A Scalable Architecture for High-Speed Digital Companding

Ortal Arazi, *Student Member IEEE*, Itamar Elhanany, *Senior Member IEEE*
Department of Electrical & Computer Engineering
The University of Tennessee
Knoxville, TN 37996-2100
Email: {ortal,itamar}@ieee.org

***Abstract*— Companding is a well-known signal processing technique exploited by a broad range of applications. It primarily offers reduction of the signal dynamic range while retaining its important attributes. Digital companders have been utilized by a variety of applications, such as voice and video coding, in which the non-linear compression/expansion is typically implemented in software. This paper proposes an efficient parallel architecture for implementing digital compander functionality at very high-speeds. A piecewise linear partitioning of the compression function is applied, driven by prescribed maximal error constraints. The scalability of the scheme in terms of speed and area is discussed. Moreover, it is shown that the architecture can be easily pipelined, yielding further speed enhancement. It is further shown that using Xilinx Virtex-II Pro (XC2VP20) FPGA devices, a 20-bit to 8-bit compander is implemented using less than 1000 gates, while operating at over 200 MHz.**

I. INTRODUCTION

Companding has been extensively incorporated as a processing technique in which a signal is compressed at the input of a system and expanded at its output. Consequently, companders have been widely deployed in communication circuits [1], as well as audio [2-4] and video [5-7] coding applications. Moreover, companding has been repeatedly proposed as a method of maintaining adequate dynamic range in integrated circuits with low power supply voltage [8]. Externally, the signal would be viewed linear while internally a non-linear transformation of the signal's dynamic range would occur.

Companding offers a variety of attractive features, such as improved noise treatment that is fundamentally different than that offered by conventional linear filters. Digital companding has been utilized for many years as means of obtaining an effect of improved non-uniform quantization in standards of digitization and compression of audio waveforms (μ-law in the U.S., and A-law in Europe) [9][10]. These techniques are based on logarithmic compressors that lead to robust quantization in the sense that the sensitivity of the signal-to-noise ratio (SNR) to the input probability density function (PDF) is dramatically reduced.

However, most digital compander implementations are carried out in software or firmware, whereby a CPU or DSP performs the required compression, quantization and expansion. With the rapid increase in data communication rates there is a growing need for a higher-speed custom compander design. This paper presents a low delay, low gate-count VLSI compander architecture. The design offers a flexible trade-off between computational accuracy and implementation resources.

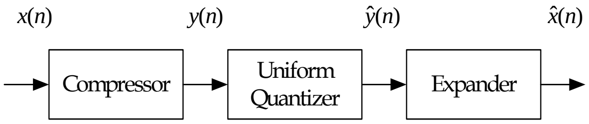

Fig. 1: A Generic compander signal flow.

The paper is structured as follows. Section II focuses on the problem statement of digital companding. Section III describes an algorithm for segmenting non-linear curves. In section IV the proposed VLSI architecture is presented while performance results are outlined in section V. In section VI the conclusions are drawn.

II. APPROXIMATE DIGITAL COMPANDER DESIGN

A generic compander signal flow is illustrated in figure 1. The original signal, $x(n)$, is non-linearly compressed yielding the signal $y(n)$ which is uniformly quantized. The quantized signal, $\hat{y}(n)$, is typically transmitted or encoded (depending on the application). At the receiver/decoder side, the quantized signal is expanded using an inverse function to that which has been used by the compressor, yielding the reconstructed signal $\hat{x}(n)$.

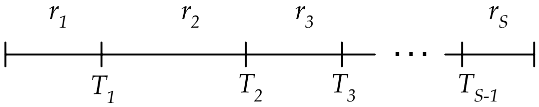

Fig. 2: Partitioning of the non-linear space into S consecutive linear segments.

Observably, the accuracy of the process depends largely on the number of bits assigned to the encoded words with respect to the resolution of the original signal. In essence, digital companding offers a non-uniform treatment of the signal's dynamic range by non-linearly mapping its encoded space.

In an aim to offer a fast and efficient compander design, we propose here an approximation model which comprises of partitioning the non-linear compression function into *S* linear segments. Each segment will be described by a linear expression in the form $2^{-k}\ x+c$. Consequently, each of the possible input values will be mapped to one of the *S* segments (regions) via the linear equation associated with each specific segment. We study the proposed scheme by utilizing the μ-law compression function given by

$$Y(X)=X_{\max}\frac{\ln\left(1+\mu X/X_{\max}\right)}{\ln\left(1+\mu\right)}\ , \qquad (1)$$

where a typical value for μ is 255. By successfully choosing the number of segments and their respective boundaries any approximation accuracy can be achieved. Notably, by selecting the number of segments to be 2^N-1, where *N* is the number of bits in each input word, we obtain perfect matching at the expense of no compression/quantization gain. A key question, addressed by the next section, is how to determine the segment parameters.

III. SEGMENTATION APPROACH

A natural criterion for partitioning the non-linear compression function into piecewise linear segments is the fitting error. In this approach, a predefined maximal error value is considered from which the number of linear segments is directly derived. Moreover, the parameters of the linear expressions describing each section are also extracted. We achieve this goal by a successive, greedy algorithm depicted in figure 3. The algorithm receives as input the quantized input range, *R*, and a maximal fitting error (ε). The algorithm progressively fits linear segments (using a mean square error criterion) whereby each segment terminates upon exceeding the maximal cumulative fitting error. All linear fits are characterized by having a slope in the form of 2^{-k}, such that the segments are in the form of $2^{-k_i}x+c_i$ *i*=1,2,…*S*, where c_i are constants and

$$k_i=\arg\min_d\left\{\left|2^{-d}-\alpha_i\right|\right\}\cdot \qquad (2)$$

α_i are the slope values obtained from a regular linear regression fitting. The error introduced by the granularity of the slopes is reflected by ε. The algorithm requires O(*W*) iterations and is carried out as a preprocessing step to the synthesis of the hardware.

IV. VLSI ARCHITECTURE

The proposed hardware compander design is depicted in figure 4. The design comprises of two primary functions: range detection and Shift-and-Add (SAA). The discrete input signal, *X*, propagates concurrently to *S*-1 comparator circuits, each comparing *X* with a different predefined value or threshold. The *S* linear segments are partitioned by these *S-1* thresholds (T_i). The output of the first comparator (T_1) will be a binary zero iff $X\in r_1$.

```
Compader_Segmentation (R, ε)
W = size of R
r = {1}
j = 1
for i  = 2 to W,
    Perform linear fit for R[j,…,i] with a slope in the
    form of 2^-k
    if (fitting error > ε)
        add (i-1) to set r
        j = i
    end
end
if (last element in set r≠W)
    add W to set r
end
```

Fig. 3: Pseudo code of a partitioning algorithm for the segmentation of non-linear compression functions into multiple linear segments.

Following a similar rationale, the outputs of each pair of neighboring comparators (*i*,*i*+1) is examined (by means of an AND gate) to determine whether $X\in r_i$. Since *X* may reside in only one of the possible *S* segments, a single binary enable signal (e_i) will indicate which segment *X* belongs to. The next phase consists of a set of parallel SAA blocks. Each block receives as input both *X* and a corresponding enabling signal. The slope parameters k_i and additive constants c_i uniquely characterize each SAA block. The slope coefficient in block *i* is in the form 2^{-k_i} where $k_i\in\{N-M+1,N\}$, while c_i are *M*-bits wide. The results of the SAA are, by definition, *M*-bit wide words. However, since only one of the enabling signals is set, by applying a bitwise OR to all results produced by the SAA blocks, we obtained the correct reconstructed value, *Y*. The segmentation procedure should be carried out as a preprocessing step, prior to synthesizing the design.

Given that the comparators of the first phase can be implemented using $\log_2 N$ gate levels (due to comparisons with constants), the critical path propagation delay is approximately

$$\tau\approx\left(\log_2 N+1\right)t_{gate}+t_{ADD-M} \qquad (3)$$

where t_{ADD-M} denotes the equivalent delay of an *M*-bit adder.

An important characteristic of the architecture is that the propagation delay does not depend on the number of segments deployed, but rather on the number of bits in the input and output words. The number of segments does, however, dictate the gate count; implementing the comparator function requires O(*N*) gates (adder complexity), while the SAA blocks require O(*M*) gates. Since a power-of-two shift

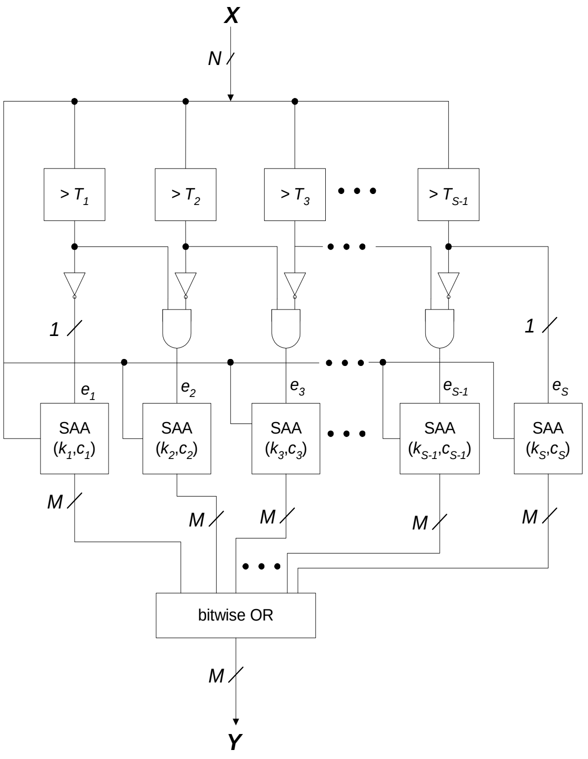

Fig. 4: The proposed compander architecture. Concurrency in all stages of the design results in the independence of the propagation delay and the number of segments.

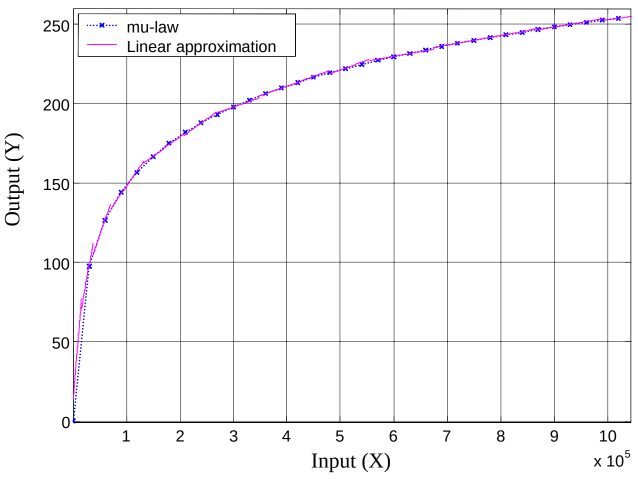

Fig. 5: Piecewise linear µ-law approximation with slopes in the form of 2^{-k}.

function consumes no logic, the aggregate gate count is thus O(S(N+M)).

V. PERFORMANCE EVALUATION

In order to evaluate the performance of the scheme, we have chosen, as first step, to compare µ-law compression results with the piecewise linear approximation proposed in this paper. The input words were 20-bit wide and the output words were 8-bit wide with μ=255. The maximal allowed cumulative error for each segment was set to 10,000.

Figure 5 shows the µ-law function with respect to the approximation scheme, as described above. The segmentation algorithm has converged to 14 segments. As expected, the segment density is higher for the lower value range where the gradient is higher. The linear segments appear to follow the non-linear mapping quite accurately indicating that, despite the small number of segments used, the signal reconstruction will be satisfactory. Considering the settings in this example, the expected number of gates would be in the order of 400, which is considerably moderate. Such a module can be repeatedly incorporated in a large design.

Next, VHDL code corresponding to the proposed design was written for a 20-bit to 8-bit compander. The target device used was a Xilinx VIRTEX-II Pro (XC2VP20) FPGA [11]. The post place-and-route results indicated that the total equivalent gate count for the design was 948 gates, while the critical path (register-to-register) was 4.68 nsec, yielding a design that operates at more than 200 MHz.

VI. CONCLUSIONS

This paper presents a scalable, pragmatic digital compander design. Based on partitioning the non-linear compression function into multiple linear segments, flexible trade-offs between accuracy and speed are achieved. The architecture realization requires moderate logic resources and can operate at very high speeds.

References

[1] S. Mahapatra, A. Ionescu, K. Banerjee, M. Declercq, “A SET Quantizer Circuit aiming at Digital Communication System”, *IEEE International Symposium on Circuits and Systems (ISCAS) 2002*.

[2] F. O. Witte, R. Backes, M. Klumpp, E. Schidlack, M. Ullrich, “Multipurpose Audio Signal Compander,” *IEEE Trans. on Consumer Electronics*, Vol. 39, No. 3, pp. 260-268, 1993.

[3] “Mu-Law and A-Law Companding Using the TMS320C2xx DSP”, *Application Report SPRA349*, Texas Instruments, Oct. 1997.

[4] A. Aggarwal, L. Shankar L. K. Rose, “Compander-Domain Approach to Scalable AAC,” *110th Convention of the Audio Engineering Society*, Netherlands, May 2001.

[5] N. Merhav, “Embedding Companders in JPEG Compression,” *Technical Report HPL-98-141*, 1998.

[6] M. Bakshi and D. R. Fuhrmann, “Improving the Visual Quality of JPEG-Encoded Images via Companding," *Journal of Electronics Imaging*, vol. 6, no. 2, pp. 189{197, April 1997.

[7] P. Chaturvedi, M. F. Insana, T. J. Hall, "2-D local companding for noise reduction in strain imaging," *IEEE Trans. Ultrason. Ferroelec. Freq. Control*, vol. 45, pp. 179-191, 1998.

[8] L. Toth, Y. Tsividis, and N. Krishnapura, "On the Analysis of Noise and Interference in Instantaneously Companding Signal Processors", *IEEE Transactions on Circuits and systems-II*, vol. 45, no. 9, pp. 1242-1249, Sep. 1998.

[9] ITU-T Recommendation G.711 (11/88) http://www.itu.int/itudoc/itut/rec/g/g700-799/g711.html.

[10] ITU-T [G.728] Recommendation G.728 (09/92) - http://www.itu.int/itudoc/itu-t/rec/g/g700-799/G.728.html.

[11] *Virtex II Pro FPGA Family Data Sheets*, available at: www.xilinx.com.