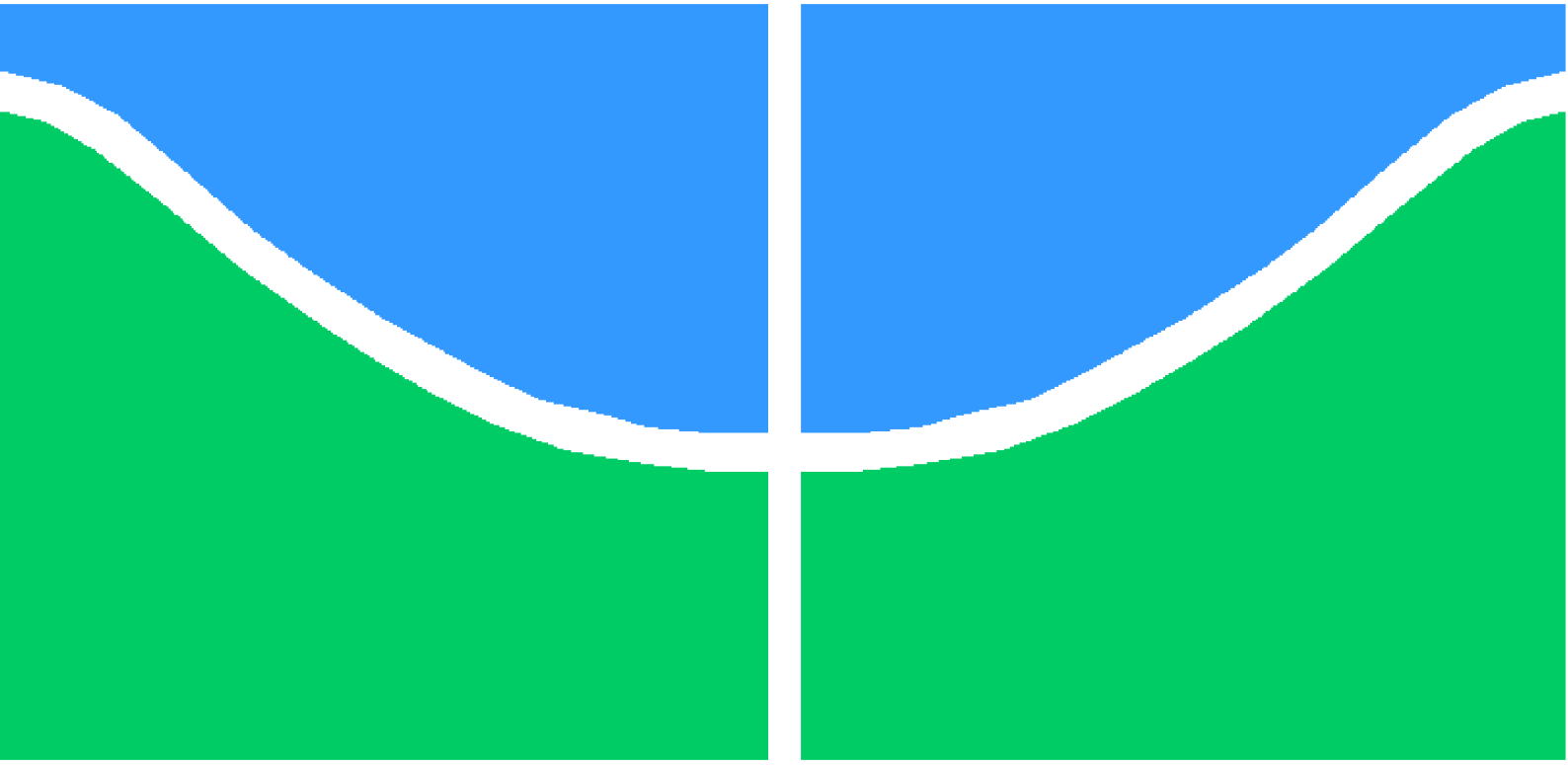

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Engenharia Eletrônica

Estudo Comparativo da Implementação Coprocessada em Sistemas em Chip do Algoritmo de Treinamento do Classificador LDA Aplicado em Interfaces Cérebro-Máquina

Autores: Heleno da Silva Morais e Oziel da Silva Santos

Orientador: Dr. Marcus Vinícius Chaffim Costa

Coorientador: PhD. Daniel Mauricio Muñoz Arboleda

Brasília, DF

2018

Heleno da Silva Morais e Oziel da Silva Santos

Estudo Comparativo da Implementação Coprocessada em Sistemas em Chip do Algoritmo de Treinamento do Classificador LDA Aplicado em Interfaces Cérebro-Máquina

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Dr. Marcus Vinícius Chaffim Costa

Coorientador: PhD. Daniel Mauricio Muñoz Arboleda

Brasília, DF

2018

Heleno da Silva Morais e Oziel da Silva Santos

Estudo Comparativo da Implementação Coprocessada em Sistemas em Chip do Algoritmo de Treinamento do Classificador LDA Aplicado em Interfaces Cérebro-Máquina/ Heleno da Silva Morais e Oziel da Silva Santos. – Brasília, DF, 2018-

57 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Marcus Vinícius Chaffim Costa

Coorientador: PhD. Daniel Mauricio Muñoz Arboleda

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2018.

1. Sistemas em Chip. 2. LDA. I. Dr. Marcus Vinícius Chaffim Costa. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Estudo Comparativo da Implementação Coprocessada em Sistemas em Chip do Algoritmo de Treinamento do Classificador LDA Aplicado em Interfaces Cérebro-Máquina

CDU 02:141:005.6

Errata

Heleno da Silva Morais e Oziel da Silva Santos

Estudo Comparativo da Implementação Coprocessada em Sistemas em Chip do Algoritmo de Treinamento do Classificador LDA Aplicado em Interfaces Cérebro-Máquina

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 06 de Julho de 2018 – Data da aprovação do trabalho:

Dr. Marcus Vinícius Chaffim Costa
Orientador

PhD. Daniel Mauricio Muñoz Arboleda
Coorientador

Dr. Cristiano Jacques Miosso Rodrigues Mendes
Convidado 1

MSc. Renato Coral Sampaio
Convidado 2

Brasília, DF
2018

Heleno da Silva Morais

Ao meu pai, Joaquim Albino de Morais †

Oziel da Silva Santos

Aos meus pais, José Antônio e Gezonita Maurício.

Agradecimentos

Heleno da Silva Morais

Agradeço primeiramente a Deus, por há 6 anos atrás ter escutado o pedido de um jovem sonhador, e que ao longo de toda minha vida esteve ao meu lado, mesmo quando não mereci.

Agradeço à toda minha família, minha mãe e meus irmãos que sempre me motivaram para conquistar meus sonhos. À todos os meus amigos, em especial Ebenezer Andrade e Oziel da Silva, que estiveram presentes nos bons e maus momentos desde o início dessa jornada e a todos que contribuíram de alguma forma durante estes 6 anos para a realização deste sonho pessoal, mas que faz parte do sonho de muitos envolvidos. À minha namorada, que desde o primeiro dia tornou desse meu sonho, nosso sonho, e que me apoiou em todos os momentos.

Aos meus orientadores pela paciência e dedicação por sempre estarem a disposição para a realização deste trabalho.

E por fim, mas não menos importante um agradecimento em especial para minha madrinha e sogra, Maria, que esteve presente nos momentos mais difíceis financeiramente ou emocionalmente. Sem ela não poderia estar aqui.

Oziel da Silva Santos

Aqui tem vários nomes de quem apoiou e de forma direta ou indireta colaborou para esse trabalho, ainda que não houvesse espaço, teria que citar: O nome de Deus que me concedeu a oportunidade de cursar Engenharia Eletrônica nesta universidade. Agradecer aos meus pais pelo apoio e amor sem fim. Aos meus orientadores, professores Marcus Vinícius Chaffim Costa e Daniel Mauricio Muñoz Arboleda pelo grande interesse em ajudar e ensinar. A minha namorada Brunna Siqueira, que sempre esteve ao meu lado. Aos meus amigos: Heleno da Silva, Ebenezer Andrade, Pedro Ivo, Ithallo Junior, Lucas Raposo, Gabriel Henrique, Gabriela Volpato, por suas ideias, conselhos e auxílio. Aos autores que são base teórica para este trabalho, em especial Fabien Lotte. E à esta Universidade.

Resumo

Palavras-chaves: BCI; LDA; FPGA; SoC; Sistemas Embarcados.

Abstract

Key-words: BCI; LDA; FPGA; SoC; Embedded Systems.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

BCI	*Brain Computer Interface* (Interface Cérebro Máquina)
EEG	Eletroencefalografia
SVM	*Support Vector Machine* (Máquina de Vetor de Suporte)
LDA	*Linear Discriminant Analysis* (Análise Discriminante Linear)
SoC	*System on Chip* (Sistema em Chip)
FPGA	*Field Programmable Gate Array* (Matriz de Portas Programáveis por Campo)
SNC	Sistema Nervoso Central
VHDL	*VHISC Hardware Description Language* (VHISC Lingagem de Descrição de Hardware)
VHISC	*Very High Speed Integrated Circuit* (Circuito Integrado de Alta Velocidade)
LUT	*Look Up Table* (Tabela da Verdade)
FF	Flip-Flop
DSP	*Digital Signal Processor* (Processador de Sinal Digital)
RAM	*Random Access Memory* (Memória de Acesso Aleatório)
MUX	Multiplexador
CSP	*Common Spatial Pattern* (Padrão Espacial Comum)
ECoG	Eletrocorticografia
SO	Sistema Operacional
RNA	Rede Neural Artificial
ADC	*Analog Digital Converter* (Conversor Digital Analógico)

Lista de símbolos

μ	Micro
θ	*Theta*
α	*Alpha*
β	*Beta*
γ	*Gamma*

Sumário

1 Introdução

Através de uma rede de mais de 100 bilhões de células nervosas interconectadas, o cérebro realiza o controle de nossas ações, percepções, emoções e etc (KANDEL, 2013). Estas células são chamadas de *neurônios*, e neles são armazenados sinais elétricos, que representam todas as informações de controle (SIULY; LI; ZHANG, 2017). Estes sinais podem ser medidos pela eletroencefalografia (EEG), que é um sistema de medição de sinais elétricos produzidos pelo cérebro durante atividades cerebrais (LOTTE; GUAN, 2011).

De acordo com (SIULY, 2012), a EEG é uma das mais importantes ferramentas para diagnosticar doenças cerebrais. Além do diagnóstico de doenças cerebrais uma outra aplicação para os sinais adquiridos pela EEG são as *Brain Computer Interfaces* (BCIs) (LOTTE; GUAN, 2011). Uma BCI é um sistema que realiza a comunicação entre o cérebro e um computador (SIULY; LI; ZHANG, 2017), onde sua principal função é a tradução dos sinais elétricos cerebrais em comandos de controle para qualquer dispositivo eletrônico (SIULY; LI; ZHANG, 2017).

A BCI realiza a tradução destes sinais através de seis passos: 1) medição dos sinais provenientes de atividades cerebrais, normalmente através da EEG, 2) pré-processamento destes sinais, 3) extração de características, 4) classificação, 5)tradução dos sinais em comandos e 6) realimentação (MASON; BIRCH, 2003). Um dos principais passos para a implementação de uma BCI é a *classificação*, pois é após este passo que é realizada a tradução dos sinais provenientes da EEG em comandos de controle (MASON; BIRCH, 2003).

Em aprendizado de máquina e reconhecimento de padrões, a classificação é caracterizada como um algoritmo que atribui parte de um sinal de entrada a um número de classes ou categorias (BRUNELLI, 2009). Um exemplo é a classificação de um e-mail como spam ou não-spam. Os algoritmos que realizam a classificação dos sinais de entrada são chamados de **classificadores** (SIULY; LI; ZHANG, 2017). De acordo com (LOTTE, 2008, p. 41), "estes classificadores são capazes de aprender como identificar um vetor de características, graças aos processos de treinamento".

O algoritmo que realiza a classificação é caracterizado por uma função matemática que mapeia um sinal de entrada em sua respectiva classe (LOTTE, 2008). Os classificadores preferidos pelos pesquisadores são os **classificadores supervisionados**.Estes classificadores fazem uso de um conjunto de dados para realizar o processo de treinamento do classificador. O conjunto de dados de treinamento é formado por vetores de características previamente atribuídos às suas respectivas classes (LOTTE, 2008). Portanto os

classificadores supervisionados são implementados a partir de dois processos: *treinamento* e *testes* (SIULY; LI; ZHANG, 2017). As *Support Vector Machines* (SVMs), em português, Máquinas de Vetor de Suporte, os *Linear Discriminant Analysis* (LDAs), em português, Análise Discriminante Linear, as Redes Neurais Artificiais (RNAs) e as árvores de decisões são alguns exemplos mais conhecidos de classificadores do tipo supervisionados (SIULY; LI; ZHANG, 2017).

O LDA como dito anteriormente é um dos classificadores supervisionados e tem como principais vantagens a simplicidade e atratividade computacional, por se tratar de um classificador linear (THEODORIDIS; KOUTROUMBAS et al., 1999). O objetivo do LDA é usar uma transformação linear para encontrar um conjunto otmizado de vetores discriminantes e remapear o conjunto de características original, em um outro conjunto de dimensão inferior (SHASHOA et al., 2016).

A simplicidade e robustez do LDA possibilita sua implementação em um sistema embarcado, o que viabiliza sua portabilidade considerando as restrições de desempenho computacional e consumo energético dos Sistemas em Chip (SoCs - do inglês *System on Chip*.

Um SoC é caracterizado pela implementação de todo um sistema computacional, composto por: memórias, processadores, entradas e saídas, conversores de dados, controladores de periféricos, entre outros, em um único chip de silício (CROCKETT et al., 2014). Diferente dos computadores tradicionais, que possuem seu sistema implementado a partir de módulos isolados e combinados em uma placa de circuito impresso, ou placa-mãe, os SoCs possuem como principais características um baixo custo de implementação, além de baixo consumo de potência, menor tamanho físico, maior confiabilidade e, dependendo dos recursos disponíveis, maior desempenho computacional, se comparado com um computador tradicional (CROCKETT et al., 2014). Um exemplo de um SoC é a plataforma *Zynq* que combina em um único chip processadores *Advanced Risc Machine* (ARM) e *Fieald Programmable Gate Arrays* (FPGA), conversores analógicos digitais (ADC - do inglês) (CROCKETT et al., 2014).

Tendo em vista a grande vantagem dos SoCs sobre os computadores tradicionais, onde são implementados e executados os algoritmos de classificação, este trabalho apresenta um estudo da viabilidade da implementação em hardware e em software embarcado, do algoritmo de treinamento do classificador LDA desenvolvido previamente por (LOTTE; GUAN, 2011) na plataforma *Matlab*, realizando a comparação de consumo computacional (hardware), desempenho computacional (tempo de execução) e consumo energético entre as implementações em hardware e software. Em particular a implementação em hardware consiste no mapeamento do algoritmo na FPGA da plataforma *Zynq*, afim de paralelizar seus processos, enquanto que a implementação em software consiste em executar o algoritmo em um sistema embarcado utilizando os cores ARM, também da plataforma *Zynq*,

além da comparação com sua implementação inicial em *Matlab*.

1.1 Justificativa

As aplicações das BCIs apresentam um crescente desenvolvimento graças ao aumento do interesse em pesquisas voltadas para o tema (BLANKERTZ et al., 2006) . As BCIs utilizam-se de algoritmos de classificação para realizar o processo de tradução dos sinais cerebrais em comandos de controle (MASON; BIRCH, 2003). O LDA é um tipo de classificador utilizado para realizar tal processo. Um dos principais processos para implementação de um classificador é o processo de treinamento, é com este processo que o classificador é capaz de classificar corretamente os sinais de entrada, quando realizado um bom processo de treinamento(LOTTE; GUAN, 2011). Por se tratar de um algoritmo linear e consequentemente não exigir um grande esforço computacional, torna-se viável sua implementação em um sistema embarcado. Como os SoCs apresentam características de baixo consumo de energia e tamanho físico pequeno, a implementação de algoritmos de classificação em sistema deste porte podem tornar as BCIs mais acessíveis, tendo em vista que um algoritmo de treinamento embarcado em um SoC reduzirá a necessidade de um sistema computacional tradicional, além de um melhor processamento computacional, pois sua implementação em plataformas FPGAs possibilitam a paralelização de seus processos (CROCKETT et al., 2014).

1.2 Objetivos

1.2.1 Objetivos Gerais

O objetivo geral do presente trabalho é implementar o algoritmo de treinamento de um classificador LDA em um sistema embarcado coprocessado (hardware/software), utilizando um SoC na plataforma *Zynq* da *Xilinx*, no intuito de estudar os ganhos em relação ao tempo de execução e consumo de energia se comparado com implementações tradicionais.

1.2.2 Objetivos Específicos

- Reproduzir os resultados obtidos por (LOTTE; GUAN, 2011) no desenvolvimento do algoritmo de treinamento do classificador LDA;
- Implementar em sistema embarcado o algoritmo de treinamento utilizando os cores ARM da *Zynq*;
- Realizar uma análise de *profile* para determinação de funções que exigem um maior esforço computacional;

- Mapear a(s) função(ões) que exigem um maior esforço computacional em hardware FPGA utilizando a linguagem VHDL;
- Validar as implementações utilizando as bases de dados do *BCI Competition III*, em especifico o conjunto de dados BCI III *dataset IVa*.
- Realizar uma análise estatística do erro associado a ambas implementações, comparadas com a implementação na plataforma *Matlab*.
- Realizar uma análise de ganhos ou perdas de processamento computacional, comparando a implementação em software com o coprocessamento;

2 Referencial Teórico

Este capítulo apresenta os conceitos teóricos que abordam este trabalho, detalhando a estrutura cerebral responsável pela geração dos sinais de interesse na subseção 2.1, o sistema de captura dos sinais (a EEG) na subseção 2.2, o sistema que se refere às BCIs que realizam a tradução dos sinais em comandos para dispositivos na subseção 2.3, os detalhes técnicos a respeito dos sinais oferecidos pela base de dados *BCI Competition* na subseção 2.4, a estrutura geral de um classificador LDA na subseção 2.5, os conceitos de um SoC na subseção 2.6 e por fim o estado da arte das implementações de algoritmos de classificação na subseção 2.7.

2.1 O Cérebro

O Sistema Nervoso Central (SNC) é o responsável direto pelo comando do nosso comportamento geral (CLARK NASHAAT BOUTROS, 2005). Ele pode ser dividido em três principais áreas: tronco encefálico ou medula espinhal, o cérebro e o cerebelo (Figura 1) (ALVAREZ; LEMOS, 2006).

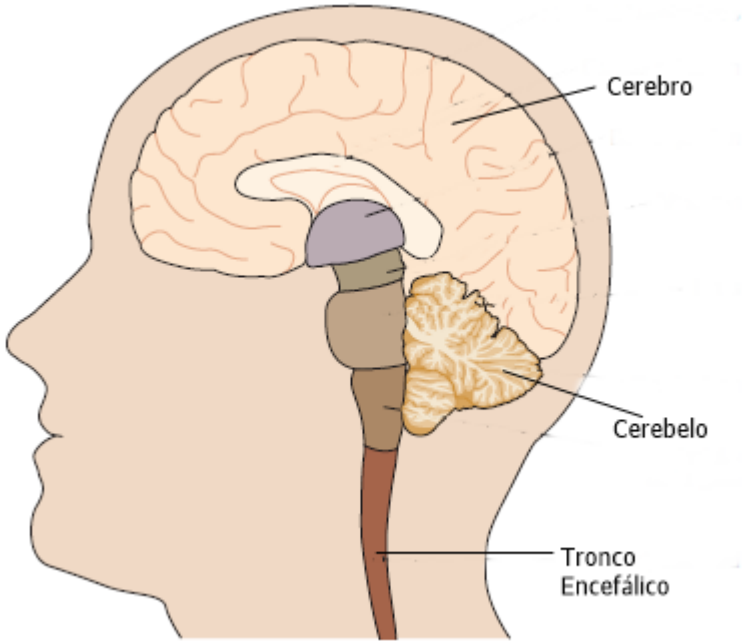

Figura 1 – Três principais áreas do sistema nervoso central. Fonte: (KANDEL, 2013)

O tronco encefálico, parte caldal do SNC, recebe e processa todos os sinais dos sensores corporais, além de realizar o controle dos membros e do tronco humano (KANDEL, 2013).

O cérebro é o processador do SNC, nele são recebidos e processados os sinais da medula espinhal, além de fornecer todos os sinais de controle para a própria medula, que por sua vez, distribui os sinais para os membros e para o tronco (KANDEL, 2013).

O cerebelo está localizado logo atrás do tronco encefálico, onde, através de fibras chamadas de pedúnculos, realizam a comunicação entre sí (KANDEL, 2013). O cerebelo é a segunda maior estrutura do SNC contendo mais da metade dos neurônios cerebrais (SIULY, 2012). É ele o responsável nossa percepção (ALVAREZ; LEMOS, 2006), além do controle da região motora e a memória de movimentos (SIULY, 2012; ALVAREZ; LEMOS, 2006)

Os sinais de controle e de sensoriamento são sinais elétricos armazenados nos neurônios (KANDEL, 2013). Suas características eletroquímicas permitem que os mesmos armazenem e transmitam sinais elétricos para qualquer outra célula receptora mesmo que a longa distâncias (SIULY, 2012). Sua estrutura pode ser dividida em três principais partes: 1) Corpo da celula, 2) axônio e 3) dendrito, conforme apresentado na Figura(2).

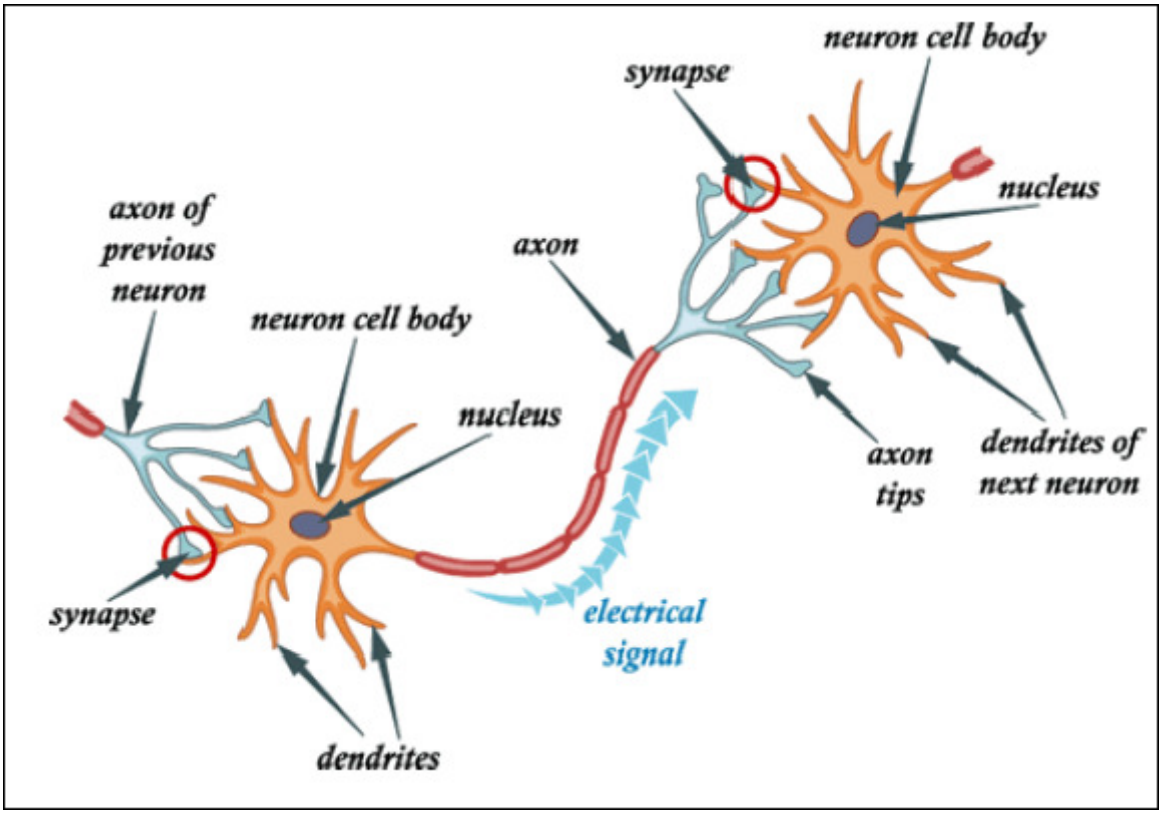

Figura 2 – Estrutura de um neurônio. Fonte: (SIULY, 2012)

A ativação dos neurônios é dada por meio de um gradiente de concentração eletroquímico, resultando na produção de uma corrente elétrica cerebral que flui através do axônio, o que torna possível a comunicação com outras celulas (SIULY, 2012). A corrente elétrica cerebral consiste comumente de íons de Na+, K+, Ca++ e Cl- (TEPLAN et al., 2002).

As atividades elétricas cerebrais podem ser divididas em dois conjuntos: os potenciais de ação (AP) e os potenciais de sinapse (SP) (SIULY, 2012).

Sempre que um potencial de sinapse atinge o limite de condução, ou seja, se o potencial é carregado (estimulado por alguma atividade cerebral) até o ponto em que se é gerada uma corrente elétrica no axônio, um AP é iniciado (SIULY, 2012).

Como dito antes, o cerebelo é o responsável pelo controle da percepção e dos movimentos, além da memória de movimento, portanto a percepção, os movimentos e a imagética motora são os estímulos que iniciam uma AP na região do cerebelo (ALVAREZ; LEMOS, 2006).

Os potenciais elétricos gerados a partir dos potenciais de sinapse são armazenados na escalpe (ou couro cabeludo), onde durante uma atividade cerebral é criado um dipolo elétrico entre os dentritos e a soma (região ao redor do núcleo) (SIULY, 2012).

2.2 Eletroencefalografia

A Eletroencefalografia (EEG) é um sistema de aquisição de potenciais elétricos, que refletem atividades cerebrais humanas (SIULY; LI; ZHANG, 2017). É muito usado por profissionais de saúde e cientistas para avaliar e estudar funções cerebrais e diagnosticar distúrbios neurológicos (SIULY; LI; ZHANG, 2017).

A técnica popular que registra sinais do cérebro usando eletrodos estrategicamente colocados sobre o couro cabeludo do paciente, portanto não invasiva, é muito importante nos diagnósticos de doenças neurológicas (RAO, 2013). Em um exame de EEG, dependendo da sua aplicação, podem ser usados de 1 a 256 eletrodos a fim de captar paralelamente os sinais (RAO, 2013). Cada par de eletrodos formam um canal, portanto com 256 eletrodos tem-se uma leitura de 128 canais, também conhecido como EEG multi canais (RAO, 2013). Cada canal recebe um amplificador de instrumentação e um equipamento de gravação do EEG (RAO, 2013).

Devido às diferentes camadas e tecidos interpostos entre a fonte do sinal (que é a atividade neural do córtex) e os sensores colocados no couro cabeludo, tudo isso atuando como um filtro passa-baixa, faz com o que a resolução espacial do EEG seja ruim, em contra partida tem boa resolução temporal na faixa de milisegundos (RAO, 2013).

Em um adulto normal, o sinal típico de EEG tem sua amplitude variando de 1 a 100 microvolts, se for medido usando eletrodos de tipo agulha o seu módulo pode variar de 10 a 20 milivolts. Considerando a não uniformidade do cérebro humano e a organização funcional do córtex, o sinal de EEG pode variar bastante de acordo com a disposição dos eletrodos (SIULY, 2012).

Com a baixa amplitude desse tipo de sinal, ele pode sofrer facilmente atenuações,

contaminações em seu espectro, como por exemplo interferência da rede elétrica de 60 Hz e seus harmônicos associados, e principalmente atividades musculares exercidas no ato da extração dos sinais. Portanto no momento do exame os pacientes são orientados a não realizar nenhum tipo de movimento (RAO, 2013).

O sistema internacional 10-20 é utilizado para especificar um padrão de alocação dos eletrodos na cabeça dos pacientes ou voluntários. Esse padrão determina as distâncias entre os eletrodos (SIULY; LI; ZHANG, 2017). São posicionados eletrodos primeiramente em pontos estratégicos, os pontos são nomeados por: *mastoids*, eletrodos de referência localizados atŕas de cada orelha (A1 e A2), *Nasion*, referência no segmento do topo do nariz, porém nivelado com os olhos e por fim, o *onion*, referência situada na base do crânio no ponto médio da parte de trás da cabeça, as medidas são feitas a partir desses pontos, no sentido transversal e médio, com intervalos de 10 e 20 % (RAO, 2013), assim representado na Figura 3.

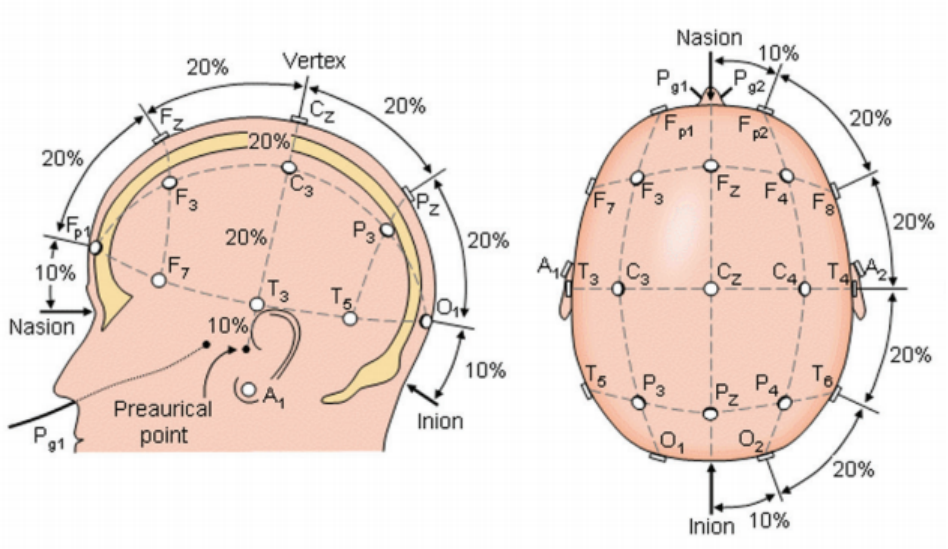

Figura 3 – Sistema Internacional de Posicionamento 10-20. Fonte: (CAMPISI; ROCCA; SCARANO, 2012).

A Frequência do sinal é um dos parâmetros mais importantes na avaliação clínica de anomalias em EEGs, e também para compreender o comportamento operacional na pesquisa cognitiva (SIULY, 2012). Com inúmeras oscilações não periódicas, milhares de comunicação inter-neurônios e comportamentos estocásticos o EEG humano é comumente ordenado em faixas de frequências específicas. Tais bandas são divididas em: delta (0.5-4 Hz) , theta(θ)(4-8Hz), alpha(α)(8-13Hz), beta(β)(13-30Hz) e acima de 30Hz gama(γ), conforme Figura 4.

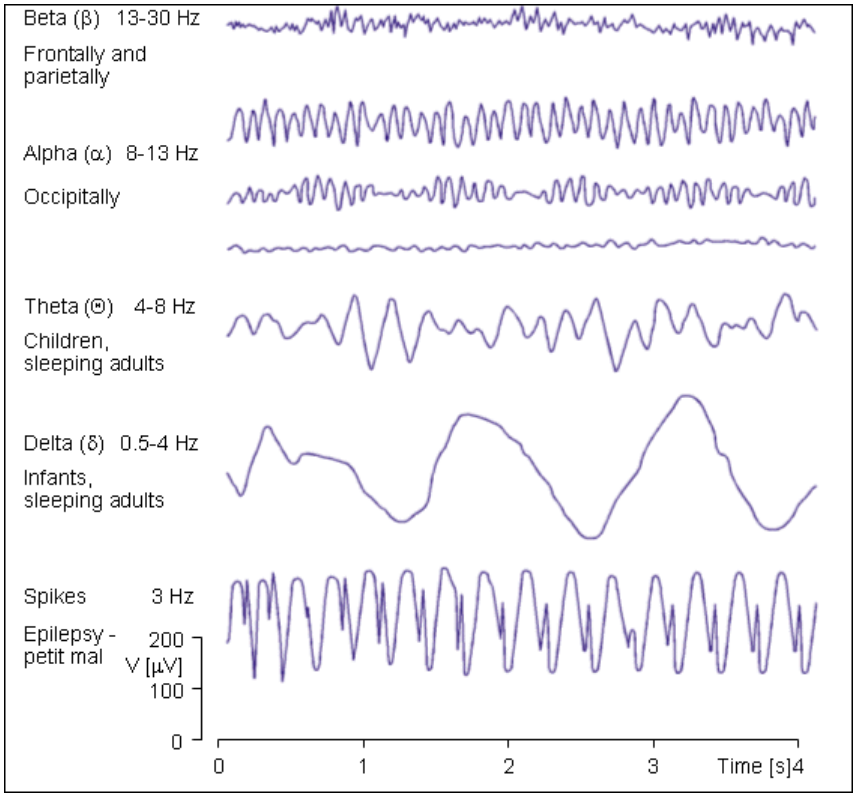

Figura 4 – Exemplos de diferentes tipos de EEG. Fonte: (CAMPISI; ROCCA; SCARANO, 2012)

.

2.3 *Brain Computer Interface*

Uma das aplicações dos sinais adquiridos pela EEG são as *Brain Computer Interface* (em português, Interface cérebro máquina) (LOTTE; GUAN, 2011; SIULY, 2012). Uma Interface Cérebro Máquina (BCI, do inglês *Brain Computer Interface*) é um sistema que analisa sinais cerebrais e os convertem para um novo sinal de saída (WOLPAW, 2012), ou seja, uma BCI é um sistema que realiza a comunicação entre o cérebro e o mundo externo sem a interação neuromuscular (WOLPAW, 2012).

O termo BCI foi utilizado pela primeira vez por Jacques Vidal em 1970, onde foram utilizadas técnicas invasivas para aquisição dos sinais cerebrais, sistema de eletrocorticografia (ECoG) (WOLPAW, 2012), sistema em que os sensores de aquisição eram instalados na região do córtex. Em 1980 o mesmo Jacques Vidal publicou pela primeira vez a utilização de uma BCI não invasiva utilizando o sistema da EEG (GUGER et al., 2013), as duas técnicas são apresentadas na Figura 5.

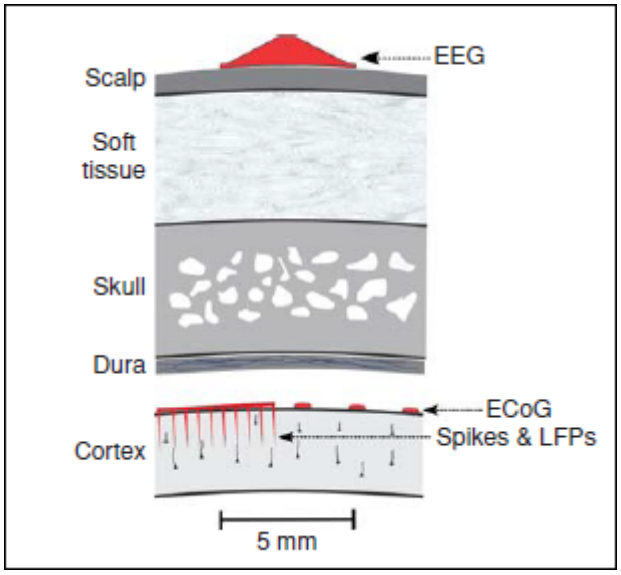

Figura 5 – Diferença entre os sistemas ECoG e EEG. Fonte: (WOLPAW, 2012).

Atualmente é mais comum a utilização do sistema da EEG para aquisição de sinais cerebrais por se tratar de um sistema não invasivo (GUGER et al., 2013). A BCI realiza a tradução dos sinais cerebrais por meio de seis passos, de acordo com o fluxograma apresentado na Figura 6.

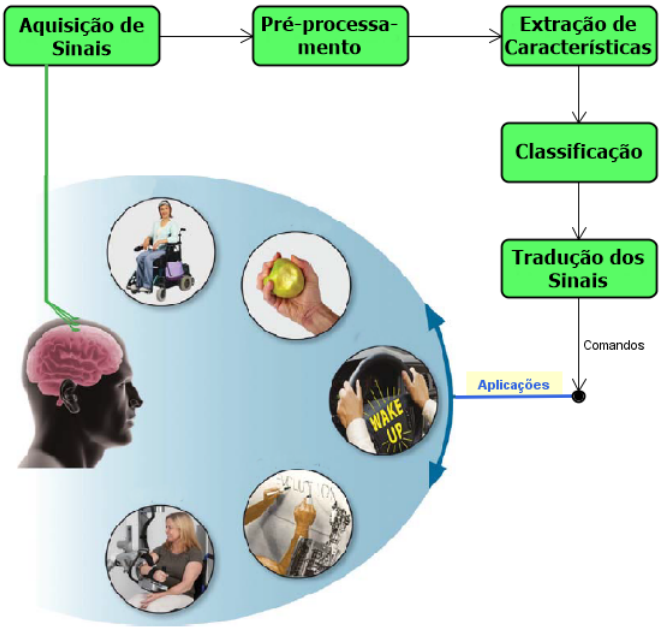

Figura 6 – Fluxograma de processos de uma BCI, que pode ser aplicado em mecanismos motorizados, reabilitação muscular, alarmes, entre outros. Adaptado de (WOLPAW, 2012).

1. **Aquisição de sinais**: Consiste na primeira etapa de um BCI, na qual são aferidos potenciais elétricos provenientes de atividades cerebrais (SIULY, 2012). Os sinais

utilizados neste trabalho foram adquiridos utilizando a EEG e provenientes de uma base de dados de acesso livre, discutida adiante, na seção 2.4.

2. **Pré-Processamento**: Nesta etapa são utilizados filtros para atenuar a presença de ruídos no sinal e amplificar as informações relevantes dentro do sinal (SIULY, 2012).

3. **Extração de Características**: Nesta etapa são extraídos dos sinais alguns parâmetros relevantes, chamados de características (SIULY, 2012).

4. **Classificação**: Nesta etapa as características extraídas no processo anterior são rotuladas em determinadas classes (SIULY, 2012).

5. **Tradução em comandos**: Nesta etapa um comando é associado a cada uma das respectivas classes (SIULY, 2012).

6. **Realimentação**: Por fim é fornecido ao usuário da BCI uma informação a respeito do seu estado mental, qual atividade cerebral foi detectada (SIULY, 2012).

Um dos principais passos para a implementação de uma BCI é a **classificação**, pois é após este passo que é realizada a tradução dos sinais provenientes da EEG em comandos de controle (MASON; BIRCH, 2003). Para isso são utilizados algoritmos classificadores, dentre os quais um tipo de algoritmo de classificação é o LDA, a ser abordado na seção 2.5.

São utilizadas estratégias mentais para definir ao usuário tarefas mentais afim de gerar características padronizadas nos sinais cerebrais de acordo com determinada tarefa, para que o classificador possa interpretá-las corretamente (SIULY, 2012). Uma das abordagens mais comuns é imagética motora, que caracteriza a imaginação do movimento de um membro do corpo humano (SIULY, 2012).

2.4 *BCI Competition*

A *BCI Competition* é uma competição que promove o desenvolvimento e melhoria da tecnologia voltada para as BCIs, sendo submetidas diferentes técnicas de análise de dados cerebrais (BLANKERTZ et al., 2006). Já foram realizadas quatro edições da competição, nos anos de 2001, 2002, 2004 e 2008 (BLANKERTZ et al., 2006). Em cada uma destas competições são fornecidos publicamente sinais cerebrais, adquiridos em laboratórios especializados (BLANKERTZ et al., 2006). Estes sinais são divididos em dois conjuntos de dados, os dados de treinamento e os dados de teste, que são utilizados para treinamento e teste dos algoritmos dos participantes (BLANKERTZ et al., 2006).

2.4.1 *BCI Competition III*

O objetivo do *BCI Competition III* é validar as metodologias de classificação e processamento de sinais cerebrais aplicados em BCIs desenvolvidas pelos participantes da competição (BLANKERTZ et al., 2005). Esta edição foi realizada entre Maio e Junho de 2004, período no qual foram disponilizados 8 *datasets* (I, II, IIIa, IIIb, IVa, IVb, IVc e V), desenvolvidos com a participação de 49 laboratórios especializados (BLANKERTZ et al., 2006). Para cada um dos *datasets* foram realizadas diferentes tarefas que estimulam atividades cerebrais durante a aquisição dos sinais, configurando assim um objetivo especifico para cada um dos *datasets* (BLANKERTZ et al., 2005).

2.4.2 *BCI Competition III - Dataset IVa*

O *dataset IVa* refere-se a um conjunto de dados adquiridos por meio da EEG, em que os sujeitos (indivíduos nos quais foram capturados os sinais) foram submetidos a estimular o cérebro por imagética motora, através de indicações visuais (BLANKERTZ et al., 2006). Os indivíduos foram submetidos a realizarem três tarefas, indicadas visualmente por 3.5s cada tarefa, sendo interrompidas em períodos aleatórios entre 1.75s e 2.25s, onde o sujeito era submetido a um período de relaxamento (BLANKERTZ et al., 2006). As três tarefas de imagéticas motoras foram: (L) mão esquerda, (R) mão direita e (F) pé direito (BLANKERTZ et al., 2006).

Os sinais foram adquiridos de 5 sujeitos rotulados como *aa, al, av, aw* e *ay*, onde foram executadas no total 280 tarefas por cada sujeito. Algumas destas tarefas foram previamente classificadas, ou seja, foram mapeadas características do sinal cerebral em sua respectiva classe, formando o conjunto de dados de treinamento, enquanto que as tarefas não classificadas formam o conjunto de dados de teste (BLANKERTZ et al., 2005). Estes sinais foram adquiridos, pré-processados e disponibilizados por *Fraunhofer FIRST, Intelligent Data Analysis Group (Head: Klaus-Robert Müller), and Charité University Medicine Berlin, Campus Benjamin Franklin, Department of Neurology, Neurophysics Group* (BLANKERTZ et al., 2006). A Tabela 1 apresenta a quantidade de tarefas previamente classificadas (nomeados #tr) e a quantidade de tarefas não classificadas (nomeadas #te) para cada sujeito.

Tabela 1 – Número de tarefas classificadas e não classificadas por sujeito (BLANKERTZ et al., 2006).

Sujeitos	**#tr**	**#te**
aa	168	112
al	224	56
av	84	196
aw	56	224
ay	28	252

2.5 *Linear Discriminant Analisys*

O LDA é um classificador desenvolvido para explorar as informações no reconhecimento de padrões supervisionados, as informações conhecidas são contidas num vetor de treinamento previamente disponibilizado (IZENMAN, 2008). No algoritmo do LDA as informações de maior relevância são descobertas, enquanto que as de menor são eliminadas. O critério usado pelo algoritmo é obter as dimensões que possuem as características mais distintas das classes padrão (KORKMAZ et al., 2017).

Para um melhor entendimento, supõe-se a existência de um conjunto de dados $\vec{x}$, em que $\vec{x} = (x_1, x_2, ..., x_n)^T$ e T indica a transposição, com características multivariadas, e que cada dado seja conhecido devido ser proveniente de uma das classes y, tal que, são predefinidas com características semelhantes aos dados. As classes podem ser exemplificadas como sendo: espécies de plantas, presença ou ausência de uma condição médica específica, diferentes tipos de tumores, tipos de veículos automotores entre outros. Para separar as classes conhecidas uma das outras, é atribuído um rótulo a cada classe, então os dados são representados como dados rotulados (IZENMAN, 2008).

Devido a indispensabilidade de diminuir as dimensões dos dados de um determinado conjunto, o objetivo do LDA é reduzir a dimensão do espaço de conjunto de dados, resolvendo o inconveniente da sobreposição (SINGH; PRAKASH; CHANDRASEKARAN, 2016).

O LDA tem a proposta de encontrar uma transformação ótima para maximizar a proporção de acordo com a Equação 2.1. com isso encontrando o vetor W que proporciona a melhor descriminação (KETSUWAN; PADUNGWEANG, 2017).

$$J(W) = \frac{W^T S_B W}{W^T S_W W} \tag{2.1}$$

S_B é a matriz de dispersão entre as classes e S_W a matriz de dispersão dentro das classes.

S_B é caracterizado pela matriz de covariância entre as classes, obtido sengundo a Equação 2.2

$$S_B = \sum_{j=1}^{c} n_j (m_j - \bar{m})(m_j - \bar{m})^T \tag{2.2}$$

c é o número de classes, e $\bar{m}$ o vetor média e m_j é o vetor dos dados pertinentes à classe j e x é o vetor de treinamento.

$$\bar{m} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.3}$$

$$m = \frac{1}{n_j} \sum_{x \in cj} x \tag{2.4}$$

A matriz de dispersão entre classes é definida pela equaçao 2.5,

$$S_B = \sum_{j=1}^{c} \sum_{x \in Cj} (X - m_j)(X - m_j)^T \tag{2.5}$$

Para S_W sendo uma matriz não-singular, a solução da equação 2.1, pode ser escrita como:

$$S_W^{-1} S_B W = \lambda W \tag{2.6}$$

onde λ é o auto-valor correspondente ao auto-vetor W.

Ou seja, a proposta do classificador LDA é encontrar o hiperplano de maior separação entre as classes analisadas. O cálculo deste hiperplano é feito a partir dos dados de treinamento. Com os dados de treinamento, a distribuição de probabilidade é conhecida, podendo ser representada pela média da dispersão dos sinais. Em outras palavras os hiperplanos são obtidos através da correlação inter-classes e intra-classe, maximizando a separabilidade das classes e minimizando a variabilidade dentro da classe.

2.6 *System-on-Chip*

O termo Sistema em Chip (SoC, do inglês *System-on-Chip*), implica que todo sistema que contém funcionalidades implementadas em hardware e software se encontra em um único chip de silício, combinando processamento, lógica de alta velocidade, interface, memória entre outros componentes ao invés de uma implementação maior em vários chips físicos diferentes agrupados em uma placa-mãe (CROCKETT et al., 2014).

São vários os argumentos a favor da escolha de um SoC a uma arquitetura que utiliza placa-mãe, pode-se citar que a solução é de menor custo, viabiliza transferências de dados mais rápidas e seguras entre vários elementos do sistema, possui maior velocidade geral do sistema, menor consumo de energia entre vários outros elementos que fortalecem a escolha de um SoC em sistemas discretos com componentes equivalentes (CROCKETT et al., 2014).

2.6.1 Aquitetura Simplificada de um SoC

O conjunto da arquitetura pode se dividir em dois sistemas, sistema de hardware e sistema de software.

No **Sistema de Hardware** encontram-se todos os periféricos, memórias e processadores, para conecta-los existe um barramento de comunicação responsável por isso. Já

no **Sistema de Software**, o software funciona sobre o processador, que também sustenta os aplicativos (geralmente em um Sistema Operacional) e também possui uma camada num nível mais baixo de software que faz a interface com o sistema de hardware. Um diagrama de blocos simplificado de um SoC é apresentado na Figura 7 (CROCKETT et al., 2014).

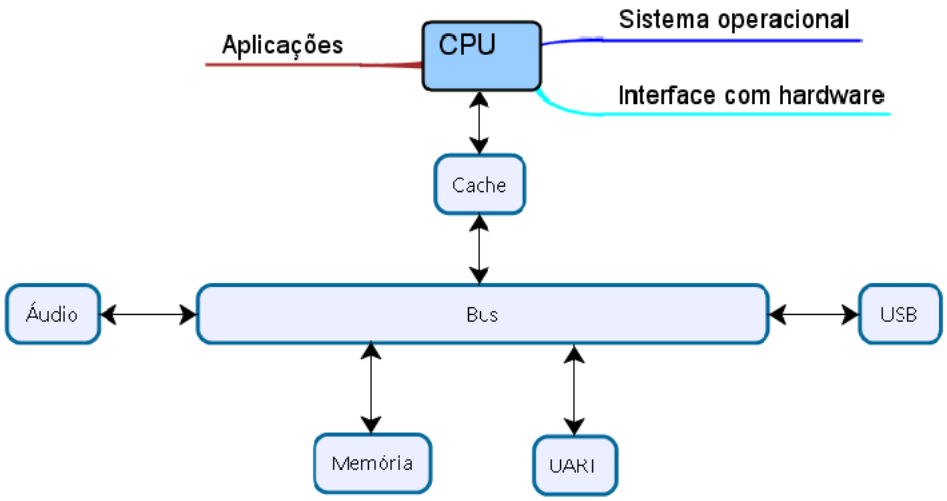

Figura 7 – Diagrama simplificado da relacão entre hardware e software em um SoC. Adaptado de (CAO et al., 2017)

2.6.2 *Zybo-Board*

A Zybo é uma plataforma de desenvolvimento (Figura8). Baseada na arquitetura Xilinx® *All-Programmable System-on-Chip* (AP SoC) do tipo Z-7010, o qual possui em seu encapsulamento um processador ARM Cortex-A9 de dois núcleos e um FPGA da família Xilinx 7-Series. Esse chip combinado com memórias, entradas e saídas de áudio e vídeo, USB, Ethernet, controlador de memória SD entre outros periféricos, proporciona um *kit* para desenvolvedores que procuram uma plataforma plataforma de desenvolvimento SoC com flexibilidade e boa capacidade de processamento (DIGILENT, 2017).

A Figura 9 mostra o diagrama funcional do Zynq-7000 AP SoC, o *Processing System e a Programmable Logic* têm sistemas de alimentação individual, permitindo o usuário habilita-los ou não para gerenciamento de energia.

O Zynq-7000 AP SoC é composto pelos seguintes blocos majoritários

- Sistema de Processamento (PS do inglês *Processing System*)
  - Unidade de Processador de Aplicação (APU - do inglês *Application Processor Unit*)
  - Interfaces de Memória(do inglês *Memory interfaces*)

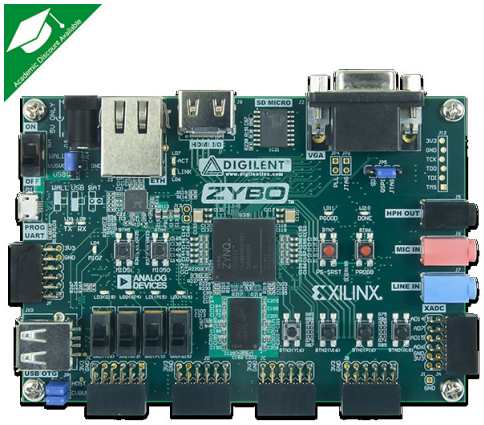

Figura 8 – Placa de Desenvolvimento Zybo-Board. Fonte: (DIGILENT, 2017)

  - Periféricos de entrada (IOP do inglês *I/O peripherals*)
  - Iterconexões (do inglês *Interconnect*)

- Lógica Programável (PL - do inglês *Programmable Logic*)
  - *Look-Up Tables* - LUT São tabelas da verdade capazes de implemtarem quaisquer funções lógicas combinacionais.
  - Flip-Flop Unidade de armazenamento que quando combinado com a LUT é utilizado para implementar circuitos sequênciais.
  - BRAM Bloco de memoria de acesso randômico utilizado para armazenar dados.
  - DSP Processador digital de sinais, basicamente é uma unidade lógica aritimética.

2.6.2.1 Comunicação entre PS e PL

Conforme apresentado na figura 9 a PL realiza a comunicação com o PS através da interface de comunicação AXI-32bits. O AXI é uma das partes da família de protocolos *ARM-AMBA*. Em 2010 a ultima versão desenvolvida do AMBA (versão 4.0), foi incluida a segunda versão do AXI, o AXI4 (XILINX, 2011).

Existem três tipos de barramento AXI, o AXI4, AXI4-Lite e AXI4-Stream.

- AXI4 - usado quando os requisitos de projeto necessitam de alta performance no mapeamento de memória.
- AXI4-Lite usado para um simples mapeamento de memória.
- AXI4-Stream usado quando necessita-se fluxo de dados em alta velocidade.

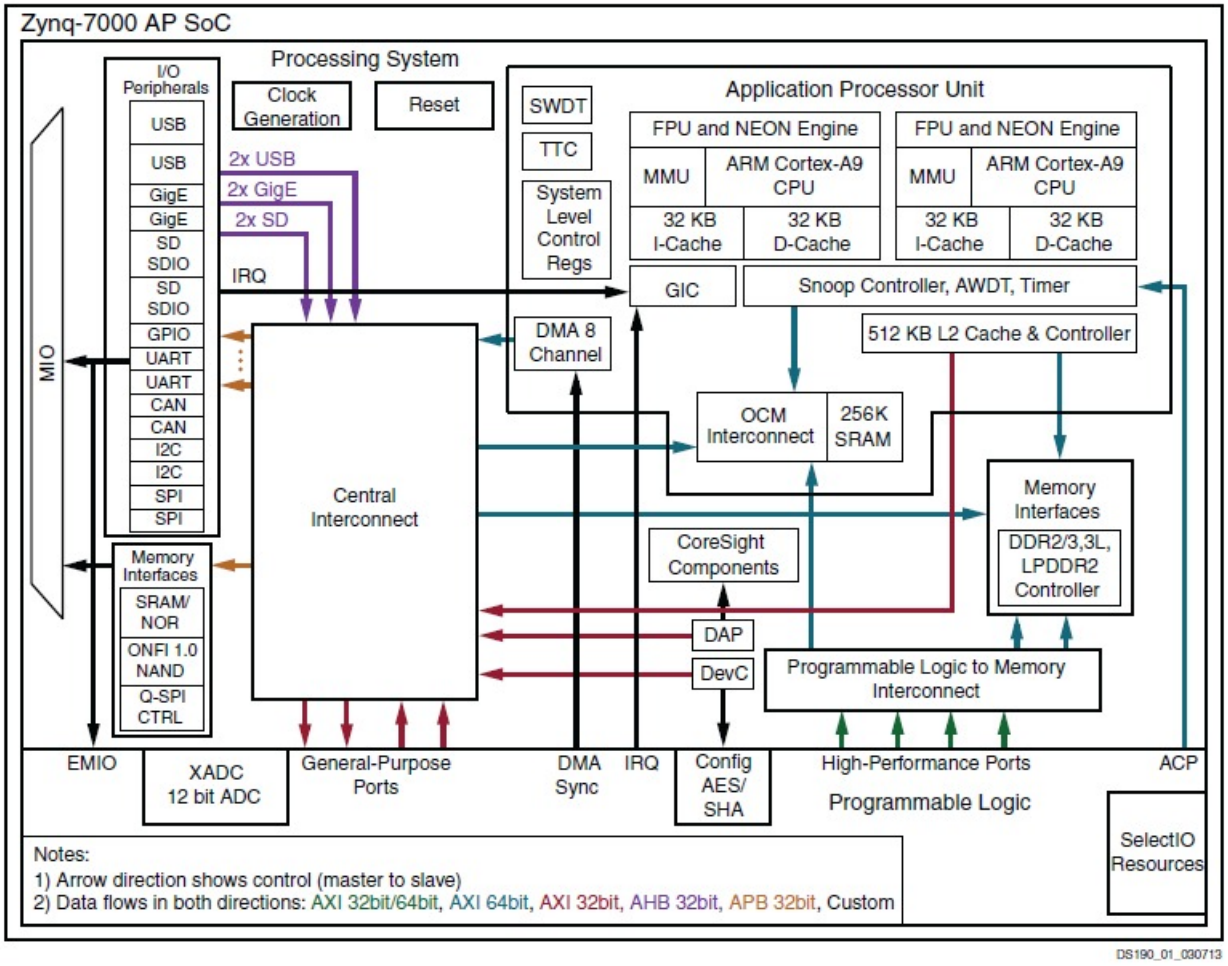

Figura 9 – Diagrama funcional do Zynq-7000 AP SoC

2.7 Estado da Arte

A implementação de algoritmos de classificação em FPGA já foi relatada anteriormente na literatura, sendo utilizados para diferentes aplicações. A Tabela 2 apresenta a relação de alguns autores que realizaram a implementação de classificadores em SoCs utilizando FPGA, apresentando a acurácia obtida por cada autor, bem como os sinais de entrada e sua respectiva aplicação.

Os melhores resultados obtidos com a base de dados *BCI Competition - Dataset IVa*foram de (WANG et al., 2004), onde utilizando do classificador LDA, conseguiu acurácias de 94.17% geral, 95.5% sobre o sujeito (aa), 100.0% sobre o sujeito (al), 80.6% sobre o sujeito (av) 100.0% sobre o sujeito (aw) 97.6% sobre o sujeito (ay).

A Tabela 3 apresenta a acurácia obtida por (LOTTE; GUAN, 2011) no ano de 2010, no desenvolvimento do algoritmo de classificação LDA, utilizando do algoritmo *Common Spatial Pattern* (CSP) para maximizar a variância do filtro passa-faixa utilizados pelo EEG para uma das classes, enquanto minimiza a variância para as outras demais

Tabela 2 – Resultados obtidos por outros autores em implementações de algoritmos de classificação em FPGA.

Autores	**Algoritmo**	**Acurácia**	**Sinal**
(YANG, 2003)	RNA	92%	Imagem
(IRICK et al., 2008)	SVM	88.6%	Imagem
(GLETTE; TORRESEN, 2009)	EHW	**97.5%**	Imagem
(GLETTE; TORRESEN, 2007)	EHW	91.4 %	Ondas Sonoras
(ALKIM; KILIç, 2011)	LVQ	85%	EMG
(YUAN et al., 2017)	SVM	**97%**	Ondas Ultrassônicas
(CHEN et al., 2016)	NB	92%	EMG
(KAIS et al., 2014)	LDA	72%	EEG

classes, utilizando a base de dados *BCI Competition - Dataset IVa*.

Tabela 3 – Acurácia de classificação do algoritmo LDA utilizando do algoritmo CSP para maximização de variância de classes. (LOTTE; GUAN, 2011)

Sujeitos	**Acurácia do LDA com algoritmo CSP**
aa	66,70%
al	96,43%
av	47,45%
aw	71,88%
ay	49,60%

3 Metodologia

Neste capítulo é apresentado como foi realizada a implementação do algoritmo de treinamento do classificador LDA em um sistema coprocessado hardware-software, detalhando todos os módulos que compõem a implementação.

3.1 Algoritmo Implementado

Em 2010 o francês Fabien Lotte publicou um trabalho com objetivo de comparar as implementações de algoritmos de extração de características de sinais provenientes de atividades cerebrais, além de propor um novo algoritmo para extração de características dos sinais (LOTTE; GUAN, 2011).
Para coleta de resultados de seu trabalho, Lotte utilizou o algoritmo de classificação LDA e a base de dados *BCI Competition III - Dataset IVa*, obtendo como melhor resultado o extrator de características CSP, conforme apresentado anteriormente na Tabela 3.
O algoritmo foi desenvolvido utilizando a plataforma *Matlab*, onde foram utilizados os recursos e funções da plataforma. A Figura 10 apresenta um diagrama de atividades que descreve as funções utilizadas no algoritmo.

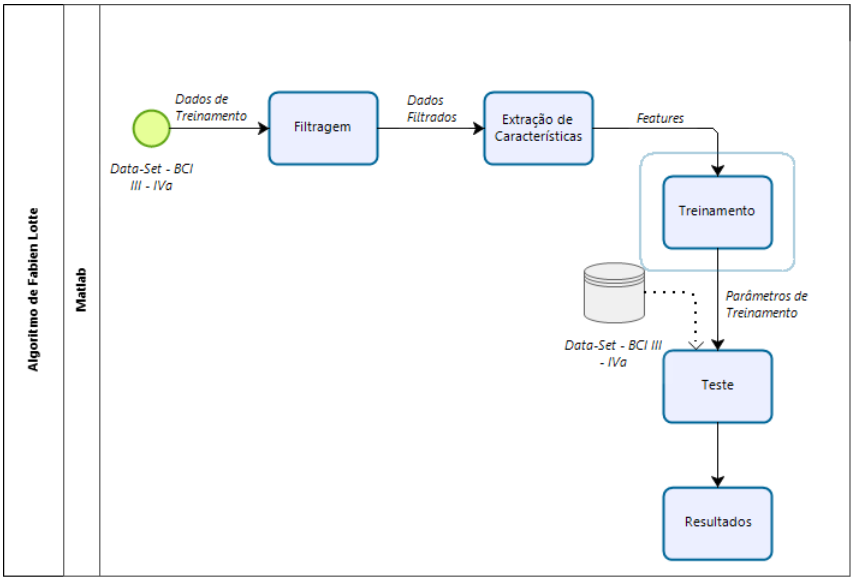

Figura 10 – Atividades realizadas pelo algoritmo de (LOTTE; GUAN, 2011).

Utilizando-se dos dados de treinamento do *BCI Competition III - Data Set -IVa*, o algoritmo inicia-se realizando a filtragem dos sinais a partir de um filtro *Butterworth* passa faixas de 5^a ordem, mantendo os sinais das faixas α e β e atenuando as demais frequências. Em seguida, é realizado o processo de extração de características utilizando o algoritmo

CSP. Logo após, é realizado o processo de treinamento, onde são calculados hiperplanos que melhor separam as classes. Por fim, para o processo de testes o algoritmo utiliza dos hiperplanos gerados pela função de treinamento para classificar os sinais de testes, também fornecidos pelo *BCI Competition III - Data Set -IVa*. Pro fim, são disponibilizados os resultados de acurácia e tempo de treinamento.

3.2 Implementação em Sistema Coprocessado

Conforme apresentado, o objetivo deste trabalho é o estudo dos ganhos ou perdas no processamento do algoritmo de treinamento do classificador LDA, implementado em um coprocessamento hardware-software, utilizando-se do SoC *Zynq* embarcado no kit de desenvolvimento *Xilinx Zybo Board* modelo 70-10, comparado com a implementação realizada por (LOTTE; GUAN, 2011) e a implementação em software executado sobre um SO Linux embarcado nos processadores ARM, também disponíveis no SoC.
Sendo assim, o trabalho se restringiu à implementação em sistema coprocessado as atividades que compõem a função de treinamento do classificador LDA (atividades destacadas) da Figura 10.
A Figura 11 apresenta um diagrama de atividades que descreve a função de treinamento do classificador LDA.

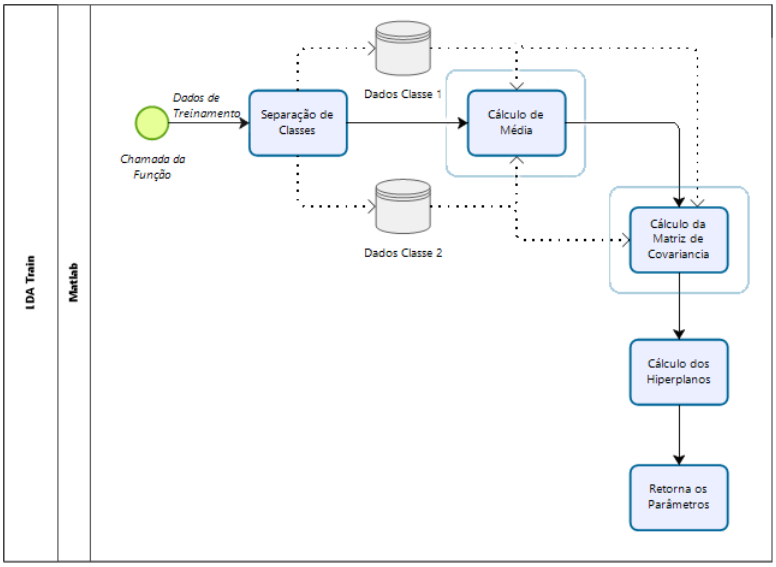

Figura 11 – Atividades realizadas pelo algoritmo de treinamento do classificador LDA, desenvolvido por (LOTTE; GUAN, 2011).

Inicialmente é realizada a separação dos sinais em dois conjuntos de dados que representam os dados referentes às duas classes em estudo. Em seguida são calculadas as médias dos sinais de cada classe. Estes valores são repassados para a função de cálculo da matriz de covariância (ou dispersão), onde são realizados cálculos matriciais conforme apresentado na Equação 2.2. Posteriormente são calculados os hiperplanos a partir da

covariância das características dos sinais. Estes hiperplanos representam os parâmetros de treinamento do classificador. São a partir deles que os sinais de teste são classificados.

Na realização do coprocessamento as funções de cálculo de média e cálculo de covariancia são as duas funções que apresentam um maior esforço computacional. Portanto ambas as funções foram mapeadas em hardware, utilizando-se da linguagem VHDL, a fim de acelerar o algoritmo de treinamento explorando o paralelismo de processos em hardware *FPGA Artix-7* disponível no SoC, enquanto as demais funções foram compiladas em software na linguagem C, executada através de um SO Linux embarcado nos processadores *ARM Cortex-A9*.

3.2.1 Ferramentas utilizadas

Para estudo das implementações (*Matlab*, software e coprocessamento) utilizamos cinco ferramentas principais:

- Software*Matlab R2016a - Student License* - Utilizado para reproduzir os resultados obtidos por (LOTTE; GUAN, 2011);
- SO Linux - Utilizado como ambiente de desenvolvimento de software embarcado;
- Software *Vivado - v.2017.4* - Utilizado para desenvolver, integrar e sintetizar os IP's de cálculo em ponto flutuante das funções média e covariância;
- *SDK - v.2017.4* - Utilizado para desenvolver o arquivo fsbl.elf e para compilar o arquivo BOOT.bin.

Para implementação em software e implementação coprocessada utilizamos o SoC *Zynq* embarcado no kit de desenvolvimento *Zybo Board*

3.2.2 Metodologias de desenvolvimento

3.2.2.1 Implementação em Hardware

Para implementação das funções de cálculo de média e cálculo da matriz de covarinância em FPGA foi adotada a metodologia *bottom-up*, na qual cada sub-bloco desenvolvido foi testado antes de ser inserido ao bloco principal, bloco de integração de todos sub-blocos do projeto, também conhecido como *Top module*. Os cálculos foram realizados através dos blocos de propriedade intelectual (IP) desenvolvidos por (MUÑOZ et al., 2010). Os IPs realiazam cálculos matemáticos em unidades de ponto flutuante de acordo com o padrão IEEE-754 com registradores de 27 bits, representados com:

- expoente: 8 bits;
- mantiza: 18 bits;
- sinal: 1 bit.

Foram criados componentes para paralelizar o processo dos cálculos, buscando o maior nível de paralelismo. Para a função média foram implementados um total de 20(vinte) componentes em paralelo. Já para a função de covariância foram implementados 4 componentes em paralelo, conforme apresentado nas Figuras 12 e 13.

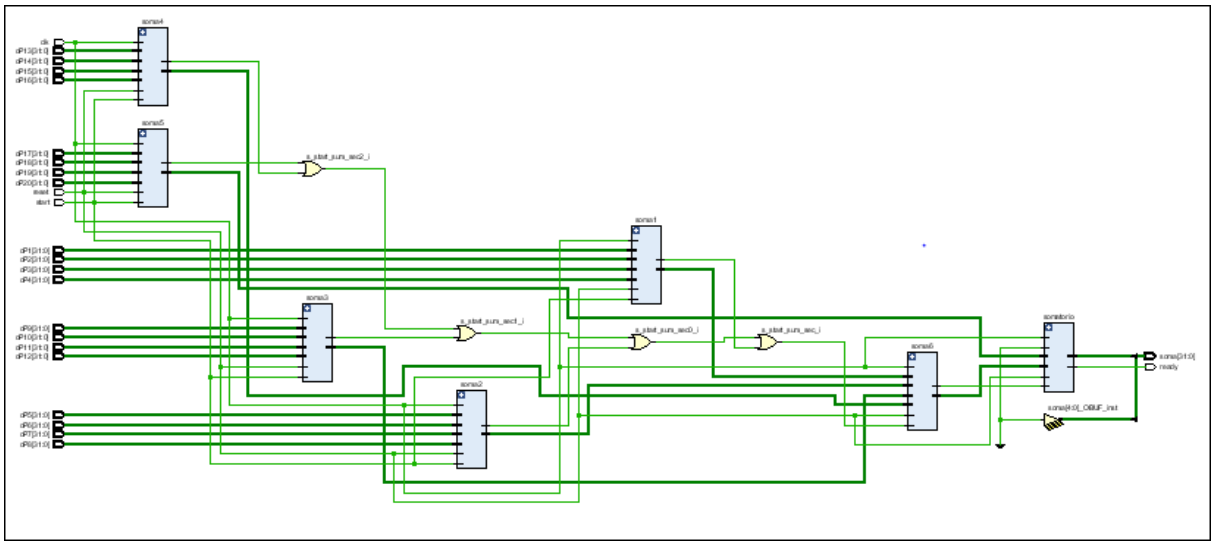

Figura 12 – Bloco para cálculo de média com vinte somas em paralelo.

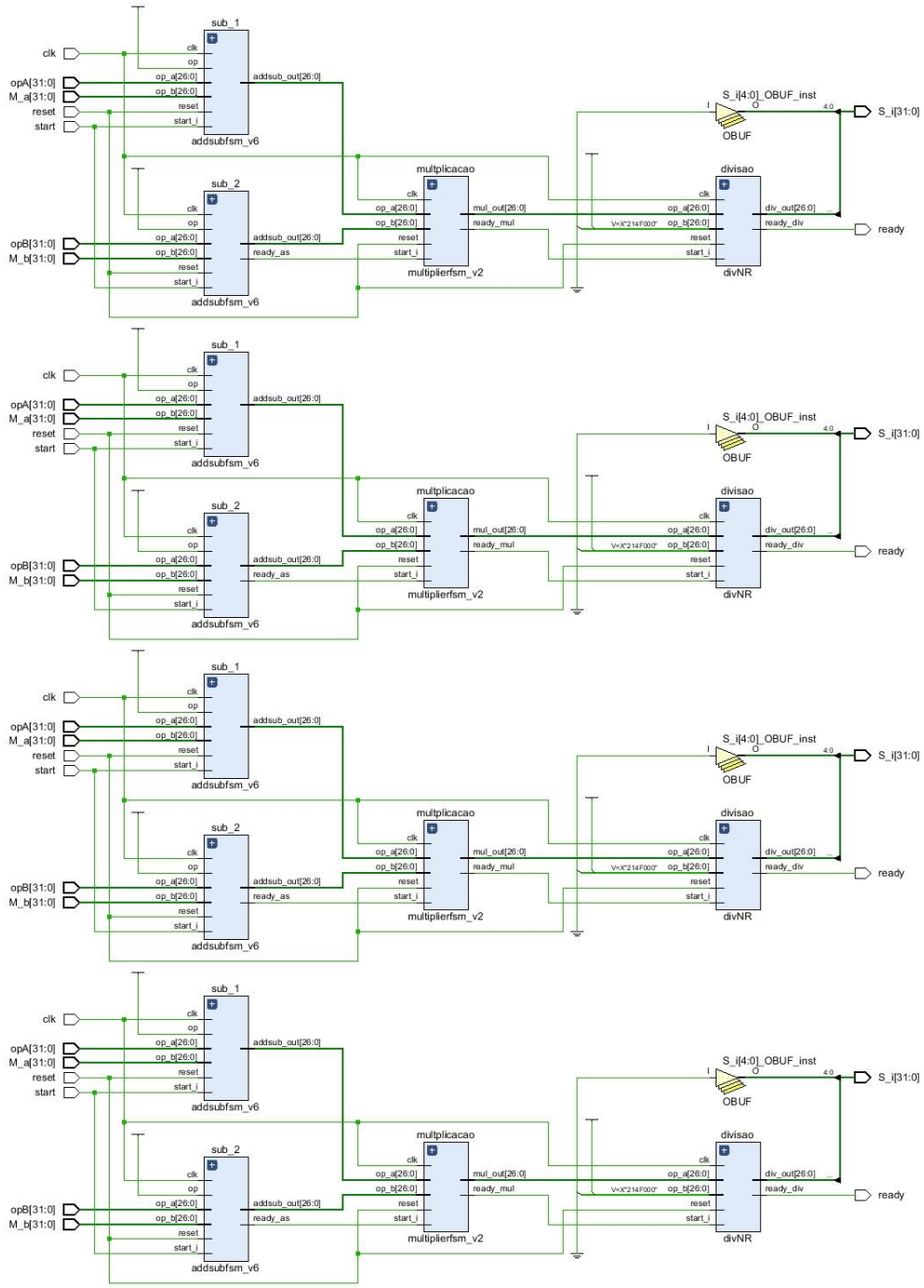

Figura 13 – Bloco para cálculo da matriz de covariancia com quatro cálculos em paralelo.

A técnica utilizada na implementação do projeto foi a *pipelining* que se baseia na divisão de uma tarefa em subtarefas sequenciais de forma simultânea, onde cada novo cálculo só é realizado após o fim do cálculo anterior.

Todos os módulos desenvolvidos foram atribuidos à um novo IP com barramento AXI4-Lite para realizar a comunicação com o procesador ARM. O diagrama de comunicação entre o IP e o processador é apresentado no Anexo 01 deste de trabalho.

Os processos realizados para implementação são apresentados na Figura 14.

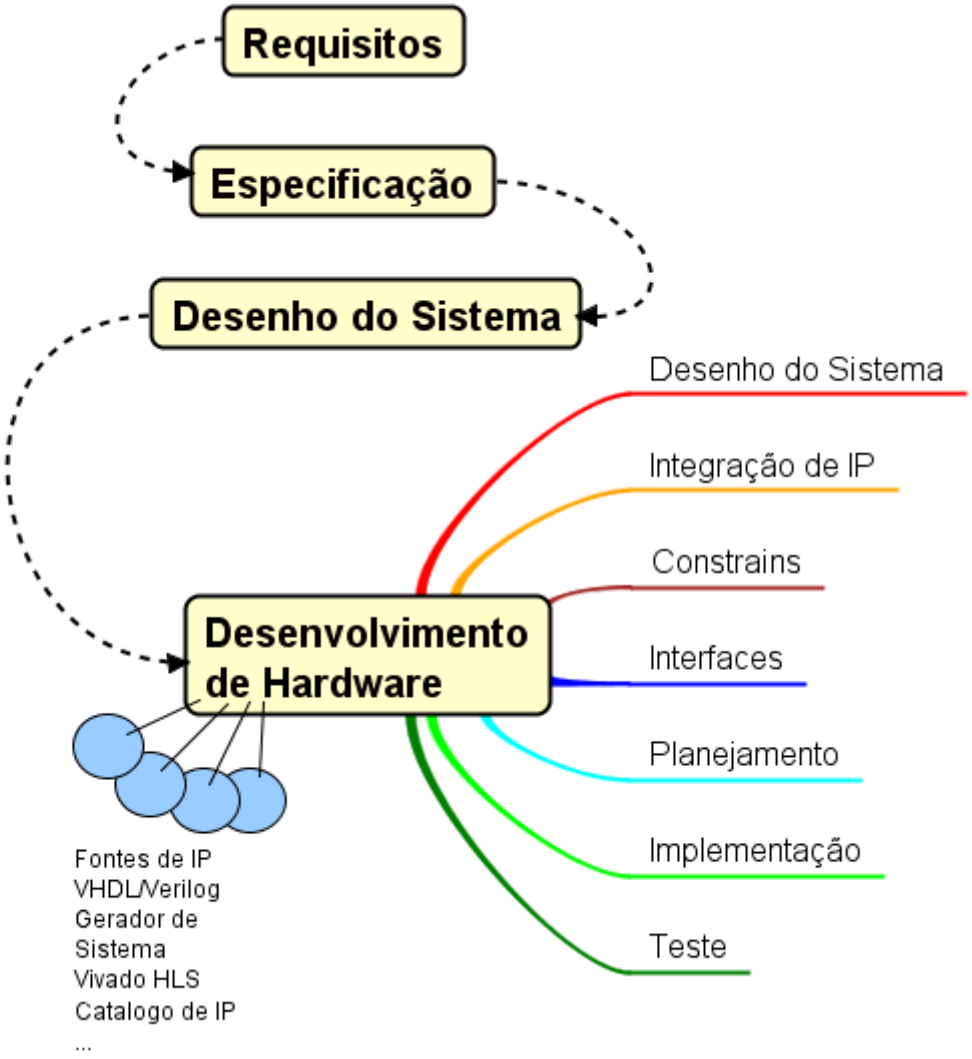

Figura 14 – Fluxograma de implementação de hardware em um SoC. Adaptado de (CROCKETT et al., 2014)

3.2.3 Análise estatística

Para efeito de análise comparatória foram coletados as seguintes informações pós síntese e implementação:

- Consumo de hardware: LUTs, FFs, blocos de DSP, blocos de memória RAM, I/O;
- Dados de desempenho: frequência de operação e tempo de execução;
- Estimação do consumo de energético.
- Estimação estatística do erro quadrático de implementação.

3.3 Implementação em software

3.3.1 Métodos e Técnicas

Para o auxílio desta implementação, foi instalado o sistema operacional *Debian Linux* sobre os cores ARM, cujo os recursos e passos necessários para a instalação do SO estão contidos no Anexo 02.

O processo para implementação seguiu o método *bottom-up*, blocos de códigos menores foram implementados e testados separadamente, com a finalidade de todos os blocos serem integrados e testados em um único bloco principal. Todas as codificações foram implementadas utilizando a linguagem de programação C, para compilar o código principal usamos o compilador GCC *(GNU Compiler Collection)*. Com esse método conseguimos realizar uma análise de perfil das funções implementadas onde foram reportados os tempo de execução de cada uma das funções, sendo assim possível deduzir as funções que necessitam de maior esforço computacional. As entradas para programa desenvolvido são os sinais de treinamento fornecidos pelo *dataset IVa* do *BCI Competition III*.

Como parâmetro de análise estatística foram coletados os seguintes dados:

- Consumo de memória
- Tempo de execução
- Erro da implementação em comparação aos resultados obtidos por (LOTTE; GUAN, 2011)

3.3.2 Integração

Toda a integração do projeto coprocessado é possibilitado pelo uso do barramento AXI4-Lite, que interliga a PL com a PS, um esquemático do processo de escrita e leitura nos IP através do barramento AXI4-Lite pode ser observado na Figura 15

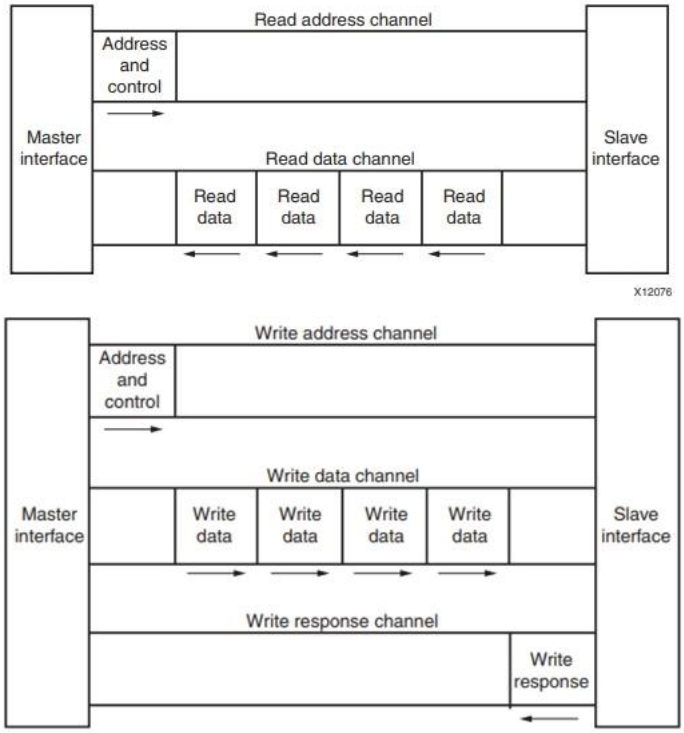

Figura 15 – Fluxo de dados para leitura e escrita no barramento AXI Fone: (XILINX, 2011)

Para escrita e leitura de dados nos IPs através da aplicação implementada em software, ultilizamos a função mmap() que faz parte da biblioteca <sys/mman.h>, esta função cria um novo mapeamento de memória no endereço de memória virtual no seu processo de chamada. Esse novo endereço começa no endereço de memória do IP (com AXI), com isso é possivel ler e escrever no endereço memória de de qualquer IP, desde que a memória esteja devidamente mapeada.

3.4 *Data Set IVa*

Em ambas as implementações serão utilizadas o *Data Set IVa* da *BCI Competition III* (BLANKERTZ et al., 2006), para efeito de testes e validações do sistema. Os dados *Data Set IVa* foram adquiridos e armazenados utilizando amplificadores do tipo *BrainAmp* e uma capa de eletrodos de 128 canais. Foram utilizados 118 canais de EEG posicionados de acordo com o sistema 10-20. Cada um destes canais foram filtrados em banda passante, utilizando um filtro *butterworth* de quinta ordem entre as frequências de 0,05 e 200 Hz, posteriormente foram digitalizados com uma frequência de amostragem de 1 kHz com precisão de 16 bits, apresentando uma resolução de 0,1 μV, além disso também foram disponibilizados os mesmos dados com uma frequência de amostragem de 100 Hz (BLANKERTZ et al., 2005).

4 Resultados Parciais

4.1 Resultados obtidos em MATLAB

Em 2010 fabien lotte publicou um trabalho, demosntrando a classificação de sinais de EEG com LDA (LOTTE; GUAN, 2011),usando o extrator de características *common spatial pattenrs*(CSP) e outras variações deste algoritmo. A base de dados usada foi a base do BCI *competition* III, com os datasets BCI_III_DSIVa, BCI_III_DSIIIa e BCI_IV_DSIIa.

As implementações foram codificadas em *Matlab*, e o autor chegou aos seguintes resultados mostrados na tabela 4.

Tabela 4 – Acurácia da classificação em % usando o extrator CSP e classificador LDA.

BCI III - data set IVa					
Sujeito	A1	A2	A3	A4	A5
CSP	66.07	96.43	47.45	71.88	49.6

Reproduzir os resultados do autor, é de suma importância para este trabalho, tanto para comparação quanto para o entendimento do algoritmo. Para executar o classificador já implementado por (LOTTE; GUAN, 2011), usou-se um laptop DELL, equipado com processador Intel Core I5-4210U CPU 2.4 GHz, 8 GB RAM e sistema operacional Windows 10, ultilizando o mesmo data set, obteve-se os resultados contidos na Figura .

```
reading EEG data...
reading BCI competition III, data set IVa
done !
Learning CSP filters assuming invertible covariance matrices
test set accuracy for subject1 = 66.0714 %
Learning CSP filters assuming invertible covariance matrices
test set accuracy for subject2 = 96.4286 %
Learning CSP filters assuming invertible covariance matrices
test set accuracy for subject3 = 47.449 %
Learning CSP filters assuming invertible covariance matrices
test set accuracy for subject4 = 71.875 %
Learning CSP filters assuming invertible covariance matrices
test set accuracy for subject5 = 49.6032 %
Training time: 0.068064 - std: 0.037339
LDA time: 0.01089 - std: 0.014697
>>
```

Figura 16 – Resultados obtidos com CSP-LDA

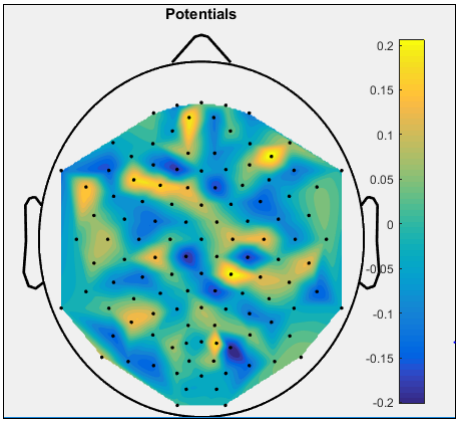

Figura 17 – Resultados obtidos com CSP-LDA

Observando os dados contidos na figura 19, pode ser ver que a reprodução do algoritmo, não teve um alto tempo de execução, sendo o tempo médio de classificação de 0.01089 e desvio padrão de 0.014697 segundos.

Referências

ALKIM, E.; KILIç, E. Chip design for intelligent data classification algorithms and implementation on an fpga: A case study to classify emg signals. In: *2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU)*. [S.l.: s.n.], 2011. p. 307–310. ISSN 2165-0608. Citado na página 42.

ALVAREZ, A.; LEMOS, I. d. C. Os neurobiomecanismos do aprender: a aplicação de novos conceitos no dia-a-dia escolar e terapêutico. *Revista Psicopedagogia*, Associacao Brasileira de Psicopedagogia, v. 23, n. 71, p. 181–190, 2006. Citado 3 vezes nas páginas 29, 30 e 31.

BLANKERTZ, B. et al. *BCI Competition III*. 2005. Disponível em: <http://www.bbci.de/competition/iii/desc_IVa.html>. Citado 2 vezes nas páginas 36 e 50.

BLANKERTZ, B. et al. The bci competition iii: validating alternative approaches to actual bci problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 14, n. 2, p. 153–159, June 2006. ISSN 1534-4320. Citado 5 vezes nas páginas 17, 27, 35, 36 e 50.

BRUNELLI, R. *Template matching techniques in computer vision: theory and practice*. [S.l.]: John Wiley & Sons, 2009. Citado na página 25.

CAMPISI, P.; ROCCA, D. L.; SCARANO, G. Eeg for automatic person recognition. *Computer*, IEEE, v. 45, n. 7, p. 87–89, 2012. Citado 3 vezes nas páginas 15, 32 e 33.

CAO, Y. et al. A post-silicon trace analysis approach for system-on-chip protocol debug. In: IEEE. *2017 IEEE 35th International Conference on Computer Design (ICCD)*. [S.l.], 2017. p. 177–184. Citado 2 vezes nas páginas 15 e 39.

CHEN, X. et al. Soc-based architecture for robotic prosthetics control using surface electromyography. In: *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. [S.l.: s.n.], 2016. v. 01, p. 134–137. Citado na página 42.

CLARK NASHAAT BOUTROS, M. M. D. *The Brain and Behavior: An Introduction to Behavioral Neuroanatomy*. 2. ed. [S.l.]: Cambridge University Press, 2005. ISBN 0521840503,9780521840507. Citado na página 29.

CROCKETT, L. H. et al. *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. [S.l.]: Strathclyde Academic Media, 2014. Citado 6 vezes nas páginas 15, 26, 27, 38, 39 e 48.

DIGILENT. *Zybo Board Reference Manual*. 2017. Disponível em: <https://reference.digilentinc.com/reference/programmable-logic/zybo/reference-manual>. Citado 3 vezes nas páginas 15, 39 e 40.

GLETTE; TORRESEN. An online ehw pattern recognition system applied to sonar spectrum classification. *Evolvable Systems: From Biology to Hardware*, Springer, p. 1–12, 2007. Citado na página 42.

GLETTE, K.; TORRESEN, J. Intermediate level fpga reconfiguration for an online ehw pattern recognition system. In: *2009 NASA/ESA Conference on Adaptive Hardware and Systems*. [S.l.: s.n.], 2009. p. 19–26. Citado na página 42.

GUGER, C. et al. *Brain-Computer Interface Research: A State-of-the-Art Summary*. 1. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2013. (SpringerBriefs in Electrical and Computer Engineering). ISBN 978-3-642-36082-4,978-3-642-36083-1. Citado 2 vezes nas páginas 33 e 34.

IRICK, K. et al. A hardware efficient support vector machine architecture for fpga. In: *2008 16th International Symposium on Field-Programmable Custom Computing Machines*. [S.l.: s.n.], 2008. p. 304–305. Citado na página 42.

IZENMAN, A. J. *Modern multivariate statistical techniques*. [S.l.]: Springer, 2008. v. 1. Citado na página 37.

KAIS, B. et al. An embedded implementation of home devices control system based on brain computer interface. In: *2014 26th International Conference on Microelectronics (ICM)*. [S.l.: s.n.], 2014. p. 140–143. ISSN 2159-1660. Citado na página 42.

KANDEL, e. a. E. R. *Principles of Neural Science*. 5. ed. The address: Mc Graw Hill, 2013. v. 2. An optional note. ISBN 978007181001-2. Citado 4 vezes nas páginas 15, 25, 29 e 30.

KETSUWAN, R.; PADUNGWEANG, P. A linear discriminant analysis using weighted local structure information. In: IEEE. *Computer Science and Software Engineering (JCSSE), 2017 14th International Joint Conference on*. [S.l.], 2017. p. 1–5. Citado na página 37.

KORKMAZ, S. A. et al. A expert system for stomach cancer images with artificial neural network by using hog features and linear discriminant analysis: Hog_lda_ann. In: IEEE. *Intelligent Systems and Informatics (SISY), 2017 IEEE 15th International Symposium on*. [S.l.], 2017. p. 000327–000332. Citado na página 37.

LOTTE, F. *Study of electroencephalographic signal processing and classification techniques towards the use of brain-computer interfaces in virtual reality applications*. Tese (Doutorado) — INSA de Rennes, 2008. Citado na página 25.

LOTTE, F.; GUAN, C. Regularizing common spatial patterns to improve bci designs: Unified theory and new algorithms. *IEEE Transactions on Biomedical Engineering*, v. 58, n. 2, p. 355–362, Feb 2011. ISSN 0018-9294. Citado 14 vezes nas páginas 15, 17, 25, 26, 27, 33, 41, 42, 43, 44, 45, 49, 51 e 53.

MASON, S. G.; BIRCH, G. E. A general framework for brain-computer interface design. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 11, n. 1, p. 70–85, March 2003. ISSN 1534-4320. Citado 3 vezes nas páginas 25, 27 e 35.

MUÑOZ, D. M. et al. Tradeoff of fpga design of a floating-point library for arithmetic operators. *Journal of Integrated Circuits and Systems*, v. 5, n. 1, p. 42–52, 2010. Citado na página 45.

RAO, R. P. *Brain-computer interfacing: an introduction*. [S.l.]: Cambridge University Press, 2013. Citado 2 vezes nas páginas 31 e 32.

SHASHOA, N. A. A. et al. Classification depend on linear discriminant analysis using desired outputs. In: IEEE. *Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2016 17th International Conference on*. [S.l.], 2016. p. 328–332. Citado na página 26.

SINGH, A.; PRAKASH, B. S.; CHANDRASEKARAN, K. A comparison of linear discriminant analysis and ridge classifier on twitter data. In: IEEE. *Computing, Communication and Automation (ICCCA), 2016 International Conference on*. [S.l.], 2016. p. 133–138. Citado na página 37.

SIULY, S. *Analysis and Classification of EEG Signals*. Dissertação (Mestrado) — University of Southern Queensland, https://www.springer.com/gp/book/9783319476520, 7 2012. Citado 8 vezes nas páginas 15, 25, 30, 31, 32, 33, 34 e 35.

SIULY, S.; LI, Y.; ZHANG, Y. *EEG Signal Analysis and Classification: Techniques and Applications*. [S.l.]: Springer, 2017. Citado 4 vezes nas páginas 25, 26, 31 e 32.

TEPLAN, M. et al. Fundamentals of eeg measurement. *Measurement science review*, v. 2, n. 2, p. 1–11, 2002. Citado na página 30.

THEODORIDIS, S.; KOUTROUMBAS, K. et al. *Pattern recognition*. [S.l.]: Academic press London, 1999. Citado na página 26.

WANG, Y. et al. Bci competition 2003-data set iv:an algorithm based on cssd and fda for classifying single-trial eeg. *IEEE Transactions on Biomedical Engineering*, v. 51, n. 6, p. 1081–1086, June 2004. ISSN 0018-9294. Citado na página 41.

WOLPAW, E. W. W. J. R. *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, 2012. ISBN 9780195388855 0195388852. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=46D2894771F9769A0D5B429B681CBF33>. Citado 3 vezes nas páginas 15, 33 e 34.

XILINX. *AXI Reference Guide*. 2011. Disponível em: <https://www.xilinx.com/support/documentation/ip_documentation/ug761_axi_reference_guide.pdf>. Citado 3 vezes nas páginas 15, 40 e 50.

YANG, F. Implementation of an rbf neural network on embedded systems: real-time face tracking and identity verification. *IEEE Transactions on Neural Networks*, v. 14, n. 5, p. 1162–1175, Sept 2003. ISSN 1045-9227. Citado na página 42.

YUAN, Y. et al. Comparison of gpu and fpga based hardware platforms for ultrasonic flaw detection using support vector machines. In: *2017 IEEE International Ultrasonics Symposium (IUS)*. [S.l.: s.n.], 2017. p. 1–4. Citado na página 42.