



**Universidade Federal de Campina Grande**

**Centro de Engenharia Elétrica e Informática**

Coordenadoria do Curso de Graduação em Engenharia Elétrica

NELSON CARLOS DE SOUSA CAMPOS

**RELATÓRIO DE ESTÁGIO SUPERVISIONADO NO  
LABORATÓRIO DE ARQUITETURAS DEDICADAS (LAD)**

Campina Grande, Paraíba  
Julho de 2015

NELSON CARLOS DE SOUSA CAMPOS

RELATÓRIO DE ESTÁGIO SUPERVISIONADO NO  
LABORATÓRIO DE ARQUITETURAS DEDICADAS (LAD)

*Relatório de Estágio  
Supervisionado submetido à  
Unidade Acadêmica de  
Engenharia Elétrica da  
Universidade Federal de  
Campina Grande como parte  
dos requisitos necessários para  
a obtenção do grau de bacharel  
em Ciências no Domínio da  
Engenharia Elétrica*

Área de Concentração: Processamento Digital de Sinais

Orientador:

Prof. Dr. Marcos Ricardo Alcântara Moraes

Campina Grande, Paraíba

Julho de 2015

NELSON CARLOS DE SOUSA CAMPOS

## RELATÓRIO DE ESTÁGIO SUPERVISIONADO NO LABORATÓRIO DE ARQUITETURAS DEDICADAS (LAD)

*Relatório de Estágio  
Supervisionado submetido à  
Unidade Acadêmica de  
Engenharia Elétrica da  
Universidade Federal de  
Campina Grande como parte  
dos requisitos necessários para  
a obtenção do grau de bacharel  
em Ciências no Domínio da  
Engenharia Elétrica*

Área de Concentração: Processamento Digital de Sinais

Aprovado em     /     /

**Prof. Dr. Jaidilson Jó da Silva**  
Universidade Federal de Campina Grande  
Avaliador

**Prof. Dr. Marcos Ricardo Alcântara Morais**  
Universidade Federal de Campina Grande  
Orientador, UFCG

Dedico este trabalho a meu pai (*in memoriam*) que me ensinou que um homem de verdade honra a sua família.

## AGRADECIMENTOS

Agradeço primeiramente à Deus, que me concedeu a vida e que nunca me decepcionou mediante a fé que sempre tive em Sua Palavra.

Agradeço também à minha mãe, Salete, pelo seu esforço sobrenatural para prover o sustento de seus filhos.

Agradeço também a toda minha família, que sempre me apoiou e me incentivou a seguir em frente.

Agradeço ao professor Marcos Ricardo Alcântara Moraes, por ter aceitado orientar este trabalho e por todos os conselhos ao longo da graduação. Ao professor Elmar Uwe Kurt Melcher, pela formação profissional que adquiri no Brazil-IP. Ao professor Heron Aragão Monteiro, pela parceria neste trabalho.

Agradeço a toda equipe do LAD, que foram essenciais para o progresso deste trabalho.

Enfim, agradeço a todas as pessoas que estiveram presentes em minha vida e que não as listarei aqui, mas elas merecem toda a consideração por terem cruzado no meu caminho.

עיסתו על מעיד הנחתום אין  
*Eyn Hanah'tom Meid Al Eesato*

O padeiro não deve comentar sobre sua massa.

Significado: Uma pessoa que faz algo não deve louvar a si própria.

Provérbio hebraico

## LISTA DE ILUSTRAÇÕES

Figura 1. Chip do MPEG-4 .....	2
Figura 2. Chip do SPVR .....	3
Figura 3. Demonstração em FPGA do PLC .....	3
Figura 4. O dotProject.....	4
Figura 5. O Gerente Administrativo do LAD.....	5
Figura 6. Arquitetura do PLCM.....	7
Figura 7. Representação das possíveis configurações de uma característica.....	8
Figura 8. Representação da soma dos pixels na região D utilizando a integral da imagem.....	9
Figura 9. Representação dos classificadores em cascata .....	10
Figura 10. Aproximação do logaritmo da função de Bessel .....	12
Figura 11. Síntese do circuito da raiz quadrada .....	13
Figura 12. Representação de um buffer circular.....	15
Figura 13. Integração de SystemC com OpenCV .....	16

## LISTA DE SIGLAS E ABREVIATURAS

- AVI – Audio Video Interleave
- Brazil-IP - Brazil Intellectual Property
- DSC – Departamento de Sistemas e Computação
- FPGA – Field Programmable Gate Array
- IP Core – Intellectual Property Core
- LAD – Laboratório de Arquiteturas Dedicadas
- LPCA – Linear Predictive Coefficients Analysis
- PLC – *Power Line Communications*
- PLCM – *Power Line Communication Modem*
- S-FSK – Spread Frequency Shift Keying
- SoC – System on Chip
- SPVR – Speaker Verification
- UFMG – Universidade Federal de Campina Grande
- UVM – Universal Verification Methodology
- UVMC – UVM Connect
- VLSI – Very Large Scale Integration

## SUMÁRIO

<b>1 Introdução</b> .....	1
1.1 O LAD .....	1
<b>2 Objetivos</b> .....	5
<b>3 Embasamento</b> .....	5
3.1 PLC.....	5
3.2 VideoHD.....	7
<b>4 Atividades Desenvolvidas</b> .....	11
<b>5 Conclusões</b> .....	17
<b>Referências</b> .....	18

# 1 INTRODUÇÃO

Este trabalho descreve as atividades realizadas durante o estágio supervisionado no Laboratório de Arquiteturas Dedicadas (LAD) da Universidade Federal de Campina Grande (UFCG). Foram desenvolvidos algoritmos em hardware para o Projeto PLC do Brazil-IP, bem como a construção de modelos de referência em SystemC de alguns módulos do projeto. Em paralelo, estudos de técnicas de Visão Computacional foram necessárias para a elaboração de uma plataforma de captura e tratamento de vídeo em Full HD. A integração de SystemC com OpenCV foi necessária, uma vez que pretende-se criar modelos de referência de alto nível para extração de características de faces utilizando o algoritmo de Viola Jones.

## 1.1 O LAD

O LAD começou suas atividades no fim da década de 1990 e é coordenado pelo Professor Elmar Uwe Kurt Melcher, sendo composto atualmente por alunos de graduação, mestrado e doutorado em Ciência da Computação e Engenharia Elétrica. Desde 2003 o laboratório é vinculado ao programa do governo federal Brazil IP [1].

O Brazil-IP (Brazil Intellectual Property) é um consórcio de laboratórios e universidades brasileiras que visa à formação de recursos humanos em projeto de circuitos integrados, circuitos VLSI e IP Cores.

O consórcio surgiu em 2003 e era composto de 8 universidades brasileiras: USP, UFMG, UFCG, UFPE, UNB, UFRGS, PUCRS e UNICAMP, sendo financiado pelo Ministério da Ciência e Tecnologia. Hoje o programa está em sua segunda edição, sendo composto de 15 Universidades: UFPE, UFCG, UFPA, UFC, UFRN, UFPB, UFMS, UFS, UNIFEI, UNICAMP, UNESP, UFSC, UNIPAMPA, UNIVALI, UFSM.

Na primeira edição do programa, foram desenvolvidos os primeiros chips decodificadores de MP3 (sob-responsabilidade da Unicamp), MPEG4 e 8051 (sob-responsabilidade da UFPE) no Brasil, sendo o MPEG4 o chip mais complexo desenvolvido

por universidades brasileiras, desenvolvido pela UFCG, Universidade Federal de Campina Grande [2].

Na segunda edição do programa, o LAD desenvolveu o SPVR (Speaker Verification), cujo objetivo do chip é identificar a identidade vocal de um locutor. A técnica de extração de características da voz foi implementada utilizando LPCA (Linear Predictive Coefficients Analysis). Na terceira edição do programa, está sendo desenvolvido um modem PLC para transmissão de dados em banda estreita. A transmissão é feita utilizando a modulação S-FSK e o chip encontra-se em processo de desenvolvimento e verificação funcional.

Outros projetos são desenvolvidos no LAD, no âmbito da Pós-Graduação. Um deles é uma plataforma para captura e tratamento de vídeo em Full HD. O objetivo é a construção de uma arquitetura dedicada capaz de extrair características de faces humanas utilizando o algoritmo de Viola Jones.

As Figuras 1 e 2 ilustram respectivamente os chips do MPEG-4 e do SPVR.

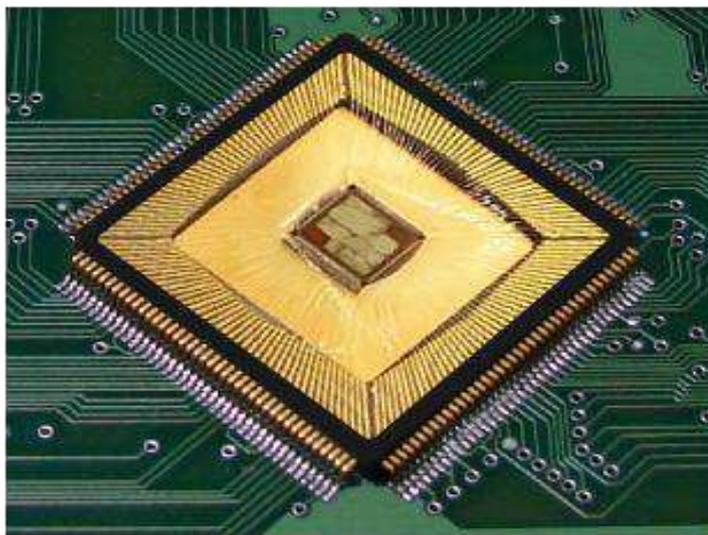


Figura 1 – Chip do MPEG-4

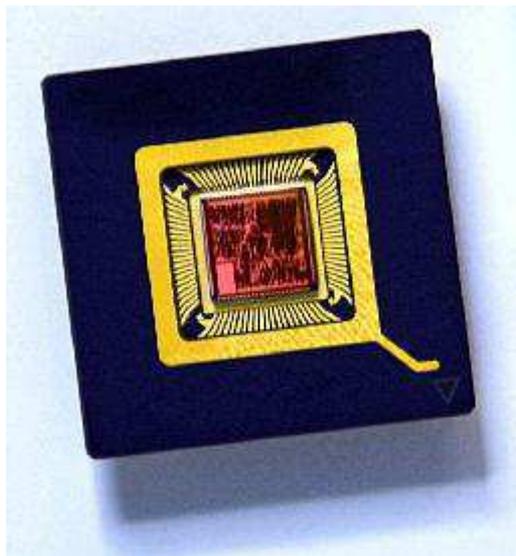


Figura 2 – Chip do SPVR

A Figura 3 ilustra a demonstração em FPGA do PLC.



Figura 3 – Demonstração em FPGA do PLC

O estágio no laboratório teve uma jornada de 20 horas semanais, onde 12 horas eram dedicadas às atividades do Brazil-IP, no projeto PLCM e as 8 horas restantes eram dedicadas ao projeto VideoHD.

O gerenciamento de horas de trabalho e cumprimento de tarefas no LAD é realizado pelo dotProject, ferramenta open source cuja finalidade é bem conveniente para este caso. Na Figura 4 ilustra-se uma janela do dotProject.

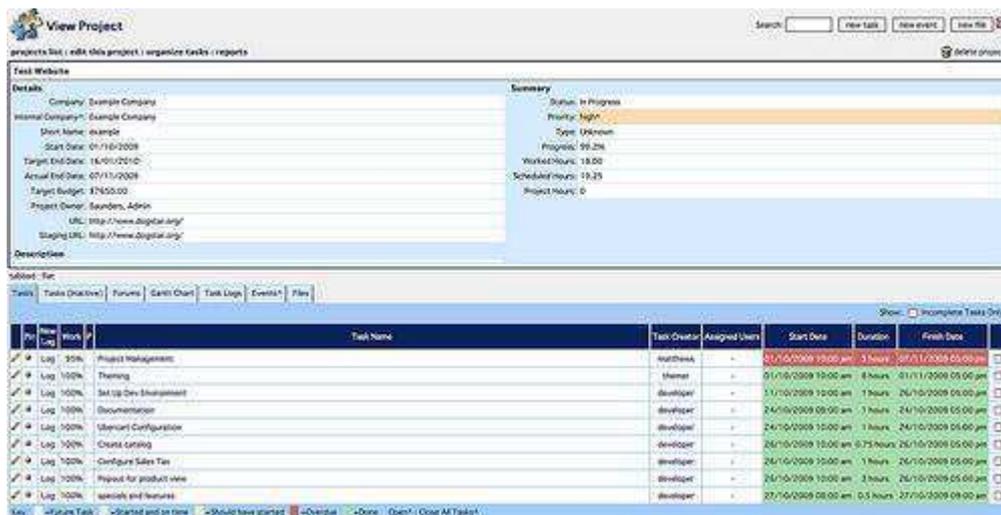


Figura 4 – O dotProject

Há duas câmeras de segurança no LAD, uma na entrada do corredor e outra dentro do laboratório. As câmeras são utilizadas como ferramentas de segurança e como maneira de gerenciar a presença das pessoas que trabalham no laboratório.

A equipe do laboratório contém o assistente administrativo Ozeías Lima, que está encarregado em gerenciar a monitoração das câmeras e do dotProject. Na Figura 5 ilustra-se o gerente administrativo do laboratório em plena atividade.



Figura 5 – Gerente Administrativo do LAD

## 2 OBJETIVOS

O objetivo deste trabalho é a implementação em Hardware de módulos para um modem Power Line Communication (PLC), construção de modelos de referência para verificação funcional dos mesmos, além do estudo de técnicas de visão computacional e a integração de SystemC com OpenCV.

## 3 EMBASAMENTO

Esta seção trata do embasamento teórico necessário para o desenvolvimento das atividades realizadas durante o estágio.

### 3.1 PLC

A transmissão de dados pela rede de distribuição de energia é geralmente cunhada com o termo PLC (Power Line Communication) e a tecnologia é estudada desde os anos 70. [3]

Redes PLC têm sido aplicadas em automação residencial, permitindo assim medições automáticas de consumo energético de dispositivos, como por exemplo, geladeiras, televisores, micro-ondas, etc.

Uma das vantagens do uso da rede elétrica para comunicação é a sua disponibilidade, não requerendo uma estrutura adicional para conexão dos nós, não sendo necessária a implantação de uma nova infraestrutura de cabos e interligações.

O PLC em questão tem taxas de transmissão em banda estreita (ordem de kbps) e as frequências de modulação são 65KHz e 87KHz, para a modulação dos bits 0 e 1 respectivamente. [3]

Nesse sentido, um padrão relevante é o IEC 61334, em particular, sua parte 5-1, que especifica as camadas física e MAC de um sistema PLC, definindo a modulação conhecida como S-FSK (Spread Frequency Shift Keying). O padrão, contudo, não especifica a técnica de demodulação, apesar de indicar um princípio heurístico que pode ser usado, dando espaço para avanços tecnológicos [3].

O projeto PLCM é um módulo IP-Core para comunicação pela rede elétrica e está sendo desenvolvido no Laboratório de Arquiteturas Dedicadas (LAD) da UFCG como parte do programa do governo federal Brazil-IP. O módulo deve implementar as camadas físicas e MAC de um sistema de comunicação pela rede elétrica em banda estreita. A arquitetura do PLCM está representada na Figura 1 e cada submódulo é descrito a seguir:

- TX\_MAC: Cálculo de CRC e composição das mensagens.
- ENC: Codificação para correção de erros.
- MOD: Modulação digital S-FSK.
- PLL: Multiplicação de frequência para sincronização.
- DEM: Demodulação digital S-FSK.
- DEC: Decodificação para correção de erros.
- RX\_MAC: Verificação de CRC e de endereçamento.
- STATUS/CTRL: Interface de controle e monitoramento (não incluso na arquitetura atual do projeto).

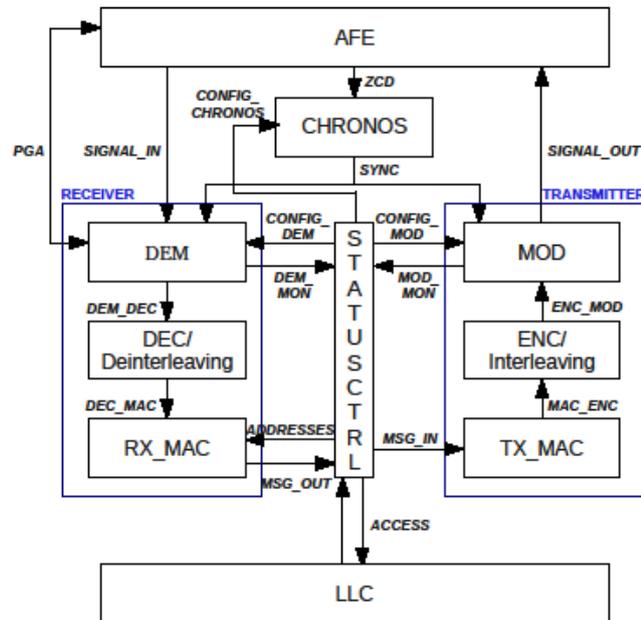


Figura 6 – Arquitetura do PLCM [3]

## 3.2 VIDEOHD

Detecção de Faces é uma tecnologia computadorizada que é amplamente utilizada em várias aplicações para identificar faces humanas em imagens digitais. Outra definição para a detecção de faces seria um processo fisiológico no qual seres humanos são capazes de identificar o rosto numa cena visual. [4]

São amplamente utilizados em biometria, geralmente como parte de um sistema de reconhecimento de faces. Aplicações em segurança, interface homem-máquina e gerenciamento de bancos de dados [4].

Um dos algoritmos mais utilizados para detecção de faces é o de Viola-Jones, proposto em 2001 por Paul Viola e Michael Jones [5]. Embora esta técnica seja utilizada para detectar outras classes de objetos, ela vem sendo amplamente empregada em detecção de faces.

O algoritmo de Viola-Jones tem quatro estágios:

1. Haar Features Selection;
2. Integral da Imagem;

3. Algoritmo de Treinamento Adaboost;
4. Classificadores em Cascata;

Os Haar Features são as unidades básicas do método de Viola-Jones, e também são chamados de características retangulares. Na Figura 2 ilustra-se as quatro possibilidades de configuração da característica. Teoricamente, uma característica pode ter dimensões e posições arbitrárias em uma janela.

O valor de uma característica sobre uma imagem pode ser calculado pela equação (1):

$$f(w) = \sum^w p_{preto} - \sum^w p_{branco} \quad (1)$$

onde:

$f(w)$ : valor da característica na janela  $w$

$\sum^w p_{preto}$  : somatório dos pixels na região preta

$\sum^w p_{branco}$  : somatório dos pixels na região branca

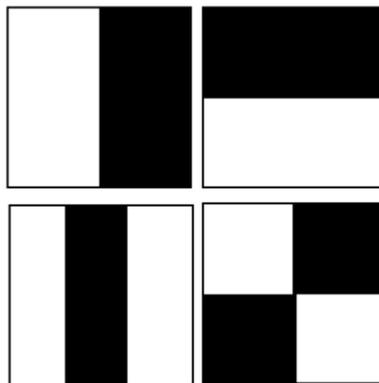


Figura 7 –Representação das possíveis configurações de uma característica

A integral da imagem é utilizada para acelerar o cálculo do valor de uma característica. O valor da integral é dado pela equação (2):

$$ii(x, y) = \sum_{x \leq x', y \leq y'} i(x', y') \quad (2)$$

onde:

$i(x', y')$ : valor do pixel na imagem na coluna  $x'$  e linha  $y'$

$ii(x, y)$ : valor da integral da imagem até a coluna  $x'$  e linha  $y'$

Por meio da equação (2), é possível calcular o valor do somatório dos pixels de uma dada área com complexidade  $O(1)$ , como ilustra a Figura 3.

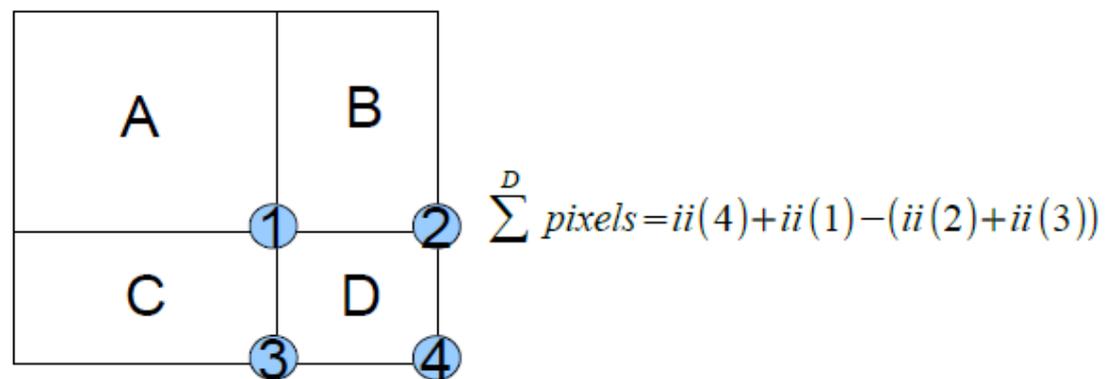


Figura 8 – Representação da soma dos pixels na região D utilizando a integral da imagem

A classificação de uma janela como uma que contém uma face ocorre com um classificador fraco. Um classificador fraco é definido pela função da equação (3):

$$h(w, p, f, \theta) = \begin{cases} 1, & \text{se } pf(w) < p\theta \\ 0, & \text{caso contrário} \end{cases} \quad (3)$$

onde:

$w$ : subjanela de  $24 \times 24$  pixels;

$f$ : característica;

$p$ : polaridade;

$\theta$ : threshold;

O próximo nível de classificação contém os classificadores fortes, que são composições de vários classificadores fracos. Eles são definidos pela função da equação (4):

$$C(w) = \begin{cases} 1, & \text{se } \sum_{t=1}^T \alpha_t h_t(w) \leq \frac{1}{2} \sum_{t=1}^T h_t(w) \\ 0, & \text{caso contrário} \end{cases} \quad (4)$$

onde:

$\alpha_t$ : constante calculada durante o treinamento;

$h_t(w)$ : valor do  $t$  – ésimo classificador fraco;

$T$ : número de classificadores fracos;

Os classificadores estão organizados em cascata, e a Figura 4 ilustra o diagrama de blocos dos classificadores. O uso destes classificadores a detecção de casos negativos, acelerando a execução do algoritmo. [6]

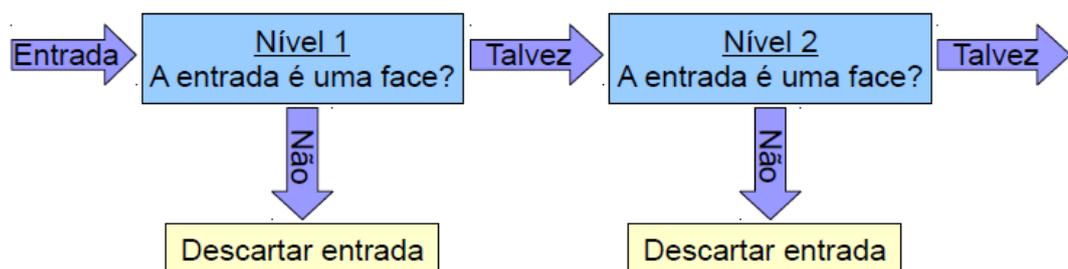


Figura 9 – Representação dos classificadores em cascata [6]

O AdaBoost é um algoritmo de aprendizado de máquina, inventado por Yoav Freund e Robert Schapire<sup>1</sup>. É um algoritmo meta-heurístico, e pode ser utilizado para aumentar a performance de outros algoritmos de aprendizagem. [7]

O nome **AdaBoost** deriva de *Adaptive Boosting* (em português, impulso ou estímulo adaptativo). O AdaBoost é adaptável no sentido de que as classificações subsequentes feitas são ajustadas a favor das instâncias classificadas negativamente por classificações anteriores. [7]

O AdaBoost é sensível ao ruído nos dados e casos isolados. Entretanto para alguns problemas é menos suscetível a perda da capacidade de generalização após o

aprendizado de muitos padrões de treino (overfitting) do que a maioria dos algoritmos de aprendizado de máquina. [7]

O AdaBoost chama um classificador fraco repetidamente em iterações  $t = 1, \dots, T$ . Para cada chamada a distribuição de pesos  $D_t$  é atualizada para indicar a importância do exemplo no conjunto de dados usado para classificação. A cada iteração os pesos de cada exemplo classificado incorretamente é aumentado (ou alternativamente, os pesos classificados corretamente são decrementados), para que então o novo classificador trabalhe em mais exemplos. [7]

## 4 ATIVIDADES DESENVOLVIDAS

Durante o período de estágio supervisionado no LAD, no período de abril a julho de 2015, foram realizadas atividades para o desenvolvimento do PLCM (Power Line Communication Modem). Concorrentemente, os estudos de visão computacional foram necessários para a construção da plataforma de captura e processamento de imagens em Full HD.

O demodulador do PLC tem uma unidade de verossimilhança, que por sua vez utiliza uma função que computa o logaritmo da função de Bessel de primeiro tipo de ordem zero. A implementação em Hardware dessa função seria inviável, então uma aproximação foi feita com interpolação. Esta tarefa levou cerca de um mês para ser executada. Uma função em Octave, compatível com Matlab foi criada para modelar a aproximação e o gráfico da função versus sua aproximação que está ilustrada na Figura 5. O erro percentual da aproximação da do logaritmo da função de Bessel em relação ao modelo da função que já está incluso no Matlab foi de 1%, o que implica em um bom resultado na aplicação proposta.

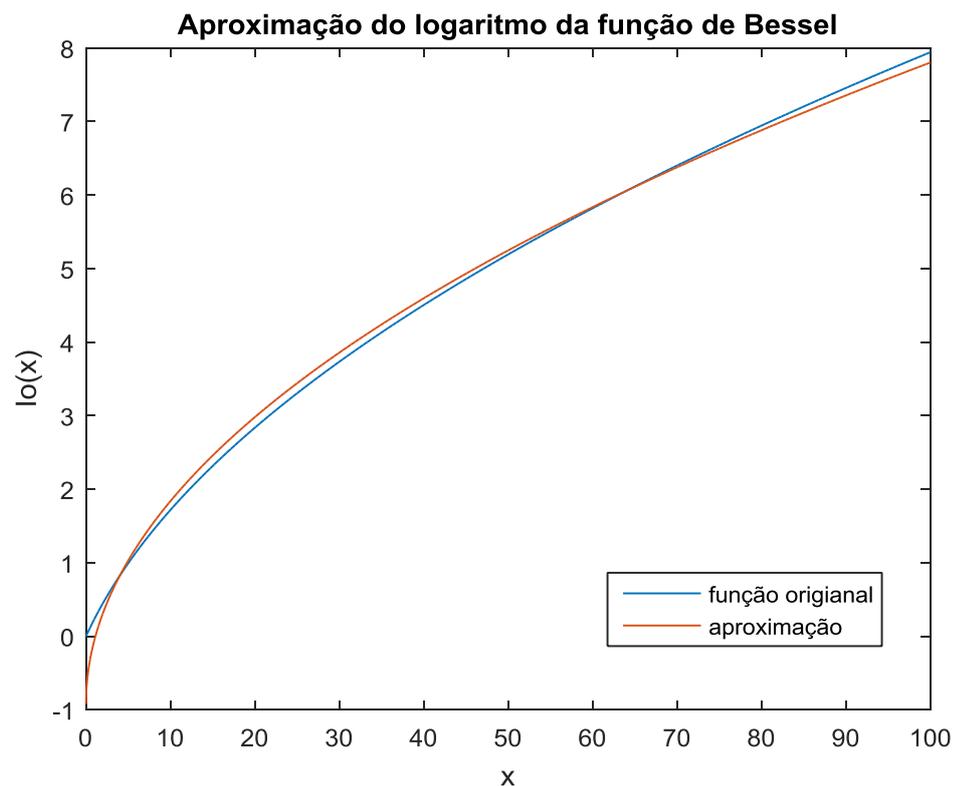


Figura 10 – Aproximação do logaritmo da função de Bessel

A função modificada de Bessel de primeiro tipo de ordem zero [8] é dada pela equação (3):

$$I_0(x) = \frac{1}{2\pi i} \oint e^{\left(\frac{x}{2}\right)\left(\frac{t+1}{t}\right)} t^{-1} dt \quad (3)$$

A vantagem desta função se dá pelo fato de não ser necessário utilizar Lookup Tables (LUT). A operação mais custosa do algoritmo é o cálculo de uma raiz quadrada e o algoritmo da raiz descrito em [9] foi implementado em FPGA. De acordo com este algoritmo, dado um número  $N$  definido conforme a equação (4), calcula-se a raiz quadrada de  $N$  encontrando-se os coeficientes  $a_i$ .

$$N = (a_1 + a_2 + \dots + a_n)^2 = a_1^2 + [2a_1 + a_2]a_2 + \dots + \left[2\left(\sum_{i=1}^{n-1} a_i\right) + a_n\right] a_n \quad (4)$$

A expressão da equação (3) permite a determinação da raiz quadrada de  $N$  sequencialmente, calculando-se o valor de cada  $a_i$  individualmente. Supondo que os valores

$a_1, \dots, a_{m-1}$  já foram determinados, então o  $m$ -ésimo termo da equação (4) é dado pela equação (5):

$$Y_m = [2P_{m-1} + a_m]a_m \quad (5)$$

onde  $P_{m-1} = \sum_{i=0}^{m-1} a_i$  é o valor aproximado da raiz quadrada a cada iteração. Para cada nova iteração, am deve satisfazer a recursão  $X_m = X_{m-1} - Y_m$ , com o valores de  $X_0 = N$  e  $X_n=0$ . [9]

Utilizando palavras de 32 bits (20 bits + 12 bits de precisão), o circuito sintetizado pelo Quartus II teve um total de 679 elementos lógicos (máquina de estados, contadores, somadores, subtratores, registradores de deslocamento e multiplexadores) e 155 registradores. Este trabalho teve a duração de cerca de uma semana e foi desenvolvido em conjunto com outro aluno do laboratório. O circuito da raiz quadrada está ilustrado na Figura 6.

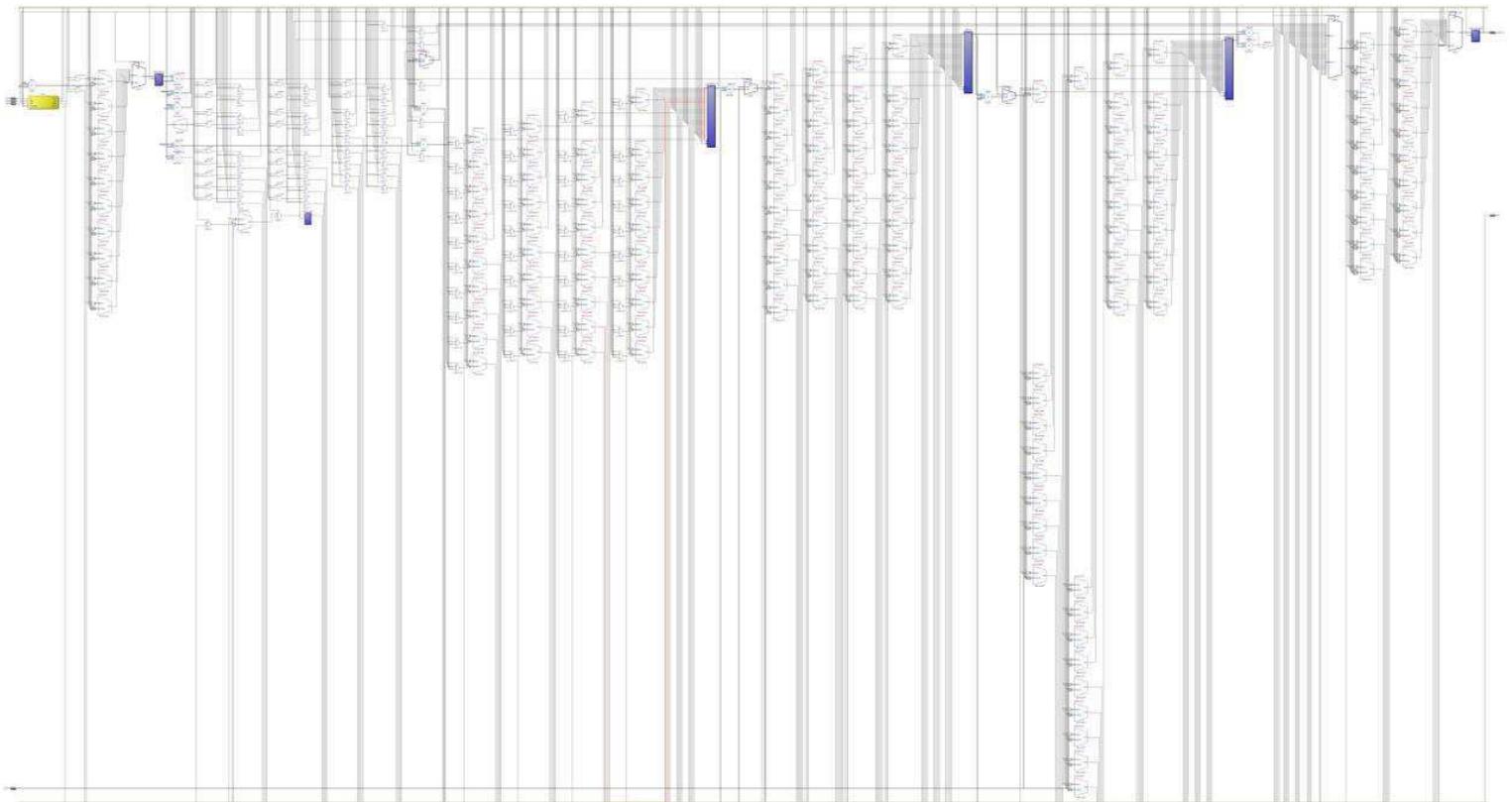


Figura 11 – Síntese do circuito da raiz quadrada

Os conceitos das disciplinas de Circuitos Lógicos, Arquitetura de Sistemas Digitais e Análise de Sinais e Sistemas tiveram fundamental importância no desenvolvimento do módulo da raiz quadrada e também nos outros trabalhos aqui desenvolvidos.

No desenvolvimento de sistemas em chip, do inglês SoC (System on Chip), a construção de módulos complexos se dá pela integração de elementos de complexidade menor, e cada elemento deve garantir um perfeito funcionamento não apenas isolado, mas também em conjunto com os outros elementos do sistema. A verificação funcional de sistemas é uma metodologia que analisa o funcionamento de todas as funções dos módulos, avaliando o comportamento do sistema em função de testes simulados, visando encontrar bugs e garantir a funcionalidade do sistema como um todo.

O algoritmo de Goertzel é uma técnica de Processamento Digital de Sinais que calcula de forma eficiente os termos individuais da Transformada Discreta de Fourier. O cálculo do Goertzel tem a forma de um filtro digital, que opera em uma sequência de dados  $x(n)$ . O filtro do Goertzel tem dois estágios. O primeiro estágio calcula uma sequência intermediária  $s(n) = x(n) + 2 \cos(2\pi f) s(n-1) - s(n-2)$ . O segundo estágio produz uma saída que é a sequência  $y(n) = s(n) - e^{2\pi i f} s(n-1)$ . [11]

Um módulo para calcular a função de Goertzel, que é parte do demodulador do PLC, foi desenvolvido em SystemC, utilizando a biblioteca UVMC (UVM Connect), que interliga SystemC com UVM (Universal Verification Methodology), para construção de modelos de referência (uma função de um modelo que tem sua funcionalidade garantida) em alto nível e verificação funcional de circuitos digitais. Esta foi uma tarefa difícil, levando cerca de dois meses para ser completada. A grande dificuldade estava na criação de um Makefile que ligava as duas bibliotecas.

A biblioteca UVMC interliga módulos escritos em SystemC com módulos escritos em SystemVerilog (que contém as classes de UVM). Um artigo que introduz esta técnica está descrita em [12] e muita informação pode ser encontrada no site <https://verificationacademy.com/>.

Nos trabalhos da parte de visão computacional, além de estudar as técnicas descritas no capítulo 3, foram estudadas as implementações de um Buffer Circular, para a construção das janelas em tempo real no processo de extração de características das faces. Esta etapa foi

realizada em paralelo com a implementação do Goertzel e na ligação do UVM com SystemC e durou cerca de duas semanas.

Um buffer circular é uma estrutura de dados de comprimento fixo que conecta o último elemento (cauda) ao primeiro elemento da estrutura (cabeça). A distância entre a cauda e a cabeça determina o comprimento do buffer. Quando esta distância é nula, diz-se que o buffer está cheio. Quando a distância é igual ao total de elementos do buffer, o buffer está vazio. Essa estrutura é amplamente utilizada em streaming (fluxo de dados). A disciplina de Sistemas em Tempo Real foi cursada em paralelo ao estágio supervisionado, e ela foi de fundamental importância para o estudo e aplicações dos buffers circulares, uma vez que este conceito foi introduzido na disciplina. Na Figura 7 ilustra-se a representação de um Buffer Circular.

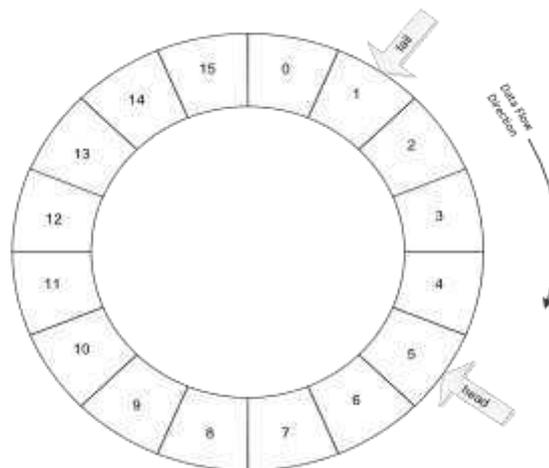


Figura 12 – Representação de um buffer circular (fonte [13])

A construção de um modelo de referência para a plataforma de visão computacional é uma necessidade para um trabalho que está em andamento no LAD, e um ambiente de integração em SystemC com OpenCV foi instalado em ambientes Linux. A integração das bibliotecas tem como base [10]. O último mês do estágio foi concentrado nesta tarefa. Uma dificuldade a ser resolvida é a sua instalação no sistema operacional CentOS. Em Ubuntu a plataforma funcionou sem nenhuma dificuldade. O pacote de instalação fornecido por [10] contém a versão 2.4.2 do OpenCV e a versão 2.3.0 do SystemC.

O filtro de Sobel é uma operação utilizada em processamento de imagem, aplicada sobretudo em algoritmos de detecção de contornos. Em termos técnicos, consiste num operador que calcula diferenças finitas, dando uma aproximação do gradiente da intensidade dos *pixels* da imagem. Em cada ponto da imagem, o resultado da aplicação do filtro Sobel devolve o gradiente ou a norma deste vetor. [14]

Como exemplo default, a biblioteca integrada SystemC/OpenCV carrega um vídeo no formato AVI (Audio Video Interleave) com as função de leitura do OpenCV, aplica um filtro de Sobel escrito em SystemC e reproduz o vídeo filtrado com uma função do OpenCV. Uma tela de captura deste processo está ilustrada na figura 13.



Figura 13 – Integração de SystemC com OpenCV

## 5 CONCLUSÕES

A realização do estágio supervisionado foi uma transição entre a graduação e a atuação profissional como engenheiro. Problemas de situações reais, cujas soluções não são respostas prontas, são comuns e requerem dedicação e planejamento para uma solução do que é proposto.

O fato de o estágio ter sido realizado em um laboratório de pesquisa, com problemas pertinentes ao âmbito acadêmico, são simulações de um cenário industrial, cujo mercado é inexistente no Brasil. Contudo, o período de atuação no estágio foi essencial para a minha formação como engenheiro e as ferramentas adquiridas não apenas neste período, mas durante toda a graduação foram de fundamental importância para o exercício da profissão que está prestes a se iniciar.

## REFERÊNCIAS

- [1] <https://lad.dsc.ufcg.edu.br>, acesso em julho de 2015.
- [2] [https://pt.wikipedia.org/wiki/Brazil\\_IP](https://pt.wikipedia.org/wiki/Brazil_IP), acesso em julho de 2015.
- [3] ASSIS, FELIPE G. Hardware Dedicado para Demodulação S-FSK utilizando Verossimilhança, Trabalho de Conclusão de Curso, UFCG, 2014.
- [4] [https://en.wikipedia.org/wiki/Face\\_detection](https://en.wikipedia.org/wiki/Face_detection), acesso em julho de 2015.
- [5] [https://en.wikipedia.org/wiki/Viola%E2%80%93Jones\\_object\\_detection\\_framework](https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework), acesso em julho de 2015.
- [6] SANTOS, Túlio L., Detector de Faces através do algoritmo de Viola-Jones, UFRJ, 2011.
- [7] <https://pt.wikipedia.org/wiki/AdaBoost> , acesso em julho de 2015.
- [8] <http://mathworld.wolfram.com/ModifiedBesselFunctionoftheFirstKind.html> , acessado em julho de 2015.
- [9] [https://en.wikipedia.org/wiki/Methods\\_of\\_computing\\_square\\_roots#Digit-by-digit\\_calculation](https://en.wikipedia.org/wiki/Methods_of_computing_square_roots#Digit-by-digit_calculation), acesso em julho de 2015.
- [10] SystemC/OpenCV Tutorial  
[http://smarter.ual.edu/images/razorcam/systemc\\_opencv\\_tutorial.pdf](http://smarter.ual.edu/images/razorcam/systemc_opencv_tutorial.pdf), Smarter Lab, University of Arkansas, acesso em junho de 2015.
- [11] [https://en.wikipedia.org/wiki/Goertzel\\_algorithm](https://en.wikipedia.org/wiki/Goertzel_algorithm), acesso em julho de 2015
- [12] [https://www.doulos.com/knowhow/sysverilog/uvm/tutorial\\_0/](https://www.doulos.com/knowhow/sysverilog/uvm/tutorial_0/), acesso em abril de 2015
- [13] <http://www.embedded.com/electronics-blogs/embedded-round-table/4419407/The-ring-buffer>, acesso em maio de 2015
- [14] [https://pt.wikipedia.org/wiki/Filtro\\_Sobel](https://pt.wikipedia.org/wiki/Filtro_Sobel), acesso em julho de 2015