Low Power Trellis Decoder with Overscaled Supply Voltage

Yang Liu*, Tong Zhang*, and Jiang Hu†
* ECSE Department, Rensselaer Polytechnic Institute, Troy, NY
† Department of ECE, Texas A&M University, College Station, TX

Abstract— **This paper is interested in applying voltage overscaling (VOS) to reduce trellis decoder energy consumption, where the key issue is how to minimize the decoding performance degradation due to VOS-induced errors. Based on the fact that the integrity of different bits in the trellis state metric has (largely) different effect on the overall trellis decoding performance, we proposed an importance-aware clock skew scheduling technique that assigns those more important bits with longer timing slacks and hence better immunity to VOS-induced errors. This will provide system-level tolerance to VOS-induced errors in trellis decoders. With Viterbi and Max-Log-MAP decoders as test vehicles, we demonstrated that about 30% energy savings on trellis state metric computation can be realized with negligible decoding performance degradation.**

I. Introduction

Voltage scaling is an effective means of reducing energy consumption in CMOS integrated circuits (IC). In conventional practice, voltage scaling is lower bounded by $V_{\text{dd-crit}}$ under which the critical path delay equals the desired clock period. Overscaling supply voltage below $V_{\text{dd-crit}}$ will result in logic errors. Leveraging the fact that most digital signal processing (DSP) functions mainly concern certain quantitative performance criteria, such as bit error rate (BER) and signal to noise ratio (SNR), Shanbhag [1] proposed a design methodology called arithmetic noise-tolerance (ANT) to enable the use of voltage overscaling (VOS) in DSP IC to further reduce the energy consumption. The key idea of ANT is to apply a separate and simple error control block to compensate the signal processing performance degradation due to the VOS-induced errors. Its effectiveness has been successfully demonstrated for various filters [2]–[4] and fast Fourier transform (FFT) [5]. However, this methodology is not directly applicable to the family of trellis decoding, where an explicit error control block is not readily available.

This work proposed a technique to enable the use of VOS in trellis decoding. Although it follows the same principle behind ANT, i.e., to provide system-level tolerance to VOS-induced errors, this proposed technique applies a fundamentally different approach to realize such system-level error tolerance. In synchronous circuits, whether a path is subject to potential VOS-induced errors is determined by its timing slack that depends on both the path propagation delay and the clock skew (i.e., the clock arrival time difference) between the state holding elements at the two ends of this path. Path timing slacks can be adjusted by intentionally tuning the clock skews, which is called clock skew scheduling [6]–[8]. Conventionally, clock skew scheduling has been used as a physical level implementation technique that is completely separate from higher level system and architecture design.

Intuitively, in most DSP functions (including trellis decoding), VOS-induced errors on different paths may lead to (largely) different signal processing performance degradation. This leads to the *key idea* of this work: we apply the clock skew scheduling in such an importance-aware manner that those more important paths have larger timing slacks and hence are more immune to VOS-induced errors. Therefore, the signal processing performance degradation due to VOS-induced errors can be minimized. Such importance-aware clock skew scheduling can be realized by incorporating an application-dependent parameter, called importance factor, that quantifies the importance of each individual circuit signal regarding to the overall signal processing performance. A random perturbation method is proposed to quantitatively determine such importance factors. This technique has been applied to convolutional code Viterbi decoder and Turbo code Max-Log-MAP decoder. Simulation results show that, under aggressive VOS, it can reduce the energy consumption on trellis state metric computation by about 30% while maintaining almost the same trellis decoding performance.

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 $D^{i,j}_{MAX}$ 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 $[-D^{i,j}_{min},\ T_{CP} - D^{i,j}_{MAX}]$ that is called permissible range [8]. To improve the circuit reliability, a safety margin presented by M may be introduced between the clock skew and the ends of the permissible

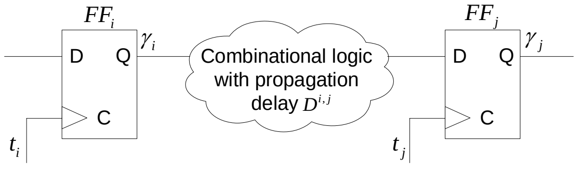

Fig. 1. Example synchronous digital circuits.

range. Clock skew scheduling refers to a process that optimizes the safety margins subject to certain criteria, which can be mathematically formulated and solved using various optimization techniques such as linear programming or some graphical methods. One of the most widely used clock skew formulations is shown as follows:

$$\begin{array}{rl} \text{Max} & M \\ \text{Subject to :} & s_{i,j} \leq T_{CP} - D^{i,j}_{MAX} - M \\ & s_{i,j} \geq -D^{i,j}_{min} + M \end{array} \tag{1}$$

B. Convolutional Code and Turbo Code Decoders

Convolutional code and Turbo code are being widely used in digital communications for realizing forward error correction (FEC). Convolutional code decoders typically use the well-known Viterbi algorithm, and Turbo code decoders may employ several different iterative soft-input soft-output (SISO) trellis decoding algorithms such as Log-MAP (Maximum A Posteriori), Max-Log-MAP, and SOVA (soft-output Viterbi algorithm). This work considers the Max-Log-MAP algorithm for Turbo code decoding. Because the recursive trellis state metric computation constitutes the essential critical path and main logic operation in both Viterbi and Max-Log-MAP decoders, we only consider the VOS-induced errors in trellis state metric computation. In Viterbi decoders, trellis state metric computation is realized by add-compare-select (ACS). In Max-Log-MAP decoders, trellis state metric computation is realized by ACS followed by metric normalization to avoid metric overflow. A technique has been proposed in [9] to transform the trellis state metric normalization to branch metric normalization so that the trellis state metric computation simply reduces to ACS. This technique can effectively reduce the Max-Log-MAP decoder critical path and hence is assumed in this work. Fig. 2 illustrates the structure of one ACS unit.

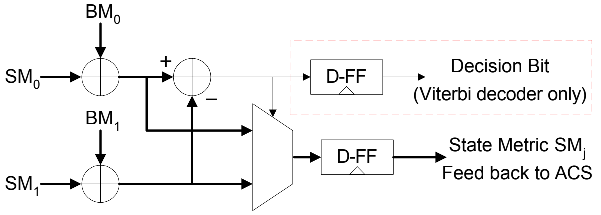

Fig. 2. ACS structure.

III. Importance-Aware Clock Skew Scheduling

Based on the conventional clock skew scheduling as described in Section II-A, we have an intuitive formulation for the importance-aware clock skew scheduling as follows:

$$\begin{array}{rl} \text{Max} & M \\ \text{Subject to :} & s_{i,j} \leq T_{CP} - D^{i,j}_{MAX} - \gamma_j \cdot M \ , \\ & s_{i,j} \geq -D^{i,j}_{min} + \gamma_j \cdot M \end{array} \tag{2}$$

where the importance factor $\gamma_j \in (0, 1)$ quantitatively represents the importance of the destination signal of each individual path, and a more important signal has a larger importance factor. Similar to the conventional clock skew scheduling, it can be solved by various optimization techniques such as linear programming or some graphical methods. Clearly, the critical issue in the importance-aware clock skew scheduling is how to determine the importance factor of each signal in the circuit.

In this work, we developed an approach to determine the importance factors inspired by the following intuition: If we randomly perturb (or flip) a signal with certain probability, the resulted signal processing performance degradation may indicate the importance of this signal regarding to the overall signal processing performance. This leads to the basic idea of the developed approach: We conduct trial-and-error computer simulations to empirically search a random perturbation probability $p_r \in (0\%, 100\%)$ for each signal, which will result in a signal processing performance that falls into a fixed target degraded performance range $[Y_{target} - \delta, \ Y_{target} + \delta]$. 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.

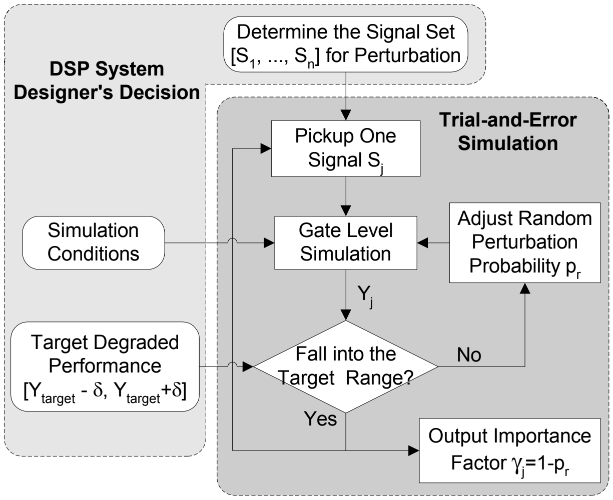
Determine the Signal Set [S1, ..., Sn] for Perturbation
DSP System Designer's Decision
Trial-and-Error Simulation
Pickup One Signal Sj
Simulation Conditions
Gate Level Simulation
Adjust Random Perturbation Probability pr
Yj
Target Degraded Performance [Ytarget - δ, Ytarget+δ]
Fall into the Target Range?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

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

For practical realization of this random perturbation method, we need the involvement of DSP system designers to tackle two issues: (i) If we conduct trial-and-error simulations to search the random perturbation probabilities exhaustively for all the signals, it may lead to very high computational overhead. Fortunately, most DSP systems contains a large number of identical (or similar) basic building blocks, in which those counterpart signals may have (roughly) the same importance to the overall signal processing performance. Therefore, DSP system designers may analytically identify those signals that may have the same or similar importance in order to largely reduce the computational overhead for searching the importance factors. (ii) DSP system designers should determine the appropriate target degraded performance range and the simulation condition (such as the input data SNR in Viterbi and Max-Log-MAP decoders) under which the trial-and-error simulations will be carried out. Fig. 3 shows the overall flow diagram of this random perturbation method, which is further illustrated by the following example.

Example 3.1: In Viterbi and Max-Log-MAP decoders, all the ACS units equally contribute to the decoding and hence have the same importance. Thus, we only need to carry out trial-and-error simulations on the outputs of one ACS unit. In this work, we set the finite word-length of state metrics in Viterbi decoder and Max-Log-MAP decoder as 8 and 9, respectively. Thus, the output of one ACS unit in Viterbi decoders contains 8-bit state metric and 1-bit decision, while the output of one ACS unit in Max-Log-MAP decoders is 9-bit state metric. Consider a rate-1/2 convolutional code with a 128-state trellis and a rate-1/3 Turbo code with an 8-state trellis. We assume these codes are modulated by BPSK (binary phase shift keying) and transmitted over an AWGN (additive white Gaussian noise) channel, and assume the decoding performance is measured in terms of SNR vs. BER. The Viterbi decoder has a BER of 1.7×10^{-5} under 4dB input data SNR, and the Max-Log-MAP decoder has a BER of 3.9×10^{-5} under 2dB input data SNR. We set the target degraded BER ranges for the Viterbi and Max-Log-MAP decoders as $[5.8 \times 10^{-4},\ 6.2 \times 10^{-4}]$ under 4dB and $[6.4 \times 10^{-3},\ 6.6 \times 10^{-3}]$ under 2dB, 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.

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

IV. APPLICATION TO VITERBI AND MAX-LOG-MAP DECODERS

A. ACS Unit Gate-Level Realization and Delay Model

Because of the relatively small finite word-lengths, the adders in each ACS unit of the Viterbi and Max-Log-MAP decoders can simply use the carry ripple adder structure. Fig. 4 shows the gate-level 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 (we set the branch metric as 8-bit in this work). Each ACS unit mainly 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.

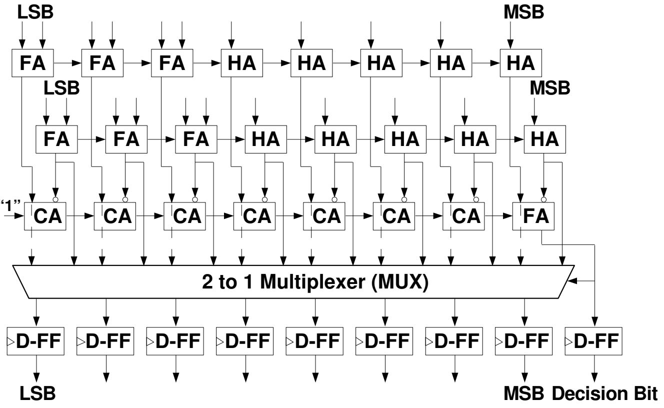

Fig. 4. Gate-level ACS unit structure in the Viterbi decoder.

Because the occurrence of VOS-induced errors is highly data dependent, we implemented gate-level logic simulators to empirically evaluate the Viterbi and Max-Log-MAP decoding BER performance in presence of VOS-induced errors. In order to reach the BER that is of practical interest (e.g., $10^{-3} \sim 10^{-5}$) in a reasonable amount of simulation time, the simulators employ a simplified gate delay model described as follows: For each HA, the input-to-carry and input-to-sum delays are 1; for each FA, the input-to-carry and input-to-sum delays are 1 and 2, respectively; for each CA and MUX, the delay is 1; for each D-FF, the clock-to-Q delay is 2 and the setup and hold time are approximated as zero.

B. Effectiveness of Importance-Aware Clock Skew Scheduling

We applied the importance-aware clock skew scheduling to 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 importance-aware clock skew scheduling according to (2). Applying a linear programming solver, we obtain the the optimized clock signal delay at each D-FF as listed in Table II.

TABLE II
IMPORTANCE-AWARE CLOCK SKEW SCHEDULING RESULTS

	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4
Viterbi	0.0000	1.0253	1.6983	2.2456	2.7651
Max-Log-MAP	0.0000	1.0252	1.7926	2.3548	2.8812

	Bit 5	Bit 6	Bit 7	Bit 8
Viterbi	3.2791	3.7917	4.3043	3.2604
Max-Log-MAP	3.3964	3.9104	4.4232	4.6796

It should be pointed out that, due to the recursive structure of the ACS computation, the supply voltage lower bound $V_{\text{dd-crit}}$ remains the same no matter how we tune the clock skews. 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 $V_{\text{dd-crit}}$. Following the discussion in [2], the path propagation delay is linearly proportional to $V_{\text{dd}}/(V_{\text{dd}} - V_{\text{t}})^{\alpha}$, where V_{t} is device threshold voltage and α is the velocity saturation index. The value of α is between 1 and 2 and determined empirically by curve fitting. The ACS computation energy saving under VOS is approximated as $(1 - K_v^2)$.

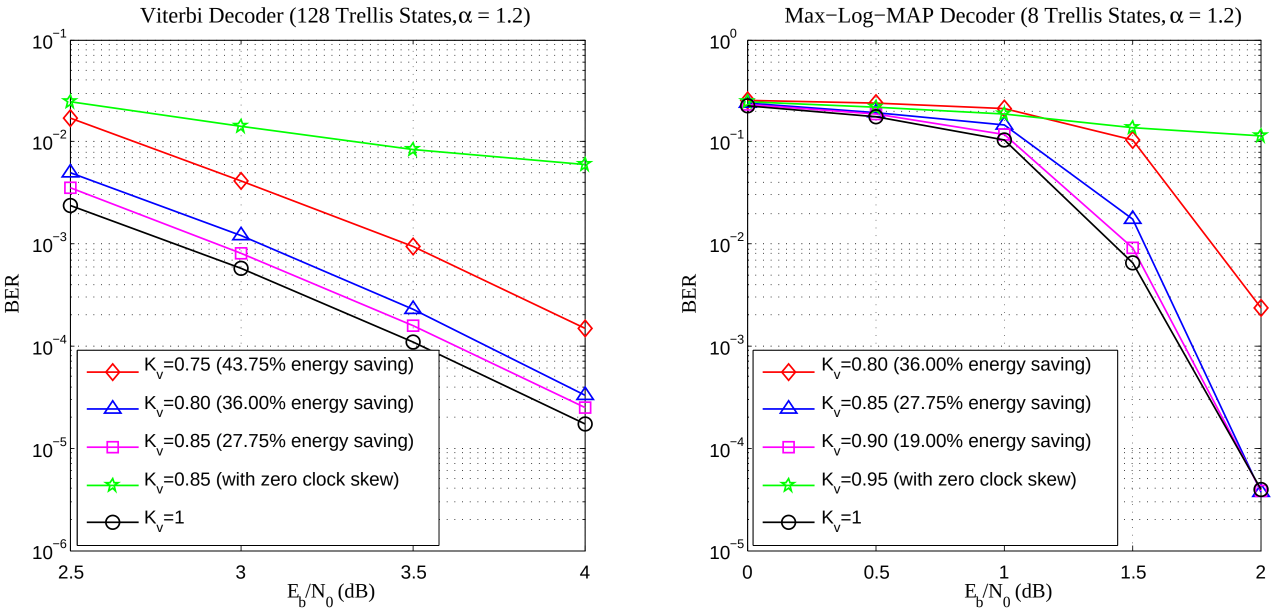

Fig. 5. BER performance after applying the importance-aware clock skew scheduling on Viterbi and Max-Log-MAP decoders.

Based on the above clock skew scheduling optimization results, we carried out gate-level Monte Carlo simulations to evaluate the Viterbi and Max-Log-MAP decoding performance under VOS. In this work, we assume the device threshold voltage is 0.62V and the velocity saturation index α is 1.2. Based on the gate-level sonte Carlo simulations, we obtain the SNR vs. BER performance curves in Fig. 5 under several different values of K_v. For the purpose of comparison, we include the ideal decoding BER performance (i.e., $K_v = 1$ and hence no logic errors will occur), and one simulated performance curve under VOS but without importance-aware clock skew scheduling (i.e., all the D-FFs have zero clock skew). Let P_{se} and P_{le} represent the average frequency of the occurrence of VOS-induced errors in each ACS unit with and without clock skew scheduling. Table III shows the average values of P_{se} and P_{le} based on the simulations. Although P_{se} is much larger than P_{le} because of the late arrival time of MSBs due to clock skew, most timing faults only occur in the lower bits and the error frequencies associated with the higher bits (or those more important bits) are very low. The above simulation results demonstrate that, by using the importance-aware clock skew scheduling, we may achieve a significant energy savings on the trellis state metric computation while maintaining very good decoding performance, even though the frequency of the occurrence of VOS-induced errors is very significant.

TABLE III
VOS-Induced Errors Statistics Under Different K_v ($\alpha = 1.2$).

	Viterbi			Max-Log-MAP		
K_v	0.85	0.80	0.75	0.95	0.90	0.85
P_{le}	9.15%	9.17%	9.37%	3.33%	3.36%	11.61%
P_{se}	28.17%	35.29%	39.22%	17.10%	29.80%	31.15%

V. Conclusions

In this paper, we present a solution to enable the use of voltage overscaling to reduce the energy consumption of trellis decoders. The basic idea is to apply importance-aware clock skew scheduling to realize system-level tolerance to the errors induced by voltage overscaling. A general formulation of importance-aware clock skew scheduling is presented and a random perturbation method has been proposed to empirically quantify the importance of different signals. Its effectiveness has been demonstrated using Viterbi and Max-Log-MAP decoders as test vehicles.

References

[1] N. R. Shanbhag, "Reliable and energy-efficient digital signal processing," in *Proc. of Design Automation Conference*, June 2002, pp. 830–835.

[2] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, pp. 813–823, Dec. 2001.

[3] L. Wang and N. R. Shanbhag, "Low-power filtering via adaptive error-cancellation," *IEEE Transactions on Signal Processing*, vol. 51, pp. 575–583, Feb. 2003.

[4] R. Hegde and N. R. Shanbhag, "A voltage overscaled low-power digital filter IC," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 388–391, Feb. 2004.

[5] B. Shim, S. R. Sridhara, and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, pp. 497–510, May 2004.

[6] J.P. Fishburn, "Clock skew optimization," *IEEE Transactions on Computers*, vol. 39, pp. 945–951, July 1990.

[7] R.B. Deokar and S.S. Sapatnekar, "A graph-theoretic approach to clock skew optimization," in *Proc. of IEEE International Symposium on Circuits and Systems*, June 1994, pp. 407–410.

[8] J.L. Neves and E.G. Friedman, "Optimal clock skew scheduling tolerant to process variations," in *Proc. of Design Automation Conference (DAC)*, June 1996, pp. 623–628.

[9] J.H. Han, A.T. Erdogan, and T. Arslan, "High speed max-log-MAP turbo SISO decoder implementation using branch metric normalization," in *Proc. of IEEE Computer Society Annual Symposium on VLSI*, May 2005, pp. 173–178.