Transactions Briefs

Design of Voltage Overscaled Low-Power Trellis Decoders in Presence of Process Variations

Yang Liu, Tong Zhang, and Jiang Hu

Abstract—**In hardware implementations of many signal processing functions, timing errors on different circuit signals may have largely different importance with respect to the overall signal processing performance. This motivates us to apply the concept of *unequal error tolerance* to enable the use of voltage overscaling at minimal signal processing performance degradation. Realization of unequal error tolerance involves two main issues, including how to quantify the importance of each circuit signal and how to incorporate the importance quantification into signal processing circuit design. We developed techniques to tackle these two issues and applied them to two types of trellis decoders including Viterbi decoder for convolutional code decoding and Max-Log-Maximum A Posteriori (MAP) decoder for Turbo code decoding. Simulation results demonstrated promising energy saving potentials of the proposed design solution on both trellis decoding computation and memory storage at small decoding performance degradation.**

Index Terms—**Clock skew scheduling, low-power, process variations, trellis decoders, unequal error tolerance, voltage overscaling (VOS).**

I. Introduction

Trellis decoders are pervasive in digital communication and data storage for realizing froward error correction and signal detection. Hence, it is of great practical importance to minimize the energy consumption of trellis decoders. Voltage scaling is an effective means of reducing energy consumption in CMOS integrated circuits (IC) [1]–[3]. In conventional practice, voltage scaling is lower bounded by $V_{\text{dd-crit}}$ under which the critical path delay equals the desired clock period. As the CMOS technology continuously scales down, process variations are becoming increasingly significant and leaves less room for conventional voltage scaling. It has been recently demonstrated [3]–[5] that voltage overscaling (VOS) (i.e., overscale the supply voltage below $V_{\text{dd-crit}}$) accompanied with appropriate error compensation may further reduce the overall energy consumption while maintaining satisfactory system performance. Leveraging the fact that most signal processing functions mainly concern certain quantitative performance criteria, such as bit error rate (BER) and signal-to-noise ratio (SNR), Shanbhag [4] proposed a design methodology, called arithmetic noise-tolerance (ANT), to design voltage overscaled low-power signal processing integrated circuits (ICs). However, this methodology is not directly applicable to the family of trellis decoding, where an explicit error control block is not readily available.

As an attempt to fill this gap, this paper presents a solution to design voltage overscaled trellis decoders. The basic idea is that, in most signal processing functions, VOS-induced errors on different circuit signals may result in different signal processing performance degradation. Intuitively, to minimize the signal processing performance degradation while pushing the envelope of energy efficiency, different circuit signals should have different immunity to VOS-induced errors in proportional to their system-wise importance. This is referred to as *unequal error tolerance*. To quantify the importance of each circuit signal with respect to the signal processing performance, we propose an approach based on the principle of random perturbation inspired by the following intuition: If we randomly perturb (or flip) a signal with certain probability, the corresponding signal processing performance degradation may indicate the importance of this signal regarding to the overall signal processing performance. In synchronous circuits, path timing slacks can be adjusted by intentionally tuning the clock skews, which is called clock skew scheduling [6]–[8]. Intuitively, unequal error tolerance can be realized by embedding the importance measurements of circuit signals into clock skew scheduling formulation, which is referred to as soft clock skew scheduling [9]. In this work, we propose a new formulation, called variation-aware soft clock skew scheduling, to realize unequal error tolerance for signal processing circuits in presence of process variations. In the context of memory storage, the realization of unequal error tolerance may be achieved through appropriate memory partition and unequal voltage scaling based on the importance measurement of the data to be stored.

In particular, we considered both convolutional code decoder using the Viterbi algorithm and Turbo code decoder using the Max-Log-Maximum A Posteriori (MAP) algorithm. We further developed different techniques to design voltage overscaled memory systems in Viterbi decoder and Max-Log-MAP decoder, respectively, that can tolerate significant VOS-induced memory operation errors at small decoding performance degradation.

Manuscript received October 09, 2007; revised February 08, 2008. First published January 13, 2009; current version published February 19, 2009. This work was supported by the National Science Foundation under Grant ECCS 0522457.

Y. Liu and T. Zhang are with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute Troy, NY 12180 USA (e-mail: liuy8@rpi.edu; tzhang@ecse.rpi.edu).

J. Hu is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: jianghu@ee.tamu.edu).

Digital Object Identifier 10.1109/TVLSI.2008.2004545

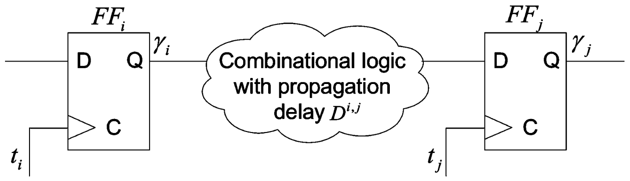

Fig. 1. Example of synchronous digital circuits.

II. Background

A. Conventional Clock Skew Scheduling

As illustrated in Fig. 1, In a synchronous circuit, clock skew is defined as the time difference, $s_{i,j} = t_i - t_j$, between the clock arrival times t_i and t_j of two sequentially adjacent flip-flops (FFs) FF_i and FF_j. Let T_{CP} denote the clock period, and $D^{i,j}_{\text{MAX}}$ and $D^{i,j}_{\min}$ denote the maximum and minimum propagation delays from FF_i and FF_j, respectively. The value of clock skew $s_{i,j}$ should fall into $\left[-D^{i,j}_{\min}, T_{\text{CP}} - D^{i,j}_{\text{MAX}}\right]$ that is called permissible range [8].

To improve the circuit reliability, a safety margin M may be introduced between the clock skew and the ends of the permissible range. Clock skew scheduling refers to a process that optimizes the safety margins subject to certain criteria, which can be formulated and solved using various optimization techniques such as linear programming [6]

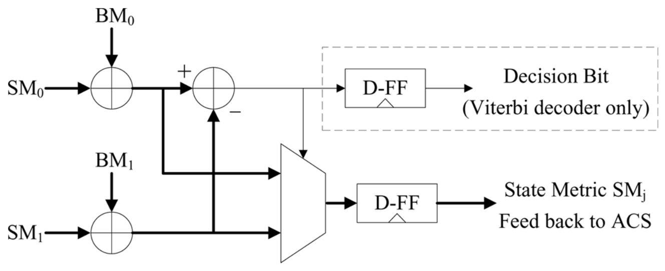

Fig. 2. ACS structure.

or some graphical methods [7]. One of the most widely used clock skew formulations is shown as follows:

$$\begin{aligned} &\text{Max} \quad M \\ &\text{Subject to:} \quad s_{i,j} \le T_{\mathrm{CP}} - D_{\mathrm{MAX}}^{i,j} - M \\ &\qquad\qquad\quad s_{i,j} \ge -D_{\min}^{i,j} + M. \end{aligned} \tag{1}$$

Under significant process variations, Monte Carlo simulation or various statistical timing analysis techniques, e.g., see [10] and [11], can be used to estimate $D_{\mathrm{MAX}}^{i,j}$ and $D_{\min}^{i,j}$ based on certain probability criterion.

Finally, Lu *et al.* [12] recently showed that clock skew may possibly reduce the clock network energy consumption if the register placement is done properly.

B. Trellis Decoders

Trellis decoders can be either hard- or soft-output. Hard-output trellis decoders typically use the Viterbi algorithm, and soft-output trellis decoders may employ several different algorithms including Log-MAP, Max-Log-MAP, and soft-output Viterbi algorithm (SOVA) [13]. This work considers the Max-Log-MAP algorithm for soft-output trellis decoding. In Viterbi decoders, trellis state metric computation is realized by add-compare-select (ACS). The memory fabric in Viterbi decoders stores the decision bits from ACS units and generates the decoder output. In Max-Log-MAP decoders, the trellis state metric computation can also be simply reduced to ACS, as proposed in [14]. The memory fabric in Max-Log-MAP decoders stores the state metrics based on which the soft output is calculated. Fig. 2 illustrates the structure of one ACS unit.

III. Proposed Techniques for Realizing Unequal Error Tolerance

A. Determination of Importance Factors

We conduct trial-and-error computer simulations to empirically search a random perturbation probability $p_r \in (0\%, 100\%)$ for each signal so that it degrades the signal processing performance into a pre-specified performance range $[Y_{\mathrm{target}} - \delta, Y_{\mathrm{target}} + \delta]$. The higher the random perturbation probability associated with one signal is, the less important this signal will be. Thus, we may use $1 - p_r$ as the importance factor of each signal.

System designers should appropriately determine the target performance degradation range and the simulation environment. Fig. 3 shows the overall flow diagram of this random perturbation method, which is further illustrated by the following example.

1) Example 3.1: Let the finite word-length of state metrics in Viterbi decoder and Max-Log-MAP decoder be 8 and 9, respectively. We consider a rate-1/2 convolutional code with a 128-state trellis and a rate-1/3 Turbo code with a 8-state trellis, and assume these codes are modulated by binary phase shift keying (BPSK) and transmitted over an AWGN channel. The decoding performance is measured in terms of

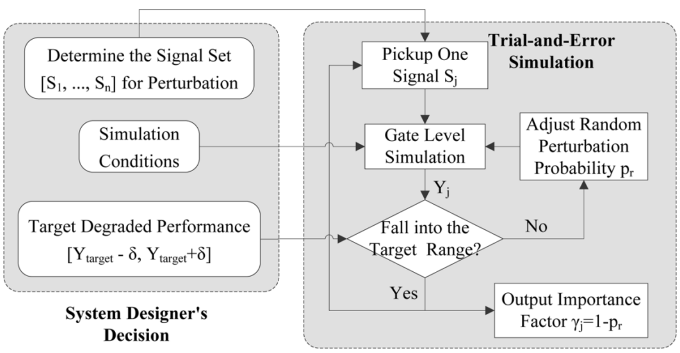
Determine the Signal Set [S1, ..., Sn] for Perturbation
Simulation Conditions
Target Degraded Performance [Ytarget - δ, Ytarget+δ]
System Designer's Decision
Trial-and-Error Simulation
Pickup One Signal Sj
Gate Level Simulation
Adjust Random Perturbation Probability pr
Yj
Fall into the Target Range?
No
Yes
Output Importance Factor γj=1-pr

Fig. 3. Diagram of the random perturbation method for determining the importance factors.

TABLE I
Importance Factors of the ACS Unit Outputs

	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4
Viterbi	~0.0000	0.7165	0.9401	0.9885	0.9981
Max-Log-MAP	~0.0000	0.6141	0.9192	0.9786	0.9951
	Bit 5	Bit 6	Bit 7	Bit 8	
Viterbi	~1.0000	~1.0000	~1.0000	0.9998	
Max-Log-MAP	0.9981	0.9998	~1.0000	~1.0000	

SNR versus BER. The Viterbi decoder achieves a BER of 1.7×10^{-5} under 4-dB input data SNR, and the Max-Log-MAP decoder has a BER of 3.9×10^{-5} under 2-dB input data SNR. We set the target BER degradation ranges for the Viterbi and Max-Log-MAP decoders as $[5.8 \times 10^{-4}, 6.2 \times 10^{-4}]$ under 4 dB and $[6.4 \times 10^{-3}, 6.6 \times 10^{-3}]$ under 2 dB, respectively. We conducted trial-and-error simulations to search the corresponding random perturbation probability for each signal and obtain the importance factors listed in Table I. For the Viterbi decoder, Bit 0 and Bit 7 are the least significant bit (LSB) and most significant bit (MSB) in the 8-bit state metric, and Bit 8 is the decision bit; for the Max-Log-MAP decoder, Bit 0 and Bit 8 are the LSB and MSB in the 9-bit state metric.

B. Variation-Aware Soft Clock Skew Scheduling

For each pair of sequentially adjacent FFs (e.g., FF_i and FF_j), let $\mathcal{F}_{\mathrm{MAX}}^{i,j}(\tau)$ $\left(\mathcal{F}_{\min}^{i,j}(\tau)\right)$ and $\sigma_{\mathrm{MAX}}^{i,j}$ $\left(\sigma_{\min}^{i,j}\right)$ represent the cumulative distribution function (CDF) and standard deviation of the maximum (minimum) propagation delay from FF_i to FF_j. Let γ_j represent the importance factor of the output signal of FF_j. The formulation of variation-aware soft clock skew scheduling is obtained by scaling the safety margin M by the product of destination signal importance factor and the path delay deviation. We take the path delay variation into account because, intuitively, the path with larger delay deviation should have more conservative timing constraints. Therefore, based on the conventional clock skew scheduling formulation in (1), the variation-aware soft clock skew scheduling is formulated as

$$\begin{aligned} &\text{Max} \quad M \\ &\text{Subject to:} \quad s_{i,j} \le T_{\mathrm{CP}} - D_{\mathrm{MAX}}^{i,j} - \gamma_j \cdot \sigma_{\mathrm{MAX}}^{i,j} \cdot M \\ &\qquad\qquad\quad s_{i,j} \ge -D_{\min}^{i,j} + \gamma_j \cdot \sigma_{\min}^{i,j} \cdot M. \end{aligned} \tag{2}$$

IV. Voltage Overscaled Viterbi and Max-Log-MAP Decoders

A. Voltage Overscaled ACS Units Array

1) ACS Unit Gate-Level Realization and Delay Model: Because of the relatively small finite word-lengths, the adders in each ACS unit can simply use the ripple-carry adder structure. Fig. 4 shows the gate-level

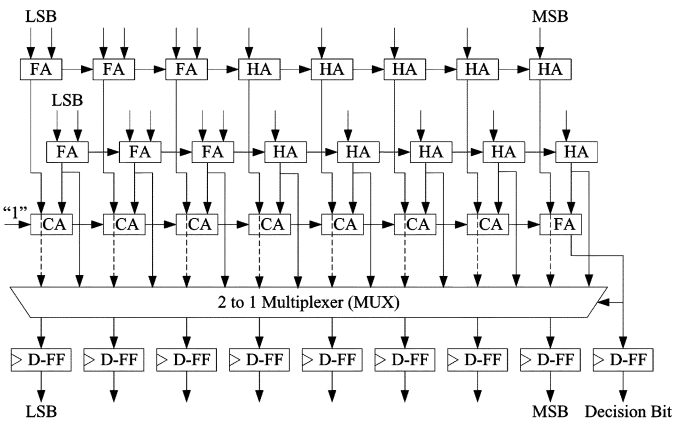

Fig. 4. Gate-level ACS unit structure in the Viterbi decoder.

TABLE II
IMPORTANCE-AWARE CLOCK SKEW SCHEDULING RESULTS

	Viterbi			Max-Log-MAP		
σ	0.05	0.10	0.15	0.05	0.10	0.15
Bit 0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bit 1	0.4556	0.0294	0.0000	0.3076	0.0403	0.0000
Bit 2	1.2624	0.5303	0.3313	1.2564	0.6348	0.4528
Bit 3	1.6086	1.2143	1.0215	1.7921	1.1220	0.8738
Bit 4	2.2087	1.7206	1.5132	1.6528	1.6351	1.2094
Bit 5	2.8902	2.1280	1.9735	2.8192	2.0971	1.5033
Bit 6	3.2634	2.6202	2.3991	3.1706	2.5740	1.8347
Bit 7	3.5588	2.8018	2.4292	3.6512	3.0372	2.1645
Bit 8	2.3467	1.1856	0.7013	3.6722	3.1706	2.5790

realization of an ACS unit in the Viterbi decoder, where the branch metric is 3-bit. The structure of ACS units in the Max-Log-MAP decoder can be obtained similarly. Each ACS unit contains five types of basic gates including 1-bit full adder (FA), 1-bit half adder (HA), 1-bit carry-only adder (CA) that only generates carry-out bit, 2-to-1 multiplexer (MUX), and D-FF.

The delay variations of all the basic gates are modelled as independent Gaussian random variables. We assume all the basic gates use static CMOS circuit structure and have following normalized delay distributions: For each HA, the input-to-carry and input-to-sum delays are modelled as $\mathcal{N}(1,\sigma)$; for each FA, the input-to-carry and input-to-sum delays are modelled as $\mathcal{N}(1,\sigma)$ and $\mathcal{N}(2,\sqrt{2}\sigma)$, respectively; for each CA and MUX, the delay is modelled as $\mathcal{N}(1,\sigma)$; for each D-FF, the clock-to-Q delay is modelled as $\mathcal{N}(2,\sqrt{2}\sigma)$ while we assume that the setup and hold time can be approximated as zero.

2) Simulation Results: We applied the variation-aware soft clock skew scheduling to the ACS units array in the Viterbi and Max-Log-MAP decoders considered in Example 3.1. Given the importance factors of ACS outputs in Table I, we formulate the clock skew scheduling according to (2). Applying a linear programming solver, we obtain the the optimized clock signal delay at each D-FF for different process variations as listed in Table II, where we take the the delay of a HA as the unit delay. Clearly, in practice clock skew cannot be scheduled in such high precision. Therefore, during the gate-level simulation, we quantize these values listed in Table II into 3-bit values with 2-bit integer and 1-bit fraction. Since the unit delay is the delay of one HA, it should be feasible to tune the skew with a precision of a half unit delay.

Let $K_v \in (0,1]$ represent the VOS factor, i.e., the supply voltage V_{dd} is scaled by K_v from the lower bound $V_{\mathrm{dd\text{-}crit}}$. Following the discussion in [15], the path propagation delay is linearly proportional to $V_{\mathrm{dd}}/(V_{\mathrm{dd}}-V_t)^{\alpha}$, where V_t is device threshold voltage and $\alpha \in [1,2]$ is the velocity saturation index that is set as 1.2 in this work. It is well known that the dynamic power consumption is proportional to the square of the supply voltage, i.e., $P_{\mathrm{dynamic}} \propto V_{\mathrm{dd}}^2$. Therefore, the ACS computation energy saving under VOS is approximated as $(1-K_v^2)$.

Based on the gate-level Monte Carlo simulations, we obtain the SNR versus BER performance curves shown in Fig. 5 under several different values of K_v. It shows that the proposed design approach can considerably reduce the decoding performance degradation incurred by VOS. Hence, this approach can readily support graceful performance degradation in presence of significant voltage overscaling. Assume we can consider less than 0.1 dB performance degradation at BER of $\sim 10^{-5}$ as negligible and hence acceptable in practice, Table III shows the power savings gained by using the proposed variation-aware soft clock skew scheduling at no cost of decoding performance.

B. Voltage Overscaled Memory System Design

1) Voltage Overscaled Memory System Design in Viterbi Decoders: In Viterbi decoders, memory stores the decision bits and generates the decoder output using a trace-back procedure [16]. Logically, we may consider the memory consists of three blocks, corresponding to three different phases called write, trace back, and decode. At write phase, decision bits are written into one entire column of memory. At trace back phase, we trace the survivor path from an arbitrary state. After L-step trace back, where L is called trace-back length, we reach a convergence trellis state, from which decode phase starts and generates decoding output using the same trace-back procedure as in the trace back phase.

Intuitively, read errors in decode phase and/or write errors in write phase tend to greatly degrade the decoding performance. On the other hand, our simulations suggest that read errors in in the trace back phase, especially those in early stage of the trace back phase, may not result in significant decoding performance degradation. Hence, we may use the random perturbation approach as described in Section III-A to determine the importance factor of the bits at different depths along the trace back, which is illustrated in the following example. For the 128-state Viterbi decoder considered above, we set $L = 56$ and the target degraded BER ranges as $[6.4\times10^{-3}, 6.6\times10^{-3}]$ under 2 dB, under which we obtain the importance factor for each bit through trial-and-error simulations.

In this work we use the 3-pointer even scheme [17], which is widely used in practice [18], [19] to design the Viterbi decoder memory system, to evaluate the energy saving potential. We propose to use the following *dynamic VOS* scheme. Each memory bank may operate with three different overscaling factors K_v, i.e., $1 > K_v^H > K_v^L$. When one memory bank operates in write or decode phase, it sets $K_v = 1$ (i.e., without VOS); while it operates in the second half of trace back phase, we have $K_v = K_v^H$, otherwise we have $K_v = K_v^L$. Let P_e^H and P_e^L represent the memory error rates due to the overscaled voltages with K_v^H and K_v^L, respectively. Fig. 6 shows the simulated decoding performance using such dynamic VOS scheme. Recently, Kurdahi *et al.* [20] investigated such quantitative relationship at 70-nm CMOS technology with $\sigma = 0.107$ process variations in V_{th}, and showed 1% memory errors may correspond to about 25%–30% power savings.

2) Voltage Overscaled Memory System Design in Max-Log-Map Decoders: In Max-Log-MAP decoders, the memory stores all the intermediate trellis state metrics generated by ACS units. As shown in Table I, bits at different positions have different importance factors. Therefore, ideally they should be stored in different memory banks, and the supply voltages of all the memory banks should be scaled so that the VOS-induced error rates are inversely proportional to the corresponding importance factors. In practice, such bit-wise voltage-scaling granularity may not be practical and/or necessary. We may partition the bits in each state metric into certain groups, each group of bits are

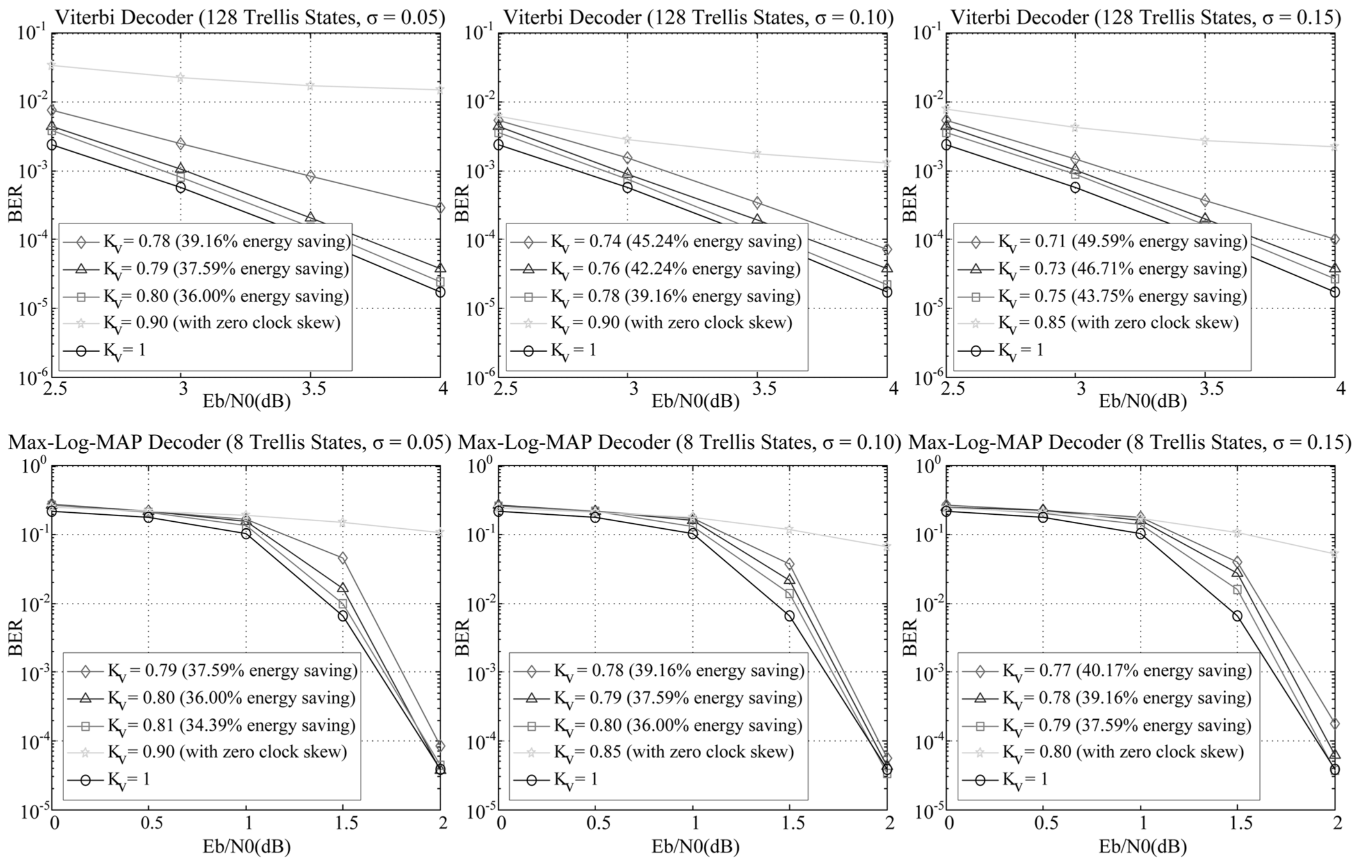

Fig. 5. BER performance after applying the importance-aware clock skew scheduling on Viterbi and Max-Log-MAP decoders.

TABLE III
POWER SAVINGS AT NEGLIGIBLE PERFORMANCE DEGRADATION

σ	0.05	0.1	0.15
Viterbi	36.00%	39.16%	43.75%
Max-Log-MAP	34.39%	36.00%	37.59%

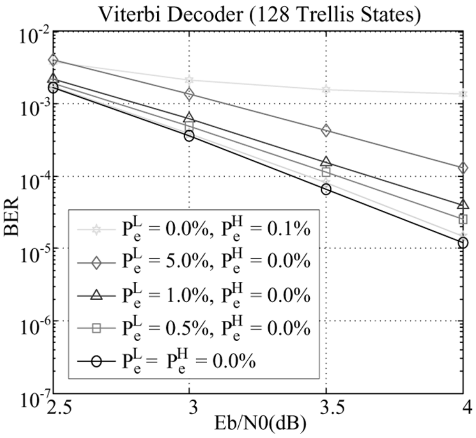

Fig. 6. BER performance with memory errors in Viterbi decoders.

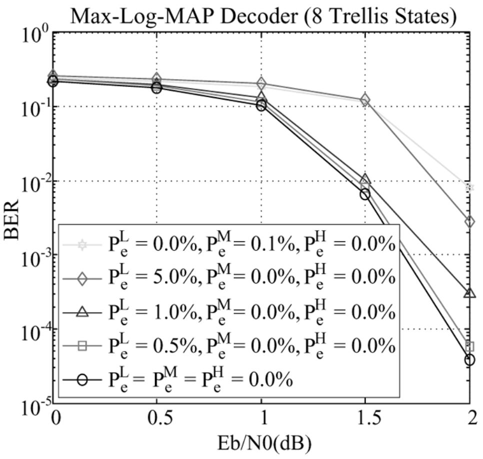

Fig. 7. BER performance with memory errors in Max-Log-MAP decoders.

stored in the same memory bank with the same supply voltage. This work simply partitions each 9-bit metric into three 3-bit groups that are stored in three memory banks. The memory operation error rates associated with the three memory banks are denoted as P_e^H, P_e^M, and P_e^L, respectively, where P_e^H and P_e^L correspond to the 3-bit MSBs and LSBs. Fig. 7 shows the simulation results under different operation error rates. As previously pointed out, the allowable memory operation error rates can be translated to memory energy savings and the quantitative relationship depends on the specific memory implementation.

V. CONCLUSION

This paper presents an unequal error tolerance design framework to enable the aggressive use of voltage overscaling for low-power trellis decoder IC implementation under process variations. This design framework is motivated by the fact that errors on different circuit signals may have largely different importance with respect to the system level performance. We developed an approach to quantify the importance measurement and a method to embed the importance measurement into computation datapath in order to realize unequal error tolerance. Under this unequal error tolerance framework, we further developed approaches to use voltage overscaling in memory systems of trellis decoders. Effectiveness of such an unequal error tolerance framework and the developed techniques have been successfully demonstrated using computer simulations.

References

[1] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, vol. 83, no. 4, pp. 498–523, Apr. 1995.

[2] R. Gonzalez, B. M. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE J. Solid-State Circuits*, vol. 32, no. 8, pp. 1210–1216, Aug. 1997.

[3] G. Karakonstantis, N. Banerjee, K. Roy, and C. Chakrabarti, "Design methodology to trade off power, output quality and error resiliency: Application to color interpolation filtering," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2007, pp. 199–204.

[4] N. R. Shanbhag, "Reliable and energy-efficient digital signal processing," in *Proc. Des. Autom. Conf.*, Jun. 2002, pp. 830–835.

[5] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "A self-tuning DVS processor using delay-error detection and correction," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, Apr. 2006.

[6] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. Computers*, vol. 39, no. 7, pp. 945–951, Jul. 1990.

[7] R. B. Deokar and S. S. Sapatnekar, "A graph-theoretic approach to clock skew optimization," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 1994, pp. 407–410.

[8] J. L. Neves and E. G. Friedman, "Optimal clock skew scheduling tolerant to process variations," in *Proc. Des. Autom. Conf. (DAC)*, Jun. 1996, pp. 623–628.

[9] Y. Liu, T. Zhang, and J. Hu, "Low power trellis decoder with overscaled supply voltage," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS): Des. Implementation*, 2006, pp. 205–208.

[10] A. Agarwal, V. Zolotov, and D. T. Blaauw, "Statistical timing analysis using bounds and selective enumeration," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 9, pp. 1243–1260, Sep. 2003.

[11] H. Chang and S. S. Sapatnekar, "Statistical timing analysis under spatial correlations," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 9, pp. 1467–1482, Sep. 2005.

[12] Y. Lu, C. N. Sze, X. Hong, Q. Zhou, Y. Cai, L. Huang, and J. Hu, "Register placement for low power clock network," in *Proc. Asia South Pacific Des. Autom. Conf. (ASP-DAC)*, Jan. 2005, pp. 588–593.

[13] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 3, pp. 429–445, Mar. 1996.

[14] J. H. Han, A. T. Erdogan, and T. Arslan, "High speed Max-Log-Map turbo SISO decoder implementation using branch metric normalization," in *Proc. IEEE Comput. Soc. Annu.Symp. VLSI*, May 2005, pp. 173–178.

[15] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 6, pp. 813–823, Dec. 2001.

[16] Y.-N. Chang, H. Suzuki, and K. K. Parhi, "A 2-Mb/s 256-state 10-mW rate-1/3 Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 35, no. 6, pp. 826–834, Jun. 2000.

[17] G. Feygin and P. Gulak, "Architectural tradeoffs for survivor sequence memory management in Viterbi decoders," *IEEE Trans. Commun.*, vol. 41, no. 3, pp. 425–429, Mar. 1993.

[18] Li H.-L and C. Chakrabarti, "A new architecture for the Viterbi decoder for code rate k/n," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 158–164, Feb. 1996.

[19] C.-C. Lin, Y.-H. Shih, H.-C. Chang, and C.-Y. Lee, "Design of a power-reduction Viterbi decoder for WLAN applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 6, pp. 1148–1156, Jun. 2005.

[20] F. J. Kurdahi, A. M. Eltawil, Y.-H. Park, R. N. Kanj, and S. R. Nassif, "System-level sram yield enhancement," in *Proc. Int. Symp. Quality Electron. Des.*, Mar. 2006, p. 6.

Fast Scaling in the Residue Number System

Yinan Kong, *Student Member, IEEE*, and
Braden Phillips, *Member, IEEE*

***Abstract*—A new scheme for precisely scaling numbers in the residue number system (RNS) is presented. The scale factor K can be any number coprime to the RNS moduli. Lookup table implementations are used as a basis for comparisons between the new scheme and scaling schemes from the literature. It is shown that new scheme decreases hardware complexity compared to previous schemes without affecting time complexity.**

***Index Terms*—Computational complexity, digital arithmetic, table lookup, residue arithmetic.**

I. Introduction

A. Residue Number System and Scaling

The residue number system (RNS) provides a means for efficient multiplication and addition of integers; however, scaling within RNS is less efficient and this problem has long prevented wider adoption of RNS. In this context, scaling an integer X means reducing its word length by dividing by a constant K

$$Y = \left\lfloor \frac{X}{K} \right\rfloor . \tag{1}$$

In binary arithmetic, K is usually chosen to be a power of 2 such that word length reduction is achieved by simply truncating a number's binary representation. There is no equivalent operation in RNS with the consequence that a result accumulated through a sequence of multiplications [as is often the case in digital filters or multiple-point fast Fourier transfers (FFTs)] can grow in word length until it overflows the dynamic range of the RNS.

An RNS [1] is characterized by a set of N coprime moduli $\{m_1, m_2, \ldots, m_N\}$. In the RNS, a number X is represented in N channels: $X = \{x_1, x_2, \ldots, x_N\}$, where x_i is the residue of X with respect to m_i, i.e., $x_i = \langle X \rangle_{m_i} = X \bmod m_i$. Within the RNS there is a unique representation of all integers in the range $0 \leq X < M$, where $M = m_1 m_2, \ldots, m_N$. M is therefore known as the dynamic range of the RNS. Two other values, M_i and $\langle M_i^{-1} \rangle_{m_i}$ are commonly used in RNS computations and are worth defining here. $M_i = (M/m_i)$ and $\langle M_i^{-1} \rangle_{m_i}$ is its multiplicative inverse with respect to m_i such that $\langle M_i \times M_i^{-1} \rangle_{m_i} = 1$.

If X, Y, and Z have RNS representations given by $X = \{x_1, x_2, \ldots, x_N\}$, $Y = \{y_1, y_2, \ldots, y_N\}$, and $Z = \{z_1, z_2, \ldots, z_N\}$, then denoting * to represent the operations +, -, or $\times$, the RNS version of $Z = X * Y$ satisfies

$$Z = \{\langle x_1 * y_1 \rangle_{m_1}, \langle x_2 * y_2 \rangle_{m_2}, \ldots, \langle x_N * y_N \rangle_{m_N}\}.$$

Thus addition, subtraction, and multiplication can be concurrently performed on the N residues within N parallel channels, and it is this high

Manuscript received June 21, 2007; revised January 15, 2008. First published January 13, 2009; current version published February 19, 2009.. This work was supported by the Australian Research Council's Discovery Project Scheme (DP0559582).

The authors are with the Centre for High Performance Integrated Technologies and Systems (CHiPTec), the School of Electrical and Electronic Engineering, the University of Adelaide, Adelaide, SA 5005, Australia (e-mail: ykong@eleceng.adelaide.edu.au; phillips@eleceng.adelaide.edu.au).

Digital Object Identifier 10.1109/TVLSI.2008.2004550