A Low Power, Single Chip Realization of a Low-Speed, Low-Delay CELP Coder/Decoder for Indoor Wireless Systems

D.P. Noel*, T.A. Kwasniewski*, S. Mahmoud* and W.P. Leblanc*  
* Electronics, Carleton University; † Systems and Computer Engineering, Carleton University

Abstract - 32 kb/s Adaptive Differential Pulse Coded Modulation (ADPCM) is a widely accepted high quality speech coding algorithm. Single chip devices exist for 32 kb/s ADPCM. This algorithm requires a relatively large bandwidth and does not perform exceptionally well in noisy wireless environments. The 16 kb/s low-delay code excited linear prediction (LD-CELP) coding algorithm does perform well in noisy environments and requires a reduced bandwidth. Through algorithm optimization and simulated verification of mixed analog and digital VLSI partitioning, a single chip implementing the 16 kb/s LD-CELP voice coding algorithm will be possible. The performance of this coder chip will be comparable to the quality of 32 kb/s ADPCM. In this paper we propose the use of a link between Comsiaco's SPW and Synopsis to determine the hardware related performance degradation and the technology/complexity related power dissipation and area parameters. It will both prove the concept of design and determine the VLSI implementation feasibility.

I. INTRODUCTION

There have been many high quality low-rate speech coding schemes developed in recent years. The most prominent and widely accepted speech coding algorithm adopted internationally is Adaptive ADPCM. ADPCM has been standardized at several coding rates with 32 kb/s being the most widely used. Efforts to reduce coding rate, thereby reducing the channel bandwidth requirement but maintaining immunity to noise has lead away from the ADPCM coding algorithm.

Several institutions have investigated the merits of 32 kb/s ADPCM resulting in single chip realizations of the algorithm [1,2,3,4,5,6,7,8,9]. Typically each implementation has utilized a version of complementary metal oxide semiconductor (CMOS) technology ranging from 0.8 μm to 2 μm. Gate counts have reached 140,000 while die size approached 1 square centimeter. Due to the CMOS fabrication technology, implementation power consumption was kept to a minimum. As CMOS devices only consume power during switching mode the processing rate influences the overall power consumption of a specific chip. By lowering the coding rate and thereby the gate switching rate, the power consumption of the integrated circuit implementation will be reduced proportionately.

The quest for a lower rate speech coding algorithm led to several innovative designs utilizing Vector Quantization (VQ) [1,10,11], linear predictive coding (LPC) [10], codebook excitation [12] and code excited linear prediction (CELP) [13,14]. Several successful designs emerged with a reduced data rate (less than 32 kb/s). In most cases the performance, as reported from subjective mean opinion scoring (MOS) and other performance measures, of the encoding/decoding algorithms met or exceeded that of 32 kb/s ADPCM. Unfortunately the signal delay created within the different algorithms was too great to be successfully used in areas where high quality (MOS of greater than 4) is required. Imposing a maximum algorithm throughput delay of 2 ms eliminated all algorithms from the search. One algorithm did warrant further research. A version of CELP was found to have a delay near the 2 ms limit[15]. Further development resulted in an algorithm capable of a 16 kb/s information coding rate with a delay factor of much less than the maximum 2 ms. The specified algorithm produced speech quality equivalent to or better than that of 32 kbps ADPCM. This algorithm was adopted by the CCITT as recommendation G.728.

Starting with the CCITT G.728 recommendation, the LD-CELP algorithm was modified and optimized for translation to hardware. Several portions of the algorithm, considered to be expensive in terms of complexity and power consumption, were replaced with simpler code. Software simulation and MOS testing verified that the performance had not been degraded.

The software was then prepared for conversion to hardware. To facilitate this conversion a systematic methodology was developed with the appropriate verification techniques. The following sections of this paper will discuss the LD-CELP algorithm, modifications to it and the conversion methodology. With the completion of the methodology, using the modified LD-CELP algorithm as a test, most software coding algorithms may be easily converted to hardware.
II. LD-CELP BACKGROUND

LD-CELP was standardized by the CCITT in 1992. It is a 16 kb/s voice coding standard with a one way coding delay of less than 2 ms [15]. The principle of CELP is retained but to reduce the delay to the 2 ms requirement the LPC predictor and excitation gain are made backward adaptive and a small excitation vector size of 5 samples is used. The conventional CELP pitch predictor is sensitive to channel errors and is therefore not used in LD-CELP. The resulting performance loss is compensated for by increasing the LPC predictor order in the accepted CCITT G.728 recommendation. This predictor order was reduced in the modified algorithm to minimize the potential hardware complexity. Performance is maintained by increasing such attributes as the subframe size.

The input to the encoder, as represented by the block diagram of Figure 1, must be uniform PCM. Conversion from A-law or μ-law PCM to uniform PCM may be required. The uniform PCM is then partitioned into blocks of 5 consecutive input signal samples. For each of these input blocks the encoder passes each of the 1024 codebook vectors. From these 1024 vectors the encoder identifies one which minimizes the frequency weighted mean square error measure with respect to the input signal vector. The 10 bit codebook index of this best codebook vector is the output of the encoder. The selected codebook vector is then passed through a gain scaling unit and a synthesis filter to establish the correct filter memory in preparation for the encoding of the next signal vector. The synthesis filter coefficients and gain are updated periodically in a backward adaptive manner based on the previously quantized signal and gain scaled excitation.

The output of the encoder is received by the decoder. The decoder as illustrated in Figure 2 is much less complex than the encoder. Upon receiving each 10-bit index, the decoder performs a table look-up to extract the corresponding codevector from the excitation codebook. This codevector is passed through a gain scaling unit and synthesis filter to produce the current decoded signal vector. The synthesis filter coefficients and the gain are updated in the same manner as in the encoder. The decoded signal vector is passed through an adaptive postfilter to enhance the perceptual quality. The postfilter coefficients are updated periodically using information available at the decoder. The postfilter signal vectors are converted on output to A-law or μ-law PCM as required.

Figure 1: LD-CELP Encoder Block Diagram

Figure 2: LD-CELP Decoder Block Diagram
III. DESIGN METHODOLOGY

The CCITT standard (LD-CELP) was researched by Systems and Computer Engineering (SCE) at Carleton University. The algorithm detailed by standard G.728 was programmed in ANSI-C for experimentation. Speech recordings were converted to 64 kHz PCM and used as inputs to the encoder software. The coded speech samples output from the encoder software (speech codevectors) were passed to the LD-CELP decoder software from which speech was reconstructed.

Initially the software was generated as specified by the CCITT standard. Further research lead to the optimization of portions of the C-code keeping in mind that the end product would be a hardware device. The elimination of any mathematical division from the software was necessary. In a hardware realization the mathematical function of division is expensive in terms of complexity and chip real estate. Other modifications were made to the CCITT standard to reduce the potential complexity and related increase in power consumption. Such modifications make the resultant hardware implementation non-compatible with the CCITT LD-CELP. Extensive MOS testing performed by SCE on simulated speech coding, have indicated that the optimized C-coded algorithm will produce reconstructed speech quality equivalent to that of 32 kHz ADPCM.

The development of a procedure for rapid conversion from software to hardware was a major goal in this research. Establishing such a process would allow a trained designer to quickly and accurately estimate the power and size requirements of a resultant hardware realization. In developing a procedure one must start with the less complex software. In the case of LD-CELP the most convenient starting point was the conversion of the decoder as the encoder will also contain a version of the decoder.

The C-code performance verified, the code was syntactically modified and broken into individual modules. The software from each module was used to generate functional simulator models in a development tool from Comdisco Systems of the USA. The tool, known as Signal Processing Workstation (SPW), is a high level simulator. It uses hierarchical structure to generate the voice coding system. Once verified as functionally equivalent to the C-coded CCITT standard, the SPW tool may indirectly generate VHDL code. Figure 3 shows a flow chart of the design methodology. As can be seen from this Figure, this VHDL code may be used by other simulators and/or synthesizers.

The first task of the software conversion was the preparation of the software for integration into the SPW system. The C-code modules were originally written to access constants and memory on a global basis. Each module had to be converted to function as stand alone code. Once this was completed pictorial representations were generated in the SPW system and the respective (converted) C-code linked to the appropriate SPW representations. Simulations were performed on this SPW version of C-code using the exact input files as used with the original C-code. Output files generated by the SPW simulator were compared to the C-code output files. The optimum conversion to the SPW models was confirmed through this process. With this functional simulator verified the actual algorithm could be easily modified and verified without much additional effort. An optimum version of the coder could be easily determined through such a process.

An advantage to using the SPW simulator is that either fixed-point or floating-point software may be used. For hardware partitioning purposes it is assumed that...
digital circuitry will use fixed point mathematics and analog circuits use the equivalent of floating point math. The ability of specifying either fixed or floating point C-code is required for determining partitioning of mixed mode (analog and digital) circuitry. For a circuit to consume the least amount of power a digital version would be more desirable while if size is more critical then analog circuitry is preferred. A trade-off results in minimized area and power consumption as well as optimized performance. Varying the converted C-code in the SPW modules from fixed to floating point arithmetic will permit the optimum partitioning of the circuitry.

To facilitate realistic on-chip conditions the appropriate noise model must be researched and included in the SPW simulations. Noise will affect digital and analog circuits in different ways. It is very important to model the noise in the circuitry as accurately as possible. Improperly simulated hardware injected noise will result in poor performance of the hardware once fabricated.

Once the designer has been satisfied by the performance generated by the SPW modeling hardware, synthesis may commence. The SPW C-coded modules may be used to generate VHDL code. VHDL synthesis code may then be transported to a hardware simulator such as Synopsis (shown in the flow chart of Figure 3). A translator has been developed at Carleton's Department of Electronics (DOE) for converting the SPW C-code into VHDL code. This VHDL code may be used by the Synopsis environment. Synopsis is a high level function synthesizer. From the SPW generated VHDL listing the Synopsis synthesizer will generate a hardware representation. This will be in the form of a schematic. Further simulations may be performed in the Synopsis environment to verify the hardware functionality. This will prove the translation of the VHDL. The synthesized hardware representation is next transferred to a hardware generation tool. Two options have been explored and are explained.

For the generation of an application specific integrated circuit (ASIC) the Cadence Opus system was chosen. The CMOS4S standard cell library was slated as the fabrication technology. CMOS4S is the Northern Telecom process provided to Carleton University by the Canadian Microelectronics Corporation (CMC). To permit the automated placement and routing of the integrated circuit the CMOS4S library in Cadence was manipulated and translated to a Synopsis database. This database creation, when linked to Synopsis, permitted synthesized netlist to be directly translated to the Cadence environment. The netlist is used by the Cadence environment to perform the chip layout. This layout may be modified as the designer pleases. Die area and power consumption may be estimated at this point.

The fabrication of an ASIC for experimental purposes can be very costly and time consuming. The implementation of the synthesized hardware in a programmable device such as a field programmable gate array (FPGA) would be much faster and less costly for experimentation. Once the conversion of the algorithm to hardware via a FPGA is verified and meets the performance criterion the development of the ASIC may proceed with a high probability of success.

Motorola has created a development tool which utilizes the schematic (as synthesized by Synopsis) to perform hardware complexity and size estimations for implementation of a circuit on a FPGA. These estimations are based on the fabrication technology utilized in the generation of the FPGA. This tool, known as the open architecture conversion system (OACS), is based on the Motorola series of FPGA devices. Verification of the performance of the hardware using a Motorola FPGA does not restrict the final fabrication of the circuit to using a Motorola technology.

IV. CONCLUSIONS

To aid in the transfer of this speech coding algorithm into hardware (a chip) several simulators and synthesizers were used. A methodology for rapidly converting a speech coding software algorithm into hardware would be devised. This methodology would allow multiple versions of an algorithm to be adapted for hardware simulation before actually deciding upon the final version of hardware. Once completed any speech coding algorithm could be exercised in hardware through the use of the conversion and testing methodology. The CCITT LD-CELP standard was used as a design guideline for the development of the methodology. Based on the complexity and functionality of the LD-CELP algorithm, the proposed methodology is verified as being capable of the successful conversion of the software to hardware. Other algorithms may be converted into hardware via this automated process. By learning the process developed from this research, a designer should be able to very quickly indicate the complexity of software/hardware and where analog/digital partitions would be required.

In developing this conversion methodology, the feasibility of implementing the modified LD-CELP algorithm in hardware is determined.
V. ACKNOWLEDGMENTS

The authors would like to thank the following organizations for their continued support: the Telecommunications Research Institute of Ontario, the Canadian Microelectronics Center, Motorola and Synopsis. The programming skills of Arthur Castonguay at Carleton's Department of Electronics are greatly appreciated.

VI. REFERENCES


