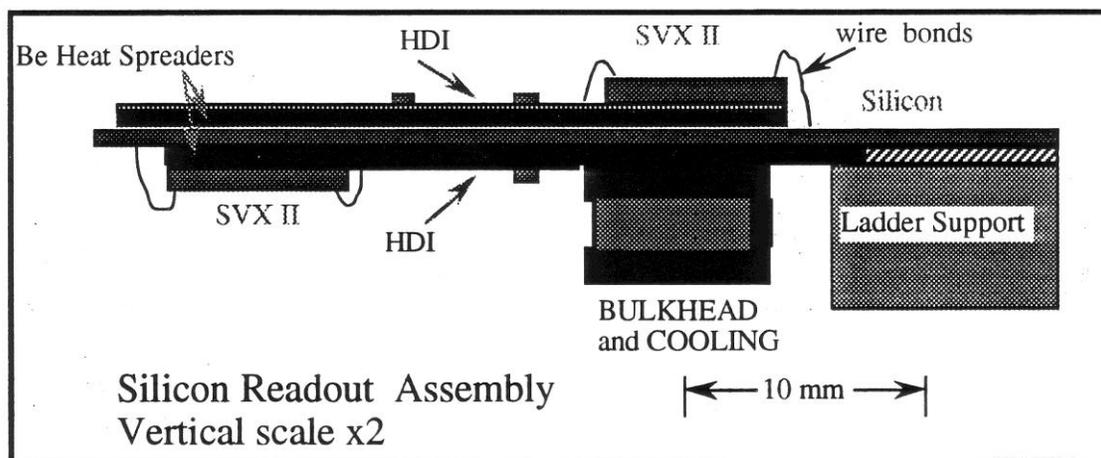


Figura 1.3.6 – Vista lateral de uma das placas do detector de silício mostrando o C.I. de leitura montado sobre esta.

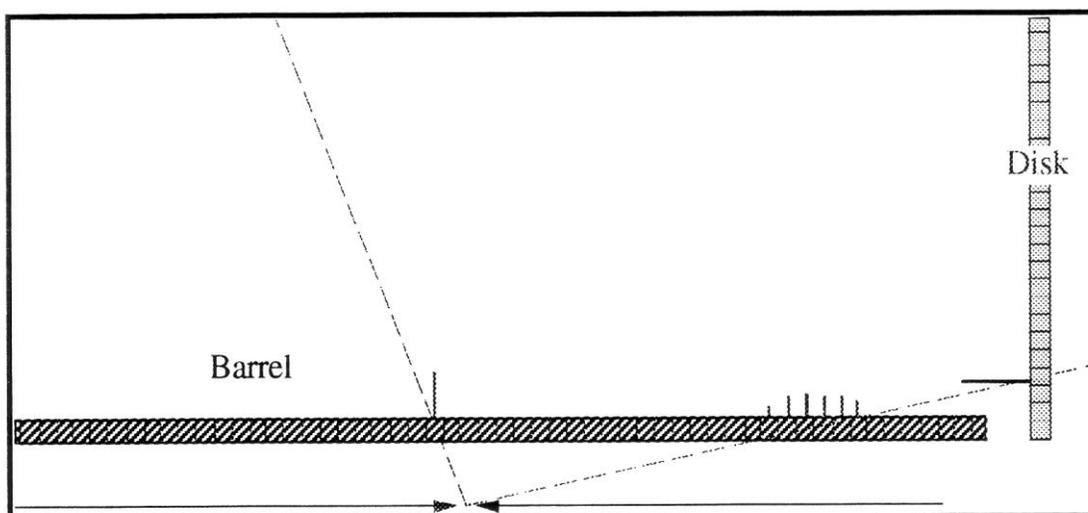


Figura 1.3.7 – Ilustração do efeito combinado do detector em barril e em disco para elevar a resolução de detector central de silício

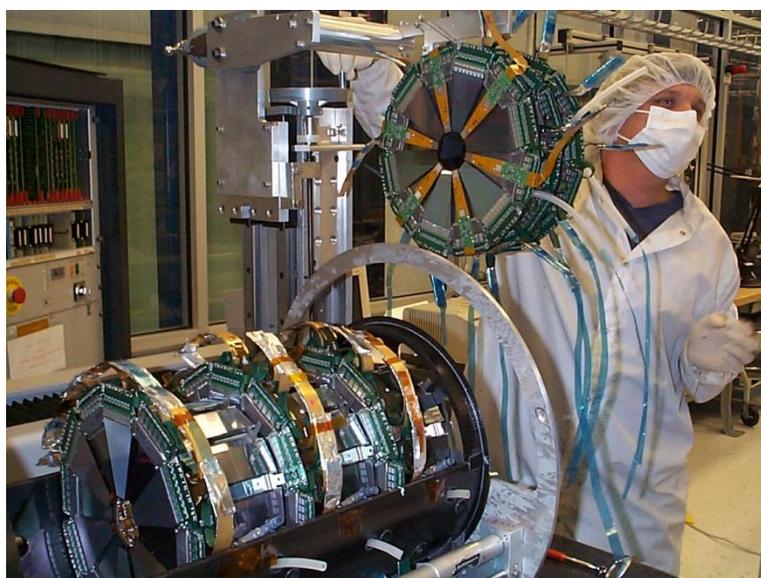


Figura 1.3.8 – Foto mostrando o detector central de silício durante sua montagem

3) O Detector de fibras cintilantes:

Ao redor do Detector de silício temos o detector de Fibras cintilantes (CFT-Central Fiber tracker). O CFT tem duas funções: em primeiro lugar em conjunto com o detector central de silício ele possibilita a reconstrução acurada de trajetórias e calculo do momento das partículas de maior energia gerada pela colisão. Em segundo lugar o CFT prove o trigger de alta velocidade de nível 1 (level 1 trigger). Este trigger é crítico, pois é o primeiro nível de trigger que seleciona quais das colisões serão salvas para análise pelo trigger nível 2. Assim ele tem de decidir de maneira irreversível e de maneira rápida quais das

colisões devem ser rejeitadas. A partir do próximo capítulo descreveremos uma proposta de trigger nível 1 baseado em processamento paralelo.

Uma das varias possibilidades propostas pelos físicos é combinar a informação do CFT com o Muon e o preshower detector para obtermos uma melhor definição para o trigger.

As Fibras cintilantes são mostradas na figura 1.3.2 com a legenda “Sci-Fi”. Um total de 80.000 canais de fibras são montadas em quatro cilindros concêntricos com um raio médio de 20 cm, 30 cm, 40 cm e 50 cm. Cada um dos cilindros serve de base para uma supercamada (“superlayer”) para quatro camadas doublet (doublet significa que cada camada é composta de 2 sub-camadas entrelaçadas). Duas das camadas doublet contem fibras correndo na direção axial paralela ao feixe. Estas duas camadas axiais são separadas por 1.5cm devido à espessura do cilindro de suporte (uma das camadas é montada internamente enquanto a outra é montada externamente ao cilindro de fibra de carbono). As outras duas camadas são orientadas de forma a formar a um angulo de +/- 1.5 graus a +/- 3 graus com a camada axial. Isto é uma das camadas esta orientada a +3 graus enquanto a outra mais interna esta orientada a -3 graus em relação às fibras axiais. Esta orientação melhora a precisão na reconstrução de partículas.

Todas as camadas doublet são construídas de forma que cada uma das camadas de fibras é deslocada de cerca da metade do espaçamento entre as fibras. Esta configuração elimina as zonas mortas entre as fibras e melhora ainda mais a resolução do detector.

As fibras cintilantes são feitas de um núcleo de poliestireno envolto em uma fina camada de acrílico a qual por sua vez é envolto por uma camada de um material fluorescente envolto em acrílico. Devido aos diferentes índices de refração das camadas acrílicas (1.59, 1.49 e 1.42), o efeito resultante é que a luz produzida fica a maior parte retida no núcleo onde ela se propaga até o fotodetector. Estudos feitos mostram que esta fibra apresenta um aumento de cerca de 70% na eficiência com respeito às fibras comuns além de aumentar a sua resistência mecânica.

As fibras têm um diâmetro mínimo de 830 um, o núcleo tem 770 um e cada uma das camadas tem cerca de 15um de espessura. A fibra emite uma luz entre o verde e o amarelo no espectro visível com um comprimento de onda de 530 nm.

Fibras multicamadas porem transparentes (clear fibers) de cerca de 8 metros são usadas para conduzir a luz para os fotodetectores onde elas são casadas com os fotodetectores com o uso de conectores óticos específicos. Os fotodetectores para o CFT ficam localizados na plataforma do detector em baixo do calorímetro.

Uma vez que a partícula ao cruzar a fibra produz um sinal bem fraco (não raro de apenas um fóton) o fotodetector usado no Dzero deve ser capaz de detectar um único fóton com uma alta eficiência, a uma alta velocidade e com um alto ganho. A solução foi um dispositivo de estado-sólido desenvolvido pelo Fermilab em conjunto com a Rockwell International. Chamado VLPC (Visible Light Photon Counter). O VLPC é baseado numa variação do principio do fotomultiplicador de estado-sólido e apresenta as seguintes características: 70% eficiência quântica operando no comprimento de onda do CFT, ganho de aproximadamente 20.000 e pode ser operado com eficiência máxima e simultaneamente com baixo ruído. Os VLPCs são fabricados sob a forma de “arrays” de 8 pixels circulares cada um com 8mm de diâmetro de forma a casar perfeitamente com as extremidades das fibras óticas que trazem o sinal do CFT. O VLPC opera a uma tensão de cerca de 7 Volts e necessita de uma temperatura ao redor de 6 – 8 K para máxima eficiência. Este último item requer que o VLPC seja imerso num ambiente criogênico.

Cassetes Criogênicos são módulos que foram projetados de forma a conter 128 arrays (1024 Canais) e mantê-los numa temperatura de operação estável (variações menores que ± 15 mK).

A figura 1.3.9 mostra uma foto do CFT onde podemos ver as várias camadas de fibras doublet que o compõe.

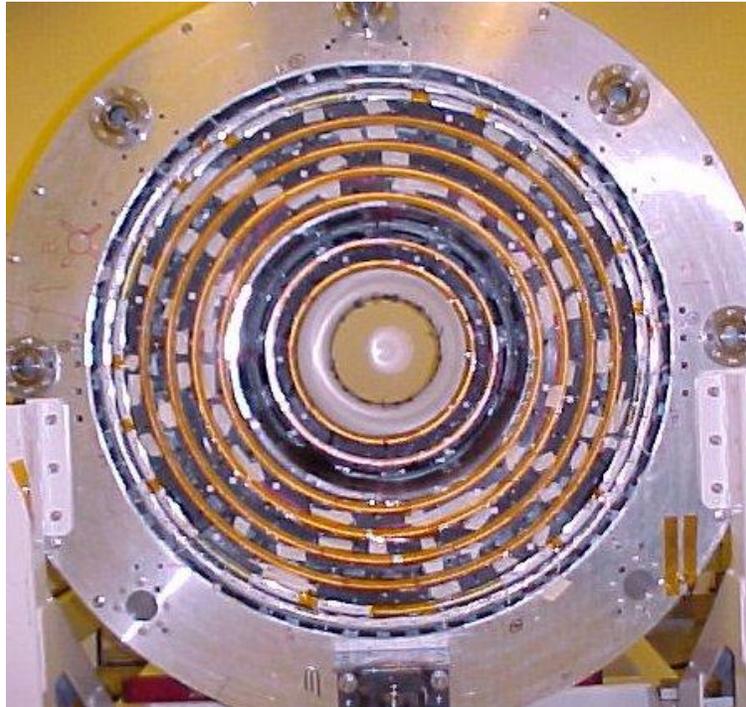


Figura 1.3.9 – Foto do CFT durante montagem

4) O Detector de Preshower

O detector de preshower está localizado ao redor do magneto supercondutor, o qual por sua vez envolve o detector central de silício e o CFT. Sendo assim o Preshower se encontra fora do campo magnético do solenoide.

O Detector de Preshower se divide em dois sub-detectores: O detector central e o detector de extremidade (forward).

O detector central de preshower auxilia a identificação de elétrons e fornece informações que permitem a correção de quaisquer distorções na trajetória devido ao campo eletromagnético do solenoide não ser homogêneo. Ele funciona como uma espécie de pré-calorímetro uma vez que no seu exterior ele é revestido de uma camada de chumbo de forma a produzir um decaimento das partículas. Além disso, o Preshower faz parte do sistema de trigger conforme iremos ver no próximo capítulo. O preshower central tem uma forma cilíndrica e está alojada no espaço entre o solenoide e o calorímetro (cerca de 51 mm disponíveis) e consiste de várias camadas de fibras cintilantes. Estas fibras estão dispostas tanto em camadas no sentido axial assim como em camadas formando ângulos positivos e negativos de 20 graus em relação às camadas de fibras axiais de forma similar ao CFT. Uma visão deste sensor é mostrada na figura 1.3.10 assim como uma visão em

corte das fibras cintilantes. Cada um dos lados da fibra tem cerca de 5 mm o que produz um espaçamento entre fibras de 5 mm. A luz emitida pelas fibras é transmitida por fibras transparentes (clear fibers) até os VLPC. A eletrônica de leitura deste sensor é a mesma do CFT e será vista em detalhes no capítulo 2.

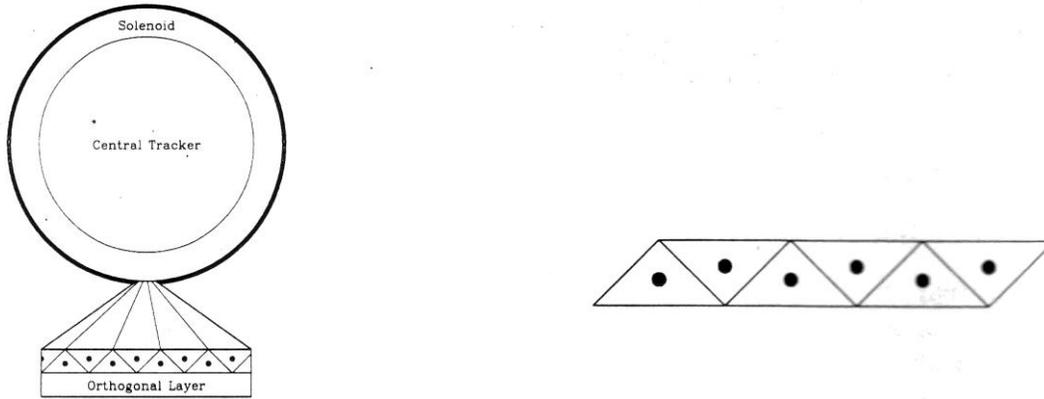


Figura 1.3.10 – Vista em corte do preshower central e das fibras

O CFT e o central preshower usam a mesma tecnologia de fibra cintilante o que permite uma medida rápida do momento e da trajetória da partícula. Por esta razão estes dois sensores serão usados para formar o trigger nível 2. Quando adicionamos a informação do preshower central à informação do CFT os físicos podem provar por simulações de Monte-Carlo usando o Geant que a detecção melhora por um fator de 5 vezes, ou seja, os triggers falsos no fim da cadeia de triggers é reduzida por um fator de 5.

O detector preshower central contém 5760 canais de leitura o que o torna muito complexo para usá-lo no trigger nível 1.

O sensor preshower de extremidade (Forward preshower sensors - FPS) representa o outro subsistema que constitui o preshower. Ele consiste de uma série de estruturas em forma de cunhas montadas juntas de forma a criar um detector cônico que funciona como uma espécie de tampa nas duas extremidades do detector conforme mostrado na figura 1.3.11. Cada cunha consiste de um material absorvente (chumbo) contida entre duas camadas de sensores (fibras cintilantes). A camada antes do material absorvente é usada para traçar as trajetórias e os sensores após a camada de chumbo é usada para detectar os jatos de partículas causados pelo material absorvente.

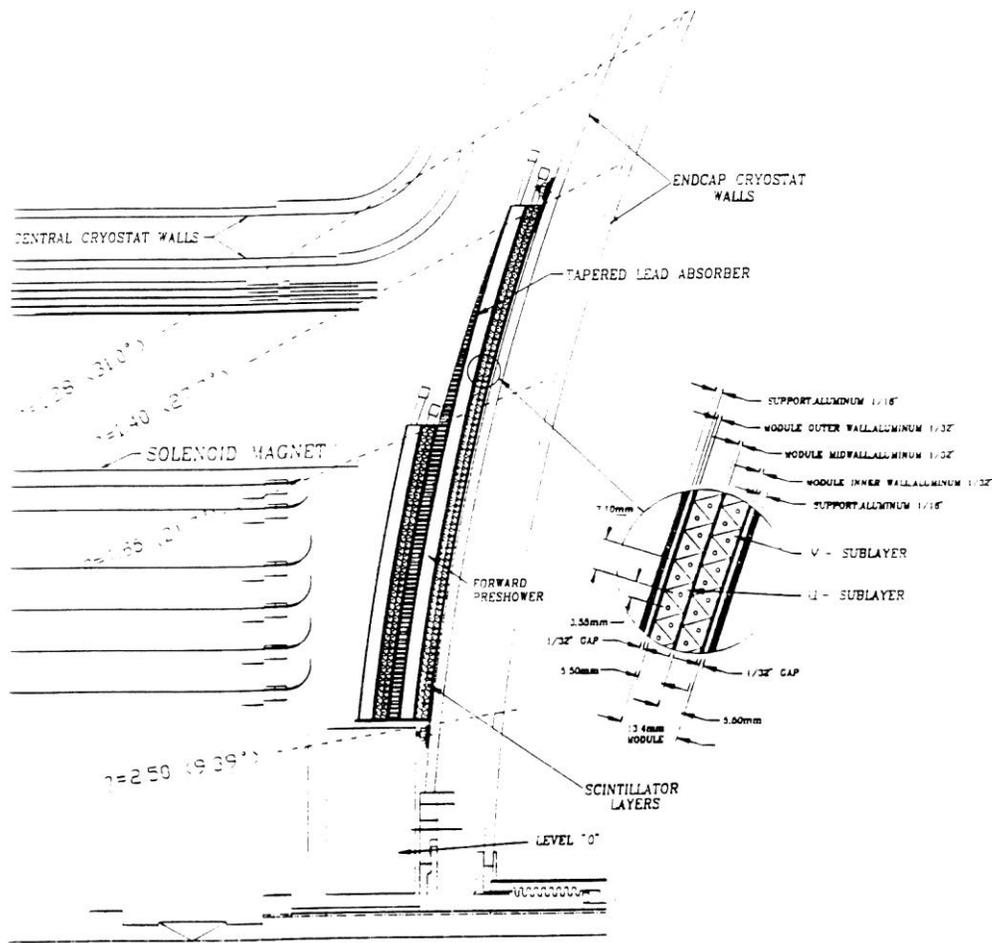


Figura 1.3.11 – Vista em corte do FPS

5) O Calorímetro

Os sensores do Calorímetro não sofreram nenhum upgrade para a run II. O calorímetro no Dzero consiste num grande numero de módulos. Cada um dos quais consiste de várias camadas de placas de material absorvente (titânio/chumbo) intercaladas com sensores de fio. A figura 1.3.12 mostra uma visão em corte de uma seção dos módulos. As camadas absorventes são separadas dos sensores de fio por um espaço de 2.3 mm preenchido por Argônio liquido. Os sensores consistem de fios de cobre revestidos por uma camada de material isolante (G10). No exterior do revestimento uma fina camada de epóxi condutivo de baixa resistividade é depositada. Durante a operação do detector as placas absorventes são aterradas enquanto uma voltagem de cerca de 2.0- 2.5 KV é aplicada aos fios de cobre. Quando as partículas atingem a camada absorvente e decaem num jato de partículas, estas ao cruzarem o argônio liquido produzem uma trilha de íons. Os elétrons liberados são atraídos para o alto potencial do fio aonde chegam depois de um curto intervalo de drift de cerca de 450 ns. Porem um sinal é induzido capacitivamente no revestimento externo pelos elétrons ao cruzarem a camada a caminho do fio interno e

transmitido para a eletrônica de leitura. Para podermos detectar a posição exata aonde a partícula cruza o revestimento, este é subdividido em pequenos segmentos, cada segmento associado a um canal dedicado.

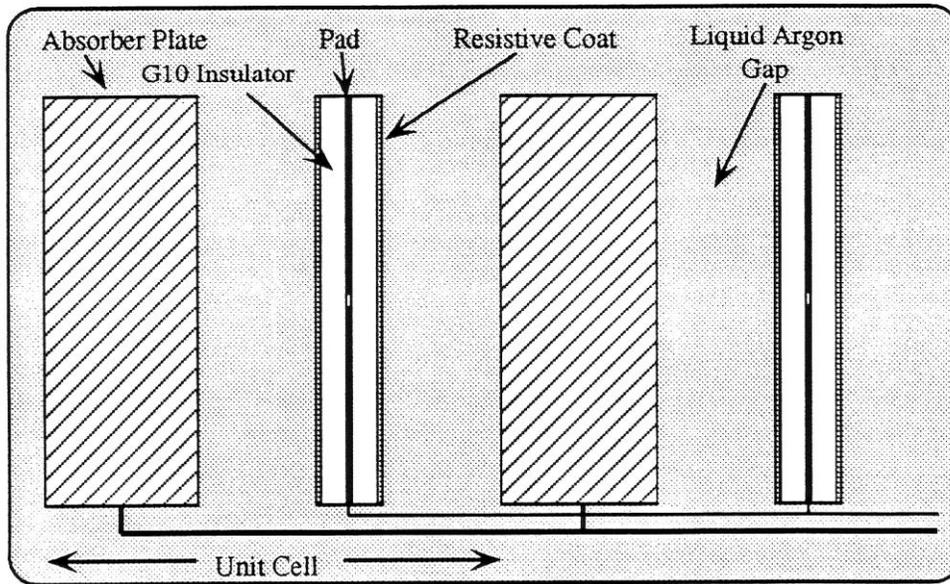


Figura 1.3.12- Uma célula do sensor do calorímetro

Uma vez que o Argônio necessita de Criogenia, todos os módulos são alojados dentro de um Criostato de forma a manter a temperatura constante. Porém para facilitar o acesso ao interior do detector, o calorímetro é dividido em três partes principais cada um com um criostato separado: O calorímetro central e os dois calorímetros de extremidade.

O calorímetro central tem um formato toroidal e consiste de 3 camadas concêntricas de módulos. A camada interior contém 32 módulos os quais são espessos o suficiente para conter a maioria das partículas. A camada intermediária contém 16 módulos com segmentos pequenos para uma maior acurácia enquanto a camada exterior contém 16 módulos porém com uma segmentação mais fina. Os números de canais para leitura somam 10.368 canais para a camada interna, 3.000 canais para a camada intermediária e 1224 canais para a camada externa. Estes números indicam que o cabeamento para todos estes sensores envolvem um elaborado projeto mecânico

O Calorímetro de extremidade é colocado fechando os dois lados do calorímetro central de forma a cobrir as partículas que deixam o detector a baixos ângulos. Estes calorímetros também contêm 3 camadas de módulos sensores porém as geometrias dos sensores é totalmente diferente. O centro do detector consiste de um módulo em forma de disco seguido por duas outras camadas de módulos em forma de cilindro. Montados em forma de anel ao redor dessas 3 camadas temos uma outra camada de módulos.

Uma visão destas camadas assim como a segmentação dos módulos por ângulo de partida pode ser vista na figura 1.3.13 abaixo.

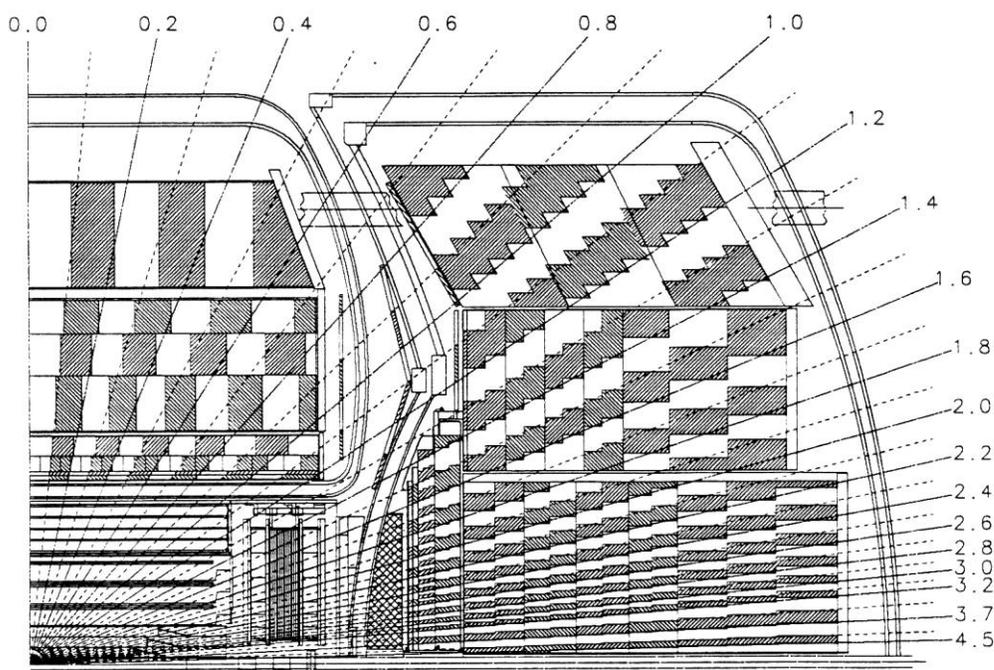


Figura 1.3.13 – Sensores de Calorímetro e sua segmentação

Em termos de eletrônica os sinais do calorímetro passam por um amplificador de carga (charge amplifiers) que além de elevar sua amplitude ainda modificam a forma do pulso de maneira a facilitar a sua detecção. Essencialmente o pulso é integrado de forma a termos o pico da amostragem cerca de 400 ns após o início do pulso e termos um sinal unipolar simétrico. O pulso é tão rápido que técnicas de RF são usadas no projeto dos conectores e a impedância dos cabos que conduzem os pulsos são casadas com a impedância dos amplificadores. Como os cruzamentos de prótons e anti-prótons ocorrem a cada 396 ns e a amostragem é feita a cada 400 ns o uso de uma estrutura em pipeline é obrigatória. O sinal é amostrado em dois instantes: antes da colisão e no pico de amostragem, a diferença entre estas duas amplitudes é enviada para o conversor A/D. O sinal proveniente da amostragem do pico do sinal também é enviado para o sistema de trigger, onde ele é usado para formar o trigger nível 3. O calorímetro requer 12 bits de precisão nos conversores A/D. O pipeline Analógico e os conversores A/D são implementados num C.I. usando-se circuitos a capacitor chaveado.

Finalmente a figura 1.3.14 mostra uma foto do Calorímetro central mostrando ainda o CFT sendo introduzido no interior do solenoide durante a montagem final do detector.

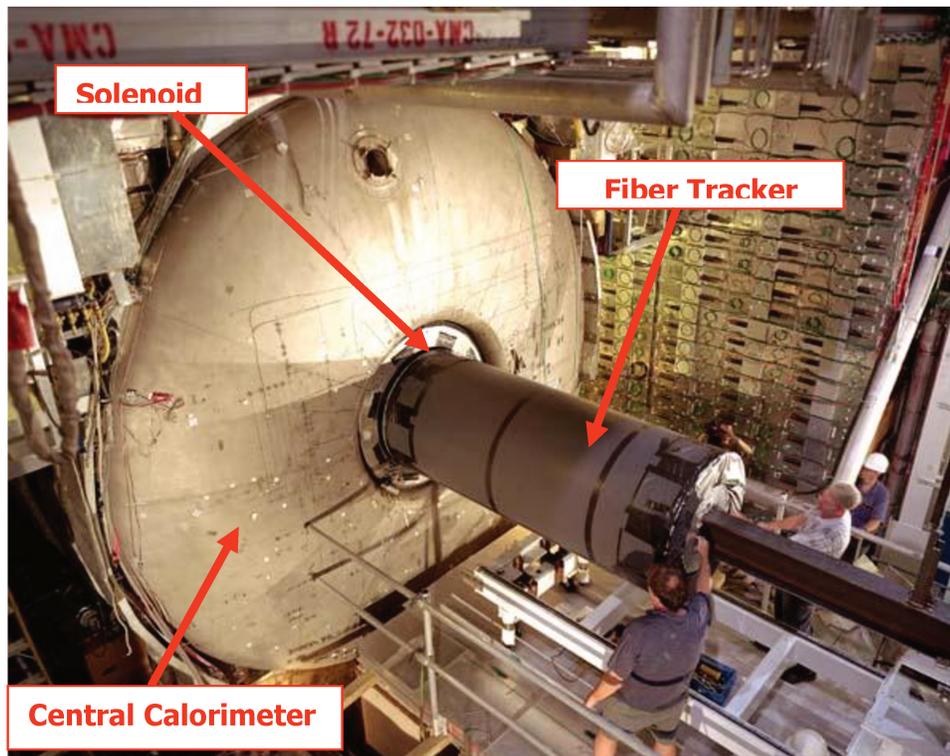


Figura 1.3.14 – Vista do Calorímetro durante a instalação do CFT

6) O sensor de Muons

O sensor mais externo do detector Dzero é o sensor de Muons. Apesar dos muons decaírem em elétrons, o seu tempo de vida médio é bem longo (2.2us), de forma que, do ponto de vista do detector, os Muons não interagem de nenhuma forma com o detector. Ainda mais se considerarmos que sua massa é muito grande (aproximadamente 200 vezes a do elétron) para produzir um jato eletromagnético no calorímetro. Assim o Muon cruza sem interagir todas as camadas do Dzero e é a única partícula que cruza a câmara de muons onde é enfim detectado.

O sistema de Muon do Dzero consiste de 5 magnetos em forma de toróide os quais são circundados por 3 camadas de câmaras de fio similares as existentes no Calorímetro. Esta estrutura está mostrada na figura 1.3.15.

As câmaras de fio medem a trajetória dos muons antes e depois de passarem pelo magneto, desta maneira podemos medir o momento dos muons. Os cinco magnetos conforme indicado na figura são: CF (central), dois EF (extremidades) e dois SAMUS (small angle Muon system) cobrindo os ângulos pequenos. O CF e os EF são coletivamente chamados de WAMUS (wide angle muon system). Cada uma das seções tem uma camada de câmaras de fio antes do magneto (camada A) e outra camada após o magneto (camada B). Existe ainda uma terceira camada (camada C) envolvendo o detector a cerca de 1 – 3m distante do magneto. Como é preciso suportar mecanicamente o detector, algumas regiões na parte inferior do detector não possuem estes sensores.

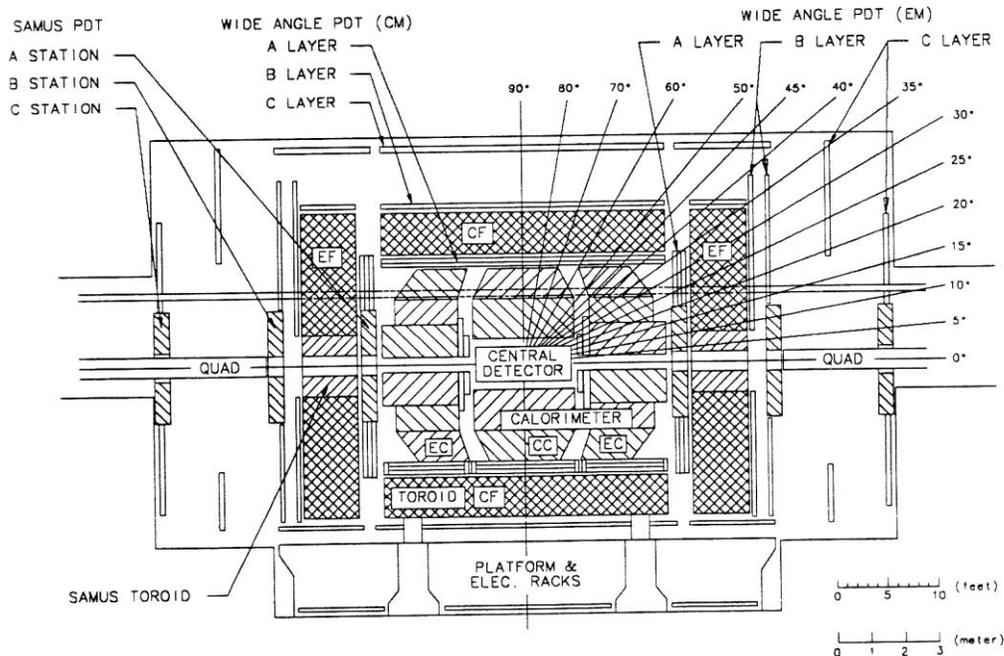


Figura 1.3.15 – Sensores de Muons

As câmaras de fio que formam WAMUS têm a seção perpendicular retangular com um fio sensor por fio. Elas são formadas de alumínio extrudido em módulos retangulares contendo 4 planos de câmaras de fio (camada A) ou 3 planos de câmaras (camada B e C). Existem 164 módulos cada um contendo entre 14 e 24 drift tubes de comprimentos variáveis entre 191 e 579 cm. Os tubos estão orientados de maneira paralela a direção do campo magnético. Deste modo a deflexão das partículas devido ao magneto é medida pelo tempo de deriva (drift time). A distância da trajetória de passagem de uma partícula a um fio sensor é medida com uma resolução de cerca de 0.5 mm. A figura 1.3.16 nos dá uma ideia da construção destas câmaras.

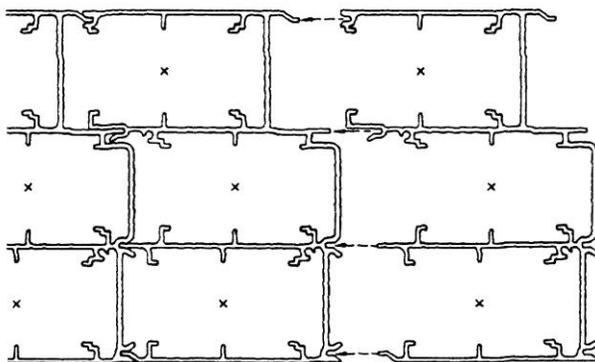


Figura 1.3.16 – Vista perpendicular da Câmara de fios usada nos sensores de Muons

Devido ao espaço disponível no Dzero não podemos ter acesso a ambas as extremidades dos fios, assim a leitura dos pulsos é feita por apenas uma das extremidades dos fios. Para isto cada par de fios adjacentes são conectados na sua extremidade inacessível.

A eletrônica de leitura mede o tempo de chegada dos pulsos chegando em cada extremidade assim como ela calcula a diferença de tempo entre os pulsos chegando às extremidades de cada par de fios interligados. Usando esta diferença de tempo podemos estimar a posição da partícula ao longo do fio com uma precisão entre 10 – 20 cm.

Para obtermos uma medida mais precisa da posição inserimos no topo e no fundo de cada câmara um conjunto de pads de catodo do tipo vernier (Vernier Cathode pads) conforme mostra a figura 1.3.17. Estes pads são feitos de um material isolante (Glasteel) coberto com um fino revestimento de cobre e formando um desenho em forma de diamante o qual se repete com um padrão de cerca de 60 cm. O revestimento de cobre se divide em uma região interna e outra externa. Como já sabemos os elétrons produzidos pela ionização do gás ao chegarem ao fio detector irão induzir pulsos no pads de catodo. A razão entre a carga depositada na área externa e a carga depositada na área interna do catodo pode ser usada para localizar a partícula com uma precisão de cerca de 3 mm quando usada em conjunto com a informação dos fios (o catodo determina em qual segmento de 60 cm a partícula passou). Além disso, os catodos em camadas adjacentes são desalinhados em cerca de 10 cm entre si de forma a resolver ambiguidades que por acaso ocorram nos cantos do padrão em diamante. Cada tubo também fornece um sinal de 1 bit se este tiver um pad de catodo atingido. Estes bits formam a entrada para o trigger de muon.

A informação das trajetórias do Muon antes e depois do magneto são medidas pelas camadas B e C (após magneto) e pela camada A (antes magneto) e são usadas para cálculo do momento da partícula durante a reconstrução do evento. Além disso, as trilhas na camada A são casadas com trilhas nos detectores centrais e no calorímetro de forma a eliminar trilhas falsas.

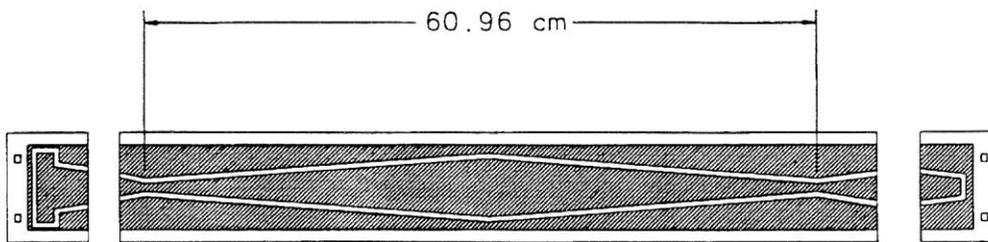


Figura 1.3.17 – Pad de catodo usado na câmara de Muons

Devido à alta densidade de trilhas nas extremidades, o sistema SAMUS usa câmaras de fios menores. Ele é composto de três estações cada uma composta de 3 camadas de câmaras de detectores. Cada plano é composto de dois sub-planos os quais estão desalinhados entre si por meia largura da câmara além de terem diferentes ângulos de rotação entre si. A construção mecânica das câmaras é a mesma do WAMUS porém não há a necessidade do Pad de catodo, uma vez que a câmara já é bem menor.

Em termos de polarização os fios são mantidos a um potencial de cerca de 4.5 kV e o pad de catodo (WAMUS) é polarizado a 2.3 kV. Um gás composto a base de Argônio é usado no WAMUS enquanto no SAMUS usa-se o gás CF_4 . O número de canais é de 11386 canais no WAMUS e 5308 no SAMUS.

Em termos de eletrônica de leitura, os sinais vindos dos fios são enviados para amplificadores de baixo ruído e com acoplamento a transformador de modo a evitar caminhos para a terra de baixa impedância entre as câmaras e os amplificadores.

A medida dos tempos de chegadas dos pulsos é feita por um C.I. dedicado feito pelo Laboratório KEK no Japão. A digitalização da carga detectada é feita por um conversor A/D de 15 Milhões de amostras por segundo e com 10 bits de resolução. Os sinais para os A/D passam por um pipeline de modo a adequar a taxa de chegada dos pulsos com a velocidade de conversão do conversor A/D.

Todas as camadas de sensores proveem dados que nos permitem traçar as trajetórias de quase todas as partículas geradas pela colisão próton - antipróton. Entretanto se fossemos gravar todas as centenas de milhares de canais de informação referente a todos os sensores, teríamos que ter uma capacidade em disco que não seria viável. E de fato nem seria uma boa ideia, pois a maioria das colisões produzem resultados já conhecidos de longa data. O que os físicos procuram são tipos específicos de colisões que ocorrem muito raramente. Portanto faz sentido termos um sistema que filtre toda esta informação de forma que só as colisões mais interessantes sejam registradas em fita. Esta é a tarefa do sistema de trigger, o qual iremos descrever no próximo capítulo.

Capítulo 2 – O sistema de trigger e o sensor CFT.

2.1- Princípio básico de um sistema de trigger

Uma vez que já vimos que não é viável armazenarmos todos os dados de todos os detectores durante todas as interações, devido ao fato de que a maioria dos resultados não oferece informações sobre novos fenômenos, temos que usar alguma forma de filtragem.

Quando uma interação ocorre num detector o sistema de sensores e de aquisição de dados registra o evento num buffer ou numa memória temporária. Esta etapa já reduz a taxa de eventos para um nível aceitável e após isto armazena as interações remanescentes numa fita magnética.

Conceitualmente um sistema de trigger em um experimento de alta energia pode ser considerado como um sistema em pipeline de múltiplos estágios como ilustrado na Figura 2.1.

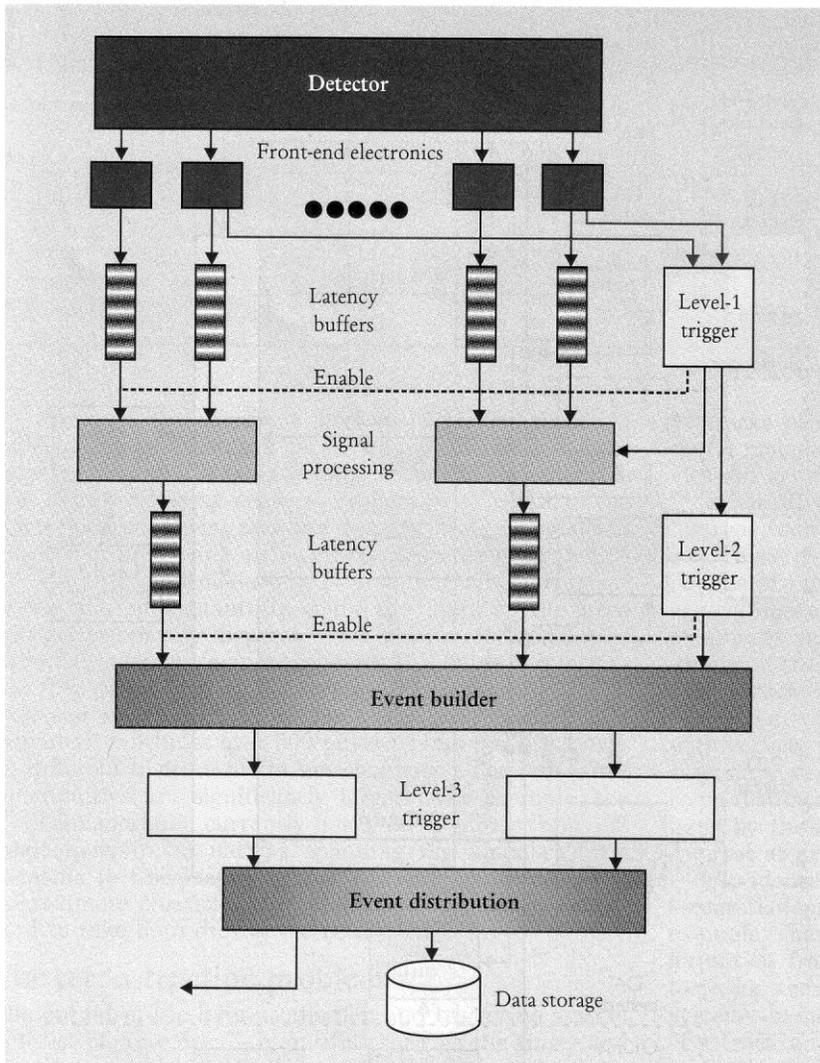


Figura 2.1.1 – Conceitos de um sistema de trigger

Dentro de um detector como o Dzero as colisões entre prótons e anti-prótons (eventos) ocorrem a uma taxa muito rápida para serem Analisadas e armazenadas pelo sistema de aquisição de dados e armazenando de um dado experimento. De qualquer maneira a maioria dos eventos são irrelevantes do ponto de vista da Física de partículas envolvida ou contem dados que podem não ser utilizáveis. Assim ao mesmo tempo em que a informação de um dado evento é armazenada na memória temporária, processadores de trigger fazem uma rápida e aproximada reconstrução de evento de modo a eliminar os eventos não interessantes. Os eventos que são aceitos são enviados para o próximo nível de memórias temporárias (queues). Uma vez que o trigger nível 1 opera com as taxas de eventos mais altas (em termos de eventos/segundo) ele tipicamente usa informações somente de um fração dos sensores disponíveis de modo a reduzir o numero total de canais que ele tem de processar durante a reconstrução. O Trigger nível 1 geralmente consiste de circuitos eletrônicos dedicados (ASIC) projetados para operar a uma alta frequência de eventos.

O trigger nível 2 e os triggers abaixo deste trabalham com taxa bem menores e assim são implementados usando-se processadores comerciais de alta velocidade para efetuar a reconstrução mais detalhada do evento (trajetória). Normalmente estes níveis de trigger (sobretudo o nível 3) são implementados usando-se redes de processadores operando em paralelo de forma que cada processador trabalhe reconstruindo um pequeno fragmento de cada evento a partir dos dados armazenados de todos os sensores.

2.2- O sistema de Trigger do Detector Dzero

Após o upgrade em 1997 o Tevatron teve a sua luminosidade elevada (luminosidade essencialmente é o numero de prótons por pacote) de 1.52×10^{31} partículas/cm² para 1.01×10^{32} partículas/cm² com o Main injector operando a plena carga.

Além disso, o espaço entre colisões será paulatinamente reduzido dos 3.56 μ s atuais para 396 ns. O aumento da luminosidade e a redução do intervalo entre colisões requerem mudanças no detector (para suportar as altas doses de radiação que inevitavelmente danificarão os sensores internos) assim como no sistema de trigger (para lidar com as altas frequências de colisão).

A figura 2.2.1 mostra o timing das colisões: As colisões são espaçadas de 396 ns entre si e estas estão agrupadas em “superbunches” cada um com 12 colisões, Entre os superbunches há um intervalo de 2.64 μ s o qual o acelerador necessita para realinhar o feixe e re-calibrar a energia. Este intervalo também é usado pela eletrônica de leitura para re-calibrar os conversores A/D e quaisquer outros ajustes necessários.

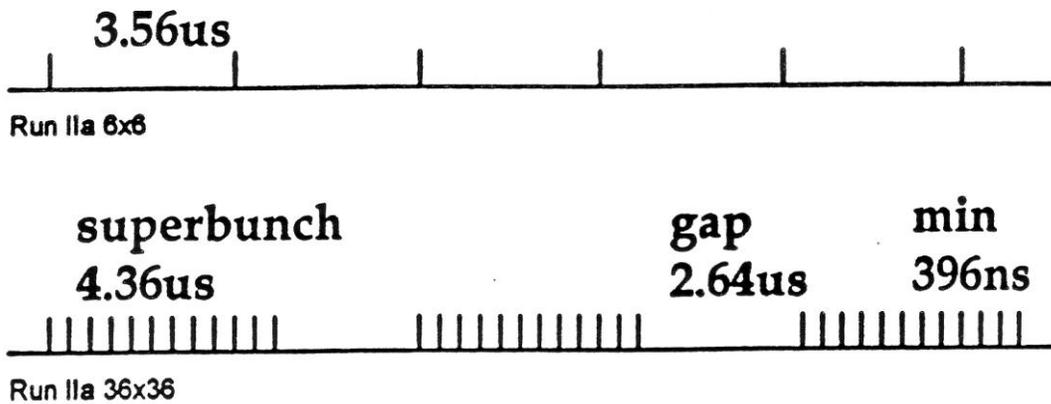


Figura 2.2.1 – Timing das colisões

No capítulo 1 descrevemos todos os sensores do detector Dzero assim como descrevemos de maneira sucinta a sua eletrônica de leitura. A figura 2.2.2 mostra uma visão geral da eletrônica de leitura de cada sensor que compõe o detector e sua eletrônica de leitura associada.

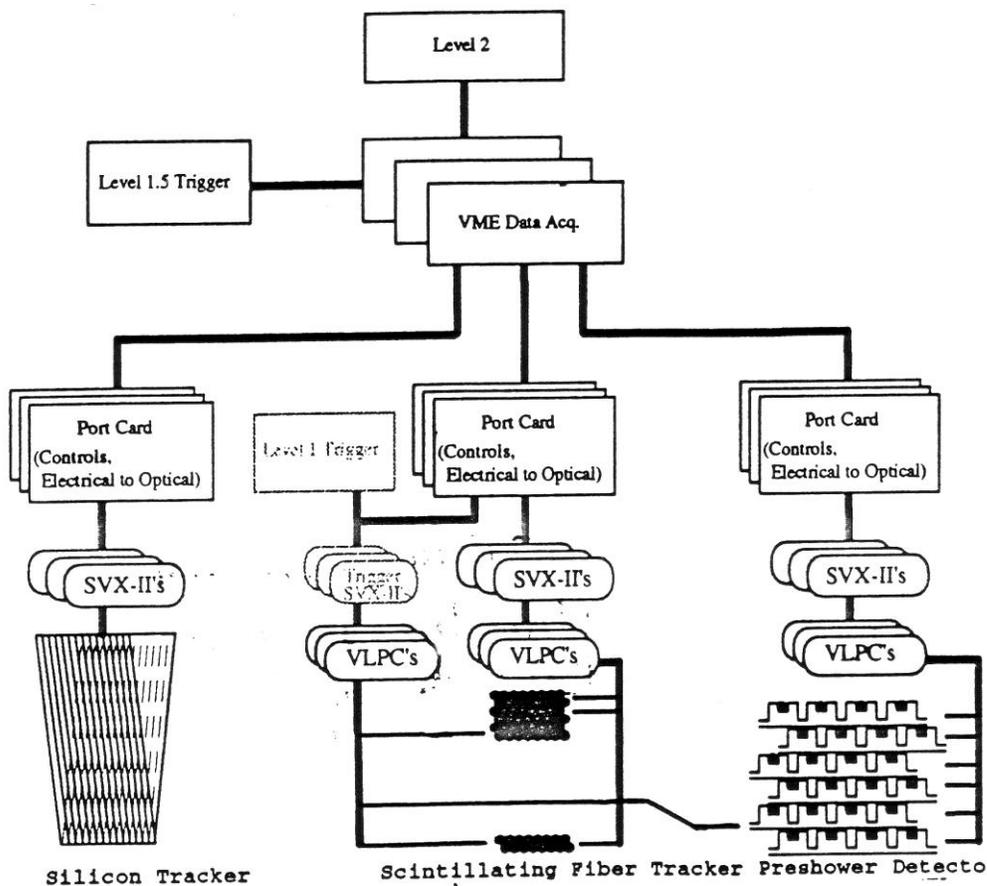


Figura 2.2.2 – Eletrônica de Leitura dos sensores centrais

Conforme já visto as fibras são terminadas em fotodetectores chamados VLPC, especialmente projetados para serem extremamente sensíveis ao comprimento de onda da luz emitida pelas fibras. Após o VLPC o sinal passa por um Circuito integrado CMOS com um estágio amplificador de baixo ruído cuja função é melhorar a relação sinal-ruído na entrada do detector. Uma visão simplificada deste CI é mostrada na figura 2.2.3.

Como vemos, ele consiste de dois estágios amplificadores CMOS de baixo ruído e um comparador, além de várias chaves. As chaves SW2 e SW3 em conjunto com o Capacitor C2 amostram o sinal na saída dos amplificadores de modo que o C.I. responsável pela conversão A/D (chamado SVXII) tem um sinal estável para digitalizar. O comparador compara o sinal na saída do primeiro amplificador com um valor de referencia de forma a indicar se a fibra associada foi atingida ou não. É este sinal digital que o trigger nível 1 usa como entrada para sua decisão.

Este amplificador extra é necessário apenas para o VLPC devido ao seu reduzido sinal, como também para amostrar o sinal proveniente do VLPC e rapidamente envia-lo para o sistema de trigger nível 1 (a latência do SVXII é de cerca de $6\mu\text{s}$ o que é excessivo para o trigger L1). Os sinais provenientes do detector central de silício, devido a sua maior amplitude, são enviados diretamente para o SVXII para digitalização.

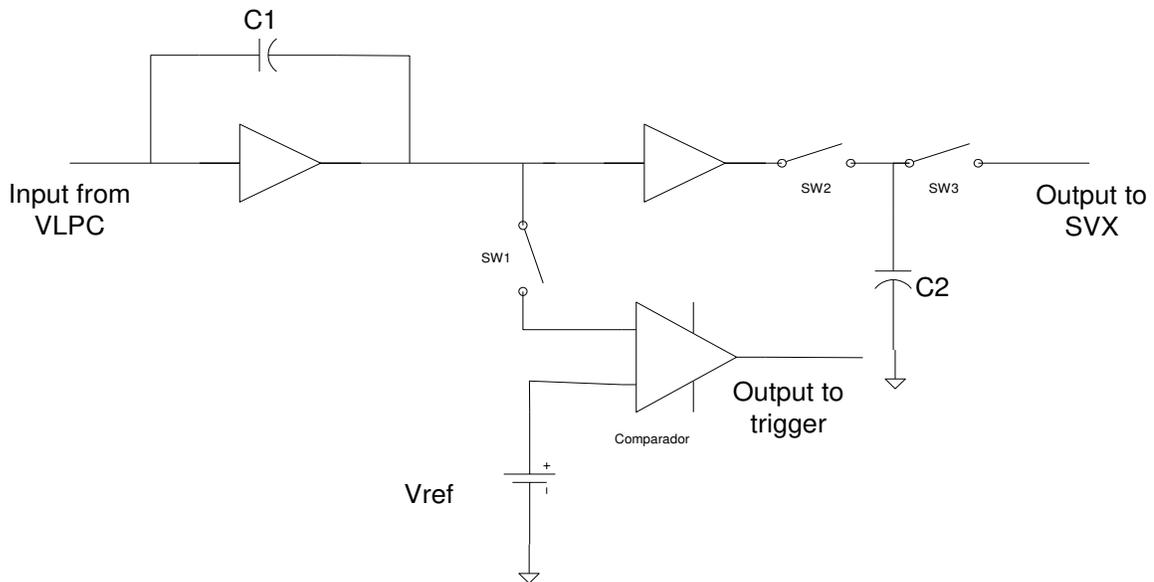


Figura 2.2.3 – Diagrama simplificado do Pré-amplificador CMOS usado no CFT

Como mostrado acima o pré-amplificador também integra a carga recebida do SVXII e a converte em tensão através do capacitor C1.

Os detectores de fibras cintilantes do CFT, do preshower e o sensor central de silício são digitalizados por um mesmo C.I. : o SVXII projetado pela Universidade da Califórnia Lawrence Livermore Laboratory. Devido a sua importância tanto na leitura dos sensores

como na formação do sinal de trigger, é importante entendermos o seu funcionamento a partir da figura 2.2.4.

A primeira seção do chip é um pré-amplificador de carga de baixo ruído. Ele essencialmente integra a carga que recebe dos detectores no capacitor C_f . A chave S_a reseta a carga armazenada no integrador após cada cruzamento de feixe. Entretanto isto não funciona quando os intervalos entre cruzamento são tão baixos, pois a resistência da chave (mesmo se fizermos o transistor MOS ter uma alta razão W/L) em conjunto com o capacitor de integração forma uma constante RC parasítica muito alta. Este problema foi resolvido no SVXII pelo uso do método de dupla amostragem (double sampling), amostrando-se o sinal antes e depois do cruzamento de forma a armazenar a diferença. Porém a chave é mantida, pois ainda temos que resetar o integrador de forma a evitar que a saída do amplificador operacional atinja VCC. No entanto este reset é aplicado somente durante os espaços entre os superbunches.

Após o pré-amplificador temos uma linha de atraso, essencialmente um shift register analógico (pipeline) de 32 estágios. Este estágio nos permite armazenar até 32 eventos (cruzamentos) do sinal de forma a criar um intervalo de tempo até que o trigger nível 1 decida qual sinal é interessante o suficiente para passar para a análise do trigger nível 2. Como cada cruzamento ocorre a cada 396 ns o SVXII nos dá um tempo máximo de pipeline de 12.67 μ s. Assim o trigger nível 1 tem agora cerca de 12 μ s para tomar a sua decisão o que torna o seu projeto eletrônico viável. Quando o sistema de trigger toma uma decisão este envia uma mensagem via interface paralela para o SVXII que informa qual das amostras o trigger quer que seja digitalizada. De posse desta informação o SVXII transfere a carga do capacitor correspondente para o estágio comparador. A conversão A/D baseia-se na comparação entre uma rampa que varia de próximo a zero volts até VCC e o sinal amostrado. Um contador de 8 bits incrementado à medida que a rampa aumenta serve para criar o código binário correspondente a tensão.

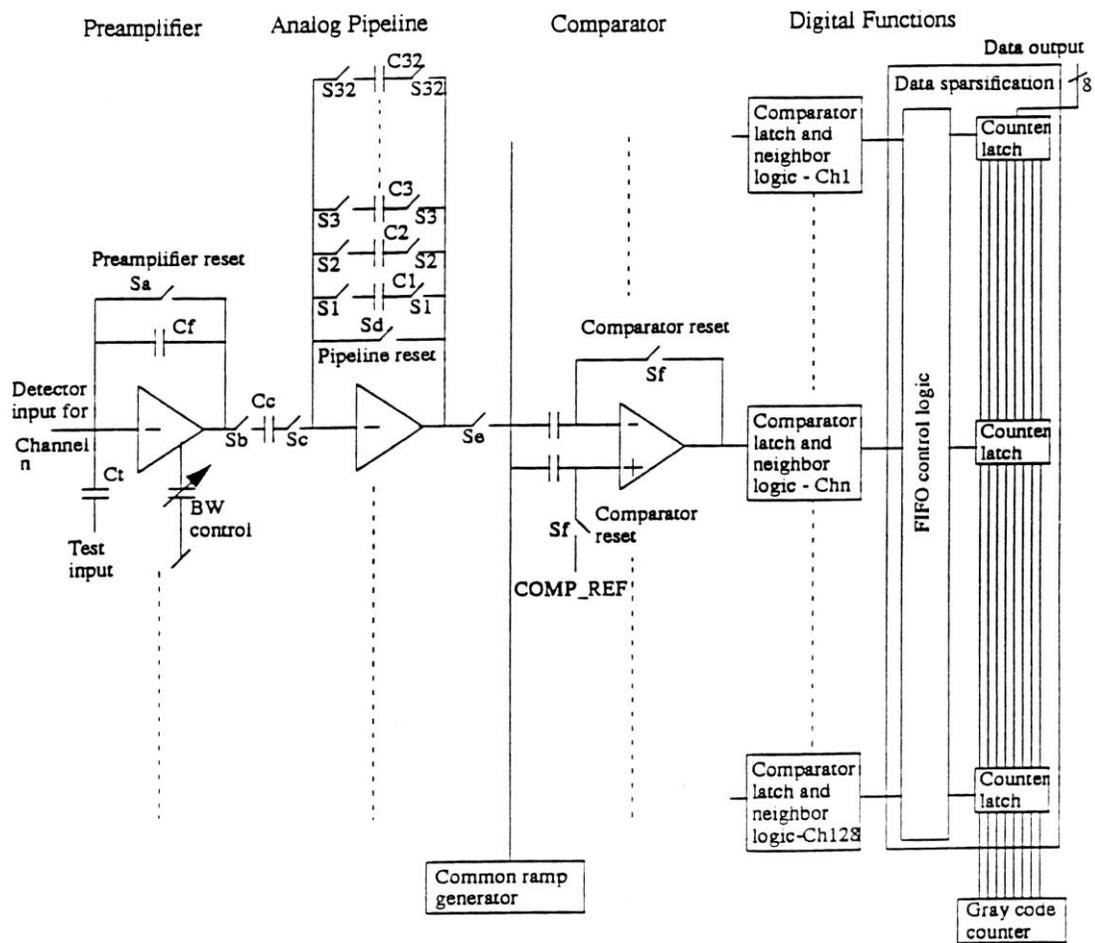


Figura 2.2.4 – Diagrama em blocos do SVXII

Quando o contador atinge um valor igual à voltagem amostrada num dos capacitores, o comparador correspondente muda de estado e armazena a contagem correspondente num registro.

O último estágio consiste numa lógica que, caso existam registros que tenham valores abaixo de um certo limiar (que é programável), estes são resetados para zero e todas as outras células com valores válidos são lidas sequencialmente.

O Chip tem várias outras funções interessantes tais como “self-test” porem estas funções não são relevantes para o funcionamento do trigger nível 1 e caso o leitor necessite de mais informação a referencia (10) provê uma descrição detalhada do C.I.

Como discutido anteriormente, a cada 396 ns ocorre um cruzamento de prótons/anti-prótons no anel. Porém com os níveis de luminosidade atuais uma colisão inelástica ocorrerá apenas em cerca de $\frac{3}{4}$ das colisões. Entretanto as colisões de interesse que geram as novas partículas ocorrem numa frequência muito menor. Uma vez que ,como já dito não é viável gravar e processar os dados de todas as colisões, é necessário criar um processo de triagem que nos separe o pequeno numero de colisões interessantes para que elas possam ser armazenadas em disco. Este processo é chamado “triggering”.

O esquema de triggering proposto pelo Dzero e no qual esta tese se baseia está mostrada na figura 2.2.5.

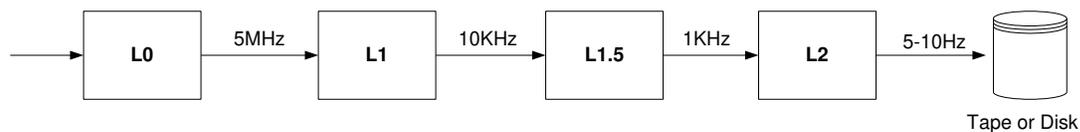


Figura 2.2.5 – Triggering proposto para o Dzero

O Nível L0 baseia suas decisões a partir da leitura do sensor de fibras cintilantes. Uma vez que este nível de trigger tem de tomar a decisão com o mínimo de latência, ele tem que se basear no sensor mais rápido disponível no detector, que no caso do Dzero é o sensor de fibras. Outra alternativa seria o sensor central de silício, mas o elevado número de canais que fazem parte deste sensor fariam que a decisão inevitavelmente fosse mais lenta.

O nível L1 baseia suas decisões na combinação do CT, do sensor de pré-shower, do sensor central de silício e do sensor do calorímetro.

O nível L1.5 combina a saída do trigger L1 com o sensor de Muons.

Finalmente o trigger L2 é implementado em software, rodando em um cluster distribuído de várias centenas de plataformas Intel de alto desempenho (servers). O nível L2 essencialmente usa todas as informações disponíveis em todos os sensores para fazer uma rápida estimativa da trajetória das partículas geradas na colisão. Esta reconstrução é computacionalmente intensa e, portanto requer um tempo considerável (cerca de 100ms – 200ms). Se após a reconstrução feita pelo L2 o evento é considerado interessante ele é salvo em fita magnética, memória flash ou disco para análise off-line posterior.

Esta análise off-line envolve a reconstrução detalhada dos eventos através do cruzamento dos dados colhidos por todos os sensores. No Dzero, esta análise requer meses rodando num Supercomputador especialmente projetado para isto e que consiste de 128 CPU's de alto desempenho operando em paralelo e usando Linux.

2.3- Uma visão detalhada do sensor de Fibras Cintilantes.

As fibras óticas de poliestireno usadas no CFT têm 835 μm de diâmetro e recebem uma deposição de alumínio na extremidade não conectada ao VLPC de forma que o numero de fótons detectados seja aproximadamente dobrado. No entanto esta fibra tem uma alta atenuação por metro de tal forma que a cada 10 metros, 95% do sinal se perde dentro da fibra. Por razões de espaço e confiabilidade o VLPC é montado exteriormente ao detector (na plataforma externa) e o sinal da fibra tem que ser conduzido por todo este percurso até o VLPC. Para reduzir esta atenuação parte da transmissão do sinal luminoso é feita por uma fibra não dopada conforme mostrado na figura 2.3.1. Como as fibras cintilantes são capazes de responder em alguns nano segundos, o CFT é a escolha natural para fornecer os dados para o trigger nível 1.

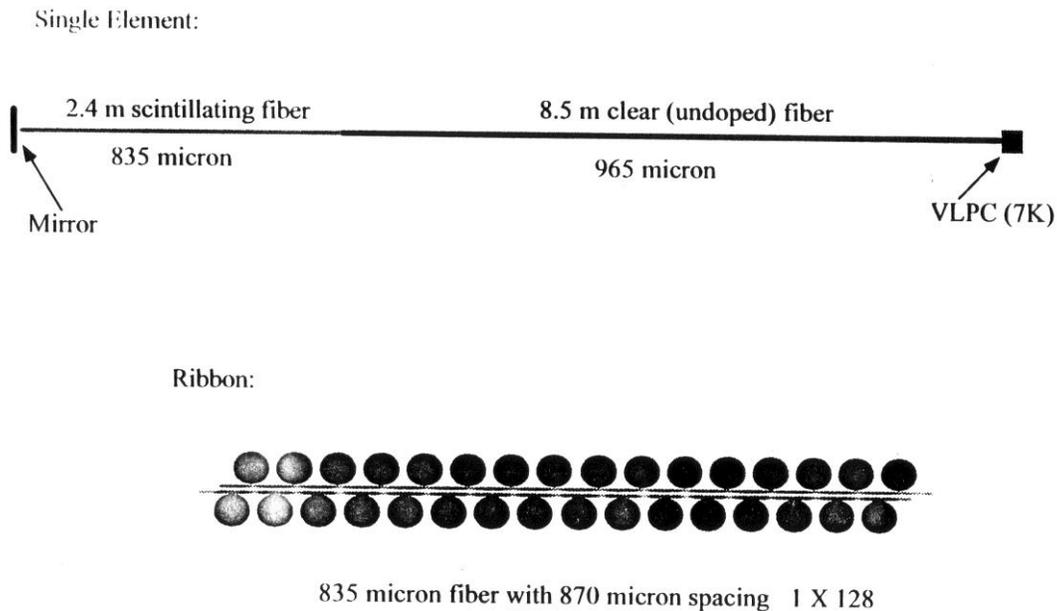


Figura 2.3.1 – Fibras cintilantes

O CFT é montado sobre quatro cilindros de fibra carbono cada um deles envolto em quatro camadas de fibras óticas cintilantes conforme mostrado na figura 2.3.2. Os raios destes cilindros são 20 cm, 30 cm, 40 cm e 50 cm respectivamente. Cada conjunto de 4 camadas de fibras envolvendo um dado cilindro é chamado de uma superlayer (SL). Cada camada na verdade consiste de duas subcamadas (chamadas doublets) com as fibras

vizinhas se sobrepondo de metade do seu diâmetro conforme mostrado na figura 2.3.1. Este esquema de superposição aumenta a precisão ao eliminar pontos cegos entre as fibras. Das quatro camadas SL por cilindro duas camadas são paralelas ao eixo do cilindro e são as camadas usadas pelo sistema de trigger (axiais) e duas camadas são enroladas ao redor do cilindro formando um ângulo de cerca de +3 graus (stereo +) e -3 graus (stereo -) com o eixo de forma a fornecer informação de profundidade ao longo do eixo do cilindro conforme mostrado na figura 2.3.2.

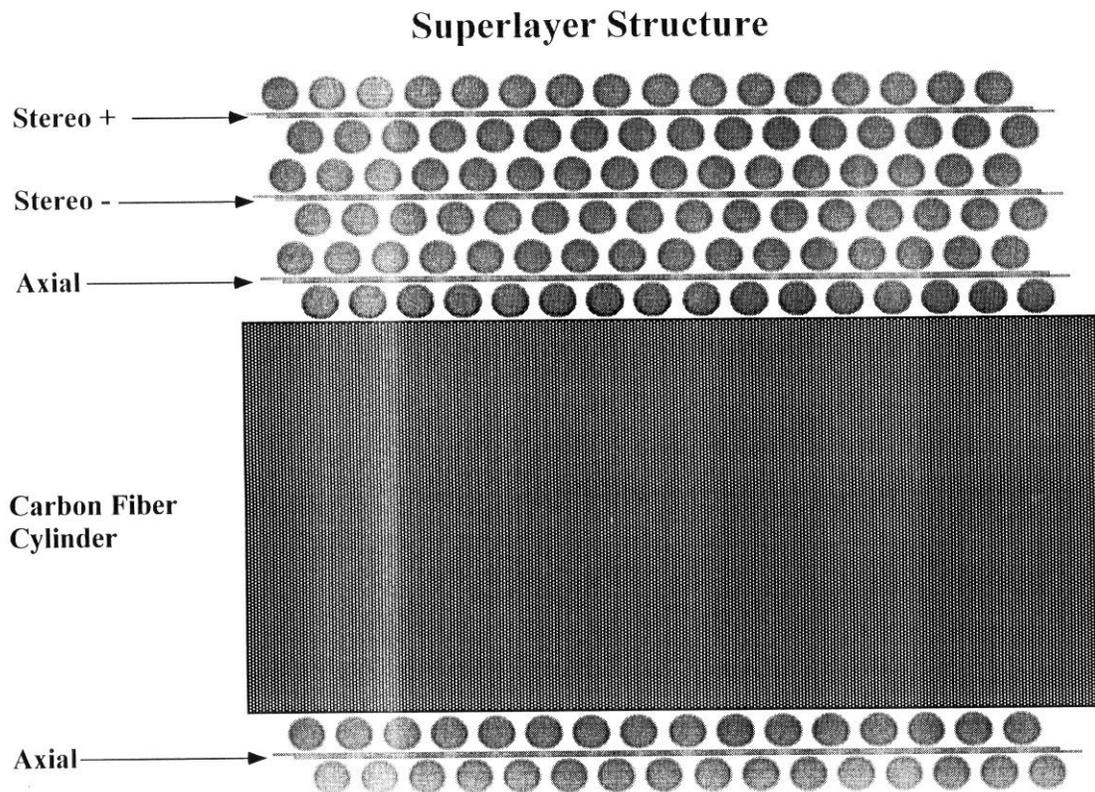


Figura 2.3.2 – Visão em corte de uma Superlayer

Se imaginarmos uma seção perpendicular ao eixo deste cilindro e dividirmos o anel resultante em 80 partes obteremos pequenos segmentos de arco de 4.5 graus que chamaremos de setores. As fibras associadas a cada setor (incluindo duas das quatro camadas SL usadas no trigger nos dá um total de 512 fibras) são processadas por uma mesma placa. Todas as 80 placas falam entre si através de um barramento de alta velocidade. Conforme mostrado na figura 2.3.3 só duas camadas de cada superlayer (as camadas axiais) são usadas no sistema de trigger. Isto é feito no interesse da velocidade, pois quanto mais canais o trigger L1 processar maior será a latência do trigger.

O sistema de trigger está baseado numa unidade básica formada por duas fibras, uma em cada camada de um dado doublet. Estas fibras estão mostradas coloridas na figura 2.3.3 no canto direito. Chamaremos esta unidade básica de **fibra doublet**.

Conforme implementado atualmente (presente estado da arte) o algoritmo de trigger é bastante simples: Para cada par formado por um par de fibras doublet na camada 1 (mais interna) e na camada 8 (mais externa) temos definido um intervalo possível de momento

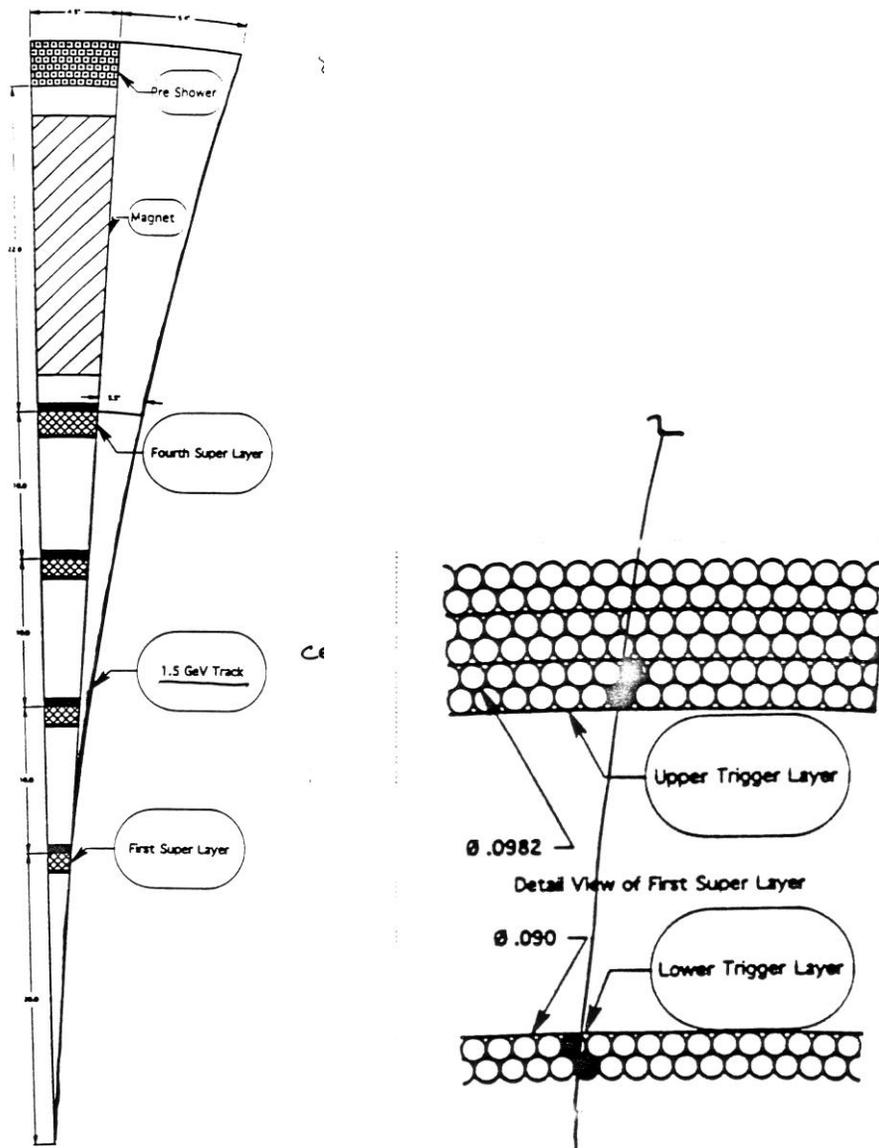


Figura 2.3.3 – Um setor do CFT

da partícula. A seguir formamos uma equação booleana incluindo todas as fibras doublet que uma partícula nessa faixa de momentos possíveis iria cruzar. Todos os elementos numa dada camada são combinados usando-se um AND lógico e todas os possíveis cruzamentos em todas as 8 camadas são também combinadas com um AND lógico. A informação de quais fibras são atingidas em cada faixa de momento de uma partícula é obtida a partir de simulações do detector usando o software GEANT em conjunto com uma descrição detalhada da geometria do CFT. Este processo cria um numero considerável de equações como veremos adiante.

Um fator importante no projeto de um sistema de trigger é o numero de trilhas falsas detectadas para cada trilha real. O VLPC possui um ruído extremamente baixo de modo que isto não é um problema por este lado, entretanto triggers falsos também são causados

por trilhas próximas que criam um efeito “Aura” disparando fibras vizinhas. Exatamente para eliminar este efeito é que se combina os resultados do trigger nível 1 com as leituras de outros sensores de cruzar os resultados de vários detectores para eliminarmos as trilhas falsas.

Cada setor têm de se comunicar uns com os outros e a mais importante comunicação é com os setores adjacentes do CFT de forma a rastrear traços que comecem nos setores adjacentes e terminem no setor corrente ou vice-versa

(trilhas que começam no setor corrente e terminal nos setores adjacentes).

Uma vez que cada setor do CFT é associado fisicamente a uma Placa de circuito impresso, a comunicação entre setores do CFT é feita por um barramento de alta velocidade conectando os vários setores.

Por outro lado existem outros sistemas de trigger (por exemplo, o sistema de Muons) que estão localizados em áreas mais distantes e que são acessados através de um protocolo Serie de alta-velocidade usando pares de fios de cobre para reduzir o custo total.

2.4 – Esquemas de distribuição do trigger atual

Nesta seção discutiremos os esquemas propostos para o circuito eletrônico do trigger nível 0 (a partir deste ponto chamaremos o trigger L1 de trigger nível 0 conforme a terminologia mais usada) para o CFT no detector Dzero. Esta seção descreve os esquemas sugeridos quando da proposta desta tese. A razão disto é que a nossa proposta parte destas estruturas e, portanto, é interessante apresentarmos esta parte primeiro como introdução.

Como já visto o detector consiste de 4 cilindros de Fibra acrílica (com raios de 20 cm, 30cm, 40cm e 50cm respectivamente) com cerca de 3 metros de comprimento. Cada cilindro tem 4 camadas de fibras dispostas sobre eles. Cada conjunto de 4 camadas é chamado de superlayer e cada uma das quatro camadas na verdade é composto de duas camadas de fibras desalinhadas entre si (doublets). Das quatro camadas duas são perfeitamente alinhadas com o eixo (chamadas de camadas X) e das outras duas uma forma um ângulo de +3% (camada U) e a outra camada forma -3% (camada V). Para fins de trigger somente as camadas X são usadas nos cálculos do trigger uma vez que as camadas U e V cruzam entre um numero muito grande de fibras ao longo dos 3 metros do detector. Isto implica que para calcularmos a posição onde uma dada fibra foi atingida necessitaríamos lermos um numero tão grande de fibras que seria inviável em termos de velocidade e custo. No entanto as camadas U e V são lidas e salvas em disco (se o evento é selecionado pelo trigger) para uma análise off-line posterior.

A circunferência completa é dividida em 80 setores com cada setor sendo servido por uma placa dedicada. As 80 placas se comunicam entre si por um barramento VME dedicado e cada placa é responsável por tomar uma decisão de trigger por setor por cada cruzamento de partículas.

As equações de quais fibras em cada caso são atingidas pelas partículas são obtidas a partir de simulações Monte-Carlo usando o simulador Geant em conjunto com arquivos descrevendo a estrutura física do detector.

A primeira proposta de formação do trigger que apresentamos é a proposta em (11), e que se baseia na formação de centroides em cada superlayer. Inicialmente vamos definir alguns termos conforme usados no artigo acima:

Fibra Atingida (F): Verdadeiro em caso da fibra ter sido atingida

Camada Doublet atingida (DL): Calculada a partir dos valores de F nas camadas internas e externas para uma dada camada Doublet.

Superlayer atingida (SL): Calculada a partir dos valores de DL nas camadas internas e externas para uma dada Superlayer.

Centroide (CT): Calculada combinando-se os valores de SL no interior de uma dada SuperLayer.

Trilha(T): Calculada combinando-se os valores de CT de todas as Superlayers.

Este esquema é baseado em uma serie de passos seqüenciais:

O primeiro passo é o cálculo de DL a partir de F (as fibras atingidas). A figura 2.4.1 ilustra todas as combinações possíveis dentro de uma camada doublet. O desalinhamento entre as camadas externa e interna age de forma a elevar a precisão da detecção da trilha. A possibilidade 1' na verdade é a mesma de 1 apenas calculadas usando-se as próximas 2 fibras. DL nos mostra por onde a partícula passou em uma dada camada doublet com precisão de meia fibra.

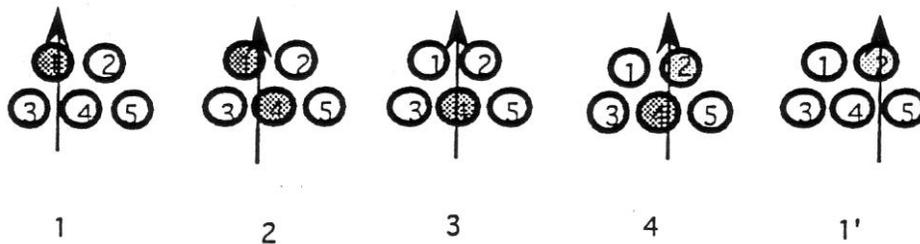


Figura 2.4.1 – Cálculo de DL

Assim para o caso 1 temos: $DL1 = F1 \text{ AND } (\text{NOT}(F3) \text{ OR } \text{NOT}(F4))$.

Usando o mesmo método temos para os casos 2, 3 e 4:

$DL2 = F1 \text{ AND } F4$

$DL3 = F4 \text{ AND } (\text{NOT}(F1) \text{ OR } \text{NOT}(F2))$

$DL4 = F4 \text{ AND } F2$

E em 1' voltamos a equação do caso 1 de novo, só que usando F2, F4 e F5 como variáveis.

O próximo passo é calcularmos SL. Para isto combinamos os DL achados no passo anterior para as duas camadas DL que formam uma superlayer conforme mostra a figura 2.4.2. As duas camadas de uma superlayer se localizam sobre um mesmo cilindro de fibra. Uma das camadas DL esta interna ao cilindro enquanto a outra camada DL esta situada no exterior.

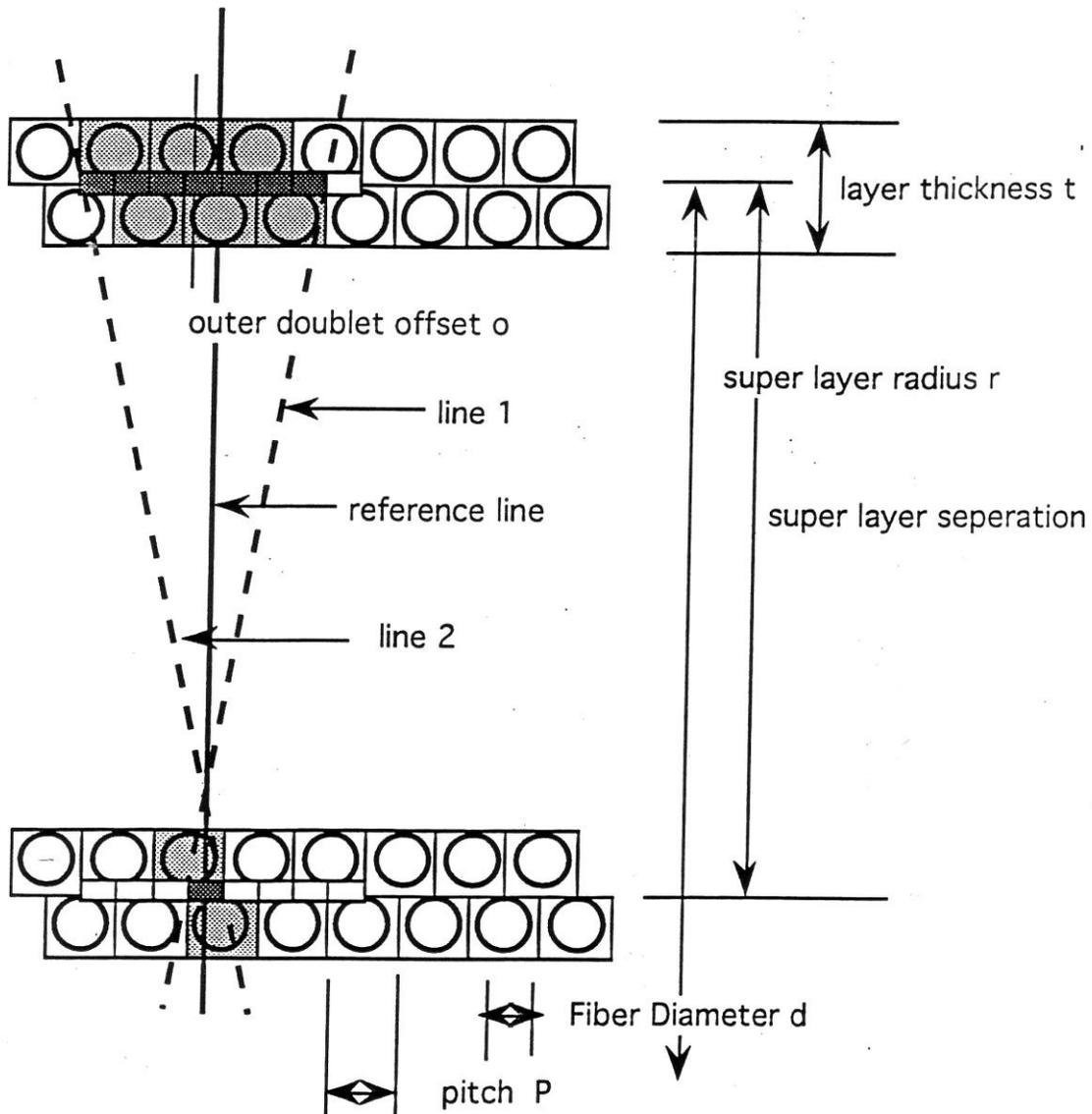


Figura 2.4.2- Formando SL's

No artigo (11) é mostrado que podemos criar equações do tipo:

$$SL_i = DL_i^{\text{inner}} \text{ AND } (DL_i^{\text{outer}} \text{ OR } DL_{i+1}^{\text{outer}} \text{ OR } \dots)$$

Onde é indicado que combinamos para cada fibra interna um conjunto de fibras externas por onde a partícula estatisticamente pode passar.

Devido ao fato de que temos muitas trilhas fantasmas causadas por ruído (efeito Aura) e também por partículas de momento muito baixo (de fato tão baixo que a sua trajetória são círculos no interior do detector), temos que usar de bom senso ao escolhermos o número de fibras externas que usamos no termo OR da equação acima. Pois se usarmos muitas esperando elevar a eficiência do detector podemos na verdade detectar um número alto de trilhas fantasmas. A Figura 2.4.3 mostra um vista perpendicular ao eixo do detector de uma colisão típica onde podemos ver a densidade de trajetórias (sobretudo nas camadas mais internas) assim como as partículas de muito baixo momento as quais essencialmente ficam girando em círculos no interior do detector criando ruído nas fibras.

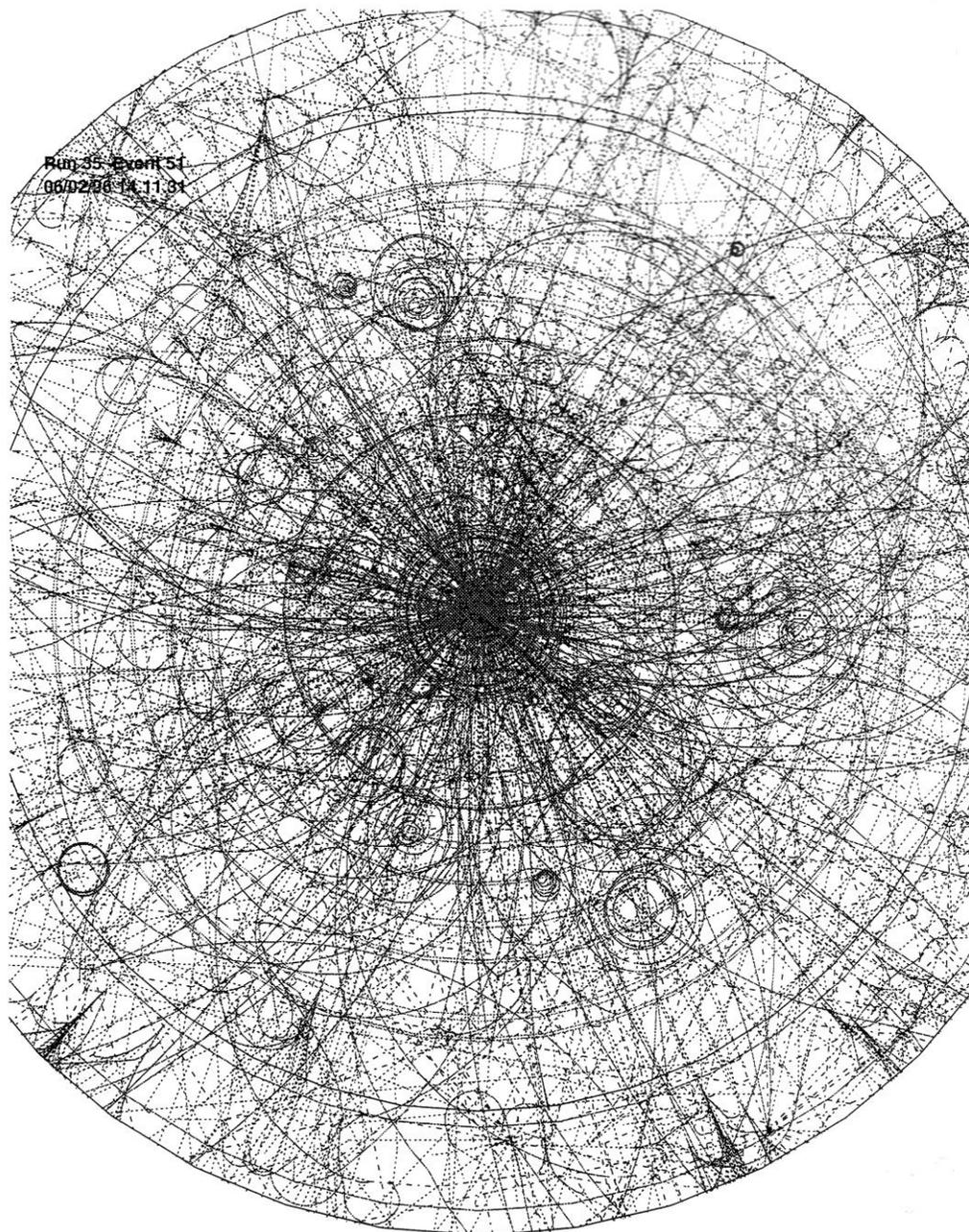


Figura 2.4.3 – Uma colisão típica

O artigo (11) faz uma análise aprofundada do número ótimo de fibras que devem ser usadas no OR. A conclusão que o autor chega é que o número ótimo é de duas fibras da camada externa.

O próximo passo é a formação de Centroides a partir dos SL. Os centroides são formados pelo grupamento de vários SL. Como os raios das quatro camadas estão na proporção 2:3:4:5 e como a geometria do detector é projetiva, ou seja, todos os setores têm o mesmo

ângulo e o mesmo número de fibras, os centroides são formados agrupando-se dois SL adjacentes.

O ultimo passo e a formação das trilhas para os vários momentos da partícula (Pt). Isto é feito pela combinação de vários centroides das varias camadas em uma equação do tipo:

$$P_j = CT_j^A \text{ AND } [CT_j^B \text{ OR } CT_{j+1}^B \text{ OR } \dots \text{ OR } CT_{j+m}^B] \\ \text{ AND } [CT_j^C \text{ OR } CT_{j+1}^C \text{ OR } \dots \text{ OR } CT_{j+n}^C] \\ \text{ AND } [CT_j^D \text{ OR } CT_{j+1}^D \text{ OR } \dots \text{ OR } CT_{j+q}^D]$$

Onde P_j representa um faixa de momento de partículas, A, B, C e D são as quatro Superlayers, J varia de 1 a 16 e m, n e q representam os offsets Máximos dos CT para cada Superlayer. Para o caso de uma partícula de momento mínimo 1.5 GeV (o valor especificado pelos físicos), o valor de q é 12 e m, n tem valores ligeiramente menores que 12 dependendo dos resultados da simulação no GEANT.

A partir dos resultados do GEANT, as equações são montadas e carregadas nas várias FPGAs que implementam cada um dos estágios. A arquitetura proposta para esta arquitetura é mostrada na figura 2.4.4 onde três camadas de FPGAs são usadas para calcular DL, SL e CT. Esta lógica na figura abaixo é chamada de “superlayer – Centroide logic”. As linhas chamadas de overlap trazem os sinais de algumas fibras dos setores adjacentes para serem processados.

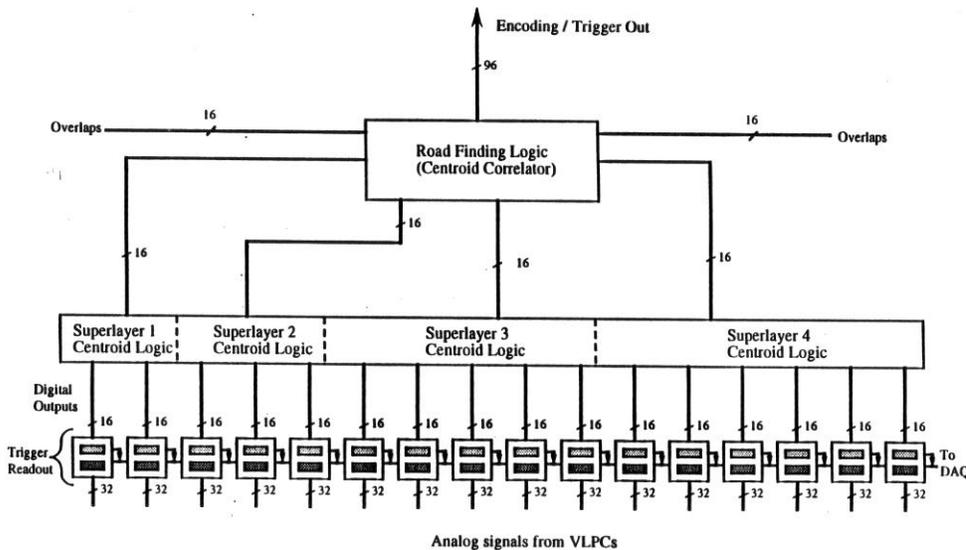


Figura 2.4.4 – Arquitetura proposta para a versão 1

Esta solução original não foi implementada por dois motivos:

- O uso de centroides limita a flexibilidade do sistema de trigger para ser reprogramado de acordo com as necessidades dos físicos
- A informação dos centroides dos setores adjacentes devem estar disponíveis antes do cálculo das trajetórias em um dado setor serem calculadas. Como há atrasos da

ordem de 20 ns - 120 ns para o cálculo dos centroides de um dado setor. Temos portanto um problema de timing.

Porém, os conceitos básicos foram aproveitados e refinados em propostas posteriores. Um dos conceitos importantes introduzidos por este primeiro estudo conforme visto é o de trilhas fantasmas e seus efeitos no trigger. Outros efeitos estudados em testes experimentais foram o efeito do elevado nível de radiação nas fibras e o efeito de canais estatisticamente danificados ao longo da vida útil do detector (que devido ao elevado nível de radiação envolvido não é muito longa). Uma análise detalhada do ponto de vista dos Físicos é apresentada em (11) porém para fins de projetar a eletrônica do trigger, podemos tomar as seguintes providências para minimizar os seus efeitos:

Primeiramente as camadas mais internas do detector são as que sofrem mais os efeitos da degeneração das fibras devido à radiação (pois são as mais próximas do ponto de colisão) e são, portanto as primeiras a falhar. No projeto do detector podemos assim fazer um OR das camadas mais internas de forma a não necessitar de um evento nas duas camadas mais internas simultaneamente, mas apenas em uma delas. Isto é claro envolve uma degradação da eficiência do detector em troca de uma maior robustez.

Com relação aos canais mortos seja por defeitos na eletrônica ou fibra, não há muito a fazer a não ser usar a redundância natural dos doublet e adicionar mais termos ao OR das fibras externas conforme descrito anteriormente. De novo, esta solução reduz a eficiência do detector e faz o detector ser mais sensível a trilhas fantasmas.

A segunda proposta conforme apresentada em (12) elimina o conceito de Superlayers, DL, SL e centroides. As fibras são consideradas individualmente no cálculo do trigger e também propõe uma nova distribuição de fibras por camada que foi usada também neste trabalho de tese. Conforme mostra a figura 2.4.5 na versão original de (11) (na figura esta versão é chamada CFL) as camadas usadas para trigger eram montadas diretamente sobre os cilindros de fibra, uma externamente outra internamente. Já na versão proposta por (12) e chamada EFL3 as camadas axiais usadas no trigger são montadas sobre as camadas U e V, maximizando assim a distância entre elas e, portanto a detecção da curvatura da partícula.

Além disso (12) propõe que usemos a camada D (mais externa) como camada âncora para o trigger. A camada âncora é definida como a camada referencia a partir do qual o trigger começa a “procurar” a trajetória da partícula. No caso proposto por (12) para cada fibra atingida na camada D existe uma equação para cada faixa de momento P_t da partícula que indicam quais as fibras nas camadas A,B e C que devem ser atingidas. Porém estas equações devem incluir não só as fibras do setor atual como também fibras adjacentes. A figura 2.4.6 mostra o resultado de simulações GEANT para uma partícula de 3 GeV, onde se mostra a trajetória de uma partícula passando pela última fibra da camada A (logo consideramos a camada A como a camada âncora neste caso) assim como os números de fibras (e ângulos em radianos) dos setores adjacentes por onde a partícula passará.

Se usarmos simulações para determinarmos quantas fibras de deslocamento uma partícula com 1.5 GeV de energia (mínima energia estimada de uma partícula) obtemos a tabela 2.4.1.

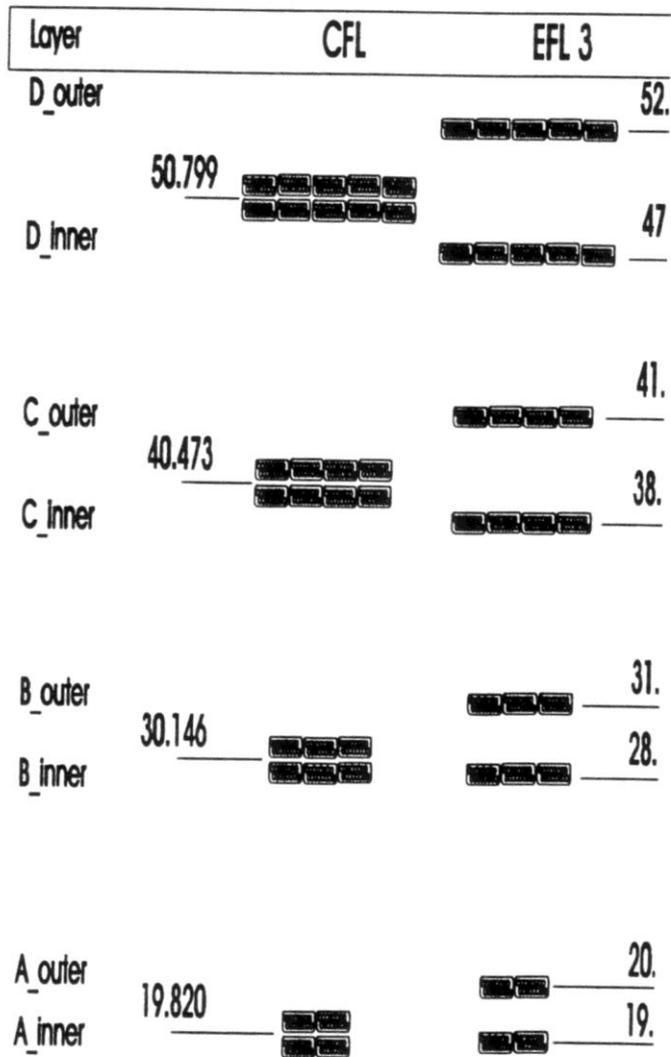


Figura 2.4.5 – Nova distribuição das fibras

Como podemos ver se ancorarmos as equações na camada A temos um número significativo de fibras de setores adjacentes entrando nas equações. Porém intuitivamente vemos que se ancorarmos na camada mais exterior o numero de fibras necessários dos setores adjacentes se reduz sensivelmente. Isto é provado matematicamente em (12) onde comparando o numero de fibras adjacentes para os dois casos temos os resultados mostrados na tabela 2.4.1.

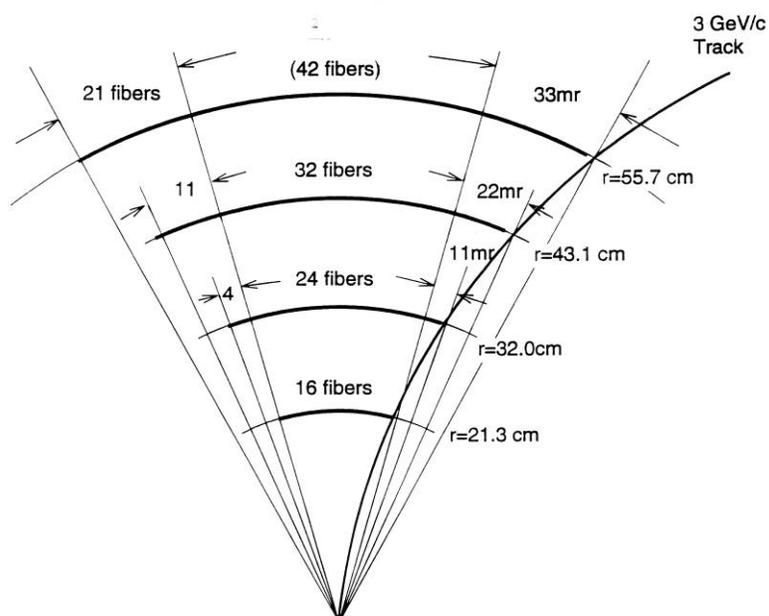


Figura 2.4.6 – trajetória de uma partícula de 3 GeV

Camada	Número de fibras das camadas adjacentes Ancora na camada A	Número de fibras das camadas adjacentes Ancora na camada D
D_externa	68	0
D_interna	60	10
C_externa	38	18
C_interna	32	24
B_externa	16	26
B_interna	12	30
A_externa	2	26
A_interna	0	28

Tabela 2.4.1- Comparação entre Âncoras na Camada A e D

Outra discussão importante abordada no artigo é como iremos fazer fluir os sinais tanto pelo barramento VME (que são limitados na placa a dois conectores de 96 pinos cada) como pelos pinos do IC do trigger (limitado pelo encapsulamento a 208 pinos).

Se considerarmos o caso das equações de trigger ancoradas na camada mais externa (camada D) que obviamente é o que apresentará o menor número de equações teremos para o máximo número de fibras que devem ser lidas para um dado setor conforme mostrado na tabela 2.4.2:

Camada	Vindas do mesmo setor	Vindas de setores adjacentes
D_externa	80	0
D_interna	80	16
C_externa	64	24
C_interna	64	24
B_externa	48	32
B_interna	48	32
A_externa	32	32
A_interna	32	32
	448	192

Tabela 2.4.2- Número de fibras necessárias por Camada

Onde o número de fibras foi arredondado para ser um múltiplo de 8 por motivos de montagem mecânica do detector devido ao numero fixo de fibras por cassete que podem ser montadas nos conectores.

O número total de fibras que devem ser usadas nas equações de trigger para o caso acima é de 832 fibras por setor. Se usarmos a estimativa do autor e considerarmos que necessitaremos de cerca de 20 pinos para VCC e terra do IC dos 208 pinos disponíveis na FPGA, nos restam 188 pinos para I/O dos sinais. Entretanto temos:

$$Numero_fibras_total = 2 * 192 + 448 = 832 \text{ fibras}$$

Entretanto se considerarmos que usaremos quatro períodos de clock para entrarmos todos os dados das fibras e além disso usaremos 4 FPGA' s em paralelo no calculo das equações teremos então:

$$Numero_fibras_por_slot = 2 * \frac{192}{4} + \frac{448}{4 * 4} = 124 \text{ Pinos}$$

O que indica que é possível pelo uso de slots na entrada de dados, evitarmos problemas com a limitação no numero de pinos do encapsulamento. Além disso, temos que trazermos de setores adjacentes um numero reduzido de fibras via barramento VME, no nosso caso 96 fibras. Se notarmos que um dado setor recebe dados dos setores adjacentes porem também manda seus dados para os setores adjacentes temos que o numero total de fibras que um dado setor tem de enviar é de 2*96 fibras. Que é acomodado pelos dois barramentos VME de 96 pinos da placa.

O esquema de sincronização da transferência entre os setores esta mostrado na figura 2.4.7, onde vemos que um dos conectores de 96 pinos é usado para entrada de dados do setor adjacente enquanto o outro conector é usado para a saída de dados.

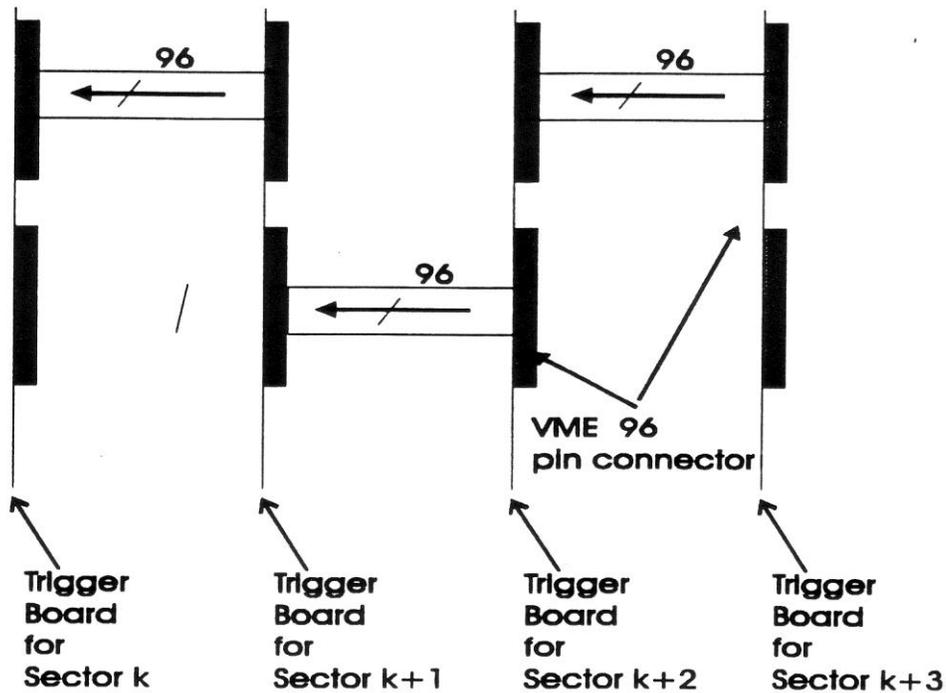


Figura 2.4.7 – Esquema de transferência de dados entre os setores

O número de slots proposto baseando-se em (13) o qual também foi usado por esta proposta é de 4 slots por cruzamento a um clock de 53 MHz. Este mesmo artigo também propõe uma sequência de leitura de fibras a qual não iremos entrar em detalhes uma vez que não a usamos nesta proposta, porém o esquema de armazenamento/timing do sistema é mostrado na figura 2.4.8. Inicialmente os sinais lógicos provenientes do CI leitor das fibras é armazenado em latches de alta velocidade (como o CI SN74ABT16373) e distribuídos para os barramentos de I/O para os setores vizinhos via VME e também apresentados para a FPGA que fará o cálculo do trigger. Os sinais estão disponíveis nos pinos tanto do barramento VME como nos pino do CI 23.8 ns após o latch armazenar os dados e fica estável por cerca de 60 ns antes de ser apagado pelos resultados do próximo slot (14). O resultado do CI leitor da fibra também deve manter o dado estável por 132 ns. Se usarmos um clock de 106 MHz (o clock mestre de 53 MHz multiplicado por dois) podemos armazenar um conjunto de dados da fibra a cada 3 períodos de clock (28.3ns). A cada 396 ns (42 ciclos de clock de 106 MHz) os latches armazenam o resultado de uma nova colisão. Isto nos dá uma folga considerável para armazenarmos em 4 slots todas as fibras antes da próxima colisão.

Uma outra especificação da lógica: A lógica deve ser capaz de processar os dados de cada slot em $(396/4) \text{ ns} = 99 \text{ ns}$ de modo a não perturbar o pipeline.

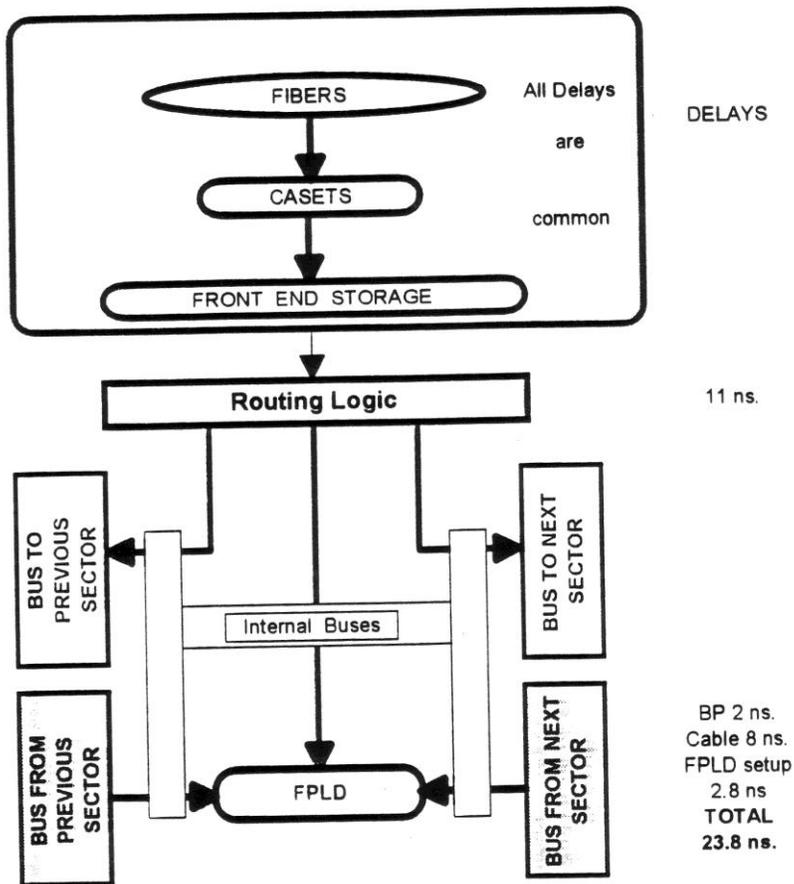


Figura 2.4.8 – Barramento interno da placa

2.5 – Metodologia usada no projeto do trigger atual

Para se criar os arquivos representando as fibras atingidas pelas partículas inicialmente os físicos rodam uma simulação Monte Carlo usando-se o software Geant para várias partículas com uma energia mínima de 1.5 GeV e obtém um arquivo do formato mostrado na figura 2.5.1 onde as primeiras 8 colunas representam o índice das fibras atingidas por uma dada trilha e a última coluna indica o número de vezes que esta sequência de fibras apareceu na simulação Monte Carlo. Para simplificar a simulação os físicos rodam estas simulações para apenas $\frac{1}{4}$ de um setor e considerando partículas com momento positivo. Esta porção de um setor é chamada de Célula e representa um padrão de fibras que são atingidas e que se repetem nas outras três células que formam o setor. Assim as equações da primeira célula são obtidas pelas simulações e as equações das outras células são obtidas da primeira célula simplesmente pela translação das fibras para as outras células. Matematicamente isto é obtido adicionando-se um offset ao índice das

fibras atingidas da célula mãe, offset este diferente para cada camada e proporcional ao numero da célula (1, 2, 3 e 4).

23	28	34	41	47	53	60	67	29
23	28	35	41	47	53	60	67	114
23	29	35	41	47	53	60	67	151
23	29	35	41	47	54	60	67	79
23	29	35	41	47	54	61	67	44
23	29	35	41	48	54	60	67	21
23	29	35	41	48	54	61	67	53
23	29	35	41	48	54	61	68	19
23	29	35	42	48	54	60	67	17
23	29	35	42	48	54	61	67	70
23	29	35	42	48	54	61	68	254

Figura 2.5.1 – Arquivo texto proveniente da simulação Monte Carlo

Os números representando os índices das fibras são representados conforme mostrado na figura 2.5.2 onde mostramos a sequência de numeração do Setor presente (Home sector) e dos setores adjacentes (previous sector e next sector). Na parte inferior da figura é mostrada a subdivisão do setor presente em quatro células básicas.

Previous Sector	Home Sector	Next Sector
1,2,3.....44	45.....88	89.....132
1,2,3.....40	41.....80	81.....120
1,2,3.....36	37.....72	73.....108
1,2,3.....32	33.....64	65.....96
1,2,3.....28	29.....56	57.....84
1,2,3.....24	25.....48	49.....72
1,2,3.....20	21.....40	41.....60
1,2,..16	17.....32	33.....48

1st cell	2nd cell	3rd cell	4th cell
45.....55	56.....66	67.....77	78.....88
41.....50	51.....60	61.....70	71.....80
37.....45	46.....54	55.....63	64.....72
33.....40	41.....48	49.....56	57.....64
29.....35	36.....42	43.....49	51.....56
25.....30	31.....36	37.....42	43.....48
21.....25	26.....30	31.....35	36.....40
17.....20	21.....24	25.....28	29.....32

Figura 2.5.2 – Numeração das fibras

Os índices das fibras indicados na figura 2.5.1 são transladados para as outras células e convertido em VHDL por uma planilha Excel dedicada com parte do código escrito em VBA. O resultado desta conversão tomando-se como exemplo o arquivo mostrado na figura 2.5.1 seria:

Track1=AL[23] AND BL[28] AND CL[34] AND DL[41] AND EL[47] AND FL[53]
AND GL[60] AND HL[67];

Track2=AL[23] AND BL[28] AND CL[35] AND DL[41] AND EL[47] AND FL[53]
AND GL[60] AND HL[67];

Track3=AL[23] AND BL[29] AND CL[35] AND DL[41] AND EL[47] AND FL[53]
AND GL[60] AND HL[67];

Track4=AL[23] AND BL[29] AND CL[35] AND DL[41] AND EL[47] AND FL[54]
AND GL[60] AND HL[67];

....

As últimas linhas do código simplesmente implementam o OR das trilhas, separando as trilhas por faixas de momento, de um mínimo de 1.5 GeV ate um momento infinito.

Para cada célula existem 306 equações do tipo acima por célula, porém como explicado anteriormente estas equações devem ser expandidas para compensar o efeito da alta dose de radiação danificando as camadas mais internas do CFT. Na pratica no projeto do trigger consideraremos o pior caso de que qualquer camada pode ser danificada pela radiação, para elevarmos a redundância. Assim digamos uma equação do tipo, digamos:

TrackX=AL[23] AND BL[28] AND CL[34] AND DL[41] AND EL[47] AND FL[53]
AND GL[60] AND HL[67];

Se considerarmos que devido a radiação uma ou mais das camadas podem ser danificadas, devemos expandir as equações de modo a incluir um OR das várias camadas conforme mostrado abaixo:

TrackX1=AL[23] AND BL[28] AND CL[34] AND DL[41] AND EL[47] AND FL[53]
AND (GL[60] OR HL[67]);

TrackX2=AL[23] AND BL[28] AND CL[34] AND DL[41] AND (EL[47] OR FL[53])
AND GL[60] AND HL[67];

TrackX3=AL[23] AND BL[28] AND (CL[34] OR DL[41]) AND EL[47] AND FL[53]
AND GL[60] AND HL[67];

TrackX4=(AL[23] OR BL[28]) AND CL[34] AND DL[41] AND EL[47] AND FL[53]
AND GL[60] AND HL[67];

Considerando esta redundância o numero de equações agora passa a ser 1224 equações por célula. Este numero indica apenas as equações das partículas com um momento positivo, as trajetórias das partículas com momento negativo podem ser obtidas simplesmente espelhando as trajetórias em relação ao eixo radial.

Assim sendo o número de equações totais a ser programada é 1224 equações * 4 células * 2 polaridades de momento, dando um total igual a 9792 equações de 8 termos.

Após a identificação de todas as trilhas de Momento maior do que 1.5 GeV, estas trilhas são classificadas em 44 faixas de momento e as 6 trilhas com os maiores valores de momento são serializadas e enviadas para o próximo nível de trigger.

Todas estas equações foram programadas em FPGAs no trigger atual usando-se 4 FPGAs, uma FPGA para cada célula. Todos os dados das fibras são entrados usando-se 4 ciclos do clock de 53 MHz e armazenados em latches. Os sinais ficam disponíveis para cálculo das trilhas após 110ns. Após cerca 270 ns os sinais de saída podem ser lidos. A serialização dos resultados conforme explicada acima é feita usando-se outras quatro FPGAs. A cada 396 ns os dados chegam provenientes do VLPC. Os dados provenientes dos setores presentes e vizinhos são armazenados nos latches após 90 ns e permanecem estáveis por cerca de 50 ns.

A figura 2.5.3 mostra a arquitetura presente do trigger assim como o atraso ao longo da cadeia de processamento do sinal.

Os VLPC em conjunto com o CI discriminador convertem os sinais óticos em pulsos elétricos e os envia após 80 ns de latência. Estes sinais são distribuídos pelo barramento VME para os setores vizinhos assim como sinais dos setores vizinhos são recebidos dos mesmos setores via VME.

As trilhas por célula são achadas 260 ns após a colisão e serializadas por máquinas de estado implementadas em outras 4 FPGAs. As 6 trilhas mais energéticas são finalmente apresentadas após 380 ns. Após isto esta informação é formatada de forma apropriada e se torna disponível para envio para o nível dois de trigger após 420 ns.

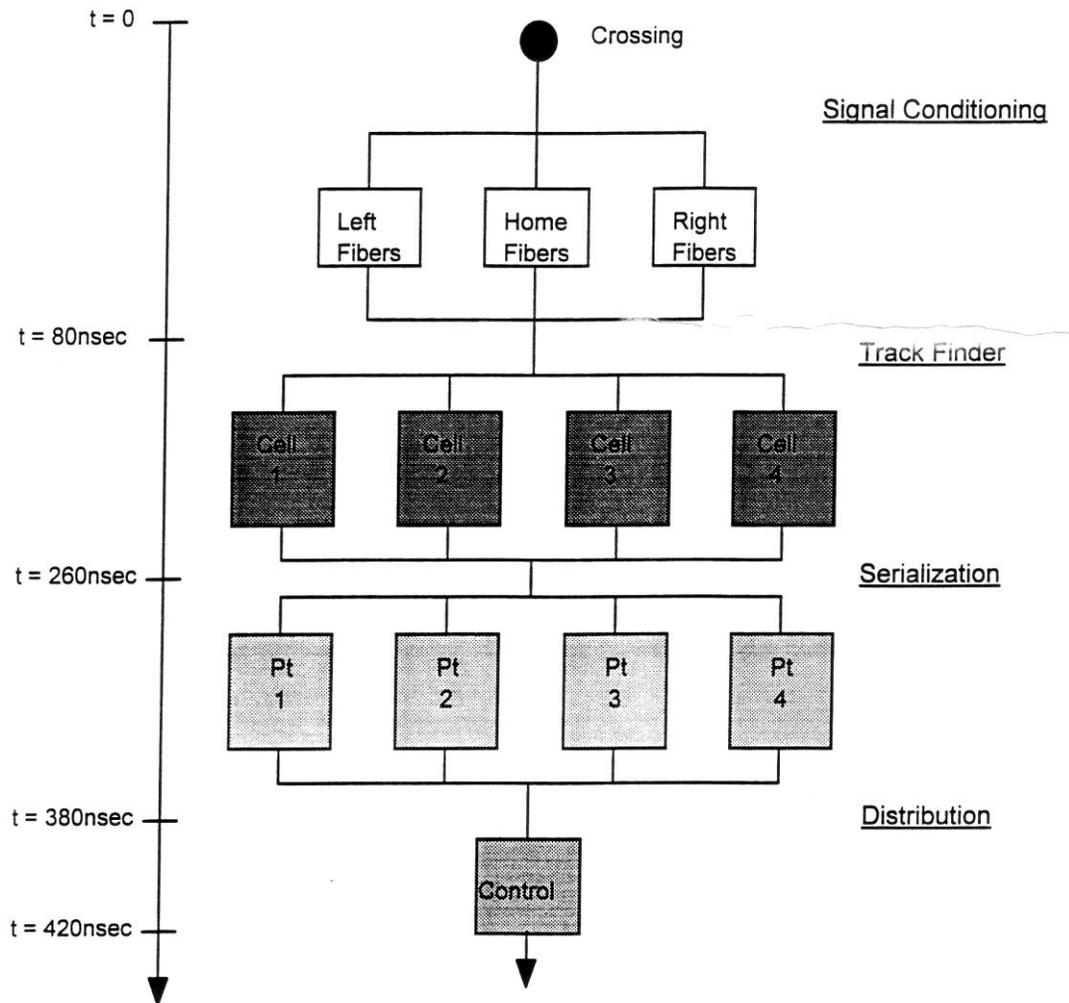


Figura 2.5.3 – Arquitetura do trigger 1 (proposta corrente)

Neste ponto podemos apreciar o ponto fraco desta proposta: em um dado momento apenas uma das 9792 possíveis equações implementadas na FPGA estará ativa, todo o resto da FPGA estará apenas dissipando energia. Isto representa um acréscimo considerável no custo do sistema e mais importante, devido à ineficiência do sistema, calor desnecessário é gerado. Calor este que necessita ser dissipado rapidamente pelo sistema de refrigeração de modo a não afetar a leitura das fibras.

Capítulo 3 - Estudo e proposta de nova arquitetura

3.1 - Estudo e simulação

Inicialmente usamos o software Mathematica (15) para escrevermos um código simples representando as equações que descrevem a deflexão causada numa partícula carregada ao passar por um campo magnético constante com o objetivo de investigarmos as propriedades das trajetórias das partículas. Para isto nos baseamos nas equações apresentadas em (16).

$$\text{Sen}(\Phi - \Phi_0) = \frac{0.3 * R}{P_t}$$

Ou se considerarmos o ângulo de deflexão bastante pequeno:

$$(\Phi - \Phi_0) = \frac{0.3 * R}{P_t}$$

Nas equações acima P_t representa o momento da partícula, Φ e Φ_0 representam o ângulo de deflexão da partícula e o ângulo de referencia. O ângulo de referencia indica o ângulo pela qual a partícula seguiria se tivesse momento infinito (que no nosso caso seria 90 graus)

No Mathematica implementamos estas equações e um modelo aproximado de localização das fibras e os índices correspondentes

A análise usando-se o Mathematica indicou que existe um passo mínimo para uma dada geometria do detector (distâncias entra as fibras) abaixo da qual o detector não detecta nenhuma variação nos índices das fibras atingidas.

A seguir escrevemos um programa em C++ que nos permite mapear os passos mínimos e as fibras atingidas supondo uma partícula com carga +1 (carga de 1 elétron porém positiva) dentro de um campo de 2 Tesla (intensidade do campo magnético gerado no interior do Dzero). A listagem deste programa esta incluída no CD anexo.

A ideia é achar um passo mínimo de momento de tal forma que os índices das fibras atingidas só variem em no máximo 3 camadas. Devido a deflexão da partícula estas 3 camadas serão as mais externas, onde a partícula certamente será mais defletida.

O programa usa uma separação entre as fibras de 238 μm (1/4 de fibra) e os raios das camadas do CFT usados são: $R[0]=0.189865\text{m}$, $R[1]=0.206535\text{ m}$, $R[2]=0.293135\text{m}$, $R[3]=0.309805\text{m}$, $R[4]=0.396405\text{m}$, $R[5]=0.413075\text{m}$, $R[6]=0.499665\text{m}$, $R[7]=0.516335\text{m}$ respectivamente.

Do resultado do programa podemos ver pela tabela 3.1 que de uma partícula de momento 1.5GeV para outra de 1.53GeV (0.03 GeV de diferença) temos apenas três camadas mudando o índice das fibras atingidas e mesmo assim a fibra atingida passa ser a imediatamente á direita ou á esquerda da fibra anteriormente atingida. Estas fibras estão assinaladas na tabela 3.1.

Camadas do CFT	Partícula Pt=1.5GeV Fibra atingida Numero	Partícula Pt=1.53GeV Fibra atingida Numero
A	3	3
B	5	5
C	7	7
D	11	11
E	12	12
F	13	13
G	16	15
H	29	28

Tabela 3.1 – Fibras atingidas conforme o modelo em C++

O resultado completo da varredura dos momentos de partículas a partir de 1.5 GeV é uma longa listagem mostrando os vários momentos nos quais as fibras atingidas variam de apenas 1 fibra em apenas 1 das camadas. Isto representa a mínima diferença de momento que o detector pode realizar. Abaixo é mostrado apenas um pedaço desta listagem. Para cada momento temos os índices das fibras atingidas em cada camada (considerando-se a fibra índice zero como aquelas pela qual a partícula passaria se tivesse energia infinita). Para cada momento indicamos com asteriscos qual camada teve o índice da fibra modificado.

Pt= 1.500236 GeV

Radius 0 phi= 0.037967 arc =0.007209 fiber No 30 hit
 Radius 1 phi= 0.041301 arc =0.008530 fiber No 35 hit
 Radius 2 phi= 0.058618 arc =0.017183 fiber No 72 hit
 Radius 3 phi= 0.061951 arc =0.019193 fiber No 80 hit
 Radius 4 phi= 0.079269 arc =0.031422 fiber No 132 hit
 Radius 5 phi= 0.082602 arc =0.034121 fiber No 143 hit
 Radius 6 phi= 0.099917 arc =0.049925 fiber No 209 hit
 Radius 7 phi= 0.103251 arc =0.053312 fiber No 223 hit ****
 #####

Pt= 1.500544 GeV

Radius 0 phi= 0.037959 arc =0.007207 fiber No 30 hit
 Radius 1 phi= 0.041292 arc =0.008528 fiber No 35 hit
 Radius 2 phi= 0.058606 arc =0.017179 fiber No 72 hit
 Radius 3 phi= 0.061939 arc =0.019189 fiber No 80 hit
 Radius 4 phi= 0.079252 arc =0.031416 fiber No 131 hit ****
 Radius 5 phi= 0.082585 arc =0.034114 fiber No 143 hit
 Radius 6 phi= 0.099897 arc =0.049915 fiber No 209 hit
 Radius 7 phi= 0.103230 arc =0.053301 fiber No 223 hit

```

#####
Pt= 1.504064 Gev
Radius 0 phi= 0.037870 arc =0.007190 fiber No 30 hit
Radius 1 phi= 0.041195 arc =0.008508 fiber No 35 hit
Radius 2 phi= 0.058469 arc =0.017139 fiber No 72 hit
Radius 3 phi= 0.061794 arc =0.019144 fiber No 80 hit
Radius 4 phi= 0.079067 arc =0.031342 fiber No 131 hit
Radius 5 phi= 0.082392 arc =0.034034 fiber No 142 hit ****
Radius 6 phi= 0.099663 arc =0.049798 fiber No 209 hit
Radius 7 phi= 0.102988 arc =0.053176 fiber No 223 hit
#####
Pt= 1.504344 Gev
Radius 0 phi= 0.037863 arc =0.007189 fiber No 30 hit
Radius 1 phi= 0.041188 arc =0.008507 fiber No 35 hit
Radius 2 phi= 0.058458 arc =0.017136 fiber No 71 hit ****
Radius 3 phi= 0.061782 arc =0.019140 fiber No 80 hit
Radius 4 phi= 0.079052 arc =0.031337 fiber No 131 hit
Radius 5 phi= 0.082376 arc =0.034028 fiber No 142 hit
Radius 6 phi= 0.099644 arc =0.049789 fiber No 209 hit
Radius 7 phi= 0.102969 arc =0.053166 fiber No 223 hit
#####
Pt= 1.505762 Gev
Radius 0 phi= 0.037828 arc =0.007182 fiber No 30 hit
Radius 1 phi= 0.041149 arc =0.008499 fiber No 35 hit
Radius 2 phi= 0.058403 arc =0.017120 fiber No 71 hit
Radius 3 phi= 0.061724 arc =0.019122 fiber No 80 hit
Radius 4 phi= 0.078978 arc =0.031307 fiber No 131 hit
Radius 5 phi= 0.082299 arc =0.033996 fiber No 142 hit
Radius 6 phi= 0.099551 arc =0.049742 fiber No 208 hit ****
Radius 7 phi= 0.102872 arc =0.053116 fiber No 223 hit
#####
Pt= 1.506964 Gev
Radius 0 phi= 0.037798 arc =0.007176 fiber No 30 hit
Radius 1 phi= 0.041116 arc =0.008492 fiber No 35 hit
Radius 2 phi= 0.058356 arc =0.017106 fiber No 71 hit
Radius 3 phi= 0.061675 arc =0.019107 fiber No 80 hit
Radius 4 phi= 0.078915 arc =0.031282 fiber No 131 hit
Radius 5 phi= 0.082233 arc =0.033968 fiber No 142 hit
Radius 6 phi= 0.099471 arc =0.049702 fiber No 208 hit
Radius 7 phi= 0.102790 arc =0.053074 fiber No 222 hit ****

```

(....)

Este programa varre todos os momentos a partir de 1.5GeV (mínimo momento especificado pelos físicos) até 360 GeV (máximo momento a partir do qual o CFT não

detecta nenhuma diferença). Do resultado fornecido por este programa nós obtemos um dado importante: Para um dado quadrante temos um máximo de 925 trilhas possíveis, cada uma com um momento diferente. A seguir este conjunto de trilhas foi importado pelo Microsoft Excel de modo a calcularmos a diferença entre o número da fibra atingida entre as camadas extremas (camada H e camada A). Parte desta tabela Excel é mostrada na Tabela 3.2:

Tabela 3.2- Parte das trilhas resultantes como função das camadas

O resultado da análise feita no Excel mostra que usando este modelo geométrico como

Layer A	Layer B	Layer C	Layer D	Layer E	Layer F	Layer G	Layer H	#H - #A
30	35	72	80	132	143	209	223	193
30	35	72	80	131	143	209	223	193
30	35	72	80	131	142	209	223	193
30	35	71	80	131	142	209	223	193
30	35	71	80	131	142	208	223	193
30	35	71	80	131	142	208	222	192
30	35	71	80	130	142	208	222	192
30	35	71	79	130	142	208	222	192
30	35	71	79	130	142	207	222	192
30	35	71	79	130	142	207	221	191
29	35	71	79	130	142	207	221	192
29	35	71	79	130	141	207	221	192
29	35	71	79	130	141	206	221	192
29	35	71	79	130	141	206	220	191
29	35	71	79	129	141	206	220	191
29	35	71	79	129	140	206	220	191
29	35	70	79	129	140	206	220	191
29	35	70	79	129	140	206	219	190
29	35	70	79	129	140	205	219	190
29	35	70	78	129	140	205	219	190
29	35	70	78	129	140	205	218	189
29	35	70	78	129	140	204	218	189
29	35	70	78	128	140	204	218	189
29	34	70	78	128	140	204	218	189
29	34	70	78	128	139	204	218	189
29	34	70	78	128	139	204	217	188
29	34	70	78	128	139	203	217	188
29	34	69	78	128	139	203	217	188
29	34	69	78	128	138	203	217	188

uma aproximação para o CFT, que para cada valor de diferença entre os números de índices entre as fibras nas camadas A e H existem no máximo 10 (dez) trajetórias diferentes que a partícula pode fazer entre estes dois pontos conforme mostrado na tabela a seguir:

Numero da fibra Da Camada A	Numero da Fibra Da Camada H	Numero de trilhas Possíveis
30	223	6
30	222	4
30	221	1
29	221	3

29	220	4
29	219	3
29	218	5
29	217	5
29	216	3
29	215	4
29	214	2
28	214	3
28	213	4
28	212	5
28	211	3
28	210	5
28	209	3
28	208	4
28	207	4
27	207	1
27	206	5
27	205	4
27	204	3
27	203	4
27	202	3
27	201	5
27	200	3
27	199	5
26	199	1
26	198	5
26	197	3
26	196	4
26	195	5
26	194	3
26	193	5
26	192	2
25	192	3
25	191	4
25	190	4
25	189	4
25	188	3
25	187	4
25	186	5
25	185	3
25	184	2
24	184	3
24	183	5
24	182	3
24	181	5
24	180	3

24	179	5
24	178	4
24	177	2
23	177	2
23	176	5
23	175	2
23	174	6
23	173	4
23	172	3
23	171	4
23	170	4
22	170	1
22	169	4
22	168	4
22	167	5
22	166	4
22	165	3
22	164	5
22	163	3
22	162	2
21	162	4
21	161	5
21	160	3
21	159	4
(...)	(...)	(...)

Tabela 3.3 – Máximo numero de trilhas por combinação (parte)

Esta análise Matemática foi feita manualmente pra todas as combinações das camadas A e H baseando-se na tabela de fibras atingidas.

Este resultado ocorre devido à finita precisão do detector causada pelo diâmetro da fibra. Esta descoberta possibilita uma significativa redução no tempo de reconstrução de trajetória e de espaço em memória para o algoritmo de busca uma vez que não temos de pesquisar todas as combinações possíveis.

3.2 - Proposta de Arquitetura

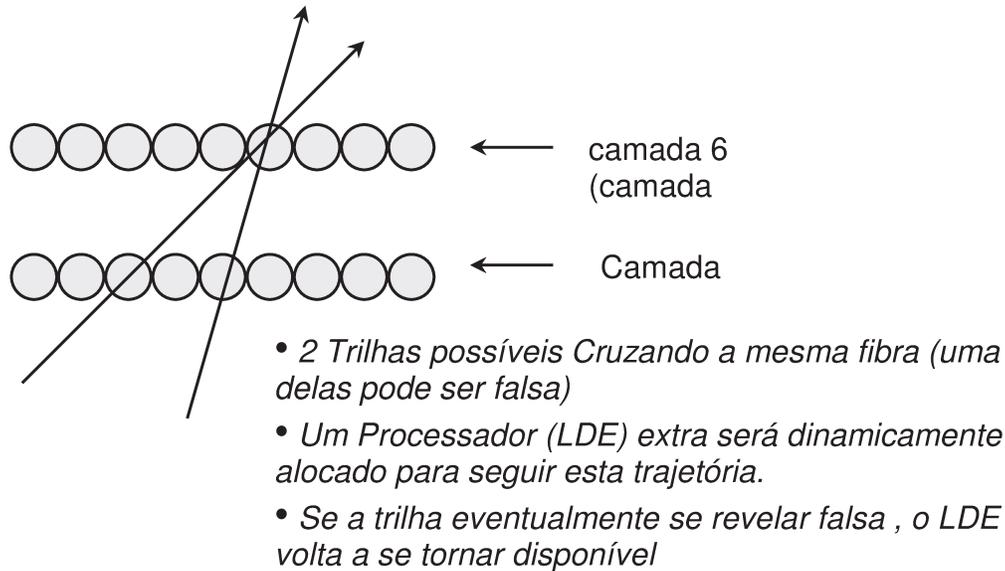
Baseando-se nesta análise a nossa primeira proposta de algoritmo foi:

1. A partir de simulações no Geant, obtemos um conjunto de fibras atingidas possíveis de acontecerem na faixa de momentos desejados pelos físicos.

2. Criar em Memória uma tabela com todas as fibras possíveis de serem atingidas por uma partícula para cada valor de diferença do número de fibra da camada H atingida e o número de fibra da camada A atingida pela partícula. Como já visto para cada valor desta diferença existem apenas 10 trajetórias diferentes possíveis
3. A partir das informações armazenadas provenientes do CFT pode-se calcular a diferença entre as fibras atingidas nas camadas H e A. O valor desta diferença limita a nossa busca como já vimos a 10 trajetórias.
4. Numa segunda etapa lemos as camadas intermediarias e comparamos com as entradas das tabelas a fim de identificarmos uma trilha de interesse.

A esta ideia básica devemos adicionar algum mecanismo para lidar com o fenômeno de trilhas falsas. Estas trilhas conforme já discutido anteriormente são causadas por partículas de baixa energia as quais essencialmente ficam girando no interior de detector causando o disparo de um grande número de fibras. A consequência disto é o de causar um excesso de fibras disparadas o que é mostrado na figura 2.4.3 (capítulo 2). Nesta figura notamos que a figura está excessivamente carregada pelo acréscimo de trilhas “fantasmas” causadas pela circulação das partículas de baixa energia.

Para lidar com este problema teremos que considerar que uma dessas trilhas “fantasma” pode disparar algumas fibras válidas entre a Camada A e H e causar uma perda de eficiência no algoritmo uma vez que isto acarreta um aumento das possibilidades de trilhas a seguir entre as camadas A e H. Este problema pode ser minimizado pela adição de processadores adicionais que tentam recompor estas trilhas extras. Os números destes processadores adicionais devem ser determinados a partir de simulações Geant e nesta proposta o número é alocado dinamicamente durante a programação do dispositivo de forma a ter o máximo de flexibilidade de uso. Este efeito é chamado de bifurcação (Branching) e é ilustrado na figura 3.2.1. O Processador dinamicamente alocado a cada passo compara as fibras atingidas com a sua lista na memória e se em alguma camada (como veremos a seguir as fibras são lidas sequencialmente de duas camadas em duas camadas) ela não coincide ele entende que é uma trilha falsa e volta a ficar disponível para traçar qualquer outra trilha extra que apareça.



Este esquema é aplicado entre a camada âncora e quaisquer das outras camadas. No caso Mostrado acima é aplicado entre a âncora e a camada cinco.

Figura 3.2.1-Bifurcacao devido a trilhas falsas

Assim podemos propor o seguinte algoritmo básico:

- Usaremos como camada Âncora a camada seis (F), pois como visto em (12) teremos o mínimo numero de equações. Cada Chip será responsável por um setor e não usaremos o conceito de centroides (Resolução de uma fibra)
- Particionamento da leitura das fibras em três passos em sequência:
 - Camadas E e F
 - Camadas G e H
 - Camadas A, B, C e D (num esquema 3-de-4 para termos redundância)
 - $(A + B) \times C \times D$
- Atribuir um processador dedicado (LDE) a cada fibra atingida na camada 6 (36 fibras por setor)
- Se a fibra foi atingida o processador é responsável por pesquisar se ha uma combinação correta de fibras atingidas nas outras camadas. Esta busca será baseada em tabelas carregadas na memória a partir de simulações GEANT.
- Esta busca por padrões de fibras atingidas será feita por etapas: primeiro usando-se as camadas E F, a seguir usando-se as Camadas G e H e por último usando-se as camadas A, B, C e D em conjunto.
- O número de possibilidades pesquisadas será cada vez menor à medida que completamos cada etapa sucessiva conforme já esperado pelos resultados da simulação em C++.

Uma descrição mais detalhada em sequencia seria:

1. Inicialmente as fibras das Camadas E e F são lidas e registradas em Latches.
2. Um LDE é associado a cada fibra atingida na camada 6. O LDE começa a busca pelas possíveis combinações de fibras das Camadas E e F obtidas da simulação GEANT em busca de um padrão similar de fibras. Se ele as acha ele se antecipa e lê quais seriam as fibras atingidas nas Camadas G e H preditas pela simulação.
 - a. Máximo de 9 possibilidades
3. As Camadas G e H são lidas e registradas.
4. Os LDEs associados novamente comparam as fibras atingidas com a que ele esperava como resultado do passo anterior. Se há um acerto ele de novo se antecipa e lê da memória as possíveis combinações de Fibras nas camadas A, B, C e D preditas pela simulação.
 - a. Máximo de 14 possibilidades
5. As Camadas A, B, C e D são lidas e registradas.
6. Baseando-se na predição anterior buscamos pelas fibras atingidas da seguinte maneira: $(A + B) \times C \times D = A \times C \times D + B \times C \times D$. A razão disto é que necessitamos de redundância devido ao potencial dano das camadas mais interiores devido à radiação.
 - a. Máximo de 20 possibilidades (2×10)

Desta maneira em três passos podemos achar todas as trilhas de interesse e os seus momentos (Pt) associados. A sequência de operações de entrada e saída é a seguinte:

Passo 1: Ler 204 fibras (E e F)

Passo 2: Ler 252 fibras (G e H)

Passo 3: Ler 264 fibras (A, B, C e D) e dar saída com as trilhas resultantes.

O número de fibras especificado acima é por setor e vem do seguinte:

No Passo 1 lemos as camadas E (108 fibras) e F (96 Fibras) e, portanto nesta etapa necessitamos de 204 pinos de entrada.

No Passo 2 lemos as camadas G (132 fibras) e H (120 Fibras) e, portanto nesta etapa necessitamos de 252 pinos de entrada.

Por fim no passo 3 lemos as camadas A, B C e D ($48 + 60 + 72 + 84$ fibras) e necessitamos assim de 264 pinos de entrada.

Assim sugerimos dividir o barramento de entrada da seguinte maneira:

$66+66+72+84 = 288$ Bits no total, sendo que os dois barramentos de 66 nos passos 1 e 2 são combinados em um único de 122 bits e se somam ao de 72 para prover os 204 bits. De forma análoga os dois barramentos de 72 e 84 são combinados num único nos dois passos iniciais.

A figura 3.2.2 mostra a arquitetura proposta em detalhes. Esta figura será usada como referência para as explicações posteriores.

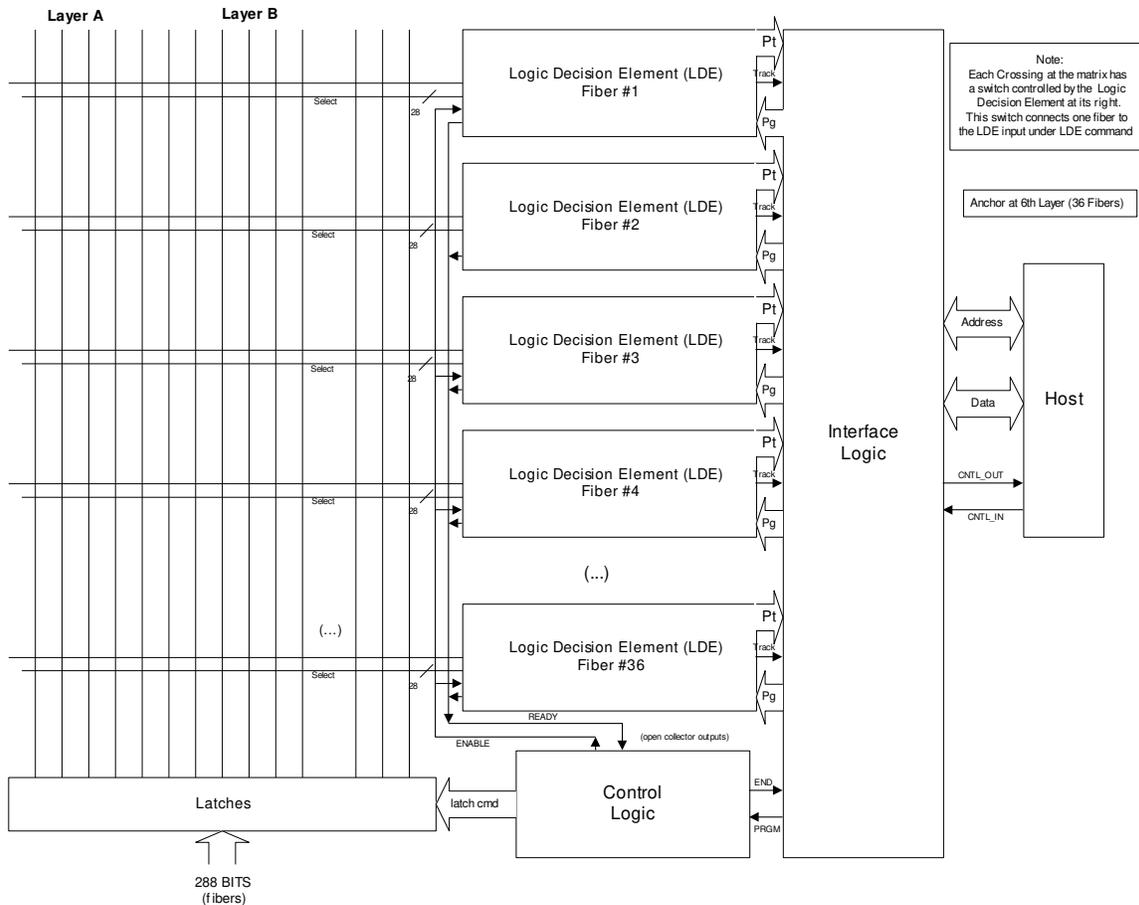


Figura 3.2.2 – Arquitetura proposta (1 chip)

A solução proposta se baseia numa arquitetura paralela formada por vários processadores simples (LDE) sendo que cada LDE é responsável pelo rastreio de uma trilha. Os LDE são gerenciados por um processador central (LDE Control) que cuida da programação de cada LDE individualmente com as equações de cada trilha assim como da leitura dos resultados da colisão e a sua formatação para envio para o host.

Como podemos ver na figura as fibras de cada setor são selecionadas por um barramento de 7 bits (total de 128 linhas de endereço). Em um dado passo de algoritmo apenas uma chave por setor será selecionada. Uma vantagem desta proposta é que o mesmo hardware é reusado sequencialmente a cada passo para avaliar as equações matemáticas economizando área e energia. As chaves são implementadas usando-se portas AND no modo corrente de modo a serem rápidas, compactas e terem um “fan-in” elevado. Estas chaves de modo corrente são montadas numa matriz de chaves de forma a controlar quais fibras serão conectadas para quais portas AND. Cada porta AND implementa um termo da equação booleana de trigger.

Como já indicamos temos um processador associado a cada fibra da camada 6 (que chamamos Logical decision Element – LDE). Cada LDE tem 20 portas AND para checar condições do tipo X.Y (dois termos) ou X.Y.Z.W (4 termos). Esta operação AND é feita através de uma soma de correntes num nó do circuito (modo corrente). O LDE tem 128

bits de saída que seleciona (usando a matriz de chaves) quais fibras serão conectadas a cada porta AND de forma a criar os termos das equações.

O bloco lógica de controle (control logic) sincroniza os 36 LDEs com o comando de “latch” de forma a armazenar a cada passo os níveis lógicos correspondentes às fibras. A saída do LDE “READY” (que é tipo coletor aberto) quando em nível alto (aberto) informa a lógica de controle que o LDE correspondente já terminou de processar as equações. Uma vez que o sinal “READY” de todos os LDEs são conectados juntos, somente quando o ultimo LDE termina os seus cálculos é que a linha “READY” vai a “1” lógico (devido ao pull-up).

Quando o sinal READY se torna “1” a lógica de controle comanda o armazenamento do próximo grupo de fibras e indica este fato para os LDEs através do sinal ENABLE = '1'.

Quando detecta o sinal ENABLE igual a '1' todos os LDEs começam a trabalhar no próximo passo em paralelo e este ciclo se repete a cada passo ate o passo 3.

Ao fim do passo 3 os LDEs que tiveram sucesso no processo de identificação de trilhas irão reportar uma trilha achada e qual o Pt (momento transversal) desta partícula para a lógica de controle (LDE Control). A lógica de controle (por intermédio do modulo IL – interface logic) formata estes dados de forma apropriada e os retransmite para o host.

A outra função do modulo IL é fazer o download das equações (tabelas) para a memória de cada um dos LDEs quando requisitado pela lógica de controle (LDE control) a qual por sua vez obedece a uma requisição do processador host.

A figura 3.2.3 mostra a estrutura básica do LDE.

O LDE usa comparadores de modo corrente para implementar os termos AND de uma dada equação (porta AND). Se a condição AND ocorre a saída correspondente vai a nível lógico “1”. A cada passo do algoritmo apenas uma das saídas vai a nível “1” indicando qual das possíveis trajetórias ocorreu.

Como indicado nas figuras, chaves na entrada selecionam quais fibras são combinadas em uma dada porta AND a cada passo do algoritmo.

A tabela look-up (RAM estática), dependendo de qual linha vai a “1” vai decidir quais chaves serão fechadas no próximo passo. Esta decisão é baseada nos dados de simulação carregados de forma apropriada na tabela.

A seguir as saídas da tabela look-up são armazenados em latches após um atraso enquanto os dois bits C2, C1 (que indicam em qual dos passos o LDE está) mudam para indicar passo 2. Após isto a lógica de Controle interna do LDE eleva o sinal READY para “1” indicando que o LDE esta pronto para o próximo passo.

Quando o LDE recebe o sinal ENABLE = 1 da Lógica de controle principal (indicando que um novo passo se inicia) a lógica de controle local espera por um certo tempo e após este atraso envia um comando de latch para armazenar as saídas da tabela look-up (posições das chaves para o próximo passo – passo 3). Os bits C1 e C2 mudam para indicar passo 3 e o pino READY vai a “1” de novo para indicar LDE pronto para o passo 3.

Durante o passo 3 o sinal ENABLE vai a “1”. De novo a lógica de controle local espera por um tempo de atraso e após isto ocorre a saída das trilhas que o LDE (se por acaso existir) achou e o Pt (momento transversal) correspondente.

Após isto a lógica de interface formata e envia os resultados para o host de acordo com o protocolo apropriado ao host.

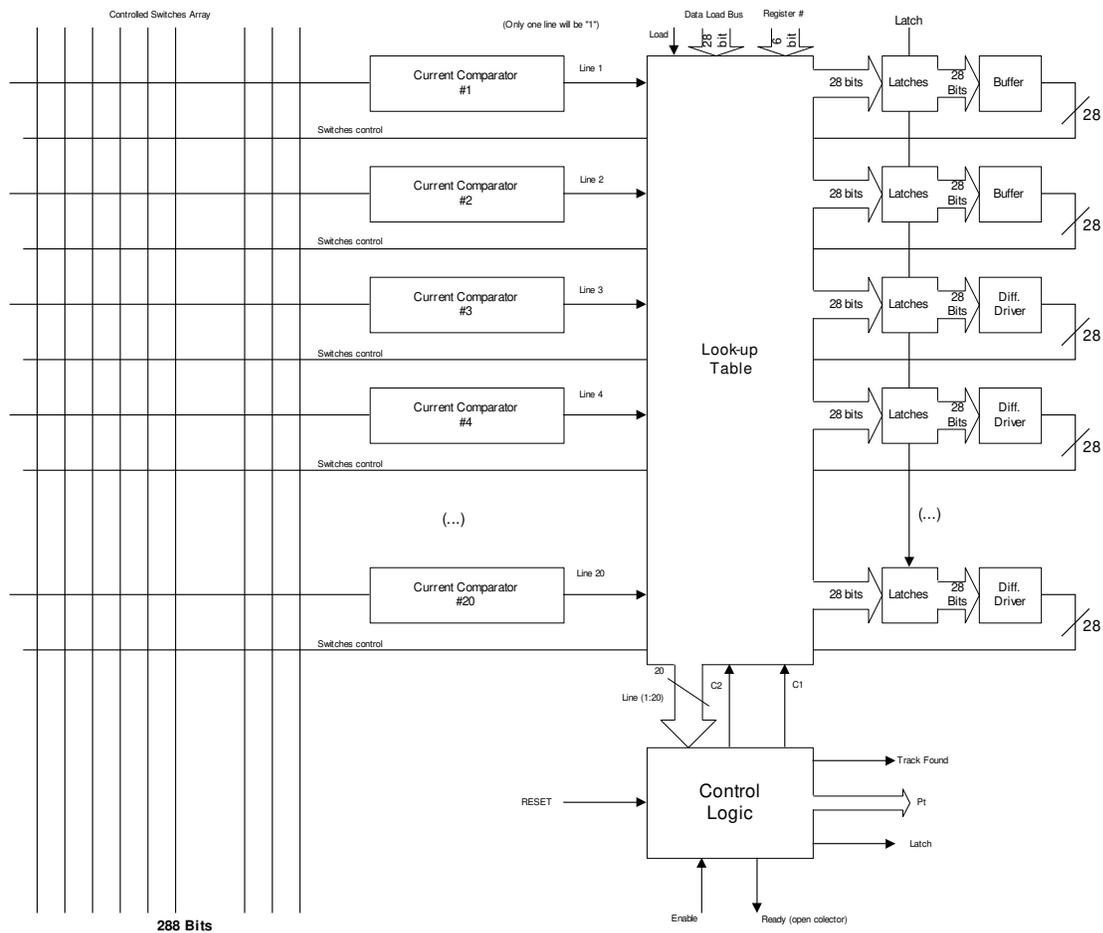


Figura 3.2.3 – Arquitetura do LDE

O comparador de modo corrente que implementa a porta AND está mostrado na figura 3.2.4. Esta comparação é implementada através de soma e subtrações de corrente num nó do circuito. Por exemplo, digamos que apenas um espelho de corrente injete corrente no nó A, neste caso o potencial no nó A vai ser abaixo do limiar do inversor. Se por outro lado dois espelhos de corrente injetam corrente no mesmo nó, o mesmo potencial muda para um nível acima do limiar do inversor e um nível lógico alto é gerado. Desta forma implementamos de uma forma analógica uma porta AND de duas entradas.

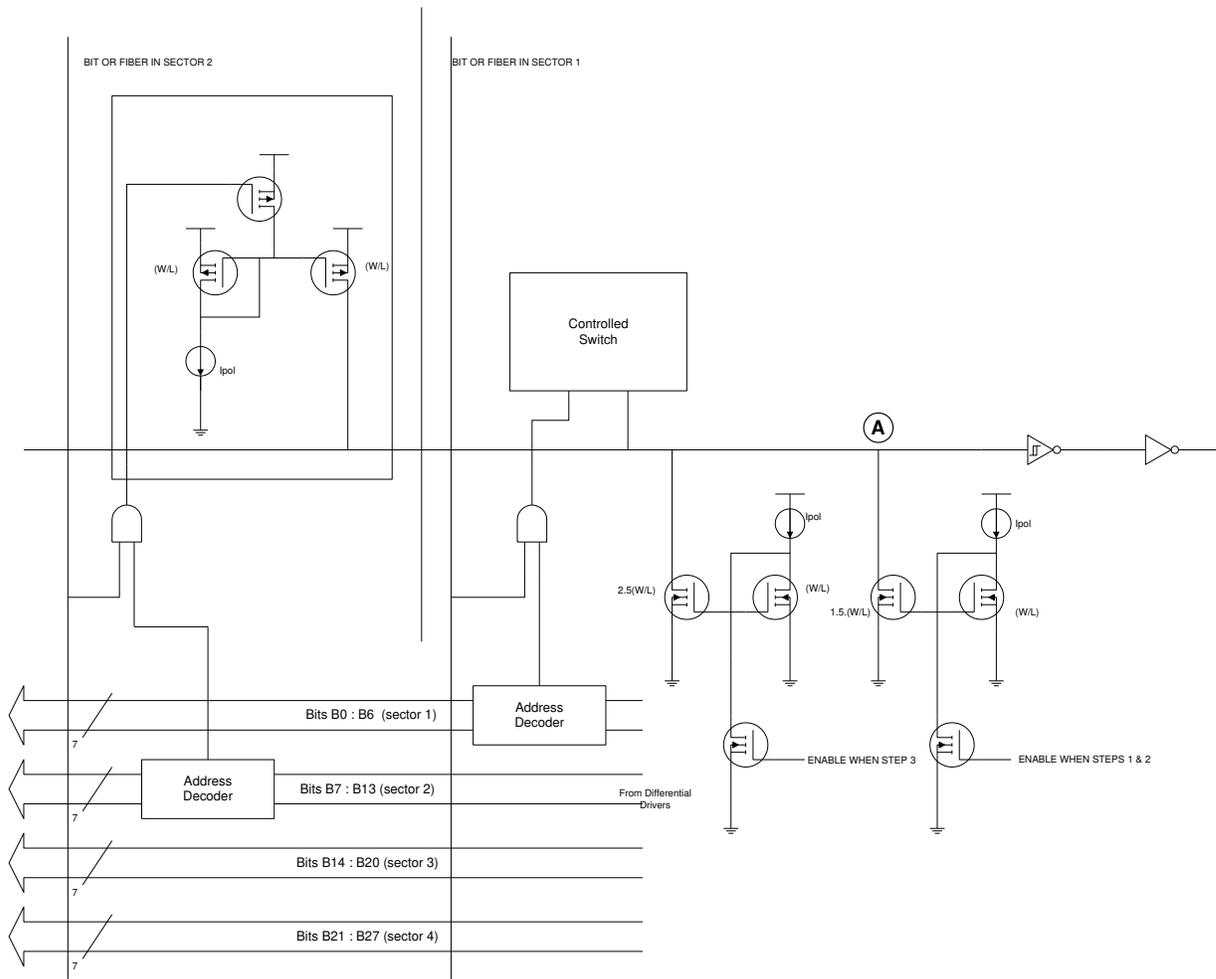


Figura 3.2.4 – Comparador de modo Corrente (porta AND)

A estrutura da tabela look-up é mostrada na figura 3.2.5. Por razões de velocidade usamos uma RAM estática nesta implementação. Esta estrutura foi implementada em VHDL conforme veremos adiante. Cada um dos três bancos de registros contém informações de trilha correspondente a cada um dos três passos do algoritmo (selecionáveis pelos bits C1 e C2). A estrutura interna de cada um dos 3 bancos de registros é mostrado na figura 3.2.6.

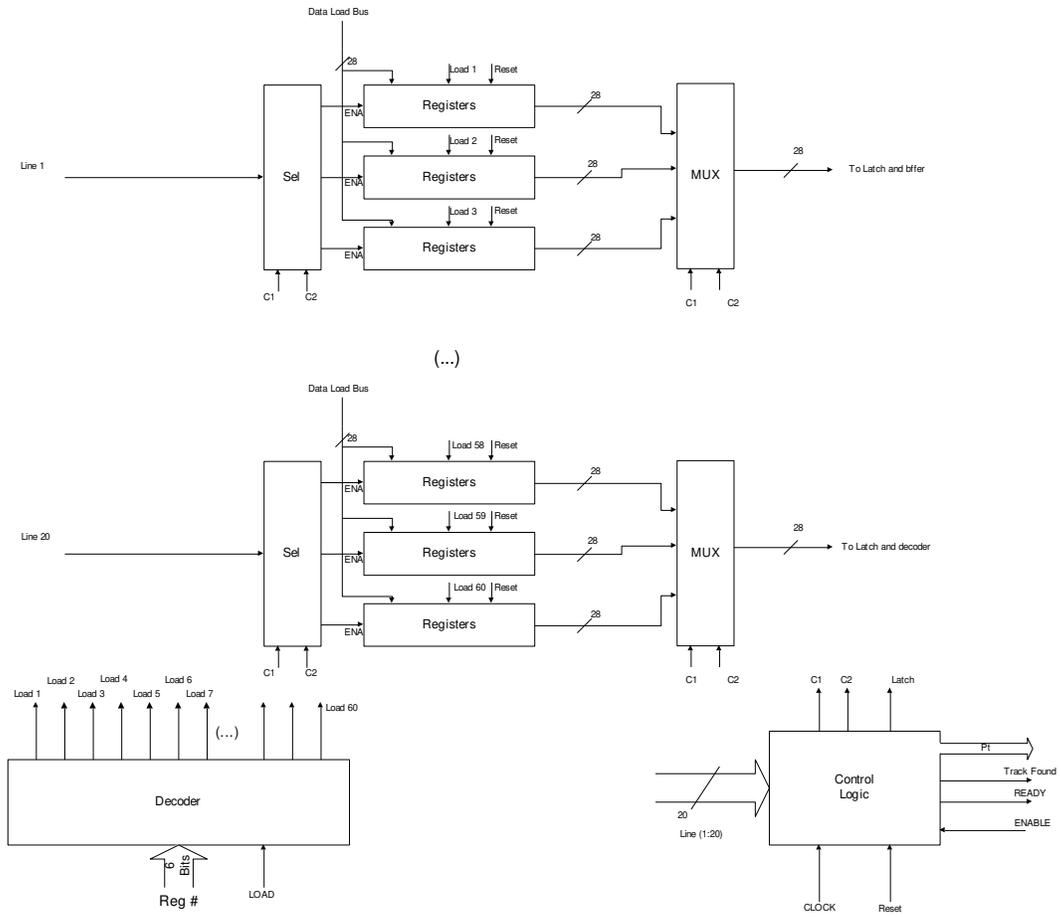


Figura 3.2.5 – Estrutura da RAM estática

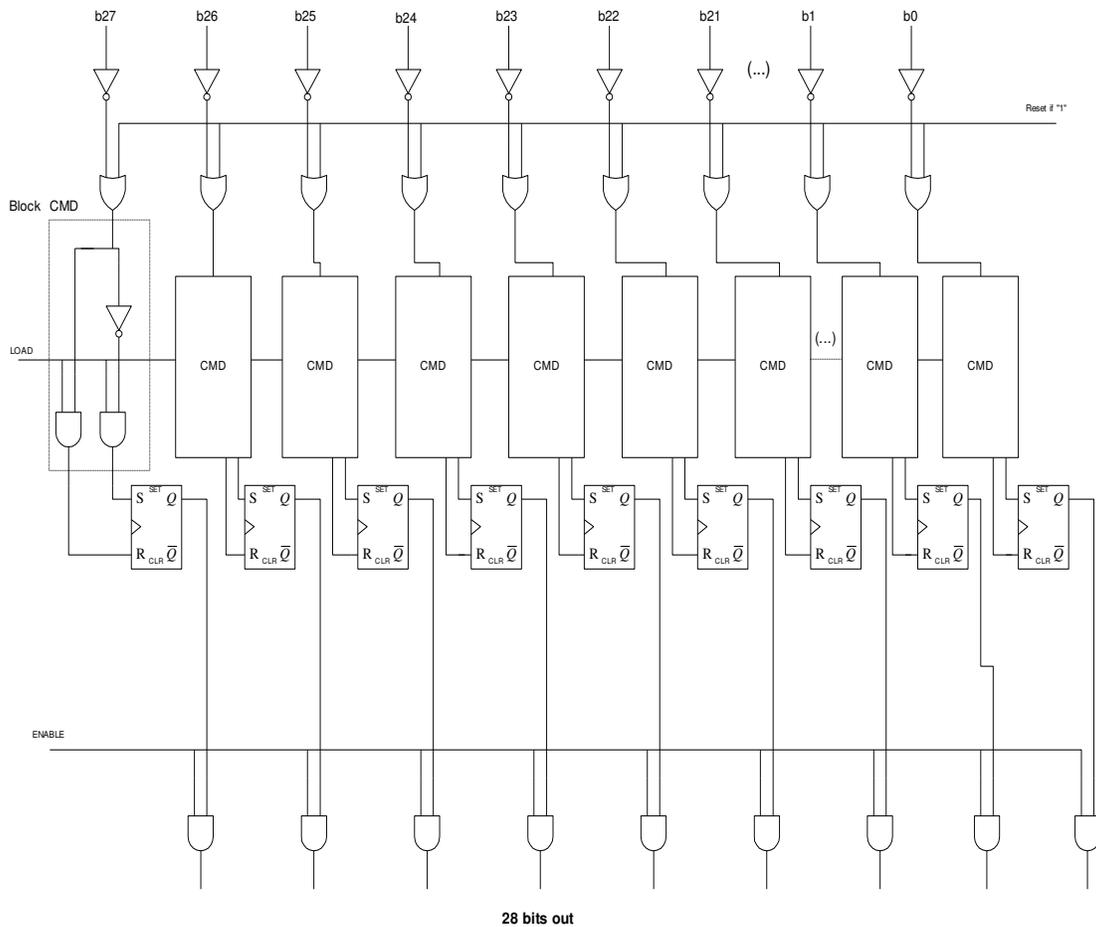


Figura 3.2.6 – Estrutura Interna dos registros

A lógica de trigger quando recebe um sinal de “enable” do discriminador vai simplesmente setar um bit interno (Flag) indicando que dados válidos sobre um hit necessitam ser mandados. Após o término do pacote de colisão, o controlador de trigger interroga simultaneamente em todas as camadas o primeiro IC da cadeia de sinais Token-in/Token-out.

- Uma vez que os vários Chips estão encadeados em sequencia (daisy-chain) com respeito a uma mesma camada, o controlador de trigger interroga somente o primeiro chip da fila, se este tem algum dado para enviar ele o enviaram via barramento de dados. Se ele não tiver nada a enviar ele passará o sinal de token para o próximo chip ate que o ultimo chip da fila tenha tido a chance de responder.
- Assim em essência a leitura dos dados de cada chip é feita pela passagem de um token pelos vários chips alocados para uma mesma camada. Quando o CI tem dados para reportar ele retém o token e envia os dados, caso contrário ele passa o token para o próximo chip da fila.

- Durante a fase de configuração inicial, este mesmo princípio é empregado. Uma linha intitulada “config” e mantida em 0V enquanto o sinal chamado “interrogate” e propagado ao longo da cadeia de ICs. Cada IC retém o token até que ele tenha lido todos os bits de necessários para configurar as equações que ele vai seguir na sua memória. Após isto ele passa o token para o próximo CI na fila. Com isto cada chip é programado em sequência começando com os responsáveis pela camada mais externa e terminando com os da camada mais interna.
- É importante notar que todas estas operações são controladas pelo bloco lógica de I/O.

Esta arquitetura pode ser expandida de forma a combinarmos os sinais de outros setores usando-se os sinais de Token-in e token-out conforme está mostrado na figura 3.2.7 e 3.2.8. Nesta figura e mostrado que se passarmos o sinal de token de chip para chip de uma maneira circular todos os chips que tem algum hit a reportar terão a chance de fazerem isto. Cada chip tem seu próprio endereço de forma que cada setor pode ter sua própria programação. Figura 3.7 mostra os chips combinados num arranjo “cascade” para chips de uma mesma camada e a figura 3.8 mostra vários arranjos “cascade” combinados em paralelo, uma “cascade” para cada camada do detector.

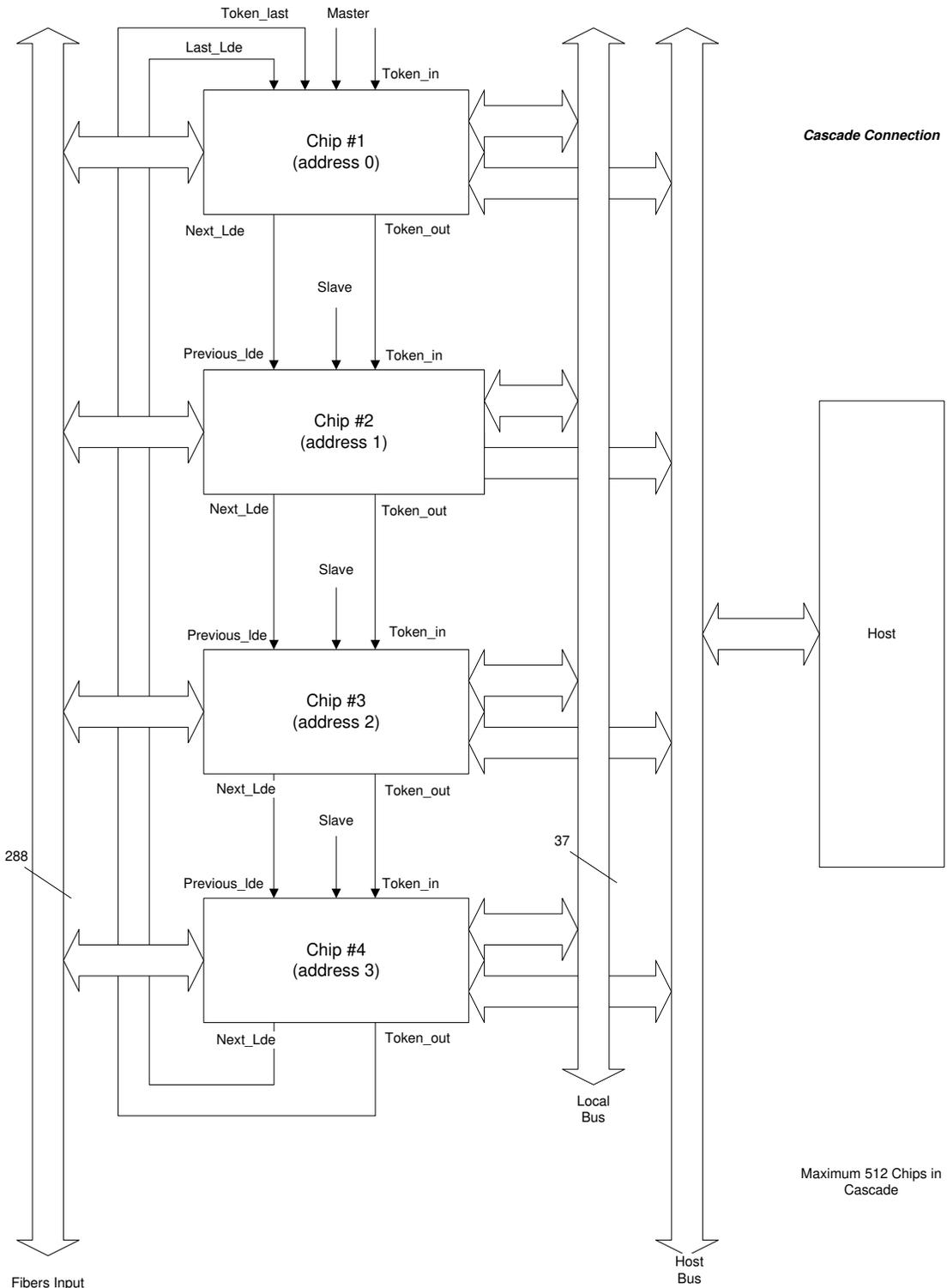


Figura 3.2.7 – Associando vários LDEs por camada formando uma cascata

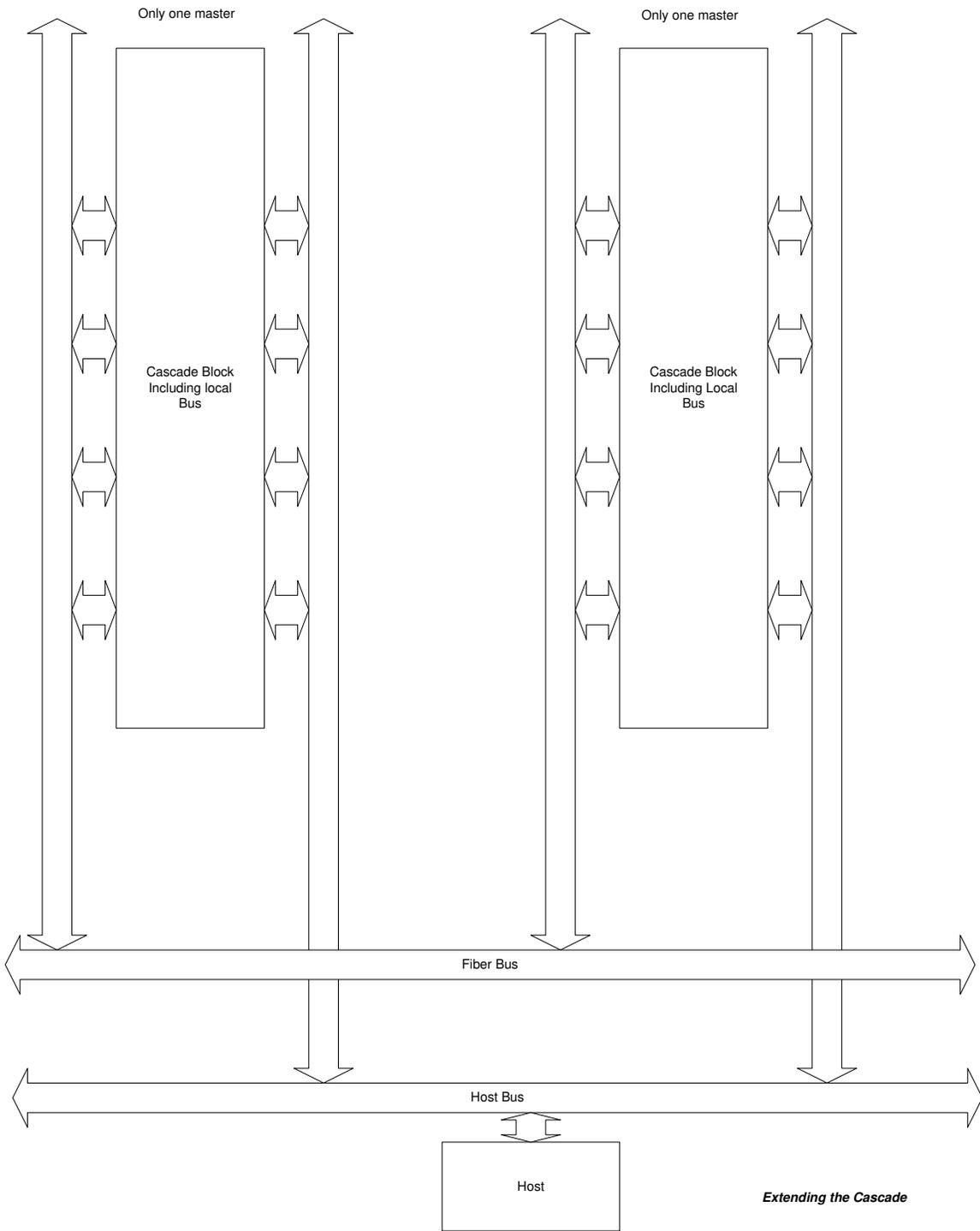


Figura 3.2.8 – associando uma cascata para cada camada

3.3 - Algoritmos usados no LDE

Nesta seção nós detalhamos os algoritmos usados na implementação do sistema em forma de diagramas de fluxo. Estes diagramas implementam com mais detalhes os algoritmos que discutimos na seção anterior. É importante notar que os diagramas que veremos a seguir já foram mostrados de maneira simplificada em seções anteriores, onde foi descrito a filosofia de operação do sistema. Daqui em diante nos concentramos em detalhar os sinais de controle e a criar o protocolo de comunicação.

Iniciamos esta discussão mostrando uma visão mais detalhada do sistema conforme mostrado nas figuras 3.9, 3.10, 3.11, 3.12. Estas figuras serão referenciadas varias vezes ate o fim desta dissertação.

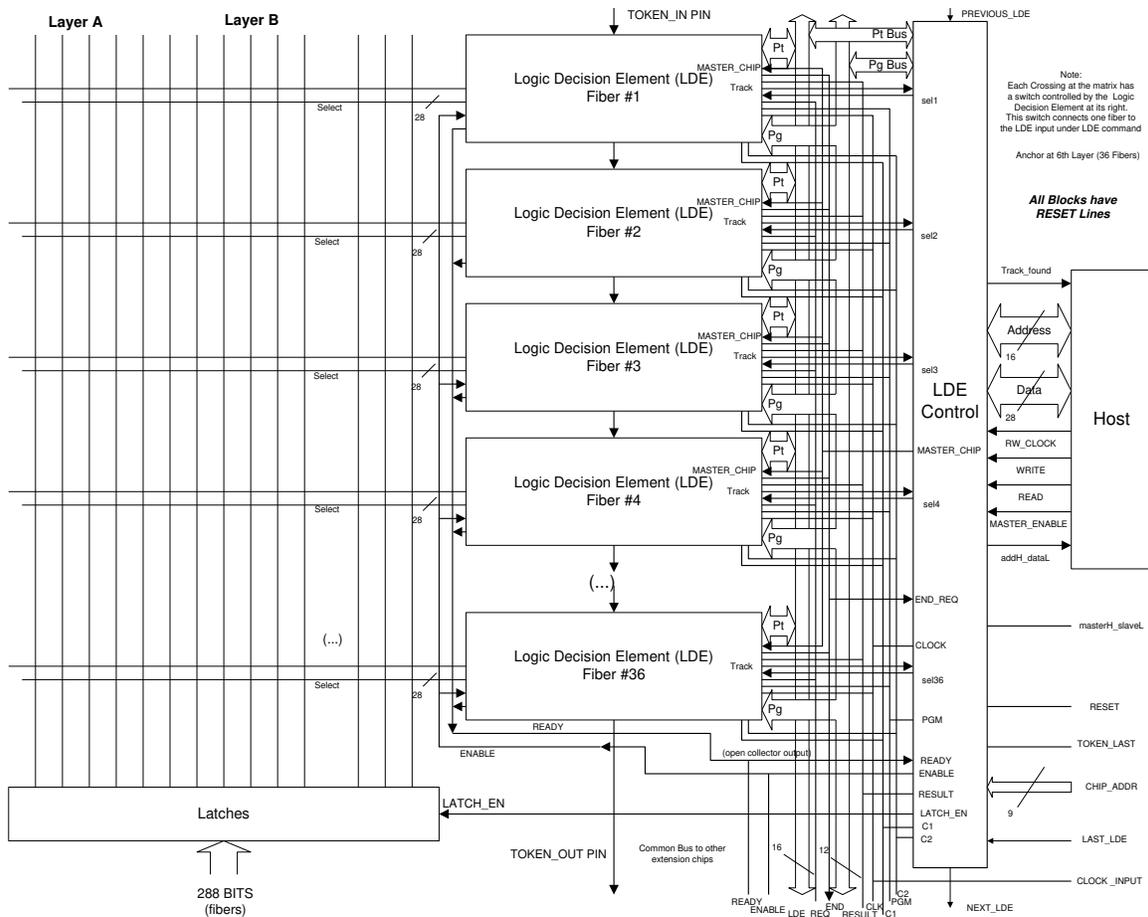


Figura 3.3.1 – Visões detalhada do CI

A figura 3.3.1 mostra uma visão detalhada do CI básico. Este CI foi projetado para acolher até 36 fibras podendo ser expandido conforme vimos na seção anterior. Estão também mostrados na figura os nomes dos sinais de controles e barramentos usados na operação do sistema. Poderíamos listá-los um a um porém achamos que é mais prático ver o seu uso através de gráficos de fluxo na próxima seção.

A figura 3.3.2 nos mostra o interior de cada um dos LDEs desta vez detalhando os sinais de controle e barramento.

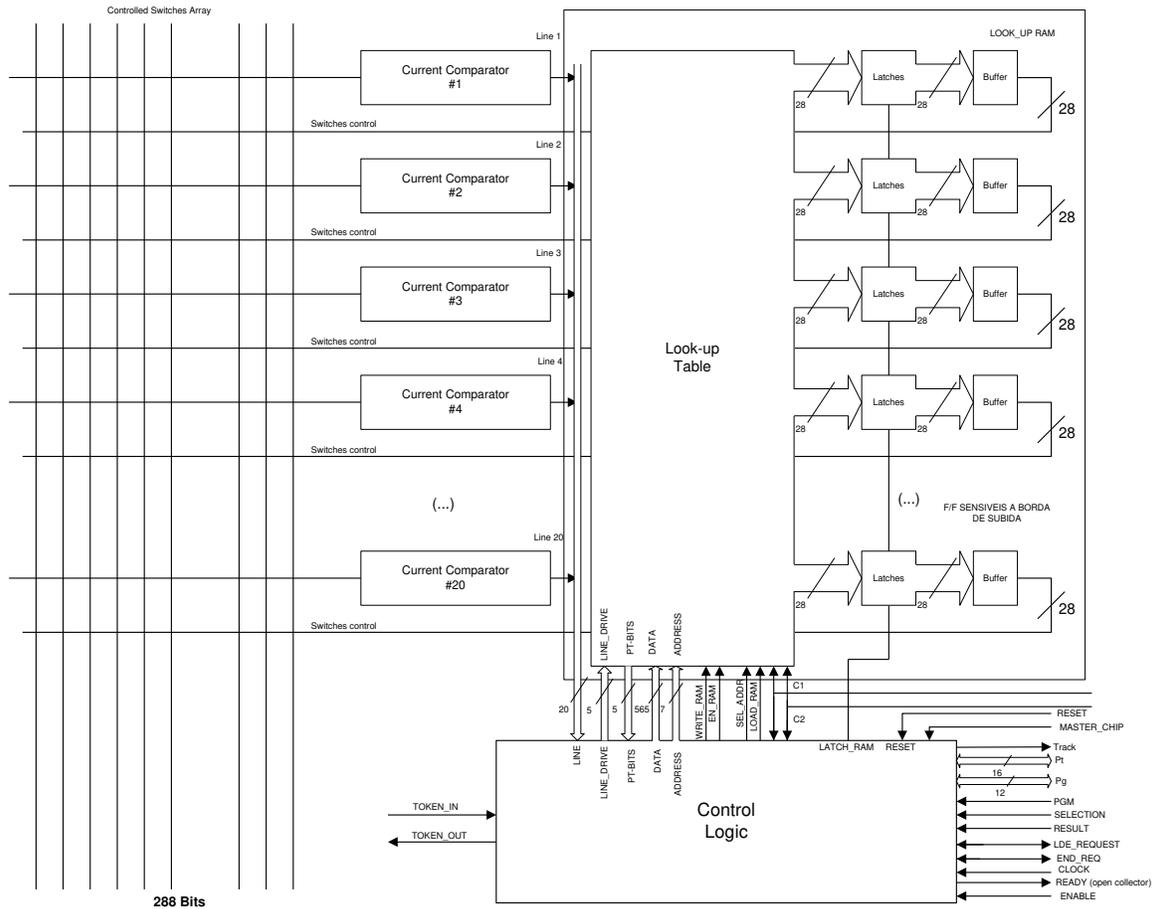


Figura 3.3.2 – Visão detalhada de cada um dos LDE

A figura 3.3.3 mostra o circuito eletrônico dos comparadores de corrente (current comparators) onde usamos o modo corrente de operação para aumentarmos a velocidade do bloco.

E importante notar que conforme mostrado na figura 3.3.3 os comparadores de corrente são programados através dos sinais “enable_when_step1&step 2” e “enable_when_step3” para um limiar de “1” nos passos 1 e 2 (Esquema 1-of-2) e para um limiar de “3” no passo 3 (esquema 3-of-4) para incluir tolerância contra os danos por radiação nas fibras conforme já discutido.

Finalmente a figura 3.3.4 nos mostra detalhes da look-up table onde podemos ter detalhes dos sinais internos e dos multiplexers usados para controlar o acesso à memória nos modos de configuração e no modo de operação.

O uso de todos estes sinais será detalhado a seguir onde descreveremos o protocolo de comunicação entre os vários blocos.

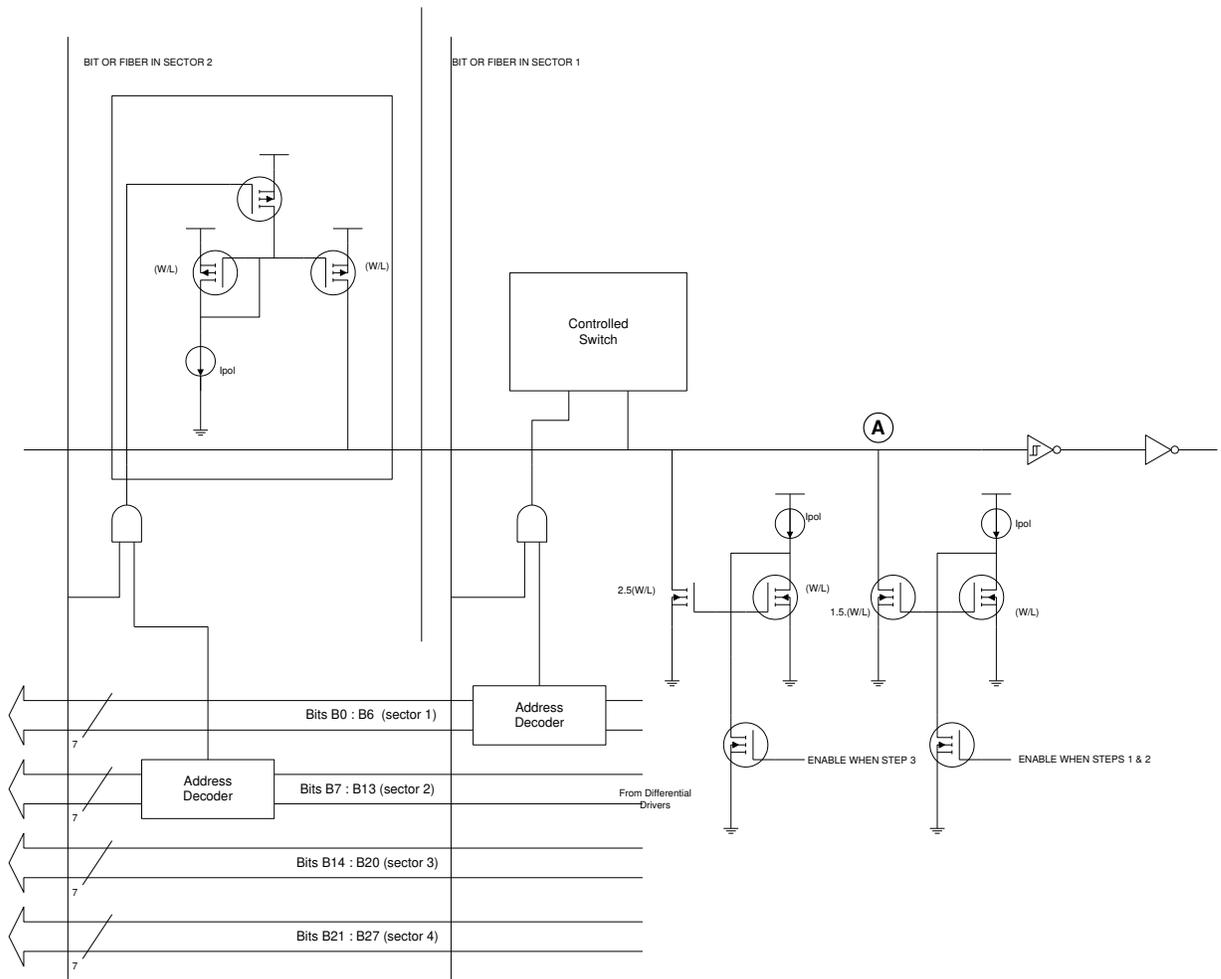


Figura 3.3.3 – Comparador de corrente

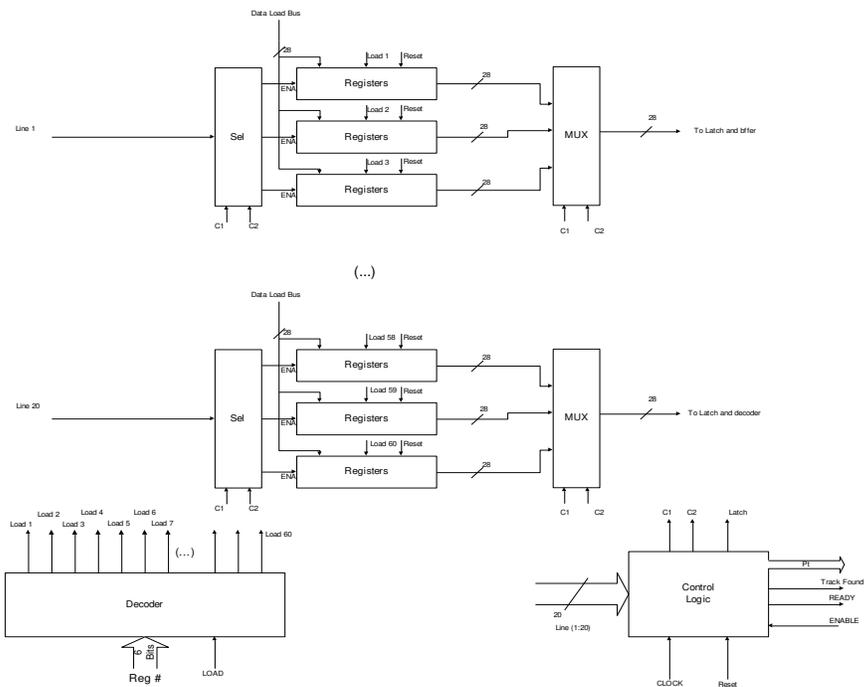
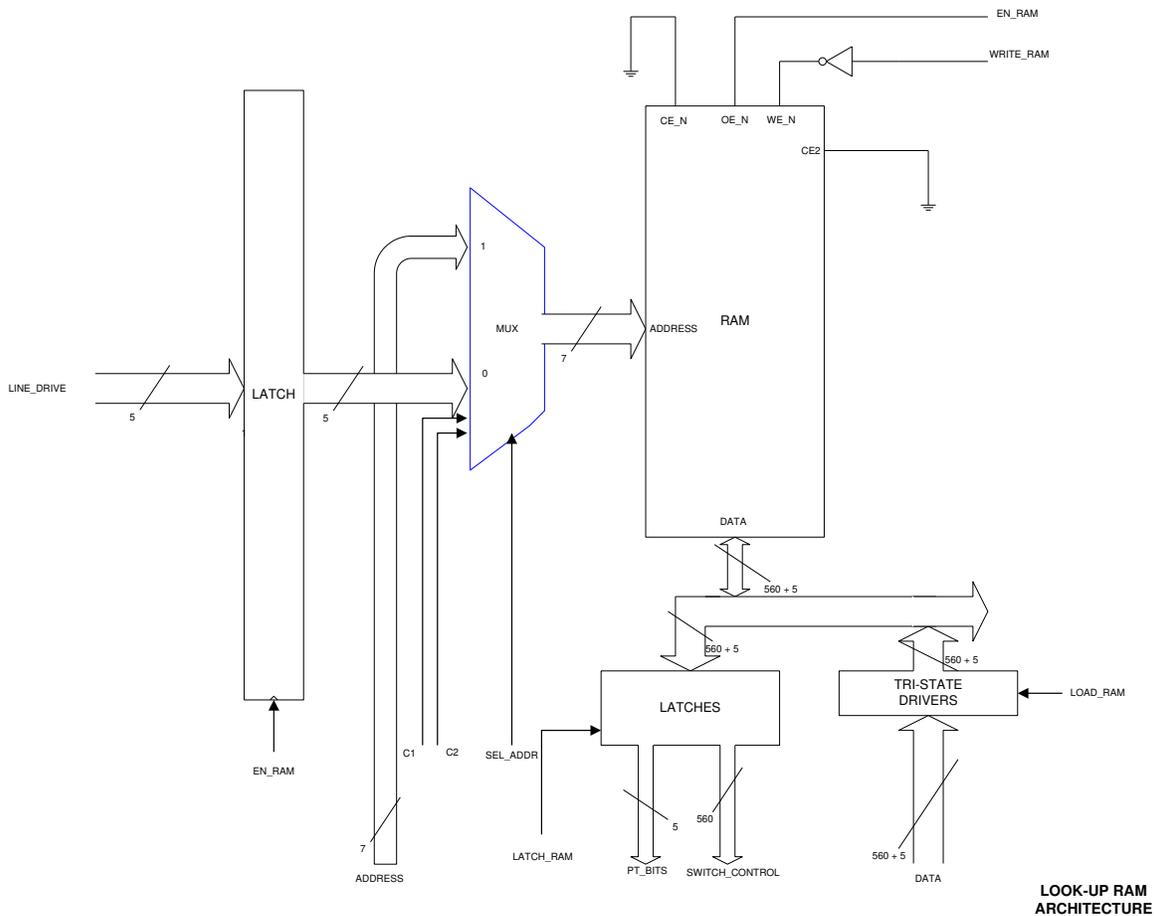


Figura 3.3.4 – Visão detalhada da tabela Look-up

Como veremos adiante cada bloco será mapeado em uma máquina de estado em VHDL. Na próxima seção descreveremos como os blocos interagem entre si e a criação do protocolo usado.

3.4 – Protocolo de comunicação

3.4.1 -Algoritmo do LDE:

A operação do sistema passo a passo:

1. O bloco “interface logic” inicializa os vários LDEs com as equações necessárias na memória:
 - O host avisa o bloco LDE_CONTROL que ele deseja acessar a primeira posição da memória de um dado LDE para fazer o download dos dados colocando um “1” lógico na linha “WRITE” e colocando os dois endereços que são necessários: o endereço da equação que vai ser escrita no barramento “DATA” (28 bits) e o endereço do LDE o qual o Host quer acessar no barramento “ADDRESS” (16 bits). Vamos supor que o endereço 00H e posto no barramento ADDRESS o que seleciona o primeiro LDE. O bloco LDE_CONTROL seleciona o primeiro LDE (no topo da figura), elevando a linha “SELECT 1” para “1”. A seguir o endereço 00H e colocado no barramento PG. O LDE responde a isso fazendo uma ponte entre a posição 00H em RAM (28 bits) e o barramento composto pela concatenação dos barramentos PT (14 bits) e PG (14 bits). A seguir o LDE_CONTROL pulsa a linha PGM para armazenar este dado e, após baixar a linha ADD_H/DATA_L para zero de forma a ordenar ao host para que ponha os dados que serão gravados no barramento DATA. O host põe os dados no barramento DATA (que são copiados no barramento composto por PG e PT uma vez que estes estão interconectados internamente) e pulsa a linha PGM de forma a armazenar os dados na memória. Esta operação de escrita se repete por 21 vezes de forma a carregar todos os bits (28 bits) associados com cada passo na RAM (20 equações mais 1 endereço que nos fornece o Bin do momento – Pt-bin). Este ciclo se repetirá para cada posição na memória e sempre sob comando do host. O host alterna entre os dados propriamente ditos e o endereço da equação no barramento DATA enquanto o barramento address mantém o endereço do LDE fixo. As linhas de endereço do HOST também afetam o contador modulo-3 interno ao LDE_CONTROL que gera os bits C1 e C2 como parte do esquema de endereçamento. À medida que cada LDE é programado o LDE_CONTROL mantém uma lista em memória de todos os LDEs usados e qual equação é programado em cada um deles.
 - Após o término do passo acima, o host programa o registro (IN_USE) com um “1” lógico para cada um dos 36 LDEs na Sexta camada.

Blocos envolvidos: LDE_CONTROL, HOST AND LDE

Sinais envolvidos: *do HOST para LDE_CONTROL* : WRITE, DATA, ADDRESS

do LDE_CONTROL para HOST : ADD_H/DATA_L

do LDE_CONTROL para LDE: SELECT, PGM, PG bus, C1 and C2

- 2) Quando o host indicar MASTER_ENABLE="1", o host autoriza a operação do sistema.

Quando o LDE_CONTROL recebe o comando acima, um ciclo de aquisição de dados é iniciado. Primeiro o LDE_CONTROL manda um sinal LATCH_EN=1 que armazena em latches os dados das 288 fibras no bloco LATCH. Após isto o LDE_CONTROL envia o sinal ENABLE=1 o qual autoriza os LDEs a avaliar as equações baseando nos dados colhidos das fibras. No interior de cada LDE o bloco CONTROL_BLOCK monitora as saídas de cada uma das equações e, se mais do que uma equação é verdadeira, a lógica de controle irá requisitar um LDE vago para acompanhar estas possibilidades extras. Esta operação é feita da seguinte maneira começando com a linha localizada no topo:

- O LDE no topo da cadeia tem o seu pino TOKEN_IN fixo em "1" lógico de tal maneira que este começa a transmitir seus pedidos de LDEs adicionais (se os tiver). Esta requisição é feita pela colocação no barramento PT da seguinte informação: O número da fibra que ele está processando (por exemplo: fibra 1) usando 6 bits e o número da linha o qual ele quer que outro LDE vago processe usando 5 bits. A seguir o LDE baixa a linha LDE_REQUEST (que é coletor aberto).
- O LDE_CONTROL tem uma lista de todos os LDEs disponíveis (isto é todos os LDEs que foram programados previamente com as equações de uma dada fibra e não estão em uso) e baseando-se nesta informação e no dado que ele lê no barramento Pt ele seleciona um LDE extra (não em uso) através do sinal SELECT. O LDE selecionado estará carregado com as equações da mesma fibra que originalmente requisitou o LDE extra. O LDE selecionado irá decodificar isto e setar um "1" lógico num dos registros de entrada para armazenar os dados da fibra. Ele também irá escrever no registro IN_USE="1". O LDE que foi endereçado a seguir sinaliza o fim desta operação pelo abaixamento da linha END (também open-coletor).
- Quando o LDE que requisitou o LDE vago recebe o sinal END, ele coloca no barramento PT a sua próxima requisição e o ciclo se repete até que não haja mais requisições. No caso do LDE que requereu ajuda não receber resposta após 3 ciclos de clock, o LDE irá setar um registro ERROR interno e prosseguir com a sua próxima requisição.
- Se o LDE_CONTROLER não consegue achar um LDE vago com a mesma equação dentro de sua tabela ele eleva o pino NEXT_LDE para "1". Isto faz com que a vez passe para o próximo chip (lembrando que o pino NEXT_LDE está conectado ao pino PREVIOUS_LDE do próximo chip LDE_CONTROLER da cadeia) o qual em paralelo também pesquisou a sua própria tabela e talvez possa atender a

requisição. Se o segundo chip também não consegue atender à requisição ele passa a vez para o próximo e assim vai até que o último chip da cadeia passa a vez de volta para o primeiro chip (através do pino LAST_LDE). Se o LDE_CONTROLER do chip que mandou a requisição recebe de volta a vez (sinal lógico alto no pino LAST_LDE) ele sabe que a requisição não pode ser satisfeita pelo sistema. Em resposta ele escreve ERROR_ALLOC=1 e retorna END=1 para o LDE que fez a requisição.

- Quando não há mais requisições, o LDE passa a vez para o seu vizinho através do sinal TOKEN_OUT=1. Esta linha voltará à zero no início do próximo ciclo quando o sinal ENABLE=1 é recebido.
- Quando o último LDE termina o seu processamento, a sua linha TOKEN_OUT vai a 1. Uma vez que o pino TOKEN_OUT do último LDE da cadeia é conectado ao pino TOKEN_LAST do primeiro LDE_CONTROLER (que iremos chamar de mestre), quando o mestre receber de volta a vez (TOKEN_LAST=1), este irá esperar pela chegada do sinal READY=1. Quando este sinal chega o LDE_CONTROLER abaixa a linha ENABLE=0 e incrementa o contador C1 e C2 apontando o próximo passo. Esta mudança em C1 e C2 dispara o reset de todas as linhas TOKEN_OUT de todos os LDEs. Após isto o LDE_CONTROLER sinaliza ENABLE=1.

Blocos Envolvidos: LDE_CONTROL (LDE), HOST, CONTROL_BLOCK (LDE), LATCH and LDE

Sinais Envolvidos: *do HOST para LDE_CONTROL:* MASTER_ENABLE
do LDE_CONTROL para LATCH : LATCH_EN
do LDE_CONTROL para LDE : ENABLE
do LDE para LDE: TOKEN_IN, TOKEN_OUT, IN_USE, LDE_REQUEST, END, barramento PT

- 3) Se o bloco CONTROL_LOGIC (dentro de cada um dos LDEs) detecta uma ou mais linhas em nível alto, ele irá processá-las de maneira apropriada: Se existem várias, a linha mais ao topo será processada localmente e as demais serão alocadas para LDEs vagos conforme visto no passo anterior. Quando não há mais requisições ele escreve TOKEN_OUT=1 e então habilita as entradas para a RAM usando o EN_RAM=1 (lembrando que a esta altura somente uma linha está ativa). As saídas da RAM são armazenadas pelo sinal LATCH_RAM (além dos comandos que afetam as chaves a saída da RAM também contêm 4 bits que informam o bin do momento relativo da partícula). O CONTROL_LOGIC também armazena este número de 4 bits num registro. A cada passo este registro será atualizado conforme veremos. Após isto o LDE eleva o sinal READY (open-collector) para 1.

Blocos Envolvidos: CONTROL_LOGIC (LDE) e LDE_CONTROL

Sinais Envolvidos: *Interno ao LDE:* EN_RAM e LATCH_RAM
do LDE para LDE_CONTROL : READY
entre LDEs : TOKEN_OUT

OBS: O LDE que estava vago e recebeu dados para processar em um dado momento irá armazenar o numero da equação (numero da linha) em seu próprio registro. Isto ira acarretar a geração de um Bin de momento (4-bits) para este novo dado neste LDE. No entanto este passo estará correto uma vez que todos os LDEs (vagos e ativos) são incrementados juntos via C1 e C2.

- 4) Se a CONTROL_LOGIC detecta que nenhuma linha foi a "1", esta reseta o registro interno IN_USE=0 assim como todos os seus registros internos e espera pelo sinal TOKEN_IN. Quando este chega o LDE põe a mensagem "LDE numero XX está livre" (a equivalência binária desta mensagem esta mostrada no fim deste capitulo) no barramento PT. Esta mensagem é usada pelo LDE controller para atualizar a tabela de alocação. Após atualizar esta tabela o LDE_CONTROL retorna END=1.

Blocos Envolvidos: (CONTROL_LOGIC) LDE e LDE_CONTROLLER

Sinais Envolvidos: *entre LDE e LDE_CONTROL: Barramento Pt e END*

- 5) Ao fim do terceiro ciclo, o LDE_CONTROL envia um sinal RESULT="1" para todos os LDEs. Este sinal reseta todos os pinos TOKEN_OUT de todos os LDEs para zero. O LDE mais ao topo tem o sinal TOKEN_IN fixo em "1" e assim ele começa o processo de enviar o conteúdo dos registros Pt, ERROR e todos os demais registros. Após receber o sinal RESULT, o bloco CONTROL_LOGIC do LDE mais ao topo ira sinalizar TRACK="1" e pôr o conteúdo do registro Pt (4 bits) juntamente com os conteúdos dos registros ERROR e IN_USE no barramento PT. O LDE ira manter os valores até que o LDE_CONTROL pulse a linha PGM. Neste momento o LDE_CONTROL armazena este dado no seu registro interno. Após o LDE mais ao topo terminar o seu processamento, ele eleva o seu pino TOKEN_OUT para "1" de forma a habilitar o próximo LDE a transferir seus resultados para o LDE_CONTROL. E assim vai de LDE em LDE ate que o último LDE reporte os seus resultados. Se um dado LDE não tiver resultados para reportar ele simplesmente passa a vez para o próximo.

Blocos Envolvidos: CONTROL_LOGIC (LDE) e LDE_CONTROL

Sinais Envolvidos: *Entre LDEs: TOKEN_IN, TOKEN_OUT*

de LDE_CONTROL para LDE : RESULT, PGM

de LDE para LDE_CONTROL : Barramento PT e TRACK

- 6) Após isto o LDE_CONTROL ordena os valores de Bins de momentum (Pt) em ordem decrescente (somente os seis maiores serão reportados para o host).

Blocos Envolvidos: LDE_CONTROL

Sinais Envolvidos: *Nenhum*

- 7) *Quando o host sinalizar READ="1", o LDE_CONTROL ira reportar as 6 trilhas com os maiores valores de Pt no barramento DATA, uma trilha por clock sincronizado com o sinal RW_CLOCK (borda de descida)*

Blocos Envolvidos: LDE_CONTROL, HOST

Sinais Envolvidos: de HOST para LDE_CONTROL: READ, R/W_CLOCK
de LDE_CONTROL para HOST : DATA

8) Fim deste ciclo. Contador C1, C2 retorna a 00h

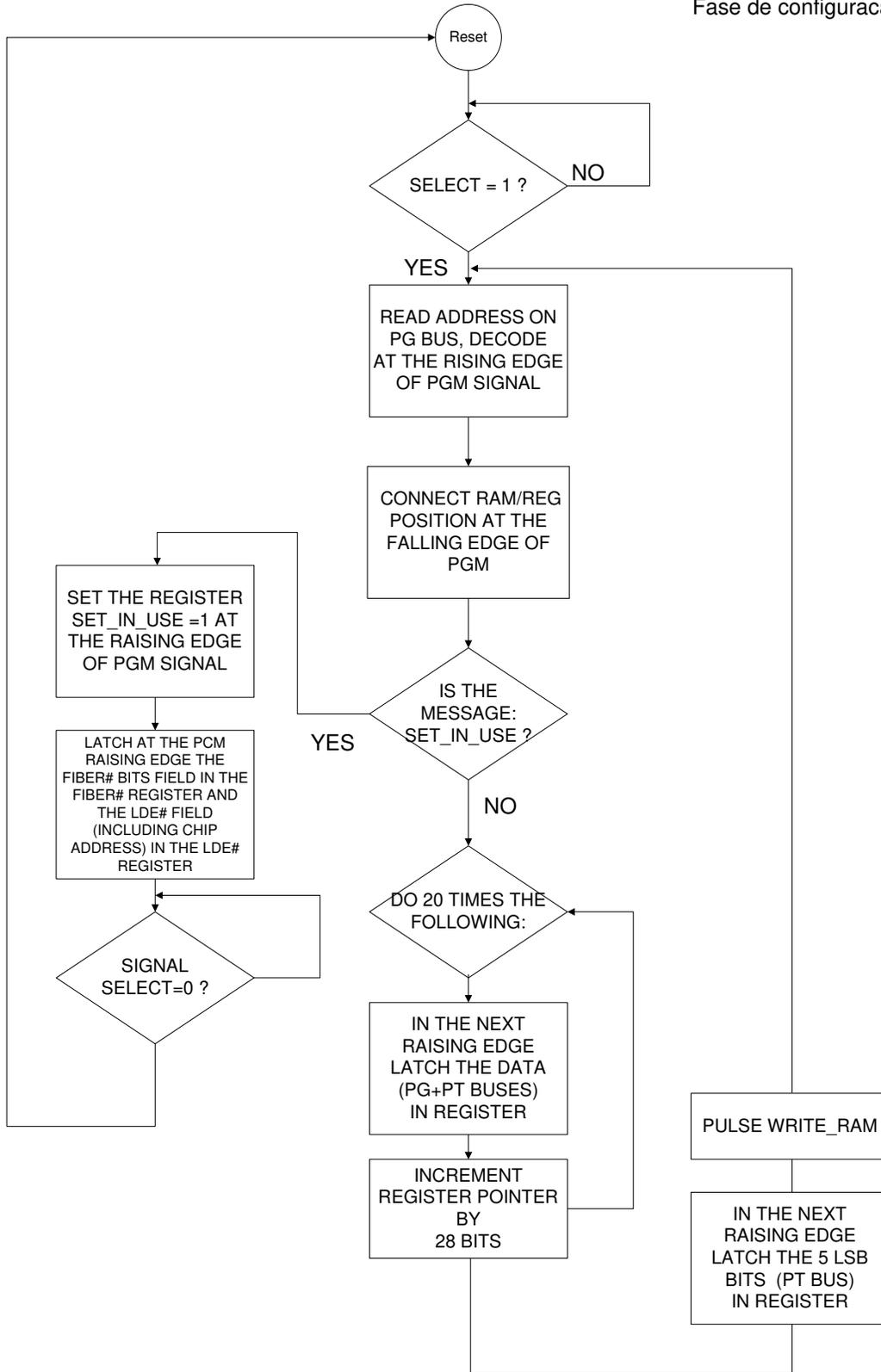
Nota Importante: Os bits C1, C2 são controlados pelo LDE_CONTROL através da contagem do número de pulsos MASTER_ENABLE.

Nota 1: Durante a fase inicial de configuração, alguns dos LDEs extras devem ser programados para serem cópias de alguns dos 36 LDEs inicialmente ativos (1 por fibra). O número de LDEs alocados como cópias para cada fibra (isto é carregado com as mesmas equações) será determinado por simulações de Monte Carlo do detector. O número de LDEs pode ser expandido pela conexão dos pinos dos barramentos comuns e definindo qual dos chips é o mestre usando o pino MASTER_H/SLAVE_L. Quando o chip é usado como master ele terá todas as funções descritas acima. Quando não usado como mestre o chip somente usa as suas funções de endereçamento para programar o chip durante o start-up. As funções de leitura e controle são todas feitas pelo chip mestre apenas.

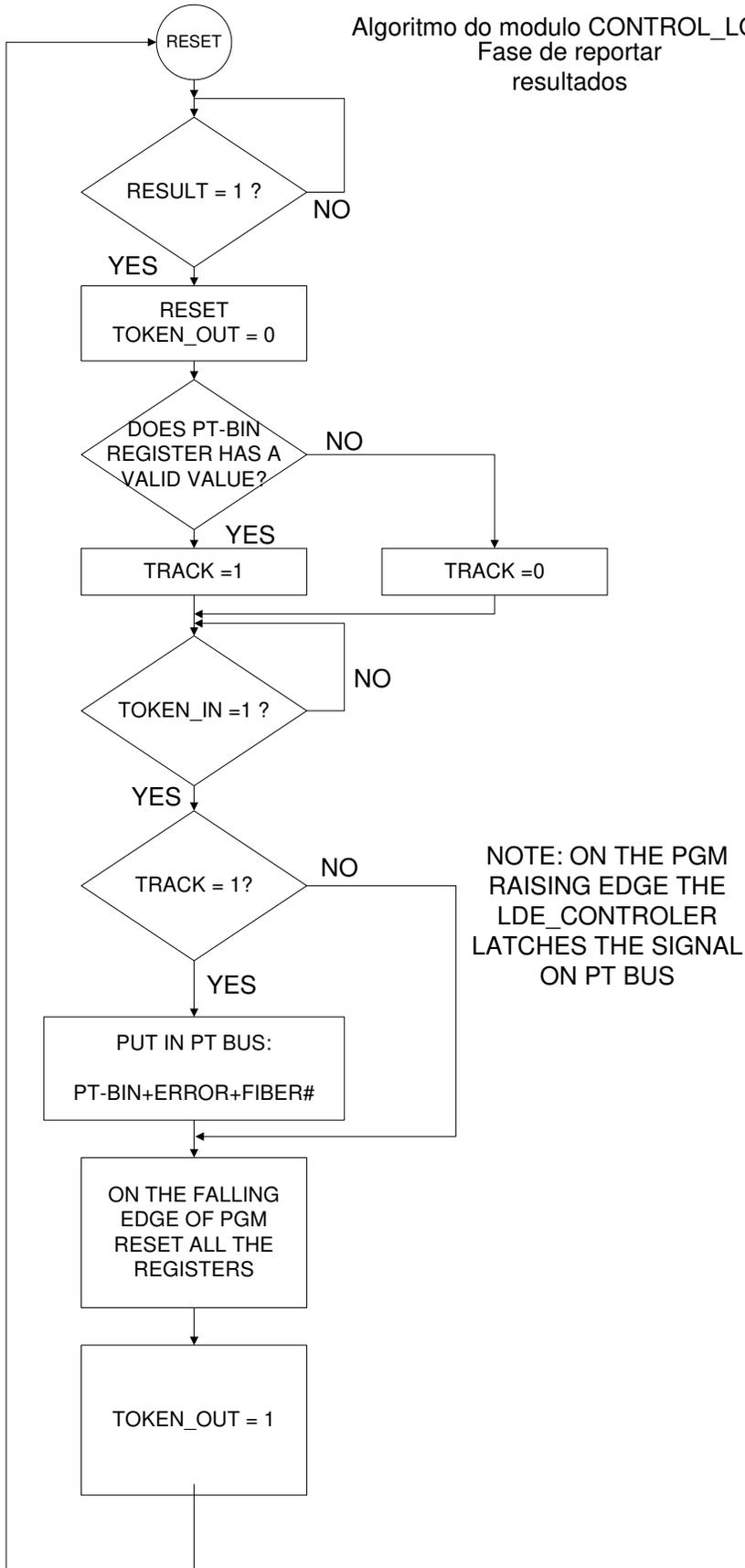
Nota 2: Cada um dos 36 LDEs dentro do chip é endereçado pelos 6 bits menos significativos do barramento de endereços. Um endereço de 10 bits é associado a cada chip externamente por meio de jumpers. Isto nos dá 1024 chips possíveis de serem interligados e, portanto 36.864 LDEs disponíveis. Cada chip irá acessar somente os 16 bits menos significativos do barramento de endereços durante seu período de configuração.

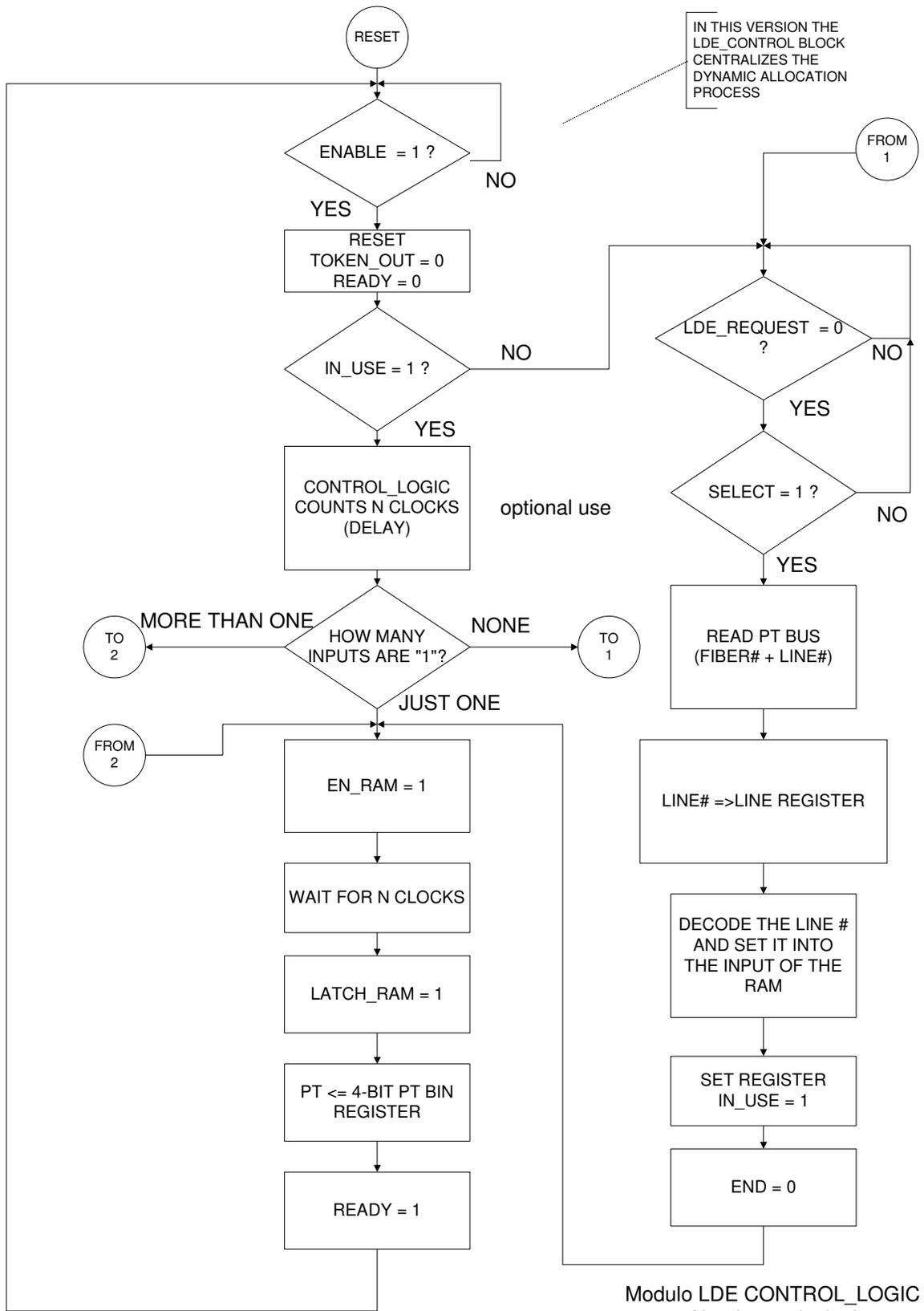
Nas Páginas seguintes temos os algoritmos do módulo control logic (o mais complexo de todos – o qual controla todos os LDE) detalhados sob a forma de fluxogramas. Estes serviram de base para a criação do código VHDL.

Algoritmo do modulo Control logic
Fase de configuracao

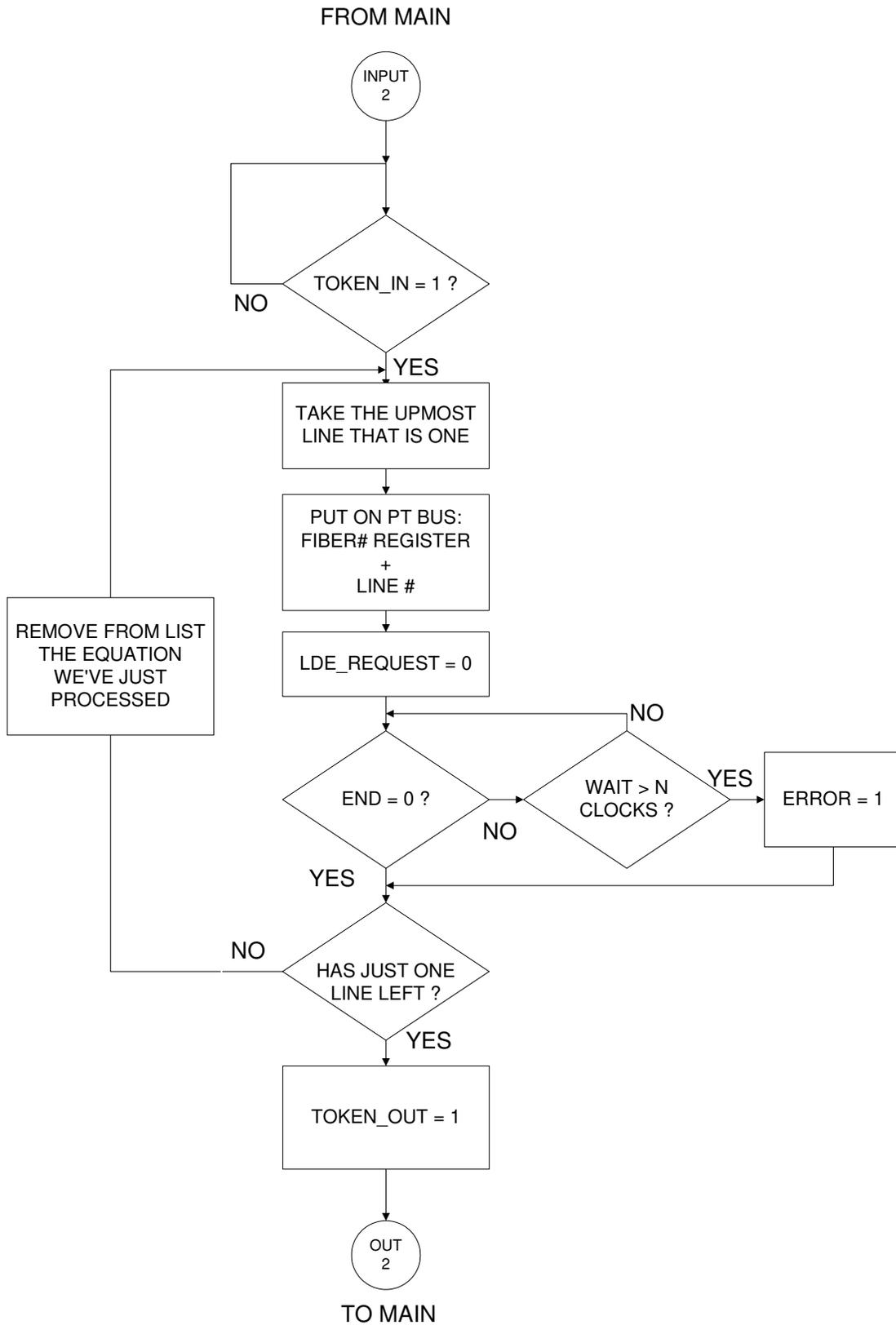


Algoritmo do modulo CONTROL_LOGIC
Fase de reportar resultados

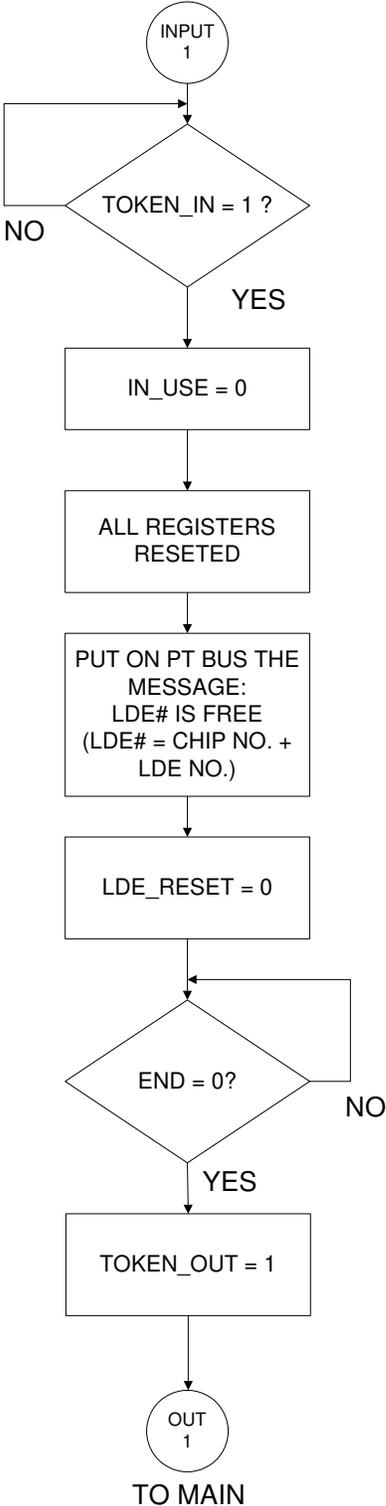




SUB-PROCESSO 2

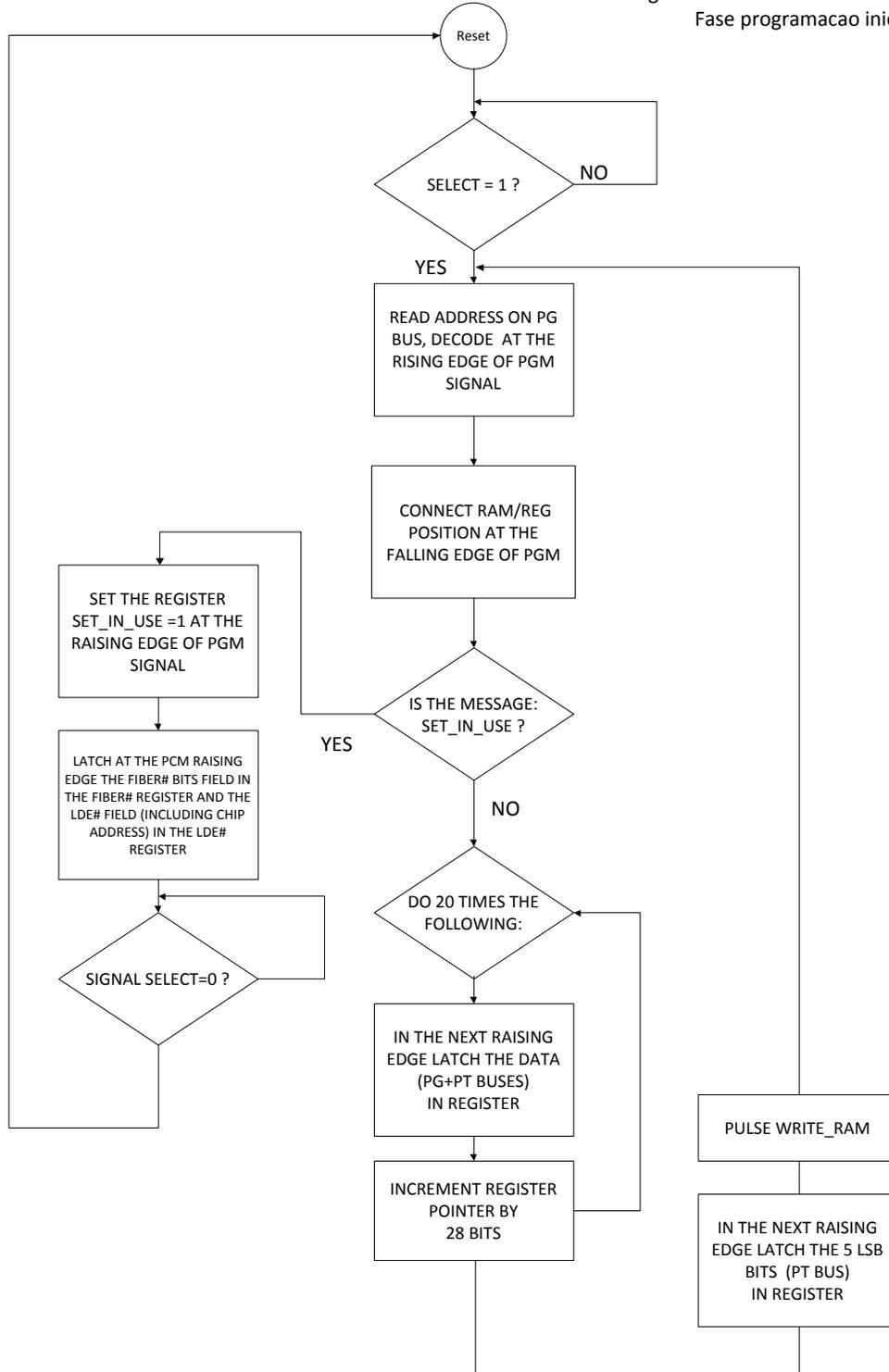


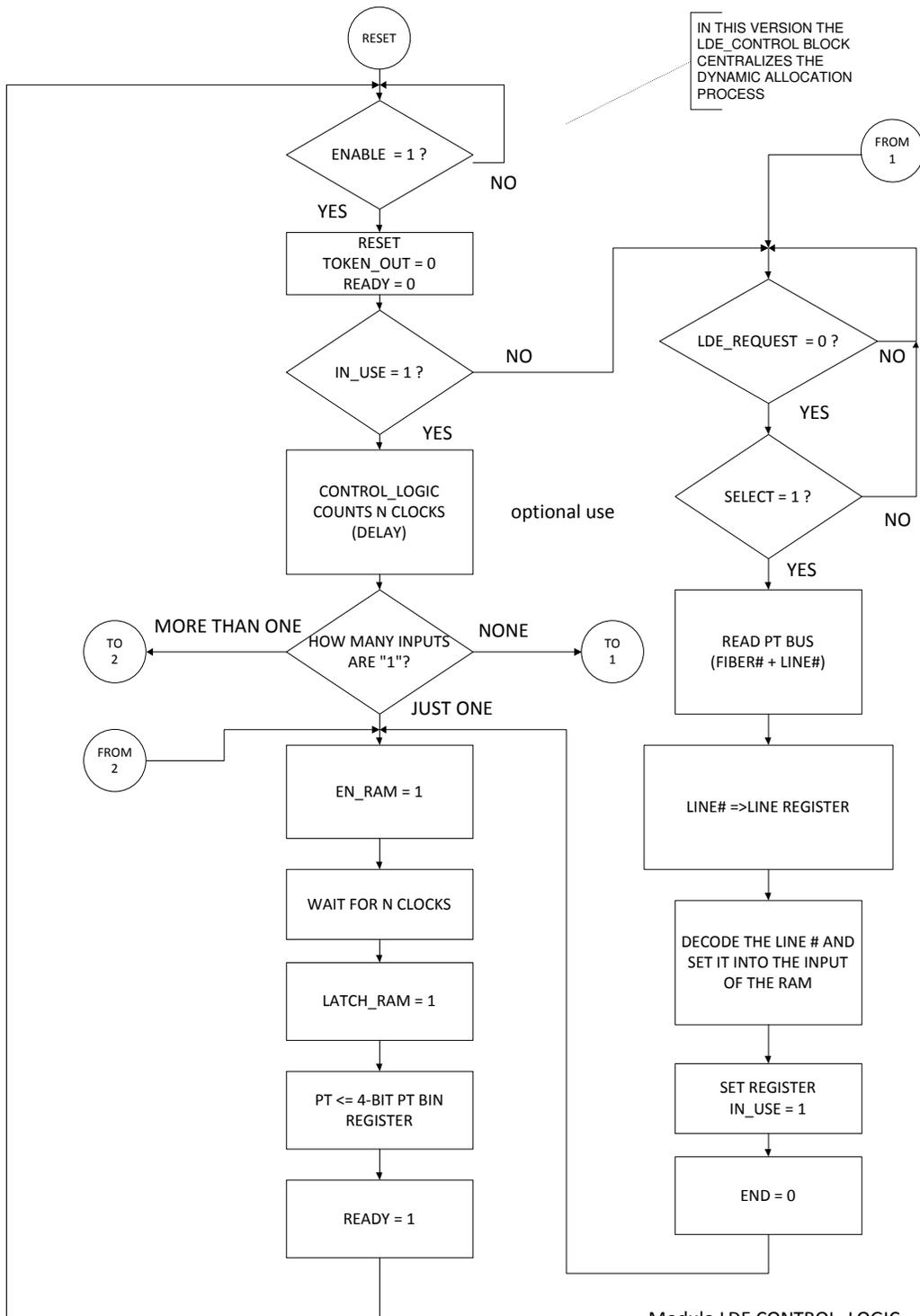
SUB-PROCESSO 1



Em continuação, nas páginas seguintes, temos os algoritmos do módulo LDE detalhados sob a forma de fluxogramas. Estes serviram de base para a criação do código VHDL. Cada página detalha um modo de operação:

Algoritmo – Modulo CONTROL_LOGIC
Fase programacao inicial

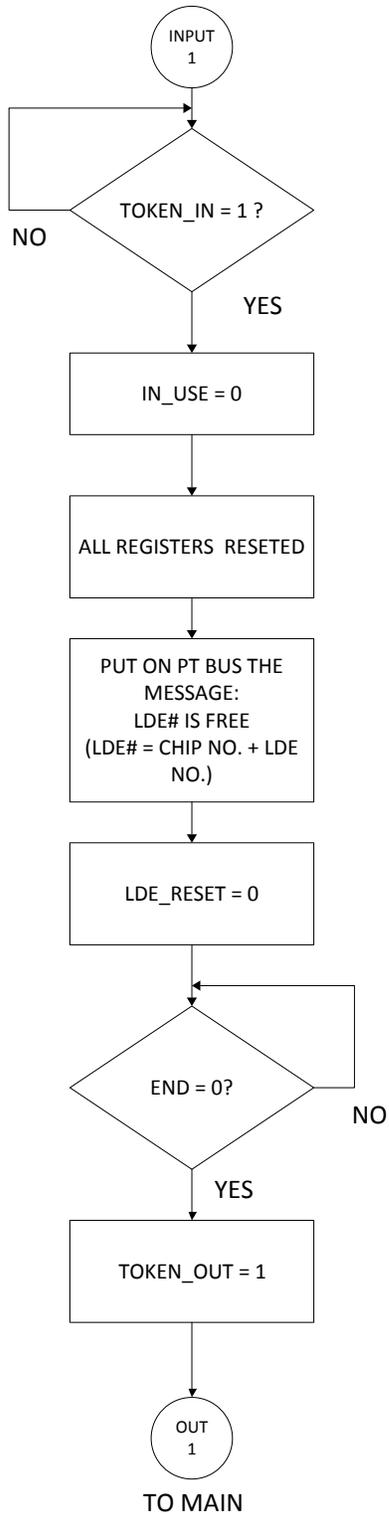




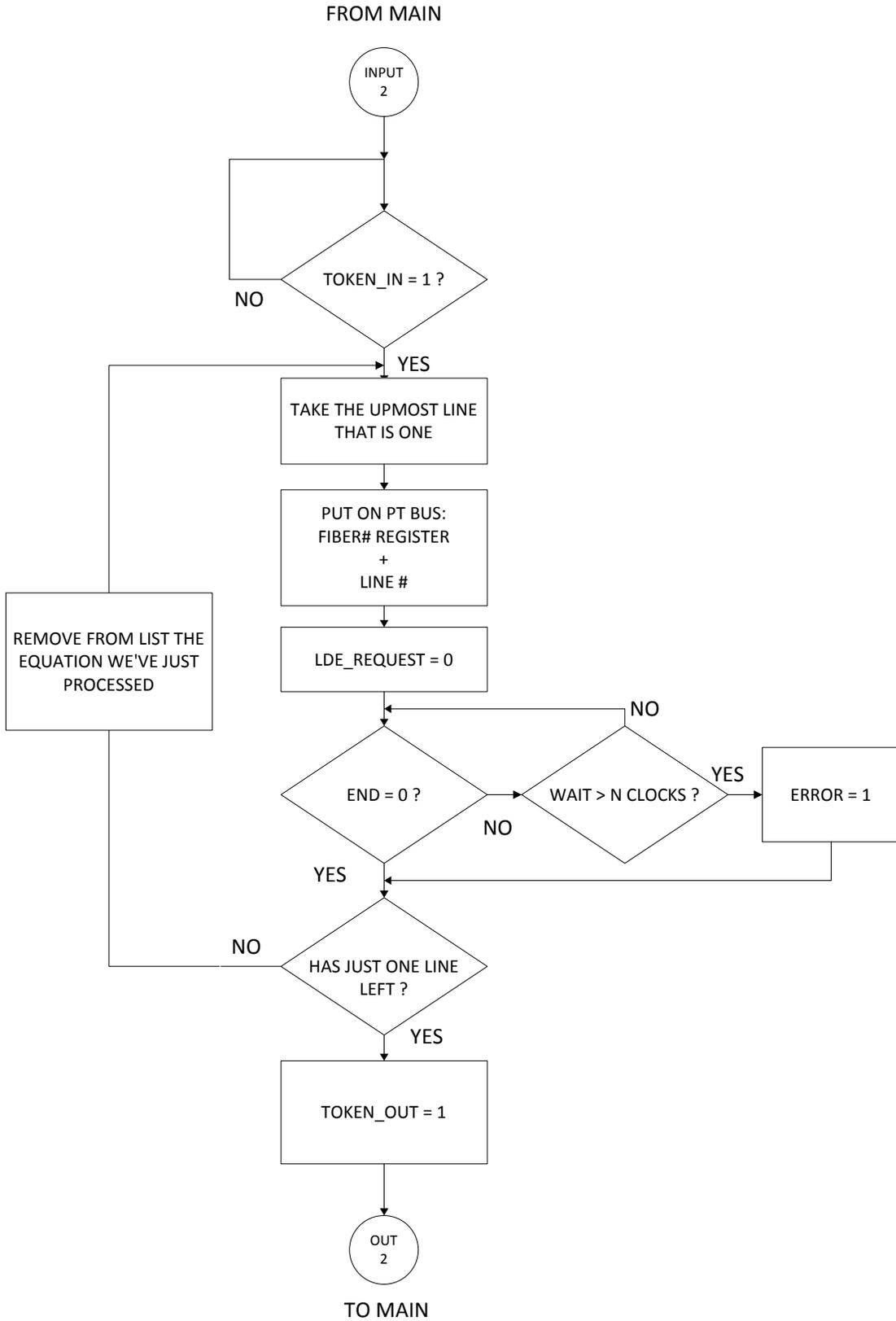
IN THIS VERSION THE LDE_CONTROL_BLOCK CENTRALIZES THE DYNAMIC ALLOCATION PROCESS

Modulo LDE CONTROL_LOGIC
Processo Principal

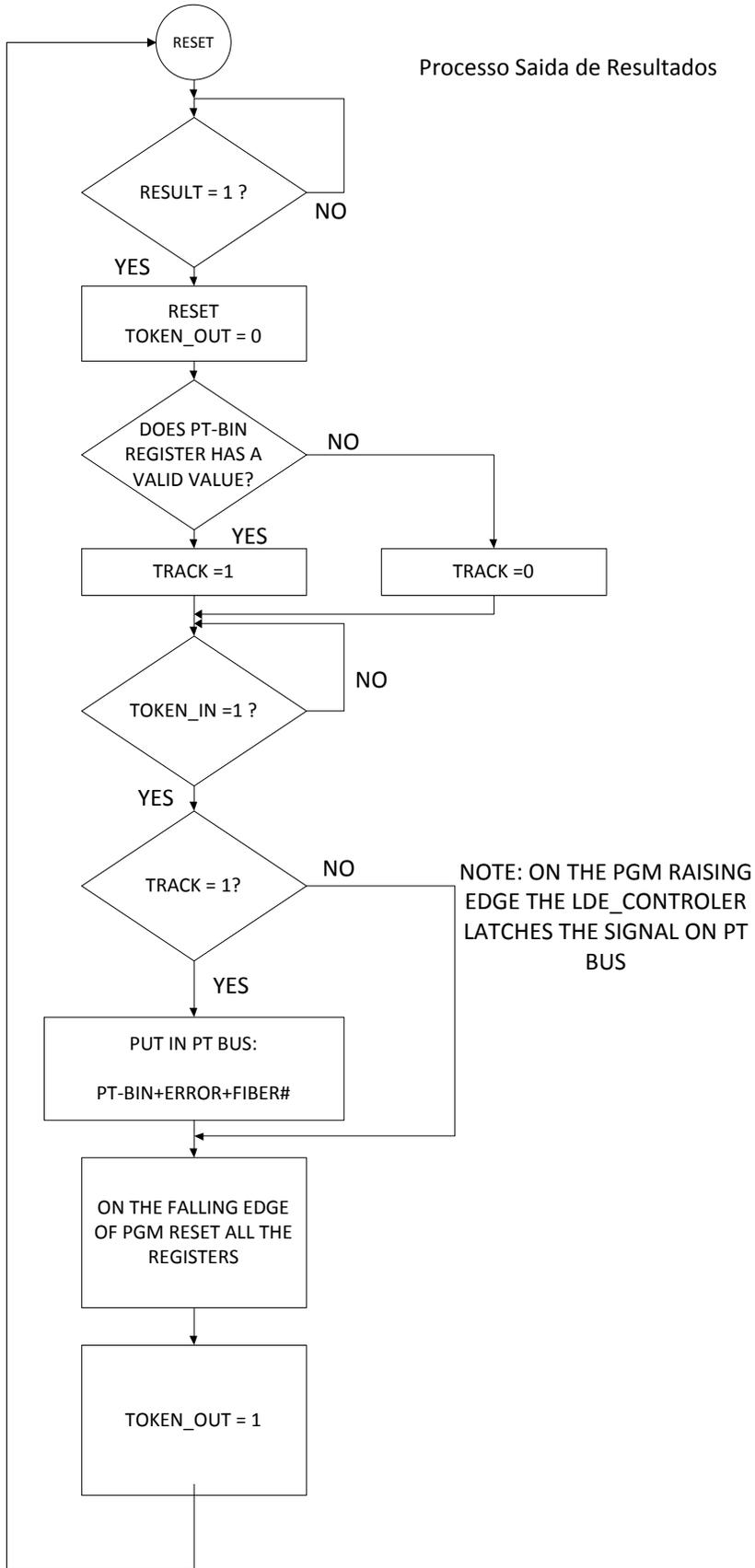
SUB-PROCESSO 1



SUB-PROCESSO 2



Processo Saida de Resultados



Capítulo 4 – Código VHDL e simulação Digital

4.1-Codificação em VHDL da parte digital

Usaremos a linguagem de descrição de hardware VHDL para escrevermos o código da lógica visando a sua posterior síntese em hardware usando-se ferramentas da Synopsys. O código foi subdividido nos seguintes sub-blocos:

1) LDE_CONTROLLER:

- LDE_CONT.vhd=> Bloco topo do LDE_CONTROLLER. Ele contém os seguintes sub-blocos interligados: *saida_cont*, *program_cont*, *master*, *slave*, *bus_cntl*, *track* e *initial*.
 - INITIAL.vhd → Controla a inicialização do *lde_controller*
 - PROGRAM.vhd → contém o bloco que coordena a programação de cada um dos LDEs
 - MASTER.vhd → Gerencia a operação do *lde_controller* quando no modo mestre
 - SLAVE.vhd → Gerencia a operação do *lde_controller* quando no modo escravo
 - BUS_CNTL.vhd → Gerencia contenção nos barramentos *pt* e *pg*
 - TRACK.vhd → Identifica as trilhas achadas pelos LDEs
 - Saida_cont.vhd → Gerencia a saída de dados para o processador *host*

2) LDE

- Lookup_t2.vhd → Tabela de busca onde as equações são armazenadas
- LDE_LOG.vhd → Lógica principal do LDE. Este bloco contém quatro sub-blocos: *Program*, *operação*, *result*, *saída*.
 - PROGRAM.vhd → Bloco ativo durante a programação inicial das equações.
 - OPERACAO.vhd → Bloco ativo durante a operação normal do LDE
 - RESULT.vhd → Calcula e ordena a lista das trilhas achadas em termos de maior momento (*pt*)
 - SAIDA.vhd → Gerencia a saída de dados para o LDE_CONTROLLER

As Figuras 4.1.1 e 4.1.2 mostram os blocos de topo para o LDE_CONTROLLER e LDE respectivamente. Nelas se pode ver a interconexão entre os diversos blocos. Nesta figura cada bloco corresponde a um componente codificado em VHDL.

Os códigos correspondentes a cada bloco estão no CD anexo. Todo os blocos foram codificados de forma compatível com o subconjunto da linguagem VHDL que o Synopsys consegue mapear para o nível de gates.

A única exceção é o bloco de memória (*table_lookup*), o qual foi codificado usando blocos LPM da Altera com o objetivo de acelerar a simulação. Durante a síntese usaremos as células de memória definidas para o processo usado.

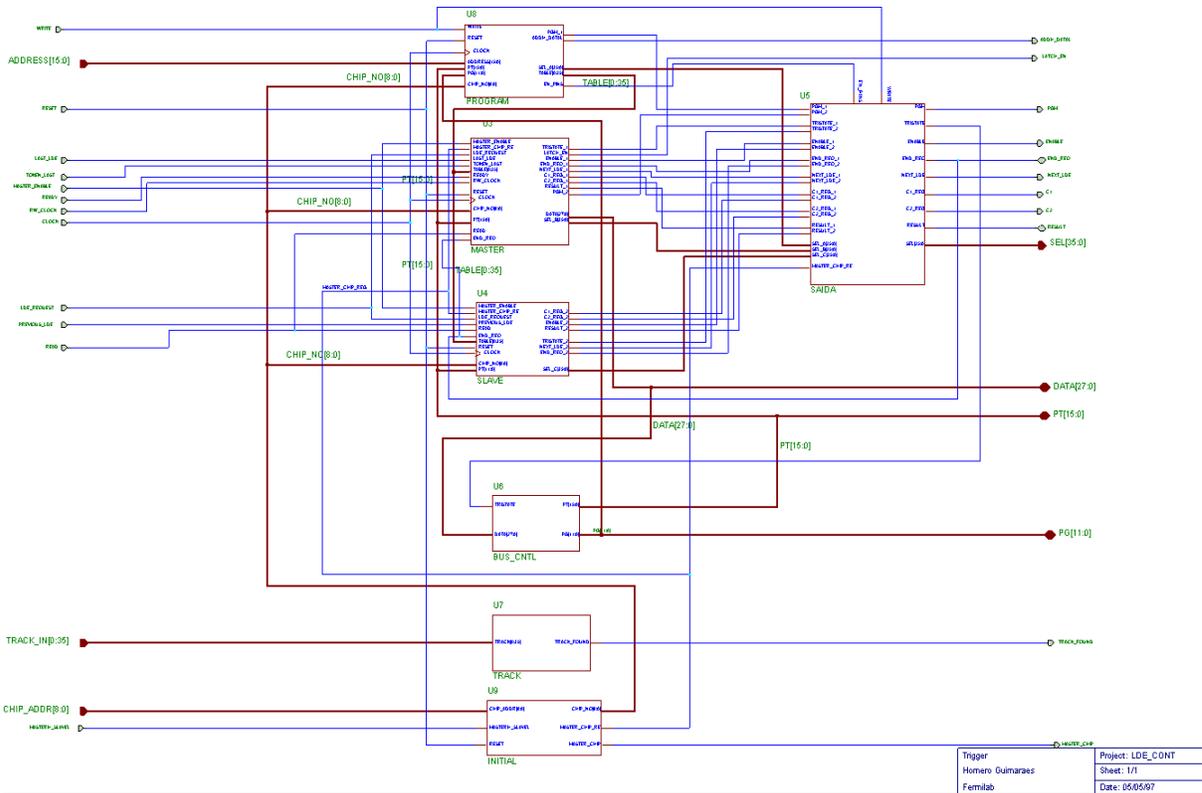


Figura 4.1.1 –Esquemático de topo do LDE_CONTROLLER

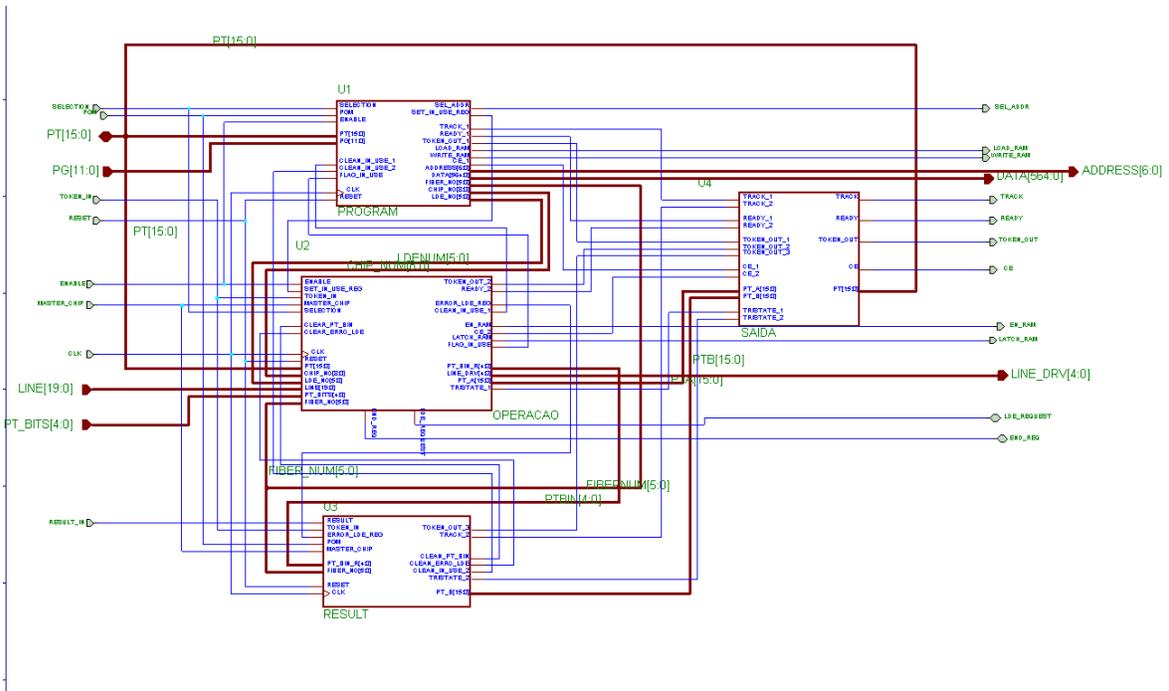
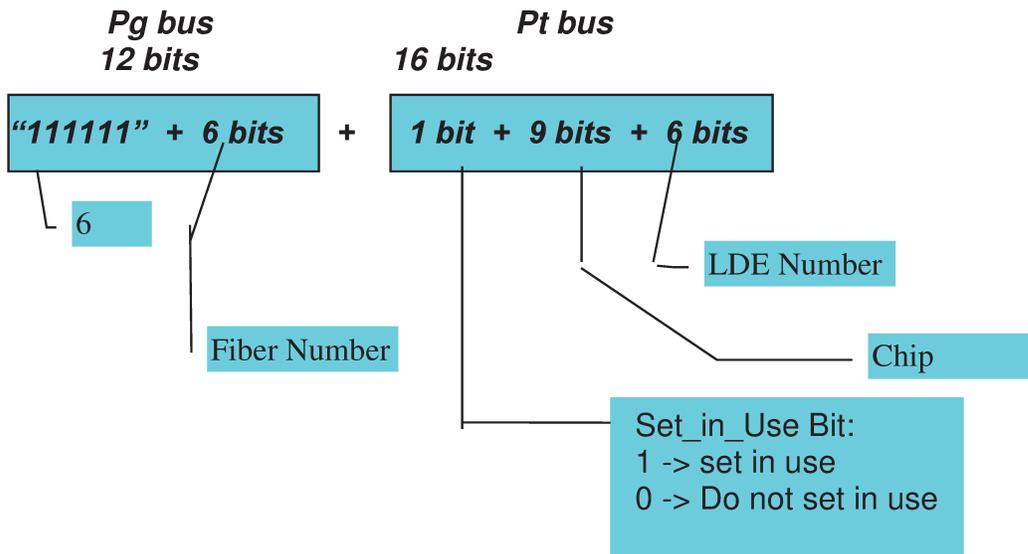


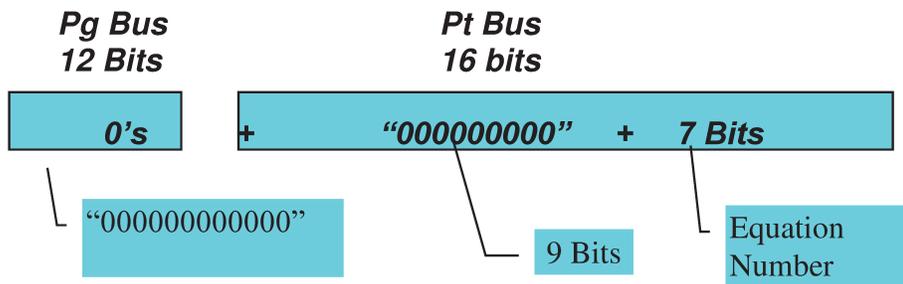
Figura 4.1.2 –Esquemático de topo do LDE

Para que possamos entender os códigos VHDL a seguir vamos inicialmente definir todas as mensagens trocadas entre os vários blocos:

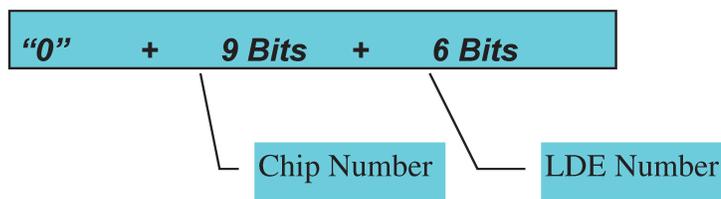
1) Mensagem SET_IN_USE: (Barramentos Pg + Pt)



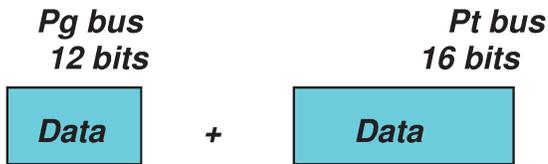
2) Programando o endereço da equação : (Barramento Pg + Pt)



3) Endereço do LDE (Host coloca palavra no barramento de endereços com 16-Bits):

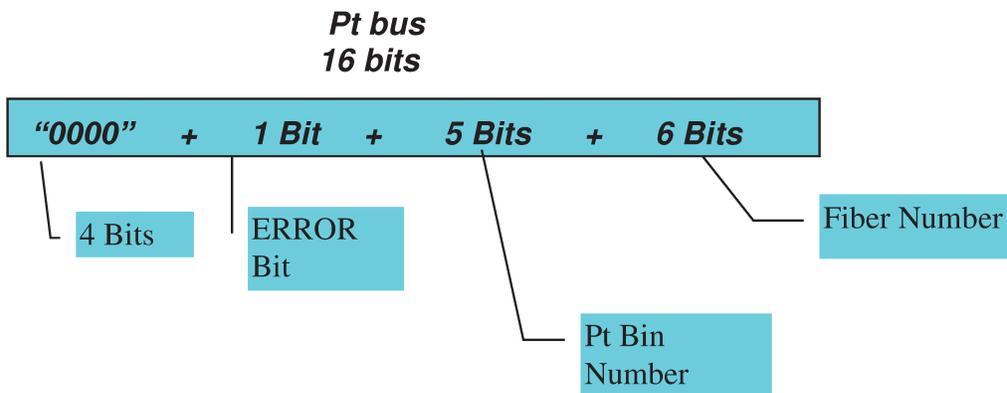


4) Dados a serem escritos na RAM durante a programação inicial (Barramentos Pg + Pt):

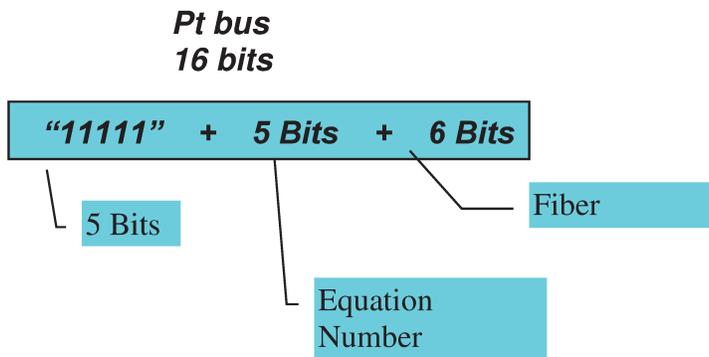


Nota: estas dados são baixados diretamente do host

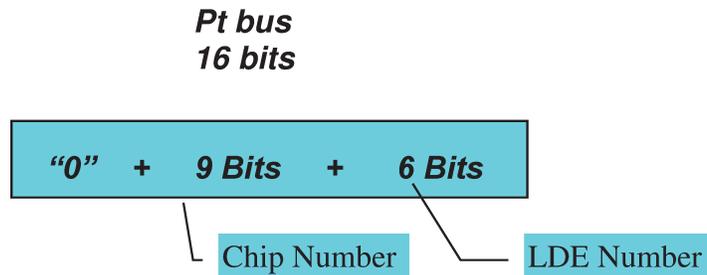
5) Reportando resultados (Barramento Pt):



6) Requerer extra LDE (Barramento Pt):



7) LDE Numero X está livre ! (Barramento Pt):



Importante :A maior diferença entre as duas últimas mensagens está no MSB!

Tendo em mente as mensagens acima, o código VHDL de cada um dos blocos está apresentado no apêndice B (CD em anexo).

Três VHDL Packages também foram criados:

1. Pack.vhd: Necessário para a síntese do código do LDE Controller. Esse package define as estruturas (types) de dados que são usado em partes do código.
2. Pack_log.vhd: Necessário para a síntese do código usado na lógica do LDE, este Package define funções para a conversão do formato inteiro (integer) para vector e vice-versa.
3. Pack.vhd: Mesmo nome que o primeiro package mas definido no diretório da lógica do LDE. Define alguns componentes usados na lógica tais como Multiplexers, tri-state drivers e flip-flops.

4.2- Circuitos de teste e Resultados das Simulações digitais

Vários circuitos de teste (test benches) foram criados para testar o código. Estes circuitos de teste nos permitem checar se todas as linhas de comunicação entre os vários elementos se comportam de acordo com o protocolo.

- O arquivo testbench_lde_cont testa o sistema considerando que o LDE controller está configurado como Mestre.
- O arquivo testbench_lde_cont_slave testa o sistema considerando que o LDE controller está configurado como escravo.

Uma leitura detalhada do código mostra que em ambos os casos os passos são semelhantes.

- Inicialmente o sistema é resetado e programado como master (ou slave) e tem o seu endereço programado (neste caso o endereço 10 decimal)
- A seguir cada LDE é programado com equações-exemplo
- A resposta a um requerimento para um LDE extra é testada
- A resposta a uma mensagem “LDE está vago” é testada.
- As duas respostas acima são testadas a cada um dos três passos do algoritmo.
- Ao fim do terceiro passo todos os LDEs reportam os seus resultados.
- O Host lê os resultados do LDE Controller.

Usamos o simulador VSIM (Modelsim, versão Windows) para compilar e rodar os test benches e checar o funcionamento do circuito.

Os primeiros blocos a serem testados serão os LDEs. Os testes serão bem simples: testbench2_lde checará o modo mestre e testbench 3_lde testará o modo slave. Em ambos os casos vamos programar as 20 posições de memória com as trajetórias desejadas e em seguida fazemos o controlador avançar os 3 passos definidos via (C1,C2).

As formas de onda são as mesmas para os dois casos.

A figura 4.2.1 mostra a etapa de programação de um LDE com uma equação (neste caso a equação tem coeficientes todos iguais a “1” para todas as 20 posições de memória):

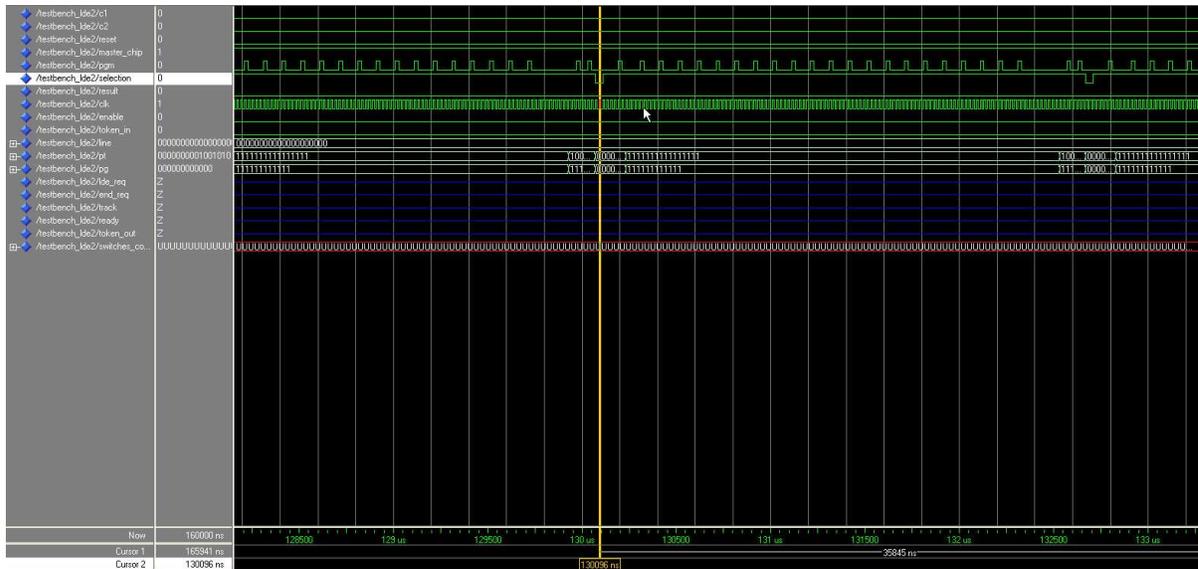


Figura 4.2.1 – Programação de um LDE

Após carregarmos as memórias com os coeficientes simulamos a movimentação do LDE por cada um dos 3 passos , conforme indicado pelos bits C1,C2 na figura 4.2.2

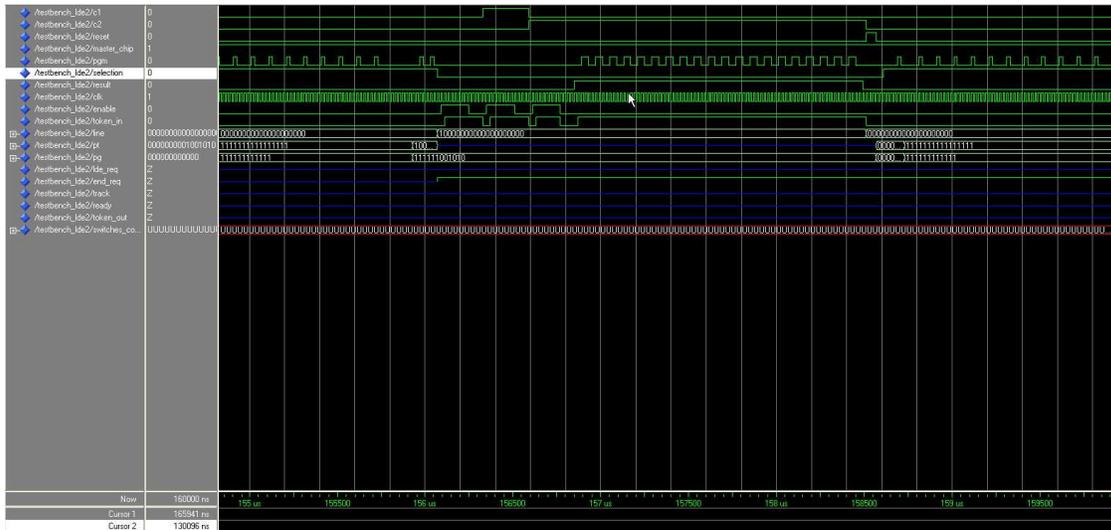
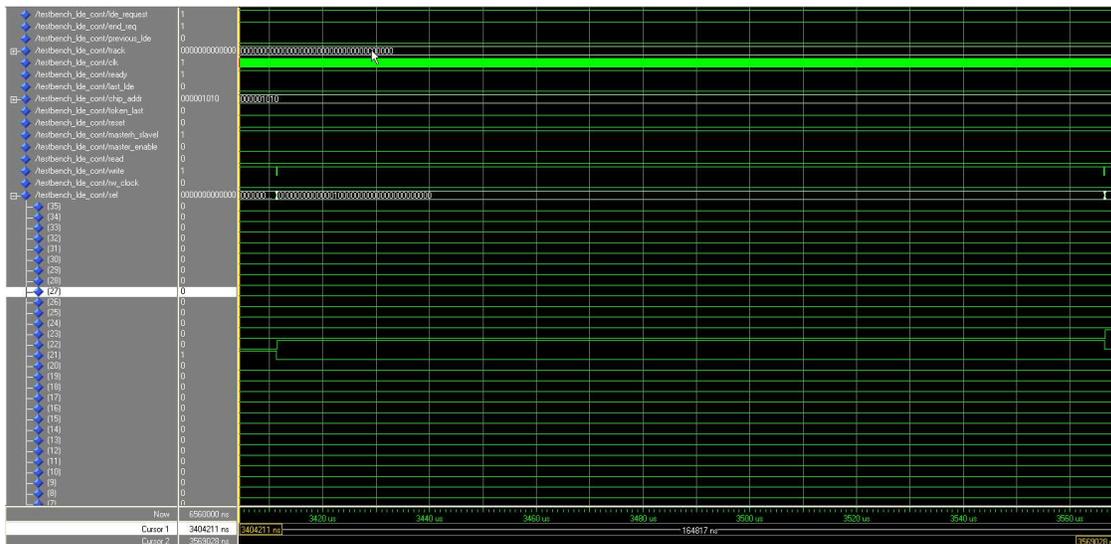


Figura 4.2.2 - Operação do LDE

A seguir rodamos o arquivo contendo o testbench da entidade testbench_lde_cont. Este testbench irá checar a operação do Lde_controller durante a etapa de programação dos LDEs.

Inicialmente programamos o Lde_controller como um mestre (master) e observamos a sua resposta a um requerimento de um LDE extra. A seguir o testbench testa a reação do LDE a uma mensagem do tipo “LDE is free”. A seguir o testbench avança para o passo 2 do algoritmo e em seguida passa para o terceiro passo em cada passo aplicamos os mesmos testes descritos acima para testar a logica. Após o terceiro passo testamos o algoritmo de ordenação de resultados e o algoritmo de saída do LDE_controller simulando os LDEs achando 3 trilhas.

As figura 4.2.3 e 4.2.4 mostram o reset inicial do LDE_controller e a programação dos LDEs em cada uma das 20 equações do passo 1.



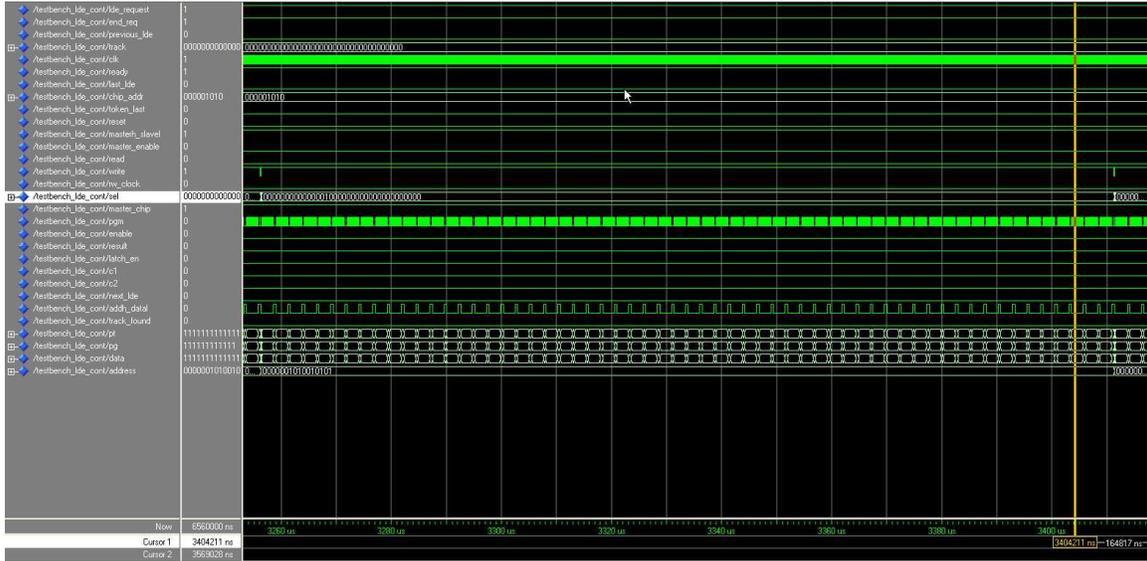


Figura 4.2.4 – Programação de um LDE

E os 36 LDEs são programados em sequencia conforme mostrado na figura 4.2.5:

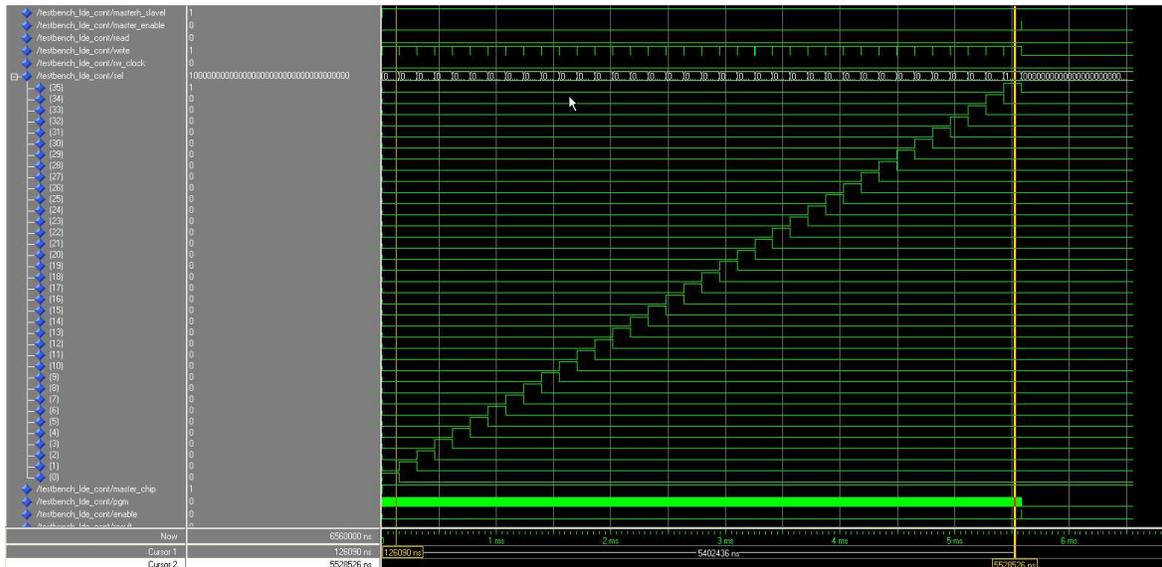


Figura 4.2.5 – Programação dos 36 LDEs de um chip

Na figura 4.2.6 é mostrada a resposta a uma mensagem de requerimento de um LDE extra seguida pela resposta a um requerimento de liberação de um LDE:

A linha next_ide é ativada porque não há um LDE para responder ao requerimento e este eh passado adiante. Isto acontece porque marcamos todas as equações como “em uso”

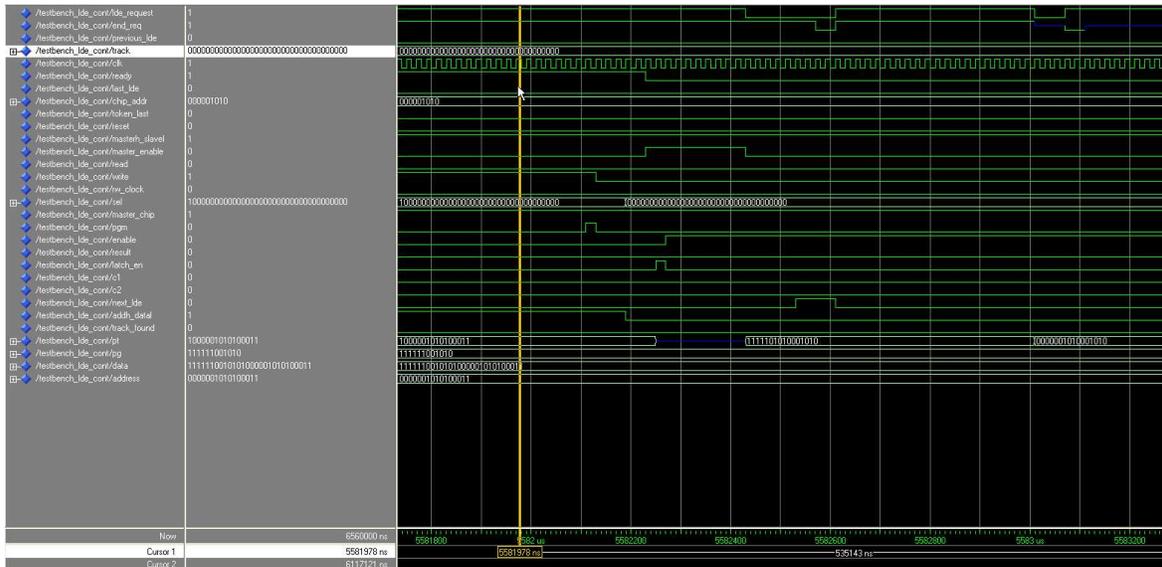


Figura 4.2.6 – Resposta a um requerimento de um LDE extra

A seguir simulamos o LDE controller no modo slave com resultados similares. Neste caso mostramos a resposta (figura 4.2.7) a um evento “Ide_request” e “Ide_free” repetidas 3 vezes. Neste caso as equações requeridas são achadas, pois nos preenchemos a tabela worktable_s com esta equação e as marcamos como não em uso:

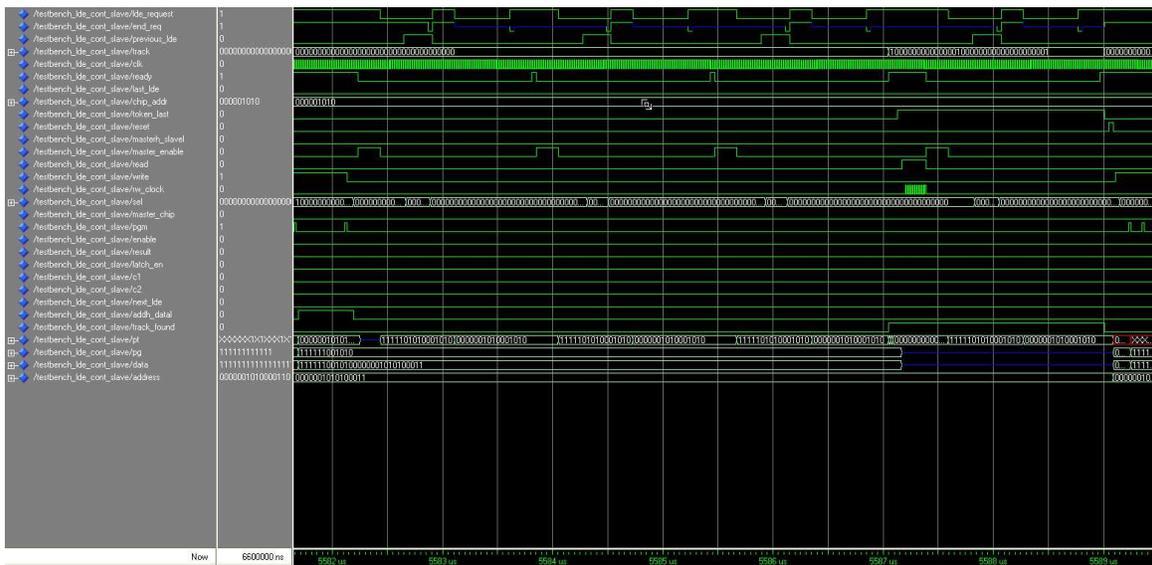


Figura 4.2.7 – Resposta a um requerimento de um LDE extra no modo escravo

A seguir finalmente simulamos o sistema completo com dois chips, cada um dos chips contendo um LDE controller controlando 2 LDEs (usamos somente 2 LDEs ao invés dos 36 LDEs usados nas simulações anteriores para tornar esta simulação possível em termos de recursos computacionais disponíveis). Um dos chips é programado para ser o mestre e o outro no modo slave.

A forma de onda mostrada na figura 4.2.8 mostra a operação normal (conforme indicado pelos bits C1, C2) seguida pela indicação de trilha achada e a leitura pelo host das trilhas achadas.

Após isto se segue um reset e uma reprogramação completa.

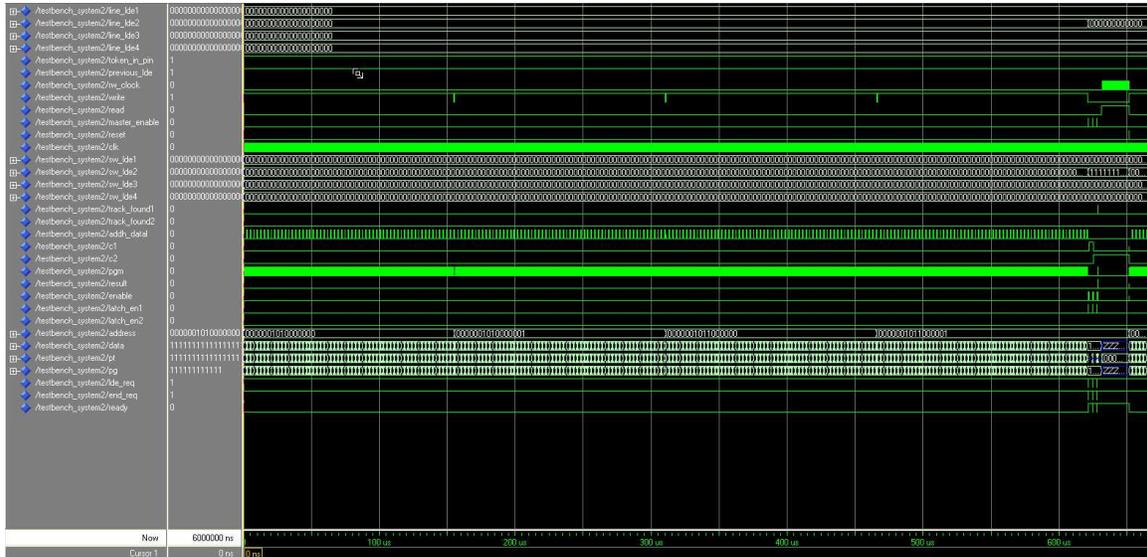


Figura 4.2.8 – Operação normal com dois chips

Na Figura 4.2.9 está mostrada a etapa de programação do LDEs. As formas de onda são similares ao caso anterior porém neste caso uma vez que temos os LDEs e o lde_controller no mesmo testbench os dados são escritos em memória e podemos testar a sincronização da rede.

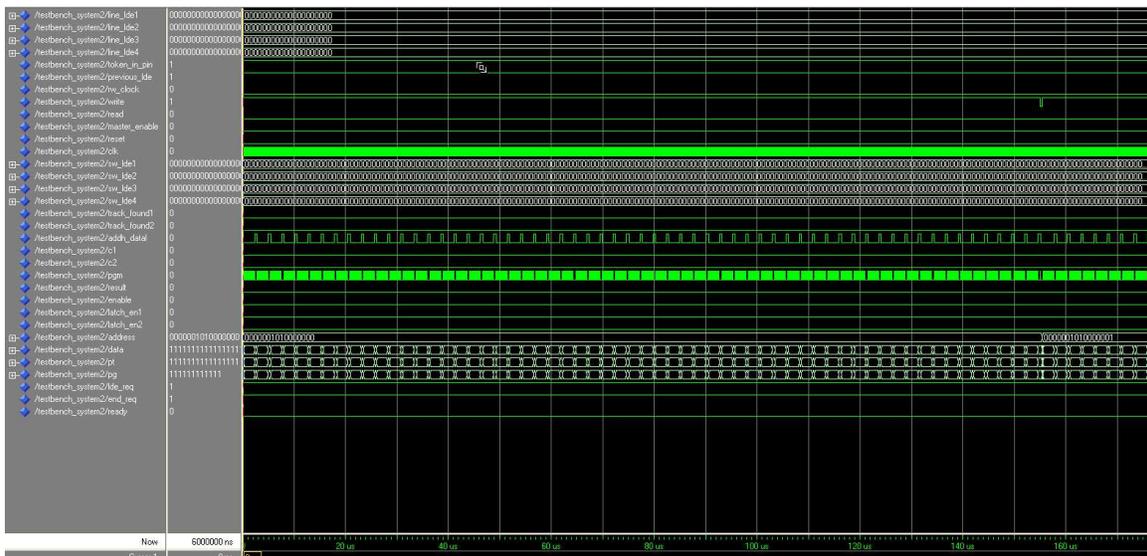


Figura 4.2.9 – Programação dos LDEs com dois chips

Na figura 4.2.10 temos a resposta do sistema a uma requisição de uma LDE para a fibra 10 (que é rapidamente atendida). Isto é repetido mais duas vezes, uma vez a cada passo conforme indicado pelos bits C1 e C2.

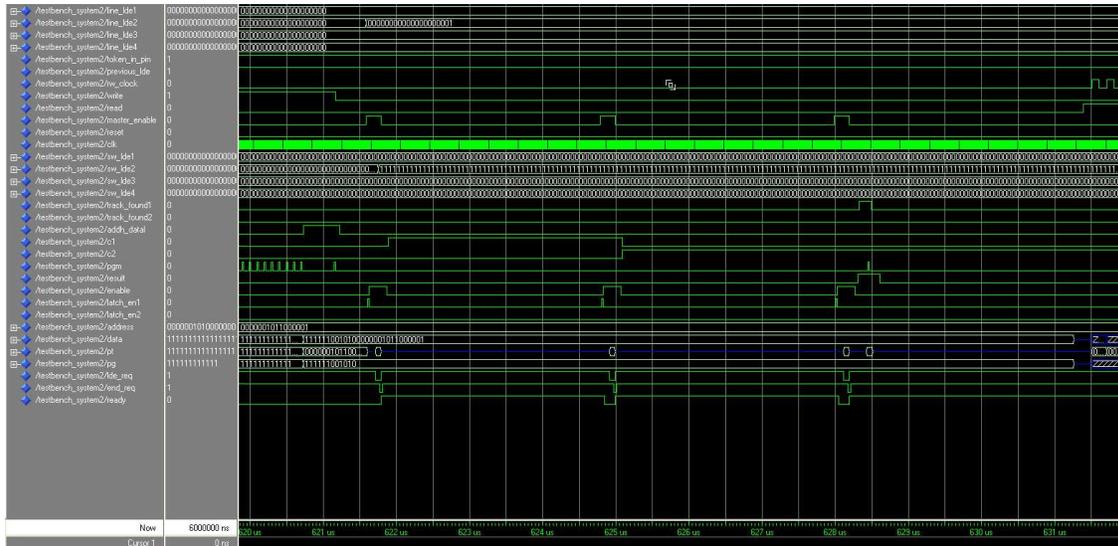


Figura 4.2.10 – Resposta a requisições de LDEs extras com dois chips

Na figura 4.2.11 temos a etapa de leitura de trilhas achadas pelo sistema. Neste testbench só temos uma trilha a qual é reportada através do barramento Pt. O valor reportado corresponde ao momento da partícula associada à trilha.

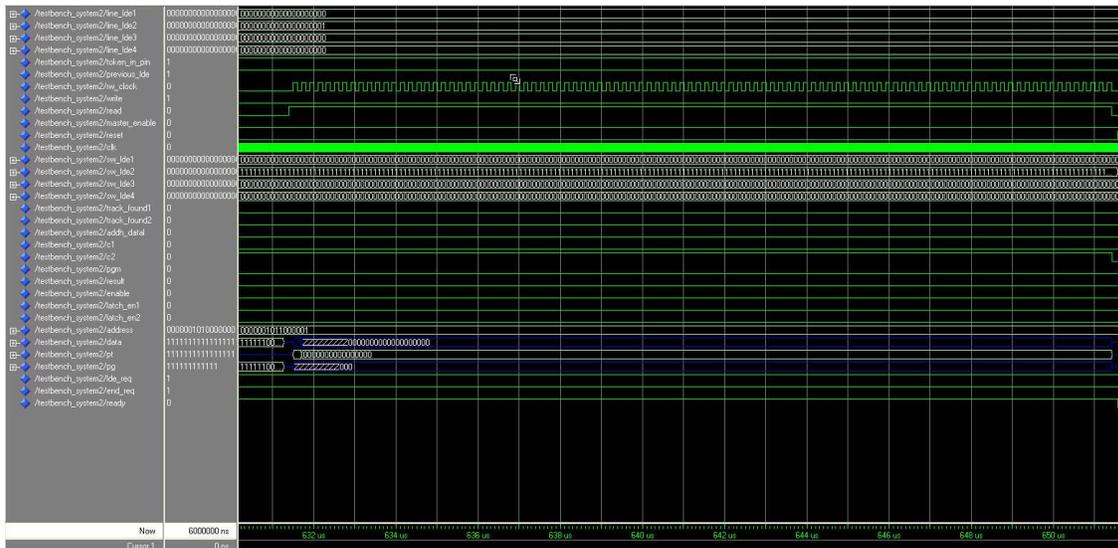


Figura 4.2.11 – Resposta a requisições de leitura com dois chips

A Figura 4.2.12 mostra a ampliação da forma de onda aonde podemos ver a única trilha sendo reportada:

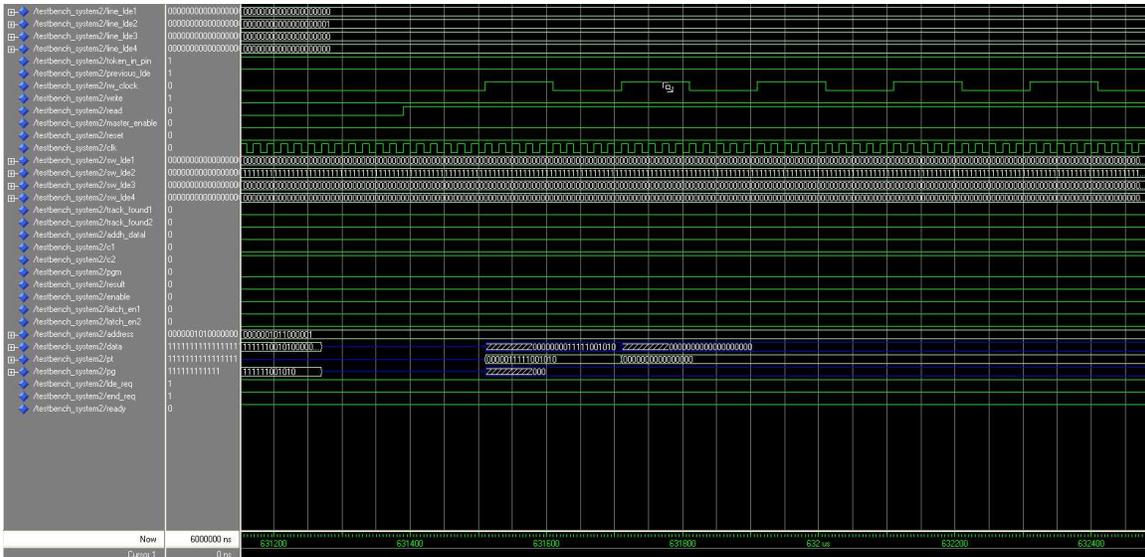


Figura 4.2.12 – Resposta a requisições de leitura com dois chips – foco numa única trilha

O testbench testbench_system usa o mesmo esquema de dois chips que o teste anterior porém ele testa a condição de resposta à mensagem de liberação de LDE. Na figura 4.2.13 mostramos a resposta completa do sistema aos vetores de teste:

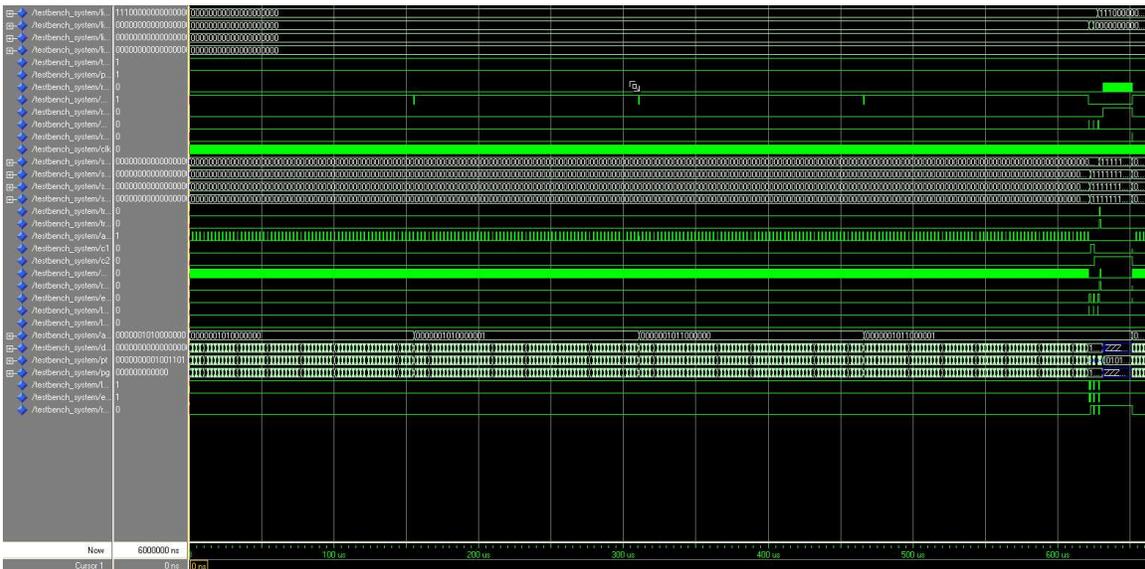


Figura 4.2.13 – Resposta Completa

A etapa de programação é a mesma que a mostrada anteriormente, assim iremos focar no modo de operação e as respostas do sistema as mensagens. A Figura 4.2.14 mostra as etapas de operação e leitura dos resultados:

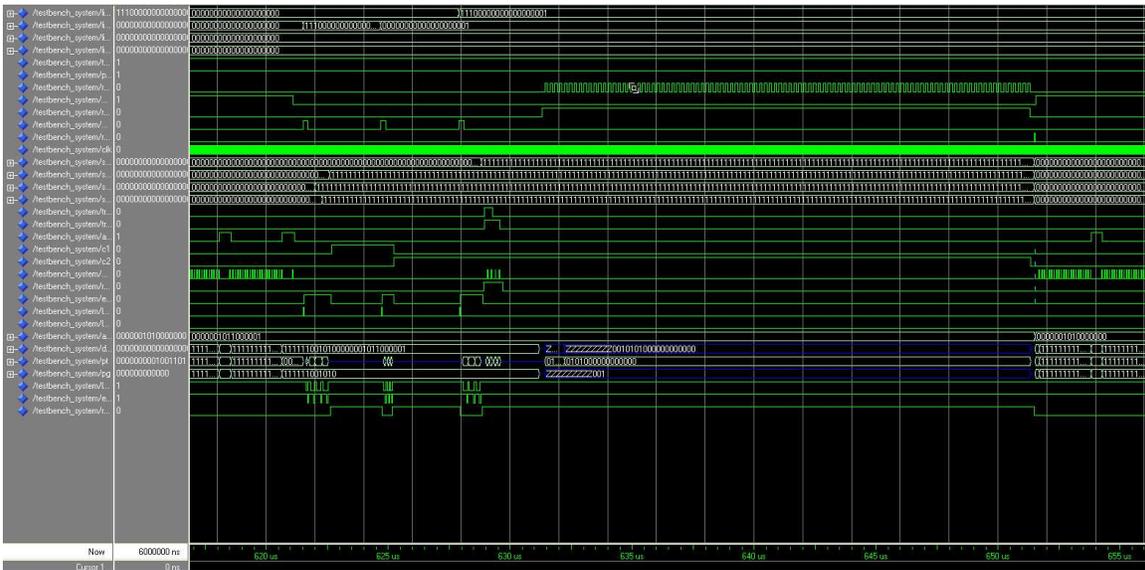


Figura 4.2.14 – Etapas de operação e Leitura dos resultados

Conforme vemos na figura 4.2.15 no passo 1 (bits C1,C2=”00”), os vetores testam a resposta do sistema a uma mensagem do tipo “LDE is free” seguida de três mensagens para alocar um LDE extra as quais são respondidas afirmativamente.

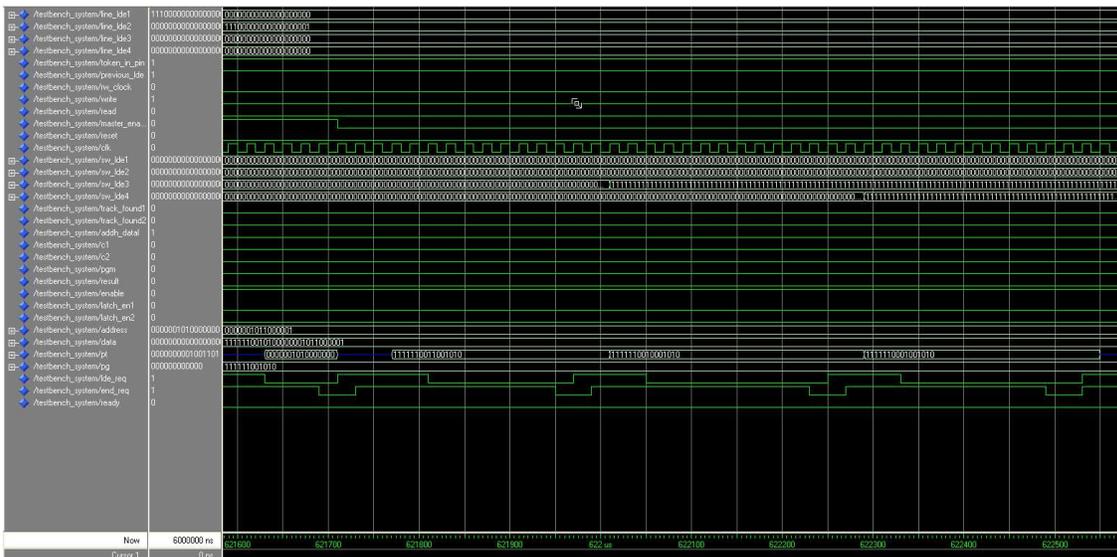


Figura 4.2.15 – Alocação de 3 LDEs extras durante passo 1

No segundo passo liberamos os 3 LDEs que alocamos no passo 1. Isto é mostrado na figura 4.2.16.

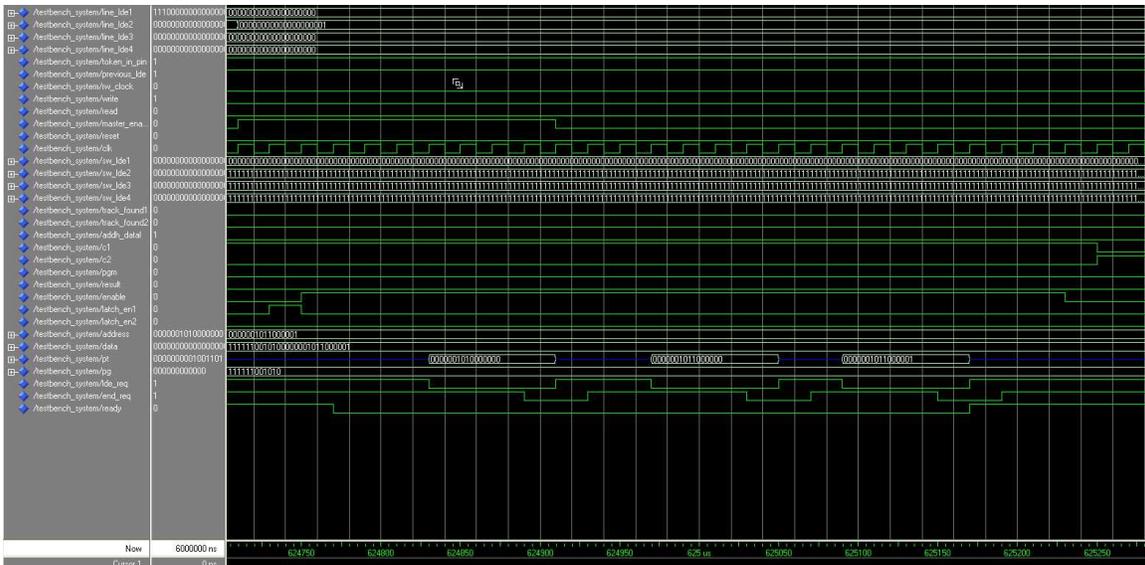


Figura 4.2.16 – Liberação dos 3 LDEs extras durante passo 1

No Passo 3 ,nos testamos a resposta a três mensagens de alocação de adicionais LDEs , conforme mostrado na figura 4.2.17.

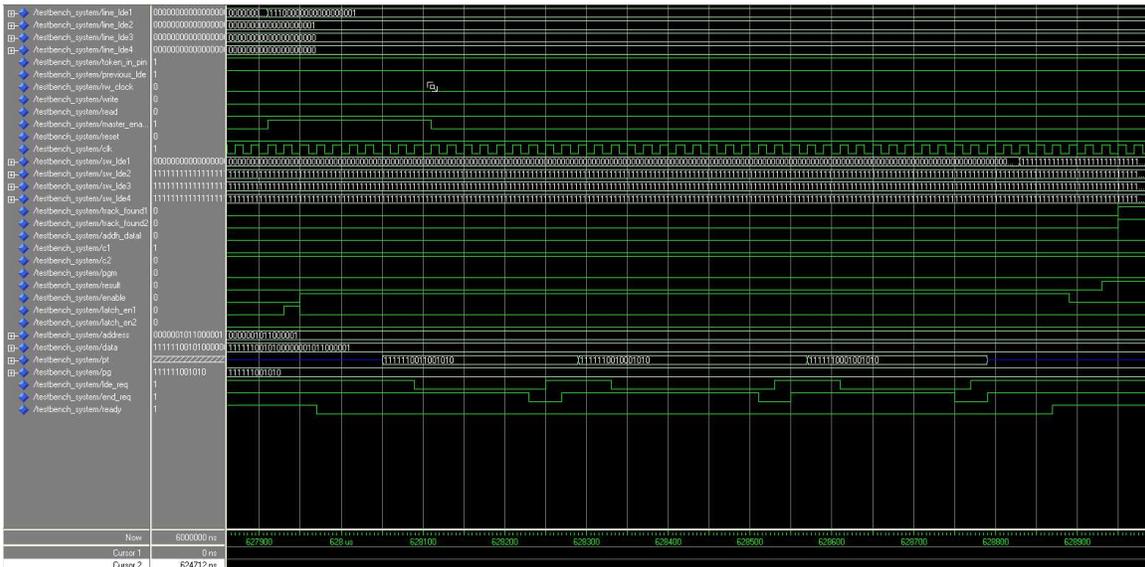


Figura 4.2.17 – Alocação de 3 LDEs extras no passo3

E finalmente temos a saída da trilha encontrada conforme mostrado na figura 4.2.18:

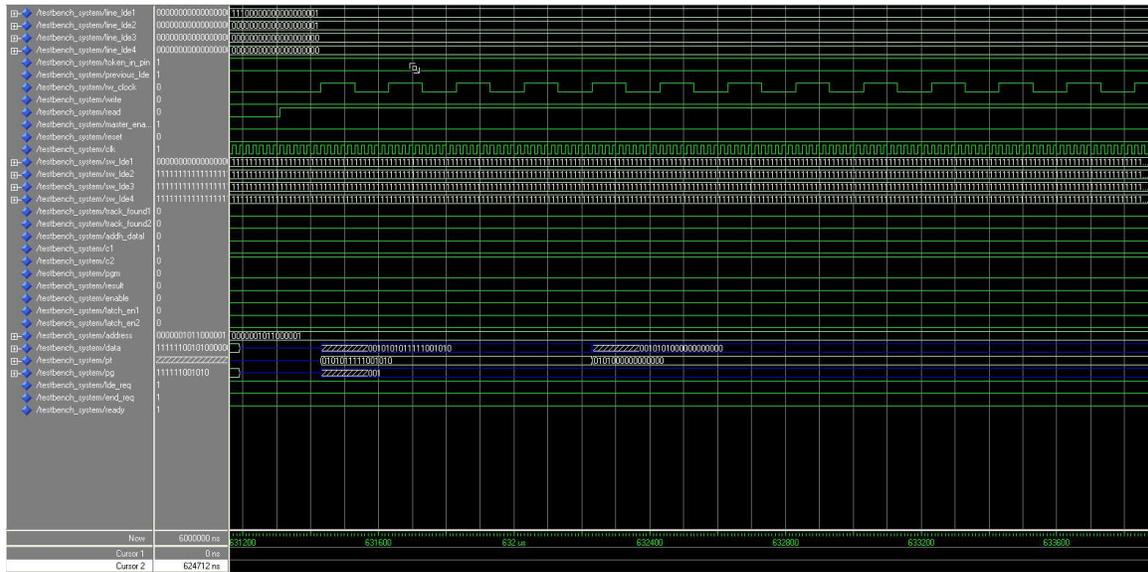


Figura 4.2.18 – Leitura das trilhas encontradas

Capítulo 5 – Circuitos Analógicos

5.1-Comparador modo corrente

A ideia básica do comparador modo corrente foi apresentada no capítulo 3 e nesta seção apresentaremos mais detalhes da sua implementação usando o processo IBM CMOS90. Este processo foi usado uma vez que é um processo estável e com dispositivos com uma frequência de transição apropriada para a aplicação (140 GHz) conforme reportado em (17).

O esquema básico é apresentado novamente como referência na figura 5.1.1 abaixo:

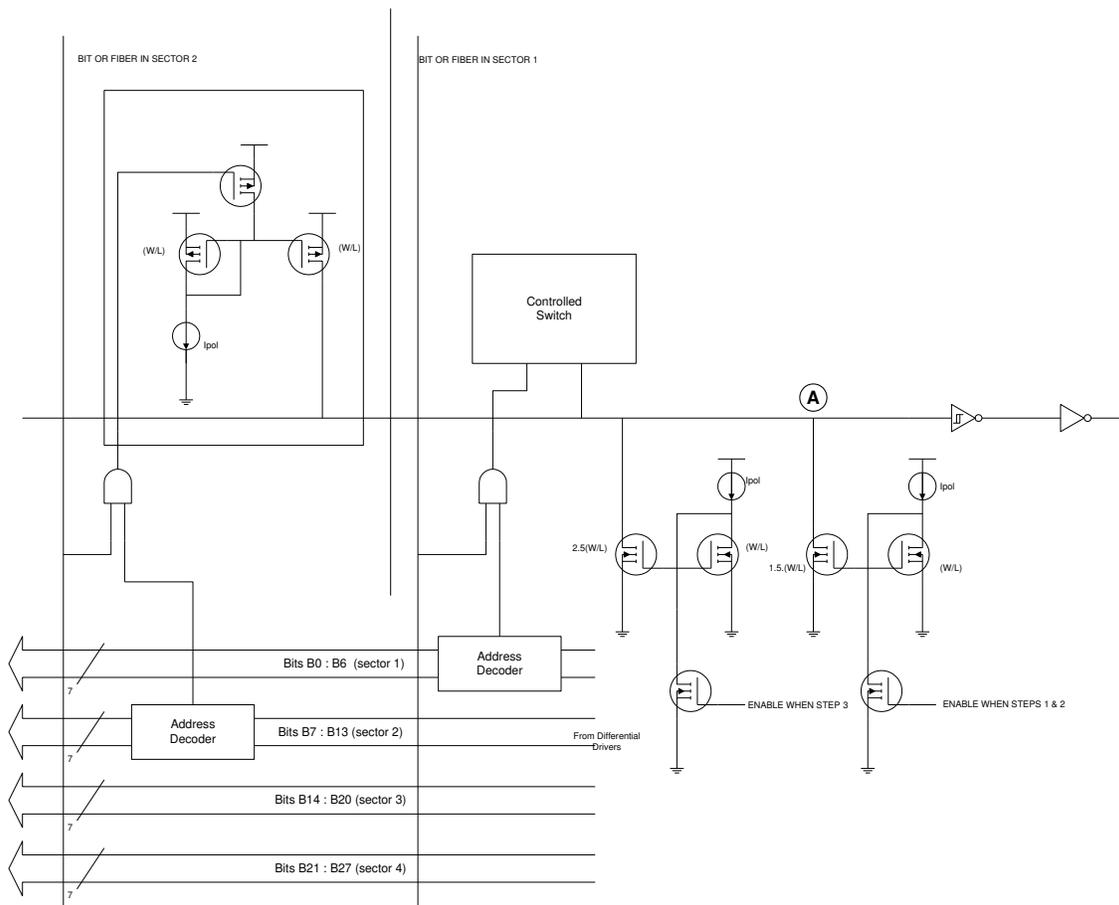


Figura 5.1.1 – Modelo do comparador modo corrente

Neste caso os comparadores operam como portas AND de alta velocidade com um enorme número de entradas validando ou não todos os possíveis caminhos. Conforme explicado anteriormente nos passos 1 e 2 a comparação é feita com dois termos $A \times B$ enquanto no terceiro passo fazemos uma comparação do tipo $(A+B) \times C \times D$ para termos redundância nas fibras mais interiores devido aos danos provenientes da radiação. Esta

comparação é feita em modo corrente no nó A. O bloco “controlled switch” será repetido 288 vezes com cada instância sendo responsável por uma fibra amostrada a cada passo. O bloco “decodificador de endereço” é responsável pela seleção de uma dada fibra dentro de determinado setor. O projeto dos decodificadores de endereço e o inversor com histerese são triviais (18) e estes blocos são amplamente disponíveis nas bibliotecas de células (como exemplo podemos citar a biblioteca da Dolphin Technologies (19)), desta forma não serão descritos em maiores detalhes. O esquemático Cadence do comparador de corrente usando o processo CMOS90 eh mostrado na figura 5.1.2:

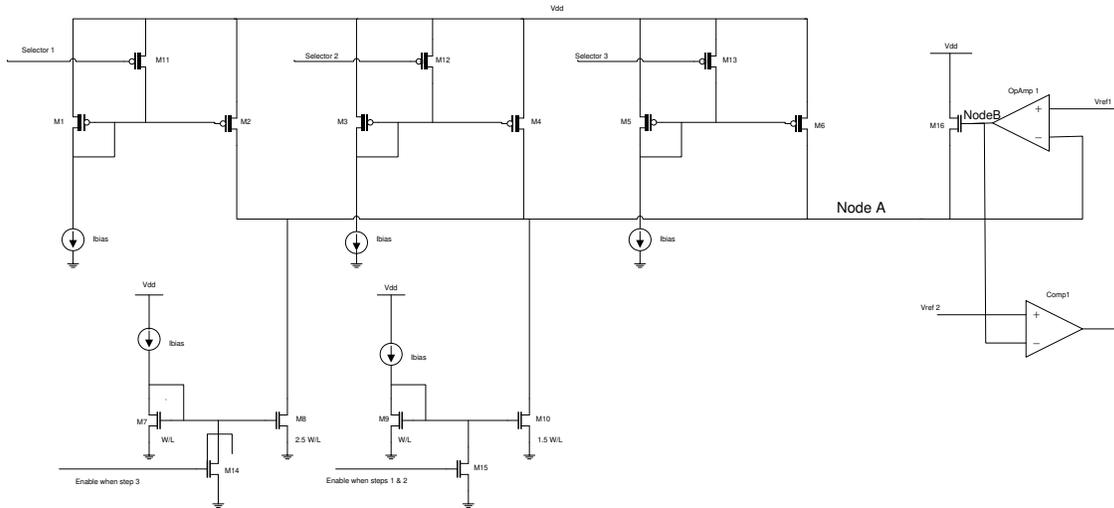


Figura 5.1.2- Esquemático do comparador modo corrente

Neste esquemático temos espelhos de corrente baseados em PMOS refletindo 100 μ A de corrente dependendo do estado das fibras (que controlam as chaves PMOS que desabilitam ou não os espelhos de corrente).

Na parte inferior temos os espelhos de corrente que selecionam em qual passo (1, 2 ou 3) o processador está operando, controlando o quanto de corrente eles absorvem do nó onde a soma de corrente ocorre (node A). O Amplificador operacional a direita opera num laço de realimentação de forma a manter o nó A no potencial de 0.6V ($V_{ref1} = V_{ref2} = 0.6$ V).

M1-M6 implementam os espelhos de corrente PMOS que são ativados seletivamente pelos decodificadores de endereço como mostrado na Figura 5.1.1. No nosso caso, a saída lógica de baixo nível (“0”) dos decodificadores irão fazer conduzir os switches M11-M13 que por sua vez garantem $V_{gs}=0$ V nos espelhos PMOS incapacitando a injeção da corrente correspondente no nó A.

M7-M10 implementam os espelhos NMOS de corrente e que são igualmente habilitados pelos transistores M14-M15 operando como chaves. Esses parâmetros são controlados pela lógica que seleciona em que passo do algoritmo que o chip está operando.

Um amplificador operacional (OpAmp1) é usado para manter a tensão no nó A constante e igual a V_{ref1} em todos os momentos. A tensão de controle do laço de realimentação é então utilizada como o sinal de entrada para o comparador final (comp1) a fim de gerar o nível lógico que indica se a fibra foi atingida ou não. A vantagem desta topologia é que o estágio de entrada do comparador (Comp1) não carrega o nó A, melhorando assim a velocidade de resposta de todo o circuito.

A seguir mostramos os resultados de simulações usando os dois esquemas: passos 1 e 2 (é necessário hits em 2 fibras) e no passo 2 (somente hit em 3 fibras). Na simulação mostrada na Figura 5.1.3 cada espelho é ativado em sequencia. O simulador usado foi o Cadence Spectre (20).

Passos 1 e 2 ==> degrau de 514.3 mV (Figura 5.1.3)

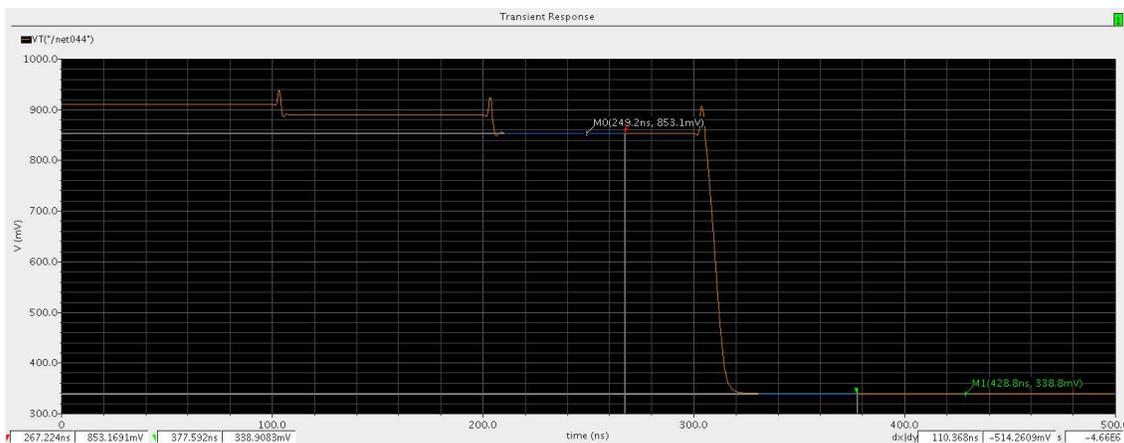


Figura 5.1.3- Resposta do comparador modo corrente passos 1 e 2

Passo 3 ==> degrau de 521.6 mV (Figura 5.1.4)

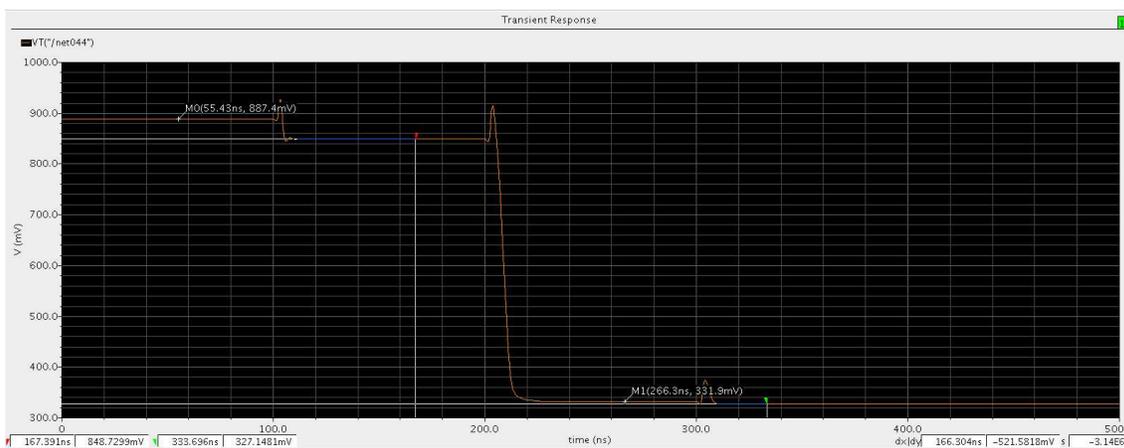


Figura 5.1.4- Resposta do comparador modo corrente passo 3

A Figura 5.1.5 mostra o esquemático final incluído o circuito de latch de saída. O bloco no canto inferior esquerdo é o circuito de polarização que será descrito adiante.

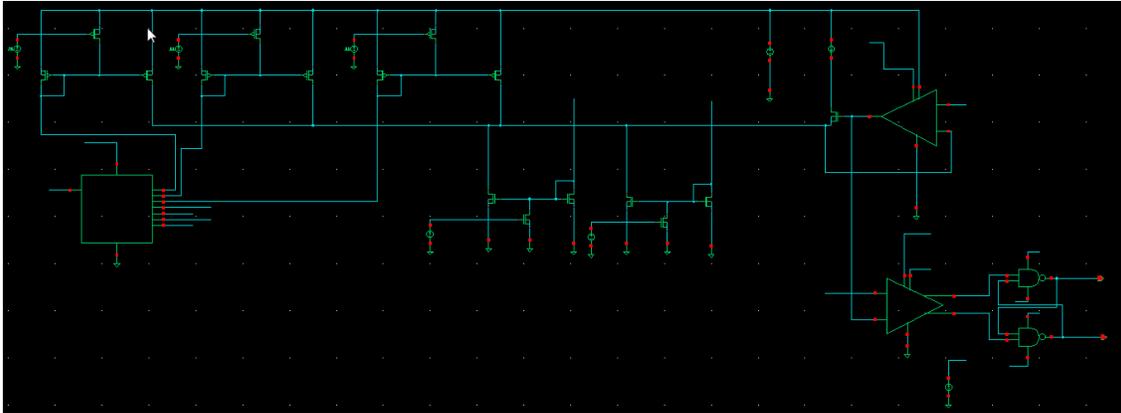


Figura 5.1.5 – Esquemático final do comparador modo corrente

O comparador usado é um comparador simples de um estágio com realimentação conforme mostrado na Figura 5.1.6:

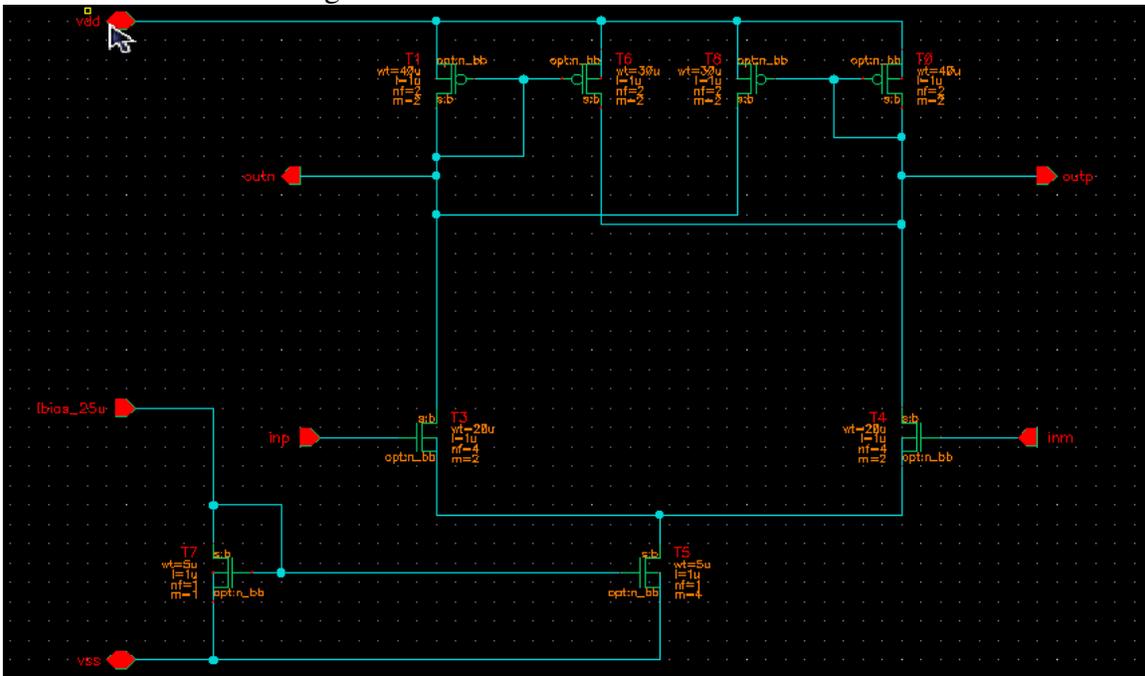


Figura 5.1.6 – Esquemático final do primeiro comparador

O esquemático da porta NAND usada e mostrado na Figura 5.1.7:

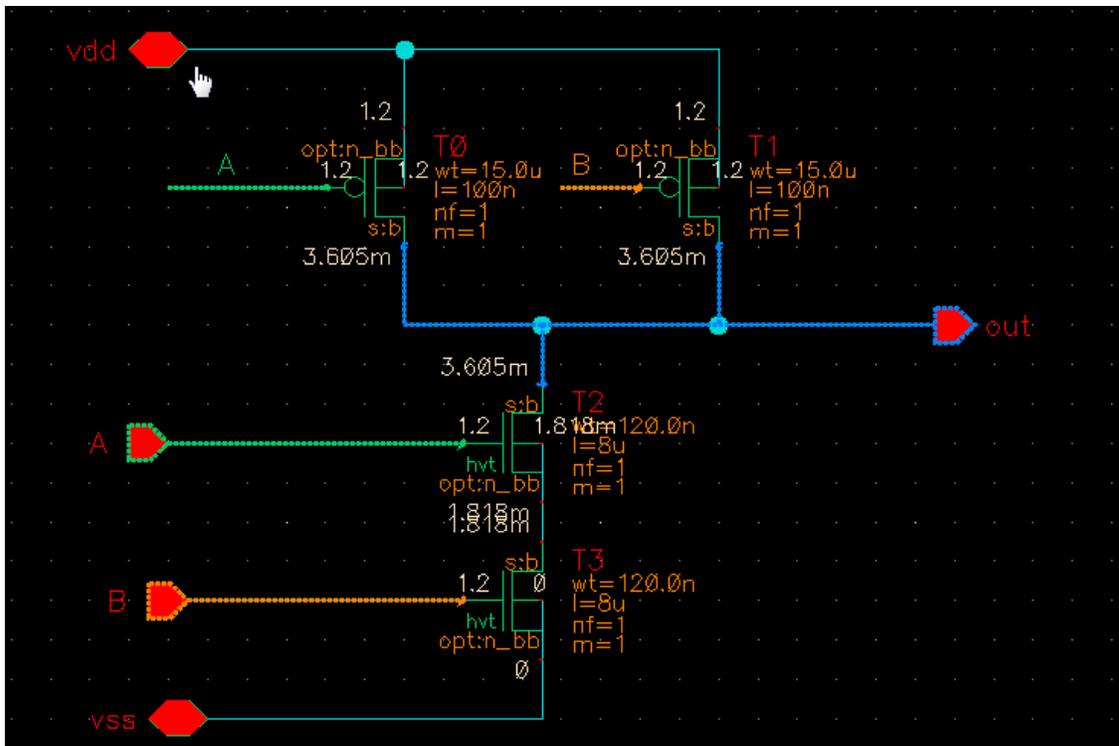


Figura 5.1.7 – Esquemático final da porta NAND

As formas de onda mostradas na Figura 5.1.8 documentam o atraso do comparador de modo corrente: o valor resultante é de 51 ns. A forma de onda no canto superior e a saída no Nó A e a forma de onda de maior amplitude representa a saída do latch:

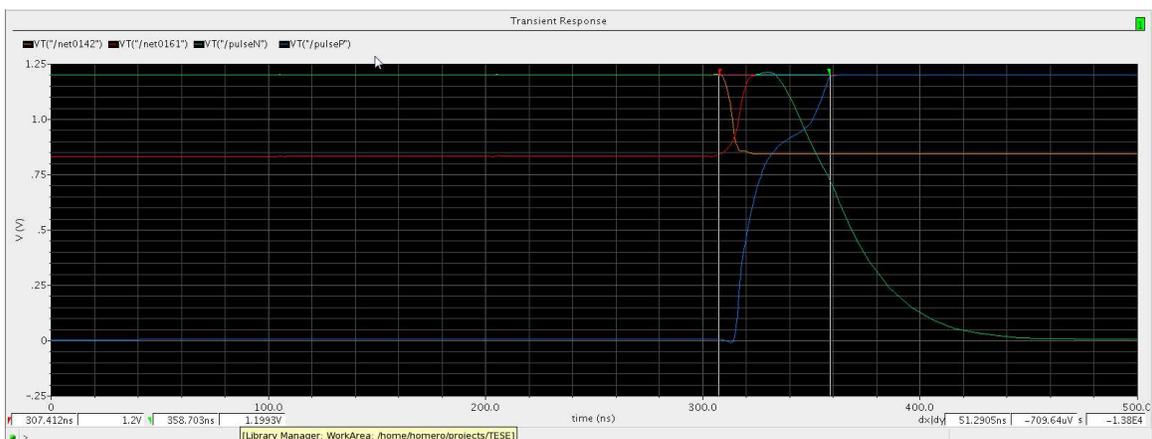


Figura 5.1.8 – Resposta do primeiro comparador

O Amplificador Operacional usado é uma versão de um só estágio classe A. A margem de fase do laço de realimentação foi verificada e é de 65 graus. O seu esquemático é mostrado na Figura 5.1.9.

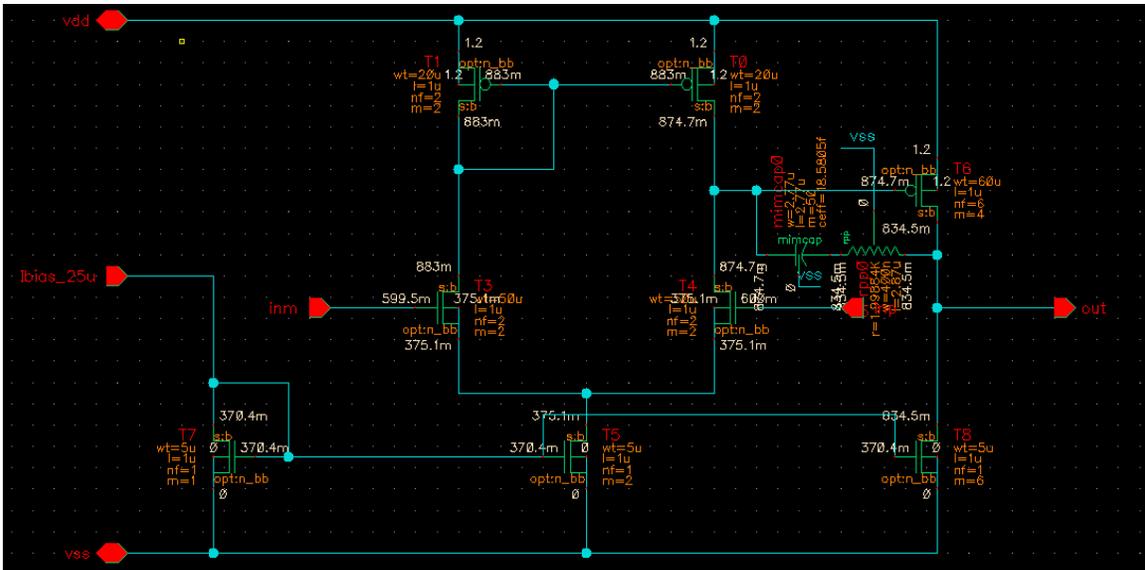


Figura 5.1.9 – Esquemático do Operacional usado no comparador de corrente.

O bloco de polarização (Figura 5.1.10) gera todas as correntes necessárias a operação do circuito. A margem de fase deste laço é de 61 graus e ele usa o mesmo amplificador operacional que o mostrado na Figura 5.1.9.

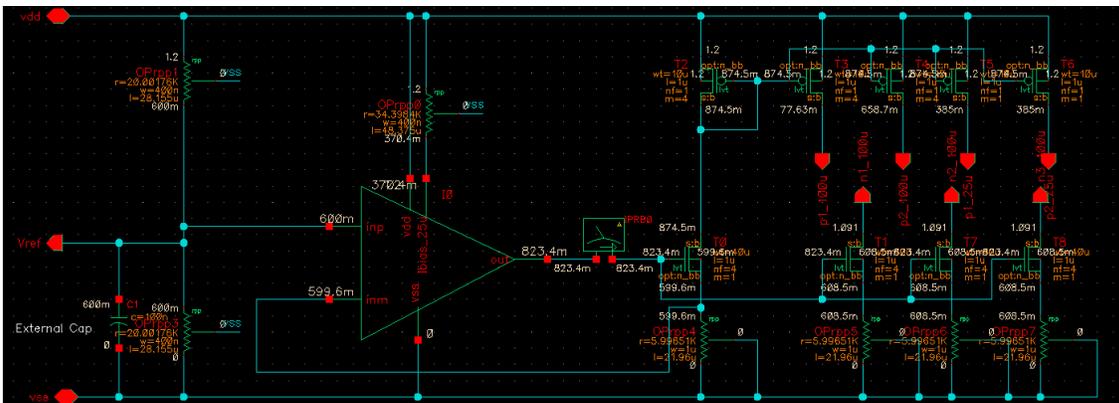


Figura 5.1.10 – Esquemático do bloco de polarização.

5.2- Projeto do front-end Analógico

O front-end analógico é o responsável pela conversão da carga recebida do sensor em um nível logico indicando se a fibra foi atingida ou não. Para isto ele se subdivide em dois blocos o pré-amplificador que essencialmente integra a carga e a transforma em tensão na sua saída seguido por um comparador que compara esta tensão com uma referência para decidir se a fibra foi atingida ou não.

No projeto deste bloco foram usadas como um ponto inicial as mesmas especificações que o C.I. SVXII projetado pelo Fermilab (10). Este CI é usado em detectores do Fermilab e em outros laboratórios ao redor do mundo.

Suas especificações básicas são:

- Período entre colisões: 132 ns
- Capacitância do detector: 10 pF a 30 pF
- Ganho: 5 mV/fC
- 1 MIP (minimum Ionizing Particle) = 5 fC
- Máxima Carga provida pelo VLPC: 90 fC
- Ruído: 1fC (1 sigma)
- Carga externa: 10 pF
- Tensão de alimentação: 2.5 V
- Ganho DC=2400 V/V (67.6 dB) com um GBW=150 MHz

A figura 5.2.1 mostra como o detector é conectado ao pré-amplificador. O detector prove um pulso negativo de carga que, ao ser integrado cria um degrau positivo na saída do pré-amplificador (que na verdade se comporta como um integrador).

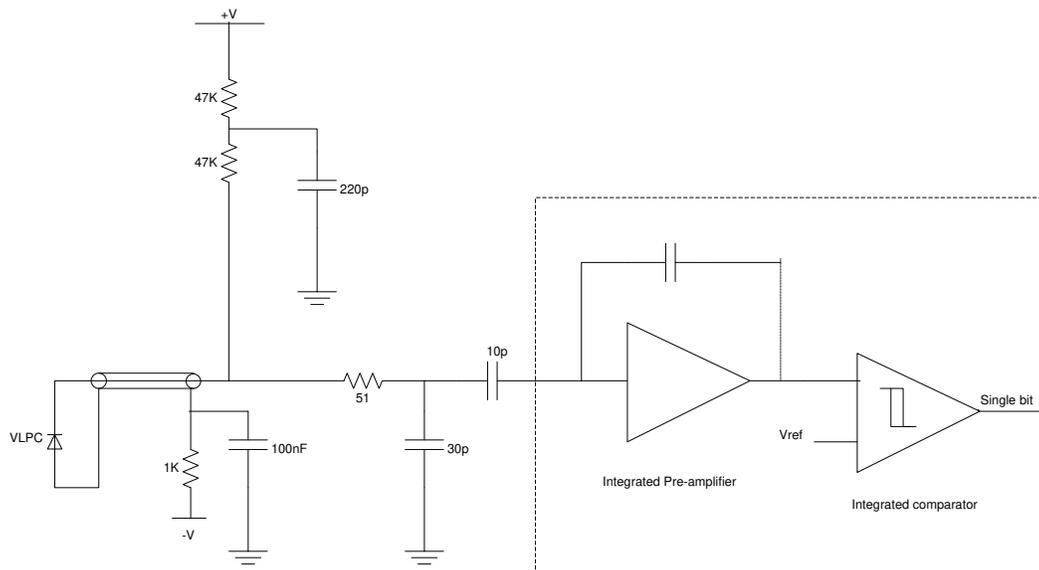


Figura 5.2.1: Front-end e sua integração com o VLPC

Para o pré-amplificador necessitaremos um amplificador de alto ganho em malha aberta e uma largura de banda elevada de forma a termos um integrador com um boa precisão. Partiremos das especificações usadas no SVXII e tentaremos obter o melhor possível. Usaremos neste trabalho o processo IBM CMOS 90nm, pois ele tem um bom compromisso velocidade versus Custo e está disponível para projetos acadêmicos via projetos multiusuário (21).

Neste projeto usaremos uma metodologia Gm/Id modificada a partir de (22) e, para obtermos o máximo ganho com a menor corrente possível, precisaremos do máximo valor de Gm/Id e do mínimo valor de Gds/Id para um dado comprimento de canal.

Para explorarmos a variação de Gm/Id e Gds/ID com o nível de inversão do canal (Vgs) e com o comprimento de canal (L) usamos o simulador Spectre em conjunto com a linguagem Skill para criarmos um script que plota estas variáveis como função de L e da tensão Vgs (porta-fonte) enquanto mantemos Vds constante e igual a 1.2V.

A variável W (largura de canal) será escalada na mesma proporção em que aumentamos L de forma a manter uma razão de 100. No circuito final iremos escalonar W de forma a obtermos a corrente que queremos.

A figura 5.2.2 mostra um exemplo de um gráfico mostrando a variação de Gm, Gm/Gds (ganho intrínseco) e Gds como função de L e Vgs:

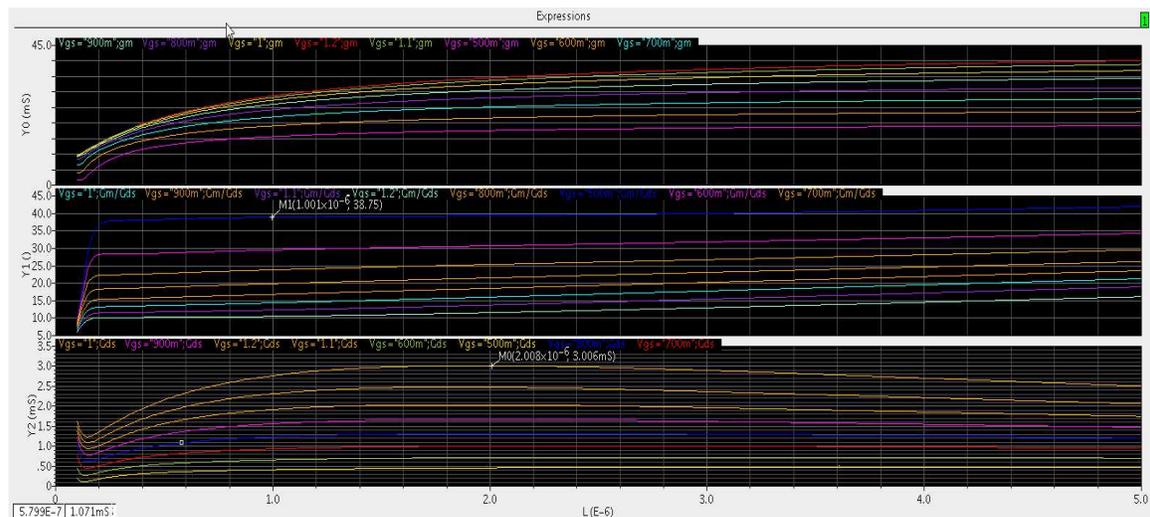


Figura 5.2.2: Caracterização de um NMOS

Estas Curvas foram extraídas para todos os dispositivos usados no circuito: os dispositivos de oxido fino (pmos e nmos) e os dispositivos de oxido grosso (dgnfet e dgpfet).

Conforme esperado da teoria quando mais Vgs aumenta, dois efeitos ocorrem:

- Gm diminui
- Gm/Gds diminui

Pois o dispositivo vai de inversão fraca ate a inversão forte com a correspondente diminuição de Gm.

Após isto as tabelas de dados são importadas em Microsoft Excel e processados para extrairmos a variação dos parâmetros Gm/Id e Gds/I versus comprimento de Canal e Vgs, conforme mostrado na figura 5.2.3:

	A	I	J	K	L	M	N	O	P	Q	
1	Vgs	6.00E-01								7.00E-01	
2	L	getData("T0:gm	getData("T0:id	Gm/Id	getData("T0:gd	Gds/Id	getData("T0:vd	getData("T0:cg	getData("T0:gm	getData("T0:id	
79	8.4480000E-07	1.8187600E-02	3.7828600E-03	4.8078967E+00	6.2410300E-04	1.6498179E-01	3.0798600E-01	-5.4034100E-13	2.0868300E-02	5.7404500E-03	
80	8.5460000E-07	1.8241600E-02	3.7971800E-03	4.8039861E+00	6.2560700E-04	1.6475569E-01	3.0894700E-01	-5.5325000E-13	2.0935100E-02	5.7608100E-03	
81	8.6440000E-07	1.8294800E-02	3.8112600E-03	4.8001973E+00	6.2708300E-04	1.6453430E-01	3.0989200E-01	-5.6631100E-13	2.1000800E-02	5.7808300E-03	
82	8.7420000E-07	1.8347100E-02	3.8251100E-03	4.7964895E+00	6.2852900E-04	1.6431658E-01	3.1082300E-01	-5.7952600E-13	2.1065600E-02	5.8005300E-03	
83	8.8400000E-07	1.8398600E-02	3.8387400E-03	4.7928747E+00	6.2994900E-04	1.6410307E-01	3.1173900E-01	-5.9289400E-13	2.1129300E-02	5.8199100E-03	
84	8.9380000E-07	1.8449200E-02	3.8521500E-03	4.7893254E+00	6.3134100E-04	1.6389315E-01	3.1264000E-01	-6.0641400E-13	2.1192000E-02	5.8389900E-03	
85	9.0360000E-07	1.8499100E-02	3.8653400E-03	4.7858921E+00	6.3270700E-04	1.6368728E-01	3.1352900E-01	-6.2008700E-13	2.1253800E-02	5.8577600E-03	
86	9.1340000E-07	1.8548100E-02	3.8783300E-03	4.7824966E+00	6.3404800E-04	1.6348480E-01	3.1440300E-01	-6.3391400E-13	2.1314700E-02	5.8762400E-03	
87	9.2320000E-07	1.8596400E-02	3.8911100E-03	4.7792018E+00	6.3536400E-04	1.6328605E-01	3.1526500E-01	-6.4789300E-13	2.1374700E-02	5.8944300E-03	
88	9.3300000E-07	1.8644000E-02	3.9037000E-03	4.7759818E+00	6.3665600E-04	1.6309040E-01	3.1611400E-01	-6.6202500E-13	2.1433700E-02	5.9123500E-03	
89	9.4280000E-07	1.8690800E-02	3.9161000E-03	4.7728097E+00	6.3792500E-04	1.6289804E-01	3.1695100E-01	-6.7631000E-13	2.1491900E-02	5.9299900E-03	
90	9.5260000E-07	1.8736900E-02	3.9283100E-03	4.7697101E+00	6.3917100E-04	1.6270890E-01	3.177500E-01	-6.9074800E-13	2.1549300E-02	5.9473700E-03	
91	9.6240000E-07	1.8782400E-02	3.9403400E-03	4.7666953E+00	6.4039400E-04	1.6252252E-01	3.1858800E-01	-7.0533900E-13	2.1605800E-02	5.9645000E-03	
92	9.7220000E-07	1.8827100E-02	3.9521900E-03	4.7637133E+00	6.4159600E-04	1.6233936E-01	3.1938900E-01	-7.2008300E-13	2.1661500E-02	5.9813700E-03	
93	9.8200000E-07	1.8871200E-02	3.9638700E-03	4.7608019E+00	6.4277600E-04	1.6215870E-01	3.2017800E-01	-7.3498000E-13	2.1716400E-02	5.9979900E-03	
94	9.9180000E-07	1.8914700E-02	3.9753800E-03	4.7579602E+00	6.4393600E-04	1.6198099E-01	3.2095700E-01	-7.5003000E-13	2.1770500E-02	6.0143800E-03	
95	1.0016000E-06	1.8957500E-02	3.9867300E-03	4.7551502E+00	6.4507500E-04	1.6180554E-01	3.2172500E-01	-7.6523300E-13	2.1823900E-02	6.0305300E-03	
96	1.0114000E-06	1.8999700E-02	3.9979100E-03	4.7524081E+00	6.4619500E-04	1.6163320E-01	3.2248200E-01	-7.8058800E-13	2.1876500E-02	6.0464500E-03	
97	1.0212000E-06	1.9041300E-02	4.0089400E-03	4.7497094E+00	6.4729600E-04	1.6146313E-01	3.2322900E-01	-7.9609700E-13	2.1928400E-02	6.0621500E-03	
98	1.0310000E-06	1.9082400E-02	4.0198100E-03	4.7470900E+00	6.4837800E-04	1.6129568E-01	3.2396600E-01	-8.1175900E-13	2.1979600E-02	6.0776300E-03	
99	1.0408000E-06	1.9122800E-02	4.0305300E-03	4.7444877E+00	6.4944100E-04	1.6113042E-01	3.2469300E-01	-8.2757400E-13	2.2030100E-02	6.0928900E-03	
100	1.0506000E-06	1.9162700E-02	4.0411100E-03	4.7419397E+00	6.5048700E-04	1.6096741E-01	3.2541000E-01	-8.4354200E-13	2.2079900E-02	6.1079500E-03	
101	1.0604000E-06	1.9202000E-02	4.0515400E-03	4.7394324E+00	6.5151500E-04	1.6080675E-01	3.2611800E-01	-8.5966300E-13	2.2129100E-02	6.1228100E-03	
102	1.0702000E-06	1.9240800E-02	4.0618400E-03	4.7369665E+00	6.5252600E-04	1.6064788E-01	3.2681700E-01	-8.7593700E-13	2.2177600E-02	6.1374600E-03	
103	1.0800000E-06	1.9279100E-02	4.0719900E-03	4.7345647E+00	6.5352000E-04	1.6049155E-01	3.2750700E-01	-8.9236400E-13	2.2225400E-02	6.1519200E-03	
104	1.0898000E-06	1.9316900E-02	4.0820100E-03	4.7322030E+00	6.5449800E-04	1.6033719E-01	3.2818700E-01	-9.0894400E-13	2.2272700E-02	6.1661900E-03	
105	1.0996000E-06	1.9354100E-02	4.0919100E-03	4.7298450E+00	6.5545900E-04	1.6018412E-01	3.2885900E-01	-9.2567800E-13	2.2319300E-02	6.1802700E-03	

Figura 5.2.3: Caracterização Gm/Id e Gds/Id como função de Vgs e L

Com isto temos todos os dados para polarizar e dimensionar os transistores MOS dos circuitos a serem projetados.

5.3-Circuito do pré-amplificador de Carga

O circuito do pré-amplificador de carga e mostrado na figura 5.3.1 abaixo em sua forma básica:

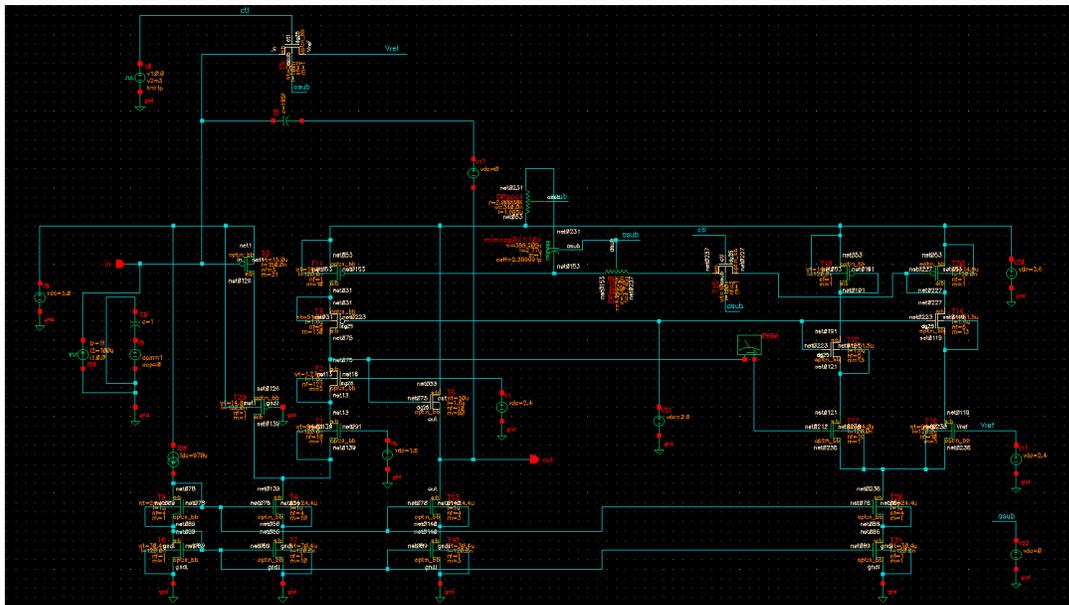


Figura 5.3.1: Circuito Pré-amplificador de Carga

Devido aos requisitos de baixo ruído, o circuito consiste de um folded Cascode (transistores T0, T29, T1-T3 e T11) cujas razões (w/l) foram dimensionadas pelo método Gm/C de modo a maximizar o ganho possível na tecnologia usada. Devido a necessidade de máximo ganho, os transistores de oxido fino foram usados onde necessitamos uma alta transcondutância (drivers) e usamos os dispositivos de oxido grosso (T3, T2) para proteger os de oxido fino uma vez que eles requerem um baixo Vds (1.2 V típico).

O ganho desta topologia pode ser estimado como:

$$Av_0 = gm_{T0} * R_{out}$$

onde $R_{out} \cong (gm_{11} * rds_{11} * rds_3) / (gm_1 * rds_1 * rds_2)$

O Transistor T0 é o transistor principal do circuito, o qual converte a tensão em corrente e também o que domina o ruído equivalente do bloco uma vez que o ruído de todos os outros dispositivos aparecem divididos por (gm_{T0}^2). De maneira apropriada ele foi dimensionado para ser largo o suficiente para termos um baixo ruído 1/f e a sua corrente é alta o suficiente para termos um alto Gm o que além de um alto ganho, reduz o ruído equivalente na entrada. Para polarizarmos de maneira apropriada este importante transistor nós usamos uma tensão de alimentação de 3V. Para o resto do circuito que é menos crítico usamos 3.6 V. Uma vantagem desta abordagem é que poderemos ter uma fonte de alimentação separada de 3V de muito baixo ruído especificamente para este bloco de forma a criar uma referência limpa e isenta de interferência de outros blocos.

O Transistor T6 implementa um seguidor de fonte usado para reduzir a impedância de saída do bloco e tornar o integrador um pouco mais próximo do ideal.

Devido à necessidade de criarmos uma tensão estável nos drenos de T2 e T1, um circuito de realimentação foi adicionado: os transistores (T18, T30, T14, T12, T17, T16, T28 e T31) implementam um amplificador que compara a tensão presente nos drenos de T2/T1 com uma voltagem de referência de 2.4V e ajusta a polarização do estágio de entrada de forma a igualar as duas. A rede RC conectada a linha de controle implementa uma compensação “lead-lag” de forma a manter o loop estável com uma margem de fase igual a 60 graus conforme mostrado na figura 5.3.2.

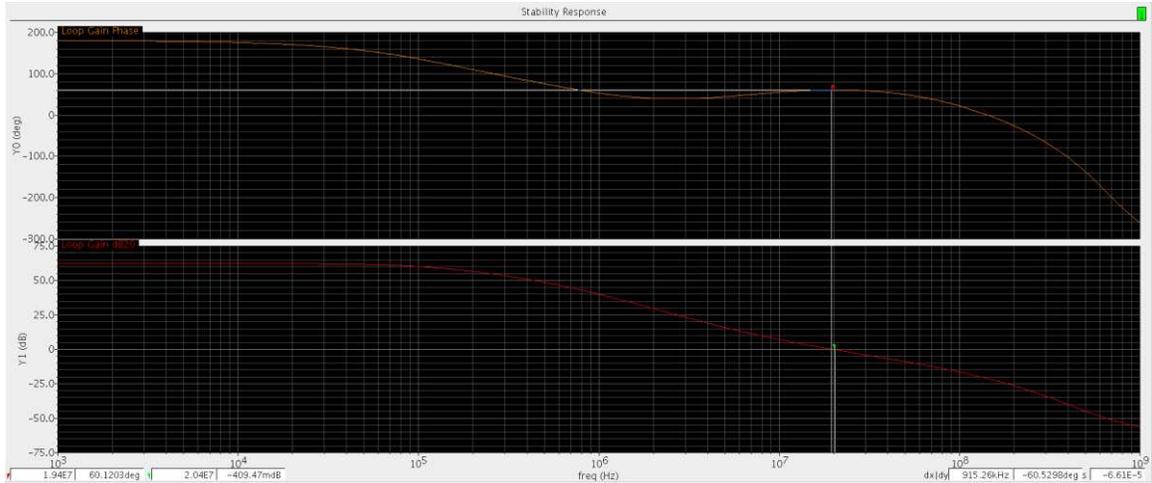


Figura 5.3.2: Margem de phase do loop

T15 é uma chave MOS que desconecta o sistema de realimentação durante o curto período de colisão onde necessitamos a máxima velocidade de operação do circuito. O capacitor mimcap0 mantém a tensão de controle estável até o fim do período de colisão (150 ns). Durante o período de reset o loop é fechado de forma a fazer uma atualização da tensão de controle

O reset é feito pela chave T5 que recarrega o capacitor de volta a tensão quiescente V_{ref} (2.4V) durante o período de reset.

Simulações AC do circuito quando em malha aberta mostram um ganho de malha aberta de 65.3 dB e um produto Ganho-banda igual a 763 MHz, o qual é melhor do que o chip SVXII usado como referência.

A Figura 5.3.3 mostra a resposta em frequência do circuito em malha aberta.

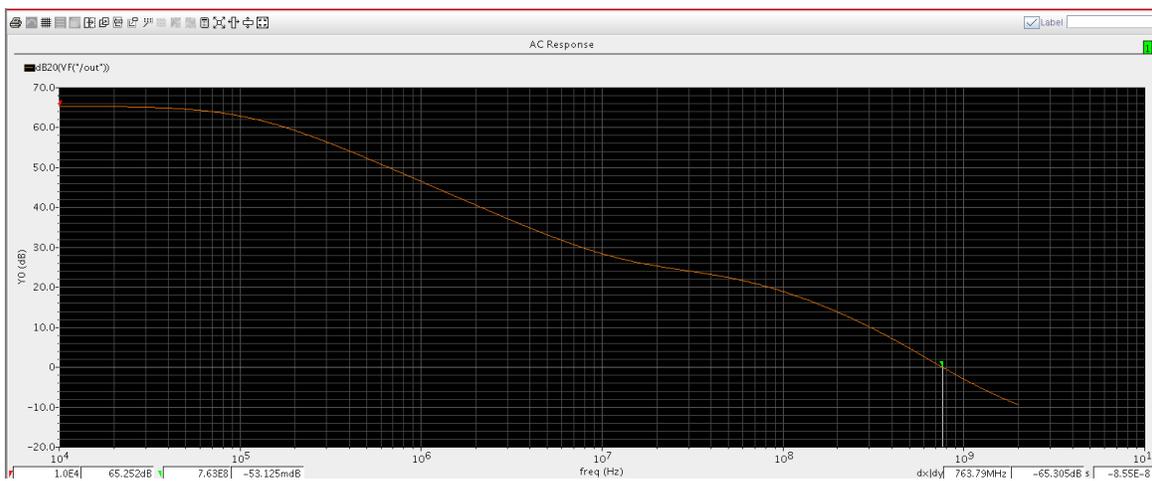


Figura 5.3.3: Resposta AC em malha aberta

O circuito usa duas voltagens de alimentação reguladas conforme já visto: 3 V e 3.6V. Estas serão geradas pela placa de alimentação do Rack. A tensão de referência (2.4V) será gerada por divisores resistivos externos, com um capacitor de desacoplamento também externo em paralelo para filtrar ruído.

Os tamanhos dos dispositivos usados são mostrados na tabela 5.3.1:

Transistor	Dimensões
To	15u/150n
T29	14.8u/120n
T11	14.8u/120n
T3	51.5u/1.6u
T2	1270u/1.6u
T1	94u/120n
T6	30u/1.6u
T9/T14/T13/T28	24.4u/1u 120útiplos: 1/10/3/1
T8/T7/T10/T31	30.4u/120n 120útiplos: 1/10/3/1
T17/T16	94u/120n
T12/T14	51.5u/1.6u *13
T18/T30	14.8u/120n
T15	10u/100n
T5	2u/100n
C8 (Capacitor integração)	185 fF
Mimcap0	24 pF
Opppc1	2K
Opppc0	5K

Tabela 5.3.1 – Dimensões dos transistores do Pré-amplificador

A seguir aplicamos pulsos de carga igual à mínima carga detectável (4fC), típica (10fC) e máxima (55 fC) baseando-se nas especificações do experimento. Para cada carga medimos o degrau em tensão na saída do integrador:

5fC (min) → degrau of 29.03 mV
 10fC → degrau of 28.48 mV
 55fC (max) → degrau of 298.6 mV

Esta amplitude do degrau será usada como especificação para o projeto do comparador.

Na Figura 5.3.4 mostramos a forma de onda correspondente. O transiente no início da forma de onda corresponde ao fechamento da malha de realimentação para ajustar o potencial de saída. O loop é aberto após 3 μ s.

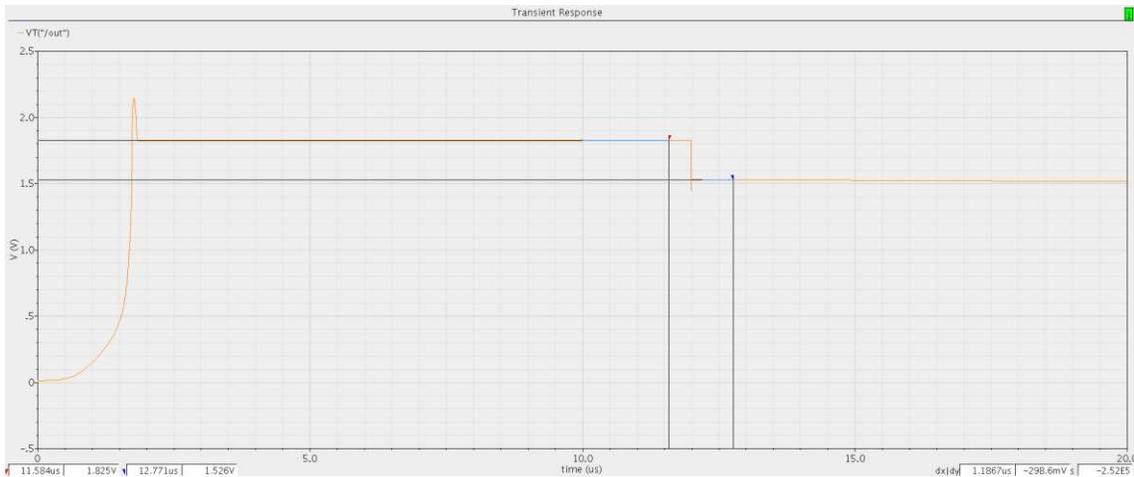


Figura 5.3.4: Resposta a um pulso de carga

Usando-se a análise transient-noise do simulador Spectre (que inclui todas as fontes de ruído de todos os dispositivos) poderemos investigar se o ruído interfere na detecção do salto DC correspondente a mínima carga (5 fC). Usaremos o parâmetro noise f_{max} = 40GHz para esta simulação.

O resultado está mostrado na figura 5.3.5 onde vemos que o nível de ruído é aceitável e não mascara a detecção conforme esperado: o salto continua sendo 30 mV.

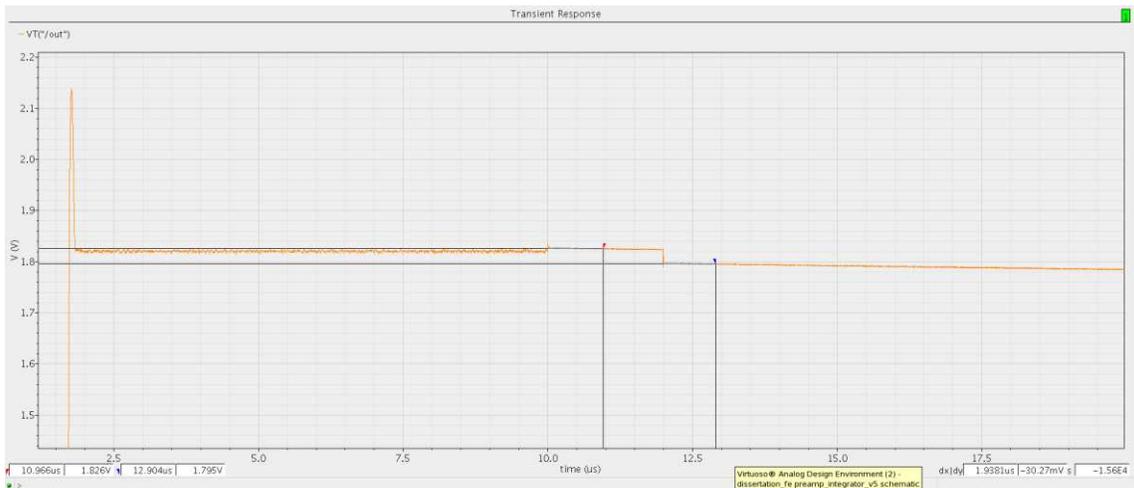


Figura 5.3.5: Resposta a um pulso de carga incluindo ruído dos dispositivos

5.4 – Comparador CMOS

O comparador é composto de 3 estágios, um pré-amplificador seguido de um latch analógico e finalmente um latch digital. O pré-amplificador aumenta a velocidade de comparação e domina o ruído. Sendo assim ele foi projetado para ter o mais alto ganho

possível com o compromisso de não termos transistores muito grandes os quais poderiam impactar a velocidade de resposta.

No latch analógico usaremos uma nova técnica pra evitar o efeito de “Kickback” a qual melhora a precisão geral do comparador (23).

A figura 5.4.1 mostra o estágio pré-amplificador do comparador. O ganho desejado é de 40 dB e as transcondutâncias dos dispositivos do estágio de entrada após o projeto usando a mesma metodologia Gm/Id são:

$G_m(\text{nmos})=875.7\mu\text{S}$ para o par diferencial (T0, T1) e $G_m(\text{PMOS})=253.8\mu\text{S} / 261.2\mu\text{S}$ para os dispositivos de carga.

O transistor T7 reseta o latch curto-circuitando as duas saídas de forma a não termos nenhum efeito de memória entre as amostras.

A faixa de variação da tensão de entrada esperada é de 2 V a 1.65 V a qual se encaixa nas simulações do pré-amplificador de carga.

Realimentação positiva foi usada no PMOS (T3-T6) de carga de forma a aumentarmos o Ganho conforme a equação:

$$A_{vo} = \frac{g_{m1}}{g_{mT6} - g_{mT3}}$$

O ganho simulado foi de 41.5 dB

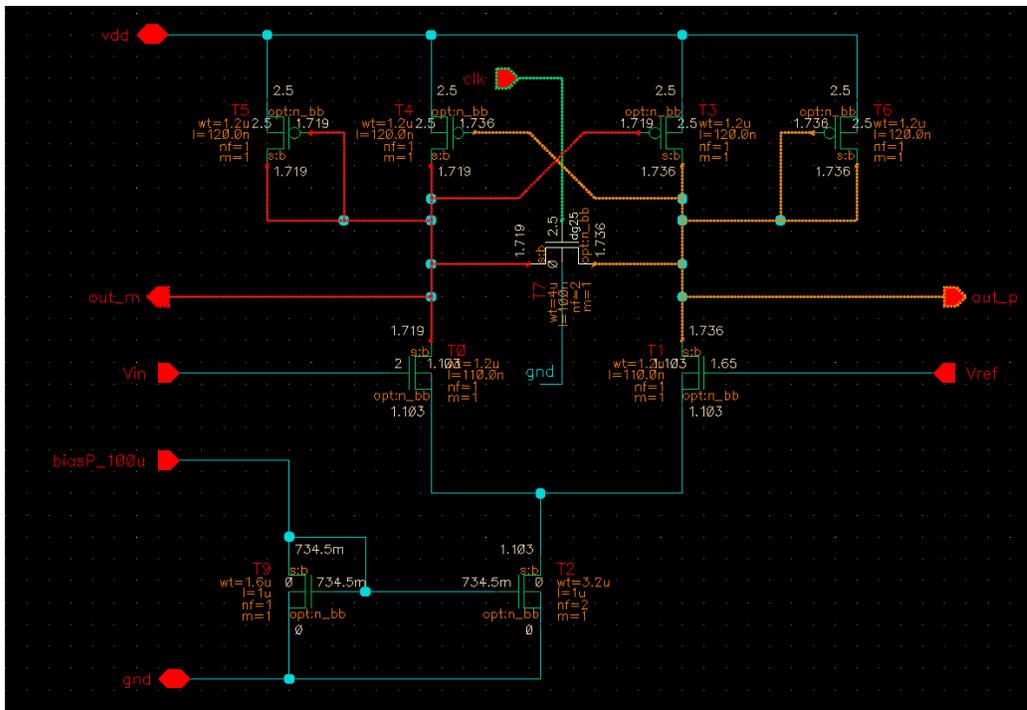


Figura 5.4.1: Primeiro estágio do Comparador (pré-amplificador)

Para simularmos este bloco usamos o esquemático de teste mostrado na figura 5.4.2. A única corrente de polarização necessária é 100 μ A.

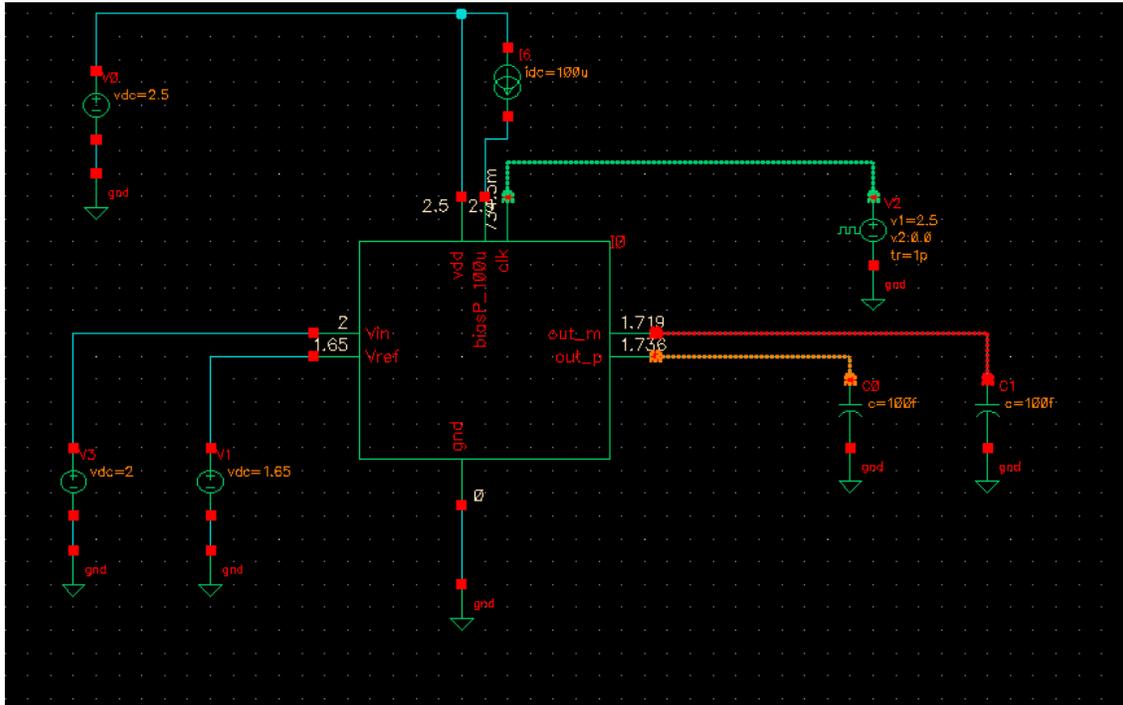


Figura 5.4.2: Testbench do Pré-amplificador

Aplicando o degrau mínimo (29 mV) obtido durante a simulação do amplificador de carga verificamos o chaveamento do comparador conforme mostra a figura 5.4.3: a saída é forçada para 1.7 V durante o reset e uma vez livre cria uma saída diferencial de 147 mV a qual é suficiente para o próximo estágio.

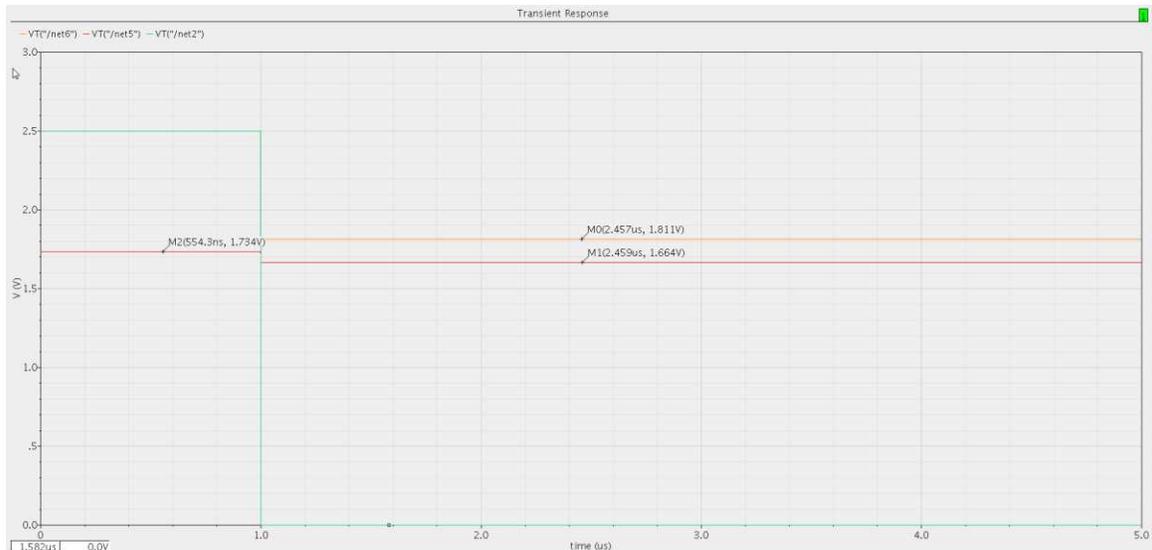


Figura 5.4.3: Simulando o Pré-amplificador

O atraso deste estágio foi simulado como 3.5 ns conforme mostra a figura 5.4.4:

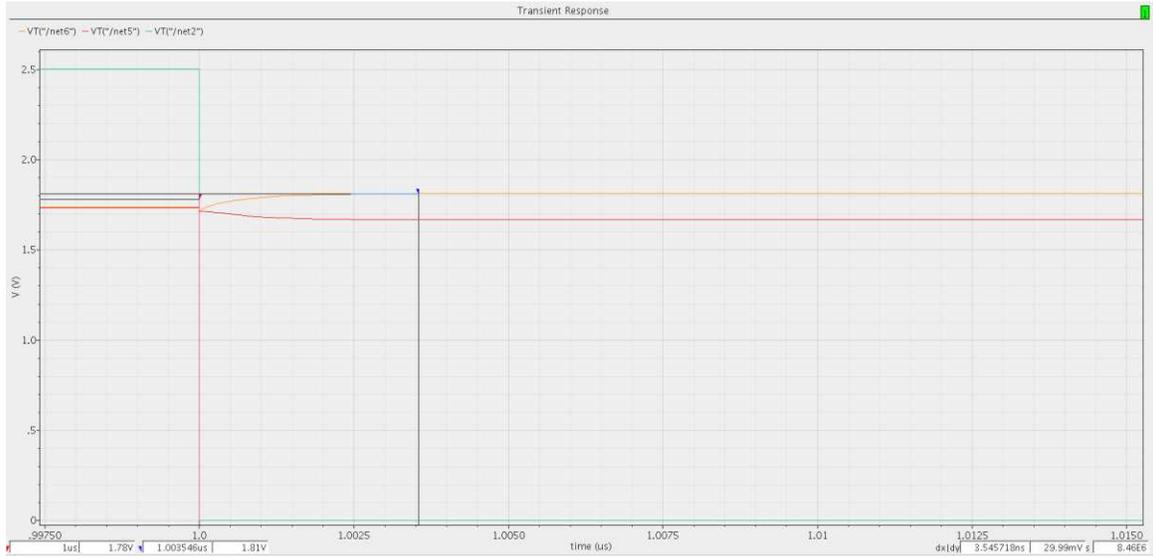


Figura 5.4.4: Atraso no Pré-amplificador

A seguir temos o latch analógico: essencialmente dois amplificadores CMOS acoplados conforme mostrado na figura abaixo. Os transistores T5/T0 e T3/T1 implementam os dois amplificadores realimentados e o transistor T7 reseta este estágio conforme mostra a figura 5.4.5.

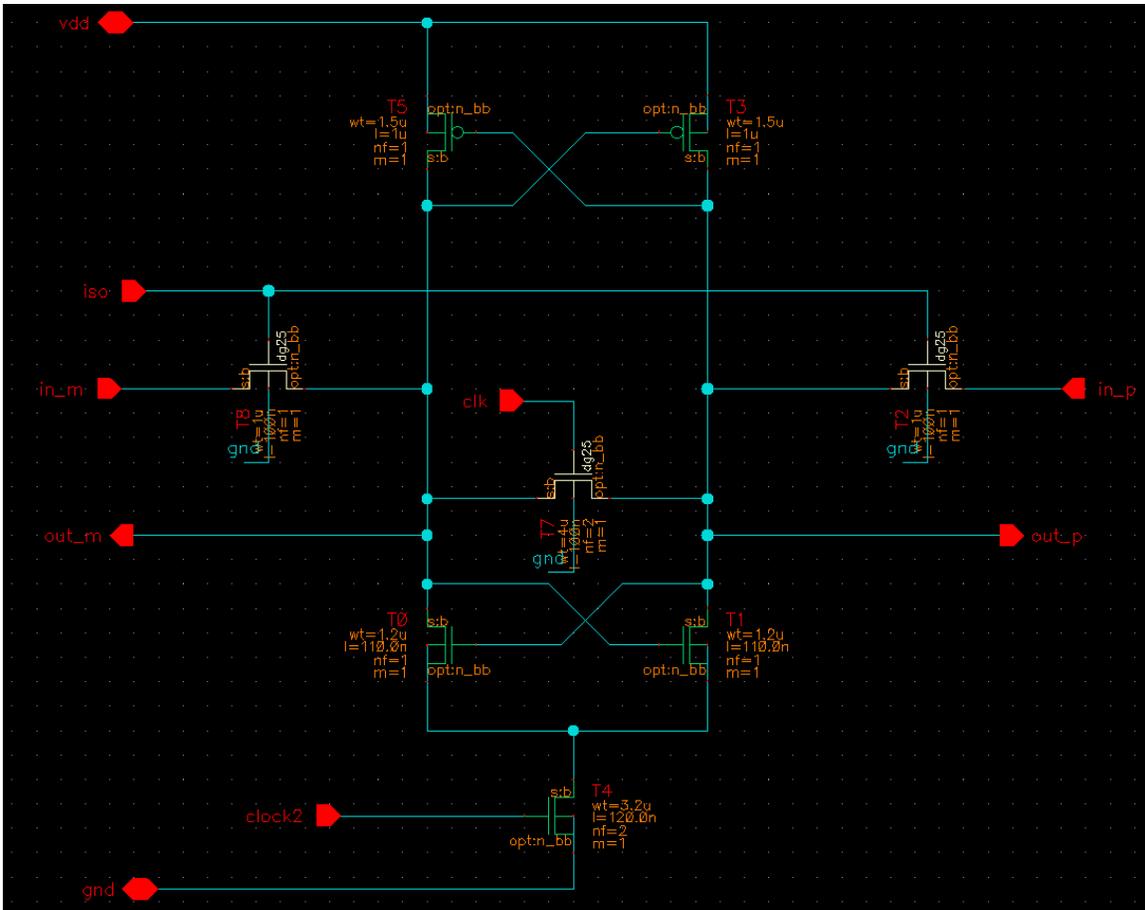


Figura 5.4.5: Latch Analógico

Os transistores T8 e T2 operam como chaves temporizadas de forma a evitar que o “Kickback” deste estágio seja transmitido de volta para o primeiro estágio. Para isto as chaves são abertas durante um determinada fase. O timing do comparador será abordado a seguir baseando-se na figura 5.4.6:

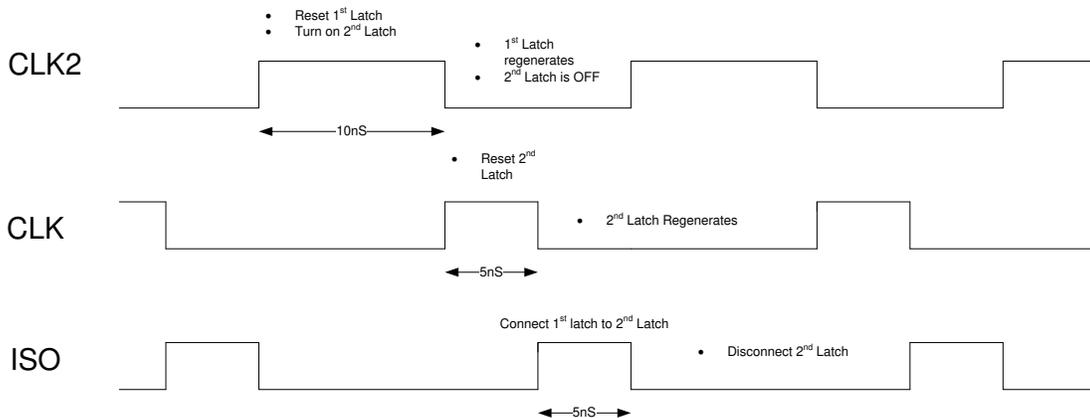


Figura 5.4.6: Reduzindo o efeito Kickback

Os sinais CLK2 e CLK comandam, de forma escalonada, o reset do pre-amplificador e do latch analógico. O sinal ISO comanda as chaves de isolamento entre o latch analógico e o pre-amplificador de forma a evitar qualquer injeção e carga de volta ao primeiro estágio durante a fase final de regeneração. Estas mesmas formas de onda são confirmadas na simulação elétrica na figura 5.4.7.

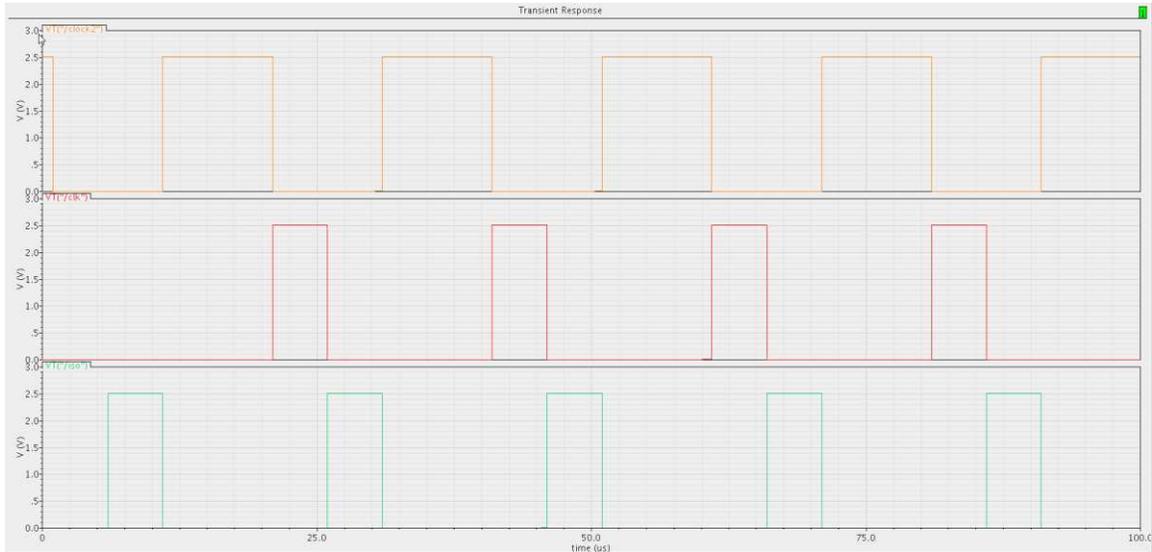


Figura 5.4.7: Reduzindo o efeito Kickback (Simulação)

A seguir colocamos os dois estágios em cascata. O circuito de teste dos dois estágios em série está mostrado na figura 5.4.8:

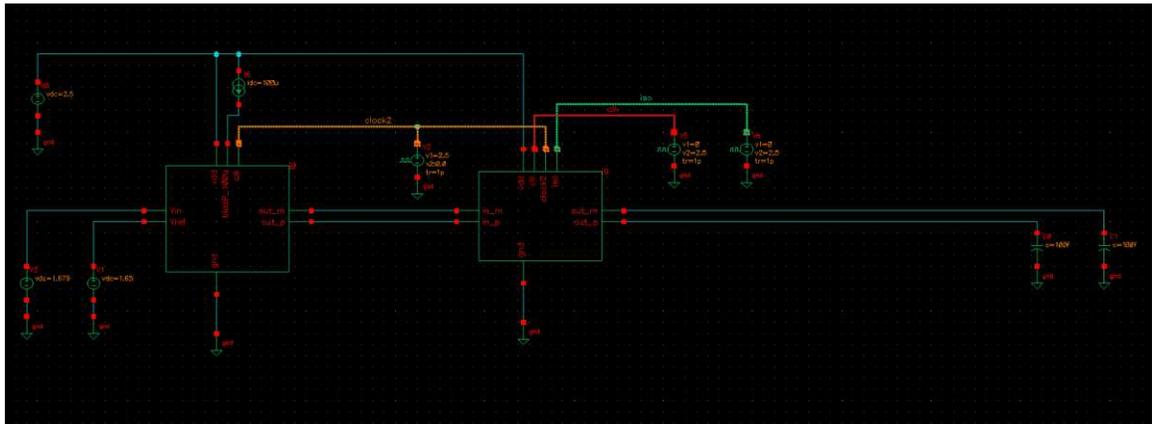


Figura 5.4.8: Pre-amp e latch analógica em cascata

As formas de onda na figura 5.4.9 mostram o resultado da simulação para o mínimo degrau. As três primeiras linhas são as 3 linhas de clock as quais já foram discutidas acima. Os sinais net5/net6 representam os nós de saída do primeiro latch e os nós net39/net40 são os nós de saída do latch analógico. Com este resultado confirmamos que a regeneração é rápida o suficiente.

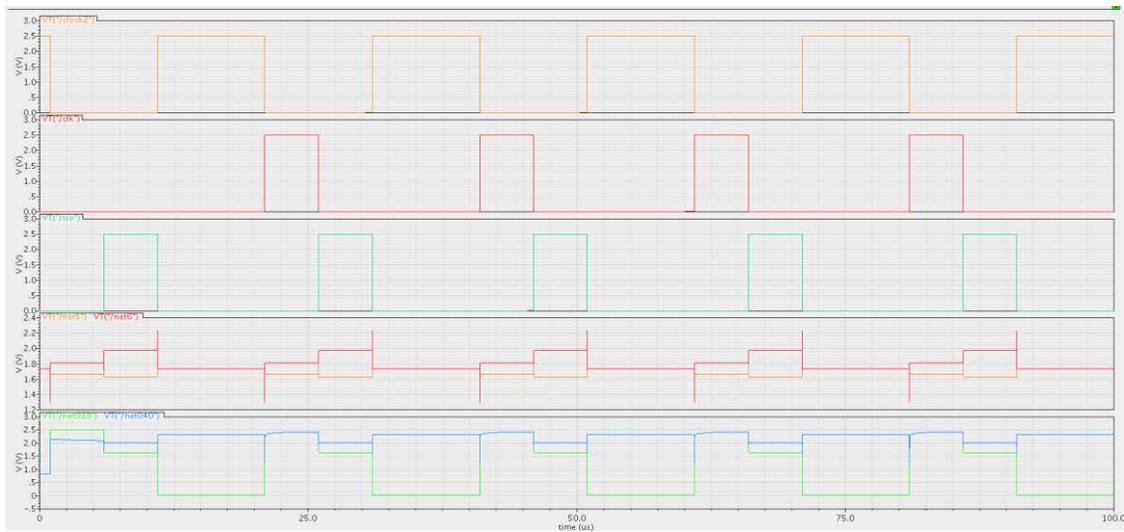


Figura 5.4.9: Aplicando o degrau mínimo na entrada do Pre-amp + Latch analógico

Como próximo passo projetamos o segundo latch (digital) e o integramos ao resto do circuito. O comparador está então completo e pode ser simulado por inteiro conforme mostra a figura 5.4.10.

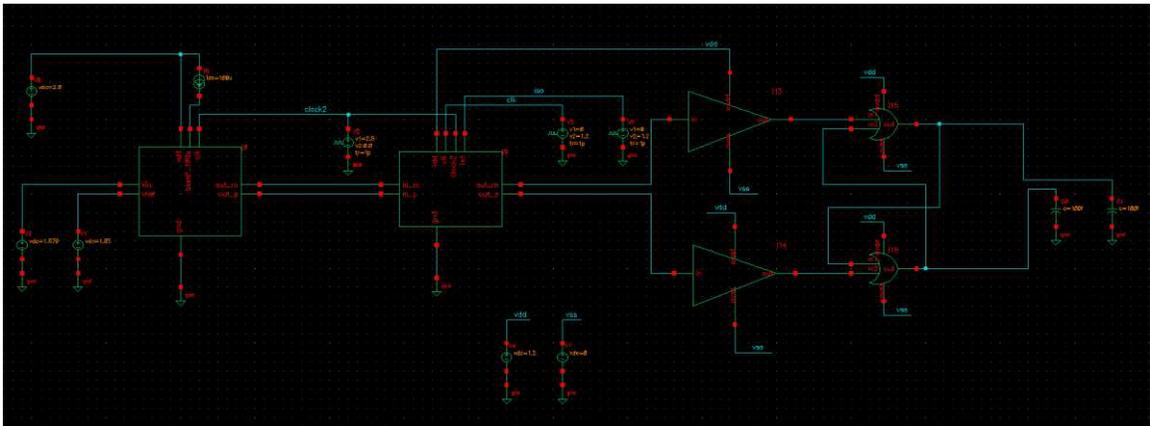


Figura 5.4.10: Comparador mostrando os seus 3 estagios

Aplicando o degrau mínimo a entrada de novo confirmamos a operação do comparador conforme mostrado na figura 5.4.11:

A primeira linha do topo para baixo mostra o degrau na entrada.

As linhas número 2, 3 e 6 (contando de cima para baixo) mostram as três linhas de clock.

As linhas 4 e 5 mostram a saída do comparador regenerando para uma excursão completa de 0V até 1.2V conforme esperado. Esta será a entrada do CI digital descrito nos capítulos anteriores, o qual terá uma lógica operando a partir de 1.2V.

O atraso do comparador foi estimado em 10.5 ns conforme mostrado na mesma figura.

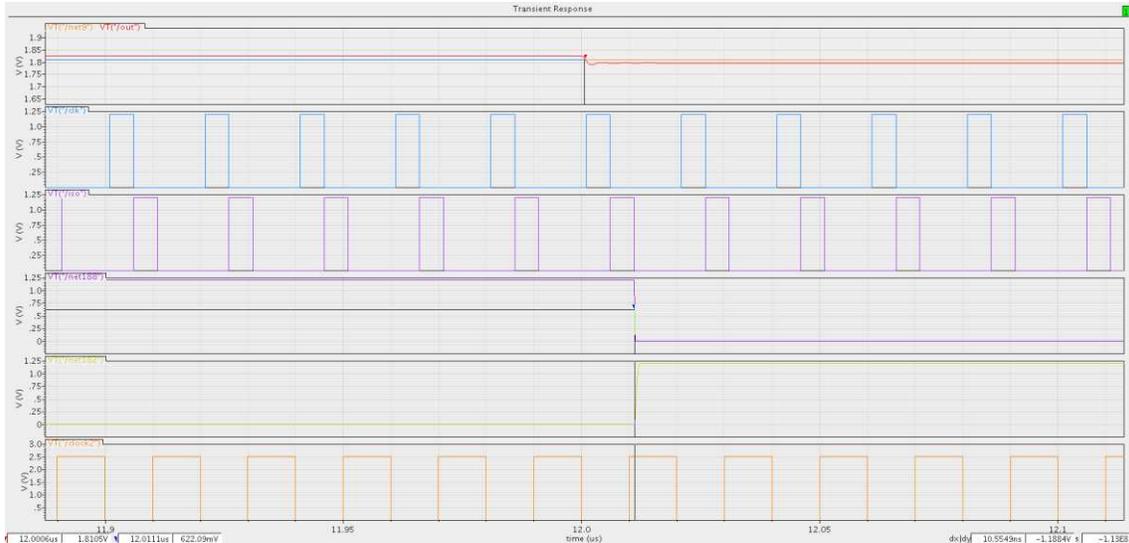


Figura 5.4.11: Comparador detectando mínimo degrau

5.5 - Simulando o front-end analógico Completo

Finalmente integramos todos os blocos analógicos num único circuito e verificamos a sua correta operação detectando um degrau mínimo de carga conforme mostrado na figura 5.5.1:

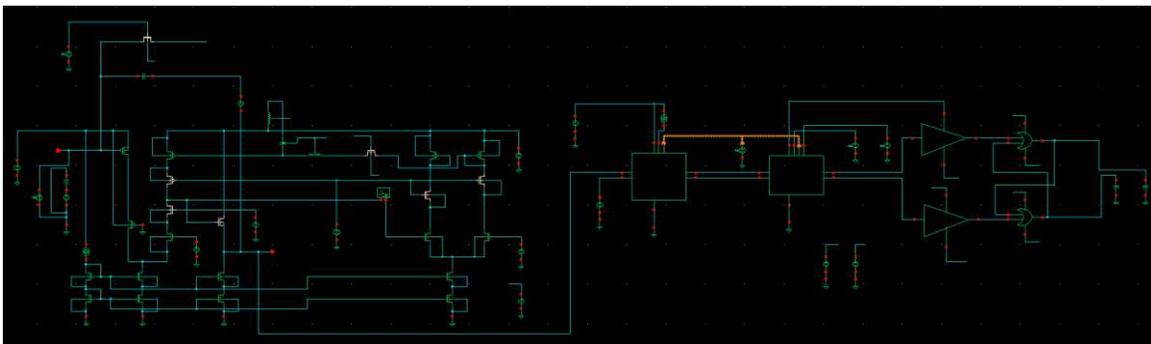


Figura 5.5.1: Simulando o front-end analógico completo

As cargas capacitivas na saída do latch digital são de 100 fF cada representando a carga dos gates digitais.

Nesta simulação usaremos uma carga mínima e aplicaremos a análise transient-noise com um $F_{max}=1\text{GHZ}$ para sermos mais precisos. As formas de onda da figura 5.5.2 mostram que o circuito detecta corretamente o degrau mesmo com a adição de ruído.

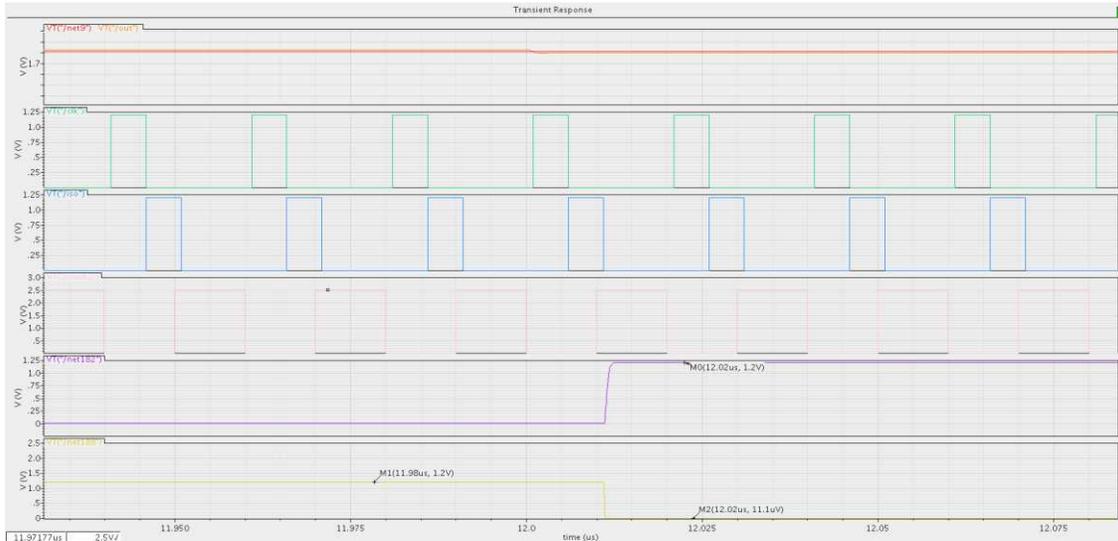


Figura 5.5.2: Comparador detectando mínima carga

Verificamos a operação de reset do comparador, o que é mostrado na figura 5.5.3. O pulso Vctl é o sinal de reset e podemos verificar a operação de reset fazendo a voltagem sobre o capacitor retornar ao valor quiescente.

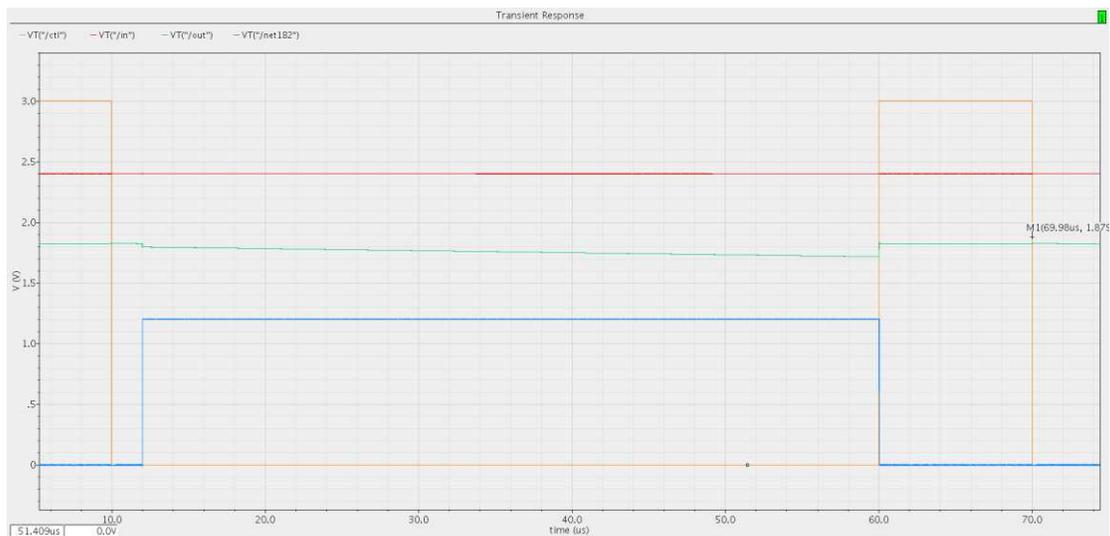


Figura 5.5.3: Reset do comparador

Capítulo 6 - Conclusão

Propomos neste trabalho uma abordagem sob a forma de um CI “mixed-signal” que implementa um circuito de trigger nível 0 através do reuso, de maneira escalonada, dos mesmos circuitos analógicos/digitais para determinar a validade de milhares de equações que determinam a trajetória de uma dada partícula no interior de um detector como o Dzero. Isto permite uma maior eficiência em termos de utilização de hardware e, portanto tanto uma menor área de pastilha como uma menor dissipação de potência quando comparado ao método atualmente empregado, o qual consiste em implementar todos os termos possíveis de todas as equações booleanas usando portas em FPGAs.

A redução da potência dissipada é importante para o detector uma vez que uma menor potência dissipada pela eletrônica significa menos calor gerado e, portanto é mais fácil manter os racks de instrumentação refrigerados a uma temperatura conveniente usando menos energia e de maneira mais eficiente.

A contribuição original deste trabalho é um novo algoritmo que nos permite reusar os mesmo circuitos a cada passo de execução de forma a consumirmos menos potência e reduzirmos a área do circuito integrado. Incluído na proposta está uma nova abordagem de como gerenciar o problema das “trilhas fantasmas” que surgem dentro do detector.

Também foi proposto um novo front-end analógico que converte os sinais enviados pelos sensores óticos em níveis lógicos de tensão que podem ser usados então pelo circuito de trigger.

Simulações efetuadas usando uma descrição VHDL de alto nível do circuito confirmaram a correta operação do sistema na parte digital enquanto simulações elétricas confirmaram funcionamento do front-end analógico.

Referencias Bibliográficas

1. *What is LHCb*. **Group, CERN Communication**. January 2008., CERN FAQ.
2. *The Dzero detector*. **S. Abachi et al.** 1994, Nuclear Instrumentation and Methods in Physics Research Vol 338, pp. 185-253.
3. *Luminosity calculations for Dzero*. **N. Amos, J. Linnemann, R. Partridge, and L. Paterno**. 1994, Dzero note 2031.
4. *Change to the D0 luminosity monitor constant*. **N. Amos et al.** 1994, DO note 2186.
5. *The Tevatron*. **Lederman, Leon M.** March 1991, Scientific American, Vol 264, number3, pp. 48 - 55.
6. *The Tevatron energy doubler: A superconducting accelerator*. **Edwards, Helen T.** 1985., Annual Reviews of Nuclear and Particle Science Vol. 35, pp. 605 - 660.
7. *Geant4—a simulation toolkit*. **Agostinelli, S. et al.** s.l. : Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated equipment, 2003, Vol. 3.
8. *The CDF trigger Silicon Vertex Tracker (SVT)*. **Belforte, S., et al.** 1994, Nuclear Science Symposium and Medical Imaging Conference Vol. 2, pp. 824-828.
9. *D0 Silicon Tracker Technical Design report*. **D0 collaboration**. s.l. : D0 note 2169, 1994.
10. *A beginners Guide to the SVXII*. **Yarema, Ray**. s.l. : Fermilab Internal Report., 1994.
11. *Level 1 Trigger design for the D0 upgrade Central Fiber Tracker*. **Borcherding, F.** s.l. : Fermilab D0 Note 2359, 1994.
12. *Proposal for a new architecture for the new trigger system using the scintillation fiber detector*. **Martin, Manuel**. s.l. : Fermilab D0 Note 3127, 1995.
13. *Routing of signals for the SF trigger using 8 layers geometry*. **Martin, Manuel**. s.l. : Fermilab D0 Note 3156, 1996.
14. *A specification for the timing of the signal to the SVX and trigger logic for the Upgrade CFT electronics*. **Borcherding, Fred**. s.l. : Fermilab D0 Note 2170, 1995.
15. *Math software packs new power; new programs automate such tedious processes as solving nonlinear differential equations and converting units*. **Shanley, Agnes**. s.l. : Chemical Engineering, 2002.
16. *Trigger equation and error calculations using Analytical Techniques*. **Johnson, Marvin**. s.l. : Fermilab D0 Note 2615, 1995.
17. **Pekarik, John J.** *Scaled CMOS Technology and Models to Support Wireless Applications*. s.l. : IBM Microelectronics report, 2006.
18. **Neil H.E. West, David Harris.** *CMOS VLSI Design - 3rd Edition*. s.l. : Addison-Wesley, 2005.
19. **Technologies, Dolphin**. [Online] http://www.dolphin-ic.com/ibm_9sf_cell.html.
20. **Inc, Cadence**. Virtuoso Spectre simulator. [Online] http://www.cadence.com/products/rf/spectre_circuit.
21. IBM 9LP/9RF CMOS Process . *Mosis MPW service*. [Online] <http://www.mosis.com/vendors/view/ibm/9lprf>.
22. **Jespers, Paul**. *The Gm/Id methodology, a sizing tool for low-voltage analog Circuits*. s.l. : Springer, 2010.

23. *Kickback Noise Reduction Techniques for CMOS latched Comparators.* **Pedro M. Figueiredo, Joao C. Vital.** 7, s.l. : IEEE Transactions on Circuits and Systems II, July 2006, Vol. 53.