

LOW COST HARDWARE FOR REAL-TIME LPC ANALYSIS AND SYNTHESIS OF SPEECH

PAUL GRIFFITH AND IAN H. WITTEN

DEPARTMENT OF ELECTRICAL ENGINEERING SCIENCE, UNIVERSITY OF ESSEX

Linear predictive coding (LPC) of speech is currently being used in applications as diverse as articulation studies, low bit-rate vocoders and "speaking toys". Consequently, a need exists for simple and inexpensive systems capable of encoding and/or synthesis in real time.

The synthesis process is fairly straightforward and many digital hardware implementations abound. Indeed, Texas Instruments have recently developed a single chip speech synthesiser [1] which forms the basis of their \$50 "Speak & Spell" toy.

The encoding process is more complex and requires both the LPC analysis and pitch extraction. This, coupled with the lower sales potential of encoding systems compared to synthesis-only ones, makes it unlikely that custom LPC encoding chips will be developed for some time. However, there will be an increasing need for analysis systems, if only to produce vocabularies for the many applications of the cheap synthesis chips.

Real-time LPC encoding is difficult because it demands both (a) repetitive, high-speed arithmetic operations and (b) substantial general-purpose (g.p.) data manipulation. It is this mix which gives the problem piquancy. The g.p. processing requirement complicates a custom-hardware approach while (minicomputer) programmed implementations are too slow for all but low quality analyses [2]. Simply adding a fast arithmetic unit (AU) does not solve the problem because the overheads of data transfer, address calculations and other housekeeping remain.

One solution is to build what is, in effect, a g.p. computer with the requisite high-speed arithmetic capability [3]. The main drawbacks of this approach are that the bulk of the circuitry must operate at the high speed of the AU, and the highly custom nature of the device precludes the use of inexpensive g.p. LSI components such as microprocessors.

This paper outlines a new approach which promises a simple and economic realisation of an LPC vocoder.

Dual-Processor Structure. Instead of attempting to fit the requirements of (a) and (b) above onto a single machine, we propose to implement them on two co-operating processors. Specifically, a conventional 16-bit microprocessor handles the g.p. processing while a fast but simple, fixed-point vector processor (VP) performs arithmetic operations upon command from the microprocessor. This achieves economy, firstly by using high-speed circuitry only where it is needed, and secondly by making extensive use of standard LSI devices.

We chose this approach for several reasons [4]. Briefly: (a) The analysis and synthesis algorithms can be succinctly expressed in the form of vector

Proceedings of The Institute of Acoustics

LOW COST HARDWARE FOR REAL-TIME LPC ANALYSIS AND SYNTHESIS OF SPEECH

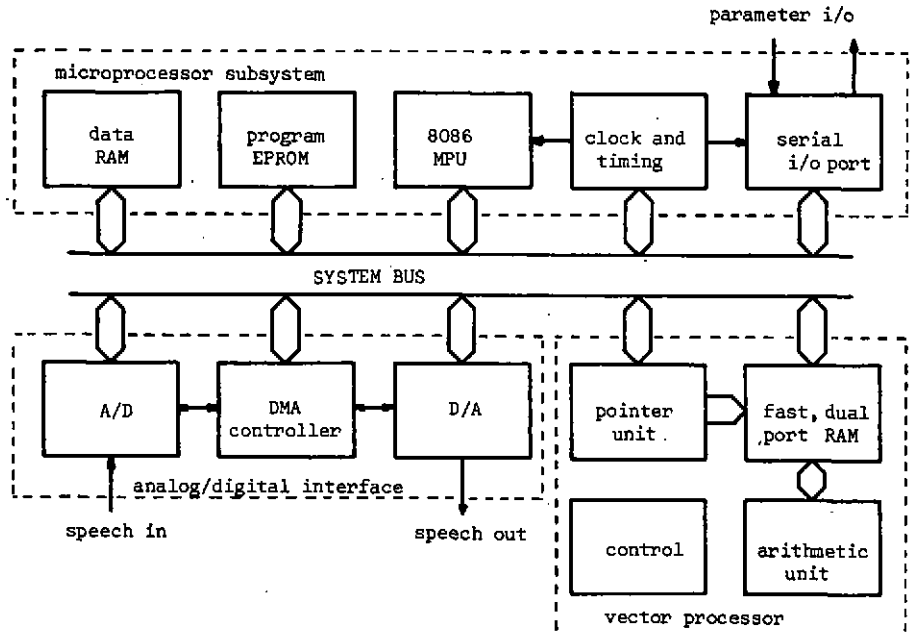
operations. (b) The natural ordering of vector data reduces the task of operand address arithmetic to simply incrementing pointers to the elements concerned. (c) Only a few, simple arithmetic operations are needed and therefore the VP can be kept simple and inexpensive. (Contrast this to the complexity and cost of the AP-120B array processor implementation described by Cole & Boynton [5]).

The VP instruction set contains the basic vector operations together with two specialised instructions - these being included for efficiency. We favour 3-address memory to memory instructions of the form $D \leftarrow A \text{ op } B$ since this minimises data transfers. All vector operations act upon data in an area of fast store which is accessible to both processors. The VP instruction set is summarised below. Note that $A[]$, $B[]$ etc represent vector quantities and A , D etc are scalars.

move:	$D[] \leftarrow A[]$	subtract:	$D[] \leftarrow A[] - B[]$
add:	$D[] \leftarrow A[] + B[]$	inner product:	$D \leftarrow A[0]*B[0] + A[1]*B[1] + \dots$
scale:	$D[] \leftarrow A * B[]$	outer product:	$D[] \leftarrow A[] * B[]$

lin_scale: $D[] \leftarrow A * B[]$ with $A \leftarrow A + \Delta A$ at each iteration
synthesis: Specialised speech synthesis instruction

Implementation. The system's architecture is shown below. It partitions into three modules, namely a microprocessor subsystem, the VP and the Analog/Digital interface. These will be briefly discussed in turn.



Proceedings of The Institute of Acoustics

LOW COST HARDWARE FOR REAL-TIME LPC ANALYSIS AND SYNTHESIS OF SPEECH

The microprocessor subsystem uses Intel's 16-bit 8086 device. EPROM will be used for program storage in the final version to allow stand-alone operation, but the prototype is RAM centred and uses the services of a host machine for software development and down-loading. A serial port is provided for i/o of encoded speech. The necessary timing signals to control events are generated by an LSI counter/timer device driven from the system clock.

The VP is controlled by the 8086. Once invoked it proceeds autonomously until the vector operation is complete when it causes an interrupt. Parameters are pre-loaded (by the 8086) into the lower locations of its local store and are automatically copied into internal registers at the start of execution. The local store has separate ports for the system bus and VP AU so that vector operations can take place concurrently with, and at a faster rate than system bus transfers.

The AU performs 16-bit fixed-point arithmetic and uses a single-chip LSI multiplier/accumulator to minimise package count and complexity. The pointer unit is responsible for operand address and iteration count calculations and is built from 2901 ALU/register file slices. A microprogrammed control unit was chosen because of its flexibility and ease of design. Real-time requirements are met by a microcycle time of 200 ns and therefore standard AMD 2900 series parts are adequate.

The Analog/Digital interface comprises the usual analog signal-conditioning and conversion hardware. Input and output data transfers are performed under DMA control to reduce the load on the 8086.

Analysis algorithm. For LPC analysis we use the well-known autocorrelation technique. This produces a set of reflection coefficients together with the speech energy. We intend to meet LPC-10 specifications, namely a 10th order analysis with a frame period of 22.5 ms.

The first task is to copy speech samples from the i/p ring buffer to a working buffer. This is done with the VP move instruction. Digital pre-emphasis could be applied at this stage but we prefer to pre-emphasise the analog speech signal. Windowing is then performed using the VP outer product instruction.

Autocorrelation coefficients are computed with inner product instructions. For accuracy, these are accumulated in double-precision. Since we are only concerned with their relative ratios, we normalise them (by left shifting until the magnitude of the zero'th coeff. is between 0.5 and 1) and then truncate to single length. The original value of the zero'th coeff. is also retained since this equals the rms energy of the windowed, input speech. Overflow can be prevented by suitable scaling [4].

The Durbin/Levinson recursion is normally employed to obtain the reflection coefficients and analysis filter parameters. We require only the former which allows us to use an alternative recursion due to Green [6]. This recursion requires less than half the computation of the D/L one, is readily structured into vector form and has the valuable property that all the quantities are fractional.

Proceedings of The Institute of Acoustics

LOW COST HARDWARE FOR REAL-TIME LPC ANALYSIS AND SYNTHESIS OF SPEECH

Synthesis algorithm. While analysis requires buffering of data, previous implementors have employed strictly real-time synthesis with the o/p samples being passed to the D/A converter as soon as they are calculated. We differ from this and generate a whole frame of speech in a working buffer.

Our first task is to fill this buffer with excitation samples. Voiced sounds utilise pulse waveforms stored in PROM. The `lin_scale` instruction performs the necessary transfers and also provides gain control and interpolation. Unvoiced sounds use a pseudo-random excitation. We use a Fibonacci sequence since it provides adequate randomness and is readily implemented, requiring only a single VP add operation. As before, `lin_scale` is used to control and interpolate the excitation level.

The synthesis filter is the well-known normalised lattice form. For reasons of efficiency a specialised VP instruction is used. This reads an excitation sample from the buffer, calculates the o/p value, stores that in the buffer, then reads the next excitation and so on until the buffer's end is reached. Filter parameter interpolation at every sample point is also performed. Look-up tables containing k and $\sqrt{1 - k^2}$ value pairs are employed to hasten this task. De-emphasis is left to analog hardware.

Pitch-extraction and other tasks. At the time of writing we have not decided upon a pitch-extraction technique. However, we will probably use either a time-domain method such as Gold/Rabiner or the SIFT technique. Without discussing their relative performances we note that the former will require additional (analog) hardware to extract features from the speech waveform, while the computational demands of the latter preclude any possibility of full-duplex operation.

Finally, although analysis, synthesis and pitch-extraction are the major operations, real-time working requires a host of other tasks such as parameter i/o, coding and decoding, channel error correction, buffer manipulation, synchronisation etc.

References.

- [1] Wiggins, R. & Brantingham, L., "Three-chip system synthesises human speech", Electronics, (August 31 1978), 109-116.
- [2] Crichton, R.C. & Fallside, F., "Linear model of speech production with application to deaf speech training", Proc IEE 121 (1974) 865-873.
- [3] Hofstetter, E.M. et al, "Microprocessor realisation of a linear predictive vocoder", IEEE Trans ASSP, vol ASSP-25 (1977) 379-387.
- [4] Witten, I.H. & Griffith, P., "A dual-processor structure for high-quality real-time linear predictive analysis of speech", Internal report, Dep't of EES, Essex University, 1979.
- [5] Cole, R.E. & Boynton, T.L., "A real time floating point variable frame rate LPC vocoder", Technical report, USC Information Sciences Institute.
- [6] Green, N., "An algorithm for calculating reflection coefficients", CESC memorandum T5A/1/77, Gov't Comm'n HQ, Cheltenham, 1977.