



UNIVERSIDADE FEDERAL DA BAHIA
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Igo Amauri dos Santos Luz

SISTEMA ONLINE EM FPGA PARA REDUÇÃO DA
SOBREPOSIÇÃO DE SINAIS NUM DETECTOR DE
PARTÍCULAS

DISSERTAÇÃO DE MESTRADO

Salvador
Dezembro de 2016

Página em branco



Igo Amauri dos Santos Luz

SISTEMA ONLINE EM FPGA PARA REDUÇÃO DA SOBREPOSIÇÃO DE SINAIS NUM DETECTOR DE PARTÍCULAS

Dissertação de Mestrado apresentada ao
Programa de Pós-graduação em Engenharia
Elétrica, PPGEE, da Universidade Federal
da Bahia, como parte dos requisitos
necessários à obtenção do título de Mestre
em Engenharia Elétrica.

Orientadores: Eduardo F. de Simas Filho
Paulo C. M. de Abreu Farias

Salvador
Dezembro de 2016

Luz, Igo Amaurí dos Santos
Sistema Online em FPGA para Redução da Sobreposição de Sinais
num Detector de Partículas / Igo Amaurí dos Santos Luz. --
Salvador, 2016.
130 f. : il

Orientador: Eduardo Furtado de Simas Filho.
Coorientador: Paulo Cesar Machado de Abreu Farias.
Dissertação (Mestrado - Programa de pós-graduação em
Engenharia Elétrica) -- Universidade Federal da Bahia, Escola
Politécnica, 2016.

1. Instrumentação Eletrônica. 2. Processamento Digital de
Sinais. 3. Filtro Digital. 4. Deconvolução. 5. FPGA. I. Filho,
Eduardo Furtado de Simas. II. Farias, Paulo Cesar Machado de
Abreu. III. Título.

SISTEMA ONLINE EM FPGA PARA REDUÇÃO DA SOBREPOSIÇÃO DE
SINAIS NUM DETECTOR DE PARTÍCULAS

Igo Amauri dos Santos Luz

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO PROGRAMA
DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (PPGEE) DA
UNIVERSIDADE FEDERAL DA BAHIA COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS
EM ENGENHARIA ELÉTRICA.

Aprovada por:



Prof. Eduardo F. Simas Filho, D.Sc. - Orientador



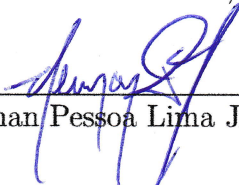
Prof. Paulo Cesar M. Abreu Farias, D.Sc. - Coorientador



Prof Amauri Oliveira, D.Sc. - Avaliador UFBA



Prof Edson Pinto Santana, D.Eng. - Avaliador UFBA



Prof Herman Pessoa Lima Jr , D.Sc. - Avaliador CBPF

SALVADOR, BA – BRASIL

DEZEMBRO DE 2016

dos Santos Luz, Igo Amauri

Sistema Online em FPGA para Redução da Sobreposição de Sinais num Detector de Partículas/Igo Amauri dos Santos Luz. – Salvador: UFBA/PPGEE, 2016.

XVII, 112 p.: il.; 29, 7cm.

Orientadores: Eduardo F. de Simas Filho

Paulo C. M. de Abreu Farias

Dissertação (mestrado) – UFBA/PPGEE/Programa de Engenharia Elétrica, 2016.

Referências Bibliográficas: p. 90 – 97.

1. Instrumentação Eletrônica. 2. Processamento Digital de Sinais. 3. Filtro Digital. 4. Deconvolução. 5. FPGA. 6. Física de Partículas. 7. Calorímetro. 8. ATLAS. I. de Simas Filho, Eduardo F. *et al.*. II. Universidade Federal da Bahia, PPGEE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Agradeço, primeiramente, aos meus pais, pelo irrestrito esforço para me proporcionar uma educação de qualidade.

A Rafael Ramos, pelo constante apoio e incentivo.

Aos meus amigos, em especial à Nildo Mascarenhas, Leilane Paixão, Eudes Benício, Rosileide Santos, Edjane Moreira, Nelson Alves e Camila Pinheiro.

Aos meus orientadores, Prof. Dr. Eduardo Simas e Prof. Dr. Paulo Cesar. Obrigado por todo o ensinamento e contribuições para o meu crescimento enquanto pesquisador. À todos os professores e funcionários do PPGEE/UFBA.

Agradeço, também, aos pesquisadores da Colaboração do ATLAS, em especial à Me. João Paulo Duarte, Prof. Dr. Luciano Andrade e Prof. Dr. José Seixas, pelas contribuições essenciais no desenvolvimento desse trabalho.

E, por fim, à FAPESB, FAPERJ, CAPES, CNPq e RENAFAP (MCTI) pelo apoio financeiro.

Resumo da Dissertação apresentada à PPGEE/UFBA como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica (M.Sc.)

SISTEMA ONLINE EM FPGA PARA REDUÇÃO DA SOBREPOSIÇÃO DE SINAIS NUM DETECTOR DE PARTÍCULAS

Igo Amauri dos Santos Luz

Dezembro/2016

Orientadores: Eduardo F. de Simas Filho

Paulo C. M. de Abreu Farias

Programa: Engenharia Elétrica

A sobreposição de sinais pode ocorrer em diversos sistemas de instrumentação. Esse fenômeno está relacionado à taxa de ocorrência dos eventos e ao tempo de resposta dos sensores. No âmbito dos detectores de partículas, que caracterizam-se por possuírem altas taxas de ocorrência de eventos, o sistema responsável pela medição de energia (calorímetro) possui limitações em termos do tempo de resposta, provocando restrições em relação ao aumento na taxa de ocorrência das colisões das partículas. Essa situação apresenta-se como um desafio quando o avanço nas pesquisas está atrelado ao aumento da taxa de colisões gerada pelos aceleradores de partículas, como é o caso do *Large Hadron Collider* (LHC). O ATLAS é um dos detectores do LHC e seu sistema de instrumentação não está preparado para lidar com o aumento da taxa de ocorrência de eventos, pois isso irá gerar sobreposição entre eventos sequenciais. Para atenuar esse problema, torna-se necessário realizar, de modo *online*, o processamento digital da informação. Nesse sentido, esse trabalho propõe a implementação de um circuito eletrônico dedicado em plataforma digital reconfigurável utilizando técnicas de filtragem de sinais. Cada filtro é uma aproximação da inversa da resposta ao impulso do canal de medição, realizando a deconvolução dos sinais medidos. A avaliação do funcionamento e desempenho dos

filtros foi realizada através de um conjunto de dados simulados, que consideram o funcionamento real do detector. A partir dos resultados alcançados, pôde-se constatar que as técnicas implementadas atenderam aos requisitos de funcionamento exigido.

Abstract of Dissertation presented to PPGEE/UFBA as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

FPGA-BASED ONLINE SYSTEM TO REDUCE THE SIGNAL PILEUP EFFECT IN A PARTICLE DETECTOR

Igo Amauri dos Santos Luz

December/2016

Advisors: Eduardo F. de Simas Filho

Paulo C. M. de Abreu Farias

Department: Electrical Engineering

The pileup effect can occur in several instrumentation systems. This phenomenon is related to the rate of occurrence of the events and the response time of the sensors. In the particle detectors, the electronic system responsible for measuring energy (calorimeter) has limitations in terms of response time when it operates with high rate of occurrence of events. This situation represents a challenge when the advance in the research is linked to the increase of the collision rate generated by the particle accelerators, such as the LHC. ATLAS is one of the LHC detectors and its instrumentation system is not prepared to deal with the increase rates of occurrence of events, because it will generate superimposition of information from subsequent events. To mitigate this problem it is necessary to perform an online digital signal processing. So, this work proposes a dedicated signal processing system embedded on a digital high-performance reconfigurable electronic platform that implements digital signal filtering techniques. Each filter is an inverse approximation of the inverse impulse response of the measuring channel, performing the deconvolution of the measured signals. The evaluation of the operation and performance of the implemented filters was performed through the use of a simulated data set, which considers the actual operation of the detector. From the achieved results, it was verified that the implemented techniques met the system operating requirements.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xiii
Lista de Abreviaturas	xiv
1 Introdução	1
1.1 Objetivos	3
1.2 Resultados em Publicações	4
1.3 Organização da Dissertação	7
2 Introdução à Física de Partículas e ao Acelerador LHC	8
2.1 Histórico da Física de Partículas	8
2.2 O LHC	11
2.3 O Detector ATLAS	14
2.3.1 A Calorimetria do ATLAS	16
2.3.2 O Sistema Online de Seleção de Eventos (<i>Trigger</i>) do ATLAS	20
3 Atualização do LHC e os Métodos de Estimação de Energia	23
3.1 Introdução	23
3.2 Sistema Eletrônico Atual do Primeiro Nível de <i>Trigger</i> Baseado em Calorimetria	26
3.3 Sistema Eletrônico que será Utilizado após a Atualização do LHC . .	29
3.4 Técnica Atual de Estimação de Energia	32
3.4.1 O Filtro Casado	33
3.4.2 Detector de Picos	35
3.5 Proposta de Nova Técnica de Estimação de Energia	37

3.5.1	Filtros Digitais e a Teoria da Convolução	38
3.5.2	Método Determinístico	43
3.5.3	Método Iterativo	47
4	Arquitetura de Hardware Desenvolvida	51
4.1	Introdução	51
4.2	Processamento Paralelo e Diferentes Domínios de <i>Clock</i>	55
4.3	Implementação do Método Determinístico	57
4.4	Implementação do Gradiente Descendente Positivo	58
4.5	Verificação Funcional	62
5	Resultados	64
5.1	Geração dos Sinais Simulados	64
5.2	Método Determinístico	66
5.2.1	Simulação do Método Determinístico	66
5.2.2	Síntese FPGA do Método Determinístico	73
5.3	Método Iterativo: Gradiente Descendente Positivo	74
5.3.1	Simulação do Método Iterativo	75
5.3.2	Síntese FPGA do Método Iterativo	77
5.4	Análise dos Resultados	83
6	Conclusões	87
	Referências Bibliográficas	90
A	Projeto de Hardware Digital	98
A.1	Introdução	98
A.2	Field-Programmable Gate Arrays - FPGA	100
A.2.1	Família Virtex-7 da Xilinx	102
A.2.2	Estruturas de DSP	104
A.2.3	Bloco de Lógica Configurável	105

Lista de Figuras

2.1	Esquema do aparelho usado para medir a deflexão elétrica e magnética dos raios catódicos [15].	9
2.2	Diagrama do modelo padrão [22].	12
2.3	Fotografia do túnel do LHC [32].	13
2.4	Ilustração da localização dos quatro maiores detectores do LHC ao longo do percurso do acelerador [22].	14
2.5	Esquemático dos detectores do ATLAS [22].	15
2.6	Eixo de coordenadas do ATLAS [22].	16
2.7	Corte transversal do ATLAS [22].	18
2.8	Diagrama dos calorímetro eletromagnético e hadrônico do ALTAS [42].	18
2.9	Ilustração do formato do <i>TileCal</i> [45].	19
2.10	Exemplo de Segmentação de uma parte do <i>TileCal</i> [47].	20
2.11	Sistema de filtragem do ATLAS [22].	21
2.12	Diagrama de Blocos do Nível 1 do <i>Trigger</i> [4].	21
3.1	Representação gráfica da Resposta ao Impulso Típica de um sensor [47].	24
3.2	Representação gráfica do efeito de empilhamento de sinais medidos [47].	24
3.3	Arquitetura do <i>Trigger</i> L1 baseado em calorimetria atual [47].	25
3.4	Arquitetura do <i>Trigger</i> L1 baseado em calorimetria em desenvolvimento [47].	25
3.5	Representação gráfica do módulo do <i>TileCal</i> [32].	26
3.6	Esquemático da eletrônica de uma gaveta do <i>TileCal</i> [32].	27
3.7	Representação gráfica do caminho do sinal na arquitetura atual do <i>TileCal</i> [5].	28

3.8	Representação gráfica do caminho da informação na atualização dos sistemas eletrônicos do <i>TileCal</i> [51]	29
3.9	Protótipo da Proposta de Arquitetura de Modificação do Front-End Baseado no Cartão 3 em 1 [53].	30
3.10	Diagrama do Fluxo do Sinal do <i>Front-End</i> Baseado no Chip <i>QIE</i> [52].	30
3.11	Diagrama do Fluxo do Sinal do <i>Front-End</i> Baseado no Chip <i>FATA-LIC</i> [52].	31
3.12	Protótipo <i>sROD</i> para Atualização do Sistema Eletrônico [57]	32
3.13	Representação gráfica do sinal transmitido e recebido pela atual arquitetura do BCID [9].	33
3.14	Diagrama da representação do problema da teoria da decisão [58]. . .	34
3.15	Diagrama de blocos do sistema de BCID [9].	36
3.16	Representação gráfica da detecção da energia no primeiro nível do <i>Trigger</i> na arquitetura atual do <i>TileCal</i> [12].	36
3.17	Representação gráfica do efeito de sobreposição entre sinais adjacentes [6].	37
3.18	Diagrama do modelo da sobreposição de sinais no Calorímetro	38
3.19	Diagrama de blocos da filtragem de um sinal digital.	39
3.20	Representação da Forma Direta do Filtro FIR [60].	40
3.21	Representação da Forma Transposta do Filtro FIR [60].	40
3.22	Representação da Forma em Cascata do Filtro FIR [61].	40
3.23	Representação da Forma Treliça do Filtro FIR [61].	41
3.24	Representação do bloco básico utilizado na Forma Treliça [61]. . . .	41
3.25	Representação da Forma Polifásica [61].	42
3.26	Representação Gráfica do Plano Z [12].	44
3.27	Representação de duas possíveis RDCs para sistemas com função racional.	45
3.28	Exemplo de recuperação do sinal do <i>TileCal</i> usando o método determinístico.	47
3.29	Representação da lógica do Gradiente Descendente Positivo [12]. . . .	50
4.1	Arquitetura do Sistema Eletrônico Implementado em FPGA.	52
4.2	Diagrama do Bloco Configura Filtros.	53

4.3	Diagrama do Bloco Conversor Serial/Paralelo.	54
4.4	Diagrama do Bloco Conversor Paralelo/Serial.	54
4.5	Diagrama do Bloco Controle.	54
4.6	Diagrama do Bloco Processamento Sinal.	55
4.7	Diagrama dos Estágios do Paralelismo.	56
4.8	Diagrama da Implementação em hardware do método determinístico.	57
4.9	Diagrama da arquitetura da implementação do filtro FIR.	58
4.10	Diagrama da Máquina de Estados Finita utilizada para o método iterativo.	60
4.11	Arquitetura da Verificação Funcional.	62
5.1	Representação gráfica do pulso característico do <i>TileCal</i> [12].	65
5.2	Representação gráfica de exemplos do pulso característico do <i>TileCal</i> com desvio de fase [12].	65
5.3	Representação gráfica dos coeficientes nas diferentes ordens utilizadas na simulação.	68
5.4	Representação gráfica de sinais de energia simulados com e sem o efeito de empilhamento.	69
5.5	Representação gráfica da aplicação do filtro com diferentes ordens no FPGA.	70
5.6	Gráficos da dispersão entre a energia ideal e a estimada em FPGA.	71
5.7	Gráficos da dispersão entre a energia estimada no FPGA e no MATLAB®.	72
5.8	Representação gráfica da utilização dos recursos lógicos na implementação do método determinístico	74
5.9	Representação gráfica do pulso característico do <i>TileCal</i> utilizado na simulação do método iterativo.	76
5.10	Representação gráfica do sinal reconstruído com diferentes quantidade de iterações.	76
5.11	Representação gráfica da relação entre a raiz quadrada do erro quadrático médio e o número de iterações.	78
5.12	Gráficos da dispersão entre a energia ideal e a estimada em FPGA no método iterativo.	79

5.13	Representação gráfica da relação entre a raiz quadrada do erro quadrático médio e o número de iterações considerando o FPGA e o modelo de referência.	80
5.14	Gráficos da dispersão entre a energia estimada no FPGA e no MATLAB® do método iterativo.	81
A.1	Diagrama do fluxo de projeto digital.	99
A.2	Diagrama conceitual da estrutura de um FPGA [77].	102
A.3	Diagrama conceitual de uma célula lógica [77].	102
A.4	Otimização da família 7 da Xilinx [79].	103
A.5	Diagrama das funcionalidades básicas do DSP48E1 [71].	105
A.6	Diagrama das funcionalidades básicas do bloco de lógica configurável[81].	106

Lista de Tabelas

4.1	Transição de Estados e Saídas	61
5.1	Valores do erro dos resultados da simulação FPGA	69
5.2	Valores do erro entre os resultados da simulação FPGA e o modelo de referência (MATLAB®)	72
5.3	Relatório de potência na síntese.	74
5.4	Valores do erro dos resultados da simulação FPGA no método iterativo.	78
5.5	Valores do erro entre os resultados da simulação FPGA e o modelo de referência (MATLAB®) do método iterativo	80
5.6	Utilização dos recursos lógicos do FPGA no método iterativo.	86
5.7	Relatório de potência na síntese do método iterativo.	86
5.8	Comparação dos resultados obtidos com a implementação dos métodos.	86
A.1	Características dos dispositivos da família Virtex-7	104

Lista de Abreviaturas

ADC	Conversor Analógico Digital, do inglês <i>Analog Digital Converter</i> , p. 27
ASIC	Circuito Integrado de Aplicação Específica, do inglês <i>Application Specific Integrated Circuit</i> , p. 31
BCID	Identificação do Bunch Crossing, do inglês <i>Bunch Crossing IDentification</i> , p. 32
CERN	Centro Europeu para Pesquisas Nucleares, do francês <i>Conseil Européen pour la Recherche Nucléaire</i> , p. 1
CI	Circuito Integrado, p. 98
CLB	Bloco de lógica configurável, do inglês <i>Configurable Logic Block</i> , p. 105
DMU	Unidade de gerenciamento de informações, do inglês <i>Data Management Unit</i> , p. 26
DSP	Processador Digital de Sinais, do inglês <i>Digital Signal Processor</i> , p. 39
FIR	Resposta ao Impulso Finita, do inglês <i>Finite Impulse Response</i> , p. 3
FSM	Máquina de Estados Finita, do inglês <i>Finite State Machine</i> , p. 60
HDL	Linguagem de Descrição de Hardware, do inglês <i>Hardware Description Language</i> , p. 99

HEP	Física de Altas Energias, do inglês <i>High-Energy Physics</i> , p. 1
IIR	Resposta ao Impulso Infinita, do inglês <i>Infinite Impulse Response</i> , p. 39
LHC	Grande Colisor de Hádrons, do inglês <i>Large Hadron Collider</i> , p. 1
LUT	<i>Look-Up Table</i> , p. 101
PMT	Tubos Fotomultiplicadores, do inglês <i>Photomultiplier Tubes</i> , p. 19
QSFP	<i>Quad Small Form-factor Pluggable</i> , p. 31
RDC	Região de Convergência, p. 44
ROD	<i>Drivers de Saída</i> , do inglês <i>Read-out Driver</i> , p. 28
RTL	<i>Register-Transfer Level</i> , p. 62
SNR	Relação sinal-ruído, do inglês <i>Signal-to-Noise Ratio</i> , p. 23
VHDL	Linguagem de Descrição de Hardware VHSIC, do inglês <i>VH-SIC Hardware Description Language</i> , p. 99

*Dedico esse trabalho aos meus pais, Nilda Nascimento dos
Santos Luz e Antonio José Luz*

Capítulo 1

Introdução

A Física de Partículas (em alguns casos chamada de Física de Altas Energias ou *High-Energy Physics* - HEP) é a área da ciência responsável por estudar a composição da matéria, em outras palavras, estuda as partículas subatômicas e suas formas de interação.

Desde a Grécia antiga já se desenvolviam teorias acerca dos elementos constituintes da matéria [1]. Desde então, diversos grupos de pesquisadores no mundo inteiro, vem desenvolvendo experimentos a fim de explorar questões relacionadas às partículas elementares. Mais recentemente, os experimentos de física de altas energias vêm sendo projetados para observar e caracterizar as partículas de interesse, que podem ser geradas por fontes naturais (ex. raios cósmicos) ou artificiais (aceleradores de partículas) [2]. Esses experimentos têm como objetivo comprovar os modelos teóricos previamente elaborados pelos pesquisadores e também fundamentar o desenvolvimento de novos modelos.

Nesse sentido, o Centro Europeu para Pesquisas Nucleares (CERN, da abreviação do nome em francês *Conseil Européen pour la Recherche Nucléaire*), um dos centros de pesquisa de referência em Física de Partículas, projetou e construiu o Grande Colisor de Hádrons (LHC, do inglês *Large Hadron Collider*) [3], sendo este o maior acelerador de partículas atualmente em operação. Ao longo da trajetória do LHC existem 6 detectores, cada um responsável por identificar características específicas das partículas. O ATLAS [4] é um desses equipamentos, caracterizado como de propósito geral, sendo constituído de alguns sub-detectores.

As informações dos sensores do ATLAS são processadas por sistemas eletrônicos

e computacionais dedicados, e, por fim, armazenadas em mídia permanente para análises posteriores [4]. Com o intuito de avançar nas pesquisas, o LHC está em processo de atualização para aumentar a quantidade de interações por cm^2 por segundo (que se traduz no aumento de luminosidade) e também a energia de colisões [5]. Consequentemente, essa atualização acarretará uma taxa de eventos maior a serem detectados e processados. Considerando o ambiente de funcionamento do calorímetro, que é o sub-detector do ATLAS dedicado à medição de energia e à estimação da posição da interação, o aumento da ocorrência de eventos provoca sobreposição da informação de eventos subsequentes medidos, fenômeno conhecido como *pileup*, ou empilhamento [6, 5].

O problema da sobreposição está associado à alta taxa de ocorrência dos eventos e ao tempo de resposta dos equipamentos responsáveis pela detecção dos mesmos. Além do ambiente do ATLAS, esse fenômeno ocorre em outras áreas que possuem sistemas eletrônicos com alta taxa de geração de eventos. Aparelhos biomédicos, por exemplo, são equipamentos que exigem alta precisão em seus sistemas de aquisição, como é o caso dos tomógrafos por emissão de pósitrons (PET), que exigem altas resoluções no processamento das imagens [7]. Em outra aplicação, na espectroscopia Gama, o efeito do *pileup* afeta a identificação das fontes radioativas desconhecidas, pois provoca a distorção do histograma das energias de fótons. Nesse cenário, a sobreposição ocorre devido a alta atividade da fonte radioativa associada à limitada resolução dos detectores [8]. Outra área que merece destaque nesse cenário são os sistemas de comunicações ópticas, nos quais a sobreposição pode modificar a informação transmitida, afetando diretamente sua eficiência.

No atual sistema eletrônico em operação no ATLAS, o pulso gerado em cada célula do detector é digitalizado através de um conversor analógico-digital e enviado para um sistema eletrônico responsável por aplicar técnicas de processamento de sinais para extrair as informações de interesse. A atual técnica de estimação de energia utilizada no primeiro nível de seleção do ATLAS é o Filtro Casado, utilizado para detectar o pulso do calorímetro [9, 10]. Com o aumento da luminosidade, em função do processo de atualização do LHC, e a inerente alta taxa de ocorrência de eventos, a eletrônica e as técnicas de estimação atualmente utilizadas tornam-se ineficientes no processo de identificar os eventos de interesse. Por isso, os pesquisadores

que compõem a colaboração do ATLAS vem desenvolvendo alterações no sistema eletrônico do detector com o objetivo de atender aos requisitos das novas fases de atualização do LHC.

Nesse cenário, torna-se necessário desenvolver um sistema digital capaz de recuperar a informação dos sinais de energia afetados pelo *pileup* no primeiro nível de seleção de eventos. Dessa forma, este trabalho apresenta uma solução que atende aos requisitos eletrônicos da atualização do ATLAS. A solução proposta consiste no desenvolvimento de um sistema digital em *Field Programmable Gate Array* (FPGA) que implementa técnicas de filtragem para reduzir os efeitos do empilhamento dos eventos detectados.

A partir da consideração do calorímetro como um sistema linear [11], o processo de sobreposição das informações foi interpretado como uma convolução do sistema de calorimetria com os sinais de interesse das colisões. Consequentemente, as técnicas de estimação das informações propostas baseiam-se na filtragem inversa, ou deconvolução. Dentre as soluções apontadas em [12], nesse trabalho foram implementadas duas abordagens: uma determinística e outra iterativa. A primeira baseou-se em filtro de resposta ao impulso finita (FIR, do inglês *Finite Impulse Response*) [13], que foi projetado para aproximar a função inversa da resposta em frequência do canal de medição e tende a utilizar poucos recursos lógicos do FPGA. A segunda é baseada na técnica de convergência de Van Cittert [14], através do Gradiente Descendente Positivo, que prioriza as amplitudes positivas, zerando as energias negativas a cada iteração. Esta solução utiliza mais recursos lógicos do FPGA devido a maior complexidade em suas operações, mas em contrapartida tende a ter melhor eficiência na reconstrução do sinal.

1.1 Objetivos

Este trabalho tem como objetivo desenvolver uma solução em FPGA capaz de reduzir os efeitos de sobreposição de eventos (*pileup*) no calorímetro do ATLAS. Para isso, foram implementadas diferentes técnicas de estimação de energia em *hardware* capazes de recompor o sinal original das amostras afetadas pela sobreposição. O desenvolvimento dessa solução levou em consideração os requisitos de funciona-

mento da eletrônica embarcada no *TileCal*, considerando o tempo de processamento, ocupação de recursos lógicos do FPGA e análise do consumo de potência. Vale destacar que a contribuição deste trabalho está na implementação da solução em *hardware*, com a proposta de um sistema adequado a ser embarcado na eletrônica do *TileCal*, e baseou-se nos projetos e avaliações de filtros realizados em [12].

Como objetivos específicos desse projeto, cita-se:

1. Compreender o funcionamento do sistema de calorímetro do ATLAS.
2. Compreender o funcionamento do sistema de aquisição e seleção das informações.
3. Compreender o funcionamento de diferentes abordagens de projeto de filtros digitais.
4. Descrever o *hardware* de forma a atender os requisitos de funcionamento do sistema eletrônico do ATLAS.
5. Desenvolver a verificação funcional a fim de garantir o funcionamento do *hardware*.
6. Avaliar o desempenho na estimação do sinal de energia das técnicas implementadas.
7. Avaliar os resultados obtidos com a síntese em FPGA das soluções propostas.

1.2 Resultados em Publicações

Os resultados obtidos com essa pesquisa foram apresentados e publicados no Congresso Brasileiro de Automática (CBA2016) 2016, Encontro de Física 2016 e Encontro Nacional de Física de Partículas e Campos 2015. Abaixo seguem os resumos dos trabalhos apresentados, sendo que o artigo completo do CBA 2016 está anexado.

LUZ, I. A. S., SIMAS FILHO, E. F., FARIAS, P. C. M. A., BECK, M., ANDRADE FILHO, L. M., DUARTE, J. P. B. S., de Seixas, José Manoel. **Sistema para Redução da Sobreposição de Sinais em Calorimetria de Altas Energias em Operação com Alta Taxa de Eventos**. Congresso Brasileiro de Automática CBA, 2016, Vitória. Anais do CBA 2016.

Resumo: Sistemas que funcionam em tempo real possuem diversos desafios em termos de desempenho que precisam ser atendidos em seu projeto, tais como: tempo de processamento, consumo de potência, área utilizada, entre outros. Dessa forma, para atender a essas restrições, esses projetos precisam ser desenvolvidos em *hardware* dedicado. Um problema que ocorre nos sistemas de processamento digital de sinais é a sobreposição entre informações provenientes de eventos subsequentes. Essa distorção do sinal é chamada empilhamento (do inglês *pileup*) e ocorre com muita frequência no âmbito dos detectores de partículas. Esse trabalho propõe um circuito eletrônico dedicado implementado em um sistema digital reconfigurável de alto desempenho para redução do empilhamento nos sinais medidos em um calorímetro. Para isso, são utilizados filtros digitais de resposta ao impulso finita implementados em paralelo. Cada filtro é uma aproximação da inversa da resposta ao impulso do canal de medição, realizando a desconvolução dos sinais medidos. Os resultados obtidos mostram a eficiência da técnica proposta e a adequação aos requisitos de tempo de processamento da aplicação.

DUARTE, J. P. B. S., HUFNAGEL, M., ANDRADE FILHO, L. M., LUZ, I. A. S., SIMAS FILHO, E. F., SEIXAS, J. M. **Estimação de Energia Online para o Tilecal: comparação entre os métodos linear e não-linear.** Encontro de Física, 2016, Natal. Anais do Encontro de Física 2016.

Resumo: A estimação de energia em calorímetros operando em ambientes de alta taxa de eventos pode sofrer com a sobreposição de informação devido à sensibilização do sistema de medição por eventos anteriores, esse fenômeno é conhecido como empilhamento de sinais (ou *pileup*). No calorímetro hadrônico do detector ATLAS (*Tile calorimeter* ou *Tilecal*), a ocorrência desse fenômeno se intensificará à medida que a luminosidade das colisões do LHC aumentar. O atual sistema de processamento de sinais do detector não está preparado para lidar com o *pileup*, o que pode ocasionar baixa eficiência para a detecção (*trigger*) de eventos. Para lidar com esse problema um novo sistema eletrônico (*readout driver*) baseado em *field-programmable gate arrays* (FPGA) está sendo projetado, permitindo a estimação de energia online antes do primeiro nível de *trigger*. Neste trabalho é proposta uma cadeia de processamento de sinais para ser embarcada nos *readout drivers* do *Tilecal* visando a redução do *pileup*. Para isso, está sendo utilizada a técnica da

de-convolução através de métodos lineares e não-lineares. A metodologia linear se baseia na aplicação de filtros digitais que aproximam o modelo inverso do canal de medição. Para modelar, também, não-linearidades existentes no problema, uma rede neural foi utilizada. Resultados com dados simulados possibilitam a comparação entre os métodos propostos considerando aspectos como a precisão na estimação e a complexidade computacional.

LUZ, I. A. S., BECK, M., SIMAS FILHO, E. F., FARIAS, P. C. M. A., ANDRADE FILHO, L. M., SEIXAS, J. M. **Implementação em FPGA do Sistema Digital para redução do Ruído de empilhamento dos Sinais do Calorímetro Tilecal do ATLAS**. Encontro Nacional de Física de Partículas e Campos, 2015, Caxambu. Anais do ENFPC 2015.

Resumo: O LHC está em processo de atualização, de modo a aumentar sua luminosidade e energia de colisões, acarretando numa maior taxa de eventos para serem detectados e processados. O aumento da ocorrência de eventos pode provocar a sobreposição da informação de eventos subsequentes (fenômeno conhecido como empilhamento ou *pileup*). Este trabalho apresenta a implementação do sistema digital de processamento de sinais para reduzir os efeitos do ruído de *pileup* no principal calorímetro hadrônico do ATLAS (o *TileCal*). Um conjunto de filtros digitais foi implementado em FPGA de alta performance para garantir o cumprimento dos requisitos de tempo de processamento das amostras. Os filtros utilizados são do tipo Resposta ao Impulso Finita (FIR *Finite Impulse Response*), com implementação na forma direta ou transposta. O projeto dos filtros foi baseado na técnica da desconvolução da resposta do calorímetro. Para estimação da inversa da função de transferência do canal de medição foi utilizado um estimador ótimo (no sentido da minimização do erro quadrático médio). O sistema digital projetado é flexível e pode utilizar filtros de diferentes ordens, variando-se também a resolução de quantização dos coeficientes. Para atender aos requisitos de tempo de processamento, foram utilizadas técnicas de *pipeline* e de múltiplos domínios de *clock* no projeto do *hardware*. Para o desenvolvimento do trabalho utilizou-se o Verilog como linguagem de descrição de hardware (HDL *Hardware Description Language*) e o System Verilog como linguagem para verificação do sistema. O FPGA utilizado para o projeto é da família Kintex-7, da Xilinx. Utilizando-se um conjunto de dados

simulados observou-se que o ruído de empilhamento é consideravelmente reduzido com a utilização dos filtros projetados. Um estudo a respeito do tempo de resposta do sistema proposto será apresentado.

1.3 Organização da Dissertação

No Capítulo 2 é contextualizada a Física de partículas através do seu histórico e alguns conceitos fundamentais para o entendimento da aplicação desse projeto. Além desse ponto, nesse capítulo é apresentado o LHC e seus subsistemas, com ênfase no detector ATLAS.

O Capítulo 3 aborda a atualização do LHC e seus efeitos no atual sistema eletrônico. Pontua, também, a atualização da eletrônica do *TileCal* para suportar o aumento da luminosidade e os métodos de estimação propostos para a recuperação dos sinais de energia.

O Capítulo 4 apresenta o sistema em FPGA desenvolvido nesse trabalho, descrevendo a arquitetura para ambas as soluções propostas e também a técnica de verificação funcional adotada. No Capítulo 5 são apresentados os resultados obtidos com a abordagem determinística e iterativa.

No Capítulo 6 são apresentadas as conclusões deste trabalho. O Apêndice A apresenta uma breve descrição do desenvolvimento de um projeto de *hardware* digital, expondo as características do dispositivo FPGA utilizado nesse trabalho. Por fim, no final do documento foi anexado o texto completo do artigo apresentado no CBA 2016.

Capítulo 2

Introdução à Física de Partículas e ao Acelerador LHC

2.1 Histórico da Física de Partículas

Desde o tempo dos filósofos gregos, pesquisa-se a respeito da composição da matéria, com a proposição de teorias que serviram de base para experimentos futuros. Na Grécia antiga, em torno de 600 a 400 a.C., acreditava-se que os elementos fundamentais eram fogo, água, terra e ar. Somente em 400 a.C, Leucipo e Demócrito, filósofos gregos, disseminaram a teoria atômica, ou do atomismo, a qual afirma que toda matéria é formada por elementos indivisíveis, por átomos [1]. Durante muito tempo, não foram realizadas pesquisas acerca desse tema, até que, em 1808, um pesquisador Inglês, chamado John Dalton, baseando-se na teoria de Leucipo e Demócrito, publicou um livro sobre a teoria atômica [15]. Nesse livro, Dalton definiu as características do primeiro modelo atômico: Toda matéria é composta por minúsculas partículas chamadas átomos; Todos os átomos de um mesmo elemento são idênticos; entre outras. Nesse estudo, John Dalton também definiu a lei das proporções definidas, afirmando que: “quando dois corpos A e B se combinam para formar um corpo C, as massas de A e de B necessárias para a reação estão numa razão definida” [15].

Além de Dalton, outros pesquisadores também propuseram modelos atômicos. Em 1897 o pesquisador J. J. Thomson abriu as portas para os estudos da Física de Partículas do século XX através do descobrimento do elétron. Essa descoberta

foi realizada através da análise do comportamento dos raios catódicos na presença de forças elétricas e magnéticas. Na Figura 2.1 apresenta-se o esquema do aparelho utilizado nesse experimento [15].

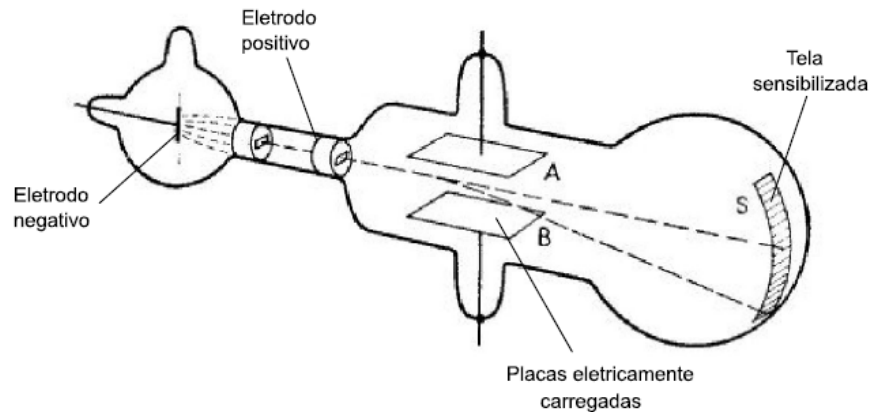


Figura 2.1: Esquema do aparelho usado para medir a deflexão elétrica e magnética dos raios catódicos [15].

A partir dessa descoberta, em 1904 Thomson apresenta um segundo modelo atômico. Nesse modelo, considera-se que o átomo neutro era constituído de uma esfera carregada positivamente e que possuía, também, elétrons. Esse modelo ficou popularmente conhecido como “Pudim de Ameixa”, do inglês “*Plum Pudding*” [15].

Em 1911, Rutherford apresentou um terceiro modelo atômico, que acabou substituindo o modelo de Thomson. A partir das conclusões do segundo modelo atômico, e realizando experimentos com feixes de partículas α sobre uma folha fina de ouro, Rutherford observou um comportamento diferente ao que havia sido definido por Thomson. Nesse novo modelo, o átomo é definido como consistindo de um núcleo carregado positivamente envolto por um sistema de elétrons, mantidos próximos uns dos outros [15]. No ano de 1913, o pesquisador Niels Bohr aperfeiçoou o modelo de Rutherford e definiu que os elétrons circulam em torno do núcleo numa órbita de acordo a um nível de energia do átomo.

A partir do terceiro modelo atômico, em 1919, Rutherford realizou um experimento onde partículas α colidiam com átomos de nitrogênio. A partir desse experimento, foi descoberta mais uma partícula elementar, o Próton [15].

Após a descoberta do próton, em 1923 Arthur Compton confirmou a existência da partícula chamada de Fóton [15], através de um experimento onde foi realizado

o espalhamento elástico de raios X em elétrons. Essa nova partícula pode ser interpretada como luz, sendo assim, é uma das mais notáveis partículas elementares, visto que está presente no dia-a-dia das pessoas.

Em 1932, Sir James Chadwick descobriu o nêutron. Essa partícula está diretamente relacionada às descobertas que foram realizadas durante a última década do século XIX [15]. A descoberta do nêutron foi estimulada a partir de um questionamento sobre o que mantém as cargas positivas confinadas em uma região espacial tão minúscula como a do núcleo atômico. Dessa forma, a partir de pesquisas da transmutação dos elementos com feixe de partículas α , repetindo o experimento com uma variedade de gases, pode-se, então, atestar a existência e também definir a massa do nêutron.

A partir das descobertas de elétrons, prótons e nêutrons, acreditava-se que esses eram os elementos fundamentais da constituição da matéria. Sendo os átomos constituídos pelos prótons e nêutrons como estrutura molecular e os elétrons como orbitais [15]. Contudo, ao longo dos anos novos estudos e experimentos foram realizados e constatou-se que os léptons e quarks são os verdadeiros elementos fundamentais, pois não possuem estrutura interna [16, 17]. Nesse sentido, foi constatado que os prótons e nêutrons são formados por quarks e que os elétrons são um tipo de lépton. Finalmente, chegou-se ao Modelo Padrão, o qual define as partículas quarks e léptons como constituintes fundamentais da matéria [16, 17].

Existem seis espécies de quarks conhecidos, são eles: u (*up*), d (*down*), c (*charm*), s (*strange*), b (*bottom*) e t (*top*) [18]. Cada uma dessas espécies pode assumir três cores: vermelho, verde e azul. Além disso, cada um possui sua antipartícula. Dessa forma, existem trinta e seis diferentes tipos de quarks [19]. Assim como os quarks, também existem seis tipos de léptons. São eles: elétron, múon, tau, neutrino do elétron, neutrino do múon e neutrino do tau [16]. Sendo que, cada um também possui sua antipartícula, resultando, assim, em um total de doze léptons.

A descoberta dos quarks ocorreu a partir das pesquisas acerca da constituição dos hádrons. Em 1964, os pesquisadores Gell-Mann e Zweig propuseram que todos os hádrons eram formados por elementos chamados quarks [20], sendo que suas partículas interagem com as outras através de interação forte [18]. Os hádrons são partículas que possuem estrutura interna e são classificadas como: bárions,

quando formados por três quarks ou três antipartícula de quarks; e mesóns, quando constituído por um quark e uma antipartícula de quark [21]. Dessa forma, seguindo esse modelo, os prótons são formados por dois quarks do tipo u e um do tipo d (uud) e os nêutrons por dois do tipo d e um do tipo u (udd) [20].

Apesar do *quark* compor a maior parte da massa observada no universo, os *léptons* não são compostos de *quarks*. Essas partículas não participam de interações fortes, mas sim das interações eletromagnéticas e fracas. As massas dos léptons são menores do que as dos quarks, sendo que os neutrinos caracterizam-se por serem os mais leves [18].

Além do conhecimento sobre as partículas, para se ter noção da constituição da matéria, torna-se necessário, também, o conhecimento acerca das interações entre elas. Existem quatro tipos de interações fundamentais: eletromagnética, gravitacional, nuclear forte e nuclear fraca. Cada uma delas está relacionada a uma propriedade fundamental: carga elétrica, massa, cor e carga fraca, respectivamente [17]. A primeira pode ser exemplificada como sendo a interação entre um elétron e um núcleo atômico; A segunda regula o movimento dos corpos celestes, ocorre entre todas as partículas; a terceira ocorre entre os quarks; a quarta, e última, ocorre, por exemplo, quando um nêutron decai para um próton, devido a emissão de um elétron ou neutrino [19].

O modelo padrão define, ainda, as partículas mediadoras, ou virtuais, responsáveis pela transmissão da mensagem da força entre as partículas interagentes [17]. Esses elementos são chamados de virtuais pois não possuem massa, tem apenas energia. Cada interação possui uma determinada partícula mediadora: na eletromagnética temos os fótons, na forte os glúons, W e Z na interação fraca e os grávitons na gravitacional [17].

Na Figura 2.2 é apresentado um esquema resumido do modelo padrão, considerando tanto as partículas elementares como as interações fundamentais.

2.2 O LHC

No cenário de estudo da física de partículas elementares, o acelerador de partículas apresenta-se como um equipamento fundamental para as pesquisas. Essa máquina

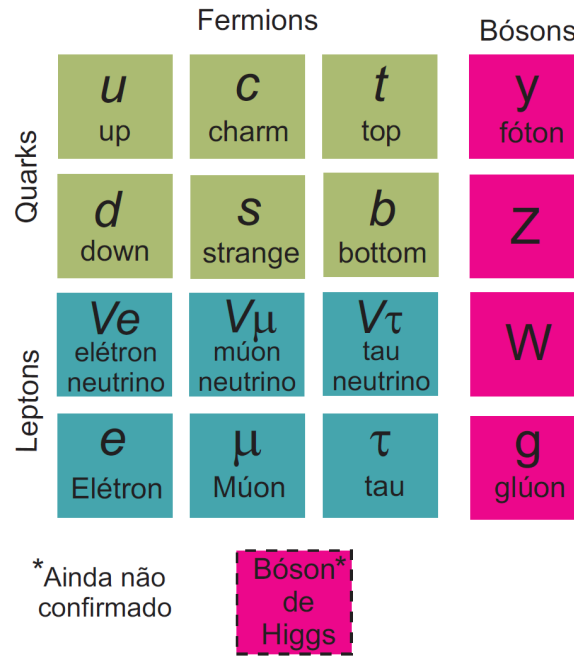


Figura 2.2: Diagrama do modelo padrão [22].

utiliza fontes de partículas de baixa energia e são responsáveis por aumentar a energia das partículas estáveis e carregadas eletricamente [22]. Os aceleradores, quanto às características construtivas, podem ser classificados como de alvo fixo ou colisores de feixes. Em relação ao percurso do feixe, podem ser classificados como lineares ou circulares [23]. As pesquisas envolvendo aceleradores de partículas normalmente requerem cooperação entre diversos pesquisadores de várias instituições do mundo, tornando-se experimentos de grandes proporções.

Os experimentos envolvendo aceleradores de partículas são compostos de diversos equipamentos, dentre eles os detectores de partículas. Os detectores precisam ter alta precisão para que as partículas detectadas possam ser adequadamente estudadas [23].

Para alcançar a precisão necessária nos experimentos, foram desenvolvidos diferentes tipos de detectores, sendo cada um específico para identificar determinadas características das partículas. São alguns exemplos: Detector a gás, responsável por detectar a ionização produzida pela passagem de uma partícula eletricamente carregada por um determinado gás; Contador de cintilação, responsável por identificar a perda de energia que é convertida em luz e convertê-la em impulso elétrico; Calorímetro, usado para medir a energia e a posição da partícula através da absorção total de sua energia [23].

Existem grandes laboratórios que realizam pesquisas na área da física de partículas, dentre eles destacam-se: CERN [24], DESY [25], KEK [26], Fermilab [27], SLAC [28] e Brookhaven [29].

O CERN (Centro Europeu para Pesquisas Nucleares) apresenta-se como um dos maiores centros de pesquisas na área de física de partículas. Criado em 1954, localizado em Genebra, na fronteira Franco-Suíça, esse centro de pesquisas vem desenvolvendo experimentos fundamentais no que diz respeito à descoberta dos constituintes da matéria [30]. No CERN foi projetado e desenvolvido o LHC, o maior acelerador de partículas já construído, tendo suas atividades iniciadas em 2008 [3, 31]. Esse acelerador tem a capacidade de gerar energia de colisão de até 14 TeV¹ e possui um percurso circular de aproximadamente 27 km de comprimento e 100 m de profundidade [22]. Na Figura 2.3 é mostrado o túnel do LHC.



Figura 2.3: Fotografia do túnel do LHC [32].

O LHC tem como objetivo provocar colisões próton-próton a uma velocidade igual a 99,9999% da velocidade da luz, para tentar recriar o ambiente frações de segundos após o *big bang* [32]. As colisões próton-próton tem uma luminosidade projetada de $10^{34} \text{cm}^{-2} \text{s}^{-1}$ enquanto que as colisões de íons pesados tem picos de luminosidade acima de $10^{27} \text{cm}^{-2} \text{s}^{-1}$ [33].

A luminosidade gerada nesse experimento é definida pela equação 2.1, onde n representa o número de feixes de partículas, N_1 e N_2 o número de partículas por

¹O Elétron-volt é uma unidade de medida de energia que representa a quantidade de energia cinética de uma partícula. TeV representa Tera Elétron-volt e corresponde a 10^{12} elétrons-volt.

feixe, A a área da seção transversal do feixe e f a frequência de colisão.

$$L = n \frac{N_1 N_2}{A} f \quad (2.1)$$

A frequência de colisões nesse acelerador é de 40 MHz, sendo que, a cada colisão espera-se que aconteçam 25 choques próton-próton, dessa forma, a taxa de interações será da ordem de 1 GHz [22].

O LHC é composto por 6 detectores, e estes tem a função de selecionar, a partir de um grande conjunto de eventos, aqueles de real interesse. Os detectores do LHC são: ATLAS [4], ALICE [34], CMS [35], LHCb [36], LHCf [37] e TOTEM [38]. Na Figura 2.4 é ilustrada a distribuição física dos quatro maiores detectores ao longo do percurso do LHC.

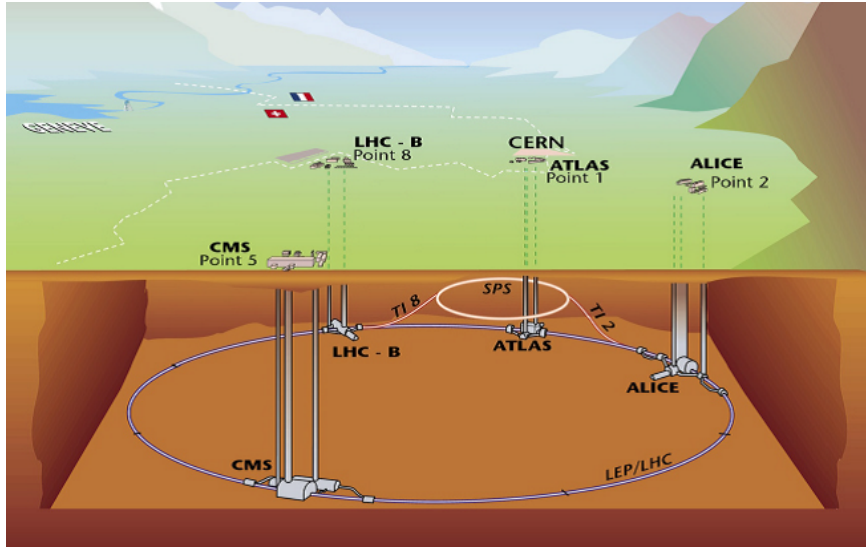


Figura 2.4: Ilustração da localização dos quatro maiores detectores do LHC ao longo do percurso do acelerador [22].

2.3 O Detector ATLAS

O ATLAS é um detector de propósito geral com formato cilíndrico e capacidade de cobrir um ângulo sólido próximo a 4π ao redor da região de colisão [22]. Tem como função detectar os resultados das colisões próton-próton (pp) [32] e pode ser considerado um detector de camadas, pois é composto por sub-detectores. Detector de Camadas é a definição atribuída para equipamentos que precisam extrair

o máximo de informações das partículas com alta precisão. Dessa forma, têm a necessidade de abrigar sistemas específicos para cada característica que se deseja identificar (energia, carga elétrica, momento, etc.).

Dentre as características do detector ATLAS, destaca-se: sistema de calorimetria eletromagnética para medição e identificação de elétrons e fótons com alta performance; sistema de calorimetria hadrônica com total cobertura para medição precisa de energia transversa; alta precisão na medição do momento de múons; alta eficiência no sistema de medição e seleção das partículas, entre outras [39].

O ATLAS tem aproximadamente 44 metros de comprimento, 25 metros de altura e 7.000 toneladas, sendo assim o maior detector de propósito geral do LHC. Na Figura 2.5 é ilustrada a composição do ATLAS. Dentre os sub-detectores ilustrados na Figura 2.5, os principais são: detector de traços, detector de múons, calorímetro eletromagnético e o calorímetro hadrônico.

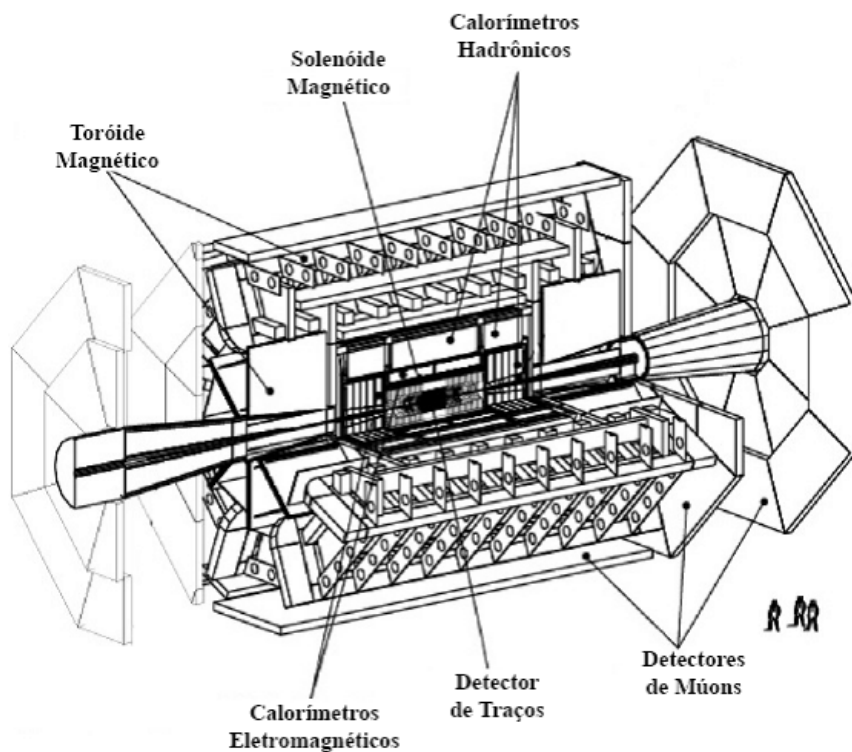


Figura 2.5: Esquemático dos detectores do ATLAS [22].

O detector de traços é responsável pela identificação do momento e do vértice das partículas carregadas geradas a cada colisão [4, 40]. Devido à alta frequência de ocorrência dos eventos, para garantir a medição com a alta precisão requerida, foi dividido em três sub-sistemas, dois utilizando semicondutores com a tecnologia

planar e outro uma câmara de tração preenchida com gás, também conhecida como detector gasoso [40, 41]. Os calorímetros têm a função de medir a energia das partículas e a posição de interação no detector [22, 42]. Por fim, o detector de múons contém o sistema de detecção para essas partículas [39, 43], que interagem pouco com o restante do detector.

Na Figura 2.6 é apresentado o sistema de coordenadas do ATLAS. Os feixes das partículas seguem o eixo z . Os eixos x e y formam um plano transversal à direção dos feixes de partículas. O ângulo azimutal ϕ é obtido através da equação 2.2 [22]. A direção de projeção das partículas após a colisão é representada por η , que é chamada de pseudo-rapidez e definida na equação 2.3 [22].

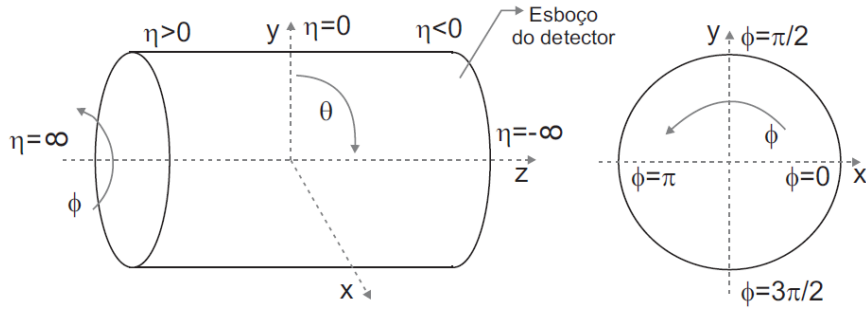


Figura 2.6: Eixo de coordenadas do ATLAS [22].

$$\phi = \arctan\left(\frac{x}{y}\right) \quad (2.2)$$

$$\eta = -\ln \tan\left(\frac{\theta}{2}\right) \quad (2.3)$$

2.3.1 A Calorimetria do ATLAS

A calorimetria é uma área da física que trata da troca de energia entre elementos. Inicialmente, os estudos nesse campo estavam voltados para fenômenos relacionados aos raios cósmicos. Após aperfeiçoamentos, a calorimetria passou a ser utilizada nos experimentos da física de partículas baseados em aceleradores [44].

Calorímetros [11] são dispositivos utilizados para medir a energia das partículas, como elétrons, próton e hádróns [44]. À medida que interagem com o material que compõe o calorímetro, as partículas vão perdendo energia (e consequentemente, velocidade) e ao final são totalmente absorvidas, sendo transformadas em energia

mensurável pelo sistema eletrônico de medição. Esses equipamentos são utilizados em detectores e precisam ter alta precisão, para, assim, serem capazes de identificar o evento ao qual as partículas são associadas [23].

As colisões que ocorrem na área do calorímetro geram um efeito conhecido como chuva de partículas carregadas, que tem sua energia absorvida ao longo da sua trajetória nas diversas camadas sobrepostas do calorímetro. Essa energia depositada na área ativa dos sensores é detectada sob a forma de carga elétrica ou luz. Efetuando a soma das medições de energia em todos os sensores do calorímetro, obtêm-se uma estimativa da energia total das partículas que colidiram [44]. Esses dispositivos podem ser classificados em dois tipos: calorímetro eletromagnético e hadrônico. O primeiro é responsável por medir as interações eletromagnéticas de elétrons e fótons, enquanto que o segundo as interações eletromagnética e forte dos elementos hadrônicos [44].

Dentre as características que tornam os calorímetros elementos importantes no estudo da física de partículas, destaca-se [44]:

- Calorímetros são sensíveis a quase todos os tipos de partículas, tanto carregadas, como neutras;
- São adaptados para experimentos de altas energias;
- Além de serem utilizados para medição de energia, podem funcionar, também, para detectar a posição, direção e o tempo de chegada da partícula.

O conjunto dos calorímetros do ATLAS é dividido em 7 camadas, 4 referentes ao calorímetro eletromagnético e 3 aos calorímetro hadrônico [22]. Na Figura 2.7 é ilustrado um corte transversal no conjunto de calorímetros, apresentando suas camadas. Na Figura 2.8 é ilustrada a disposição dos calorímetros do ATLAS.

As camadas eletromagnéticas são o pré amostrador (PS *Pré-Sampler*), E1, E2 e E3. As camadas hadrônicas são H1, H2 e H3. Cada uma dessas camadas apresenta diferente granularidade, ou seja, concentração de células detectoras por unidade área. O *Pré-Sampler*, localizado na região onde $|\eta| < 1,8$, é responsável por corrigir medições quando ocorre perda de energia no caminho até os calorímetros [39]. O calorímetro eletromagnético é constituído de finas folhas de chumbo separadas por sensores de argônio líquido [22] e cobre a região onde $|\eta| < 3,2$ [39]. Suas camadas

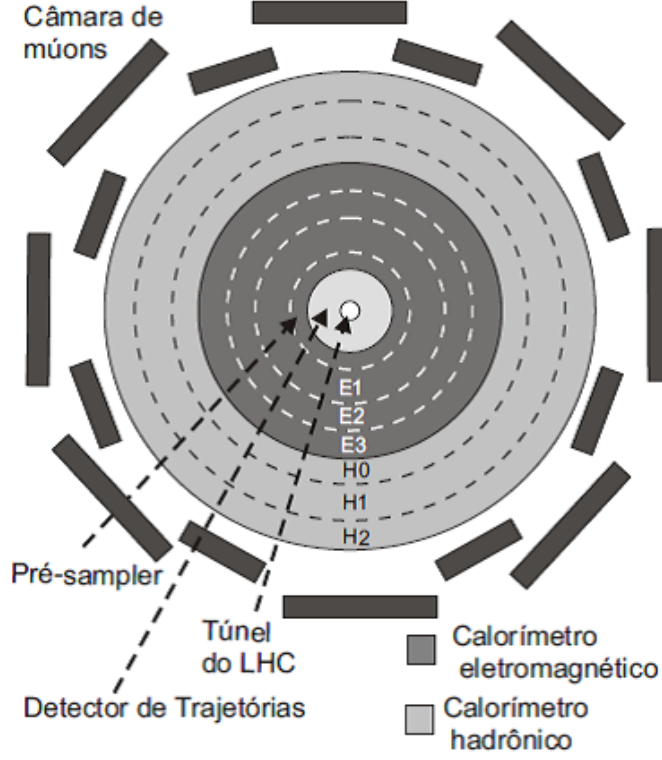


Figura 2.7: Corte transversal do ATLAS [22].

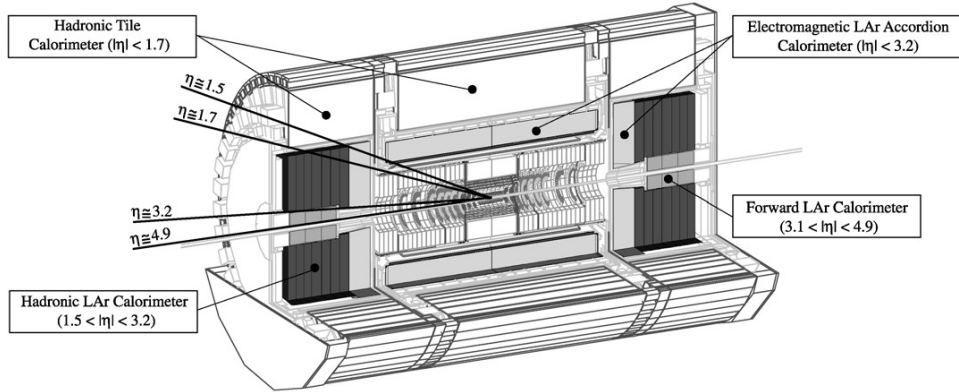


Figura 2.8: Diagrama dos calorímetros eletromagnético e hadrônico do ATLAS [42].

são divididas em uma parte central chamada barril, região onde $|\eta| < 1,475$, e duas tampas finais, onde $1,375 < |\eta| < 3,2$ [39, 42].

O calorímetro hadrônico é constituído por placas absorvedoras de aço separadas por telhas de material plástico cintilante, cobrindo a região onde $|\eta| < 1,7$. Devido a essa estrutura, também é conhecido como calorímetro de telhas (*Tile Calorimeter* ou *TileCal*). As partículas emitem luz ao passar pelas telhas cintilantes, sendo essa de intensidade proporcional à sua energia incidente [45]. Sendo assim, através de sensores chamados de tubos fotomultiplicadores (PMT, do inglês *Photomultiplier*

Tubes), a luz é transformada em sinal elétrico. Esse calorímetro tem a estrutura de um cilindro, sendo dividido em três partes: barril ($|\eta| < 1,0$) e dois barris estendidos ($0,8 < |\eta| < 1,7$) [46, 39]. Na Figura 2.9 é ilustrado o formato do *TileCal*.

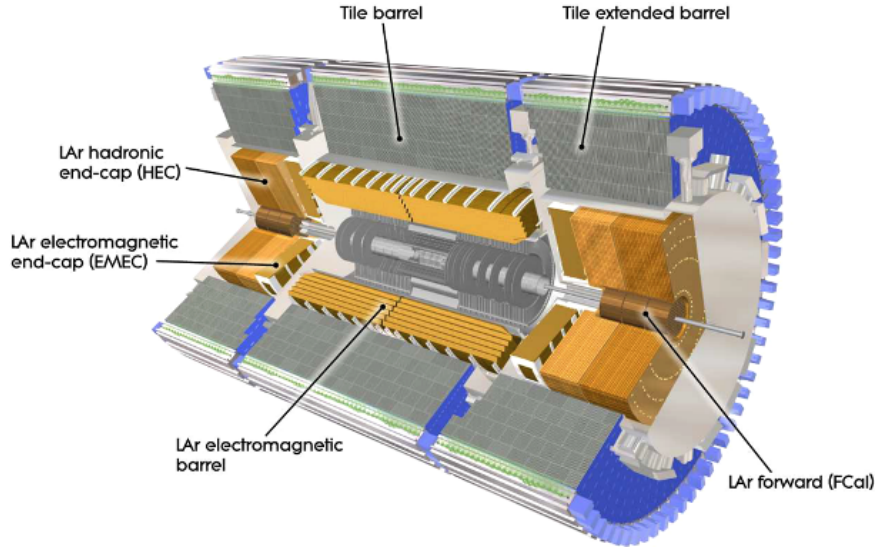


Figura 2.9: Ilustração do formato do *TileCal* [45].

As informações de energia das partículas detectadas pelos calorímetros são importantes, pois quando um evento é selecionado, ou seja, atende aos requisitos de interesse do processo de seleção online (*Trigger*), as informações são armazenadas para futuras análises *off-line*.

No Calorímetro do ATLAS, a granularidade das células detectoras é diferente em cada camada. Na Figura 2.10 é ilustrada essa distribuição, para uma seção do *TileCal*. Observa-se que a primeira camada, de A1 até A10, apresentam mais sensores, sendo assim, apresenta uma maior precisão na identificação da posição de interação da partícula com o detector. A segunda camada, de BC1 até BC8 e B9, apresenta menos células sendo estas com maior profundidade e, sendo assim, absorvendo maior parcela de energia. A terceira e última camada, de D0 até D3, possui ainda menos sensores e são responsáveis por capturar os detalhes finais do “chuveiro” de energia [22]. As linhas tracejadas na Figura 2.10 ilustram as possíveis trajetórias da partícula em relação às camadas.

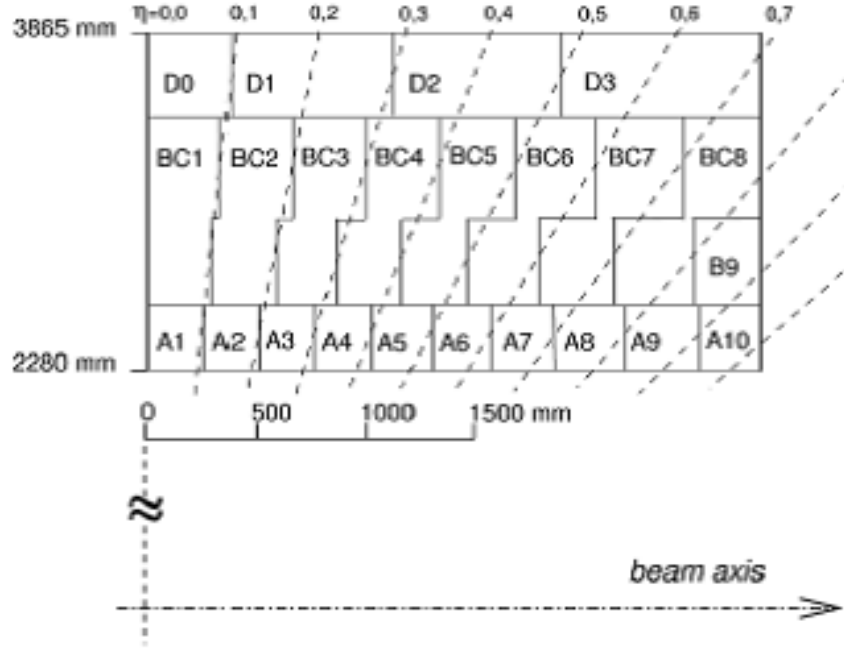


Figura 2.10: Exemplo de Segmentação de uma parte do TileCal [47].

2.3.2 O Sistema Online de Seleção de Eventos (*Trigger*) do ATLAS

O ATLAS possui um sistema online de seleção de eventos (*trigger*), tendo como função selecionar os eventos de interesse produzidos pelas colisões do LHC. Esse sistema de *trigger* é composto por três níveis sequenciais e acessa as informações de três sub-detectores: o detector de trajetória, os calorímetros e a câmara de múons. Na Figura 2.11 é ilustrado esse sistema.

Cada nível do sistema de filtragem é responsável por descartar as assinaturas não relevantes, passando para o próximo nível, apenas as prováveis assinaturas de interesse [22]. O primeiro nível de seleção, L1, baseia-se na redução da granularidade das informações, tanto dos calorímetros, como do detector de múons e tem como principal função identificar os eventos de interesse [39]. Na Figura 2.12 é apresentado o diagrama de blocos contendo o fluxo dos sinais durante o processo de decisão desse nível. Nessa figura, observa-se que a análise dos eventos de interesse é realizada por um processador central de seleção (*Central Trigger Processor*) e o caminho das informações para o *Front-end* do detector, para o nível 2 e para o sistema de aquisição de sinais, estão destacados nas cores vermelha, azul e preta, respectivamente.

O nível L1 de seleção recebe todas as amostras geradas pelas colisões, a uma taxa

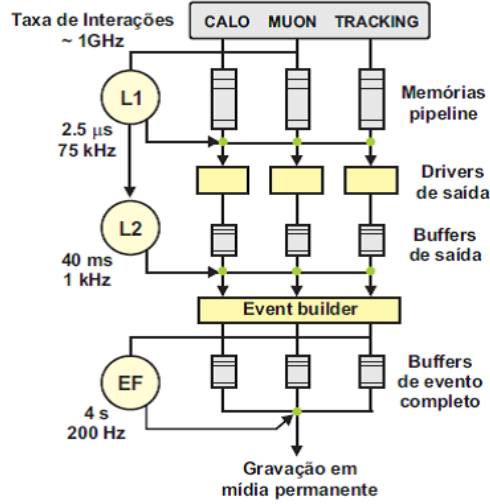


Figura 2.11: Sistema de filtragem do ATLAS [22].

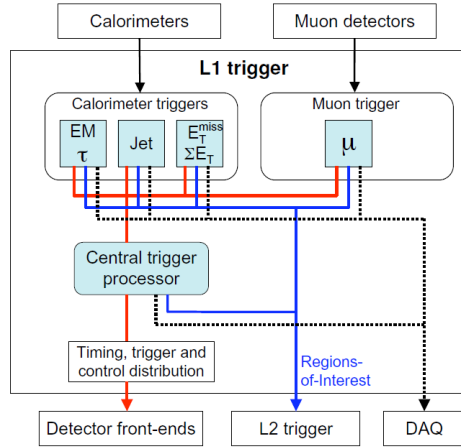


Figura 2.12: Diagrama de Blocos do Nível 1 do *Trigger* [4].

de 40 MHz, gerando uma taxa de interações da ordem de 1 GHz, sendo responsável por reduzir de 40 MHz para 75 kHz [39]. Dessa forma, o L1 possui uma latência máxima de até $2,5 \mu s$ após a colisão. Logo, devido a necessidade de uma alta performance, esse nível foi implementado em hardware dedicado e para atender a restrição de tempo, utiliza-se apenas de parte da resolução disponível no detector. Como ilustrado na Figura 2.11, no L1 existe um *pipeline* de memória, e nessas memórias ficam armazenadas todas as informações do detector referentes ao evento até que a seleção ou rejeição dos dados seja concluída.

O nível L2 já não possui a mesma restrição de tempo de processamento do nível L1, ou seja, lida com uma taxa menor de eventos e possui um tempo de resposta maior. Nessa etapa, a taxa de eventos é reduzida de 75 kHz para 1 kHz.

Com isso, esse nível de seleção possui critérios mais rigorosos de seleção e também utiliza-se da resolução máxima dos calorímetros. Esse nível é executado por cerca de 700 computadores através de software especializado. O segundo nível executa a seleção baseada nas informações de região de interesse disponibilizadas pelo primeiro nível, tendo acesso às informações com máxima granularidade e precisão, mesmo que seu algoritmo normalmente utilize apenas informações de regiões centrais das informações do L1 [39].

O nível L3, conhecido como *Event Filter* (EF) [4], é o último nível de seleção do sistema de *Trigger*. Esse nível utiliza-se de todas as informações não consideradas pelos outros níveis, dos eventos de interesse selecionados. Esse nível opera em software, utilizando-se de algoritmos executados por cerca de 2000 computadores em modo *off-line*. Os eventos selecionados nesse nível são armazenados definitivamente em mídia permanente. Nessa fase, a taxa de eventos é reduzida para cerca de 200 Hz [4].

Capítulo 3

Atualização do LHC e os Métodos de Estimação de Energia

3.1 Introdução

O LHC vem passando ao longo dos anos por fases de atualização. No que diz respeito ao sistema eletrônico de instrumentação dos calorímetros, é necessário que seja modificado para suportar o aumento da luminosidade. A fase 0 de atualização ocorreu entre os anos 2013 e 2014, tendo como objetivo alcançar uma luminosidade de aproximadamente $10^{34} \text{cm}^2 \text{s}^{-1}$ [5, 48]. Uma próxima parada do detector está prevista para o período entre 2018 e 2019, com o objetivo de preparar o sistema eletrônico para uma luminosidade de aproximadamente $2 - 3 \times 10^{34} \text{cm}^2 \text{s}^{-1}$, chamada de fase 1 [5, 48]. Por último, a fase 2, prevista para ocorrer entre 2022 e 2023, tem como objetivo alcançar uma luminosidade de $5 - 7 \times 10^{34} \text{cm}^2 \text{s}^{-1}$ [5, 48].

Através do aumento da luminosidade, esse programa de atualizações permitirá o avanço dos experimentos no âmbito da Física de Partículas. Para viabilizar esse cenário, torna-se necessário realizar modificações no sistema eletrônico do *TileCal*. Dentre os principais objetivos da atualização na eletrônica do *TileCal*, destaca-se: melhorar o desempenho do nível 1 do *trigger*, aumentar a confiabilidade, melhorar a precisão das informações e a relação sinal-ruído (SNR, do inglês *Signal-to-Noise Ratio*) [5].

Para alcançar o desempenho desejado, o sistema de seleção de eventos precisa ter acesso a mais informações e, conseqüentemente, ter maior capacidade de proces-

samento para a execução de algoritmos mais complexos [5]. Esse novo cenário de operação do ATLAS, considerando a atual arquitetura de funcionamento, provocará um problema no sistema de seleção de eventos: o empilhamento de informações subsequentes, também conhecido como *pileup*. Esse problema irá afetar, diretamente, o funcionamento do sistema de seleção, pois provocará a distorção do padrão dos eventos. O problema de empilhamento ocorre devido ao tempo de resposta dos sensores do *Tilecal*. Como pode ser observado na Figura 3.1, o pulso típico desses sensores tem duração da ordem de 150 ns.

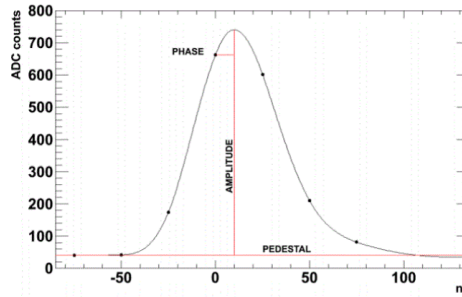


Figura 3.1: Representação gráfica da Resposta ao Impulso Típica de um sensor [47].

A frequência de ocorrência de colisões entre pacotes no LHC é de 40 MHz. Dessa forma, o intervalo entre duas colisões é de 25 ns. Um mesmo sensor permanece sensibilizado por até 6 colisões em sequência. Na Figura 3.2 é ilustrado (através de simulação) o efeito do empilhamento em um sensor do *Tilecal*.

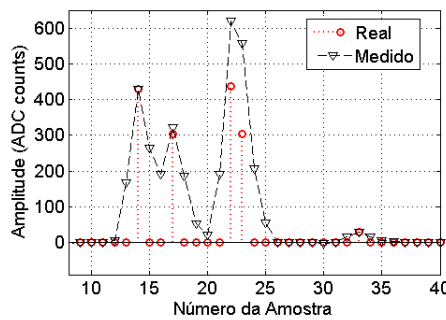


Figura 3.2: Representação gráfica do efeito de empilhamento de sinais medidos [47].

Para alcançar os objetivos do programa de atualização do LHC, a colaboração do ATLAS propôs alterações no nível 1 do sistema de *trigger* do detector. Na configuração atual, o nível 1 é alimentado por sinais resultantes de somadores analógicos, os quais têm como entradas as informações dos tubos fotomultiplicadores. Esses

somadores são responsáveis por compactar a informação, reduzindo o tempo de resposta. A alteração propõe que o nível 1 de decisão utilize todas as amostras de cada uma das PMTs, com isso, aumentando a eficiência de seleção dos eventos de interesse. Na Figura 3.3 é ilustrada a arquitetura atual do nível L1, baseado em calorimetria, e na Figura 3.4 a nova arquitetura proposta.

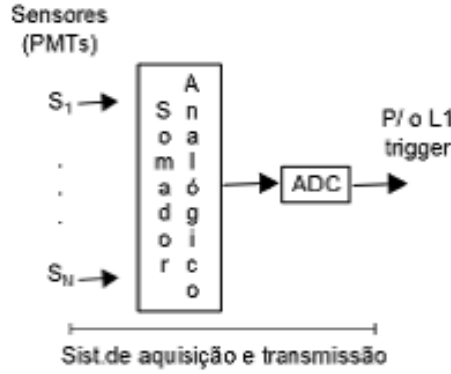


Figura 3.3: Arquitetura do *Trigger* L1 baseado em calorimetria atual [47].

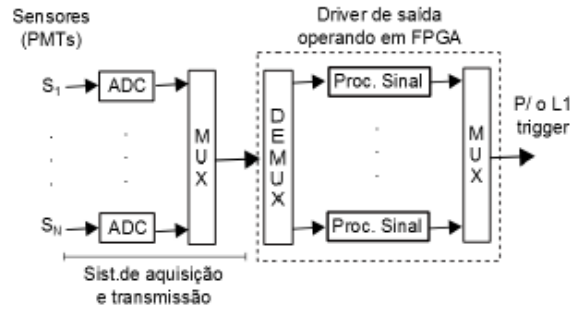


Figura 3.4: Arquitetura do *Trigger* L1 baseado em calorimetria em desenvolvimento [47].

Devido às novas fases de operação, além da modificação do sistema eletrônico, torna-se necessário remodelar o processo de filtragem do sinal para o nível 1 de *trigger*. A técnica utilizada atualmente, o Filtro Casado, não é eficiente considerando o novo cenário de operação devido a distorção do sinal. Com isso, uma alternativa para a técnica de estimação de energia precisa ser desenvolvida e embarcada na nova eletrônica do detector.

3.2 Sistema Eletrônico Atual do Primeiro Nível de *Trigger* Baseado em Calorimetria

O sistema que compõe o Calorímetro Hadrônico do ATLAS, o *TileCal*, é dividido em uma parte óptica, uma eletrônica de *Front-End* (FE) e de digitalização e, por fim, uma eletrônica de *back-end*. A primeira é responsável por transmitir a luz depositada pelas partículas durante o seu trajeto para as Fotomultiplicadoras, onde são transformadas em pulsos elétricos [32]. As energias depositadas pelas partículas são absorvidas pelas telhas cintilantes e propagadas até suas bordas, onde são absorvidas e transportadas através de uma fibra, composta por material fotofluorescente, até a PMT. Nesse sistema, ilustrado na Figura 3.5, os fótons produzidos em cada telha é enviada para duas PMTs, garantindo a redundância do sinal, enquanto que as fibras de diferentes telhas são conectadas a uma PMT [32].

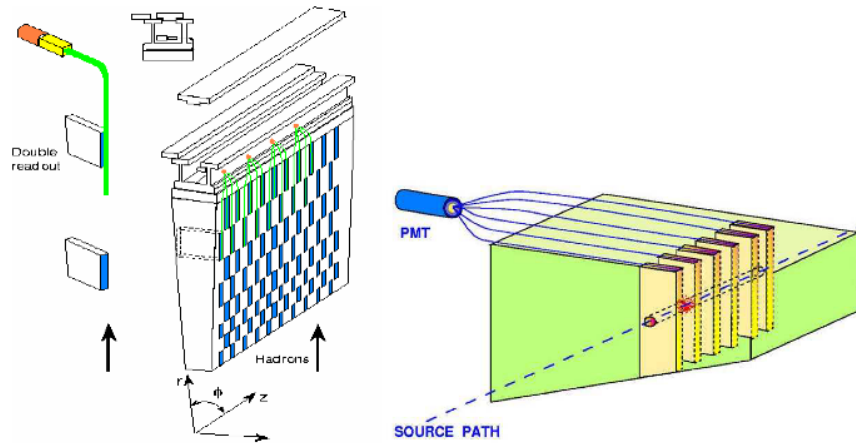


Figura 3.5: Representação gráfica do módulo do *TileCal* [32].

A eletrônica de *front-end* e de digitalização está localizada em uma estrutura de gavetas removíveis, com a finalidade de facilitar sua montagem e manutenção. Duas gavetas são interligadas, formando uma “super-gaveta”, sendo que as localizadas no barril central são conectadas a 45 PMTs, enquanto as localizadas nos barris estendidos, a 32 PMTs.

Como pode ser observado na Figura 3.6, as gavetas do sistema de *front-end* e de digitalização são compostas por cinco elementos, são eles: Fotomultiplicadora (PMT), Placa Mãe (*Motherboard*), Digitalizador (*Digitizer*), Unidade de gerenciamento de informações (DMU) e Placa de Interface (*Interface Board*).

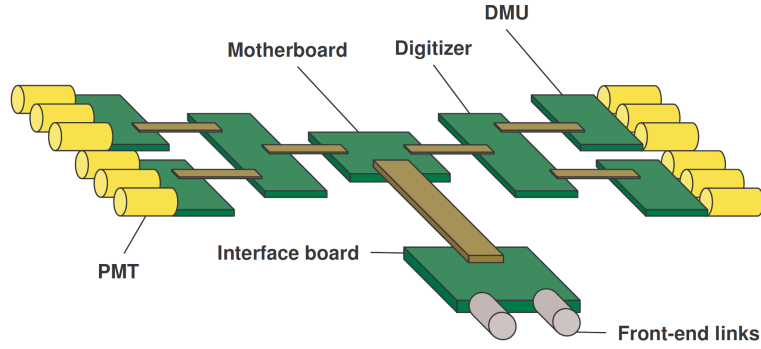


Figura 3.6: Esquemático da eletrônica de uma gaveta do *TileCal* [32].

Cada gaveta é composta por seis fotomultiplicadoras organizadas em grupos de 3. Esses dispositivos são divididos em quatro partes: misturador de luz, do inglês *mixer*; um tubo fotomultiplicador, a PMT; divisor de alta voltagem, *High Voltage Divider*; e o Cartão 3 em 1 [32]. O *mixer* é responsável por misturar a luz proveniente de todas as fibras de uma célula, a PMT realiza a conversão do sinal de luz em elétrico, o divisor particiona as altas voltagens entre os dínodos da PMT, e, por fim, o cartão 3 em 1 possui três funcionalidades: disponibilizar pulsos com ganho baixo e alto para a digitalizadora, carregar a injeção de sinais para calibração e realizar a integração lenta dos sinais da PMT para calibração e monitoramento.

A *Motherboard* conecta-se com as PMTs, com o Digitalizador e com a placa de interface através de três camadas [32]. A mais inferior garante a transmissão de energia de baixa tensão e sinal de controle para as PMTs, em uma camada intermediária é realizada a transmissão dos sinais analógicos das PMTs para o Digitalizador. Por fim, uma última camada é formada pelo circuito de soma e os cabos de saída do nível 1 do *Trigger*.

O Digitalizador é responsável por amostrar, converter e armazenar as informações de seis PMTs [32]. O processo de amostragem é realizado através de Conversores Analógico Digital, do inglês *Analog Digital Converter* (ADC), com resolução de 10 bits. Esses sinais digitalizados são armazenados nas duas DMUs. Além desses componentes, o equipamento possui um circuito integrado responsável pelo sistema de sincronização e programação, atuando na distribuição do *clock* juntamente com as informações de sincronização e controle para a eletrônica do *Front-End*. Os sinais digitais ficam armazenados na unidade de gerenciamento de informações a espera da

decisão do nível 1 do sistema de *Trigger* e, quando selecionados, são disponibilizados para os *buffers* de saída [32].

Dentre outros dispositivos, os *Drivers* de Saída, *Read-out Driver* (ROD), são os elementos centrais da eletrônica de *Back-End* [49]. Como pode ser observado na Figura 2.11, esses elementos ficam localizados entre o primeiro e o segundo nível do sistema de *Trigger*. As informações de energia produzidas pelo ATLAS são identificadas e digitalizadas pelo sistema de *Front-end* e então enviadas para esses *drivers* [32]. Nesses dispositivos, as informações dos módulos do calorímetro, enviadas a uma taxa de 100 kHz, são processadas e transmitidas para o nível 2 [32, 49].

Na Figura 3.7 é possível identificar o caminho do sinal na arquitetura atual do sistema eletrônico do ATLAS no nível 1. Nessa arquitetura, os sinais do primeiro nível são produzidos por somas analógicas de seis sinais de energia, tendo suas saídas transmitidas para o nível 1. Quando ocorre um evento físico, gera-se um sinal óptico no calorímetro, que é transmitido para a eletrônica de *Front-End*. Nessa etapa, o *mixer* realiza a mistura das informações das fibras de uma célula, que é enviada para a PMT. Esses sinais são processados pelo cartão 3 em 1, os quais são digitalizados a 40 MHz por ADCs de 10 bits. Por fim, os sinais digitalizados das células de um evento selecionado pelo primeiro nível são transmitidos para a eletrônica de *Back-End*.

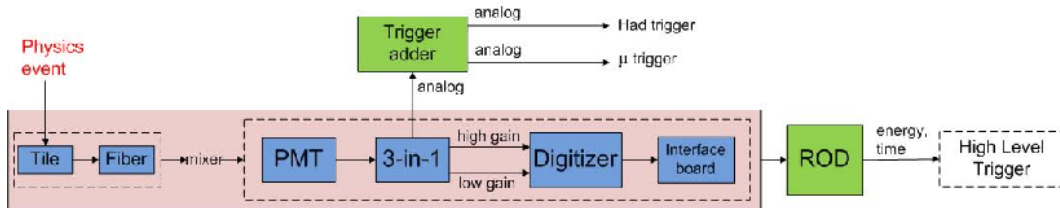


Figura 3.7: Representação gráfica do caminho do sinal na arquitetura atual do *TileCal* [5].

O nível 1 do sistema de *Trigger* do *TileCal* é alimentado com os sinais analógicos provenientes das estruturas de soma analógica, ou seja, das torres de *trigger*. Esses dispositivos são constituídos de somadores de alto desempenho e um circuito adicional de linha de condução [50]. As informações resultantes desse sistema, que representam combinações das camadas de segmentação do calorímetro, contribuem com a redução da quantidade de sinais processados nesta etapa, reduzindo para uma

taxa de 75 kHz [50]. Sendo assim, esse nível define as regiões de interesse a partir de combinações de sinais, ou seja, com reduzida granularidade [50].

3.3 Sistema Eletrônico que será Utilizado após a Atualização do LHC

O programa de atualização do *TileCal* propõe modificações em todo o sistema eletrônico do detector, tanto no *Front-end* como no *Back-end*, inclusive estuda-se realizar alterações também em seu sistema mecânico, além das atualizações necessárias no sistema de fontes de alimentação [5]. Na Figura 3.8 pode-se observar o caminho da informação na proposta de atualização dos sistemas eletrônicos do *TileCal*.

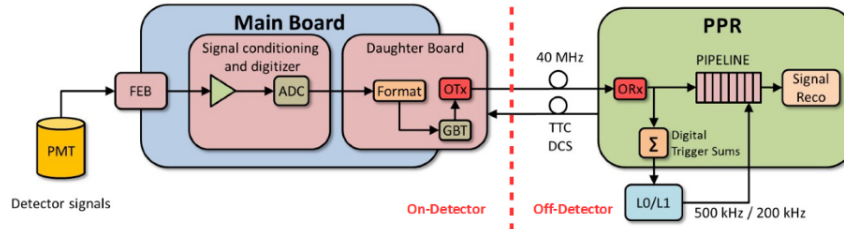


Figura 3.8: Representação gráfica do caminho da informação na atualização dos sistemas eletrônicos do *TileCal* [51]

Através da Figura 3.8 pode-se observar que a eletrônica embarcada no detector, representada pela região *On-Detector*, é composta por: Uma nova placa de *Front-End*, uma placa principal, do inglês *Main Board* (MB) e uma placa filha, do inglês *Daughter Board* (DB). A primeira é responsável, entre outras funções, pelo condicionamento e amplificação dos sinais. A MB realiza a digitalização, enquanto que a DB processa a informação e transmite para a eletrônica não embarcada do detector, representa pela região *Off-Detector* [5, 52, 51, 53].

O grupo de pesquisadores do ATLAS está avaliando o desenvolvimento de três diferentes arquiteturas para a eletrônica do FE, denominadas: *Front-End-Modified 3-in-1 card*, *Front-end-QIE chip* e a *Front-End-FATALIC ASISC* [52, 54]. A primeira propõe uma modificação na estrutura do Cartão 3 em 1 presente na arquitetura atual do FE e apresenta-se como uma solução que utiliza elementos discretos, proporcionando melhor linearidade e menor ruído [53]. Baseado na Figura 3.9 pode-se

observar que nessa proposta de arquitetura o sinal analógico da PMT é moldado e seus ganhos, tanto o alto como o baixo, são amplificados com uma taxa de 1:32, digitalizados com 12 bits e amostrados a uma frequência de 40 MHz [55].

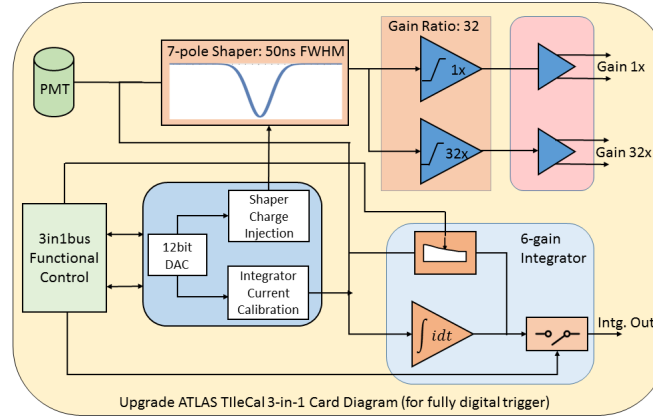


Figura 3.9: Protótipo da Proposta de Arquitetura de Modificação do Front-End Baseado no Cartão 3 em 1 [53].

As outras duas propostas de arquitetura para o *Front-End* utilizam circuitos integrados dedicados. Na solução baseada no chip *QIE*, que representa Carga, Integrador e Codificador, do inglês *Charge, Integrator and Encoder*, não realiza-se a modelagem do pulso, reduzindo o problema do efeito de sobreposição e permitindo que os pulsos da PMT sejam medidos [53]. Como pode ser observado na Figura 3.10, o sinal de entrada passa por um divisor de corrente que divide a informação em quatro ganhos. Cada informação então será processada por um integrador e digitalizada por estruturas de ADCs de 6 bits.

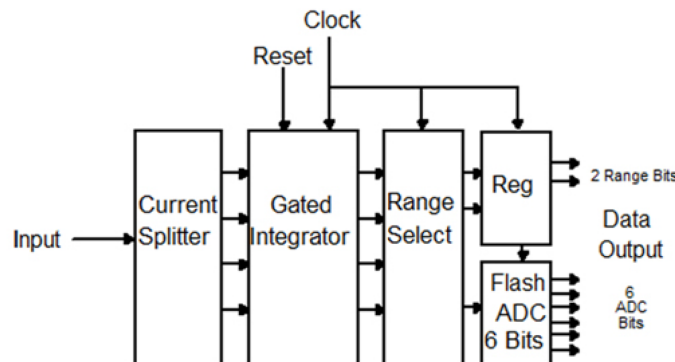


Figura 3.10: Diagrama do Fluxo do Sinal do *Front-End* Baseado no Chip *QIE* [52].

A terceira opção de arquitetura para o sistema eletrônico do *Front-End* também baseia-se na utilização de um chip dedicado, um Circuito Integrado de Aplicação Es-

pecífica (ASIC, do inglês *Application Specific Integrated Circuit*). Através da Figura 3.11 pode-se observar um diagrama básico deste circuito. Denominada de *FATALIC*, abreviação para *Front-end for Atlas TileCal Integrated Circuit*, essa proposta utiliza um transportador de corrente para três diferentes faixas de ganho (1x, 8x e 64x). Para cada ganho, o sinal será digitalizado por ADC de 12 bits [53].

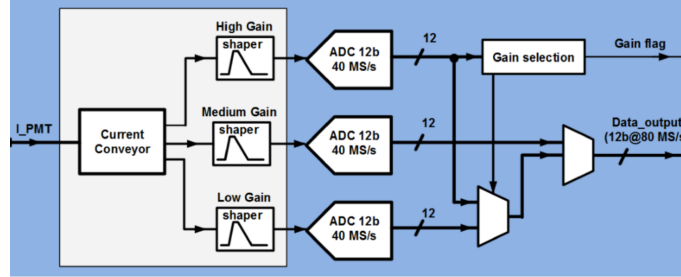


Figura 3.11: Diagrama do Fluxo do Sinal do *Front-End* Baseado no Chip *FATALIC* [52].

Em relação a eletrônica de *Back-End*, representada pela região *Off-Board* na Figura 3.8, uma nova *Read-Out Driver* (*ROD*) está sendo desenvolvida, chamada de “Super” *Read-Out Driver* (*sROD*) [5, 48]. Dentre as funções dessa eletrônica de componentes não embarcados no detector, destaca-se: processamento e recepção da informação proveniente das placas filhas (*Daughter Boards*); Reconstrução e envio dos sinais para o sub-sistema de saída (do inglês *ReadOut Subsystem - ROS*); pré processamento e transmissão da informação para o nível 1 do sistema de seleção de eventos; implementação de algoritmos de reconstrução de sinal; entre outras [5, 53, 48].

Como pode ser visto na Figura 3.12, a placa da *sROD* é composta por: módulos *QSFP*, do inglês *Quad Small Form-factor Pluggable*, responsáveis pela comunicação com a eletrônica de *Front-End*; dispositivos FPGA, um Xilinx Virtex 7 e um Kintex 7, responsáveis pelo pré processamento e envio da informação para o nível 1 do *Trigger*, pela compactação e descompactação das informações de energia e controle, armazenamento dos sinais em memórias, recebimento e decodificação das informações de controle, temporização e seleção, entre outras [56, 53, 48]. Além desses elementos, a placa possui ainda módulos para gerenciamento de energia, comunicação *Ethernet*, memórias DDR3, entre outros.

Essas alterações nos sistemas eletrônicos do *TileCal* afetará o funcionamento

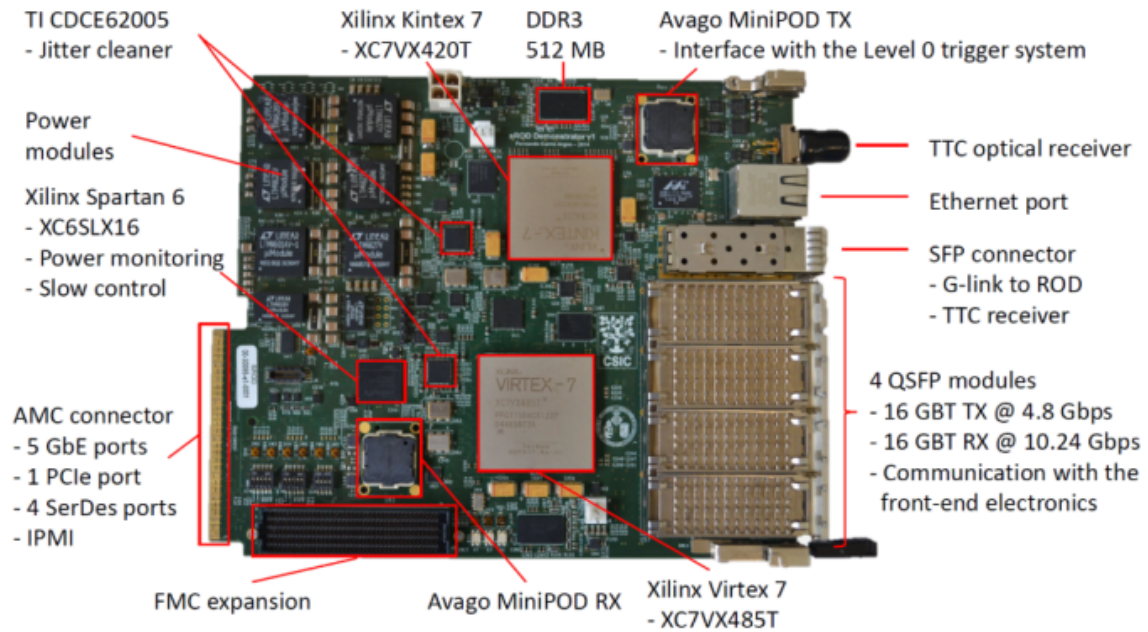


Figura 3.12: Protótipo *sROD* para Atualização do Sistema Eletrônico [57]

do primeiro nível do sistema de seleção de eventos. Na arquitetura anterior, esse nível recebia os sinais provenientes da placa analógica somadora. Com a retirada da mesma, deverão ser processados todos os eventos gerados pelas colisões. Dessa forma, a *sROD* será responsável por estimar a energia da célula de cada *bunch crossing* a uma taxa de 40 MHz.

Nesse cenário, este trabalho concentra-se na implementação de métodos de estimação de energia por *bunch crossing* no primeiro nível de *Trigger* em FPGA. É proposta, assim, uma solução que atende as especificações do projeto *sROD* desenvolvido pelos pesquisadores da colaboração ATLAS.

3.4 Técnica Atual de Estimação de Energia

Na arquitetura atual do primeiro nível do sistema de *Trigger*, conforme mencionado no tópico 3.2, os sinais analógicos são somados, gerando, assim, sinais maiores do que um único *bunch crossing* [10]. Com isso, torna-se necessário identificar o pulso referente a cada *bunch crossing*, processo denominado de Identificação do *Bunch Crossing* (BCID, do inglês *Bunch Crossing IDentification*) [9]. Os ruídos inerentes da eletrônica do calorímetro afetam diretamente a performance desse procedimento, pois, para realizar essa tarefa no primeiro nível de seleção utiliza-se dois algoritmos:

um Filtro Casado, responsável pela atenuação do ruído, e um circuito detector de picos.

3.4.1 O Filtro Casado

Para encontrar a solução ótima de um algoritmo e contornar o problema do ruído na eletrônica do *TileCal*, foi considerada a teoria de um sistema de comunicação digital, onde o transmissor é representado pelo componente *ADC* e o receptor, pelo sistema de processador do *BCID* [9]. Nesse sentido, as informações transmitidas podem ser consideradas como um vetor de amostras digitalizadas por *bunch crossing*, representado na Figura 3.13 como p_n , e os sinais recebidos pelo *BCID* são amostras afetadas pelo ruído, apresentadas na Figura 3.13 como o sinal x . O *BCID* deve ser capaz de rejeitar o ruído e gerar saídas considerando apenas os sinais de energia. Para isso, na arquitetura atual é utilizada a técnica do Filtro Casado para estimar o valor de energia verdadeiro.

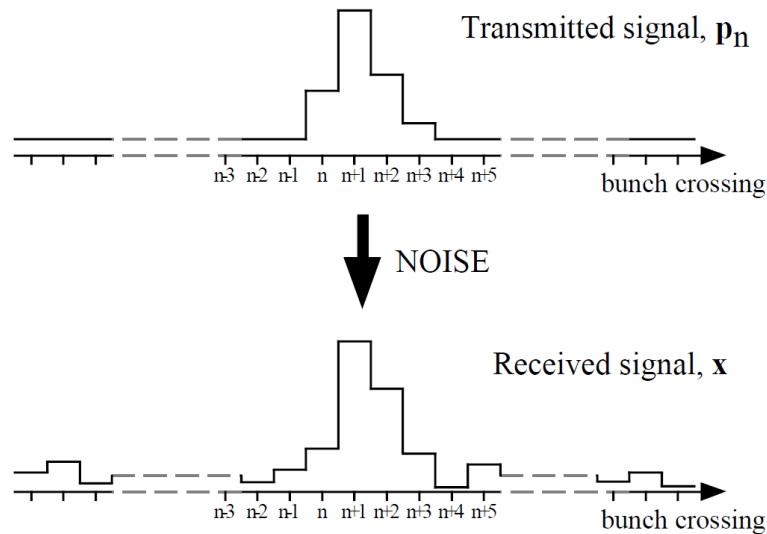


Figura 3.13: Representação gráfica do sinal transmitido e recebido pela atual arquitetura do *BCID* [9].

Nesse sistema, os sinais determinísticos são inseridos em ruído branco gaussiano e os coeficientes desse filtro são proporcionais a altura do pulso de cada amostra [10]. Para alcançar os requisitos do sistema de *BCID*, esses coeficientes são configurados para maximizar a relação sinal/ruído, sendo indicado para cenários de detecção de pulso que tenha sua forma conhecida [9]. O projeto desse filtro baseia-se no teste

de hipóteses [58], as quais, considerando uma sequência de N amostras de sinais digitalizados, são definidas como:

1. H_0 : Somente ruído $w[n]$ foi transmitido;
2. H_1 : Sinal de energia $s[n]$ mais ruído $w[n]$ foi transmitido;

Na Figura 3.14 observa-se a representação gráfica do problema da teoria da decisão baseada em duas hipóteses. Nessa ilustração, o elemento *source* gera saída que pode ser uma dentre as duas hipóteses. O elemento seguinte, *probabilistic transition mechanism*, é um dispositivo que conhece qual hipótese é verdadeira, sendo que, baseado nesse conhecimento, gera um ponto no *observation space* de acordo com a lei da probabilidade, a qual está relacionada com a densidade de probabilidade das hipóteses [58]. O *decision rule* refere-se à regra de decisão que é atribuída a cada ponto das hipóteses.

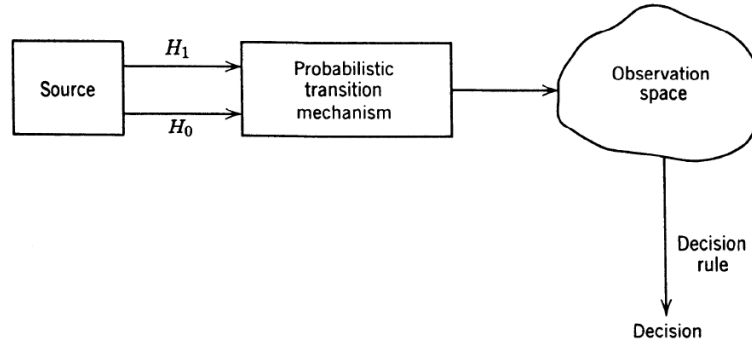


Figura 3.14: Diagrama da representação do problema da teoria da decisão [58].

Considerando que n representa o índice das amostras na sequência, as hipóteses listadas anteriormente para o cenário do *TileCal* podem ser representadas da seguinte forma:

1. H_0 : $r[n] = w[n]$ $n = 1, 2, 3, \dots, N$;
2. H_1 : $r[n] = s[n] + w[n]$ $n = 1, 2, 3, \dots, N$;

Utilizando as equações de densidade de probabilidade para H_0 e H_1 , obtém-se a razão que maximiza a eficiência de detecção, a qual é dada pela razão de semelhança da Equação 3.1 [12].

$$L(r) = \frac{f_{R|H_1}(r|H_1)}{f_{R|H_0}(r|H_0)} \quad (3.1)$$

A partir da Equação 3.1, considerando que as densidades de probabilidade do ruído são gaussianas, caracterizando-se como ruído branco (possuindo média zero [12, 9]) e o sinal $s[n]$ sendo determinístico, obtém-se a Equação 3.2 [12].

$$\sum_{i=1}^N r_i s_i \underset{H_0}{\overset{H_1}{\gtrless}} \gamma' \quad (3.2)$$

Desse modo, analisando a Equação 3.2 observa-se que a decisão entre as hipóteses é obtida através do somatório de produtos entre o sinal recebido $r[n]$ e o sinal determinístico $s[n]$, caracterizando-se, portanto, como um Filtro de resposta ao impulso finita [9, 10, 12].

O desempenho com a utilização dessa técnica de filtragem no nível 1 do *Trigger* do *TileCal* pode ser observado em [10, 59].

3.4.2 Detector de Picos

Os sinais de energia digitalizados, após passarem pelo Filtro Casado, são analisados por um algoritmo de detecção de picos, responsável por extrair a informação de onde ocorreu o pico de energia, ou seja, a deposição da energia no *bunch crossing*.

Nesse sistema, os sinais resultantes do processo de filtragem são armazenados em três registradores, representados na Figura 3.15 por f_1 , f_2 e f_3 , sendo estes os três mais recentes *bunch crossing*. Através desses valores, o algoritmo detector de pico faz a comparação desses sinais para encontrar um valor máximo utilizando a lógica $f_1 < f_2 > f_3$ [9, 10]. O resultado que satisfaz essa condição é comparado a um *threshold* e submetido a uma subtração de um valor de *offset* [9]. Na Figura 3.16 pode-se observar uma representação gráfica do funcionamento desse algoritmo, à esquerda o sinal antes da aplicação do Filtro Casado e à direita apresenta-se a detecção do pico onde ocorreu a deposição de energia.

Na Figura 3.15 está ilustrado o diagrama de blocos da estrutura do processador do BCID. Através da mesma pode-se observar o caminho da informação, desde a saída dos dispositivos ADCs, passando pelo Filtro Casado e Detector de Picos, até chegar no sistema de seleção de eventos (*Trigger*).

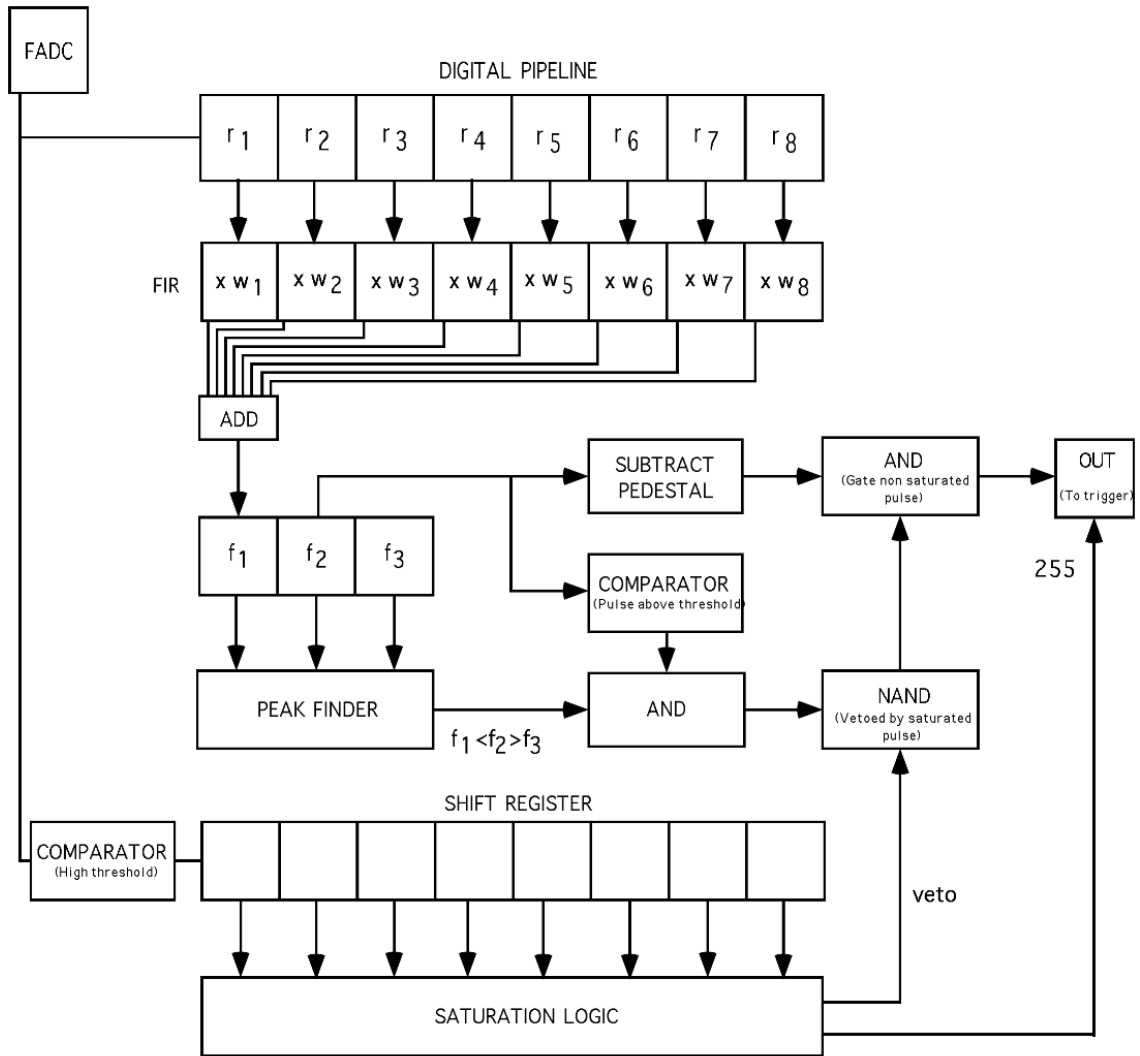


Figura 3.15: Diagrama de blocos do sistema de BCID [9].

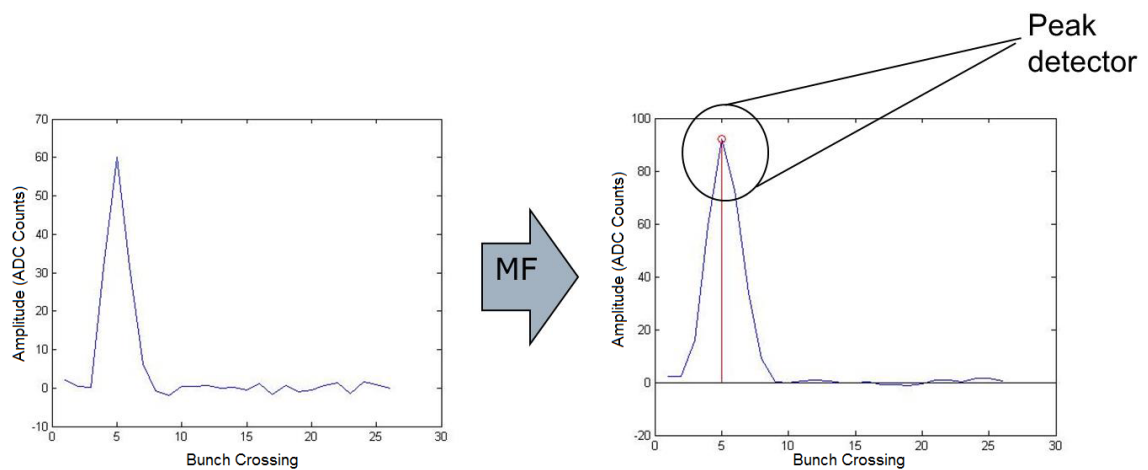


Figura 3.16: Representação gráfica da detecção da energia no primeiro nível do *Trigger* na arquitetura atual do *TileCal* [12].

3.5 Proposta de Nova Técnica de Estimação de Energia

Conforme abordado na seção 3.3, o Detector ATLAS está passando por fases de atualização com o objetivo de aumentar sua luminosidade, consequentemente, elevando a quantidade de colisões e informações geradas a cada evento. Esse cenário elevará a probabilidade de ocorrer múltiplas interações próton-próton em um mesmo *bunch crossing*, ou seja, energias de múltiplas colisões poderão ser depositadas em uma mesma célula em um intervalo inferior ao tempo de resposta dos sensores [6]. Esse fenômeno, conhecido como sobreposição, do inglês *pileup*, ocorre pois o tempo entre dois eventos subsequentes é inferior ao tempo de processamento da eletrônica das células.

O *pileup* é a sobreposição de informações provenientes de eventos subsequentes, provocando a deformação do pulso característico. Na Figura 3.17 pode-se observar um exemplo desse efeito, onde o sinal em vermelho se sobrepõe ao sinal em preto, resultando no sinal roxo deformado por *pileup* [6].

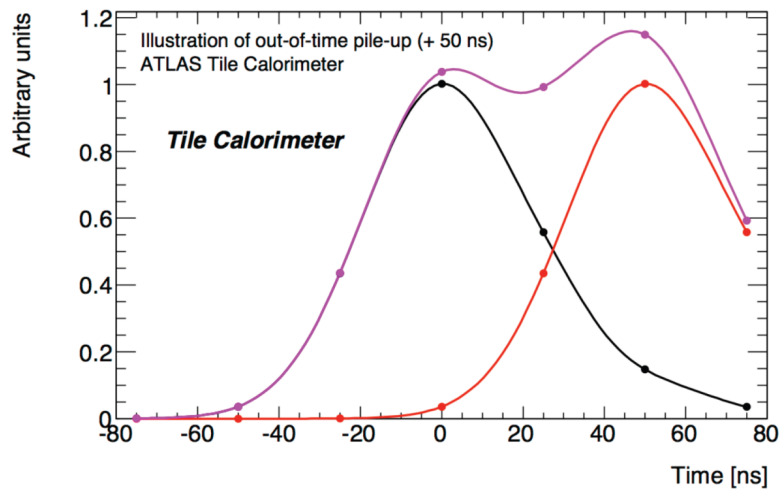


Figura 3.17: Representação gráfica do efeito de sobreposição entre sinais adjacentes [6].

Considerando o novo cenário de funcionamento da eletrônica do *TileCal*, torna-se necessário propor técnicas mais eficientes de estimação da energia no primeiro nível do *Trigger*, já que o Filtro Casado depende do conhecimento prévio da forma do pulso afetado pelo ruído. O desempenho desse método não é adequado para o

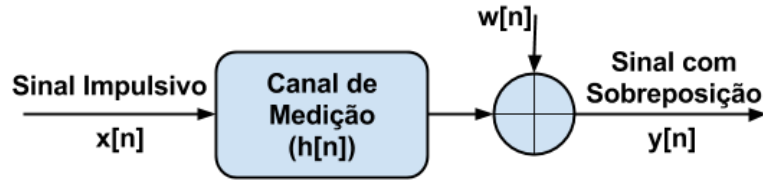


Figura 3.18: Diagrama do modelo da sobreposição de sinais no Calorímetro

Calorímetro, uma vez que o *pileup* altera a forma do pulso.

Na Figura 3.18 observa-se o diagrama que representa o modelo de distorção do sinal impulsivo no Calorímetro. $\mathbf{x}[n]$ representa o sinal de energia referente a uma colisão, $\mathbf{h}[n]$ a resposta ao impulso do canal de medição do Calorímetro modelado, $\mathbf{w}[n]$ representa o ruído de medição e, por fim, $\mathbf{y}[n]$ o sinal resultante do processo de medição. Nessa abordagem, o problema consiste na recuperação de $\mathbf{x}[n]$ a partir do conhecimento de $\mathbf{h}[n]$ e $\mathbf{y}[n]$.

Em seu trabalho, Duarte [12] analisou o desempenho de diferentes técnicas de detecção e estimação de energia, por *bunch crossing*, com a finalidade de reconstruir o sinal verdadeiro de energia (ou ao menos obter uma boa aproximação do sinal original) através da utilização de filtros digitais. Os métodos propostos em [12] baseiam-se no princípio da convolução, dentre os quais, como escopo de implementação em *hardware* nesse trabalho, foram escolhidos duas abordagens: uma determinística e outra iterativa [12].

3.5.1 Filtros Digitais e a Teoria da Convolução

Filtros digitais são estruturas utilizadas tipicamente para alterar atributos de um determinado sinal no domínio do tempo ou da frequência [60]. Essas estruturas possuem diferentes funções, como por exemplo a separação e a reconstrução de sinais. Em outras palavras, os filtros permitem a passagem de determinadas frequências enquanto que outras são atenuadas [13].

A filtragem de sinais é uma das aplicações mais comuns do Processamento Digital de Sinais, sendo utilizada em diversas áreas, dentre elas: robótica, comunicação de informações, biomédica, processamento de áudio, entre outras [60]. O processamento de sinais de forma digital vem sendo cada vez mais utilizado em detrimento da forma analógica. Este fato está relacionado aos benefícios da utilização de estruturas

digitais, como: maior robustez às mudanças de temperatura e menor custo [60].

Essas estruturas são caracterizadas, basicamente, como algoritmos implementados em hardware ou em software. A depender dos requisitos, os filtros podem ser executados por um Processador Digital de Sinais (DSP, do inglês *Digital Signal Processor*), em *hardware* dedicado e também em *software*. Na Figura 3.19 é ilustrado um diagrama básico da filtragem de um sinal digital $x[n]$, sendo a informação desejada representada por $y[n]$.



Figura 3.19: Diagrama de blocos da filtragem de um sinal digital.

No âmbito desse trabalho, a eletrônica do calorímetro é considerada como um sistema linear e invariante no tempo (LTI). Os sinais de entrada ($x[n]$) interagem com a sequência de resposta ao impulso do filtro ($f[n]$) através do processo de convolução linear [60]. A convolução é representada por: $y = f * x$, onde f é a resposta ao impulso do filtro, x é o sinal de entrada e y é a saída do sistema. A operação de convolução é definida pela Equação 3.3 [60].

$$y[n] = x[n] * f[n] = \sum_K f[k]x[n - k] \quad (3.3)$$

Os filtros digitais lineares e invariantes no tempo são classificados em dois tipos: resposta ao impulso finita e resposta ao impulso infinita (IIR, do inglês *Infinite Impulse Response*). O primeiro tipo possui quantidade finita de amostras de entrada, tendo assim, um número finito de somas em seu somatório. Em contrapartida, o segundo tipo requer uma quantidade infinita de somas [60].

Filtros FIR possuem duração finita de valores de saída e, portanto, sua resposta ao impulso torna-se nula após um determinado tempo. Esses filtros tem resposta de fase linear, ou seja, não são produzidas distorções de fase no sinal filtrado. Além disso, não possuem realimentação e são sempre estáveis.

A função que define o filtro FIR, apresentada na Equação 3.4, é uma versão finita da convolução da Equação 3.3. Os coeficientes desse filtro são constantes, caracterizando-se como um filtro digital linear e invariante no tempo, não recursivo,

sendo dependente apenas dos sinais atrasados de $x[n]$ [60]. Na Equação 3.4, L representa a ordem do filtro, e $f[n]$ as respostas ao impulso do FIR.

$$y[n] = x[n] * f[n] = \sum_{K=0}^{L-1} f[k]x[n - k] \quad (3.4)$$

Existem diferentes estruturas para os filtros FIR, portanto, diferentes formas de implementação do mesmo. São exemplos: forma direta, estrutura transposta, cascata, entre outras. Na Figura 3.20 apresenta-se graficamente a forma direta, enquanto que na Figura 3.21 pode-se observar sua forma transposta e, por fim, na Figura 3.22 a forma em cascata.

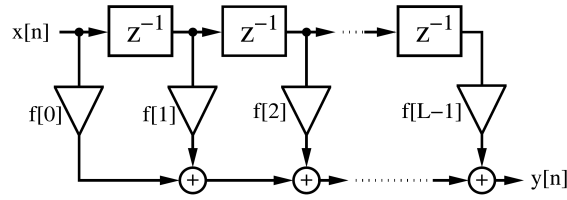


Figura 3.20: Representação da Forma Direta do Filtro FIR [60].

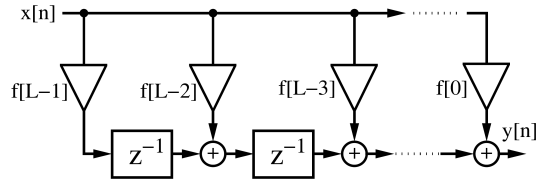


Figura 3.21: Representação da Forma Transposta do Filtro FIR [60].

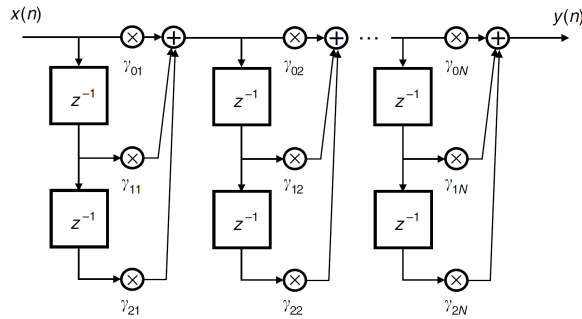


Figura 3.22: Representação da Forma em Cascata do Filtro FIR [61].

Como podem ser observadas nas Figuras 3.20, 3.21 e 3.22, as diferentes estruturas dos filtros são compostas, basicamente, por: somadores, multiplicadores e atrasos

unitários. A forma direta, em relação à implementação, é a arquitetura mais simples. Seus coeficientes são obtidos diretamente a partir da função de transferência [61]. Essa forma segue, diretamente, as operações apresentadas pela Equação 3.4.

A forma transposta foi definida baseando-se na forma direta, ou seja, apresenta-se como uma alternativa a essa estrutura. Em relação aos benefícios, afirma-se que não é necessário registrador de deslocamento extra para o $x[n]$ e também não é necessário estágio extra de pipeline para os somadores do produto para que se obtenha alta taxa de saída (ou *throughput*) [60]. Por fim, a forma em cascata é obtida através da conexão, em cascata, de filtros FIR de segunda ordem. Os coeficientes, representados por γ_{ik} , são determinados a partir dos $f(k)$ da função de resposta ao impulso do filtro [61].

Existem alternativas de estruturas FIR mais eficientes do que as já mencionadas. Dentre elas pode-se destacar a forma treliça e a estrutura polifásica. Através da Figura 3.23 pode-se observar o diagrama de blocos da forma treliça não-recursiva e de ordem M, sendo este formado pela concatenação da estrutura apresentada na Figura 3.24. Destaca-se a aplicação desse modelo em bancos de filtros biortogonais com fase linear e reconstrução perfeita [61].

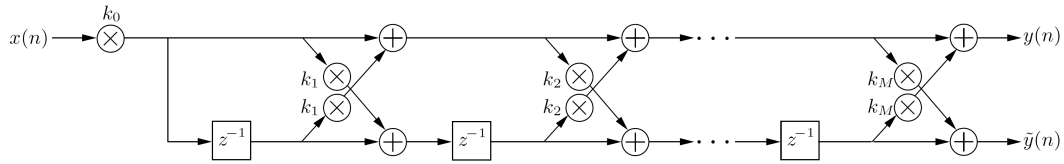


Figura 3.23: Representação da Forma Treliça do Filtro FIR [61].

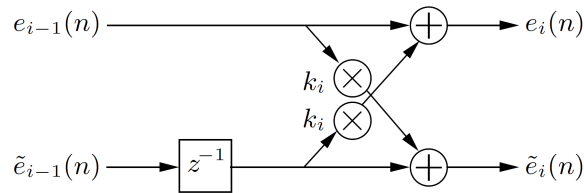


Figura 3.24: Representação do bloco básico utilizado na Forma Treliça [61].

Para o entendimento da forma treliça não-recursiva, é necessário analisar a recursividade presente no bloco básico representado na Figura 3.24. Essas relações são apresentadas nas Equações 3.5 e 3.6, sendo que: $i = 1, 2, \dots, M$; $e_0(n) = \tilde{e}_0(n) =$

$k_0x(n)$; e $e_M(n) = y(n)$ [61].

$$e_i(n) = e_{i-1}(n) + k_i \tilde{e}_{i-1}(n-1) \quad (3.5)$$

$$\tilde{e}_i(n) = \tilde{e}_{i-1}(n-1) + k_i e_{i-1}(n) \quad (3.6)$$

Outra estrutura de FIR eficiente é a forma Polifásica, a qual destaca-se pela sua aplicação em processamento paralelo. Essa forma de implementação caracteriza-se por sua decomposição polifásica, podendo o filtro ser dividido em diferentes M filtros. Além disso, apresenta-se também como uma solução eficiente para implementações de banco de filtros que operam com diferentes faixas [61]. Através da Figura 3.25 pode-se observar a estrutura dessa forma Polifásica, na qual cada $H_i(z)$ é encontrada pela Equação 3.7, na qual $i = 0, 1, \dots, K-1$ [61].

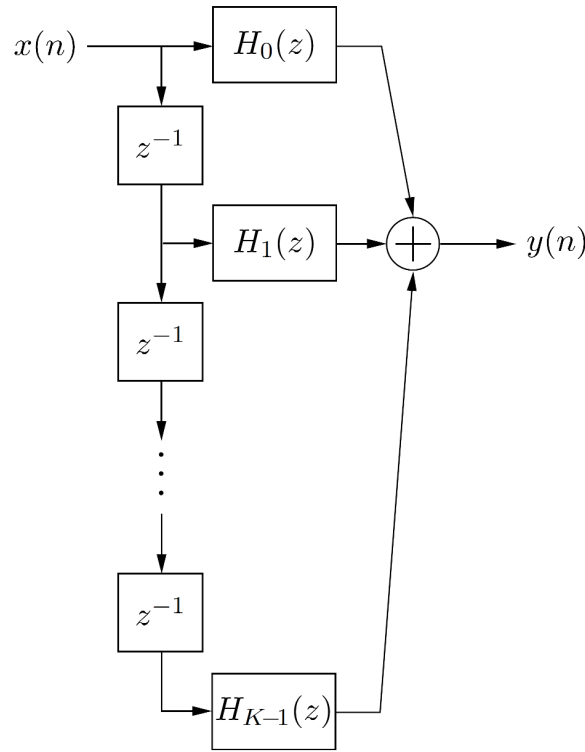


Figura 3.25: Representação da Forma Polifásica [61].

$$\mathbf{H}_i(\mathbf{z}^{\mathbf{K}}) = \sum_{\mathbf{n}=0}^{\bar{\mathbf{N}}-1} \mathbf{h}(\mathbf{K}\mathbf{n} + \mathbf{i}) \mathbf{z}^{-\mathbf{K}\mathbf{n}} \quad (3.7)$$

Além dos Filtros FIR, temos também os de resposta ao impulso infinita. As estruturas IIR são recursivas, sendo, no geral, uma solução mais adequada quando

o projeto possui limitação em relação a ordem. Nesses casos, são necessários menos coeficientes do que os filtros FIR para atender a um determinado requisito, tendo, assim, um menor tempo de execução. Em contrapartida, o IIR apresenta também algumas desvantagens em relação ao FIR. São exemplos: a realimentação pode introduzir instabilidade; mais difícil de implementar com pipeline, entre outras [60]. Filtros IIR não serão detalhados nesse trabalho, pois, diante das características do *TileCal* e da especificação de funcionamento do filtro, estudos anteriores apontaram que a estrutura FIR é mais adequada para o problema [12].

3.5.2 Método Determinístico

A abordagem determinística consiste na aplicação da técnica de Filtro FIR para realizar a deconvolução, ou seja, recuperar a informação de cada *bunch crossing* afetado pelo efeito de sobreposição. Através da modelagem da eletrônica do calorímetro como um sistema linear, foi possível obter o filtro inverso com o auxílio da Transformada Z .

3.5.2.1 Transformada Z

A Transformada Z é uma importante ferramenta para a análise e projeto de filtros digitais, pois esta técnica apresenta-se como uma alternativa à Transformada de Fourier no tempo discreto, que em muitos casos não atende às condições de convergência [62]. Além disso, permite a avaliação da estabilidade e causalidade dos sistemas [12]. A Transformada Z no tempo discreto, considerando um sistema LTI com sequência $q[n]$, pode ter sua resposta ao impulso definida pela Equação 3.8, na qual z é uma variável complexa [61].

$$Q(z) = \sum_{n=-\infty}^{\infty} q[n]z^{-n} \quad (3.8)$$

Devido a existência de uma variável complexa, essa transformada pode ser interpretada utilizando o plano complexo, z , ilustrado graficamente na Figura 3.26. A circunferência unitária é a região onde $|z| = 1$, sendo assim caracterizada por ter raio unitário e por representar o conjunto de pontos onde $z = e^{jw}$ para $0 \leq w \leq 2\pi$. Nesse ponto, a Transformada Z é reduzida a Transformada de Fourier, pois a magnitude da variável z é a *unidade* [63].

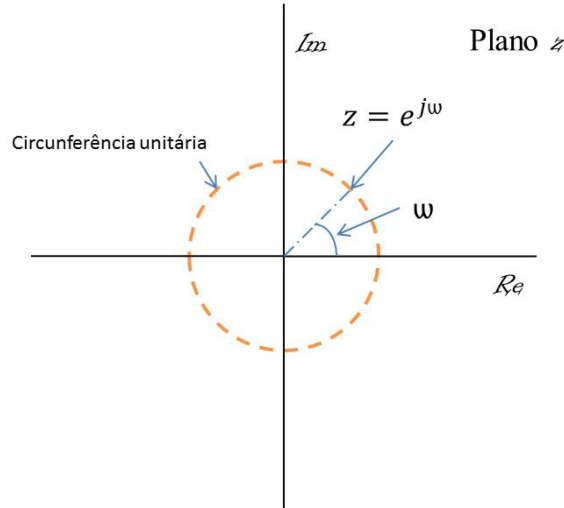


Figura 3.26: Representação Gráfica do Plano Z [12].

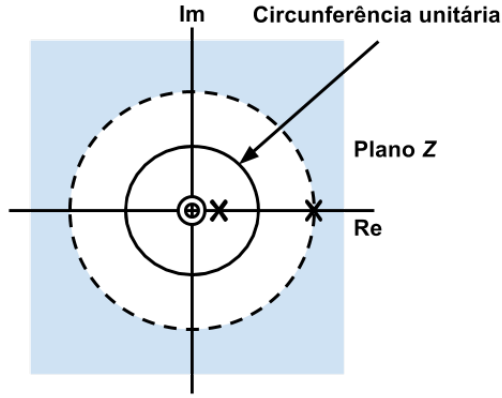
Nessa transformada, existem condições para a convergência do somatório infinito apresentado na Equação 3.8, ou seja, valores para os quais o somatório converge para um valor finito. O conjunto de valores de z que proporcionam a convergência de $Q(z)$ chama-se Região de Convergência (RDC) [62, 63]. A RDC pode ser expressa de acordo com a Equação 3.9 [64].

$$\sum_{n=-\infty}^{\infty} |q[n]| |z|^{-n} < \infty \quad (3.9)$$

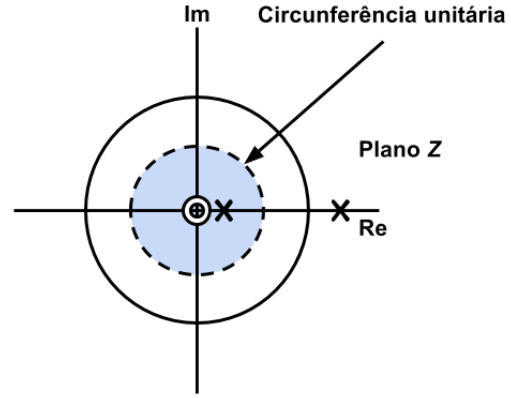
A Transformada Z pode ser utilizada, inclusive, para análise de sistemas cujo seu somatório infinito seja definido como uma função racional formada por polinômios em z , como exemplificado na Equação 3.10. Nessa Equação, as raízes de $O(z)$ representam os zeros de $Q(z)$ enquanto que as raízes de $F(z)$ os pólos de $Q(z)$ [65]. Nesses sistemas, a região de convergência é definida em relação a localização dos seus pólos [62]. Dentre outras propriedades desses sistemas racionais, destaca-se que: a RDC não pode conter pólos; se a sequência for um sinal lateral direito, a RDC corresponde à direita do pólo mais afastado da origem; se sequência for um sinal lateral esquerdo, a RDC corresponde à área esquerda ao pólo mais próximo da origem [62, 63].

$$Q(z) = \frac{O(z)}{F(z)} \quad (3.10)$$

Na Figura 3.27 pode-se observar duas possíveis regiões de convergências com seus



(a) Representação gráfica da RDC e dos pólos e zeros com sequência lateral direita.



(b) Representação gráfica da RDC e dos pólos e zeros com sequência lateral esquerda.

Figura 3.27: Representação de duas possíveis RDCs para sistemas com função racional.

zeros e pólos de um sistema formado por função racional. As RDCs são indicadas pela região em destaque. Os pólos são indicados pelo X e, em todos os casos, o zero encontra-se na origem. Na Figura 3.27a a região de convergência está associada a uma sequência lateral direita enquanto que na Figura 3.27b a uma sequência lateral esquerda.

Além das características apresentadas, a Transformada Z possui outras propriedades, destacando-se: Linearidade; Inversão de Tempo; Convolução; dentre outras. Maiores detalhes e especificações dessas propriedades pode ser encontrado em [64].

3.5.2.2 Projeto do Filtro Inverso Realizável

Para a implementação do método determinístico foi necessário projetar um filtro inverso que fosse realizável, viabilizando a estimação da energia do TileCal através da técnica de deconvolução.

A Eq 3.11, na qual $h[n]$ e $g[n]$ são as respostas ao impulso e $\delta[n]$ representa um impulso unitário, apresenta uma relação em que dois sistemas lineares, invariantes no tempo e causais no tempo discreto são inversos.

$$h[n] * g[n] = \delta[n] \quad (3.11)$$

Em um sistema definido pela relação $y[n] = x[n] * h[n]$, temos que $y[n]$ representará o sinal $x[n]$ distorcido pelo canal $h[n]$. Relacionando esse sistema com o Calorímetro apresentado na Figura 3.18, para efetuar a recuperação de $x[n]$, deve-se convoluir $y[n]$ com $g[n]$, que representa a resposta ao impulso do canal inverso de $h[n]$. Sendo assim, o resultado desse processamento ($v[n]$) será uma aproximação da entrada $x[n]$. A Eq. 3.12 ilustra essas relações.

$$y[n] = x[n] * h[n] \rightarrow v[n] = y[n] * g[n] \simeq x[n] \quad (3.12)$$

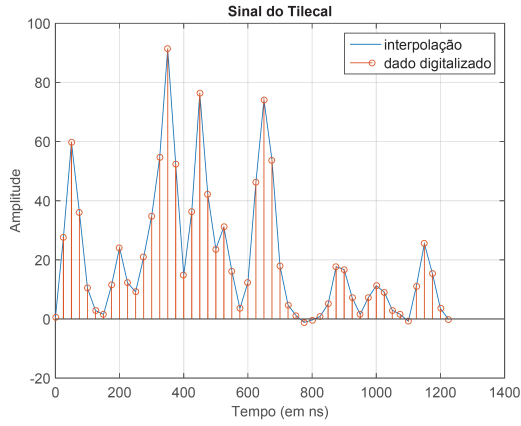
Este sistema inverso pode ser representado no domínio Z . Para isso, aplica-se a Transformada Z em ambos os lados da Eq. 3.11, obtendo-se o sistema inverso de $\mathbf{H}(z)$ (Eq. 3.13).

$$G(z) = \frac{1}{H(z)} \quad (3.13)$$

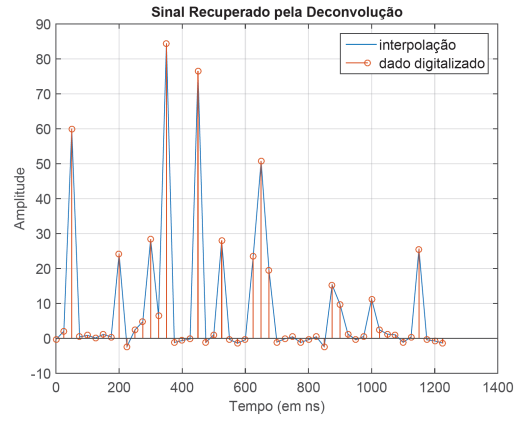
Na abordagem determinística, encontram-se os coeficientes do filtro inverso a partir do pulso característico $h[n]$ do detector. No domínio z , obtém-se o filtro inverso $G(z)$ e, conseqüentemente, os coeficientes utilizados para deconvoluir os sinais com sobreposição [12].

Ao se analisar o espectro do pulso característico do *TileCal* no domínio z ($H(z)$), considerando um sistema causal, o filtro inverso ($G(z)$) não é realizável. Para se obter uma aproximação de um filtro realizável, adotou-se uma estratégia a partir da separação de $G(z)$ em dois polinômios $G_1(z)$ e $G_2(z)$, sendo o primeiro composto pelos pólos internos ao círculo unitário, caracterizando-se como estável, enquanto que o segundo possui pólos externos ao círculo unitário, caracterizando-se como instável. Sendo assim, trata-se $G_2(z)$ para obter uma aproximação estável e causal do mesmo. A partir da resposta ao impulso obtida de $G_2(z)$, realiza-se a convolução com a resposta ao impulso do sistema $G_1(z)$, obtendo, assim, a resposta ao impulso de uma aproximação da função inversa $G_A(z)$ realizável [12].

Na Figura 3.28 pode-se observar o resultado alcançado por [12] na reconstrução de um sinal utilizando o projeto desse filtro inverso. Na Figura 3.28a apresenta-se o gráfico contendo o sinal do detector afetado pelo efeito da sobreposição e na Figura 3.28b apresenta-se o sinal recuperado usando a abordagem determinística.



(a) Representação gráfica do sinal do *TileCal* com efeito de sobreposição [12].



(b) Representação gráfica do sinal recuperado [12].

Figura 3.28: Exemplo de recuperação do sinal do *TileCal* usando o método determinístico.

3.5.3 Método Iterativo

Na busca por uma técnica eficiente na reconstrução dos sinais de energia do *TileCal* além de métodos diretos, Duarte [12] também avaliou o desempenho de métodos iterativos. Os métodos diretos, como o determinístico, baseiam-se em Filtros FIR, realizando o processamento de cada amostra diretamente. Em contrapartida, os métodos iterativos realizam o processamento a partir de uma janela de amostras. Na abordagem iterativa, o conjunto de sinais é processado iterativamente, formando sua saída a partir da solução de um sistema de equações lineares, a qual, a cada iteração, converge para a estimativa desejada.

Os métodos analisados foram o algoritmo *Gold* e duas variações do Gradiente Descendente [66]. Para o entendimento dessa abordagem, torna-se necessário o conhecimento do modelo de deconvolução de Van Cittert. Nos tópicos a seguir, além do modelo de Van Cittert, serão abordados o Gradiente Descendente e suas variações. O algoritmo *Gold* não será abordado pois não faz parte do escopo desse trabalho. Maiores informações sobre o mesmo podem ser encontradas em [12, 14, 67].

3.5.3.1 O Modelo de Van Cittert

Baseada na teoria da convolução e na implementação de filtros digitais abordada na seção 3.5.1, temos que a equação da convolução entre $x[n]$ e $h[n]$ pode ser definida por $y[n] = x[n] * h[n]$, sendo sua implementação na estrutura de um FIR definida pela Equação 3.14. Contudo, uma convolução discreta e linear pode ser descrita de forma matricial. Dessa forma, a Equação 3.14 pode ser definida de acordo com a Equação 3.15 [14].

$$y[n] = \sum_{k=0}^{L-1} x[k]h[n-k] \quad (3.14)$$

$$\begin{bmatrix} y0 \\ y1 \\ y2 \\ y3 \\ y4 \end{bmatrix} = \begin{bmatrix} h0 & 0 & 0 \\ h1 & h0 & 0 \\ h2 & h1 & h0 \\ 0 & h2 & h1 \\ 0 & 0 & h2 \end{bmatrix} \begin{bmatrix} x0 \\ x1 \\ x2 \end{bmatrix} \quad (3.15)$$

Na Equação 3.15, $[h0 \ h1 \ h2]$ representa a resposta ao impulso do filtro, enquanto que $[x0 \ x1 \ x2]$ os sinais a serem convoluídos. O conjunto $[y0 \ y1 \ y2 \ y3 \ y4]$ representa o resultado dessa operação. Logo, a convolução linear em sua forma matricial é definida pela Equação 3.16.

$$\mathbf{y} = \mathbf{H}\mathbf{x} \quad (3.16)$$

A partir da Equação 3.16, com a multiplicação de ambos os lados por $\mathbf{H}^T$, obtém-se uma matriz *Toeplitz* ($\mathbf{H}^T\mathbf{H}$) [14], sendo esta uma matriz positiva, com seus autovalores reais [12]. Logo, têm-se a Equação 3.17 que modela o processo de reconstrução de um sinal através da deconvolução [14].

$$\mathbf{H}^T\mathbf{y} = \mathbf{H}^T\mathbf{H}\mathbf{x} \quad (3.17)$$

A partir da Equação 3.17 propõe-se, então, o método de deconvolução iterativa de *Van Cittert*, que apresenta-se como uma solução menos sensível ao erro [14]. O equacionamento desse modelo está representado em 3.18.

$$\hat{\mathbf{x}}^{\text{iter}+1} = \hat{\mathbf{x}}^{\text{iter}} + \mu\mathbf{H}^T(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^{\text{iter}}) \quad (3.18)$$

Na Equação 3.18, *iter* representa o índice da iteração atual e μ um fator de relaxamento real, sendo este necessário para garantir a convergência do sistema [12]. A partir das análises do critério de convergência, obtém-se que os valores assumidos por μ devem respeitar a relação apresentada em 3.19, sendo λ_{max} o maior autovalor de $\mathbf{H}^T\mathbf{H}$ [12].

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (3.19)$$

3.5.3.2 Gradiente Descendente e suas Variações

O algoritmo Gradiente Descendente [66] apresenta-se como uma técnica de deconvolução que parte da minimização do erro médio quadrático. Esse método é modelado de acordo com a Equação 3.20, onde $\mathbf{r}$ é um vetor de amostras recebido pelo sistema, $\mathbf{H}\hat{\mathbf{x}}$ é o modelo linear para convolução, conforme apresentado em 3.16, e $\mathbf{x}$ o vetor resultante do processo de estimação [12].

$$\mathbf{J}(\hat{\mathbf{x}}) = (\mathbf{r} - \mathbf{H}\hat{\mathbf{x}})^T(\mathbf{r} - \mathbf{H}\hat{\mathbf{x}}) \quad (3.20)$$

De forma análoga ao modelo de Van Cittert, apresentado em 3.18, a Equação 3.21 apresenta uma abordagem iterativa que minimiza a Equação 3.20 [12]. Nesse método, o μ é definido com o intuito de otimizar a taxa de convergência, ou seja, obter o menor valor da função custo $\mathbf{J}$ na direção contrária ao gradiente da função $\mathbf{J}$ [12].

$$\hat{\mathbf{x}}^{iter+1} = \hat{\mathbf{x}}^{iter} + \mu\mathbf{H}^T(\mathbf{r} - \mathbf{H}\hat{\mathbf{x}}^{iter}) \quad (3.21)$$

Uma variação do método do Gradiente Descendente, que considera em seu processamento apenas amostras positivas do sinal a ser recuperado, é chamado de Gradiente Descendente Positivo. Esse algoritmo é executado da mesma forma que o Gradiente Descendente, através do equacionamento matemático apresentado nessa seção (Equações 3.20 e 3.21). Sua particularidade consiste na anulação (atribui-se valor 0) de amostras com valores abaixo de determinado limiar antes de ser iniciada uma próxima iteração [12]. Na Figura 3.29 é ilustrado o fluxograma desse algoritmo.

O Gradiente Descendente Positivo é uma técnica que tende para uma solução esparsa positiva. Considerando o ambiente do calorímetro, esse método concentra a

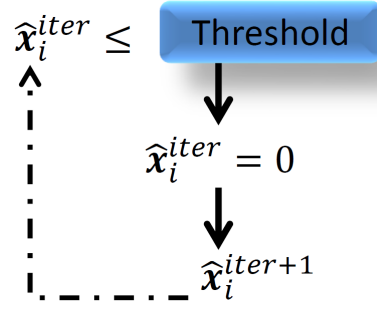


Figura 3.29: Representação da lógica do Gradiente Descendente Positivo [12].

energia em componentes positivas, ou seja, não estima sinais positivos e negativos de eventos adjacentes do *TileCal*.

Em uma outra variação do Gradiente Descendente, ao invés da anulação das componentes abaixo de um limiar, estas são excluídas da próxima iteração. Esse método é conhecido como Gradiente Descendente que Exclui Zeros [12]. O benefício desse método em relação ao Positivo, consiste na necessidade de menos recursos computacionais durante o seu processamento, pois, com a exclusão de componentes, o tamanho das matrizes poderão reduzir de uma iteração para outra.

Para este trabalho, o método iterativo adotado para ser desenvolvido em *hardware* foi o Gradiente Descendente Positivo. O benefício para a utilização do método que exclui zeros não justificaria sua escolha, pois, em FPGA não é possível excluir componentes lógicos em tempo de execução, ou seja, não representaria ganho em relação a redução dos recursos utilizados.

Capítulo 4

Arquitetura de Hardware Desenvolvida

4.1 Introdução

Este trabalho tem como objetivo o desenvolvimento de uma nova solução em hardware para a reconstrução dos sinais de energia do calorímetro Hadrônico do ATLAS, uma vez que a eletrônica atual terá seu desempenho comprometido com o problema da sobreposição de sinais após a atualização do LHC. Segundo as especificações de funcionamento do detector, o principal requisito desta nova solução é a adequação à nova frequência de ocorrência das colisões.

Conforme abordado no Capítulo 3, o LHC está passando por um processo de atualização com o objetivo de aumentar a luminosidade dos seus experimentos. A frequência de ocorrência das colisões nesse acelerador será de 40 MHz, ou seja, um novo evento será gerado a cada 25 nanosegundos e todos os sinais de energia gerados com essa frequência serão enviados para o nível 1 do sistema de *trigger* [5]. Com a arquitetura atual do sistema eletrônico do calorímetro hadrônico, essa atualização provocará o efeito de sobreposição ou empilhamento dos sinais de energia, afetando diretamente o funcionamento do sistema de seleção de eventos (*trigger*) [6]. Dessa forma, torna-se necessária a utilização de um sistema digital capaz de realizar a reconstrução do sinal proveniente dos sensores considerando a taxa de 40 MHz.

Conforme abordado na seção 3.3, pode-se observar que a Colaboração ATLAS definiu a utilização dos dispositivos FPGAs Virtex-7 e Kintex-7, da Xilinx [57]. O

primeiro é responsável pela reconstrução dos sinais de energia. Com isso, a solução proposta neste trabalho considerou o ambiente de desenvolvimento da Xilinx, tendo como base o dispositivo XC7VX485T [68].

A definição da arquitetura do *hardware* levou em consideração os requisitos definidos pela colaboração ATLAS em relação à atualização do sistema eletrônico. Nesse novo cenário, todos os sinais de energia gerados à cada colisão devem ser processados. Sendo assim, a arquitetura proposta desse trabalho precisa lidar com a restrição de tempo de processamento de 25 nanosegundos (intervalo de tempo entre duas colisões). Além dessa característica, os sinais dos sensores digitalizados são enviados para o sistema de processamento de forma serial e os sinais recuperados precisam ser enviados para o nível 1 do *Trigger* no mesmo formato. Para garantir o funcionamento de acordo com as especificações, foi necessário implementar técnicas como processamento em paralelo e diferentes domínios de *clock*.

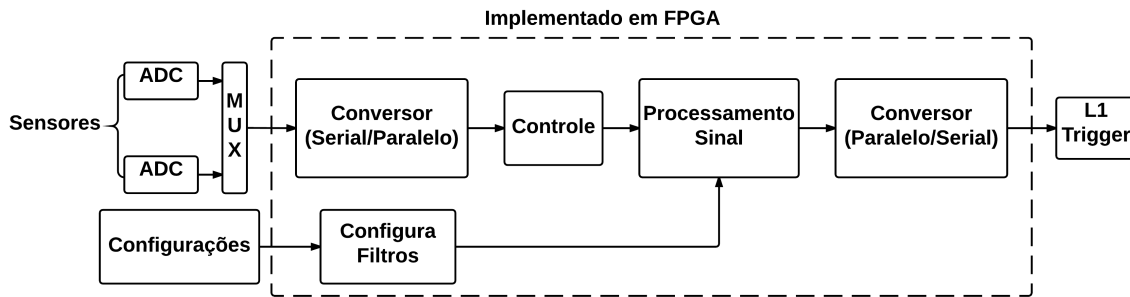


Figura 4.1: Arquitetura do Sistema Eletrônico Implementado em FPGA.

Na Figura 4.1 mostra-se a arquitetura de *hardware* proposta. Os elementos dentro da região tracejada pertencem ao escopo desse trabalho, os demais ilustram a interação externa com o hardware desenvolvido. Os conversores analógicos digitais realizam a digitalização dos valores analógicos de energia. O multiplexador (“MUX”) controla o envio desses sinais, bit a bit, para serem reconstruídos. “Configurações” representa o envio de palavras de configuração para o *hardware*.

Com o intuito de tornar o sistema reconfigurável, definiu-se utilizar registradores de configuração que podem ser alterados externamente pelo usuário do sistema. Com essa alternativa, os filtros podem ter seu funcionamento alterado externamente, sem a necessidade de reconfiguração do FPGA. Esses registradores estão representados pelo bloco “Configura Filtros”, ilustrado na Figura 4.2. Esse bloco possui interface com o topo do projeto, permitindo comunicação externa, e com o bloco

responsável pelo processamento do sinal, disponibilizando, diretamente, as palavras de configuração. As entradas desse módulo são: “clk”, que representa o sinal de *clock*; “rst_n”, que representa o sinal de *reset*; “addr”, que é o endereço do registrador de configuração onde a palavra será escrita; “Input data”, que representa a palavra de configuração; “Input valid”, responsável por informar que existe uma palavra de configuração válida disponível. As saídas “Config reg” representam as informações de configuração utilizadas para cada implementação de filtragem.



Figura 4.2: Diagrama do Bloco Configura Filtros.

Para o processamento eficiente dos sinais de energia, estes precisam estar disponíveis em forma paralela. Portanto, torna-se necessário fazer a conversão das entradas de serial para paralelo. Esse processo é representado no bloco “Conversor Serial/Paralelo”, na Figura 4.3. Como o nível 1 de *trigger* precisa receber os sinais de forma serial, foi necessário fazer a conversão dos sinais de energia reconstruídos em serial. Sendo este processo realizado no bloco “Conversor Paralelo/Serial”, representado na Figura 4.4. Ambos possuem interfaces com o topo do projeto, pois controlam os sinais que entram e que saem do FPGA, além dessa interface, o primeiro é conectado ao bloco de controle e o segundo ao bloco de filtragem. Nesses dois blocos, as entradas “clk” e “rst_n”, representam os sinais de *clock* e *reset*, respectivamente, enquanto que “Input valid” informam a existência de um sinal válido disponível. A entrada “Input data bit” representa o bit válido de um sinal de energia recebido de um sensor, enquanto que “Filtered data”, representa um sinal paralelo filtrado para ser convertido em serial. As saídas “Valid parallel data” e “Valid bit” informam a existência de sinal válido para o bloco subsequente, enquanto que “Parallel data” e “Data bit” referem-se às informações de energia em sua forma paralela e serial disponibilizadas por cada bloco de conversão.

Cada módulo desse sistema, ilustrado na Figura 4.1, é responsável por processar os sinais de até 12 sensores. A cada colisão, as fotomultiplicadoras são sensibilizadas



Figura 4.3: Diagrama do Bloco Conversor Serial/Paralelo.



Figura 4.4: Diagrama do Bloco Conversor Paralelo/Seria.

e, então, enviam os sinais de energia para o FPGA. Dessa forma, o *hardware* precisa identificar a origem de cada sinal, para que seja realizado o correto processamento do mesmo. Para garantir essa organização foi criado o bloco “Controle”, como mostrado na Figura 4.5, que está conectado ao bloco de conversão serial/paralelo e ao bloco de filtragem. Esse módulo possui como sinais de entrada: “clk”; “rst_n”; “valid_parallel_data”, corresponde ao sinal de controle que identifica uma amostra válida de sinal proveniente do conversor serial paralelo; e “parallel_data”, que representa o sinal de energia em sua forma paralela. As saídas, são: “data”, referente ao sinal de energia; e o conjunto de sinais “Valid Channel”, de 0 a 11, que informa ao bloco “Processamento Sinal” a qual sensor a amostra disponível pertence.

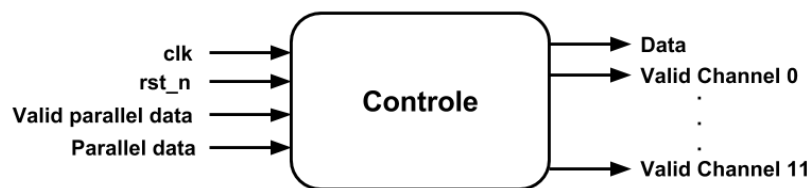


Figura 4.5: Diagrama do Bloco Controle.

O “Processamento Sinal”, ilustrado na Figura 4.6, representa os filtros digitais, responsáveis por efetuarem a reconstrução dos sinais de energia afetados pela sobreposição. Esse bloco é composto pelas instancias dos filtros que, nesse trabalho, considerou dois diferentes métodos, o Determinístico e o Iterativo. Suas implementações foram abordadas nas seções 3.5.2 e 3.5.3. Suas interfaces estão conectadas aos blo-

cos de controle e de conversão paralelo/serial. As entradas desse bloco são: “clk”; “rst_n”; as entradas “Valid Channel”, de 0 até 11, referem-se aos sinais do “Controle” informando a qual canal o sinal pertence; os “Config reg” representam as informações de configuração utilizadas em cada implementação de filtragem. Em relação às saídas, “Filtered data” representa o sinal de energia reconstruído e “Valid data” é o sinal de controle informando que existe uma saída válida.

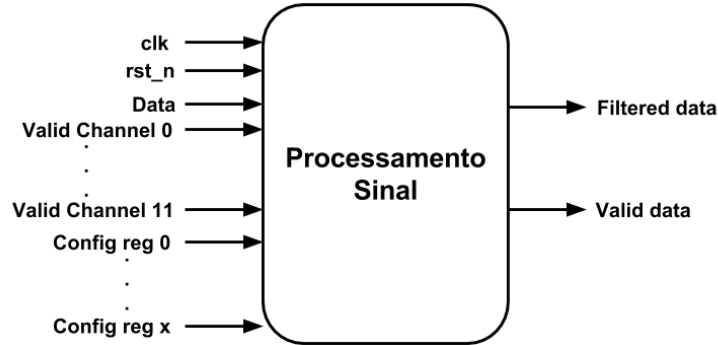


Figura 4.6: Diagrama do Bloco Processamento Sinal.

Essa forma de arquitetura permite que diferentes estruturas e técnicas de filtragem sejam avaliadas. A modularização das partes do sistema permite que apenas o bloco “Processamento Sinal” precise ser alterado para que seja utilizada uma implementação de filtro diferente. Com essa estratégia, toda a estrutura de conversão dos sinais de entrada e saída, configuração e controle não precisa ser modificada.

4.2 Processamento Paralelo e Diferentes Domínios de *Clock*

A técnica de paralelismo teve como objetivo reduzir o tempo de processamento do sistema de reconstrução dos sinais de energia do TileCal. Foi possível dividir o funcionamento do sistema em três estágios e torná-los independentes. Os estágios são: conversão dos sinais de entrada de serial em paralelo; processamento das amostras de energia; conversão do sinal de saída, as amostras reconstruídas, de paralelo em serial. Na figura 4.7 são ilustrados os estágios desse paralelismo.

Como pode ser observado na Figura 4.7, a saída de cada estágio é armazenada em um registrador, representado pelos blocos REG. Com isso, quando cada estágio

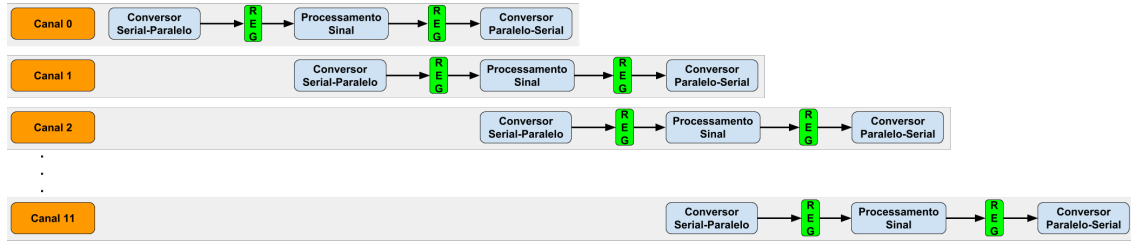


Figura 4.7: Diagrama dos Estágios do Paralelismo.

finaliza o seu processamento, sua saída é armazenada em um registrador, ficando disponível para o estágio seguinte. Dessa forma, o bloco fica disponível para processar outros sinais, independentemente do estado de processamento dos demais estágios.

O paralelismo garante que o sistema não precisa fazer todo o processamento do sinal de um determinado canal para que as informações de um outro sensor possam, também, serem processados. Isso garante que os três estágios funcionem de forma paralela, ou seja, enquanto a informação de entrada referente a um canal está sendo convertido, o sinal de energia de outro sensor pode ser processado no segundo estágio. Consequentemente, uma outra amostra, que já passou pelo processo de filtragem, pode ser convertida em serial e disponibilizada na saída do sistema.

Outra solução também adotada no projeto foi a utilização de diferentes domínios de *clock*. Caso todos os estágios funcionem com uma mesma frequência de *clock*, como a quantidade de bits de entrada e de saída são as mesmas, seria necessário criar uma estrutura de armazenamento no terceiro estágio. Como ambas as conversões utilizam a mesma quantidade de pulsos, ao longo da execução do sistema, o último estágio não estaria disponível para fazer a conversão quando os sinais estivessem sendo filtrados. Dessa forma, seria necessário armazenar esses sinais em alguma estrutura até que pudessem ser disponibilizadas na saída do sistema.

Para evitar a utilização de mais recursos do FPGA e, também, que esse problema se tornasse um gargalo no tempo de processamento do sistema, decidiu-se fazer com que esse terceiro estágio funcione com uma frequência de *clock* duas vezes mais rápida. Com isso, garante-se que esse bloco finalizará a conversão do sinal de saída mais rápido que os demais estágios. Ou seja, quando um sinal de energia filtrado for disponibilizado pelo segundo estágio, o último bloco estará pronto para fazer a conversão.

4.3 Implementação do Método Determinístico

O método determinístico consiste na aplicação da deconvolução, ou seja, uma filtragem inversa utilizando filtros de Resposta ao Impulso Finita (FIR) [13]. Esse filtro é obtido através da resposta ao impulso que modela o sistema eletrônico do *TileCal*, encontrando, assim, uma aproximação por um filtro FIR realizável.

O projeto do hardware dessa técnica consiste na implementação da expressão que define o filtro FIR, apresentada na Equação 4.1.

$$y[n] = x[n] * h[n] = \sum_{K=0}^{L-1} h[k]x[n-k] \quad (4.1)$$

De forma geral, apresenta-se como um somatório de produtos. A estrutura escolhida para ser implementada foi a forma Transposta do Filtro FIR [60], como apresentado na seção 3.5.1. Na Figura 4.8 apresenta-se um diagrama contendo as representações das entradas e saídas desse bloco. As entradas representam os sinais de *clock*, *reset*, palavra de configuração, sinal de energia e coeficientes. A ordem máxima desse filtro foi definida em 30, possuindo 31 entradas referentes aos coeficientes. A entrada “Config” representa a configuração de funcionamento do filtro, nesse caso, define a ordem do mesmo. Pode-se alterar a ordem de operação do filtro FIR externamente, através do bloco “Configura Filtros”, sem a necessidade de reprojetá-lo.

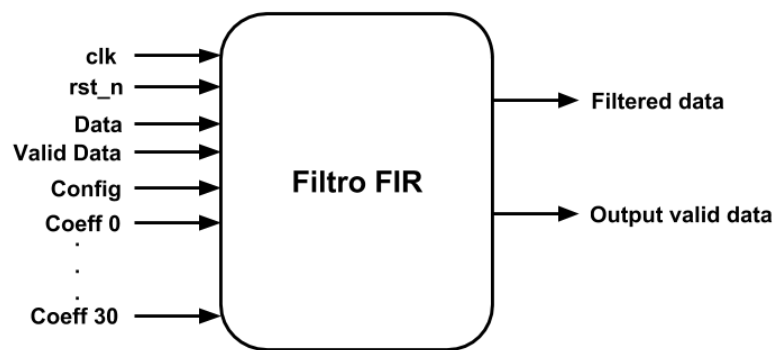


Figura 4.8: Diagrama da Implementação em hardware do método determinístico.

O sistema deve realizar o processamento das amostras de até 12 sensores. Para evitar que seja necessário ter 12 instâncias de implementação FIR, foi projetada uma arquitetura do fluxo dos sinais de forma que fosse utilizada apenas uma instância. Essa estratégia teve como objetivo evitar que as estruturas do FPGA, responsáveis

pelas operações aritméticas, fossem replicadas a fim de reduzir a utilização desses recursos. O diagrama desse fluxo das informações está ilustrado na Figura 4.9.

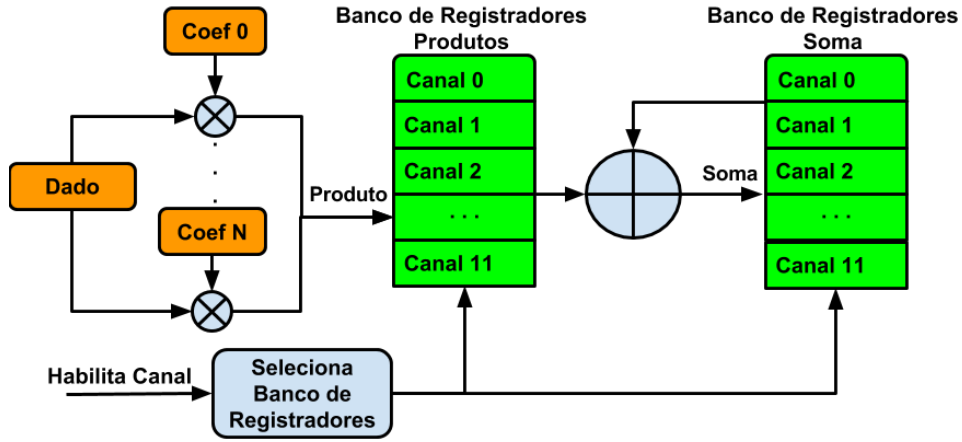


Figura 4.9: Diagrama da arquitetura da implementação do filtro FIR.

Como pode ser observado na Figura 4.9, foram utilizados dois bancos de registradores para armazenar os resultados das operações de multiplicação e soma referentes a cada canal. Através dessa lógica é possível reutilizar as mesmas estruturas aritméticas para efetuar o processamento das amostras de todos os sensores. Com isso, garante-se o correto funcionamento da forma transposta do filtro FIR, pois as amostras atrasadas de cada canal permanecem armazenadas para acessos posteriores.

A estrutura de paralelismo implementada nesse projeto garante que, quando a amostra de um sensor tiver sido convertida em paralelo no primeiro estágio, o bloco “Filtro FIR” já vai ter finalizado a filtragem de um outro sinal. Isso ocorre porque a quantidade de ciclos necessários para a execução do segundo estágio é menor do que o primeiro. Torna-se, portanto, desnecessário ter estruturas de soma e multiplicação exclusivas para cada canal, reduzindo a utilização das estruturas aritméticas do FPGA e tendo melhores resultados na síntese.

4.4 Implementação do Gradiente Descendente Positivo

No escopo desse trabalho, o método iterativo implementado foi o algoritmo do Gradiente Descendente Positivo. No estudo realizado em [12], os dois algoritmos que

tiveram os melhores desempenhos foram o Gradiente Descendente que Exclui Zeros e o Descendente Positivo. A diferença entre o funcionamento de ambos é que no primeiro a matriz resultante de cada iteração é reduzida, enquanto que no segundo essa matriz permanece com a mesma dimensão mas suas componentes negativas são zeradas. Em termos de implementação em hardware, o benefício do método que exclui zeros não seria possível de ser aproveitado. E, por isso, foi utilizado o método Descendente Positivo.

Conforme apresentado na seção 3.5.3.2, esse algoritmo define que para os valores negativos seja atribuído zero. Portanto, a cada iteração, as posições da matriz resultado que possuírem valores negativos devem ser zeradas. A equação geral que define esse método é apresentada na Equação 4.2.

$$\mathbf{X}^{iter+1} = \mathbf{X}^{iter} + \mu \mathbf{H}^T (\mathbf{r} - \mathbf{H} \mathbf{X}^{iter}) \quad (4.2)$$

A complexidade computacional inerente ao método, devido às operações envolvendo matrizes, faz com que sua implementação em FPGA torne-se um processo com relevante custo de hardware. A descrição da equação 4.2 de forma direta em Verilog é inviável para essa aplicação, pois no processo de síntese iria gerar hardware não otimizado e ineficiente, com o caminho dos sinais muito extenso, o que exigiria tempo de processamento elevado. Para segmentar o processamento da informação a fim de alcançar os requisitos de *timing*, e ocupação do FPGA, decidiu-se dividir a Eq. 4.2 em 5 partes, executadas em momentos distintos. As equações parciais são apresentadas à seguir:

$$Eq1 = \mu \mathbf{H}^T \quad (4.3)$$

$$Eq2 = \mathbf{H} \mathbf{X}^{iter} \quad (4.4)$$

$$Eq3 = \mathbf{r} - Eq2 \quad (4.5)$$

$$Eq4 = Eq2 \times Eq3 \quad (4.6)$$

$$X^{iter+1} = \mathbf{X}^{iter} + Eq4 \quad (4.7)$$

O controle de fluxo da informação foi realizado através da implementação de uma Máquina de Estados Finita (FSM, do inglês *Finite State Machine*), apresentada na Figura 4.10. Cada estado habilita a execução de uma das etapas do algoritmo do Gradiente Descendente Positivo. Com essa estratégia, pôde-se reduzir o fluxo dos sinais, possibilitando que o circuito suportasse uma frequência de *clock* maior. Na Tabela 4.1 apresenta-se os detalhes referente às transições dos estados e saídas.

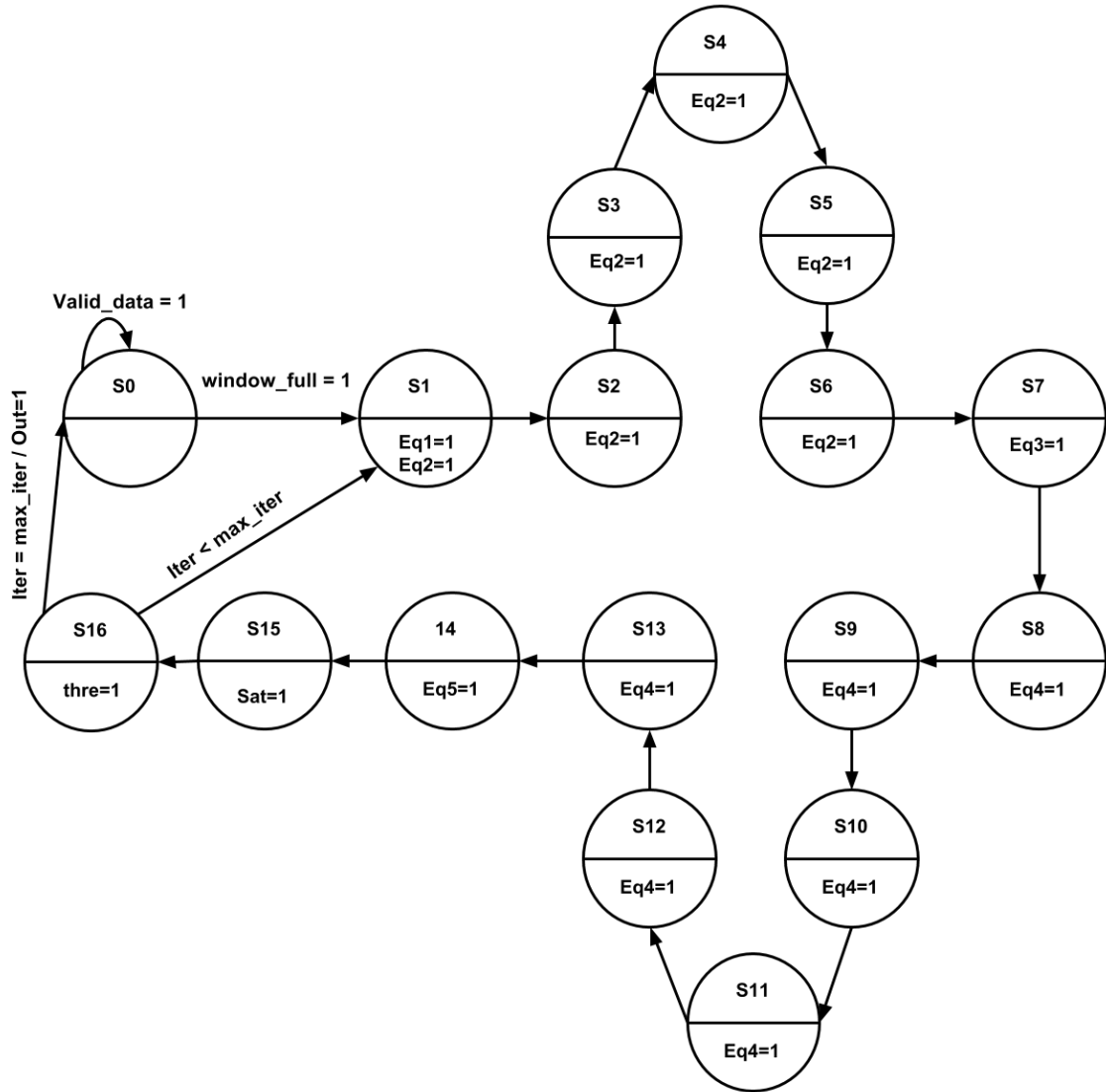


Figura 4.10: Diagrama da Máquina de Estados Finita utilizada para o método iterativo.

O estado S0 é responsável por controlar o armazenamento dos sinais de energia

Tabela 4.1: Transição de Estados e Saídas

Estado Atual	Entradas	Próximo Estado	Saída
S0	Valid_Data = 1 e Window_full = 0	S0	x
S0	Window_full = 1	S1	x
S1	x	S2	Eq1 = 1 e Eq2 = 1
S2	x	S3	Eq1 = 0 e Eq2 = 1
S3	x	S4	Eq1 = 0 e Eq2 = 1
S4	x	S5	Eq1 = 0 e Eq2 = 1
S5	x	S6	Eq1 = 0 e Eq2 = 1
S6	x	S7	Eq1 = 0 e Eq2 = 1
S7	x	S8	Eq3 = 1
S8	x	S9	Eq4 = 1
S9	x	S10	Eq4 = 1
S10	x	S11	Eq4 = 1
S11	x	S12	Eq4 = 1
S12	x	S13	Eq4 = 1
S13	x	S14	Eq4 = 1
S14	x	S15	Eq5 = 1
S15	x	S16	Saturation = 1
S16	Iter < Iter_max	S1	Threshold = 1 e Output = 0
S16	Iter = Iter_max	S0	Threshold = 1 e Output = 1

de uma determinada janela. Até que seja recebida a quantidade de bits da janela, a máquina de estados permanece em S0. Quando alcança a quantidade de amostras, o controle passa para o estado S1.

O estado S1 representa o início do processamento do algoritmo. Nesse estado, realiza-se a execução das equações (4.3) e (4.4). A segunda equação representa uma multiplicação entre duas matrizes, sendo um dos gargalos do processamento devido à geração de um caminho de informação com elevada quantidade de elementos lógicos, caso fosse projetada para execução em apenas um ciclo de *clock*. Para evitar perda de desempenho em termos de *timing*, decidiu-se realizá-las em seis diferentes estados, utilizando seis pulsos de *clock*. No primeiro estado realiza-se as operações de multiplicação entre os elementos das matrizes e nos outros cinco, de S2 até S6, executa-se as operações de adição.

O estado S7 controla a execução da equação 4.5. Nos estados S8 a S13 são realizados os cálculos referente a equação 4.6. Esta possui a mesma complexidade

da equação 4.4 e, dessa forma, adotou-se a mesma estratégia para que se obtivesse um melhor desempenho em relação ao tempo de processamento.

Por último, no estado S14 executa-se a equação 4.7, resultando na matriz de saída do método em uma determinada iteração. O estado S15 controla a aplicação da técnica de saturação, onde os elementos da matriz resultante são analisados e saturados para serem representados na quantidade de bits de saída do sistema.

Por fim, o estado S16 controla a aplicação do limiar do algoritmo, ou seja, zera os elementos da matriz resultante que tiverem valores negativos. Nesse estado, também é realizado o controle da quantidade de iterações que foram executadas com uma determinada janela. Quando a quantidade máxima de iterações é alcançada, finaliza-se o processamento dessa janela e habilita-se o sinal que controla a saída do resultado da janela processada.

4.5 Verificação Funcional

Com o intuito de validar o correto funcionamento do processo de filtragem dos sinais de energia do *TileCal*, foi desenvolvida a verificação funcional dos métodos implementados. A Figura 4.11 apresenta o ambiente de verificação projetado. A linguagem utilizada para a implementação desse ambiente foi *SystemVerilog*.

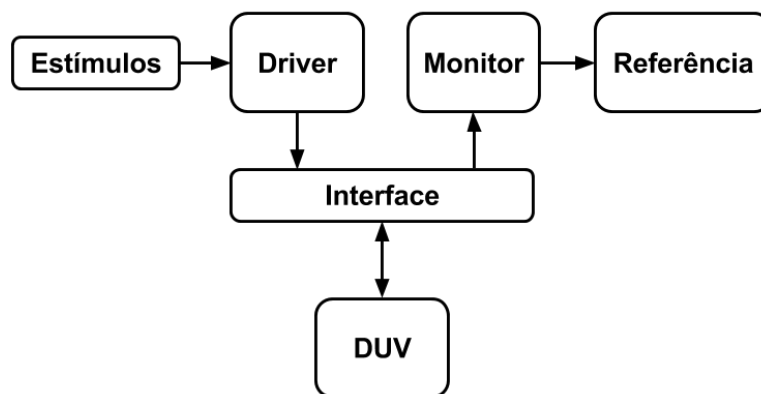


Figura 4.11: Arquitetura da Verificação Funcional.

Na Figura 4.11, o bloco denominado de *Design Under Verification* (DUV) representa o *Register-Transfer Level* (RTL) que está sendo verificado, ou seja, os métodos de filtragem implementados em hardware nesse projeto. A Interface representa a conexão do RTL com os demais elementos da verificação, garantindo que os estímulos

e as respostas do projeto possam ser controlados pelo ambiente.

Os Estímulos representam os padrões que são enviados para o DUV. Esses valores representam os sinais de entrada utilizados pelo *hardware* para executar o processo de filtragem. O *Driver* é responsável por identificar os estímulos e enviar para o RTL respeitando seu protocolo de funcionamento.

O Monitor é responsável por observar as saídas do RTL. Identifica, de acordo com o protocolo, cada amostra de energia processada, gerada pelo DUV, e envia esses valores para a Referência. Esse último bloco é responsável por validar essa saída através da comparação com o resultado gerado pelo modelo de referência. Esse modelo foi desenvolvido em Matlab, implementando o mesmo método de filtragem utilizado pelo RTL. Logo, através desse processo de validação é possível identificar se o comportamento do RTL, tanto em relação ao seu protocolo de comunicação, como à reconstrução dos sinais de energia, está de acordo com o esperado.

Capítulo 5

Resultados

Nesse capítulo são apresentados os resultados obtidos na implementação em hardware dos métodos de estimação de energia para o primeiro nível de *Trigger* do *TileCal*. Serão expostos tanto os resultados de simulação, para avaliação do desempenho dos algoritmos na reconstrução do sinal, como também os resultados da síntese em FPGA, para a análise da viabilidade da prototipação dessa solução nos dispositivos que serão utilizados na nova arquitetura do sistema eletrônico do *TileCal*.

5.1 Geração dos Sinais Simulados

Para a execução das simulações do hardware desenvolvido e a validação dos resultados encontrados, foi necessário utilizar um simulador capaz de gerar sinais de energia do *TileCal*. Nesse sentido, foi utilizado uma *Toy Monte Carlo Simulation*, que gera sinais com as características de sobreposição similares aos encontrados para o *TileCal* [69]. O uso de sinais simulados é fundamental para os testes da arquitetura proposta, pois como a configuração experimental do detector para qual o sistema está sendo projetado ainda não está operacional, não existem sinais experimentais disponíveis.

Na geração do pulso característico do *TileCal*, utilizado para o desenvolvimento dos filtros propostos, foi considerado um conjunto normalizado de 7 amostras, correspondentes à resposta ao impulso de um canal do Calorímetro. Na Figura 5.1 observa-se a representação gráfica desses 7 pontos com fase zero.

O simulador considera, ainda, o desvio de fase no momento da aquisição do sinal

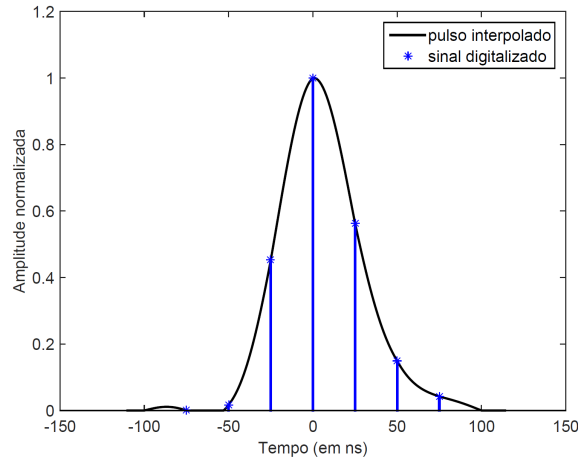


Figura 5.1: Representação gráfica do pulso característico do *TileCal* [12].

devido aos diferentes tempos de voo (*time of flight*) das partículas nos eventos de colisão. Nesse sentido, no *TileCal* considera-se a possibilidade da ocorrência de um desvio de fase em uma faixa de ± 5 ns (com distribuição uniforme). Na Figura 5.2 observa-se a representação gráfica de exemplos do pulso característico com desvio de fase de ± 5 ns.

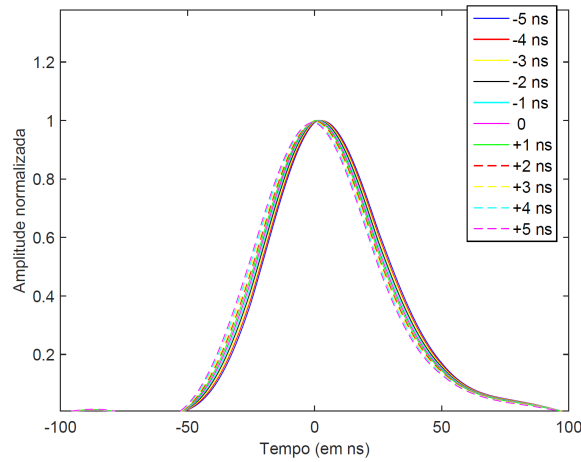


Figura 5.2: Representação gráfica de exemplos do pulso característico do *TileCal* com desvio de fase [12].

O simulador foi desenvolvido em Matlab e gera tanto sinais afetados pelo *pileup* como também as informações de energia verdadeira, considerando o cenário de funcionamento do *TileCal* sem a sobreposição. O *script* foi construído de forma parametrizável, permitindo a geração de sinais com diferentes configurações de operação. Os parâmetros são: Quantidade de amostras, Amplitude máxima e Ocupação. O

primeiro define a quantidade de *bunch crossing*, ou seja, de amostras de energia que serão gerados, tanto dos sinais distorcidos como de seus respectivos valores verdadeiros. O segundo parâmetro define o valor da amplitude máxima assumido pelos sinais simulados gerados. O terceiro, e último, representa a porcentagem de *bunch crossing* com ruído de *pileup*.

O funcionamento do simulador pode ser dividido em alguns passos. Inicialmente gera-se a quantidade de amostras desejada de ruído branco e gaussiano considerando os sinais digitalizados a 40 MHz; Em seguida, a partir da ocupância, sorteia-se as posições onde ocorreram as deposições de energia; O próximo passo é a realização do sorteio de amplitudes de energia. Em seguida, realiza-se o sorteio para a simulação do desvio de fase, considerando a faixa de ± 5 ns. No último passo, os pulsos característicos, formados com as energias sorteadas e o desvio de fase, são sobrepostos considerando que as amostras centrais estão localizadas nos *bunch crossing* sorteados.

5.2 Método Determinístico

Nessa seção serão apresentados os resultados obtidos com a implementação em FPGA da solução proposta nesse trabalho considerando o método determinístico como técnica para a estimação da energia. Nesse cenário, considerou-se a arquitetura apresentada na seção 4.3.

5.2.1 Simulação do Método Determinístico

Para a execução das simulações, foi utilizada a arquitetura da verificação funcional descrita na seção 4.5. Dessa forma, garantiu-se que os sinais seriam enviados ao hardware considerando o protocolo previamente definido, da mesma forma que as saídas foram monitoradas considerando o correto comportamento do protocolo de saída.

Para a execução das simulações em FPGA, os dados de entrada precisam ser representados em ponto fixo. A representação, tanto dos coeficientes como dos sinais de energia, foi definida em complemento de dois. Sendo assim, foram analisados os conjuntos de dados simulados e definida a quantidade de bits para representação

da parte inteira e fracionária do valor real. Além disso, para evitar *overflow*, foi utilizada a técnica de saturação [70].

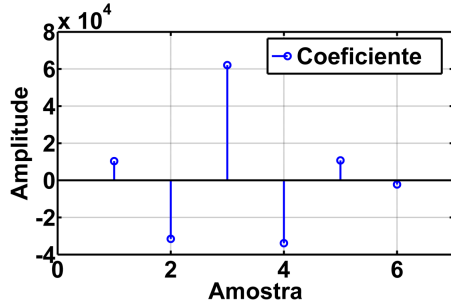
Para a validação do projeto, foram executadas simulações com diferentes cenários, alterando a ordem do filtro e utilizando o mesmo conjunto de sinais simulados. Os sinais de energia foram representados com 12 bits e os coeficientes com 18 bits, sendo esta resolução definida de acordo com análises preliminares dos valores simulados com o objetivo de reduzir as perdas de informação nesse processo de representação. As ordens definidas para a execução das simulações foram: 5, 10, 15, 20 e 30, sendo os coeficientes utilizados ilustrados nas Figuras 5.3a 5.3b 5.3c 5.3d 5.3e.

Através da Figura 5.4 pode-se observar exemplos de sinais gerados pelo simulador. Na Figura 5.4a é apresentado um gráfico com os sinais simulados ideais, ou seja, energias sem a distorção e na Figura 5.4b é apresentado o mesmo intervalo de sinais com o efeito de *pileup*. As amplitudes de energias apresentadas nesses gráficos estão em sua forma real. Em relação aos parâmetros do simulador, foi definida a ocupação de 10% e a amplitude máxima em 500.

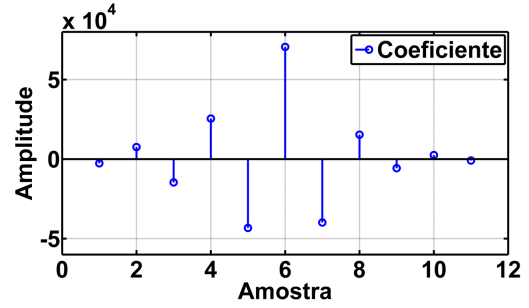
Após a execução das simulações, considerando todos os cenários propostos, verificou-se a eficiência da reconstrução do sinal graficamente. Nas Figuras 5.5a, 5.5b, 5.5c, 5.5d e 5.5e pode-se observar os gráficos construídos a partir da sobreposição do resultado da implementação FPGA com os sinais ideais de energia gerados pelo simulador.

Foram realizadas, também, análises referentes ao erro da implementação FPGA. Na Tabela 5.1 são apresentados o valor médio do erro e a raiz quadrada do erro quadrático médio da implementação FPGA. Para a geração desses dados foram utilizados os resultados dos filtros implementados em FPGA e os sinais de energia ideais gerados pelo simulador.

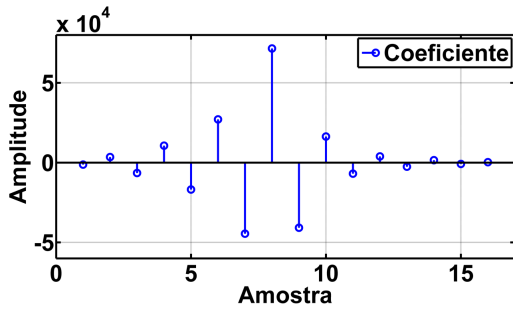
Ainda em relação a análise do erro, foram construídos gráficos de dispersão para cada cenário de simulação realizado. Esses gráficos podem ser observados nas Figuras 5.6a, 5.6b, 5.6c, 5.6d e 5.6e.



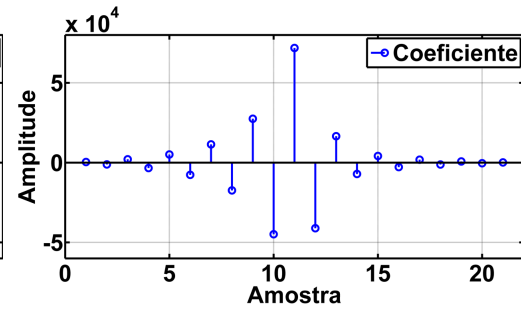
(a) Representação gráfica dos coeficientes do Filtro de Ordem 5.



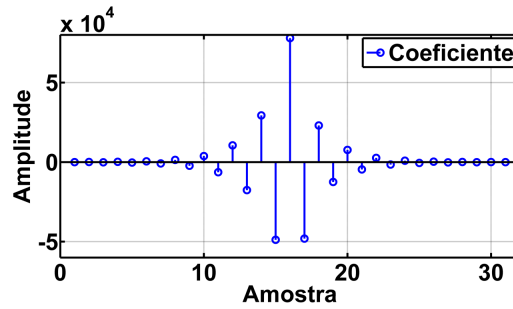
(b) Representação gráfica dos coeficientes do Filtro de Ordem 10.



(c) Representação gráfica dos coeficientes do Filtro de Ordem 15.

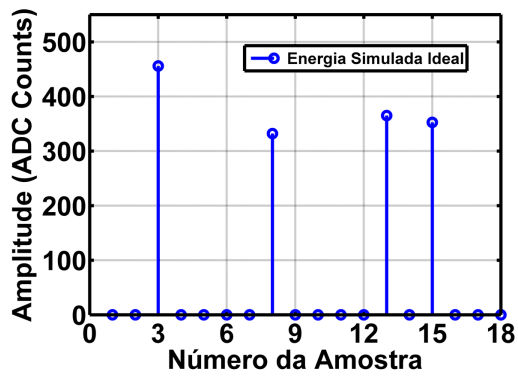


(d) Representação gráfica dos coeficientes do Filtro de Ordem 20.

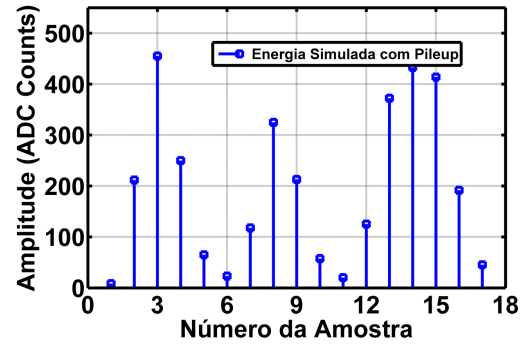


(e) Representação gráfica dos coeficientes do Filtro de Ordem 30.

Figura 5.3: Representação gráfica dos coeficientes nas diferentes ordens utilizadas na simulação.



(a) Representação gráfica de sinais de energia sem empilhamento.

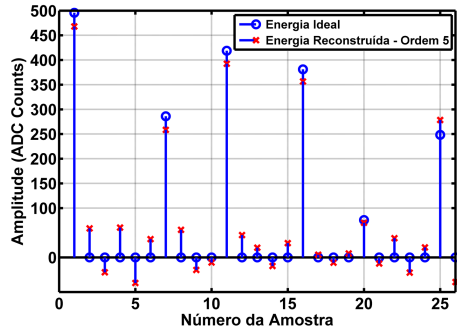


(b) Representação gráfica de sinais de energia com efeito de empilhamento.

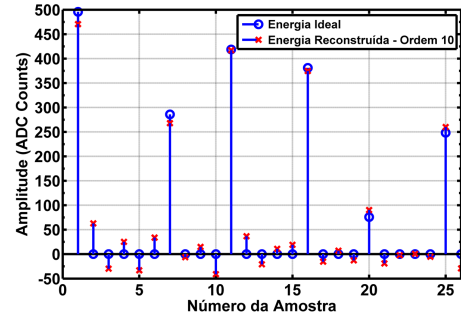
Figura 5.4: Representação gráfica de sinais de energia simulados com e sem o efeito de empilhamento.

Tabela 5.1: Valores do erro dos resultados da simulação FPGA

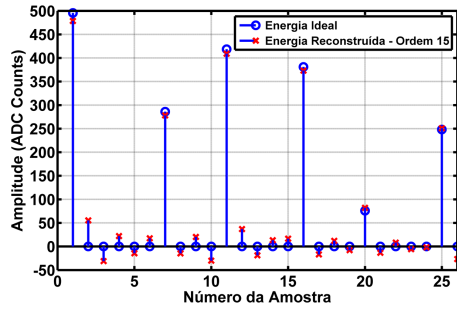
Ordem do Filtro	Valor médio	Raiz quadrada do erro quadrático médio
Ordem 5	14,44	24,95
Ordem 10	12,33	19,86
Ordem 15	11,93	19,33
Ordem 20	11,82	19,21
Ordem 30	12,39	20,30



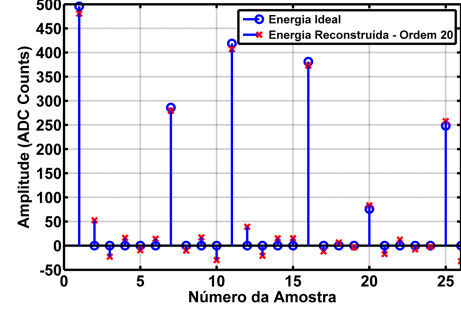
(a) Aplicação do filtro de ordem 5.



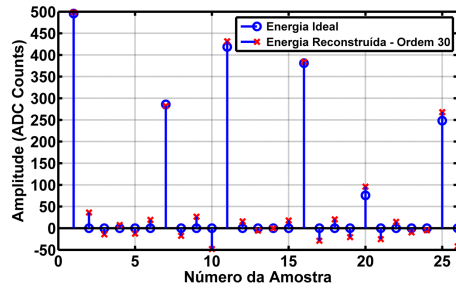
(b) Aplicação do filtro de ordem 10.



(c) Aplicação do filtro de ordem 15.

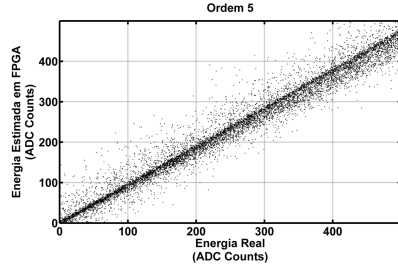


(d) Aplicação do filtro de ordem 20.

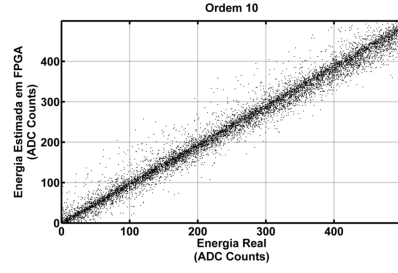


(e) Aplicação do filtro de ordem 30.

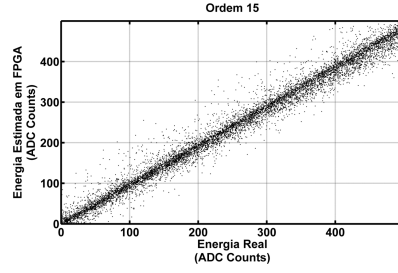
Figura 5.5: Representação gráfica da aplicação do filtro com diferentes ordens no FPGA.



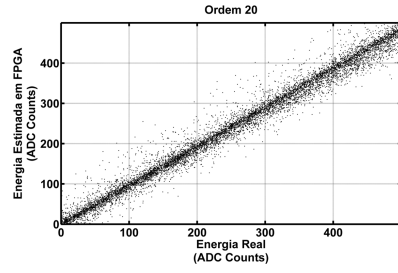
(a) Dispersão com filtro de ordem 5.



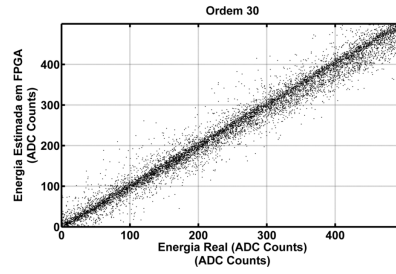
(b) Dispersão com filtro de ordem 10.



(c) Dispersão com filtro de ordem 15.



(d) Dispersão com filtro de ordem 20.



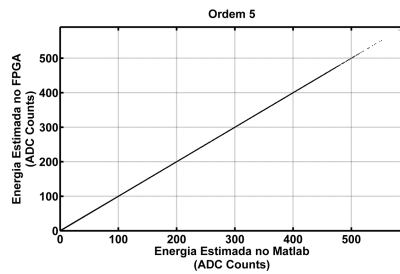
(e) Dispersão com filtro de ordem 30.

Figura 5.6: Gráficos da dispersão entre a energia ideal e a estimada em FPGA.

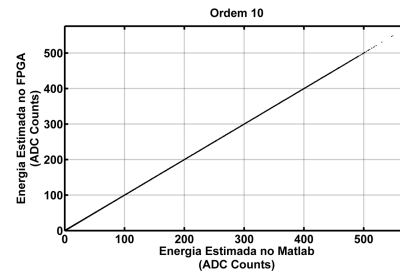
Para avaliar a reconstrução do sinal em FPGA, foi utilizado como modelo de referência a função nativa do MATLAB[®] que implementa o filtro digital FIR. A partir dos resultados obtidos, foram realizadas as análises do erro entre os sinais reconstruídos em ambas as implementações. Na Tabela 5.2 foram apresentados os resultados referentes ao valor médio do erro e à raiz quadrada do erro quadrático médio, enquanto que as Figuras 5.7a, 5.7b, 5.7c, 5.7d e 5.7e apresentam os gráficos de dispersão para as simulações de ordem 5, 10, 15, 20 e 30 respectivamente.

Tabela 5.2: Valores do erro entre os resultados da simulação FPGA e o modelo de referência (MATLAB®)

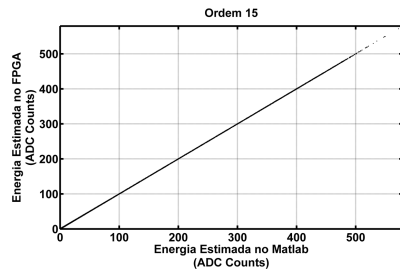
Ordem do Filtro	Valor médio	Raiz quadrada do erro quadrático médio
Ordem 5	0,30	0,37
Ordem 10	0,36	0,45
Ordem 15	0,37	0,46
Ordem 20	0,38	0,47
Ordem 30	0,42	0,52



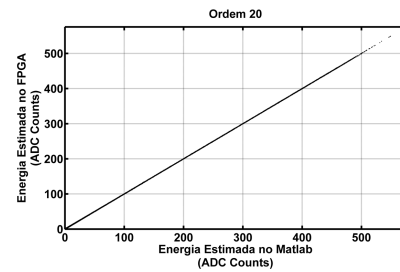
(a) Dispersão com filtro de ordem 5.



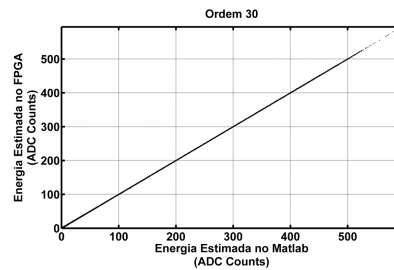
(b) Dispersão com filtro de ordem 10.



(c) Dispersão com filtro de ordem 15.



(d) Dispersão com filtro de ordem 20.



(e) Dispersão com filtro de ordem 30.

Figura 5.7: Gráficos da dispersão entre a energia estimada no FPGA e no MATLAB®.

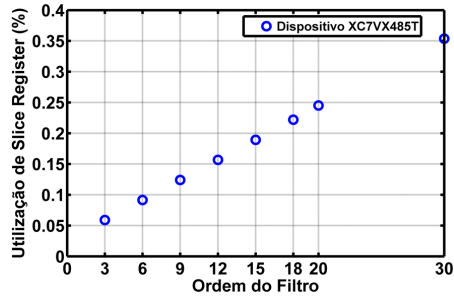
5.2.2 Síntese FPGA do Método Determinístico

Além da fase de simulação, foi desenvolvido, também, o processo de síntese FPGA do projeto considerando o método determinístico de estimação de energia. O FPGA definido para ser usado na nova eletrônica do *TileCal* é o XC7VX485T, da família Virtex-7 da Xilinx. Sendo assim, a síntese foi executada para esse dispositivo.

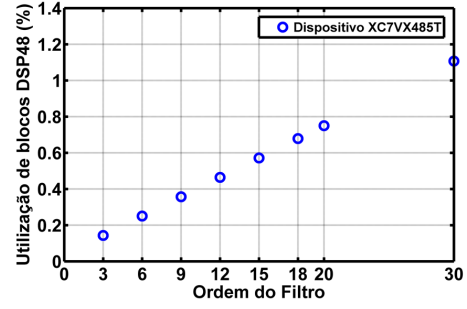
Em relação ao comportamento temporal, o relatório gerado na síntese atestou que o pior caso em relação ao caminho do sinal foi o tempo entre o *Clock-to-Q*, responsável pela estabilização do coeficiente em seu respectivo registrador, e a multiplicação desse valor pelo sinal de energia, executada através da estrutura DSP48E1 do FPGA. Em uma primeira versão, esse caminho foi sintetizado com um atraso de 3,351 ns. Após a investigação da arquitetura e funcionalidades da estrutura de DSP48E1 do FPGA, identificou-se que existem formas de melhorar o desempenho da multiplicação através da utilização de, pelo menos, três estágios de *pipeline*. Dessa forma, foram adicionados registradores de *pipeline* na descrição do hardware para que a ferramenta de síntese utilizasse esse recurso para reduzir o atraso do caminho crítico. Com essas alterações, o período de *clock* mínimo para a operação de multiplicação foi reduzido para 1,879 ns, sendo 76,2 % para a lógica e 23,8 % para o roteamento.

As operações aritméticas dessa arquitetura são realizadas no domínio de *clock* mais lento, logo, a definição das frequências baseia-se na análise do caminho crítico desse domínio. A partir do relatório obtido na síntese, define-se que o sistema suporta uma frequência máxima de 532 MHz para o domínio do *clock* mais lento e 650 MHz para o *clock* mais rápido. A arquitetura foi definida considerando que um domínio de *clock* é duas vezes mais rápido que o outro, logo, foi utilizada as frequências de 325 MHz e 650 MHz para os diferentes domínios.

Com a realização da síntese foi possível analisar a ocupação dos recursos do dispositivo FPGA. Na Figura 5.8a ilustra-se a utilização da lógica configurável (*Slice Register*), enquanto que a Figura 5.8b ilustra a utilização das estruturas de operações lógicas e aritméticas (DSP48). Em ambas variou-se a ordem do filtro.



(a) Utilização de *Slice Register*.



(b) Utilização de estruturas *DSP48*.

Figura 5.8: Representação gráfica da utilização dos recursos lógicos na implementação do método determinístico

Na Tabela 5.3 são apresentadas as informações de consumo de potência extraídas da síntese. Na segunda coluna pode-se observar os resultados em relação à potência estática, na terceira a dinâmica, e, por fim, na quarta o consumo total.

Tabela 5.3: Relatório de potência na síntese.

Ordem do Filtro	Estática (mW)	Dinâmica (mW)	Total (mW)
Ordem 5	241,72	89,83	331,55
Ordem 10	241,92	112,65	354,57
Ordem 15	242,16	139,99	382,15
Ordem 20	242,53	181,13	423,66
Ordem 30	242,65	194,57	437,22

5.3 Método Iterativo: Gradiente Descendente Positivo

Nessa seção serão abordados os resultados obtidos com a implementação do método iterativo de estimação de energia considerando a arquitetura apresentada na seção 4.4. Dentre os métodos iterativos propostos, foi implementado o Gradiente Descendente Positivo, conforme descrito na seção 3.5.3.2.

5.3.1 Simulação do Método Iterativo

A simulação do método iterativo também considerou a estrutura do ambiente de verificação apresentado na seção 4.5. Os sinais de entrada, gerados pelo simulador, foram representados em ponto fixo utilizando a representação em complemento de dois e a técnica de saturação para evitar o *overflow*.

As informações de energia afetadas pelo *pileup* nessa simulação foram representadas em ponto fixo com resolução de 12 bits. Além desse sinal, esse método exige a utilização de um fator de relaxamento, o μ . Com o objetivo de tornar o projeto viável para ser implementado em *hardware*, foi decidido não realizar o cálculo desse fator a cada iteração, considerando-o como um valor constante definido na Equação 5.1, onde λ_{max} representa o maior autovalor de $H^T H$. O fator de relaxamento também foi representado com 12 bits.

$$\mu = \frac{2}{\lambda_{max}} \quad (5.1)$$

A estimação da energia através do Gradiente Descendente Positivo utiliza a informação referente ao pulso característico do *TileCal*, responsável por compor os elementos da matriz H . Esse pulso, ilustrado na Figura 5.9, é formado por 7 amostras, representadas com 12 bits e, nessas simulações, sem desvio de fase.

O método iterativo exige que sua implementação seja realizado em janelas, ou seja, o processamento para estimação de energia é realizado considerando um determinado conjunto de sinais. De acordo com os estudos realizados, o tamanho da janela em que o erro estabiliza no Gradiente Descendente Positivo é 25 e, portanto, nesse trabalho foi utilizado esse valor.

Para avaliar o desempenho desse método implementado em FPGA foram realizadas simulações considerando diferentes cenários, alterando a quantidade de iterações e utilizando o mesmo conjunto de informações de energia. Foi considerada até 100 iterações, aumentando de 5 em 5. Na Figura 5.10a observa-se a reconstrução do sinal no cenário com 5 iterações, enquanto que na Figura 5.10b utilizou-se 100 iterações.

A partir dos resultados obtidos, esse algoritmo foi avaliado através da análise da raiz quadrada do erro quadrático médio. Na Figura 5.11 é apresentado graficamente a relação entre a raiz quadrada do erro quadrático médio e o número de iterações na reconstrução do sinal. A partir desse resultado, observou-se que o erro estabilizava

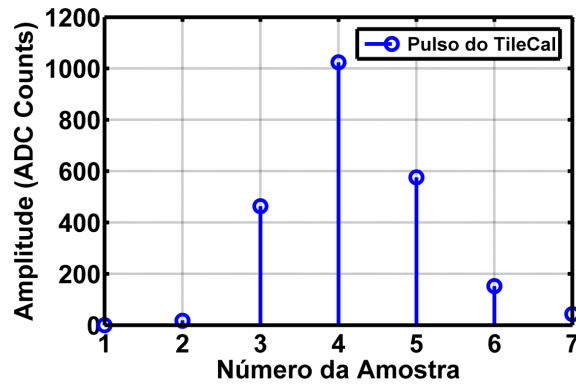
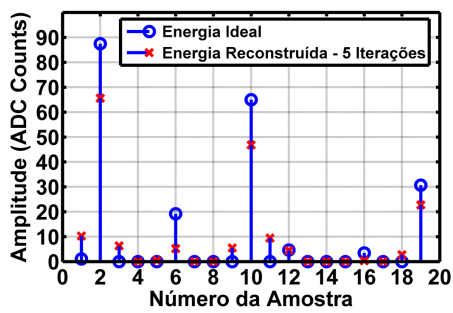
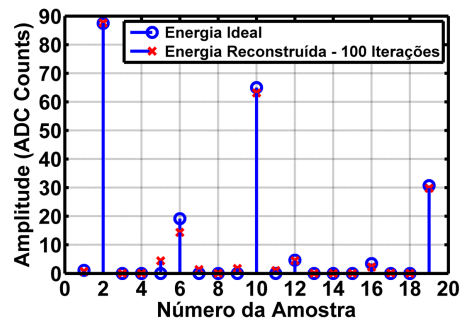


Figura 5.9: Representação gráfica do pulso característico do *TileCal* utilizado na simulação do método iterativo.



(a) Sinal reconstruído com 5 iterações.



(b) Sinal reconstruído com 100 iterações.

Figura 5.10: Representação gráfica do sinal reconstruído com diferentes quantidade de iterações.

entre 45 e 50 iterações. Sendo assim, não foram feitas análises com quantidade de iterações superior a 100. As informações utilizadas nessas análises foram os resultados obtidos com a execução da simulação do projeto em FPGA e os sinais de energia ideais fornecidos pelo simulador.

Na Tabela 5.4, além da raiz quadrada do erro quadrático médio, é apresentado o resultado referente ao valor médio do erro da implementação FPGA variando-se a quantidade de iterações. Para a geração desses dados foram utilizados os resultados da implementação do método iterativo em FPGA e os sinais de energia ideais gerados pelo simulador.

Para complementar a análise do erro da implementação FPGA, foram desenvolvidos gráficos de dispersão para alguns cenários de simulação. Esses gráficos podem ser observados nas Figuras 5.12a, 5.12b, 5.12c, 5.12d, 5.12e, 5.12f, 5.12g e 5.12h.

Para esse método foi desenvolvido um modelo de referência, também em MATLAB®. A validação do desempenho do projeto de hardware desenvolvido para essa técnica foi realizada através da comparação entre os resultados da simulação em FPGA e da referência. Com essas informações, foram feitas análises da raiz quadrada do erro quadrático médio, apresentadas na Figura 5.13.

Na Tabela 5.5 são apresentados os resultados referente ao valor médio do erro e a raiz quadrada do erro quadrático médio, enquanto que nas Figuras 5.14a, 5.14b, 5.14c, 5.14d, 5.14e, 5.14f, 5.14g e 5.14h foram apresentados os gráficos de dispersão. Para a geração desses dados foram utilizados os resultados da implementação do gradiente descendente positivo em FPGA e os sinais de energia estimados com o modelo de referência.

5.3.2 Síntese FPGA do Método Iterativo

Com o intuito de analisar a viabilidade da implementação desse método de estimação no sistema eletrônico do *TileCal*, além da simulação funcional, foi desenvolvida também a síntese. Através desse processo é possível avaliar se a solução atende aos requisitos de tempo de processamento e ocupação do FPGA. O dispositivo utilizado nesse procedimento foi o XC7VX485T, da família Virtex-7 da Xilinx.

Como essa abordagem envolve operações entre matrizes, o ponto crítico do fluxo do sinal nesse hardware localiza-se nas multiplicações necessárias para se obter a

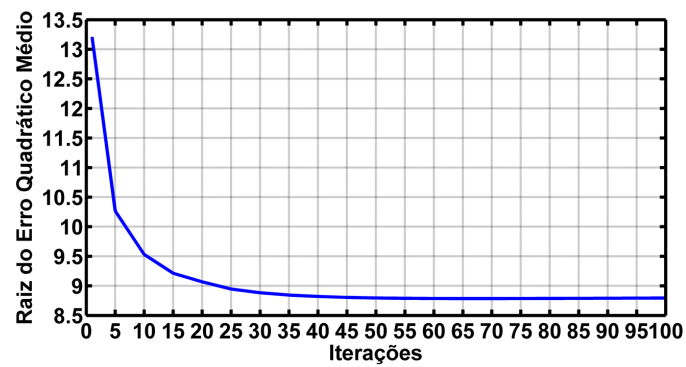
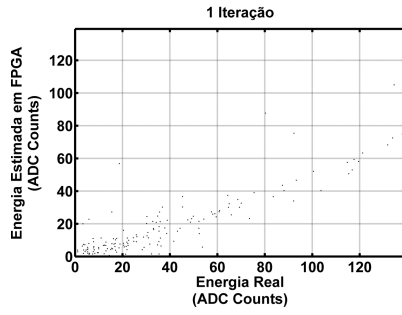


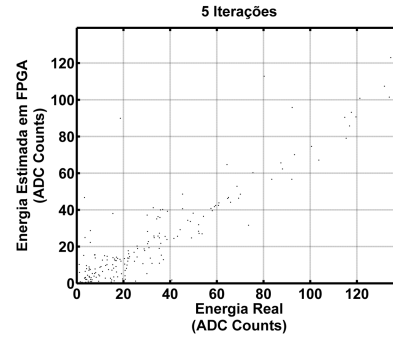
Figura 5.11: Representação gráfica da relação entre a raiz quadrada do erro quadrático médio e o número de iterações.

Tabela 5.4: Valores do erro dos resultados da simulação FPGA no método iterativo.

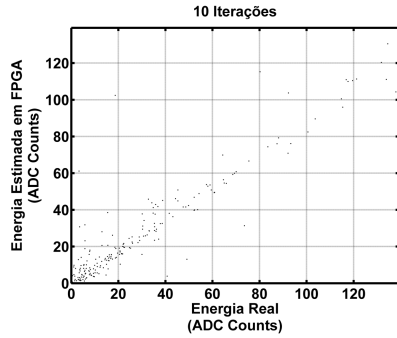
Quantidade de Iterações	Valor médio	Raiz quadrada do erro quadrático médio
1	5,65	13,20
5	4,74	10,26
10	4,49	9,52
15	3,69	9,21
20	3,48	9,06
25	3,19	8,94
30	3,08	8,88
40	2,88	8,82
50	2,77	8,79
60	2,70	8,78
70	2,66	8,78
80	2,62	8,78
90	2,60	8,79
100	2,58	8,79



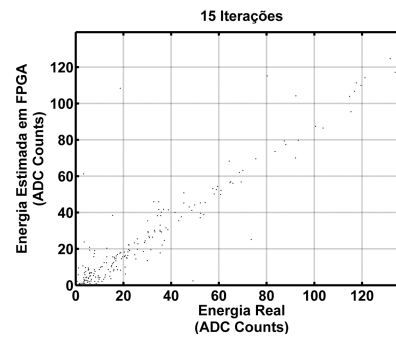
(a) Dispersão com 1 iteração.



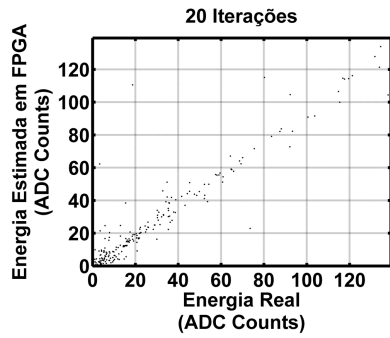
(b) Dispersão com 5 iterações.



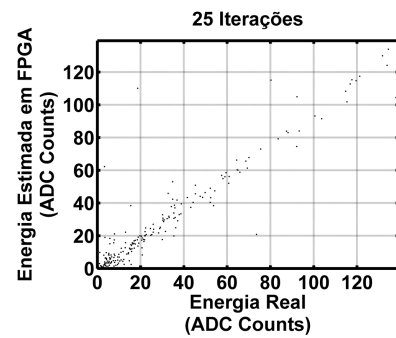
(c) Dispersão com 10 iterações.



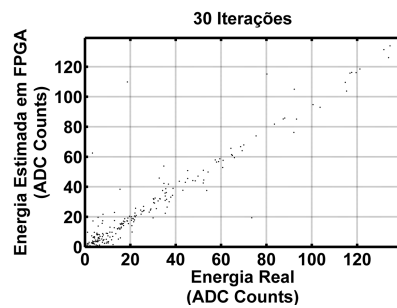
(d) Dispersão com 15 iterações.



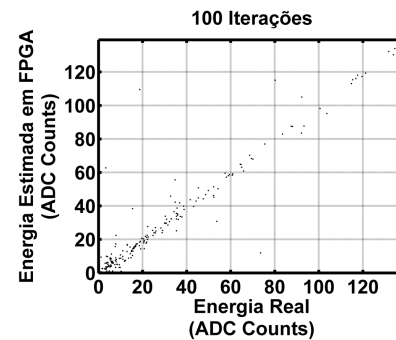
(e) Dispersão com 20 iterações.



(f) Dispersão com 25 iterações.



(g) Dispersão com 30 iterações.



(h) Dispersão com 100 iterações.

Figura 5.12: Gráficos da dispersão entre a energia ideal e a estimada em FPGA no método iterativo.

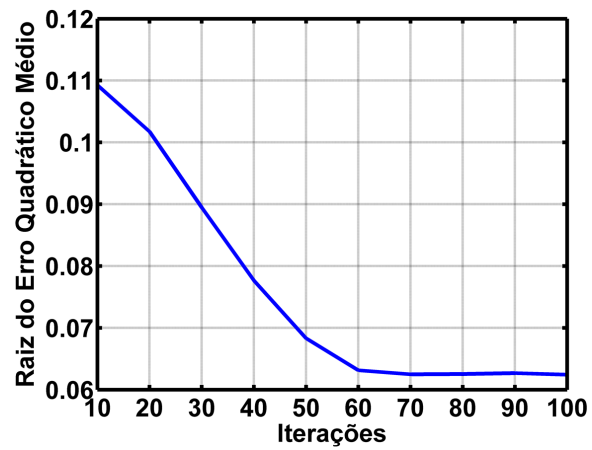
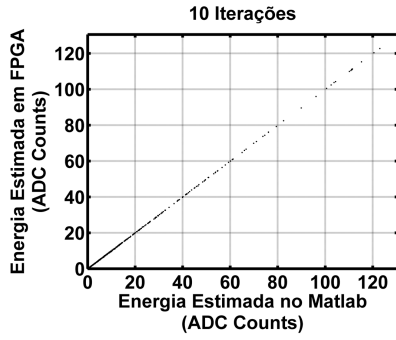


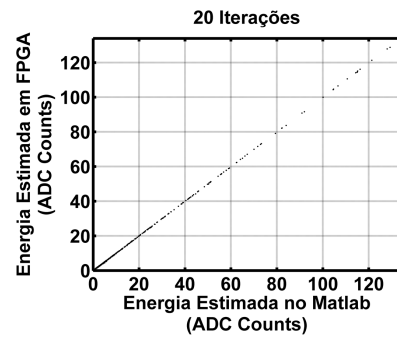
Figura 5.13: Representação gráfica da relação entre a raiz quadrada do erro quadrático médio e o número de iterações considerando o FPGA e o modelo de referência.

Tabela 5.5: Valores do erro entre os resultados da simulação FPGA e o modelo de referência (MATLAB[®]) do método iterativo

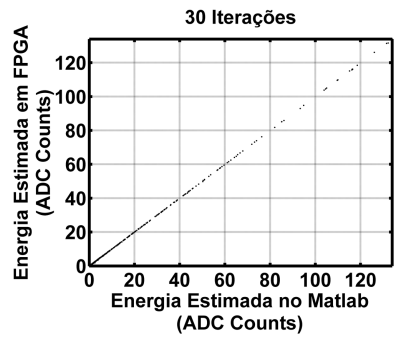
Quantidade de Iterações	Valor médio	Raiz quadrada do erro quadrático médio
10	0,07	0,10
20	0,05	0,10
30	0,05	0,08
40	0,04	0,07
50	0,04	0,06
60	0,03	0,06
70	0,03	0,06
80	0,03	0,06
90	0,03	0,06
100	0,03	0,06



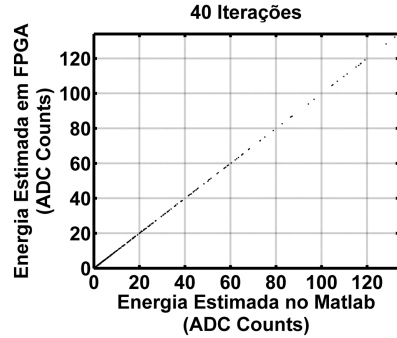
(a) Dispersão com 10 iteração.



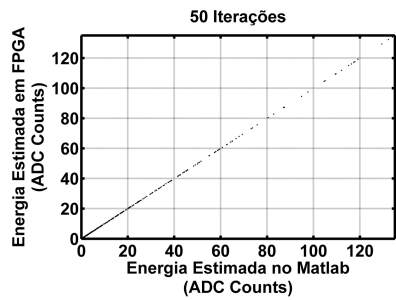
(b) Dispersão com 20 iterações.



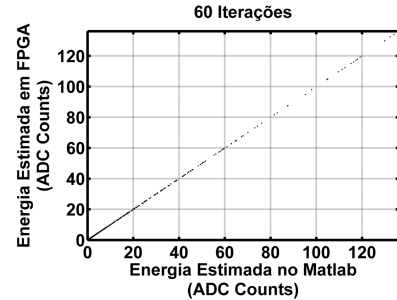
(c) Dispersão com 30 iterações.



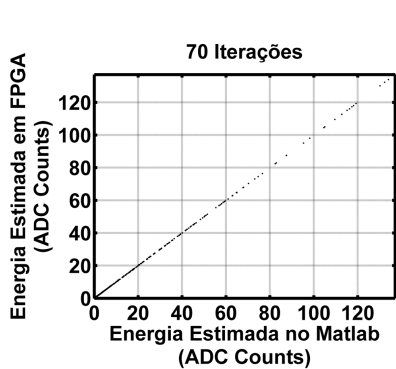
(d) Dispersão com 40 iterações.



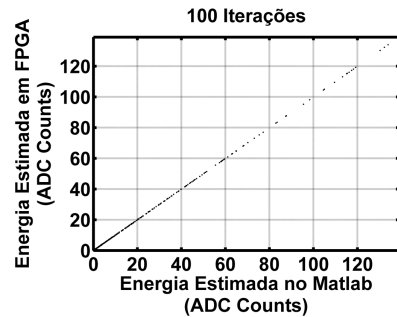
(e) Dispersão com 50 iterações.



(f) Dispersão com 60 iterações.



(g) Dispersão com 70 iterações.



(h) Dispersão com 100 iterações.

Figura 5.14: Gráficos da dispersão entre a energia estimada no FPGA e no MATLAB[®] do método iterativo.

estimação da energia em cada iteração. De acordo com a arquitetura proposta, a maior complexidade encontra-se na resolução da Eq.2 (4.4) e Eq.4 (4.6), pois a primeira envolve multiplicação entre duas matrizes de dimensões 25×19 e 19×1 , enquanto que a segunda, duas matrizes de dimensões 19×25 e 25×1 . Sendo assim, para conseguir melhorar o tempo de processamento, foi necessário utilizar as estruturas de DSP do FPGA.

Em um primeiro cenário de síntese, não foram utilizados os dispositivos DSP48E1 em sua forma otimizada, e cada termo das matrizes resultantes das Eq.2 e Eq.4 foi obtido através da execução da multiplicação e das somas em um mesmo ciclo de relógio. Nessa configuração foi encontrado o pior caso em termos de atraso na execução das operações. A síntese inferiu, para um mesmo caminho, diferentes instâncias de DSP e células lógicas em cascata. O resultado obtido foi um atraso de 12,43 ns para o caminho mais crítico. Com isso, o circuito teria que ser operado a uma frequência máxima de 81,678MHz. Esse período de *clock* tornaria inviável a utilização desse projeto, pois, devido à característica iterativa, seriam necessários para o processamento das 25 amostras com 100 iterações 1.194.987,5 ns.

Para alcançar uma frequência de operação viável, foi necessário utilizar métricas de projeto de *hardware* para melhorar a performance. Nesse caso, o caminho da informação precisou ser reduzido através da remodelagem das etapas referentes ao cálculo de Eq.2 (4.4) e Eq.4 (4.6). Essas modificações consistiram em executar a multiplicação e as somas necessárias para o cálculo de cada termo das matrizes resultantes em estados distintos através de modificação da máquina de estados. Com essa técnica, apesar do aumento da quantidade de pulsos para a execução de uma iteração, foi possível retirar as instâncias de DSP em cascata, responsáveis pelo maior atraso no caminho da informação. Associada a essa abordagem, utilizou-se também a técnica de pipeline das estruturas de DSP48E1 através da utilização de, pelo menos, três estágios, seguindo indicação do manual do usuário do dispositivo [71].

A separação das operações de multiplicação e soma permitiu alcançar reduções expressivas no atraso dos caminhos da informação. O caminho com maior atraso foi reportado com 5,092 ns e passou a ser composto pela lógica das operações de soma em cascata, e não mais o DSP48E1. A partir de então, passou-se a investigar métodos para reduzir esses caminhos. A abordagem adotada foi, também, de criar

outros estados na máquina de controle e fazer com que fosse reduzido a quantidade de níveis de somadores em cascata. Nesse ponto, passou a ser analisado o impacto do aumento da quantidade de estados em relação ao ganho com o período máximo suportado pelo *hardware*. Inicialmente, utilizando uma máquina com 11 estados, foi possível alcançar o caminho mais crítico com atraso de 3,967 ns. Após sucessivas alterações na descrição do hardware e execuções de sínteses, alcançou-se o período máximo de 1,896 ns, utilizando 17 estados.

Nesse último resultado, o atraso reportado é referente ao caminho dos sinais de dois níveis de somas em cascata. Ao ser analisado o impacto da adição de novos estados, observou-se que o ganho em relação ao período era reduzido e que não haveria benefícios expressivos na quantidade de iteração máxima suportada pelo hardware. Sendo assim, considerando o atraso de 1,896 ns em seu caminho mais crítico, o sistema pode ser executado com uma frequência máxima de 527,481 MHz. Utilizando janela de 25 amostras, podem ser executadas até 20 iterações respeitando o tempo de processamento exigido pela eletrônica do *TileCal*.

A complexidade das operações e da arquitetura necessária para alcançar uma performance viável impactou diretamente na utilização dos recursos do FPGA. Na Tabela 5.6 são apresentados os resultados extraídos da ferramenta de síntese em relação a utilização de *Slice Register*, *LUTs* e instâncias de DSP48E1. A expressiva ocupação dos blocos DSP deve-se à grande quantidade de operações de multiplicações que envolve esse método associada à necessidade da alta performance em relação ao tempo de processamento.

Além da ocupação de recursos lógicos, na Tabela 5.7 pode-se observar também as informações referentes ao consumo de potência estática, dinâmica e total.

5.4 Análise dos Resultados

Através dos resultados obtidos com a simulação do *hardware* desenvolvido para o método determinístico, observou-se que o sinal de energia foi reconstruído de forma satisfatória. Com o aumento da ordem do filtro, obteve-se melhores resultados, tanto em relação ao erro médio como em relação à raiz quadrada do erro quadrático médio (Tabela 5.1). Essa melhor performance pode ser observada através dos gráficos de

dispersão (Figura 5.6). A validação dessa simulação foi realizada através da análise dos erros obtidos com a comparação dos resultados de sua simulação e do modelo de referência (Tabela 5.2), comprovando o comportamento aceitável do *hardware*.

Nos resultados da síntese do método determinístico constatou-se que o aumento da ordem do filtro exige, como esperado, maior utilização dos recursos lógicos do FPGA (Figura 5.8a e Figura 5.8b). Essa situação deve-se ao fato de que a quantidade de operações de soma e multiplicação do filtro FIR está diretamente ligada a ordem do filtro, ou seja, quanto maior a ordem mais multiplicações e somas são necessárias para o processamento do sinal. Associado a esse fato, temos que com o aumento da ordem são necessárias mais estruturas de registradores para o armazenamento das informações. O aumento da utilização dos recursos lógicos do FPGA provocou o aumento do consumo da potência (Tabela 5.3). Enquanto que o consumo de potência estática praticamente permaneceu o mesmo, a potência dinâmica sofreu maiores variações. Esse fenômeno está associado diretamente à maior utilização de blocos DSP, pois esses elementos consomem a maior parcela da potência dinâmica.

Em relação aos resultados alcançados com a abordagem iterativa, constatou-se a eficiência desse método na reconstrução das informações de energia do *TileCal*. Através da análise dos valores de erro (Figura 5.11), observou-se que quanto maior o número de iterações, melhor o desempenho na estimação da energia, sendo que, a partir de cerca de 50 iterações o erro é estabilizado. O *hardware* desenvolvido para essa solução também foi validado através da análise do erro entre os resultados simulados e os obtidos com o modelo de referência (Figura 5.13). Verificou-se reduzida taxa de erro, comprovando o comportamento aceitável do projeto.

Os resultados obtidos com a síntese da implementação do método iterativo comprovam a complexidade envolvida na construção desse *hardware*. Conforme relatado, foram necessárias diversas tentativas de melhorias na descrição do projeto a fim de alcançar uma performance viável. Com a análise temporal realizada pela ferramenta, constatou-se que o projeto pode ser executado para processar uma janela de tamanho 25, com 20 iterações, respeitando a taxa de ocorrência das colisões. A utilização dos recursos lógicos concentrou-se nos dispositivos DSP (Tabela 5.6) devido à grande quantidade de operações de multiplicação inerentes ao método, o que impactou no consumo de potência do circuito (Tabela 5.7).

Com a análise da performance alcançada, tanto do método determinístico como do iterativo, verificou-se que ambos atenderam aos requisitos de funcionamento da eletrônica do *TileCal*. Na Tabela 5.8 apresenta-se os principais resultados obtidos em cada implementação. Em relação ao nível de reconstrução da informação de energia, o Gradiente Descendente Positivo apresentou melhor performance. O erro alcançado com 20 iterações é consideravelmente menor do que o obtido com a implementação do FIR. Em contrapartida, o tempo de processamento de uma amostra no filtro FIR é menor do que o tempo para o processamento da janela do iterativo. Em relação a ocupação do FPGA, o método iterativo, devido a sua maior complexidade em comparação ao FIR, exigiu maior utilização dos recursos lógicos, principalmente em relação aos DSPs, o que também provocou maior consumo de potência.

Tabela 5.6: Utilização dos recursos lógicos do FPGA no método iterativo.

Componente	Utilizado	Disponível	Utilização (%)
Slice Register	27738	607200	4
Slice LUTs	16036	303600	5
DSP48E1s	480	2800	17

Tabela 5.7: Relatório de potência na síntese do método iterativo.

	Estática	Dinâmica	Total
Potência (mW)	243,67	307,18	550,85

Tabela 5.8: Comparação dos resultados obtidos com a implementação dos métodos.

Resultado	Método Determinístico	Método Iterativo
Raiz Quadrada do Erro Quadrático Médio	19,21 (Ordem 20)	9,06 (20 Iterações)
Período de <i>Clock</i> Mínimo	1,879 ns	1,896 ns
Utilização de Slice LUTs	0,24 % (Ordem 20)	5 %
Utilização de DSP48E1s	0,75 % (Ordem 20)	17 %
Consumo de Potência	423,66 mW (Ordem 20)	550,85 mW

Capítulo 6

Conclusões

O acelerador de partículas LHC, construído no CERN, é o maior acelerador atualmente em operação, e está submetido a um cronograma de atualização com o objetivo de aumentar sua luminosidade e, com isso, avançar nas pesquisas relacionadas às partículas elementares. Na primeira fase do programa de atualização, que ocorreu entre 2013 e 2014, foi alcançada uma luminosidade de aproximadamente $10^{34} \text{cm}^2 \text{s}^{-1}$. Na segunda, prevista para o período entre 2018 e 2019, pretende-se chegar a aproximadamente $2 - 3 \times 10^{34} \text{cm}^2 \text{s}^{-1}$. Em sua última fase de operação, prevista para ocorrer entre 2022 e 2023, projeta-se alcançar $5 - 7 \times 10^{34} \text{cm}^2 \text{s}^{-1}$.

Devido ao aumento da luminosidade, torna-se necessário atualizar os sistemas eletrônicos dos detectores que compõem esse acelerador, como, por exemplo, o ATLAS. Nesse cenário de operação, os sinais processados pelos sensores serão afetados pelo efeito de sobreposição de eventos adjacentes, conhecido como *pileup*. Esse efeito provoca a distorção do padrão dos eventos e, conseqüentemente, irá afetar o desempenho do *trigger*. Dessa forma, além da alteração da arquitetura da eletrônica, torna-se necessário, também, alterar o método de estimação de energia utilizado atualmente, pois o filtro casado tem seu funcionamento baseado no conhecimento do padrão dos sinais, que serão distorcidos pelo *pileup*.

Um dos requisitos do processo de atualização do ATLAS consiste no aumento da granularidade das informações enviadas para o L1 do *trigger*, ou seja, esse nível receberá todas as informações dos calorímetros referentes aos eventos, pois os somadores serão removidos. Com isso, pretende-se aumentar o desempenho na identificação dos eventos de interesse. Os pesquisadores da colaboração ATLAS propuseram uma

atualização na eletrônica do detector, que inclui mudanças na arquitetura de funcionamento do nível 1 do sistema de seleção de eventos.

Esse trabalho apresentou uma solução em hardware (FPGA) para a recuperação das informações de energia afetadas pelo *pileup*. Para o desenvolvimento desse projeto, considerou-se os requisitos de funcionamento definidos pelos pesquisadores do ATLAS, como, por exemplo, o dispositivo FPGA utilizado e o tempo de processamento dos sinais.

Foram realizadas pesquisas a fim de definir métodos eficientes na reconstrução do sinal do *TileCal*. Nesse trabalho foram consideradas duas abordagens, uma determinística, baseada em técnicas de filtro FIR e outra iterativa, através do algoritmo Gradiente Descendente Positivo. Os resultados alcançados com a implementação dessas técnicas em FPGA atestam a viabilidade em embarcar esses métodos no sistema eletrônico do *TileCal*.

Em relação ao método determinístico, observou-se que com o aumento da ordem do filtro foi possível obter melhor resultado em relação à estimação da energia, alcançando os menores valores do erro com ordens entre 20 e 30. A baixa complexidade no desenvolvimento do hardware para essa solução, associada à arquitetura do sistema proposta, culminou na baixa ocupação dos recursos lógicos do dispositivo e baixo consumo de potência. A utilização das estruturas otimizadas de DSP disponíveis no dispositivo possibilitou alcançar resultado satisfatório em relação à frequência suportada pelo hardware.

Ao analisar os resultados obtidos com o método iterativo, constatou-se que essa abordagem, em relação a estimação de energia, apresenta melhor desempenho. Foram obtidos valores de erro significativamente inferiores em relação à implementação com filtro FIR. No que diz respeito a ocupação dos recursos lógicos, essa solução teve um desempenho pior em relação ao FIR. Esse resultado deve-se à complexidade nos cálculos desse algoritmo, principalmente por envolver operações de multiplicações entre matrizes, e também por exigir uma estrutura de controle do fluxo de sinal mais sofisticada.

Para alcançar um resultado satisfatório em relação ao período de *clock* suportado pelo circuito, foi necessário um trabalho exaustivo na execução de sucessivas sínteses e análise dos relatórios. A cada processo de síntese, a forma de descrever o

hardware foi sendo modificada de forma a seguir parâmetros de codificação eficiente para o sintetizador da Xilinx. Em decorrência da complexidade na codificação das operações de multiplicação entre matrizes, foi necessária uma investigação minuciosa do dispositivo DSP48E1 a fim de conseguir utilizá-lo de forma otimizada. Ao fim desse processo, foi possível alcançar um resultado que viabiliza a execução desse método com 20 iterações e uma ocupação de recursos lógicos razoável.

Conclui-se que, do ponto de vista da recuperação do sinal de energia, a abordagem iterativa, implementada utilizando o método do Gradiente Descendente Positivo, apresenta-se como mais adequada do que o método determinístico.

Referências Bibliográficas

- [1] RUSSEL, B., *The History of Western Philosophy*. Simon and SchusterTouchstone, 1967.
- [2] FERNOW, R. C., *Introduction to Experimental Particle Physics*. Cambridge University Press, Mar 1986.
- [3] EVANS, L., BRYANT, P., “LHC machine”, *Journal of Instrumentation*, v. 3, n. 08, pp. S08001, 2008.
- [4] AAD, G., ABAT, E., ABDALLAH, J., et al., “The ATLAS experiment at the CERN large hadron collider”, *Journal of Instrumentation*, v. 3, n. 8, 2008.
- [5] CERQUEIRA, A. S., “ATLAS Tile Calorimeter Readout Electronics Upgrade Program for the High Luminosity LHC”. In: *Proceedings, International School on High Energy Physics : Workshop on High Energy Physics in the near Future. (LISHEP 2013): Rio de Janeiro, Brazil, March 17-24, 2013*, 2013.
- [6] CLEMENT, C., KLIMEK, P., “Identification of pile-up using the quality factor of pulse shapes in the ATLAS Tile Calorimeter”. In: *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE*, pp. 1188–1193, Oct 2011.
- [7] HASELMAN, M. D., HAUCK, S., LEWELLEN, T. K., et al., “FPGA-based pulse pileup correction”. In: *IEEE Nuclear Science Symposium Medical Imaging Conference*, pp. 3105–3112, Oct 2010.
- [8] LOPATIN, M., MOSKOVITCH, N., TRIGANO, T., et al., “Pileup attenuation for spectroscopic signals using a sparse reconstruction”. In: *Electrical*

- Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of*, pp. 1–5, Nov 2012.
- [9] GARVEY, J., REES, D., *Bunch Crossing Identification for the ATLAS Level - 1 Calorimeter Trigger*, Tech. Rep. ATL-DAQ-96-051. ATL-D-PN-51, CERN, Geneva, May 1996.
- [10] HADLEY, D. R., “Digital Filtering Performance in the ATLAS Level-1 Calorimeter Trigger”. In: *Proceedings, 17th Conference on Computing Applications in Nuclear and Plasma Sciences on Real Time (RT2010): Lisbon, Portugal, May 24-28, 2010*, CERN: Geneva, 2010.
- [11] WIGMANS, R., *Calorimetry: Energy Measurement in Particle Physics. International series of monographs on physics*, Clarendon Press, 2000.
- [12] DUARTE, J. P. B. S., *Estudo de Técnicas de Deconvolução para Reconstrução de Energia online no Calorímetro Hadrônico do ATLAS*, Master’s Thesis, Programa de Pós Graduação em Engenharia Elétrica / UFJF, 2015.
- [13] LYONS, R. G., *Understanding digital signal processing*. 3rd ed. Pearson Education, 2010.
- [14] BANDŽUCH, P., MORHÁČ, M., KRIŠTIAK, J., “Study of the Van Cittert and Gold iterative methods of deconvolution and their application in the deconvolution of experimental spectra of positron annihilation”, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, v. 384, n. 2, pp. 506 – 515, 1997.
- [15] CARUSO, F., OGURI, V., SANTORO, A., *Partículas Elementares: 100 anos de descobertas*. Livraria da Física, 2012.
- [16] BRAIBANT, S., GIACOMELLI, G., SPURIO, M., *Particles and Fundamental Interactions: An Introduction to Particle Physics. Undergraduate Lecture Notes in Physics*, Springer Netherlands, 2011.
- [17] MOREIRA, M. A., “O Modelo Padrão da Física de Partículas”, *Revista Brasileira de Ensino de Física*, v. 31, pp. 1306.1 – 1306.11, 04 2009.

- [18] WONG, S. S., *Introductory Nuclear Physics*. 2nd ed. Wiley-VCH, 1998.
- [19] MOREIRA, M. A., *Física de Partículas: uma abordagem conceitual e epistemológica*. Livraria da Física, 2012.
- [20] GRIFFITHS, D. J., *Introduction to elementary particles; 2nd rev. version. Physics textbook*, Wiley: New York, NY, 2008.
- [21] PERKINS, D. H., *Introduction to high-energy physics; 4th ed.*. Cambridge Univ. Press: Cambridge, 2000.
- [22] FILHO, E. F. S., *Análise não-linear de componentes independentes para uma filtragem online baseada em calorimetria de alta energia e com fina segmentação*, Ph.D. Thesis, Electrical Engineering Program / COPPE / UFRJ, Rio de Janeiro, BR, 2010.
- [23] MARTIN, B. R., *Nuclear and Particle Physics: An Introduction*. Wiley, 2009.
- [24] CERN, “CERN Accelerating science”, Disponível em: <https://home.cern/>. Acesso em: 28 de Setembro de 2016, 2016.
- [25] DESY, “Deutsches Elektronen-Synchrotron”, Disponível em: <http://www.desy.de/>. Acesso em: 28 de Setembro de 2016, 2016.
- [26] KEK, “HIGH ENERGY ACCELERATOR RESEARCH ORGANIZATION”, Disponível em: <http://www.kek.jp/>. Acesso em: 28 de Setembro de 2016, 2016.
- [27] FERMILAB, “Fermi National Accelerator Laboratory”, Disponível em: <http://www.fnal.gov/>. Acesso em: 28 de Setembro de 2016, 2016.
- [28] SLAC, “SLAC NATIONAL ACCELERATOR LABORATORY”, Disponível em: <https://www6.slac.stanford.edu/>. Acesso em: 28 de Setembro de 2016, 2016.
- [29] BROOKHAVEN, “Brookhaven National Laboratory - a passion for discovery”, Disponível em: <https://www.bnl.gov/>. Acesso em: 28 de Setembro de 2016, 2016.

- [30] CERN, “Experiments”, Disponível em: [http :
//home.cern/about/experiments](http://home.cern/about/experiments). Acesso em: 27 de Setembro de 2016, 2016.
- [31] LHC-COLLABORATION, “Large Hadron Collider”, Disponível em: [http :
//lhc.web.cern.ch/lhc/](http://lhc.web.cern.ch/lhc/). Acesso em: 27 de Setembro de 2016, 2016.
- [32] BIOT, A. V., *TileCal Read-Out Drivers production and firmware developments*, Master’s Thesis, UNIVERSITAT DE VALÈNCIA ESTUDI GENERAL DE VALÈNCIA, Valencia, 2007.
- [33] PETTERSSON, T. S., LEFÈVRE, P., *The Large Hadron Collider: conceptual design*, Tech. Rep. CERN-AC-95-05-LHC, Oct 1995.
- [34] AAMODT, K., QUINTANA, A. A., ACHENBACH, R., et al., “The ALICE experiment at the CERN LHC”, *Journal of Instrumentation*, v. 3, n. 08, pp. S08002, 2008.
- [35] CHATRCHYAN, S., HMAYAKYAN, G., KHACHATRYAN, V., et al., “The CMS experiment at the CERN LHC”, *Journal of Instrumentation*, v. 3, n. 08, pp. S08004, 2008.
- [36] JR, A. A. A., FILHO, L. M. A., BARBOSA, A. F., et al., “The LHCb Detector at the LHC”, *Journal of Instrumentation*, v. 3, n. 08, pp. S08005, 2008.
- [37] ADRIANI, O., BONECHI, L., BONGI, M., et al., “The LHCf detector at the CERN Large Hadron Collider”, *Journal of Instrumentation*, v. 3, n. 08, pp. S08006, 2008.
- [38] ANELLI, G., ANTCHÉV, G., ASPELL, P., et al., “The TOTEM Experiment at the CERN Large Hadron Collider”, *Journal of Instrumentation*, v. 3, n. 08, pp. S08007, 2008.
- [39] AIRAPETIAN, A., GRABSKY, V., HAKOPIAN, H., et al., “ATLAS: Detector and physics performance technical design report. Volume 1”, 1999.
- [40] MARTI-GARCIA, S., “Tracking and vertexing performance of the ATLAS Inner Detector at the LHC”. CERN: Geneva, 2011.

- [41] VOGEL, A., “ATLAS Transition Radiation Tracker (TRT): Straw tube gaseous detectors at high rates”, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, v. 732, pp. 277 – 280, 2013.
- [42] PUZO, P., “ATLAS calorimetry”, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, v. 494, n. 1-3, pp. 340 – 345, 2002, Proceedings of the 8th International Conference on Instrumentation for Colliding Beam Physics.
- [43] PONTECORVO, L., “The ATLAS muon spectrometer”, *The European Physical Journal C - Particles and Fields*, v. 34, n. 1, pp. s117–s128, 2004.
- [44] FABJAN, C. W., GIANOTTI, F., “Calorimetry for Particle Physics”, *Rev. Mod. Phys.*, v. 75, n. CERN-EP-2003-075, pp. 1243–1286. 96 p, Oct 2003.
- [45] AAD, G., ABBOTT, B., ABDALLAH, J., et al., “Readiness of the ATLAS Tile Calorimeter for LHC collisions”, *The European Physical Journal C*, v. 70, n. 4, pp. 1193–1236, 2010.
- [46] ADRAGNA, P., ALEXA, C., ANDERSON, K., et al., “The ATLAS hadronic tile calorimeter: from construction toward physics”, *IEEE Transactions on Nuclear Science*, v. 53, n. 3, pp. 1275–1281, June 2006.
- [47] CAVALCANTI, M. M., FILHO, E. F. S., FARIAS, P. C. M. A., et al., “Sistema Digital para redução do empilhamento em medições de energia num detector de partículas”, *Anais do XX Congresso Brasileiro de Automática*, pp. 456–463, 2014.
- [48] CARRIÓ, F., CASTILLO, V., FERRER, A., et al., “The sROD module for the ATLAS Tile Calorimeter Phase-II Upgrade Demonstrator”, *Journal of Instrumentation*, v. 9, n. 02, pp. C02019, 2014.
- [49] CERQUEIRA, A. S., “Upgrade for the ATLAS Tile Calorimeter Readout Electronics at the High Luminosity LHC”, *Journal of Physics: Conference Series*, v. 404, n. 1, pp. 012060, 2012.

- [50] CERQUEIRA, A. S., SEIXAS, J. M., CALOBA, L. P., “Analog system for building the first-level triggering signal provided by the hadronic calorimeter of ATLAS detector”, *Nucl. Instrum. Meth.*, v. A570, pp. 117–125, 2007.
- [51] SPOOR, M., *Upgrade of the ATLAS Tile hadronic calorimeter for high-luminosity LHC run*, Tech. Rep. ATL-TILECAL-PROC-2016-001, CERN, Geneva, Mar 2016.
- [52] GOMES, A., “The new front-end electronics for the ATLAS Tile Calorimeter Phase 2 Upgrade”, *Journal of Instrumentation*, v. 11, n. 02, pp. C02015, 2016.
- [53] HARKUSHA, S., COLLABORATION, A. T., *Upgrade of the ATLAS hadronic Tile calorimeter for the High luminosity LHC*, Tech. Rep. ATL-TILECAL-PROC-2016-009, CERN, Geneva, Aug 2016.
- [54] USAI, G., *Upgrade of the ATLAS Tile Calorimeter for the High luminosity LHC*, Tech. Rep. ATL-TILECAL-PROC-2015-009, CERN, Geneva, Jun 2015.
- [55] TANG, F., ANDERSON, K., DRAKE, G., et al., “Design of the Front-End Readout Electronics for ATLAS Tile Calorimeter at the sLHC”, *IEEE Transactions on Nuclear Science*, v. 60, n. 2, pp. 1255–1259, April 2013.
- [56] VALERO, A., COLLABORATION, A. T., *A new read-out architecture for the ATLAS Tile Calorimeter Phase-II Upgrade*, Tech. Rep. ATL-TILECAL-PROC-2015-025, CERN, Geneva, Dec 2015.
- [57] CARRIO ARGOS, F., “The PreProcessors for the ATLAS Tile Calorimeter Phase II Upgrade”, Nov 2015.
- [58] VAN TREES, H. L., *Detection, estimation, and modulation theory*. John Wiley & Sons, 2001.
- [59] WESSELS, M., “Calibration and Performance of the ATLAS Level-1 Calorimeter Trigger with LHC Collision Data”, *Physics Procedia*, v. 37, pp. 1841 – 1848, 2012.

- [60] MEYER-BAESE, U., *Digital signal processing with field programmable gate arrays*. 3rd ed., v. 65. Springer Publishing Company, Incorporated, 2007.
- [61] DINIZ, P., NETTO, S., SILVA, E. D., *Digital Signal Processing: System Analysis and Design*. 2nd ed. Cambridge University Press, 2010.
- [62] MITRA, S. K., *Digital Signal Processing: A Computer-Based Approach*. 2nd ed. McGraw-Hill School Education Group, 2001.
- [63] OPPENHEIM, A. V., WILLSKY, A. S., *Sinais e Sistemas*. 2nd ed. Pearson Prentice Hall, 2010.
- [64] HAYKIN, S., VAN VEEN, B., *Sinais E Sistemas*. Bookman, 2001.
- [65] PROAKIS, J. G., MANOLAKIS, D. K., *Digital Signal Processing*. 4th ed. Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2006.
- [66] ELAD, M., *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. 1st ed. Springer Publishing Company, Incorporated, 2010.
- [67] GOLD, R., *An Iterative Unfolding Method for Response Matrices*. Argonne National Laboratory, Dec 1964.
- [68] XILINX, *7 Series FPGAs Overview*, May 2015, v1.17, Disponível em: <https://www.xilinx.com/support.html>.
- [69] NOVAKOVA, J., *Monte Carlo Simulations of the Tile Calorimeter and Measurement of the Z to tau tau Cross Section with the ATLAS Detector*, Ph.D. Thesis, Charles U., 2012-06-27.
- [70] CONSTANTINIDES, G. A., CHEUNG, P. Y. K., LUK, W., “Synthesis of Saturation Arithmetic Architectures”, *ACM Trans. Des. Autom. Electron. Syst.*, v. 8, n. 3, pp. 334–354, jul 2003.
- [71] XILINX, *7 Series DSP48E1 Slice*, Nov 2014, v1.8, Disponível em: <https://www.xilinx.com/support.html>.

- [72] MANO, M. M., CILETTI, M. D., *Digital Design: With an Introduction to the Verilog HDL*. Pearson, 2013.
- [73] ZEIDMAN, B., ZEIDMAN, R. M., *Verilog Designer's Library*. 1st ed. Prentice Hall PTR: Upper Saddle River, NJ, USA, 1999.
- [74] XIU, L., *VLSI Circuit Design Methodology Demystified: A Conceptual Taxonomy*. Wiley-IEEE Press, 2007.
- [75] RAMACHANDRAN, S., *Digital VLSI Systems Design: A Design Manual for Implementation of Projects on FPGAs and ASICs Using Verilog*. Springer Publishing Company, Incorporated, 2014.
- [76] LAVAGNO, L., SCHEFFER, L., MARTIN, G., *EDA for IC implementation, circuit design, and process technology*. CRC press, 2006.
- [77] CHU, P. P., *FPGA prototyping by Verilog examples: Xilinx Spartan-3 version*. John Wiley & Sons, 2011.
- [78] SASS, R., SCHMIDT, A. G., *Embedded systems design with platform FPGAs: principles and practices*. Morgan Kaufmann, 2010.
- [79] MEHTA, N., "Xilinx redefines power, performance, and design productivity with three new 28 nm fpga families: Virtex-7, kintex-7, and artix-7 devices", *Xilinx White Paper*, 2012.
- [80] HUSSEIN, J., KLEIN, M., HART, M., "Lowering power at 28 nm with Xilinx 7 series devices", *Xilinx, White paper, WP389 (v1.3)*, 2015.
- [81] XILINX, *7 Series FPGAs Configurable Logic Block*, Nov 2014, v1.7, Disponível em: <https://www.xilinx.com/support.html>.
- [82] XILINX, *Vivado Design Suite User Guide - Implementation*, Dec 2013, v2013.4, Disponível em: <https://www.xilinx.com/support.html>.
- [83] XILINX, *Vivado Design Suite User Guide - Synthesis*, May 2014, v2014.1, Disponível em: <https://www.xilinx.com/support.html>.

Apêndice A

Projeto de Hardware Digital

A.1 Introdução

Implementações de um sistema digital em software são, normalmente, menos complexas de serem desenvolvidas se comparadas a implementações em *hardware*. Implementações em *software* podem utilizar microprocessadores, multiprocessadores e processadores digitais de sinais de propósito geral. Porém, as instruções disponíveis nesses dispositivos não são apropriadas para aplicações que exigem rápido processamento computacional, pois, entre outros quesitos, são executadas sequencialmente, reduzindo a eficiência do processamento [75]. Em contrapartida, implementações em hardware podem alcançar maior desempenho, já que podem ser projetadas para utilizar técnicas de paralelismo [75].

Um sistema em hardware dedicado pode ser desenvolvido em ASIC ou utilizando dispositivos FPGA. Os ASICs são utilizados para implementação de tarefas específicas, caracterizando-se como um circuito integrado (CI) desenvolvido exclusivamente para exercer uma determinada função, em um domínio específico, não podendo ter sua estrutura alterada. Os FPGAs são dispositivos com arquiteturas reconfiguráveis que tem sua funcionalidade definida pelos projetistas, podendo ser alterada a qualquer momento.

No desenvolvimento de um projeto de hardware, a decisão da utilização de ASIC ou FPGA deve levar em consideração questões como: custo, tempo de desenvolvimento, desempenho, quantidade de unidades necessárias, entre outras. O custo no desenvolvimento de um ASIC é superior ao custo da implementação em FPGA.

Em contrapartida, é possível alcançar melhor performance em uma implementação ASIC. O tempo de desenvolvimento do projeto em ASIC também é superior ao do FPGA, pois além do tempo de implementação, existe o tempo para a fabricação do CI pela *foundry*.

O projeto de um hardware, seja ASIC ou FPGA, deve seguir um determinado fluxo de desenvolvimento com a finalidade de obter um sistema que atenda às necessidades da aplicação. Na Figura A.1 ilustra-se os elementos básicos do fluxo de um projeto de hardware digital.

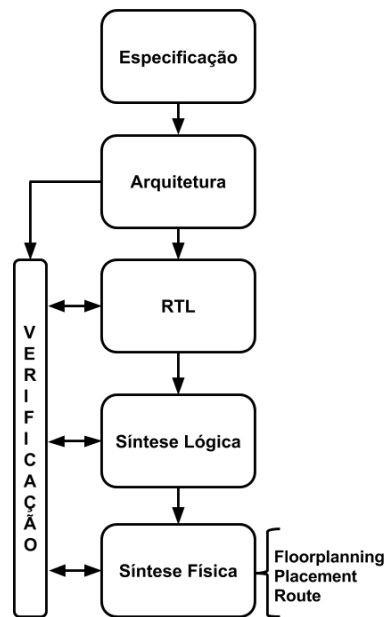


Figura A.1: Diagrama do fluxo de projeto digital.

As fases de Especificação e Arquitetura fazem parte do processo de planejamento do sistema. Nelas são levantados os requisitos, definidas as restrições de tempo, área e potência, desenvolvido o modelo de referência utilizando linguagens de alto nível, como Matlab e C. Com a arquitetura do sistema definida, inicia-se a fase de *Register-Transfer Level* (RTL) [72], que consiste na descrição comportamental do hardware, utilizando alguma Linguagem de Descrição de Hardware (HDL, do inglês *Hardware Description Language*), como Verilog [73] e Linguagem de Descrição de Hardware VHSIC (VHDL, do inglês *VHSIC Hardware Description Language*) [74]. A utilização de estruturas sintetizáveis representa um aumento no nível de abstração no processo de descrição do hardware, pois não é necessário preocupar-se com a biblioteca da tecnologia, permitindo, assim, um aumento da eficiência nessa fase.

A síntese lógica é responsável pela conversão do modelo RTL em elementos combinacionais, sequenciais e de interconexão. Nessa fase é gerado uma *netlist*, composta por portas lógicas, a qual representa a lógica de funcionamento do circuito [76, 83]. Devido à utilização de ferramentas computacionais para a realização dessa tarefa, pôde-se aumentar a produtividade dos projetistas. As ferramentas de síntese, além de traduzir o hardware descrito em uma linguagem de alto nível, realiza otimizações de área, potência e de temporização [76] levando em consideração a tecnologia.

Na fase de síntese física ocorre, basicamente, a tradução do projeto lógico em físico [74]. Nessa etapa as ferramentas de desenvolvimento em FPGA realizam tarefas como *Floorplanning*, *Placement*, Roteamento (*Route*), algumas otimizações, entre outras [82]. No *Floorplanning* é realizada a definição da área utilizada pelo projeto, a posição dos blocos, dos pinos de entradas e saídas, posicionamento das células, planejamento da distribuição de potência, entre outras. O *Placement* é responsável pela definição do local apropriado para o posicionamento das células de acordo com a posição previamente definida. Com os elementos devidamente posicionados, executa-se a etapa de *Route*, responsável pela interconexão, roteamento, entre as células lógicas utilizando metais [74]. Após o roteamento, são realizadas outras etapas referentes ao posicionamento das linhas de alimentação, verificação de regras de fabricação e equivalência lógica [74].

Como pode ser observado na Figura A.1, a etapa de Verificação pode ocorrer durante quase todo o fluxo de desenvolvimento do projeto. Essa etapa é responsável por garantir o correto funcionamento do sistema em cada etapa. Para isso, costuma-se executar a verificação funcional tanto no nível RTL como no nível de portas lógicas geradas durante as sínteses lógica e física.

A.2 Field-Programmable Gate Arrays - FPGA

FPGAs são dispositivos semicondutores reconfiguráveis, ou seja, formados por elementos lógicos e de interconexão programáveis [74]. Também conhecidos pelo termo "*sea-of-gates*", em português, "mar de portas lógicas", são dispositivos que podem ser rapidamente configurados para uma determinada aplicação [75]. A maioria desses dispositivos utilizam RAM estática para armazenar as informações de confi-

gurações, dessa forma, permitindo que sejam configurados e reconfigurados mesmo após de terem sido instalados em um determinado produto, sendo essa a característica de ser programável em campo [78]. Esse aspecto torna esses dispositivos atrativos do ponto de vista do tempo de desenvolvimento do projeto e versatilidade, pois, o mesmo pode ser reprogramado em qualquer instante, alterando o funcionamento do circuito projetado, com um mesmo dispositivo podendo ser usado para diferentes áreas de aplicações.

Assim como os ASICs, circuitos desenvolvidos para FPGA também podem explorar técnicas de desenvolvimento de *hardware* para melhorar o desempenho da aplicação, como paralelismo, *pipeline*, diferentes domínios de *clock*, etc. Para aplicações de baixo volume, os FPGAs apresentam-se como soluções mais baratas do que os ASICs, além de possuir um ciclo de desenvolvimento mais curto. A decisão de utilização desses dispositivos deve levar em consideração as especificações do projeto em relação a quantidade de elementos lógicos, período de *clock*, tensão de alimentação, portas de entrada e saída, entre outros quesitos. Ao contrário de um projeto em ASIC, em que o projetista tem maior liberdade, os dispositivos FPGA possuem uma estrutura pré-definida e, conseqüentemente, recursos pré-definidos. Sendo assim, existe uma quantidade máxima de elementos lógicos e portas de entrada e saída que poderão ser utilizados para modelar o comportamento do hardware, assim como, o período de *clock* que o mesmo poderá funcionar, a tensão de alimentação também deverá seguir opções pré-definidas, entre outros pontos [75].

A estrutura geral dos FPGAs é composta por uma matriz bidimensional de células lógicas genéricas e comutadores programáveis [77]. A Figura A.2 mostra um diagrama básico dessa estrutura. As células lógicas, apresentadas como “*logic cell*”, são os elementos configuráveis para exercer alguma função específica e os comutadores, apresentados como “*programmable switch*”, são responsáveis por fazer a interconexão entre as células lógicas.

As células lógicas são, basicamente, formadas por um circuito combinacional e por um elemento de registro, um *flip-flop*. O circuito combinacional reconfigurável é, normalmente, implementado por uma *look-up table* (*LUT*) [77]. A Figura A.3 apresenta o diagrama de uma célula com três entradas baseadas em uma LUT. Nesse diagrama a LUT representa um circuito combinacional que gera uma saída em função

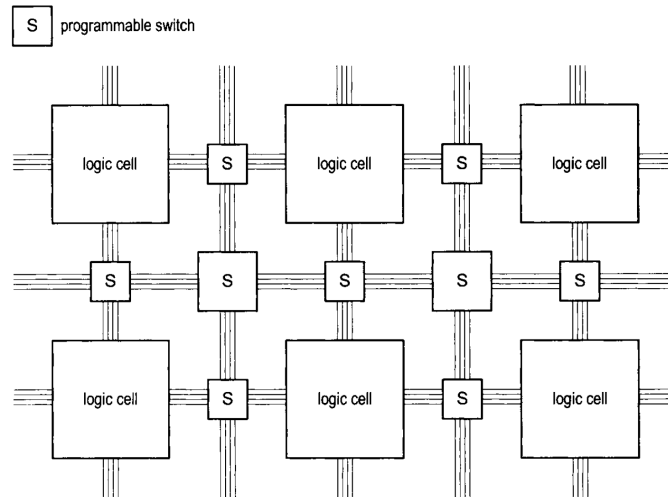


Figura A.2: Diagrama conceitual da estrutura de um FPGA [77].

das três entradas, sendo que esta pode ser utilizada diretamente ou registrada.

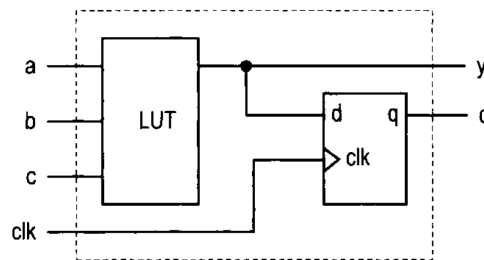


Figura A.3: Diagrama conceitual de uma célula lógica [77].

Além das estruturas mencionadas, os FPGAs disponibilizam ainda elementos chamados *Macro cell*, que são utilizados com o objetivo de melhorar a performance dos circuitos implementados, e apresentam-se como um complemento às células lógicas genéricas. Alguns exemplos dessas estruturas são: blocos de memória, multiplicadores, dispositivos de processamento digital de sinais, circuito de gerenciamento de *clock*, circuitos de interface de entrada e saída, entre outros.

A.2.1 Família Virtex-7 da Xilinx

Existe no mercado uma extensa variedade de dispositivos FPGA com diferentes características, atendendo, assim, a diversos tipos de aplicações. A escolha do FPGA é uma das mais importantes etapas durante a especificação do projeto, pois, de acordo com os requisitos de funcionamento da aplicação, precisa ser escolhido o dispositivo que possua todos os recursos necessários para o correto funcionamento

do projeto. O impacto de uma escolha errada de FPGA pode ser irreversível, pois é impossível alterar a estrutura lógica desse dispositivo, inviabilizando qualquer alteração em sua arquitetura para se adequar aos requisitos do circuito projetado.

Nesse projeto decidiu-se utilizar um dispositivo FPGA da família Virtex-7 [68] da Xilinx. Essa escolha foi realizada no âmbito da colaboração do detector ATLAS e baseou-se no fato de a família Virtex-7 ser uma das mais robustas disponíveis atualmente [68], tendo, assim, recursos lógicos e físicos suficientes para o desenvolvimento do projeto.

A crescente demanda do mercado por um dispositivo com maior largura de banda e desempenho de processamento associado ao baixo consumo de potência, impulsionou a Xilinx a criar a família 7 de dispositivos FPGAs. Para atender a esses requisitos, a Xilinx utilizou a tecnologia de fabricação 28 nm da *foundry* TSMC, obtendo um produto com preço reduzido, baixo consumo de potência, alta performance e capacidade [79, 80]. Na Figura A.4 apresenta-se uma comparação entre a tecnologia 28 nm e a de 40 nm, podendo ser observado um ganho expressivo em relação ao consumo de potência.

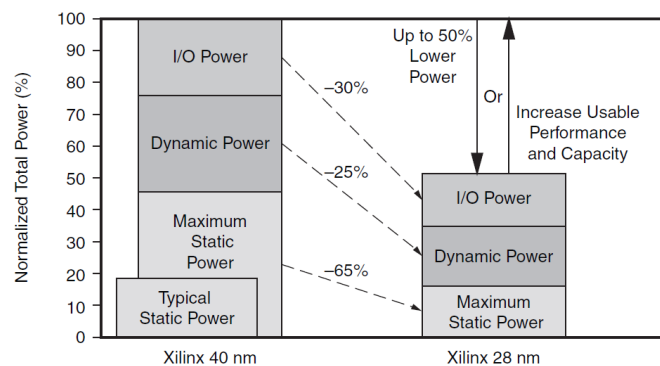


Figura A.4: Otimização da família 7 da Xilinx [79].

Algumas características da família 7 de FPGAs da Xilinx são [79] [68]:

- Blocos de lógica configurável usam as estruturas *look-up table* com 6 entradas, lógica de controle, sinais de habilitação e saídas.
- Look-up table de 6 entradas com tecnologia configurável como memória distribuída.
- Blocos de RAM com 18k e 36k, com suporte a porta única e porta dupla.

Tabela A.1: Características dos dispositivos da família Virtex-7

Dispositivo	Células Lógicas	DSP	Blocos de RAM (36Kb)	PCIe	Bancos de I/O	Tensão de I/O
XC7V585T	582.720	1.260	795	3	17	1.2V, 1.35V, 1.5V, 1.8V, 2.5V, 3.3V
XC7V2000T	1.954.560	2.160	1.292	4	24	
XC7VX330T	326.400	1.120	750	2	14	
XC7VX415T	412.160	2.160	880	2	12	
XC7VX485T	485.760	2.800	1.030	4	14	
XC7VX550T	554.240	2.800	1.180	2	16	

- Tecnologia de entrada e saída de alta performance com suporte a interface DDR3.
- Interface analógica configurável pelo usuário.
- Dispositivos DSP com multiplicadores 25x18, acumuladores de 48 bits e pré soma para filtragem de alta performance.
- Projetado para alta performance e baixo consumo de potência com a tecnologia de 28 nm

Na Tabela A.1 são apresentas as principais características de alguns FPGAs da família Virtex-7 [68]. Nessa Tabela pode-se observar algumas características do dispositivo selecionado para operar no detector ATLAS, o XC7VX485T. Esse FPGA possui grande quantidade de dispositivos DSP e células lógicas disponíveis, sendo estes fundamentais para a implementação dos sistemas de processamento dos sinais de energia do *TileCal*. Destaca-se, também, a quantidade de blocos dedicados de memória RAM, necessários para o armazenamento das informações e os bancos de I/O, utilizados para interfaceamento com os demais módulos eletrônicos.

A.2.2 Estruturas de DSP

As estruturas de DSP disponíveis nos FPGAs da Xilinx são responsáveis por garantir um melhor desempenho no processamento de sinais digitais. Essas estruturas implementam arquiteturas otimizadas de multiplicadores, acumuladores, soma, entre outras. Em aplicações que utilizem essas operações, essas estruturas são fundamentais para melhorar a performance tanto em relação ao tempo de processamento,

como em termos de ocupação dos recursos lógicos, ao contrário da utilização dos elementos lógicos genéricos para gerarem os circuitos dessas operações.

Na família 7 da Xilinx, essas estruturas recebem a denominação de DSP48E1 e podem implementar algoritmo customizado e completamente paralelo [71]. As aplicações que envolvem processamento digital de sinais baseiam-se em operações binárias de multiplicações e também acumuladores. A utilização de estruturas otimizadas para desenvolver essas operações representa benefícios em relação à velocidade e eficiência, visto que os DSP48E1 são dedicados, customizados e de baixo consumo.

Dentre as funcionalidades do DSP48E1, destaca-se [71]:

- Multiplicadores 25x18 em complemento de dois.
- Acumulador de 48 bits.
- Pipeline e barramentos dedicados opcionais para operações em cascata.
- Unidade aritmética de instrução simples e múltiplos dados.
- Suporte na detecção de *Overflow* e *Underflow*.

A Figura A.5 apresenta um diagrama das funcionalidades básicas do DSP48E1.

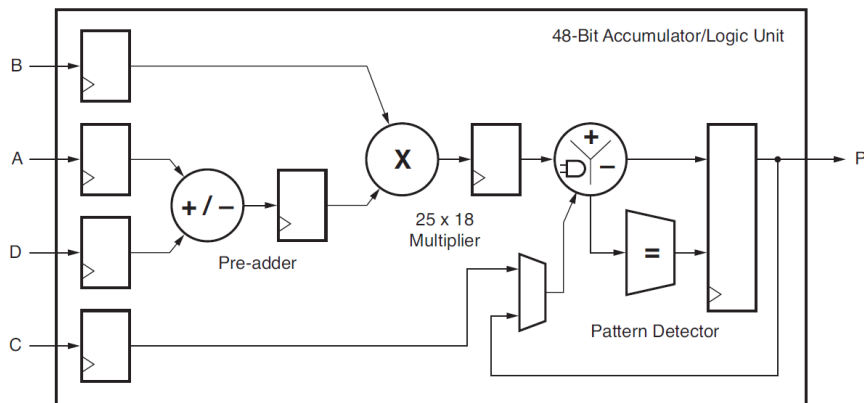


Figura A.5: Diagrama das funcionalidades básicas do DSP48E1 [71].

A.2.3 Bloco de Lógica Configurável

Os blocos de lógica configurável (*CLB*, do inglês *Configurable Logic Block*) são os principais recursos lógicos utilizados para implementação de circuitos sequenciais e

combinacionais. Essas estruturas são compostas, basicamente, por duas unidades lógicas e são conectadas a uma matriz de comutadores [81]. A Figura A.6 apresenta sua estrutura interna básica, sendo esta a implementação específica da Xilinx para o modelo conceitual apresentado na Figura A.3.

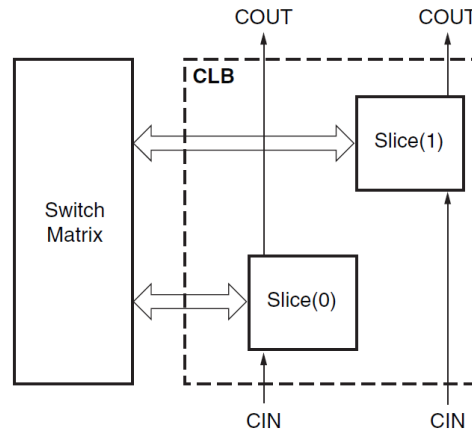


Figura A.6: Diagrama das funcionalidades básicas do bloco de lógica configurável[81].

Os elementos identificados como *Slice* na Figura A.6 são formados por quatro *look-up table*, de 6 entradas e duas saídas, todas independentes, e oito elementos de armazenamento, dos quais quatro podem ser configurados como *flip-flop* do tipo D ou *latches* e os outros quatro podendo ser utilizados apenas como *flip-flop* do tipo D. Sendo assim, cada CLB possui 8 LUTs e 16 elementos de armazenamento (*flip-flops* ou *latches*). Cada *LUT* é utilizada para realizar circuitos lógicos, operações aritméticas, função de memória ROM, registrador de deslocamento, entre outras funcionalidades.

SISTEMA PARA REDUÇÃO DA SOBREPOSIÇÃO DE SINAIS EM CALORIMETRIA DE ALTAS ENERGIAS EM OPERAÇÃO COM ALTA TAXA DE EVENTOS

IGO A. DOS S. LUZ*, EDUARDO F. S. FILHO*, PAULO C. M. A. FARIAS*, MATHEUS BECK*,
LUCIANO M. ANDRADE†, JOÃO PAULO B. S. DUARTE†, JOSÉ M. DE SEIXAS‡

*Programa de Pós Graduação em Engenharia Elétrica, Escola Politécnica
Universidade Federal da Bahia, Salvador, Bahia, Brasil

†Programa de Pós Graduação em Engenharia Elétrica
Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais, Brasil

‡Laboratório de Processamento de Sinais COPPE/POLI
Universidade Federal do Rio de Janeiro, Rio de Janeiro, Rio de Janeiro, Brasil

Emails: igoamauri@gmail.com, eduardo.simas@ufba.br, paulo.farias@ufba.br,
matheus.alencarbeck@gmail.com, luciano.andrade@engenharia.ufjf.br,
joao.duarte@engenharia.ufjf.br, seixas@lps.ufrj.br

Abstract— Real-time systems have some challenges in terms of performance that must be met in your project, such as: processing time, power consumption, area consumption, among others. Thus, due to these restrictions, sometimes, these projects need to be developed in dedicated hardware. A problem in digital signal processing systems is the superimposition of information from subsequent events. This signal distortion is called pileup and occurs very often in the particle detectors. This work proposes a dedicated signal processing system embedded on a digital high-performance reconfigurable electronic platform for reduction of the pileup in a calorimeter. To this end, a parallel implementation of digital finite impulse response filters was used. Each filter represents an approximation of the inverse of the measurement channel impulse response, performing signal deconvolution. The obtained results indicate the efficiency of the proposed solution and the compliance to the time latency required for the application.

Keywords— FPGA, Digital Filter, Deconvolution, Particle physics, Calorimeter.

Resumo— Sistemas que funcionam em tempo real possuem diversos desafios em termos de desempenho que precisam ser atendidos em seu projeto, tais como: tempo de processamento, consumo de potência, área utilizada, entre outros. Dessa forma, para atender a essas restrições, esses projetos precisam ser desenvolvidos em *hardware* dedicado. Um problema que ocorre nos sistemas de processamento digital de sinais é a sobreposição entre informações provenientes de eventos subsequentes. Essa distorção do sinal é chamada empilhamento (do inglês *pileup*) e ocorre com muita frequência no âmbito dos detectores de partículas. Esse trabalho propõe um circuito eletrônico dedicado implementado em um sistema digital reconfigurável de alto desempenho para redução do empilhamento nos sinais medidos em um calorímetro. Para isso, são utilizados filtros digitais de resposta ao impulso finita implementados em paralelo. Cada filtro é uma aproximação da inversa da resposta ao impulso do canal de medição, realizando a desconvolução dos sinais medidos. Os resultados obtidos mostram a eficiência da técnica proposta e a adequação aos requisitos de tempo de processamento da aplicação.

Palavras-chave— FPGA, Filtro Digital, Desconvolução, Física de Partículas, Calorímetro.

1 Introdução

Sistemas para aplicações de processamento de sinais que exigem melhores desempenhos em relação ao tempo de processamento, consumo de potência, entre outros quesitos, precisam ser desenvolvidos em *hardware* dedicado, com características específicas, para alcançar os resultados esperados. Nesse cenário, apresenta-se os sistemas em tempo real, que tem seu tempo de processamento restrito à taxa de geração de eventos externos, dessa forma, muitas vezes não sendo viável a utilização de equipamentos de propósito geral para o processamento dos dados.

No processamento digital de sinais, existem sistemas em tempo real que são afetados pela sobreposição da informação de eventos subsequentes, fenômeno conhecido como *pileup*. Esse problema está associado à alta taxa de ocorrência dos eventos e ao tempo de resposta dos equipamentos

responsáveis pela detecção dos mesmos. Dessa forma, a depender da frequência de geração dos eventos, torna-se necessário projetar uma solução em *hardware* dedicado para reduzir os efeitos do empilhamento e recuperar a informação real. Essa recuperação da informação pode ser feita através do processo de filtragem, utilizando filtros digitais.

O efeito de empilhamento ocorre em sistemas das mais diversas áreas, dentre elas pode-se citar a Física de Partículas. Essa área de atuação é responsável por estudar a composição da matéria, as partículas subatômicas e suas interações (Fenow, 1986). O Centro Europeu para Pesquisas Nucleares (CERN), um dos centros de pesquisa de referência em Física de Partículas, desenvolveu o *Large Hadron Collider* (LHC) (Evans and Bryant, 2008; Filho, 2010), sendo este o maior acelerador de partículas em operação. Além desse, outros centros também desenvolveram diferentes equipamentos, como o *Relativistic Heavy Ion Col-*

lider (RHIC), desenvolvido pelo Laboratório Nacional de Brookhaven, o *Tevatron*, desenvolvido pelo *Fermilab*, entre outros. Nesses equipamentos, o efeito de *pileup* ocorre nos detectores devido à relação entre a frequência de ocorrência das colisões e ao tempo de resposta dos sensores responsáveis por transformá-las em sinais digitais.

Esse trabalho propõe o desenvolvimento de um sistema digital responsável por reduzir os efeitos do empilhamento de eventos no Calorímetro do LHC, tendo como objetivo garantir o correto funcionamento do sistema de seleção de eventos de interesse. Devido à restrição de tempo da aplicação, foi necessário desenvolver uma implementação em *hardware* dedicado (em *Field Programmable Gate Array - FPGA*), para executar o processamento em paralelo do conjunto de dados dos sensores.

No FPGA, o processamento das amostras é realizado utilizando filtros digitais, projetados através de um conjunto de dados simulados, que consideram o funcionamento real dos detectores, e capazes de realizar uma filtragem inversa (desconvolução). A arquitetura do filtro desenvolvida foi a Resposta de Impulso Finita (*Finite Impulse Response - FIR*) em sua forma transposta (Meyer-Baese, 2007).

Esse artigo é dividido em cinco seções: A Seção 2 descreve o problema da sobreposição de sinais no âmbito dos detectores de partículas; a Seção 3 aborda o método proposto para recuperação do sinal; a Seção 4 descreve o sistema em FPGA; a Seção 5 apresenta os resultados simulados obtidos com a implementação do sistema proposto; a Seção 6 apresenta a conclusão.

2 Detectores de Partículas

Os detectores de partículas são equipamentos utilizados para identificar características das partículas, separando-as no tempo e no espaço, para que possam ser associadas a determinados eventos (Martin, 2009). Para alcançar a precisão necessária para os experimentos, utilizam-se diferentes tipos de detectores, cada um relacionado a uma determinada característica que deseja-se identificar. Por exemplo: detector a gás, responsável por detectar a ionização produzida pela passagem da carga de uma partícula por um determinado gás; calorímetro, usado para medir a energia e a posição da partícula através de total absorção; entre outros.

Dentre os calorímetros em funcionamento, pode-se citar o conjunto que compõe o ATLAS, um dos detectores de propósito geral do LHC (Aad et al., 2008). Este é dividido em camadas, 4 referentes aos Calorímetros Eletromagnéticos e 3 aos Hadrônicos (Aad et al., 2010). A Figura 1 ilustra um corte transversal nesse conjunto de calorímetros.

O efeito de empilhamento de eventos que

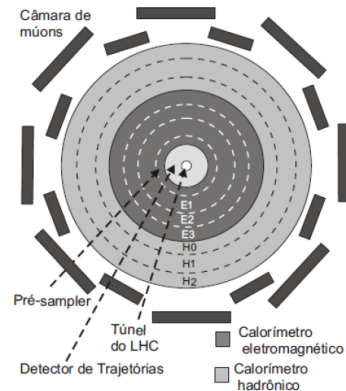


Figura 1: Corte transversal dos calorímetros do ATLAS.

acontece nos detectores de partículas, como o ATLAS, compromete o sistema de detecção e seleção das eventos de interesse desses sistemas, pois este provoca a distorção do padrão dos sinais. A Figura 2 ilustra o pulso típico do sensor de um calorímetro do ALTAS, tendo esse uma duração da ordem de 150ns. Considerando-se um sistema que tenha, por exemplo, uma frequência de ocorrência de colisões de 40MHz, esse sensor permaneceria sensibilizado por até 6 colisões subsequentes. A Figura 3 mostra o efeito do empilhamento nesse sinal.

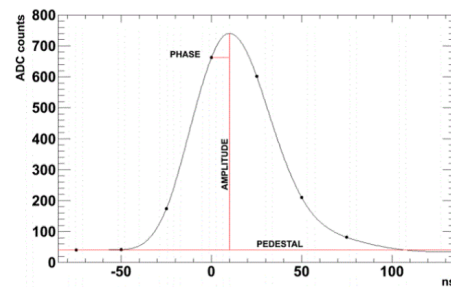


Figura 2: Resposta ao Impulso Típica de um sensor (Cavalcanti et al., 2014).

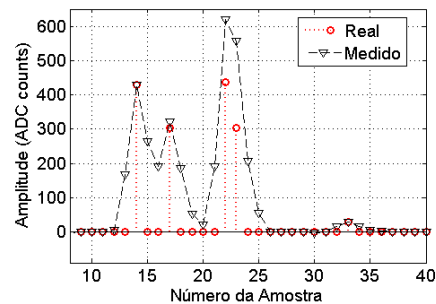


Figura 3: Efeito de empilhamento em um sinal medido (Cavalcanti et al., 2014).

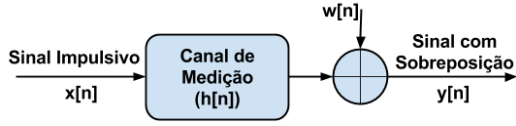


Figura 4: Modelo da sobreposição no Calorímetro.

3 Método Proposto

A técnica de estimação de energia utilizada nesse trabalho baseia-se no princípio da filtragem inversa (ou desconvolução).

A Eq 1, na qual $\mathbf{h}[n]$ e $\mathbf{g}[n]$ são as respostas ao impulso e $\delta[n]$ representa um impulso unitário, apresenta uma relação em que dois sistemas lineares, invariantes no tempo e causais no tempo discreto são inversos.

$$\mathbf{h}[n] * \mathbf{g}[n] = \delta[n] \quad (1)$$

Considerando que um determinado sinal $\mathbf{x}[n]$ que foi transmitido em um canal de transmissão com resposta ao impulso $\mathbf{h}[n]$, nesse caso, $\mathbf{y}[n] = \mathbf{x}[n] * \mathbf{h}[n]$ representará o sinal $\mathbf{x}[n]$ distorcido pelo canal $\mathbf{h}[n]$. Na Figura 4 observa-se o processo de distorção do sinal impulsivo em um Calorímetro, nesta, $\mathbf{x}[n]$ representa o sinal de energia referente a uma colisão, $\mathbf{h}[n]$ a resposta ao impulso do canal de medição do Calorímetro modelado, $\mathbf{w}[n]$ representa um ruído de medição, e, por fim, $\mathbf{y}[n]$ o sinal resultante do processo de medição.

Para efetuar a recuperação de $\mathbf{x}[n]$, deve-se convolver $\mathbf{y}[n]$ com $\mathbf{g}[n]$, que representa a resposta ao impulso do canal inverso de $\mathbf{h}[n]$. Sendo assim, o resultado desse processamento ($\mathbf{v}[n]$) será uma aproximação da entrada $\mathbf{x}[n]$. A Eq. 2 ilustra essas relações.

$$\mathbf{y}[n] = \mathbf{x}[n] * \mathbf{h}[n] \rightarrow \mathbf{v}[n] = \mathbf{y}[n] * \mathbf{g}[n] \simeq \mathbf{x}[n] \quad (2)$$

Este sistema inverso pode ser representado no domínio Z , para isso, aplica-se a Transformada Z em ambos os lados da Eq. 1, então, obtém-se o sistema inverso de $\mathbf{H}(z)$ (Eq. 3).

$$\mathbf{G}(z) = \frac{1}{\mathbf{H}(z)} \quad (3)$$

Com o intuito de propor uma solução eficiente de projeto do filtro, estudos foram realizados com diferentes técnicas (Duarte, 2015). De acordo com os resultados satisfatórios das análises (Duarte, 2015), decidiu-se utilizar, nesse trabalho, o método que baseia-se em uma abordagem determinística para obter um FIR que desconvolve $\mathbf{y}[n]$ e, então, recupera o sinal de energia real $\mathbf{x}[n]$.

Na abordagem determinística, encontram-se os coeficientes do filtro inverso a partir do pulso característico $\mathbf{h}[n]$ do detector. No domínio z , obtém-se o filtro inverso $\mathbf{G}(z)$, consequentemente,

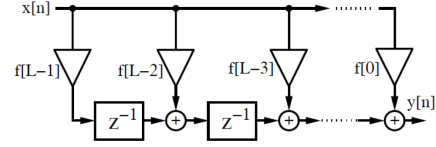


Figura 5: Forma Transposta do Filtro FIR.

os coeficientes utilizados para desconvolver os sinais com sobreposição (Duarte, 2015).

Ao se analisar o espectro do pulso característico desse detector no domínio z ($\mathbf{H}(z)$), considerando um sistema causal, o filtro inverso ($\mathbf{G}(z)$) não é realizável. Para se obter uma aproximação de um filtro realizável, adotou-se uma estratégia a partir da separação de $\mathbf{G}(z)$ em dois polinômios $\mathbf{G}_1(z)$ e $\mathbf{G}_2(z)$, sendo o primeiro composto pelos pólos internos ao círculo unitário, caracterizando-se como estável, enquanto que o segundo possui pólos externos ao círculo unitário, caracterizando-se como instável. Sendo assim, trata-se $\mathbf{G}_2(z)$ para obter uma aproximação estável e causal do mesmo. A partir da resposta ao impulso obtida de $\mathbf{G}_2(z)$, realiza-se a convolução com a resposta ao impulso do sistema $\mathbf{G}_1(z)$, obtendo, assim, a resposta ao impulso de uma aproximação do sistema inverso $\mathbf{G}(z)$ realizável (Duarte, 2015).

Nesse trabalho, optou-se pela aplicação de técnicas de filtros FIR para a implementação do filtro inverso (Duarte, 2015), pois possuem duração finita (Lyons, 2011). Esses filtros têm resposta de fase linear, não possuem realimentação, sendo sempre estáveis (Gnanasekaran and Manikandan, 2013). A saída de um filtro FIR pode ser expressa pela Eq. 4, sendo L a ordem do filtro (Lyons, 2011).

$$\mathbf{y}[n] = \mathbf{x}[n] * \mathbf{h}[n] = \sum_{k=0}^{L-1} \mathbf{h}[k] \mathbf{x}[n-k] \quad (4)$$

Existem diferentes estruturas de implementação de um filtro FIR, neste trabalho foi utilizada a forma transposta (Figura 5) (Meyer-Baese, 2007). Em relação aos benefícios dessa arquitetura, afirma-se que não é necessário registrador de deslocamento extra para o $\mathbf{x}[n]$ e também não é necessário estágio extra de *pipeline* para os somadores do produto para que se obtenha alta taxa de processamento dos sinais (Meyer-Baese, 2007).

4 Sistema Eletrônico Desenvolvido

Cada módulo desse sistema, ilustrado na Figura 6, é responsável por processar os sinais de 12 canais. A informação de energia é enviada de forma serial para o FPGA e o sistema precisa disponibilizar a informação processada também de forma serial.

Na Figura 6 pode-se identificar a arquitetura do sistema proposto. Os blocos “Conversor Serial/Paralelo” e “Paralelo/Serial” são responsáveis por fazer a conversão do sinal de entrada e do sinal de saída do sistema, respectivamente. O bloco “Configuração Filtro” é responsável por armazenar palavras de configuração que serão utilizadas pelas implementações dos filtros, por exemplo, seus coeficientes. O bloco “Controle” garante que, à medida que os sinais de entrada sejam convertidos, estes sejam enviados para a instância do filtro responsável por executar o processamento das amostras do respectivo canal. O “Banco de Filtros” agrupa os filtros digitais responsáveis pela filtragem dos sinais.

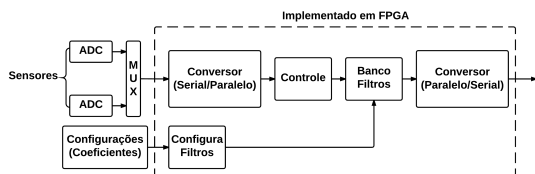


Figura 6: Arquitetura do Sistema Proposto.

Para ter um melhor desempenho em relação ao tempo de processamento, a arquitetura foi planejada para implementar técnicas de *pipeline* e de diferentes domínios de *clock*. O sistema foi dividido, basicamente, em três estágios: conversão do sinal de entrada; processamento do sinal; conversão do sinal de saída. O primeiro estágio é responsável por converter o sinal de entrada serial em paralelo, sendo assim, criando a palavra que representa o sinal de energia capturado de um determinado sensor. Essa informação é processada no segundo estágio, responsável por aplicar o filtro. E, por fim, o terceiro estágio converte o sinal processado em serial para ser disponibilizado na saída do sistema. Essa técnica garante que os três estágios funcionem de forma paralela, ou seja, enquanto o dado de entrada está sendo convertido, outro dado é processado no segundo estágio, consequentemente, uma outra amostra, que já passou pelo processo de filtragem, pode ser convertida em serial na saída do sistema.

O terceiro estágio foi projetado para utilizar uma frequência de *clock* duas vezes mais rápida do que o restante do sistema. Com isso, garante-se que esse bloco finalizará a conversão do sinal de saída mais rápido que os demais estágios. Ou seja, quando um sinal de energia filtrado for disponibilizado pelo segundo estágio, o último bloco estará pronto para fazer a conversão.

5 Resultados

O sistema eletrônico proposto foi desenvolvido utilizando-se os dispositivos Virtex-7 (XC7VX485T) (Xilinx, 2015) e Kintex-7

(XC7K480T) (Xilinx, 2015), e foi testado com sinais de energia simulados. Foi utilizado um modelo em MATLAB® que simula o comportamento dos sensores em cada colisão, gerando, assim, os sinais de energia referentes a cada evento. O simulador utiliza a resposta ao impulso típica de um canal de medição.

Para a execução dos testes da implementação em FPGA, os dados de entrada precisam ser representados em ponto fixo. A representação, tanto dos coeficientes como dos sinais de energia, foi definida em complemento de dois. Sendo assim, foi definida a quantidade de bits para representação das partes inteira e fracionária do valor real. Além disso, para evitar *overflow*, foi utilizada a técnica de saturação (Constantinides et al., 2003).

Para a validação do projeto, foram executadas simulações com diferentes cenários, alterando a ordem do filtro e utilizando o mesmo conjunto de sinais simulados. Como modelo de referência, foi utilizada a função em MATLAB® que implementa o filtro digital FIR. Dessa forma, realizou-se a filtragem utilizando o mesmo conjunto de entradas que foram utilizadas no dispositivo FPGA.

Nos resultados das simulações em FPGA, os sinais de energia foram representados com 12 bits e os coeficientes com 18 bits, sendo esta resolução definida de acordo com análises preliminares dos valores simulados. A Figura 7 a) e b) ilustra o sinal simulado sem e com empilhamento respectivamente, ambos representados em ponto fixo.

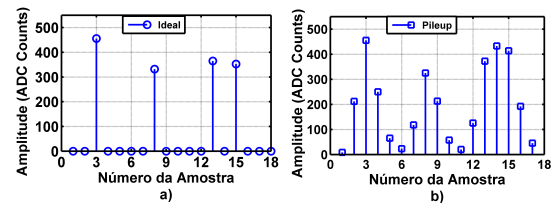


Figura 7: Exemplos de sinais simulados do Tilecal a) sem empilhamento e b) com empilhamento.

Foram realizadas análises referentes ao erro médio e à raiz quadrada do erro quadrático médio. Considerou-se os resultados obtidos com as implementações no FPGA e no MATLAB®. As Tabelas 1 e 2 mostram os resultados alcançados variando a ordem do filtro entre 5, 10, 15, 20 e 30. De acordo com esses resultados, observa-se que a ordem 20 teve um melhor desempenho.

Tabela 1: Valor médio do erro

Ordem do Filtro	FPGA (ADC counts)	Matlab (ADC counts)
Ordem 5	-0,9727	-1,0981
Ordem 10	0,5128	0,3914
Ordem 15	0,2160	0,0903
Ordem 20	0,0988	-0,0317
Ordem 30	0,3552	0,2176

A Figura 8 apresenta o resultado da implementação em FPGA utilizando filtros de ordem 20. Esse gráfico foi gerado considerando os sinais

Tabela 2: Raiz Quadrada do Erro Quadrático Médio

Ordem do Filtro	FPGA(ADC counts)	Matlab(ADC counts)
Ordem 5	24,95	24,95
Ordem 10	19,86	19,85
Ordem 15	19,33	19,32
Ordem 20	19,21	19,20
Ordem 30	20,30	20,29

resultantes da filtragem em FPGA e também os sinais simulados ideais de energia. Sendo assim, comparando o resultado da reconstrução de cada *Bunch Crossing* com o seu respectivo valor ideal.

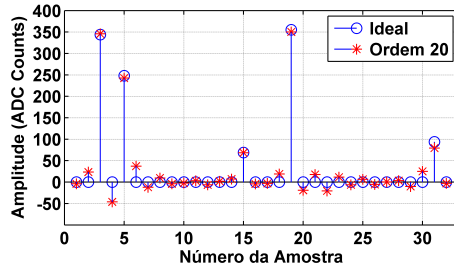


Figura 8: Sinal de saída do filtro de ordem 20 implementado em FPGA comparado ao sinal ideal.

Foram desenvolvidas, também, análises gráficas referentes ao erro. A Figura 9 a) ilustra o gráfico de dispersão entre os resultados da simulação da implementação FPGA, utilizando o filtro de ordem 20, e os valores ideais de energia simulados dos sensores. A Figura 9 b) apresenta o gráfico de dispersão construído a partir dos resultados do MATLAB® e do FPGA, considerando a mesma ordem de filtro. Através desta análise, pode-se observar que o sistema desenvolvido para o dispositivo FPGA apresentou comportamento bastante semelhante ao do modelo de referência.

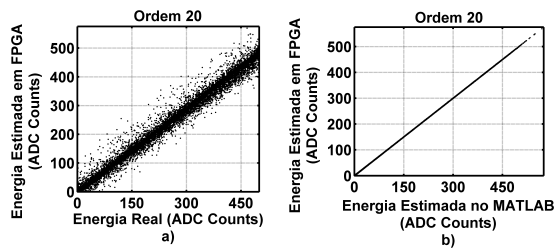


Figura 9: Gráficos de dispersão entre a energia estimada pela implementação em FPGA e a) energia real, e b) energia estimada pela implementação em MATLAB

Em relação ao comportamento temporal, o sistema foi simulado com uma frequência de clock de 800MHz. Esse sistema utiliza 144 ciclos de *clock* para processar as amostras dos canais referentes a uma colisão, ou seja, são necessários 180ns. Considerando a ocorrência de uma colisão a cada 25ns, são necessárias aproximadamente 8 colisões para se obter os sinais filtrados.

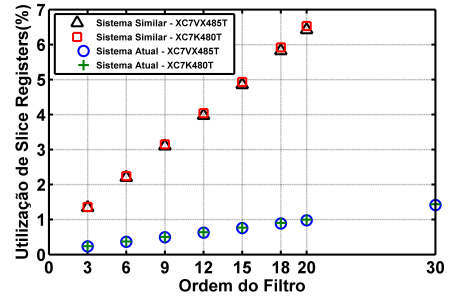


Figura 10: Utilização de *Slice Register*.

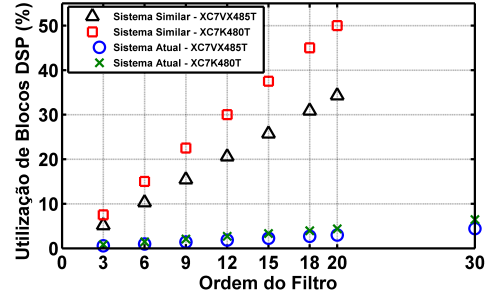


Figura 11: Utilização de blocos de *DSP*.

Em relação à ocupação de recursos dos dispositivos FPGA, a Figura 10 apresenta a utilização da lógica configurável (*Slice Register*), enquanto que a Figura 11 mostra a utilização das estruturas de operações lógicas e aritméticas (*DSP*), em ambas variou-se a ordem do filtro. Essas análises tiveram como alvo ambos os dispositivos da Xilinx, tanto o Virtex-7 como o Kintex-7. A partir desses gráficos, observa-se que a arquitetura proposta nesse trabalho utiliza menos recursos dos dispositivos FPGAs do que uma solução similar, anteriormente proposta (Cavalcanti et al., 2014). Essa outra implementação possui uma estrutura de filtro para cada canal de medição, utilizando, assim, muitas estruturas de DSP e lógica reconfigurável, as quais ficam ociosas durante determinados períodos de seu funcionamento. No sistema proposto nesse trabalho, compartilharam-se as estruturas de *DSP* entre os filtros e também foi otimizado o caminho dos sinais de energia através das técnicas de *pipeline* e múltiplos *clocks*.

6 Conclusões

Esse trabalho propõe a reconstrução de um sinal com sobreposição através de uma implementação em FPGA de Filtro Digital do tipo FIR projetado de acordo com um método determinístico. Através da análise dos resultados da filtragem do sinal, pode-se observar que o mesmo foi reconstruído de forma satisfatória. Essa análise pode ser confirmada observando os gráficos dos erros, onde constata-se a eficiência e o comportamento esperado do sistema FPGA implementado. Como

trabalho futuro, propõe-se utilizar outro tipo de projeto de Filtro, dessa vez considerando um método iterativo.

Agradecimentos

Os autores agradecem à FAPESB, FAPERJ, CAPES, CNPq e RENAFAP (MCTI) pelo apoio financeiro e à colaboração ATLAS pelas contribuições para o desenvolvimento deste trabalho.

Referências

- Aad, G., Abat, E., Abdallah, J. et al. (2008). The ATLAS experiment at the cern large hadron collider, *Journal of Instrumentation* **3**(8).
- Aad, G., Abbott, B., Abdallah, J. et al. (2010). Readiness of the ATLAS tile calorimeter for LHC collisions, *The European Physical Journal C* **70**(4): 1193–1236.
- Cavalcanti, M. M., Filho, E. F. S., Farias, P. C. M. A. et al. (2014). Sistema digital para redução do empilhamento em medições de energia num detector de partículas, *Anais do XX Congresso Brasileiro de Automática* pp. 456–463.
- Constantinides, G. A., Cheung, P. Y. K. and Luk, W. (2003). Synthesis of saturation arithmetic architectures, *ACM Trans. Des. Autom. Electron. Syst.* **8**(3): 334–354.
- Duarte, J. P. B. S. (2015). *Estudo de técnicas de deconvolução para reconstrução de energia online no calorímetro hadrônico do ATLAS*, Master's thesis, Programa de Pós Graduação em Engenharia Elétrica / UFJF.
- Evans, L. and Bryant, P. (2008). LHC machine, *Journal of Instrumentation* **3**(08): S08001.
- Fernow, R. C. (1986). *Introduction to Experimental Particle Physics*, Cambridge University Press, New York, USA.
- Filho, E. F. S. (2010). *Análise não-linear de componentes independentes para uma filtragem online baseada em calorimetria de alta energia e com fina segmentação*, PhD thesis, Electrical Engineering Program / COPPE / UFRJ, Rio de Janeiro, BR.
- Gnanasekaran, M. and Manikandan, M. (2013). High throughput pipelined implementation of reconfigurable FIR filter for sdr, *Current Trends in Engineering and Technology (IC-CTET), 2013 International Conference on*, IEEE, pp. 362–364.
- Lyons, R. G. (2011). *Understanding Digital Signal Processing*, third edn, Prentice Hall, Boston, MA, USA.
- Martin, B. R. (2009). *Nuclear and Particle Physics: An Introduction*, second edn, Wiley, USA.
- Meyer-Baese, U. (2007). *Digital Signal Processing with Field Programmable Gate Arrays*, third edn, Springer, Tallahassee, Florida, USA.
- Xilinx (2015). *7 Series FPGAs Overview*. v1.17.