VLSI IMPLEMENTATION OF A LOW-ENERGY SOFT DIGITAL FILTER [1]

Rajamohana Hegde and **Naresh R. Shanbhag**
Coordinated Science Laboratory/ ECE Department
University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL 61801
email: [rhegde, shanbhag]@icims.csl.uiuc.edu

Abstract - In this paper, we present a VLSI implementation of an energy-efficient digital filtering algorithm developed using the *soft DSP* framework. Soft DSP refers to *overscaling* the supply voltage without sacrificing speed and employing algorithmic error-control to restore the resulting performance degradation. It is shown that delay imbalance at the circuit level inherent in existing arithmetic structures result in improved energy savings with marginal degradation in performance. The proposed scheme implemented in $0.35\mu m$ $TSMC$ technology provides an overall energy savings of up to 76% with performance degradation of less than $1dB$ in the signal-to-noise ratio (SNR_o) at the filter output.

1 INTRODUCTION

For a given technology, reduction in energy dissipation has been made possible due to energy-efficient design techniques at all possible levels of design hierarchy. Schemes at the lower levels of the design process [1] are usually application independent. At the algorithmic and architectural levels, features that are specific to a class of applications are exploited to develop application specific energy reduction techniques [2, 3]. Voltage scaling [3] is an effective means of achieving reduction in energy dissipation as a reduction in supply voltage V_{dd} by a factor K, reduces the dominant capacitive component of energy dissipation by a factor K^2 [3]. However, the extent of voltage scaling [3] is limited by the critical path delay of the architecture and the throughput requirements of the application. If the supply is scaled beyond this limit (denoted as $V_{dd-crit}$), input dependent intermittent errors appear at the output. Hence $V_{dd-crit}$ is seen as the lower bound on voltage scaling for a given architecture.

Recently, we have shown [4, 5] that V_{dd} can be *overscaled* beyond $V_{dd-crit}$ for multimedia communication systems in which statistical metrics such as bit-error rate (BER) and signal-to-noise ratio (SNR) are used to specify the algorithmic performance. It was shown that the errors introduced due to sub-critical voltage operation lead to degradation in algorithmic performance. The notion of algorithmic noise-tolerance (ANT) was proposed to restore the

[1]This work was supported by NSF award CCR-0000987.

performance degradation due to sub-$V_{dd-crit}$ operation. The use of ANT in this context leads to substantial energy savings at no loss of throughput and was referred to as *soft DSP*. Several error-control algorithms that exploit the correlation in the output samples have been proposed for *frequency selective filters* to restore performance degradation due to soft errors. It was shown that up to 80% energy savings are possible for narrowband filters. It was also shown that the proposed scheme is effective in restoring performance degradation due to random errors induced by deep submicron (DSM) noise [7].

In this paper, we present a VLSI implementation of the frequency-selective filtering algorithm developed in [4]. The proposed implementation involves a single multiply-accumulate (MAC) structure with programmable coefficients and a single MAC predictor which is employed to perform error-control via the backward prediction algorithm presented in [4]. It is shown that, the delay imbalance at the circuit level improves the effectiveness of the proposed implementation. It is also shown that existing designs exhibit substantial delay imbalance and hence result in considerable energy savings under the soft DSP framework.

The rest of this paper is organized as follows: In next section, the notions of algorithmic noise-tolerance, soft DSP, and prediction-based error-control [4] are reviewed. The delay imbalance of the structures employed in the proposed implementation at the circuit level are studied in section 3. The design details and a new MAC structure are then presented in section 4. We then report the algorithmic performance vs. energy savings characteristics of the proposed implementation in section 5 and finally, in section 6 conclusions and scope for future work on this topic are presented.

2 ALGORITHMIC NOISE-TOLERANCE (ANT)

The notion of algorithmic noise-tolerance, proposed in [4], refers to algorithmic error-control to restore performance degradation due to errors in the system output. As the algorithmic performance of DSP/communication systems is specified in terms of statistical metrics such as (BER) and (SNR), errors in system output lead to degradation in BER or SNR. If the error frequency is high enough, it can lead to substantial degradation in BER or SNR making them fall below the minimum desired level. In [4], two scenarios that can lead to degradation in BER/SNR in a DSP system are considered. For both, we employ ANT to restore the performance degradation. In the first scenario, referred to as *soft DSP*, errors are deliberately introduced by scaling supply voltage without sacrificing throughput in order to achieve energy savings. In the second scenario, errors are due to spurious voltage deviations introduced at the circuit level due to DSM noise.

2.1 Soft DSP

A typical DSP system is designed in such a way that the critical path delay [6] T_{cp} (defined as the *worst case delay over all possible input patterns*), should

be less than or equal to the sample period T_s, i.e., $T_{cp} \leq T_s$. Hence, given T_s, the DSP system is designed such that, at the rated supply voltage V_{dd}, the *delay condition*, $T_{cp} \leq T_s$, is satisfied. The relationship between V_{dd} and circuit delay τ_d is given by [6],

$$\tau_d = \frac{C_L V_{dd}}{\beta(V_{dd} - V_t)^\alpha}, \tag{1}$$

where C_L is the load capacitance, α is the velocity saturation index, β is the gate transconductance, and V_t is the device threshold voltage. We refer to the voltage at which $T_{cp} = T_s$ as the *critical supply voltage* $V_{dd-crit}$ of a given architecture. Note that violating the delay condition by reducing V_{dd} beyond $V_{dd-crit}$ [4], i.e., setting

$$V_{dd} = K_v V_{dd-crit}, \tag{2}$$

where $0 < K_v < 1$, leads to erroneous output when the critical paths are excited. In conventional voltage scaling, $V_{dd-crit}$ is seen as a lower bound on the supply voltage for a given architecture and throughput. In [4], it was proposed to operate the DSP architecture at voltages lower than $V_{dd-crit}$ in order to achieve higher energy savings. Such operation leads to errors in the system output when the critical paths and other longer paths are excited, which are corrected using ANT techniques. This approach typically gives up to 80% additional energy savings after current day voltage scaling approaches have been applied.

2.2 Impact of errors on algorithmic performance

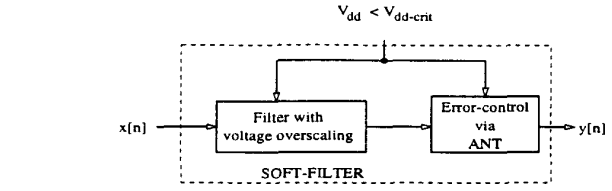
In the absence of errors, the performance of a DSP transfer function $H(z)$ is measured in terms of the output SNR given by $SNR = 10\log(\sigma_s^2/\sigma_n^2)$, where σ_s^2 is the signal power and σ_n^2 is the signal noise power. In presence of soft errors, we have, $SNR = 10\log(\sigma_s/(\sigma_{n+c}^2))$, where σ_{n+c}^2 is the total noise power. In this scenario, ANT is employed to achieve a value of σ_{n+c}^2 that is as close to σ_n^2 as possible. The block diagram of a general soft DSP scheme is shown in Figure 1(a).

2.3 Noise-tolerant filtering algorithm based on linear prediction

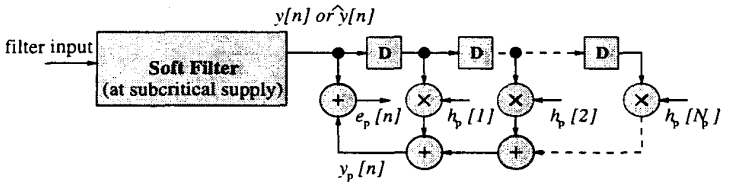
The error-control algorithms presented in [4, 5] are based on backward and forward-backward linear prediction [8]. In this section we present a brief review of these algorithms.

The output of a digital FIR filter shown in Fig. (1(b)), when the filter is error-free, is denoted by $y[n]$ and is given by,

$$y[n] = \sum_{k=0}^{N-1} h[k]x[n-k], \tag{3}$$



(a)



(b)

Figure 1: The soft DSP framework: (a) voltage overscaling, and (b) prediction-based algorithmic noise-tolerance scheme.

where $h[k]$ denotes the filter impulse response, $x[n]$ is the filter input and N is the number of taps in the filter. Let $\hat{y}[n]$ denote the filter output when the filter is operating under reduced voltage, with $\hat{y}[n] = y[n] + y_{err}[n]$, where $y_{err}[n]$ denotes the error in the filter output due to soft errors. Note that $y_{err}[n]$ is non-zero only when the input pattern is such that longer paths in the filter implementation are excited. Let $y_p[n]$ denote the output of an N_p-tap predictor when the filter is noiseless, i.e.,

$$y_p(n) = \sum_{k=0}^{N_p} h_p[k]y[n-k], \tag{4}$$

where $h_p[k]$ denotes the *optimum predictor coefficients* [8] that minimize the mean squared value (MSE) $< e_p^2[n] >$ of the prediction error $e_p[n]$, given by, $e_p[n] = y[n] - y_p[n]$. The minimum mean square error (MMSE) depends on the autocorrelation function of $y[n]$ and the order of the predictor. Let $\hat{y}[n]$, $\hat{y_p}[n]$, and $\hat{e_p}[n]$ denote the filter output, the predictor output, and the prediction error, respectively, in presence of soft errors. It can be shown that $\hat{e_p}[n] = y_{err}[n] + e_p[n]$. Assuming that no more errors occur in the next N_p output samples, we can show that, $\hat{e_p}[n+m] = -h_p[m]y_{err}[n] + e_p[n+m]$, for $m = 1, \cdots, N_p$.

The error-control algorithms presented in [4, 5] are based on the fact that, in absence of errors due to soft computations, the magnitude of the prediction error is small and is inversely proportional to the correlation in the digital

filter output. In presence of a soft error, the magnitude of the prediction error increases substantially and the erroneous sample leads to errors in the subsequent output samples of the predictor. Hence a pattern of errors is generated. This pattern is exploited to detect errors in the filter output. In case an error is detected, the predictor output corresponding to the erroneous sample is resolved to be the actual filter output.

The effectiveness of this scheme depends on the following factors:

- **output correlation:** The correlation in the filter output determines the accuracy of the predictor. Hence, for highly correlated outputs such as narrowband filters, the prediction error is extremely small in absence of soft errors. Hence the probability of detecting a soft error increases.

- **error frequency:** The frequency of soft errors depends on the factor by which the supply voltage has been scaled and, more importantly, on the delay imbalance present in the architecture employed. As it is shown in the next section, delay imbalance at both circuit and the architectural level present in several existing arithmetic units improve the effectiveness of the soft DSP approach.

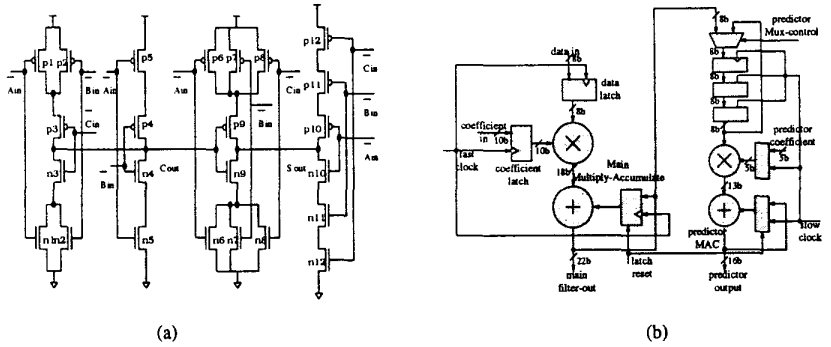
3 CIRCUIT LEVEL DELAY IMBALANCE

It was shown in [4] that the energy savings attainable via soft DSP depends on the path delay distribution of the DSP block. Specifically, we showed that in case of a ripple carry adder, due to its wide delay distribution(or large delay imbalance), up to 75% energy savings are possible if an error probability of 0.1 can be tolerated. It was also shown that the delay balance of sign-magnitude computation is larger compared to that of two's complement. Hence a new MAC structure that employs a sign-magnitude multiplier was proposed. In this section, we discuss the delay imbalance present in existing arithmetic blocks at circuit level.

The soft-DSP implementation presented in this paper employs the fully symmetric, static CMOS 24-transistor full-adder [6] shown in Fig. 2(a). Note that, in this structure, the carry output gets computed before the sum output is computed. In the full adder structure shown below, different input combinations lead to different evaluation delays leading to delay imbalance.

- The input combinations $(\bar{A_{in}} = 0, \bar{B_{in}} = 1, \bar{C_{in}} = 0)$, $(\bar{A_{in}} = 1, \bar{B_{in}} = 0, \bar{C_{in}} = 0)$, and $(\bar{A_{in}} = 0, \bar{B_{in}} = 0, \bar{C_{in}} = 1)$ can lead to a low-to-high transition at the output C_{out} in which two PMOS transistors act in series. Similarly, the combinations $(\bar{A_{in}} = 1, \bar{B_{in}} = 0, \bar{C_{in}} = 1)$, $(\bar{A_{in}} = 0, \bar{B_{in}} = 1, \bar{C_{in}} = 1)$, and $(\bar{A_{in}} = 1, \bar{B_{in}} = 1, \bar{C_{in}} = 0)$ can lead to a high-to-low transition at C_{out} with two NMOS transistors acting in series. Note that this forms the worst case delay for the node C_{out}. For a full adder designed in $0.35\mu m$ TSMC CMOS technology, this delay turns out to be $0.33ns$.

- However, when we have, $(\bar{A_{in}} = 0, \bar{B_{in}} = 0, \bar{C_{in}} = 0)$ and $(\bar{A_{in}} = 1, \bar{B_{in}} = 1, \bar{C_{in}} = 1)$, the transition at C_{out} will be much faster as there will be two paths to V_{dd} or V_{ss}. For the full adder designed for this implementation, this delay is $0.18ns$. Hence there are two possible delay values for C_{out}.
- Similarly, it can be seen that there are two possible delay values for the S_{out} node which is also dependent on the delay at C_{out}. These delay values for the adder designed TSMC technology are $0.39ns$ and $0.14ns$ respectively.



(a) (b)

Figure 2: The soft DSP implementation: (a) schematic diagram of the 24 transistor full-adder, and (b) the filter architecture.

Several DSP algorithms comprise of the multiplication operation and the multipliers are constructed using a binary full adder as their building block. The critical path delay of the multiplier structure is usually computed in terms of full-adder delays [6]. But as we have shown here, the full adder is not a delay-balanced structure and hence the worst case full-adder delay (which in this case turns out to be $0.33ns + 0.39ns = 0.72ns$) is considered. However, when one of the critical-paths in a multiplier is excited, the probability that each one of the full-adders in that path receive a worst case input combination is quite low and hence, the actual critical path delay could be much lesser than the worst case one. This provides an opportunity for sub-critical voltage operation with very low error probabilities (due to critical path violations).

4 VLSI DESIGN

The block diagram of the proposed soft DSP implementation is shown in Fig. 2(b). The main MAC is operated at sub-critical supply voltage. The main MAC output is fed to the predictor MAC which employs the past output samples to compute the current predictor output. If the prediction error is large, then the filter output is assumed to be in error and the predictor output is resolved to be the actual filter output.

4.1 Latches

The structure is implemented for most part in static CMOS. The precisions employed are marked on the block diagram in Fig. 2(b). Note that the precisions of the predictor are much lesser than that of the main filter. We have employed C^2MOS implementation [6] for all the latches in this implementation. The latches are powered with a separate supply that is held constant. The main MAC consists of an unsigned array multiplier followed by a modified 2's complement adder.

4.2 Main MAC

As the signed-magnitude representation is more attractive from the soft-DSP point of view (as shown in [4]), a modified version of the MAC structure proposed in [4] is employed in this design. The MAC structure proposed in [4] requires two additional adders to add a bias term to take $-ve$ partial products into account. A further modification of this structure that eliminates the additional adders is employed in the proposed implementation. The structure of this new MAC is derived as follows:

Let the sign-magnitude multiplier output (K bits in length) be denoted by

$$S(n) = s_{K-1}(n)\, s_{K-2}(n) \cdots s_0(n) \tag{5}$$

where $s_i(n)$ denote the i^{th} bit. Let the adder output (of the MAC) be $K+M$ bits wide. If the product (output of the multiplier) is $+ve$, then the adder input is set to

$$S_a(n) = \underbrace{0\,0\cdots0}_{Mbits}|\, s_{K-1}(n)\, s_{K-2}(n) \cdots s_0(n) \underbrace{\leftarrow 0}_{carry-in} . \tag{6}$$

If the product is negative, the adder input needs to be $-s(n)$. Instead of a full two's complement of the multiplier output, we apply partial two's complement of the multiplier output to the adder which is obtained as

$$S_a(n) = \underbrace{0\,0\cdots0}_{Mbits}|\, \bar{s}_{K-1}(n)\, \bar{s}_{K-2}(n) \cdots \bar{s}_0(n) \underbrace{\leftarrow 1}_{carry-in} . \tag{7}$$

where $\bar{s}_i(n)$ is the complement of $s_i(n)$. It can be shown that the partial two's complement is equivalent to

$$S_a(n) = -2^{K+1} - S(n). \tag{8}$$

Hence, the bias term -2^{K+1} needs to be subtracted from the final accumulated output. Let

$$B = b_{M-1}2^{M-1} + \cdots + b_1 2^1 + b_0 \tag{9}$$

be the total number of $-ve$ partial products for a single filter output. The bias term that needs to be subtracted from the final output is $B[2^{K+1}]$. Note that B can be obtained by employing a counter that is incremented when the product is $-ve$. The proposed MAC structure eliminates the need for additional adders employed in the MAC structure presented in [4].

4.3 Predictor

To compute the estimate of the current output, an additional two's complement MAC is employed. The predictor MAC in the proposed implementation is clocked at a frequency that is $8X$ slower than the main MAC. Hence the supply to the predictor MAC can be reduced to minimize the additional energy dissipation. The proposed implementation allows for up to 3 taps in the predictor for a 32-tap filter. Note that this configuration covers most of the frequency selective filtering cases presented in [4]. To implement the difference based scheme the first tap of the predictor is set to unity and rest of the taps are set to zero. The transistor complexity of the main MAC is 3872 while that of the predictor is 2423. In addition, the predictor is clocked at $1/8^{th}$ the clock speed of the main MAC. Hence, in terms of energy dissipation, the contribution from the predictor is less than 10%.

5 SIMULATION RESULTS

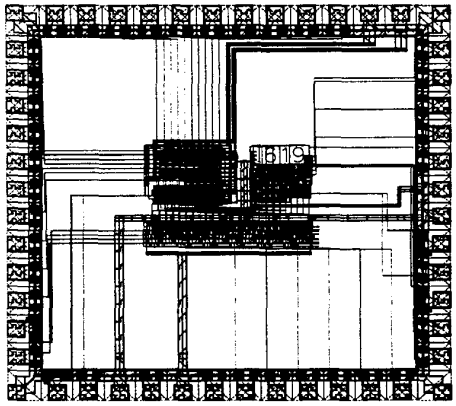


Figure 3: Layout of the soft DSP filter.

The layout of the soft DSP filter implemented in $0.35\mu m$ $TSMC$ CMOS technology is shown in Fig. 3. The main MAC is designed to be clocked at $150MHz$ at $3.3V$. The core circuitry including the multiplexer at the output comprises of aproximately $10,000$ transistors. The performance of the proposed implementation is measured in the same frequency selective filtering setup as employed in [4].

5.1 Performance Measures

The algorithmic performance of the proposed implementation is measured in terms of the SNR at the output of the filter given by [4], SNR_o =

$10log_{10}(\sigma_s^2/\sigma_n^2 + \sigma_c^2)$ where σ_s^2 is the variance of the signal component, σ_n^2 is the variance of the signal noise, and σ_c^2 is the variance due to soft errors.

The savings in energy dissipation in the proposed implementation due to voltage reduction is estimated as follows: Let the critical path length of the main MAC in terms of full-adder delays be $N_c T_{FA}$, where N_c is the number of full-adders in the critical path of the conventional two's complement array multiplier with the same precision as the multiplier employed in this implementation and T_{FA} is the delay of a full adder. Using HSPICE, we first compute the value of $V_{dd-crit}$, the supply voltage at which we get $T_p = N_c T_{FA}$. We then compute the supply value V_{dd-k} at which we get $T_p = (N_c - k)T_{FA}$. Then the reduction in energy dissipation (energy savings ES)at each supply value V_{dd-k} is given by

$$ES = \frac{E_{original} - E_k}{E_{original}} \times 100\%, \tag{10}$$

where $E_{original} = tC_L V_{dd-crit}^2$ and $E_k = tC_L V_{dd-k}^2$, where t is the average transition activity and C_L is the total load capacitance of the main MAC. Hence we get

$$ES = 1 - \frac{V_{dd-k}^2}{V_{dd-crit}^2}. \tag{11}$$

At each value of V_{dd-k}, we obtain an estimate of the algorithmic performance by setting $T_p = (N_c - k)T_{FA}$ in the high-level algorithmic model of the soft-DSP system that is similar to the one employed in [4].

5.2 SNR vs. Energy Dissipation

The plot of energy-savings vs. algorithmic performance for the above implementation for a filter of bandwidth $\omega_b = 0.2\pi$ is shown in Fig. 4. Note that the energy vs. algorithmic performance characteristics are very similar to the ones obtained in [4], though the energy values in [4] were obtained via a high level estimation tool. In estimating the energy savings as mentioned above, the contribution from the predictor has been ignored for the following reasons: 1.) The size of the predictor and hence its overall capacitance is smaller than that of the main MAC due to lower precisions, and 2.) The predictor is clocked at $8X$ slower than the main MAC. The implementation is currently under fabrication and measured results will be published at a later date.

6 CONCLUSIONS

In this paper, we have presented a VLSI implementation of a digital filter developed in the soft DSP framework. We have shown that, delay imbalance at the circuit-level aids the soft-DSP approach in obtaining higher energy savings with marginal degradation in performance. Future work on this topic involves the development error-control techniques for widely used DSP algorithms where energy dissipation is a critical issue.

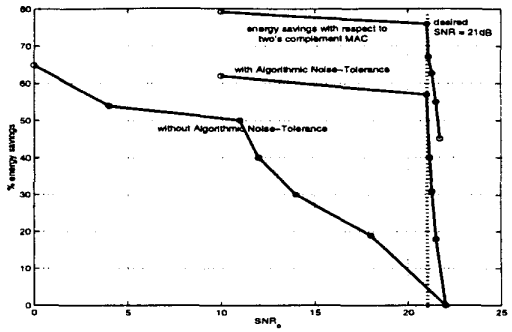


Figure 4: energy-dissipation vs. algorithmic performance for the soft DSP filter.

References

[1] R. Gonzalez, et. al., "Supply and threshold voltage scaling for low power CMOS," *IEEE Journal of Solid State Circuits,* vol. 32, No. 8, Aug. 1997.

[2] R. Hegde and N. R. Shanbhag, "A Low-power phase splitting passband equalizer," *IEEE Trans. on Signal Processing,* vol. 47, no. 3, March 1999.

[3] A. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proceedings of the IEEE*, vol. 83, no. 4, pp. 498-523, April 1995.

[4] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," *Proceedings of 1999 International Symposium on Low Power Electronics and Design,* pp. 30-35, August, 1999.

[5] R. Hegde and N. R. Shanbhag, "Low-power digital filtering via soft DSP," *ICAASP, 2000,* Instanbul, Turkey, June, 2000.

[6] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective.* Prentice-Hall, New Jersey, 1996.

[7] K. L. Shepard "Conquering Noise in Deep-Submicron Digital ICs," *IEEE Design and Test of Computers*, Jan-Mar 98, pp. 51-62.

[8] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications* Prentice-Hall, New Jersey, 1996.