International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 2, October 2014

Transformation Techniques for Compensating Memory Errors in Jpeg2000

G. James Samuel[1], J. Sam Suresh[2]

P.G. Scholar, Department of VLSI Design, ACET, Tirupur, Tamil Nadu [1]

Assistant Professor, Department of ECE, ACET, Tirupur, Tamil Nadu [2]

Abstract: This paper presents novel techniques to mitigate the effects of SRAM memory failures caused by low voltage operation in JPEG2000 implementations. We investigate error control coding schemes; specifically single error correction double error detection code based schemes, and proposes an unequal error protection scheme tailored for JPEG2000 that reduces memory overhead with minimal effect in performance. Furthermore, we propose algorithm-specific techniques that exploit the characteristics of the discrete wavelet transform coefficients to identify and remove SRAM errors. These techniques do not require any additional memory, have low circuit overhead, and more importantly, reduce the memory power consumption significantly with only a small reduction in image quality.

Keywords: Error compensation, error control coding, JPEG2000, SRAM errors, voltage scaling.

I. Introduction

JPEG2000 is a widely used image coding standard that has applications in digital photography, high definition video transmission, medical imagery, etc., Since it processes one entire frame at a time, it has large memory requirements, and consequently, large memory power consumption. Voltage scaling is an effective way of reducing memory power. For instance, it was shown in that for a JPEG2000encoder, 25% to 35% power saving is possible when the memory operate sat scaled voltages. However, aggressive voltage scaling exacerbates RAM memory errors especially in scaled technologies.

SRAM failure rate is affected by threshold voltage variations, which in turn, is affected by process variations such as those due to random do pant fluctuation (RDF), length, width and oxide thickness, soft errors and others. Of these, RDF and channel length modulations are the most dominant factors.

In scaled technologies, the standard deviation of are fairly large; with voltage scaling, this increases causing the number of memory errors to increase. However, image and video codec's have some degree of error tolerance and full correction of memory errors is not needed to provide good quality performance. Several circuit, system and architecture level techniques have been proposed to mitigate and/or compensate for memory failures. In circuit level, different SRAM structures such as 8 T and 10 T have been proposed , the error locations in the cache are detected using built in self test circuitry and an error address table is maintained to route accesses to an error-free locations. Many techniques make us of error control coding (ECC) such as orthogonal Latin square co-design and extended Hamming codes in. More recently, algorithm specific way techniques have been developed for codec's such as JPEG2000 MPEG-4 to compensate for memory errors caused by voltage scaling. In linearization and second derivative of the image are used to detect error locations in different sub-bands in JPEG2000.

Digital Object Identifier are then corrected in an iterative fashion by flipping one bit at a time starting from the most significant bit (MSB). The overall procedure has fairly high latency and power overhead.

The method in uses combination of 6 and 8 T SRAM cells based on read reliability, area and power consumption and applies it to a MPEG-4 implementation. In this work, we propose several ECC and algorithm-specific techniques to mitigate the effect of memory failures caused by low voltage operation of JPEG2000. This work is an extension. We first investigate single error correction, double error detection (SECDED) codes and propose an unequal error protection (UEP) scheme that is customized for JPEG2000. The UEP scheme assigns ECCs with different strengths to different sub-bands so that the overall memory overhead it is reduced with minimal effect in performance. Next, we propose four algorithm-specific techniques with different levels of complexity that do not require additional SRAM memory. These techniques will be exploit the characteristics of the discrete wavelet transform

(DWT) coefficient sand use these to identify and correct errors in the high frequency components.

They allow the codec to operate at a high performance level even when the number of memory errors is quite high. For instance, use of these techniques enables the memory to be operated at 0.7 V resulting in an estimated 62%reduction in memory power with only 1Db quality degradation when the bit rate is 0.75 bits per pixel. Also, these techniques have very low overhead and are thus ideal for reducing the power consumption of JPEG2000 implementations.

The rest of this paper is organized as follows. In Section II, SRAM analysis and a brief summary of JPEG2000 is presented. In Section III, we present the UEP methods followed by four algorithm-specific methods in Section IV. Simulation and power estimation results for well-know test images are given.

A. Sram Failure Analysis

The overall failure rate in memories is dominated by variation caused by RDF and channel length modulation. The effect of RDF on threshold voltage is typically modeled with an additive iid Gaussian distributed voltage variation. Variance of threshold voltage of a MOSFET is proportional to , where is oxide thickness, and are length and width of the transistor, respectively. For 32 nm, is approximately between 40 to 60 mV; for 22 nm, the numbers increase to 60 to 80 SRAM failure analyses have been investigated by several researchers statistical models of RDF are used to determine read, write failure and access time variations.

In read and write noise margin of 6 T SRAM structure is used to calculate reliability of the memory. The read/write failure rate of 6 TSRAM structure at a typical corner for 45-nm technology can be as high as at 0.6 V .We simulated SRAM failures caused by RDF and channel length variation for 32 nm technology using Hospice with high performance TM from [12]. An SRAM cell with bit line load value equal to those of256 cells, with half of them storing "1" and the other half storing "0", is simulated using Monte Carlo simulations. The overall bit error rate in the given images are been produce (BER) is calculated for different levels of RDF with 5% channel length variation at different.

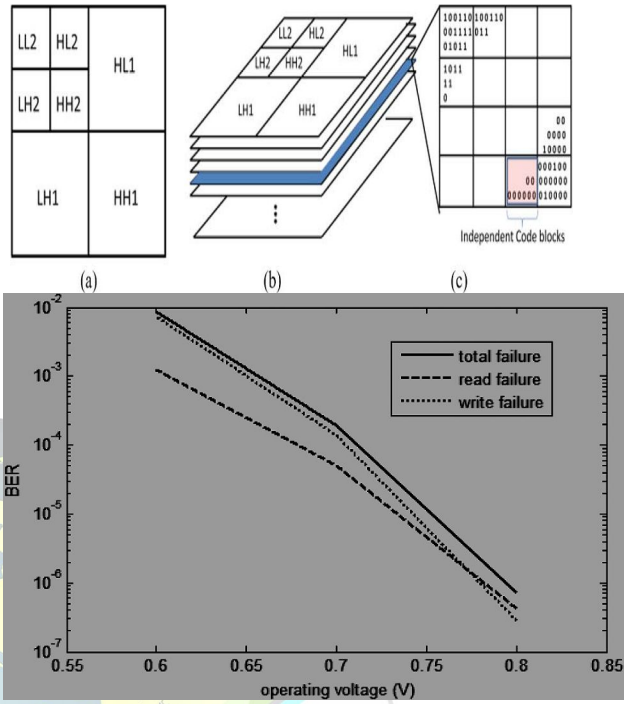


Fig.1. Read, write, and total failure probability of SRAM in 32-nm technology for different voltage.

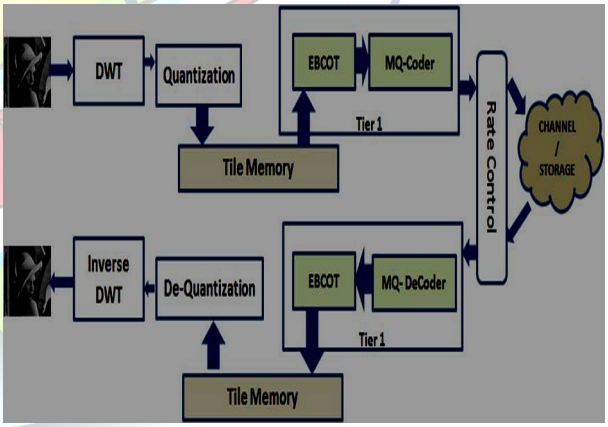


Fig.2. Block diagram of JPEG2000.

B. Jpeg2000 Summary

The general block diagram of theJPEG2000 encoder/decoder is illustrated in Fig. 2. The original image (in pixel domain) is transformed into frequency sub-bands using the DWT engine followed by quantization and stored into tile memory. Compressed image is obtained after embedded block coding with optimal truncation (EBCOT) processing followed by rate control. The two level sub-band representation of the DWT output is shown.The input image

of size is processed by high-pass (H) and low-pass (L) filters along rows and columns followed by sub sampling.

For a natural image, the low frequency sub-bands are likely to contain coefficients with larger values while higher frequency sub-bands such as HL1, LH1, HH1 contain coefficients with small values. Moreover, the coefficients in a sub-band have similar magnitudes. We exploit these facts in developing the memory error compensation techniques. In most implementations, DWT coefficients are stored in the tile memory, which is typically an SRAM data memory. Aggressive voltage scaling introduces errors in these memories. As a result the DWT coefficients that are stored in the tile memory are different. Sub-band representation of DWT output. (b)Bit plane representation of Sub-band. (c) Code block representation of bit planes different levels of DWT from the data that is read out of the memory for EBCOT processing.

II. ECC SCHEMES

In this section, we study the use of ECC in combating errors in memories. We study the use of 3 different SECDED codes: (137, 128), (72, 64) and (39, 32). Of these three codes, (39, 32) is the strongest followed by (72, 64), and (139, 128). The memory overhead of an ECC code is defined as where represents the number of information bits for an ECC codeword length of. Since the memory area Overhead of the stronger codes is very large; we propose to use an UEP scheme where the more important bits are given higher protection by encoding them with stronger codes.

Thus by using a combination of strong and weak codes, the memory overhead can be reduced without sacrificing performance. In order to quantitatively measure the importance of a bit, we introduce which is the mean square error due to bit failures in memory. This is the same for all images and is solely a function of the sub band level and location of the error bit position in the sub band coefficient and the normalized for different sub-band outputs of a 3-level DWT as a function of a 1 bit error in different bit positions starting with the most significant bit (MSB). The values are normalized with respect to maximum of the LL3 sub-band. We see that level-3 sub band outputs are the most sensitive to bit errors and thus should be protected with the strongest code. Also, errors inMSB-2 bit of level 3 outputs (LL3, HL3, LH3, and HH3) generate approximately same degradation in image quality as errors in MSB-1 bit of level 2 outputs (HL2, LH2, and HH2) and errors in MSB bit of level1 outputs (HL1, LH1, and HH1).We use the same strength code for the bits that generate similar.

In a system that uses 3 codes, we break Fig. 4 into 3 regions bounded by line-1 and line-2. We use the strongest code, which is(39, 32), for the points above line-1, (72, 64) code for the points. Between line-1 and line-2 and the (137, 128) code for the rest of the points. In the proposed method, the optimal settings of line-1 and line-2 depend on memory overhead and quality degradation. Overall memory overhead (MO) is sum of memory overheads due to each ECC code.

We study two schemes: (i) fixed overhead scheme (UEP-1) which minimizes degradation subject to, where is the memory constraint (ii) fixed performance loss scenario (UEP-2) which minimizes memory overhead subject to, where is the allowable performance loss. Consider an example of UEP-1 when and the memory overhead constraint is 0.125. Of the candidate schemes, minimum degradation is achieved with line-1 corresponding to Set-3 and line-2 corresponding to Set-2. Using this configuration, drops by 35% compared to when only (72, 64) is used. Note that these settings change with different memory BER since the coddler for the constituent codes are a function of the memory BER.

III. ALGORITHM-SPECIFIC TECHNIQUES

It is well-known that in natural images neighboring pixels are highly correlated. It turns out that in the frequency domain, neighboring coefficients have similar magnitudes. Moreover, for a natural image, DWT outputs at high sub-band (higher frequency) typically consist of smaller values. For these coefficients, isolated non-zero MSBs are unlikely and can be used to flag errors. We propose four methods for HL, LH and HH sub-bands and an additional method for LL sub-band to mitigate the impact of memory errors on system quality. We consider both random errors and burst of errors.

Method 1: In this method, we erase all the data (ones) in the bit planes that are higher than a certain level for high sub-bands. Through simulations, we found that for 16 bit data very little information is lost by discarding 2 to 4MSB planes. Furthermore, since EBCOT skips coding bit planes that consist of all-zero bits, this technique helps reduce the power consumption due to EBCOT.

Method 2: Although Method 1 is very simple, there can be false erasures in MSB planes resulting in loss of quality.Method2 addresses this problem by exploiting the image statistics. Here, the number of ones in a given bit plane is counted and if the number is below a certain threshold, all the bits in that plane are erased and the all-zero bit plane information is passed to EBCOT. The threshold value here is dynamic and is set to be equal to twice the expected number of errors in a bit plane. The overhead of this method is the counter. Fortunately, it is not triggered

often since it operates only on the high bit planes. Also, it is disabled after it identifies the first bit of the given plane that is not erased.

Method 3: Discarding all the bits in a given bit plane when a threshold condition is satisfied may sometimes result in losing valuable MSBs. We propose a third method which looks at data in current bit plane and also data in one upper and two lower bit planes. This is motivated by the fact that bits in a given bit plane are correlated with bits in their neighboring bit planes. This method first decides whether to process the current bit plane by counting the number of bits in the bit plane and comparing it with a dynamic threshold.

Next, to process the current non-zero bit in the selected bit-plane, it uses a neighborhood. Specifically, if a non-zero bit is detected in position of the plane, it checks the 3 block of bits around the position in the and plane in addition to its 8 neighbors in the plane. If it detects another 1within this search group, it decides that the current bit is a correct 1. Method-3 also stops after identifying the first bit-plane that is not eligible for erasure.

Method 4: This is an extension of Method 3 to handle burst errors. The steps are the same as that of Method 3 except that if the other non-zero bit in the group is consecutive to the current bit, then an error is flagged for the current bit.

Correction Method for LL Sub band: Here simple filtering is sufficient to correct the errors. For each DWT output, a 3 3 block of its neighbors is used to calculate its expected value. If the current value differs from the mean by, the original value is kept; otherwise it is updated by the mean value. In order to reduce the amount of computation, we only consider 8 MSB of a 16 bit output to calculate the mean. Through our simulations, we found that for this scenario is been generated and gives very good results.

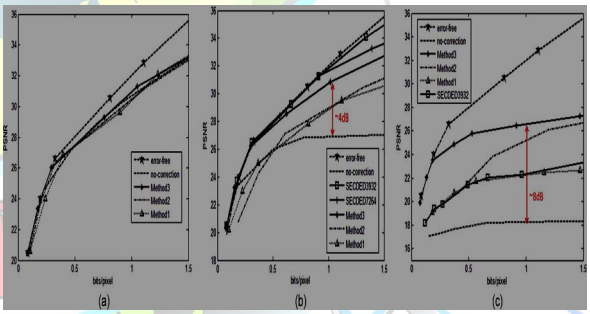
IV. Simulation Results

In this section, we describe the quality performance and overhead of the algorithm-specific and ECC-based methods for JPEG2000. The quality performance is described in terms of peak signal to noise ratio (PSNR) between the original and decoded image, and compression rate is determined by the number of bits required to represent one pixel (bpp).

We implemented (9, 7) lossy JPEG2000 with 3 levels DWT and 32 EBCOT block size. Four representative images, Lena, Cameraman, Baboon and Fruits, are used in the simulations .MATLAB is used to compute the performance curves. The overall BER rate in the tile memory is changed from to which is compatible with the BER results obtained for 6 T SRAM under voltage scaling.

Model-1 which represents fully random errors, and ***Model2*** *which* represents burst errors characterized by probability density function of number of failures given by: where probability of an error in one bit is 0.5, errors in two consecutive bits is 0.2 and errors in three consecutive bits is 0.03. If some amount of image quality degradation is tolerable (below 0.5 dB for 0.5 bpp), the no-correction method is sufficient and there is no benefit of using any of the algorithm-specific methods. In addition, extended Hamming of (39, 32) can provide almost error-free performance at this level. The performance curves show a divergent trend when BER increases. Algorithm-specific methods can provide good results for low and medium quality/rate regions. ***Method3*** follows the error-free curve very closely in the range 0.2 to 1 bpp. For example, it improves quality approximately by 4 dB at 1 bpp rate compared to the no-correction case. If some degradation is tolerable.

Methods 1 and 2 are good candidates when compression rate is above 0.8 bpp, since they have lower complexity compared to ***Method 3.***



Algorithmic methods improve the quality by 3 to 8 dB for compression rate around 0.75 bpp. Method 3 follows the no-error curve closely under 0.25 bpp compression rate. For medium compression rate, it improves the performance noticeably and achieves the best performance quality among all techniques. The performance of ECC methods, even the strongest ECC, namely (39, 32) is not good for fully random error model, and deteriorates for burst error model.

The memory overhead constraint for UEP-1 is 0.125 and the performance constraint for UEP-2 is (normalized) is . From the table, we see that Method 3 (for random errors) and Method 4 (for burst errors) can provide approximately 8 dB improvement compared to the no-correction case, and their average degradation is around 3 dB compared to the no-error case. Method 4, which has been optimized for burst errors, has an average of 1.9 dB improvement over Method 3 for burst error model. On the other hand, Method 3 has a superior performance compared to Method 4 for the random error model.

Overhead: Area, Delay, Power Consumption:

The circuits overhead of the algorithm-specific and ECC-based techniques are obtained using Design Compiler from Synopsys and 45 nm models from.

ECC-Based Techniques:

To support the various UEP schemes, combination of multiple ECCs to lower the overall circuit is used. The encoder and the decoder have a hierarchical structure and are based on the design. The power consumption, area and latency of encoder/decoder for the (137, 128), (72, 64), and (39, 32) codes are listed in Table II. The clock period is 2 ns. In addition to circuitry overhead, ECC techniques require extra memory to store the parity bits. This can be significant for stronger codes such as (39, 32). As mentioned earlier, the memory overhead can be reduced by implementing UEP instead of fixed ECC schemes.

TABLE I
AREA, LATENCY, AND POWER CONSUMPTION OVERHEAD OF ECC

	Encoder (137,128) /(72,64)/(39,32)	Decoder (137,128) /(72,64)/(39,32)
Area (um^2)	622.35	1139.93
Worst-case delay (ps)	508/390/270	1608/1142/610
Active Power (uW)	413.45/230.30/93.38	683.32/347.18/155.42
Leakage Power (uW)	6.34	11.22

TABLE II
ALGORITHM-SPECIFIC TECHNIQUES

	9-bit counter	All-zero detector	4-bit comparator
Area (um^2)	90.81	22.39	17.88
Worst-case delay (ps)	292	80	42
Active Power (uW)	31.77	0.78	0.58
Leakage Power (uW)	1.01	0.081	0.064

Algorithm-Specific Techniques:

The given power consumption, area and latency of the overhead circuitry, namely, 9-bit counter combined with the comparator used in Methods 2, 3, and 4, the all-zero detector used in Methods 3 and 4, and the 4-bit comparator used in Method 4, are illustrated in Table III when the clock period is 2 ns. Unlike the ECC based methods, there are no additional memory requirements.

V. CONCLUSION

In this paper, we presented use of UEP schemes and algorithm-specific techniques to the mitigate memory failures caused by aggressive voltage scaling in JPEG2000 implementations. The UEP schemes can achieve better performance and lower overhead compared to generic ECC schemes by coding bit planes of different levels of DWT sub-bands according to their importance.

However these schemes do not have good performance for high error rates. Next, algorithm-specific techniques are presented which require no additional memory, have low circuit overhead and outperform the best ECC-based schemes for high bit error rates for both random and burst error scenarios.

They exploit the redundancy in DWT outputs of JPEG2000 to identify and correct memory errors. These techniques enable us to drop the operating voltage of memory while causing acceptable reduction in image quality for low to medium compression rates. For instance, they enable the memory to be operated at 0.7 V resulting in 62% memory power saving (assuming same number of reads and writes) and 25% overall power saving (assuming that the memory power is on average 40% of the overall power as in [2]) with only 1 dB loss in PSNR. If 4 dB loss in PSNR is acceptable, they enable the memory to be operated at 0.6 V resulting in 78% memory power saving and 31% overall power saving.

REFERENCES

[1]. P. Schelkens, A. Skodras, and T. Ebrahimi, The JPEG 2000 Suite.New York: Wiley, 2009.

[2]. M. A. Makhzan, A. Khajeh, A. Eltawil, and F. J. Kurdahi, "A low power JPEG2000 encoder with iterative and fault tolerant error concealment,"IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 6, pp. 827–837, Jun. 2009.

[3]. S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 24, no. 12, pp. 1859–1880, Dec. 2005.

[4]. A. Agarwal, B. C. Paul, S. Mukhopadhyay, and K. Roy, "Process variation in embedded memories: Failure analysis and variation aware architecture," IEEE J. Solid-State Circuits, vol. 40, no. 9, pp. 1804–1813,Sep. 2005.

[5]. K.Agarwal and S.Nassif, "Statistical Analysis of SRAMcell stability," in Proc. Design Autom. Conf., 2006, pp. 57–62.

[6]. B. H. Calhoun and A. Chandrakasan, "A 256 kb subthreshold SRAM in 65 nm CMOS," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, 2006, pp. 628–629.

[7]. G. K. Chen, D. Blaauw, T. Mudge, D. Sylvester, and N. S.Kim, "Yielddriven near-threshold SRAM design," in Proc. Int. Conf. Comput.- Aided Design, 2007, pp. 660–666.

[8]. Z. Chishti, A. R. Alameldeen, C. Wilkerson, W. Wu,and S. Lu, "Improving cache lifetime reliability at ultra-low voltages," in Proc.Micro, 2009, pp. 89–99.

[9]. T. R. N. Raoand and E. Fujiwara, Error Control Coding for Computer Systems. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[10]. I. J. Chang, D. Mohapatra, and K. Roy, "A voltage-scalable & process variation resilient hybrid SRAM architecture for MPEG-4 video processors," in Proc. Design Autom. Conf., 2009, pp. 670–675.

[11]. H. Yamauchi, "A discussion on SRAMCircuit Design Trend in Deeper Nanometer-Scale Technologies," IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol. 18, no. 5, pp. 763–774, May 2010.

[12]. Arizona State University, Tempe, AZ, "Predictive Technology Models," 2008. [Online]. Available: http://ptm.asu.edu/