A HIGH-SPEED ARCHITECTURE FOR ADPCM CODEC[1]

Naresh R. Shanbhag   Keshab K. Parhi

Dept. of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455

Abstract

A pipelined architecture for adaptive pulse code modulation (ADPCM) is presented. The architecture is developed by the application of *relaxed form of look-ahead*. The hardware overhead is only the the pipelining latches and is independent of the number of quantizer levels, the predictor order and the pipelining level. The codec latency is smaller than the level of pipelining. Under the assumption of small quantization error, the convergence properties of the pipelined architecture are compared with that of the serial one. Speech and image coding examples are presented to support the conclusions in this paper.

1 Introduction

Many real-time applications require high-speed digital signal processing systems. The techniques of pipelining [1] and parallel processing [2] are two major ways to achieve the desired speed-up. While the look-ahead technique [1] has been successfully applied to pipeline adaptive filters [3, 4], the accompanying hardware increase is extremely prohibitive. This is because the look-ahead technique, developed primarily for fixed coefficient filters, maintains exact input-output mapping. For pipelining adaptive filters, we may sacrifice this exact input-output mapping if the average convergence behaviour is not changed substantially. Thus, we may use a *relaxed form of look-ahead* for pipelining adaptive filters.

The relaxed look-ahead has been succesfully employed to develop a pipelined LMS adaptive filter architecture [5], referred to as the PIPLMS (to be read as Pipe LMS) filter. It has also be applied to develop pipelined architectures for adaptive lattice filters [6]. In this paper, we apply relaxed look-ahead to pipeline the adaptive pulse code modulation (ADPCM) coder and decoder. The ADPCM has been recently pipelined with the look-ahead [7], but the hardware increase is strongly dependent on the number of quantizer levels L, the order of predictor N and the pipelining level M. The pipelined ADPCM architecture presented in this paper, has a hardware increase which is independent of L and N and is weakly dependent on M.

In section 2, we derive the pipelined ADPCM (PIPADPCM) architecture and discuss its convergence properties. Comparison of the convergence properties and the performance of PIPADPCM with the serial architecture (SADPCM) is carried out in section 3. Speech and image coding examples are presented in section 4, to support the conclusions of the paper.

2 The Pipelined ADPCM Architecture

Consider the conventional serial ADPCM architecture (or SADPCM) in Fig.1, which employs the least-mean-square (LMS) algorithm for adapting the coefficients of the predictor block **F**. The quantizer **Q** and the weight-update block **WUD** are also shown. The computation time (and therefore the clock period) of the SADPCM is given by

$$T_c = 3T_m + (N+1)T_a + T_q, \tag{2.1}$$

where T_m and T_a are the compute times of a two-operand multiplier and adder, respectively, and T_q is the quantizer computation time. It is desired that the clock period of the pipelined ADPCM architecture (called as PIPADPCM) is less than T_c/M, where M is the desired speed-up.

The input to **F** (see Fig.1), which is the reconstructed signal $\tilde{s}(n)$, and is given by :

$$\tilde{s}(n) = s(n) - q(n), \tag{2.2}$$

where $q(n)$ is the *quantization error* due to quantization of the *prediction error* signal $e(n)$.

Under the assumption of small quantization error (i.e. $\tilde{s}(n) \approx s(n)$) the SADPCM architecture of Fig.1 reduces to that of the conventional linear predictor which has already been pipelined [5]. From the pipelined linear predictor, a pipelined ADPCM architecture (or PIPADPCM) can be developed by feeding the quantized value of the error signal ($e_q(n)$) and $\tilde{s}(n)$ to the **WUD** block. The PIPADPCM architecture is shown in Fig.2, where it is clear that the increase in pipeline latches constitute the sole increase in hardware. This is in remarkable contrast to the hardware explosion which results on the application of look-ahead [7]. Also, the hardware increase is independent of the number of quantizer levels, the filter order and the level of pipelining. The codec latency in PIPADPCM equals $L_c + L_d + 1$, which is usually smaller than the speed-up M. The M_1 latches in Fig.2 (in block **F**) can be used

[1] This research was supported by the army research office by contract number DAAL03-90-G-0063.

0-7803-0593-0/92 $3.00 1992 IEEE

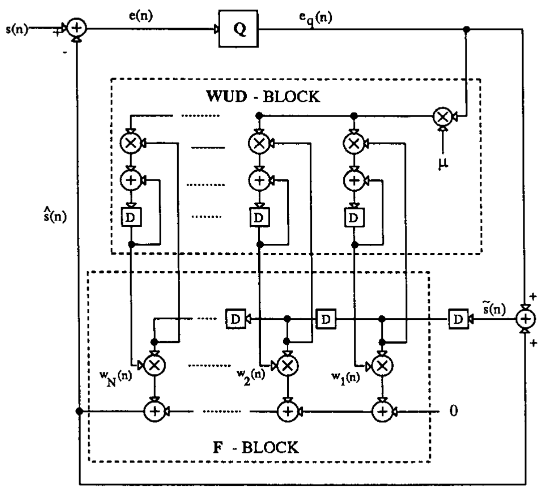

Figure 1: The SADPCM architecture

to pipeline the **F** block, the quantizer **Q** and the adders. This fact will be illustrated via an example in section 4.

3 Convergence Properties and Performance of PIPADPCM

Under the assumption of small $q(n)$ the PIPADPCM reduces to PIPLMS [5]. In this section we briefly describe the convergence properties of PIPLMS.

3.1 Convergence analysis

A sufficient condition on the adaptation constant μ, for the convergence of weights, is given by

$$0 \leq \mu \leq \frac{1}{4\lambda_{max}}, \qquad (3.1)$$

where λ_{max} is the largest eigenvalue of the input correlation matrix $\mathbf{R}$. This upper bound is eight times tighter than that of SADPCM.

The convergence time-constant for PIPADPCM (τ_{PIP}) would be $M_1 + M_{21}$ times that of SADPCM (τ_S). However, the convergence time in seconds for PIPADPCM (t_{PIP}), is in fact less than that of SADPCM (t_S). This is because the clock period of PIPADPCM is a fraction of that of SADPCM. With the assumption of (3.1), this fraction equals 1/2.

From Fig.2, we notice that PIPADPCM does a $M_1 + 1$-step forward prediction. Thus, PIPADPCM would always give a higher steady state prediction error than SADPCM. The PIPADPCM misadjustment $\mathcal{M}_{PIP}$, defined with respect to the minimum mean-squared error of SADPCM ($J_{min,S}$), is given by

$$\mathcal{M}_{PIP} = (\mathcal{M}_S + 1)\alpha - 1 \qquad (3.2)$$

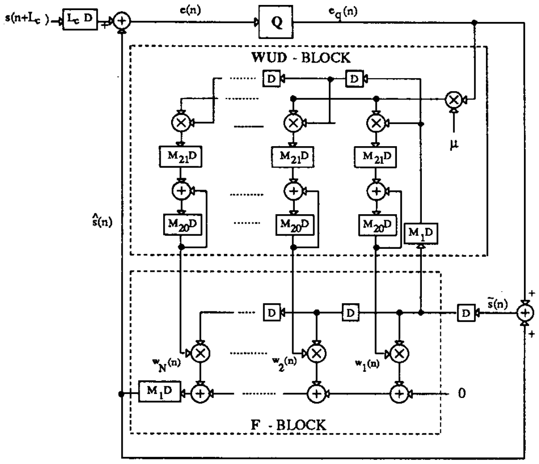

Figure 2: The PIPADPCM architecture.

where $\mathcal{M}_S$ is the misadjustment of SADPCM and the constant $\alpha = J_{min,PIP}/J_{min,S}$ ($J_{min,PIP}$ is the minimum mean-squared error of PIPADPCM).

3.2 Performance

The performance of ADPCM codec can be quantified in terms of the ratio of the input signal power σ_s^2 to the reconstruction noise power σ_r^2 (SNR). Assuming fixed uniform quantizers, the quantization error power σ_q^2 is related to the prediction error power σ_e^2 as

$$\sigma_q^2 = \epsilon_* 2^{-2R} \sigma_e^2, \qquad (3.3)$$

where ϵ_* is a constant and R is the number of quantizer bits.

From (3.3) and from the discussion in the previous subsection, it is clear that PIPADPCM would have a higher $r(n)$ as compared to SADPCM. Employing (3.2) and (3.3), a relation between the number of quantizer bits for PIPADPCM (R_P) and that for SADPCM (R_S) for the same σ_r^2, can be derived easily and is given by

$$R_P = R_S + \frac{1}{2} log_2 \alpha. \qquad (3.4)$$

Thus, PIPADPCM has a lower SNR than SADPCM for the same R. However, it will be shown that for speech and image applications there is no perceptual degradation even if the number of quantizer bits is not increased. For speech, we use the *Articulation Index* (AI) [8] defined as

$$AI = 0.05 \sum_{i=1}^{20} [min(SNR_i(dB), 30)/30] \qquad (3.5)$$

where SNR_i is the signal-to-noise ratios calulated in 20 predefined [8] frequency bands.

4 Examples

In this section, we first present a pipelining example to demonstrate the increase in speed and then present a speech coding application.

4.1 Pipelining Example

To illustrate the actual reduction in the clock-period, we assume $T_a = T_q = 20$, $T_m = 40$ and $N = 2$. Therefore, from (2.1) it follows that $T_c = 200$. For a speed-up of 5, the clock-period of PIPADPCM should be 40. This can be achieved with $M_1 = 2$, $M_{20} = 1$, $M_{21} = 2$, $L_c = 1$ and $L_d = 0$. The distribution of latches to obtain the desired clock-period is shown in Fig.3 for the coder. The decoder can similarly be obtained.

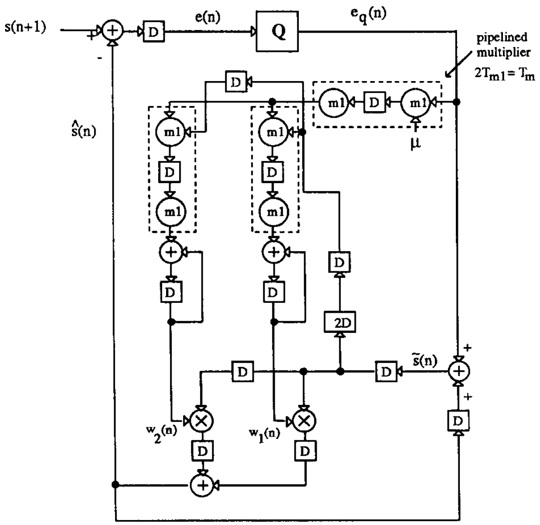

Figure 3: The encoder example.

4.2 Speech Coding Example

In this section, we take a speech signal corresponding to the word 'zoos', sampled at 8 kHz for a duration of 1 s. This signal (see Fig.4) was encoded with SADPCM and PIPADPCM with a 12th order predictor and a 4-bit fixed uniform quantizer.

For PIPADPCM, we chose $M_1 = 4$, $M_{21} = 1$, $M_{20} = 1$, $L_c = 0$, $L_d = 0$, which corresponds to a speed-up of $M = 5$. The μ for SADPCM was 0.05 while that of PIPADPCM was 0.009. These values of μ were chosen for optimal tracking performance. The quantizer dynamic range for SADPCM was 0.32 and that of PIPADPCM was 0.8. This was done to account for the differing prediction error powers. As expected, the prediction error power (Fig.5(a)) for SADPCM is much lower than that of PIPADPCM (Fig.5(b)). The σ_r^2 for SADPCM was upper bounded by 4.5×10^{-4}, while that of PIPADPCM was bounded by 0.003. Thus, the signal-to-noise ratio (SNR) for PIPADPCM (15.62 dB) is inferior to that of SADPCM (23.33 dB). However, the AI was 90% for both the architectures, which is the maximum achievable under the given sampling rate. This confirms our claim that in speech applications the PIPADPCM can have a performance identical to that of SAPDCM inspite of a degraded SNR. Note that PIPADPCM has the same latency as SADPCM in spite of the speed-up of 5. Next, we present an image coding example, where the image of Lenna (Fig.6) is encoded via PIPADPCM and SADPCM. A uniform quantizer with $L = 16$ was chosen. The predictor order was 3. The reconstructed image for SADPCM (Fig.7) and PIPADPCM (for speed-up of 20) (Fig.8) are perceptually identical. Similar results have been obtained for higher speed-ups and coarser quantization.

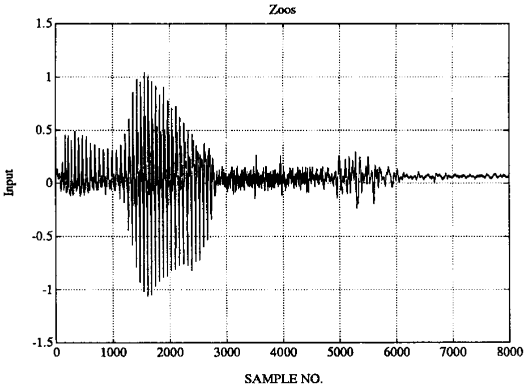

Figure 4: The input speech signal.

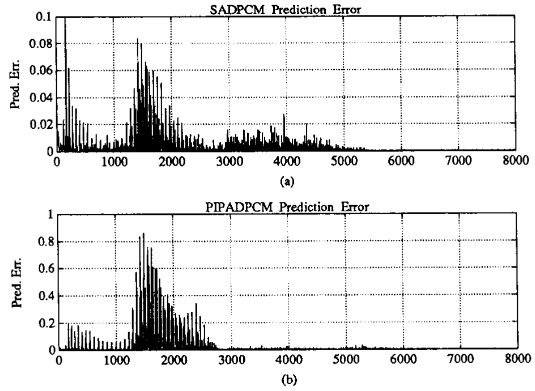

Figure 5: Prediction error powers : (a) SADPCM and (b) PIPADPCM

5 Summary and Conclusions

A pipelined ADPCM architecture (PIPADPCM) has been developed by employing a relaxed form of look-ahead, resulting in an enormous savings in hardware. Applications to speech and image coding are presented.

Improvements in the PIPADPCM performance are possible through the use of adaptive quantizers (specially for speech signals).

References

[1] K.K. Parhi and D.G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters - Part I : Pipelining using scattered look-ahead and decomposition", *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol. 37, pp. 1099-1117, July 1989.

[2] K.K. Parhi and D.G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters - Part II : Pipelined incremental block filtering", *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol. 37, pp. 1118-1134, July 1989.

[3] K.K. Parhi and D.G. Messerschmitt, "Concurrent cellular VLSI adaptive filter architectures", *IEEE Trans. on Circuits and Systems*, vol. 34, pp. 1141-1151, Oct. 1987.

[4] T. H.-Y. Meng and D.G. Messerschmitt, "Arbitrarily high sampling rate adaptive filters", *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol. 35, pp. 455-470, April 1987.

[5] N.R. Shanbhag and K.K. Parhi,"A pipelined LMS adaptive filter architecture", *Proc. Asilomar Conf. on Sig., Syst. and Comput.*, 1991.

[6] N.R. Shanbhag and K.K. Parhi, "A pipelined adaptive lattice filter architecture", *Proc. IEEE Intl. Symp. on Circuits and Systems*, San Diego, 1992.

[7] K.K. Parhi, "Pipelining in algorithms with quantizer loops", *IEEE Trans. on Signal Processing*, vol. 38, pp. 745-754, July 1991.

[8] N.S. Jayant and P. Noll, *Digital Coding of Waveforms.* Englewood Cliffs, NJ: Prentice Hall, 1984.

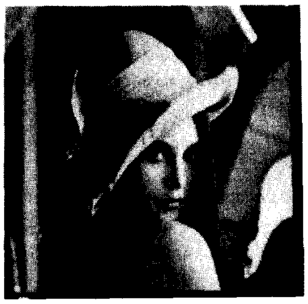

Figure 6: The original image.

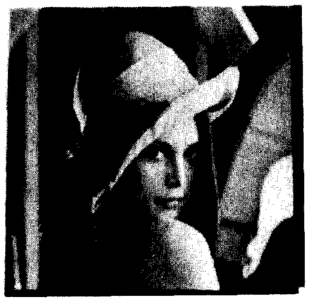

Figure 7: SADPCM reconstructed image.

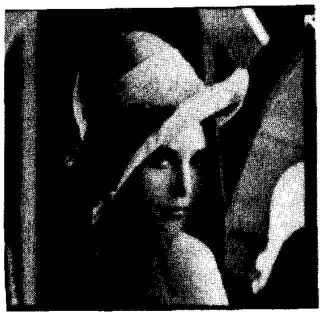

Figure 8: PIPADPCM reconstructed image.