

ASIC IMPLEMENTATION OF VECTOR QUANTIZATION FOR SPEAKER RECOGNITION

M. Li, J. T. Proudfoot and G. King
Department of Electrical & Electronic Engineering,
University College of Swansea
&

Informatics Research Centre
Southampton Institute of Higher Education, Southampton, SO9 4WW, UK
Tel: (44703) 319237, Fax: (44703) 319322
email: li_m@uk.ac.southampton-institute

ABSTRACT

This paper describes the high-level design and specification of an Application-Specific Integrated Circuit (ASIC) to implement a pattern matching algorithm based on vector quantization (VQ) and speaker verification algorithms. The ASIC uses a fully synchronous top-down design methodology with a hardware description language, ELLA. As well as presenting details of the algorithm and register-transfer level structure, the paper presents the hierarchical design methodology used to implement the algorithm in low cost ASIC form.

The approach can be applied to a wide range of digital signal processing functions. The goal of this effort is to develop pattern matching system based on the VQ and codebooks which can be used as a alternative to the currently used software system.

1 INTRODUCTION

Achieving the performance required from a modern digital signal processing (DSP) system often necessitates the real time application of reliable numerical algorithms for least squares estimation, distance measuring, solving linear equations, performing singular value decomposition and so on. In order to perform such computations at the required data rate, it is sometimes necessary to design a highly dedicated parallel processor which can be implemented using advanced VLSI technology. The research of this paper presents the design of the pattern matching chip at high level using ELLA tools: ELLA hardware description language and simulator.

The VQ technique is widely applied in the speech processing research community for speech analysis and recognition. In addition, the VQ and codebooks is interesting for its applications specially in speaker recognition, since it displays implementation-simplicity and time-independence. The algorithm motivated by distance processing, notably those developed by Mason et al [1], has demonstrated some degree of improvement on restricted recognition tasks.

The speaker recognition system shown in Figure 1, is discussed in [1]. The feature extraction is based on linear predictive coding (LPC) technology [2] to determine the short-term prediction with 12 coefficients. The raw speech is pre-processed by a low-pass filter after A/D conversion, then segmented into 25.6 msec frames with 50% overlap. The segmented speech is multiplied by a Hamming window prior to linear prediction analysis using autocorrelation method.

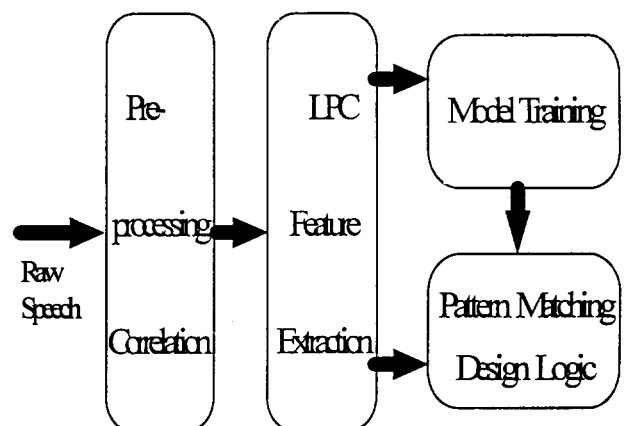


Figure 1 Automatic Speaker Recognition System

A 32-bit floating point vector quantization scheme is used to code the 12 LPC coefficients. The simulation of the design is over 10 VQ codebooks: one for each person, and each codebook has 32 codewords with 12 coefficients. The pattern matching technique is employed to perform the recognition. Finally, a decision is made by decision logic unit. The speaker recognition system has been designed and specified. It consists of three ASIC chips. One is for pre-processing and autocorrelation calculation, the second for LPC feature extraction and the third for pattern matching. The model training can be done by the software system developed by the relevant research work because the codebooks are fixed before real time processing, which makes the chip flexible. To make it simple, this paper only

demonstrates the pattern matching technique on a single chip for VQ, personalised codebooks with a top-down design methodology using a CAD package--ELLA.

The remainder of this paper is organised as follows: Section 2 will discuss the design and development of the chip using ELLA. Section 3 will give an overview of both pattern matching based on VQ codebooks and verification algorithms. This is followed by a description of the hardware architecture of the real-time implementation of the pattern matching technique in Section 4. Section 5 presents simulation results and future work. The conclusion is in Section 6.

2 PATTERN MATCHING AND VERIFICATION ALGORITHMS

The pattern matching technique is based on VQ which is widely applied in the speech processing research community for speech analysis and recognition. In addition, the VQ and Codebooks (VQCBs) technique is interesting for its applications specially in speaker recognition [1], because of its previously stated implementation-simplicity and time-independence.

Mapping of the pattern matching algorithm based on VQCBs into a single chip is discussed below. The key issue in implementing the pattern matching concerns searching of the codebooks. When test data is acquired, the codebook searching process uses the Euclidean distance measure $d(\bar{c}_1, \bar{c}_2)$ between a test vector \bar{c}_1 and a reference vector \bar{c}_2 within each codebook in order to make the first intra-codebook selection, see equation (1),

$$d^2(\bar{c}_1, \bar{c}_2) = \sum_{i=1}^k (c_{1,i} - c_{2,i})^2 \quad (1)$$

where k is the order of a vector while there are 32 vectors in a codebook. For the whole utterance the process is repeated on the sequence vectors of the utterance. Then the smallest distances for the full test vector sequence from matched reference vectors within the book are accumulated in order to make the second inter-codebook selection. Finally, the accumulated distances are compared with each other and the codebook which corresponds to the smallest accumulated distance is assumed to identify the speaker.

For speaker verification a threshold is set to compare with the identified distance. If the distance is less than the threshold the speaker is accepted, otherwise, it is rejected.

3 ASIC DESIGN METHODOLOGY

This project uses a fully synchronous top-down design methodology with the ELLA design tools. ELLA[3] is an integrated high-level CAD environment for the design and simulation of an architecture described in a hardware

description language. It models architectures in a modular way, and its support environment is used to simulate different design alternatives.

To cope with the overwhelming complexity of VLSI circuits, it is possible to consider a hierarchy of partition. If the whole system is regarded as the highest level, it is possible to subdivide the system into modules each of which performs a specific function. That function may also occupy an identifiable area of the chip layout. Within each module further subdivision should be possible until a primary unit of the type under consideration is reached. This partition can be applied to most design representations to some extent. It is widely understood now that the partitioning of a design can bring major benefits to the accuracy, integrity and efficiency of the system especially when it is a VLSI device.

Corresponding to ELLA description, ELLA describes systems at various levels of abstraction: architectural, structural and gate level. At a higher level of abstraction each building block is treated as mathematical operator with its parallel and pipelined structure and interconnections with ELLA function declarations. This application has shown that the ELLA environment enables the designer to gradually add detail to the modules, establishing the partitioning, structure and communication requirements which will apply in the layout. One design may combine all levels. This obviously reflects design hierarchy.

The hierarchical design methodology inherent in ELLA enables various degrees of parallelism and pipeline to be conveniently exploited to achieve efficient ASIC implementation. The ASIC implementation of the pattern matching techniques utilises both parallelisation and pipeline to achieve an efficient architecture which matches the computational requirements of the algorithm to the processing bandwidth of the target ASIC design and fabrication technology.

It is a modular structure and consequently yields simple complexity management, design modification, or implementation change with different components by rebinding the design via the function declaration in ELLA.

4. ARCHITECTURE

As stated in Section 2, the codebook searching is an example of pattern recognition. The main constraint on the algorithm is that a test vector has to pass through each entry in each codebook consecutively for vector distance measure. The number of processing engines for distance calculation is open to consideration, for the test vector could be run through ten separate parallel processors or one processor multiplexed ten ways for ten codebooks. Intermediate configurations are also possible, such as two processors handling five codebooks each with the same test vector.

A rough calculation on the number of operations for the vector distance iteration measure used here shows that the most economical solution appears to multiplex one processing engine ten ways to construct the ten-book comparison. So a custom bit-parallel processor will be exploited

Having decided on the recursive architecture, therefore, an implementation for the pattern matching technique can be partitioned into its natural arithmetic components: a vector distance recursive processor (VDP), a comparator (MIN), an adder, a divider and a decision maker, see Figure 2. These major operators are then augmented with registers (e.g. minimum value register (MVR) and optimum person register (OPR)), memories and multiplexing elements.

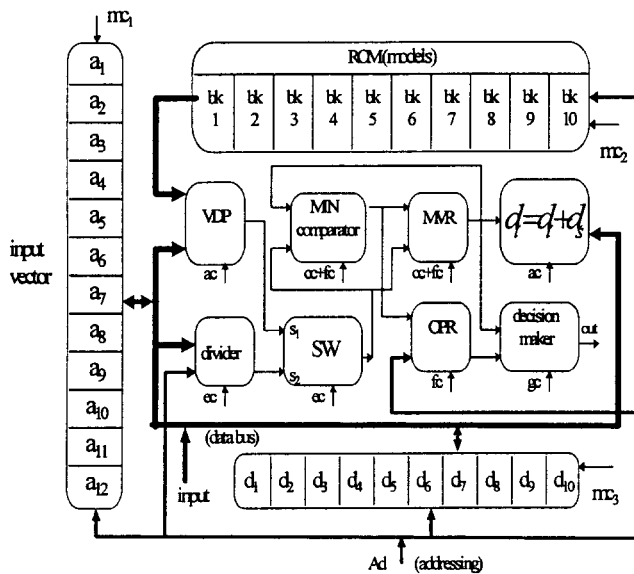


Figure 2 Block Diagram of Pattern Matching and Logic Decision

4.1 Storage

The system has three individual memory blocks. Since the input source vector V is repeatedly accessed for each distance computation, it is stored in an on-chip buffer. An intermediate memory receives results from the execution unit, sorts it and sends it to the specified data port. The codebooks' memory is partitioned off chip. This is motivated by the fact that the codebook sizes are usually quite large (typically at least 16K for a 12-dimensional 32×10 code-words vector quantizer).

4.2 Data Bus Structure

The chip uses a common data bus through which all components can communicate with each other except the

codebook storage memory which is directly connected to one of the processor input ports so that the processor can read data from the input memory and the codebook memory in parallel. This is because the whole system spends more than 90% of the time reading data from the codebook memory. The bus structure alleviates time-contest of one data bus structure, thereby increasing throughput and minimising system component count.

4.3 Vector Distance Recursive Processor (VDP)

For vector distance computation, based on equation (1), the computational core of the chip consists of a 32-bit

subtractor and a 32×32 bit multiplier to perform the minus-square operation and produce a full precision 32-bit product which is input to an accumulator. It is a highly parallel and pipelined design to exploit the inherent parallelism in DSP algorithms.

4.4 Minimisation

A comparator (MIN) for intra-book and inter-book selections of the smallest distance (minimisation of distances). It accepts a vector distance from the VDP and compares the current distance with the previous smaller one obtained last time and kept in the minimum value register (MVR). If the current distance is smaller or equal to the previous best minimum, a flag "newout" is set high. Then the current one is put into the register for next comparison. The MVR is initialised by exerting max which loads the largest 32-bit number into the MVR at the beginning of any new codebook search. The inter-book comparison after the codebooks search shares the comparator in different time so that the usage of the comparator is increased.

4.5 Adder

An adder operates as $d_i = d_i + d_s$ for accumulation of the matched distances for whole test sequence of the utterance and for each book.

4.6 Divider

A divider normalises the accumulated smallest distances by the number of the test sequence vectors for speaker verification,

4.7 Controller

Designing ASICs is, in essence, about the control of an ASIC. If the control mechanism is correct, the operation of the chip is within a proper framework. The main task of the controller is to generate both addresses for the memories and control-signals required in the codebooks search by the processor,

such as ac, bc and ec and so on in Figure 2. The Algorithm State Machine technique [4] is used to express hardware algorithms.

The codebooks structure for codebooks search is arranged like that ten codebooks are located subsequently in the codebooks memory. Each codebook has 32 codevectors each of which occupies 12 contiguous locations of the codebook memory. Therefore, a set of increment counters is used to generate their addresses for supporting codebooks search. The codebook memory addresses are presented by 12 bits.

Figure 2 shows the block diagram of the system, which represents both its functional flow diagram and its logical and physical structure. The realisation of the circuit from this level of description only requires few operator module designs implementing the desired functions, with mutually compatible interfaces and a controller for the timing.

32-bit floating point arithmetic is used throughout the data-bus and 64-bit floating point arithmetic is exploited within the VDP processor, but its output is rounded off to 32 bits to keep bus size, real estate usage and memory size low.

5 SIMULATION RESULTS AND FUTURE WORK

High level functional simulations were performed during development of the ASIC architecture to verify circuit functionality, timing and performance. Test vectors were generated by extracting parameters from real data. The results are shown in Table 1.

The development is currently underway to synthesise the system using VIEWlogic package.

Models	ERROR	
	True Speaker	Impostor Speaker
1	4.02%	5.11%
2	6.93%	7.11%
3	5.00%	5.26%
4	4.45%	5.53%
5	4.21%	3.32%
6	4.08%	5.41%
7	6.00%	7.64%
8	3.97%	4.25%
9	8.11%	9.36%
10	5.26%	5.78%
Average	5.20%	5.88%

Table 1 Verification Equal Error Rate for 10 Models and 20 Speakers in 3 Versions of 25 Alphabet

6 CONCLUSIONS

A new dedicated speech signal processor has been presented which employed multiple processing units on a single chip. Starting from the algorithm-level description, it is ended with the structure-level implementation of the pattern matching, for the design of which, a high level top-down design methodology is presented. This application has shown that at behavioural levels the ELLA environment enables the designer to gradually add detail to the modules, establishing the partitioning, structure and communication requirements which will apply in layout. And ELLA can provide useful feedback on design performance at a very early stage in the design process so that the system architecture can be perfected rapidly and efficiently. Furthermore, this performance information was made available at the cost of only a few days effort. The finished model stands as a functional specification for the pattern matching hardware which could be developed into a complete, gate-level description of the circuit ready for implementation as a single ASIC.

REFERENCES

- [1] J. S. Mason, J. Oglesby and L. Xu, "Codebooks to Optimise Speaker Recognition", Proc. Eurospeech, pp. 267-270, 1989.
- [2] L. R. Rabiner and R. W. Schafer, "Digital Processing of speech signals", pp 396-412, Englewood Cliffs, N. J.: Prentice Hall, Inc., 1978.
- [3] The ELLA User Manual, Praxis Electronic Design Ltd, UK, 4.0 edition, 1990.
- [4] David Green: " Modern Logic Design " 1986 Addison Wesley Publishing Company, Inc.