Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

NELSON CARLOS DE SOUSA CAMPOS

ARQUITETURA E IMPLEMENTAÇÃO EM HARDWARE DE UM MÓDULO MFCC PARA EXTRAÇÃO DE CARACTERÍSTICAS DA VOZ

Campina Grande, Paraíba
Março de 2015
NELSON CARLOS DE SOUSA CAMPOS

ARQUITETURA E IMPLEMENTAÇÃO EM HARDWARE DE UM MÓDULO MFCC PARA EXTRAÇÃO DE CARACTERÍSTICAS DA VOZ

Trabalho de Conclusão de Curso 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 da Informação

Orientador:
Professor Edmar Candeia Gurjão, Dr.

Campina Grande, Paraíba
Março de 2015

NELSON CARLOS DE SOUSA CAMPOS

ARQUITETURA E IMPLEMENTAÇÃO EM HARDWARE DE UM MÓDULO MFCC PARA EXTRAÇÃO DE CARACTERÍSTICAS DA VOZ

Trabalho de Conclusão de Curso 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 da Informação

Aprovado em / /

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Professor Edmar Candeia Gurjão, Dr.
Universidade Federal de Campina Grande
Orientador, UFCG

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

AGRADECIMENTOS

Agradeço primeiramente a 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 Edmar Candeia Gurjão, por ter aceitado orientar este trabalho. Ao professor Elmar Uwe Kurt Melcher, pela formação profissional que adquiri no Brazil-IP. Ao professor Roberto de Medeiros Faria, por sua ajuda e companheirismo.

Agradeço à toda equipe do LAD, em especial à Plateny e Fabrício, que foram essenciais para o progresso deste trabalho.

Um agradecimento ao professor Tarso Vilela Ferreira, pela sua contribuição na formatação deste trabalho e ao professor Marcos Barbosa de Melo, pelo seu altruísmo.

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.

“Aller Anfgang ist schwer.”

Provérbio alemão (“Todo começo é difícil.”).

RESUMO

Técnicas de Processamento Digital de Sinais de Voz são utilizadas em Sistemas de Reconhecimento de Locutor. No enfoque deste trabalho, foi abordada a técnica dos coeficientes mel-cepstrais (MFCC), e a implementação em hardware de um Logaritmo, que é parte integrante do sistema de extração de características da voz foi abordada aqui. Simulações de consumo de energia e área ocupada no silício foram realizadas para uma tecnologia de 28nm. Técnicas de redução de consumo foram estudadas e aplicadas, visando-se eliminar o uso de multiplicadores, ao empregar a multiplicação egípcia, que realiza operações de soma e deslocamento de bits.

Palavras-chave: PDSV, Hardware, Baixo consumo, MFCC.

ABSTRACT

Techniques of Digital Processing of Speech Signals are being used in Speaker Recognition Systems. However, in the focus of this study, it was approached the technique of mel-cepstral coefficients (MFCC), and the hardware implementation of a logarithm, which is part of the features extraction from speech system that was accomplished here. Simulations were performed for consumption and area occupied on the chip. Consumption reduction techniques have been studied and applied in order to eliminate the use of multipliers by employing Egyptian multiplication, that performs sums and bit shift operations.

Keywords: DPSS, Hardware, Low Power, MFCC.

LISTA DE SIGLAS E ABREVIATURAS

A/D	Analógico/Digital
AMBA	Advanced Microcontroller Bus Architecture
ASIC	*Application Specific Integrated Circuits*
CDE	Comparação por Distância Euclidiana
DCT	*Discrete Cosine Transform*
DM	*Decision Maker*
DSP	*Digital Signal Processor*
DV	Detector de Voz
EF	Energia Final
EI	Energia Inicial
FFT	*Fast Fourier Transform*
FIR	*Finite Impulse Response*
FPGA	*Field Programmable Gate Array*
HMM	*Hidden Markov Model*
LBG	*Linde-Buzo-Gray*
LI	Limiar Inferior
LPC	*Linear Prediction Coding*
LS	Limiar Superior
MFCC	*Mel-Frequency Cepstral Coefficients*
PDSV	Processamento Digital de Sinais de Voz
PE	Pré-ênfase
PM	*Parting Matching*
QA	Quantidade de Acertos
QD	Quantidade de Desconhecidos
QE	Quantidade de Erros
QV	Quantização Vetorial
RD	Regra de Decisão
SRAM	*Static Random-Access Memory*
SRF	Sistemas de Reconhecimento de Fala
SRL	Sistemas de Reconhecimento de Locutor

SRV	Sistemas de Resposta Vocal
TF	Tempo de Fim
TI	Tempo de Início
VLSI	*Very Large Scale Integration*
WIN	Windowing

SUMÁRIO

Resumo VII
Abstract VIII
Lista de Siglas e Abreviaturas IV
Sumário XI
1. Introdução 1
    1.1 Objetivos 3
    1.1.1 Objetivos específicos 3
2. Reconhecimento de Locutor 4
    2.1 Aquisição 4
    2.2 Pré-processamento 5
    2.3 Normalização 5
    2.4 Detecção de voz 5
    2.5 Pré-ênfase 6
    2.6 Segmentação e Janelamento 6
    2.7 Extração de características 7
        2.7.1 Estimação dos Coeficientes MFCC 9
        2.7.2 Geração de padrões 11
3. Desenvolvimento 13
4. Conclusão 21
5. Bibliografia 22
6. Anexo A – Área e consumo de Multplicador 23
7. Anexo B – Área e consumo do Peasant 25
8. Anexo C – Área e consumo do Logaritmo 27

1 INTRODUÇÃO

Reconhecimento de locutor é uma área de pesquisa do campo Processamento Digital de Sinais de Voz (PDSV). Esta área, por sua vez, está dividida em três grandes subáreas:

1. Resposta Vocal (síntese);
2. Reconhecimento de Fala;
3. Reconhecimento de Locutor.

Na Figura 1.1, apresenta-se uma classificação geral do processamento de voz, com ênfase no reconhecimento de locutor, objeto de estudo da pesquisa ora descrita (CAMPBELL, 1997).

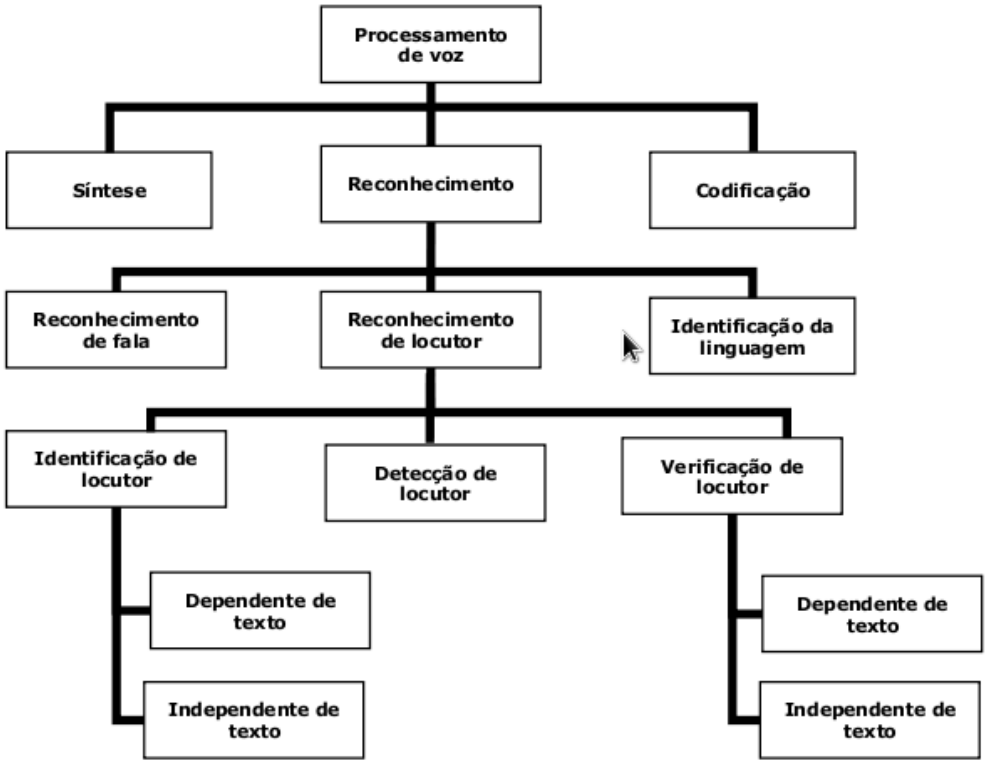


Figura 1.1: Classificação geral da área de processamento digital de sinais de voz (Adaptado de CAMPBELL, 1997)

> "Os Sistemas de Resposta Vocal utilizam sinais de voz armazenados para compor a mensagem de saída, que é formada pela concatenação dos sinais armazenados, com base em regras linguísticas dos idiomas (regras gramaticais e fonéticas)." (LÉLIS DE MELO, 2015)

Os sistemas de verificação (validação ou autenticação) confirmam a identificação do locutor (LI, 2012). O funcionamento típico de um sistema de verificação de locutor, acontece em duas etapas: treinamento e teste. O treinamento é a etapa em o que locutor precisa cadastrar o modelo vocal representativo do seu sinal de voz, junto com um identificador para esse modelo, que pode ser um número. O locutor repetirá a sentença algumas vezes, construindo-se assim, um modelo com as suas características vocais representativas. Na fase de teste, o locutor fornece sua identificação digitando um número ou outra forma de identificação. Em seguida, o sistema solicita que o usuário pronuncie sua sentença. Uma vez enunciada, a sentença é comparada com o modelo armazenado. O locutor é então rejeitado ou aceito pelo sistema se o valor resultante exceder um limiar pré-definido.

1.1 OBJETIVOS

O presente trabalho tem por objetivos o estudo e a implementação em hardware de módulos MFCC para a extração de características da voz.

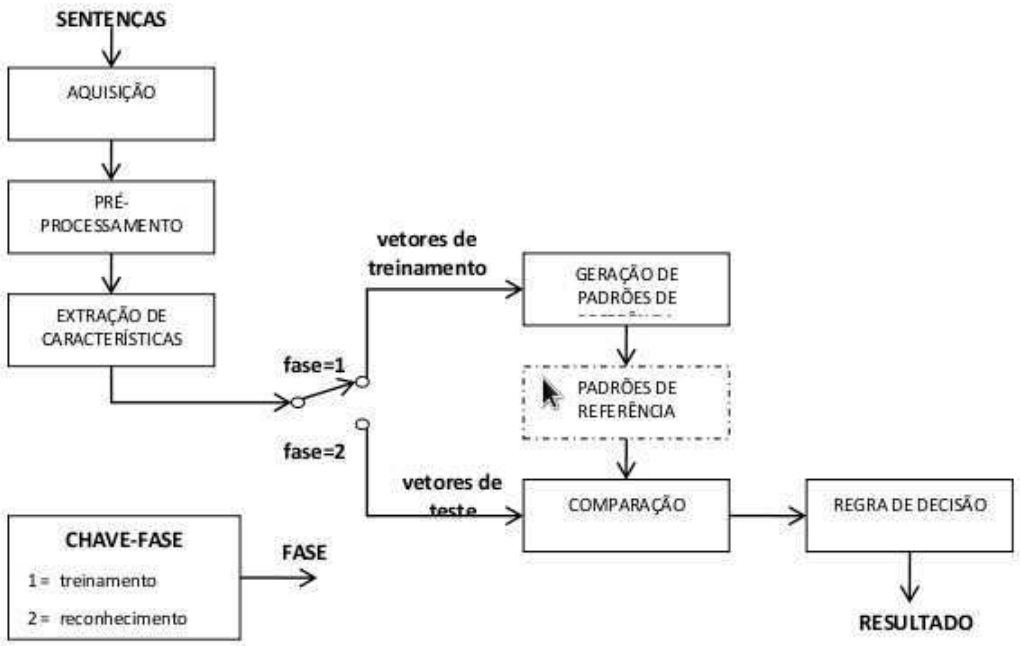
1.1.1 OBJETIVOS ESPECÍFICOS

- O estudo dos blocos constituintes do MFCC e suas aplicações em reconhecimento de locutor.
- Desenvolvimentos de módulos IP-Core para os projetos Brazil-IP/PLC e Brazil-IP/SPVR.

- Aprimoramento pessoal nas áreas de Processamento Digital de Sinais e Design de Hardware Digital.

2 RECONHECIMENTO DE LOCUTOR

Os sistemas de Reconhecimento de Locutor são utilizados para identificar a pessoa que fala a partir de extração de características da voz inerentes ao falante. Estes sistemas têm duas fases: uma de treinamento e outra de teste. Na Figura 2.1, é apresentado o diagrama de blocos de um sistema de reconhecimento de padrões.



Fonte:(RABINER; SCHAFER, 2010)

Figura 2.1: Diagrama de blocos de um sistema de reconhecimento de padrões.

A seguir, serão descritas as etapas presentes na tarefa de reconhecimento de padrões. As etapas de aquisição, pré-processamento e extração de características são comuns às duas fases (treinamento e reconhecimento).

2.1 AQUISIÇÃO

Nesta etapa, a captura do sinal é realizada, usualmente, por um microfone, que

converte a vibração sonora em sinal elétrico (sinal analógico contínuo no tempo e em amplitude).

Na fase de conversão, o sinal analógico é discretizado no tempo, para permitir seu processamento por qualquer sistema digital. Esse processo de conversão é feito pela amostragem do sinal analógico, $s_a(t)$, com período de T segundos, e as amostras são quantizadas, para se obter um sinal amostrado $s(n)$ (RABINER; SCHAFER, 2010), dado por:

$$s(n) = s_a(n \times T), n = 0,1,2, \ldots \qquad (2.1)$$

2.2 PRÉ-PROCESSAMENTO

Esta etapa reduz os efeitos indesejados incorporados ou presentes no sinal de voz, preparando-o para as etapas seguintes do processo de reconhecimento. Esta etapa é composta pelas seguintes subetapas: normalização, detecção de voz, pré-ênfase, segmentação e janelamento (CARDOSO, 2009; O'SHAUGHNESSY, 2000; RABINER; SCHAFER, 2010).

2.3 NORMALIZAÇÃO

Durante a normalização, há uma redução na variabilidade do sinal de voz em relação ao ambiente de gravação (ruído de fundo, canal de comunicação). A variabilidade causada pelas diferenças de intensidade de voz dos locutores também é restringida (O'SHAUGHNESSY, 2000), ou seja, a amplitude do sinal de voz dos locutores é limitada a uma dada faixa de valores.

2.4 DETECÇÃO DE VOZ

No processo de aquisição, há muitas amostras com informações redundantes. Portanto, para realizar uma extração de características dos locutores de maneira

eficiente, é necessário eliminar trechos do sinal adquirido que contenham informações desnecessárias ou redundantes (ruídos de fundo ou trecho com silêncio, por exemplo).

Uma técnica para detectar a voz consistem em mensurar a energia do segmento do sinal (vide Equação 2.2) para identificar os trechos de voz ativa (FECHINE et al., 2010). Na Equação 2.2, E_{seg}, é a energia do segmento, $s(n)$ é a n-ésima amostra do sinal de voz e N_a é o número de amostras por segmento do sinal.

$$E_{seg} = \sum_{n-0}^{N_a-1} [s(n)]^2 . \qquad (2.2)$$

2.5 PRÉ-ÊNFASE

A pré-ênfase realiza uma filtragem das componentes presentes nas altas frequências utilizando um filtro FIR (*Finite Impulse Response*) de primeira ordem. Esta filtragem se faz necessária devido aos efeitos da variação da glote e da impedância de radiação, causados pelos lábios no processo de produção da voz.

A função de transferência da pré-ênfase, é obtida pela transformada *z* e é dada pela Equação 2.3, sendo α o fator de pré-ênfase, um número entre 0,9 e 1,0 (RABINER; SCHAFER, 1978).

$$H(z) = 1 - \alpha z^{-1}, \quad 0 \leq \alpha \leq 1. \qquad (2.3)$$

2.6 SEGMENTAÇÃO E JANELAMENTO

Uma das características do sinal de voz é não ser estacionário nem periódico . No entanto, em intervalos de tempo de 10 a 30 ms, os segmentos de voz possuem

caráter estacionário e a segmentação do sinal se dá utilizando técnicas de janelamento.

Uma janela retangular (vide Equação 2.4) é uma partição em blocos consecutivos de mesmo tamanho N_a do sinal. Porém, essa divisão abrupta, causa, no domínio da frequência, fugas espectrais que alteram o espectro do sinal" (FECHINE, 2000).

$$J(n) = \begin{cases} 1, & 0 \leq n \leq N_a - 1 \\ 0, & \text{caso contrário} \end{cases}. \qquad (2.4)$$

A janela de Hamming, descrita pela equação 2.5, elimina essas transições abruptas presentes nas extremidades entre blocos consecutivos. (FECHINE, 2000).

$$J(n) = \begin{cases} 0{,}54 - 0.46\cos\left[\frac{2\pi n}{(N_a - 1)}\right], & 0 \leq n \leq N_a - 1 \\ 0, & caso\ contrário \end{cases}. \qquad (2.5)$$

O janelamento de Hanning (equação 2.6) assemelha-se à janela de Hamming, gerando um esforço menor nas amostras do centro e uma atenuação maior nas amostras da extremidade (FECHINE, 2000).

$$J(n) = \begin{cases} 0{,}5 - 0{,}5\cos[2\pi n/(N_a - 1)], & 0 \leq n \leq N_a - 1 \\ 0, & caso\ contrário \end{cases}. \qquad (2.6)$$

As janelas de Hamming e Hanning atenuam fortemente as amostras da extremidade do quadro. Sendo assim, é comum o uso de superposição de quadros adjacentes, a fim de garantir que a variação dos parâmetros entre as janelas adjacentes seja mais gradual e que a análise das amostras localizadas nos extremos das janelas não seja prejudicada." (DA SILVA, 2009)

2.7 EXTRAÇÃO DE CARACTERÍSTICAS

A etapa de extração de características é essencial em sistemas de reconhecimento, pois os elementos que possibilitam a geração de padrões são obtidos nesta fase. No presente trabalho, utiliza-se a técnica dos Coeficientes Mel-Cepstrais.

Os coeficientes MFCC (Mel-Frequency Cepstrum Coefficients) surgiram devido aos avanços na área de psicoacústica. Esta ciência mostra que a percepção das frequências de tons puros ou de sinais de voz não seguem uma escala linear, estimulando assim, a ideia de criar uma escala, chamada Mel.

A percepção humana de frequências de tons puros ou de sinais de voz não seguem uma escala linear e é baseada nas distribuições de bandas críticas. Para cada tom com sua respectiva frequência, medida em Hz, corresponde um valor medido na escala Mel. O Mel é uma unidade de medida de frequência. Como referência, foi definida a frequência de 1 kHz, com potência 40 dB acima do limiar mínimo da audição do ouvido humano, equivalendo a 1000 *mels."* (CARDOSO, 2009)

O mapeamento da frequência em Hz para uma frequência na escala Mel é definida pela equação 2.7

$$F_{mel} = 2595 \log\left(1 + \frac{F_{linear}(Hz)}{700}\right) \qquad (2.7)$$

Os valores 2595 e 700 foram obtidos empiricamente. (LÉLIS DE MELO, 2015).

Por meio destes valores, é possível verificar a relação entre a escala linear e escala Mel a partir da Figura 2.3.

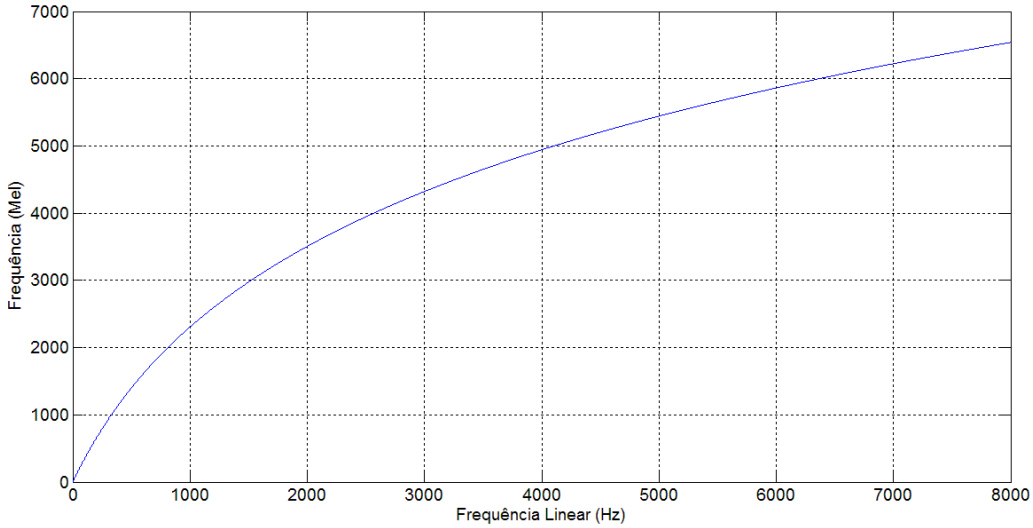


Figura 2.3: Mapeamento da frequência linear na escala Mel.

2.7.1 Estimação dos Coeficientes MFCC

Na Figura 2.4 está ilustrado o diagrama de blocos do processo de obtenção dos coeficientes mel-cepstrais . A entrada do sistema é o sinal de voz já pré-processado e janelado. O espectro do sinal é obtido a partir da FFT. Em seguida, um banco de filtros triangulares cujas frequências centrais obedecem à escala Mel recebem como entrada o espectro dos segmentos de voz. O banco de filtros enfatiza as componentes de frequência centrais, atenuando as demais. O logaritmo da energia resultante é calculado e os coeficientes mel-cepstrais são fornecidos com a aplicação de uma DCT (Discrete Cossine Transform).

Fonte: (CARDOSO, 2009).

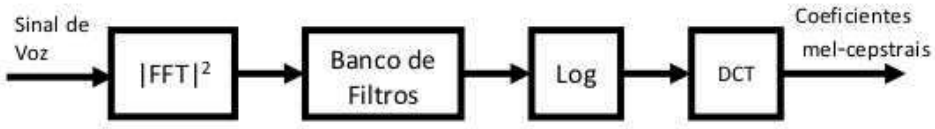
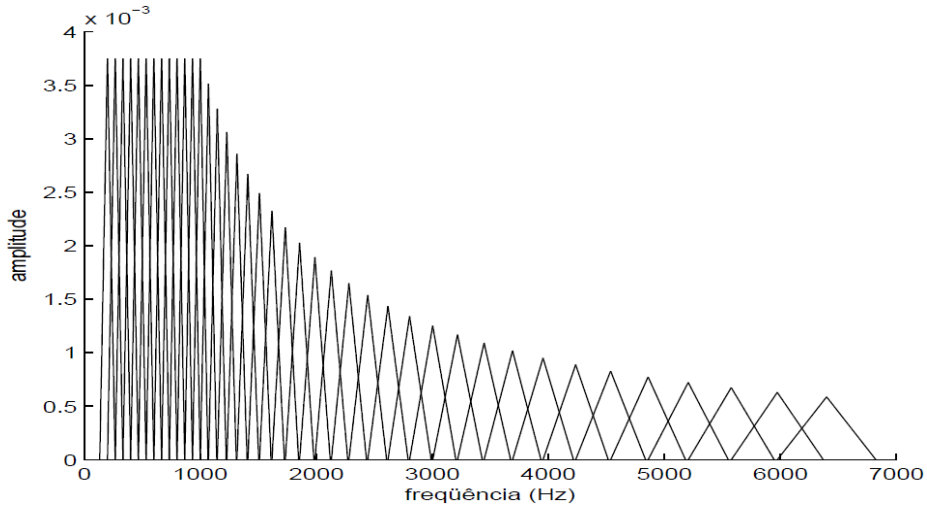


Figura 2.4: Obtenção dos coeficientes mel-cepstrais a partir de banco de filtros.

O uso de bancos de filtros pode reduzir a dimensionalidade da informação extraída do sinal de voz. A quantidade de informação é proporcional ao número de filtros utilizados. Os filtros estão linearmente espaçados segundo a escala Mel. Sendo assim, há mais filtros nas frequências mais baixas, onde a energia do sinal de voz está mais concentrada. Portanto, o banco de filtros enfatiza as frequências onde ocorrem variações perceptíveis de acordo com a escala Mel.

Na Figura 2.5 está ilustrado o espectro do banco de filtros triangulares da implementação de Stanley (1998).



Fonte: (CARDOSO, 2009)

Figura 2.5: Espectro do banco de filtros triangulares de (SLANEY, 1998)

O processo de obtenção dos coeficientes MFCC são representados matematicamente pela equação 2.8:

$$S(k) = \sum_{k=1}^{N\,FFT} W_k(j)X(j), \quad k = 1, \ldots, M, \qquad (2.8)$$

onde,

$W_k(j)$ são as janelas de ponderação triangulares conforme a escala Mel;

$X(j)$ é o espectro da magnitude da FFT de N pontos;

$S(k)$ é o sinal de saída do banco de filtros;

M é o número de filtros utilizados.

Aplicando a DCT na saída do banco de filtros, obtém-se os coeficientes MFCC, conforme mostra a equação 2.9:

$$C_{mel}(n) = \sum_{k=1}^{M} \log(S(k)) \cos\left[n(k-\frac{1}{2})\right]\frac{\pi}{M}, \quad n = 0,1, \ldots, M, \qquad (2.9)$$

onde $C_{mel}(n)$ é o n-ésimo coeficiente mel-cepstral.

Os métodos de extração de características baseados em MFCC apresentam melhores taxas de reconhecimento, embora aumentem a complexidade de sua implementação em hardware. O aumento da complexidade algorítmica se dá devido ao fato da FFT, da DCT, do banco de filtros e do logaritmo.

O objetivo e principal contribuição deste trabalho foi o desenvolvimento do módulo do logaritmo. No Capítulo 3 serão apresentadas as técnicas utilizadas na implementação dos módulos aritméticos, que foram desenvolvidos tanto para sua utilização no projeto Brazil-IP/PLC tanto quanto Brazil-IP/SPVR.

2.7.2 GERAÇÃO DE PADRÕES

Esta tarefa consiste na comparação realizada entre padrões que são obtidos a partir da representação de características extraídas de um sinal de voz, pertencentes a um locutor a ser avaliado durante a fase de testes, com padrões de referências previamente armazenados na fase de treinamento.

A técnica utilizada neste trabalho para a geração dos padrões de referência foi a Quantização Vetorial. Esta técnica é indicada por remover a redundância, que geralmente existe, entre os sucessivos vetores de observação (O'SHAUGHNESSY, 2000). Além disso, apresenta um baixo custo computacional. (LÉLIS DE MELO, 2015)

"A quantização vetorial (QV) é um método de compressão de dados que mapeia um conjunto de vetores de observação X, com o objetivo de gerar um conjunto de M vetores-código C. Os conjuntos X e C são definidos formalmente por (CIPRIANO, 2001)":

$$X = \{x_t \in \mathcal{R}^p \mid t = 1 \ldots T\}. \qquad (2.10)$$

$$C = \{y_i \in \mathcal{R}^p \mid i = 1 \ldots M\}. \qquad (2.11)$$

"O conjunto corresponde ao dicionário (*codebook*) do quantizador que possui níveis e os vetores x_t e y_i, são p-dimensionais. Portanto, tem-se um quantizador vetorial p-dimensional de M-níveis (ou M-partições)."

"A tomada de decisão é baseada em dois limiares: um superior e outro inferior. Quando o valor de distância fica abaixo do limiar inferior, o locutor é aceito, ou seja, o locutor é realmente quem ele alega ser. Entretanto, se o valor de distância fica acima do limiar superior, o locutor é rejeitado. Por fim, se o valor de distância ficar entre os dois limiares, é dito que o locutor é desconhecido e então o mesmo repete sua sentença."(LÉLIS DE MELO, 2015)

3 DESENVOLVIMENTO

Conforme foi visto no Capítulo 2, para a obtenção dos coeficientes MFCC, é necessário que a saída do banco de filtros seja aplicada em um bloco que computa o cálculo do logaritmo.

Para a implementação do logaritmo em hardware, inicialmente foi utilizada a abordagem descrita em (Saambhavi, 2012) que se utiliza de look-up tables (LUTs) para uma aproximação do cálculo. De acordo com a técnica descrita, qualquer número a pode ser escrito de acordo com a expressão (3.1), onde p é um inteiro e N é um escalar entre 0.5 e 1.0.

$$a = N * 2^p \qquad (3.1)$$

Aplicando o logaritmo binário em (3.1), obtém-se a seguinte expressão:

$$\log_2 a = p + \log_2 N \qquad (3.2)$$

Aplicando uma mudança de base, para a obtenção do logaritmo natural, a expressão (3.2) pode ser reescrita como:

$$\log_e a = (p + \log_2 N) * \log_e 2 \qquad (3.3)$$

Uma vez que $\log_e 2$ é uma constante, o logaritmo natural de a pode ser obtido a partir da soma de p e do logaritmo binário de N, e o resultado é multiplicado por uma constante. O valor de p é obtido encontrando-se a posição do bit mais significativo de a, que obviamente é 1. $\log_2 N$ é encontrado em uma tabela armazenada em uma LUT.

A tabela contém 256 valores que armazenam os logaritmos dos números entre 0.5 e 1.0 na base 2, com um intervalo de 0.5/256=0.01953, entre cada célula da LUT. Com uma manipulação matemática, o índice da LUT é encontrado pela relação $(N - 0.5) \ll 9$, uma vez que $N = a \gg p$.

O uso de LUTs (Lookup Tables) com grandes quantidades de valores pode encarecer o projeto do IP-Core, razão pela qual o algoritmo baseado na tabela foi descartado.

Uma maneira elegante e simples de computar o logaritmo binário é apresentada por Clay S. Turner (LYONS, 2012). Este método foi adotado por sua simplicidade e como ponto de partida, deseja-se encontrar

$$y=\log_2(x) \qquad (3.4)$$

Invertendo-se a expressão 3.4, tem-se:

$$x=2^y \qquad (3.5)$$

Assumindo que $1\le x<2$, caso contrário deve-se escalar o valor de x, pode-se calcular o logaritmo de x de maneira iterativa da maneira descrita a seguir.

Uma vez que x esteja escalado adequadamente, o que implica $0\le y<1$, pode-se expandir y como uma série binária conforme a expressão 3.6.

$$y=y_1 2^{-1}+y_2 2^{-2}+y_3 2^{-3}+y_4 2^{-4}+... \qquad (3.6)$$

A equação 3.6 pode ser reescrita pela expressão (3.7) como é mostrado a seguir:

$$y=2^{-1}\left(y_1+2^{-1}\left(y_2+2^{-1}\left(y_3+2^{-1}\left(y_4+...\right)\right)\right)\right) \qquad (3.7)$$

Introduzindo 3.7 em 3.5, obtém-se a expressão 3.8:

$$x=2^{2^{-1}\left(y_1+2^{-1}\left(y_2+2^{-1}\left(y_3+2^{-1}\left(y_4+...\right)\right)\right)\right)} \qquad (3.8)$$

Elevando a expressão 3.8 ao quadrado, vem que:

$$x^2=2^{y_1}2^{2^{-1}\left(y_2+2^{-1}\left(y_3+2^{-1}\left(y_4+...\right)\right)\right)} \qquad (3.9)$$

Ao observar a equação 3.9, percebe-se que dependendo do valor de y_1 , haverão dois casos a serem analisados:

$$y_1=0: \qquad x^2=2^{2^{-1}\left(y_2+2^{-1}\left(y_3+2^{-1}\left(y_4+...\right)\right)\right)} \qquad (3.10)$$

$$y_1=1: \qquad x^2=2.2^{2^{-1}\left(y_2+2^{-1}\left(y_3+2^{-1}\left(y_4+...\right)\right)\right)} \qquad (3.11)$$

Analisando as expressões 3.10 e 3.11, nota-se que o bit mais significativo de y será 1 se $x^2\ge 2$, caso contrário, y_1 será 0. Baseado nessas informações, um algoritmo para o cômputo do logaritmo é descrito pelos seguintes passos:

1. Escale x, tal que $1\le x<2$;
2. Se $x^2\ge 2$, atribua o bit mais significativo de y à 1 e divida x por 2, senão apenas atribua o bit mais significativo de y à 0;

O logaritmo é computado em N passos, onde N é o tamanho da palavra do argumento. Uma vez que o algoritmo utiliza apenas comparações e deslocamento de bits, a parte mais custosa dos cálculos é a comparação de um número quadrado, que requer o uso de multiplicadores.

O método aqui proposto é uma modificação da técnica exposta por Turner, pelo fato de eliminar o multiplicador. A seguir será descrita uma técnica de multiplicação milenar amplamente utilizada no Egito. O módulo foi desenvolvido e incorporado ao módulo do logaritmo e foi denominado *Peasant Multiplication*.

Para ilustrar o uso dessa multiplicação, utiliza-se como exemplo o produto de 9 por 8, que obviamente deve resultar em 72. Construindo-se uma tabela sendo a primeira coluna A = 9 e B = 8, efetua-se o processo de multiplicação dividindo-se B por 2 até que A seja 1, ao mesmo tempo que B é reduzido para a metade a cada passo, multiplica-se A por 2. O resultado está ilustrado na Tabela 3.1.

Tabela 3.1: Ilustração do método de Multiplicação Peasant

A	**B**
9	8
18	4
36	2
72	1

Geometricamente, o que ocorre é que dobrando-se A e divindo-se B por 2, o produto AB permanece constante, como pode ser ilustrado na Figura 3.1

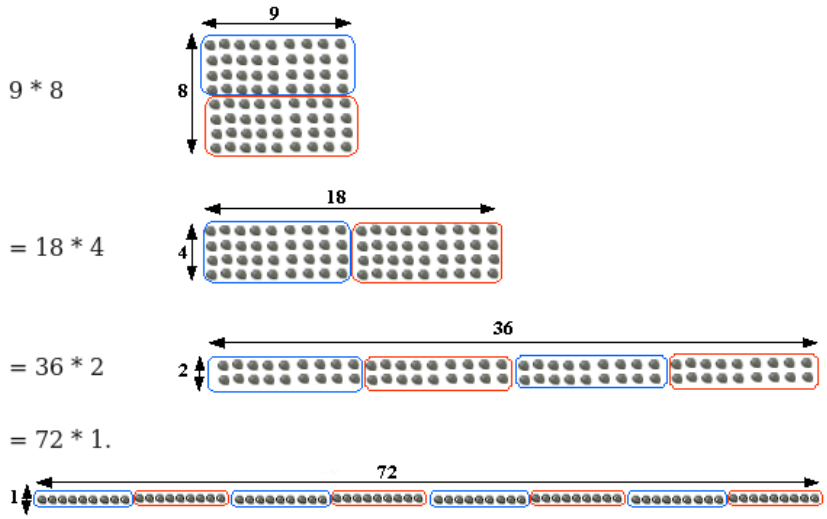


Figura 3.1: Uso do Peasant (9x8)

Como a multiplicação é comutativa, ao efetuar-se o produto de 8*9, temos os valores de A e de B distribuídos conforme a Tabela 3.2

Tabela 3.2: Ilustração do método de Multiplicação Peasant

A	B
8	9
16	4
32	2
64	1

O resultado é do produto é novamente 72. Contudo, conforme ilustra a Figura 3.2, ao dividir-se 9 por 2, obtém-se 4 com resto 1. Contudo, este resto é multiplicado por 8, pois para obter-se o resultado correto, 16*4+8 = 72, o resto deve ser acumulado. Repetindo-se o processo, tem-se 32*2+8=72 e finalmente, 64*1+8=72.

Nota-se assim, de acordo com as tabelas 3.1 e 3.2, que o resultado do produto de A por B é a soma dos valores de A quando os valores de B são ímpares. Esta técnica é fascinante, uma vez que sua implementação em hardware requer apenas o uso de deslocadores de deslocamento e acumuladores. Para acumular apenas os valores de A quando B é ímpar, faz-se uma operação AND do bit menos significativo de B com o valor de A. Sendo assim, o algoritmo do Peasant é executado pelos seguintes passos:

1. Desloque B para a direita até que B seja 1. Simultaneamente, desloque A para a esquerda;
2. Acumule o valor de A and LSB(B), onde LSB(B) é o bit menos significativo de B.

O módulo do peasant foi implementado em SystemVerilog e com a ferramenta dc_shell da Synopsys, estimou-se a área e o consumo do módulo, utilizando-se uma tecnologia de 28nm. Foi utilizado na simulação um clock de 5MHz. A tabela 3.3 compara a área e a potência consumida do módulo Peasant e do multiplicador normal utilizando entradas de 8 bits.

Tabela 3.3: Comparação entre Peasant e Multiplicação Convencional (Área e Consumo)

Módulo	**Área (μm^2)**	**Consumo (uW)**
Peasant	216.729603	1.7358
Multiplicador	273.088008	1.7223

De acordo com a tabela 3.3, o multiplicador convencional ocupa 26% a mais de área em relação ao Peasant. Por outro lado, o Pesant consome 0.78% a mais de potência que o método convencional. O *tradeoff* (Área x Consumo) é aceitável, uma vez que a técnica implementada reduz o espaço do circuito ocupado no chip e o multiplicador convencional se utiliza de blocos patenteados, que encarecem o valor final do produto.

De posse destes resultados, decidiu-se utilizar o Peasant Multiplication como base para o comparador quadrático do Logaritmo. O IP-Core foi desenvolvido em SystemVerilog e foi utilizada uma notação em ponto fixo parametrizável, com M bits para a parte inteira do argumento e N bits para a parte fracionária.

Uma vez que o módulo irá ser parte integrante de um sistema, é necessário uma interface de comunicação entre dispositivos. O protocolo utilizado foi o AMBA® AXI.

O AMBA (*Advanced Microcontroller Bus Architecture*) AXI (*Advanced eXtensible Interface*), é um padrão aberto de interconexões de blocos de Sistemas-em-um-Chip (SoC), introduzido pela ARM. Neste protocolo, a comunicação acontece quando ambos os sinais de handshake (READY e VALID) são nível alto e durante a borda de subida do clock.

O sinal READY indica que o dispositivo está pronto para receber informação, ao passo que o VALID indica que está pronto para enviar dados. A Figura 3.2 ilustra o handshake de um dispositivo fonte que está pronto para enviar os dados no mesmo instante em que o destino está pronto para receber a informação.

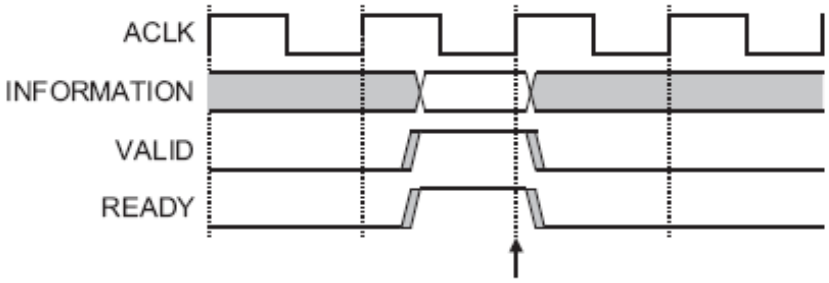


Figura 3.2: Protocolo AMBA – READY e VALID ativos

Novamente, utilizando-se o dc_shell, calculou-se a potência consumida e a área ocupada pelo logaritmo utilizando-se a mesma tecnologia de 28nm e um clock de 5MHZ como parâmetros nas simulações. Utilizou-se 4 bits para a parte inteira (M=4) do argumento e 4 bits para a parte fracionária (N=4) e o módulo teve um consumo total de 3.9715mW. A área ocupada no chip foi de aproximadamente 538.4512 μm^2 .

Por meio da ferramenta RTL Viewer do Quartus II, foi sintetizado o circuito o circuito do logaritmo foi sintetizado e a Figura 3.3 ilustra o diagrama do circuito para M=4 e N=4.

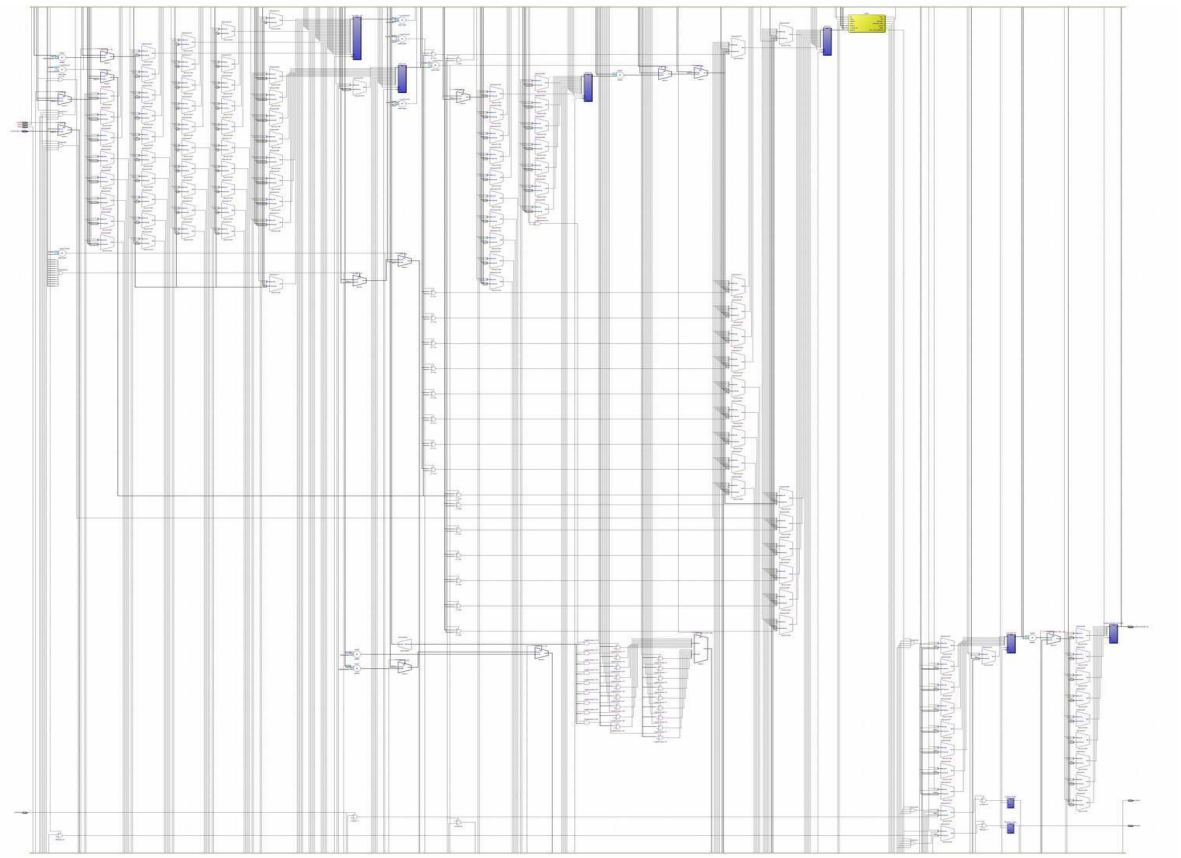


Figura 3.3: Sintetização do Logaritmo (M = 4bits, N = 4 bits)

A ferramenta do Quartus II também gerou a máquina de estados do Logaritmo. Na Figura 3.4 está apresentada a Máquina de Estados Finita que executa os passos do algoritmo.

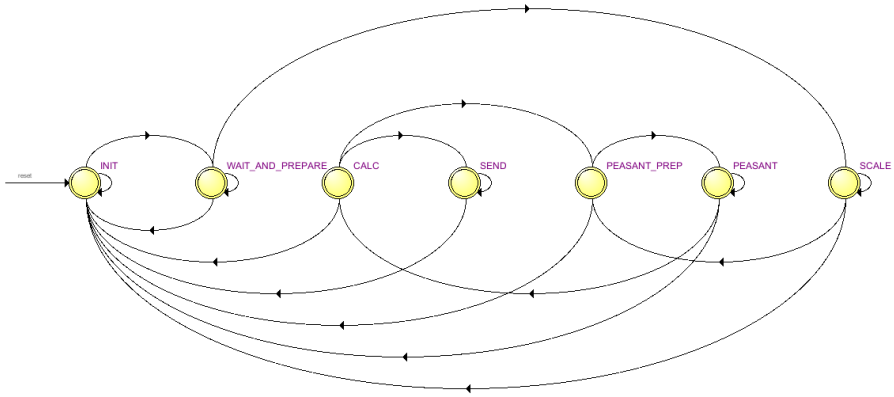


Figura 3.4: Máquina de Estados Finita do Logaritmo

A descrição de cada estado se dá a seguir:

- INIT: reseta todos os registradores para os valores default. Vai para o estado WAIT_AND_PREPARE;
- WAIT_AND_PREPARE: espera o argumento do Logaritmo e prepara o número para ser escalado entre 1 e 2. Vai para o estado SCALE;
- SCALE: escala o argumento entre 1 e 2. Vai para o estado PEASANT_PREPARE;

- PEASANT_PREPARE: variáveis auxiliares recebem o argumento escalado para fazer a mutliplicação pelo Peasant. Vai para o estado PEASANT;
- PEASANT: eleva o argumento escalado ao quadrado. Vai para o estado CALC;
- CALC: efetua o cálculo final do Logaritmo. Vai para o estado SEND;
- SEND: envia o resultado via protocolo AMBA. Volta para o estado INIT.

4 CONCLUSÃO

O presente trabalho teve como objetivo a implementação em hardware de um Logaritmo para o MFCC, que é parte integrante de um sistema de reconhecimento de locutor. Ao término desta etapa de minha formação profissional, foram abordados os realizados os seguintes pontos:

- O estudo e embasamento teórico de Processamento Digital de Sinais de Voz para um sólido entendimento do estado da arte;
- Estudo de técnicas de implementação de algoritmos em Hardware, bem como estudo de métodos de eficiência de algoritmos;
- Utilização de ferramentas de Hardware Design para estimação de consumo de energia e área ocupada em um chip;
- Estudo e utilização de protocolos de comunicação entre dispositivos;
- Trabalho em equipe, uma vez que este projeto forneceu um IP-Core para um aluno de Mestrado no âmbito do Brazil-IP.

5 BIBLIOGRAFIA

CAMPBELL, J. P. Speaker Recognition: A Tutorial. Proceedings of the IEEE, v.85, n. 9, p. 1437-1462, set. 1997.

CARDOSO, D. P. Identificação de Locutor usando Modelos de Misturas Gaussianas. 88p. Dissertação (Mestrado em Engenharia) – Departamento de Engenharia de Sistemas Eletrônicos, Universidade de São Paulo, São Paulo, 2009.

CIPRIANO, J. L. G. Desenvolvimento de Arquitetura para Sistemas de Reconhecimento Automático de Voz Baseados em Modelos Ocultos de Markov. 125 p. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. 2001.

CIPRIANO, J. L. G.; NUNES, R. P.; BARONE, D. A. C. Implementation of Voice Processing Algorithms in FPGA Hardware. IX WORKSHOP IBERCHIP, La Habana. Anais... La Habana: IBERCHIP, 2003.

DA SILVA, A. G. Reconhecimento de Voz para Palavras Isoladas. 2009, 60p. Estudo de Conclusão de Curso (Graduação em Engenharia da Computação) -Centro de Informática, Universidade Federal de Pernambuco, Recife. 2009.

FECHINE, J. Reconhecimento automático de identidade vocal utilizando modelagem híbrida:Paramétrica e Estatística. 2000, 212p. Tese (Doutorado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Campina Grande, Campina Grande. 2000.

LÉLIS DE MELO, Fabrício G. Avaliação de coeficientes Mel-Cepstrais na Representação das Características Vocais de um Locutor (Dissertação de Mestrado em Ciência da Computação), Departamento de Sistemas e Computação, Universidade Federal de Campina Grande, Campina Grande, 2015.

LYONS, Richard G. Streamlining Digital Signal Processing: A Tricks of the Trade Guidebook IEEE Press, 2012.

O'SHAUGHNESSY, D. Speech Communications: Human and Machine. 2. ed. Wiley-IEEE Press, 548p, 2000

O'SHAUGHNESSY, D. Interacting with computers by voice: automatic speech recognition and synthesis. Proceedings of IEEE, 2003 v.91 n.9, p.1272-1305.

RABINER, L.; SCHAFER, R. Digital processing of speech signals. New York: Prentice Hall, 512p, 1978.

RABINER, L.; JUANG, B. Fundamentals of Speech Recognition. New York: Prentice Hall, 1993. 507 p.

RABINER, L.; SCHAFER, R. Theory and Applications of Digital Speech Processing. Prentice Hall, 1056p, 2010.

SAAMBHAVI, V.; RAO, S.; RAJALAKSHMI, P. Design of feature extraction circuit for speech recognition applications. TENCON 2012 - 2012 IEEE REGION 10 CONFERENCE, 2012, Cebu. Anais... Cabu: IEEE, p. 1-5, 2012.

SLANEY, M. Auditory Toolbox. Palo Alto: Interval Research Corporation. Disponível em: <https://engineering.purdue.edu/~malcolm/interval/1998-010/>. , 1998.

http://en.wikipedia.org/wiki/Ancient_Egyptian_multiplication (acesso em dezembro de 2014)

6 ANEXO A – ÁREA E CONSUMO DO MULTIPLICADOR

```
Loading db file
'/usr/local/cmos28fdsoi_24/C28SOI_SC_8_CORE_LL@2.1@20131011.0/libs/C28SOI_SC_8_CORE_LL_tt28_0.80V_0.00
V_0.00V_0.00V_25C.db'
    Information: Updating design information... (UID-85)
    Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
    Warning: Design has unannotated primary inputs. (PWR-414)
    Warning: Design has unannotated sequential cell outputs. (PWR-415)


    ****************************************
    Report : power
        -analysis_effort low
    Design : multiplier
    Version: H-2013.03-SP3
    Date   : Sun Mar 22 14:52:45 2015
    ****************************************


    Library(s) Used:

      C28SOI_SC_8_CORE_LL (File:
/usr/local/cmos28fdsoi_24/C28SOI_SC_8_CORE_LL@2.1@20131011.0/libs/C28SOI_SC_8_CORE_LL_tt28_0.80V_0.00V_0.00
V_0.00V_25C.db)


    Operating Conditions: tt28_0.80V_0.00V_0.00V_0.00V_25C   Library: C28SOI_SC_8_CORE_LL
    Wire Load Model Mode: enclosed

    Design       Wire Load Model          Library
    ------------------------------------------------
    peasant              wl_zero          C28SOI_SC_8_CORE_LL
    peasant_DW_mult_uns_0  wl_zero          C28SOI_SC_8_CORE_LL


    Global Operating Voltage = 0.8
    Power-specific unit information :
      Voltage Units = 1V
      Capacitance Units = 1.000000pf
      Time Units = 1ns
      Dynamic Power Units = 1mW    (derived from V,C,T units)
      Leakage Power Units = 1mW


   Cell Internal Power  = 491.2795 nW   (91%)
   Net Switching Power  =  48.0510 nW    (9%)
                         ---------
```

```
Total Dynamic Power    = 539.3305 nW  (100%)

Cell Leakage Power     =   1.1829 uW


                 Internal         Switching         Leakage           Total
Power Group      Power            Power             Power             Power   (  %    ) Attrs
--------------------------------------------------------------------------------------------------
io_pad           0.0000           0.0000            0.0000            0.0000  (  0.00%)
memory           0.0000           0.0000            0.0000            0.0000  (  0.00%)
black_box        0.0000           0.0000            0.0000            0.0000  (  0.00%)
clock_network    0.0000           0.0000            0.0000            0.0000  (  0.00%)
register         4.5496e-04       4.0526e-06        4.3544e-04        8.9445e-04  ( 51.93%)
sequential       0.0000           0.0000            0.0000            0.0000  (  0.00%)
combinational    3.6324e-05       4.3998e-05        7.4748e-04        8.2781e-04  ( 48.07%)
--------------------------------------------------------------------------------------------------
Total            4.9128e-04 mW    4.8051e-05 mW     1.1829e-03 mW     1.7223e-03 mW
1
****************************************
Report : area
Design : multiplier
Version: H-2013.03-SP3
Date   : Sun Mar 22 14:54:26 2015
****************************************

Library(s) Used:

    C28SOI_SC_8_CORE_LL (File:
/usr/local/cmos28fdsoi_24/C28SOI_SC_8_CORE_LL@2.1@20131011.0/libs/C28SOI_SC_8_CORE_LL_tt28_0.80V_0.00V_0.00
V_0.00V_25C.db)

Number of ports:                38
Number of nets:                167
Number of cells:               117
Number of combinational cells:          80
Number of sequential cells:             36
Number of macros/black boxes:            0
Number of buf/inv:                      35
Number of references:                   13

Combinational area:            186.918408
Buf/Inv area:                  11.097600
Noncombinational area:          86.169600
Macro/Black Box area:           0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:               273.088008
Total area:                 undefined
```

7 ANEXO B – ÁREA E CONSUMO DO PEASANT

```
Loading db file
'/usr/local/cmos28fdsoi_24/C28SOI_SC_8_CORE_LL@2.1@20131011.0/libs/C28SOI_SC_8_CORE_LL_tt28_0.80V_0.00V_0.00
V_0.00V_25C.db'
        Information: Updating design information... (UID-85)
        Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
        Warning: Design has unannotated primary inputs. (PWR-414)
        Warning: Design has unannotated sequential cell outputs. (PWR-415)


        ****************************************
        Report : power
              -analysis_effort low
        Design : peasant
        Version: H-2013.03-SP3
        Date   : Wed Mar 18 17:15:15 2015
        ****************************************


        Library(s) Used:

           C28SOI_SC_8_CORE_LL (File:
/usr/local/cmos28fdsoi_24/C28SOI_SC_8_CORE_LL@2.1@20131011.0/libs/C28SOI_SC_8_CORE_LL_tt28_0.80V_0.00V_0.00
V_0.00V_25C.db)


        Operating Conditions: tt28_0.80V_0.00V_0.00V_0.00V_25C   Library: C28SOI_SC_8_CORE_LL
        Wire Load Model Mode: enclosed

        Design       Wire Load Model          Library
        ------------------------------------------------
        peasant              wl_zero         C28SOI_SC_8_CORE_LL
        peasant_DW01_add_0   wl_zero         C28SOI_SC_8_CORE_LL


        Global Operating Voltage = 0.8
        Power-specific unit information :
          Voltage Units = 1V
          Capacitance Units = 1.000000pf
          Time Units = 1ns
          Dynamic Power Units = 1mW    (derived from V,C,T units)
          Leakage Power Units = 1mW


         Cell Internal Power  = 681.7836 nW   (94%)
         Net Switching Power  =  42.6859 nW    (6%)
```

```
                    ---------
Total Dynamic Power    = 724.4695 nW  (100%)

Cell Leakage Power     =   1.0113 uW


                Internal         Switching         Leakage           Total
Power Group     Power            Power             Power             Power   (  %    ) Attrs
--------------------------------------------------------------------------------------------------
io_pad            0.0000           0.0000          0.0000          0.0000  (   0.00%)
memory            0.0000           0.0000          0.0000          0.0000  (   0.00%)
black_box         0.0000           0.0000          0.0000          0.0000  (   0.00%)
clock_network     0.0000           0.0000          0.0000          0.0000  (   0.00%)
register       6.5530e-04       8.8033e-06       6.2851e-04       1.2926e-03  (  74.47%)
sequential        0.0000           0.0000          0.0000          0.0000  (   0.00%)
combinational  2.6481e-05       3.3883e-05       3.8283e-04       4.4319e-04  (  25.53%)
--------------------------------------------------------------------------------------------------
Total          6.8178e-04 mW     4.2686e-05 mW     1.0113e-03 mW     1.7358e-03 mW
1


****************************************
Report : area
Design : peasant
Version: H-2013.03-SP3
Date   : Wed Mar 18 17:16:00 2015
****************************************

Library(s) Used:

   C28SOI_SC_8_CORE_LL
(File:/usr/local/cmos28fdsoi_24/C28SOI_SC_8_CORE_LL@2.1@20131011.0/libs/C28SOI_SC_8_CORE_LL_tt28_0.8
0V_0.00V_0.00V_0.00V_25C.db)

Number of ports:                        38
Number of nets:                        246
Number of cells:                       211
Number of combinational cells:         159
Number of sequential cells:             51
Number of macros/black boxes:            0
Number of buf/inv:                      58
Number of references:                   15

Combinational area:              94.656004
Buf/Inv area:                    13.273600
Noncombinational area:          122.073599
Macro/Black Box area:            0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                216.729603
Total area:                 undefined
```

1

8 ANEXO C – ÁREA E CONSUMO DO LOGARITMO

```
Loading db file '
/usr/local/cmos28fdsoi_24/C28SOI_SC_8_CORE_LL@2.1@20131011.0/libs/C28SOI_SC_8_CORE_LL_tt28_0.80V_0
.00V_0.00V_0.00V_25C.db'
Information: Updating design information... (UID-85)
Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
Warning: Design has unannotated primary inputs. (PWR-414)
Warning: Design has unannotated sequential cell outputs. (PWR-415)


****************************************
Report : power
        -analysis_effort low
Design : log
Version: H-2013.03-SP3
Date   : Sun Mar 22 12:32:54 2015
****************************************


Library(s) Used:

    C28SOI_SC_8_CORE_LL
(File:/usr/local/cmos28fdsoi_24/C28SOI_SC_8_CORE_LL@2.1@20131011.0/libs/C28SOI_SC_8_CORE_LL_tt28_0.8
0V_0.00V_0.00V_0.00V_25C.db)


Operating Conditions: tt28_0.80V_0.00V_0.00V_0.00V_25C   Library: C28SOI_SC_8_CORE_LL
Wire Load Model Mode: enclosed

Design        Wire Load Model           Library
------------------------------------------------
log                 wl_zero          C28SOI_SC_8_CORE_LL
log_DW01_add_0        wl_zero          C28SOI_SC_8_CORE_LL
log_DW01_dec_0        wl_zero          C28SOI_SC_8_CORE_LL
log_DW01_inc_0        wl_zero          C28SOI_SC_8_CORE_LL
log_DW01_add_1        wl_zero          C28SOI_SC_8_CORE_LL


Global Operating Voltage = 0.8
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1mW
```

```
 Cell Internal Power  =   1.3181 uW   (96%)
 Net Switching Power  =  53.8053 nW    (4%)
                         ---------
Total Dynamic Power    =   1.3719 uW  (100%)

Cell Leakage Power     =   2.5997 uW


                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (  %    ) Attrs
--------------------------------------------------------------------------------------------------
io_pad           0.0000           0.0000           0.0000           0.0000  (   0.00%)
memory           0.0000           0.0000           0.0000           0.0000  (   0.00%)
black_box        0.0000           0.0000           0.0000           0.0000  (   0.00%)
clock_network    0.0000           0.0000           0.0000           0.0000  (   0.00%)
register         1.2855e-03       1.2866e-05       1.3065e-03       2.6048e-03  (  65.59%)
sequential       0.0000           0.0000           0.0000           0.0000  (   0.00%)
combinational    3.2608e-05       4.0939e-05       1.2932e-03       1.3667e-03  (  34.41%)
--------------------------------------------------------------------------------------------------
Total            1.3181e-03 mW    5.3805e-05 mW    2.5997e-03 mW    3.9715e-03 mW
1
****************************************
Report : area
Design : log
Version: H-2013.03-SP3
Date   : Sun Mar 22 12:34:22 2015
****************************************

Library(s) Used:

    C28SOI_SC_8_CORE_LL
(File:/usr/local/cmos28fdsoi_24/C28SOI_SC_8_CORE_LL@2.1@20131011.0/libs/C28SOI_SC_8_CORE_LL_tt28_0.8
0V_0.00V_0.00V_0.00V_25C.db)

Number of ports:                        24
Number of nets:                        589
Number of cells:                       511
Number of combinational cells:         401
Number of sequential cells:            106
Number of macros/black boxes:            0
Number of buf/inv:                     107
Number of references:                   36

Combinational area:             284.729610
Buf/Inv area:                    28.505600
Noncombinational area:          253.721599
Macro/Black Box area:             0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                538.451208
```

Total area: undefined

1