A FPGA-BASED VITERBI ALGORITHM IMPLEMENTATION FOR SPEECH RECOGNITION SYSTEMS.

Fabian Luis Vargas[1] *Rubem Dutra Ribeiro Fagundes*[2] *Daniel Barros Junior*[3]
vargas@ee.pucrs.br *rubem@ee.pucrs.br* *dbarros@ee.pucrs.br*

DEE – Departamento de Engenharia Elétrica – Faculdade de Engenharia

Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS
Av. Ipiranga 6681, prédio 30 sala 152
Porto Alegre - RS - Brasil

ABSTRACT

This work proposes a speech recognition system based on a hardware/software co-design implementation approach. The main advantage in this approach is an expressive processing time reduction in speech recognition, because part of the system is implemented by dedicated hardware. This work also discuss another way to implement "Hidden Markov Models" (HMM), a probabilistic model extensively used in speech recognition systems. In this new approach, the Viterbi algorithm, used to compute the HMM likelihood score, will be "built in" together with the HMM structure designed in Hardware, and implementing probabilistic state machines that will run as parallel processes each one for each word in the vocabulary handled by the system. So far, we have a dramatic speed up performance, getting meseaures around 500 times faster than a classic implementation with the correctness comparable with others isolated word recognition systems.

1. INTRODUCTION

1.1 Speech Recognition System Structure

A speech recognition system (SRS) is basically a pattern recognition system dedicated to detect speech, or in other words, to identify language words into a sound signal achieved as input from the environment.

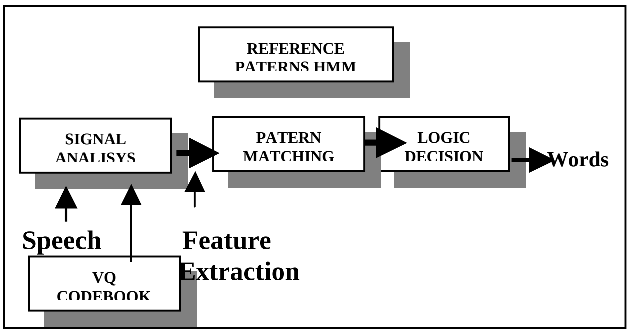

Figure 1: Speech Recognition System

Figure 1 shows the main steps to process a front-end speech recognition system. In the signal analysis step a speech sampling will be made with an A/D converter. Those samples are processed in order to extract some relevant features from speech signal input. [FAGU1993] [RABI1993]. The next step, pattern matching, makes a comparison among source reference patterns (also sets of signal parameters from reference patterns) previously stocked on the system and scores the likelihood of this reference patterns against the input set. The next step, decision logic, chooses one of those reference sets that match with the signal parameters set from the input (usually called "test set").

2. SECTION II

2.1 Signal analysis implementation.

Signal analysis is responsible for signal sampling, its conversion in a digital representation, and vector quantization. At end, the speech signal will be replaced by sequences of label-codes. (figure 2)

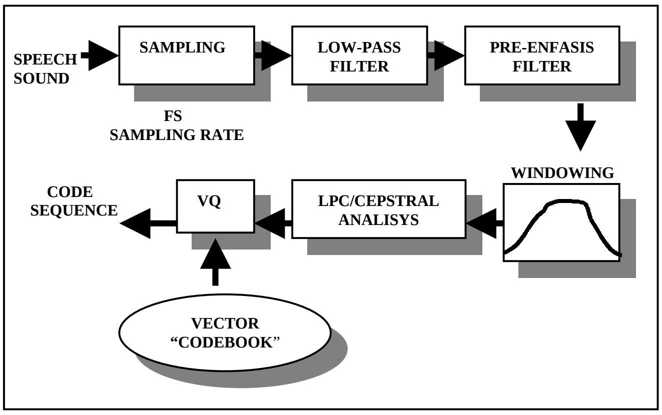

Figure 2: Signal Analysis main tasks

Figure 2 shows the six sequential tasks to be executed in the signal analysis step, which are:

[1] Professor in Electrical Department, Engineering School at Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre – RS - Brazil

[2] Professor in Electrical Department, Engineering School at Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre – RS - Brazil

[3] Researcher in the Electrical Department, Engineering School at Pontificia Universidade Católica do Rio Grande do Sul, Porto Alegre – RS - Brazil

- **Sampling**: the SRS converts speech sound from the outside world into digital representation. Essentially, this task will include a sample and hold device, and an analogic-digital (A/D) converter. It is necessary to choose the appropriate sampling rate, according with Nyquist-Shannon theorem.

- **Low pass filter**: cuts those high frequencies found on the signal due to sampling. Usually this filter is adjusted by sampling rate [DELL1993].

- **Pre-emphasis filter**: adjusts the high variations on spectrum frequencies due to glottal pulse and lips radiation found in the speech signal behavior. [GRAY1982],[RABI1978].

- **Windowing:** cuts the speech signal into blocks of 10 ms signal frame each. A hamming window adjusts those frame samples [O'SH1987],[HARR1978].

- **LPC/Cepstral analysis:** algorithms process each frame in order to complete the cepstral coefficients from linear predictive coefficients [GRAY1982][MAKH1975].

- **VQ – Vector quantization:** each vector of cepstral coefficients is evaluated by distance measure. Using, as a map, a codebook with reference vectors in the acoustic space. The final output is a sequence of label codes[1] (usually called observation sequence) that will be evaluated by the pattern matching process. [RABI1985][LIND1980][DELL1993].

3. SECTION III

3.1 The pattern matching process

Pattern matching is the identification step, where the words spelled in the speech signal are identified generating a sequence of text words. The sequence observation is evaluated using Hidden Markov Models (HMM), which, as the acoustic reference pattern, plays the main role in the recognition process. [RABI1989a], [FAGU1993] [FAGU1998].

3.1.1 Hidden Markov Models (HMM)

Figure 5 shows a HMM structure usually applied in speech recognition systems [RABI1986][FAGU1993]:

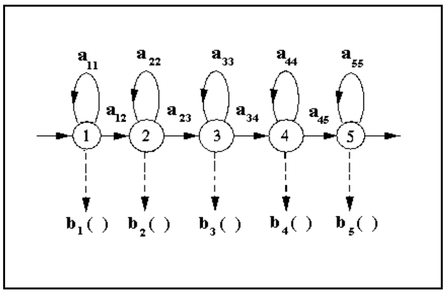

Figure 5: Hidden Markov Model

As seen in figure 5, we can define a HMM $\lambda = (A, B, \pi)$ with this set of parameters:

A : $\{a_{ij}\} = P\{q_j / q_i\}$, Probability Transition Matrix, with dimension N^2 , and N is the number of states. This matrix describes a probability transition from state q_i to q_j.

B : Matrix $b_j(k) = P\{V_k / q_j\}$, the probability to get the symbol V_k in the state q_j. and for DDHMM[2] $b_j(k) = \{b_{jk}\}$; $1 \le j \le N$ e $1 \le k \le M$.

for all N model states and M symbols used on VQ.

π **:** Initial probability vector $\pi(i)$. Concerning HMM from figure 5 , this vector will always be defined as [1 0 0 0 ..],

$\{V_1, \ldots, V_k, \ldots, V_M\}$ **:** set of M symbols

$O = \{O_1, \ldots, O_T\}$ **:** observation sequence in the interval [1,T]

$Q = \{q_1, \ldots, q_T\}$ **:** state sequence through the HMM in the interval [1,T].

N – Number of states

M – Number of symbols (number of centroids or, also, number of label-codes)

For a more didactical approach in Hidden Markov Models with applications in speech recognition, we would recommend the following references [RABI1989b] [HUAN1990] [FAGU1993]

3.1.2 FPGA Implementation

Our approach proposes the use of HMM implementation with FPGA as a tool for the evaluation of the code label sequences (instead of the traditional utilization of Viterbi algorithm).

The main idea is to use the FPGA to implement another kind of HMM, by the use of adders, comparators and logic operators, performing the pattern matching process without running Viterbi algorithm. This new HMM can be seen in the figure 6:

[1] These labels are numbers, relating the codebook reference vectors, usually called centroids, in VQ. Each label-code is really just a label to those centroids. After VQ, each label-code in the sequence output is called observation.

[2] Discrete Density function Hidden Markov Model

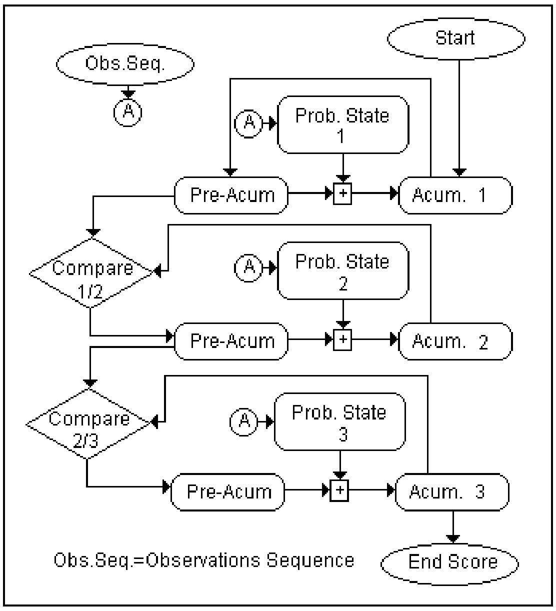

Figure 6 : HMM implementation with FPGA

In this new approach, the recognition process is done by scoring likelihood from the code-label sequence achieved by VQ (as done in Viterbi algorithm) directly by logical blocks in FPGA chip. The whole process is built by the structure shown in figure 7.

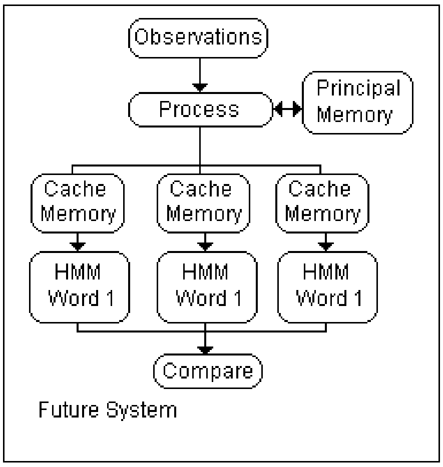

Figure 7: Pattern Matching step implemented by dedicated hardware.

Figures 6 and 7 shows all the tasks that pattern matching process performs, which are:

1. To fetch the probability score addressed by each code label from the observation sequence, for each current HMM state.
2. To accumulate those scores thorough all HMM states, generating a final accumulated score.
3. To choose the highest HMM accumulated score using a comparison logic block.

This new approach saves time in the speech recognition process, because instead of running an algorithm to score, it computes directly the likelihood scoring.

4. SECTION IV

4.1 Methodology

As can be seen in figure 8, the left side of the internal bus comprises the Motorola 56002 microprocessor and is responsible for executing the software part of the speech recognition procedure (i.e., data acquisition and signal processing). The right side of the internal bus is composed by the Altera FPGA design kit, which is based on the FPGAs MAX 7K128 [UP1BOARD] and FLEX 10K20 chips [FLEX10K]. This part implements the HMMs in a dedicated hardware, and is responsible for the pattern (voice) recognition procedure itself. This task is dramatically speedup by implementing into the FPGAs a dedicated hardware to perform parallel arithmetic operations.

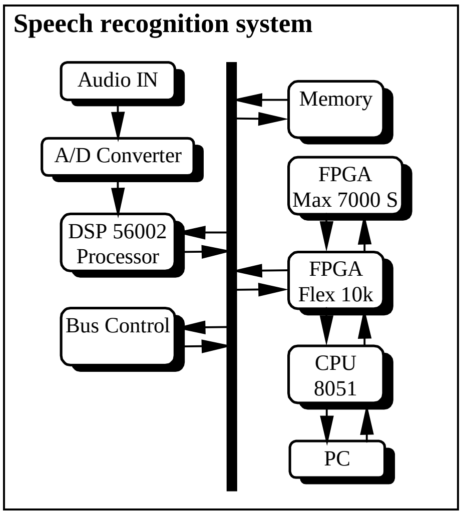

Figure 8: *Speech Recognition System:* the left side of the internal bus (bold line) is based on the DSP 56002 microprocessor development kit, from Motorola; the right side is based on the Altera FPGA development kit.

4.1.1 Signal Analysis

The signal analysis step is done by the Motorola 56002 EVM, with a 16 kHz sampling rate. The following tasks are programmed with C and Motorola assembler languages and stored into Motorola 56002 EVM. Furthermore, we use 10 ms sample frames, with 2/3 frame overlapping in windowing and LPC/Cepstral steps, in order to avoid loosing samples during parameters extraction. After running a short speech detector algorithm (in order to discard silence segments in the input), the signal analysis provides an observation sequence to the pattern matching step.

4.1.2 Pattern Matching

The pattern matching to implement these structures as shown in figures 6 and 7, uses two Altera FPGA chips: MAX 7K128 and FLEX 10K20, placed into an Altera development board. In addition, we use VHDL (VHSIC Hardware Description Language) [BERG1996] [OTT1996] to describe the whole structure proposed: using 6-state HMM and 128 vectors in the codebook. For each state, there is a probability table relating each code-label with the probability of occurrence in the input sequence for that state. Due to space limitations in the FPGA, we use a memory module storage, controlled by a 8051 Intel microcontroller, to store those probability tables and provide those values during score computation. Such tables plays the role of Matrix B in a HMM parameters, providing a discrete probability density function of the VQ symbols associated for each state.

5. RESULTS

Table 1 below shows the results achieved in this new fpga-based approach:

Words	*Obs. Sequence size*	*Total [clk]*	*FPGA-BASED Time [ms]*	*Classic Viterbi Time [s]*
2	66	5280	1.320	0.577
3	66	7920	1.980	0.917
4	66	10560	2.640	1.240

Table 1 : *Comparison results between fpga-based approach and classic Viterbi approach.*

The time measured in an fpga viterbi implementation and a classic vitebi implementation can be seen in table 1, for different amount of words (using maximum 4 words due to FPGA space limitations). The number of observations (observation sequence size) and the number of clock pulses is also presented for performance evaluation.

6. CONCLUSIONS

Our research group is now working towards the integration of the speech-recognition system. This task is being executed by developing a system based on a single board containing the 56002 Motorola microprocessor, the Altera FPGAs, mass memory modules and the overall glue logic that the system requires in order to operate in a stand-alone configuration. At the same time we have the isolated word recognition system already trained with MATLAB, using a VQ codebook with 128 centroids, and a six states HMM. We have also tried another configuration such as to use 8 states in the HMM. However, due space limitations in the FPGA, we have decided for 6 states as the best choice. In our test procedures we have achieved a very good performance with 99.7% correctness rate in a vocabulary with more than 10 words. From the results above it should be note the dramatic speed up achieved, around 500 times faster than a classic Viterbi, due to parallel hardware Viterbi implementation. Also, we must point out that the few amount of words currently reached in this implementation will be overcome by the new VLSI technologies, supporting this approach to be used in small speech recognition circuits in the future.

7. REFERENCES

[BERG1996] BERGÉ, Jean-Michel. *VHDL designer's reference.* Dordrecht: Kluwer, 1996.

[DELL1993] DELLER, J., PROAKIS, J.G., HANSEN, J. H.L. *Discrete-time processing of speech signals.* New York : Macmillan, 1993.

[FAGU1993] FAGUNDES, Rubem Dutra Ribeiro. *Reconhecimento de Voz, Linguagem Contínua, Usando Modelos de Markov.* São Paulo, 1993. Dissertation (mestrado) - Escola Politécnica, Universidade de São Paulo.

[FAGU1998] FAGUNDES, Rubem Dutra Ribeiro. *Abordagem Fonético-Fonológica em Sistemas de Reconhecimento de Voz de Linguagem Contínua.* São Paulo, 1998. Tese de Doutorado - Escola Politécnica, Universidade de São Paulo.

[FLEX10K] FLEX 1OK Literature (internet source) http://www.altera.com/html/literature/lf10k.html

[GRAY1982] GRAY JR., A.H.; MARKEL, J.D. *Linear prediction of speech.. In: Communication and cybernetics,* 3 ed., Berlin : Springer, 1982.

[HUAN1990] HUANG, X.D.; ARIKI, Y.; JACK, M.A. *Hidden Markov models for speech recognition.* Edinburgh, Edinburgh University Press, 1990.

[LIND1980] LINDE, Y.; BUZO, A.; GRAY, R.M. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, v.28, n.1, p.84-95, jan. 1980.

[MAKH1975] MAKHOUL, J. Linear prediction: a tutorial review. *Proceedings of the IEEE.* v.63, n.4, p. 561-580, apr 1975.

[OTT1996] OTT, Douglas et alii. *A designer's guide do VHDL synthesis.* Boston: Kluwer, 1996.

[O'SH1987] O'SHAUGHNESSY, D. *Speech Communication Human and machine.* Massachusetts : Addison-Wesley, 1987.

[RABI1985] RABINER, L.R.; LEVINSON, S.E. A speaker-independent, syntax-directed, connected word recognition system based on hidden markov models and level building. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* v.33, n.3, p.561-73, June 1985.

[RABI1986] RABINER, L.R., JUANG, B.H. An introduction to Hidden Markov Models. *IEEE Acoustics, Speech, and Signal Processing*, p. 4-16, jan. 1986.

[RABI1989a] RABINER, L.R.; WILPON, J.G.; SOONG, F.K. High performance connected digit recognition using Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* v.37, n.8, p.1214-1225, aug. 1989.

[RABI1989b] RABINER, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, v.77, n.2, p.257-86, Feb. 1989.

[RABI1993] RABINER, L., JUANG, B.H., *Fundamentals of speech recognition.* New Jersey : Prentice Hall, 1993.

[UP1BOARD] UP1 Board Literature (internet source) http://www.altera.com/html/univ/features.html#products

[WOLF1980] WOLF, J.J. Speech recognition and understanding. In: FU, K.S. et al. *Digital pattern recognition.* 2.ed. Berlin : Springer, 1980. p.167-203.