Copyright

by

Ku He

2012

The Dissertation Committee for Ku He Certifies that this is the approved version of the following dissertation:

Adaptive Low-Energy Techniques in Memory and Digital Signal Processing Design

Committee:

Michael Orshansky, Co-Supervisor

Andreas Gerstlauer, Co-Supervisor

Adnan Aziz

Constantine Caramanis

Rouwaida Kanj

Adaptive Low-Energy Techniques in Memory and Digital Signal Processing Design

by

Ku He, B.E., M.E.

Dissertation

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

The University of Texas at Austin
May 2012

Dedication

To my family

Acknowledgements

First, I would like to thank my research advisors: Prof. Michael Orshansky and Prof. Andreas Gerstlauer, for their guidance and support during my time in graduate school at the University of Texas at Austin.

Secondly, I would like to thank the committee members of the Chinese Student and Scholar Association (CSSA) at the University of Texas, as well as the members of Austin Tsinghua Alumni Association (ATAA). We have worked together to hold many successful events.

Thirdly, I would like to thank my friends in Austin, and also my college classmates in the US. Your support helped me to overcome many difficulties during my research in the last five years.

Finally, I am grateful to my family for their love and steady support. I am proud of all of them: my parents Hualin He and Fuyu Liu, sister Xiangyu He, brother-in-law Shuanglin Liu, niece Muen Liu, uncle Huaxin He, aunt Di Yang, cousins Le He and Qian He.

Adaptive Low-Energy Techniques in Memory and Digital Signal Processing Design

Ku He, Ph.D.

The University of Texas at Austin, 2012

Co-Supervisors: Andreas Gerstlauer and Michael Orshansky

As semiconductor technology continues to scale, energy-efficiency and power consumption have become the dominant design limitations, especially, for embedded and portable systems. Conventional worst-case design is highly inefficient from an energy perspective. In this dissertation, we propose techniques for adaptivity at the architecture and circuit levels in order to remove some of these inefficiencies. Specifically, this dissertation focuses on research contributions in two areas: 1) the development of SRAM models and circuitry to enable an intra-array voltage island approach for dealing with large random process variation; and 2) the development of low-energy digital signal processing (DSP) techniques based on controlled timing error acceptance.

In the presence of increased process variation, which characterizes nanometer scale CMOS technology, traditional design strategies result in designs that are overly conservative in terms of area, power consumption, and design effort. Memory arrays, such as SRAM-based cache, are especially vulnerable to process variation, where the penalty is a power and bit-cell increase needed to satisfy a variety of noise margins. To improve yield and reduce power consumption in large SRAM arrays, we propose an intra-array voltage island technique and develop circuits that allow for a cost-effective deployment of this technique to reduce the impact of process variation. The voltage tuning architecture makes it possible to obtain, on average, power consumption reduction of 24% iso-area in the active mode, and the leakage power reduction up to 52%, and, on average, of 44% iso-area in the sleep mode. Alternatively, bitcell area can be reduced up to 50% iso-power compared to the existing design strategy.

In many portable and embedded systems, signal processing (SP) applications are dominant energy consumers. In this dissertation we investigate the potential of error-permissive design strategies to reduce energy consumption in such SP applications. Conventional design strategies are aimed at guaranteeing timing correctness for the input data that triggers the worst-case delay, even if such data occurs infrequently. We notice that an intrinsic notion of quality floor characterizes SP applications. This provides the opportunity to significantly reduce energy consumption in exchange for a limited signal quality reduction by strategically accepting small and infrequent timing errors. We propose both design-time and run-time techniques to carefully control the quality-energy tradeoff under scaled V_{DD}. The basic philosophy is to prevent signal quality from severe degradation, on average, by using data statistics. We introduce techniques for: 1) static and dynamic adjustment of datapath bitwidths, 2) design-time and run-time reordering of computations, 3) protection of important algorithm steps, and 4) exploiting the specific patterns of errors for low-cost post-processing to minimize signal quality degradation. We demonstrate the effectiveness of the proposed techniques on a 2D-IDCT/DCT design, as well as several digital filters for audio and image processing applications. The designs were synthesized using a 45nm standard cell library with energy and delay evaluated using NanoSim and VCS. Experiments show that the introduced techniques enable 40~70% energy savings while only adding less than 6% area overhead when applied to image processing and filtering applications.

Table of Contents

List of Tables ix

List of Figures x

Chapter 1: Introduction 1

1.1 Adaptivity in SRAM 1

1.2 Adaptiviy in Image Processing System 6

1.3 Adaptivity in Digital Filter 12

Chapter 2: Low-power techniques in SRAM 15

2.1 Adaptive Framework in SRAM 15

2. 2. Experiment 20

2.3 Summary 31

Chapter 3: Timing Error Acceptance in Image Processing 32

3.1 Error Control through Knowledge of Operand Statistics 32

3.2 Error Control by Dynamic Reordering of Accumulations 40

3.3 Preventing Error Spread and Amplification 42

3.4 Reducing Residual Image Artifacts through Post-processing 45

3.6 Timing Errors in Big Adder and Small Adders 73

3.7 Summary 76

Chapter 4: Digital filtering 77

4.1 Error Control through Operand Statistics 77

4.2 Error Control through Reordering 87

4.3 Experiment 90

4. 4 Summary 99

Chapter 5: Conclusion 100

Bibliography/References 102

Vita 107

List of Tables

Table 2.1: On-chip voltage scheme area overhead estimation. 24

Table 3.1: Energy Saving and Area of IDCT. 58

Table 3.2: Energy under different combinations. 58

Table 3.3: Area/energy of IDCT with post processing. 71

Table 4.1: Energy and area overhead.. 92

Table 4.2: Energy savings and performance... 92

List of Figures

Figure 1.1: Dependence of threshold voltage standard deviation on transistor area.2

Figure 1.2: The design specification increases due to long tail of Gumbel distribution for high number of cells 4

Figure 2.1: Adjustment of a partition's V_{DD} shifts access times of all cells in a partition to fix violations or save power 16

Figure 2.2: SRAM architecture using row-based multiple voltage control.......... 17

Figure 2.3: PMOS switch network for a partition selects one of available voltages.18

Figure 2.4: SRAM bank architecture 21

Figure 2.5: SRAM voltage delivery architecture. 22

Figure 2.6: Expected power vs. bitcell area Pareto curves for different voltage and switch complexities in active mode. 28

Figure 2.7: Expected leakage vs. bitcell area Pareto curves for different voltage and switch complexities in sleep mode. 29

Figure 2.8: Monte Carlo simulation for SNM in active mode............................. 30

Figure 2.9: Monte Carlo simulation for SNM in sleep mode. 30

Figure 2.10: Monte Carlo simulation for RCM. 30

Figure 2.11:. Monte Carlo simulation for WRM in active mode. 31

Figure 3.1: Frequency distribution of IDCT coefficients for sample image......... 33

Figure 3.2: Partitioning of input matrix. 33

Figure 3.3: Quality-energy tradeoffs in Adder 1 and Adder 2............................. 34

Figure 3.4: Energy vs. quality loss Pareto front - comparison............................. 37

Figure 3.5: DCT coefficient partitioning. 38

Figure 3.6: Reduced width adder. 40

Figure 3.7: Accumulation reordering architecture. 42

Figure 3.8: Rescheduling of algorithm steps. 44

Figure 3.9: Images with/without median filtering. 47

Figure 3.10: Modified median filtering algorithm, window length 3. 48

Figure 3.11: Bitmap for all components. 50

Figure 3.12: Comparison between median filtering and error limiting. 53

Figure 3.13: 2D-IDCT design architecture. 54

Figure 3.14: Q-E tradeoff under different compression ratios. 56

Figure 3.15: Individual PSNR vs. energy profiles. 57

Figure 3.16: Combined PSNR vs. energy profile. 59

Figure 3.17: Q-E curve for various images with/without our techniques. 60

Figure 3.18: Frequency of errors in individual bit positions. 61

Figure 3.19: Image quality under different energy budgets. 62

Figure 3.20: Approximate 2D-DCT design: comparison of this work and CSHM. 63

Figure 3.21: An unfolded IDCT design. 64

Figure 3.22: An IDCT design using sign-magnitude representation. 65

Figure 3.23: An IDCT design using 20 bits vs. 24 bits. 66

Figure 3.24: Q-E tradeoff using various adder architectures. 66

Figure 3.25: Reduced-width CLA adder. 67

Figure 3.26: QE tradeoff for balanced/unbalanced design. 68

Figure 3.27: PSNR curve for filtered and unfiltered image. 71

Figure 3.28: MS-SSIM curve for filtered and unfiltered image. 71

Figure 3.29: Image quality after post-processing. 72

Figure 3.30. Comparison between median filtering and error limiting using quantized data. 73

Figure 3.31: The bitmap for adder 1 and adder 2. 75

Figure 3.32: The four subcases for C_2. 75

Figure 4.1: Digtial filter in direct form II. 78

Figure 4.2: Distribution of input data in an audio application. 79

Figure 4.3: Dynamic-width adder architecture. 81

Figure 4.4: Quality loss in two adders vs. Adder 2 bitwidth. Behavior for several values of timing budget is shown. 82

Figure 4.5: Total quality loss as a function of Adder 2 bitwidth. 82

Figure 4.6: Quality loss vs. adder number. 83

Figure 4.7: Dependence of optimal Adder 2 width on input data variance. 84

Figure 4.8: Quality loss and average adder width.. 84

Figure 4.9: Predicted W_2 and simulated W_2 under different W_1.......................... 87

Figure 4.10: Technique abstraction of arithmetic reordering. 90

Figure 4.11: Single MAC filtering architecture. 90

Figure 4.12: Individual SSNR vs. energy profiles in FIR filters. 93

Figure 4.13: FIR filters using reorder technique.. 94

Figure 4.14: FIR filter for different input data with dynamic bitwidth optimization. 95

Figure 4.15: Individual SSNR vs. energy profiles in IIR filters. 96

Figure 4.16: Image quality under different energy budgets................................. 98

Chapter 1: Introduction

Energy efficiency is one of the paramount concerns in the design of embedded and portable system. Conventional worst-case design strategies are highly inefficient in terms of energy consumption. Hence, we propose adaptivity at both the architecture and circuit levels as a means to remove some of these inefficiencies. Specifically, this dissertation deals with two types of adaptivity: 1) adaptivity to process variations in SRAM design; 2) adaptivity to input data variations while giving up on timing correctness in DSP design.

In the following discussions, we will first review the background knowledge and the previous works of low energy techniques in SRAM and DSP. Details of our low-energy design techniques will be introduced in subsequent chapters.

1.1 ADAPTIVITY IN SRAM

In nanometer technologies, the increase of process variation significantly impacts circuit yield. The impact of variability on the design of large SRAM arrays is especially severe. Some patterns of variability are highly systematic, such as those in photolithography and chemical-mechanical polishing. Among the random variability patterns, the threshold voltage variation due to random dopant placement is paramount. SRAM cells are typically sized to be of minimum area, because of the tight layout requirements for large arrays. Because the variance of threshold voltage variation is inversely proportional to the transistor area, the V_{th} variance of small size transistors is large and is growing as reported in Figure 1.1 [1].

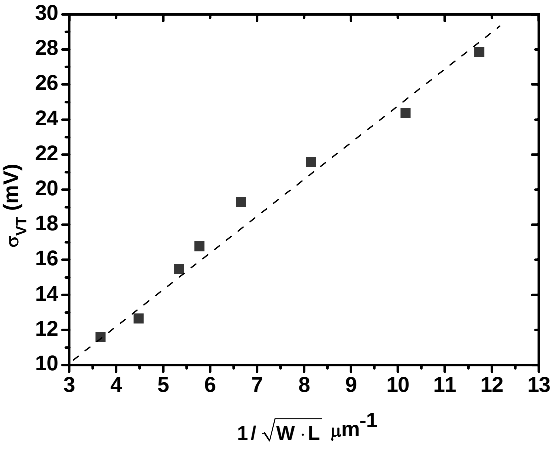


Figure 1.1: Dependence of threshold voltage standard deviation on transistor area.

SRAM cell design is driven by the need to satisfy static noise margin, write margin and read current margin over all cells in the array and these constraints determine both the minimum cell size and supply voltage. Increasing cell area and supply voltage can ensure that the noise margins are met. The requirement to meet noise margin constraints sets the limit on the smallest possible cell size and also on the minimum usable supply voltage V_{DD} for the array, commonly known as V_{min}. Because threshold voltage variation impacting the cells in the SRAM array is independent, the margin specification needs to be met at very high sigma corners, five or six sigma, in order to reach acceptable yield, requiring significant cell upsizing and increased V_{min}. SRAM cell area is an important metric of the success of technology scaling, and variability makes such traditional scaling hard to sustain. Thus, reducing the negative impact of random V_{th} variability on SRAM area is an important goal.

One effective strategy for dealing with variability is post-silicon adaptivity. Adaptive circuit-level solutions, such as adaptive supply voltage and adaptive body bias have been employed in order to increase frequency, reduce standby leakage, and reduce switching power in logic circuits. For example, the approach of [2] is based on applying high V_{DD} and forward body bias for slow dies and low V_{DD} and reverse body bias for high power, thus shrinking the yield violation region. In techniques not directly addressing variability, multiple adaptive strategies for SRAM arrays have been explored to save leakage power or enhance cell stability. In [3], in order to achieve adequate SRAM cell stability, the authors use different V_{DD}s for read and write operation. In this case, assigning V_{DD} in a row-based manner is not suitable because there could be simultaneous read/write operations in the same row. In [4], the author uses dynamic voltage in a row-by-row manner and unaccessed rows are set to low V_{DD} to reduce leakage.

Earlier work has addressed the techniques for mitigating the impact of global die-to-die variation on the operation of SRAM arrays. In [5] adaptive body bias has been employed to increase SRAM yield. The high rate of read/hold failures at low V_{th} corners has been reduced by employing reverse body bias and at high V_{th} corners forward body bias has been used to reduce the write/access failures. In [6] the concept of column-based voltage assignment to reduce voltage overhead was described. However, the work has not developed the theoretical models that would be needed to guide adaptive design at specific levels of variability and for reduced bitcell area.

The key contribution of our work is that we propose a new architecture to reduce the overhead of high-sigma margin design on V_{DD} and cell area by employing an adaptive voltage scheme in a partitioned SRAM array. The key idea is to be able to shift empirical distributions (realizations) of the design margins in a partition to meet the target specification. Because the partition is smaller than the whole array, the tail of the Gumbel distribution is significantly reduced. For the partitions whose worst margin violates the specification, a higher voltage is selected to gain yield, otherwise voltage is reduced for power reduction.

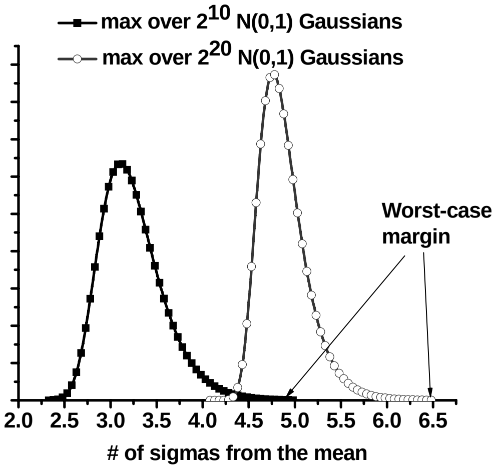


Figure 1.2: The design specification increases due to long tail of Gumbel distribution for high number of cells

The central intuition we introduce is the possibility of smaller bit cell area design due to the use of adaptive techniques which enable reducing the pessimism in selection of design-time optimal solutions. Our basic argument is that accepting a larger single-cell spread, due to sizing-down the cell, and compensating it by post-silicon adjustment of the empirical realization of margins in the partition allows solutions that are area- and energy-superior to the traditional ones. In the sleep mode operation, the voltage of SRAM arrays is often reduced to a voltage level that guarantees that cell content will not be destroyed. Only SNM is relevant in the sleep mode. The key factor that determines the amount of power reduction in sleep mode is how low can the voltage, known as *data retention voltage* (DRV), be. The data retention voltage of an SRAM is determined by the worst-case tail of the DRV distribution of all the cells in the SRAM [7]. Under the normally distributed within-die threshold voltage variation, the 6σ tail of the DRV distribution is almost 3X of the mean value. On the other hand, leakage power has exponential dependence on V_{DD}: a small amount of reduction of V_{DD} can lead to significant reduction of leakage power. Therefore, if we can reduce the mean value of

DRV of a SRAM in sleep mode using tuning, we can achieve significant leakage power saving.

Given the importance of minimum area design, the impact of post-silicon tuning must therefore be rigorously taken into account in order to size the cell optimally. This requires taking into account the underlying distribution of the process parameters in the device optimization to produce power-area Pareto analysis. We develop models to quantitatively predict the statistics of noise margins depending on the partitioning strategy. The ultimate objective of the model is to guide SRAM design for minimum area and minimum equivalent (mean) power superior to traditional design. The model predicts the benefits for the given cost. First, the models we derive include the realistic constraint that only a small finite number of voltage levels are available to be assigned to partitions. Second, the analysis is cognizant of the cost of the tuning architecture due to the need to assign different voltages to partitions depending on the realization. The cost includes the cost of generating additional voltage levels, creating extra routing, and control logic. Our solution for avoiding the prohibitive cost of continuous voltage adjustment is to provide a choice of several discrete but optimized voltage levels chosen at manufacture time. This drastically reduces the cost of adaptivity. While the cost of this adaptivity is quite low, even with only 2-4 voltage levels, as we demonstrate, our architecture allows a significant cell area reduction.

We use the framework of two-stage stochastic optimization for capturing the interaction between the design-time sizing and adaptivity. The analytical models for design margins are based on the extreme-value statistics. They allow to express yield as a function of supply voltage and size. We consider cases of continuously adaptable voltage and discrete adaptable voltage and express the yield as a function of tuning architecture parameters (partition size, range of continuous adaptable voltage or the values of discrete adaptive voltages). In this dissertation, we provide a framework to handle multiple design margins taking into account the mutual correlation. Furthermore, we allow more general class of objective functions which allow minimization of power or a linear combination of mean and standard deviation of adaptive supply voltage. The method is based on application of stochastic dynamic programming [8] and exploits the recursive structure of the objective function.

1.2 ADAPTIVIY IN IMAGE PROCESSING SYSTEM

Nowadays, the gap between the limited battery life and the need to support more complex functionality of embedded systems is growing. Mitigating this gap requires continued advances in low energy design. To solve this problem, we propose to exploit error-tolerance of certain signal processing circuits to reduce their energy consumption. Our strategy focuses on circuit-level Timing ERRor Acceptance (TERRA) as a way to reduce energy. In a conventional design methodology, driven by static timing analysis, timing correctness of all operations is guaranteed by construction. The design methodology guarantees that every circuit path regardless of its likelihood of excitation must meet timing. Traditional design strategies do not consider the possibility of accepting timing errors. When V_{DD} is scaled even slightly, large timing errors occur and rapidly degrade the output signal quality. This rapid quality loss under voltage scaling significantly reduces the potential for energy reduction. In this dissertation, we will show how the above quality-energy tradeoff can be dramatically improved. We achieve this by identifying the sources of early and, from the quality loss point-of-view, worst timing errors and modify the circuit such that the overall tradeoff between quality and energy is improved.

Several efforts in the past have explored the possibility of trading quality in DSP systems for lower energy. In [9, 10], energy is reduced by discarding algorithm steps or iterations that contribute less to the final quality. In [11], adaptive precision of the arithmetic unit output is used to save energy. In [12, 13], energy reduction is enabled by using lower voltage on a main computing block and employing a simpler error-correcting block that runs at a higher voltage and is thus, error-free, to improve the results impacted by timing errors of the main block. In [14], a low-power DCT core is implemented by identifying and skipping the unnecessary computations. In [15], power is reduced by applying aggressive voltage scaling to the memory of a multimedia system, and then filtering out the resulting memory faults. The most similar approach to ours is described in [16, 17, 18]. In this work, combinational logic blocks are restructured to enable utilization of intermediate results, which are arranged such that the more important ones, from the quality point of view, are obtained first.

An important distinction between prior work and our TERRA strategy is that in other work, the results produced by blocks subject to timing errors are not directly accepted. From the point of view of gate-level design, such techniques still guarantee timing correctness of all digital operations. In [12, 13], an estimated value of the result is used in downstream computation in case of timing errors. In [16, 17], computation is terminated early and intermediate results impacted by timing errors are ignored entirely. In contrast, our strategy allows using the erroneous results directly, providing, of course, that the magnitude of error is carefully controlled. As a result, we are able to achieve large energy savings in the low range of quality loss.

The available data from literature suggests that our design is effective. The energy savings achieved are higher than in earlier work: for example, the saving are 55% in [14], 40% in [16], and 62.8% in [18]. Because an exact comparison is difficult for designs in different technologies, we implemented one of the prior designs (the CSHM-based DCT design described in [18]) and compared it with a DCT design based on our techniques. The results show that TERRA techniques can achieve substantially lower energy for an image quality of about 30dB.

We also anticipate that our strategy is extendable to a larger class of algorithms. Our approach does not require changing the algorithm itself, e.g. to allow for early termination. Instead, we directly re-design the implementation to tolerate timing errors. Another difference with [16, 17] is that their approach only allows a discrete set of quality-energy points. By contrast, our technique enables a range of trade-offs along a continuous quality-energy profile.

The proposed TERRA strategy is based on a statistical treatment of errors: while we give up on guaranteeing the worst-case timing, we have to satisfy timing requirements on average to keep global signal quality from severe degradation. Here we considerably extend earlier work by: (a) substantially extending formal analysis of the design choices that need to be made in implementing the timing-error accepting strategy, and (b) presenting novel post-processing techniques to improve the perceptive quality of the image produced by the error-accepting circuits. The new post-filtering techniques are motivated by our search for further ways to reduce the degradation of perceived image quality.

Experiments suggest that two images may have similar values of PSNR but be assessed as being of different quality by a human subject. It turns out that an image with fewer localized errors is more acceptable to human perception: the PSNR metric captures only the overall average signal quality, but does not capture well the local quality. MSB errors tend to cause significant local quality degradation. Thus, we have to ensure that if timing errors occur, they are limited to the LSBs as much as possible. The introduced post-processing techniques aim to do that.

A widely used post-processing technique is filtering, such as a 2-D median filter. In [19], median filtering is used to remove noise. In this dissertation, we implement a simplified median filter that can quickly estimate the median of an array of pixels, such that the computational complexity is reduced and a low-energy design is achieved. Another existing approach for error reduction is to identify the erroneous results and then replace them with an approximated one [12]. We propose an image filter with error limiting that performs a partial substitution on the output pixel instead of replacing all of its bits. This significantly simplifies the error checking and correction logic. In contrast to previous work, our focus is on energy minimization under performance constraints instead of pure performance or throughput optimization [19].

We advance architecture-level techniques that significantly reduce algorithm quality loss under V_{DD} scaling, as compared to direct V_{DD} reduction. This leads to a superior quality-energy tradeoff profile. Fundamentally, this is enabled by (i) reducing the occurrence of early timing errors with large impact on quality, (ii) using control and data flow analysis to disallow errors that are spread and get amplified as they propagate through the algorithm, and (iii) applying post-processing techniques to reduce localized large magnitude errors that significantly degrade local image quality.

To address the first goal, we specifically focus on the behavior of timing errors in addition as a fundamental building block of most signal and image processing algorithms. Simple analysis shows that the magnitude of timing errors depends on the values of operands. A specific important class of operands leading to early and large-magnitude timing errors is the addition of small numbers with opposing signs. We develop two distinct techniques at two levels of granularity - one at the operation and one at the block level - to reduce such errors. Note that depending on knowledge about data statistics, both techniques can be applied at design or at run time. For the design chosen in this dissertation, however, we limit discussions to static operation-level and dynamic block-level optimizations.

Targeting the first two goals, we present four quality-energy (Q-E) optimizations at the operation, block, algorithm and system levels. Techniques are introduced and demonstrated on the designs of an Inverse Discrete Cosine Transform (IDCT) and a Discrete Cosine Transform (DCT) as widely used image and video processing kernels. Specifically, the key contributions for architecture Q-E profile shaping are:

1) Controlling large-magnitude timing errors in operations by exploiting the knowledge of statistics of operands. In many cases, we have knowledge of data distributions that can be exploited at design or at run time. Specifically, in the IDCT/DCT algorithm, high-frequency coefficients tend to have small magnitude values, often of opposite sign. And such components tend to cause early and large quality loss (See Figure 3.3 (c) and Figure 3.19). Our technique is based on the realization that an adder with reduced bitwidth can be used to process such operands. Two objectives are achieved by using such adders: the magnitude of quality loss is reduced and its onset is delayed. In the IDCT/DCT algorithm, the classification can be done at design time, with higher-frequency components being processed in reduced-width adders while the rest of the matrix components are processed on the regular-width adder.

2) Controlling the frequency of error-generating additions by dynamically re-arranging the sequence of operations, e.g. in accumulation. Similar to the previous technique, this strategy aims at reducing the quality loss in addition stemming from processing of small-valued opposite-sign numbers, but at a level higher than that for a single addition. Specifically, it is targeted at reducing the cumulative quality loss

resulting from multiple additions. Such multi-operand addition occurs, for example, in accumulation, which is a key component of many DSP algorithms, and, specifically, of IDCT/DCT.

3) Preventing occurrence of errors which can spread and get amplified throughout the algorithm. An important aspect of a design methodology that allows some timing errors is controlling the impact of these errors on output quality from the perspective of the entire algorithm. Specifically, a result impacted by timing errors early in the algorithm can have a dramatic impact on the overall quality by affecting downstream computations through repeated reuse of incorrect data. Therefore, we can not afford to allow errors in certain critical steps, and we propose a technique to avoid such errors based on rescheduling of the algorithm.

4) Post-processing to mitigate the effects of errors that do occur and to assist the aforementioned Q-E optimizations. In the pixel domain, timing errors appear as discontinuous outliers and MSB errors that significantly degrade *perceived* image quality. Such errors can be effectively detected and mitigated via median filtering or error limiting. A straightforward implementation of median filtering requires several full-bitwidth comparators, resulting in large energy and area overhead. Instead, we propose a low-cost scheme which computes an approximation of the median by using only two MSBs to perform the comparisons which reduces the overhead. Alternatively, in the 2D-IDCT algorithm, the average pixel value can be computed accurately even under scaled V_{DD} just from the DC component. Furthermore, in order for a block of pixels to have MSBs that are different from the average, it has to either have large AC coefficients or an average which is closed to 2^N, N here is any integer greater than 0. Hence, we can compute an error-free average and maximal range for each pixel. For each input block, we check the AC and DC components to determine the maximum possible difference

between pixel and average. If there are pixels for which the actual difference is smaller, we set their MSBs to their average value in order to limit possible errors in pixel MSBs.

1.3 ADAPTIVITY IN DIGITAL FILTER

Digital filter is another important DSP application which is heavily used in multimedia tasks such as speech, image, and video processing. Such tasks are often responsible for much of energy consumption on portable electronic devices. Extending battery time requires continued innovation in low-power methods for such multimedia applications. To enable a low-power digital filter design, we propose techniques based on timing error tolerance to significantly reduce energy consumption in digital filter circuits, which are an important building block of many such applications.

Because of their importance, much work has been done in the area of low-power implementation of digital filtering circuits over the previous decades. In general, finite-impulse response (FIR) filters tend to be more power-consuming than infinite-impulse response (IIR) ones [20]. A very incomplete list of approaches to reduce power consumption in FIR filters at the architectural level includes techniques such as multirate filtering, subfilter approaches and multiplierless architectures [20]. At the circuit level, optimal selections of filter bitwidth and realizations of adders and multipliers to reduce power consumption have been done either in a static [21] or dynamic [22] fashion. Furthermore, optimally choosing filter parameters for given target metrics such as gain, phase linearity, bandwidth, pass-band ripple or stop-band attenuation for low power has also been investigated [23].

It is widely recognized that voltage scaling is one of the most effective ways to reduce power consumption of any digital system. In [24, 11], this is exploited by implementing the filter using fastest possible filter structures and then using generated timing slack to reduce power via voltage scaling. In the traditional paradigm of V_{DD} scaling, scaling is limited by the worst-case delay through any combinational logic. In other words, a conventional methodology guarantees timing correctness of all operations by construction. In this dissertation, we describe techniques that allow pushing V_{DD} scaling beyond this point and achieving further energy savings. In traditional approaches, scaling of V_{DD} beyond the point of worst-case delays immediately leads to large timing errors and rapidly degrades the output signal quality. This rapid quality drop eliminates the possibility of an efficient tradeoff between quality and energy. In this dissertation, we show how to achieve a graceful degradation of filter quality under scaled V_{DD}. We achieve this by identifying the sources of early and worst timing errors and designing filtering architecture to eliminate such errors.

In developing this approach we work in the wider framework of error-tolerant low-power design. In previous publications, techniques for trading quality for energy in digital filtering and digital signal processing (DSP) applications have been studied at varying levels of abstraction. In [10], the authors propose a technique that dynamically minimizes the order of a digital filter to reduce the switched capacitance and hence the total energy. In [25], the authors restructure the filter computations such that voltage scaling affects less important filter taps first.

Finally, in [26], energy is saved by using lower voltage on the main computing block and running a simplified estimating block at higher voltages to correct timing errors in the main block.

The common feature of prior work is that results produced by blocks subject to timing errors are not directly accepted. By contrast, our strategy allows using the erroneous results, provided, of course, that the frequency and magnitude of errors is carefully controlled. In this dissertation, we propose the concept of controlled timing error acceptance for low energy DSP applications, and demonstrate a significantly improved quality-energy tradeoff for a 2D-IDCT block. And then we adapt and generalize this approach to develop an architecture and design strategy for low-power, timing error accepting digital filters with applications in a wide range of DSP systems.

We specifically propose two architecture-level techniques aimed at selectively accepting and controlling timing errors in the datapath of filters: 1) dynamic bitwidth adjustment to control the occurrence of large magnitude errors in small operand additions at run time, and (2) static filter tap reordering to control, at design time, the frequency of small valued operations leading to large errors. Fundamentally, both techniques are based on the observation that the largest and earliest timing errors stem from processing small inputs in the adder of the multiply-accumulate (MAC) units. By controlling the frequency and magnitude of such errors, we can allow for larger V_{DD} scaling while maintaining a high output signal quality.

Chapter 2: Low-power techniques in SRAM

2.1 ADAPTIVE FRAMEWORK IN SRAM

The SRAM yield is set by the need to satisfy three noise margins: read current margin (RCM), write margin (WRM), and static noise margin (SNM). The SNM is defined as the minimum voltage noise required to flip the state of the cell. WRM is required to perform successful write in the period during which the word line is turned on. RCM is needed to ensure that there is enough time to build sufficient bit line voltage difference for the sense amplifier during the cell read. Thus, RCM can be captured as cell read delay time. For RCM, an upper limit will be imposed on the maximal cell read time over all cells. For the SNM and WRM, we have a lower bound on the minimal value of these margins over all cells. By negating SNM and WRM, we can assume without loss of generality that we have an upper bound on the maximal value of noise margins among all cells. The sizing of the SRAM cell transistors must be carefully adjusted in order to meet the conflicting requirements of the three margins. The total array yield is determined by the probability that the worst margin meets the target specification.

The satisfaction of noise margins is driven by the maximum of the cell margins in an array. Stochastically, the maximum over a large number of random variables follows extreme order statistics. For a typical SRAM consisting of a large number of cells, the worst case margins are asymptotically distributed according to a Gumbel distribution, which is characterized by long tails (Figure 1.2 in the introduction section). This requires that each cell's margins must be sufficiently high at high sigma corners in order to achieve the target yield.

Our adaptive framework for SRAM relies on partitioning the SRAM array into a set of *tunable blocks* which can be set to different supply voltages. Due to the smaller number of cells in each block, the realization of the worst margin is significantly smaller (indeed, exponentially smaller [27]). As shown in Figure 2.1, because each partition contains only a sub-set of all cells in the array, there will be many partitions in which the worst realizations among the cells will still have margins which are less than the margin upper bound. Partition-based voltage assignment means that the entire set of realized bitcell margins in a partition is shifted. That ability allows us to accomplish one of two things: (1) for fixed cell area, we are able to reduce supply voltage in partitions whose worst realizations are below the relevant constraint, and thus reduce the average supply voltage; or (2) to reduce cell size which results in the larger spread in V_{th}, but which can now be tolerated because of the presence of tunability.

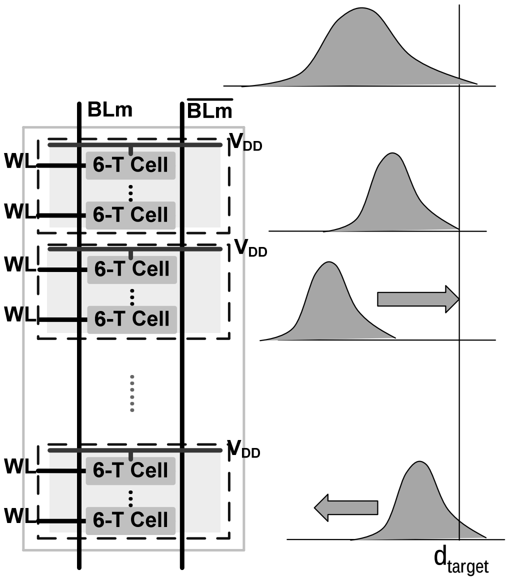


Figure 2.1: Adjustment of a partition's V_{DD} shifts access times of all cells in a partition to fix violations or save power

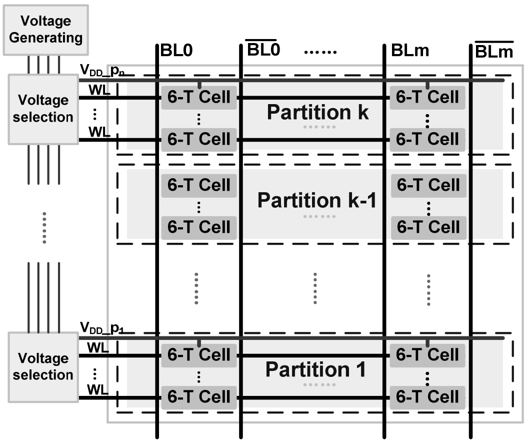


Figure 2.2: SRAM architecture using row-based multiple voltage control.

We now describe the SRAM architecture organization to implement our scheme. The array is partitioned in a row-by-row manner, as shown in Figure 2.2. The partitions are sets of rows that can be set at a distinct voltage. The supply voltage of all the bitcells in the partition and the corresponding wordlines can be set to one of the allowed voltage levels. The voltage levels between partitions can be different. The bitline voltages are fixed. There are two reasons why we choose the row-by-row partitioning strategy for the array: (1) If partitioning is done in a column-by-column way, the voltage for each partition is determined by the slowest cell in the column, but since SRAM is accessed in a row-by-row manner, for most of the read/write operations, it would waste power; (2) we fix the bitline voltages, while if the voltage island is partitioned in a column-by-column manner, the bitcell supply voltage would be different between the wordline voltage and bitline voltage. That would make it hard to achieve high SNM and WRM.

In Figure 2.1, each partition contains only a sub-set of all cells in the array. There will be many partitions in which the worst realizations among the cells will still have margins which are less than the margin upper bound. Partition-based voltage assignment means that the entire set of realized bitcell margins in a partition is shifted. That ability allows us to accomplish one of two things: (1) for fixed cell area, we are able to reduce supply voltage in partitions whose worst realizations are below the relevant constraint, and thus reduce the average supply voltage; or (2) to reduce cell size which results in the larger spread in Vth, but which can now be tolerated because of the presence of tunability.

The voltages are generated by the on-chip voltage regulators. The area and power overhead associated with generation of additional voltage levels is estimated and is reported in Chapter 2.2. A PMOS switch network is used to carry the selected voltage to each partition, as shown in Figure 2.3.

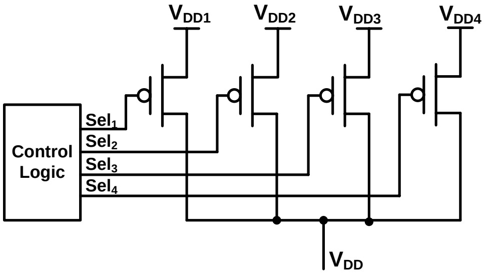


Figure 2.3: PMOS switch network for a partition selects one of available voltages.

After the SRAM is manufactured, the RCM (or read time), SNM and WRM of each partition are measured. Based on the worst-case partition margins, a voltage is selected for that particular partition. The measurement of the RCM and WRM has been addressed in the work in [28], and the measurement of SNM has been addressed in [29].

We now develop an optimization framework for optimally designing, and understanding the impact of, a small number of distinct supply voltage levels that can be adjusted in independently tunable blocks of cells. As discussed in the previous chapter, the goal is to mitigate the impact of selecting parameters based on expected or realized worst-case randomness over a very large block of cells, as is required in order to satisfy the margins described above. We show that allowing independent process parameters across a larger number but smaller blocks allows a more energy- and area-efficient design. Specifically, we want to study the reduction of area and power resulting from allowing different numbers of distinct, and optimized, voltage levels.

To choose the optimal cell size, voltage levels and partition numbers for the partitioned SRAM, we formalize the problem for SRAM operating in active mode as follows:

$$\mathrm{MIN}_{\mathrm{W,\,VDD}} : \quad \mathrm{P}_{\mathrm{mean}}(\mathrm{W},\, \mathrm{V}_{\mathrm{DD}}) \qquad (1)$$

$$\text{s.t.} : \mathrm{Yield} \geq \mathrm{Yield}_{\mathrm{target}}$$

We now describe the formulation, if we allow independent V_{DD} tuning for p tunable blocks of m cells each (so that $p \cdot m = n$). At design time, a single value of supply $V_{nominal}$ is set. Let the t size vector $\text{block\_margin}^{nom}{}_{i}$ denote the worst case margins of a tunable block i in the absence of adaptive voltage. The tunable voltage is used to adjust the untuned margins of block i. The voltage for block i is modified to $V_{nominal}$ $+\Delta V_{adapt,i}(\text{block\_margin}^{nom}{}_{i}\ )$ in the post-silicon tuning phase. In this initial idealized treatment, the exact value of $\Delta V_{adapt,i}$ is different for each block but its distribution can be quantitatively captured based upon our tuning policy which we describe later. The expected yield expression becomes:

$$Yield = Pr(min_{1\leq i\leq p}\ block\_margin_i(W,\ V_{nominal} + \Delta V_{adapt,\,i}) \geq margin_{target})$$

The power minimization problem can be re-formulated as: (P_{mean} is the expected value of single cell mean power under V_{th} variability and below its expected value is taken over variability of ΔV_{adapt}.)

$$MIN_{W,V_{adapt}} \ : \ E[P_{mean}(W, V_{\mathrm{nominal}} + \Delta V_{\mathrm{adapt}})]$$

$$s.t. \ : \ Yield \geq Yield_{\mathrm{target}}$$

In the above, P_{mean} and *Yield* are monotonic increasing functions of tunable voltage. Thus in the adjustable post-silicon tuning, the tuning policy is determined by the worst-case margin vectors over each tuning block, and chosen to meet the margin target vector, $margin_{target}$. If the worst-case margins for a tunable block are above the margin target $margin_{target}$ even for a single component among the t margins, then we increase V_{DD} until all the margin components are safely below $margin_{target}$. This allows us to meet yield constraints, although at the cost of a power penalty. If the worst-case margins are lower in all components with respect to $margin_{target}$, we can reduce V_{DD} as long as worst case margins continue to be lower than the margin targets in order to reduce power. After developing the above model, we need to set up experiments to extract necessary coefficients for the models and evaluate the effectiveness of our proposed framework.

2. 2. EXPERIMENT

The following SRAM array organization, shown in Figure 2.4, is used for experimentation and for the extraction of power models. The array size is 1Mb and is divided into 16 banks. The word width of 32 bits is used, so that each access activates one row in a bank but only 32 out of 256 bitcells of the row are used. Each bitline has 256 bitcells. Sense amplifiers are used to recover the full-swing signals at the bitlines.

For the on-chip voltage distributing network, we note that in our earlier work [30] the models assumed an on-chip low-dropout voltage regulator to generate the voltage values because we focused on the array running only in the active mode, without the need to generate a lower voltage for the standby mode. Since we study both the active and sleep modes, the voltage may need to change by up to 70%, making low-dropout voltage regulator impossible. To resolve this problem and for the purpose of modeling and analysis, we assume that two off-chip voltage supply lines are available and the on-chip voltage scheme generates all the intermediate voltages based on the two off-chip supplies. The on-chip voltage generating scheme utilizes the PMOS supply control network introduced in [31], as shown in Figure 2.5.

The advantages of using PMOS supply control scheme are: it does not require any on-chip passive components, resulting in a smaller area footprint; the output voltage values are digitally controlled, resulting in higher accuracy and can be easily interfaced with other digital logic; it directly uses the off-chip voltages, which is more stable because off-chip voltage regulator uses bigger capacitors and inductors to reduce power-line noise; it is passive and can support both active and sleep modes, while other types of on-chip voltage regulators would waste a lot of power in the sleep mode because it is almost impossible to bring down the voltage of the biasing circuits in these voltage regulator. The downside of the supply control is that it requires one extra external voltage, we assume this is acceptable in the SRAM design. Finally, we also note that we assumed a bank size of 256×256.

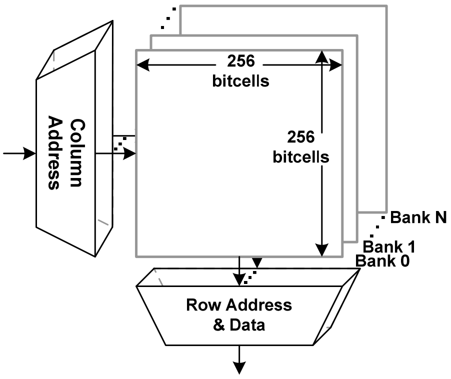


Figure 2.4: SRAM bank architecture

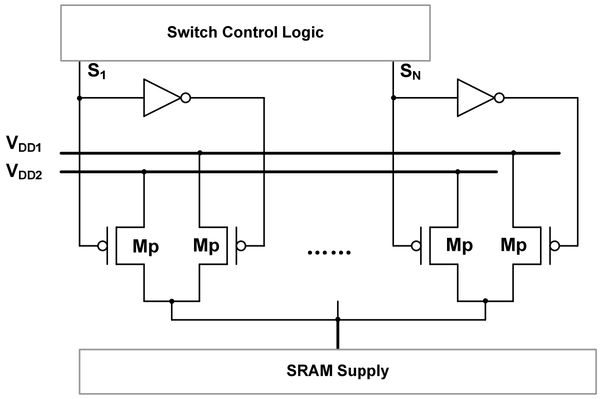


Figure 2.5: SRAM voltage delivery architecture.

Using the SRAM architecture and voltage generating circuits described above, we set up the models for evaluation of the design. We run experiments using the developed model to quantify the ability of the proposed scheme to reduce the impact of randomness on SRAM array. As mentioned before, the SRAM array needs to operate in active mode and sleep mode with V_{DD} lowered to minimize leakage. In these two modes, the SRAM has different design margins to meet. In the active mode, the SRAM bitcells can be read, written or stay in hold. Hence, the SRAM design needs to satisfy the read margin, write margin, and static noise margin. In sleep mode, the supply voltage is lowered to a level above the voltage when the SRAM bitcells may flip, since the bitcells stay idle, only leakage power is consumed. The SRAM has only static noise margin to satisfy. We evaluate separately in these two modes the possible power and area reductions enabled by our tuning strategy.

The evaluation of the cost of tuning circuitry needs to be accounted for and here we provide the details of the additional circuitry to enable tunability. The sizes of the PMOS transistors in the switch network are determined by the maximum active current of the bitcell during the read and write operations plus the leakage current. The maximum current happens during the write operation because the bitcell may flip and draw current from the power line. During a write operation, one word (32 bitcells) of a row is written, while the rest of the bitcells in the same row are being read. The maximum current from the on-chip voltage dividing network is drawn when the wordline driver is activated to drive one row and all the bitcells in that row flip. We characterize this current using HSPICE. In order to estimate the overhead of PMOS switch area, we relate it to the normalized width of the bitcell. The current across the PMOS linearly depends on the voltage and the width. Therefore, the size of the PMOS can be calculated as

$$W_P = I_{max} / (i_o \cdot V_{DD})$$

where i_o is the current conducted by the unit width PMOS transistor under a given V_{DD}. We also need to include the area and power cost of generating distinct voltages. The passsive voltage generating network needs only to supply the current for the bitcell and wordline, which is a relatively small current compared to the current consumed by precharge circuitry and row/column decoder: based on our simulation, when activity factor equals to 0.2, the precharge circuitry consumes about 79.2 μW and the corresponding bitcells consume about 1.1 μW power. Here the precharge circuitry power consists of the power used to charge/discharge the 256 bitlines in one bank. And the bitcells power consists of only the 32 bitcells' dyanmic power during read/write. Although the on-chip voltage distributing network does not provide power to the precharge circuitry, the tuning which happens on the on-chip voltage network can help reducing the precharge power. This is because the partition and tuning techniques reduce the average voltage on the SRAM cell (including the pass transistor) and the bitcell size, it leads to less discharge current and smaller load on the bitlines, and hence reduces the precharge power.

We model the overall area overhead as given by a function of the number of partitions (s), the number of distinct voltage levels (v), and the normalized bitcell width (w):

$$A_O = c_0 \cdot s \cdot v \cdot w + c_1 \cdot (v-1) \cdot w$$

where c_0 is the area of the PMOS switch when $w=1$, c_1 is the area of the voltage dividing network when $w=1$, and $c_0=1.8$ and $c_1=3.5$. We assume that two of the voltages are generated externally. The estimated area overheads are shown in Table 2.1 for different partition complexities.

Table 2.1: On-chip voltage scheme area overhead estimation.

Array size	s	v	w	Area Overhead	Relative overhead(%)
1Mb	1024	4	1	7393.8	0.7%
1Mb	512	4	1	3707.4	0.4%
1Mb	256	4	1	1864.2	0.2%

In the active mode, we model the total power using the following formula:

$$P_{tot} = \alpha \cdot (0.5 \cdot P_{read} + 0.5 \cdot P_{write}) + (1 - \alpha) \cdot P_{leak}$$

where α is the activity factor, 0.2 activity is used in the experiment. P_{read} is the read power, and as mentioned before, we assume that the SRAM array is divided into banks which have a size of 256×256, during read operation, only one row in a certain bank is being accessed, the rest of the banks consume only leakage power. Therefore, the read power consists of the bitline power which is consumed when charging and discharging the bitline, the power consumed when driving the wordline, the cell power of the bank being accessed, together with the leakage power of the rest of the banks:

$$P_{read} = P_{bitline} + P_{wordline} + P_{cell} + P_l$$

Similarly, for the write power P_{write}, it consists of the bitline power, wordline power, cell power, and active leakage power.

$$P_{write} = P_{bitline} + P_{wordline} + P_{cell} + P_l$$

The activity factor of 0.5 means that read and write operations take 50% of the time respectively. P_{leak} here is the leakage power in the active mode. In contrast to the leakage power in the sleep mode, active leakage power is consumed when V_{DD} is around 1V.

We now describe the experiments conducted using the aforementioned architecture and cost models. In this dissertation we use multiple constraints. The multiple constraints used are: the bitcell delay which defines the RCM, SNM, WRM in the active mode, and SNM in the sleep mode.

The closed-form expressions for above design margins, power and leakage models in the active mode and the sleep mode were fitted to SPICE simulations of a cell designed in the 32nm process using the PTM BSIM model. The mean fitting error of the models for all the design margins was below 3%. Besides the fitted model of each single margin, we also model the covariance between any two distinct margins to characterize the correlation between them.

The fitted models for each constraint, power and leakage are as follows:

$$E[P(W,V_{DD})] = e_0 \cdot W + e_1 \cdot W \cdot V_{DD} + \sum_{i=2}^{5} e_i \cdot {V_{DD}}^i$$

$$\frac{1}{E[D(W,V_{DD})]} = f_0 \cdot W + f_1 \cdot W^{-1} + f_2 \cdot V_{DD} + f_3 \cdot V_{DD} \cdot W$$

$$\frac{1}{\sigma_D(W,V_{DD})} = g_0 \cdot W + g_1 \cdot W \cdot V_{DD} + g_2 \cdot W \cdot {V_{DD}}^2 + g_3 \cdot V_{DD} + g_4 \cdot {V_{DD}}^2$$

$$E[SNM(W,VDD)] = k_0 \cdot W^2 + k_1 \cdot V_{DD} \cdot W + k_2 \cdot {V_{DD}}^2 \cdot W + \sum_{i=-2}^{3} k_i \cdot {V_{DD}}^i$$

$$\sigma_{SNM}(W,V_{DD}) = l_0 \cdot W + l_1 \cdot V_{DD} + l_2 \cdot W \cdot V_{DD}$$

$$E[WRM(W,VDD)] = m_0 \cdot W^2 + m_1 \cdot V_{DD} \cdot W + m_2 \cdot {V_{DD}}^2 \cdot W + \sum_{i=-2}^{3} m_i \cdot {V_{DD}}^i$$

$$\sigma_{WRM}(W,V_{DD}) = n_0 \cdot W + n_1 \cdot V_{DD} + n_2 \cdot W \cdot V_{DD}$$

$$E[leak(W,V_{DD})] = EXP(y_0 \cdot W + y_1 \cdot V_{DD} + y_2 \cdot W \cdot V_{DD} + \sum_{i=3}^{4} y_i \cdot {V_{DD}}^i)$$

$$\Sigma_{mutual}(W,V_{DD}) = EXP(q_0 \cdot W + q_1 \cdot W^2 + q_2 \cdot W \cdot V_{DD} + \sum_{i=3}^{5} q_i \cdot {V_{DD}}^i)$$

We use polynomial functions to model the power, delay and noise margins, and the coefficients of the functions are generated by least-square fitting. We first run HSPICE simulation to obtain power and margin data points at discrete sets of W, V_{DD}. Based on such, we run into a recursive procedure to determine the fitted model of power and noise margins in terms of W and V_{DD}. The strategy we use is to start from the model which consists of only 1st order terms: W and V_{DD}, and then we keep adding higher order terms to reduce the fitting error. Also, we remove terms whose coefficients are close to zero while adding new terms. Finally, the procedure terminates when the fitted error is less than 5%.

The delay and leakage power can not be modeled using polynomial function directly. Instead, we model the inverse of delay and the logarithmic of leakage to bound fitted errors under the 5% of the HSPICE simulation results.

The circuit setup for noise margins characterization is as follows: For the SNM, there are two cases: one is the read SNM, and the other is the hold SNM. To measure read SNM, the pass transistor is turned on and we use the method introduced in [29]. For other margins like WRM, and RCM, we measure the write delay and the read delay of the SRAM cells. All the measurements are performed in the form of Monte Carlo simulation. Finally we compute the mean, deviation, and the covariance of all noise margins.

Similarly, we build the fitted models in sleep mode, as follows:

$$E[leak(W,V_{DD})] = EXP(a_0 \cdot W + a_1 \cdot V_{DD} + a_2 \cdot W \cdot V_{DD} + \sum_{i=3}^{4} a_i \cdot {V_{DD}}^{i})$$

$$E[SNM(W,VDD)] = b_0 \cdot W + b_1 \cdot W^2 + b_2 \cdot V_{DD} \cdot W + b_3 \cdot {V_{DD}}^2 \cdot W + b_4 \cdot V_{DD} + b_5 \cdot V_{DD}^2$$

$$\frac{1}{\sigma_{SNM}(W,V_{DD})} = c_0 \cdot W + c_1 \cdot W^{-1} + c_2 \cdot V_{DD}^{-1} + c_3 \cdot V_{DD} + c_4 \cdot V_{DD} \cdot W$$

The *E[SNM]* expression represents the mean value for SNM. Since all mutual covariance expressions $\Sigma_{mutual}(W, V_{DD})$ can use the same fitting formula but different fitting parameters, we use a common mutual covariance expression above to represent the covariance between any two constraints. e.g. read delay and SNM, write margin and SNM, etc.

Because the model already includes the correlation between any two margins, the optimization process automatically handles the case when two margins are conflicting. For example, increasing SNM may reduce WRM, the dynamic programming we use will search a balance point between these two margins according to the covariance matrix and guarantee that both margins are satisfied. The size of the SRAM array is fixed to be 1Mb. The nominal V_{DD} is *1V* for the active mode and *0.4V* for the sleep mode. Adaptive voltage is limited to *20%* of the nominal value. The ratio of transistor widths in the 6T cell is kept constant. The area changes are produced by varying the normalized width (*w*), which uniformly sizes all transistors in the cell. The yield constraint of 90% was used.

Using dynamic programming, we can solve the optimization problem (1). And based on the output of the optimization problem, we obtain the tradeoff curve between power and cell area, shown in Figure 2.6. In Figure 2.6, the experiment result shows that by applying the partition technique, we can achieve power saving and area saving. In the figure, *v* is voltage complexity, which is the number of voltages available for selection, and *s* is the switching complexity, which is number of partitions in the SRAM array.

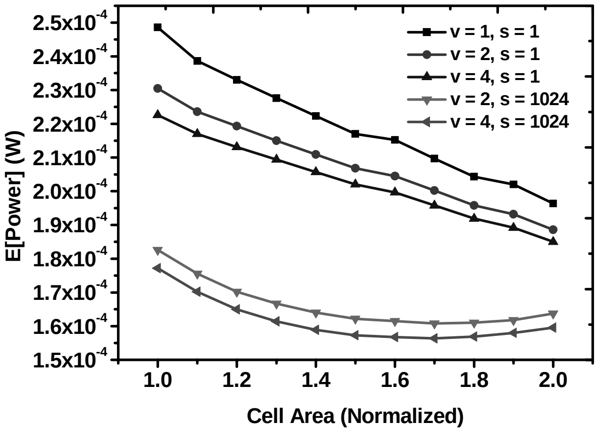


Figure 2.6: Expected power vs. bitcell area Pareto curves for different voltage and switch complexities in active mode.

In Figure 2.6, one important question we sought to answer is the dependence of improvements in area and expected power on the number of voltage levels (v) available. We find that when no spatial partitioning is available (switch complexity s=1), there is little improvement with higher v. However, once spatial partitioning is available (s=1024), area savings are larger for a higher number of voltage levels. Yet, the difference between v=4 and v=2 is not dramatic, which indicates that even a small number of different voltage levels can be effective in tuning circuits.

Similarly, for SRAM operating sleep mode, based on the same principle we use for the active mode optimization, we derive the optimal voltage selection for different partition sizes and cell sizes, and we explore the area reduction under different switching complexity, the experiment result is shown in Figure 2.7.

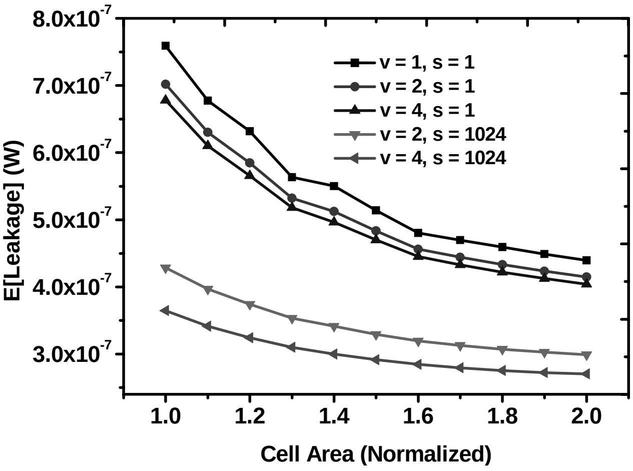


Figure 2.7: Expected leakage vs. bitcell area Pareto curves for different voltage and switch complexities in sleep mode.

Figure 2.7 shows that by using larger partition size, for all values of switching complexity, the average leakage saving of 44% can be achieved.

Overall, we propose an SRAM tuning strategy for reducing intra-array randomness through shifting empirical distributions (realizations) of RCM, WRM, and SNM in a partition to meet the target. We develop quantitative models that for the first time allow rigorous design of adaptive SRAM arrays with large intra-array randomness.

To verify the effectiveness of our partition technique SRAM on saving power while keeping all noise margins, we run Monte Carlo simulations on the SRAM block. In these simulations, without partitioning, the SRAM has to run at the highest voltage to satisfy a high yield target. With partitioning, only a few partitions have to run at the highest voltage to satisfy the margins. The results are shown from Figure 2.8 to Figure 2.11. Each of Figure 2.8~2.11 shows three cases: 1) No partition with nominal V_{DD}; 2) No partition with the highest V_{DD}; 3) 1024 partitions with four different V_{DDS}.

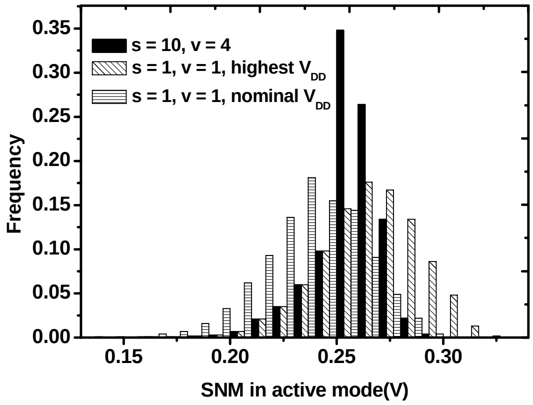


Figure 2.8: Monte Carlo simulation for SNM in active mode.

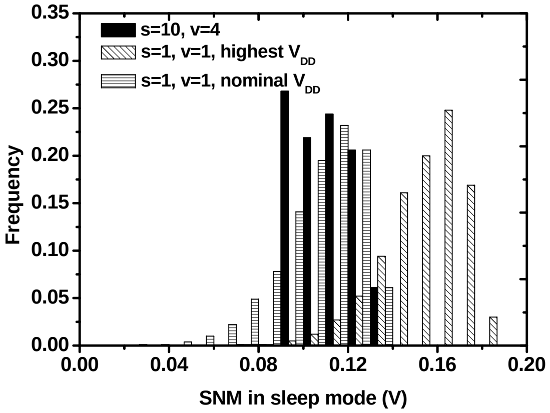


Figure 2.9: Monte Carlo simulation for SNM in sleep mode.

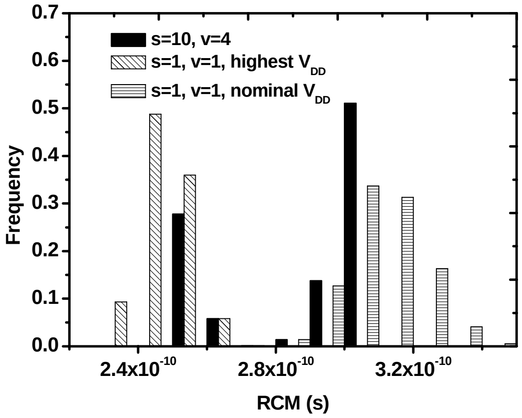


Figure 2.10: Monte Carlo simulation for RCM.

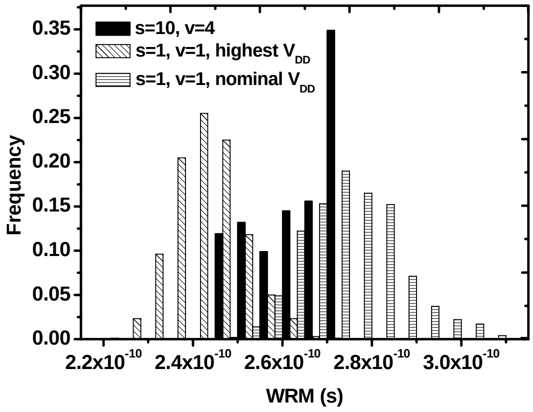


Figure 2.11:. Monte Carlo simulation for WRM in active mode.

2.3 SUMMARY

In this chapter, we propose an SRAM tuning strategy for reducing intra-array randomness through shifting empirical distributions (realizations) of RCM, WRM, and SNM in a partition to meet the target. We develop quantitative models that for the first time allow rigorous design of adaptive SRAM arrays with large intra-array randomness.

Chapter 3: Timing Error Acceptance in Image Processing

The 2D-IDCT and 2D-DCT computations can be represented by $I = C^T \cdot A \cdot C$ and $I = C \cdot A \cdot C^T$, respectively, where C is the orthogonal type-II DCT matrix and A is the spectrum coefficient matrix. It is customary to implement the 2D-IDCT/DCT as a sequence of two 1D-IDCT/DCTs. For each 1D-IDCT/DCT, the core algorithm is a matrix-vector dot product. For IDCT, the transformation is:

$$T(k) = \frac{c(k)}{2} \cdot \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2k+1)n}{2N}\pi\right]$$

$$N = 8, c(0) = \frac{1}{\sqrt{2}}, c(n) = 1, 0 \le k \le N-1$$

where *x(n)* is the data being processed. The DCT is very similar, except that the coefficient matrix is transposed. The following discussions will focus on a 2D-IDCT. Application to a corresponding 2D-DCT will be discussed later.

3.1 ERROR CONTROL THROUGH KNOWLEDGE OF OPERAND STATISTICS

When V_{DD} is scaled down, large magnitude timing errors are very likely to happen in additions of small numbers with opposing sign. Such additions lead to long carry chains and are the timing-critical paths in the adder. The worst case for carry propagation occurs in the addition of -1 and 1. In 2's complement representation, this operation triggers the longest possible carry chain and, thus, experiences timing errors first. Crucially, when a timing error occurs, the apparent result will also have a very large possible numerical error due to carry propagation into the MSBs leading to a large magnitude mismatch compared to the error-free result. For example, in an 8-bit computation, the error magnitude can be up to 64. This analysis and this problem is, of course, specific to the 2's complement representation of signed numbers. However, our techniques can also be used in sign-magnitude representation. As will be detailed later, in sign-magnitude arithmetic, subtractions or opposing-sign additions are internally computed using 1's or 2's complement logic. This results in similar timing error behavior and our techniques remain effective.

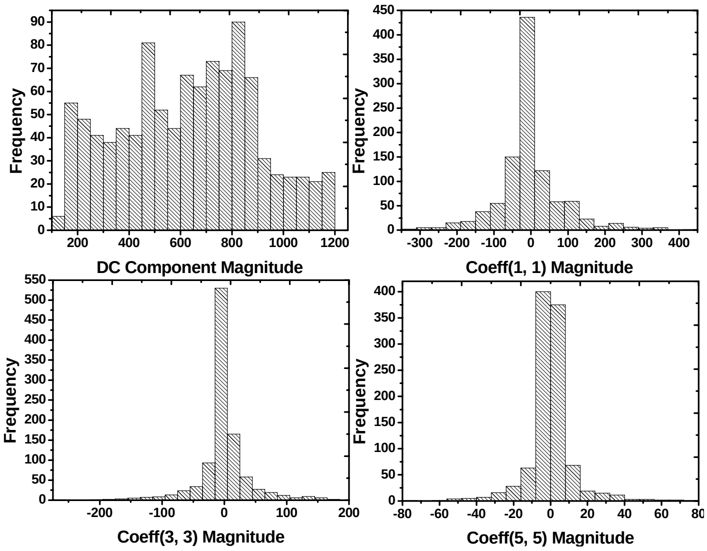


Figure 3.1: Frequency distribution of IDCT coefficients for sample image.

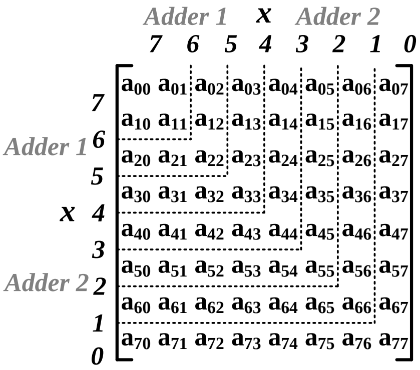
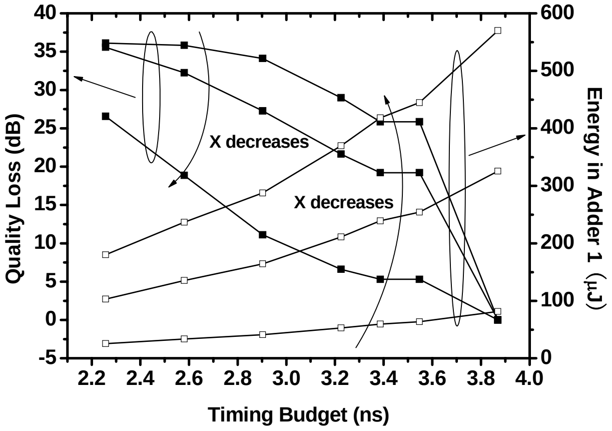
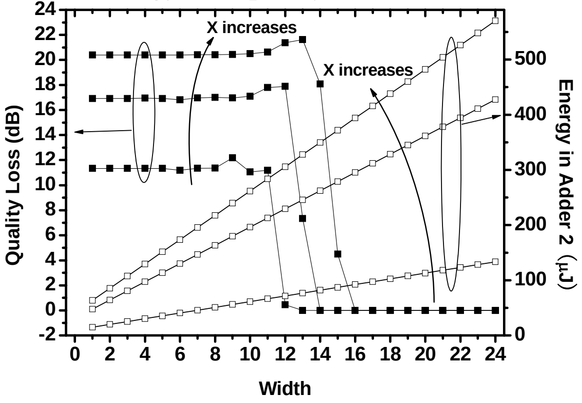

Adder 1
x
Adder 2
7 6 5 4 3 2 1 0
7
Adder 1 6
5
x 4
3
Adder 2 2
1
0
a00 a01 a02 a03 a04 a05 a06 a07
a10 a11 a12 a13 a14 a15 a16 a17
a20 a21 a22 a23 a24 a25 a26 a27
a30 a31 a32 a33 a34 a35 a36 a37
a40 a41 a42 a43 a44 a45 a46 a47
a50 a51 a52 a53 a54 a55 a56 a57
a60 a61 a62 a63 a64 a65 a66 a67
a70 a71 a72 a73 a74 a75 a76 a77

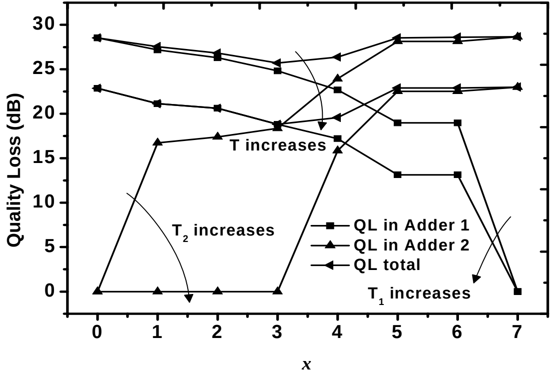
Figure 3.2: Partitioning of input matrix.



(a) Energy and quality loss in Adder 1.



(b) Energy and quality loss in Adder 2.



(c) Quality loss vs. component classification.

Figure 3.3: Quality-energy tradeoffs in Adder 1 and Adder 2.

In the 2D-IDCT algorithm, the additions that involve small-valued, opposite-sign operands occur in the processing of high-frequency components. This is because the first 20 low-frequency components contain about 85% or more of the image energy [18]. Hence, the magnitude of high-frequency components tends to be small, and coefficients follow a Laplace distribution with high probability densities concentrated in a narrow range [32], as shown in Figure 3.1. Furthermore, the Laplace distributions are zero-centered, which implies that high frequency components also tend to have opposing signs. As such, a significant amount of quality loss at scaled V_{DD} can be attributed to additions involving such components. The first specific technique we employ is based on the realization that an adder with a bitwidth smaller than required by other considerations can be used to process such operands. Two objectives are achieved by using such adders: the magnitude of quality loss is reduced and its onset is delayed. Large-valued operands, of course, require a regular-width adder. Note that in an actual implementation it is possible to utilize a single adder with variable bitwidth.

In the IDCT algorithm, the classification of matrix elements can be done at design time. This raises the question of (a) how to best perform this classification; and (b) how to identify the optimal bitwidth of the reduced-width adder. In the following, we develop a model to enable such a design optimization. We define Adder 1 as the regular-width adder and Adder 2 as the reduced-width adder. In classifying the components, we seek to find the boundary, within the data matrix, between the upper-left low-frequency components and the lower-right high-frequency components. We therefore define the following parameters of our model:

x: Boundary between high-/low-frequency coefficients (Figure 3.2).

D_1: Worst-case delay of Adder 1.

D_2: Worst-case delay of Adder 2.

T_1: Timing budget of Adder 1.

T_2: Timing budget of Adder 2.

We assume throughout this discussion that $T_2=D_2$, i.e. that no timing errors are allowed to occur in Adder 2. Furthermore, we assume that $T_1=T_2$, which implies that both adders are affected by V_{DD} scaling in an identical manner.

Based on this notation, we can study the Q-E characteristics of the two adders under scaled V_{DD}. By exploring adder characteristics, we are able to identify the optimal partitioning strategy from the point of view of achieving a globally optimal Q-E result. For simplicity, we substitute in this analysis the equivalent notion of timing budget for the value of V_{DD}.

We first study the Q-E relation for the regular width adder, shown in Figure 3.3 (a). The right axis shows the energy value at different timing budgets T_1. As expected, allotting a smaller timing budget, which entails an equivalent lowering of V_{DD}, results in a reduction of energy. Increasing the number of matrix components processed in the reduced-width adder, i.e. increasing x, results in fewer additions performed by Adder 1, and thus a lower energy at the same timing budget. The quality loss (shown on the left axis) is initially low when the allotted timing budget is high and few computations experience error. As T_1 is reduced, however, we begin to observe that the quality loss is smaller for larger x. This corresponds to the scenario in which fewer operations are performed by Adder 1, and thus there is less opportunity for timing errors to occur.

The Q-E behavior of the reduced-width adder is shown in Figure 3.3 (b). We are specifically interested in finding the Q-E behavior as a function of the bitwidth. Note that because no *timing* errors are allowed in Adder 2, an exploration with respect to timing budget, as shown for Adder 1 above, would have no purpose. We see that for large bitwidths of Adder 2, there is no quality loss. A significant reduction in quality occurs with the onset of overflow errors when the magnitude of data being processed is larger than the available adder width.

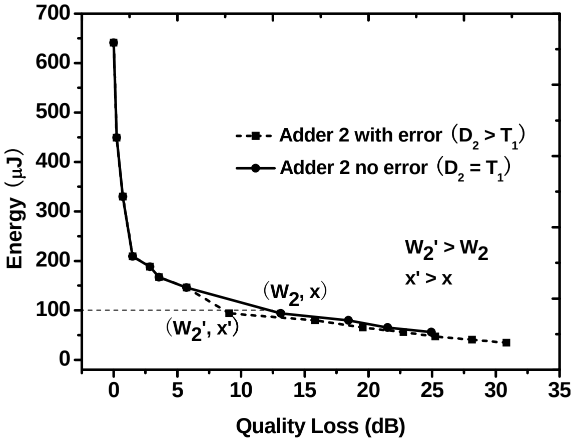
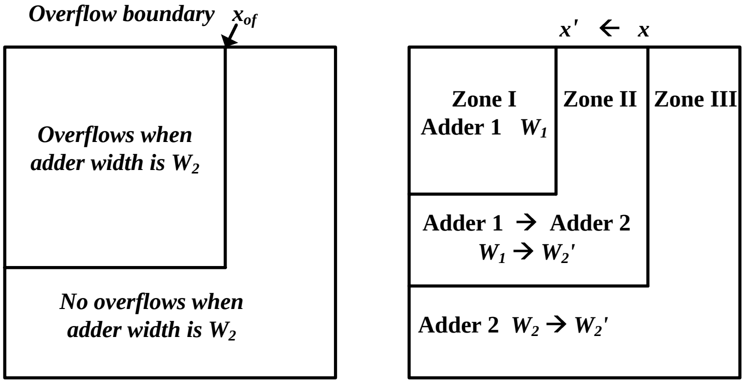


Figure 3.4: Energy vs. quality loss Pareto front - comparison.

The analysis of the system Q-E behavior combines the behavior of Adder 1 and Adder 2. This enables exploration of the x, D_2, W_2, and T_1 design space in order to find an optimal Q-E solution. The primary trade-off involves the choice of x. From Figure 3.3 (c), we can see that the total quality loss reaches a minimum when x is around 4. For larger values, the quality loss due to Adder 2 becomes excessive. For smaller values, the quality loss is dominated by errors from Adder 1. However, the optimal choice of x also depends on both the total timing budget available as well as the bit-width of Adder 2. The set of optimal design decisions is best represented as a Pareto curve in the energy-quality space as shown in Figure 3.4. The figure shows the Pareto points, i.e. $min(Q \mid E)$, that are generated by different choices of x and W_2 at different T_1.

To understand the behavior of Pareto points in Figure 3.4 and trace the dependence of the optimal x on T_1 and W_2, we first study the simple case when $D_2 \leq T_1$. Then, we relax the constraint to allow $D_2 > T_1$, and we adjust x and W_2 under a fixed T_1 to determine the new optimal set of x and W_2 under the relaxed constraint.

Under the constraint that $D_2 \leq T_1$, we can observe: 1) the optimal x is set by the overflow boundary (x_{of}); and 2) the optimal Adder 2 width is the maximum Adder 2 width ($W_{2\max}$), which is set by T_1. The x_{of} here is defined as the maximum possible x for a given W_2 without having overflows in Adder 2, as shown in Figure 3.5 (a).


Overflow boundary x_{of}
Overflows when adder width is W_2
No overflows when adder width is W_2
$x' \leftarrow x$
Zone I
Adder 1 W_1
Zone II
Zone III
Adder 1 → Adder 2
$W_1 \rightarrow W_2'$
Adder 2 $W_2 \rightarrow W_2'$

(a) D2 ≤ T1

(b) D2 > T1

Figure 3.5: DCT coefficient partitioning.

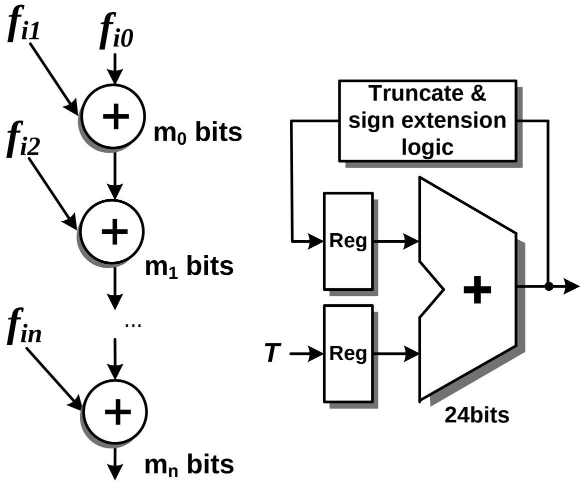
For a given timing budget T_1 and $D_2 \leq T_1$, there must not be any timing errors and we can define a maximum Adder 2 width as W_{2max}. Since the onset of overflow immediately leads to large errors (Figure 3.3 (b)), W_{2max} also sets a maximum x of x_{of} at the boundary at which overflows appear. At the same time, we always aim to send as much data as possible to the error-free reduced-width adder (Adder 2), so as to reduce timing errors in the full-width adder (Adder 1). Hence, we choose x to be at its maximum limit (x_{of}) to allow as much data as possible being processed in Adder 2.

To further explore the design space beyond the point for which no timing errors are allowed in Adder 2, we state a theorem, and present its complete proof in Section 2.5:

Theorem1: *In the absence of overflows, the output timing error in a wide adder is greater than or equal to the error in a smaller-width adder when both adders process the same operands.*

According to this theorem, sending more data to Adder 2, i.e. increasing x, will make it possible to further reduce the quality loss, even in the presence of timing errors in both adders. However, in order to avoid exceeding the overflow boundary with its large quality loss, we also have to increase W_2 and hence x_{of}, relaxing the timing constraint for Adder 2 to $D_2 > T_1$. As shown in Figure 3.5 (b), data in Zone I and Zone II is originally processed by Adder 1 using width W_1. Data in Zone III is processed by Adder 2 using width W_2. After increasing x, data in Zone II is processed by Adder 2 using width W_2', s.t. $W_2 < W_2' < W_1$. The quality loss in Zone II is reduced while the quality loss in Zone III increases according to Theorem 1. Since increasing x, i.e. sending more data to Adder 2, can reduce timing errors in Adder 1, but increasing W_2 leads to more timing errors in Adder 2, there exists an x (and W_2) with maximum quality loss reduction. These points correspond to the Pareto front of the dashed line in Figure 3.4.

In the implementation, the reduced-width addition is actually realized using the truncated result of a regular-width adder sharing the same core logic. The combined adder architecture is shown in Figure 3.6. The indexes of the frequency coefficients are used by the control logic to determine whether to feed them into a full-width or reduced-width addition. The control logic compares the index of the matrix component currently being processed with the predetermined classification constant x. The output of this comparison is used to activate a truncation logic. The truncation logic takes a reduced number of LSBs from the full-width adder output according to the pre-designed Adder 2 width, and sign extends them back to the full width and feeds the result back into the destination accumulator.



(a) Technique abstraction (b) Implementation

Figure 3.6: Reduced width adder.

3.2 ERROR CONTROL BY DYNAMIC REORDERING OF ACCUMULATIONS

The technique introduced in Section 3.1 is able to delay the onset of large-magnitude errors in individual two-operand additions. The second technique presented in this section is based on reduction of the cumulative quality loss resulting from multiple additions, such as accumulations, which are a key component of many DSP algorithms, and, specifically, of IDCT. The key observation is that if positive and negative operands are accumulated separately, and added only in the last step, the number of error-producing operations is reduced to one last addition that involves operands with opposite sign. At the same time, the operands involved in this last addition are guaranteed to be larger in absolute value than any individual opposite-sign operands involved in the original sequence of additions. This guarantees that the reordered accumulation will result in a smaller quality loss under scaled timing.

Let us illustrate how the order of operations in accumulation affects the timing errors occurring at a given timing budget. The difference between optimized and unoptimized sequences is significant. As an example, consider four numbers (-1, 1, -1, 1) being accumulated. There are three possible sequences of accumulation:

Case 1: 11111111+00000001+11111111+00000001

Case 2: 11111111+11111111+00000001+00000001

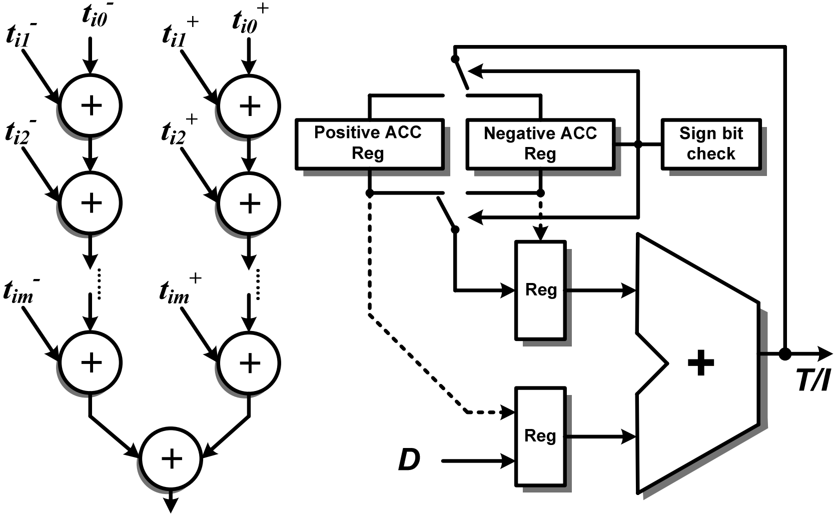
Case 3: (11111111+11111111)+(00000001+00000001)

For Case 1, the 1st and the 3rd additions have large delay, each with a carry chain length of 8. For Case 2, the 3rd addition has large delay with a carry chain of 8. For Case 3, only the addition outside the brackets has large delay with a carry length of 7. The total timing budget in Case 3 is roughly half of that of Case 1. Thus, we observe that the order of accumulation can significantly affect the frequency of worst-case delay as well as the length of the longest carry chain.

Using the observation that additions of small numbers with opposite sign tend to cause large-magnitude errors, we now show how the sequence of additions can be changed to reduce overall error. As described above, we propose a strategy in which we first group operands with the same sign. Then, the operands in each group are accumulated and finally the results of two group-accumulations are added. This is akin to the strategy that Case 3 illustrates. Because the best grouping of operands cannot be known at design time, this technique is dynamic and is based on execution-time observation of operand values.

The proposed implementation uses the sign bits in the MSB to separate the positive and negative operands when loading data. The implementation is shown in Figure 3.7. The control logic checks the sign bits and accumulates positive and negative numbers in separate accumulation registers. Then, in a final step, the results are added together. This final addition can in turn be protected against timing errors using either one of the techniques presented in Section 3.1 or 3.3.

Compared to the original implementation, the reordered accumulation carries extra overhead for the reordering logic and duplicate accumulation registers. Nevertheless, simulation results show (Section 3.5) that the technique can significantly improve the quality-energy profile under scaled timing.



(a) Technique abstraction (b) Implementation

Figure 3.7: Accumulation reordering architecture.

3.3 PREVENTING ERROR SPREAD AND AMPLIFICATION

In previous sections, we presented techniques for targeting individual error sources at the operation and block level. With knowledge of the application, we now further focus on control of sources of errors that have the potential to be spread and amplified at the algorithm level. More specifically, we propose a technique using algorithm-level retiming to explicitly prevent errors in critical steps that may have a large impact on downstream results and hence overall quality.

For the 2D-IDCT algorithm, analysis of control and data flow is relatively simple because it consists of two nearly-identical steps:

$$T = C^T \cdot A$$

$$I = T \cdot C$$

We address the problem of a timing error in Step 1. Such an error can generate multiple output errors in *I* because each element of *T* is used in multiple computations of Step 2. We can model this behavior by introducing an error matrix *E*, which is added to *T* such that the two algorithm steps become:

$$T' = T + E$$

$$I = T \cdot C + E'$$

Here, $E' = E \cdot C$ is the final error. Although *E* may have only one non-zero entry, the matrix product results in up to *size(A)* errors vertically or horizontally in E'. As a result, the noise in the decoded image of an unmodified IDCT has a stripe pattern (see Figure 3.19 in Section 3.5).

Thus, to avoid such wide-spread quality loss, we need to ensure that no errors occur in Step 1. We assume an architecture in which supply voltage can only be scaled uniformly. If timing budgets are allocated to steps based on worst-case analysis, any reduction in V_{DD} would lead to a reduced timing slack in Step 1 and hence un-allowable levels of errors being generated there. We therefore propose a strategy to allocate extra timing margins to critical steps, such as Step 1. Importantly, given overall latency constraints for the design, as is the case for many real-time image or video coding applications, end-to-end algorithm timing must remain constant and performance must not be degraded. Thus, an important element of protecting the early algorithm steps is a re-allocation strategy that shifts timing budgets between steps. Maintaining a constant total time, we show how to borrow computing time from non-critical algorithm steps in order to increase timing margins in critical ones, all while reducing overall quality loss.

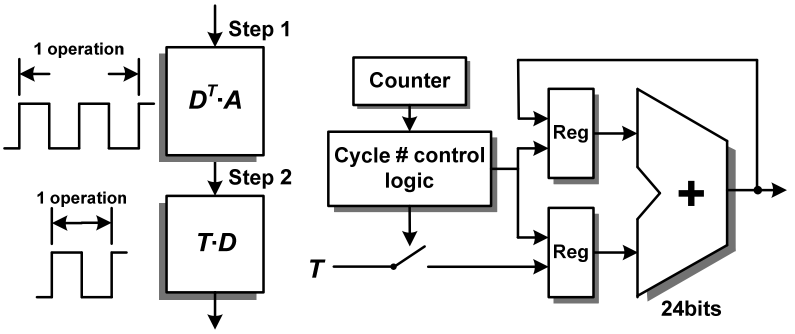
To implement such a strategy, we make the timing budget in each step adjustable. The original minimum error-free timing budget for each step is:

$$T_{step1} = N_1 \times T_{clk}$$
$$T_{step2} = N_2 \times T_{clk}$$

Where T_{clk} is the clock period, and N_1 and N_2 are the number of cycles in each step. In the original 2D-IDCT implementation, steps are identical and $N_1=N_2=N$. To adjust the budget, we need to divide it into multiple parts. A division factor M (M is greater than 1, and it is an integer) is used to make $T_{step1}=NM\times T_{clk}/M$, and $T_{step2}=N\times T_{clk}/M$. V_{DD} is then scaled down, increasing the propagation delays. Consequently, T_{clk} is scaled to T_{clk}' such that $2N\times T_{clk}$ is equal to $NM\times T_{clk}'/M+N\times T_{clk}'/M$, i.e. $T_{clk}'=2T_{clk}/(1+1/M)$. Hence, the new clock frequency is:

$$f_{clk}' = \frac{T_{clk}'}{M} = \frac{2}{((M+1)T_{clk})}$$

Since the total budget is fixed, we disproportionally shift timing budgets under scaled V_{DD} from Step 2 to Step 1. Note, however, that the factor M cannot become too large. Otherwise, the clock frequency would be too high and timing errors would not remain restricted to the adder in Step 2.



(a) Technique abstraction (b) Implementation

Figure 3.8: Rescheduling of algorithm steps.

The implementation includes logic to allocate different timing budgets to each step (Figure 3.8). We empirically choose M to be 2 and increase clock frequency accordingly. The control logic includes a 1-bit counter to keep track of the cycle counts for each step. In Step 1, each operation is assigned 2 cycles, while each operation in Step 2 is assigned 1 cycle.

3.4 REDUCING RESIDUAL IMAGE ARTIFACTS THROUGH POST-PROCESSING

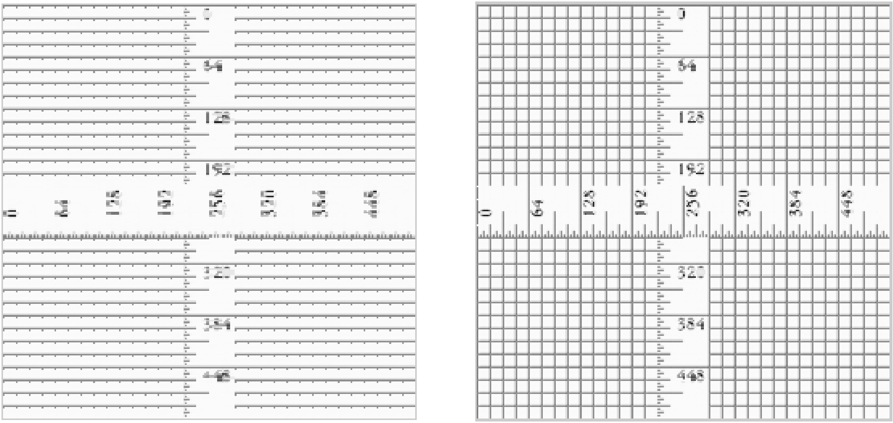
Techniques discussed so far have dealt with preventing or minimizing errors in the output image. While the described techniques significantly reduce energy at an acceptable PSNR, they result in some undesired localized visual artifacts. The reason is that good PSNR alone is not a guarantee of acceptable visual quality of the image.

In the following, we develop energy-efficient means of reducing such image artifacts for the 2D-IDCT design. Artifacts can be divided into two categories: salt-and-pepper noise and stripe artifacts. Salt-and-pepper noise is a pattern of randomly occurring white and black pixels. In our 2D-IDCT system, this type of artifact is caused by timing errors in step 2 (Figure 3.8 (a)). By contrast, stripe artifacts are error patterns appearing as black and white lines of pixels, and are produced when timing errors happen in step 1 of the IDCT: errors in step 1 are amplified through the matrix multiplication in step 2, resulting in a stripe shape. Depending on the multiplication order, the stripe can be vertical or horizontal. In our algorithm, we perform an operation $T \cdot D$, where errors in the intermediate matrix T spread across different rows in the final output, resulting in a vertical stripe. If the order of two steps is reversed, artifact stripes would be horizontal. This property will affect the implementation of post-processing techniques.

Importantly, because the logic for post-processing is quite simple we are able to guarantee that its delay is less than the adder delay. In this way, we are certain that post-processing logic is free of timing-induced errors.

To reduce the artifacts, we propose two separate filtering techniques. These two techniques can be implemented individually, or combined together. In this dissertation, we demonstrate how to implement them separately. The first technique is median filtering. The algorithm uses a sliding window to replace each entry with the median of its neighboring entries. While preserving edges, a median filter is effective at removing localized high-frequency image artifacts, such as the aforementioned salt-and-pepper and stripe distortions, when the noise level is low [33]. Compared to other filters, it is also less complex and only requires comparisons. Therefore, the hardware implementation of a median filter can be made simpler and more energy-efficient.

In our implementation, median filtering is performed on the converted output stream, and is applied to the entire image. In the design without median filtering, the output image is stored in memory and then sent out after each 8×8 block IDCT computation is done. With median filtering, each pixel is filtered when being sent out. As discussed before, stripe-shaped artifacts manifest themselves as single vertical lines. We therefore perform horizontal median filtering, which limits and localizes artifacts in the filter window. To minimize hardware overhead and maximize energy savings, we use the simplest possible 1-D median filter with window length 3. The filter checks a current pixel and all other pixels within the window to determine which one should be outputted. Hence the buffer needed for a length-3 median filter window is of size 2. However, in our experiments, the implementation of such a filter still results in up to 60% area overhead. To further reduce complexity, we can develop an approximate median filter. The most visible artifacts are usually due to errors in MSBs of pixels. Hence, we can apply median filtering by comparing pixel MSBs only. Experiments show that the two most significant bits are sufficient to generate an output with only 0.3dB PSNR degradation compared to the case when all bits are used for median filtering. However, the area overhead is reduced to 6%.



(a) With conventional median filtering: PSNR=17.4dB

(b) With modified median filtering: PSNR=29.7dB

Figure 3.9: Images with/without median filtering.

A drawback of a conventional median filter is that it will produce a modified output even in the absence of timing errors (Figure 3.9). To reduce this effect, we modify the median filtering algorithm as follows: if the MSB of the median is the same as that of the unfiltered pixel, the filter will output the original pixel instead of the median. As such, if there is no or only a small timing error, the unfiltered output will be passed through. This modification is based on the observation that median filtering is effective at removing large single pixel outliers. Hence, if the MSB of the median is different from that of the unfiltered pixel, an outlier is likely detected and the filtered pixel is used to remove it. Otherwise, it means that there is no large outlier, and the unfiltered pixel is used. In this way, we can expect that with very high probability, both the LSB errors introduced by median filtering as well as the outliers can be reduced. Such modification can significantly improve the image quality at all energy levels. For error-free images at high energy levels, the drop in PSNR due to median filtering is reduced from 14dB to 5dB. The modified technique also significantly reduces the distortions caused by conventional median filtering. In the absence of timing errors, as seen in Figure 3.9, the conventional median filter leads to the loss of vertical lines. By contrast, our modified median filter preserves them.

```
$m1=D_1[7:6]$, $m2=D_2[7:6]$, $m3=D_3[7:6]$.
if $m1 \geq m2$ & $m1 \geq m3$ & $m2 \geq m3$
        out = D_2;
else if $m1 \geq m2$ & $m1 \geq m3$ & $m3 \geq m3$
    if $D_3[7] = D_2[7]$
        out = $D_2$;
    else
        out = $D_3$;
else if $m2 \geq m1$ & $m2 \geq m3$ & $m1 \geq m3$
    if $D_1[7] = D_2[7]$
        out = $D_2$;
    else
        out = $D_1$;
else if $m2 \geq m1$ & $m2 \geq m3$ & $m3 \geq m1$
    if $D_3[7] = D_2[7]$
        out = $D_2$;
    else
        out = $D_3$;
else if $m3 \geq m1$ & $m3 \geq m2$ & $m1 \geq m2$
    if $D_1[7] = D_2[7]$
        out = $D_2$;
    else
        out = $D_1$;
else if $m3 \geq m2$ & $m3 \geq m1$ & $m2 \geq m1$
        out = $D_2$;
end
```

Figure 3.10: Modified median filtering algorithm, window length 3.

Another drawback in conventional median filtering is the boundary issue. Pixels at the boundary of each 8×8 block can not be filtered because the data in the filter window is insufficient. To solve this problem, we use the following strategy to generate a window for the boundary pixels. Let us assume that the pixel row being filtered is D_0, D_1, D_2, D_3...., where D_0 sits at the left block boundary. We use the same window (D_0, D_1, D_2) for computing both D_0 and D_1. The problem with this windowing strategy is that for both D_0 and D_1, the filtered output is the same. This can cause visible vertical stripe patterns at the 8×8 image block boundaries. However, such patterns are automatically mitigated by the previous technique, in which the filter, in most cases, will output the original pixel instead of the filtered one.

The complete modified median filtering algorithm is shown in Figure 3.10. In this algorithm, we assume that the input is (D_1, D_2, D_3) and the output is *out* (without filtering, the output *out* would be D_2).

Our second proposed post-processing technique uses error limiting. It relies on the observation that typical images have low local spatial variations. Hence, pixels within each block are likely to have the same MSBs. In the frequency domain, this manifests itself as a large DC and small AC components. Based on this observation, each pixel in a 8×8 block can be represented as the sum of a baseline and a deviation. The baseline value is derived directly from the DC component. It is the same for all pixels in a block, and is the average over all 8×8 pixel values. The deviation is obtained from the AC components, and it differs from pixel to pixel.

In typical smooth image regions, deviations will be small and pixel values are likely to have similar values, i.e. the same MSBs as the baseline average. Under scaled V_{DD}, the baseline can be easily obtained without errors. In a 8×8 2D-IDCT, it is simply *1/8* of the DC value, i.e. it can be computed by shifting the DC component by 3 bits. At the same time, pixels tend to have timing errors in their MSBs first. Therefore, if there are only small local deviations, we can substitute the pixel MSBs (which may be affected by timing errors) with the error-free baseline MSBs, limiting the impact of any timing errors. Because error checking incurs area and energy overhead, we want to perform such substitution blindly. To do so and not introduce additional errors, we have to ensure that baseline MSBs are a correct predictor of error-free pixel MSBs, i.e. that *Pixel[N:i]* = *Baseline[N:i]*. The question therefore becomes 1) when this equality holds, and 2) if it holds, for what values of *i*.

We address these questions by looking at a pixel representation as follows:

$$\begin{aligned} Pixel &= Baseline + \Delta \\ &= Baseline_{MSBs} + Baseline_{LSBs} + \Delta \end{aligned}$$

where $Baseline_{MSBs}$ and $Baseline_{LSBs}$ represent zero-padded splits of *Baseline* at the *i*th bit position, and Δ represents the deviation, see Figure 3.11. We call $Baseline_{LSBs} + \Delta$ the residue term.

To guarantee *Pixel[N:i]* = *Baseline[N:i]* for a given *i*, we need to ensure that the MSBs (bits *N*:*i*) of the residue term are zero, i.e. that $0 \le Baseline_{LSBs} + \Delta \le 2^i$. We can rewrite this inequality as:

$$-\Delta \le Baseline_{LSBs} \le 2^i - \Delta \qquad (3.1)$$

To find the *i* that satisfies inequality (3.1), we need to know both $Baseline_{LSBs}$ and Δ. We now show how to estimate both terms for the 2D-IDCT algorithm.

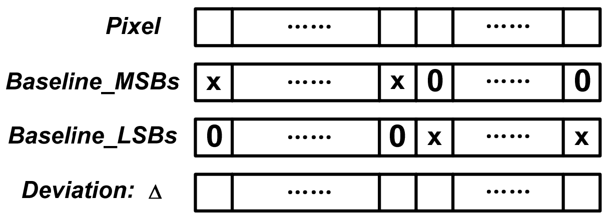


Figure 3.11: Bitmap for all components.

To determine Δ, we can rewrite the 2D-IDCT algorithm, plug in the coefficients and separate the baseline and deviation terms:

$$\begin{aligned} x_{i,j} &= \frac{c(i,j)}{2} \cdot \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C_{u,v} X_{u,v} \\ &= \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_{i,j} \cos[\frac{\pi u}{16}(2i+1)] \cos[\frac{\pi v}{16}(2j+1)] X_{u,v} \qquad (3.2) \\ &= C_{0,0} X_{0,0} \cdot \cos[\frac{\pi 0}{16}(2i+1)] \cos[\frac{\pi 0}{16}(2j+1)] + \Delta \end{aligned}$$

The first term is the baseline:

$$\begin{aligned} Baseline &= C_{0,0} X_{0,0} \cdot \cos[\frac{\pi 0}{16}(2i+1)] \cos[\frac{\pi 0}{16}(2j+1)] \\ &= \frac{1}{8} X_{0,0} \end{aligned}$$

From this, we can see that the baseline value can be simply computed by shifting the DC component ($X_{0,0}$).

The deviation Δ becomes:

$$\Delta = \frac{1}{4} \sum_{\substack{v \neq 0 \\ if\ u=0}}^{7} \sum_{\substack{u \neq 0 \\ if\ v=0}}^{7} C_{i,j} \cos[\frac{\pi u}{16}(2i+1)] \cos[\frac{\pi v}{16}(2j+1)] X_{u,v}$$

Let t represent the upper bound of $|X_{u,v}|$, i.e.:

$$| X_{u,v} | \leq t \qquad (3.3)$$

and

$$| \Delta | \leq C \cdot t \qquad (3.4)$$

where C is the following constant:

$$C = \max_{i,j} (| \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_{i,j} \cos[\frac{\pi u}{16}(2i+1)] \cos[\frac{\pi v}{16}(2j+1)] |)$$

We now rewrite (3.1) as:

$$-\Delta \leq C \cdot t \leq Baseline_{LSBs}$$

$$Baseline_{LSBs} \leq 2^i - C \cdot t \leq 2^i - \Delta$$

To derive a tighter bound for $Baseline_{LSBs}$:

$$C \cdot t \leq Baseline_{LSBs} \leq 2^i - C \cdot t \qquad (3.5)$$

Again, this is the inequality that needs to be satisfied to guarantee that *Pixel[N:i] = Baseline[N:i]*.

Based on inequalities (3.3) and (3.5), we can perform error limiting. We first partition t into five regions and pre-calculate the corresponding values of $C \cdot t$ and $2^i - C \cdot t$ for different i. At runtime, we check the AC components $X_{u,v}$ of the 8×8 input block to determine a smallest upper bound t. Using the pre-computed bounds for the given t and the *Baseline* computed from the DC component, we then find the smallest i for which (3.5) holds. If there is such an i, we replace pixel bits [N:i] with their baseline equivalents. Otherwise, no substitution is performed.

To further improve the error limiting technique, we can reduce the upper bound t to allow more bits to be substituted. In practice, the deviation only reaches the upper bound when all $|X_{u,v}|$s equal to t and their signs are the same as the corresponding 2D-IDCT algorithm coefficients. This rarely happens. Therefore, we can choose a smaller t. Such tweaking may lead to mis-substitution, but it removes other severe timing errors. We can empirically determine a practical value of t to use.

Another way to improve the performance of error limiting is to introduce an allowed and forbidden state: when (3.4) is violated, we still have some knowledge about what the MSBs of the pixels ought to be. For example, if the first three MSBs of the baseline value are binary 100, and if (3.4) is violated, but the value of $C \cdot t$ is still below a certain level, then the correct pixel MSBs can be 011 or 101, but definitely not 000 or 111. In such cases, we can remove the MSBs error by defining 000 and 111 as forbidden states, and use control logic to change these two states to either 100 or 011. Assuming the minimum difference between the allowed state and the forbidden state is D, we introduce another bound t':

$$t' = D/C$$

where C is the constant mentioned before. If all AC components $X_{u,v}$ are less than t', we know that the MSBs within a certain 8×8 block cannot take any value in the forbidden state. In our implementation, we use this concept to prevent the case when a white pixel becomes black and vice versa. We pick two sets of baseline MSBs values (100 and 001, which are found to be most likely to have significant errors) and determines the corresponding forbidden states for them. At run-time, we check whether the output MSBs fall into the forbidden states. If so, the MSBs will be substituted with the baseline MSBs.

To determine which filtering method to use for a given application, we apply each of them to multiple images. The simulation results are shown in Figure 3.12. Figure 3.12 shows that median filtering results in images with lower maximum quality. However, it outperforms error-limiting in the low-energy region. The cross-over point between the two techniques depends on the image, and we define the lowest cross-over point as a threshold. The threshold needs to be determined through simulations of representative images under scaled voltage. The quality required by a specific application scenario determines the technique to be employed.

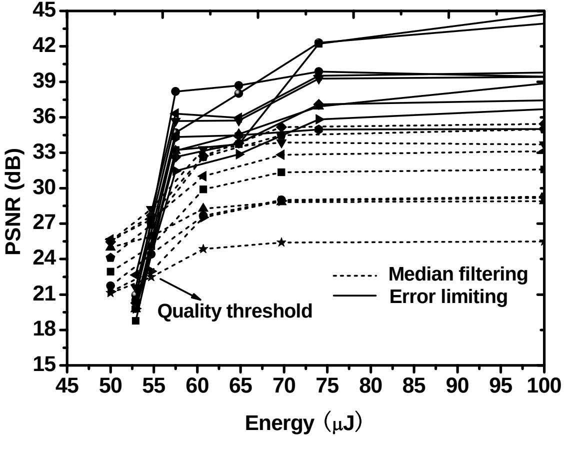


Figure 3.12: Comparison between median filtering and error limiting.

3.5 EXPERIMENTAL RESULTS

We applied our techniques to folded and unfolded 2D-IDCT and 2D-DCT realizations. In a folded architecture [34], each 1D-IDCT/DCT shares the same physical, pipelined multiplier-accumulator (MAC) unit containing an adder and a multiplier, which minimizes the area of the whole design. As shown for a 2D-IDCT in Figure 3.13, the first 1D-IDCT computes intermediate results using the arithmetic unit and stores data in memory. The next 1D-IDCT then uses data from memory and the same arithmetic unit to compute the final result. In order to compare the effectiveness of our techniques on different architectures, we also implemented both a similar, folded DCT as well as an unfolded, pipelined 2D-IDCT.

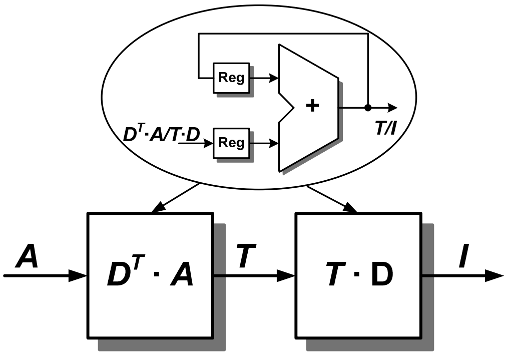


Figure 3.13: 2D-IDCT design architecture.

The IDCT data and coefficient matrices *A* and *C* have 16-bit and 8-bit resolution, respectively. By contrast, in the DCT case, both data and coefficient matrices have 8-bit resolution while the output resolution is 16-bit. The multiplier in the arithmetic unit is pipelined and has a width of 8×16 bits. The adder has a width of 24 bits and it operates as an accumulator in the IDCT/DCT process. The error control techniques we introduce can be applied to various types of adders, and we realized them on a ripple-carry adder, a carry-select adder and a carry-lookahead adder. We conduct most of our experiments using a ripple-carry adder because it has better Q-E tradeoff at low energy compared to a tree adder, as we will demonstrate below. Such a design restricts the timing errors entirely to the adder. As our results will show, however, our implementation of a scaled slow adder requires less energy than a balanced design that uses a fast adder to achieve the same quality and performance. The test images are from the USC-SIPI image database [35]. Only the Y signal of each Y:Cb:Cr format image is used.

The 2D-IDCT/DCT is implemented in Verilog-HDL and synthesized using Design Complier with the OSU 45nm PDK. And we use Synopsys Hercules to translate the RTL code into a SPICE netlist. Then, we build a NanoSim + VCS testbench to enable both RTL-level and SPICE-level simulations to obtain final output images and energy-delay results, respectively. IDCT and DCT follow the same computational process. Other than using different coefficients and reversing the partitioning of computations along output instead of input matrices, the same circuit and timing error control can be applied in both cases. In the following, unless otherwise specified, we first show results for the 2D-IDCT case and then extend those to a 2D-DCT.

A 2D-IDCT block is usually used in an image decompression system in which input data is quantized. We therefore added a quantization step before the IDCT block to generate realistic 8×8 input data. The quantization table is taken directly from the JPEG standard (Table *K.1*). We experimented with different compression ratios to test the effectiveness of proposed techniques; the results are in Figure 3.14. In these experiments, the ratio is defined as the total number of bits required for the original images divided by the total number of bits required for quantized DCT data. A high compression ratio means that the numbers in the quantization table are large and more high frequency components are reduced to zero. At low compression ratio of 2, the impact of quantization on the effectiveness of proposed techniques is not noticeable. At compression ratio of 40, the initial quality is lower, as is expected. However, the rate of quality degradation is also lower. This is because aggressive quantization leads to many high-frequency DCT coefficients becoming zero, which reduces the likelihood of timing errors due to addition of small opposing-sign operands. Despite this intrinsic benefit of quantization, the proposed techniques are effective even at high compression ratios. This is because there are still many non-zero entries left in the DCT matrix.

The experiments show that the proposed techniques with quantized data achieve about 60% to 80% energy savings for various levels of quantized data. In the following discussions, we use quantized data with a compression ratio of 40 to measure the achieved energy savings.

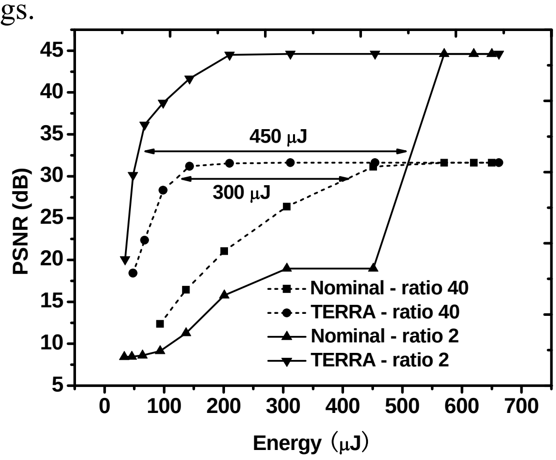
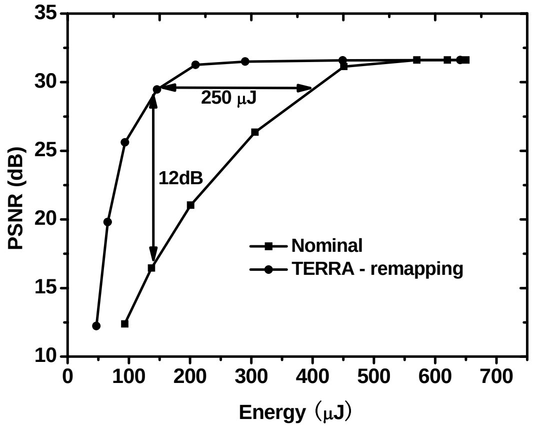
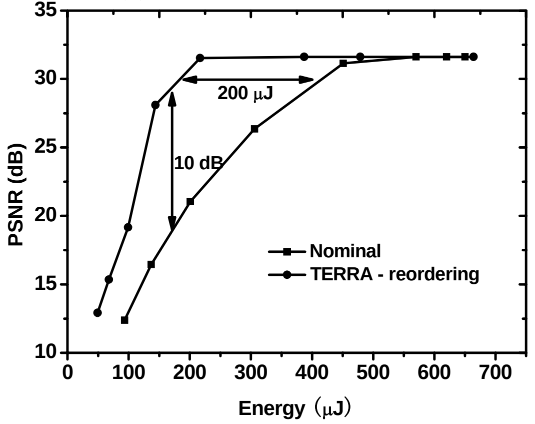


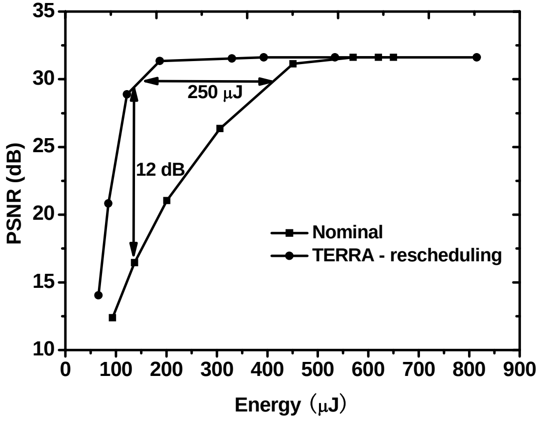
Figure 3.14: Q-E tradeoff under different compression ratios.



(a) Reduced-width adder



(b) Accumulation reordering



(c) Re-budgeting of two cycles to step 1

Figure 3.15: Individual PSNR vs. energy profiles.

Table 3.1: Energy Saving and Area of IDCT.

	V_{DD}	Energy saving	Area μm^2	Delay@1.1V
Original	1.1	0%	149949.39	3.87ns
Adder	0.95	49.1%	150482.45	3.90ns
Reorder	0.95	32.1%	154611.35	3.90ns
Step1&2	0.95	42.1%	150073.69	3.87ns
All three	0.90	63.1%	155175.45	3.91ns

Table 3.1 shows the energy savings for each technique and their combination. Energy savings are computed at PSNR = 30dB with the processing rate being a constant 11ms per *256×256* frame. The resulting PSNR vs. energy profiles for each technique are shown in Figure 3.15.

Table 3.2: Energy under different combinations.

Component	0	1	2	3	4	5
Energy (μJ)	570	440	329	344	354	365

Individual techniques can be combined to achieve maximum energy savings. However, since the described techniques all have varying impact on the different frequency components, their optimal combination is not obvious. Using the technique of Section 3.3, a larger timing budget is given to the earlier algorithm step. This change impacts all frequency components. On the other hand, the technique of Section 3.1 impacts mainly the high-frequency components (since they are the components that involve small-valued operands). Finally, the technique of Section 3.2 impacts operands with opposing sign, no matter if they are low- or high-frequency components.

Based on these observations, we devised the following strategy for selectively applying techniques to different algorithm steps and frequency components: (1) In Step 1, we allocate more cycles only to the low-frequency components while using dynamic reordering and a reduced-width adder to process the high-frequency components; (2) In Step 2, timing errors are not propagated into later steps, so only the reduced-width adder and dynamic reordering are applied. In this combination, the total number of clock cycles needed in Step 1 is smaller than what the technique introduced in Section 3.3 would require to achieve the same quality level. Hence, under a fixed total time, the adjusted clock period T_{clk}' is larger and there exists more timing slack for energy savings.

The key problem is to determine which low-frequency components in Step 1 require more cycles for their processing after applying techniques from Sections 3.1 and 3.2. Since the size of the frequency coefficient matrix in a 2D-IDCT is small, we can do a brute-force exploration to determine the best assignment. Table 3.2 shows the results of such simulations. Results indicate that the smallest energy is obtained when allocating more time (two cycles in our implementation) to the computation of the first two low-frequency components.

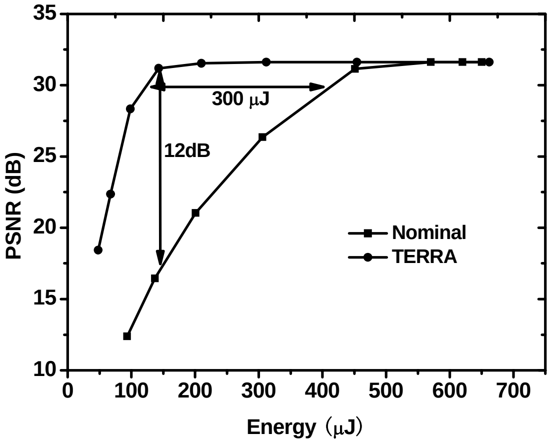


Figure 3.16: Combined PSNR vs. energy profile.

The PSNR vs. energy curve for the combination of techniques is shown in Figure 3.16. A significantly improved trade-off curve is generated by a non-trivial combination of individual techniques. Finally, a set of sample images under scaled V_{DD} is shown in Figure 3.19. Note that achieving a similar energy reduction by conventional V_{DD} scaling would result in unacceptable degradation of image quality (Figure 3.19(b)). To further demonstrate the effectiveness of our TERRA techniques, we test the 2D-IDCT design with various images [35], and the Q-E tradeoff curves are shown in Figure 3.17. As shown in the figure, for all the test images, our design can significantly reduce the probability of large timing errors, and thus improve PSNR at smaller energy values.

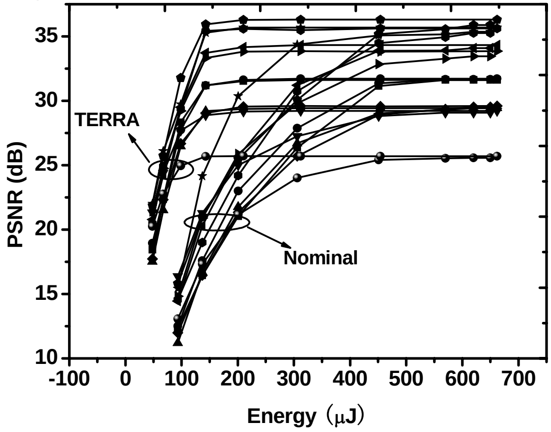
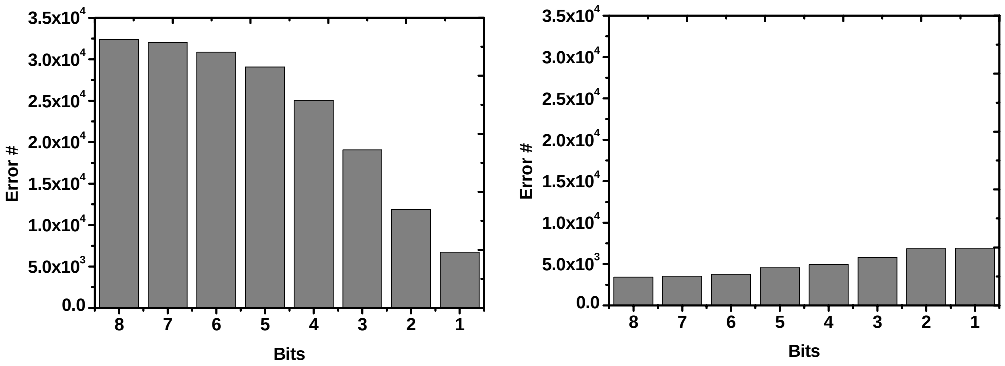


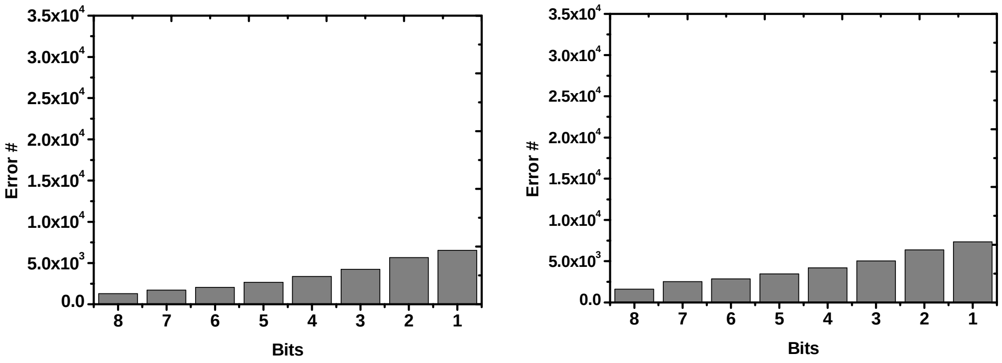
Figure 3.17: Q-E curve for various images with/without our techniques.

Figure 3.18 shows the effectiveness of techniques at the level of individual bits. Originally, when no design optimizations are applied, the output image of the 2D-IDCT has severe MSBs errors (Figure 3.18 (a)). After applying the three optimizations (Figure 3.18 (b)), both MSB and LSB errors are reduced. However, MSB errors are reduced much more significantly than LSB errors.



(a) Bit errors in nominal design.

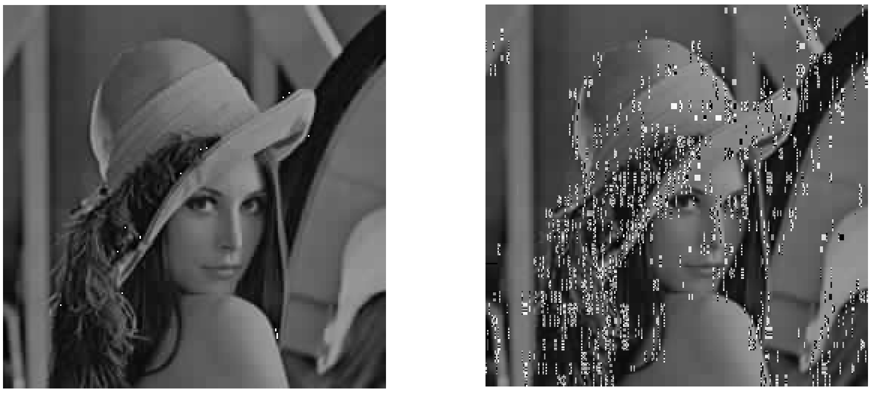
(b) Bit errors with optimizations.



(c) Bit errors with optimizations and error limiting.

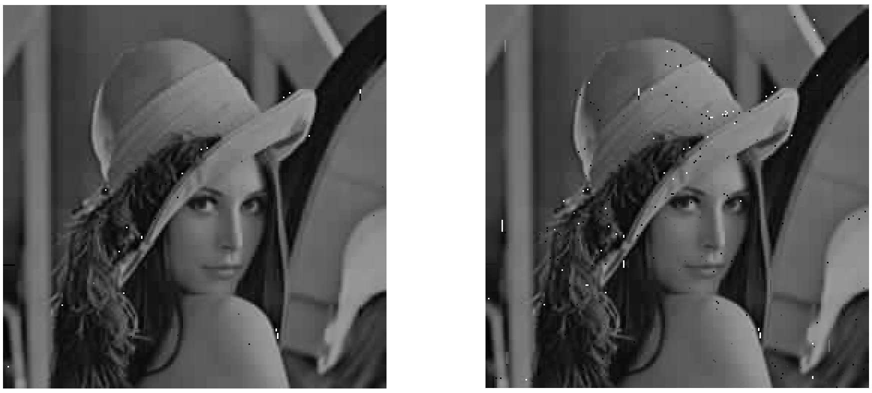
(d) Bit errors with optimizations and median filtering.

Figure 3.18: Frequency of errors in individual bit positions.



(a) Nominal:
Energy=570 μJ
PSNR=31.6dB

(b) Nominal:
Energy=137 μJ
PSNR=16.5dB



(c) TERRA:
Energy=143 μJ
PSNR=31.2dB

(d) TERRA:
Energy=98.5 μJ
PSNR=28.3dB

Figure 3.19: Image quality under different energy budgets.

To better quantify the effectiveness of this work, we compare the achieved energy savings with those produced by alternative approaches to approximate implementations of image processing circuits. For that, we also applied the TERRA techniques described in this dissertation to a 2D-DCT design. We also implemented an approximate, folded 2D-DCT design using the optimized computation sharing multiplication technique

(CSHM) described in [18]. Both designs use the same folded architecture described before, while the CSHM-based 2D-DCT replaces the pipelined MAC unit with a CSHM arithmetic unit. The two designs are synthesized using the same 45nm OSU library, and they are compared at identical performance. Results in Figure 3.20 show that the initial quality allowed by the CSHM design is slightly lower than in our implementation. This is due to the coefficient restructuring performed in CSHM. Under scaled voltage, the CSHM design discards long but less important paths, i.e., the ones for high frequency components. Since the DCT data in our experiments is quantized, many high frequency components are already zero (which equals to discarding them). As a result, we can see that the quality difference between the CSHM-based design, which computes all *8×8* entries and the design which computes only the top-left (*5×5*) entries is negligible. Results suggest that the energy savings with the CSHM technique are about 50% at a quality of around 29dB, while using our TERRA techniques on a 2D-DCT can save about 71% energy. At lower quality, however, the CHSM design performs slightly better.

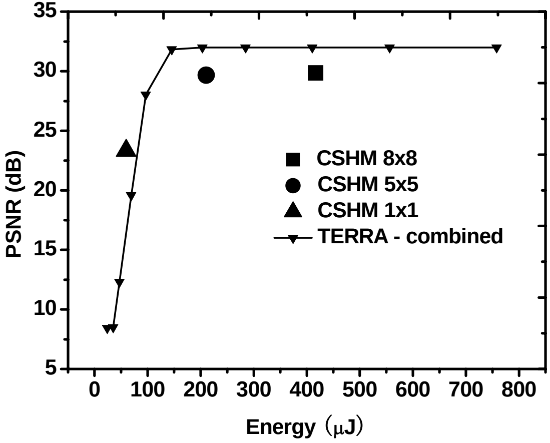


Figure 3.20: Approximate 2D-DCT design: comparison of this work and CSHM.

We also implemented an unfolded 2D-IDCT system with our techniques. In this design, there are physically separate MAC units for step 1 and step 2. At the expense of roughly doubling the area, this allows the two steps to be pipelined. Simulation results for the unfolded IDCT are shown in Figure 3.21.

The energy saving is 68% in the unfolded design after using TERRA techniques. These results suggest that the techniques are effective on both folded and unfolded realizations. Due to the higher base area and hence energy of an unfolded system, achievable energy savings are higher than in the folded case under the same performance.

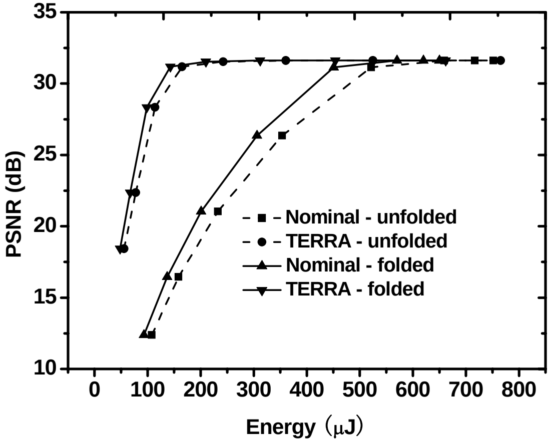


Figure 3.21: An unfolded IDCT design.

In the discussion so far, TERRA techniques have been implemented using 2's complement data representation. Experiments demonstrate that they are also effective in systems based on sign-magnitude representation. We implemented three sign-magnitude 2D-IDCT systems using the same folded architecture with a single pipelined MAC unit as described before. The MAC units in these three implementations are designed based on different methods for performing sign-magnitude opposing-sign addition [36]: (1) translating sign-magnitude data to a 2's complement representation and using a regular adder to perform operations; (2) using a separate subtractor to handle opposing-sign additions, which internally are realized in 2's complement logic; and (3) employing a 1's complement subtractor to add opposing-sign numbers. Simulation results are shown in

Figure 3.22. Experiments show that by applying TERRA techniques about 50% energy saving can be achieved at a quality of 30dB. This is because even in sign-magnitude representation, additions of small opposite-sign numbers trigger the longest carry or borrow propagation in the adder or subtractor, respectively. Furthermore, due to the hardware overhead for conversion, sign-bit logic and additional subtractors, the base energy of sign-magnitude systems is higher.

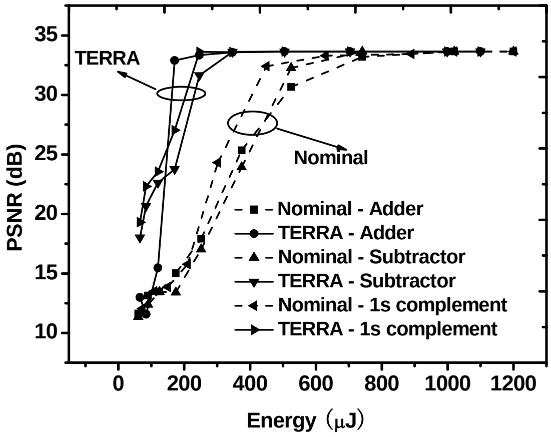


Figure 3.22: An IDCT design using sign-magnitude representation.

In earlier work, a widely used energy-saving technique has been used to reduce the internal data precision through quantization. To compare the effectiveness of different strategies, we implemented a 20-bit, folded 2D-IDCT design and compared its energy efficiency with our original 24-bit 2D-IDCT. In the 20-bit design, we employed a single MAC unit with 20-bit bitwidth, the input data and coefficients are 16-bit and 8-bit, respectively, while internal operations are truncated to 20 bits. Simulation results are shown in Figure 3.23. Our optimizations are independent of the selection of a particular precision and we applied them to both designs. Results show that by using our techniques, about 60% energy savings can be achieved for a 20-bit design. Importantly, we find that the 24-bit TERRA design has uniformly better quality and energy than the reduced-precision implementation.

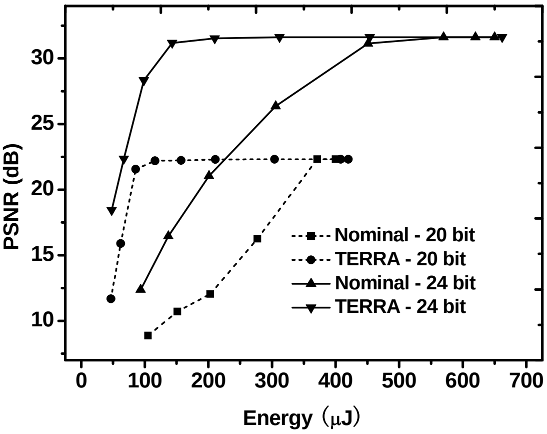


Figure 3.23: An IDCT design using 20 bits vs. 24 bits.

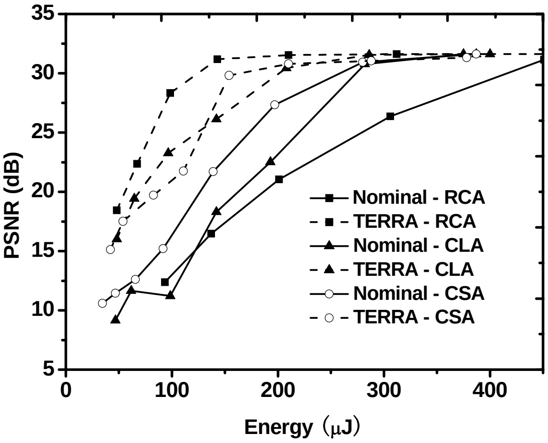


Figure 3.24: Q-E tradeoff using various adder architectures.

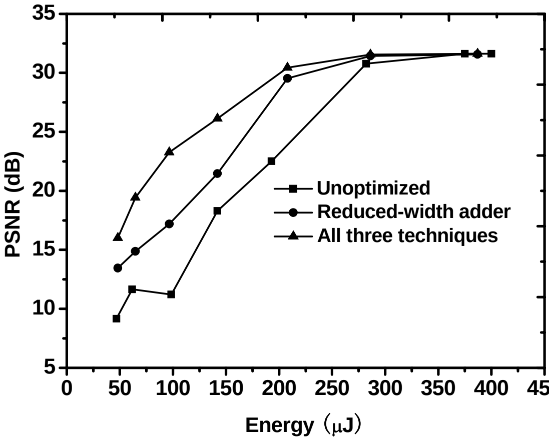


Figure 3.25: Reduced-width CLA adder.

The experiments so far relied on a ripple-carry adder. We now compare the error-energy behavior of different adder architectures, specifically, of a ripple-carry adder (RCA), carry-select adder (CSA) and carry-lookahead adder (CLA), see Figure 3.24. We find that the 2D-IDCT designs using CLA, and to a lesser extent CSA, have smaller base energy if timing errors are not permitted. Since CLA and CSA have shorter critical paths, their initial energy advantages are due to our ability to reduce voltage more significantly without causing timing errors. Under scaled voltages, our techniques also enable significant energy savings of 48% and 56%, respectively, for CLA and CSA based designs. Our techniques are applicable because timing errors are still primarily caused by small operands. This is confirmed by the effectiveness of solely applying a reduced-width adder to such operands in a CLA-based design, as shown in Figure 3.25. From Figure 3.24 we observe that the magnitude of energy reduction is largest for a RCA. As a result, while a RCA has a higher base energy, once timing errors are allowed, RCA has lower energy than CLA and CSA under equivalent performance and quality. We hypothesize that the reason is due to the narrower, more balanced distribution of timing paths in the

CLA or CSA compared to RCA. This appears to be an example of the known behavior that an overly optimized design is less resilient to errors.

Our design so far relied on a fast pipelined multiplier that is paired with a slow adder, where the latter is overscaled using TERRA techniques to control timing errors under a common timing budget. By contrast, in a traditional balanced design, a fast, pipelined multiplier would either be paired with a fast adder such as a CLA (balanced design 1), or a slow adder, such as a RCA, would be combined with a slower, non-pipelined multiplier (balanced design 2) to meet a certain performance goal. To compare the different design philosophies, we implemented both balanced approaches. The resulting Q-E tradeoffs are shown in Figure 3.26. Balanced designs have a lower timing-error free base energy. This is because we can exploit timing slack for additional energy savings. However, under a timing error acceptance strategy, the unbalanced design can be scaled even further while maintaining almost perfect quality. As already indicated above (Figure 3.24), at least in some cases, a slower RCA scaled to the same performance achieves a lower energy than a design that balances slack by using a multiplier with a fast adder.

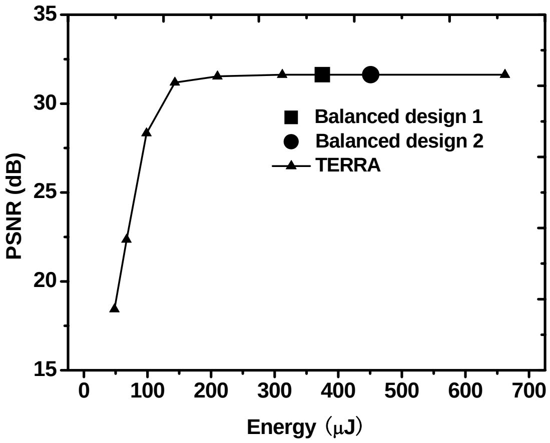


Figure 3.26: QE tradeoff for balanced/unbalanced design.

Nevertheless, remaining errors lead to visually noticeable salt-and-pepper and stripe artifacts. In section II-D, we proposed two types of post-processing techniques for 2D-IDCT to filter out such artifacts. The combination of post-processing with error shaping optimizations is straightforward. For the median filtering technique, output data is buffered in memories with a size equal to the filtering window length and the approximate median of one data window is selected as the filtered result. This process operates directly on the output data and is independent from the error shaping techniques. Figure 3.18 (d) shows the bit-level error count of a whole image after using median filtering. Compared to Figure 3.18 (a), we can see that median filtering is effective in further reducing the MSB errors in the output image. However, it also leads to a slight increase in LSB errors.

Post-processing via error limiting involves both input and output data. Input data is analyzed to determine whether value limiting should be applied and how many MSBs should be affected. Output MSBs are subsequently overwritten if the control logic on the input side triggers the substitution. This technique is also independent of core error shaping techniques. As shown in Figure 3.18(c), error limiting reduces MSB errors without increasing errors in the LSBs.

Area, energy and image quality of the post filtering techniques are shown in Table 3.3, where energy and quality is measured at the same voltage level as in Figure 3.19 (d) (PSNR of around 32dB). Corresponding images after post-processing are shown in Figure 3.29 (a) and Figure 3.29 (b). Both filtering techniques introduce a slight area and energy overhead. However, their 3 to 5dB PSNR improvement is larger than what simple scaling of voltages to the same energy level would deliver in an unfiltered design. Figure 3.27 shows the PSNR vs. energy profile of post-filtering techniques. Although image quality is generally improved, we can note that at high energy levels, the PSNR curve for median filtering is worse than the unfiltered case. Yet, even at such high energy levels, the median filtered image looks as good as, if not better, than the unfiltered one. Furthermore, in intermediate regions, the PSNR metric is highly non-monotonic, which makes it difficult to fairly evaluate energy-quality tradeoffs. Therefore, we utilize an alternative multi-scale structural similarity (MS-SSIM) [37] metric, which is designed to accurately assess humanly perceived image quality. The MS-SSIM curve of different techniques under varying energy levels is shown in Figure 3.28. MS-SSIM results confirm that, compared to the original case, both post processing techniques improve the perceived image quality over the whole energy range. This coincides with the visual appearance of the images (Figure 3.29), which have less salt-and-pepper noise and look better. In addition, in MS-SSIM profiles, quality drops off at lower energy levels. As such, further energy savings can be achieved while maintaining an overall excellent image quality with an MS-SSIM > *0.90*. Figure 3.29 (c) and Figure 3.29 (d) show resulting test images and energy levels.

As discussed previously (Section 3.4, Figure 3.12), there is a quality-energy tradeoff and cross-over point for choosing between different filtering techniques. At high energy levels, median filtering introduces intrinsic errors and the error limiting technique is better in terms of PSNR. But median filtering has a slightly lower area overhead and outperforms error limiting when energy levels begin to drop. This effect is more pronounced when looking at MS-SSIM profiles, where perceptual base quality is less affected by the intrinsic smoothing introduced through median filtering. Overall, based on such tradeoffs, designers can select which post-processing technique to use depending on desired quality levels and energy budgets.

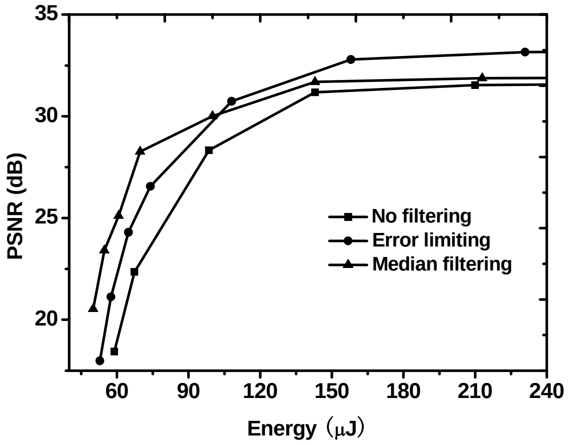


Figure 3.27: PSNR curve for filtered and unfiltered image.

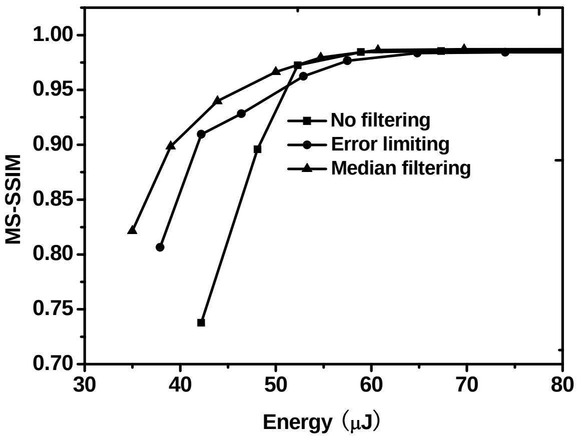
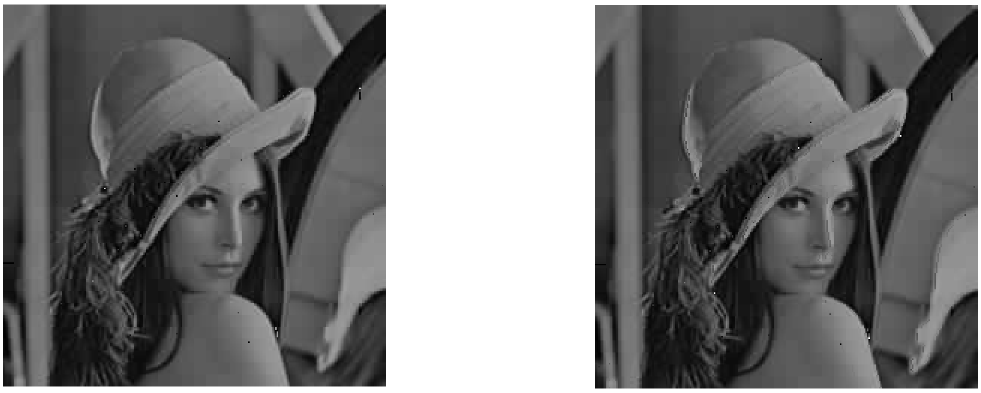


Figure 3.28: MS-SSIM curve for filtered and unfiltered image.

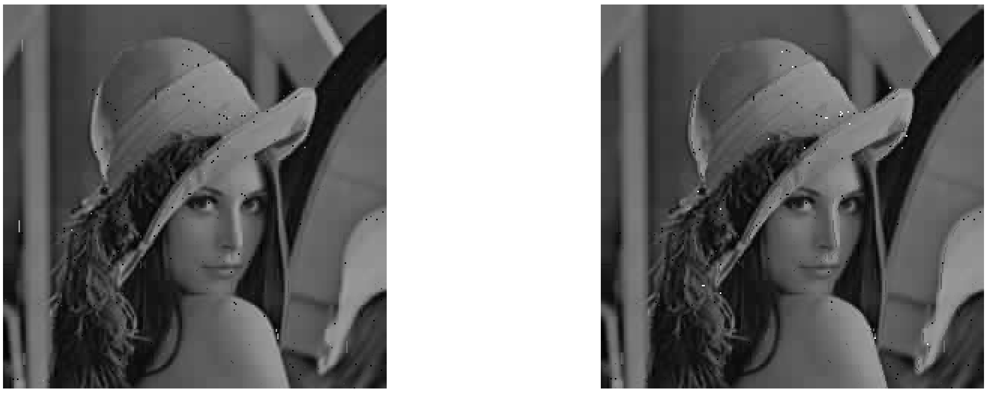
Table 3.3: Area/energy of IDCT with post processing.

Type	All three	Error Limiting	Median Filtering
Energy (μJ)	143	108	100
Area (μm^2)	155175.45	159166.38	159009.63
PSNR (dB)	31.2	30.7	30.0
MS-SSIM	0.9856	0.9844	0.9873



(a) Test image with error limiting: Energy=158.0μJ PSNR=32.8dB MS-SSIM=0.9844

(b) Test image with median filtering: Energy=143.0μJ PSNR=31.7dB MS-SSIM=0.9873



(c) Test image with error limiting: Energy=108μJ PSNR=30.7dB MS-SSIM=0.9844

(d) Test image with median filtering: Energy=100μJ PSNR=30.0dB MS-SSIM=0.9873

Figure 3.29: Image quality after post-processing.

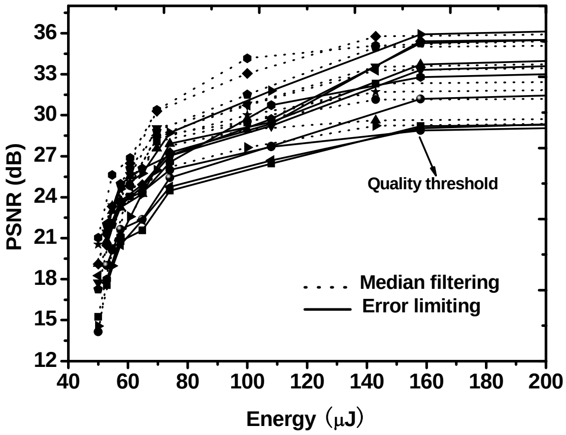


Figure 3.30. Comparison between median filtering and error limiting using quantized data.

3.6 TIMING ERRORS IN BIG ADDER AND SMALL ADDERS

In this section, we will prove the following theorem for 2'complement number additions.

Theorem 3.1. In the absence of overflows, the output timing error in a wide adder is greater than or equal to the error in a smaller-width adder when both adders process the same operands.

We make the following definitions and assumptions:

1) There are two adders, Adder 1 and Adder 2. Adder 1 has a width of N_1 bits and Adder 2 has a width of N_2 bits, with $N_2 > 1$ and $N_1 > N_2$. Adder 2 is sign-extended to N_1 bits, as shown in Figure 3.31;

2) The output of Adder 1 is r_1 with an error of e_1, and the output of Adder 2 is r_2 with an error of e_2. Assuming that the correct result of an addition is r, $r = r_1 + e_1 = r_2 + e_2$;

3) Adder 1 and Adder 2's N_2 LSBs are the same;

4) No overflow in Adder 1 and Adder 2.

The objective is to prove: $|e_1| \geq |e_2|$

We use a divide-and-conquer strategy to prove the above inequality. In the absence of overflows (assumption 4), the correct result \$r\$ can be represented using N_2 bits, i.e.:

$$-2^{N_2-1} \le r \le 2^{N_2-1} - 1$$

This can be rewritten as $|r| < 2^{N_2-1}$. Similarly, since Adder 2 only has N_2 bits, $|r_2| < 2^{N_2-1}$ and $e_2 = r - r_2$. Therefore, $|e_2|<|r|+|r_2|$ and we can bound e_2 as

$$|e_2| < 2^{N_2-1}$$

There are four cases for r_1 and r_2:

C_0: $r_1 = r_2$;

C_1: $r_1 \neq r_2$, r_1 is correct, r_2 is incorrect;

C_2: $r_1 \neq r_2$, r_1 is incorrect, r_2 is incorrect;

C_3: $r_1 \neq r_2$, r_1 is incorrect, r_2 is correct;

We need to prove that the condition $|e_1| \ge |e_2|$ holds for all four cases.

For C_0, $|e_1| = |e_2|$, i.e. it is trivially satisfied.

For C_1, r_1 is correct and it follows that bits $[N_2:N_1]$ in r_1 are all the same. Likewise, bits $[N_2:N_1]$ in r_2 are sign extended and must always be the same. Hence, r_1's $[N_2:N_1]$ bits must be equal to r_2's $[N_2:N_1]$ bits, and $r_1 = r_2$. Intuitively, if there are no overflows and no timing errors in r_1, then r_2 in the smaller adder must also be correct. This contradicts $r_1 \neq r_2$. Hence, C_1 never happens.

Case C_2 can be subdivided into four subcases according to the N_1th bit of r_1 and r_2, as shown in Figure3.32.

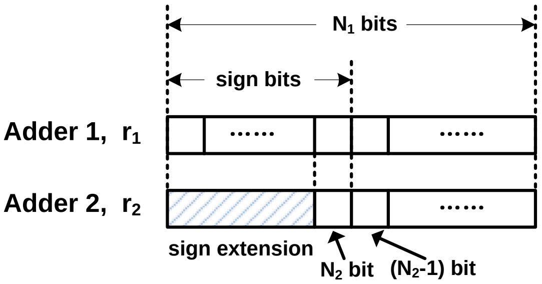


Figure 3.31: The bitmap for adder 1 and adder 2.

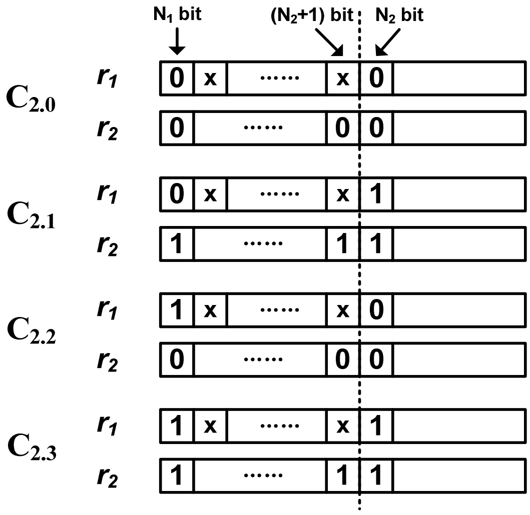


Figure 3.32: The four subcases for C_2.

In subcase $C_{2.0}$, both r_1 and r_2 are positive and $r_1 = r_2+d$ with $d \ge 2^{N_2+1}$. In subcase $C_{2.1}$, $r_1 > 0$ and $r_2 < 0$ with $r_1 = r_2+d$ and $d \ge 2^{N_2+1}$. In subcase $C_{2.2}$, $r_1 < 0$ and $r_2 \ge 0$ with $r_2 = r_1+d$ and $d \ge 2^{N_2+1}$. Finally, in subcase $C_{2.3}$, $r_1 < 0$ and $r_2 < 0$ with $r_2 = r_1+d=r_2$ and $d \ge 2^{N_2+1}$. For subcases $C_{2.0}$ and $C_{2.1}$, from assumption 2 it follows that $r_2+d+e_1 = r_2+e_2$, i.e. $|e_1| = |e_2 - d| \ge |d| - |e_2| \ge 2^{N_2+1} - |e_2|$. Similarly, for subcases $C_{2.2}$ and $C_{2.3}$, $r_1 + e_1 = r_1 + d + e_2$, i.e. $|e_1| = |e_2 + d| \ge |d| - |e_2| \ge 2^{N_2+1} - |e_2|$. Since $|e_2| \le 2^{N_2} - 1$, it follows that $|e_1| \ge 2^{N_2+1} - |e_2| \ge 2^{N_2+1} - (2^{N_2} - 1) \ge 2^{N_2} + 1 \ge max|e_2|$. Therefore, $|e_1| \ge |e_2|$ for all four subcases.

Finally, for case C_3, $|e_2| = 0$, i.e. $|e_1| > |e_2|$. The above proof relies only on the arithmetic of addition, and it does not depend on hardware implementation of the adder. Hence it is valid to various types of adders.

3.7 SUMMARY

This chapter presented techniques that enable architecture-level shaping of the quality-energy tradeoff under aggressively scaled V_{DD} through controlled timing error acceptance. We demonstrated the implementation of these techniques on a design of a folded 2D-IDCT/DCT architecture. Results show that significant energy savings can be achieved while maintaining a constant performance and good image PSNR. To further improve the visual quality, filtering techniques can be implemented to reduce visual image artifacts.

Chapter 4: Digital filtering

This chapter focuses on developing modifications for digital filter implementations to allow them to tolerate timing errors. Any digital filter implements an affine function of the input signal and recorded output signals:

$$y(n) = \sum_{i=0}^{N} b_i \cdot x(n-i) + \sum_{j=1}^{N} a_j \cdot y(n-j) \qquad (4.1)$$

where $x(i)$ is the input sequence, $y(n\text{-}j)$ is the recorded output signal and a_j, b_i are the designable filter coefficients. These coefficients and their ordering can determine the timing error magnitude and frequency during the filtering operations. In the later discussion, we will show how to manipulate a_j and b_i to reduce the quality loss.

4.1 ERROR CONTROL THROUGH BITWIDTH ADJUSTMENT

We first present a technique that exploits the properties of operand statistics to achieve energy savings. We demonstrate this technique on a digital filter represented in direct Form II in Figure 4.1. The digital filtering process basically involves two types of operations: addition and multiplication. We implement the digital filter based on an industry-standard low overhead, low-power design in which the core multiply accumulate (MAC) operations are realized using a multiplier and adder [23] that are chained to operate in one clock cycle. In such architecture, the critical path is defined by the multiplier-adder chain, where under timing starvation, the addition at the end of the chain will experience timing errors first. Note that other components in the architecture are not on the critical path and can be treated as timing-error free for the amount of slack and range of voltage scaling considered in our experiments. We focus techniques in this dissertation on controlling timing errors in the adder.

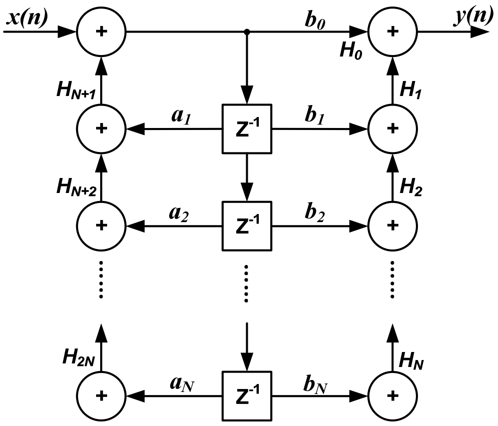


Figure 4.1: Digital filter in direct form II.

When V_{DD} begins to scale down, timing errors impact the results of computation as data moves through the datapath. Timing errors impacting the highly significant bits cause the largest signal quality degradation. Therefore, the objective of the techniques we introduce is to prevent such errors as much as possible. It has been demonstrated before that the early-onset MSB errors are caused largely by processing small opposing-sign additions as discussed in the previous chapter. This is because in 2's complement code, the higher significance bits of small operands are filled with extended sign bits. As a result, in opposing-sign addition of small operands, the carry propagation chain tends to be long and requires longer time to settle. And because the actual operands are small, an early termination of the carry propagation results in large errors.

In previous chapter a method to dramatically reduce the incidence and impact of such timing errors has been introduced. The idea is to utilize a reduced-width adder for small operands to reduce the length of the longest carry. The necessary condition, of course, is that the bitwidth of a reduced-width adder is large enough to represent the accurate result and avoid overflow. In filters, the input data and coefficient bitwidths are determined by the dynamic range specification, while the bitwidth of the datapath is determined by the gain of the transfer function. The core of the technique we now develop is allowing a dynamic reduction in the bitwidth of the adders used in a filter implementation. We exploit the fact that common applications of digital filters operate on data characterized by distributions of a specific type. It is well-known, for example, that speech and music data usually follows a Laplacian distribution [38], as shown in the Figure 4.2 for audio data sample.

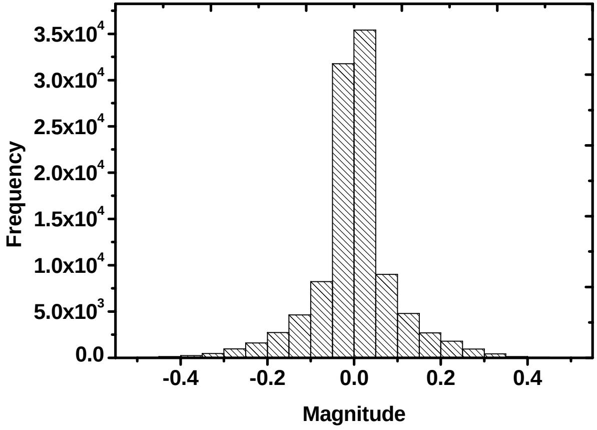


Figure 4.2: Distribution of input data in an audio application.

The key property of this distribution is that most values are close to zero. Nevertheless, in the traditional design paradigm in which timing errors are not allowed, the datapath bitwidth has to be designed to be able to process the largest possible inputs, which in fact occur very rarely.

The proposed architecture a filter that uses a dynamic-width adder is shown in Figure 4.3. The idea is to adjust adder bitwidth dynamically with the purpose of eliminating the early and large timing errors. The architecture requires checking the magnitude of the input data and processing the operands on an adder with bitwidth sufficient for the particular inputs. To allow the results produced by the dynamic-width adder to be used for downstream computation, we perform sign extension on the results to match the full bitwidth. An actual implementation does not require the bitwidth to be continuously adjustable: according to our experiment, just two bitwidth values are sufficient to enable a significant quality-energy tradeoff.

Operand magnitude-checking logic is activated on each addition. The checking logic checks whether MSBs in both operands are either all 1s or all 0s. If so, the checking logic asserts that the bitwidth of both operands are less than or equal to the reduced width, and then the operands are processed by Adder 2, otherwise the operands are processed by Adder 1.

The dynamic-width adder architecture is shown in Figure 4.3. In the figure, it is assumed that only one physical adder is used. The inputs are first sent to the magnitude-checking logic block, which can be implemented compactly. The checking logic uses *AND* gates to determine whether a specified number of higher-significance bits of the inputs are all zeros or all ones. If the condition is true, then a reduced-width adder should be used and the checking logic activates the width-control logic to perform truncation and sign-extension on the adder output. In essence, each time the magnitude-checking logic initiates a reduced-width addition, a smaller effective adder is used for that particular computation. Otherwise, a full-width adder is used.

The overhead of implementing the described technique includes delay, energy and area costs. The magnitude-checking block operates on 5 to 15 bits with a maximum delay of about $log_2(15)$ gate equivalents. Since it runs in parallel to and is faster than the *MAC* unit, no overall delay overhead is incurred. The truncation and sign extension logic adds a multiplexer with one gate delay on the critical path. Note that the critical path is defined when operating in full-bitwidth mode, where sign extension is disabled and the multiplexer is in pass-through configuration. For the range of voltage scaling considered, truncation and sign extension logic, which is active only in reduced-width mode, is free of timing errors and otherwise only contributes to the load on the adder. Overall energy and area overhead includes the magnitude-checking block, *MUX* gates and sign extension logic, which we quantify in the experimental section. Energy overhead also comes from the cost of switching the dynamic-width adder. Switching between a full-width and a reduced-width adder needs to occur on a per-sample basis. The frequency of switchings depends on the statistics of input operands. Experiments show that the incurred energy overhead is, in the end, justified because the entire technique enables significantly higher energy savings through an increased potential for voltage scaling.

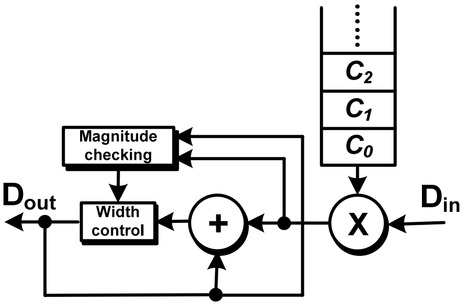


Figure 4.3: Dynamic-width adder architecture.

Because the dynamic-width adder has only a discrete number (two) of allowed bitwidths, we need to address the question of how to find the optimal bitwidth for the smaller adder. We investigate this question on a practical digital filter design, which implements a 5th order FIR filter based on the single MAC architecture. We define Adder 1 as the full-width adder and Adder 2 as the reduced-width adder. The widths of Adder 1 and Adder 2 are W_1 and W_2 separately. Formally, the goal is to find the W_2 which leads to the least quality loss at a given energy budget. We define the following parameters: D_1: the worst-case delay of Adder 1; D_2: the worst-case delay of Adder 2; T: the timing budget of Adder 1 and Adder 2. We assume in this discussion that $T < D_2 \leq D_1$, so timing errors may occur in both Adder 1 and 2.

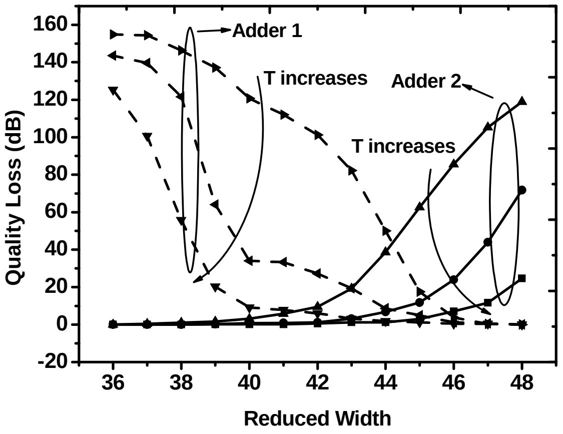


Figure 4.4: Quality loss in two adders vs. Adder 2 bitwidth. Behavior for several values of timing budget is shown.

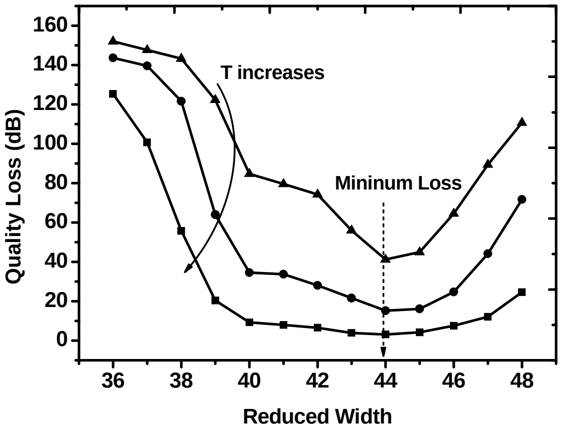


Figure 4.5: Total quality loss as a function of Adder 2 bitwidth.

Through simulation we obtain quality-energy profiles for both adders individually, Figure 4.4, and then jointly, Figure 4.5. From Figure 4.4 we see that as W_2 increases, quality loss in Adder 2 increases while quality loss in Adder 1 decreases. As a result, we see in Figure 4.5 that there exists a width W_{opt} that results in minimal total quality loss. The optimal width value appears to be largely insensitive of the allotted timing budget. When Adder 2 width is greater than W_{opt}, the overall quality loss grows because errors in Adder 2 increase and dominate. Conversely, if Adder 2 width is below

W_{opt}, a larger fraction of the input data is now processed by Adder 1. The result is that more small operands are processed by the full-width adder leading to more frequent and larger errors.

So far, our discussion assumed that we realize a dynamic-width adder with only two possible bitwidths. In principle, it is possible to have a larger number of bitwidths available. We find, however, that increasing the number of bitwidths available does not substantially improve the quality-energy tradeoff. A simulation-based experiment show in Figure 4.6 suggests that quality loss can be significantly reduced by having two bitwidth values. However, a further increase in dynamic-adder complexity to enable a number of bitwidth levels beyond does not improve quality appreciably.

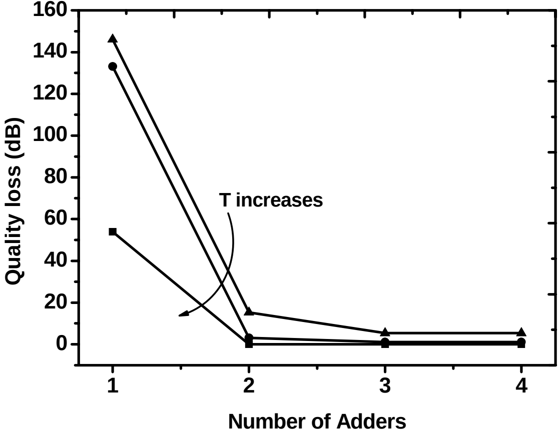


Figure 4.6: Quality loss vs. adder number.

Further, we investigate dependence of the optimal dynamic-width adder parameters on operand statistics, specifically, on the variance of the operand distribution. Quality loss at different W_2 is shown in Figure 4.7 for several values of operand variance.

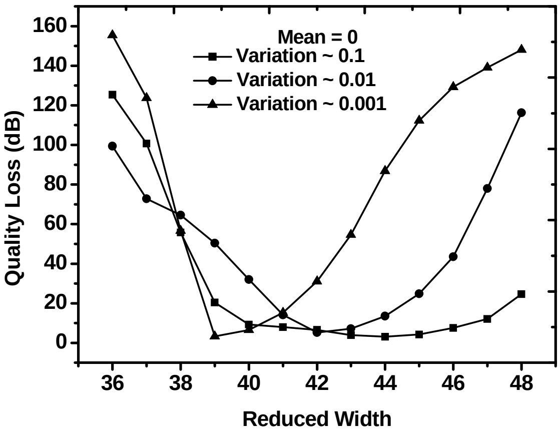


Figure 4.7: Dependence of optimal Adder 2 width on input data variance.

We observe that the optimal Adder 2 width changes notably as the variance of the input data changes. For smaller variance, a larger fraction of data has values that are small, and it is advantageous to make the width of Adder 2 (W_2) smaller.

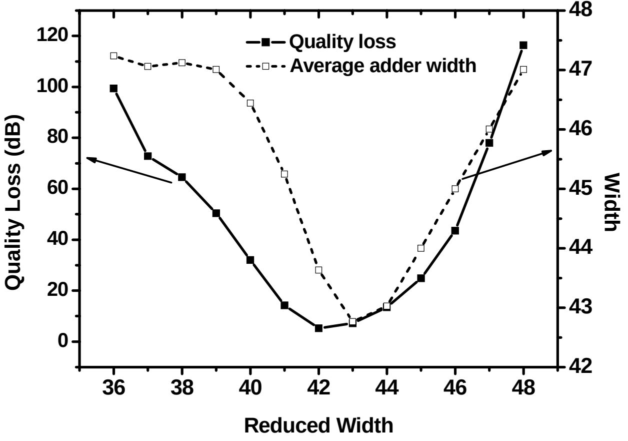


Figure 4.8: Quality loss and average adder width.

To better understand the dependence of optimal dynamic-adder features on input data statistics, we develop an analytical model that allows an estimation of the optimal design parameters. Having an analytical model also removes the need to rely on time consuming simulation-based analysis such as presented above. Recall that the maximum error in each adder is proportional to the width of that adder. Thus, the model is based on the intuition that the bitwidth of Adder 2 (W_2) that minimizes quality loss at any given level of variance also minimizes the average effective adder width. We define average effective adder width as $W_{avg} = W_1 \times p_1 + W_2 \times p_2$, where p_1 and p_2 are the probabilities of using Adder 1 or Adder 2 respectively. As Figure 4.8 shows, we observe that quality loss and W_{avg} track well as W_2 is swept and that their minima coincide to a good degree.

Relying on the tracking of the two metrics established above, we formulate the search for the optimum W_2 via a minimization problem for W_{avg}. The model assumes that input data distribution is given by the Laplacian distribution. We consider the case with only two bitwidths available. Let d_1 and d_2 be the two input operands to the adder, and x be the magnitude threshold for determining whether a larger (Adder 1) or a smaller adder (Adder 2) is used for processing the operands. Then, the problem of minimizing W_{avg} is given as:

$$\begin{aligned} \min_x : \quad & P(|d_1| < x, |d_2| < x) \cdot W_2 \\ & + [1 - P(|d_1| < x, |d_2| < x)] \cdot W_1 \end{aligned}$$

We make the simplifying assumption that the operands are independent which allows us to re-write the minimization problem as:

$$\begin{aligned} \min_x : \quad & P(|d_1| < x) \cdot P(|d_2| < x) \cdot W_2 \\ & + [1 - P(|d_1| < x) \cdot P(|d_2| < x)] \cdot W_1 \end{aligned}$$

The probabilities in the above expression can be evaluated under the assumption that the inputs follow the Laplacian distribution. We further assume that distribution is zero-centered, i.e., μ=0, which is true of many practical instances. The probability density function is given by:

$$f(t \mid \mu, b) = \frac{1}{2b} e^{-\frac{|t-\mu|}{b}}$$

where b is the scale parameter related to variance as $\sigma^2 = 2b^2$. The sought probability can be computed by

$$P(|d_1| < x) = \left(\int_{-x}^{x} \frac{1}{2b} e^{-\frac{|t|}{b}} dt \right)$$

Substituting this probability into the minimization problem, we have:

$$\left(\int_{-x}^{x} \frac{1}{2b} e^{-\frac{|t|}{b}} dt \right)^2 \log_2 x + \left[1 - \left(\int_{-x}^{x} \frac{1}{2b} e^{-\frac{|t|}{b}} dt \right)^2 \right] W_1$$

Finally, this function being minimized can be reduced to a simple form:

$$W_1 - (W_1 - \log_2 x)(1 - e^{-\frac{x}{b}})^2 \qquad (4.2)$$

The minimum of function (4.2) can be computed by setting its 1st derivative to zero and solving the equation. The resulting equation is:

$$\frac{1}{\ln 2 \cdot x}(1 - e^{-\frac{x}{b}})^2 + \frac{2}{b} e^{-\frac{x}{b}} (W_1 - \log_2 x)(1 - e^{-\frac{x}{b}}) = 0$$

We find that the model provides good matching to the simulation-based exploration that we performed earlier. For example, in Figure 4.5, which is generated by gate-level simulation, the optimal values of bitwidth for Adder 2 that minimize quality loss are 44, 42, and 39 for three levels of input data variance. For the same values of variance, the analytical model described above predicts optimal W_2 to be 46, 43 and 40.

To further test the effectiveness of the model, we compare the simulated optimal W_2 and the model-predicted W_2 under different W_1, and the results are plotted in Figure 4.9. Overall, we can see that our model and simulated results match well at inputs with low variance, which are typical in music and speech (with typical values < 0.01).

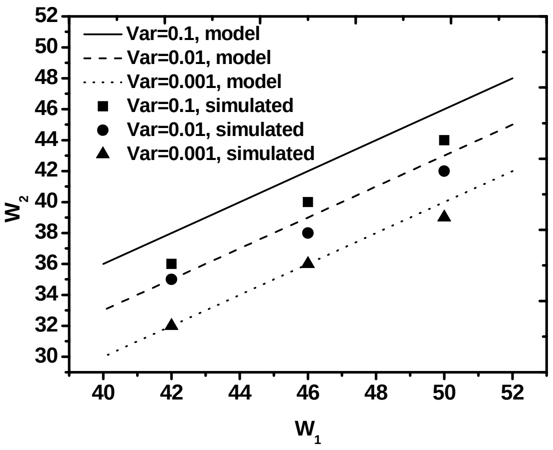


Figure 4.9: Predicted W_2 and simulated W_2 under different W_1.

4.2 ERROR CONTROL THROUGH REORDERING

In the previous section, we discussed how the onset of large timing errors can be controlled by using a reduced-width adder for small operands. Since errors for large operands are unavoidable, such an approach is most effective if the relative fraction of small operand additions is increased. In this section, we introduce techniques to manipulate the input data distribution of intermediate MAC operations by reordering of filter taps. In a traditional single MAC unit design, the width of the MAC unit is determined by the maximal bitwidth over all taps, which is generally independent of any intermediate reordering. However, under a timing error acceptance philosophy, such techniques will allow us to statically apply adders of different width to different taps in order to reduce timing errors. Furthermore, in combination with dynamic bitwidth adjustment (Section 4.1), reordering can, on average, reduce the magnitude of data in intermediate operations and hence increase the effectiveness of this technique.

From the filter expression (4.1), we can see that in computing the final output $y(n)$, the filter needs to generate a set of intermediate results, which correspond to a set of intermediate transfer functions denoted as H_0-H_{2N} in Figure 4.1. These transfer functions determine the maximum possible gain over all frequencies at intermediate nodes and, hence, the minimum required bitwidth for each intermediate result in the datapath. Without affecting the transfer function at the output of the filter, intermediate transfer functions vary when the order of filter taps is changed. As such, we can reduce intermediate gains and hence the bitwidth of intermediate operations by optimally reordering the taps. Such a reordering can be done at design time. It allows us to apply an adder of smaller width to intermediate taps in order to reduce the timing errors under voltage scaling.

To optimize the order of filter taps, we discuss the cases for finite impulse response (FIR) and infinite impulse response (IIR) filters separately. For a simpler FIR filter, the filtering expression (4.1) reduces to:

$$\begin{aligned} y(n) &= \sum_{i=0}^{N} b_i \cdot x(i) \\ &= b_0 x(0) + b_1 x(1) + \ldots + b_N x(N) \end{aligned}$$

where intermediate transfer functions can be expressed as:

$$H_0 = b_0$$

$$H_1 = b_0 + b_1 \cdot z^{-1}$$

$$\ldots$$

$$H_N = b_0 + b_1 \cdot z^{-1} + \ldots + b_N \cdot z^{-N}$$

and the maximum gain G_i over all inputs at intermediate nodes is determined by:

$$G_1 = | b_0 |$$

$$G_2 = | b_0 | + | b_1 |$$

$$\ldots$$

$$G_N = | b_0 | + | b_1 | + \ldots + | b_N |$$

Therefore, minimizing gains is achieved by processing of filter taps in ascending order of absolute filter coefficient values (b_0, b_1, ..., b_N). After reordering, the filter expression becomes:

$$y(n) = b_{i_0} x(i_0) + b_{i_1} x(i_1) + \ldots + b_{i_N} x(i_N)$$

where $i_0 \ldots i_N$ denote the reordered mapping of indices.

For IIR filters, obtaining the optimized order of filter taps requires exhaustively search for all the possible orders, which is extremely inefficient. In practice, we divide the IIR filter coefficients into feedforward and feedback sections. For the feedforward and feedback sections, the coefficient sets (b_0, b_1, ..., b_N) and (a_1, a_2, ..., a_N) represent the coefficients of the denominator and numerator of the filter transfer function, respectively. We reorder the two coefficient sections separately based on the aforementioned FIR filter optimization method.

The implementation of the reordering technique involves changing the order of arithmetic operations and applying a smaller adder to each tap depending on intermediate filter gains G_i. For a single MAC architecture, the abstraction of this reordering technique is shown in Figure 4.10. In the implementation, the tap control logic changes the order in which data and coefficient pairs are fed into the MAC unit. Furthermore, the tap control also truncates and sign-extends results for taps whose gains are small. Note, however, that reordering can not achieve a smaller gain for intermediate transfer functions in all cases. For example, if taps are already optimally ordered or if all coefficients are the same (as is the case in FIR filters using rectangle windows), no further optimizations are possible, but reduced adder widths may still be applicable.

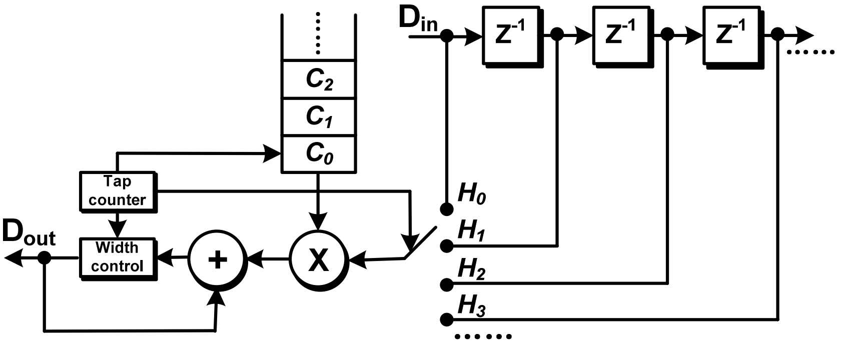


Figure 4.10: Technique abstraction of arithmetic reordering.

4.3 Experiments and Results

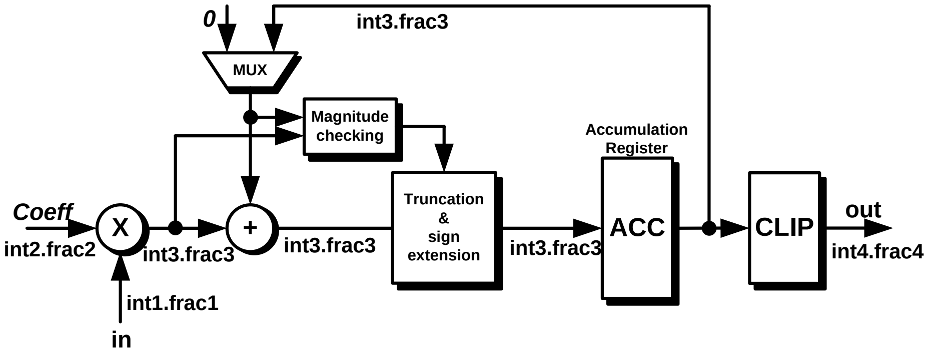


Figure 4.11: Single MAC filtering architecture.

4.3 EXPERIMENT

We base our implementation of a digital filtering architecture with timing error control on a single MAC unit design as shown in Figure 4.11. In this architecture, data is processed using a fixed-point format. Coefficients are represented in a Q*int2.frac2* format, i.e. with *int2* integer and *frac2* fractional bits. Similarly, input, output and intermediate data is assumed to be in Q*int1.frac1*, Q*int4.frac4* and Q*int3.frac3* format, respectively. By changing coefficients and data precisions, such a generic architecture can be used in different filtering applications.

We have applied our approach to several filtering examples in audio, speech and image processing. For audio and speech applications, we designed both FIR and IIR filter implementations. Filter coefficients were obtained using MATLAB's *fdatool*. In all cases, we excited unoptimized and optimized versions of each filter with the same 4 seconds of input data. The quality of the filtered output signals is measured using a segmental SNR (SSNR) metric, which is known to be a better estimator of perceived audio quality than regular SNR [39]. SSNR is measured by dividing the output signals into 20ms segments and averaging over the regular SNR values computed for each segment.

For image processing applications, we designed a 2-D sharpening filter based on our proposed filtering architecture. The filter kernel is generated using default settings of MATLAB's *fspecial* function for sharpening images. The quality of filtered output images is measured using a standard peak signal-to-noise ratio (PSNR) metric.

All designs are implemented in Verilog-HDL and synthesized using Design Complier with the OSU 45nm PDK. Timing and energy values are obtained through Spice-level simulations using NanoSim and VCS. In all cases, the nominal voltage of the filters is 1.1V. Energy savings and area overheads of implementing a combination of our techniques for each of the filter designs are summarized in Table 4.1 and Table 4.2. Energy levels are measured at commonly accepted good quality levels of around 120dB SSNR and 23dB PSNR, respectively.

Table 4.1: Energy and area overhead

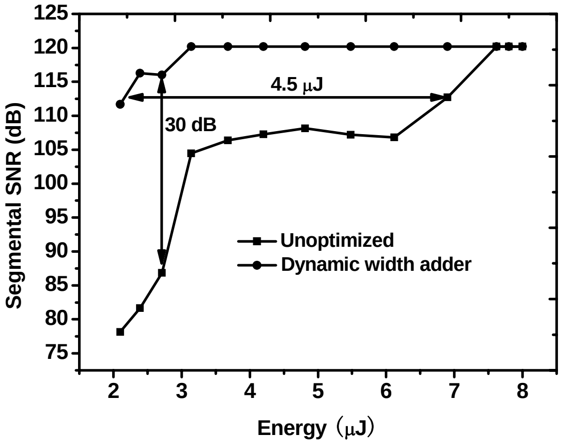
	Energy Overhead			Area Overhead		
	Width	Reorder	Comb.	Width	Reorder	Comb.
FIR	0.5%	0.7%	0.9%	1.6%	1.1%	1.7%
IIR	0.5%	0.3%	0.7%	1.1%	1.0%	1.2%
Sharpen	0.5%	0.4%	0.9%	1.4%	1.1%	2.1%

Table 4.2: Energy savings and performance.

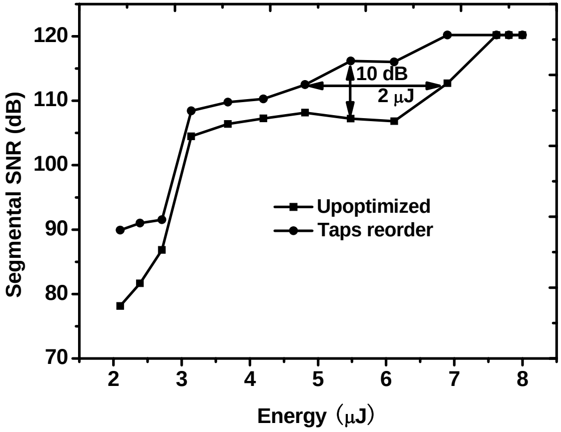
	V_{DD}	SSNR/PSNR	Energy Saving	Speed
FIR	0.75V	119.7dB	58.8%	207.5Mhz
IIR	0.75V	121.7dB	58.1%	207.9Mhz
Sharpen	0.70V	23.3dB	69.7%	303.0Mhz

4.3.1 FIR filter for audio processing

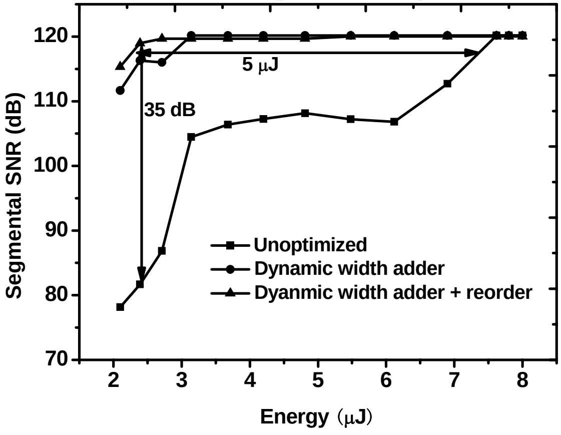
The FIR implemented is a typical 5th-order low-pass filter based on a least-squares design method. The sampling frequency is 22kHz, the pass band ends at 6kHz, and the stop band starts at 7.5kHz. The coefficients are b=(-0.1145, 0.0558, 0.5177, 0.5177, 0.0558, -0.1145). The format for coefficients, input data, intermediate results and final outputs is Q1.21, Q3.29, Q4.50, and Q3.29, respectively. As such, the full adder width is 54 bits, while the reduced-width adder has a precision of 39 bits.



(a) Dynamic bitwidth adjustment



(b) Filter tap reordering



(c) Combined techniques

Figure 4.12: Individual SSNR vs. energy profiles in FIR filters.

The simulation results for the FIR filter are shown in Figure 4.12. Figure 4.12 shows that the dynamic bitwidth adjustment can significantly delay the onset of timing errors when the voltage is scaled down. Furthermore, both dynamic adjustment as well as static reordering (Figure 4.12 (b)) improve the shape of the quality-energy profile, achieving a graceful quality degradation over a wide energy range. Combined (Figure 4.12 (b)), significant energy savings can be obtained while maintaining almost perfect signal quality. To further test the effectiveness of reordering, we applied the technique to a range of FIR filters with orders ranging from *N=15* to *N=63*. Resulting quality improvements at an energy budget of $45\mu J$ are shown in Figure 4.13. In this experiment, we tested 7 different FIR filters. The Figure 4.13 shows that for most FIR cases, the reordering technique can have more than 10dB quality gain compared to the original case.

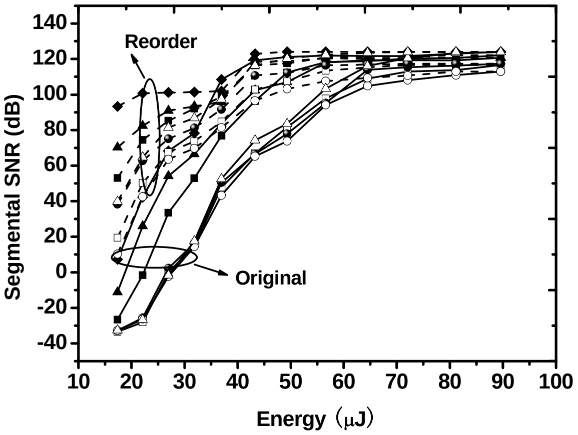


Figure 4.13: FIR filters using reorder technique.

We also tested the sensitivity of the dynamic bitwidth adjustment technique to different types of input data. Figure 4.14 shows the results of feeding jazz, pop and classical music as well as speech audio files into a 63rd-order FIR filter. Results suggest that specifics of speech data trigger worse timing error behavior than in music. This is due to longer segments of silence, which are characterized by small-valued operands triggering early and large timing errors. In all cases, dynamic bitwidth adjustment significantly improves quality-energy behavior, maintaining close to perfect quality over a large energy range. With dynamic adjustment, variations in data magnitudes and their effects on timing errors can be transparently evened out across different characteristics of input data.

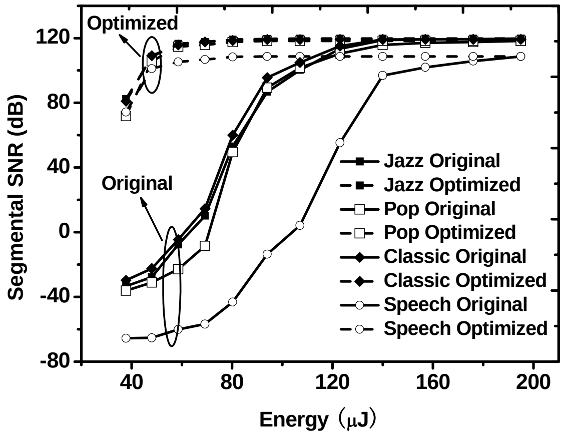
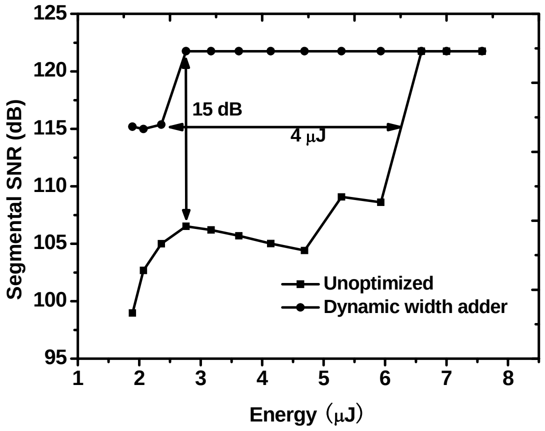


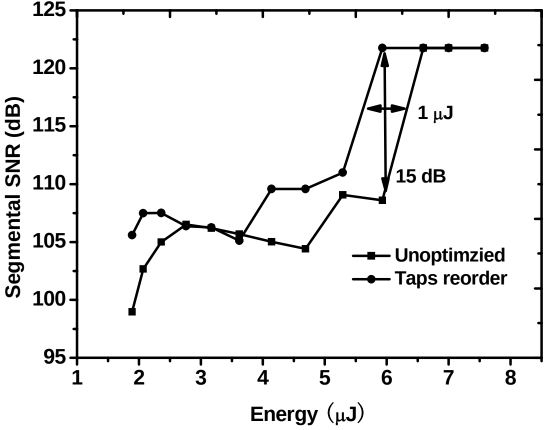
Figure 4.14: FIR filter for different input data with dynamic bitwidth optimization.

4.3.2 IIR filter for audio processing

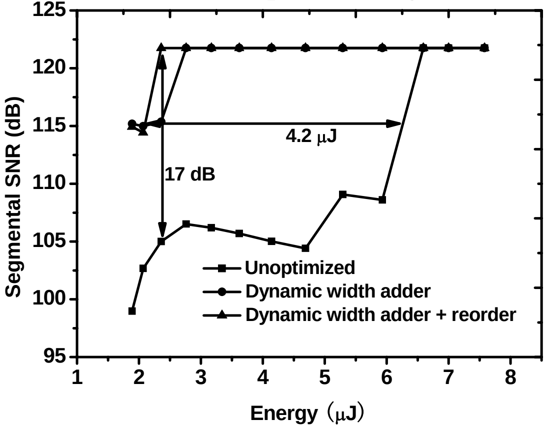
We also implemented a 3rd-order low-pass type II Chebyshev filter as a typical IIR filter example. The sampling frequency is 22kHz and the cutoff frequency is 8kHz. The coefficient sets are *b*=(0.2282, 0.5612, 0.5612, 0.2282) and *a*=(-0.1652, -0.3835, -0.0300). The formats for coefficients, input data, intermediate results and final output are Q1.21, Q3.29, Q4.50, and Q3.29. The full-width and reduced-width adders in this case have 54 and 38 bits, respectively.



(a) Dynamic bitwidth adjustment



(b) Filter tap reordering



(c) Combined techniques

Figure 4.15: Individual SSNR vs. energy profiles in IIR filters.

Results for the IIR filter are shown in Figure 4.15. Figure 4.15 (a) and Figure 4.15(b) show the quality-energy profiles when applying dynamic bitwidth adjustment Both techniques can delay the onset of timing errors, but in the IIR case, dynamic bitwidth adjustment is far more effective than reordering. Furthermore, once errors start to happen, quality drops are overall more severe than in the FIR case. This is due to the feedback loop in the IIR filter, which leads to erroneous results being reused and propagating into subsequent computations. Nevertheless, as shown in Figure 4.15 (c), the combination of techniques is very effective in improving the timing error behavior of the system.

4. 3. 3 Image sharpening filter

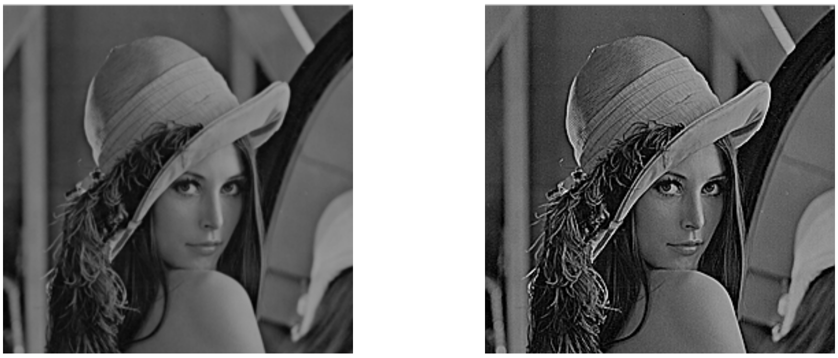
Sharpening of images is used to increase the contrast between bright and dark regions by applying a high-pass FIR filter. The following array represents the coefficient kernel we generated in MATLAB:

$$h = \begin{vmatrix} -0.1667 & -0.6667 & -0.1667 \\ -0.6667 & 4.3333 & -0.6667 \\ -0.1667 & -0.6667 & -0.1667 \end{vmatrix}$$

The filter is usually realized as a 2-D convolution of each pixel with this kernel (except for the boundaries). We implement a 1-D version on our architecture using the following algorithm:

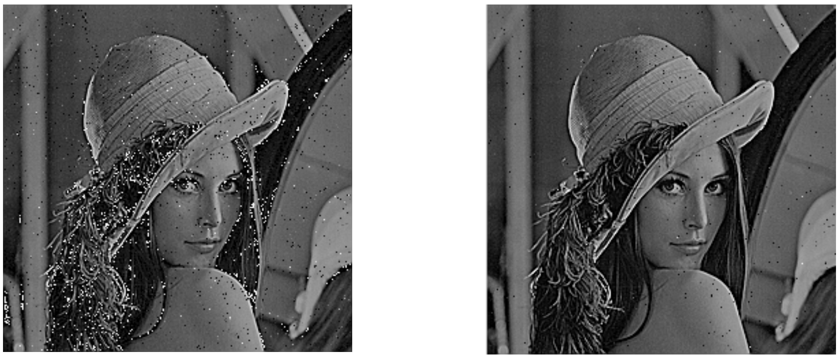
$$P(m,n) = \sum_{k=1}^{9} h(i,j) \cdot I(m+i, n+j)\,,\;\; i = \left\lceil \frac{k}{3} \right\rceil,\;\; j = k\%3+1$$

where *I(m, n)* and *P(m, n)* are the original and filtered pixel, at location (*m*, *n*), respectively.



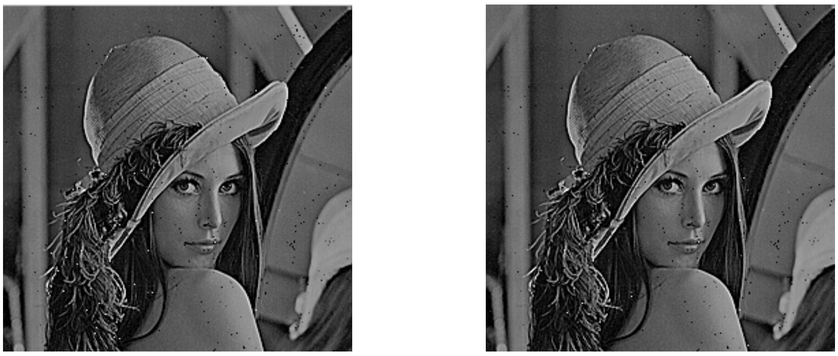
(a) Original

(b) Sharpened:
Energy=2.19 μJ
PSNR=23.9dB



(c) Unoptimized:
Energy=1.27 μJ
PSNR=19.6dB

(d) Dynamic width
adder: Energy=1.27 μJ
PSNR=23.0dB



(e) Reorder:
Energy=1.27 μJ
PSNR=23.0dB

(f) Combined:
Energy=1.27 μJ
PSNR=23.2dB

Figure 4.16: Image quality under different energy budgets.

This represents a 9th-order FIR filter, where the format for coefficients, input data, intermediate results and final output is Q4.12, Q8.0, Q12.12, and Q8.0. The full and reduced width adders in this case have 24 and 20 bits, respectively. Sample images after applying the sharpening filter with and without our error control are shown in Figure 4.16. From Figure 4.16 (c) we can see that, compared to a sharpened image at nominal voltage (Figure 4.16 (b)), voltage scaling without error control causes a lot of visually noticeable salt-and-pepper artifacts. By contrast, using our techniques (Figure 4.16 (f)), such noise is significantly reduced and resulting images exhibit good perceived quality at the same reduced energy.

4. 4 SUMMARY

This chapter presented techniques that enable architecture-level shaping of the quality-energy tradeoff under aggressively scale V_{DD} through controlled timing error acceptance. The implementation of these techniques is demonstrated on the general digital filtering architecture. Results show that significant energy saving can be achieved while maintaining a constant performance and good SNR/PSNR. This dissertation presented techniques that enable architecture-level shaping of the quality-energy tradeoff under aggressively scale V_{DD} through controlled timing error acceptance. The implementation of these techniques is demonstrated on the general digital filtering architecture. Results show that significant energy saving can be achieved while maintaining a constant performance and good SNR/PSNR.

Chapter 5: Conclusion

The limited battery life has become a bottleneck for embedded system. To prolong the battery life, we develop low-energy techniques in this dissertation. Since the typical building blocks in an embedded systems consists of a memory system and a DSP system. Our techniques are specifically designed for these two systems.

In memories systems, such as SRAM arrays, the increase of process variation significantly impacts circuit yield. Some patterns of variability are highly systematic, such as those in photolithography and chemical-mechanical polishing. Such variability has severe impact on the design of large SRAM arrays. This is because SRAM cells are typically sized to be of minimum possible area. And SRAM also needs to satisfy noise margin constraints over all cells in the array and these constraints determine both the minimum cell size and supply voltage. Increasing cell area and supply voltage can ensure that the noise margins are met. The requirement to meet noise margin constraints sets the limit on the smallest possible cell size and also on the minimum usable supply voltage V_{DD} for the array. Because threshold voltage variation impacting the cells in the SRAM array is independent, the margin specification needs to be met at very high sigma corners, five or six sigma, in order to reach acceptable yield, requiring significant cell upsizing and increased V_{DD}. SRAM cell area is an important metric of the success of technology scaling, and variability makes such traditional scaling hard to sustain. We introduce adaptive circuit scheme on optimization methods to allow low-power SRAM design under high-sigma constraints. The key contributions of this dissertation for the SRAM part is: (i) developing quantitative theoretical models for guiding the adaptive tuning for intra-array randomness; (ii) proposing a new architecture to reduce the overhead of high-sigma margin design on V_{DD} and cell area by employing an adaptive voltage scheme in the SRAM array. Result show that proposed technique can achieve significant power savings in both active and sleep mode, meanwhile the technique can also reduce the SRAM array size.

In embedded DSP systems, quality is set by a signal-to-noise ratio (SNR) floor. Conventional digital design strategies guarantee timing correctness of all operations, which leaves large quality margins in practical systems and sacrifices energy efficiency. In this dissertation, we present techniques to significantly improve energy efficiency by shaping the quality-energy tradeoff achievable via V_{DD} scaling. In an unoptimized design, such scaling leads to rapid loss of quality due to the onset of timing errors. We proposed techniques that modify the behavior of the early and worst timing error offenders to allow for larger V_{DD} reduction. The key contribution for the DSP part is: (i) developing both design-time and run-time techniques to reduce the occurrence of early timing errors with large impact on quality; (ii) using control and data flow analysis to disallow errors that are spread and get amplified as they propagate through the algorithm; (iii) developing low-cost post-processing techniques to improve signal quality. The experimental results on image processing and digital filter applications demonstrate that significant energy saving can be achieve using our techniques.

Bibliography/References

[1] K. Kuhn, C. Kenyon, A. Kornfeld, M. Liu, A. Maheshwari, W. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, and K. Zawadzki, "Managing process variation in intel's 45nm CMOS technology," Intel Technology Journal, vol. 12, no. 2, pp. 93–109, June 2008.

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variation and impact on circuits and microarchitecture," in Proceedings of the Design Automation Conference, vol. 40, June 2003, pp. 338–342.

[3] K. Zhang, U. Bhattacharya, Z. Chen, F. Hamzaoglu, D. Murray, N. Vallepalli, Y. Wang, B. Zheng, and M. Bohr, "A 3-Ghz 70-Mb SRAM in 65-nm CMOS technology with integrated column-based dynamic power supply," IEEE Journal of Solid-State Circuits, vol. 41, no. 1, pp. 146–151, January 2006.

[4] K. Kanda, T. Miyazaki, M. Sik, H. Kawaguchi, and T. Sakurai, "Two orders of magnitude leakage power reduction of low voltage SRAM's by row-by-row dynamic V_{DD} control(RRDV) scheme," in IEEE International ASIC/SOC Conference, vol. 15, September 2002, pp. 381–385.

[5] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Reduction of parametric failure in sub-100-nm SRAM array using body bias," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 1, pp. 174–183, January 2008.

[6] B. Mohammad, S. Bijansky, A. Aziz, and J. Abraham, "Adaptive SRAM memory for low power and high yield, in IEEE International Conference on Computer Design, October 2008, pp. 176–181.

[7] J. Wang, A. Singhee, R. Rutenbar, and B. Calhoun, "Statistical modeling for the minimum standby supply voltage of a full SRAM array," in European Solid State Conference, September 2007, pp. 400–403.

[8] M. Sniedovich, "Dynamic programming : foundations and principles", Boca Raton : Chapman & Hall/CRC Press, 2010.

[9] S. H. Nawab, A. V. Oppenheim, A. P. Chandrakasan, J. M. Winograd, and J. T. Ludwig, "Approximate signal processing," VLSI Signal Processing, vol. 15, pp. 177-200, 1997.

[10] J. T. Ludwig, S. H. Nawab, and A. P. Chandrakasan, "Low-power digital filtering using approximate processing," IEEE Journal of Solid-State Circuits, pp. 395-400, 1996.

[11] A. Sinha and A. P. Chandrakasan. "Energy efficient filtering using adaptive precision and variable voltage," ASIC SOC Conference, pp. 327-331, 1999.

[12] R. Hedge and N. R. Shanbhag, "Soft digital signal processing," TVLSIS, pp. 379-391, 2000.

[13] L. Wang and N. R. Shanbhag, "Low-power filtering via adaptive error cancellation," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 51, no. 2, pp. 575-583, 2003.

[14] T. Xanthopoulos and A. Chandrakasan, "A low-power DCT core using adaptive bitwidth and arithmetic activity exploiting signal correlations and quantization," J. Solid-State Circuits, vol. 35, no. 5, pp. 740-750, 2000.

[15] F. Kurdahi, A. Eltawil, K. Yi, S. Cheng, and A. Khajeh, "Low power multimedia system design by aggressive voltage scaling," IEEE Transactions on Very Large Scale Integration Systems, vol. 18, no. 5, pp. 852-856, 2010.

[16] J. Park, S. Kwon, and K. Roy, "Low power reconfigurable DCT design based on sharing multiplication," ICASSP, pp. III-3116-III-3119, 2002.

[17] G. Karakonstantis, D. Mohapatra, and K. Roy, "System level DSP synthesis using voltage overscaling, unequal error protection and adaptive quality tuning," SIPS, 2009.

[18] N. Banerjee, G. Karakonstantis, and K. Roy, "Process variation tolerant low power DCT architecture," DATE, pp. 1-6, 2007.

[19] R. L. Swenson and K. R. Dimond, "A hardware FPGA implementation of 2-D median filter using a novel rank adjustment technique," Intl. Conference on Image Processing and Its Application, vol. 1, pp. 103-106, 1999.

[20] Y. Lian and Y. J. Yu. "Low-power digital filter design techniques and their applications," Circuits, Systems, and Signal Processing, 29(2):1-5, 2010.

[21] H. Choi and W. P. Burleson. "Search-based wordlength optimization for VLSI/DSP synthesis," VLSI Signal Processing, pp. 198-207, 1994.

[22] O. Chen, R. Shen, and S. Wang. "A low-power adder operating on effective dynamic data ranges," IEEE Transaction on very large scale integration (VLSI) systems, vol. 10, no. 4, pp.435-453, 2002.

[23] T. W. Parks and C. S. Burrus. "Digital filter design," Wiley, New York, 1987.

[24] K. Johansson, O. Gustafsson, L. DeBrunner and L. Wanhammar, "Minimum adder depth multiple constant multiplication algorithm for low power FIR filters", ISCAS 2011, pp.1439-1442.

[25] N. Banerjee, J. H. Choi, and K. Roy. "A process variation aware low power synthesis methodology for fixed-point FIR filters," ISLPED, pp. 147-152, 2007.

[26] B. Shim, S. R. Sridhara, and N. R. Shanbhag. "Reliable low-power digital signal processing via reduced precision redundancy," IEEE Transaction on very large scale integration (VLSI) systems, vol. 12, no. 5, pp. 497-510, 2004.

[27] A. Dembo and O. Zeitouni, "Large Deviations Techniques and Applications," Springer-Verlag, 1998.

[28] M. Yamaoka, N. Maeda, Y. Shimazaki, and K. Osada, "65nm low-power high-density SRAM operable at 1.0v under 3s systematic variation using separate V_{TH} monitoring and body bias for NMOS and PMOS," in IEEE International Solid-State Circuits Conference, February 2008, pp. 384-385.

[29] E. Seevinck, F. J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," in IEEE J. Solid-state Circuits, October 1987, pp. 748–754.

[30] A. K. Singh, K. He, C. Caramanis, and M. Orshansky, "Mitigation of intra-array SRAM variability using adaptive voltage architecture," in IEEE/ACM International Conference in Computer-Aided Design, November 2009, pp. 637–644.

[31] O. Hirabayashi, A. Kawasumi, and A. Suzuki, "A process-variation tolerant dual-power-supply SRAM with 0.179µm2 cell in 40nm CMOS using level-programmable wordline driver," in IEEE International Solid-State Circuits Conference, February 2009, pp. 458-459.

[32] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distribution for images," IEEE transaction on image processing, vol. 9, no. 10, pp. 1661-1666, 2000.

[33] E. Arias-Castro and D. L. Donoho, "Does median filtering truly preserve edges better than linear filtering?" The Annals of Statistics, vol. 37, no. 3, pp. 1172-1209, 2009.

[34] S. Uramoto, Y. Inoue, A. Takabatake, J. Takeda, and Y. Yamashita, "A 100-Mhz 2-D discrete cosine transform core processor," JSSC, vol. 27, pp. 492-499, 1992.

[35] USC SIPI image database. [Online]. Available: http://sipi.usc.edu/database/

[36] R. K. Richards, Arithmetic Operations in Digital Computers. New York: Van Nostrand, 1955.

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, 2004.

[38] S. Gazor and W. Zhang. "Speech probability distribution," IEEE Signal Processing Letters, vol. 10, no. 7, pp. 390-399, 2003.

[39] P. Mermelstein. "Evaluation of a segmental SNR measure as an indicator of the quality of ADPCM coded speech," J. Acoust. Soc. Am., vol. 66, no. 8, pp. 1664-1667, 1979.

Vita

Ku He was born in February 10th, 1982, he is the son of Hualin He and Fuyu Liu. He received this B.S. in Electrical Engineering from Tsinghua University, China, 2004. After obtaining his M.S. in Electrical Engineering from Tsinghua University, China, in 2007, he went on to study in University of Texas at Austin, and entered the Ph.D program in the Department of Electrical and Computer Engineering in the same year.

Permanent address (email): #102, Entrance 4, Building 34,
Liu Jiang Zao Zhi Chang (Liujiang Paper Mill),
Liubei district, Liuzhou, Guangxi, 545011
(kuhe@utexas.edu)

This dissertation was typed by the author.