A FORMANT SPEECH SYNTHESISER ASIC: FUNCTIONAL DESIGN

Clive D Summerfield†, and Marwan A Jabri†
† School of Electrical Engineering
The University of Sydney

ABSTRACT - This paper is the first of two companion papers on the design and implementation of a multi-channel formant speech synthesiser Application Specific Integrated Circuit (ASIC). The objective of this research is the development of an efficient VLSI structure which can be implemented as a single VLSI device and yet retains the acoustical performance necessary to generate high quality and high intelligibility synthetic speech and have sufficient processing bandwidth for multi-channel operation. This paper concentrates on the functional design of a VLSI formant speech synthesis structure for achieving these objectives.

INTRODUCTION

Over the past 40 years, Formant Speech Synthesis has become an established and widely used method for generating synthetic speech output. With the rapid expansion in Information Technology projected for the next decade, multi-access speech response services will play an increasing important role in information dissemination and communication.

Extensive research by Holmes has shown that parallel formant synthesiser structures are capable of exceptionally high quality and intelligibility speech synthesis (Holmes, 1982 and Clark, Summerfield & Mannell, 1986). However, as well this, there is very strong engineering motivations for adopting a parallel synthesis architecture. Structural regularity has a profound effect on the efficiency of VLSI implementation. Both serial and hybrid synthesiser suffering from high degrees of structural irregularity and require more second order sections to perform the same task as the parallel synthesiser. As well as this, the cascade filter structures used in these design demand very careful implementation if dynamic range problems are to be avoided.

However, as it stands, the Holmes design has a number of structural problems. Irregularity in the high frequency filter arrangements is very inefficient from a VLSI engineering standpoint. The VLSI formant filter structure these have been replaced by a parallel connection of two formant filters to produce a regularised parallel structure similar to that reported by Clark, Summerfield and Mannell (1986). The complete structure contains six formant filter channels, designated, F1 to F5 and a supplementary resonator, FN. Channels F1 to F5, model the time varying acoustical transfer characteristics of the oral cavity. The supplementary resonator, FN, models nasal cavity resonance and provides a mechanism for controlling the low frequency response of the synthesiser. Five fixed filters are used at the output of formant channels F1 to F5 modify the response of the resonators to allow correct mixing of the formant channels and are similar to those as described by Holmes (1982).

Crucial to the efficiency of the VLSI implementation and the acoustical performance of the synthesiser is the internal data representation and resolution. The bit-serial data representation is ideal for the VLSI implementation of speech signal processing functions. As bit-serial data communications are along single wires, the communications overhead which can dominate the structure in complex signal processing functions is small. For good quality speech output it is necessary to synthesise speech with a very high dynamic range. Experiments by the author show clearly that critical parts of the circuit need to be implemented with extremely high precision to prevent "limit-cycles" affecting the speech quality (Summerfield, 1988). In fully synchronous bit-serial architectures, processing bandwidth and data resolution (the number of bits) can be "traded-off" against each other without unduely affecting the overall physical size of the VLSI device. Although this not a simple linear relationship far more preferable than the relationship that exists for bit-parallel representations. The data word length used in the present implementation is 16 bits, with double precision bit-serial representation used to implement critical functions.

The VLSI formant speech synthesiser has been developed exclusively using the methods of "functional design" as implemented in the Denyer/Renshaw FIRST Silicon Compiler (Denyer & Renshaw, 1985). FIRST offers a hierarchical design environment which is designed for the rapid implementation of fully synchronous bit-serial signal processing architectures. FIRST provides four hierarchical level of circuit abstraction; SYSTEM; CHIP; OPERATOR and PRIMITIVE, in descending order. The description in this paper is restricted to the OPERATOR and PRIMITIVE levels of circuit abstraction.

FUNCTIONAL OPERATOR STRUCTURE

The first step in achieving a functionally correct VLSI design is to define a functional operator hardware description of the synthesis circuit. The functional operator structure used to implement the parallel formant synthesiser filtering functions is shown in figure 1. This structure contains five functional operators which perform the excitation mixer/gain controls, coefficient generation, resonance filtering, the F1 fixed filter and the output combination.

At the functional operator level, the complete formant synthesiser filtering operation is implemented using a fully multiplexed six phase clocking scheme. In this strategy, acoustic parameter are queued at the input nodes to the mixer/gain and coefficient generator operators. The computational latencies of these operators is matched so that the composite excitation function values and resonance filter coefficients arrive at the the inputs to the second order filter operator synchronously. The output nodes from this operator connect to the F1 fixed filter and output combiner operators. Multiplexers within the output combiner operator selects the appropriate input node corresponding to the phase of the multiplexing clock and accumulates the formant channel sample values to generate the output speech waveform.

Detailed descriptions of the primitive structure of these functional operators have been published elsewhere (Summerfield, 1986, 1987(1), 1987(2), 1988). Here, only a brief algorithmic description is present which is sufficient to clarify the operation of the VLSI synthesiser.

The mixer-gain operator controls the degree of voiced, $U_g'(t)$, (where $U_g'(t)$ represents the radiation corrected version of the glottal volume-velocity function), and frication, $U_f(t)$, components in the composite excitation function applied to the resonant filters. The algorithmic description of the mixer-gain operator is given by

$$e_j(t) = Gv_j U_g'(t) + Gf_j U_f(t), \tag{1}$$

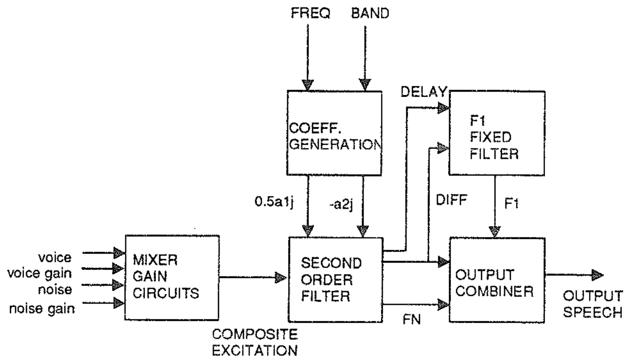

Figure 1: Functional operator description of the VLSI formant synthesiser.

where Gv_j and Gf_j are the voiced and fricative gains for the jth formant channel, respectively. This model is a variation of the "sliding scale" mechanism used by Holmes (1982). It provides absolute control is over the amount of voiced and fricative excitation applied to the formant filter. This was done primarily to minimise the number of multiplier primitives necessary to implement the operator, although it also provides a more flexible approach to excitation function control which could potentially be useful for controlling the output voice quality.

The resonance filter coefficients, $a1_j$ and $a2_j$, (for the jth formant channel) are calculated from the formant frequencies, f_j, and bandwidths, b_j, using the product terms $a1_j/2 = F'_jB'_j$ and $-a2_j = B'_j{}^2$, where F'_j and B'_j, are provided by external look-up tables defined by $F' = cos(2\pi f\tau)$ and $B' = e^{(-\pi b\tau)}$. (τ is the sampling interval.)

The resonator difference equation is modified to compensate for the coefficients generated by this method as is given by

$$o_j(t) = e_j(t) + 2o_j(t-\tau)(a1_j/2) - o_j(t-2\tau)(-a2_j). \quad (2)$$

The output of the jth formant channel, $o_j(t)$ is construct from the input excitation function, $e_j(t)$ combined with two product terms computed from previous resonance filter output values and the coefficients. Previous output values, $o(t-\tau)$ and $o(t-2\tau)$, are stored in two recursively connected shift registers and the difference calculation is performed by a primitive structure consisting of two multipliers and three adder/subtractor primitives.

TWO-MULTIPLIER PRIMITIVE ARCHITECTURE

The operator structure shown in figure 1 was developed originally using a six multiplier bit-serial primitive structure (Summerfield, 1987(1), 1987(2)). A pipeline arrangement of three, two-multipler structures implement the mixer-gain, coefficient generator and resonance filter operators following the data flows shown in figure 1. Because of the high number of multiplier primitives used, the structure proved to be extremely large, and was impractical to fabricate as a single VLSI device. However, even for the obsolete 5 micron nMOS technology, in which the circuit was originally designed, the processing bandwidth was far in excess of that necessary for real-time speech synthesis. Even in this form, the device was capable of achieving the multi-channel operation. The greater processing bandwidth with 2 micron CMOS fabrication technology enables higher levels of multiplexing to be used. This reduces the number of multiplier primitives needed to implement the synthesis function, making a single chip implementation feasible and, at the same time, retaining sufficient processing bandwidth for multi-channel synthesis operation.

The two-multiplier primitive structured developed for CMOS fabrication is shown in figure 2. Central to this structure are the two double precision, 16 bit, bit-serial multiplier primitives. Inputs to the multipliers are controlled by a bank of multiplexers which schedule the multiplier operations. The multiplier output connect to a set of double precision bit-serial adder and subtractor primitives which perform the primitive functions to complete the mixer-gain and resonance filter operations.

Synthesis of each formant channel is controlled by a three phase multiplexer clock. On the first phase, the multiplexers select the mapped formant frequency and bandwidth value, F'_j and B'_j. The coefficients $a1_j/2$ and $-a2_j$ are fed back via format converters and synchronising delay line elements (D2) to the input of the multiplexer to be used on a later phase during the resonator calculation. On the second phase, the multiplier inputs are connected to the excitation function generators and the gain controls. The output of the multipliers is combined in adder primitive, A2, to generate the composite excitation sample value. Shift register delay lines (D1) are used at the to synchroise the arrive of the excitation function with the resonance filter calculation. This calculation is controlled on the third clock phase by connecting the multiplier inputs to the previously generated coefficients and the bit-serial recursive delay lines, SR1 and SR2. Adder A1 performs the doubling function and is followed by a subtractor primitive S1 which subtracts the products to compensate for the modified resonator coefficients. Adder A3 combines the composite excitation function to complete the resonance filter difference calculation. A third format converter primitive converts the resonance filter output, $o(t)$, to single precision before connecting to the resonance filter recursive shift registers.

The three phase clocking scheme is repeated six times for each formant channel to complete the synthesis operation. In a 16 bit synthesis system, the synthesis of a single speech sample value requires a total of 288 clock cycles. For real-time speech synthesis (at 10 kHz) a clocking rate of 2.88 MHz is required. This is well within the processing bandwidth available from modern CMOS fabrication technology which, generally, can be clocked in excess of 35 MHz. (Although this figure can vary considerably and is highly dependent upon the electrical characteristics of the circuit layout.)

Multi-channel synthesis is achieved by expanding the recursive shift registers in increments of 288 bits and increasing the clocking rate accordingly. This enables the device to be time multiplexer amongst multiple synthesis operation. In this way, the number of synthesis channels available from a single VLSI device can be increased until the maximum clocking speed of the synthesis device is

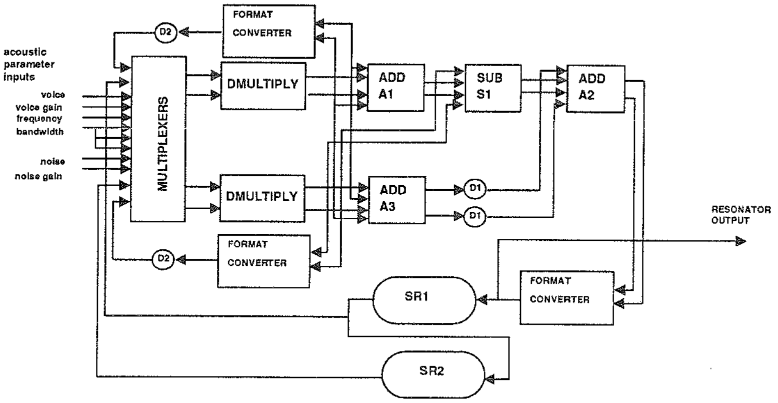

Figure 2: The two-multipler formant synthesis structure.

reached. The design goal for the present device is eight channels. This requires a clocking rate of 23.04 MHz which is well within the capabilities of 2 micron CMOS fabrication technology.

CONCLUSIONS

The two-multiplier structure presented in this paper performs the central formant filtering operations of the formant speech synthesiser. The companion paper on the implementation of the formant speech synthesiser uses the two-multiplier primitive description as the starting point. Much of the detailed description of the F1 fixed filter and output combiner have necessary been omitted. Discussion on the structure of these operators can be found in the literature (Summerfield, 1987(1), 1988).

Current research is directed toward development of strategies for incorporating the excitation function generators and coefficient mapping functions on the same chip. In particular, research is underway to develop a two multiplier implementation of the glottal excitation generator which can be incorporated into the design by employing further level of multiplexing.

There are several directions for this research to proceed. As well as the multi-channel operation discussed in this paper, the two multiplier synthesis structure is small enough to be considered as a macro-cell for inclusion in a much larger ASIC containing a microprocessor structure on which can be implemented a synthesis-by-rule or even a full text-to-speech system.

ACKNOWLEDGEMENTS

This work is funded by the Overseas Telecommunications Commission (Australia). Acknowledgements to Mr. Mike McGregor and other at the Department of Electrical Engineering, University of Edinburgh for thier assistance with the implementation of this chip. The synthesis structure described in this paper is the subject of a patent application.

REFERENCES

Denyer P. & Renshaw D. (1985) *VLSI Signal Processing: A Bit-Serial Approach*, (Addison-Wesley).

Holmes J. N. (1982) "Formant synthesizers: cascade or parallel?" JSRU Research Report 1017.

Summerfield C. D. Clark J. E. & Mannell R. H. (1986) "A high performance digital hardware synthesiser", First Australian Conference on Speech Science and Technology, pages 342 -- 347.

Summerfield C. D. (1987(1)) "VLSI structures for the implementation of a high performance formant speech synthesiser", 1st IASTED International Symposium on Signal Processing and its Applications.

Summerfield C. D. (1987(2)) "VLSI structures for the implementation of a formant synthesiser", European Conference on Speech Technology, pages 393--396.

Summerfield C. D. (1986) "A review of VLSI structures for the implementation of formant speech synthesisers", First Australian Conference on Speech Science and Technology pages 348 -- 353.

Summerfield C. D. (1988) "VLSI structures for speech synthesis", 7th Australian Microelectronics Conference, pages 9 -- 15.