Energy-efficient Motion Estimation using Error-Tolerance

Girish V. Varatkar and Naresh R. Shanbhag
Coordinated Science Laboratory, University of Illinois at Urbana-Champaign
1308 W Main St., Urbana, IL 61801
[varatkar, shanbhag]@uiuc.edu

ABSTRACT

Presented is an energy-efficient motion estimation architecture using error-tolerance. The technique employs overscaling of the supply voltage (voltage overscaling (VOS)) to reduce power at the expense of timing errors, which are then corrected using algorithmic noise-tolerance (ANT) techniques. Referred to as input subsampled replica ANT (ISR-ANT), the proposed technique incorporates an input subsampled replica of the main sum of absolute difference (MSAD) block for obtaining the motion vectors in the presence of errors induced by VOS. Simulations show that the proposed technique can save up to 60% power over an optimal error-free present day system in a 130*nm* CMOS technology. Power savings increase to 79% in a 45*nm* predictive process technology.

Categories and Subject Descriptors: B.8.2 [Performance and reliability]: Performance Analysis and Design Aids

General Terms: Algorithms, Design, Reliability

Keywords: Low-power, noise-tolerance

1. INTRODUCTION

Next generation wireless multimedia communications standards such as digital video broadcast (DVB) [1], fourth generation (4G) mobile systems [2] 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. 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 cycles [3]. Therefore, low-power motion-estimation architectures and implementations are of great interest.

Low-power motion estimation is a well-studied subject [4]-[6]. However, most, if not all, approaches assume error-free computation. The proposed work relaxes this assumption in order to push the boundaries of achievable energy-efficiency. In particular, most algorithmic low-power approaches focus on heuristics to reduce the number of macroblocks processed per motion vector [3]. These include employing a simpler distance criterion [8], fixed or adaptive search area [9] and temporal or spatial prediction [10]. Several VLSI architectures are proposed for ME with various trade-offs between gate-count, I/O bandwidth and throughput [3][11][12][13]. A typical motion estimation accelerator consists of a RAM for search area and current block, an address generation unit, a datapath consisting of a processor element and a control unit. The datapath power consumption is found to be 75% of the total ME power consumption for full search motion estimation algorithm and 50% of the total ME power consumption for three step search algorithm [7]. Therefore, it is important to reduce the datapath power consumption. Scaling of supply (V_{dd}) and threshold voltage (V_t), has been commonly employed to reduce the total datapath power consumption [14]-[15]. The benefits of conventional voltage scaling are limited by the (V_{dd},V_t) combination at which the worst case critical path delay is equal to the clock period [15].

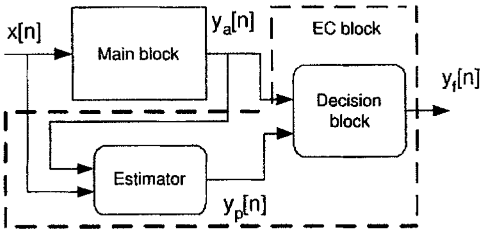

Figure 1: An ANT-based system.

In this paper, we present a novel low-power ME architecture that is based on the concept of algorithmic noise-tolerance (ANT) [16]. In ANT (see Fig. 1), a main block is assumed to make intermittent errors which are then corrected by an error-control block (**EC**). The **EC** block includes an estimator and a decision block. The error mechanism can vary but if power reduction is being targeted then the errors are tailored to arise from voltage overscaling (VOS) [16]. In VOS, the supply voltage is reduced beyond $V_{dd-crit}$, i.e.,

$$V_{dd} = K_{vos} V_{dd-crit}, \qquad (1)$$

where $0 < K_{vos} \leq 1$ and $V_{dd-crit}$ is the supply voltage below which timing violations occur. These violations are referred to as VOS errors which are then corrected by employing ANT techniques. Thus, the combination of VOS and ANT can reduce power beyond that achievable by conventional voltage scaling alone.

ISLPED'06, October 4–6, 2006, Tegernsee, Germany.
Copyright 2006 ACM 1-59593-462-6/06/0010 ...$5.00.

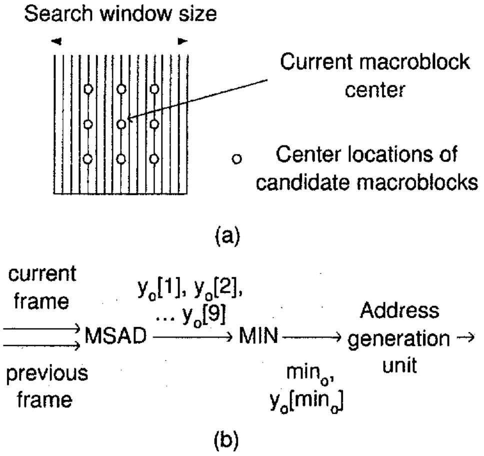

Figure 2: The three step search (TSS) algorithm: (a) search window, and (b) block level implementation.

Many ANT techniques exist. These include, the reduced precision replica (RPR) ANT [17] technique where the estimator is a reduced precision replica of the main block. In prediction-based ANT [16], the estimator is a predictor that exploits the correlation in the output of the main block. As VOS errors are input-dependent, the adaptive error-cancellation technique [18] employs an error estimator to estimate and cancel VOS errors at the main block output.

In this paper, we present a novel ANT technique for correcting VOS errors in ME architectures and study the achievable power savings. Simulations using an IBM 130nm CMOS process show that up to 60% power savings can be achieved over an optimal error-free architecture. Power savings increase to 79% in a 45nm predictive process technology [19][20].

Section 2 of the paper describes the ME algorithm and a straightforward application of ANT to ME referred to as motion-vector replica ANT (MVR-ANT). In section 3, we present our main contribution: a new technique referred to as input subsampled replica ANT (ISR-ANT) for energy-efficient motion estimation. In section 4, we present simulation results for ISR-ANT datapath designed using IBM 130nm process technology and using 45nm predictive technology models.

2. PRELIMINARIES

In this section, we present preliminaries of ME and ANT. This is done by first introducing the three-step search ME algorithm and then demonstrating a straightforward but ineffective application of ANT resulting in the MVR-ANT. The latter will then be modified to form the proposed ISR-ANT in section 3.

2.1 The Three Step Search (TSS) Algorithm

A motion estimation algorithm reduces temporal redundancy between consecutive video frames. In block matching motion estimation algorithms, the current video frame is partitioned into non-overlapping macroblocks of size sixteen pixels by sixteen pixels. For each macroblock in the current frame, the motion estimation algorithm searches for the best matching macroblock in the previous frame. There exist hundreds of algorithms for efficient search [3]-[13] since the ME algorithm is not standardized. For energy-efficiency purpose, we select an algorithm that is suitable for VLSI implementation. The full-search block matching algorithm is the most optimal and the most suitable algorithm for VLSI implementation due to regularity in computation. However, it demands a huge amount of computation.

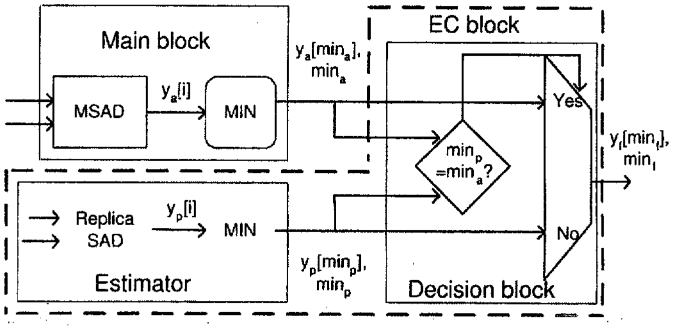

Figure 3: The MVR ANT architecture.

The TSS algorithm is a commonly employed sub-optimal block matching algorithm [21] because of its simplicity, reduced amount of computation compared to full search, and near optimal performance. In this paper, we choose TSS algorithm to demonstrate effectiveness of ANT technique. Our proposed ANT technique does not depend upon the choice of block matching algorithm used and it can be extended to other block matching algorithms. In TSS algorithm, an initial step size Δ, typically equal to half of the search window size is chosen. The location of the center of the current macroblock inside the search window is shown in Fig. 2(a). 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 **MSAD** block by summing up the absolute difference between the corresponding pixels in the candidate macroblocks and the current macroblock. Let **MSAD** block have input pixel streams denoted by $a[k]$ and $b[k]$ from the current macroblock and the candidate macroblock, respectively. The output of **MSAD** are the nine candidate SAD values denoted by $y_o[i]$ for $1 \leq i \leq 9$, where,

$$y_o[i] = \sum_{k=1}^{256} |a[k] - b[k]| \tag{2}$$

for $1 \leq i \leq 9$. The index corresponding to the best match is obtained as,

$$\begin{aligned} y_o[min_o] &= min\{y_o[1], y_o[2], ..., y_o[9]\} \\ min_o &= argmin\{y_o[1], y_o[2], ..., y_o[9]\} \end{aligned} \tag{3}$$

The motion vector is the vector difference between $M[min_o]$ and the current block. Next, $\pm\Delta$ is halved and the center of the search window is moved to coincide with that of $M[min_o]$. Previous steps are repeated till the Δ becomes less than 1. A block level implementation of TSS is shown in Fig. 2(b). The **MSAD** block calculates the SAD in Eq. 2 while the **MIN** block determines min_o using Eq. 3.

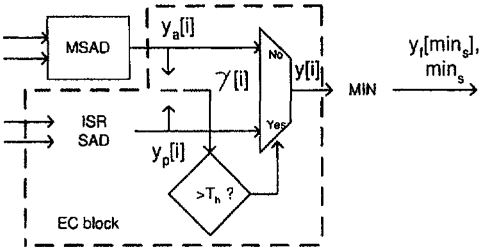

Figure 4: The ISR-ANT architecture.

2.2 Motion Vector Replica (MVR) ANT

A straightforward application of the ANT framework (see Fig. 1) results in the motion vector replica (MVR) ANT. A MVR ANT-based ME (see Fig. 3) has a main block and an error control block (**EC**). The address generation block is not shown. The main block is a complete ME engine that includes the **MSAD** and **MIN** blocks. The main block is made energy-efficient via VOS but makes intermittent errors. These VOS errors degrade the output peak signal-to-noise ratio ($PSNR$) if left uncorrected. Let the error-free **MSAD** block outputs be denoted as $y_o[i]$ for $i = 1, .., 9$. Under VOS, the **MSAD** output denoted as $y_a[i]$ is given by

$$y_a[i] = y_o[i] + \eta[i] \tag{4}$$

where $\eta[i]$ is the VOS error. Next, we define the main block output min_a as follows.

$$\begin{aligned} y_a[min_a] &= min\{y_a[1], y_a[2], ..., y_a[9]\} \\ min_a &= argmin\{y_a[1], y_a[2], ..., y_a[9]\} \end{aligned} \tag{5}$$

The **EC** block has an estimator and a decision block. The estimator estimates the correct motion vector and is designed to have low complexity and hence error-free operation, compared to the main block. This means that the estimator output will not be as accurate as the correct main block output. Therefore, the estimator block operates in an error-free albeit inaccurate manner. If the main block and the estimator outputs differ, then the decision block employs the estimator output as the final corrected output $y_f[min_f]$ as shown in Fig. 3.

A simple **EC** block is one whose estimator is a reduced precision replica of the main block [17]. For example, the **MSAD** block can have 8-bit pixels as input while the replica **SAD** block can employ reduced input bit precision. In fact, it is known [22] that the average $PSNR$ degrades by less than $0.5dB$ if 3-bit inputs are employed for computing SAD. If the **MSAD** block employs 3-bit inputs, then the replica **SAD** must operate with less than 3-bit input pixel values resulting in very inaccurate estimates of SAD. Thus, the MVR-ANT is not very effective in power reduction.

3. INPUT SUBSAMPLED REPLICA (ISR) ANT ARCHITECTURE

In this section, we describe the main contribution of this paper referred to as the input subsampled replica (ISR) ANT-based ME architecture. We make the following modifications to the MVR ANT **EC** block to generate ISR-ANT as shown in Fig. 4.

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.

$$y_p[i] = m \times \sum_{k=1}^{\lfloor 256/m \rfloor} |a[mk] - b[mk]| \tag{6}$$

Let $e_p[i]$ denote the SAD estimation error.

$$e_p[i] = y_p[i] - y_o[i] \tag{7}$$

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

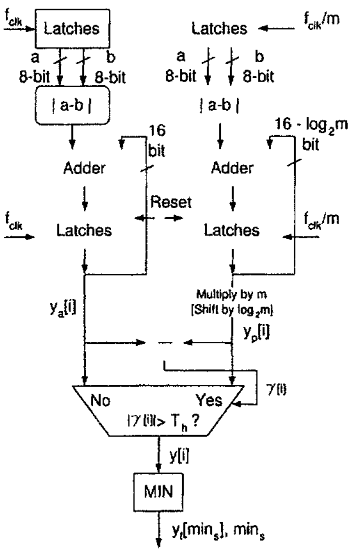

Figure 5: The ISR-ANT datapath.

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 =max$|e_p[i]|$ can be chosen in such a way that $|e_p[i]| < T_h$. Let $\gamma[i]$ denote the difference between the **MSAD** output $y_a[i]$ and **ISR-SAD** output $y_p[i]$, i.e.,

$$\gamma[i] = y_a[i] - y_p[i] \tag{8}$$

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 error is detected. If there is no error, the **MSAD** output $y_a[i]$ is employed as input to the **MIN** block.

The ISR-ANT algorithm is described by the following steps.

1. An initial step size Δ is chosen. Eight blocks at a distance of $\pm\Delta$ from the center (around the center block) are picked for SAD computation and comparison. The **MSAD** and the **ISR-SAD** blocks, each calculate the nine candidate SADs.
2. An error is declared if $|\gamma[i]| > T_h$ where T_h =max$|e_p[i]|$.
3. If an error is declared then $y[i] = y_p[i]$ else $y[i] = y_a[i]$.
4. The direction in which the block distortion $y[i]$ is minimum is chosen.

$$\begin{aligned} y_f[min_s] &= min\{y[1], y[2], ..., y[9]\} \\ min_s &= argmin\{y[1], y[2], ..., y[9]\} \end{aligned} \tag{9}$$

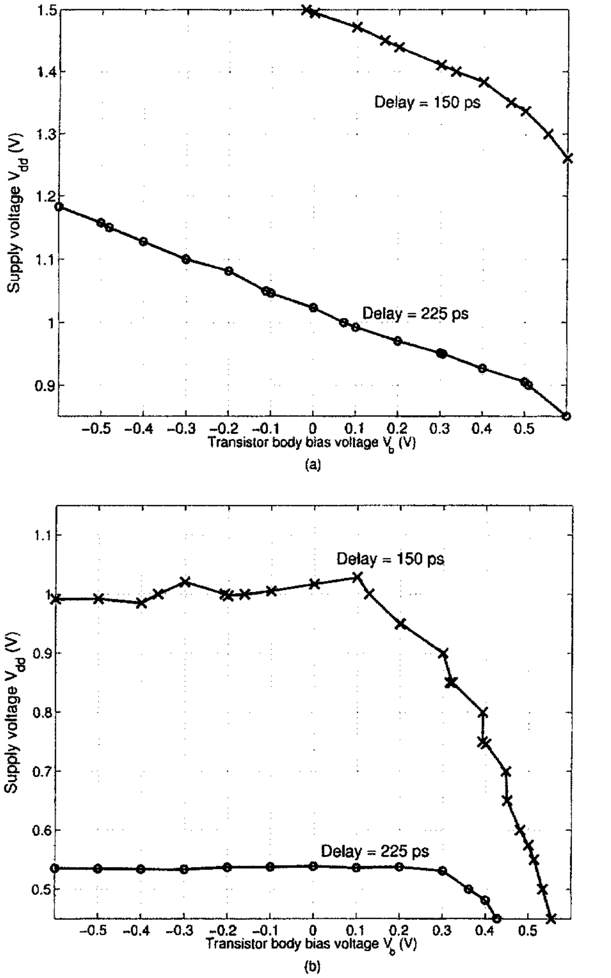

Figure 6: Supply voltage and body bias for full adder delay = $150ps$ and $225ps$ for: (a)$130nm$, and (b) $45nm$ process technology.

5. The step size Δ is halved. The center of the search window is moved to the point with the minimum distortion. Previous steps are repeated till the step size becomes less than 1.

ISR-ANT works well under the following assumptions:

1. The magnitude of VOS error in **MSAD** block output is large so that it is easy to detect errors in the output.

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

The errors due to voltage-overscaling 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 VOS 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.

In ISR-ANT architecture as shown in Fig. 5, the **MSAD** block consists of a modulus block computing the absolute difference between 8-bit luminance values. These absolute differences are accumulated by an 16-bit adder whose outputs are latched at a frequency f_{clk}. The **ISR-SAD** block uses pixel luminance values which are subsampled by a factor of m. Hence, the adder in **ISR-SAD** has $16 - log_2 m$ bits and its outputs are latched at f_{clk}/m. After 256 cycles of the frequency f_{clk} clock, the absolute value of $\gamma[i]$ is compared with the threshold T_h and the multiplexer output $y[i]$ is fed to the MIN block.

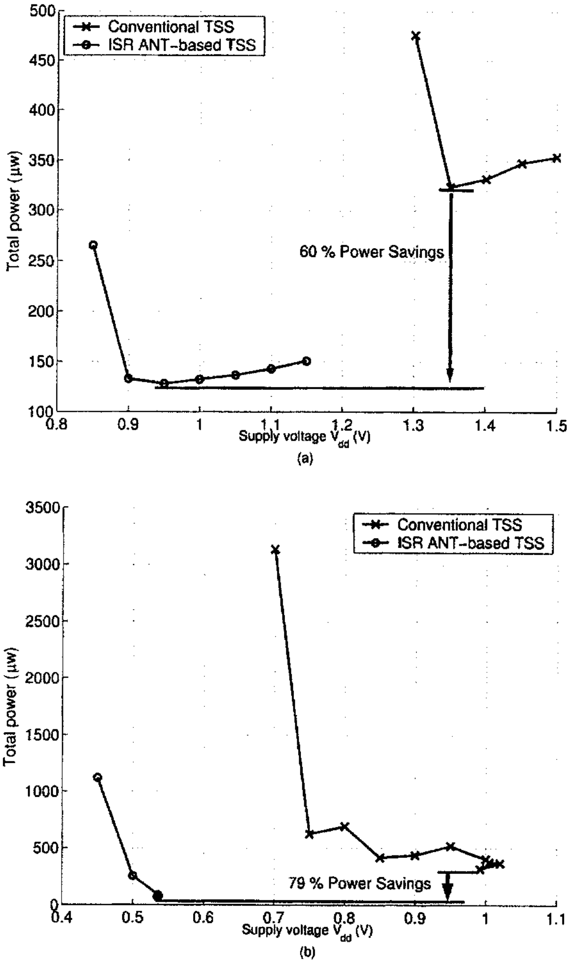

Figure 7: Power consumption for conventional TSS and ISR-ANT-based TSS for: (a)$130nm$, and (b) $45nm$ process technology.

4. SIMULATION RESULTS AND DISCUSSION

In this section, we compare the power vs. performance trade-offs between conventional error-free architecture and ISR-ANT in an IBM $130nm$ CMOS process and $45nm$ predictive technology models [19][20]. Three different video clips are evaluated: flower garden (low motion), mobile calendar (medium motion) and football (high motion).

4.1 Simulation Set-up

We first determined the system level throughput requirements for motion estimation in real time encoding of MPEG-II main profile at main level [24] to be a CIF frame size of 288 by 352 pixels at the rate of 30 frames/s. Next, we simulated both architectures with ripple carry adders using an HDL simulator. This was done to determine the maximum delay

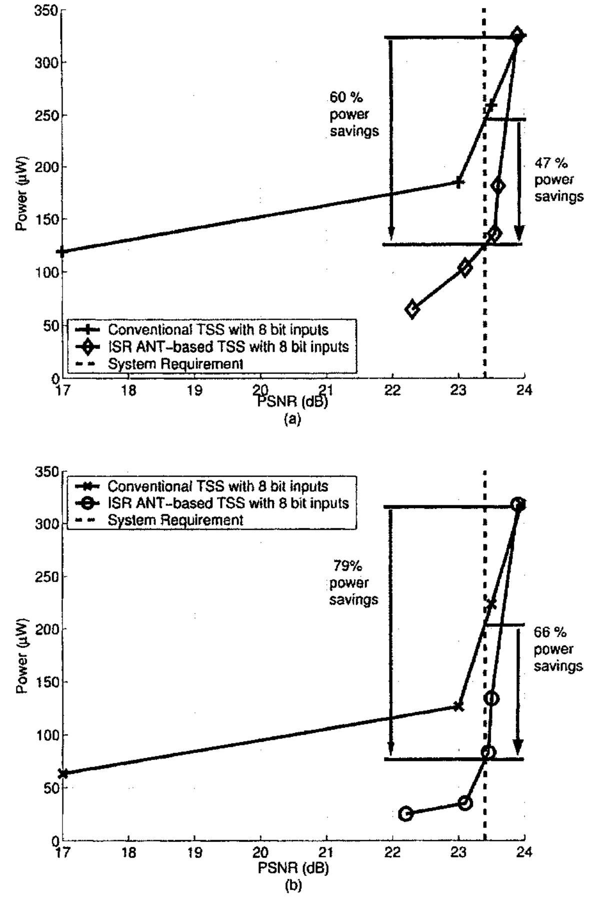

Figure 8: Power vs. *PSNR* plot for 8-bit input in: (a)130*nm*, and (b) 45*nm* process technology.

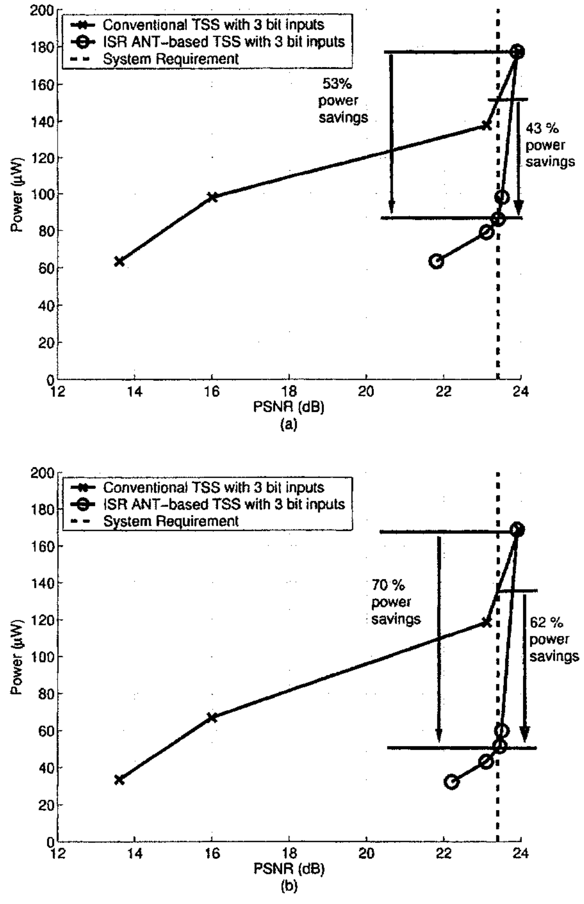

Figure 9: Power vs. *PSNR* plot for 3-bit input in: (a)130*nm*, and (b) 45*nm* process technology.

of a 1-bit adder (T_{FA}) necessary to support this system level throughput. The conventional architecture was found to require a $T_{FA} \leq 150ps$ for error-free operation. We simulated ISR-ANT with various subsampling ratios and determined that the system performance requirement as defined in 4.3 is met for $m \leq 4$ for the ISR-ANT. Therefore, the subsampling ratio is set to $m = 4$ for all simulations. With $m = 4$, we determined that T_{FA} needs to be less than $225ps$ in order for the **EC** block to correct the resulting errors effectively. For $T_{FA} \geq 225ps$, the VOS errors in **MSAD** degrade the ISR-ANT performance significantly.

Next, we characterized a full adder with the mirror structure [23] in terms of delay, dynamic power and leakage power consumption. Isodelay curves were obtained by varying the supply and body bias voltage combinations (V_{dd},V_b) via HSPICE in 130*nm* IBM process technology and 45*nm* predictive technology. The adder outputs were loaded with identical mirror full adders to determine the worst case output delay.

4.2 Power vs. Supply and Body Bias Voltage

As mentioned earlier, the error-free conventional architecture and ISR-ANT architecture needs to operate with $T_{FA} \leq 150ps$ and $T_{FA} \leq 225ps$ respectively. Fig. 6(a) shows the (V_{dd},V_b) combinations that result in a constant full-adder delay $T_{FA} = 150ps$ and $T_{FA} = 225ps$ in 130*nm* IBM technology. Similarly, Fig. 6(b) shows the isodelay plots in the 45*nm* process technology. Similar isodelay curves are derived for intermediate delay values. These plots are useful in determining the power-optimum (V_{dd},V_b) combination for the conventional and ISR-ANT architectures.

We simulate the conventional architecture and the ISR-ANT architecture using HSPICE to obtain the power consumption for both the architectures operating at the (V_{dd},V_b) combinations obtained from the isodelay curves similar to Fig. 6. Thus we determine the power-optimum (V_{dd},V_b) combination for the conventional and ISR-ANT architectures. We plot the power dissipation of the two architectures in Fig. 7.

In the 130*nm* IBM process technology (see Fig. 7(a)), the conventional architecture achieves a minimum power consumption of $324\mu W$ at ($V_{dd-crit} = 1.35V$, $V_b = 0.5V$), while the ISR-ANT architecture achieves a minimum power consumption of $128\mu W$ at ($V_{dd} = 0.95V$, $V_b = 0.3V$). The **EC** block operates at lower voltage, $V_{dd-EC} = 0.6V$. Thus, ISR-ANT achieves power-reduction of 60% over the conventional system.

The results for 45*nm* technology are shown in Fig. 7(b). We see that the conventional architecture achieves a minimum power consumption of $318\mu W$ at ($V_{dd-crit} = 1.0V$, $V_b = -0.6V$), whereas ISR-ANT achieves a minimum power consumption of $69\mu W$ at ($V_{dd} = 0.53V$, $V_b = -0.6V$). Here, the **EC** block operates at a voltage of $V_{dd-EC} = 0.4V$. Power reduction of 79% is achieved via ISR-ANT.

4.3 Power vs. Performance Trade-off

In this subsection, we compare the power savings for the conventional and ISR-ANT architectures as a function of the *PSNR*. We simulate both architectures using the HDL simulator in order to determine the output motion vectors. We reconstruct the current frame from these motion vectors and the previous frame. Reconstruction error is calculated as the difference between the reconstructed frame and the actual current frame luminance values. The *PSNR* is calculated as

$$PSNR(dB) = 20log_{10}\frac{255}{\sigma_r} \qquad (10)$$

where σ_r^2 is the reconstruction noise power. As mentioned earlier, the *PSNR* calculations were done for three different clips. The flower garden clip has slowly moving garden background. The mobile calendar clip shows objects moving with medium speed. The football clip has fast player movement. We show a plot of power dissipation vs. *PSNR* for mobile calendar clip in Fig. 8 for 8-bit inputs. The plots for other clips are similar and hence are not shown. We set the desired *PSNR* requirement to be $0.5dB$ less than the *PSNR* obtained using the error-free conventional architecture. Plots in Fig. 8(a) and (b) for $130nm$ and $45nm$ process nodes, respectively, are obtained by reducing the supply voltage from $V_{dd-crit}$ where the full-adder delay is $T_{FA} = 150ps$. A power-optimum value for V_b is obtained for each value of V_{dd} from the isodelay curves similar to those in Fig. 6. First, note that *PSNR* of the conventional architecture drops severely as the supply voltage is reduced from $V_{dd-crit}$. The ISR-ANT architecture is seen to be robust to VOS. At a $PSNR = 23.9dB$, we see that 60% and 79% total power savings are achieved in $130nm$ and $45nm$, respectively, with the error-free conventional architecture as a reference. The power savings become 47% and 66% in $130nm$ and $45nm$, respectively, when compared at the desired $PSNR = 23.4dB$.

We can reduce the complexity of the **MSAD** block by reducing its precision from 8-bits to 3-bits. Figure 9 shows the power savings when the input is quantized to 3-bits. At a $PSNR = 23.9dB$, we see that 53% and 70% total power savings are achieved in $130nm$ and $45nm$, respectively, with the error-free conventional architecture as a reference. The power savings become 43% and 62% in $130nm$ and $45nm$, respectively, when compared at the desired $PSNR = 23.4dB$.

We note that if we subsample the inputs of **MSAD** in the conventional architecture by $m = 4$ in order to reduce its complexity, then the *PSNR* degrades from $PSNR = 23.9dB$ to $PSNR = 22.8dB$. This *PSNR* loss is unacceptable and it indicates that ISR-ANT is a unique approach to power reduction.

5. CONCLUSIONS

In this paper we have applied error-tolerance for energy-efficient motion estimation. We minimized the total power by jointly optimizing supply voltage and body bias in $130nm$ and $45nm$ process technologies. The proposed ISR-ANT technique was shown to be robust to VOS errors. The ISR-ANT technique is agnostic to the actual source of errors and thus it would be interesting to study its performance in the presence of random errors such as soft-errors due to particle hits as well as errors due to process variations.

6. REFERENCES

[1] http://www.dvb.org

[2] http://www.4gmf.org

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

[4] M. A. Elgamel, et. al., "A comparative analysis for low power motion estimation VLSI architectures," in *IEEE Workshop on Signal Processing* , pp. 149–158, 2000.

[5] F. Dufaux, et. al., "Motion estimation techniques for digital TV: A review and a new contribution," in *Proc. of the IEEE* , vol. 83, No. 6, pp. 858–876, June 1995.

[6] Po-Chih Tseng, et. al., "Advances in hardware architectures for image and video coding- a survey," in *Proc. of the IEEE*, vol. 93, No. 1, pp. 184–197, Jan. 2005.

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

[8] M. Ghanbari, "The cross-search algorithm for motion estimation," in *IEEE Trans. on Comm.*, vol. 38, No. 7, pp. 950–953, July 1990.

[9] J. Minocha, et. al., "A low power data-adaptive motion estimation algorithm," in *IEEE Workshop on Multimedia Signal Processing*, pp. 685–690, Sept. 1999.

[10] B. Zeng, et. al., "Optimization of fast block motion estimation algorithms," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, No. 6, pp. 833–844, Dec. 1997.

[11] S. Kim et. al., "A fast motion estimator for real-time systems," in *IEEE Trans. on Consumer Electronics*, vol. 43, No. 1, pp. 24–33, Feb. 1997.

[12] S. Dutta et. al., "A flexible parallel architecture adapted to block matching motion estimation algorithms," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, No. 1, pp. 74–86, Feb. 1996.

[13] P. Lakamsani, "An architecture for enhanced three step search generalized for hierarchical motion estimation algorithms," in *IEEE Trans. On Consumer Electronics*, vol. 43 no. 2, pp. 221-227, May 1997.

[14] A. P. Chandrakasan, et. al., "Minimizing power consumption in digital CMOS circuits," in *Proc. of IEEE*, Vol. 83, April 1995.

[15] R. Gonzalez, et. al., "Supply and threshold voltage scaling for low-power CMOS," in *IEEE Journal of Solid-state Circuits*, Vol. 31, No. 3, pp. 395-400, March 1999.

[16] R. Hegde, et. al., "Soft digital signal processing," in *IEEE Trans. on VLSI*, vol. 9 pp. 813-823, December 2001.

[17] B. Shim, et. al., "Low-power digital signal processing via reduced-precision redundancy," in *IEEE Trans. on VLSI Systems*, vol. 12 pp. 497-510, May 2004.

[18] L. Wang, et. al., "Low-power filtering via adaptive error-cancellation," in *IEEE Trans. on Signal Processing*, vol. 51 pp. 575-583, February 2003.

[19] http://www.eas.asu.edu/ ptm

[20] Y. Cao et. al., "New paradigm of predictive MOSFET and interconnect modeling for early circuit design," in *Proc. of CICC*, pp. 201–204, 2000.

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

[22] Y. Baek, et. al., "An efficient block matching criterion for motion estimation and its VLSI implementation," in *IEEE Trans. on Consumer Electronics*, vol. 42 no. 4, pp. 885-892, November 1996.

[23] J. M. Rabaey, et. al., "Digital integrated circuits A design perspective,"' Pearson Education, NJ: Prentice-Hall, 2003.

[24] S. C. Hsia, "VLSI implementation for low-complexity full-search Motion estimation," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, No. 7, pp. 613–619, July. 2002.

[25] B. Liu, et. al., "New fast algorithms for the estimation of block motion vectors," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3 no. 2, pp. 148-157, April 1993.