Energy and Quality-Aware Multimedia Signal Processing

By

Yunus Emre

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved August 2012 by the
Graduate Supervisory Committee:

Chaitali Chakrabarti, Chair
Bertan Bakkaloglu
Yu Cao
Antonia Papandreou-Suppappola

ARIZONA STATE UNIVERSITY

December 2012

ABSTRACT

Today’s mobile devices have to support computation-intensive multimedia applications with a limited energy budget. In this dissertation, we present architecture level and algorithm-level techniques that reduce energy consumption of these devices with minimal impact on system quality.

First, we present novel techniques to mitigate the effects of SRAM memory failures in JPEG2000 implementations operating in scaled voltages. We investigate error control coding schemes and propose an unequal error protection scheme tailored for JPEG2000 that reduces overhead without affecting the performance. Furthermore, we propose algorithm-specific techniques for error compensation that exploit the fact that in JPEG2000 the discrete wavelet transform outputs have larger values for low frequency subband coefficients and smaller values for high frequency subband coefficients.

Next, we present use of voltage overscaling to reduce the data-path power consumption of JPEG codecs. We propose an algorithm-specific technique which exploits the characteristics of the quantized coefficients after zig-zag scan to mitigate errors introduced by aggressive voltage scaling.

Third, we investigate the effect of reducing dynamic range for datapath energy reduction. We analyze the effect of truncation error and propose a scheme that estimates the mean value of the truncation error during the pre-computation stage and compensates for this error. Such a scheme is very effective for reducing the noise power in applications that are dominated by additions and multiplications such as FIR filter and transform computation. We also present a novel sum of absolute difference (SAD) scheme that is based on most significant bit truncation. The proposed scheme exploits the fact that most of the absolute difference (AD) calculations result in small values, and most of the large AD values do not contribute to the SAD values of the blocks that are selected. Such a scheme is highly effective in reducing the energy consumption of motion estimation and intra-prediction kernels in video codecs.

Finally, we present several hybrid energy-saving techniques based on combination of voltage scaling, computation reduction and dynamic range reduction that further reduce the energy consumption while keeping the performance degradation very low. For instance, a combination of computation reduction and dynamic range reduction for Discrete Cosine Transform shows on average, 33% to 46% reduction in energy consumption while incurring only 0.5dB to 1.5dB loss in PSNR.

.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Chaitali Chakrabarti for her guidance, support, and encouragement. Working with her has been my best learning experience so far. I believe that I am capable of working independently now and I have not felt better prepared than I feel now to undertake challenges that may lie ahead. I have been feeling amazingly fortunate that I could work with her in the past years.

I would like to extend my gratitude to Professors Antonia Papandreou-Suppappola, Bertan Bakkaloglu and Yu Cao for serving as members of my committee. I thank all my friends at Arizona State University, Osman Emir Erol, Berkay Kanberoglu, Ender Yilmaz, Ahmet Cemal Durgun who made my stay here pleasant and enjoyable, other friends, family and relatives who wished me success with my program and have been in touch with me for a very long time. I am profoundly indebt to my mentors at Intel Corporation, Umit Ogras, Mike Kishinevsky, and Jianping Xu, for teaching me the importance of crtitical thinking and hardworking. I also wish to thank my fellow members of Low Power Systems Lab, Chengen Yang, Ming Yang, and Rupa Mahadevan, for providing a fun environment in which to discuss, and learn more. I am also thankful to Science Foundation of Arizona (SFAz) and National Science Foundation (NSF) for supporting my work during my PhD study.

I am deeply grateful to my father, Cafer Emre, my mother, Saliha Emre, and my sister, Nilufer Emre. None of this would have been achieved without their support. Special thanks go to my wife, Refika Koseler Emre, who has always been my joy and my pillar. I would like to thank her for her company, understanding, and encouragement during the past years.

TABLE OF CONTENTS

Page

LIST OF TABLES viii

LIST OF FIGURES x

CHAPTER

1. INTRODUCTION 1

I. Overview of Existing Work 2

II. Problems Addressed 3

A. Voltage Scaling and Error Compensation in Memories: 3

B. Voltage Scaling and Error Compensation in Data-path: 5

C. Dynamic Range Reduction in Datapath: 6

D. Hybrid Schemes for Energy Reduction in Data-path 8

III. Organization 9

2. BACKGROUND 10

I. Sources of Power Consumption 10

II. Power Reduction in Dynamic Component 10

A. Power Reduction by Voltage Scaling 11

B. Power Reduction by Voltage Scaling 11

C. Power Reduction by Architecture Modification 13

D. Power Reduction by Algorithm Modification: Reducing Number of Computations 13

CHAPTER Page

III. Power Reduction of Static Component 15

IV. Overview of Image/ Video Coding Algorithms 16

A. JPEG [44] 17

B. JPEG2000 19

C. Video Codec (H.264) 22

3. COMPENSATING FOR MEMORY ERRORS CAUSED BY VOLTAGE SCALING: CASE STUDY JPEG2000 26

I. Introduction 26

II. Power Reduction in Memories 26

A. SRAM Failure Analysis 27

B. Previous Work on Compensating for SRAM Errors 31

III. Proposed Unequal Error Protection for JPEG2000 33

IV. Proposed Algorithm-Specific Techniques for JPEG2000 40

A. Method-1 41

B. Method-2 41

C. Method-3 42

D. Method-4 43

V. Simulation Results for the Proposed Techniques 43

A. System Performance Results 43

B. Overhead: Area, Delay, Power Consumption 49

VI. Summary 53

4. COMPENSATING FOR DATAPATH ERRORS CAUSED BY VOLTAGE SCALING: CASE STUDY JPEG 54

I. Introduction 54

II. Voltage Scaling Induced Errors in Datapath 54

A. Failure Analysis 54

B. Previous Work on Compensating Datapath Errors 56

III. Proposed Algorithm-Specific Technique: Case Study DCT 58

IV. Simulation Results 61

A. Algorithm Performance 61

B. Hardware Overhead 63

V. Summary 64

5. COMPENSATING FOR DATAPATH ERRORS CAUSED BY DYNAMIC RANGE REDUCTION 65

I. Introduction 65

II. Effects of Dynamic Range Reduction 65

III. Reducing Dynamic Range by Low Order Bit Truncation 66

IV. Reducing Dynamic Range by High Order Clipped Computation (HOC) 76

A. Proposed SAD Algorithm 80

B. Proposed SAD Architecture 82

CHAPTER Page

C. Performance Analysis of Proposed HOC 86

Algorithm Performance 86

D. Energy Quality Trade-off 89

V. Summary 92

6. HYBRID ENERGY SAVING TECHNIQUES FOR DATAPATH 95

I. Combining Computation Reduction and Voltage Scaling 95

II. Combining Voltage Scaling and Dynamic Range Reduction 96

III. Combining Computation Reduction and Dynamic Range Reduction 102

IV. Summary 109

7. CONCLUSION 110

Works Cited 114

LIST OF TABLES

Table Page

3.1. PSNR values of different techniques at 0.75bpp compression rate for BER=10 − 2, using error models 1&2 49

3.2. Necessary modules for algorithm-specific methods 50

3.3. Area, latency and power consumption overhead of circuit components used in the algorithm-specific methods 51

3.4. Area, latency and power consumption overhead of ECC Schemes for (137,128), (72,64) and (39,32) 52

4.1. Number of bits necessary to represent each group of 2D DCT coefficients for natural images 59

4.2. PSNR values of proposed technique at 0.75bpp compression rate when BER(VOS)=10 − 4 63

4.3. Power Consumption and latency of the three units in the voltage overscaling compensation scheme 63

5.1. Quality, power and latency of DCT engine for different levels of truncation 74

5.2. Percentage of AD Values during Inter-prediction that can be represented by 4 to 8 bits for Football Video sequence 78

5.3. Percentage of AD values during Inter-prediction that can be represented by 4 to 8 bits for Foreman Video Sequence 78

5.4. Statistics of Selected and Unselected SAD Values for Foreman Video Sequence during Inter Prediction 80

5.5. Average increase in compressed data rate and reduction in PSNR *Thr*=16,32,64..... 82

5.6 Average increase in compressed data rate and reduction in PSNR for Inter-prediction 88

Table Page

5.7 Average increase in compressed data rate and reduction in PSNR for Intra-prediction 88

6.1. MSE Improvement for different Gaussian Filters.. 101

6.2. Reduction Order of 6 Sample Images 105

6.3. Final Order for Scheme I, II and III 106

6.4. Delay, Power and Area for Original and Reduced Computation 1D-DCT.............. 107

LIST OF FIGURES

Figure Page

1.1 Goal: Reduce energy consumption with minimal quality degradation 2

2.1. Power delay profile of 14-bit RCA adder under voltage scaling 12

2.2. Sub-sampling patterns for ½ and ¼ schemes 15

2.3. Block Diagram of JPEG 17

2.4. 2D DCT architecture using 1-D DCTs 17

2.5. a) Luminance Quantization Matrix for $Q = 50$; b) Zigzag scan order 19

2.6 Block diagram of JPEG2000 20

2.7 a) Sub-band representation of DWT output b) Bit plane representation of Sub-band c) Code block representation of bit planes 22

2.8. Block Diagram of the Video Encoder 23

2.9. a) Nine prediction modes of Intra_4x4 b) Diagonal Down Right 25

3.1. σVT scaling trend with and without oxide scaling [35] [53] 27

3.2. Schematic of a SRAM cell 28

3.3. Delay distribution of a single SRAM read operation 30

3.4. Read, write and total failure probability of SRAM in 32nm technology for different voltage levels when σVT=40mV 31

3.5. ΔDE (normalized) due to error in different bit positions for different levels of DWT a) Linear b) Logarithmic Scale 35

3.6. a) Overhead and b) Performance Degradation for different settings of line-1 and line-2. 37

3.7. Performance Degradation x Overhead for different settings of line1 and line2 39

Figure Page

3.8.Parity generation of (39, 32) code from (72, 64) code 39

3.9. Number of non-zero bits in different subbands in a) Bridge b) Lena images 41

3.10. Lena Image a) Original b) HL1 c) HL2 subband 41

3.11. Bit planes used in error detection mechanism 42

3.12. Quality vs. Compression rates when BER=$10-4$ for Lena a) for error model-1 b) for error model-2 46

3.13. Quality vs. Compression rates when BER=$10-3$ for Lena a) for error model-1 b) for error model-2 47

3.14. Quality vs. Compression rates for Lena when BER=$10-2$ using algorithmic techniques a) for error model-1 b) for error model-2 48

3.15. Block diagram of encoder for (137,128), (72, 64) and (39, 32) codes 52

4.1. Block Diagram of 14-bit RCA 55

4.2. Probability of error distribution for 14-bit RCA for different voltage settings 56

4.3. Block Diagram of Algorithm Noise Tolerant (ANT) Scheme 57

4.4. Magnitude of DC and AC coefficients a) averaged over all blocks; b) first 20 blocks of Bridge image 59

4.5. PSNR vs. Compression rate for Bridge when BER=$10-4$ 62

4.6. PSNR vs. Compression rate for Bridge when BER=$10-3$ 62

5.1. Energy and Noise distribution of 16-bit RCA wrt bit-width 66

5.2. Average performance loss due to bit truncation a) using DS and TSS in motion estimation b) intra prediction of H.264 67

5.3: Noise power of $Qmul$ for 16-bit multiplication with and without μ-compensation.. 70

Figure Page

5.4. Architecture of 1-D DCT coefficients. a) First stage butterfly b) $w0$ and $w4$ computation units, c) $w2$ unit, d) $w1$ unit 72

5.5. Modified DCT computation of $W1$ 73

5.6. Performance comparison between uncompensated and compensated bit truncation for DCT computation of Baboon image 74

5.7. MAC implementation of a low pass filter 76

5.8. Histogram plots of AD computations in motion estimation of a) Football and b) Foreman video sequences 77

5.9. Histogram plots of AD computations in intra-prediction of a) Football and b) Foreman video sequences 77

5.10. Histogram of SAD values of all blocks a) Inter Prediction b) Intra Prediction 79

5.11. Histogram of SAD values of selected blocks a) Inter Prediction b) Intra Prediction 79

5.12. a) Baseline SAD calculation unit, b) 8-bit absolute difference calculator 82

5.13. Proposed SAD calculation unit 83

5.14. Flow Chart of the proposed SAD calculation scheme 84

5.15. 6-bit Half Adder Chain 85

5.16. Delay vs Energy Consumption of the baseline and proposed schemes 86

5.17. PSNR vs Compressed Data Rate of a) Football, b) Claire sequences under different configurations for inter-prediction 87

5.18. Cost of using proposed scheme in terms of percentage increase in compressed data size and PSNR reduction for inter-prediction. 88

5.19. Algorithm performance for different SAD architectures 90

Figure Page

5.20. Comparison of latency and energy consumption at nominal voltage of all candidate schemes 91

5.21. Inter-Prediction a) Compressed Data Rate Increase (ΔCR) vs Normalized Energy, b) PSNR degradation ($\Delta PSNR$) vs Normalized Energy 93

5.22. Intra-Prediction a) Compressed Data Rate Increase (ΔCR) vs Normalized Energy, b) PSNR degradation ($\Delta PSNR$) vs Normalized Energy 94

6.1. 16-bit RCA a) Delay, b) Energy, c) Average Error per Operation distribution with and without precision reduction under voltage scaling 98

6.2. 16-bit CLA Energy distribution with and without precision reduction under voltage scaling for comparable average error per operation 99

6.3. Noise Power vs Energy Consumption of 3x3 Gaussian filter on Lena image under voltage scaling with a) precision reduction, b) precision reduction with voltage scaling 100

6.4. Performance comparison of 3x3 Gaussian filter on Lena image; original and reduced precision filter with and without compensation 101

6.5. DCT coefficient deactivation and low order bit truncation for the 1D-DCT coefficients 103

6.6. Comparison between bit truncation level and DCT coefficient deactivation for a) $W_{7,}$ b) W_5 for Baboon image when Q=55 104

6.7. Binary decision tree to choose combination of coefficient deactivation and truncation level 104

6.8. Performance of the resulting decision order generated using 6 training samples on test images a) Lake, and b) Elaine 106

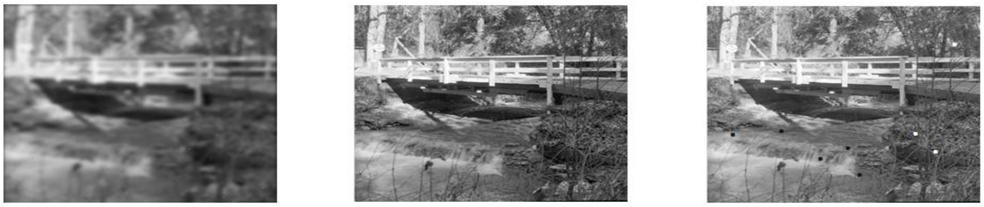
6.9. Energy saving at different quality levels for three candidate schemes 108

CHAPTER 1

INTRODUCTION

Portable multimedia devices have proliferated in the last two decades, and the number of applications supported by these devices has increased significantly. Each additional application results in higher energy consumption and since most of these devices are battery powered, it is important that every effort be made to reduce the energy cost. The challenge is to minimize this cost while executing increasingly complex functionalities with minimal degradation in algorithm performance quality. Fortunately, many of the multimedia applications do not need 100% correctness during computation and energy saving transformations are favored as long as the output quality is only mildly affected [1] [2].

There are many multimedia systems which achieve low energy consumption at the expense of low system quality. Our goal is to achieve low energy consumption with minimal degradation in quality. Fig. 1.1 shows the DCT energy consumption and final image quality for a JPEG coded sample image corresponding to three different algorithm configurations. On two ends of the spectrum are configurations with low energy and low quality (left sub-figure) and high energy and high quality (middle sub-figure). Our goal is to derive configurations with fairly low energy and reasonably high quality performance as shown in the right sub-figure. In this work, we propose several architecture and algorithm level techniques that help derive such configurations for image and video codecs.



Low Energy (9.2 nJ)
Low Quality (PSNR = 19dB)

High Energy (16.2 nJ)
High Quality (PSNR = 31dB)

Low Energy (11.3 nJ)
High Quality (PSNR = 28 dB)

Fig. 1.1 Goal: Reduce energy consumption with minimal quality degradation

I. OVERVIEW OF EXISTING WORK

Three of the most effective techniques for reducing energy consumption are voltage scaling [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15], reduction in number of computations [2] [14] [16] [17] [18] [19] [20] and dynamic range adjustment [17] [20] [21] [22] [23] [24]. While voltage scaling results in significant reduction in energy consumption due to the quadratic dependence between supply voltage and energy consumption, voltage over-scaling (VOS) can lead to failures. Techniques have been developed to mitigate the errors due to critical path violation in the computation unit and memory due to VOS. While circuit-level techniques such as Razor [25] are quite effective, we propose low overhead algorithm-level techniques that use the inherent redundancy and characteristics of the data to detect and correct errors that occurred during the computation.

Unlike general purpose computing, most multimedia applications can provide decent quality even with reduced number of computations as long as the significant computations are retained. The basic idea is that all components of the computation are not equally significant and so for systems with limited resources, the more important computations are done first and the less important computations are performed later or even eliminated. Such a methodology has been successfully applied to many image and video processing algorithms [2] [19] [20].

Another popular energy-saving technique is dynamic range reduction in the datapath computation. Typically, low order bits are less important and so can be truncated to save energy. Such a methodology has been used in many multimedia kernels such as filtering [20], DCT [21] [23], FFT [5] etc. While truncation reduces energy consumption, it also introduces errors due to operation with a reduced dynamic range.

II. PROBLEMS ADDRESSED

We describe several energy-saving techniques that achieve minimum degradation in quality with low overhead for image and video codecs. While some of these techniques are general, others have been geared to exploit the algorithmic features and result in superior performance both in terms of energy consumption and algorithm performance quality.

A. Voltage Scaling and Error Compensation in Memories:

Voltage scaling is an effective way of reducing memory power. However, aggressive voltage scaling exacerbates SRAM memory errors especially in scaled technologies. Several circuit, system and architecture level techniques have been proposed to mitigate and/or compensate for memory failures. At the circuit level, different SRAM structures such as 8T and 10T have been proposed [26] [27]. In [28], 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 use of error control coding (ECC) such as orthogonal latin square codes in [12] and extended Hamming codes in [29]. More recently, algorithm-specific techniques have been developed for codecs such as JPEG2000 [13], MPEG-4 [10] to compensate for memory errors caused by voltage scaling. In [13], binarization and second derivative of the image are used to detect error locations in different sub-bands in JPEG2000. These 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 [10] uses a combination of 6T and 8T SRAM cells based on read reliability, area and power consumption and applies it to a MPEG-4 implementation.

Contribution [9] [30]**:** We propose techniques that make use of voltage overscaling to reduce the energy consumption of memories with minimal effect on the system quality. We choose JPEG2000 as the demonstration platform since it is a widely used image coding standard that has applications in digital photography, high definition video transmission, medical imagery, etc [31]. This codec has large memory requirements since it processes one entire frame at a time, and consequently, large memory power consumption. For instance, it was shown in [9] [13] that in a JPEG2000 encoder, 25% to 35% power saving is possible when the memory operates at scaled voltages.

In our work, first, we characterize the different types of memory error under scaled voltage levels. Majority of the errors are exacerbated by process variations. These include variation of the transistor sizes (W/L) and variation in V_{th} due to random dopant fluctuation (RDF). Then, we propose several techniques to mitigate the effect of memory failures in JPEG2000 due to operating the memory at scaled voltages. The first scheme is an unequal error protection (UEP) scheme based on single error correction double error detection (SECDED) codes that is customized for JPEG2000. The UEP scheme assigns error control coding (ECCs) with different strengths to different subbands so that the overall memory overhead is reduced with minimal effect in performance. Next, we propose four algorithm-specific methods with different levels of complexity that do not require additional SRAM memory. These methods exploit the fact that the discrete wavelet transform (DWT) outputs have larger values for low frequency subband coefficients and smaller values for high frequency subband coefficients and that the coefficients in a subband have similar magnitudes. Thus errors in most significant bits (MSBs) of high frequency components can easily be identified and removed.

The algorithm-specific methods allow us to operate at very high compression ratio for high memory bit error rate (BER) with small drop in performance. For instance, for compression ratio of 0.75bpp and BER=10^{-3}, the proposed techniques can achieve less than 1dB drop in PSNR from the no-error case. Even for very high bit error rate of BER=10^{-2}, there is only a 2.8dB degradation at 0.75bpp with no memory cost. ECCs, on the other hand, can provide decent performance when BER=10^{-3} and the high memory overhead of these codes can be reduced by using our UEP scheme. However, at BER=10^{-2}, the error correcting capability of fixed ECC or UEP scheme is significantly worse than the algorithm-specific methods. For compression ratio of 0.75bpp, the best ECC scheme is approximately 5dB less than the best algorithm-specific method. Furthermore, the overhead of algorithm-specific methods is very small, making it possible to easily compensate for errors introduced due to low voltage operation of SRAM in JPEG 2000 implementations.

B. Voltage Scaling and Error Compensation in Data-path:

As in memory, an effective way of reducing the power consumption in the datapath is lowering the supply voltage. However, this could result in critical path violations leading to failures. In our work, we investigate use of voltage overscaling to reduce the energy consumption of datapath with minimal effect on the system quality. We focus on JPEG since this is part of many embedded devices for multimedia where power consumption is a very important metric.

Several JPEG architectures have been proposed that trade-off power consumption and quality. The DCT architecture in [32] exploits correlation between DCT coefficients in conjunction with standard techniques such as voltage scaling, data parallelism and pipelining. Data bit-width adaptation is used in [21] to reduce the processing load of high frequency coefficient computations. A similar scheme is also investigated in [23] where truncation of up to 4 low order bits is shown to achieve 40% reduction in energy consumption of the memory and data-path.

Contribution [8] [33]**:** We first characterize the distribution of timing violations in datapath and model the errors due to aggressive voltage scaling. Most of these errors reside in high order bits that incur very excessive degradation in system quality. To compensate for these errors, we introduce an algorithm-specific technique tailored for Discrete Cosine Transform (DCT) engine in JPEG codecs. The technique exploits the fact that in 8×8 DCT, two adjacent AC coefficients after zig-zag scan have similar values and two coefficients corresponding to higher frequencies generally have smaller values. These features are used to detect the datapath errors and then compensate for them. Operating the datapath at 0.83V (instead of the nominal 1V), results in BER=10^{-4} due to voltage overscaling. For this error rate, the proposed technique achieves 3.4dB PSNR improvement compared to no correction case and approximately 1.2dB degradation compared to error-free performance for a 20% reduction in power consumption. Thus the proposed techniques allow JPEG codecs to have lower power consumption with only a mild degradation in image quality.

C. Dynamic Range Reduction in Datapath:

Reducing dynamic range is an effective technique to reduce energy consumption. However, they incur degradation in system quality. In [16], low order bit truncation is applied to motion estimation in video encoding; instead of using 8 bits, only 4 or 5 of the higher order bits are kept to reduce dynamic energy consumption by approximately 60%. The truncation based technique in [23] achieves up to 40% energy saving for DCT while causing approximately 0.2 reduction in mean structural similarity (MSSIM) (which corresponds to approximately 15dB loss in peak signal to noise ratio (PSNR)).

Contribution [17] [34]**:** We investigate use of bit truncation to reduce the power consumption of signal processing kernels such as FIR filter and DCT kernels. Low order bit truncation introduces errors which have to be compensated to minimize quality degradation. To handle the errors, we propose a compensation unit based on unbiased estimation of the truncation noise. For 4-bit truncation in DCT computation in JPEG, such a scheme achieves 23% power savings with only 0.6dB drop in PSNR. Thus, this technique is very effective in improving the PSNR performance with a small circuit overhead.

We also propose a MSB truncation scheme for sum of absolute difference (SAD) computation in motion estimation and inter-prediction. The proposed scheme uses the statistics of absolute difference (AD) and SAD computations to approximate the computations. Specifically, it exploits the fact that: i) most of the AD values are small, due to locality of current and reference blocks, and that ii) most of the large AD values are for blocks that are likely not to be selected, and thus these values can be approximated. In the proposed scheme, large AD values are detected using special logic and the corresponding SAD value updated with a correction factor. We also propose variants of this scheme based on sub-sampling which further reduce the energy consumption. The corresponding architecture has a lower critical path delay compared to the baseline SAD architecture and achieves 28% energy reduction at nominal voltage.

For iso-throughput, i.e., when the throughput is the same as the baseline architecture, it achieves 54% energy reduction while incurring less than 2% increase in compressed data rate and approximately 0.1% reduction in PSNR. If larger increase in compressed data rate is tolerable, the proposed scheme with ½ sub-sampling has approximately 90% reduction in energy consumption with ~7% increase in compressed data rate and less than 1% reduction in PSNR.

D. Hybrid Schemes for Energy Reduction in Data-path

Voltage scaling, reducing number of computation and dynamic range reduction all achieve energy savings; however their combination achieves even higher energy savings. Selective deactivation of DCT coefficient based on the operating voltage has been proposed in [35]. It has been shown that 41% to 90% power saving is possible compared to baseline scheme with up to 10dB degradation in PSNR. A similar approach is applied in [36] for color interpolation where only less important computations are affected by voltage scaling and process variation. Such a scheme achieves 40% power savings with 5dB PSNR degradation.

Contribution [34]**:** We study the combination of voltage scaling and dynamic range reduction for low pass FIR filter kernels. The errors due to increase in critical path delay during voltage scaling are reduced by truncating the lower order bits which causes a reduction in the critical path. The noise that is introduced due to truncation is compensated by using an unbiased estimator to calculate the average noise power due to truncation and compensating for it. For a MAC based FIR filter, such a scheme achieves 85% energy saving for fairly low noise level. We also study the combination of computation reduction and dynamic range reduction for low-pass FIR filter and DCT. We propose a scheme that chooses which DCT coefficients have to be deactivated and the number of bits to be truncated based on the quality metric, Q. We derive combinations of deactivation and truncation for different acceptable PSNR degradations across the whole range of Q. Simulation results show on average, 33% to 46% reduction in energy consumption while incurring 0.5dB to 1.5dB degradation in PSNR performance of JPEG.

III. Organization

The layout of this report is as follows. Chapter 2 briefly describes the schemes to reduce power consumption in CMOS circuits. It also include a brief review of the image and video codecs that were used in this work. In Chapter 3, failures due to voltage scaling in memories are investigated. Unequal error protection and algorithm specific techniques are proposed to mitigate errors in tile memory of JPEG2000 image codecs. Chapter 4 describes the voltage scaling and error compensation techniques in data-path for JPEG. An algorithm-specific technique is proposed that exploits inherent redundancy in the DCT transform of JPEG to minimize the voltage scaling induced performance degradation. Chapter 5 addresses the effects of reducing dynamic range in datapath computations. Techniques to compensate truncation noise due to low order bit truncation using unbiased estimator are presented along with a high order clipped computation scheme that is applicable for motion estimation and intra prediction schemes. Chapter 6 describes combination of different energy saving techniques such as voltage scaling, number of computation and dynamic range reduction for two case studies, FIR filtering and DCT transform in JPEG. Chapter 7 concludes the thesis.

CHAPTER 2

BACKGROUND

I. SOURCES OF POWER CONSUMPTION

There are three main sources of power consumption in CMOS circuits: dynamic (switching), short circuit current, static (leakage) as illustrated in equation (2.1) [3].

$$P = \underbrace{\alpha C_L V_{dd}^2 f_{clk}}_{dynamic} + \underbrace{I_{sc} V_{dd}}_{short\ ciruit} + \underbrace{I_{leak} V_{dd}}_{static} \tag{2.1}$$

The dynamic component is a function of $\alpha,$ the probability of transition, ie. average number of times a node makes a power consuming transition in one clock period, $C_L,$ the load capacitance, V_{dd}, the supply voltage and f_{clk}, the clock frequency. I_{sc} is the short circuit current which arises when both the pmos and nmos are on during switching, thereby conducting a direct path current from supply to ground; I_{leak} is the leakage current due to subthreshold leakage, direct tunneling gate leakage, source-substrate and drain-substrate reverse biased pn junction leakage. Of the three components, the dynamic component is the largest. For instance, in 45nm technology for an SRAM memory, it is about 55% of the total power consumption, while static power consumption is about 40% [37]. Next we describe some well-known techniques to reduce power consumption in CMOS circuits.

II. POWER REDUCTION IN DYNAMIC COMPONENT

The dynamic power consumption is given by $\alpha C_L V_{dd}^2 f_{CLK}$ and so any reduction in the switching activity or capacitance or supply voltage or clock frequency results in lower power.

A. Power Reduction by Voltage Scaling

The product of switching activity (α) and load capacitance (C_L) is defined as effective capacitance (C_{eff}). While C_L reduces with technology scaling, C_{eff} can be minimized through choice of logic function, logic style, and exploiting signal statistics [38].

B. Power Reduction by Voltage Scaling

Voltage scaling is one of the most effective techniques to reduce dynamic power consumption due to its quadratic relation as shown in equation (2.1). It is effective for reducing the power consumption of datapath, memory and even interconnects. However, it increases the latency of the circuitry and promotes delay induced errors for a given clock period. The relation between V_{dd} and latency of a circuit can be expressed as follows:

$$T_d \propto \frac{C_L V_{dd}}{(V_{dd} - V_{th})^{\gamma}} \tag{2.2}$$

where γ is 2 for long channel devices and approaches to ~1.2, 1.3 for short channel devices [38]. Delay of the device increases dramatically as V_{dd} approaches to V_{th}, which makes voltage scaling harder for scaled technologies due to smaller $\frac{V_{dd}}{V_{th}}$ ratio.

Fig. 2.1 illustrates the normalized power saving and delay increase of a 14-bit ripple carry adder (RCA) with respect to nominal voltage using 45nm PTM models [40]. When the voltage is scaled to 0.8V, there is approximately 40% reduction in power consumption of the adder and a 46% increase in the delay. Thus aggressive voltage scaling can lead to timing violations.

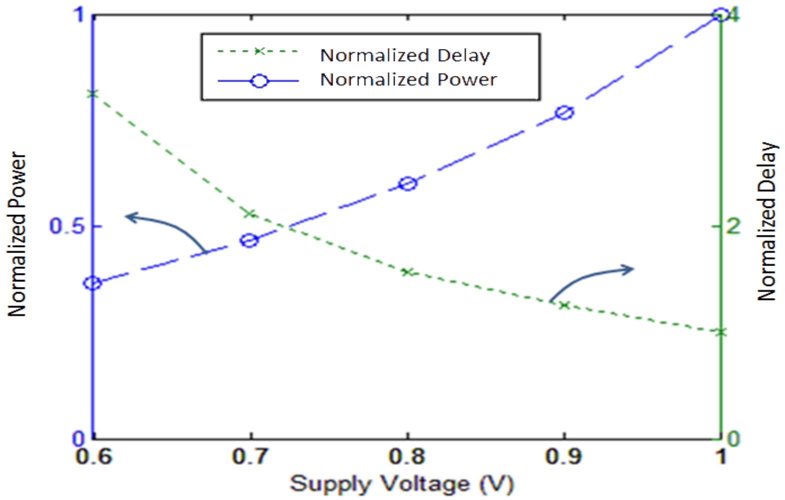


Fig. 2.1. Power and delay profiles of 14-bit RCA adder under voltage scaling

Even larger power savings can be obtained by operating the SRAM memory at scaled voltages. For instance, for a 32nm 6T SRAM memory structure whose nominal voltage is 0.9V, operating at 0.6V results in power saving of approximately 80% in write and read while leakage power also drops by more than 80%. However, such aggressive voltage scaling results in 10^{-2} BER for the SRAM due to incorrect voltage sharing and timing violations.

To exploit voltage scaling with minimal effect on overall system performance, schemes based on fine-grained multiple supply voltage (V_{dd}) levels have been introduced. For a given combinational logic, nominal voltage level (V_{dd}^{H}) is assigned for circuit components in the critical path while lower voltage level (V_{dd}^{L}) is used for components in the other non-critical paths. Level converters are placed between high and low voltage levels [41] which consume extra power and diminish the effect of multiple voltage levels. To reduce the overhead, only a few voltage levels are used and a large chunk of computing resources are operated at the same voltage [42].

C. Power Reduction by Architecture Modification

Circuit parallelization and pipelining are two popular techniques used to reduce power without throughput degradation. The throughput of a single processing unit at clocked at f Hz can be obtained by using M parallel units operating at f/M Hz. Now as operating frequency is reduced, each of the processing units has M times longer time to complete the processing. This allows the voltage to be scaled down to exploit the timing slack, and results in significant reduction in power. For instance, in 45nm technology, if two processing units are used, the voltage can be scaled to 0.7V instead of nominal voltage (1V) resulting in ~50% power saving. Adding pipeline latches in the critical path also allows for operating at lower voltages without affecting the throughput. While additional pipeline latches introduce extra capacitance, the overhead is quite small [39].

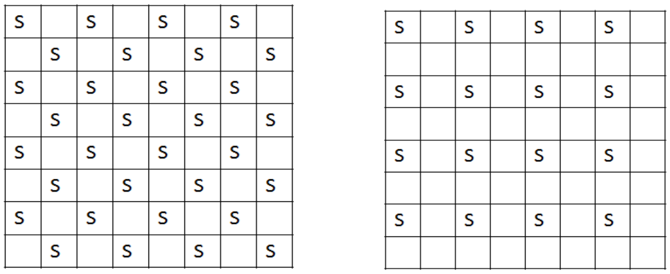
Other techniques include ordering of input signals to reduce the transition activity, which also reduces the power consumption. In [43], it has been shown that for a 9 tap FIR filter implemented on a DSP board, the power consumption can be reduced by 25 to 40% by reordering the filter coefficients. Such a reordering scheme is very effective for symmetric filters that are widely used in multimedia such as 2D-Gaussian filtering.

Resource sharing is another power reduction technique that utilizes multiplexed architectures in which multiple functional operations can be mapped to a less number of computational units. Even though this increases the switching activity, the circuit area is reduced which in turn reduces leakage power consumption.

D. Power Reduction by Algorithm Modification: Reducing Number of Computations

The number of computations can be reduced by choosing a smarter algorithm with a lower complexity. Examples include Fast Fourier Transform implementation of the Discrete Fourier Transform, differential tree search based vector quantization [3], etc. However, most of the time reduction in the number of computations comes with a performance hit. For instance, in block matching that is used in motion estimation, heuristic algorithms such as three step search and diamond search have lower complexity but sub-optimal performance. These search algorithms are used when the performance requirements are not that stringent and the available energy is low [18]. Now for each search algorithm, sub-sampling can also be used to further reduce the number of computations [6] [16] [17]. If 1/s is the subsampling ratio, then these schemes reduce the number of absolute difference computations in motion estimation by 1/s and result in significant energy reductions. Two subsampling schemes for s=2 and s=4 tailored for 8x8 block based motion estimation are illustrated in Fig. 2.2 [17]. These schemes reduce the number of AD computations by 1/s, and for iso-throughput, they also allow us to scale the voltage of the system, resulting in significant energy reduction.

Another way of reducing the number of computations with minimal effect in image quality is by exploiting the fact that different portions of the computation have different levels of significance on the overall system quality. For instance, most of the image energy of the DCT resides in the low frequency coefficients and higher frequency coefficients can be sacrificed when good enough quality is achieved [2] [20]. Similarly, in FIR filtering, larger filter taps contribute more to system performance, and so sorting the impulse response filter taps in decreasing order of magnitude and computing on larger coefficients first helps achieve energy saving with reduced overall quality degradation [20]. The concept is related to progressive transmission where data or image transmission is halted when decent quality acquired as in [19].



S		S		S		S	
	S		S		S		S
S		S		S		S	
	S		S		S		S
S		S		S		S	
	S		S		S		S
S		S		S		S	
	S		S		S		S

S		S		S		S	
S		S		S		S	
S		S		S		S	
S		S		S		S	

Fig. 2.2. Sub-sampling patterns for ½ and ¼ schemes

III. Power Reduction of Static Component

The static component of power consumption is due to subthreshold leakage, direct tunneling gate leakage, source-substrate and drain-substrate reverse biased pn junction leakage. The significance of static power consumption in todays' systems is rapidly increasing due to continually shrinking transistor geometry and larger die size.

The subthreshold current (I_{sub}) is due to weak conduction between source and drain of the MOSFET when the gate voltage is below the threshold voltage (V_{th}). It occurs due to thermal emission of carriers over the potential barrier set by the threshold [38]. I_{sub} can be expressed as

$$I_{sub} = I_0 e^{\frac{V_{gs} - V_{th} + \eta V_{ds} - k\gamma V_{sb}}{n v_T}} \left(1 - e^{-\frac{V_{ds}}{v_T}}\right) \tag{2.3}$$

where I_0 is the current at V_{th} and is dependent on geometry and process. n is a process dependent term and is typically 1.3-1.7 for CMOS. V_{ds} is the drain to source voltage, and v_T is the thermal voltage. The drain induced barrier lowering (DIBL-ηV_{ds}) effect reduces the threshold voltage of the transistor and increases the subthreshold current that results in higher static power consumption. The body biasing voltage (V_{sb}) also modulates the threshold voltage and changes the leakage current.

Techniques based on use of multiple threshold voltage levels have been used to manage leakage power. However, threshold voltage not only affects leakage power, but also affects dynamic power and latency. Thus, fine-grained multiple threshold voltage schemes use higher threshold voltage for the components that are off the critical path and nominal threshold voltage levels for those on the critical path. Body biasing is another widely used technique for leakage control. It is used to control the energy gap between the conduction bands of the gate (polysilicon) and body (SiO_2), which in turn changes the threshold voltage. Use of stacked transistors also reduces the V_{ds} value per transistor which consequently reduces the leakage current.

Gate leakage is due to tunneling of carriers from the bulk silicon to gate which is inversely proportional to gate thickness and material conductivity. With technology approaching 1nm for gate thickness, high-k materials are used to maximize the insulation between gate and body, thereby reducing the gate leakage problem [38]. Reverse biased pn junction between source/drain and body causes leakage under high electric field ($> 10^6\ V/cm$) in which electrons tunnel through the junction [41]. Changing the doping concentration of the body and/or applying reverse bias voltage are two common techniques to reduce pn junction current [38].

IV. OVERVIEW OF IMAGE/ VIDEO CODING ALGORITHMS

In this section, we review general flow of image and video processing standards such as JPEG, JPEG2000 image coding standards and H.264 video coding and their important processing kernels.

A. JPEG [44]

The general block diagram of a JPEG encoder/decoder is shown in Fig. 2.3. The original image in pixel domain is divided into 8×8 blocks which are transformed into frequency domain using 2 dimensional (2-D) DCT. This is followed by quantization, where the coefficients are scaled by factors that depend on the desired image quality and/or compression rate. Next, zig-zag scanning is used to order the 8×8 quantized coefficients into a one dimensional vector (1×64 format) where low frequency coefficients are placed before the high frequency coefficients. The entropy coder generates the compressed image using Huffman coding.

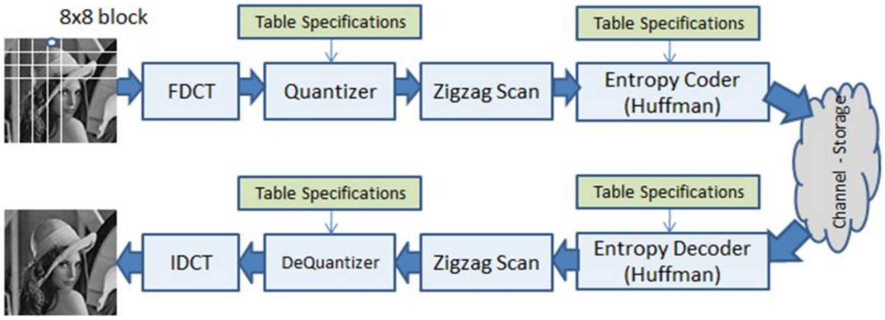


Fig. 2.3. Block Diagram of JPEG

Discrete Cosine Transform (DCT): 2-D DCT is typically implemented using 1-D DCTs along rows (columns) followed by 1-D DCT along columns (rows) as illustrated in Fig. 2.4.

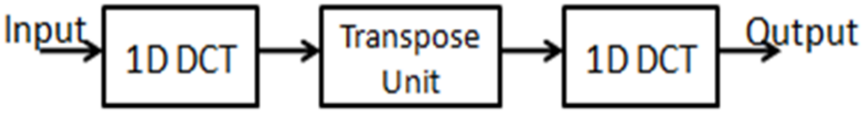


Fig. 2.4. 2D DCT architecture using 1-D DCTs

1-D DCT transform of size 8 that is used in JPEG can be expressed as follows:

$$W_i = \frac{c_i}{2}\sum_{k=0}^{7} x_k \cos\frac{(2k+1)k\pi}{16} \qquad c_i = \begin{cases} \frac{1}{\sqrt{2}} & i = 0 \\ 1 & i = 1, \ldots, 7 \end{cases} \tag{2.4}$$

where x_k's are input pixels in row or column order and W_i's are the corresponding outputs. Typically 8-point DCT is computed along rows and the coefficients stored in the transpose unit so that data for the 8-point DCT along columns can be obtained efficiently. The properties of the coefficient matrix are used to reduce the number of multiplications. We use the following method of implementing the odd and even coefficients.

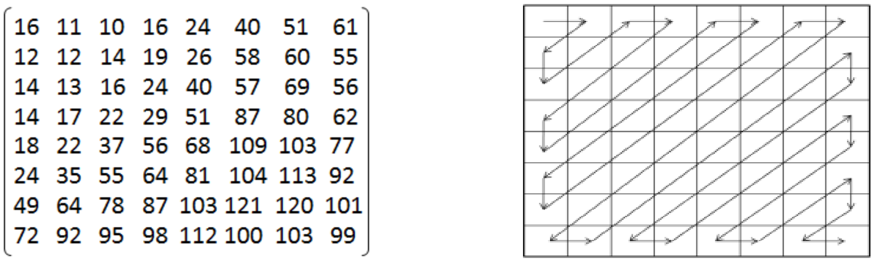
$$\begin{bmatrix} W_0 \\ W_2 \\ W_4 \\ W_6 \end{bmatrix} = \begin{bmatrix} d & d & d & d \\ b & f & -f & -b \\ d & -d & -d & d \\ f & -b & b & -f \end{bmatrix} \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix} \tag{2.5}$$

$$\begin{bmatrix} W_1 \\ W_3 \\ W_5 \\ W_7 \end{bmatrix} = \begin{bmatrix} a & c & e & g \\ c & -g & -a & -e \\ e & -a & g & c \\ g & -e & c & -a \end{bmatrix} \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \tag{2.6}$$

a= $\frac{1}{2}\cos\left(\frac{\pi}{16}\right), b = \frac{1}{2}\cos\left(\frac{2\pi}{16}\right), c = \frac{1}{2}\cos\left(\frac{3\pi}{16}\right), d = \frac{1}{2}\cos\left(\frac{4\pi}{16}\right), e = \frac{1}{2}\cos\left(\frac{5\pi}{16}\right), f = \frac{1}{2}\cos\left(\frac{6\pi}{16}\right), g = \frac{1}{2}\cos\left(\frac{7\pi}{16}\right)$. The DCT engine is implemented by 12 bit integer operations in [21] [45].

Quantizer: The rate and quality of the image is determined by the quantization level. In order to achieve different quality and compression rates, the quantization matrix is scaled by quality factor that is a function of the quality metric (Q), which ranges from 1 to 100 [46]. A lower Q results in lower image quality and compression rate. Fig. 2.5 illustrates JPEG luminance quantization table for Q=50. Note that high frequency components which are at the bottom right corner are quantized aggressively while low frequency components which are at the top left corner are mildly quantized. Fig. 2.5 also shows the zig-zag scanning order. The very first element is the DC coefficient which is encoded in differential order by subtracting the DC coefficient of the previous block and encoding the difference using a Huffman table in baseline JPEG; the rest of the coefficients are AC coefficients, which are encoded using another Huffman table specified in the standard [44].

In Chapter 4, we use the fact that two adjacent AC coefficients after zig-zag scan have similar values, and that coefficients corresponding to higher frequencies generally have smaller values. These are used to derive schemes that correct errors in the datapath.



$$\begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

(a) (b)

Fig. 2.5. a) Luminance Quantization Matrix for $Q = 50$; b) Zigzag scan order

Entropy Coder: Entropy coding of JPEG exploits the fact that most of the quantized coefficients at high frequencies consists are small numbers, if not zeros. Thus, instead of coding each zero pixel value separately, the entropy coder counts the number of zeros and uses it to specify the run-length of the zeros. Furthermore, the coder uses small codewords for small pixel values to effectively compress the non-zero pixels.

B. JPEG2000

JPEG2000 is a widely used still image compression technique. It supports progressive image and region of interest coding and has better visual quality and compression performance compared to JPEG. In fact, it achieves more than 2 dB improvement in PSNR for the same compression rate compared to JPEG [31].

The general block diagram of the JPEG2000 encoder/decoder is illustrated in Fig. 2.6. The original image (in pixel domain) is transformed into frequency sub-bands using the discrete wavelet transform (DWT) engine followed by quantization and stored into tile memory. Compressed image is obtained after embedded block coding with optimized truncation (EBCOT) processing followed by rate control. While rate control can be achieved by adjusting quantization level or by EBCOT, EBCOT provides gradual improvement (unlike quantization) and is more favorable.

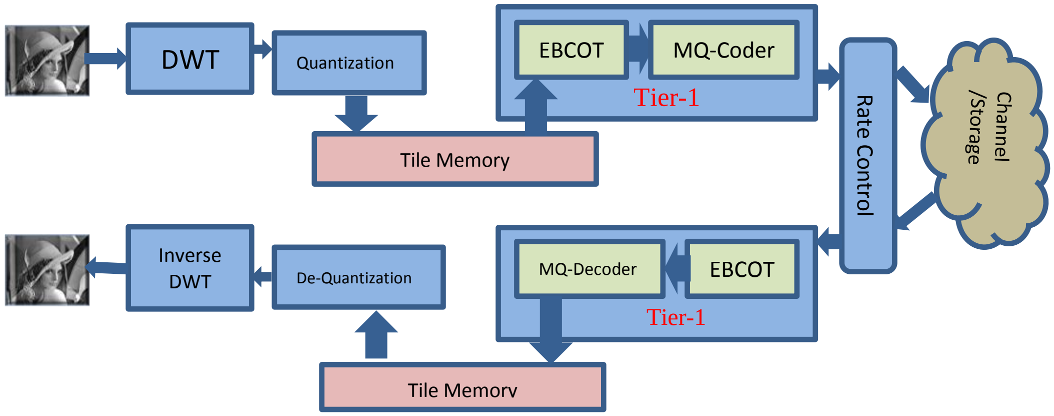


Fig. 2.6. Block diagram of JPEG2000

The 2 level sub-band representation of the DWT output is shown in Fig. 2.7-a. The input image of size NxN is processed by high-pass (H) and low-pass (L) filters along rows and columns followed by subsampling to generate the first level outputs of size N/2xN/2: HL1 (high-pass along row followed by low-pass along columns), HH1 (high-pass along rows followed by high-pass along columns), LH1 (low-pass along rows and high-pass along columns) and LL1 (low-pass along rows and columns). The LL1 outputs are further processed by high-pass and low-pass filters along the rows and columns to generate the second level outputs, HH2, HL2, LH2, and LL2, each of which is of size N/4xN/4. For a natural image, the low frequency sub-bands are likely to contain coefficients with larger values while higher frequency sub-bands contain coefficients with small values. Therefore the most significant bits of high frequency sub-bands (such as HL1, LH1, HH1) typically consist of all zeros. We exploit this fact in developing the memory error compensation techniques in Chapter 3.

The DWT outputs are processed by Tier-1 coding which exploits the redundancy of these outputs in different bitplanes. Tier-1 coding in JPEG2000 takes 32×32 or 64×64 bits from each bitplane and encodes them independently. This stage has two main sections: EBCOT and MQ-Coder. EBCOT uses context based coding for each bit in a given bit plane, MQ-Coder then calculates its probability and determines the final compressed bit stream. For the all-zero bit plane, tier-1 coding skips processing that bit plane and only adds one bit information to the header of the file. Fig. 2.7-b and c illustrate the bit plane representation and independent code blocks (32×32 and 64×64) of an image.

In most implementations, DWT coefficients are stored in a memory as shown in Fig. 2.7. In general, the tile memory is typically a SRAM because of its low latency. In scaled technologies, there may be errors in these memories as will be described in Chapter 3. As a result the DWT output that is stored in the tile memory is different from the data that is read out of the memory for EBCOT processing. In Chapter 3, we describe techniques to compensate for these errors.

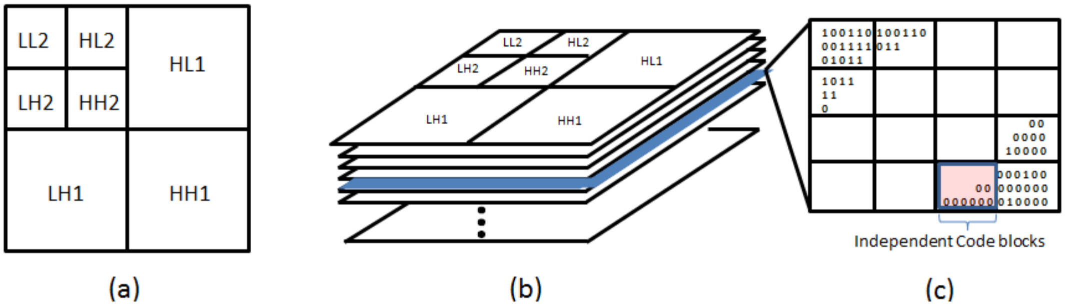


Fig. 2.7 a) Sub-band representation of DWT output b) Bit plane representation of Sub-band c) Code block representation of bit planes

C. Video Codec (H.264)

In a video codec such as in H.264 [47], frames are partitioned into smaller non-overlapping blocks typically of size 8x8 or 16x16 (macroblocks). Two types of encoding are employed for a given frame: intra (spatial) and inter (temporal). In the inter block coding, motion estimation (ME) is used to remove temporal redundancy across consecutive frames [16]. In intra block coding, pixels of the current macroblock (MB) are estimated using the adjacent pixels in the same frame.

The general block diagram of the encoding flow is illustrated in Fig. 2.8. Prediction blocks using ME and intra-prediction are computed and the best match is subtracted from the current block to remove redundancy in the video frames. The residual is transformed using 4×4 DCT followed by a quantizer with a predefined step size. Quantizer has 51 configuration each of which results in different bit rate and video quality. The remaining data is fed into an entropy coder (context adaptive variable length coding (CAVLC) or binary arithmetic coding (CABAC)) [48]. Residual frame is regenerated by back tracing the DCT and quantization steps that contracts the reference frame. In the last stage, deblocking filter removes the blocking artifacts due to DCT and quantization processes [49].

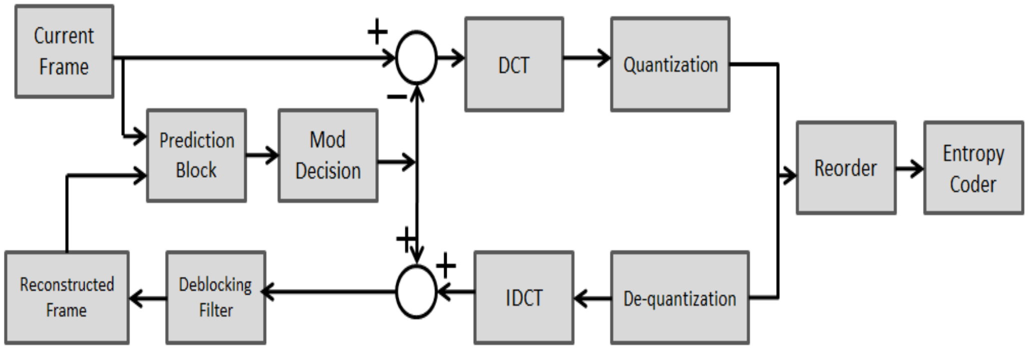


Fig. 2.8. Block Diagram of the Video Encoder

Inter Prediction (Motion Estimation): ME in video codec is used to remove the temporal redundancy in the consecutive frames. Cost functions are based on Lagrangian multiplier, sum of absolute transform difference (SATD) and sum of absolute difference (SAD) to find the best match [16]. The mode that minimizes distortion and reduces the code rate is then chosen. Due to its simplicity, the sum of absolute difference (SAD) is widely used in real time implementation. Here the absolute difference (AD) of pixels corresponding to the same location in the reference and current blocks are computed and then summed over all the pixels in the block. SAD can be expressed as:

$$SAD(P_{cur}, P_{ref}) = \sum_{m=0}^{B-1}\sum_{n=0}^{B-1}\left|P_{cur}(m,n) - P_{ref}(m,n)\right| \tag{2.7}$$

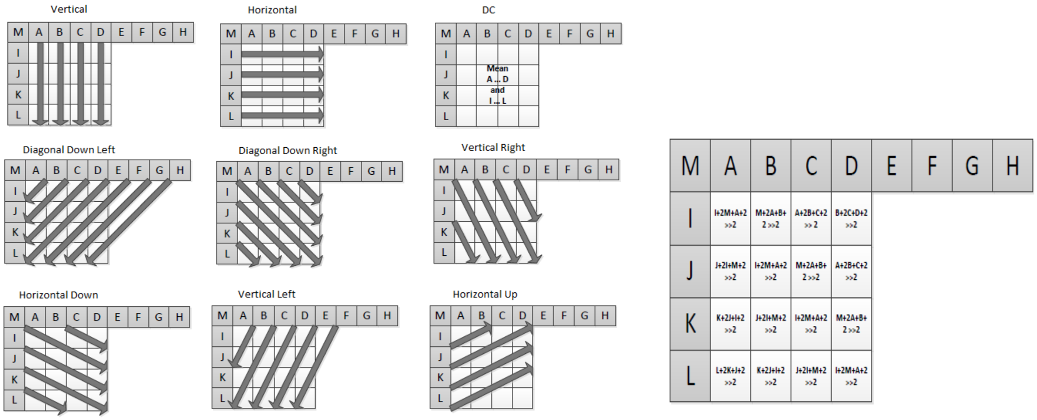
where the current block is of size $B \times B$, and P_{cur} and P_{ref} refer to current and reference blocks, respectively. The block that achieves the lowest SAD value is the block that is selected for motion compensation. In Chapter 5, we use the fact that most of the AD values are small due to locality of current and reference blocks, and that most of the large AD values are for blocks that are likely not to be selected, and thus these values can be approximated. This helps in reducing the number of SAD computations.

There are several search strategies to find the candidate blocks in a search area. Full search, where all reference block positions are searched, is computationally intensive and thus rarely used. Instead, several heuristic algorithms with lower complexity but sub-optimal performance are used. A comprehensive list of search strategies has been described in [50]. We use two popular strategies, namely, three step search (TSS) [51] and diamond search (DS) in our work [52]. The proposed techniques are essentially independent of the search algorithm and can be used in conjunction with other search methods.

H.264 supports sub-pixel motion estimation with half and quarter pixel accuracy. Following the integer ME, sub-pixels are estimated at four neighboring directions of the current pixel using interpolators. A 6-tap filter whose taps are {1,-5, 20, 20,-5, 1} is used in half-pixel and bilinear interpolation is used in quarter pixel ME of H.264.

Intra Prediction: In intra prediction mode of H.264, MBs are predicted using the adjacent MBs that are already encoded. There are two main types of prediction modes in luminance compression: Intra_4x4 (I4MB) and Intra_16x16 (I16MB) and one mode for chrominance component: Intra_8x8.

In Intra_4x4 mode, each MB is partitioned into sixteen 4x4 coding blocks. There are 9 different prediction modes for each 4x4 block that are illustrated in Fig. 2.9. These modes are called vertical, horizontal, DC, diagonal down left, diagonal down right, vertical right, horizontal down, vertical left and horizontal up where they are suited to predict regions with specified direction of intensity change. Thirteen boundary pixels from previously encoded blocks are used to generate prediction blocks (PB). Each PB has sixteen pixel values that are computed using some or all of the boundary pixels. The direction of the prediction is shown with arrows starting from boundary pixels whose weighted average generates the predicted value. For instance, predicted values for mode 4 (Diagonal down right) is illustrated in Fig. 2.9. On the other hand, predicted values for DC mode is the mean of boundary pixels L to K and A to D. Similarly, Intra_16x16 and Intra_8x8 have four combinations with different direction of prediction [47]. SAD given in equation (2.7) is widely used for mode decision in intra-prediction.



M	A	B	C	D	E	F	G	H
I	I+2M+A+2 >>2	M+2A+B+ 2 >>2	A+2B+C+2 >> 2	B+2C+D+2 >>2				
J	J+2I+M+2 >>2	I+2M+A+2 >>2	M+2A+B+ 2 >>2	A+2B+C+2 >>2				
K	K+2J+I+2 >>2	J+2I+M+2 >>2	I+2M+A+2 >>2	M+2A+B+ 2 >>2				
L	L+2K+J+2 >>2	K+2J+I+2 >>2	J+2I+M+2 >>2	I+2M+A+2 >>2				

Fig. 2.9. a) Nine prediction modes of Intra_4x4 b) Diagonal Down Right

CHAPTER 3

COMPENSATING FOR MEMORY ERRORS CAUSED BY VOLTAGE SCALING: CASE STUDY JPEG2000

I. INTRODUCTION

Voltage scaling is an effective way of reducing memory power. Unfortunately aggressive voltage scaling results in memory failures. In this chapter, we describe several ECC and algorithm-specific techniques to mitigate the effect of memory failures caused by low voltage operation of SRAM memory in JPEG2000 codecs. First, we analyze the failures in SRAM memories in Section II. Then, we propose an unequal error protection (UEP) scheme that is customized for JPEG2000 in Section III. The UEP scheme assigns ECCs with different strengths to different subbands so that the overall memory overhead is reduced with minimal effect in performance. Next, we describe four algorithm-specific techniques with different levels of complexity that do not require additional SRAM memory in Section IV and V. These techniques exploit the characteristics of the discrete wavelet transform (DWT) coefficients and use these to identify and correct the errors. They allow the codec to operate at a high performance level even when the number of memory errors is quite high.

II. POWER REDUCTION IN MEMORIES

In this section, we investigate the effects of voltage scaling on SRAM memory performance. Specifically, we analyze its effect on bit error rate (BER) of the low voltage SRAM memories.

A. SRAM Failure Analysis

There are several factors that affect the failure in SRAMs: hard errors which are mainly due to manufacturing (random dopant fluctuations (RDF), channel length, and width modulations), temperature, aging, and soft errors which are primarily due to alpha particles which cause memory cells to lose their charge. As transistor sizes scale down, the number of soft errors increases since the probability of hitting multiple transistors by the same alpha particle gets larger [12] [53] [54]. The number of soft errors also increases exponentially as voltage is scaled down. However, the overall failure rate in memories is still dominated by RDF and channel length modulation [37] [55].

RDF is the result of random distribution of dopant atoms in the transistor channel. The effect of RDF on threshold voltage is typically modeled with an additive iid Gaussian distributed voltage variation. Standard deviation of this model is highly dependent on manufacturing and transistor sizing [55]. Variance of threshold voltage of a MOSFET is proportional to $\sigma_{VT} \sim \frac{EOT}{\sqrt{L_t * W_t}}$, where EOT is oxide thickness, and L_t and W_t are length and width of the transistor, respectively. Fig. 3.1 shows the trend in threshold

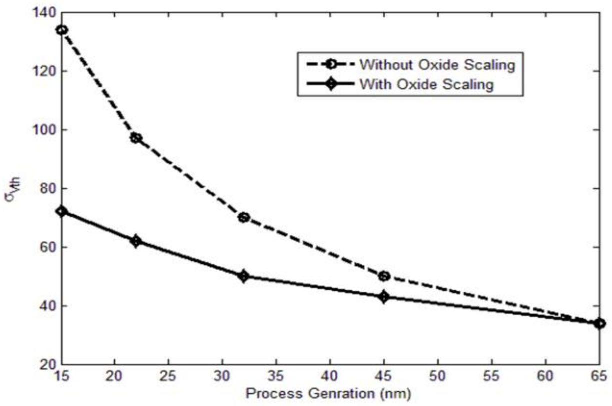


Fig. 3.1. σ_{VT} scaling trend with and without oxide scaling [37] [55]

voltage variation due to RDF in scaled technologies from [37] [55]. For 32nm, σ_{VT} is approximately between 40 to 60mV. With the help of high-K materials such as SiON, HfSiON, the thickness of oxide can be scaled down thereby reducing the variation in threshold voltage.

In addition to random variations, there are systematic variations in threshold voltage that can be modeled as correlated Gaussian random process [56]. Variance for systematic variations depends on the distance between two transistor positions in the chip. However, in this chapter, we focus on random variations since they are the major source of cell failures in SRAM. The effect of other parameters (such as temperature, aging, etc.) can be modeled as an increase in the threshold voltage variation.

Next, we describe the causes of errors in a typical 6T SRAM cell illustrated in Fig. 3.2. NCR and NCL are control transistors which are activated in the case of a read and write operation. BL and BLR lines, which have very high capacitive values, are used to retrieve or store a bit into the cell. These lines are precharged to the same value at read and opposite values at write cycles. There are three main factors that contribute to overall hard error rate: read, write and access failure [28] [57].

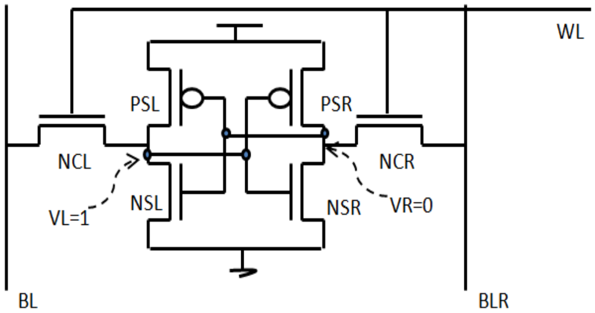


Fig. 3.2. Schematic of a SRAM cell

During read operation after control signal (WL) is turned on, voltage division occurs between one of the control transistors and the NMOS in the same side. Read failure is the result of unbalanced voltage sharing at the read node (VR - Fig. 3.2) which causes flipping of the stored bit in the cell. Increase in V_{th} of NMOS transistors NSR, NSL and/or reduction in V_{th} of control transistors NCL, NCR increase the probability of read failure.

During write operation, bit lines are charged to opposite values depending on bit value that is stored. In order to have a successful write, the node between PMOS and control transistor (VL –Fig. 3.2) should be reduced below a voltage level. Increase in V_{th} of control transistors and/or drop in V_{th} of PMOS devices increases the probability of failure in the write operation.

Increase in V_{th} of control transistors and NMOS devices can cause access time violations. However, for this application, the delay constraint is relaxed and access time violation is not an issue. So in the rest of the paper, we consider total error rate to be the union of read and write failure. For instance, Fig. 3.3 illustrates the delay variation distributions of an SRAM cell when the voltage is scaled from nominal (0.9V) to 0.8V while the target delay is 40ps.

SRAM failure analyses have been investigated by several researchers [57] [28] [58] [55] [27] [10]. In [57] and [28] statistical models of RDF are used to determine read, write failure and access time variations. One bit cell failure probability is considered and long channel transistor models are used to derive analytical current equations. In [58], read and write noise margin of 6T SRAM structure is used to calculate reliability of the memory. Importance sampling is applied in [27] to reduce the sample size and improve the speed of simulations. In [55], statistical techniques are used to model the tail distribution of access time when the error rates are very low. In [10], read failure rate of 6T SRAM structure at nominal (typical) corner for 65nm technology is shown to differ from 10^{-8} to 10^{-3} when voltage is dropped from nominal value of 1V to 0.6V. Moreover, at scaled voltage levels, write failure rate is evaluated to be 10^{-3} for the same technology.

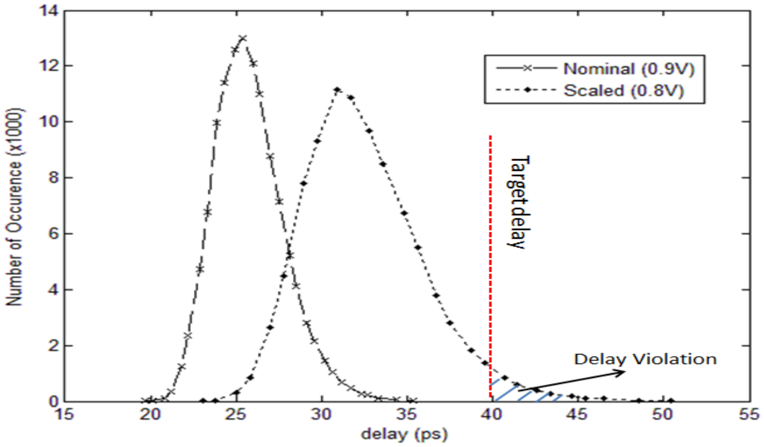


Fig. 3.3. Delay distribution of a single SRAM read operation

We studied the two types of failure due to RDF and channel length variation for 32nm technology using Hspice with high performance predictive technology models (PTM) from [40]. A SRAM cell with bitline load value equal to that 256 cells with half of them storing ‘1’ and the other half storing ‘0’ is simulated using Monte Carlo simulations. The overall BER for different levels of RDF is calculated for 5% channel length variation at different supply voltages. Each transistor is sized using minimum length (L=32nm). To estimate failure rates, we follow a procedure similar to that given in [57].

Fig. 3.4 illustrates the read, write and total failure rates for σ_{VT}=40mV from 0.8V to 0.6V. At the nominal voltage of 0.9V, the BER is estimated to be 10^{-10}. At lower voltages, the BER are very high. For instance, at 0.7V, the BER is 10^{-4} and at 0.6V, it climbs to 10^{-2}. Such high error rates were also reported in [10] [57].

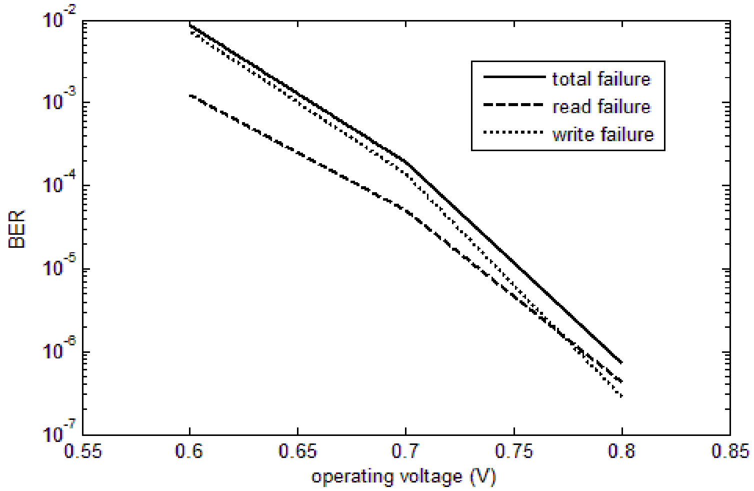


Fig. 3.4. Read, write and total failure probability of SRAM in 32nm technology for different voltage levels when σ_{VT}=40mV

Operating at low voltage levels saves both dynamic and static power. Our simulations show that the read (write) power of the SRAM cells (without read/write circuitry) can be reduced as much as 52% (71%) when the voltage is scaled to 0.7V and 72% (84%), when the voltage level is scaled to 0.6V. This is at the expense of a significant increase in the number of memory errors; techniques to compensate for these errors will be described in the following sections

B. Previous Work on Compensating for SRAM Errors

Several circuit, system and architecture level techniques have been proposed to mitigate and/or compensate for memory failures. At the circuit level, different SRAM structures such as 7T, 8T and 10T have been proposed [26] [27] [59] [60]. In 8T and 10T structures, data path for read and write operation are separated to increase the robustness of the operations. However, additional circuitry increases leakage power and circuit area by approximately 20% to 30%. To reduce the impact of variations and improve stability, boosting and suppressing voltage levels for read and write operations have been described in [37]. However, the additional voltage converters and controlling circuitry increase the complexity of the system.

Architecture level techniques have been proposed to avoid faulty bits using defect maps in [28] [61] [62]. In [28], 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 error-free locations. Row and column redundancy have been effectively used for low error rates in [61]. For high error rates, the memory area overhead is significant and it is no longer a viable option. In [62], 25% to 50% of cache memory capacity is traded-off for reliable operation at very low voltage levels. The method involves identifying and disabling defective portions of the cache at word level and bit-level granularities.

Many techniques make use of error control coding (ECC) [12] [29] [63]. In [12], orthogonal latin square codes are used to trade-off cache size with correction capability. Extended Hamming codes which provide single error correction, double error detection (SECDED) have been used for several years to combat failures in memory systems [29]. Their simple structure makes them appealing for applications that require low latency and power consumption. To increase the error correction capability, a two dimensional coding (product code) is used in [37]. Although this scheme can handle errors that occur within a certain square shaped memory region (smaller than 32x32), correction capability degrades when errors are randomly distributed over the whole memory.

More recently, algorithm-specific techniques have been developed for JPEG2000 [13], MPEG-4 [10]. These techniques mitigate system degradation due to memory failures using additional features that are intrinsic to the algorithm. In [13], binarization and second derivative of the image are used to detect error locations in different sub-bands in JPEG2000. These 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 [10] uses a combination of 6T and 8T SRAM cells based on read reliability, area and power consumption and applies it to a MPEG-4 implementation. The method in [11] is a general technique where the memory banks that store MSBs operate at a different voltage level than the ones that store LSBs. This is shown to achieve 45% power reduction with slight degradation in image quality.

III. PROPOSED UNEQUAL ERROR PROTECTION FOR JPEG2000

In this section, we first study the use of ECC in combating errors in memories in JPEG2000 codecs. 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 with memory area increase of 21.9% followed by (72, 64) with an area increase of 12.5%, and (139, 128) with an area increase of 7%. The memory overhead of an (n, k) ECC code is defined as $\frac{n-k}{k}$ where k represents the number of information bits for an ECC codeword length of n.

Since the memory area overhead of the stronger codes is very large, we propose to use unequal error protection (UEP) in the tile memory. By using a combination of strong and weak codes, the area overhead can be reduced without sacrificing performance. Note that UEP has been investigated earlier in the context of wireless transmission of images [64] [65]. The main idea of UEP is to provide superior protection for the more important bits. For instance, in JPEG2000 the higher subband DWT outputs are more important and so should be protected better with stronger codes.

In order to quantitatively measure the importance of a bit, we introduce ΔDE which is quality degradation due to bit failures in memory. This is the same for all images and is solely a function of the subband level and location of the error bit position in the subband coefficient. Errors in high level subband coefficients cause a larger degradation in image quality, as expected. In a particular subband, an error in the k^{th} bit position of a coefficient affects the value by 2^k and the ΔDE due to an error in the k^{th} bit position is 2 times as large as that due to an error in $(k-1)^{th}$ bit position.

Fig. 3.5 plots the normalized ΔDE for different subband outputs of a 3-level DWT as a function of a 1 bit error in different bit positions. The values are normalized with respect to maximum ΔDE of the LL3 subband. Bits whose failures result in higher ΔDE are clearly more important in terms of image quality. Thus, level-3 subband outputs are the most sensitive to bit errors and should be protected with stronger codes. Furthermore, as seen from the figure, errors in MSB-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 level 1 outputs (HL1, LH1 and HH1). We use the same strength code for the bits that generate similar ΔDE.

In a system that uses 3 codes, we break Fig. 3.5-a 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 weakest code (137, 128) for the rest of the points. Fig. 3.5-b illustrates the same figure in logarithmic scale with additional horizontal lines to indicate the possible locations of line-1 and line-2. In the proposed method, we choose to employ 8 settings starting from the highest ΔDE, since having more than 8 settings have little effect on overall image quality. The method can be easily extended to accommodate larger number (strength) of ECCs, different levels of DWT and larger number of ΔDE settings.

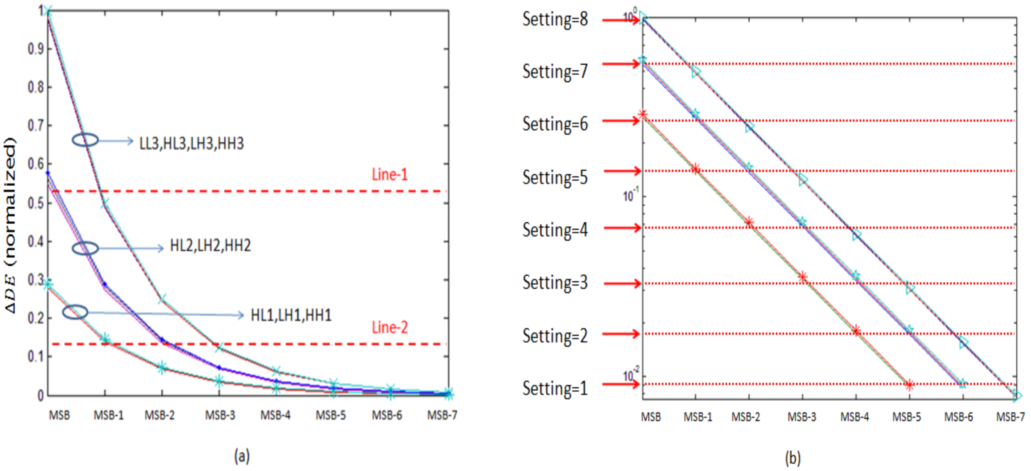


Fig. 3.5. ΔDE (normalized) due to error in different bit positions for different levels of DWT a) Linear b) Logarithmic Scale

The optimal settings of line-1 and line-2 depend on memory size overhead and quality degradation. Overall memory size overhead (MO) is sum of memory size overheads due to each ECC scheme, $MO_{overall} = (MO_{39,32} \cup MO_{64,72} \cup MO_{137,128})$. Similarly, overall degradation is sum of ΔDE degradations due to use of (39, 32) above line-1, use of (72, 64) in the region between line-2 and line-1 and use of (137, 128) in the region below line-2.

$$\Delta DE \ = \Delta DE_{39,32}(line_1) + \Delta DE_{72,64}(line_1, line_2) + \Delta DE_{137,128}(line_2) \qquad (3.1)$$

where

$$\Delta DE_{39,32}(line_1) = BER_{39,32} * \left[\sum_{m=line_1}^{8} 2^m + \sum_{n=line_1}^{7} 2^n + \sum_{k=line_1}^{6} 2^k\right] \quad (3.2)$$

$$\Delta DE_{72,64}(line_1, line_2) = BER_{72,64} * \left[\sum_{m=line_2}^{line_1-1} 2^m + \sum_{n=line_2}^{line_1-1} 2^n + \sum_{k=line_2}^{line_1-1} 2^k\right] \quad (3.3)$$

$$\Delta DE_{137,128}(line_2) = BER_{137,128} * \left[\sum_{m=1}^{line_2-1} 2^m + \sum_{n=1}^{line_2-1} 2^n + \sum_{k=1}^{line_2-1} 2^k\right] \quad (3.4)$$

and $BER_{39,32}$, $BER_{72,64}$ and $BER_{39,32}$ are the coded bit error rate of codes (39, 32), (72,64) and (137,128), respectively. The three summation terms in these equations correspond to the three subbands where the first term corresponds to the ΔDE due to errors in level-3 subband, the second term corresponds to the ΔDE due to errors in level-2 subband, and so on. More specifically, $\sum_{m=line_1}^{8} 2^m$ in equation (3.2) models errors in level-3 subband that is obtained by adding the contributions of all settings from 8 to line-1. We study 3 problem scenarios.

Scenario 1-F*ixed Overhead (UEP-1)*: For a given tolerable overhead, we find the setting of line-1 and line-2 that provides the minimum degradation. The optimization problem can be formulated as:

$$\min_{line_1, line_2} \Delta DE \quad subject\ to\ MO_{overall} \le M_f \quad (3.5)$$

where M_f is the given memory overhead. This scheme can achieve better performance as compared to the fixed ECC method.

Scenario 2 - *Fixed Performance Loss (UEP-2)*: For a given tolerable degradation, we find the setting of line-1 and line-2 that provides the minimum overhead. The optimization problem can be formulated as:

$$\min_{line_1, line_2} MO_{overall} \quad subject\ to\ \Delta DE \le S_f \quad (3.6)$$

where S_f is the given performance degradation.

Scenario 3 - ***Combination of Performance Loss and Overhead:*** We define a new metric to combine the two factors (memory overhead and performance loss) which is expressed as $combined\ metric = \Delta DE\ * MO_{overall}$.

Example UEP-1**:** Consider an example when BER= 10^{-3} and the given memory overhead is 0.125. Both (72, 64) and (137,128) codes can be used, though (72, 64) provides better performance. Fig. 3.6 illustrates the memory overhead and performance

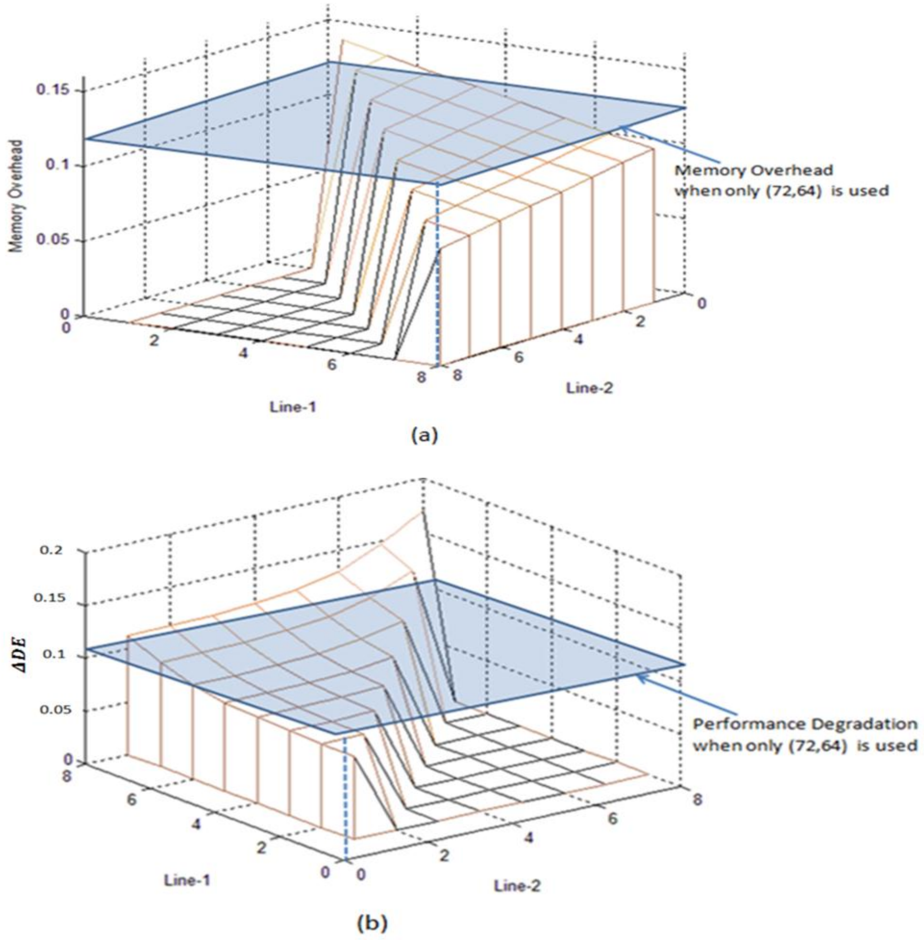


Fig. 3.6. a) Overhead and b) Performance Degradation for different settings of line-1 and line-2.

degradation for different settings of line-1 and line-2. Since line-2 cannot be over line-1 those setting are set to zero. From Fig. 3.6-a, we see that there are several UEP schemes that have lower overhead than that of (72, 64) code. Among these schemes, the one that has the minimum ΔDE degradation is chosen using the performance degradation curve shown in Fig. 3.6-a. Specifically, setting 3 of line-1 and setting 2 of line-2 provides the lowest ΔDE with memory overhead close to 0.125. This leads to use of (39, 32) code from MSB to MSB-5 of level-3 DWT coefficients, from MSB to MSB-4 of level-2 DWT coefficients and from MSB to MSB-3 of level-1 DWT coefficients. The (72, 64) code is used for MSB-6 of level-3 DWT coefficients, MSB-5 of level-2 coefficients and MSB-4 of level-1 coefficients and the (137, 128) code is used for the remaining bits of all coefficients. Using this scheme, ΔDE drops by 35% compared to when only (72, 64) is used, which is equivalent to 1.87 dB improvement in image quality.

***Example UEP-2*:** Next we compare the memory overhead of the (72, 64) code and our proposed scheme for $\Delta DE = 0.11$. Using ΔDE curves illustrated in Fig. 3.6-b, we find the minimum overhead occurs for setting 7 of line-1 and setting of 5 for line-2. Thus the (39, 32) code is used for MSB and MSB-1 of level-3 DWT output and MSB of level-2 DWT output, the (72, 64) code is used for MSB of level-2 and rest of the bits are coded using the (137, 128) code. This configuration results in 29% reduction in memory overhead compared to that of the (72, 64) code.

Finally, Fig. 3.7 illustrates the result of combined metric when BER=10^{-3}. Using this scheme we find the optimal solution corresponds to setting 5 of line-1 and setting 4 of line-2, which translates to 17% reduction in memory overhead and 26% improvement in ΔDE .

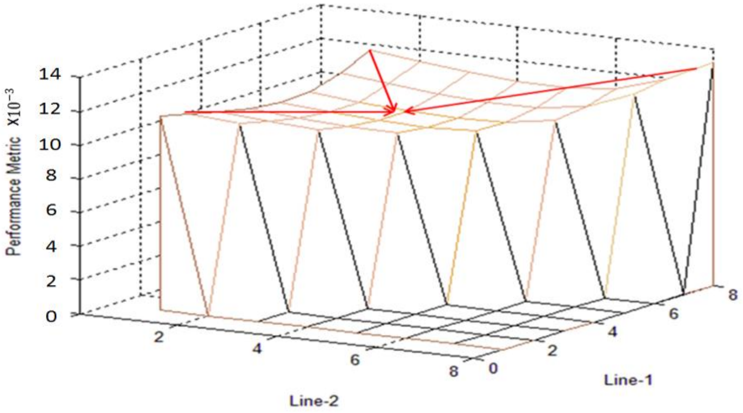
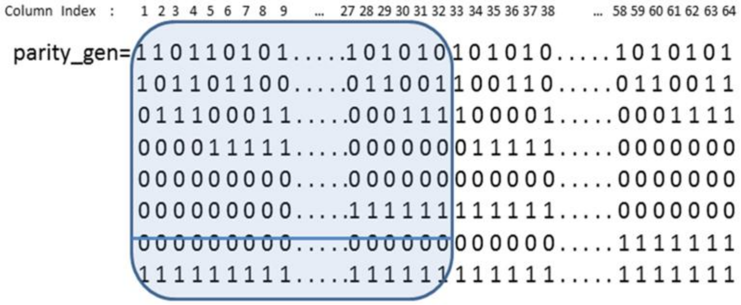


Fig. 3.7. Performance Degradation x Overhead for different settings of line1 and line2

In order to support UEP schemes with reduced circuit overhead, we derive stronger codes from weaker codes. Essentially, the parity generator matrix for shorter code (stronger) can be derived from the parity generator matrix of the longer code (weaker). This can be utilized to design hardware that can be shared for multiple codes. Consider the parity generator matrix of the (72, 64) code illustrated in Fig. 3.8. It consists of 8 rows (equal to number of parity bits). The first half of this code (column 1 to 32) except the seventh row can be used to generate the parity matrix of (39, 32) code since the seventh row consists of all zeros. Using the parity generator matrix, systematic codes can easily be generated by appending the parity part to the original information sequence. Although we need additional circuitry compared to SECDED implementations



```
Column Index  :  1 2 3 4 5 6 7 8 9   ...   27 28 29 30 31 32 33 34 35 36 37 38   ...  58 59 60 61 62 63 64
parity_gen=1 1 0 1 1 0 1 0 1.....1 0 1 0 1 0 1 0 1 0 1 0.....1 0 1 0 1 0 1
           1 0 1 1 0 1 1 0 0.....0 1 1 0 0 1 1 0 0 1 1 0.....0 1 1 0 0 1 1
           0 1 1 1 0 0 0 1 1.....0 0 0 1 1 1 1 0 0 0 0 1.....0 0 0 1 1 1 1
           0 0 0 0 1 1 1 1 1.....0 0 0 0 0 0 0 1 1 1 1 1.....0 0 0 0 0 0 0
           0 0 0 0 0 0 0 0 0.....0 0 0 0 0 0 0 0 0 0 0 0.....0 0 0 0 0 0 0
           0 0 0 0 0 0 0 0 0.....1 1 1 1 1 1 1 1 1 1 1 1.....0 0 0 0 0 0 0
           0 0 0 0 0 0 0 0 0.....0 0 0 0 0 0 0 0 0 0 0 0.....1 1 1 1 1 1 1
           1 1 1 1 1 1 1 1 1.....1 1 1 1 1 1 1 1 1 1 1 1.....1 1 1 1 1 1 1
```

Fig. 3.8.Parity generation of (39, 32) code from (72, 64) code which are optimized for a single code, generating codes in this manner allows us to adjust coding strength as required.

One drawback of the proposed UEP scheme is that it requires additional memory. Next we propose the algorithm-specific techniques with different levels of complexity that do not require additional SRAM memory.

IV. Proposed Algorithm-Specific Techniques for JPEG2000

It is well-known that in natural images neighboring pixels are highly correlated. It turns out that in the frequency domain, neighboring DWT coefficients have certain similarities. We find that, for a natural image, DWT outputs at high subbands (higher frequency) typically consist of smaller values and thus contain small number of non-zero bits in MSB planes. Fig. 3.9 illustrates the number of non-zero bits in higher bitplanes for different subbands in Lena and Bridge images. We see that for both images, the number of non-zero bits upto MSB-3 is very small. Furthermore, there is a similarity between magnitudes of neighboring coefficients. For instance, the non-zero values in the HL1 and HL2 subbands of the Lena image shown in Fig. 3.10 correspond to edges and the edges are connected. Thus isolated non-zero bits are unlikely and can be used to flag errors.

The rest of this section describes JPEG2000-specific techniques to mitigate the impact of memory errors on system quality. Unlike other circuit level and ECC methods, these schemes introduce little circuit overhead and no-additional data storage. For high frequency sub-bands such as HL, LH and HH, we propose four methods. For LL sub-band, we propose a low pass filtering technique to mitigate errors the errors.

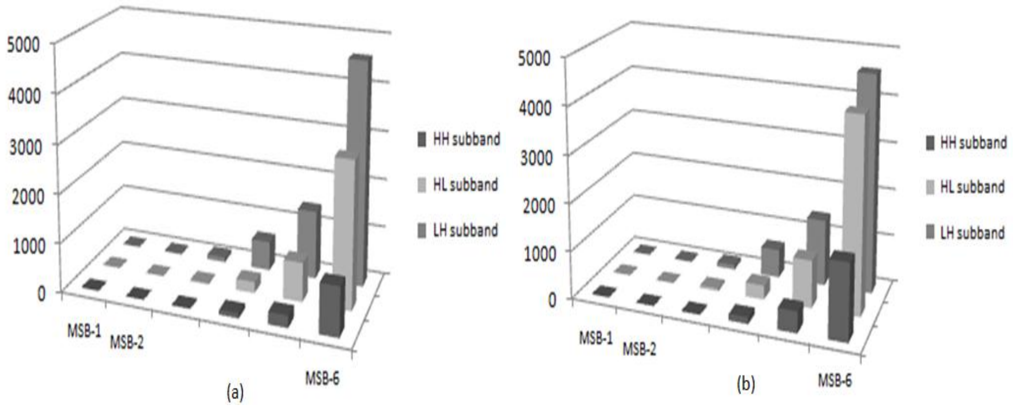


Fig. 3.9. Number of non-zero bits in different subbands in a) Bridge, b) Lena images

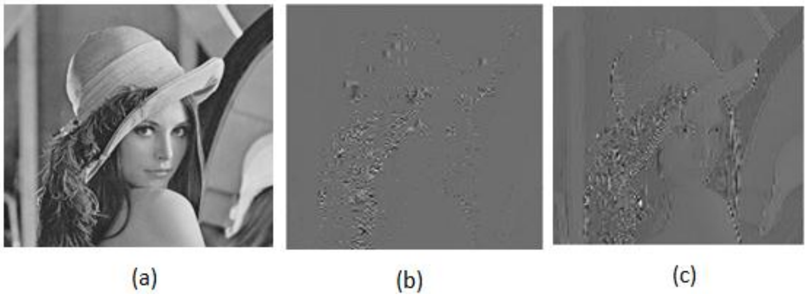


Fig. 3.10. Lena Image. a) Original, b) HL1 subband and c) HL2 subband

A. 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 4 MSB planes. Furthermore, since EBCOT skips coding bit planes that consist of all-zero bits, this technique helps reduce the power consumption due to EBCOT.

B. Method-2

Although Method 1 is very simple, there can be false erasures in MSB planes resulting in loss of quality. Method 2 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 plane that is not erased.

C. 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 3x3x4 neighborhood. Specifically, if a non-zero bit is detected in (k, l) position of the i^{th} plane, it checks the 3x3 block of bits around the (k, l) position in the $(i+1)^{th}, (i-1)^{th}$ and $(i-2)^{th}$ planes in addition to its 8 neighbors in the i^{th} plane. If it detects another 1 within this 3x3x4 search group as illustrated in Fig. 3.11, 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.

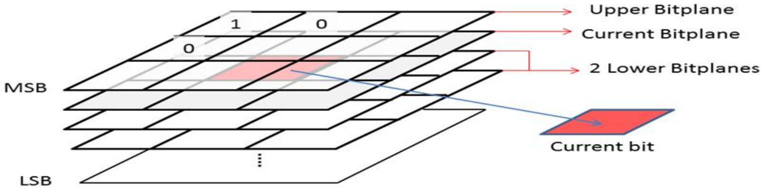


Fig. 3.11. Bit planes used in error detection mechanism

D. Method-4

So far, we have assumed that failures are uniformly distributed across the memory. However it is anticipated that with the reduction in transistor sizes, the number of burst errors will increase [12]. In order to mitigate burst errors, we modify Method-3 so that it can handle both burst and random errors. The steps are the same as that of Method 3 except that if the other non-zero bit in the 3x3x4 group is consecutive to the current bit, then an error is flagged for the current bit.

V. SIMULATION RESULTS FOR THE PROPOSED TECHNIQUES

A. System Performance 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) and change in compression rate measured as the number of bits required to represent one pixel (bpp). If C is the original frame and R is the reconstructed frame, then for frame of size MxN, the PSNR is given by

$$\mathrm{PSNR} = 10\log\left(\frac{\max(\mathrm{C}^2)}{\mathrm{MSE}}\right), \qquad \text{where}$$

$$MSE = \frac{1}{MN}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left(C_{ij} - R_{ij}\right)^2$$

We compare the performance of the algorithm-specific methods and ECC codes (39, 32) and (72, 64). We do not show the results obtained by the UEP schemes because they are as best as their strongest constituent code. In order to evaluate the performance, we implemented (9, 7) lossy JPEG2000 with 3 level DWT and 32x32 EBCOT block size. We study quality vs. compression rate characteristics. Quality is measured with respect to the reconstructed image using PSNR and the compression rate is calculated in terms of bits per pixel (bpp) based on the final compressed image 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 10^{-4} to 10^{-2} which is compatible with the BER results obtained for 6T SRAM under voltage scaling as described in Section 2.

We consider two error models:

- Model-1: represents fully random failures where occurrence of each failure is independent.
- Model-2: represents burst failure scenarios which are anticipated with reduction of transistor sizes and increase in soft error rates [12]. The burst error model is characterized by probability density function (pdf) of number of failures given by: $f_e = 0.7 * \delta(e-1) + 0.27 * \delta(e-2) + 0.03 * \delta(e-3)$, where probability of an error in one bit is 0.7, errors in two consecutive bits is 0.27 and errors in three consecutive bits is 0.03.

Case 1: BER = $\mathbf{10^{-4}}$

Fig. 3.12-a illustrates the system performance using Lena image when BER=10^{-4} for fully random error model. If some amount of image quality degradation is tolerable (below 0.5dB for 0.5bpp), 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 error-free performance at this level. Fig. 3.12-b illustrates the system performance, when burst error model is used. There is a very slight degradation for (39, 32) code; however it still provides almost error-free performance.

Case 2: BER = 10^{-3}

The performance curves show a different trend when BER increases to 10^{-3}, as shown in Fig. 3.13 for Lena image. Both for random and burst error models, algorithm-specific methods can provide good results for low and medium quality/rate regions. Method-3 follows the error-free curve very closely in the range 0.2 to 1bpp. For example, it improves quality approximately by 4dB at 1bpp rate compared to the no-correction case. If some degradation is tolerable, Methods 1 and 2 are good candidates when compression rate is above 0.8bpp, since they have lower complexity compared to Method 3. The performance of the ECC schemes: (72, 64) and (39, 32) are also illustrated in Figure 19-a. Even though they provide very good performance for low quality images, their performance diverge from error-free case for higher quality scenarios. Furthermore, their performance become worse when burst error model is used as illustrated in Fig. 3.13-b. Even the best ECC scheme (39, 32) has weaker performance compared to Method 4.

Case 3: BER = 10^{-2}

Fig. 3.14 illustrates the performance when BER=10^{-2}. Algorithmic methods improve the quality by 3 to 8dB for compression rate around 0.5bpp. Although Method-1 and 2 do not improve performance much for low rates, they can provide acceptable performance with much lower overhead. Methods 3 and 4 follow the no-error curve closely under 0.25bpp compression rate. For the mid-quality range, they improve the performance noticeably and achieve the best quality among all techniques. Thus by using algorithm specific techniques, we can achieve approximately 8dB gain in performance (compared to no-correction case) at 1bpp for both burst and random error scenarios.

We also investigate the performance of ECC using the strongest code, namely (39, 32). The performance is not good for fully random error model, and deteriorates for burst error model. Method 3 achieves 4dB better quality than (39, 32) at 1bpp for fully random error model and Method 4 can achieve 5dB better quality for burst error model.

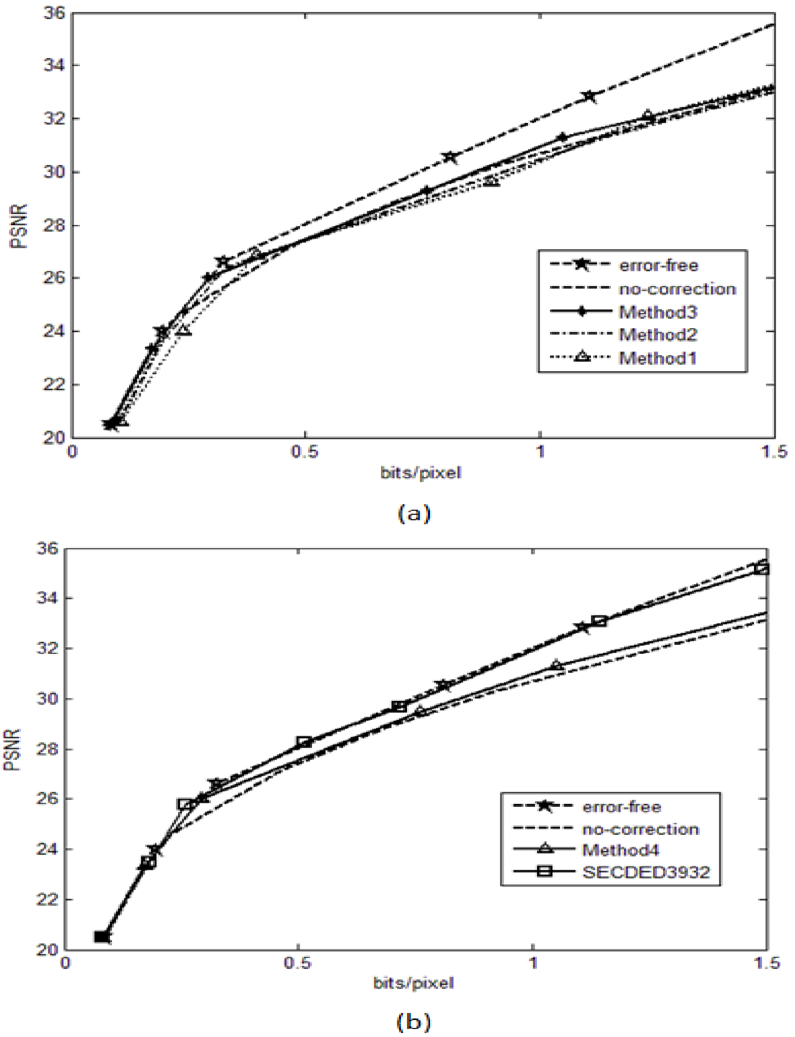
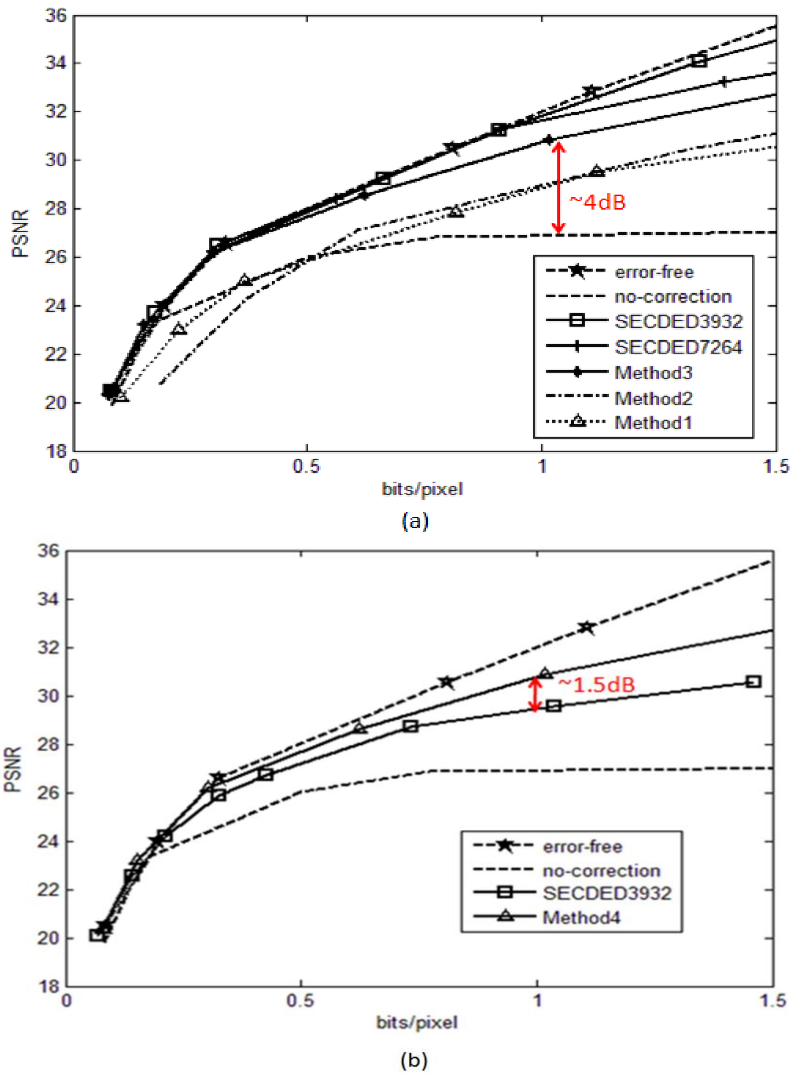


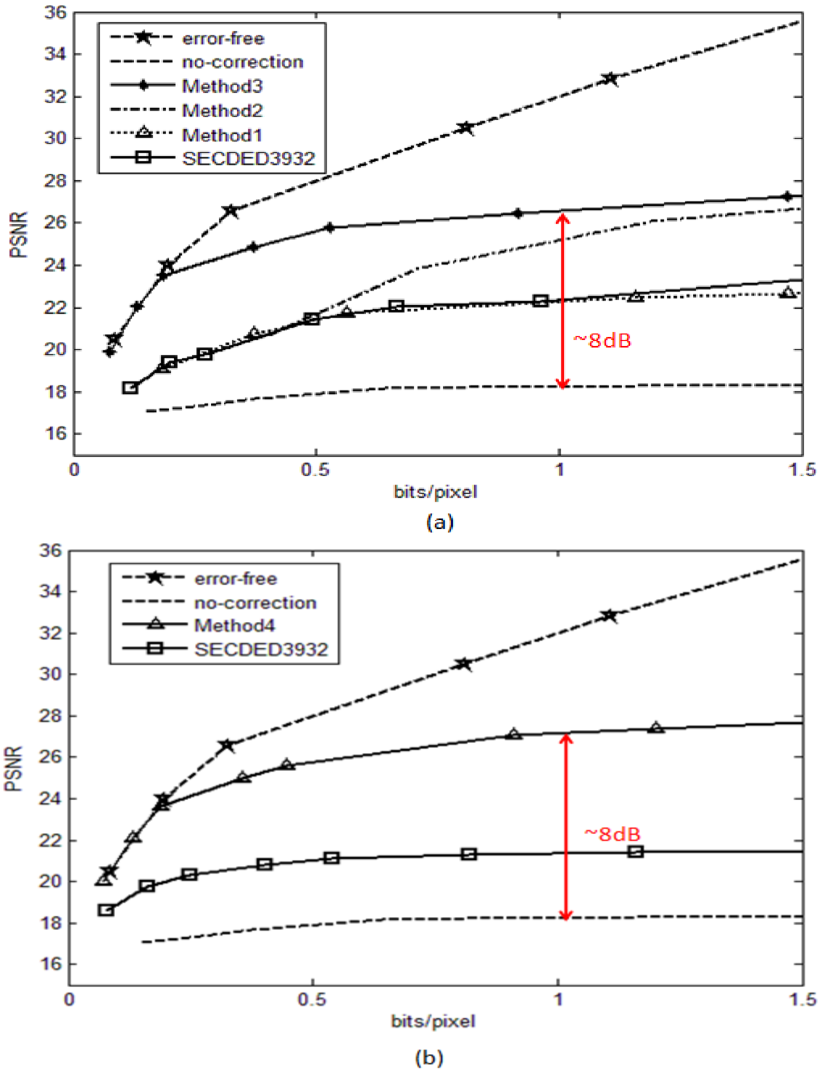
Fig. 3.12. Quality vs. Compression rates when BER=10^{-4} for Lena a) for error model-1 b) for error model-2



(a)

(b)

Fig. 3.13. Quality vs. Compression rates when BER=10^{-3} for Lena a) for error model-1 b) for error model-2



(a)

(b)

Fig. 3.14. Quality vs. Compression rates for Lena when BER=10^{-2} using algorithmic techniques a) for error model-1 b) for error model-2

Table 3.1 lists the performance of all methods for 0.75 bpp when BER=10^{-2} for random and burst error models. The memory overhead constraint for UEP-1 is 0.125 and the performance constraint for UEP-2 is ΔDE (normalized) =5x10^{-3}. From the table, we see that Method 3 (for random errors) and Method 4 (for burst errors) can provide approximately 8dB improvement compared to the no-correction case, and their average degradation is around 3dB compared to the no-error case. Method 4, which has been optimized for burst errors, has an average of 1.9dB 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.

Table 3.1. PSNR values of different techniques at 0.75bpp compression rate for BER=10^{-2}, using error models 1 and 2

	Error Model	Lena	Fruits	Baboon	Cameraman
Original	1&2	29.9	27.5	23.4	28.3
No Correction	1	18.2	15.6	12.3	16.3
	2	18.1	15.6	12.2	16.2
UEP-1	1	22.2	19.3	15.5	21.4
	2	21.4	18.5	14.8	20.1
UEP-2	1	20.9	18.2	14.6	20.3
	2	20.1	17.4	13.6	19.3
Method 1	1	21.9	18.2	14.6	20.4
	2	22	20.1	14.7	20.2
Method 2	1	23.8	20.7	18.2	22.6
	2	23.5	20.8	18.2	23.0
Method 3	1	26.1	25.7	20.3	25.3
	2	24.2	23.3	19.1	23.1
Method 4	1	25.6	24.8	19.7	24.5
	2	26.3	25.4	20.5	25.9

B. Overhead: Area, Delay, Power Consumption

The power consumption and latency circuitries that are used in algorithm-specific techniques and ECC are obtained using Design Compiler from Synopsys and 45nm models from [66].

Algorithm-Specific *Methods*

Circuitry: The simplest scheme, Method-1, requires no additional circuitry; therefore it does not add any overhead. Both Methods-2 and 3 require counter circuitry. Method-3 also requires an all-zero detector for comparing with the neighboring bit values. Method-4 requires an additional comparator that is used to detect burst errors; however it only takes 4 1-bit inputs corresponding to neighbors of the current bit in the same bit plane. Table 3.2 summarizes the necessary components that are used in different algorithm specific methods.

Table 3.2. Necessary modules for algorithm-specific methods

	9-bit counter	All-zero Detector	4-bit Comparator
Method-1	•	•	•
Method-2	✓	•	•
Method-3	✓	✓	•
Method-4	✓	✓	✓

As mentioned in Section IV, the counter circuit is not triggered much. For instance, for images sizes of sizes 256x256 with BER of 10^{-4}, 10^{-3}, and 10^{-2}, dynamic threshold value for the counter ranges between 3 and 320. Therefore, in order to support a 256x256 image with BER=10^{-2}, we need a 9 bit counter. Larger counters would be needed to support large image sizes; for instance, a 11-bit counter for 512x512 images.

The power consumption and latency of the 9-bit counter used in Methods 2 and 3, the all-zero detector used in Method 3 and the 4-bit comparator used in Method-4 are obtained using Design Compiler from Synopsys and 45nm models from [66]. Table 3.3

illustrates the power consumption, area and latency results for the different components when the clock period is 2ns.

Table 3.3. Area, latency and power consumption overhead of circuit components used in the algorithm-specific methods

	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

Memory Core Power Saving: Unlike ECC based techniques, the algorithm-specific methods do not require any additional memory. Furthermore, they can achieve full power reduction due to voltage scaling. For instance, if Method 3 is invoked to compensate BER=10^{-2} obtained by operating at 0.6V, the leakage energy of the SRAM core would drop by 82% and the active energy would drop by 85%.

ECC Methods:

Circuitry: To support UEP schemes, we combine multiple ECCs to lower the overall circuit. The encoder for an UEP scheme using (137,128), (72, 64) and (39, 32) codes is illustrated in Fig. 3.15. For (137,128) code, the input bits b1 through b32 are sent to one parity generator, bits b33 through b64 are sent to the second parity generator and b65 through b128 are sent to the third parity generator. The combiner combines the three sets of parity bits and generates parity bits for the (137, 128) code. When higher coding capability is required, as in (39, 32) code, the second and third parity generator and combiners (shaded blocks in Fig. 3.15) are disabled and the outputs of the first generator are selected. The decoder can be implemented using a similar hierarchical structure.

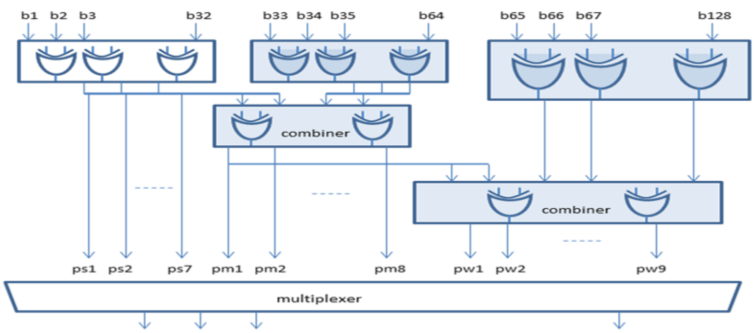


Fig. 3.15. Block diagram of encoder for (137,128), (72, 64) and (39, 32) codes

The power consumption and latency of encoder/decoder pair of ECCs are obtained using Design Compiler from Synopsys and 45nm models from [66]. Table 3.4 lists the area, power consumption, and latency results for the (137,128), (72, 64) and (39, 32) codes. The clock period is 2ns. Note that for large memory, power consumption due to encoding and decoding can be significant.

Table 3.4. Area, latency and power consumption overhead of ECC Schemes for (137,128), (72,64) and (39,32)

	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

In addition to circuitry overhead, ECC techniques require extra memory to store redundant data (parity sequence). As mentioned earlier, the memory overhead can be reduced by implementing UEP instead of fixed ECC scheme. For example, for the same performance degradation, UEP can reduce memory overhead by approximately 29% compared to fixed ECC (72, 64) as shown in example UEP-2.

VI. SUMMARY

In this chapter, we focused on use of voltage scaling to reduce energy consumption of SRAM memories while minimizing quality degradation. We choose JPEG2000 as the target application since it has a large memory requirement and is a popular image coding standard. We analyzed the errors due to voltage overscaling in SRAM memories and presented techniques to mitigate the memory failures. First, we proposed a novel unequal error protection (UEP) schemes that assign different ECCs to different parts of the memory based on their significance. Next, we presented algorithm-specific techniques 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. These techniques enable us to drop the operating voltage of memory with modest reduction in image quality for low to medium compression rates.

CHAPTER 4

COMPENSATING FOR DATAPATH ERRORS CAUSED BY VOLTAGE SCALING: CASE STUDY JPEG

I. Introduction

Voltage scaling is a very effective way of reducing power consumption in datapaths. Unfortunately, aggressive voltage scaling results in critical path violations which lead to errors. In this chapter, we describe an algorithm-specific technique for JPEG that reduces power consumption with little degradation in algorithm performance. The technique is based on exploiting the characteristics of the quantized coefficients after zig-zag scan and is described in Section III. It improves the PSNR performance with small circuit overhead as shown in Section IV.

II. Voltage Scaling Induced Errors in Datapath

A. Failure Analysis

In this section, we focus on failures in the data path which can happen because of critical path violation due to aggressive voltage scaling. Voltage overscaling (VOS) refers to scaling the voltage beyond the value imposed by the critical delay of the circuitry. This may result in timing violations in the data-path, resulting in erroneous operation.

We use the method in [8] to derive the error probability distribution of a 14-bit RCA and use the results to generate the error models under voltage scaling. The 14-bit RCA is illustrated in Fig. 4.1 where 3 of the longer paths are highlighted.

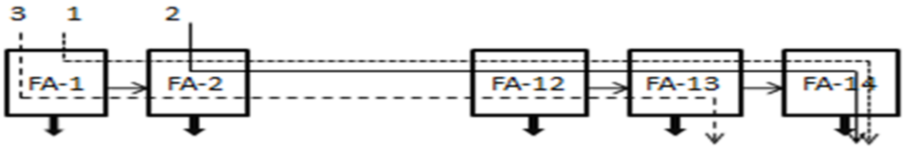


Fig. 4.1. Block Diagram of 14-bit RCA

Assume that the delay of each full adder (FA) is the sum of nominal delay, t_{FA}, systematic variation t_{SYS}, which is typically considered same for all the FAs in a 14-bit RCA, and random variation $t_{r,\_}$ which can be modeled using zero mean iid Gaussian random variable with variance σ_{FA} . Then delay of each carry chain starting from the x^{th} FA and ending at the y^{th} FA can be calculated as

$$T_{chain}\,(x,y) = (x-y)\,\times\,(t_{FA}+t_{SYS}\,) + (t_{r,x}+\ldots+t_{r,y}) \tag{4.1}$$

which can be simplified using the iid Gaussian properties as:

$$T_{chain}\,(\Delta) = \Delta\times\,(t_{FA}+t_{SYS}\,) + \sqrt{\Delta}\times\,t_r \tag{4.2}$$

where $\Delta = x-y$. Thus $T_{chain}\,(\Delta)$ is a Gaussian variable with $\mu = \Delta\times\,(t_{FA}+t_{SYS}\,)$ and $\sigma = \sqrt{\Delta}\times\sigma_{FA}$. Also, the delay of any chain can be represented using only 14 different distributions $T_{chain}(1)$ to $T_{chain}(14)$.

The probability of errors for each bit at the output of the 14-bit adder is derived as follows. Assume that the critical path delay is t_{crt}. We have 14 different paths that may lead to MSB error over the carry chain: LSB to MSB , $LSB+1$ to MSB, $LSB+2$ to MSB etc, where each has a different delay distribution. In order to calculate the probability of error for MSB, we use the Bayes' theorem and sum all the probabilities as:

$$p(t_{MSB} > t_{crt}\,) = \sum_{z=1}^{14} p(T_{chain}\,(z)\; > \; t_{crt}\,|\,chain = z) \tag{4.3}$$

where t_{MSB} is the path delay of MSB bit and $p(chain = z) = \frac{1}{2^z}$.

Thus, for each output bit, we can calculate its error probability for a given t_{crt}. The distribution of errors due to voltage scaling for different supply voltages is shown in Fig. 4.2 when the allowable critical path is 1350ps which is consistent with results in [67]. The following parameters are used. At nominal voltage of 1V, $t_{FA} = 82ps$, $t_{SYS} = 5ps$ and $\sigma_{FA} = 8ps$ for fan-out of four (FO4); at 0.6V, the values increase to $t_{FA} = 240ps$, $t_{SYS} = 5ps$ and $\sigma_{FA} = 15ps$. Note that most of the errors reside in the most significant bits, which can result in significant performance degradation.

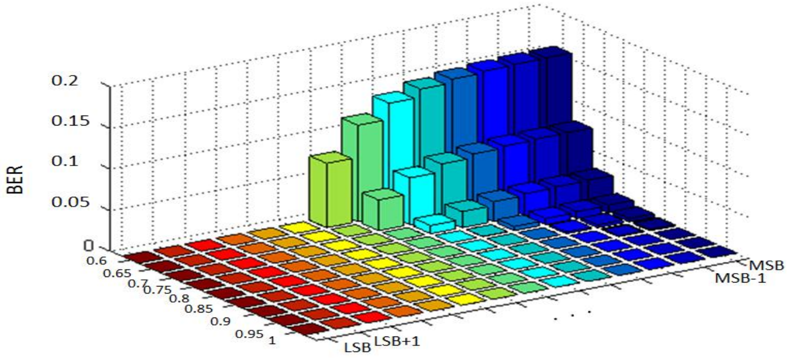


Fig. 4.2. Probability of error distribution for 14-bit RCA for different voltage settings

B. Previous Work on Compensating Datapath Errors

To mitigate the errors due to critical path violation of the computation unit under voltage scaling, algorithm noise tolerance (ANT) has been used in [4] [5] [6] [7]. Fig. 4.3 illustrates the general block diagram of the ANT scheme which consists of the main block and the reduced computation block. The main block implements the original computation at full precision; thus its output (y_m) is prone to critical path violation under voltage scaling. The reduced computation block is designed to generate a statistical replica (y_r) of the original result with a shorter critical path. These two outputs are compared to detect any errors that may have occurred in the main block. Since VOS results in large errors in magnitude, the system chooses y_m if the difference is smaller than the predetermined threshold (Thr) and y_r otherwise.

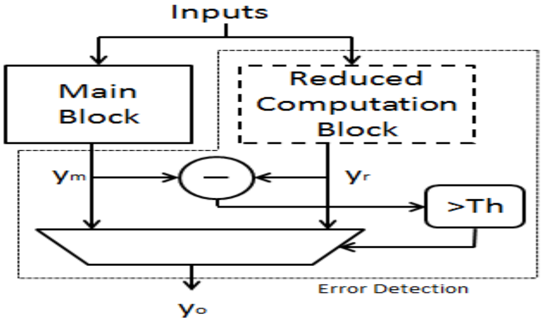


Fig. 4.3. Block Diagram of Algorithm Noise Tolerant (ANT) Scheme

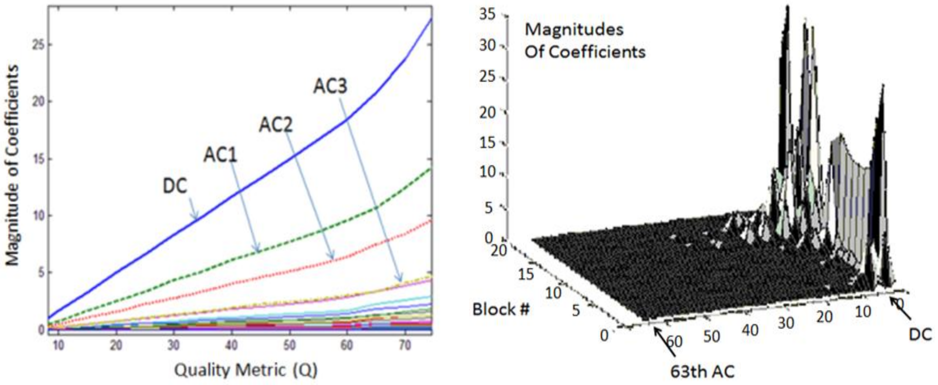
In an ANT based system, the reduced computation block needs to provide good approximation of the original output, have low complexity circuitry to minimize overall overhead, and have shorter critical path to ensure error-free operation. Reduction in computation is achieved by using reduced precision replica [4] [5], subsampling [6], and prediction based error correction [7]. ANT-based systems have been applied to multimedia applications such as FIR low pass filter [4] [5], FFT [5], and motion estimation [6]. In [4], correlation of the FIR low pass filter outputs is used to correct errors, if any. To minimize the overhead, a very simple low pass filter is employed that computes the estimates of the main block. In [5], the reduced computation block is based on 4-bit MSB implementation of FFT while the main block operates on 8-bit data. In [6], a subsampled version of the original motion estimation block is used in the reduced computation block. All these methods achieve 20% to 40% energy reduction while incurring small performance degradation.

III. Proposed Algorithm-Specific Technique: Case Study DCT

In this section, we describe the proposed error compensation technique that exploits the algorithm characteristics of JPEG to mitigate the errors that occur in DCT. The technique exploits characteristics of encoded JPEG data to detect the VOS induced errors.

The error distribution curve given in Fig. 4.2 can be extended to model carry save adders (CSA) that are used in 1D DCT implementation. We use such an error distribution curve to model errors for each output coefficient of the 1D DCT and assume that the errors are independent. The reason is as follows: assume that a single datapath violation occurs during 1D DCT along rows that result in a single miscalculated coefficient. This failure affects the values of eight 2D-DCT coefficients along a column of 8 × 8 DCT. Fortunately, after zig-zag scan, the miscalculated coefficients in a column are separated.

In order to compensate for voltage scaling induced errors, we use algorithm-specific techniques [8]. We utilize the fact that in frequency domain, neighboring coefficients have similar values. Fig. 4.4-a shows the average magnitude of the DC coefficient and several AC coefficients after zig-zag scan for different values of Q for Bridge image. These figures demonstrate that (i) there is a similarity in the magnitude between coefficients of two adjacent AC coefficients after zig-zag scan, (ii) coefficients corresponding to higher frequencies generally consist of smaller values and (iii) the magnitude of coefficients increase with Q. In addition, from our simulations, we find that coefficients of the same order but in consecutive blocks also have similar magnitudes. This is illustrated in Figure Fig. 4.4-b which shows 64 coefficient values of the first 20 blocks of Bridge image when Q=50.



(a) (b)

Fig. 4.4. Magnitude of DC and AC coefficients a) averaged over all blocks; b) first 20 blocks of Bridge image

Recall that while the 8 × 8 DCT units generates 14 bit outputs (12 bit integer + 2 bit fractional), the quantization stage determines the number of bits that are finally used to represent each coefficient. For instance, when Q=50, the 5th AC (AC5) coefficient which is originally 14 bits is quantized and rounded to $AC_q\ (5) = round(\frac{AC5}{10})$ which is represented with 9-bits (bold in Table 4.1). Table 4.1 specifies how many bits are sufficient to represent the coefficients after quantization step for different values of Q. In order to reduce the complexity, we partitioned the 64 coefficients into 4 groups: Group-1 consists of coefficients DC to AC-15, Group-2 consists of AC-16 to AC-31, and so on.

These features are used to derive a procedure for compensating the errors due to voltage overscaling in the datapath. Our procedure consists of 2 steps.

Table 4.1. Number of bits necessary to represent each group of 2D DCT coefficients for natural images

Quantizer	Group-1	Group-2	Group-3	Group-4
Q =< 5	6	5	4	3
5 < Q =<15	7	6	5	4
15 < Q =<30	8	7	6	5
30 < Q =<55	**9**	8	7	6
55 < Q =<70	9	8	7	7

Step 1: We detect and correct errors in sign extension bits. If Table 4.1 specifies that a k-bit representation is sufficient, then by definition, the sign extension bits k to MSB should be all zero for a positive number and all one for a negative number. We pick three bits from the sign extension bits and use majority logic to correct the erroneous sign extension bits. This step is applicable to the groups that can be represented using 7 bits or less. False detection probability of this scheme is $C_2^3\ (BER_s)^2\ (1 - BER_s\ ) + (BER_s)^3$, where BER_s represents error rate probability of a single bit.

$\boldsymbol{Step\ 2}$: We detect and correct an error when we find an abnormal increase in magnitude in one of the coefficients. This is motivated by the fact that coefficients that are adjacent to each other have similar magnitudes. The procedure is as follows. In order to detect an error in the j^{th} AC coefficient of the k^{th} block, we take the average of the two adjacent coefficients, namely, $(j-1)^{th}$ and $(j+1)^{th}$ coefficient, and compare it with the j^{th} coefficient. If the difference is higher than a predetermined threshold, we calculate the average of the j^{th} AC coefficient of the $(k-1)^{th}$ and $(k+1)^{th}$ block and compare again with the j^{th} coefficient in the k^{th} block. If the difference is again higher than the threshold, we change the value of the j^{th} coefficient to the average of the two neighboring coefficients in the same block. The pseudo code for this step is given in Algorithm 4.1. Since each group specified in Table 4.1 has different bit width specifications, we assign different threshold levels for each group to reduce the false detection probability. For instance, the threshold value for Group-1 is 64 whereas it is only 8 for Group-4. These threshold values were determined by experimentation with a sample set of images.

Algorithm 4.1. Pseudo Code for Step-2

```
Initialize Parameters Thresholds ($THR_i$) i = 1,2,3,4
for each AC coefficient do
    if $|AC(k,j) - \frac{AC(k,j+1)+AC(k,j-1)}{2}| > THR_i$ , then
        if $|AC(k,j) - \frac{AC(k+1,j)+AC(k-1,j)}{2}| > THR_i$ , then
            $AC(k,j) = \frac{AC(k+1,j)+AC(k-1,j)}{2}$
        end if
    end if
end for
```

IV. Simulation Results

In this section we describe the algorithm quality performance and the hardware overhead of the proposed scheme. The quality performance is described in terms of peak signal to noise ratio ($PSNR$). The compression rate is measured in number of bits required to represent one pixel (bpp) and is related to the quality metric (Q). We introduce a new metric BER(VOS) which represents the BER at scaled voltage level.

A. Algorithm Performance

The performance of the proposed algorithm-specific method when BER(VOS)= 10^{-4} and 10^{-3} are shown in Fig. 4.2 for the Bridge image using full-precision DCT. Using the error estimation from Section II, we see that 0.83V operation results in a BER(VOS) of 10^{-4} and 0.75V operation results in a BER(VOS) of 10^{-3}.

At BER(VOS) of 10^{-4}, our method has 3dB improvement over the no-correction case and a drop of approximately 1dB compared to the error-free case at 0.75 bpp compression rate (Q~30) illustrated in Fig. 4.5. At BER(VOS) of 10^{-3}, quality degradation due to errors is very high as shown in Fig. 4.6. However the proposed technique helps improve the PSNR by approximately 7.5dB at 0.75bpp. Table 4.2 summarizes the performance of the proposed technique for 4 representative images

(Bridge, Baboon, Lena and Pepper) images at compression rate of 0.75bpp when BER(VOS) is 10^{-4} corresponding to operating voltage of ~0.83V.

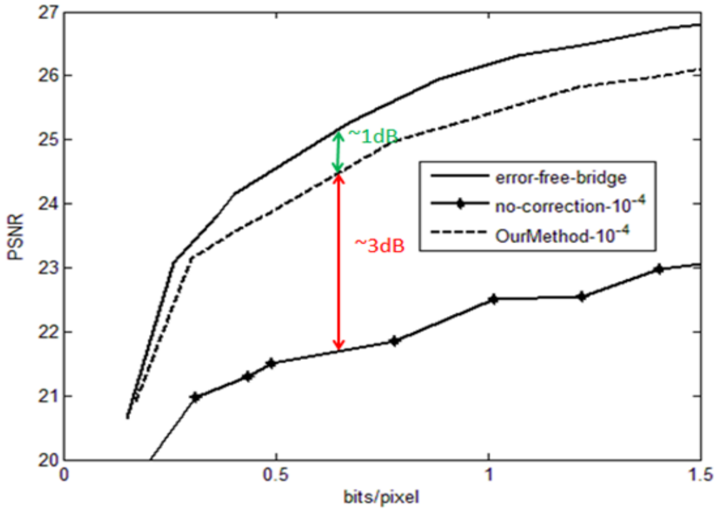


Fig. 4.5. PSNR vs. Compression rate for Bridge when BER=10^{-4}

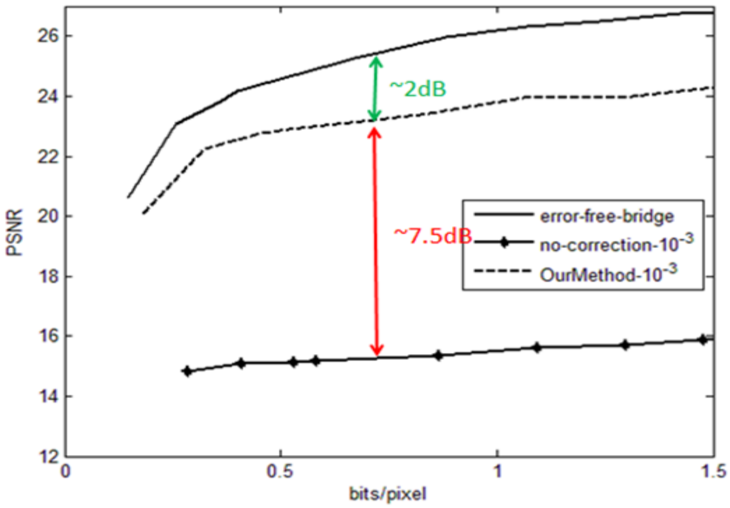


Fig. 4.6. PSNR vs. Compression rate for Bridge when BER=10^{-3}

Table 4.2. PSNR values of proposed technique at 0.75bpp compression rate when BER(VOS)=10^{-4}

Images	Error Free	No Correction	Proposed Scheme
Bridge	25.2	21.4	24.1
Baboon	25.7	20.6	24.3
Lena	32.8	27.8	31.2
Pepper	31.5	26.4	30.2

B. Hardware Overhead

The hardware overhead of the proposed algorithm-specific consists of majority voter, coefficient comparator and average calculator. Majority voter scheme is used in the first step to detect errors in the sign extension of bits. Coefficient comparator is used to detect abnormality in magnitudes of neighboring coefficients. Average calculator is used to compensate for an error bit. Fortunately, this unit is rarely activated due to small number of failures. Table 4.3 illustrates the power consumption and latency results of the three units for clock period of 4ns. We see that the overhead is fairly small, approximately 12% of full precision 2D-DCT, enabling us to operate at scaled voltage levels with small loss in image quality.

Table 4.3. Power Consumption and latency of the three units in the voltage overscaling compensation scheme

	Majority Voter	Coefficient Comparator	Average Calculator
Active Power (uW)	3.6	96	103
Latency (ps)	42	459	421

V. Summary

In this chapter, we focused on the use of voltage scaling to reduce the power consumption in the 2D DCT module of the JPEG codec. We analyzed the errors due to voltage overscaling and presented a low overhead algorithm-specific method to compensate for most of these errors. The proposed method exploits the characteristics of the quantized DCT coefficients and is able to reduce datapath energy by 20% with a slight degradation in quality.

CHAPTER 5

COMPENSATING FOR DATAPATH ERRORS CAUSED BY DYNAMIC RANGE REDUCTION

I. INTRODUCTION

In this chapter, we focus on dynamic range reduction as a technique to reduce energy [20] [21]. In dynamic range reduction, the number of bits in a computation are reduced by truncating the bits, resulting in lower dynamic as well as leakage energy. However, truncation introduces errors which need to be compensated to minimize the degradation in system performance. In this chapter, we propose an unbiased estimator that reduces the truncation noise with very small overhead circuitry. We describe this technique in Section III. While truncating low order bits is more common, in some cases, the high order bits contain less information and can be truncated. In Section IV, we present high order clipped computation that truncates the MSB bits in SAD computation during motion estimation in video coding. The proposed scheme achieves 28% energy saving at nominal voltage and 54% energy saving at iso-throughput condition compared to baseline SAD architecture.

II. EFFECTS OF DYNAMIC RANGE REDUCTION

Reducing the datapath precision to achieve low power consumption is a popular technique in signal processing systems. Typically, high order bits contain most of the information while low order bits capture the details of the application. Fig. 5.1 illustrates the savings in energy consumption and truncation induced errors of a 16-bit RCA for different bit widths in 45nm technology. Since RCA has a regular structure, the energy reduction is proportional to the bit-width of the adder. For instance, at nominal voltage, we observe 24% reduction in energy consumption of the adder when 12 bits are used instead of 16 bits.

One drawback of reduced precision arithmetic is that it introduces truncation errors. Fig. 5.1 plots truncation noise defined as the magnitude of the difference between the outputs obtained with full precision data and the output obtained with truncated data scaled by the full precision output. From Fig. 5.1, we see that while truncation noise increases logarithmically, energy saving of the adder increases linearly with increase in number of truncated bits. Low order bit truncation can easily be applied to multimedia applications such as filtering, DCT; however, one of the main challenges is to compensate for the quality degradation caused by reduced precision operation.

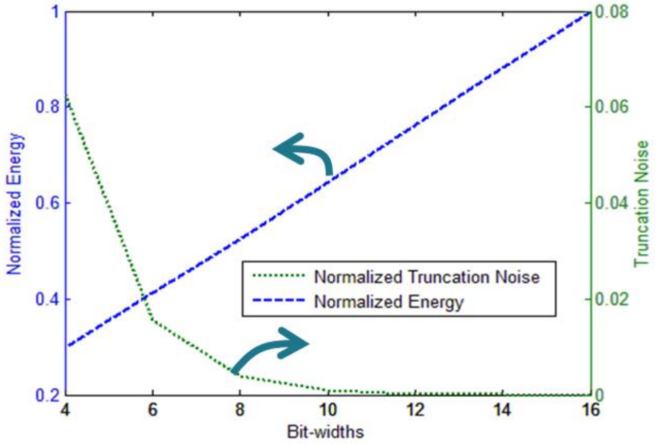


Fig. 5.1. Energy and Noise distribution of 16-bit RCA wrt bit-width

III. Reducing Dynamic Range by Low Order Bit Truncation

Bit truncation methods that remove low order bits have been very effective for motion estimation [16] [17] [24]. In [16], instead of using all 8 bits, only 4 or 5 of the higher order bits are used to represent current and reference pixels. In [17], the performance degradation and increase in compressed data rate have been studied for low order bit truncation in motion estimation used in H.264. Fig. 5.2 illustrates the average degradation over several video sequences for low order bit truncation ranging from 1-bit to 4-bits for diamond (DS) and three step search (TSS) strategies during motion estimation and intra-prediction modes of H.264. Since motion estimation and intra-prediction are based on subtraction of the pixel values, the expected performance degradation is not very high. This is because subtraction is more tolerant to truncation noise than addition or multiplication operations. As a result, motion estimation units do not need an additional compensation unit to handle the truncation errors.

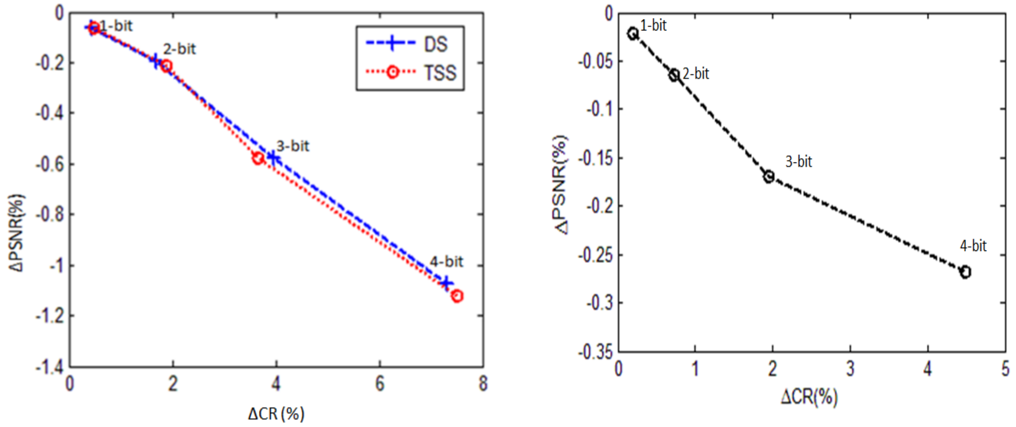


Fig. 5.2. Average performance loss due to bit truncation a) using DS and TSS in motion estimation, b) intra prediction of H.264

In algorithms whose building blocks are multiplications and additions, the truncation error has to be compensated to maintain good image quality. Next, we describe our techniques for compensating truncation errors and illustrate it using DCT and FIR filters as examples.

Truncation Induced Errors: Analysis and Compensation

We begin by analyzing the effect of bit truncation on simple arithmetic operations such as addition, subtraction and multiplication. We describe the error characteristics for operations on unsigned numbers; however the procedure can easily be extended to operations on two's complement and signed numbers. Next, we describe a method to reduce the effect of truncation based errors on system quality.

The output of a DSP system after LOB truncation at time instant k can be expressed as:

$$y_e[k] = y_o[k] + \nu[k] \tag{5.1}$$

where $y_o[k]$ is the truncation-free output and $\nu[k]$ is the truncation induced error (noise) which is a random variable with mean μ_N and variance σ_N^2. The noise power can be represented by the mean square error (MSE) defined as $\mu_N^2 + \sigma_N^2$. In order to reduce the noise power, we propose a method that estimates the mean value of the truncation error during the pre-computation stage and compensates for it. We refer to this method as μ_N compensation. The overhead of this method is very small. Moreover the noise power after μ_N-compensation does not depend on μ_N anymore and is only a function of the variance of the truncation error.

Let us consider a system whose inputs are originally represented with $M+1$ bits, $x(M{:}\,0)$. When L bit truncation is employed, where $L \leq M$, the input becomes $x(M{:}\,L)$. Assuming uniformly distributed input signals, we can express Q_x, the truncation error for the input signal x, as:

$$Q_x(L-1{:}\,0) = x(M:0) - x(M:L) = \sum_{i=0}^{L-1} 2^i b_i \tag{5.2}$$

where b_i is an independent, uniform random variable with two discrete values: 0 and 1. The expected value and variance of Q_x are given by

$$E[Q_x(L-1{:}0)] = \mu_q(L-1{:}0) = \frac{1}{2}(2^L - 1) \tag{5.3}$$

$$var[Q_x(L-1{:}0)] = \sigma_q^2(L-1{:}0) = \sum_{i=0}^{L-1} 4^i \sigma_{b_i}^2 = \frac{4^L - 1}{12} \tag{5.4}$$

where $\mu_q(L-1{:}0)$ and $\sigma_q^2(L-1{:}0)$ are mean and variance of $Q_x(L-1{:}0)$ and $\sigma_{b_i}^2$ is the variance of b_i.

Using equation (5.12)-(5.13), we can compute the expected value and variance of the truncation error (Q_{add}) of an adder with inputs *x* and *y*. Both inputs are independent and the lower L bits (out of M+1 bits) have been truncated.

$$\begin{aligned} E[Q_{add}(L-1{:}0)] &= E[x(M{:}0) + y(M{:}0) - x(M{:}L) - y(M{:}L)] \\ &= 2\mu_q(L-1{:}0) \end{aligned} \tag{5.5}$$

$$var[Q_{add}(L-1{:}0)] = 2\sigma_q^2(L-1{:}0) \tag{5.6}$$

Using similar analysis, we compute the expected value and variance of subtraction and multiplication.

$$E[Q_{sub}(L-1{:}0)] = E[x(M{:}0) - y(M{:}0) - x(M{:}L) + y(M{:}L)] = 0 \tag{5.7}$$

$$var[Q_{sub}(L-1{:}0)] = 2\sigma_q^2(L-1{:}0) \tag{5.8}$$

$$\begin{aligned} E[Q_{mul}(L-1{:}0)] &= E[x(M{:}0)y(M{:}0) - x(M{:}L)y(M{:}L)] \\ &= 2\mu_q(M{:}0)\mu_q(L-1{:}0) - \mu_q(L-1{:}0)^2 \end{aligned} \tag{5.9}$$

$$\begin{aligned} var[Q_{mul}(L-1{:}0)] = 2[&\mu_q(L-1{:}0)^2\sigma_q^2(M{:}L) + \mu_q(M{:}L)^2\sigma_q^2(L-1{:}0) \\ &+\sigma_q^2(L-1{:}0)\sigma_q^2(M{:}L)] + 2\mu_q(L-1{:}0)^2\sigma_q^2(L-1{:}0) \\ &+\sigma_q^2(L-1{:}0)\sigma_q^2(L-1{:}0) + 2\mu_q(L-1{:}0)\mu_q(M{:}L)\sigma_q^2(L-1{:}0) \end{aligned} \tag{5.10}$$

Fig. 5.3 illustrates how the noise power (MSE) of 16-bit multiplication of unsigned numbers can be reduced with μ_N compensation. We see that the analytical results and simulated results match very closely. Moreover μ_N plays an important role in determining the noise power and compensating for μ_N helps reduce the MSE by >2X.

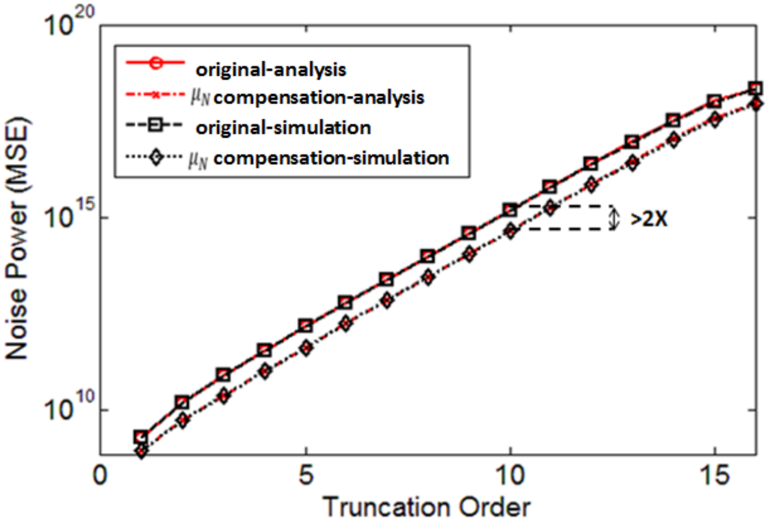


Fig. 5.3: Noise power of Q_{mul} for 16-bit multiplication with and without μ_N-compensation

Since noise power is proportional to $\mu_N^2 + \sigma_N^2$, the proposed method helps in reducing the noise power for computations such as additions and multiplications. It does not help in the case of subtractions since the μ_N of subtraction is 0. Furthermore, we see that, noise power of an ‘L’ bit truncation with compensation and ‘L-1’ bit truncation without compensation are comparable. However since the overhead of μ_N compensation is very small, a system with larger number of truncation bits has larger energy savings as will be illustrated in Chapter 6. Next we illustrate the use of the μ_N compensation method to compensate for errors in DCT and FIR filter computation.

Example 1(DCT): General description of DCT in JPEG is given in Chapter 2. Fig. 5.4 describes the architecture to compute 4 DCT coefficients ($W_{0,}$ $W_{1,}$ W_2 and W_4) of the 8-point DCT used in JPEG. Here, x_s are the input pixels, W_s are the outputs, and the DCT matrix coefficients are $a = \frac{1}{2}\cos\left(\frac{\pi}{16}\right)$, $b = \frac{1}{2}\cos\left(\frac{2\pi}{16}\right)$, $c = \frac{1}{2}\cos\left(\frac{3\pi}{16}\right)$, $d = \frac{1}{2}\cos\left(\frac{4\pi}{16}\right)$, $e = \frac{1}{2}\cos\left(\frac{5\pi}{16}\right)$, $f = \frac{1}{2}\cos\left(\frac{6\pi}{16}\right)$, $g = \frac{1}{2}\cos\left(\frac{7\pi}{16}\right)$. In our analysis, we introduce 2 extra bits to represent the fractional part of the computation in baseline mode. This results in approximately 0.1dB improvement over the 12-bit implementation. After pairwise subtraction and addition of the pixels, we obtain y_0 to y_7, where y_0=x_0+x_7, y_1= x_1+x_6, …, and y_7= x_3-x_4. For W_0 and W_4, common sub-expression elimination is used to obtain results with small number of computation units as illustrated in Fig. 5.4-b. Implementation of W_2 is illustrated in Fig. 5.4-c; a variant of which is used for W_6. Fig. 5.4-d shows the computation structure used to find W_1. The odd coefficients ($W_{3,}$ W_5 and W_7) are computed using units that are similar to the unit for W_1.

We calculate the truncation noise (TN) for the DCT outputs for a 14 bit fixed point implementation of DCT. The expected errors due to truncation in W_0 and W_1 can be expressed as follows. To simplify our analysis, we assume that all Y values are uncorrelated and so the expected value for L bit truncation is $\frac{2^L-1}{8}$. Since $W_0 = d * (y_0 + y_1 + y_2 + y_3)$, the truncation error for W_0 is given by $TN_{w0} = E[d * (y_0\ (L-1{:}\,0) + \cdots + y_3\ (L-1{:}\,0))] = \lfloor\frac{d(2^L-1)}{2}\rfloor$. Similarly the truncation errors for W_1 is given by $\lfloor(a+c+e+g)\frac{2^L-1}{8}\rfloor$, and that of W_2 is given by $(b+f-b-f) \times E[y] = 0$. The truncation noise of W_4 and W_6 are also zero.

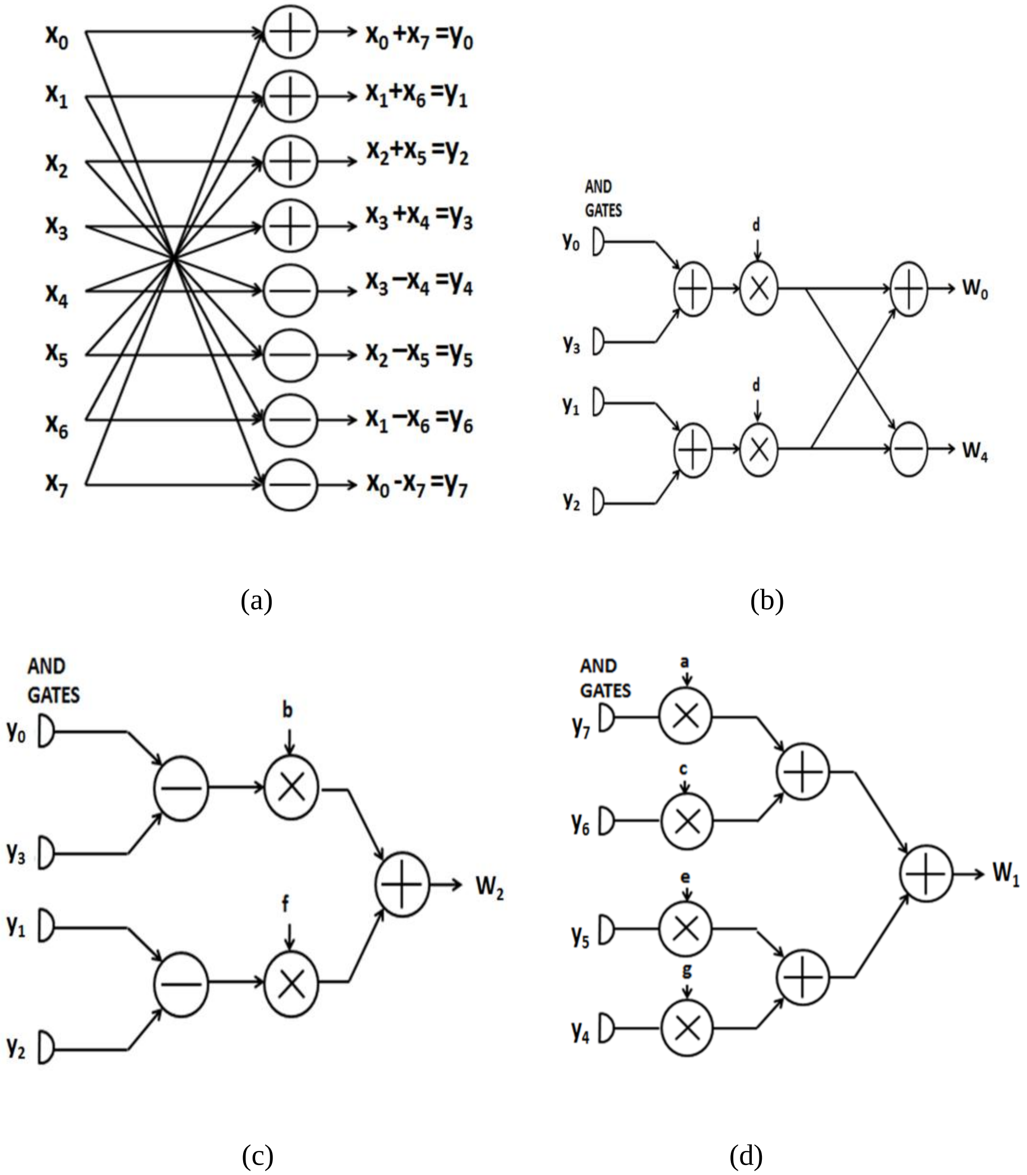


Fig. 5.4. Architecture of 1-D DCT coefficients. a) First stage butterfly, b) W_0 and W_4 computation units, c) W_2 unit and d) W_1 unit

The expected truncation noise values are used as unbiased estimators to compensate the errors. Instead of compensating for errors of all the outputs, we only compensate for errors in the computation of W_0 and W_1. The motivation for this is that these coefficients are the most important ones and the corresponding estimation errors are the largest. Also this keeps the complexity of the overhead circuitry small. Fig. 5.5 illustrates the compensation mechanism for W_1 computation. The overhead of this scheme is the 14-bit adder at the output as well as the AND gates to disable a selective set of input bits. The area and power overhead due to extra processing elements is around 2% of the overall DCT implementation.

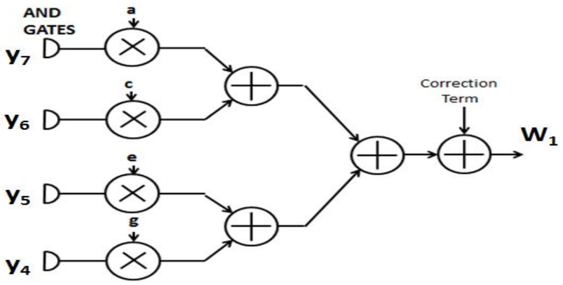


Fig. 5.5. Modified DCT computation of W_1

Fig. 5.6 illustrates the performance improvement with the use of unbiased estimators for W_0 and W_1 when low order bits are truncated for DCT computation of the Baboon image. For 1bpp compression rate, 4-bit truncation causes a degradation of 1.3dB which is reduced to 0.6dB with compensation. For the same 1bpp compression rate, when 6 bits are truncated, the performance improvement is approximately 1.2dB compared to the system without compensation. Thus as the truncation level increases, we observe higher performance improvements in systems that use compensation.

Table 5.1 lists the power consumption and latency of the 1D DCT engine with clock period of 4ns obtained using design compiler from Synopsys [68] and 45 nm Nangate libraries [66]. The 0-bit truncation scheme includes the overhead circuitry for supporting multi-bit truncation and thus has higher power and latency compared to the baseline scheme.

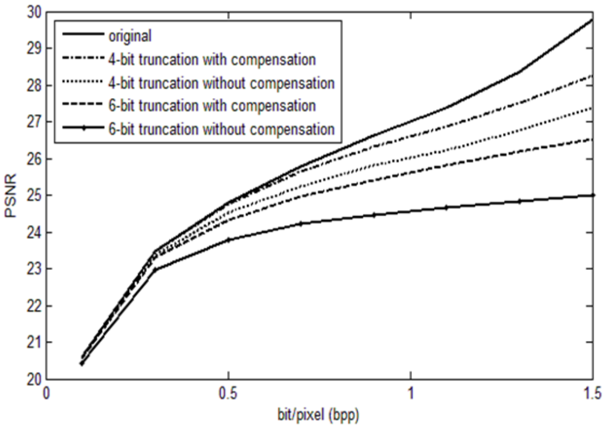


Fig. 5.6. Performance comparison between uncompensated and compensated bit truncation for DCT computation of Baboon image

The active power decreases significantly with the increase in the number of truncation bits. Specifically, we see a 23% reduction in active power compared to the baseline scheme for 4-bit truncation and 35% reduction in active power for a 6-bit truncation. Table 5.1 also lists the change in PSNR calculated at 1bpp ($Q \sim 50$) using 6 sample images namely, Lena, Pepper, Bridge, Baboon, Flight and House.

Table 5.1. Quality, power and latency of DCT engine for different levels of truncation

	ΔPSNR (dB)	Active Power (mW)	Latency (ns)
Baseline	0	5.39	2.92
0-bit Truncation	0	5.51	3.31
2-bit Truncation	0.1	4.76	2.95
4-bit Truncation	0.6	4.14	2.78
6-bit Truncation	2.4	3.49	2.51

Example 2(FIR Filter): Consider a FIR low pass filter (LPF) using unsigned inputs and coefficients, which is typical in many multimedia algorithms. The output y(n) of an N-tap filter with M+1-bit precision is given by

$$y(n, M{:}0) = \sum_{k=0}^{N-1} h(k, M{:}0)x(n-k, M{:}0) \tag{5.11}$$

where h(k,M:0) is the k'th coefficient of the filter and x(n-k,M:0) represents the input value time at n-k. Such a computation can be implemented efficiently using MAC based architectures. We can calculate the unbiased estimator for L-bit truncation assuming coefficients are less than one as:

$$\begin{aligned} E[Q_x] &= E\left[\sum_{k=0}^{N-1} h(k, M{:}0)x(n-k, M{:}0) - \sum_{k=0}^{N-1} h(k, M{:}L)x(n-k, M{:}L)\right] \\ &= \sum_{k=0}^{N-1} E[(h(k, M{:}0)x(n-k, M{:}0) - \ h(k, M{:}L)x(n-k, M{:}L))] \end{aligned} \tag{5.12}$$

When filter coefficients are known, the estimator given in equation (5.13) reduces to:

$$\begin{aligned} &= E[x(L-1{:}0)] \times \sum_{k=0}^{N-1} h(k, M{:}0) + \ E[x(M{:}L)] \times \sum_{k=0}^{N-1} h(k, L-1{:}0) \\ &= \frac{(2^L-1)}{2} DC_{gain} + \frac{(2^{M+1}-2^L)}{2} \sum_{k=0}^{N-1} h(k, L-1{:}0) \end{aligned} \tag{5.13}$$

where DC_{gain} represents the sum of filter tap coefficients for LPF given by $DC_{gain} = \sum_{k=0}^{N-1} h(k, M{:}0)$. As an example, for a 3x3 Gaussian Filter ($\sigma = 1$) with M=7 and L=2 the unbiased estimator value is 3; this value increases to 15 when L=4.

Fig. 5.7 illustrates the block diagram of the proposed MAC based architecture for LPF. Filter coefficients and input data are truncated using an array of AND gates before the multiplication; thus only high order bits become active during computation. After N cycles of MAC computation, the correction factor is applied to reduce errors due to truncation.

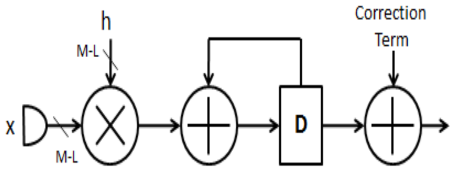


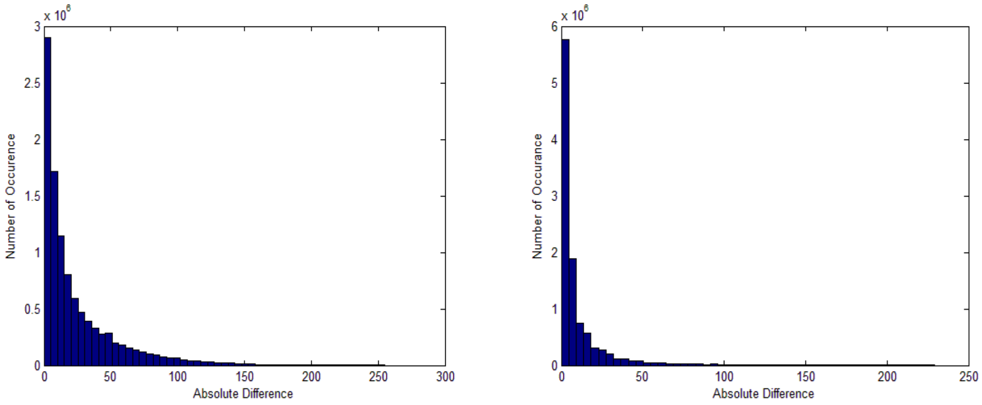
Fig. 5.7. MAC implementation of a low pass filter

IV. Reducing Dynamic Range by High Order Clipped Computation (HOC)

It is not always the case that the low order bits are less significant in computation and so can be dropped. In this part, we show that high order bits of the sum of absolute difference (SAD) computation in motion estimation can also be dropped. The proposed scheme uses the statistics of absolute difference (AD) and SAD computations to reduce the dynamic range and approximate the computations. Specifically, it exploits the fact that most of the AD values are small due to locality of current and reference blocks, and that most of the large AD values are for blocks that are likely not to be selected, and thus these values can be approximated.

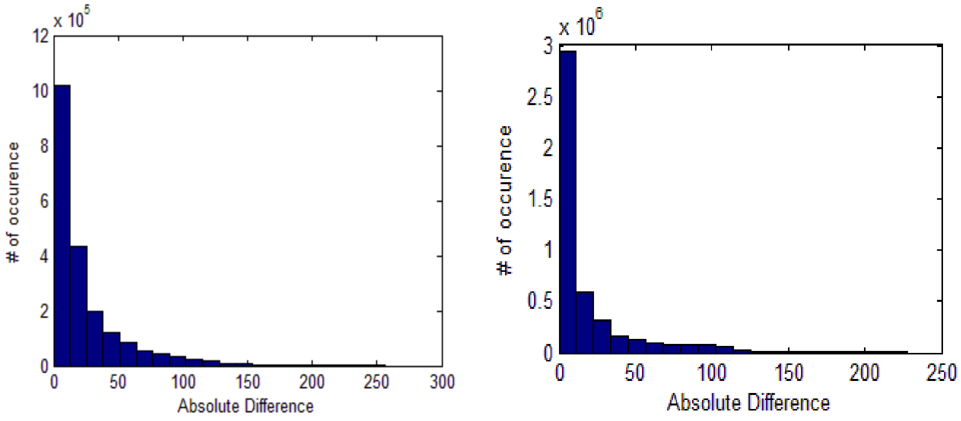
To motivate the HOC scheme, we first study the statistics of AD and SAD computations of motion estimation and intra prediction blocks in a video codec for video sequences with different amount of motion. Fig. 5.8 and Fig. 5.9 illustrate the distribution of AD values for Football and Foreman video sequences for quantizer level Q=20 in motion estimation stage. Most of the AD calculations result in very small values due to locality of current and reference blocks. We see that, on average, 82% of the AD results of Football sequence is smaller than 32, which can be represented by 5-bits. This ratio increases up to 93% for Foreman and further for slow motion video sequences (97% for Claire). We observe similar trends for intra prediction. For instance, 76% to 89% of the AD results are smaller than 32 for the same set of video sequences in intra prediction.

Table 5.2 and Table 5.3 give the percentages of AD values that can be represented by 4 to 8 bits for Football and Foreman video sequences for different values of Q. We see that these ratios are not affected by Q. Similar results have been obtained for other test video sequences.



(a) (b)

Fig. 5.8. Histogram plots of AD computations in motion estimation of a) Football and b) Foreman video sequences



(a) (b)

Fig. 5.9. Histogram plots of AD computations in intra-prediction of a) Football and b) Foreman video sequences

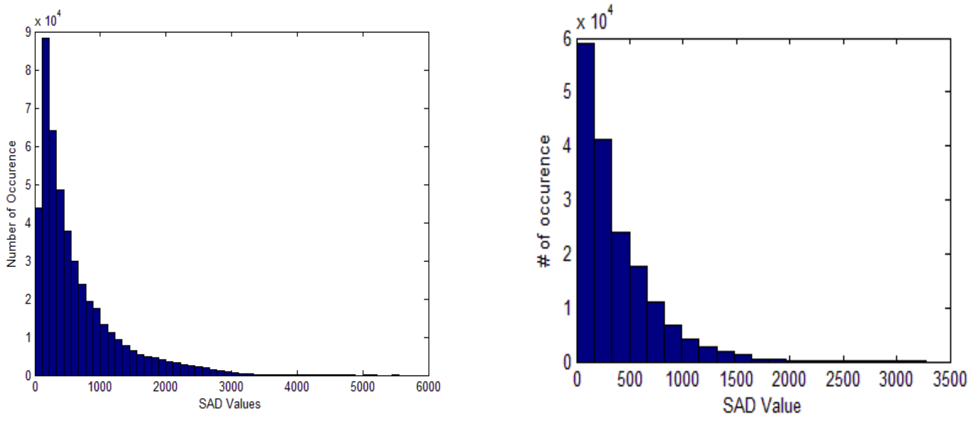
Table 5.2. Percentage of AD values during Inter-prediction that can be represented by 4 to 8 bits for Football Video sequence

# of bits	Q=10	Q=20	Q=30	Q=40
4	59%	60%	60%	60%
5	81%	82%	82%	82%
6	92%	92%	93%	93%
7	98%	99%	99%	99%
8	100%	100%	100%	100%

Table 5.3. Percentage of AD values during Inter-prediction that can be represented by 4 to 8 bits for Foreman Video Sequence

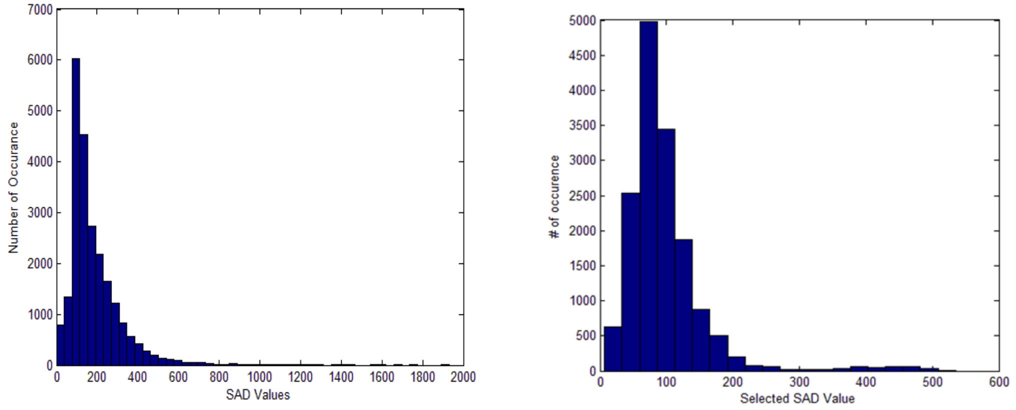
# of bits	Q=10	Q=20	Q=30	Q=40
4	85%	86%	86%	84%
5	93%	93%	94%	94%
6	98%	98%	98%	98%
7	~100%	~100%	~100%	~100%
8	100%	100%	100%	100%

Our next observation is that most of the large AD values make the corresponding SAD values large, and at the end, the blocks with the large SAD values are not selected. Fig. 5.10 shows the distribution of SAD values of all blocks for inter and intra prediction and Fig. 5.11 shows the distribution of SAD values of only the selected blocks for the Foreman video sequence. We see that SAD values of the selected blocks are not greater than a couple of hundred and large AD values could not have contributed to these SAD values.



(a) (b)

Fig. 5.10. Histogram of SAD values of all blocks for a) inter prediction, b) intra prediction



(a) (b)

Fig. 5.11. Histogram of SAD values of selected blocks for a) inter prediction, b) intra prediction

Table 5.4 lists the maximum, mean and standard deviation of SAD values of selected and unselected blocks for 4 different quantization levels for inter prediction. The SAD values of selected blocks have approximately 4 times smaller values for maximum, mean and variance. Thus, when the AD values are large, we do not need to update SAD value with the exact AD value. In fact, we can ignore the result and replace it with a correction factor which is an approximation of the actual AD value. This scheme results in significant energy reduction and little performance penalty as will be demonstrated in the following sections.

Table 5.4. Statistics of Selected and Unselected SAD Values for Foreman Video Sequence during Inter Prediction

	Max of selected blocks (max of all blocks)	Mean of selected blocks (mean of all blocks)	σ of selected blocks (σ of all blocks)
Q=10	1595 (4810)	161 (593)	126 (578)
Q=20	1576 (4810)	178 (579)	124 (578)
Q=30	1261 (4879)	233 (599)	125 (568)
Q=40	1242 (4866)	364 (624)	182 (553)

A. Proposed SAD Algorithm

The proposed scheme uses HOC computation and 1-bit LOB truncation for inter prediction and 2-bit LOB truncation for intra prediction (see Fig. 5.2). As described earlier, most of the AD computations result in small values and the large values of AD generally contribute to SAD computations of unselected blocks. Thus, we update the SAD value with an approximate value when AD is larger than a threshold value (Thr). The pseudo code for the algorithm is given in Algorithm 5.1.

In order to avoid 8-bit AD computation before comparing with Thr, we split the

Algorithm. 5.1. Proposed HOC Computation

```
Input: Pixels from reference and search blocks.
Output: SAD value of the given blocks
Initialize $Thr$, SAD=0 and BBIT=$\log_2 Thr$
    for each pixel location in the block
        AD_Low = AD of bits LSB to BBIT-1
        if (AD of pixels> $Thr$) then
                SAD = SAD + correction factor
        else
                SAD = SAD + AD_Low
    end
```

computation into two separate parts. Let BBIT be the boundary bit which is determined by the threshold value, BBIT=$log_2 Thr$. Then the lower order bits are LSB to BBIT-1 and higher order bits are BBIT to MSB. For instance, LSB is bit 1 for inter prediction since bit 0 has been truncated and LSB is bit 2 for intra prediction since intra-prediction is less sensitive to LOB truncation as illustrated in Fig. 5.2. We compute the difference of the lower order bits and higher order bits in parallel and combine these results to determine whether AD $> Thr$ and whether the SAD value should be updated by the correction factor or not.

The threshold value *Thr* has to be selected carefully. A larger value of *Thr* gives better PSNR and CR performance but has larger circuit overhead including larger critical path delay. Table 5.5 illustrates the performance degradation of three threshold values (*Thr*=16, 32, 64) averaged over 12 video sequences for inter and intra prediction modes. We see that *Thr*=32 provides very small increase in $\Delta PSNR$ and a small decrease in ΔCR for most of the video sequences for inter prediction. On the other hand, *Thr*=64 provides low performance degradation for intra prediction. For inter prediction, choosing *Thr*=32 with 1-bit LOB result in the 7-bit AD computations being split into two parts with comparable delay (a 3-bit higher order computation part and a 4-bit lower order computation part). For intra prediction, choosing *Thr*=64 with 2-bit LOB also result in a 4-bit lower order computation which allow us to use the same SAD unit for both intra and inter prediction as illustrated in the next Section.

The rest of the chapter assumes *Thr*=32 for inter prediction and *Thr*=64 for intra-prediction. If, however, the application requires higher algorithm performance, then higher *Thr* should be chosen at the expense of larger circuit overhead.

Table 5.5. Average increase in compressed data rate and reduction in PSNR *Thr*=16,32,64

		Thr=16	***Thr*=32**	*Thr*=64
Inter-Prediction	ΔCR (%)	4.8%	**1.8%**	0.5%
	ΔPSNR (%)	0.1%	**0.06%**	<0.01%
Intra-Prediction	ΔCR (%)	5.6%	2.9%	**1%**
	ΔPSNR (%)	0.14%	0.09%	**0.07%**

B. Proposed SAD Architecture

In this section, we describe a new SAD architecture based on the proposed scheme for inter and intra prediction. Fig. 5.12 illustrates the conventional SAD architecture that computes the difference of four 8-bit number pairs in parallel (A-B and B-A), picks the positive one and updates the SAD value. AD computation unit is illustrated in Fig. 5.12. Parallelization of four is due to block sizes in H.264 which is being multiples of 4 (4x4 to 16x16). The last stage adder is implemented using a 5 input carry save adder (CSA).

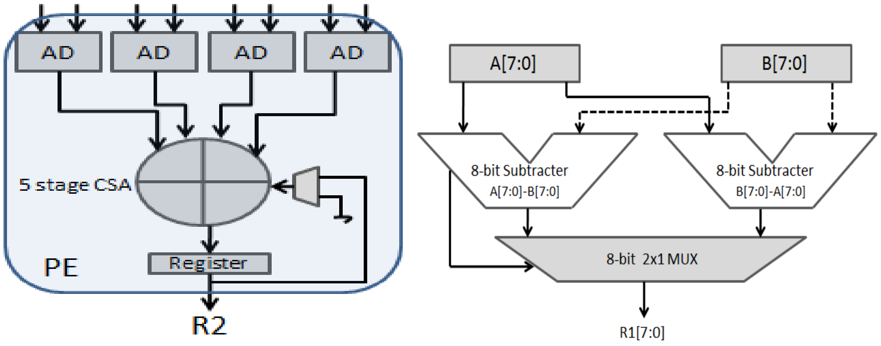


Fig. 5.12. a) Baseline SAD calculation unit, b) 8-bit absolute difference calculator

The proposed AD architecture for inter-prediction mode with $Thr = 32$ is illustrated in Fig. 5.13. The two LOB units (LOB1 and LOB2) compute $A[4:1] -$

$B[4:1]$ and $B[4:1] - A[4:1]$, respectively. When $B[4:1] > A[4:1]$, Cout_1 of LOB2 is 1. Similarly, when $B[7:5] > A[7:5]$, Cout_2 of HOC is 1.

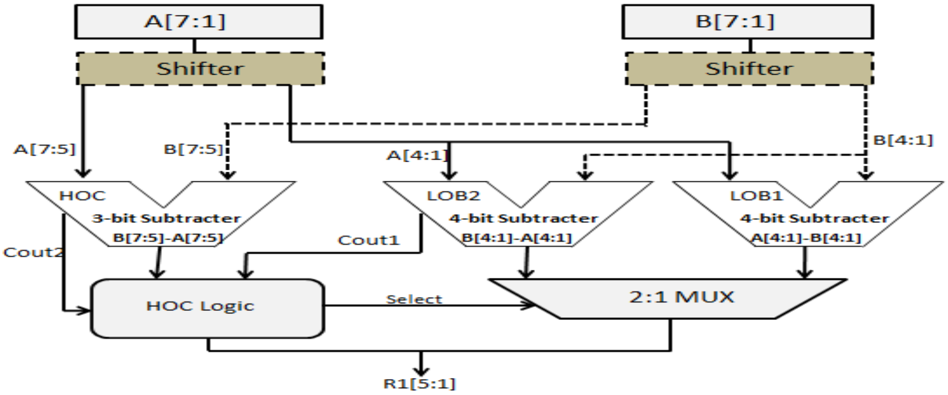


Fig. 5.13. Proposed SAD calculation unit

The HOC unit computes the difference $B[7:5] - A[7:5]$ in parallel with the two LOB units. When the result of HOC part is 001, it means that $B[7:5] - A[7:5] = 32$. Now, if Cout_1=0 then $B[7:1] - A[7:1] < 32$, and the SAD value is updated using LOB1 if Cout_2=0 and updated using LOB2 otherwise; if Cout_1=1, then SAD value is updated with the correction factor which is 32. Similarly, when the HOC part is 111, it means that $B[7:5] - A[7:5] = -32$, and if Cout_1=1 then $A[7:1] - B[7:1] < 32$ and the SAD value is updated using LOB1 if Cout_2=1 and updated using LOB2 otherwise; if Cout_1=0 it is updated with the correction factor. When the HOC part is 000, the SAD is updated using LOB1 if Cout_1=0 and LOB2 otherwise. For all the other HOC results, SAD is updated using the correction factor. The *HOC logic* unit has inputs Cout_1, Cout_2 and the 2-bit HOC output and generates 1 control signal to select either LOB1 output or LOB2 output, and a second signal corresponding to bit 5 of R1. Note that bit 5 corresponds to the correction factor 32. The *HOC logic* has a critical path of 2 NAND and 1 NOR gates. Fig. 5.14 gives the flow chart of the proposed scheme.

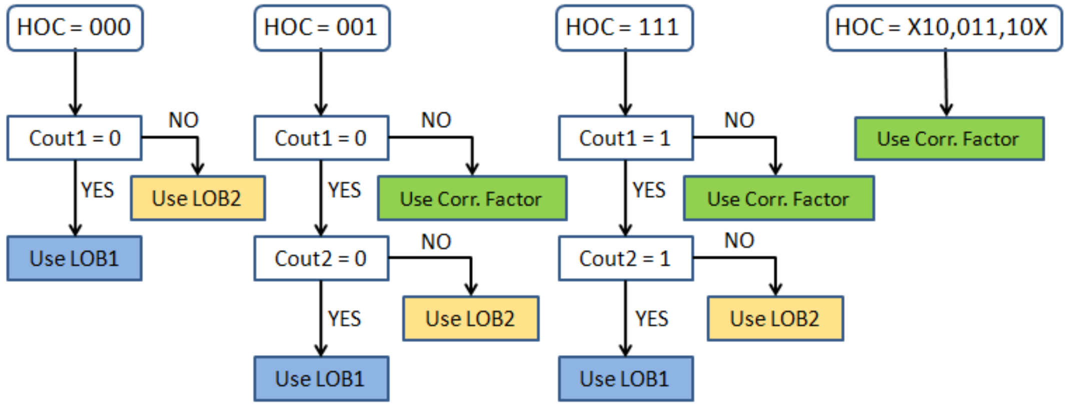


Fig. 5.14. Flow Chart of the proposed SAD calculation scheme

The SAD calculation unit for inter prediction can be extended to handle intra prediction. In the SAD calculation unit described in Fig. 5.13 two shifters are added after input registers A and B to align the active range of LOB computation. In inter prediction the active range is from LSB 4 to 1 while for intra prediction it is from LSB 5 to 2. The *HOC logic* is the same for both inter prediction and intra prediction. Thus, using two shifters, the proposed AD architecture can be configured for inter and intra prediction.

The proposed architecture has smaller critical path delay compared to the baseline architecture. First, the LOB and HOC units compute in parallel and the delay of the HOC logic unit is quite small. Next, the bit-widths of R1 (output of AD) and R2 (output of SAD) are reduced. The bit-width of R1 is reduced from 8 bits to 5 bits. As a result, the maximum SAD value is reduced from 16 bits to 13 bits. The last stage now has a carry save adder of size 13 bits which adds the first 5 bits of four AD values using full adders (6 bits due to carry propagation) and the last 7 bits using half adders. The new adder has smaller critical path and energy consumption compared to the general 16 bit adder. Furthermore, to reduce the critical path delay of the half adder chain, we merge two half adder modules as illustrated in Fig. 5.15. The carry-ripple path for a single 2 bit HA chain is now only one NAND and one NOR gate instead of two AND gates.

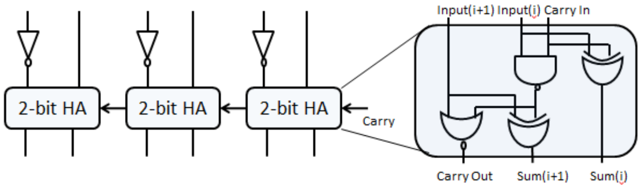


Fig. 5.15. 6-bit Half Adder Chain

We use high performance 45nm PTM models [40] to implement our system. The nominal voltage is equal to 1V. Delay and energy values of the basic gates are generated using HPSICE and PTM models. Then, delay and energy for the SAD architectures are computed for different input values using Verilog on ModelSim simulator [69]. For the addition and subtraction units, ripple carry adder (RCA) structure is used, because, for small bit widths, the energy delay product of RCA outperforms other adder structures.

We compare critical path delays vs average energy consumption of the proposed SAD unit with HOC (described in Fig. 5.13) and the baseline architecture (described in Fig. 5.12). The performance curves of the proposed and baseline schemes for different voltage levels are illustrated in Fig. 5.16. At nominal voltage, the proposed scheme can reduce critical path delay by 27% and average energy consumption by 28%. For iso-throughput, we scale the supply voltage of the proposed architecture to 0.8V, resulting in a 54% drop in energy as shown in Fig. 5.16.

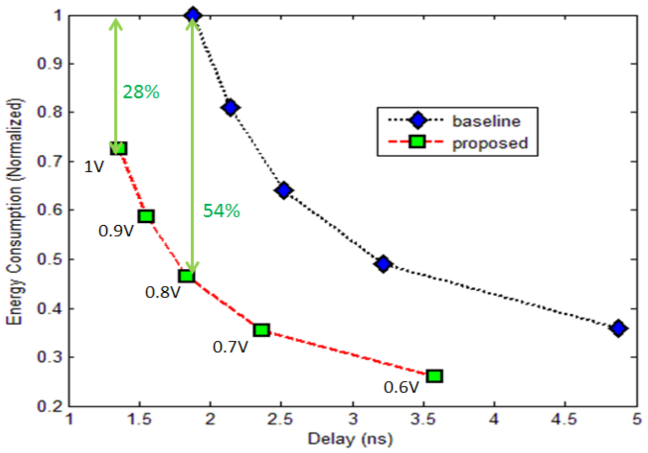


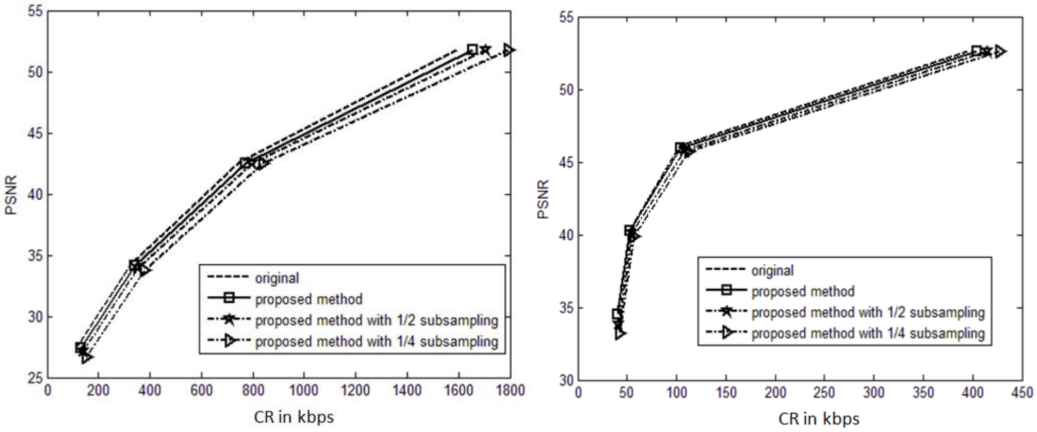
Fig. 5.16. Delay vs Energy Consumption of the baseline and proposed schemes

C. Performance Analysis of Proposed HOC

We use change in peak to noise ratio ($\Delta PSNR$) and change in compressed data rate (ΔCR) to evaluate the performance of the modified scheme. $\Delta PSNR$ is defined as the percentage increase in the PSNR value of the modified scheme compared to the baseline scheme averaged over different values of Q. Compressed data rate (CR) is defined as the size of the encoded data for 1 second of video and expressed in terms of bits per second (bps). ΔCR is defined as the percentage increase in compression rate compared to the baseline scheme. So both reduction in $\Delta PSNR$ and increase in ΔCR should be as small as possible.

Algorithm Performance

Fig. 5.17 illustrates the performance of the proposed technique for inter-prediction and its variants (subsampling described in Chapter 2) for 2 different video sequences (Football and Claire). In all cases, we find that the performance degradation is fairly low. There are several factors that help to achieve low degradation. The proposed scheme i) retains computations with low order bits, which contribute more effectively to the SAD computation of the selected block, ii) approximates the computation of higher order bits which does not affect the performance much since higher AD values typically correspond to blocks that are likely not to be selected. We also observe that ½ and ¼ sub-sampling schemes provide decent quality performance, which makes these schemes attractive especially since they do not require any change in the overall SAD architecture.



(a) (b)

Fig. 5.17. PSNR vs Compressed Data Rate of a) Football, b) Claire sequences under different configurations for inter-prediction

Next we present the performance results for the 12 video sequences. Fig. 5.18 illustrates the cost of using the proposed schemes in terms of percentage increase in compressed data rate and reduction in PSNR for inter prediction. We see that the proposed scheme without sub-sampling provides very good performance results. On average, it only increases ΔCR by 1.8% and reduces $\Delta PSNR$ by 0.06%. The proposed scheme with ½ sub-sampling increases ΔCR by 6.5%, on average, and reduces $\Delta PSNR$ by less than 1%. Table 5.5 and Table 5.6 summarize average performance degradations of the proposed scheme and its variants for inter and intra prediction, respectively. Note that the type of search has little effect on the ΔCR and $\Delta PSNR$ performance for inter prediction.

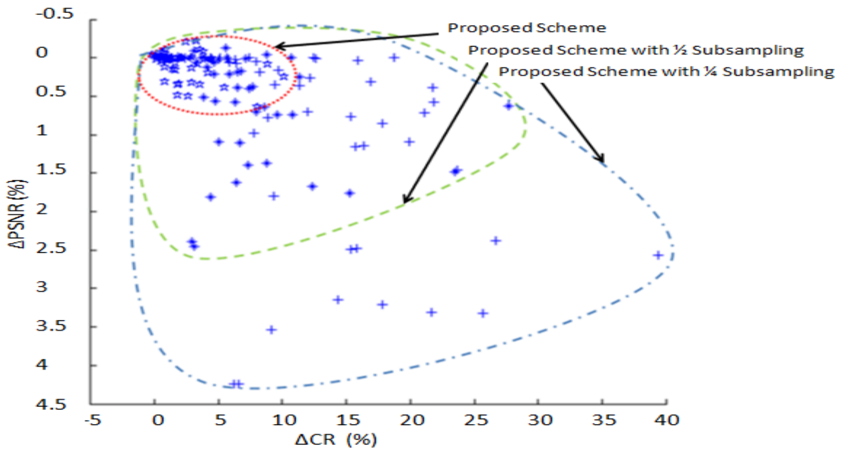


Fig. 5.18. Cost of using proposed scheme in terms of percentage increase in compressed data size and PSNR reduction for inter prediction.

Table 5.6 Average increase in compressed data rate and reduction in PSNR for inter-prediction

	HOC (Thr=32) with 1-bit LOB truncation		HOC (Thr=32) with 1-bit LOB truncation and ½ sub-sampling		HOC (Thr=32) with 1-bit LOB truncation and ¼ sub-sampling	
	DS	TSS	DS	TSS	DS	TSS
ΔCR (%)	1.8%	1.9%	6.5%	6.8%	12.4%	12.2%
ΔPSNR (%)	0.06%	0.06%	0.5%	0.6%	0.9%	1%

Table 5.7 Average increase in compressed data rate and reduction in PSNR for intra-prediction

	HOC (Thr=64) with 2-bit LOB truncation	HOC (Thr=64) with 2-bit LOB truncation and ½ sub-sampling
	Intra_4x4 and Intra_16x16	
ΔCR (%)	1%	4.8%
ΔPSNR (%)	0.07%	0.4%

D. Energy Quality Trade-off

Next, we compare the energy and algorithm quality of the proposed scheme and its variants with several candidate schemes. The candidate schemes are as follows:

A) LOB truncation. A.1: 1-bit truncation, A.2: 2-bit truncation, A.3: 3-bit truncation and A.4: 4-bit truncation. In each case, we modify the sizes of the internal computation blocks of the baseline SAD unit in Fig. 5.12.

B) Subsampling schemes. B.1: ½ sub-sampling, and B.2: ¼ sub-sampling. There is no modification in the baseline SAD unit

C) Proposed scheme. C.1: HOC (Thr=32 for inter and Thr=64 for intra) & LOB (1-bit for inter and 2-bit for intra), C.2: C.1 with ½ sub-sampling and C.3: C.1. with ¼ sub-sampling. The proposed SAD unit of Fig. 5.13 is used.

Fig. 5.19 illustrates $\Delta PSNR$ and ΔCR of all the candidate schemes in inter prediction and intra prediction. As expected, increasing LOB truncation introduce higher performance degradation. On the other hand, sub-sampling schemes result in quite high performance degradation compared to other techniques. The proposed SAD scheme (C.1) achieves less than 2% degradation for inter/intra prediction while its combination with the sub-sampling schemes achieves comparable performance with their counter-parts (B.1 & B.2).

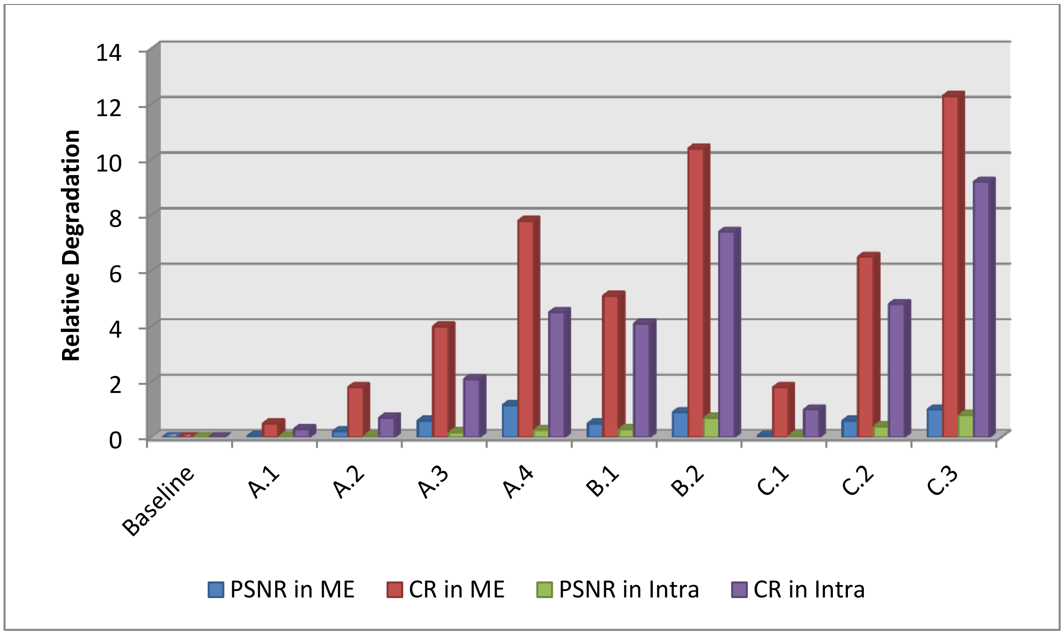


Fig. 5.19. Algorithm performance for different SAD architectures

Energy Consumption of SAD

Next, we compare the energy of the proposed SAD scheme and its variants with several candidate schemes. Each architecture is implemented using high performance 45nm PTM models [40] in which the nominal voltage equals to 1V. Delay and energy values of the basic gates are generated using HSPICE and PTM models. Then, delay and energy for each architecture is computed using Verilog on ModelSim simulator [69]. Fig. 5.20 illustrates the critical path delays, energy consumption per AD computation and total energy per SAD computation of 8x8 block of all the candidate schemes. As expected, increasing LOB truncation reduces energy and delay, and sub-sampling schemes reduce the total number of AD computations for a single search resulting in significant reduction in energy. Critical path delay of the configurations that achieves extra timing slack compared to baseline can be operated at lower voltages such as the proposed scheme C.1 which has 27% smaller latency. Furthermore, proposed schemes

C.2 and C.3 require 26% less energy compared to their counter parts B.1 and B.2 at the expense of slightly higher performance degradation.

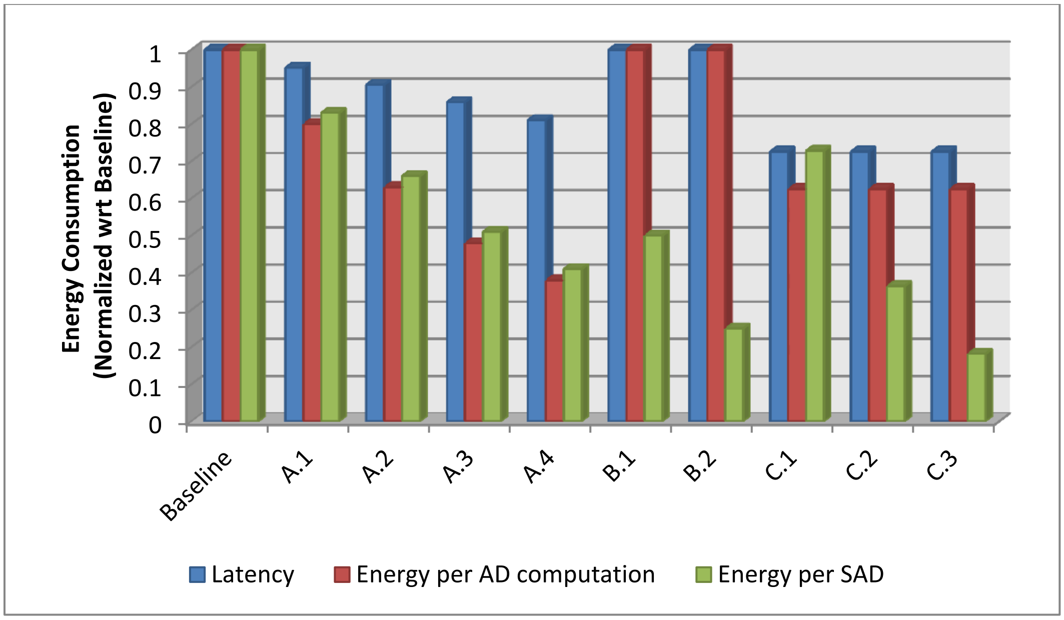


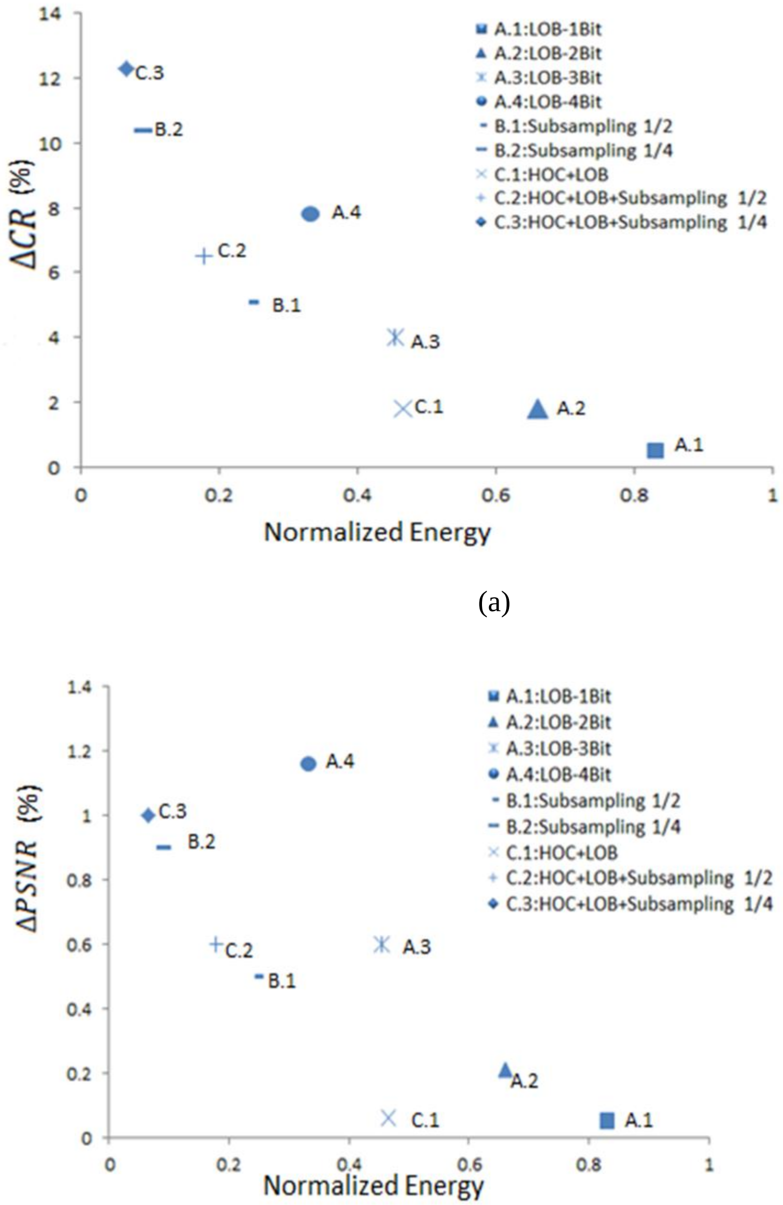
Fig. 5.20. Comparison of latency and energy consumption at nominal voltage of all candidate schemes

Fig. 5.21 and Fig. 5.22 compare the PSNR drop, CR increase and normalized energy consumption under iso-throughput conditions for inter and intra prediction modes respectively. To compute energy consumption for iso-throughput, we apply voltage scaling so that the delay is comparable to the baseline scheme. We compute the energy consumption and normalize it to that of the baseline scheme. From Fig. 5.21 and Fig. 5.22, on average, we see that the 4-bit LOB truncation scheme achieves 67% reduction in normalized energy. Sub-sampling by a factor of 2 (B.1) incurs 2.7% and 0.4% less increase in ΔCR for inter and intra prediction, respectively, and 0.6% less reduction in $\Delta PSNR$ for inter prediction for a comparable energy reduction. The proposed schemes achieve better energy and quality performance compared to single LOB and sub-sampling schemes. For instance, normalized energy consumption of C.2 is approximately

9% lower than B.1 for a comparable ΔCR. The normalized energy consumption of A.3 and C.1 are comparable but the ΔCR of C.1 is 2.2% and 1.1% lower compared to B.1 for inter and intra prediction, respectively. Also, C.1 achieves approximately 20% less normalized energy consumption compared to A.2 for comparable ΔCR. Of the other two proposed schemes, C.2 has 83% lower energy consumption with 6.5% and 4.8% increase in ΔCR for inter and intra prediction, respectively, and C.3 has 93% lower energy consumption with 12.5% and 92% increase in ΔCR for inter and intra prediction, respectively. While the trends are similar, the $\Delta PSNR$ values are very small for all the schemes and hence not discussed here. Overall, the proposed schemes achieve lower energy consumption with slightly higher performance loss compared to other candidate schemes.

V. SUMMARY

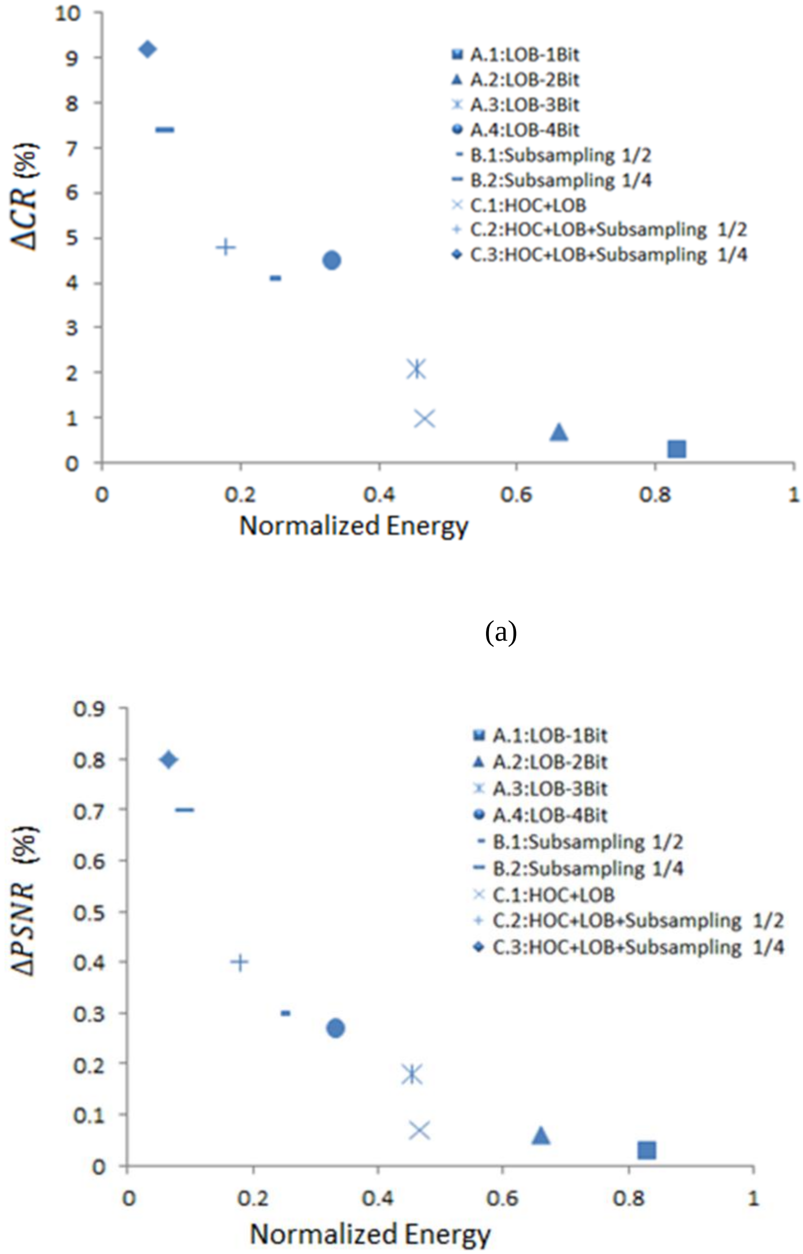
In this chapter, we focused on dynamic range reduction to reduce datapath energy. We analyzed the errors due to dynamic range reduction and proposed an unbiased estimator to minimize the effect of bit truncation errors for FIR filter and DCT. Furthermore, we also proposed a technique for MSB truncation that is applicable for SAD computation used in motion estimation and intra prediction in video codecs. The proposed scheme achieves 28% energy reduction at nominal voltage and 54% reduction for iso-throughput while incurring less than 2% increase in compressed data size and 0.1% reduction in PSNR. We also consider two variants of this scheme based on sub-sampling that further reduce the energy consumption.



(a)

(b)

Fig. 5.21. Inter prediction. a) Compressed Data Rate Increase (ΔCR) vs Normalized Energy, b) PSNR degradation ($\Delta PSNR$) vs Normalized Energy



(a)

(b)

Fig. 5.22. Intra prediction. a) Compressed Data Rate Increase (ΔCR) vs Normalized Energy, b) PSNR degradation ($\Delta PSNR$) vs Normalized Energy

CHAPTER 6

HYBRID ENERGY SAVING TECHNIQUES FOR DATAPATH

In this chapter, we investigate hybrid schemes that use a combination of voltage scaling, reduced computation and dynamic range reduction to achieve even higher energy saving with smaller performance degradation. The performance overhead and energy savings of each scheme are quantified and analyzed.

We present existing work on combination of computation reduction and voltage scaling in Section I. Then, we study the combination of computation reduction and dynamic range reduction for a sample signal processing system such as DCT transform in JPEG. We design a scheme that chooses which DCT coefficients have to be deactivated and the number of bits to be truncated based on the quality metric, Q. Next, we study the combination of voltage scaling and dynamic range reduction for some common DSP kernels in Section III. The main idea behind such a combination is that the errors due to increase in critical path delay during voltage scaling can be reduced by truncating the lower order bits which cause a reduction in the critical path. Furthermore, the noise that is introduced due to truncation is compensated by using an unbiased estimator that was introduced in Chapter 5.

I. COMBINING COMPUTATION REDUCTION AND VOLTAGE SCALING

Several significance driven techniques, where the significant components have shorter delay and the less significant components have longer delay, have been proposed in [35] [36] [70]. These techniques are very effective in reducing the energy consumption without significantly affecting the quality too much. At nominal voltage, all computations ensure no-violation in the critical path while at scaled voltage levels those which have higher critical path delay than what is allowed by the operating frequency, are disabled. For instance, selective deactivation of DCT coefficient based on reduced voltages has been proposed in [35]. Since low frequency DCT coefficients contain most of the input image energy, they are significant and implemented with the shortest critical path. Such a method results in 41% to 90% power saving compared to baseline scheme but with up to 10dB degradation in PSNR. A similar approach is applied in [36] for color interpolation where only less important computations are affected by voltage scaling and process variation. Such a scheme achieves 40% power savings with 5dB PSNR degradation.

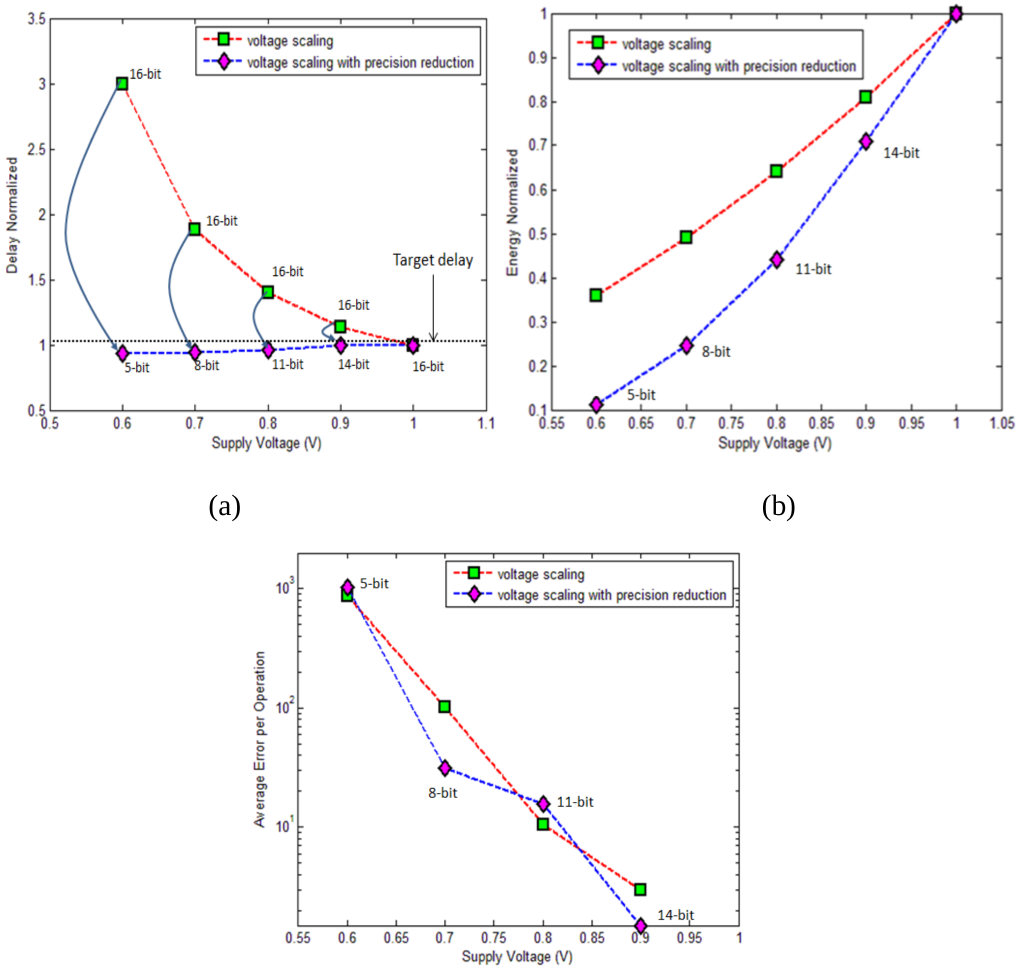
II. COMBINING VOLTAGE SCALING AND DYNAMIC RANGE REDUCTION

Voltage scaling and dynamic range reduction are two complementary energy reduction techniques. While voltage scaling reduces the energy consumption, it increases the delay of the computation unit and can cause timing errors. However, if reduced precision operation is acceptable, the critical path of the computation is lower and timing errors due to voltage scaling can be avoided. We illustrate this with the help of a 16 bit adder example and then show the effectiveness of this method in achieving energy reduction with minimal quality degradation for a low pass FIR filter.

Consider a simple 16-bit RCA adder implemented using 45nm PTM model and simulated for uniformly distributed inputs. Fig. 6.1a illustrates the change in critical path delay under voltage scaling; the target delay of the adder at nominal voltage and full precision is illustrated with a dashed line parallel to x-axis. As expected, the critical path delay of the 16-bit adder increases rapidly with voltage scaling. For instance, at 0.8V the increase in critical path delay is approximately 45% of the target delay.

The increase in critical path delay is reduced using lower precision arithmetic unit since it has shorter critical path. For instance, critical path violation of the 16-bit adder at 0.8V can be prevented by truncating 5 low order bits and operating only on the 11 MSBs. Similarly, critical path violation at 0.7V can be prevented by operating on the 8 MSBs. Fig. 6.1b shows the difference in energy saving between a scheme where only voltage scaling is used and a scheme where a combination of voltage scaling and reduced precision is used. At 0.6V, use of only voltage scaling reduces the energy consumption by 63% while the combination reduces it by 89%.

Next, we analyze the average error induced by voltage scaling and the combined technique. At nominal voltage, both systems operate at full-precision, and so the average error is zero. At scaled voltage levels, both systems have comparable average error per operation as shown in Fig. 6.1c. For instance, at 0.8V, the 11-bit adder and the 16 bit adder have the same error/operation but the 11-bit adder has 20% lower energy (Fig. 6.1b). Furthermore, the truncation noise can be lowered using the compensation technique described in Chapter 5. Even without compensation, the combined technique achieves much higher energy saving compared to when only voltage scaling is used for comparable noise levels. Note that while the results presented in Fig. 6.1 use uniformly distributed data; we expect this method to be equally effective for real image data.



(a)

(b)

(c)

Fig. 6.1. 16-bit RCA. a) Delay, b) Energy, c) Average Error per Operation distribution with and without precision reduction under voltage scaling

We see similar trends in delay, energy and error performance for more complex adders such as carry-look ahead adder (CLA). Fig. 6.2 shows normalized energy as a function of the supply voltage for a 16-bit CLA. We see that CLA supports more aggressive truncation. For instance, when operated at 0.8V, it uses only 8 bits (out of 16 bits) and thus achieves higher energy savings. However, the average error per operation for CLA is typically larger compared to RCA for the same voltage level and RCA tends to have better energy performance for the same error level. This is in agreement with the results presented in [67].

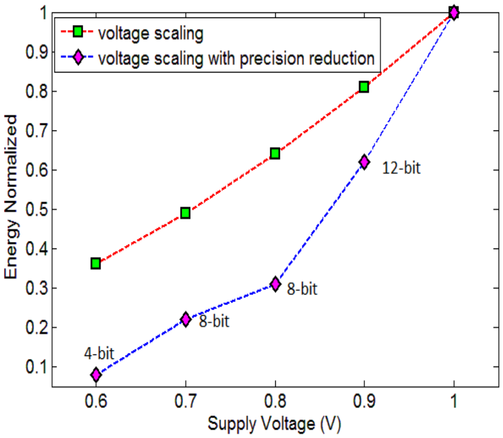
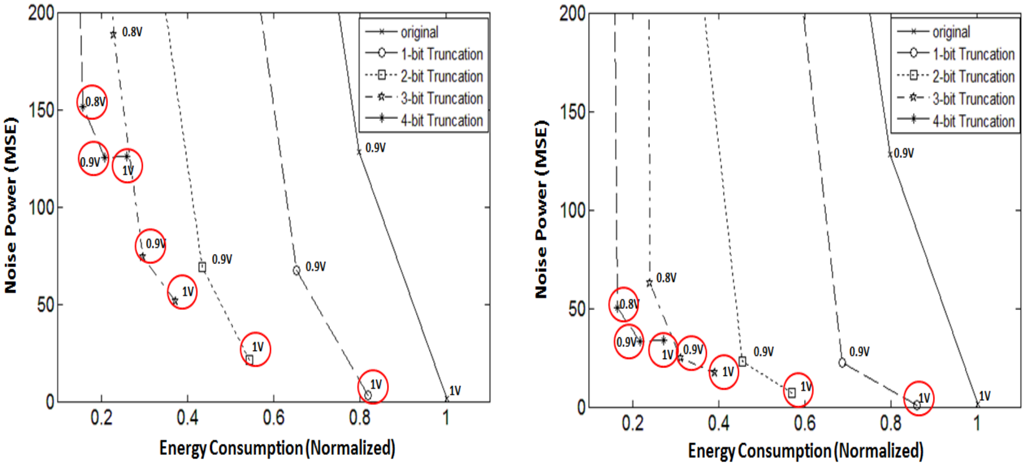


Fig. 6.2. 16-bit CLA. Energy distribution with and without precision reduction under voltage scaling for comparable average error per operation

Next, we present the results of this procedure on real image data. Consider processing the Lena image with a LPF using a MAC based architecture with 8-bit resolution. The LPF under consideration is a 3x3 Gaussian filter with $\sigma = 1.0$. The multiplier is implemented using a carry save adder tree and the final stage is implemented using RCA. Also, both the inputs and the filter coefficients are truncated with the same order. Fig. 6.3a shows the mean squared error noise power (VOS induced + truncation) vs. normalized energy consumption for various levels of low order bit truncation without compensation. Each point in the curve corresponds to a specific supply voltage level. Noise power is calculated using the mean square error (MSE) between the LPF results obtained with voltage scaling and those obtained with nominal voltage operation. From this figure, we see that full precision LPF (original) shows a large increase in noise level when the voltage is scaled to 0.9V with only about 20% energy saving. On the other hand, 2-bit truncation operating at 0.9V has lower noise power and 45% energy saving.

Thus, dynamic precision adjustment with voltage scaling achieves considerable better performance compared to when only voltage scaling is used.

Next, we study the effect of truncation noise compensation. Fig. 6.3b illustrates the performance of the LPF when the estimator described in Chapter 5 is applied. For 4-bit truncation operating at 0.9V, the noise power reduces by 66% when compensation unit is used. The overhead is very small, since the compensation unit is activated only 1/N of the time where N is the number of the filter taps. At full precision, we have approximately 5% overhead compared to original MAC unit because of the final adder illustrated in Fig. 5.7.



(a) (b)

Fig. 6.3. Noise Power vs Energy Consumption of 3x3 Gaussian filter on Lena image under voltage scaling with a) precision reduction, b) precision reduction with voltage scaling

Finally, Fig. 6.4 shows the pareto-optimal curves for voltage scaling in combination with truncation with and without compensation. These curves are generated by connecting the best configurations shown in circles in Fig. 6.3a and b. We see that the combination scheme always achieves better performance compared to sole voltage scaling at all levels. Furthermore, truncation with compensation achieves higher energy saving for the same noise power. For instance, at MSE=35, LPF using compensation achieves 20% extra energy saving compared to LPF with no compensation.

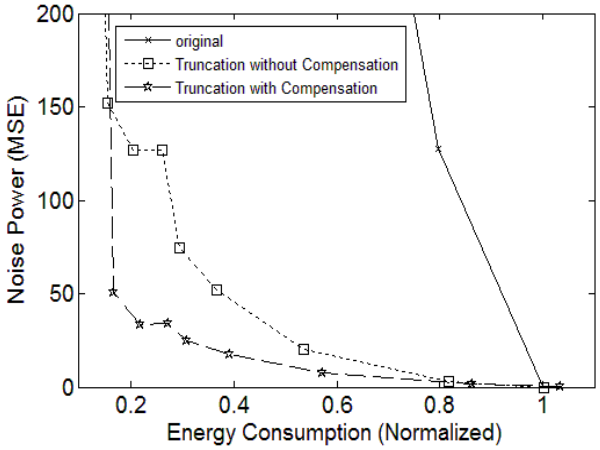


Fig. 6.4. Performance comparison of 3x3 Gaussian filter on Lena image; original and reduced precision filter with and without compensation

We repeat the analysis for three different 3x3 Gaussian filters ($\sigma = 0.5, \sigma = 1$ and $\sigma = 1.5$) and for two different MAC based architectures, one with a RCA in the final stage and the other with a CLA in the final stage. The MSE improvement is calculated as the difference between MSE of (VOS+truncation) with and without compensation. We use four sample images (Baboon, Lena, Flight, and Pepper) and list the average MSE improvement in Table 6.1. We see that the MAC with RCA has slightly

Table 6.1. MSE Improvement for different Gaussian Filters

	$\sigma = 0.75$	$\sigma = 1$	$\sigma = 1.25$
MAC with RCA	54%	66%	68%
MAC with CLA	51%	62%	63%

higher MSE improvement. While the MSE performance of the two MAC based systems is slightly different, both benefit from use of this technique.

We compare the performance of the proposed voltage scaling with dynamic range reduction technique with the ANT technique for FIR filtering [4]. The reduced computation block in the ANT system is a filter that uses 4 MSBs for both filter coefficients and input values. It consumes approximately 23% extra energy at nominal voltage but at 0.8V, the ANT system achieves 20% energy reduction for MSE=60 noise power. In comparison, the proposed technique achieves 85% energy reduction for the same level of noise power.

III. COMBINING COMPUTATION REDUCTION AND DYNAMIC RANGE REDUCTION

Computation reduction and dynamic range reduction techniques both try to keep significant computations while removing less significant portions of the computation. The combination is highly dependent on quality requirement and characteristics of the application. We illustrate this method using DCT as a case study.

Here the combination is based on DCT coefficient deactivation and low order bit truncation. The DCT architecture under consideration is given in Fig. 5.4. In DCT coefficient deactivation, DCT coefficients are deactivated starting from the highest frequency component of 1D DCT (W_7). Thus it is not possible to deactivate W_6 without deactivating W_7. In low order bit truncation, inputs are truncated for the entire computation unit with a granularity of 2-bit. These two techniques are combined in such a way that the performance degradation is minimized. Fig. 6.5 illustrates the proposed methods for 14 bit fixed point DCT implementation. The solid red line (L-shape) in Fig. 6.5 illustrates the scenario in which W_7 is deactivated and 4 low order bits are truncated in the rest of the coefficients. The above procedure can be implemented by controlling the AND gates at the inputs of each DCT coefficient computation unit as illustrated in Fig. 5.4.

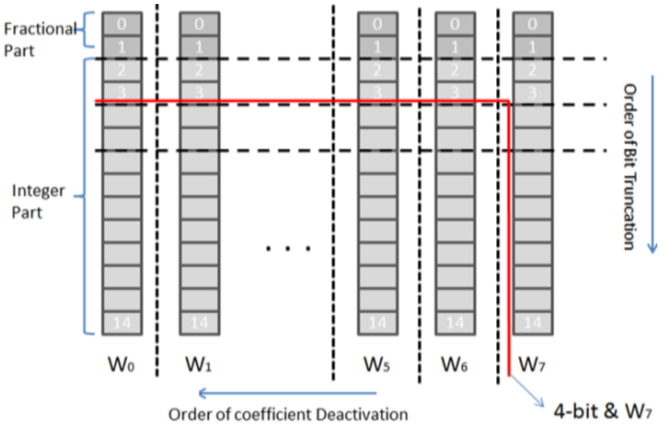
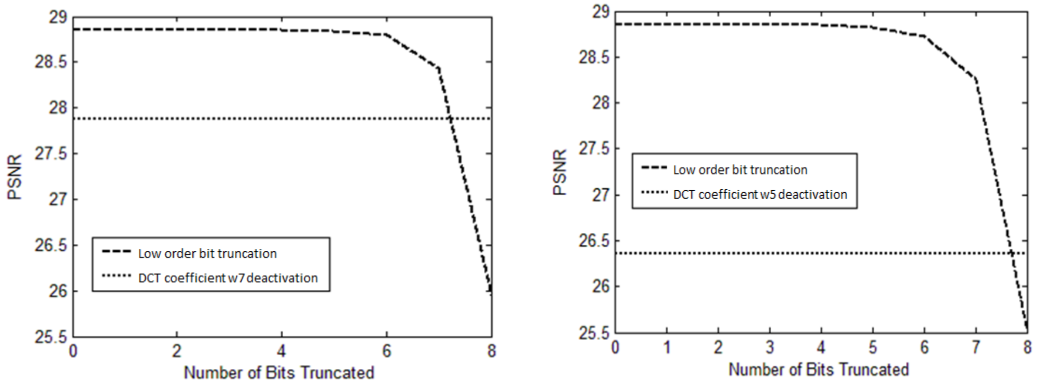


Fig. 6.5. DCT coefficient deactivation and low order bit truncation for the 1D-DCT coefficients

Next, we describe a scheme to combine coefficient deactivation and low order bit truncation. First we note that there is a crossover point in performance where it becomes better to deactivate a coefficient instead of applying aggressive bit truncation to that coefficient. Fig. 6.6 illustrates the PSNR performance of Baboon image as a function of low order bit truncation for W_5 and W_7 coefficients. We see that deactivation of W_5 and W_7 coefficients become more attractive after truncating 7 bits (out of 14). To improve confidence of the crossover point, we investigate the performance of 6 sample images (Baboon, Lena, Flight, Pepper, House and Bridge) and find that it is better to deactivate the DCT coefficient rather than truncating 6 bits. Thus, in our procedure, we limit the low order bit truncation to 4 levels with granularity of 2 bits, namely 0 bit (no truncation), 2 bit, 4 bit, and 6 bit truncation.



(a) (b)

Fig. 6.6. Comparison between bit truncation level and DCT coefficient deactivation for a) $W_{7,}$ b) W_5 for Baboon image when Q=55

Next, we determine the order in which coefficient deactivation and low order bit truncation is to be applied using a binary decision tree as illustrated in Fig. 6.7. We start from full precision, and at Level 1 choose between two competing schemes 2 bit low order truncation and W_7 deactivation based on PSNR. If 2 bit truncation provides better performance, then we pick that branch. In Level 2, we choose between 4 bit truncation (2+2) and W_7 deactivation with 2 bit low order bit truncation. If in Level 1, W_7 was deactivated, then in Level 2, we choose between W_7 deactivation and 2 bit truncation of all other coefficients or W_7 and W_6 deactivation.

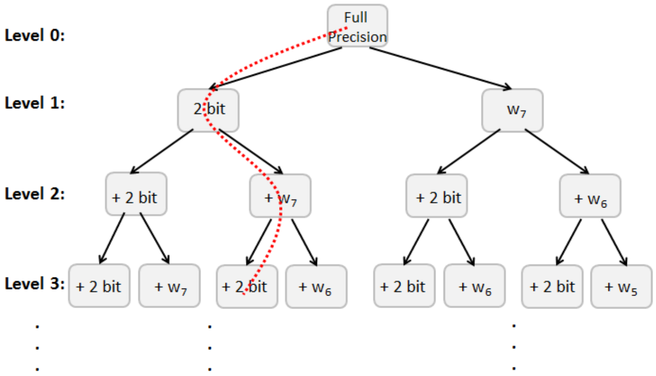
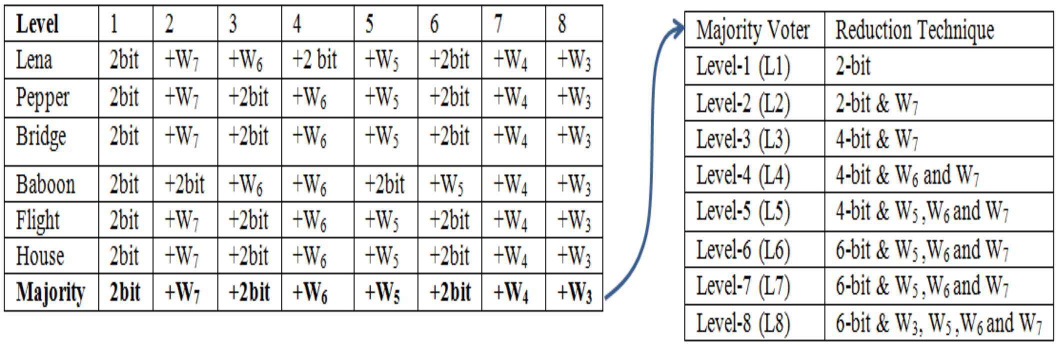


Fig. 6.7. Binary decision tree to choose combination of coefficient deactivation and truncation level

Table 6.2 lists the reduction order of 6 images using the binary decision tree method. First consider the table on the left. We read this table from left to right in increasing level number so Level 4 for Lena image corresponds to deactivation of coefficients W_6, W_7 and 2+2=4-bit truncation of all coefficients. Using majority voter scheme for each level (each column of Table 6.2), we form a general order which is given in the last row of Table 6.2. In this order, 2-bit low order truncation is followed by W_7 deactivation. Then, (2+2=) 4-bit low order bit truncation is applied to all the coefficients. Note that, since we consider the same computation units for the W_0 and W_4 pair to minimize the circuitry, we do not deactivate one of the members of this pairs unless both of them are eligible to be deactivated. Thus in Level 7, we do not deactivate W_4. The final reduction order for the first eight levels of majority voter result is illustrated in the table to the right.

Table 6.2. Reduction Order of 6 Sample Images



Level	1	2	3	4	5	6	7	8
Lena	2bit	+W_7	+W_6	+2 bit	+W_5	+2bit	+W_4	+W_3
Pepper	2bit	+W_7	+2bit	+W_6	+W_5	+2bit	+W_4	+W_3
Bridge	2bit	+W_7	+2bit	+W_6	+W_5	+2bit	+W_4	+W_3
Baboon	2bit	+2bit	+W_6	+W_6	+2bit	+W_5	+W_4	+W_3
Flight	2bit	+W_7	+2bit	+W_6	+W_5	+2bit	+W_4	+W_3
House	2bit	+W_7	+2bit	+W_6	+W_5	+2bit	+W_4	+W_3
Majority	**2bit**	**+W_7**	**+2bit**	**+W_6**	**+W_5**	**+2bit**	**+W_4**	**+W_3**

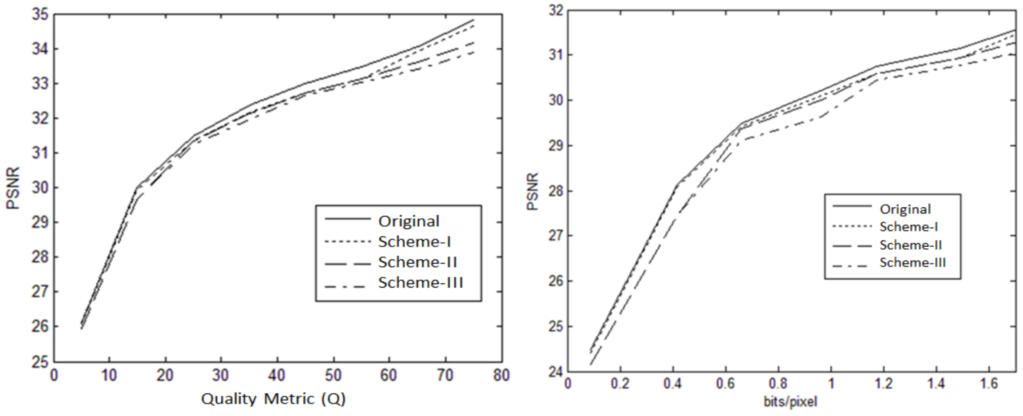
Majority Voter	Reduction Technique
Level-1 (L1)	2-bit
Level-2 (L2)	2-bit & W_7
Level-3 (L3)	4-bit & W_7
Level-4 (L4)	4-bit & W_6 and W_7
Level-5 (L5)	4-bit & W_5 ,W_6 and W_7
Level-6 (L6)	6-bit & W_5 ,W_6 and W_7
Level-7 (L7)	6-bit & W_5 ,W_6 and W_7
Level-8 (L8)	6-bit & W_3, W_5 ,W_6 and W_7

Using Table 6.2, we determine suitable configurations for three PSNR degradation schemes: i) Scheme-I (ΔPSNR<0.5dB), ii) Scheme-II (ΔPSNR<1dB), and iii) Scheme-III (ΔPSNR<1.5dB). Here, ΔPSNR is defined as the reduction in the PSNR value of the modified scheme compared to the baseline scheme. We use 6 sample images (Lena, Pepper, Bridge, Baboon, Flight and House) in our evaluation. For a given quality metric (Q) which is used in JPEG [44], we find all configurations that satisfy the PSNR constraint and choose the one that provides highest saving in computation. We use the majority voter order generated in Table 6.2 to determine the priority of a configuration. Table 6.3 lists the combination orders for the three schemes.

We test the effectiveness of the combination schemes given in Table 6.3 using five test images (Lake, Tank, Elaine, Feather and Boat). Fig. 6.8 illustrates the results for Elaine and Lake images. For instance, for Lake image at Q=50, Scheme II corresponding to $\Delta PSNR \leq 1dB,$ results in PSNR of 32.7 dB, which is only 0.6dB lower than the original PSNR. Thus the proposed method guarantees that the PSNR constraints are satisfied for Q values from 75 down to 5.

Table 6.3. Final Order for Scheme I, II and III

Q (Quality Metric)	75	65	55	45	35	25	15	5
Scheme-I (ΔPSNR<0.5dB)	L2	L2	L3	L3	L3	L3	L4	L6
Scheme-II (ΔPSNR<1dB)	L3	L3	L3	L3	L4	L4	L5	L8
Scheme-III (ΔPSNR<1.5dB)	L4	L4	L4	L4	L5	L5	L6	L8



(a) (b)

Fig. 6.8. Performance of the resulting decision order generated using 6 training samples on test images a) Lake, and b) Elaine

Next, we calculate the power consumption of the original and proposed schemes for different configurations. All multiplications are implemented using carry save adder structures. We set the clock frequency at 250MHz for all the combinations. Table 6.4 lists active power, latency and area estimations for the configuration order given in Table 6.2 using Synopsys Design Compiler [68] with 45nm Nangate technology [66]. The proposed schemes have marginal increase in circuitry area (2.9%) and leakage (3.6%) compared to the original implementation due to extra units that are used to gate inputs and compensate truncation error. Overall, the proposed scheme provides flexible performance with reduced power consumption for different quality requirements. For instance, using configuration order of L8, we save 61% power consumption and have 14% extra timing slack compared to original full precision DCT engine. The timing slack can be absorbed by operating at 0.9V (instead of 1V) resulting in 68% saving in power consumption.

Table 6.4. Delay, Power and Area for Original and Reduced Computation 1D-DCT

		Proposed Scheme: Reduced Computation +Truncation							
	Original	Baseline	L1	L2	L3	L4	L5	L6	L8
Power (mW)	5.39	5.51	4.76	4.20	3.78	3.32	2.68	2.13	1.51
Latency (ns)	2.92	3.11	2.95	2.95	2.78	2.78	2.51	2.51	2.51
Area (um^2)	4435	4539							

Finally, we present the power savings of Schemes I, II and III for Q values from 75 to 5 in Fig. 6.9. We see that Scheme III always achieves the highest power saving due to higher allowable degradation. As we move from high quality (Q large) regions to low quality regions (Q small), we see an increase in the power savings. This is because the combinations used in the low quality regions are quite aggressive in terms of power savings. On average, we achieve 33% power saving for Scheme I, 39% power saving for Scheme II and 46% power saving for Scheme III.

We compare the proposed scheme with the ANT technique [5], significance driven technique in [35] and adaptive truncation technique in [23]. ANT based scheme using 4-bit MSB replica of the DCT in the reduced computation block has a 16% overhead compared to original DCT which reduces the power saving at scaled voltage levels [35]. It achieves 20% power saving with 5dB degradation in PSNR. Significance driven technique achieves 47% energy reduction with more than 4dB degradation in PSNR [35]. The truncation based technique in [23] achieves up to 40% energy saving while inducing approximately 0.2 reduction in MSSM which corresponds to approximately 15dB loss in PSNR. In contrast, the proposed combination scheme can achieve average energy savings of 46% with <1.5dB PSNR degradation. Moreover, the proposed scheme provides a mechanism for higher energy saving for low Q settings while keeping the degradation low for high Q settings.

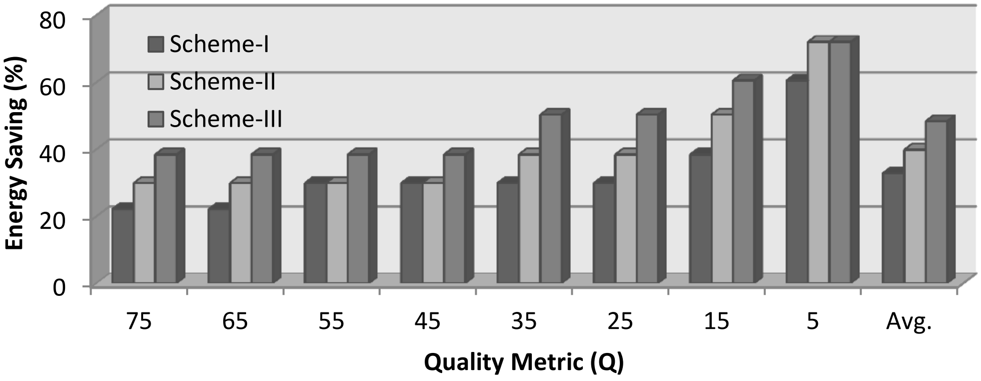


Fig. 6.9. Energy saving at different quality levels for three candidate schemes

IV. SUMMARY

In this chapter, we described several hybrid techniques that further reduce energy consumption while causing little reduction in quality. We investigated the combination of voltage scaling and dynamic range reduction and applied it to a low pass FIR filter. The proposed scheme achieved 85% energy saving for fairly low noise level. We also studied the combination of computation reduction and dynamic range reduction for DCT used in JPEG. Simulation results showed, on average, 33% to 46% reduction in energy consumption for a small 0.5dB to 1.5dB degradation in the system performance. Thus algorithm-level optimizations can help reduce the energy consumption of many multimedia signal processing algorithms with only a mild degradation in quality.

CHAPTER 7

CONCLUSION

The number of portable multimedia devices has dramatically increased in the last two decades. Many of these devices consume significant amount of energy. To increase the lifetime of these devices, the energy consumption has to be reduced without compromising on the quality. In this thesis, we presented several techniques that successfully reduce the energy consumption of image and video codecs with minimum quality degradation. The techniques were based on aggressive use of voltage scaling, reducing dynamic range and reducing number of computations. Since these techniques result in computation errors that effect quality, a great deal of emphasis is placed on compensating for these errors.

The first problem that we addressed is use of voltage scaling to reduce energy consumption of SRAM memories while minimizing quality degradation. We choose JPEG2000 as the target application since it has large memory requirements and is a popular image coding standard. First, the errors due to voltage overscaling in SRAM memories are analyzed. Majority of the errors are aggravated by process variations. These include variation of the transistor sizes (W/L) and variation in V_{th} due to random dopant fluctuation (RDF). Next, techniques to mitigate memory failures are presented. New unequal error protection (UEP) schemes are proposed that achieve better performance and lower overhead compared to generic ECC schemes. These schemes code bit planes of different DWT sub-bands according to their importance. However these schemes do not have good performance for high error rates and also have large storage overhead. Next, algorithm-specific techniques are presented which require no additional memory storage, 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 characteristics of low and high frequency subband outputs to identify and correct memory errors. These techniques enable us to drop the operating voltage of memory with modest reduction in image quality for low to medium compression rates. For instance, they enable the memory to be operated at 0.7V 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 [13]) with only 1dB loss in PSNR. If 4dB loss in PSNR is acceptable, they enable the memory to operate at 0.6V resulting in 78% memory power saving and 31% overall power saving.

The second problem that we addressed is the use of voltage overscaling in DSP data-paths to reduce energy consumption. We focused on the 2D DCT module in the JPEG codec. We saw that most of the errors due to voltage overscaling occur at MSBs which result in very high performance degradation. Next, a low overhead algorithm-specific method to compensate for most of these errors is presented. The proposed technique exploits the characteristics of the quantized DCT coefficients. Simulation results show that operating at 0.83V (instead of the nominal 1V) results in a 20% reduction in datapath energy but causes BER(VOS) of 10^{-4}. The proposed technique improves PSNR performance by approximately 3.4dB compared to the no-correction case but has a degradation of about 1.2 dB in PSNR compared to the error-free case.

The third problem that we addressed is datapath energy reduction through dynamic range reduction. We first analyzed errors due to lower order bit truncation and proposed a procedure based on pre-computing the mean of the truncation error and compensating for it at the end of the computation. Such a procedure reduces the noise power significantly for operations such as additions and multiplications. Simulation results show that in DCT, computation with 4 bit truncated data achieves 23% power saving with only 0.6dB drop in PSNR. We also proposed a technique for MSB truncation that is applicable for SAD computation used in motion estimation and intra prediction in video codecs. The proposed technique exploits the features of AD and SAD distributions to derive a scheme that carefully approximates the computations in the SAD unit. If the AD value is larger than a threshold, it is likely to contribute to the SAD value of an unselected block. Thus large AD values can be approximated and the corresponding SAD value updated with a correction term. The resulting architecture has a lower critical path delay compared to the baseline architecture and significantly lower energy consumption. It achieves 28% energy reduction at nominal voltage and 54% reduction for iso-throughput while incurring less than 2% increase in compressed data size and approximately 0.1% reduction in PSNR.

Next, we described several hybrid techniques that further reduce energy consumption while causing little reduction in quality. We investigated the combination of voltage scaling and dynamic range reduction and applied it to a low pass FIR filter. The proposed scheme achieved 85% energy saving for fairly low noise level. We also studied the combination of computation reduction and dynamic range reduction for DCT used in JPEG. Simulation results showed, on average, 33% to 46% reduction in energy consumption for a small 0.5dB to 1.5dB degradation in the system performance. Thus algorithm-level optimizations can help reduce the energy consumption of many multimedia signal processing algorithms with only a mild degradation in quality.

Future Work

The schemes proposed in this work can be applied to other power greedy multimedia applications such as speech coding (CELP, G-723), image/video segmentation and recognition etc. Like video coding, speech coding uses correlation between consecutive data samples to remove the temporal redundancy. Thus, coded data has certain characteristics that can be exploited to compensate for errors introduced due to voltage scaling or truncation.

Voltage scaling has been used to effectively reduce the energy consumption in this work. This automatically leads us to near threshold computing (NTC) where the voltage is scaled to a couple of hundred millivolts above the threshold voltage. In this voltage domain, the energy efficiency improves by more than 100X [71]. The main problem of NTC is the 10X increase in delay. So far, it has been mainly applied to nonperformance-critical modules such as sensors, low performance processors where the clock rate is on the order of couple of MHz [72]. Now, most multimedia systems (video encoders) require much higher clock frequency to complete the processing within the time frame. One way to overcome this is to use data level parallelism as has been done for diet-SODA [73]. Unfortunately, such parallelism results in very high area overhead. It may be possible to reduce the burden of this overhead using algorithm-specific techniques like those discussed in this thesis. Combining data-parallelism with algorithm-specific techniques will enable us to design the next generation of truly low power systems.

Works Cited

[1] S. Ghosh and K. Roy, "Parameter Variation Tolerance and Error Resiliency: New Design Paradigm for the Nanoscale Era," *Proceedings of the IEEE*, pp. 1718-1751, October 2010.

[2] S. H. Nawab, A. V. Oppenheim, A. Chandrakasan, J. M. Winograd, and J. T. Ludwig, "Approximate Signal Processing," *Journal of VLSI Signal Processing*, vol. 15, pp. 177- 200, 1997.

[3] A. Chandrakasan and R. Brodersen, "Minimizing Power Consumption in Digital CMOS circuits," *Proceedings of the IEEE*, pp. 498-523, April 1995.

[4] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Transactions on VLSI Systems*, vol. 9, no. 12, pp. 813–823, December 2001.

[5] B. Shim, S. R. Sridhara, and N. R. Shanbhag, "Reliable Low-Power Digital Signal Processing via Reduced Precision Redundancy," *IEEE Transactions on VLSI Systems*, vol. 12, pp. 497-510, May 2004.

[6] G. Varatkar and N. R. Shanbhag, "Error-resilient motion estimation architecture," *IEEE Transaction on VLSI Systems*, vol. 16, no. 10, pp. 1399–1412, October 2008.

[7] N. R. Shanbhag, R. A. Abdallah, R. Kumar, and D. L. Jones, "Stochastic Computation," in *Design Automation Conference*, 2010, pp. 859-864.

[8] Y. Emre and C. Chakrabarti, "Data-path and Memory Error Compensation Tecnhiques for Low Power JPEG Implementation," in *International Conference on Acoustic, Speech and Signal Processing*, 2011, pp. 1589-1592.

[9] Y. Emre and C. Chakrabarti, "Memory error compensation techniques for JPEG2000," in *IEEE Workshop on Signal Processing Systems*, 2010, pp. 36-41.

[10] I. J. Chang, D. Mohapatra, and K. Roy, "A Voltage-Scalable & Process Variation Resilient Hybrid SRAM Architecture for MPEG-4 Video Processors," in *Design and Automation Conference*, 2009, pp. 670-675.

[11] M. Cho, J. Schlessman, W. Wolf, and S. Mukhopadhyay, "Accuracy-aware SRAM: a reconfigurable low power SRAM architecture for mobile multimedia applications," in *Asia and South Pacific Design Automation Conference*, 2009, pp. 823-828.

[12] Z. Chishti, A. R. Alameldeen, C. Wilkerson, W. Wu, and S. Lu, "Improving Cache Lifetime Reliability at Ultra-low Voltages," in *International Symposium on Microrchitecture (MICRO)*, 2009, pp. 89-99.

[13] M. A. Makhzan, A. Khajeh, A. Eltawil, and F. J. Kurdahi, "A Low Power JPEG2000 Encoder with Iterative and Fault Tolerant Error Concealment," *IEEE Transactions on VLSI Systems*, vol. 17, no. 6, pp. 827-837, June 2009.

[14] K. J. Lin, S. Natarajan, and J. W. S. Liu, "Imprecise results: Utilizing partial computations in real-time systems," in *8th Real-Time System Symposium*, 1987, pp. 210-217.

[15] J. Sartori and R. Kumar, "Architecting Processors to Allow Voltage/Reliability Tradeoffs," in *Int. Conf. on Compiler Architectures and Synthesis for Embedded Systems*, 2011, pp. 115-124.

[16] C. Chen et al., "Analysis and Architecture Design of Variable Block-Size Motion Estimation for H.264/AVC," *IEEE Transactions on Circuit and System-I*, vol. 53, no. 2, pp. 578-593, February 2006.

[17] Y. Emre and C. Chakrabarti, "Low energy motion estimation via selective approximation," in *IEEE Int. Conf. on Application-Specific Systems, Architectures and Processors*, 2011, pp. 176-183.

[18] C. Lian and et al., "Power-aware Multimedia: Concepts and Challenges," *IEEE Circuits and Systems magazine*, vol. 7, no. 2, pp. 26-34, 2007.

[19] Y. Andreopoulos and M. van der Schaar, "Incremental Refinement of Computation for the Discrete Wavelet Transform," *IEEE Transactions on Signal Processing*, vol. 56, pp. 140-157, January 2008.

[20] A. Sinha, A. Wang, and A. Chandrakasan, "Energy scalable system design," *IEEE Transactions on VLSI Systems*, vol. 10, pp. 135-145, April 2002.

[21] J. Park, J. H. Choi, and K. Roy, "Dynamic Bit-Width Adaptation in DCT: An Approach to Trade Off Image Quality and Computation Energy," *IEEE Transactions on VLSI Systems*, vol. 18, no. 5, pp. 787-793, May, 2010.

[22] J. Y. F. Tong, D. Nagle, and R. A. Rutenbar, "Reducing Power by Optimizing the Necessary Precision/Range of Floating Point Arithmetic," *IEEE Transactions on VLSI Systems*, vol. 8, pp. 273-286, June 2000.

[23] S. H. Kim, S. Mukhopadhyay, and M. Wolf, "System Level Energy Optimization for Error Tolerant Image Compression," *IEEE Embedded System Letters*, vol. 2, pp. 81-84, September 2010.

[24] Z. He, C. Tsui, K. Chan, and M. L. Liou, "Low-Power VLSI Design for Motion Estimation Using Adaptive Pixel Truncation," *IEEE Transactions on Circuits and Systems For Video Technology*, vol. 10, no. 5, pp. 669-678, August 2000.

[25] D. Ernst and et al., "Razor: a low-power pipeline based on circuit-level timing speculation," in *International Symposium on Microarchitecture (MICRO)*, 2003, pp. 7-18.

[26] 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.

[27] G. K. Chen, D. Blaauw, T. Mudge, D. Sylvester, and N. S. Kim, "Yield-driven near-threshold SRAM design," in *Int. Conf. Computer-Aided Design*, 2007, pp. 660–666.

[28] A. Agarwal, B. C. Paul, S. Mukhopadhyay, and K. Roy, "Process Variation in Embedded Memories: Failure Analysis and Variation Aware Architecture," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1804-1813, September 2005.

[29] T.R.N. Rao and E. Fujiwara, *Error Control Coding for Computer Systems*. Prentice Hall, 1989.

[30] Y. Emre and C. Chakrabarti, "Techniques for Compensating Memory Errors in JPEG2000," *IEEE Transactions on VLSI Systems*, accepted for publication.

[31] P. Schelkens, A. Skodras, and T. Ebrahimi, *The JPEG 2000 Suite*.: Wiley, 2009.

[32] T. Xanthopoulos and A. Chandrakasan, "Low-Power DCT Core Using Adaptive Bitwidth and Arithmetic Activity Exploiting Signal Correlations and Quantization," *IEEE Journal of Solid State Circuits*, vol. 35, no. 5, pp. 740-750, May 2000.

[33] Y. Emre and C. Chakrabarti, "Quality-Aware Techniques for Reducing Power of JPEG Codecs," *Journal of Signal Processing Systems*, accepted for publication.

[34] Y. Emre and C. Chakrabarti, "Energy and Quality-Aware Multimedia Signal Processing," *IEEE Transaction on Multimedia*, under revision.

[35] G. Karakonstantis, N. Banerjee, and K. Roy, "Process-variation resilient and voltage scalable DCT architecture for robust low-power computing," *IEEE Transactions on VLSI Systems*, vol. 18, no. 10, pp. 1461–1470, 2010.

[36] N. Banerjee and et al., "Design methodology for low power dissipation and parametric robustness through output quality modulation: application to color interpolation filtering," *IEEE Transactions on CAD for Integrated Circuits and Systems*, vol. 28, no. 8, pp. 1127-1137, August 2009.

[37] K. Zhang, *Embedded Memories for Nanoscale VLSIs*.: Springer , 2009.

[38] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*,

3rd ed. Addison Wesley, 2004.

[39] K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. Wiley-Interscience, 1999.

[40] Yu Cao. [Online]. ptm.asu.edu

[41] C. Piguet, *Low-Power Electronics Design*.: CRC Press, 2004.

[42] TI. [Online]. http://focus.ti.com/pdfs/wtbu/smartreflex_whitepaper.pdf

[43] M. Mehendale and S. Sherlekar, *VLSI Synthesis of DSP Kernels: Algorithmic and Architectural Transformations*. Springer, 2001.

[44] William B. Pennebaker and Joan L. Mitchell, *JPEG: Still Image Data Compression Standard*. Springer, 1992.

[45] T Acharya and P.-S. Tsai, *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*. Wiley Inter-Science, 2004.

[46] The independent JPEG Group, "The sixth public release of independent JPEG Group's Free JPEG Software," C Source code of JPEG Encoder research 6b, (ftp://ftp.uu.net/graphics/jpeg) March 1998.

[47] I. E Richardson, *H.264 and MPEG-4 Video Compression*. John Wiley & Sons, 2003.

[48] G. Bjontegaard and K. Lillevold, "Context-Adaptive VLC Coding of Coefficients," Fairfax, VA, JVT Document JVT-C028 May 2002.

[49] International Telecommunication Union. [Online]. http://www.itu.int/rec/T-REC-H.264/en

[50] S. Yang, W. Wolf, and N. Vijaykrishnan, "Power and Performance Analysis of Motion Estimation Based on Hardware and Software Realization," *IEEE Transactions on Computers*, vol. 54, no. 6, pp. 714-726, June 2006.

[51] R. Li, B. Zeng, and M. L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuits and Systems For Video Technology*, vol. 4, no. 4, pp. 438-442, August 1994.

[52] S. Zhu and K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 287-290, February 2000.

[53] J. Karlsson, P. Liden, P. Dahlgren, R. Johanson, and U. Gunneflo, "Using heavy-ion radiation to validate fault handling mechanism," *IEEE Transactions in*

Microelectronics, vol. 14, pp. 8-23, 1994.

[54] R. Reed and et. al, "Heavy ion and proton-induced single event multiple upset," *IEEE Transactions Nuclear Science*, vol. 44, no. 6, pp. 2224-2229, December 1997.

[55] H. Yamauchi, "A discussion on SRAM circuit design trend in deeper nanometer-scale technologies," *IEEE Transactions on VLSI*, vol. 18, no. 5, pp. 763 - 774, May 2010.

[56] S. R. Sarangi and et al., "VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 1, pp. 3 - 13, February 2008.

[57] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of Failure Probability and Statistical Design of SRAM Array for Yield Enhancement in Nanoscaled CMOS," *IEEE Transaction on CAD of Integrated Circuits and Systems*, vol. 24, no. 12, pp. 1859-1880, December 2005.

[58] K. Agarwal and S. Nassif, "Statistical Analysis of SRAM cell stability," in *IEEE Design Automation Conference*, 2006, pp. 57-62.

[59] L. Chang and et al., "Stable SRAM cell design for the 32 nm node and beyond," in *Symp. VLSI Technology Digest*, 2005, pp. 128–129.

[60] R. Aly, M. Faisal, and M. A. Bayoumi, "Novel 7T SRAM cell for low power cache design," in *IEEE SOC Conference*, 2005, pp. 171–174.

[61] S. E. Schuster, "Multiple word/bit line redundancy for semiconductor memories," *IEEE Journal of Solid-State Circuits*, vol. 13, pp. 698–703, October 1978.

[62] C. Wilkerson et al., "Trading off Cache Capacity for Reliability to Enable Low Voltage Operation," in *International Symposium on Computer Architecture*, 2008, pp. 203-214.

[63] J. Kim, N. Hardavellas, K. Mai, B. Falsafi, and J. C. Hoe, "Multi-bit Tolerant Caches Using Two Dimensional Error Coding," in *International Conference on Microarchitecture (MICRO)*, 2007, pp. 197-209.

[64] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Error-Resilient Image and Video Transmission over the Internet Using Unequal Error Protection," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 121-131, February 2003.

[65] A. A. Alatan, M. Zhao, and A. N. Akansu, "Unequal Error Protection of SPIHT Encoded Image Bitstream," *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 6, pp. 814-817, June 2000.

[66] Nangate Sunnyvale California. 45nm Open Cell Library. [Online]. http://www.nangate.com/

[67] Y. Liu, T. Zhang, and K. K. Parhi, "Computation ErrorAnalysis in Digital Signal Processing Systems with Overscaled Supply Voltage," *IEEE Transactions on VLSI Systems*, vol. 18, no. 4, pp. 517-526, April 2010.

[68] Synopsys. [Online]. http://www.synopsys.com

[69] ModelSim HDL Simulator. [Online]. http://model.com/content/modelsim-pe-student-edition-hdl-simulation

[70] D. Mohapatra, G. Karakonstantis, and K. Roy, "Significance driven computation: A voltage-scalable, variation-aware, quality-tuning motion estimator," in *International Symposium on Low Power Electronics and Design*, 2009, pp. 195-2000.

[71] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-Threshold Computing:Reclaiming Moore’s Law Through Energy Efficient Integrated Circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253-266, February 2010.

[72] M. Seok et al., "The phoenix processor: A 30 pW platform for sensor applications," in *IEEE Symposium on VLSI Circuits*, 2008, pp. 188-189.

[73] S. Seo et al., "Diet SODA: A power-efficient processor for digital cameras," in *IEEE International Symposium on Low Power Electronics and Design*, 2010, pp. 79-84.