VARIATION-TOLERANT MOTION ESTIMATION ARCHITECTURE

Girish V. Varatkar and Naresh R. Shanbhag

Coordinated Science Laboratory/ECE Department
University of Illinois at Urbana-Champaign
1308 W Main St., Urbana IL 61801.

ABSTRACT

In this paper, we study the trade-off between energy-efficiency and variation-tolerance of an error-resilient motion estimation architecture. Error-resiliency is incorporated via algorithmic noise-tolerance (ANT) where an input subsampled replica (ISR) of the main sum-of-absolute-difference (MSAD) block is employed for detecting and correcting errors in the MSAD block. This architecture is referred to as ISR-ANT. In the presence of process variations, the average peak signal-to-noise ratio ($PSNR$) of ISR-ANT architecture increases by up to $1.8dB$ over that of the conventional architecture in $130nm$ IBM process technology. Furthermore, the $PSNR$ variation is also reduced by $7\times$ over that of the conventional architecture at the slow corner while achieving a power reduction of 33%.

Index Terms— process variation, error resiliency

1. INTRODUCTION

Next generation wireless multimedia communications standards such as fourth generation (4G) mobile systems need to provide services such as video transmission on hand-held units. These units need to be energy-efficient while providing a high quality of service. Various video compression standards have been proposed to reduce the bandwidth of multimedia data transmission. The MPEG-4 encoder is the most computationally intensive block in a video processor. The motion estimation (ME) kernel consumes 66%-94% of the encoder computational complexity [1]. The ME datapath power consumption is found to be 75% of the total ME power consumption for full search motion estimation algorithm and 60% of the total ME power consumption for the three step search algorithm [2]. Therefore, low-power motion-estimation architectures and implementations are of great interest.

The ME implementations fabricated in nanometer silicon process technologies face the problem of performing energy efficient computation in the presence of noise. The nanometer process technologies suffer from non-idealities such as process variations, voltage or temperature induced noise and soft errors. One source of process variations is the random fluctuations in the number of dopant atoms in the MOS channel [3]

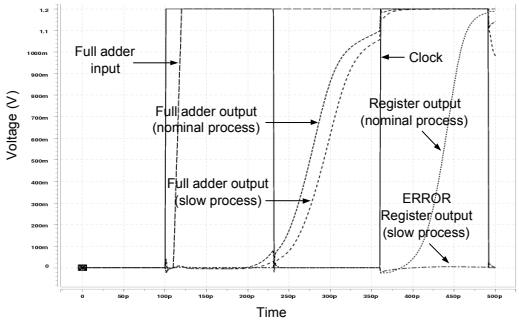

Fig. 1. A typical timing violation induced error due to process variations.

which affects the device threshold voltage V_t of the transistor. The usage of sub-wavelength lithography for patterning transistors results in width and gate-length variations. This creates delay variations which result in uncertainty in the data arrival time at the registers or memory elements causing them to latch incorrect data leading to logic errors. An example of such an error event is shown in Fig. 1 using HSPICE simulation of a latched full adder designed in an IBM $130nm$ process technology. We can see that the circuit which operates correctly at the nominal process corner produces an erroneous output at the slow process corner.

Previous schemes to *avoid* errors due to timing violations have relied on adaptive body biasing (ABB) to modulate the transistor threshold voltage V_t [4] and adaptive supply voltage (ASV) [5]. However, the effectiveness of ABB is known to decrease as the channel length shrinks while ASV requires accurate, power-hungry circuitry. Process variations cause 30% variability in operating frequency in current process technology and this variability is expected to increase to 60% within the next 10 years [6]. In the presence of such increased variations, a worst-case design has high power consumption while the nominal design, even though it is energy-efficient, will exhibit intermittent errors. Therefore, error-resilient architectures and implementations which trade-off power with reliability are of great interest [7].

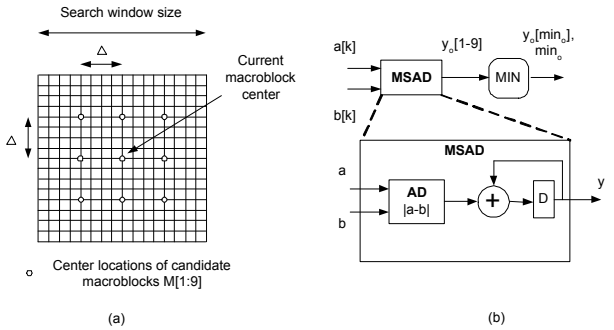

Fig. 2. The three step search (TSS) algorithm: (a) the search window, and (b) a block level implementation.

1.1. Contribution

In this paper, we study the performance of error-resilient low power ME architecture referred to as input subsampled replica ANT (ISR-ANT) [9] in the presence of errors due to process variations and voltage overscaling (VOS). In VOS, the supply voltage is reduced beyond $V_{dd-crit}$, the supply voltage below which timing violations occur, in order to push the limits of power savings using conventional voltage scaling [8]. Simulations using statistical process model of an IBM $130nm$ CMOS process technology show that ISR-ANT increases the mean peak signal-to-noise ratio ($PSNR$) by up to $1.8dB$ when compared to the $PSNR$ of the conventional architecture on a slow die. ISR-ANT also reduces the variation of the $PSNR$ due to WID variations around the slow process corner by $7\times$ and achieves up to 33% power savings for nearly equal values of $PSNR$.

Section 2 describes the ME algorithm and presents ISR-ANT, the previously proposed error-resilient architecture for energy-efficient motion estimation. Section 3 presents the characterizations of the probability of error due to process variations for the arithmetic units employed in ME implementation and explains the simulation setup. In section 4, we present simulation results showing the impact of process variations and combination of both process variations and VOS on the $PSNR$ using the conventional and the ISR-ANT architectures.

2. PRELIMINARIES

In this section, we present preliminaries of ME. We first introduce the ME algorithm and then demonstrate the application of ANT resulting in the error-resilient ISR-ANT architecture.

2.1. The Three Step Search (TSS) Algorithm

An ME algorithm reduces temporal redundancy between consecutive video frames. In block matching ME algorithms, the current video frame is partitioned into non-overlapping macroblocks of size N pixels by N pixels. For each macroblock in the current frame, the ME algorithm efficiently searches for the best matching macroblock in the previous frame.

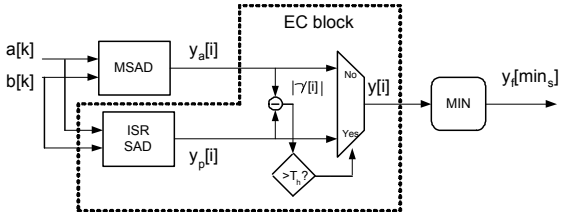

Fig. 3. The ISR-ANT based ME architecture.

There are numerous algorithms for efficient search [1] since the ME algorithm is not standardized. We select an algorithm that is suitable for VLSI implementation for energy-efficiency purposes. The three step search (TSS) algorithm [10] is a commonly employed sub-optimal block matching algorithm because of the simplicity of its implementation, robustness and near optimal performance. In this paper, we choose the TSS algorithm to demonstrate the effectiveness of the proposed ANT technique. Note that the proposed ANT technique can be applied to any other block matching algorithm.

In the TSS algorithm (see Fig. 2(a)), an initial step size Δ, typically equal to half of the search window size is chosen. Next, nine candidate macroblocks $\mathbf{M}[1:9]$ with their center locations as shown in Fig. 2(a), are chosen from the previous frame for comparison. Eight of these candidate macroblocks have their centers at a distance of $\pm\Delta$ in the x and y direction from the current macroblock. The ninth macroblock is at the same location as the current macroblock.

The sum of absolute differences (SAD) for each of the nine macroblocks are calculated by the main SAD (**MSAD**) block (see Fig. 2(b)) by summing up the absolute difference between the corresponding pixels in the candidate macroblocks and the current macroblock. The output of the **MSAD** block are the nine candidate SAD values denoted by $y_o[i]$ $(1 \leq i \leq 9)$, where,

$$y_o[i] = \sum_{k=1}^{N \times N} |a_i[k] - b_i[k]|, \qquad \text{for } 1 \leq i \leq 9 \qquad (1)$$

The index corresponding to the best match is obtained as,

$$y_o[min_o] = min\{y_o[1], y_o[2], ..., y_o[9]\}$$
$$min_o = argmin\{y_o[1], y_o[2], ..., y_o[9]\} \qquad (2)$$

The motion vector is the vector difference between $\mathbf{M}[min_o]$ and the current block. Next, Δ is halved and the center of the search window is moved to coincide with that of $\mathbf{M}[min_o]$. Previous steps are repeated till the Δ becomes less than 1. In the block level implementation of TSS in Fig. 2(b), the **MSAD** block calculates the SAD in (1) while the **MIN** block determines min_o using (2).

2.2. Input Subsampled Replica (ISR) ANT

In this subsection, we describe the error-tolerant ME architecture referred to as the ISR-ANT architecture [9]. In a generic ANT-based system, a main block is assumed to make intermittent errors due to timing violations which are corrected by an error-control block (**EC**). The **EC** block includes an estimator and a decision block. We propose the following ME architecture based on the concept of ANT to generate ISR-ANT as shown in Fig. 3.

1. We employ an estimator based on input subsampling, where an estimate of the **MSAD** output is calculated by employing an **ISR-SAD** block which subsamples the input streams $a[k]$ and $b[k]$ by a factor of m as shown below,

$$y_p[i] = m \times \sum_{k=1}^{\lfloor N^2/m \rfloor} |a_i[mk] - b_i[mk]| \qquad (3)$$

   Let $e_p[i]$ denote the SAD estimation error defined as follows:

$$e_p[i] = y_p[i] - y_o[i] \qquad (4)$$

   Note that **ISR-SAD** block will consume lower power than the **MSAD** block and can be made to operate error-free because it can operate with a lower clock frequency and performs fewer computations.

2. We modify the decision block as follows. We detect and correct errors at the output of the **MSAD** block.

Note, the **ISR-SAD** output $y_p[i]$ is an estimate of the error-free sum $y_o[i]$ for $1 \leq i \leq 9$. Hence, a threshold T_h can be chosen in such a way that $max(|e_p[i]|) < T_h$. Let $\gamma[i]$ denote the difference between the actual (potentially erroneous) **MSAD** output $y_a[i]$ and **ISR-SAD** output $y_p[i]$, i.e.,

$$\gamma[i] = y_a[i] - y_p[i] \qquad (5)$$

An error is declared if $|\gamma[i]| > T_h$. The decision block employs the **ISR-SAD** output $y_p[i]$ as input to the **MIN** block if an error is detected. If there is no error, the **MSAD** output $y_a[i]$ is employed as input to the **MIN** block.

ISR-ANT works well under the following assumptions:

1. The magnitude of error in **MSAD** block output is large. This makes it easy to detect errors.

2. The **ISR-SAD** and the decision blocks are error-free.

Both assumptions are easily met in practice. This is because the errors due to timing violations occur in the most significant bits (MSBs) due to least-significant bit (LSB) first nature of computation in **MSAD**. As a result, the magnitude of the error in **MSAD** block output is large. The **ISR-SAD** block has only N/m inputs to process as compared to N inputs for the **MSAD** block. Hence, it is able to operate in an error-free manner.

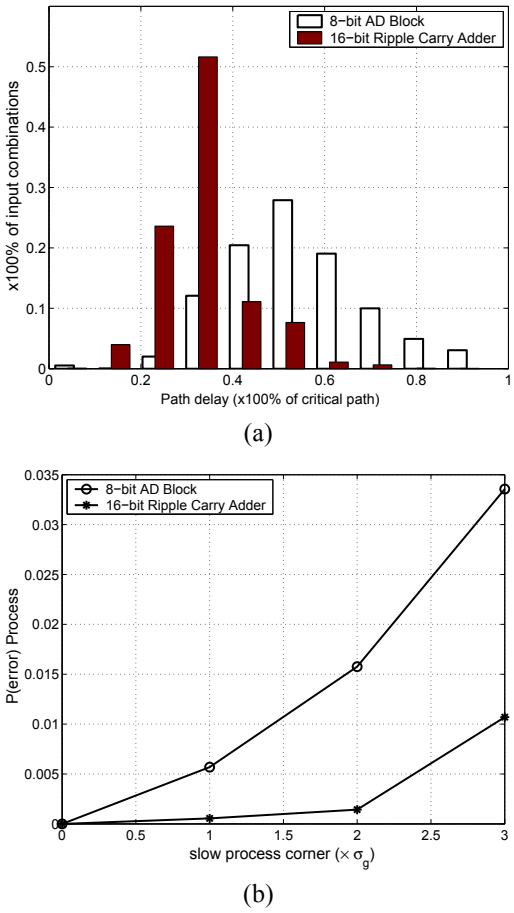

Fig. 4. Error characterization of **AD** block and a ripple carry adder: (a) path delay distribution and (b) probability of process variation error.

3. PROCESS VARIATIONS SIMULATION SETUP

Process variations are classified as die-to-die (D2D) and within-die (WID) variations. D2D variations are caused by differences in process conditions (resist thickness, aberrations in the stepper lens and others) experienced by chips on different wafers in different lots. They modify the device properties (V_t, oxide thickness, conductance and others) for all the devices on the chip in the same way. The standard deviation of the gate delay due to D2D process variations is denoted as σ_g. WID variations result in differences in device parameters for two instances of the same device on the same chip. WID variations are caused by geometric variation due to different layout conditions (nested vs. isolated, vertical vs. horizontal) and mismatch due to the placement of dopant atoms in the device channel. The mean and the standard deviation of the gate delay due to WID process variations are denoted as μ_l and σ_l respectively.

Process variations significantly affect circuit delay. The

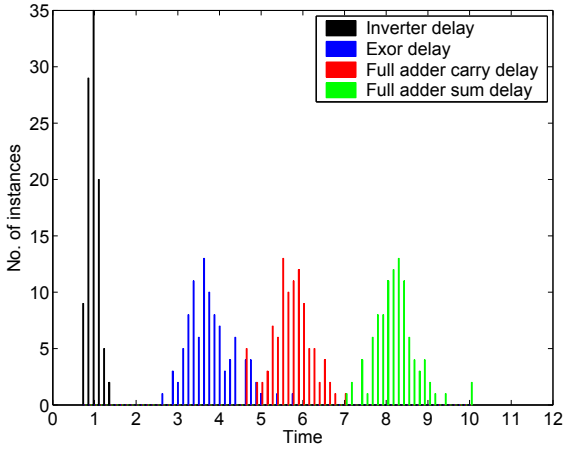

Fig. 5. Delay distributions of various gates for a $3\sigma_g$ slow die with WID variations normalized to the mean inverter delay.

Table 1. Characteristics of normalized delay distributions of various gates at the $3\sigma_g$ slow corner due to WID variations.

Gate		Supply Voltage			
		1.35 V	1.2 V	1.05 V	0.9 V
Inverter	μ_l	1.0	1.0	1.0	1.0
	σ_l	0.13	0.14	0.14	0.17
	σ_l/μ_l	0.13	0.14	0.14	0.17
Exor	μ_l	3.50	3.80	4.19	4.23
	σ_l	0.47	0.57	0.67	0.74
	σ_l/μ_l	0.13	0.15	0.16	0.17
Full adder Carry	μ_l	5.52	5.72	6.11	6.81
	σ_l	0.45	0.48	0.58	0.69
	σ_l/μ_l	0.08	0.08	0.09	0.10
Full adder Sum	μ_l	7.96	8.20	9.47	12.91
	σ_l	0.50	0.51	0.66	0.91
	σ_l/μ_l	0.06	0.06	0.07	0.07

impact of these variations are captured through measurements [11], which are then employed to generate statistical process models [12]. In this section, we first characterize the error probabilities for the arithmetic units employed in the ME implementation due to D2D process variations. Then we discuss the impact of WID process variations on circuit delay (μ_l and σ_l) using statistical process model for a $3\sigma_g$ slow die. Next, we describe the simulation setup employed for simulating the effect of process variation induced timing errors on the performance of ME algorithm.

3.1. Error Characterization of Arithmetic Units

The **MSAD** block employs an absolute difference (**AD**) block followed by an accumulator (see Fig. 2(b)). These arithmetic units are based on least significant bit (LSB) first computation. Therefore, critical path timing violations due to VOS or process variation will result in errors in the most significant bits (MSBs). These errors are large in magnitude and hence severely degrade the performance in terms of $PSNR$.

The probability of timing errors depends on the path delay distribution of the architecture and the probability distribution of the inputs. The delay distributions of an 8-bit **AD** block and a 16-bit ripple carry adder are shown in Fig. 4(a). The probability of error for the **AD** block and the ripple-carry adder are shown in Fig. 4(b) for uniformly distributed inputs at different process corners due to D2D variations. The x-axis shows the instance of the slow process due to D2D process variation in terms of σ_g. The supply voltage is kept constant such that there are no errors at the nominal process corner. We observe that the **AD** block and the ripple carry adder exhibit errors for 3% and 1% of the inputs, respectively, at $3\sigma_g$ slow process corner. This is because the **AD** block has greater number of paths with delays close to the critical path delay than the ripple carry adder as shown in Fig. 4(a). Therefore, the probability of error due to process variations is higher for the **AD** block than for the ripple carry adder. ISR-ANT architecture is shown to be very effective in correcting for these errors.

3.2. Simulation Setup

We characterized the delay distribution of basic gates such as an inverter, exor, and a full adder due to WID variations at various values of the supply and body bias voltage combinations (V_{dd},V_b). Monte Carlo simulations using statistical model files were employed for this purpose. Fig. 5 shows the normalized delay distributions resulting from the presence of WID variations at the $3\sigma_g$ slow corner with $V_{dd} = 1.2V, V_b = 0V$. Table 1 shows the mean and the standard deviation of the normalized delay for $V_b = 0$, from which we observe that the relative delay variations (σ_l/μ_l) decreases as we move from the simplest gate (inverter) to a complex gate (full adder). The relative delay variations was also found to decrease with an increase in the supply voltage V_{dd}.

Next, we sample the distribution in Fig. 5 to obtain the gate delays of a gate level implementation of the conventional and the ISR-ANT architectures at the $3\sigma_g$ slow process corner. This process is repeated 30 times in order to obtain 30 instances of the two architectures. We simulate the conventional and the ISR-ANT architectures using an HDL simulator which operates at the gate-level to determine the output motion vectors for the three clips. We predicted the current frame from these motion vectors and the previous frame to obtain the $PSNR$. The $PSNR$ is calculated as

$$PSNR(dB) = 20 \times log_{10}\frac{255}{\sigma_r} \quad (6)$$

where σ_r^2 is the prediction noise power. We set the desired $PSNR$ requirement to be $0.5dB$ less than the $PSNR$ of the error-free conventional architecture.

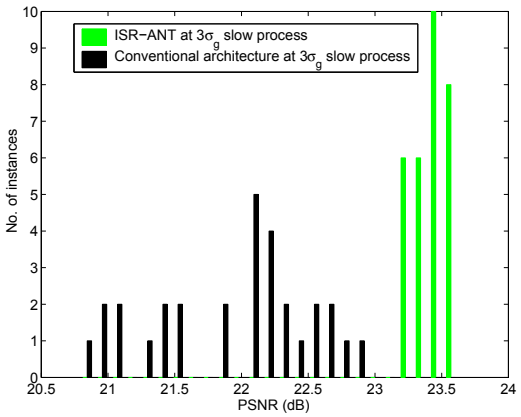

Fig. 6. $PSNR$ distribution using conventional and ISR-ANT architecture on a $3\sigma_g$ slow die due to WID variations for mobile calendar clip.

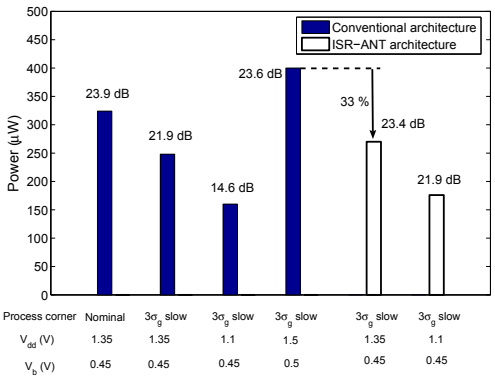

Fig. 7. Power performance trade-off for mobile calendar clip.

Table 2. Characteristics of $PSNR$ distributions for conventional architecture on a $3\sigma_g$ slow die due to WID variations.

Clip	flower garden	mobile calendar	football
μ_c (dB)	21.31	21.95	23.17
σ_c (dB)	0.49	0.59	0.39
σ_c/μ_c	0.022	0.026	0.017

4. SIMULATION RESULTS

In this section, we present simulation results showing the impact of delay variations on the $PSNR$ of ME using conventional and ISR-ANT architectures. Three different video clips are evaluated: flower garden (low motion), mobile calendar (medium motion) and football (high motion).

4.1. Impact of Process Variations on $PSNR$

Each of the 30 instances of either the conventional or ISR-ANT architecture will result in a different $PSNR$. This is because the path delay distribution and hence the timing violations will be different for each instance. Thus, the $PSNR$ is a random variable and it will have a distribution. The mean μ_c and the standard deviation σ_c of the $PSNR$ for the conventional architecture are tabulated in Table 2. We observe that the μ_c drops by approximately $2dB$ for flower garden and mobile calendar clips and $1dB$ for the football clip when compared to the error-free implementation. This drop is quite significant and results in a noticeable loss in image quality.

Next, we obtain the $PSNR$ distribution for the ISR-ANT architecture for different values of the subsampling ratio $m = 3, 4, 5$ and the **ISR-SAD** input precision $b = 8, 6, 5$. The representative distributions of the $PSNR$ for the conventional architecture and the ISR-ANT architecture ($m = 4$, $b = 8$) are shown in Fig. 6. The mean μ_i and the standard deviation σ_i of the output $PSNR$ are tabulated in Table 3. From Table 3, we can see that the improvement in the mean μ_i is significant as we increase estimator complexity from $m = 5$ to $m = 4$, but provides diminishing returns as the estimator complexity increases from $m = 4$ to $m = 3$. We also note that the performance of ISR-ANT decreases as the precision of ISR-SAD block is reduced from 8 to 5. Comparing Tables 2 and 3, we observe that the mean $PSNR$ increases but its standard deviation decreases when we use ISR-ANT architecture instead of the conventional architecture. The relative variation (σ_i/μ_i) in $PSNR$ is reduced by $7\times$ for the flower garden, $5\times$ for the mobile calendar and $4\times$ for the football clip. Since we want to limit the $PSNR$ loss to $0.5dB$, we choose $m = 4, b = 8$ in the following discussion.

4.2. Power vs. Performance Trade-off

In this subsection, we present the power overhead of using ISR-ANT and its impact on the $PSNR$ for a representative clip (mobile calendar) in the presence of VOS and process variation induced errors. We compare power consumption of the ISR-ANT architecture with the conventional architecture. We simulate the transistor level netlist of the conventional architecture and the ISR-ANT architecture using HSPICE with random input vectors to obtain the power consumption for both the architectures at different supply voltage levels. We evaluate the mean $PSNR$ employing the procedure described in the previous subsection.

We show a plot of power consumption of the two architectures along with the mean $PSNR$ for the mobile calendar clip in Fig. 7. The first bar shows the power consumed by the conventional architecture operating under error-free conditions on a nominal process die. The V_{dd} and V_b are adjusted using the mean delay characterization results. We note that the prediction $PSNR$ is $23.9dB$ for power consumption of $324\mu W$ at $V_{dd} = 1.35V, V_b = 0.45V$ at the nominal process corner. The mean performance of the conventional architec-

Table 3. Characteristics of $PSNR$ distributions for ISR-ANT architecture on a $3\sigma_g$ slow die due to WID variations.

		flower garden			mobile calendar			football		
		b=8	b=6	b=5	b=8	b=6	b=5	b=8	b=6	b=5
m=5	μ_i (dB)	22.84	22.75	21.75	22.29	22.06	21.85	22.92	22.90	22.76
	σ_i (dB)	0.27	0.28	0.67	0.24	0.25	0.51	0.21	0.21	0.31
	σ_i/μ_i	0.011	0.012	0.030	0.010	0.011	0.023	0.009	0.009	0.014
m=4	μ_i (dB)	23.12	22.80	22.19	23.42	23.33	22.07	23.58	23.48	23.19
	σ_i (dB)	0.08	0.14	0.36	0.11	0.19	0.49	0.26	0.29	0.38
	σ_i/μ_i	0.003	0.006	0.016	0.005	0.008	0.022	0.011	0.012	0.016
m=3	μ_i (dB)	23.18	23.10	22.29	23.70	23.51	23.05	24.14	24.08	23.51
	σ_i (dB)	0.01	0.16	0.4	0.02	0.2	0.3	0.01	0.04	0.14
	σ_i/μ_i	0.0004	0.007	0.018	0.0008	0.0085	0.013	0.0004	0.0017	0.006

ture decreases to $21.9dB$ for power consumption of $248\mu W$ at $V_{dd} = 1.35V, V_b = 0.45V$ at the $3\sigma_g$ slow corner. If the supply voltage is reduced to $V_{dd} = 1.1V, , V_b = 0.45V$, the errors occur from process variations as well as VOS. Hence, the mean performance degrades to $14.6dB$ while consuming $160\mu W$ of power. If we apply the conventional ABB and ASV to reduce the gate delays and correct the timing errors then the power consumption increases to $400\mu W$ at $V_{dd} = 1.5V, V_b = 0.5V$ while achieving a mean $PSNR$ of $23.6dB$. The ISR-ANT architecture, at $3\sigma_g$ slow process corner and $V_{dd} = 1.35V, V_b = 0.45V$, consumes $270\mu W$ with a $PSNR$ of $23.4dB$. Thus, at the same slow process corner, the $PSNR$ of ISR-ANT is comparable to the conventional architecture while consuming 33% lower power than the conventional architecture. When process variations and VOS occur simultaneously, the ISR-ANT improves the $PSNR$ from $14.6dB$ to $21.9dB$ while consuming an additional 10% power. Thus, ISR-ANT technique is able to trade-off power and performance effectively with robust $PSNR$ performance.

5. CONCLUSIONS

In this paper, we studied the performance of ISR-ANT architecture based on the principle of error-resilience in the presence of process variation errors. The work presented in this paper falls in the category of communication inspired low-power design techniques [8] that favors the notion of error-correction rather than error-avoidance. Such error-resiliency based techniques can be applied to other power hungry 4G media communication kernels such as discrete cosine transform (DCT) and forward error-control (FEC) decoders. Studying the effectiveness of these techniques at the video encoder system level is also of great interest.

6. ACKNOWLEDGMENT

The authors acknowledge the support of the MARCO Gigascale Systems Research Center and Texas Instruments.

7. REFERENCES

[1] P. Kuhn, *Algorithms, complexity analysis and VLSI architectures for MPEG-4 motion estimation*, Kluwer Academic Publishers, Boston 1999.

[2] R. Richmond II, et. al., "A low-power motion estimation block for low bit-rate wireless video," in *ISLPED*, 2001.

[3] X. Tang, et. al., "Intrinsic MOSFET parameter fluctuations due to random dopant placement," *IEEE Trans. on VLSI Systems*, vol. 5 pp. 369-376, December 1997.

[4] J. W. Tschanz, et. al., "Adaptive Body Bias for Reducing Impact of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage," *IEEE Journal of Solid-state Circuits*, Vol. 37, Nov. 2002.

[5] T. Chen, and S. Naffziger, "Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under the presence of process variation," *IEEE Trans. VLSI*, vol. 11, Oct. 2003.

[6] http://public.itrs.net.

[7] S. Borkar et. al., "Parameter variations and impact on circuits and microarchitecture," in *Proc. of DAC*, 2003.

[8] R. Hegde, and N. R. Shanbhag, "Soft digital signal processing," *IEEE Trans. on VLSI*, vol. 9 Dec. 2001.

[9] G. Varatkar, and N. R. Shanbhag, "Energy-efficient motion estimation using error-tolerance," in *Proc. of ISLPED*, October 2006.

[10] T. Koga, "Motion compensated interframe coding for video conferencing," in *Proc. NTC*, 1981, Ch. 9.6.1-9.6.5.

[11] K. A. Bowman, et. al., "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE Journal of Solid-state Circuits*, Vol. 37, Feb. 2002.

[12] *IBM process design manual*, May 2004.