VI CONGRESSO DE INICIAÇÃO CIENTÍFICA DA UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

PIBIC/CNPq/UFCG-2009

DETECÇÃO DE INÍCIO-FIM DE ELOCUÇÕES PARA VERIFICAÇÃO DE LOCUTOR EM SISTEMAS EMBARCADOS

João Viniccius Gomes Alves[1], Elmar U. K. Melcher[2], Joseana Macêdo Fechine[3]

RESUMO

O trabalho proposto visa à obtenção e implementação em hardware de um detector de início/fim (*endpoints*) de elocuções a ser aplicado na verificação de locutor em sistemas embarcados. A importância deste estudo consiste em determinar, a partir de parâmetros temporais da voz, o início e o fim de uma elocução, excluindo os intervalos de silêncio e o ruído nela presentes. Dessa forma, é possível diminuir a quantidade de sinal a ser processado e, consequentemente, os custos com memória e tempo de processamento para extração das características relevantes para o processo de verificação de locutor. Para a implementação em hardware do detector, utilizou-se o dispositivo programável (FPGA), com o auxílio da metodologia de concepção de IP-Core denominada ipPROCESS.

Palavras-Chave: Sistema Embarcado, Detecção de *Endpoints,* Energia do Sinal de Voz.

Abstract

The proposed work aims to achieve the hardware implementation of a detector to start / end (endpoints) from utterance to be applied in the speaker verification of embedded systems. The importance of this study is to determine, from time parameters of the voice, the beginning and end of an utterance, excluding intervals of silence and noise present in it. Thus, it is possible to decrease the amount of signal being processed and therefore the cost of memory and processing time for extraction of features relevant to the process of speaker verification. For implementation in hardware of the detector was used the programmable device (FPGA), using the methodology of design of IP-Core called ipPROCESS.

Keywords: FPGA, Endpoints detection, Energy

INTRODUÇÃO

A verificação da identidade vocal de um locutor permite a confirmação ou não da sua identidade, feita por um computador, o que possibilita tornar a relação homem-máquina mais efetiva e natural. Normalmente, ela é utilizada para acesso a informações restritas, em transações pessoais e em segurança de computadores e redes de comunicações (LI, 2000). Com o desenvolvimento crescente da tecnologia de sistemas embarcados, esses dispositivos estão se inserindo cada vez mais na vida diária das pessoas, mudando o cotidiano e o seu modo de vida. Desse modo, é natural a tendência de desenvolvimento da área de verificação

[1] Aluno do Curso de Engenharia Elétrica, Depto. de Engenharia Elétrica, UFCG, Campina Grande, PB, E-mail: joao.gomes@ee.ufcg.edu.br
[2] Ciência da Computação, Prof. Doutor, Depto. de Sistemas e Computação, UFCG, Campina Grande, PB, E-mail: elmar@dsc.ufcg.edu.br
[3] Ciência da Computação, Profa. Doutora, Depto. de Sistemas e Computação, UFCG, Campina Grande, PB, E-mail:joseana@dsc.ufcg.edu.br

de locutor em sistemas embarcados (KE et al., 2008).

Apesar disso, a maioria dos sistemas de verificação de locutor atuais ainda é baseada em softwares para computadores. Porém, devido ao crescente interesse por sistemas embarcados, pesquisas têm sido feitas na área. Entretanto, poucos são os sistemas capazes de realizar esta tarefa em tempo real (KE et al., 2008).

No contexto dos sistemas embarcados, em aplicações voltadas para a verificação de locutor em tempo real, torna-se imperativo a redução dos intervalos de silêncio da voz (início e fim da elocução), visando minimizar a quantidade de informação a ser processada, bem como minimizar as informações desnecessárias presentes nesses intervalos (ruído). Logo, este trabalho visa a desenvolver um detector de *endpoints*, ou seja, localizar o início e fim das elocuções, que atenda aos requisitos de simplicidade e robustez, realizando a detecção de maneira síncrona com a elocução da voz, utilizando, para tanto, a energia do sinal.

Este trabalho é parte integrante do Projeto *Brazil*-IP, da Universidade Federal de Campina Grande, financiado pelo Governo Federal, intitulado *Speaker Verification*.

Para a implementação do protótipo em hardware do sistema proposto, será utilizado a tecnologia FPGA (*Field Programmable Gate Array*). Esta plataforma se aproveita de um grau de paralelismo igual a um ASIC (*Application Specific Integrated Circuit*), e ao mesmo tempo proporciona flexibilidade e adaptabilidade ao projeto (ALTERA, 2009).

A metodologia de verificação funcional utilizada permitirá o acompanhamento do fluxo de projeto, de forma que o *testbench* (ambiente de simulação) seja gerado antes da implementação do dispositivo a ser verificado *(Design Under Verification* – DUV*),* tornando o processo de verificação funcional mais rápido e o *testbench* confiável, devido ao fato de ele ser verificado antes do início da verificação funcional do DUV. Esta metodologia é baseada no VeriSC (SILVA, 2007) e na OVM (Open Verification Methodology) (OVM, 2009) e é chamada BVM (Brazil-IP Verification Methodology). Além disso, para o auxílio na prototipagem em FPGA do sistema proposto, utiliza-se o ipPROCESS (LIMA et al., 2005).

Conceitos Básicos de Processamento Digital de Sinais de Voz

Uma forma eficaz, simples e natural do ser humano expressar seus pensamentos, opiniões e trocar ideias é a voz. Esta possibilita que mãos e olhos estejam disponíveis para outras atividades, o que traz consigo vantagens, tais como mobilidade (MARTINS, 1997). Usando a voz, o ser humano pode externar mais palavras por minuto do que digitando num teclado (COSTA, 1994).

A voz carrega consigo peculiaridades únicas a respeito da identidade de cada indivíduo, tais como informações do seu grupo sócio-cultural, a idade, o sexo, o estado emocional, a região de onde reside (a partir do sotaque), a língua que está sendo falada, entre outras características (FLANAGAN, 1978).

A voz é uma característica física ou comportamental mensurável utilizada para reconhecer ou verificar a identidade invocada de um usuário registrado, o que a classifica como uma característica biométrica do ser humano. Além disso, a voz obedece às seguintes exigências de uma informação biométrica: aceitabilidade, exclusividade, universalidade e coletabilidade (KOERICH, 2004).

Com esses atributos, torna-se claro que a partir do sinal de voz pode-se distinguir as características de cada pessoa. Em função das suas vantagens, existe o interesse econômico na implementação de sistemas baseados em voz, nos casos em que confiabilidade, facilidade de uso e custo são relevantes.

Diante de tal interesse, pesquisas na área de Processamento Digital de Sinais de Voz (PDSV) buscam o desenvolvimento de técnicas que possibilitem a construção de padrões, a partir da fala que permitam modelar de forma eficiente as características vocais únicas de cada locutor, por exemplo, (KLEIJN et al, 1998; De LIMA et al., 2000; BENZEGHIBA et al., 2003; Da CUNHA et al., 2003; LEMMETTY, 2004).

Na Figura 1, é apresentado uma descrição geral do processamento da voz, com ênfase na verificação de locutor – objeto de estudo deste trabalho (CAMPBELL, 1997).

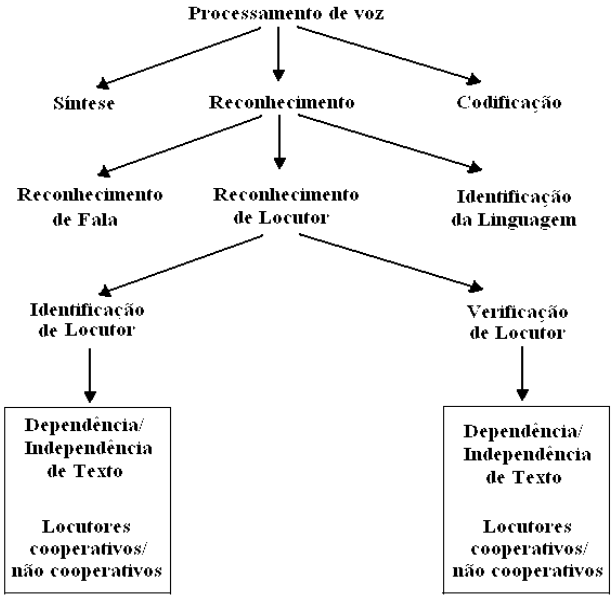

Figura 1. Descrição Geral do Processamento de Voz (Adaptado de (CAMPBELL, 1997)).

Nos sistemas de reconhecimento de locutor, determina-se a identidade de uma pessoa a partir da sua voz, com o propósito de restringir o acesso a redes, computadores, base de dados e ambientes, bem como a restrição de informações confidencias às pessoas não autorizadas, dentre outras aplicações.

A implementação de um sistema de reconhecimento de locutor é feita a partir da obtenção, para cada locutor, de um conjunto de parâmetros representativos da sua voz. Os parâmetros obtidos irão compor um modelo (ou padrão) representativo do locutor. Dessa forma, o locutor será aceito ou rejeitado, com base na comparação dos seus parâmetros (padrão) de teste com os parâmetros já armazenados (padrões de referência), utilizando-se uma regra de decisão (RABINER et al., 1978).

Tendo como entrada um sinal de voz, o objetivo da tarefa de reconhecimento de locutor é identificar a pessoa mais provável de ser o locutor, dentre uma população definida – Identificação de Locutor, ou verificar se o locutor é quem ele alega ser – Verificação de Locutor (RABINER et al., 1978). Sendo assim, os sistemas de verificação fazem a comparação com um único padrão pré-estabelecido, enquanto que os de identificação, com todos os padrões pré-estabelecidos.

Na verificação de locutor, também conhecida como autenticação de locutor, verificação de voz ou autenticação de voz (CAMPBELL, 1997), deve-se decidir se um dado locutor é a pessoa que ele alega ser, isto é, uma identidade é alegada pelo usuário e a decisão exigida pelo sistema é binária, ou seja, apenas aceita ou rejeita a identidade alegada (RABINER et al., 1978).

Quanto à identificação de locutor, o sistema é requisitado a fazer uma identificação dentre todos os locutores. Então, em substituição a uma única comparação entre um conjunto de medidas e um padrão de referência armazenado, faz-se necessário um número de comparações igual ao número de locutores. Além disso, a identificação pode ser feita de dois modos: conjunto-aberto (o locutor pode não estar na população) e conjunto fechado (sabe-se de antemão que o locutor pertence à população) (RABINER et al., 1978).

O reconhecimento de locutor também pode ser classificado como dependente ou independente de texto. A dependência de texto requer que o locutor pronuncie uma frase ou uma dada senha pré-determinada, enquanto que o sistema independente de texto não exige o caso anterior. De acordo com a área de aplicação do sistema, a dependência ou a independência torna-se crucial. Por exemplo, na área de criminalística, tem-se maior interesse em sistemas independentes de texto, pois na maioria das aplicações os locutores a serem identificados são não cooperativos. Já em aplicações que envolvem acesso a ambientes restritos, torna-se mais adequado o uso de sistemas dependentes de texto, já que neste caso os locutores são cooperativos (FECHINE, 2000).

O contexto do trabalho ora descrito está voltado para a verificação de locutor dependente do texto.

A verificação de locutor consiste em uma tarefa de reconhecimento de padrões da voz e, para tanto, é dividida em duas fases: treinamento e verificação (vide Figura 2).

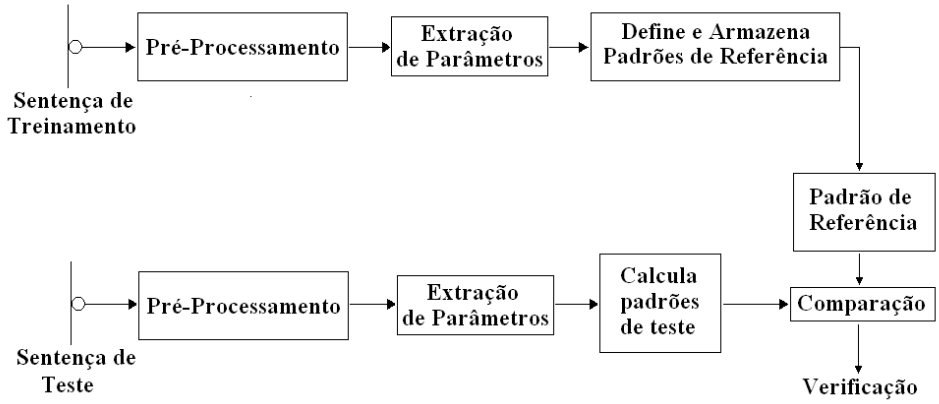

Figura 2. Sistema de Verificação de Locutor (Adaptado de (RABINER et al., 1993))

Na fase de treinamento, a partir das sentenças de treinamento, é realizado o pré-processamento do sinal, seguido da extração dos parâmetros representativos e, a partir destes, são obtidos os padrões de referência que representam os padrões únicos de um dado locutor. Na fase de verificação (reconhecimento ou teste), também é realizado o pré-processamento e extração de parâmetros representativos do locutor. Para tanto, são utilizadas as mesmas técnicas da fase de treinamento. Por fim, são obtidos os padrões de teste do locutor os quais são comparados com os padrões de referência (previamente armazenados) e, a partir de uma lógica de decisão, o locutor é considerado verdadeiro (aceito) ou falso (rejeitado) (CAMPBELL, 1997).

Fatores externos podem contribuir para erros em um sistema de verificação de locutor. Esses erros estão ligados a fatores humanos e ao ambiente, tais como: erro de elocução ou de leitura das frases pré-definidas, estado emocional do locutor, variação da posição do microfone (intra ou inter-sessões), ambiente acústico pobre ou inconsistente (ruído), erro de “casamento” do canal (microfones diferentes para treinamento e teste), problemas de saúde (um simples resfriado pode alterar as características do trato vocal) e idade (a forma do trato vocal pode ser alterada com a idade) (FECHINE, 2000; DIAS, 2000; RABINER et al., 1978; CAMPBELL 1997).

Essas fontes de erro externas geralmente não podem ser eliminadas, logo precisam ser modeladas. Para tanto, tem-se a,etapa de pré-processamento. Este passo é responsável pelo tratamento do sinal de voz com relação ao ambiente de gravação e ao canal de comunicação utilizados e tem como tarefa a redução de efeitos indesejáveis incorporados ou presentes no sinal de voz, além de prepará-lo para as etapas seguintes do processo de verificação (RABINER et al., 1978; FURUI, 1981; SHAUGHNESSY, 2000).

Na Figura 3 são apresentadas três técnicas básicas presentes na etapa de pré-processamento do sinal, sendo estas: detecção de *endpoints*, pré-ênfase e janelamento. A detecção de *endpoints* visa à eliminação dos intervalos de silêncio (ou de ruído) presentes no início e fim do sinal de voz, bem como entre as palavras que compõem a elocução. O uso dessa técnica proporciona vantagens como a redução do tempo de processamento, já que apenas o sinal de voz será processado pelo sistema de verificação (OLIVEIRA, 2001) e a melhoria no desempenho da verificação provocada pela exclusão do ruído de fundo existente antes e depois da gravação do sinal de voz (RABINER et al., 1993).

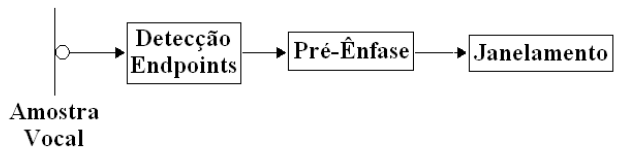

Figura 3. Etapas constituintes do pré-processamento de voz (Adaptado de (SILVA, 2006))

O Mecanismo de Produção da Voz

Uma mensagem produzida por um locutor é um sinal de voz composto por uma sequência de sons que servem como uma representação simbólica, cuja composição é determinada pelas regras de linguagem. A forma como estas regras são utilizadas e o seu estudo científico é denominado *Linguística*. As características da produção do som pelo ser humano, especialmente para a descrição, classificação e transcrição da voz, é objeto de estudo da ciência denominada *Fonética* (RABINER et al.,1978).

A voz emitida é resultante de várias transformações ocorridas em diferentes estágios: semântico, linguístico, articulatório e acústico. As diferenças oriundas destas transformações resultam em propriedades acústicas diferentes do sinal de voz, diferenças estas utilizadas, na tarefa de verificação de locutor, para a discriminação de locutores entre si. Estas distinções entre locutores são um resultado conjunto das particularidades inerentes ao trato vocal (características inerentes) e daquelas relacionadas ao movimento dinâmico do trato vocal, ou seja, a forma como o indivíduo fala (características instruídas) (CAMPBELL, 1997).

A fim de gerar o som desejado, o locutor exerce uma série de controles sobre o aparelho fonador, apresentado na Figura 4, produzindo a configuração articulatória e a excitação apropriadas. Na Figura 4, tem-se os componentes importantes do sistema vocal humano, dentre eles: o trato vocal, nome genérico dado ao conjunto de cavidades e estruturas que participam da produção sonora, que tem como limite inferior a região glótica e superior, os lábios; o trato nasal, responsável pela produção dos sons nasais, que começa na úvula ou véu platino e termina nas narinas (RABINER et al., 1978).

A fala é produzida durante a fase de exalação do ar, após a inalação deste nos pulmões. Este fluxo de ar, depois da vibração das cordas vocais situadas na laringe, excita o trato vocal, o que produz os chamados sons sonoros. Para a produção de sons nasalados, a úvula se abre permitindo a passagem do ar pelo trato nasal e a sua radiação pelas narinas (RABINER et al., 1978).

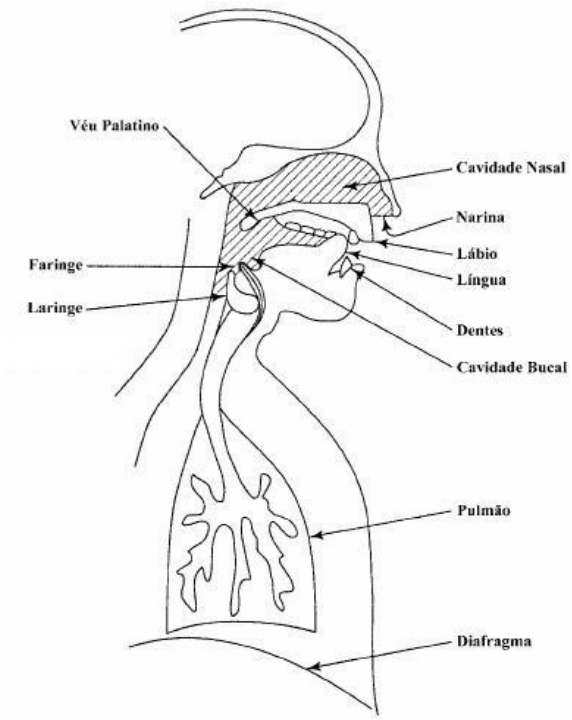

Figura 4. Anatomia do Aparelho Fonador (Adaptado de (DELLER Jr. et al., 1993)).

O trato vocal e o trato nasal são mostrados na Figura 5 como tubos cujas seções transversal são não uniforme. O som se propaga através destes tubos e o espectro de frequência é modelado pela seletividade em frequência do tubo. No ambiente de produção da voz, as frequências de ressonância do tubo do trato vocal são chamadas de frequências formantes ou *formantes.* Estas frequências dependem, primordialmente, da forma e dimensões do trato vocal, em que cada forma possui conjunto único de formantes. Como resultado desta característica, sons diferentes são obtidos de acordo com a forma assumida pelo trato vocal. Com isso, as propriedades espectrais do sinal de voz variam com o tempo e com a forma do

trato vocal (RABINER et al., 1978).

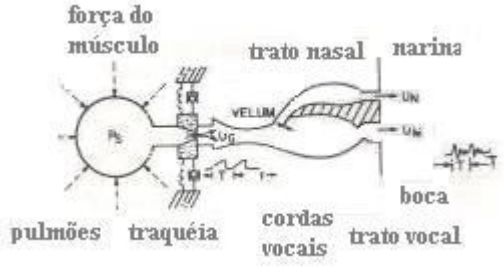

Figura 5. Diagrama do aparelho vocal (Adaptado de (RABINER et al., 1978).

Em decorrência das limitações dos órgãos humanos de produção de voz e o sistema auditivo, a comunicação humana típica está limitada na faixa de 7-8 kHz (RABINER et al.,1978; SHAUGHNESSY, 2000).

Modelo de Produção da Voz

Para a obtenção de um modelo detalhado do processo de produção da voz, os seguintes efeitos devem ser observados (RABINER et al., 1978):

1. Variação da configuração do trato vocal com o tempo;
2. Perdas próprias por condução de calor e fricção nas paredes do trato vocal;
3. A maciez das paredes do trato vocal;
4. Radiação do som pelos lábios;
5. Junção Nasal;
6. Excitação do som no trato vocal, etc.

Um modelo para a geração de sinais de voz, considerando os efeitos da propagação e da radiação conjuntamente, pode ser obtido a partir de valores adequados para a excitação e para os parâmetros do trato vocal. A teoria acústica apresenta uma técnica simplificada para a modelagem dos sinais de voz, a qual possui excitação separada do trato vocal e da radiação. Os efeitos da radiação e do trato vocal são representados por um sistema linear variante com o tempo. O gerador de excitação gera um sinal similar a um trem de pulsos glotais ou um sinal aleatório (ruído). Com isso, escolhem-se os parâmetros da fonte e do sistema para se obter na saída o sinal de voz desejado (RABINER et al., 1978). Com isso, obtém-se o modelo apresentado na Figura 6, em que $u(n)$ é o sinal de excitação e $As(n)$ e $Af(n)$ controlam a intensidade da excitação do sinal de voz e do ruído, respectivamente.

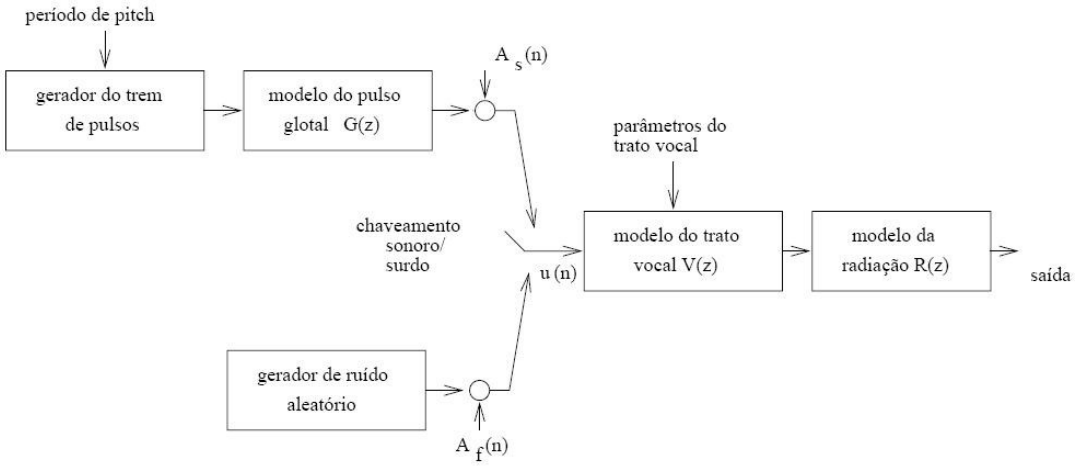

Figura 6. Modelo discreto de produção da fala (Adaptado de (RABINER et al., 1978)).

A partir da troca entre geradores de excitação sonora e não sonora, alterna-se o modo de excitação. Como trato vocal pode ser modelado em uma grande variedade de formas, obtém-se uma combinação conveniente do pulso glotal e modelos de radiação em um sistema simples. Em particular, para a análise por predição linear combina-se o pulso glotal, a radiação e os

componentes do trato vocal a fim de representá-los com uma simples função de transferência (RABINER et al., 1978):

$$H(z) = G(z)V(z)R(z) \quad (1),$$

em que:

$G(z)$ - Transformada-z do modelo do pulso glotal;

$V(z)$ - Transformada-z do modelo do trato vocal;

$R(z)$ - Transformada-z do modelo de radiação.

Esse modelo possui algumas limitações, dentre as quais a variação dos parâmetros com o tempo, a necessidade de espaçamento do pulso glotal por um múltiplo inteiro de período de amostragem, *T*. Porém, estas deficiências não limitam a aplicabilidade deste modelo (RABINER et al., 1978; VIEIRA, 1989).

Parâmetros Temporais do Sinal de Voz

A avaliação de um gráfico amplitude-*versus*-tempo de um sinal de voz permite a análise de características importantes que proporcionam uma descrição completa deste sinal. Dessa forma, a identificação de sons básicos da fala é permitida a partir de parâmetros temporais. A representação a partir destes é importante porque o processamento digital necessário é de implementação simples e, apesar desta simplicidade, proporciona uma boa estimativa das características do sinal de voz (RABINER et al., 1978). Dentre os parâmetros tradicionais para a análise de voz, destaca-se a Energia do Sinal.

Uma peculiaridade importante dos sinais de voz é a invariância de suas propriedades espectrais no tempo para curtos intervalos, sendo um valor típico 16 ms. Logo, com o objetivo de se obter os parâmetros espectrais do sinal, faz-se necessário o particionamento deste em segmentos (ou bloco de amostras), com o intuito de se trabalhar com o sinal dentro dos seus limites de estacionariedade (RABINER et al., 1978; VASSALI et al., 2000; DELLER Jr. et al., 1993; RABINER et al., 1993).

Energia do Sinal

A energia por segmento, $Eseg$, é obtida como

$$E_{seg} = Na.E\{[s(n) - \mu_{s(n)}]^2\} \quad (2)$$

A voz é considerada um sinal ergódico[4] e estacionário no sentido amplo[5], com média nula. Dessa forma, a Equação 2 é definida como (RABINER et al., 1978; VIEIRA, 1989):

$$E_{seg} = Na.E\{[s(n)]^2\} = \sum_{n=0}^{Na-1}[s(n)^2] \quad (3)$$

em que $s(n)$ é o sinal de voz, $\mu_{s(n)}$ a média de $s(n)$ e Na o tamanho da janela (bloco de amostras do sinal) em questão. Sendo assim, a energia é resultante da soma dos quadrados das amplitudes das Na amostras do sinal contido na janela em análise, devendo refletir as variações de amplitude do sinal entre intervalos ou janelas (RABINER et al., 1978; VIEIRA, 1989).

A energia do sinal está concentrada na região de baixas frequências do espectro, geralmente entre 500 e 800 Hz. Porém, isto não descarta as componentes de frequências mais altas, pois apesar de possuírem baixa energia, determinam a inteligibilidade da voz (FECHINE, 2000).

---

[4] Um processo estocástico ergódico possui as médias estatísticas iguais às médias temporais.

[5] Um processo estocástico estacionário no sentido amplo possui uma média constante e uma função de autocorrelação dependente apenas da diferença entre os intervalos de medição.

Hardware de um Sistema Embarcado

Um sistema embarcado é o resultado da combinação de *hardware* e *software*, e algumas vezes peças mecânicas, desenvolvido para realizar uma função específica (BARR, 1999; FRANCIA, 2001). Por meio do desenvolvimento em *hardware*, pode-se alcançar maior eficiência e rapidez na execução de determinadas tarefas e, a partir do *software*, é possível reduzir o tempo de desenvolvimento e aumentar a flexibilidade do sistema (EDWARDS et al., 1997).

Quanto à implementação em *hardware* de um sistema embutido e às aplicações deste, diferentes métodos podem ser usados. Para aplicações em grande escala, como para o mercado de consumo, o mais indicado é o uso de um SoC (*System-On-a-Chip*) (Carro et al., 2003). A arquitetura de *hardware* de um SoC pode conter alguns blocos dedicados. Estes blocos são chamados de *IP-core* (*Intellectual Property Cores*) e nada mais são do que componentes de *hardware* que desempenham tarefas específicas e são projetados de modo a permitir o seu reuso em diferentes sistemas (MORAES et al., 2004; SILVA, 2007). Outro ponto importante no desenvolvimento de *hardware* é o tempo necessário para a realização do projeto e validação individual de todos os componentes - blocos dedicados, processadores, entre outros -, além do tempo da validação do conjunto em um mesmo sistema (EDWARDS et al., 1997).

Para aplicações com menor volume de produção, o uso de FPGA (*Field-Programmable Gate Arrays*) é mais recomendável (SILVA, 2006). A principal vantagem no uso de FPGA é a possibilidade de modificação da estrutura de *hardware* do sistema por meio de um processo denominado reconfiguração, permitindo o desenvolvimento incremental, correção de erros de projeto, além da adição de novas funcionalidades ao *hardware* (MOARES et al, 2004).

No projeto de *hardware* de um sistema embarcado, as linguagens utilizadas são chamadas de linguagens de descrição de *hardware (HDLs – Hardware Description Language).* Estas auxiliam na tarefa de descrição dos circuitos eletrônicos, permitindo descrever a forma como os circuitos operam e também possibilitando a simulação destes circuitos antes mesmo de sua fabricação. A principal diferença entre uma linguagem de programação para software e uma HDL é que a sintaxe e a semântica desta incluem informações para expressar qual será o comportamento do *hardware* ao longo do tempo (CARRO et al., 2003).

A grande dificuldade na escolha de uma linguagem para a implementação de um sistema embutido de *hardware* reside no fato de que para cada uma das etapas da tarefa – mostradas a seguir -, níveis de abstração diferentes são exigidos. Por exemplo, a HDL chamada *Verilog*, apresenta a vantagem da sua utilização como entrada para simulação e síntese automática de circuitos descritos no nível de microarquitetura. Porém, a utilização desta mesma linguagem mostra-se ineficaz para descrição de *software* e especificações de alto nível (CARRO et al., 2003). Por outro lado, com a utilização de linguagens com um nível mais alto de abstração, como C++ ou Java, ocorre a inadequação destas para a descrição de hardware. Tendo em vista estas dificuldades, optou-se pelo uso de uma linguagem que combinasse as vantagens de uma linguagem como C++, quanto ao seu nível de abstração, com uma semântica adicional apropriada para a descrição de *hardware*. Esta linguagem denomina-se *SystemVerilog* (SYSTEMVERILOG, 2009).

Etapas de Desenvolvimento de um IP-Core

As etapas necessárias para o desenvolvimento de um sistema embarcado consistem em: especificação funcional, verificação funcional, construção do modelo RTL, síntese, simulação pós-síntese e prototipagem, como mostrado na Figura 7. A seguir, será descrita cada etapa.

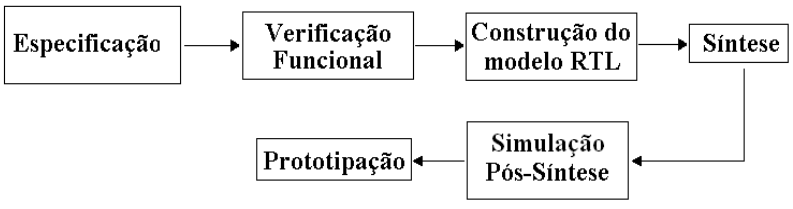

Figura 7. Etapas de desenvolvimento de um IP-Core (Adaptado de (SILVA, 2006)).

Especificação Funcional

Esta etapa consiste de uma descrição completa do que se deseja construir, ou seja, tem-se uma explicação das funcionalidades desejadas para o seu sistema de *hardware.*

Esta especificação é fundamental para o entendimento das necessidades do dispositivo a ser desenvolvido. Nesta fase, são estudados os requisitos, baseados na definição exata de cada funcionalidade que o *hardware* final deve executar. Por fim, é produzido um documento de texto com um alto nível de abstração contendo estes requisitos (SILVA, 2007). A metodologia seguida durante esta etapa é apresentada em (LIMA et al., 2005).

Verificação Funcional

Esta fase é a mais longa do fluxo de desenvolvimento de um sistema de *hardware* embarcado, consistindo em cerca de 70% de todos os recursos do projeto (BERGERON, 2003; PIZIALI, 2004). Porém, ela é crucial no sucesso de reuso do IP-*core* em desenvolvimento. De acordo com (BERGERON, 2005), verificação funcional é um processo usado para demonstrar que o objetivo do projeto é preservado em sua implementação. Ela deve ser feita a partir da comparação de dois modelos, o modelo em desenvolvimento e o modelo ideal, sem erros, que reflete a especificação, o Modelo de Referência (SILVA, 2007).

A verificação funcional é realizada a partir de simulações. Durante a simulação, são inseridos estímulos na entrada do DUV e esses estímulos são coletados em sua saída e comparados com os resultados esperados (ideais) oriundos do Modelo de Referência (SILVA, 2007).

Os estímulos a serem inseridos no DUV devem ser escolhidos de forma cuidadosa, de modo que este exerça as funcionalidades desejadas. Isto porque a etapa da verificação somente terminará quando todos os requisitos especificados forem executados. Estes requisitos compõem o chamado critério de cobertura. Dessa forma, a verificação estará concluída apenas quando todos os critérios de cobertura forem atingidos. Com isso, a verificação funcional deve obedecer ao processo chamado de verificação dirigida por cobertura, o qual se baseia na restrição dos estímulos de acordo com a cobertura e cuja determinação do término da simulação depende dos critérios desta (SILVA, 2007).

A metodologia de verificação funcional utilizada neste trabalho é denominada BVM. Derivada das metodologias OVM, a qual é baseada no padrão *SystemVerilog* do IEEE, e VeriSC (SILVA, 2007). A metodologia BVM permite o desenvolvimento de ambientes de verificação avançados, oferecendo altos níveis de integração e portabilidade.

Para este projeto, utilizou-se como ambiente de simulação a ferramenta eTBC (*Easy Testbench Creator*) (PESSOA, 2007), a qual é descrita a seguir.

eTBc

A ferramenta BVM consiste em um ambiente de simulação (*testbench*), o qual é composto de um *Source*, um *Monitor*, um *Driver*, o *Reference Model* e um *Checker*, tal como exposto na Figura 8. Os mesmos dados de entrada são enviados para o *Reference Model* e para o DUV (*Design Under Verification*). Então, as saídas do DUV e do *Reference Model* são coletadas e comparadas (PESSOA, 2007).

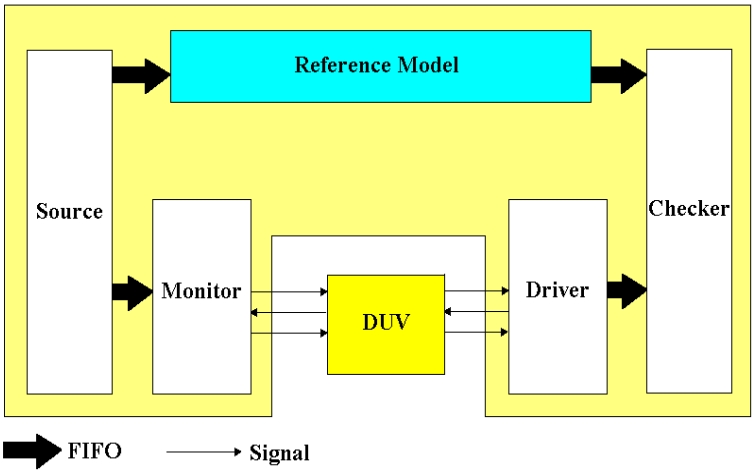

Figura 8. Ambiente de Verificação Funcional *(Testbench)* (Adaptado de (PESSOA, 2007)).

O *testbench* é um artefato escrito em linguagem formal, usado para criar simulações para o DUV, representado por uma linguagem de descrição de hardware, com o objetivo de conceber estímulos que consigam ativar as funcionalidades do DUV (BERGERON, 2003).

O Modelo de Referência (*Reference Model*) é um modelo ideal, isto é, isento de erros e já testado funcionalmente, que representa as funcionalidades que foram especificadas no projeto, enquanto o DUV é o projeto que está sendo verificado. O Modelo de Referência pode inclusive ser escrito em uma linguagem de alto nível, como C, C++, Java, Python, dentre outras (BERGERON, 2003).

O *testbench* é descrito em nível de transações, ou seja, dados que trafegam entre os módulos componentes do projeto e operações que ocorrem nas entradas e saídas de cada um deles, o qual é chamado de nível TLM – *Transaction Level Modeling* (CAL et al., 2003). O DUV está em um nível mais baixo de implementação, o nível RTL – *Register Transfer Level* (LAVAGNO et al., 2006), no qual o *IP-core* é visto a partir da transferência de dados entre os seus registradores. É necessário, portanto, que haja a conversão entre os dois níveis, responsabilidades do *Driver* e do *Monitor*. O primeiro converte as transações em TLM para sinais em nível RTL, enquanto que o segundo, de sinais novamente para transações. Além disso, tanto o Driver quanto o Monitor executam um protocolo de comunicação (*handshake*) com o DUV, através de sinais (PESSOA, 2007). Para o projeto proposto, o protocolo de comunicação seguido é AMBA AXI (AMBA AXI, 2009).

Alguns elementos da Figura 8 estão conectados por setas largas, as FIFO (*First In First Out*). As FIFO desempenham um papel importante no *testbench*, pois são responsáveis por controlar o sequenciamento e o sincronismo dos dados em transações (CAL et al., 2003).

O *Source* é responsável por criar estímulos, necessários para satisfazer os critérios de cobertura especificados durante a simulação. Todos os estímulos criados são transações e estas são enviadas, através de FIFO, diretamente ao Modelo de Referência e ao *Driver*. Os estímulos devem exercitar todas as funcionalidades especificadas, para saber se as respostas do DUV estão corretas (PESSOA, 2007).

Checker é o responsável por comparar as respostas provenientes do DUV e do Modelo de Referência para averiguar se são equivalentes (BERGERON, 2003).

Construção do Modelo RTL

O próximo passo nas etapas de desenvolvimento de um IP-core é a implementação RTL (*Register Transfer Level*). Esta consiste de um código escrito em um nível mais baixo de abstração, o nível de sinais. Nesse nível, todas as operações são controladas por ciclo de relógios *(clocks).* Além disso, este código descreve a especificação do projeto em termos do nível de fluxo de dados entre registradores, o que justifica o seu nome. Normalmente, este código é implementado em uma linguagem de descrição de hardware (SILVA, 2006; SILVA, 2007).

A implementação RTL será o modelo a ser convertido em hardware. Logo, um aspecto importante que deve ser respeitado é que os erros precisam ser captados ainda nesta etapa,

pois quanto mais cedo os erros forem detectados, menos recursos serão gastos na correção destes (LAVAGNO et al., 2006).

Síntese

A etapa de síntese é realizada a partir da conversão de uma descrição RTL em um conjunto de registradores e em lógica combinacional. Além disso, assim que a síntese do código RTL é concluída, gera-se uma *netlist* no nível de portas lógicas (SILVA, 2007).

Simulação pós-síntese

No passo seguinte, a simulação pós-síntese, aspectos específicos do dispositivo utilizado na prototipagem, no caso FPGA, do sistema, tal como atraso das portas lógicas, passam a ser importantes. Nesta fase, tanto aspectos de funcionalidade quanto de tempo são levados em consideração durante a simulação (SILVA, 2006; SILVA, 2007).

Esta etapa é essencial para determinar se os requisitos de tempo são respeitados e se pode ser obtido um desempenho melhor do dispositivo a partir de circuitos mais otimizados (SILVA, 2007).

Prototipação

Por fim, há a implementação do código gerado pela síntese, consistindo na implantação do *netlist* gerado pela síntese em algum dispositivo de hardware (SILVA, 2007).

MATERIAL E MÉTODOS

A metodologia utilizada é baseada em VeriSC (SILVA, 2007) e OVM (*Open Verification Methodology*), denominada BVM (*Brazil-IP Verification Methodology*). Para auxílio na prototipagem em FPGA do sistema proposto, é utilizada o ipPROCESS (LIMA et al., 2005).

O ipPROCESS é um processo de desenvolvimento para SOFT IP com prototipagem em FPGA. Este fornece um método disciplinado de atribuição de tarefas e responsabilidades dentro de uma organização de desenvolvimento (LIMA et al., 2005).

O OVM é uma metodologia e uma biblioteca de classes aberta e interoperável para verificação. Ela assegura a interoperabilidade entre simuladores, linguagens de alto nível e designers com ferramentas para definidas para o SystemVerilog (OVM, 2009).

Material

- Infra-estrutura de Hardware:
  - Microfone Leadership. Impedância: 1,4 KΩ; Frequência de Resposta: 50 KHz – 13 KHz; Sensibilidade: -58 dB +/- 3 dB; Tensão de Operação: 1V – 10 V; Consumo: 350 μW no máximo; Relação Sinal/Ruído: 40 dB ou superior; conector: 3,5 mm Stéreo Mini-Plug;
  - Microcomputador Core 2 Duo T5800 2,00 GHz. HD de 320 GB;
  - FPGA Altera.

- Infra-estrutura de Software:
  - Quartus 8.1;
  - eTBc 2.0 Beta;
  - GCC compiler 4.6;
  - irun 08.10-s004.

- Base de dados:

Sinais de áudio captados dentro do Laboratório de Arquiteturas Dedicadas (LAD), com o nível de ruído ambiente.

- Parâmetro temporal extraído do sinal de Voz: Energia

RESULTADOS E DISCUSSÃO

Algoritmo para detecção de *endpoints*

Após a digitalização do sinal de voz, o detector de *endpoints* segmenta este sinal em blocos de duração definida, em torno de 5 a 10 ms. Estes blocos são denominados *subframes*. O algoritmo proposto baseia-se na extração da energia de cada um destes segmentos. Foram utilizados dois limiares de energia, denominados *lowerEnergyThreshold* e *higherEnergyThreshold*, usados, respectivamente, no auxílio da determinação do fim e do início da atividade de voz, e em limiares (*thresholds*) temporais para determinar os pontos onde a atividade vocal começa e termina, respectivamente, denominados *startTimeThreshold* e *endTimeThreshold*. O fluxograma apresentado na Figura 9, retirado da Especificação Funcional do módulo *Voice Detector* (VD), componente do projeto *Speaker Verification*, representa o algoritmo concebido.

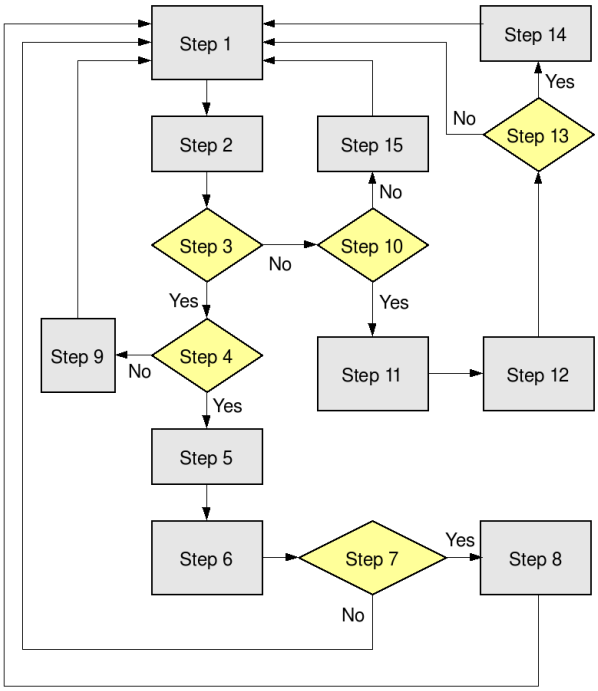

Figura 9. Detecção de *endpoints* do sistema.

No fluxograma, cada um dos passos utilizados pelo algoritmo é descrito como um *step.* O primeiro passo consiste no preenchimento de um *buffer* com todas as amostras do *subframe.* Com o *buffer* preenchido, avança-se ao *step 2,* no qual calcula-se a energia deste *subframe* por meio da Equação 3.

Em seguida, o objetivo consiste em procurar pelo início da atividade vocal, o que corresponde à pergunta do *step 3*. Para tal, caso a energia calculada seja maior do que o limiar estabelecido, *higherEnergyThreshold,* o que corresponde ao questionamento presente no *step 4,* conta-se o número de *subframes* consecutivos que possuem energia maior do que o *higherEnergyThreshold,* tarefa esta realizada pelo *step 5*. Caso contrário, o algoritmo interpreta o *subframe* atual como um falso início de atividade vocal, descarta as chances deste *subframe* corresponder ao início de voz, passo de número 9, e retorna ao passo inicial.

Uma vez no *step 5*, o algoritmo armazena o *subframe* atual em um vetor representando a atividade vocal efetiva, tarefa cumprida pelo *step 6*. Caso este número de *subframes* consecutivos que possuem energia maior do que o *higherEnergyThreshold* ultrapasse o limiar *startTimeThreshold,* pergunta presente no *step 7,* avança-se ao *step 8*. Caso contrário, retorna-se ao passo inicial para a análise de um novo *subframe*. Quando o algoritmo chegar ao passo de número 8, o início de voz foi encontrado e retorna-se ao passo inicial para a determinação do fim de atividade vocal.

Para um novo *subframe,* a resposta ao passo de número 3 será negativa, isto é, procura-se agora pelo fim de atividade vocal. Com isto, se chega ao *step 10,* o qual desempenha o papel

de fazer o questionamento sobre o valor da energia calculada para o *subframe* atual. Se este possuir energia menor do que *lowerEnergyThreshold,* chega-se ao *step 11,* o qual é responsável pelo armazenamento do número de *subframes* consecutivos, cuja energia é menor do que o limiar estabelecido. Caso contrário, o algoritmo considera o *subframe* corrente como um falso final de atividade vocal, passo de número 15, e retorna ao primeiro passo.

Uma vez no *step 11*, procede-se de maneira análoga ao procedimento de determinação do início da atividade vocal, e, no *step 12*, similar ao *step 6,* armazena-se o *subframe* corrente em um vetor representando a atividade de fala efetiva. Caso o número contido de *subframes* no *step 11* seja maior do que *endTimeThreshold,* prossegue-se ao passo de número 14, o qual determina que o final de atividade vocal foi encontrado. Caso contrário, retorna-se ao passo de número 1 para a análise de um novo *subframe.*

Determinação dos Limiares

Para determinar os valores dos dois limiares de energia acima citados, foi utilizado o software MATLAB, cuja versão utilizada foi a 7.1. A partir de amostras de áudio gravadas no Laboratório de Arquiteturas Dedicadas (LAD), com o microfone especificado, foram estimados estes dois valores.

Para a frase “Quero usar a máquina”, o sinal de voz obtido e o sinal resultante após a aplicação do algoritmo proposto são mostrados na Figura 10. Na Figura, observam-se, no sinal original, intervalos de silêncio entre as palavras pronunciadas e grandes intervalos sem fala (Figura 10a). No sinal modificado, os intervalos entre as palavras foram removidos e apenas o trecho de voz efetiva é apresentado (Figura 10b).

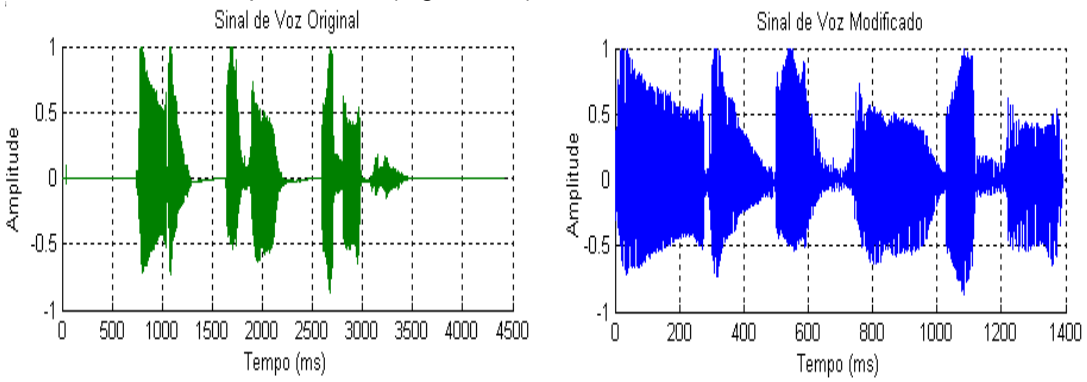

Figura 10. a) Sinal de Voz Original; b) Sinal de voz modificado (extração de *endpoints*).

O gráfico de energia por *subframes*, do sinal de voz apresentado na Figura 10, é apresentado na Figura 11. Nesta Figura, as siglas HET e LET representam, respectivamente, os dois limiares de energia *higherEnergyThreshold* e *lowerEnergyThreshold.* As linhas em vermelho representam estes limiares. Para as amostras obtidas e para o ambiente de gravação, os valores dos limiares estimados foram de 10,3 e 0,8, respectivamente.

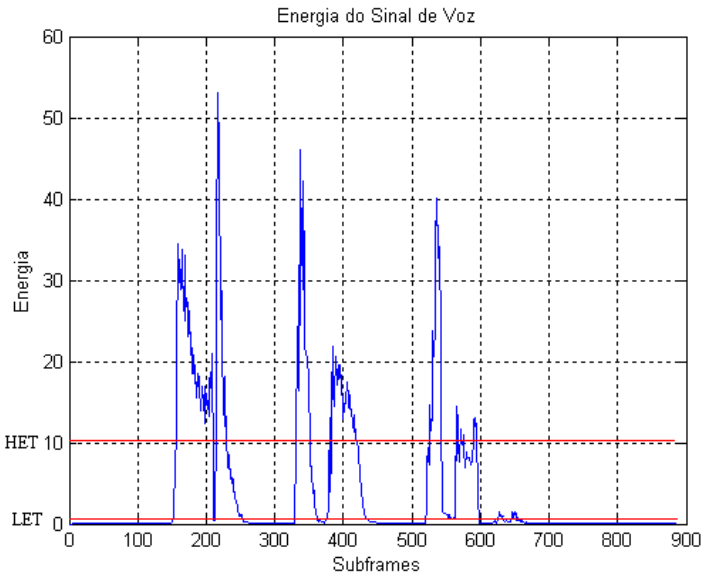

Figura 11. Gráfico de Energia por *Subframes*.

CONCLUSÕES

Diante do exposto acima, a detecção de *endpoints* se mostra eficiente para extração de silêncio presente no início e fim de uma elocução, bem como entre as palavras. Em virtude da simplicidade de implementação e eficiência, esta técnica se mostra promissora para implementação em hardware.

Após a validação da técnica (em andamento), a próxima fase do projeto consistirá na implementação do detector em linguagem de descrição de hardware (SystemVerilog) para posterior desenvolvimento do protótipo em FPGA.

AGRADECIMENTOS

À ajuda extra da professora Joseana.
Aos meus pais, pelo amor, carinho e cobranças sempre nos momentos certos.
Aos integrantes do LAD, pelo aprendizado e experiência obtidos durante o projeto.
Ao CNPQ, pela oportunidade concedida.

REFERÊNCIAS BIBLIOGRÁFICAS

AMBA AXI. **Specification.** Disponível em: < www.cse.iitk.ac.in/users/sheetesh/resources/AMBAaxi.pdf>. Acesso em 10 jul. 2009

ALTERA. Disponível em: < www.altera.com >. Acesso em: 02 jul. 2009.

BARR, M. **Programming Embedded Systems in C and C++.** O'Really Media, 1999. 194p.

BENZEGHIBA, M. F.; BOULARD, H., User-customized password speaker verification based on HMM/ANN and GMM models. In: INTENTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING, 7, 2002, Denver. **Proceedings...** Denver: International Conference on Spoken Language Processing – ICSLP, USA, 2002. p. 1325 – 1328.

BERGERON, J. **Writing Testbenches:** Functional Verification of HDL models. Second Edition. Norwell: MA. Kluwer Academic Publishers, 2003. 512p.

CAI, L; GAJSKI, D. **Transaction Level Modeling in System Level Design.** Center of Embedded Computer Systems, University of California, 2003

CARRO, L.; WAGNER, F. R. Sistemas Computacionais Embarcados. In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA, 22, 2003, Campinas. **Anais...** Campinas: Sociedade Brasileira de Computação – SBC, Brasil, 2003.

CAMPBELL, J. P. Speaker Recognition: A Tutorial. **Proceedings of the IEEE.** vol. 85, n.9, p.1437 – 1462, 1997.

CIPRIANO, J. L. G. **Desenvolvimento de Arquitetura para Sistemas de Reconhecimento Automático de Voz baseados em Modelos Ocultos de Markov**. Tese (Doutorado em Ciência da Computação), Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS. 2001. 123f.

COSTA, W. C. de A. **Reconhecimento de Fala Utilizando Modelos de Marvok Escondidos (HMM's) de Densidades Contínuas.** Dissertação (Mestrado em Engenharia Elétrica), Centro de Ciências e Tecnologia, Univerdade Federal da Paraíba, Campina Grande, PB. 1994.

Da CUNHA, A. M; VELHO, L. Laboratório VISGRAF – Instituto de Matemática Pura e Aplicada. **Métodos Probabilísticos para Reconhecimento de Voz.** Rio de Janeiro, 2003. 62p. (Relatório Técnico).

De LIMA, A. A.; FRANCISCO, M. S.; NETTO, S. L.; F. RESENDE Jr.; G. V. Análise Comparativa de Parâmetros em Sistemas de Reconhecimento de Voz. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, 23, 2000, Gramado. **Anais …** Gramado: Sociedade Brasileira de Telecomunicações, SBrT, Brasil, 2000.

DELLER Jr., R.; PROAKIS, J. G; HANSEN, J. H. L, **Discrete-time Processing of Speech Signals**, Macmillan Publishing Co., 1993. 936p.

DIAS, R. S. F. **Normalização de locutor em Sistemas de Reconhecimento de Fala.** Dissertação (Mestrado em Engenharia Elétrica), Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, Campinas, SP. 2000. 114f.

EDWADS, S; LAVAGNO, L.; LEE, E. A.; SANGIOVANNI-VINCETELLI, A. Design of Embedded Systems: Formal Models, Validation, and Synthesis. **Proceedings of IEEE.** vol. 85, n.3, p. 366 – 390, 1997.

FECHINE, J. M. **Reconhecimento automático de identidade vocal utilizando modelagem híbrida: Paramétrica e Estatística.** Tese (Doutorado em Engenharia Elétrica), Centro de Ciência e Tecnologia, Universidade Federal de Campina Grande, Campina Grande, PB. 2000. 212f.

FLANAGAN, L. J, **Speech Analysis Synthesis and Perception**. Second Edition. New Jersey. Murray Hill, 1978.

FRANCIA, G.A. Embedded Systems Programming. In: JORNAL OF COMPUTING IN SMALL COLLEGES, 15, 2001. **Proceedings...** Consortium for Computing Sciences in Colleges – CCSC, 2001, v.17, n.2, p.204-210.

FURUI, S. Cepstral Analysis Technique for Automatic Speaker Verification. **IEEE Transactions on Acoustics, Speech and Signal Processing Magazine**. v. 29, n.2, p.254-272,1981.

IBNKAHLA, M. **Signal Processing for Mobile Communication Handbook**. Florida: CRC Press, 2005. 520p.

KE S; HOU, Y.; HUANG, Z; LI, H. A HMM Speech Recognition System Based on FPGA. In: IEEE CONGRESS ON IMAGE AND SIGNAL PROCESSING. **Proceedings...,** Sanya, 2008. China, 2008. p.305–309.

KLEIJN, W. B.; PALIWAL, K. K., **Speech Coding and Synthesis**, New York: B. V. Elsevier Science, 1998. 774p.

KOERICH, A. L. Sistemas Biométricos. In: ESCOLA REGIONAL DE INFORMÁTICA, 12, 2004, Guarapuava. **Anais...** Guarapuava: Sociedade Brasileira de Computação – SBC, Brasil, 2004.

LAVAGNO L.; MARTIN, G.; SCHEFFER, L. **Eletronic Design Automation for Integrated Circuits Handbook** – 2 Volume Set. Boca Raton: F.L. CRC Press, Inc, 2006. 1152p.

LEMETTY, S. **Review of Speech Synthesis Technology**. Disponível em: < www.acoutics.hut.fi/>. Acesso em: 06 de jul. 2009.

LI, Q.; B.-H; JUANG, Q; ZHOU; C. LEE. Automatic Verbal Information Verification for User Authentication. **IEEE Trans. Speech and Audio Processing.** v.8, n.5, p.585-596, 2000

LIMA, M.; AZIZ, A.; ALVES, D.; LIRA, P.; SCHWAMBACH, V.; BARROS, E. ipPROCESS: Using a Process to Teach IP-core Development. **IEEE International Conference on Microelectronic Systems Education**. p.27-28, 2005.

MARTINS, J. A. **Avaliação de Diferentes Técnicas para Reconhecimento de Fala.** Tese (Doutorado em Engenharia Elétrica), Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP. 1997.

MORAES, F.; CALAZANS, N.; MOLLER, L.; BRIAO, E.; CARVALHO, E. **Dynamic and Partial Reconfiguration in FPGA SoCs**: Requirements Tools and a Case Study. In: ROSENSTIEL, W. New York, 2004.

OLIVEIRA, M. P. B. **Verificação Automática do Locutor, Dependente do Texto, Utilizando Sistemas Híbridos MLP/HMM.** Dissertação (Mestrado em Engenharia Elétrica), Departamento de Engenharia Elétrica. Instituto Militar de Engenharia, Rio de Janeiro, RJ. 2001. 111f.

OVM. Disponível em: < www.ovmworld.org >. Acesso em 27 jul. 2009.

PEGORARO, T. F. **Algoritmos Robustos de Reconhecimento de Voz Aplicados a Verificação de Locutor.** Dissertação (Mestrado em Engenharia Elétrica), Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, Campinas, SP. 2000. 101f.

PESSOA, I. M. **Geração Semi-Automática de *Testbenches* para Circuitos Digitais Integrados.** Dissertação (Mestrado em Ciência da Computação), Departamento de Sistemas e Computação, Universidade Federal de Campina Grande, Campina Grande, PB. 2007. 52 f.

PIZIALI, A. **Functional Verification Coverage Measurement and Analysis.** First Edition. Massuchusetts. Kluwer Academic Publisher, 2004**.** 400p.

RABINER, L. R.; SCHAFER, R.W. **Digital Processing of Speech Signals**. New Jersey: Upper Saddle River. Prentice Hall, 1978. 512p.

RABINER, L. R.; JUANG, B. H. **Fundamnentals of Speech Recognition.** Nova Jersey: Englewood Clifs. Prentice Hall, Inc., 1993. 496p.

SANTOS, A. D. Universidade Federal do Paraná. **Reconhecimento de Palavras Faladas.** Curitiba, 2007. 13p. (Relatório Técnico)

SHAUGHNESSY, D. O. **Speech Communication:** Human and machine. Second Edition. New York. Willey-IEEE Press, 1999. 548p.

SILVA, D. D. C. **Desenvolvimeto de um IP-core de Pré-Processamento Digital de Sinais de Voz para Aplicação em Sistemas Embutidos.** Dissertação (Mestrado em Ciência da Computação), Departamento de Sistemas e Computação, Universidade Federal de Campina Grande, Campina Grande, PB. 2006. 107f.

SILVA, K. R. G. da. **Uma Metodologia de Verificação Funcional para Circuitos Digitais.** Tese (Doutorado em Engenharia Elétrica), Departamento de Engenharia Elétrica, Universidade Federal de Campina Grande, PB. 2007. 119f.

SYSTEMVERILOG. Disponível em: < www.systemverilog.org > Acesso em 06 julho de 2009.

VASCONCELOS, M. C. R. De. **Construção de um Ambiente Computacional para Implementação de Aplicações na Área de Comunicação Vocal Homem-Máquina.** Universidade Federal de Campina Grande, Campina Grande, PB. 2004. (Iniciação Científica).

VASSALI, M. R., de SEIXAS, J. M. e ESPAIN, C. Reconhecimento de Voz em Tempo Real Baseado na Tecnologia dos Processadores Digitais de Sinais. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, 23, 2000. **Anais …** Gramado: Sociedade Brasileira de Telecomunicações, SBrT, Brasil, 2000.

VIEIRA, M. N. **Módulo Frontal para um Sistema de Reconhecimento Automático de Voz**. Dissertação (Mestrado em Engenharia Elétrica), Faculdade de Engenharia Elétrica, Universidade de Campinas, Campinas, SP. 1989.