

# Exhibit H

# Hardware Realization of Waveform Vector Quantizers

BERTRAM P. M. TAO, MEMBER, IEEE, HÜSEYİN ABUT, SENIOR MEMBER, IEEE,  
AND ROBERT M. GRAY, FELLOW, IEEE

**Abstract**—A real-time full search vector quantization system for speech waveform coding is implemented using LSTTL and CMOS devices. The system consists of low-pass filters, A/D and D/A converters, an algorithm for discriminating voiced and unvoiced speech, a full search vector quantizer encoder and decoder, and a microprocessor-based controller.

The system is designed to operate at two possible rates: one bit/sample using a dimension 8 vector quantizer (6500 bits/s) or 2 bits/sample using a dimension 4 vector quantizer (13 000 bits/s). In both cases the codebooks have rate 8 bits/vector. Separate codebooks were designed for voiced and unvoiced speech based on a training sequence of 640 000 samples containing five different speakers.

The subjective and quantitative results are compared to both simulations and with a real-time array processor based implementation.

## I. INTRODUCTION

WITH the increasing importance of digital speech in military, industrial, and consumer applications, a variety of techniques have been developed. The low rate systems operating under 4800 bits/s are mostly voice coders such as LPC systems which transmit a parametric representation of a frame of speech. Higher rate systems are mostly waveform coders that transmit information sufficient to produce a reproduction that “looks like” the original. The low rate systems are useful in systems with severe memory or communication capacity constraints, but they are complex and are sensitive to background noise, communication channel errors, and multiple speakers. Waveform coders require more rate, but they are usually simpler and more robust against all kinds of errors.

Numerous sophisticated waveform coders have been developed and implemented during the past few years. An example of such a sophisticated and implementable coding system is a subband coder using adaptive transform coding or adaptive DPCM in the subbands. Such techniques are implementable using special purpose chips or general pur-

pose DSP chips. Most such schemes eventually use some form of scalar quantization following a transformation on input blocks. For example, the block may be Fourier transformed and then the Fourier coefficients separately quantized with varying bit allocations. Such techniques can be further fine tuned by adaptation.

From an information theoretic point of view, however, scalar quantization, even if preceded by optimal transformations and incorporating adaptation, is not the optimal means of quantizing a sequence of random vectors if by optimal we mean in the sense of minimizing an average distortion. The basic theorems of Shannon theory state that even for memoryless processes, one can always do better by quantizing an entire vector rather than separately handling components. Intuitive ad hoc schemes such as transform coding and DPCM are popular, however, because they tend to be simple, they work quite well for a variety of sources, they are easy to implement, and, in addition, no better techniques have been available. In particular, Shannon theory promises the existence of optimal vector quantizers, but provides no constructive techniques for designing them.

During recent years many simulation studies have been conducted on a family of design algorithms for vector quantizers that use a generalization of Lloyd's original PCM design technique, a technique that is currently commonly used for minimum average cost cluster analysis and pattern recognition. (See, for example, the papers and references in the March 1982 Special Issue on Quantization of the IEEE TRANSACTIONS ON INFORMATION THEORY for both the classical papers and modern surveys of these algorithms.) The design technique is known to produce at least a locally optimal vector quantizer for a given distortion measure and for either a probabilistic source model or a very long training sequence of data produced by the source to be compressed.

Simulations have been conducted for a variety of information sources, including speech waveforms [1]. The transmitter or encoder is conceptually simple since it finds a minimum distortion or nearest neighbor codeword for an observed source vector by searching a codebook of reproduction vectors or templates in a ROM where the codebook is designed off-line by the design algorithm. This operation requires no transforming, variable bit allocation, adap-

Manuscript received September 14, 1982; revised June 15, 1983. This work was supported in part by the Joint Services Electronics Program at Stanford University, Stanford, CA 94305. Parts of this paper were presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing, Paris, France, April 1982, and the IEEE International Symposium on Information Theory, Les Arcs, France, June 1982.

B. P. M. Tao is with Hycom, Inc., Irvine, CA 92714.

H. Abut is with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA 92182.

R. M. Gray is with the Information Systems Laboratory, Stanford University, Stanford, CA 94305.

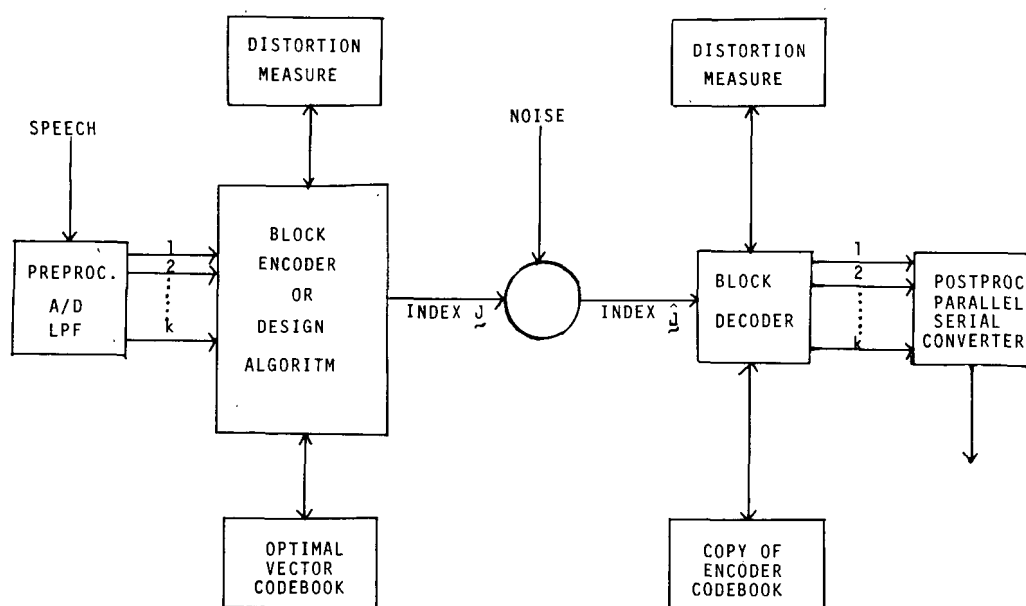


Fig. 1. General block diagram of a vector quantization system applied to speech waveforms.

tation, prediction, or other operations. It does, however, require very large memories and a large search effort. Both requirements precluded hardware implementation during the early days of algorithm development. With faster microprocessors and cheaper and larger memories these problems are no longer significant. The research reported in this paper was aimed at developing a real-time hardware vector quantizer using mostly "off-the-shelf" economical and industry standard IC devices coupled with two special purpose processors—one to perform the vector quantization and another to extract voicing information. A similar project was accomplished simultaneously but independently using a similar approach by Y. Yamada of Osaka University [4].

The quantizer was designed using such components rather than a DSP chip for two reasons. Several of the more famous DSP chips were not available to us when this project was begun, and almost all of the available chips were not well matched in structure or in timing to our particular application. While capable of doing signal processing of high complexity, the chips were not efficient for simple nearest neighbor searches of large codebooks. For example, the Texas Instruments TMS320 chip has the smallest basic cycle time of 200 ns and it is optimized for particular DSP applications. The internal architecture of this chip, however, is such that it would require a minimum of 4 cycles to perform distortion computations, which is 300 ns slower than needed here.

The paper is organized as follows. Section II contains a description of vector quantizers for waveform coding. Section III summarizes some old and new simulation results, including some real-time simulations using an array processor. Sections IV–VI describe the implementation in detail. In the final section the results of the various experiments are compared and possible improvements and modifications are described.

## II. WAVEFORM VECTOR QUANTIZERS

### A. Full Search Quantizers

A block diagram of a  $k$ -dimensional vector quantizer (VQ) applied to speech waveforms is shown in Fig. 1. This encoding scheme is described by a vector dimension  $k$ , a rate of  $R$  bits/vector (or  $r = R/k$  bits/sample or  $R_B = r \cdot F_S$  bits/s if the sampling rate is  $F_S$ ), a codebook  $\hat{A} = \{y_0, \dots, y_{N-1}\}$  of  $N = 2^R$  real  $k$ -dimensional vectors, an encoder mapping  $\alpha$  that maps an input source vector  $x = (x_0, \dots, x_{R-1})$  into a binary  $R$ -tuple or a binary channel vector  $u = (u_0, \dots, u_{R-1}) = \alpha(x)$  and a decoder mapping which transforms the received binary vector into a reproduction vector  $y = \beta(u) \in \hat{A}$ .

The decoder is a simple table lookup procedure. The optimal vector codebook  $\hat{A}$  is stored into a sufficiently large memory to hold an accurate digital representation of  $N = 2^R$   $k$ -dimensional vectors; the binary channel vector  $\hat{j}$ —possibly different from the encoder output  $j$  due to channel bit errors—is interpreted as the binary index of the reproduction vector to be output.

As in the Shannon theory of information, we define a nonnegative distortion measure  $d(x, y)$  that measures the distortion or cost of reproducing an input vector  $x$  as an output vector  $y$  and then attempt to minimize the average distortion  $Ed(x, \beta(\alpha(x)))$ . Having average distortion as a performance criterion, the optimum encoder for a given codebook  $\hat{A}$  follows the minimum distortion rule:

$$\alpha(x) = u \quad \text{if} \quad d(x, \beta(u)) \leq d(x, \beta(v)) \quad v \neq u \quad (1)$$

that is, send the binary vector which minimizes the distortion between the input vector and the reproduction. This is called a full search encoder since in general the encoder must search all  $N = 2^R$  codewords to find the minimum distortion vector. On the other hand, for a given

encoder  $\alpha$  the optimum codebook  $\hat{A}$  is formed as  $\alpha(\mathbf{u}) = \text{cent}(\mathbf{u})$  where  $\text{cent}(\mathbf{u})$  denotes the generalized centroid or center of gravity of  $\mathbf{u}$  defined as the vector  $\mathbf{y}$  minimizing the conditioned expectation  $E(d(\mathbf{x}, \mathbf{y}) | \alpha(\mathbf{x}) = \mathbf{u})$ , if such a centroid exists. Using a long training sequence to estimate averages, these two facts yield the basic iterative VQ design technique detailed in [1] and used here.

A major problem with such full search VQ's is the large search effort required to select a minimum distortion codeword. For example, for a 1 bit/sample 8-dimensional code with a squared error distortion measure, the encoder must compute 256 error energies of 8-dimensional vectors. In order to reduce the search effort of a VQ, some form of a tree searching technique can be employed in the encoder/decoder structure leading to a tree search vector quantization (TSVQ) system. Such systems trade off performance for speed and are considered for waveform coding applications in Gray and Linde [2] and in Gray and Abut [3]. Our focus here, however, is to demonstrate the implementability of a full search VQ of single and multiple codebook systems.

### B. Multiple Codebook Systems

In vector quantization and similar systems multiple codebooks have often proved a useful means of providing codes that are better suited to the local stationary behavior of sources such as speech. Single codebooks are designed for the long run average behavior and this can often result in a lack of reproduction words that match the short term behavior well. For example, words that match the mixture of a very noisy source (unvoiced speech) and an almost periodic source (voiced speech) may not well match source words from one or the other. Multiple codebooks may be designed for these separate modes of behavior and thereby better match the local variations. Such multiple codebooks often provide better performance for a given block length for sources such as speech whose short term locally stationary behavior can vary widely from the long term globally stationary behavior. For example, the tree encoding systems of Stewart *et al.* [9] match waveform coders to one of several possible LPC models for the locally stationary behavior. More simply, Wong *et al.* [8] used a simple voiced/unvoiced/silence decision to design separate codebooks for voiced and unvoiced sounds in an LPC vector quantization system. The decision mechanism is used to preclassify the incoming data and the subsets of the training sequence are then used to design the separate codes. Such codes can be viewed as a simple form of adaptive coding where only a relatively small number of codes are permitted.

We considered a simple two-codebook vector quantizer using the voiced/unvoiced/silence decision approach. The particular classifier was chosen because it was relatively simple, because it seemed a natural means of classifying speech for a two-codebook system, and because it had proven useful in an LPC vector quantization system. No claim is made that this is the best means of classification;

in fact the results indicate that this is not a good choice for a waveform coding system and that the performance gains do not justify the added complexity. The system does, however, demonstrate the feasibility of simple multiple codebook systems and other classification schemes, e.g., the vector quantized LPC classifiers of Stewart *et al.* [9] for tree encoding systems, may yield more significant improvement.

Fig. 2 is a block diagram of the two-codebook vector quantizer encoder using voiced/unvoiced classification where a decision of silence results in a single codeword rather than a codebook. Analog speech is bandlimited and sampled at a sampling rate of  $F_s = 6400$  samples/s. Every 10 ms a block of 64 samples is fed to the classifier and a vector quantizer buffer. The classifier issues a voiced/unvoiced/silence decision SUV every 10 ms based on a 20 ms data window. A 10 ms buffer is required to maintain synchronization since the quantizer needs a class decision before doing anything. The VQ takes  $k$  samples at a time and one SUV information and then searches through the voiced or unvoiced codebooks to find the minimum distortion codeword. In the case of a silence input from the classifier the quantization stage can be bypassed totally. The binary coder is a 1:1 mapping of SUV information and the quantizer outputs onto  $\{0,1\}^R$ . Every 10 ms,  $J = 64/k$  indexes and one SUV side information is transmitted. The classifier information of only 2 bits is required for each frame of 10 ms duration. If we assume that the source changes slowly compared to the vector lengths studied here, the data compression achieved by this technique can be significant as shown in the following numerical example. Let a particular speech segment of 100 frames—almost one second of speech and 64 samples/frame—be classified as 40 percent voiced, 20 percent unvoiced, and 40 percent silence. If all of these 100 frames were encoded by a rate  $r = 2$  and dimension  $k = 4$  quantizer, 12 800 bits would be needed. (There is no need for a side information channel.) If the scheme to be defined in (2) is used, then the number of bits needed for voiced, unvoiced, and silent portions would be 5200, 1320, and 80 bits, respectively. The total number of bits reduces to 6600 of which 200 bits are side information. This corresponds to a data rate reduction of nearly 50 percent.

Due to the computation requirements of the different blocks shown in Fig. 2, the system requires two parallel processors working in pipeline structure—one for the classifier and another one for the vector quantizer—both of which have sample-level fast operations. These two processors, the sample-data system, buffers, memories, the binary coder, and other units are integrated and controlled by a medium-speed general-purpose microprocessor,  $\mu P$ . Details of design and the performance results of this two-codebook VQ system will be discussed in Sections V and VI.

### III. COMPUTER SIMULATION RESULTS

Vector quantizers were designed and tested in six experi-

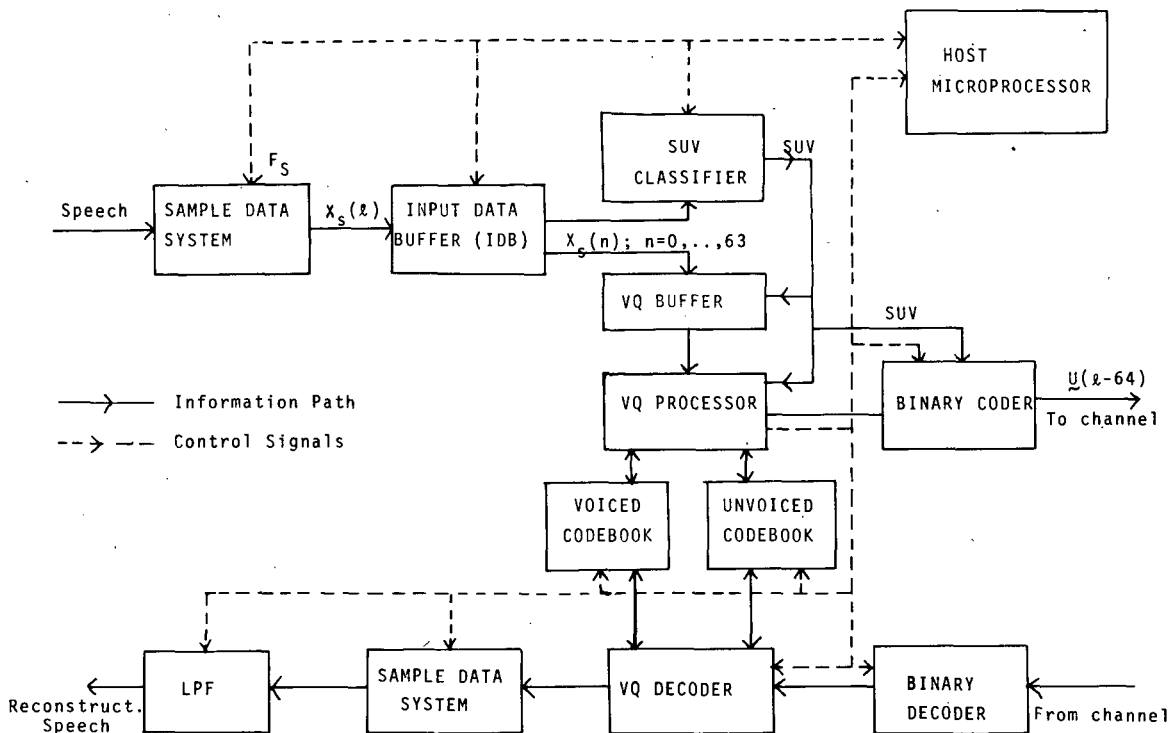


Fig. 2. Block diagram of the two-codebook VQ system.

ments for a variety of rates, dimensions, and data sets. Except for the two-codebook variable-rate cases, duplicate runs were made on separate computers using different programs. The first three experiments compare full search and tree search designs. The last three experiments investigate the performance of multiple codebook systems. Results from the first three and the last three experiments are summarized in Tables I and II, respectively.

*Experiment 1:* In Abut, Gray, and Rebolledo [1], codes were designed for rate  $r=1$  bit/sample and dimensions  $k=1, 2, \dots, 8$  and rate  $r=2$  bits/sample and dimensions  $k=1, \dots, 4$ . Full search VQ was used on a training sequence of 640 000 samples from five male speakers. The designed codebooks have been tested on a separate sequence of 78 000 samples recorded by a different speaker.

*Experiment 2:* In Gray and Abut [3], all of the above cases have been duplicated for binary tree searched vector quantizers and are presented here for comparison.

*Experiment 3:* The first experiment was repeated on training and test sequences in German. The design was accomplished off-line as usual, but the tests were conducted in real time using an array processor. There are a number of differences between the English and the German databases. The German data contained 20 min of recording from a German radio newscast spoken by several professional speakers. The data set was digitized at a rate of 8000 samples/s, whereas the English one had a rate of 6500 samples/s. Due to the limited computer storage capacity, the German data were then compressed into 8 bits using a 15 segment logarithmic quantizer. Full search VQ's were designed using a subset of 256 000 samples from the decompressed version of this database and a different set of 120 000 samples were used in testing the codebooks.

*Experiment 4:* The first database used in experiments 1 and 2 were decomposed into classes of voiced, unvoiced, and silent segments using a classifier similar to that in [5]. An optimal full search voiced codebook of rate one and dimension 8 was designed based on voiced segments only. Similarly, an unvoiced codebook of the same rate and dimension was obtained from unvoiced and silent segments. Class SNR's as well as the overall SNR were computed.

*Experiment 5:* The previous experiment was repeated for rate 2 and dimension 4.

*Experiment 6: Variable-Rate Universal Coding:* The voiced data subset was employed in designing a rate  $r=2$  dimension 4 codebook. A rate  $r=1$  dimension 4 codebook was obtained for the unvoiced segments. Finally, only 2 bits of SUV classifier side information are transmitted for every silence interval of 10 ms. The overall transmission rate including the side information channel is given by

$$R_B = (\text{SUV} + 64 \cdot B) \cdot (\text{frame rate}) \text{ bits/s} \quad (2)$$

where

$$B = \begin{cases} 0 & \text{silence} \\ 1 & \text{unvoiced} \\ 2 & \text{voiced} \end{cases}$$

the frame rate is 100 frames/s and SUV is a 2 bit side information.

Observation of Table I indicates that vector quantization provides a 6–8 dB performance improvement over optimal scalar quantization at the rates considered.

The quantitative results from the German database conform to those in English, even though the testing conditions were significantly different and, in particular, the



TABLE I  
SIMULATION RESULTS FROM SINGLE CODEBOOK EXPERIMENTS

Rate Bits per sample	Dimension k	Experiment 1		Experiment 2		Experiment 3	
		Full Search English SNR (db)		TSVQ English SNR (db)		Full Search German SNR(db)	
		Design	Test	Design	Test	Design	Real-Time Test
1	1	2.01	2.05	2.01	2.05	2.16	-
1	2	5.24	5.25	5.12	5.14	3.76	-
1	4	7.09	6.99	6.37	6.42	5.29	-
1	8	9.74	8.82	8.85	8.02	8.89	8.25
2	1	6.50	6.41	6.27	6.21	6.46	-
2	2	9.98	9.72	9.42	9.17	8.13	-
2	4	13.46	12.70	12.60	11.88	12.87	11.90

TABLE II  
SIMULATION RESULTS FROM MULTIPLE CODEBOOK EXPERIMENTS

	Experiment 4		Experiment 5		Experiment 6	
	Design	Test	Design	Test	Design	Test
Encoder Rates (bits/sample)						
Voiced Frames	1.0	1.0	2.0	2.0	2.0	2.0
Unvoiced Frames	1.0	1.0	2.0	2.0	1.0	1.0
Dimensions						
Voiced Frames	8	8	4	4	4	4
Unvoiced Frames	8	8	4	4	4	4
Class SNR's (db)						
Voiced Frames	10.73	9.05	14.84	13.24	14.84	13.24
Unvoiced Frames	9.06	6.43	10.42	8.52	5.97	5.21
Overall SNR (db)	10.32	8.74	13.47	12.53	11.12	11.59
Side Information Rate (bits/s)	100	100	100	100	200	200
Overall Transmission Rate (bits/s)	6,600	6,600	13,100	13,100	6,852	7,655

sampling rate was different. A dozen or more subjects, bilingual in both languages, concluded uniformly that the German reconstructed speech sounded better than the English ones. This is attributed to the faster sampling rate in the German database.

The SNR values for TSVQ's are within 1 dB of the full search results. Moreover, both systems were rated very close in numerous informal listening tests.

It is clear from Table II that the two-codebook system of Fig. 2 did not yield significantly superior results over that of the single codebook case. The improvement was always less than 1 dB. The variable rate cases of experiment 6, however, resulted in SNR's of 11.12 dB and 11.59 dB for the training and the test data, respectively. These numbers are within 1–2 dB of the results in experiment 5, but the observed rate reduction is very significant (41.6–47.7 percent). In addition, the recordings from these two experiments were presented to subjects and they have indicated that the differences were negligibly small.

#### IV. IMPLEMENTATION OF A SINGLE CODEBOOK SYSTEM

Here we describe the system structure for a single

codebook waveform vector quantizer with an encoding rate of either 1 bit/sample at dimension 8 or 2 bits/sample at dimension 4. Thus, both systems use 8 bit codebooks. The hardware architecture and the flowchart of the system are depicted in Fig. 3(a) and (b), respectively. It consists of the following components:

- 1) sample-data system consisting of a low-pass filter and an analog-to-digital converter (ADC).
- 2) 64-sample input data buffer (IDB).
- 3) quantization processor  $Q(\cdot)$ .
- 4) microprocessor system ( $\mu P$ ), including a Z80A CPU, RAM, EPROM, and I/O buffer logic.
- 5) codebook ROM.
- 6) output driver including a digital-to-analog converter (DAC), and a reconstruction LPF.

The overall number of devices is 82, comprising common "off-the-shelf" LSTTL, CMOS, and NMOS devices. The following is a description of the major functional modules and their respective chip count.

##### A. Preprocessor

The analog speech  $x(t)$  is filtered by a seventh-order Causer low-pass filter with a cutoff frequency of 3150 Hz

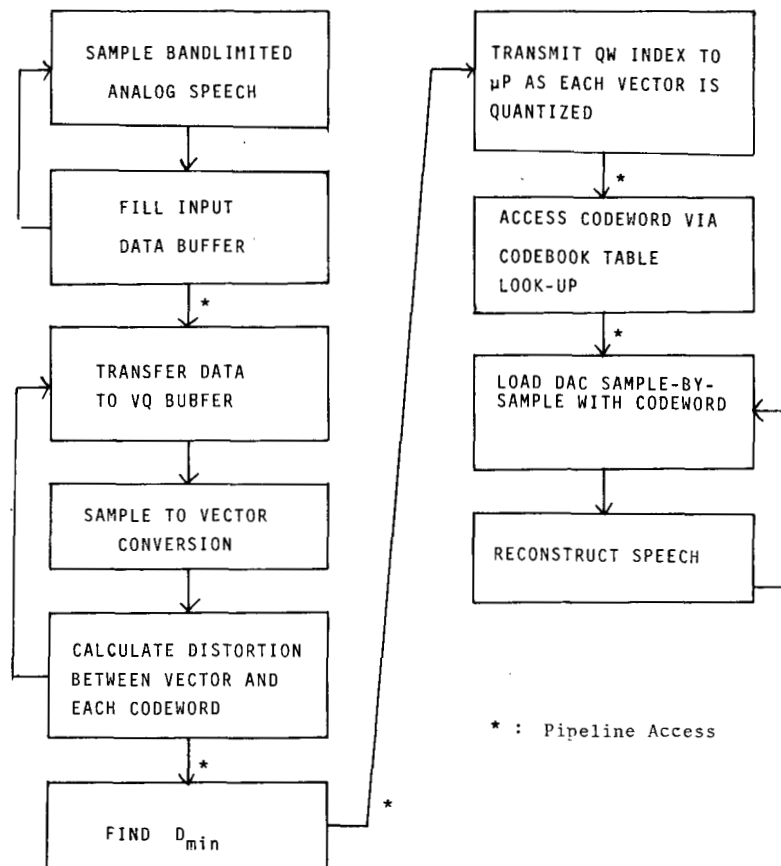
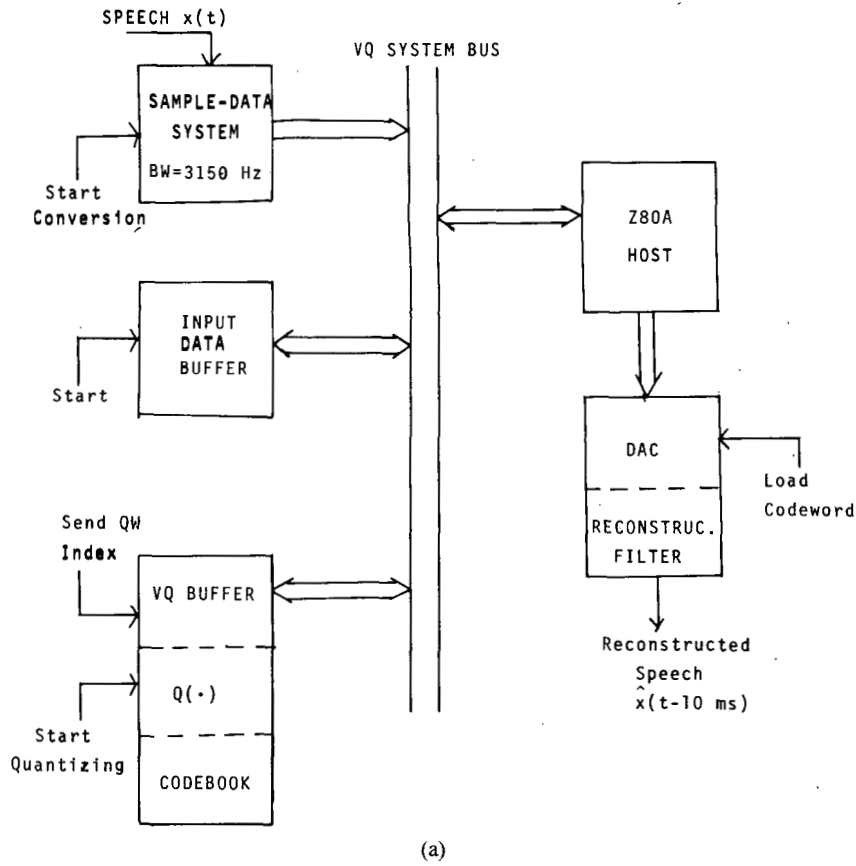


Fig. 3. (a) The hardware architecture of the single codebook VQ system. (b) Flow diagram of the single codebook VQ system.

and the gain is manually adjusted to match the dynamic range of the ADC,  $\pm 5$  V. The filtered signal is digitized at a sampling rate of  $F_S = 6400$  Hz, and the ADC accuracy is 8 bits  $\pm 1$  LSB. The sampled 8 bit data are then sequentially and byte-wise loaded into the 64-sample IDB. A new frame is started every 64 samples, and after the last sample of the current frame but before the first sample of the next frame, all 64 samples are transferred byte-wise to the quantization processor buffer (VQ buffer). This transfer is completed in  $57 \mu\text{s}$ , at a clock rate of  $F_C = 800$  ns/sample, and is initiated by the  $\mu\text{P}$  by forcing  $Q(\cdot)$  and IDB into a special direct-memory-access (DMA) mode. The pre-processor module requires 8 devices including the VQ buffer, consisting of mostly analog chips, a RAM, and a FIFO.

### B. Quantization

Following the DMA,  $Q(\cdot)$  processor starts the first of  $J = 64/k$  quantization cycles independent of the  $\mu\text{P}$ . The samples are accessed from the VQ buffer in modulo- $k$  base to simulate a serial-to-parallel converter where  $k$  is the vector size or the dimension of the system. The sequence of vectors and samples can be expressed in the following manner:

$$x^l = \{x^l(n) - ((k \cdot l + n))_m; n = 0, \dots, k-1, \\ m = 0, \dots, N-1, l = 0, \dots, J-1\} \quad (3)$$

where  $x^l$  is the  $l$ th of  $J = 64/k$  vectors in which  $x^l(n)$  is the  $n$ th sample, and  $N = 256$  is the number of codewords in the codebook ROM  $\hat{A}$ . The operation  $((\cdot))_m$  denotes the modulo addressing of the VQ buffer. Each vector  $x^l$  is quantized independently of adjacent vectors in full search cycles and each cycle is initiated by the  $\mu\text{P}$ .

The input vectors  $x^l$  are compared to 8 bit accuracy in a mean-square distortion sense to every codeword  $y_i$  for  $i = 0, \dots, 255$  in the codebook ROM, after which a minimum distortion codeword is selected. The codebook entries in ROM are extracted from the final codebooks of experiment 4. Since the system has an 8 bit architecture, high precision numbers from computer simulations are scaled down to 8 bit numbers before burning the PROM's.

The quantization process takes place in two pipelined sections of hardware. The first one computes the squared-error distortion between  $x^l$  and  $y_i$  at a particular codeword  $i$ , and the second module keeps track of a running minimum distortion.

Due to design requirements, one full search cycle time (FSC) has been kept to 500 ns per distortion computation. This speed is accomplished by the use of high-speed memories, low-power Schottky logic, and a table lookup for the squaring operation. In addition, any accumulation overflow is monitored in parallel with the distortion computation by employing a look-ahead logic. In particular, if an overflow occurs, it is flagged without waiting until the end of the calculation, therefore no extra time is required for overflow detection.

Since one distortion calculation is carried out every 500

ns, the time needed for one full pass through the codebook is

$$T_{\text{FSC}} = (8 \text{ samples/vector}) (500 \text{ ns/sample}) \\ \cdot (256 \text{ vectors/cycle}) \\ = 1.024 \text{ ms/cycle.}$$

Actually,  $T_{\text{FSC}}$  is a few microseconds more than 1.024 ms in order to account for the delay resulting from the pipeline synchronization. After each  $T_{\text{FSC}}$ , the final codebook index is loaded into the quantizer word mail box register (QW). Every  $k$  sample clocks or 1.248 ms the host  $\mu\text{P}$  reads the value of QW, but only after a full search cycle is completed. The quantizer module requires 45 chips including SSI and MSI circuits, and a high-density bipolar RAM.

### C. Postprocessing

If the system is in the encoding mode then the host  $\mu\text{P}$  sends the index of the minimum distortion codebook to the binary coder so that the digital signal is conditioned for the channel. If it is in the decoding mode, however, then the  $\mu\text{P}$  points to the codeword  $j$  in the ROM with the first byte in row  $j \cdot k + k - 1$ , the second byte in row  $j \cdot k + k - 2$ , etc. The bytes are accessed synchronously with the sampling rate  $F_S$  and are sent to the output driver section for digital-to-analog conversion. This process is completely controlled by the  $\mu\text{P}$ . After the DAC, the reconstructed signal is filtered by an LPF similar to that in the pre-processing stage.

After sample reconstruction,  $Q(\cdot)$  sends a flag to the  $\mu\text{P}$  to indicate completion. Although the  $\mu\text{P}$  software can predict the end of the cycle, this flag is monitored as a safety measure in case of synchronization problems. At this point, a new frame is loaded into the IDB, and the  $\mu\text{P}$  starts another 64-sample DMA cycle from the IDB to VQ buffer, which, in turn, begins another quantization cycle. This stage needs nine chips including analog devices.

### D. Host Microprocessor $\mu\text{P}$

It is observable from the system architecture of Fig. 4 that the primary functions of the  $\mu\text{P}$  are timing, control, and the interfacing of various modules. There is very little numerical processing undertaken by the host. In addition to the Z80A system, there are six modules on the  $\mu\text{P}$  bus: one general purpose working space RAM as a data buffer, a ROM containing the software, DAC system, an I/O buffer for interfacing with the VQ system bus, an external multiplier chip for high-speed multiplications (needed in the clipping stage of the classifier module and it is used only in the design of the two-codebook encoding scheme), and a ROM for the codebook  $\hat{A}$ . The total number of chip count here is 20.

### E. Testing

The system has been tested both quantitatively by monitoring the resulting minimum distortion  $d_{\min}$  and subjec-



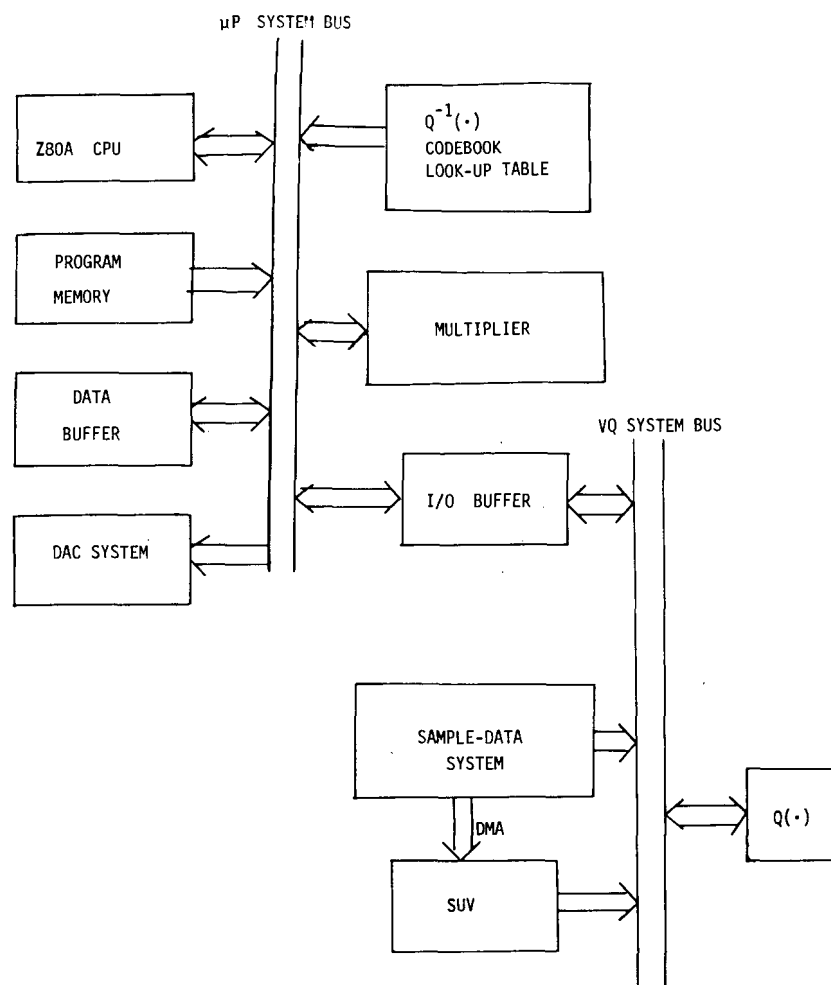


Fig. 4. Bus architecture for the VQ system and the host microprocessor.

tively by informal listening tests. The hardware was designed for a dynamic range of 8 bits with a resolution of one LSB. Therefore, the accuracy of quantization is 1 part in 256 and the squared error distance between any two vectors can be a minimum of 1 bit of full scale 8 bits. In the quantitative measurements, vectors from a test data set were used; and hence,  $d_{\min}$  for each measurement was exactly known and the minimum distortion codebook index  $j$  could be predicted. For this particular test, all 256 codewords in the ROM were used as the input data sequence and the resulting  $d_{\min}$  was zero. The minimum distortion codeword indices were found without a single error.

Next the system was tested with real speech prerecorded and played back on a cassette tape deck containing the English alphabet, digits, words, and phrases. The distortion was very low during the silent intervals because the codebook ROM contained a null vector and vectors with magnitudes close to zero. The test yielded fairly intelligible but noisy output for voiced speech segments. There were, however, severe errors in encoding fricative sounds because the employed codebook was not optimal for unvoiced sounds. The minimum distortion measurements were re-

corded for each vector and the accumulated measurements resulted in an overall SNR of 6.25 dB. This figure is about 2 dB below the simulation results of Section III. We believe that the major cause of degradation is due to the dynamic range problem (i.e., the incompatibility between the codebook and input signal variances) and the absence of a reasonable AGC mechanism to facilitate an 8 bit A/D conversion.

## V. DESIGN OF A TWO-CODEBOOK VQ SYSTEM

We present the design structure of a two-codebook waveform quantizer with a rate of 2 bits/sample and dimension 4 for voiced data, 1 bit/sample and dimension 4 for unvoiced sounds, and a null vector for silent periods. The basic block diagram of the system is given in Fig. 2. Here we discuss the actual design of various modules. The overall system of Fig. 2 can be broken into the following building blocks:

- preprocessor comprising a preamplifier, an LPF, the sample-data system (ADC), and an input buffer (IDB).
- SUV classifier.
- quantization processor  $Q(\cdot)$ .

- d) voiced and unvoiced codebooks  $\hat{A}_v, \hat{A}_u$ .
- e) host microprocessor system  $\mu P$ .
- f) transmitter coder (XMTR) and receiver decoder (RCVR).
- g) postprocessor.

We will not be discussing items a), c), and g) since they are identical to the corresponding blocks in the case of the single codebook design.

### *SUV Classifier*

It is basically a pitch detector working as a silence/unvoiced/voiced (SUV) classifier, issuing one decision every 10 ms based on 20 ms analysis windows with 50 percent overlapping. The reason why we build a full pitch detector for this restricted task has been to avoid duplication of work in our present effort on designing pitch synchronous speech compression systems based on VQ encoding techniques. The SUV classifier module is a simplified version of the trilevel clipping and autocorrelation method by Dubnowski *et al.* [6]. The simplification entails the use of 20 ms windows, instead of 30 ms, in which the clipping levels are determined every 10 ms. Every 10 ms a new segment is loaded, thus new levels are computed 100 times/s. While the new segment is loaded, the trilevel clipping process and subsequent autocorrelation process take place. The output of this module is a 2 bit SUV information. The design details of this module can be found in [7].

### *Voiced and Unvoiced Codebooks, $\hat{A}_v, \hat{A}_u$*

Once a SUV decision is made, the quantization processor  $Q(\cdot)$  switches to the appropriate codebook and it freezes there until a different SUV value shows up. This switching and freezing action is done by the host microprocessor.

### *Microprocessor System $\mu P$*

The Z80A CPU is the heart of this part. The  $\mu P$  monitors the system as in Section IV using the Z80A software. Specific tasks are:

- 1) initialize the system.
- 2) start each processing cycle.
- 3) load and control the SUV queing register,  $Q_n$ .
- 4) perform clipping level computations.
- 5) SUV decisions.
- 6) coder and decoder actions.
- 7) I/O control.

### *Transmitter Coder (XMTR) and Receiver Decoder (RCVR)*

The transmitter coder packs the data and the side information of each 10 ms frame, standardizes the package in some known format, and sends the final bit stream through the channel. In the receiver side the above functions are undone by a binary decoder.

The data packets are 66 bits long, consisting of a 2 bits SUV header and 64 bits codebook indexes. The indexes are 8 bits long for each voiced vector of dimension 4. In the case of unvoiced data, however, the indexes are only 4 bits. Finally, for silent periods a string of 64 zeros are attached to the appropriate header. Experiments have shown that more than 30 percent data compression could be achieved if the zeros in silent periods are suppressed.<sup>1</sup>

As previously mentioned, the postprocessor reconstructs a replica of the original speech with a 10 ms delay. All the design details of the functional units in this section are explained by Tao [7].

## VI. CONCLUDING REMARKS

Hardware realizations of both single codebook full search vector quantizers and multiple codebook full search vector quantizers were successfully completed and tested.

Comparison of the resulting performance with that of several old and new simulations indicates that the performance of the hardware VQ's were about 2 dB inferior to the simulated VQ's. In addition, this degradation was clearly noticeable in listening tests as well. We believe that this is primarily due to the absence of any AGC mechanism in A/D and D/A conversions. The reason for not including such a block was to follow the simulations exactly.<sup>2</sup> To improve the performance, an AGC mechanism is recommended to match the dynamic range of the input speech to that of the codebook. In this way, absolute accuracy to 8 bits can be maintained at the cost of a few bits of side information for the gain factor. However, the hardware developed in this study proved that waveform coding data compression systems can be implemented using full search 8 bit codebooks with readily available devices. It should also be noted that the architecture of the VQ processor designed here is within VLSI capabilities [7] and it can be developed into a single-chip codec. Parallel use of new 16 or 32 bit devices should also improve the quality of such coders by permitting higher accuracy codebooks and the resolution in distortion computations. The architecture of such a system would be essentially that outlined here modified to take advantage of the better microprocessor. The implementability of full search VQ of up to 8 bits was further attested by the real-time simulations conducted with the aid of an array processor. The superior quality of these simulations supports our belief that improved codebook and arithmetic accuracy will yield improved quality.

<sup>1</sup>This percentage is a very conservative figure and the thresholds in the classifier have been adjusted to make errors, if any, in favor of classifying a silent period as voiced or unvoiced depending upon what comes before or after the questionable segment. Depending on the data set and the classifier thresholds, close to 50 percent data rate reduction can be achieved as in TASI systems.

<sup>2</sup>Training database and the test sequence were used in designing the codebooks without any signal shaping or normalization.

## ACKNOWLEDGMENT

The authors would like to thank National Semiconductor Speech Processing Laboratory for permission to use their facilities and for supplying the necessary electronic parts. The authors would also like to acknowledge Prof. D. Wolf and his research team at Institut für Angewandte Physik der Johann-Wolfgang-Goethe Universität, Frankfurt, Germany, for their help in realizing the real-time simulations.

## REFERENCES

- [1] H. Abut, R. M. Gray, and G. Rebolledo, "Vector quantization of speech and speech-like waveforms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 423-435, June 1982.
- [2] R. M. Gray and Y. Linde, "Vector quantizers and predictive quantizers for Gauss Markov sources," *IEEE Trans. Commun.*, vol. COM-30, pp. 381-389, Feb. 1982.
- [3] R. M. Gray and H. Abut, "Full search and tree searched vector quantization of speech waveforms," *Proc. IEEE ICASSP'82*, May 1982, Paris, France.
- [4] Y. Yamada, private communication.
- [5] B. S. Atal and R. L. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 201-212, June 1976.
- [6] J. J. Dubnowski, R. W. Schafer, and L. R. Rabiner, "Real-time digital hardware pitch detector," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 2-8, Feb. 1976.
- [7] B. Tao, "A real-time speech data compression system using vector quantization," M.Sc. thesis, Dep. Elec. Comput. Eng., California State Univ., Long Beach, 1982.
- [8] D. Y. Wong, B.-H. Juang, and A. H. Gray, Jr., "An 800 bit/s vector quantization LPC vocoder," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 770-780, Oct. 1982.
- [9] L. C. Stewart, R. M. Gray, and Y. Linde, "The design of trellis waveform coders," *IEEE Trans. Commun.*, vol. COM-30, pp. 702-710, Apr. 1982.



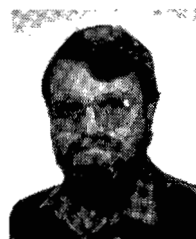
**Bertram P. M. Tao (S'77-M'82)** was born in Alexandria, LA, on March 31, 1955. He received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1977 and the M.S. degree in electrical engineering from California State University, Long Beach, in 1982.

In 1977 he joined Hughes Aircraft Co., Fullerton, CA, and he worked for Ford Aerospace, Newport Beach, CA. In both companies he was involved with the software simulation and hardware integration of image processing and pattern recognition systems. Presently, he is with Hycom Inc., Irvine, CA, as a Manager of the Speech Technology Section where he is directing advanced product development and research in speech recognition and vocoding.



**Hüseyin Abut (S'70-M'74-SM'79)** was born in Akhisar, Turkey, on November 20, 1945. He received the B.S. degree in electrical engineering from Robert College, Istanbul, Turkey, in 1968, and the M.S. and Ph.D. degrees from North Carolina State University, Raleigh, in 1970 and 1972, respectively.

In 1972 he joined the Electronics Division, Marmara Research Institute, Gebze, Turkey. During 1973-1980 he was with the Departments of Electrical Engineering and Mathematics, Boğaziçi University, Istanbul, Turkey. He was a Visiting Scholar at the Information Systems Laboratories, Stanford University, Stanford, CA, and a Visiting Associate Professor in the Department of Electrical and Computer Engineering, California State University, Long Beach, in 1979-1980. He was with the National Semiconductor Corporation Speech Laboratory during 1980-1981. Since 1981 he has been a Professor in the Department of Electrical and Computer Engineering at San Diego State University, San Diego, CA. During the Summer of 1982 he was a Visiting Professor in the Institut für Angewandte Physik, Frankfurt University, Federal Republic of Germany. His current technical interests include digital speech and image encoding, speech analysis, synthesis and recognition, and fast computational algorithms.



**Robert M. Gray (S'68-M'69-SM'77-F'80)** was born in San Diego, CA, on November 1, 1943. He received the B.S. and M.S. degrees from the Massachusetts Institute of Technology, Cambridge, in 1966 and the Ph.D. degree from the University of Southern California, Los Angeles, in 1969, all in electrical engineering.

Since 1969 he has been with the Information Systems Laboratory and the Department of Electrical Engineering, Stanford University, CA, where he is currently a Professor engaged in teaching and research in communication and information theory with an emphasis on data compression.

Dr. Gray was a member of the IEEE Information Theory Group Board of Governors (1974-1980), was Associate Editor for Shannon Theory (1977-1980) and Editor (1980-1983) of the IEEE TRANSACTIONS ON INFORMATION THEORY, and was corecipient of the 1976 Information Theory Group Paper Award. He was a Fellow of the Japan Society for the Promotion of Science (1981) and the John Simon Guggenheim Memorial Foundation (1981-1982). He is a member of Sigma Xi, Eta Kappa Nu, the Society for Industrial and Applied Mathematics, the Institute of Mathematical Statistics, the American Association for the Advancement of Science, and the Societe des Ingenieurs et Scientifiques de France. He holds an Advanced Class Amateur Radio License (KB6XQ).