# VaROT: METHODOLOGY FOR VARIATION-TOLERANT DSP HARDWARE DESIGN USING POST-SILICON TRUNCATION OF OPERAND WIDTH

by

KEERTHI KUNAPARAJU

Submitted in partial fulfillment of the requirements

For the degree of Master of Science

Thesis Adviser: Dr. Swarup Bhunia

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

May, 2011

**CASE WESTERN RESERVE UNIVERSITY**

**SCHOOL OF GRADUATE STUDIES**

We hereby approve the thesis/dissertation of Keerthi Kunaparaju

candidate for the Master of Science degree *.

Signed

Swarup Bhunia
(Chair of the comittee)

Christous Papachristou

Francis "Frank" Merat

Date: 11/11/2010

*We also certify that written approval has been obtained for any proprietary material contained therein.

To my family and friends.

TABLE OF CONTENTS

Page

# LIST OF TABLES

# LIST OF FIGURES

Figure Page

VaROT: Methodology for Variation-Tolerant DSP Hardware Design using Post-Silicon Truncation of Operand Width

Abstract
by
KEERTHI KUNAPARAJU

Dramatic improvements in semiconductor integrated circuit technology presently make it possible to integrate millions of transistors, onto a single semiconductor IC. These improvements in integration densities have been driven by aggressive scaling of technology, which has led to both increasing density and computing power.

On the flip side, constant drive towards ever decreasing feature sizes has led to a significant increase in manufacturing cost. One of the main causes of this increase in manufacturing cost is a significant decrease in manufacturing yield due to manufacturing losses. These manufacturing losses are due to increasing process parameter variations that CMOS devices face at nanometer scale.

Increasing device parameter variations in nanometer CMOS technologies cause large spread in circuit parameters such as delay and power, leading to parametric yield loss. For Digital Signal Processing (DSP) hardware, variations in circuit parameters can significantly affect the Quality of Service (QoS). Post-silicon calibration and repair have emerged as an effective solution to maintain QoS in DSP chips under large process-induced parameter variations. However, existing calibration and repair approaches rely on adaptation of circuit operating parameters such as voltage, frequency or body bias and typically incur large delay or power overhead. In this thesis, a novel low-overhead approach of healing DSP chips by commensurately truncating the operand width based on their process shifts is presented. The proposed approach exploits the fact that critical timing paths in typical DSP datapaths originate from the least significant bits. Hence, truncation of these bits, by setting them at constant values, can effectively reduce the delay of a unit, thereby avoiding delay failures. Efficient choice of truncation bits and values can minimize the impact of truncation

on QoS. Appropriate design time modifications including insertion of low-overhead truncation circuit and skewing the path delay distribution through gate sizing to maximize the delay improvement with truncation are presented.

The proposed technique is applied on two common DSP circuits, namely Discrete Cosine Transform (DCT) and Finite Impulse Response (FIR). Simulation results show significant decrease in critical path delay with the truncation of least significant input bits and a graceful degradation in the QoS. Also there is a large improvement in manufacturing yield (41.6%) with up to 5X savings in power compared to existing approaches like voltage scaling and body biasing.

# 1. INTRODUCTION

Moore's Law has been serving the semiconductor industry marvelously since its evolution but as we continue to move in sub nanometer regime we need to deal with the darker side of the Moore's law [1]. Although the semiconductor industry has seen nearly exponential increase in the device integration density and performance, it now faces some major road blocks due to intrinsic physical limitation of the devices. One of the major barriers that the CMOS devices face at nanometer scale is increasing process parameter variations. IC manufacturing processes typically involve complex physical and chemical interactions. Because it is impossible to control these interactions at scaled technology nodes, process parameters associated with these manufacturing processes tend to fluctuate around their nominal values, causing "*process parameter variations*". Such variations can significantly reduce manufacturing yield. As technology scales, the importance of understanding the effects of process variations on circuit performance is increasing further. Let's first study the impact of scaling on manufacturing yield.

## 1.1 Impact of Scaling on Manufacturing Yield

With continuous scaling, though growing transistor numbers on a chip are significantly improving chip performance and reducing manufacturing costs, the design (including the design tools and mask making) costs are significantly increasing as shown in Fig. 1.1.

Thus improving the yield of chips is no longer just the responsibility of manufacturers and has shifted to the shoulders of the circuit designers. It must be considered in the early stage of a circuit design. Traditionally, the yield of chips depends on how well the design follows the design rules but at 65 and 45 nm CMOS process nodes,

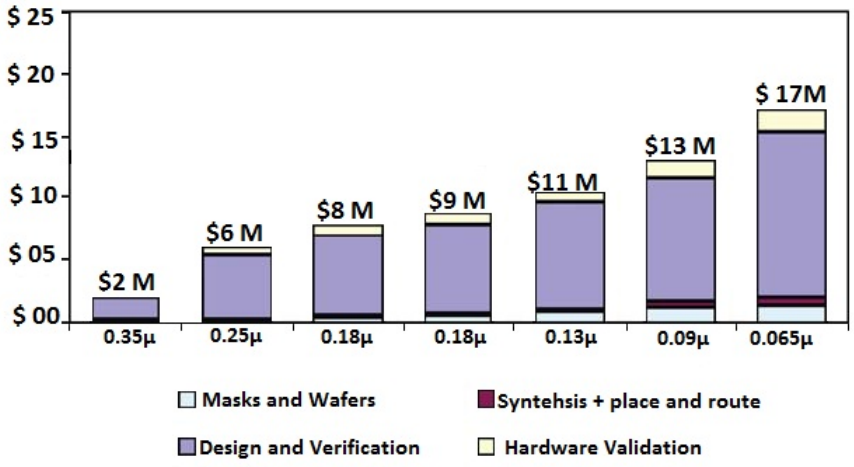



Fig. 1.1. Design cost is increasing as processes enter nano CMOS [2]

design rules become more complex. Also, in nano-scale technology nodes, yield issues cannot be fully resolved by using optical proximity correction (OPC) and phase-shift mask (PSM) techniques. It requires a more accurate prediction to issues that could happen in manufacturing. Furthermore, as transistors continue to scale down, they do not behave in a way as they do in micro-scale process nodes. The strong variations of process parameters and fabrication environment parameters, together with random effects, become key factors to be considered in yield enhancement.

The importance of accurately estimating the impact of parameter fluctuations on circuit performance is directly related to a company's overall revenue. An overestimation increases the design complexity, possibly leading to an increase in design time, an increase in die size, rejection of otherwise good designs, and even missed market windows [3]. Conversely, an underestimation can compromise the product's performance and overall yield as well as increase the silicon debug time [3]. In summary, overestimating fluctuations impacts the design effort, and underestimating fluctuations impacts the manufacturing effort.

Let's now discuss about the three main sources of these variations.

## 1.2 Sources of Parameter Variations

Sources of variations can be:

- Process Variation (P)
- Supply Voltage (V)
- Temperature (T)

Systematic and random variations in process, supply voltage and temperature (P, V, T) are posing a major challenge to the future high performance microprocessor design ( [4], [5]). Technology scaling beyond 90nm is causing higher levels of device parameter variations, which are changing the design problem from deterministic to probabilistic. The demand for low power typically causes supply voltage scaling and hence making voltage variations a significant part of the overall challenge. Finally the quest for growth in operating frequency has manifested in significantly high junction temperature and within die temperature variation. Let's now discuss the impact of P, V, T variations on circuits.

### 1.2.1 Process Variations

Fig. 1.2 plots distributions of frequency and standby leakage current($I_{sb}$) of microprocessors in a wafer. The spread in frequency and leakage distributions is due to variation in transistor parameters, causing about 20x variation in chip leakage and 30% variation in chip frequency.

This variation in frequency has introduced the concept of frequency binning. Notice that the highest frequency chips have a wide distribution of leakage, and for a given leakage, there is a wide distribution in the frequency of the chips. The highest frequency chips with more $I_{sb}$, and lowest frequency chips with less $I_{sb}$ may have to be discarded affecting the yield.

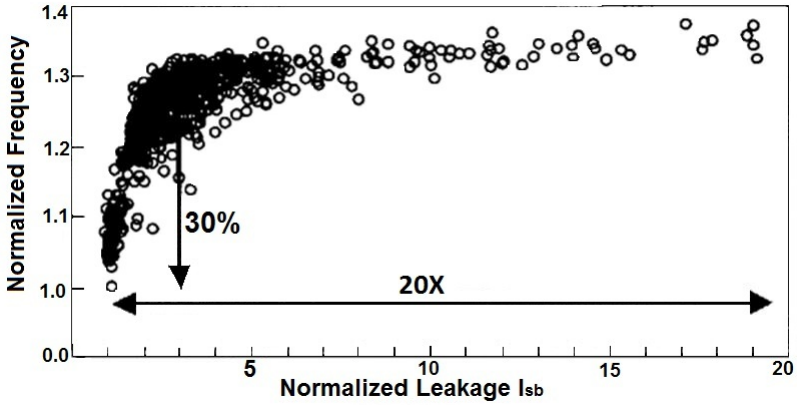



Fig. 1.2. Leakage and Frequency variations [5]

### 1.2.2 Supply Voltage Variations

Variations in switching activity across the die and diversity of the type of logic, result in uneven power dissipation across the die. This variation results in uneven supply voltage distribution and temperature hot spots, across a die, causing transistor sub threshold leakage variation across the die.

### 1.2.3 Temperature Variations

Temperature variation is unavoidable in the everyday operation of a design. Within die temperature fluctuations have existed as a major performance and packaging challenge for many years. Both the device and interconnect performance have temperature dependence, with higher temperature causing performance degradation. Additionally temperature variation across communicating blocks on the same chip may cause performance mismatches, which may lead to logic or functional failures.

The net consequence of the P, V, T variation manifests itself on the chip frequency variation. The frequency distribution across ICs may have serious cost implications with it. Low performance chips need to discarded which in turn effects the yield and hence the cost.

Lets now discuss more in detail about process variations since in this work we propose a healing technique to compensate for process variations.

## 1.3 Types of Process Variations

Process variations can be classified as follows:

1. **Inter-die variations or die-to-die variations**: Inter-chip variations are variations that occur from one die to another, meaning that the same device on a chip has different features among different dies of one wafer, from wafer to wafer and from wafer lot to wafer lot.

2. **Intra-die or within-die variations**: Intra-die variations are the variations in device features that are present within a single chip, meaning that a device feature varies between different locations on the same die. Intra-chip variations exhibit spatial correlations and structural correlations. Intra-die variations can be further classified into random and systematic variations.

    - **Random Variations**: Random variations are caused by atomic-level differences between devices even though the devices may have identical layout geometry and environment. Some examples of these variations are dopant profiles, film thickness variation, and line-edge roughness. Variation in the threshold voltage $V_{th}$ is observed due to placement of random number of dopants in the channel during manufacturing steps of implant and annealing processes. This phenomenon is called Random Dopant Fluctuation. The other random variations considered are gate oxide thickness $T_{ox}$ and effective channel length $L_e$.

    - **Systematic Variations**: Systematic variations imply spatial correlation between devices. The electrical parameters of the device vary depending on the placement of a device relative to its neighbors. These variations have a well-understood relationship between design instances or layouts and the

resulting electrical parameter values. They are predictable so that they can be modeled and the values are maintained across all corners/distributions.

Lets now discuss the impact of die-to-die and within die variations on the maximum frequency distribution of the ICs.

## 1.4 $F_{max}$ Distribution Model

An overview of the maximum operating frequency ($F_{max}$) distribution model is presented in Fig. 1.3.

The individual contributions of die-to-die (D2D) and within-die (WID) fluctuations on the nominal critical path delay distribution are shown. The simulated critical path delay distribution resulting from WID fluctuations is the distribution of one specific critical path. Then for a number of independent critical paths for the chip, the within-die maximum critical path delay distribution for the entire chip is shown. Since D2D fluctuations affect each critical path on a chip equally, the D2D maximum critical path delay distribution is represented by the D2D nominal critical path delay distribution. Next, the two maximum critical path delay distributions resulting from D2D and WID fluctuations are statistically combined and then mapped to a frequency distribution. Thus the ($F_{max}$) distribution model is obtained.

Functionality, $F_{max}$, and power consumption of individual dies are influenced by both die-to-die (D2D) and within-die (WID) variation components. The impact of within-die variation, which causes differences in path delays fabricated on the same die, is heavily influenced by circuit optimization decisions such as transistor sizing, threshold voltage assignment, and number of critical paths in the design. Fig. 1.4 shows that as the number of independent critical paths increases, the mean of the maximum critical path delay (which corresponds to the $F_{max}$) increases as well. The magnitude of the WID variation also depends on critical path depth, where paths with fewer logic stages experience less averaging of random variations resulting in larger variability. Due to increasing complexity and performance requirements for

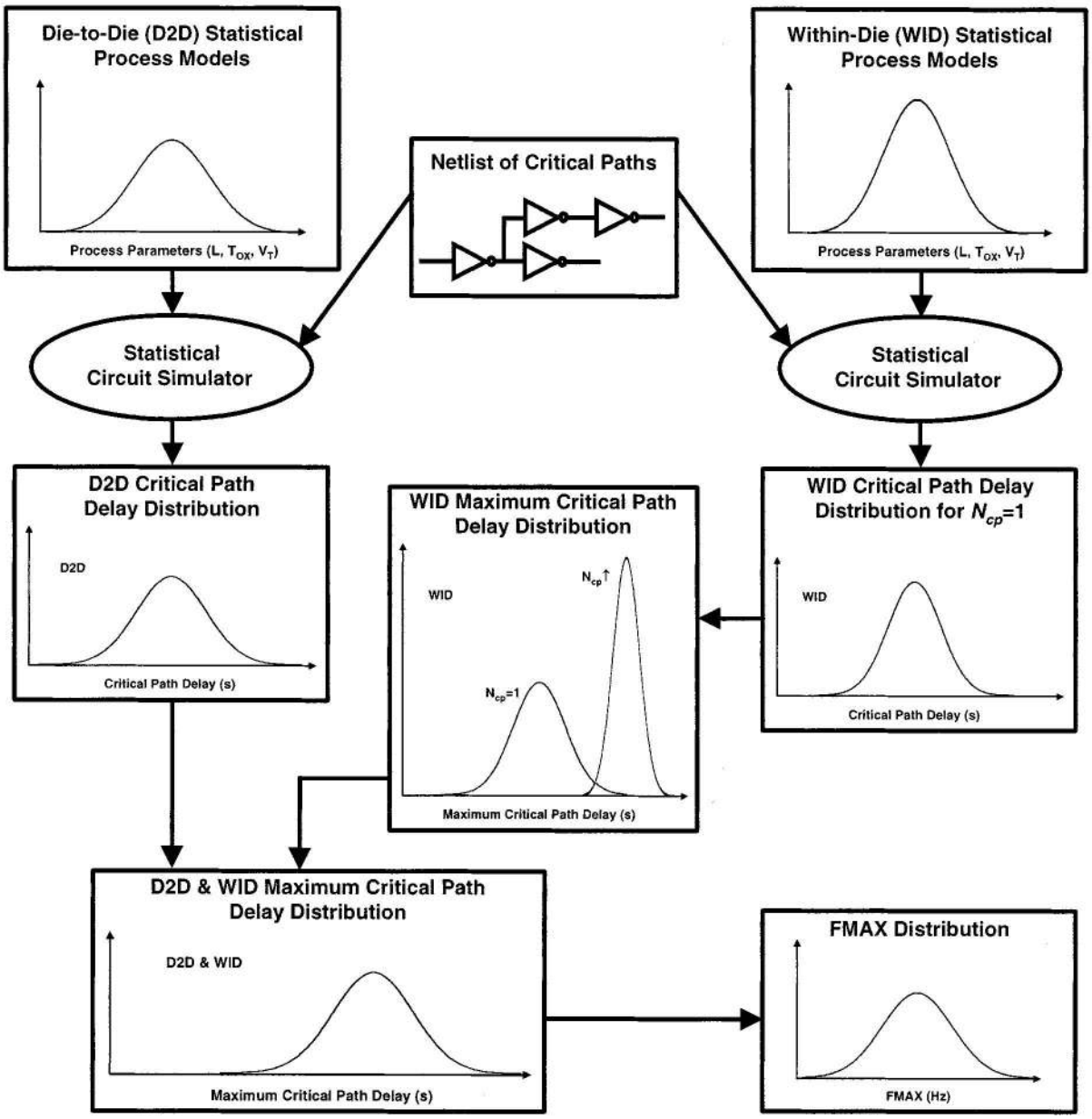



Fig. 1.3. Flowchart for describing the $F_{max}$ distribution. $N_{cp}$ is the number of independent critical paths on a chip [6]

microprocessor designs, the number of critical paths increases with each generation while the logic depth typically decreases. Both trends worsen the impact of within-die variations.

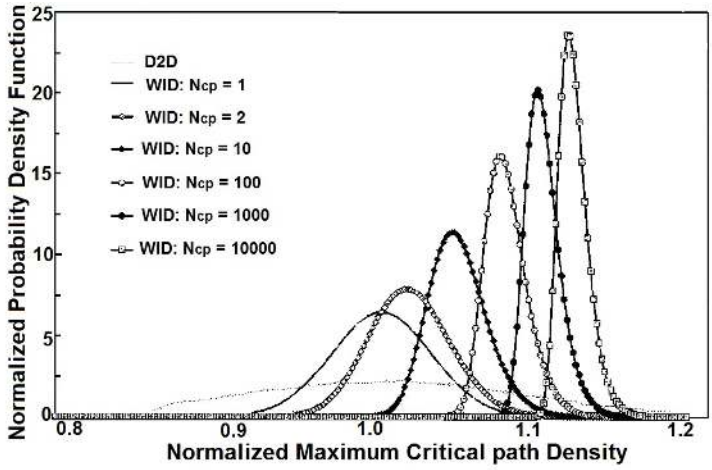



Fig. 1.4. Impact of within-die variations on product performance, as a function of the number of statistically independent critical paths ($N_{cp}$) [6]

It is to be noted that the variance of the combined distribution is determined mainly by the die-to-die component, while the mean $F_{max}$ is primarily a function of within-die variations. These variations combine to affect the frequency and power distributions for the fabricated dies, and therefore both are important to consider when optimizing a design for performance, power, and revenue. Typically, these variations are handled by a combination of design margining (which can lead to a worst-case design which operates inefficiently under normal conditions) and frequency binning (which impacts revenue and yield). When frequency binning is done, dies with a slow $F_{max}$ are either discarded or sold at a reduced price, while dies with excessive leakage or total power will violate the system power specification and must be discarded. Thus, the amount of process variations directly impacts the revenue.

## 1.5 Contribution to this Thesis

As increasing process variations in nanoscale technology nodes lead to large spread in major circuit parameters such as delay and power consumption which significantly

affects the manufacturing yield, several techniques have been proposed to deal with the impact of process variations to counteract the reduction in parametric yield. Conventional worst-case design approaches lead to overly pessimistic results in terms of area and power under large variations. On the other hand, statistical design approaches [7] [8], try to mitigate the overhead of worst-case design by optimizing a design for a target yield under statistical distribution of circuit parameters. However, statistical design represents a trade-off between target yield and design overhead in terms of area and power. With increasing parameter variations, effectiveness of statistical approaches is expected to reduce significantly. As an alternative or complementary to conservative and statistical design approaches, designers resort to two major design techniques to ensure high yield under parameter variations at low design overhead: 1) *variation-tolerant design approaches* [9] [10], where circuits are designed to account for process variations during run time such that the performance of the chips will not be affected. But this might lead to increase in design complexity and also incur more area and power overhead 2) *post-silicon calibration and repair*, where parameter shift is detected and compensated after manufacturing by changing operating parameters such as supply voltage, frequency or body bias [11] [12] [13]. But scaling up the supply voltage results in high power consumption due to quadratic dependence of dynamic power of a circuit on operating voltage and application of body bias results in increase in short channel effects which lead to increase in leakage current.

Digital Signal Processing (DSP) hardware blocks are used in numerous embedded and mobile applications. With increasing parameter variations in nanoscale technologies, these computational blocks become increasingly vulnerable to variation-induced delay failures. These failures can significantly affect the Quality of Service (QoS) for a DSP circuit, e.g. degradation in output image quality in a image encoding block, leading to degradation in parametric yield. Existing statistical design solutions or post-silicon adaptation of circuit operating parameters typically incur large area or power overhead in order to maintain QoS. Since these DSP blocks are often used in

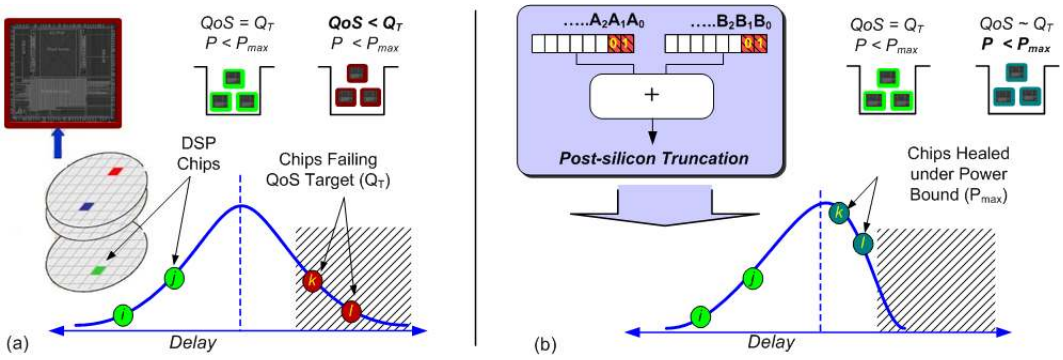



Fig. 1.5. Healing of digital signal processing chips failing QoS target using the proposed post-fabrication operand width truncation approach: a) binning of chips before healing; b) post-silicon operand truncation and binning after healing.

power-constrained applications, it is important to develop yield improvement techniques with minimal impact on power. In this work, we propose *VaROT* - a Variation Resilience through Operand Truncation approach targeting yield improvement in DSP hardware. VaROT provides a low-overhead approach for post-silicon healing of delay failures to restore system performance under large die-to-die or within-die parameter variations in nanoscale process technologies. Fig. 1.5 illustrates that post-fabrication healing of chips failing QoS target ($Q_T$) under power bound ($P_{max}$) leads to improvement in parametric yield. The proposed approach exploits the fact that in typical DSP datapath modules (such as adder, multiplier, multiply-and-accumulate units), critical timing paths originate from the least significant bits (LSBs) and they can be shortened by truncating the LSBs - i.e. setting constant values e.g. "0" to these bits. Consequently, truncation of operand width in these data paths post-manufacturing can be used to tolerate delay failures. Truncating the input bits, however, affects the output quality. However, we note that in case of common DSP computations (such as filtering, Fourier transform, color interpolation, motion estimation), truncating the least significant input bits in most datapath elements lead to minimal loss in output QoS [9] [14]. Besides, one can choose the optimal combination of constant values that can be assigned to the truncated bits to further reduce the QoS impact.

Note that, given a DSP datapath, the effect of truncation on delay reduction can be maximized by using a constrained design optimization step, e.g. gate sizing, which ensures that the critical paths originate from the input bits, which have little impact on QoS. During design of a DSP system, the truncation hardware can be inserted to select datapaths, which are vulnerable to delay failures under variations. After manufacturing, the delay shift in each chip is measured and appropriate numbers of input bits in the dataptaths equipped with truncation circuit are truncated to the predetermined truncation values based on the sensed process corner of a chip. The truncation configuration are stored in a tiny non-volatile configuration memory in the chip. Unlike the existing post-silicon repair solutions e.g. voltage or frequency scaling, simulation results show that such healing procedure avoids large impact on power dissipation, die area and throughput.

In the past, bit-width adaptation - both static and dynamic - has been used to reduce energy of computation in DSP circuits. The static approaches [15] [16] aim at choosing area or power-optimal bit-width for each datapath in a DSP circuit during design. The dynamic approaches [17] [18], on the other hand, performs bit-width adaptation at run time to trade-off energy versus accuracy (or QoS) or reduce energy on specific input data pattern. None of these approaches, however, address compensation of process-induced spread in QoS in DSP chips. The novelty of this technique lies in applying post-manufacturing bit-width truncation based on process shift in a chip to compensate for quality loss. The truncation is applied to select QoS failing chips to improve the parametric yield. Unlike the finite word length approach in [19], the paths in the design are skewed such that critical paths originate from LSBs. The proposed technique can be combined with an on-chip process-voltage-temperature (PVT) monitor such as ring-oscillator based sensor in order to apply dynamic truncation to prevent run-time failures under temporal fluctuations. Such an approach prevents designing a DSP circuit for worst-case temperature and aging conditions. Moreover, the proposed approach can be combined with dynamic voltage

scaling and ABB approaches to reduce power consumption at run time with graceful degradation in QoS. In particular, this work makes the following contributions:

1. It presents a design methodology for variation-resilient DSP circuits such that delay failures due to process variations can be corrected using a post-silicon repair mechanism that employs truncation of operand width. It evaluates the effect of truncation on output quality and investigates the optimal choice of number of operand bits and the assigned values to those bits that minimizes the impact on output quality.

2. It presents a design optimization step using gate sizing that maximizes the delay reduction due to truncation. It also presents a low-overhead implementation of the truncation hardware for varying number of bits with varying combination of values based on process corner of a chip.

3. It considers two case studies, namely Discrete Cosine Transform (DCT) and Finite Impulse Response (FIR), which are commonly-used DSP applications to verify the effectiveness of the proposed approach in improving parametric yield. Unlike the existing approaches, it does not cause large increase in circuit power and area to compensate process-induced delay variations. In fact, it can result in small *power saving* due to reduction in switching activity in the truncated bits. Simulation results show that VaROT can provide large improvement in yield with minimal impact on QoS with significant power savings compared to existing healing techniques.

4. It discusses possible extension of the approach for tolerating temporal delay variations as well as achieving graceful degradation in QoS with dynamic voltage scaling.

The rest of the thesis is organized as follows. Chapter 2 provides background of related work. The motivation behind the approach and description of the proposed healing methodology are provided in Chapter 3. Chapter 4 and Chapter 5 provides

case studies on two common DSP applications, namely Discrete Cosine Transform (DCT) and Finite Impulse Response (FIR) circuits respectively. Chapter 7 provides the possible extension of the proposed healing approach and conclusion.

# 2. BACKGROUND

Circuits are subject to inherent variation and uncertainty in both their operating and processing conditions. So, designers must verify circuit functionality and performance over ranges of those conditions by introducing manufacturing conditions into circuit simulation through the models of circuit elements.

Conventionally, designers have done this by verifying circuit functionality and performance under extreme process and operating conditions, reasoning that a circuit that functions and performs correctly at the extremes should function or perform correctly at the intermediate conditions. With properly selected conditions, verification by simulation under extreme conditions, called *Worst Case Circuit Modeling*, can insure that circuits function and perform to specification across a wide range of operating and process conditions. However, worst case circuit modeling has two serious limitations. First, it carries with it substantial risks of over or under-estimating variations and their impacts on design [20]. Over-estimation makes it harder to design circuits meeting their specifications, possibly leading to increased design effort, increased die size, and missed market windows. Conversely, under-estimation can lead to manufacturability problems and yield loss. Second, worst case methods are limited in their ability to provide with quantitative information about the robustness and sensitivities of their designs.

Also, the drive to higher performance and density in integrated circuit products has narrowed the "design window" by increasing variation in semiconductor manufacturing processes while decreasing the tolerance of integrated circuits to variation. The"guardband" inevitably associated with worst case methods further narrows, or even eliminates, this window, compromising the ability of designers to find viable solutions. This has forced designers to look for alternatives to worst case methods capable of more accurately representing the impact of variation. Broadly, three classes

of techniques are proposed to ensure/enhance yield under variations while incurring minimal impact on design overhead:

1. **Statistical design approach**, where a circuit parameter (e.g. delay or leakage) is modeled as a statistical distribution (e.g. Gaussian) and the circuit is designed to meet a constraint on yield (or to maximize it) with respect to a target value of the parameter. Gate sizing or dual-$V_{th}$ assignments are typically used to vary circuit delay or leakage distribution.

2. **Post-Silicon compensation and correction**, where parameter shift is detected and compensated corrected after manufacturing by changing operating parameters such as supply voltage, frequency or body bias. Post-silicon techniques, such as adaptive body-biasing or frequency scaling can affect both power and performance of a circuit.

3. **Variation avoidance**, where a given circuit is synthesized in such a way that the delay failures due to variations can be identified in run time and avoided by adaptively switching to lets say two-cycle operations.

Lets discuss the above three techniques in detail

## 2.1 Statistical design and Modeling

Statistical circuit modeling technique overcomes the limitations of worst case modeling, in part, by not requiring assumptions about the combination of conditions corresponding to worst case. Instead, they model uncertainty in processing and operating conditions statistically and use circuit simulators to propagate this uncertainty through to circuit responses. Statistical design methodology that either ensures or enhances certain parametric yield (e.g. with respect to delay) under specific design constraint (e.g. on area or power) has been addressed by many researchers ( [20], [21], [22], [7]). Gate-level sizing and or $V_{th}$ assignment have been primarily used as a tool to modulate the circuit delay distribution for yield improvement or

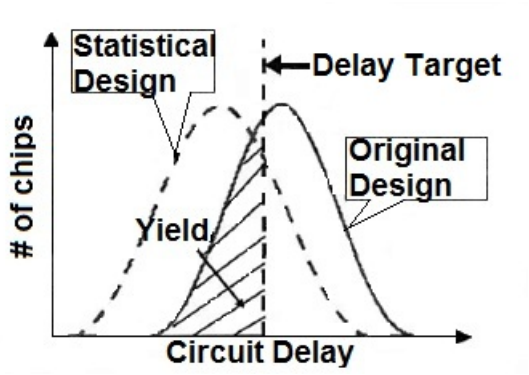



Fig. 2.1. Delay distribution of a circuit before and after statistical design. The yield, computed as the probability of meeting a delay target, is considered as an objective or constraint of the design optimization process [8]

yield-constrained area/power minimization. In a statistical design, circuit delay is typically modeled as a Gaussian distribution as shown in Fig. 2.1; timing yield is modeled as the probability to meet the target delay (shaded region in Fig. 2.1); and the delay distribution is changed in a way to improve yield during the design optimization process.

But there are some barriers which impede the statistical circuit modeling and optimization.

With continuous scaling within die variations pose significant challenges to statistical circuit as they must be modeled by at least one random variable for every circuit element. This results in unmanageable complexity. Design uncertainty poses another serious challenge and easily exceed the inherent fluctuations of manufacturing processes. Also introducing the impacts of variation and uncertainty greatly increases the computational and engineering resources required for design, diminishing the benefits of statistical techniques. Profitability of a design is conventionally equated with yield. However, large spread in the frequency distribution due to increasing uncertainties has led to the concept of speed-binning to improve the design profit. Since high-frequency ICs correspond to higher price points compared to their low-frequency counter parts,

maintaining yield at a target circuit delay (i.e. frequency) under statistical delay distribution does not ensure high profit.

Thus several significant barriers impede their wide-spread use in industrial circuit design.

## 2.2 Post-Silicon Process Compensation/Correction

In this category of solutions, process variation is detected using on-chip process sensor or in the manufacturing test. The deviation of circuit parameters due to variation is compensated/corrected by applying appropriate techniques after manufacturing by changing operating parameters such as supply voltage, frequency or body bias. We discuss below some post silicon process compensation techniques.

### 2.2.1 RAZOR Technique

One such technique, called RAZOR [23], a new approach to Dynamic Voltage Supply(DVS), is based on dynamic detection and correction of speed path failures in digital designs. Its key idea is to tune the supply voltage by monitoring the error rate during operation. Because this error detection provides in-place monitoring of the actual circuit delay, it accounts for both global and local delay variations and therefore eliminates the need for voltage margins to ensure always-correct circuit operation in traditional designs. Razor relies on a combination of architectural and circuit-level techniques for efficient error detection and correction of delay path failures. Fig. 2.2 illustrates the concept for a pipeline stage. A so-called shadow latch, controlled by a delayed clock, augments each flipflop in the design. In a given clock cycle, if the combinational logic, stage L1, meets the setup time for the main flip-flop for the clock's rising edge, then both the main flip-flop and the shadow latch will latch the correct data. In this case, the error signal at the XOR gate's output remains low, leaving the pipeline's operation unaltered. If combinational logic L1 doesn't complete its computation in time, the main flip-flop will latch an incorrect value, while the

shadow latch will latch the late-arriving correct value. The error signal would then go high, prompting restoration of the correct value from the shadow latch into the main flip-flop, and the correct value becomes available to stage L2. To guarantee that the shadow latch will always latch the input data correctly, designers constrain the allowable operating voltage so that under worst-case conditions the logic delay doesn't exceed the shadow latch's setup time.

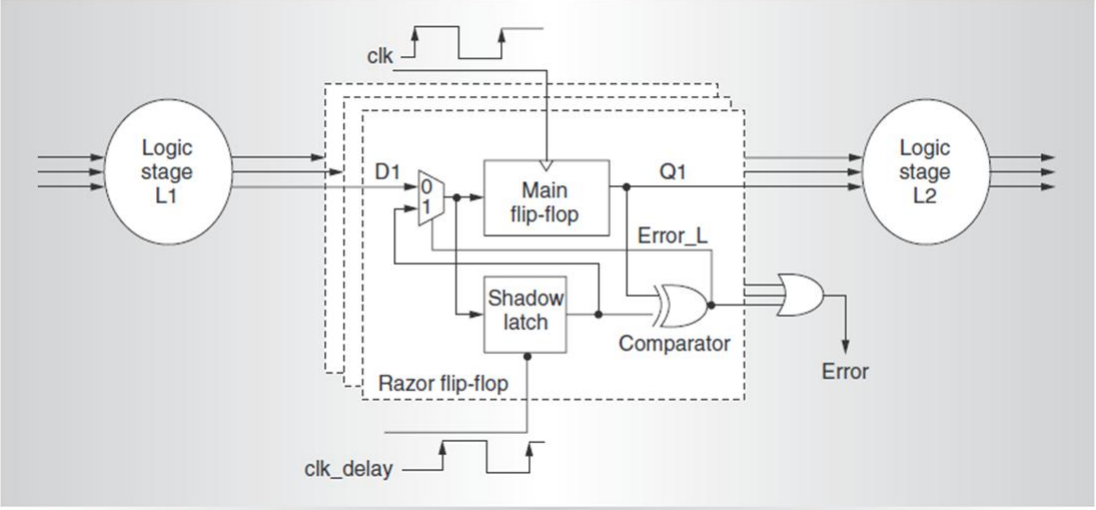



Fig. 2.2. Pipeline stage augmented with Razor latches and control lines [23]

Razor detects an error in any stage, the entire pipeline stalls for one cycle by gating the next global clock edge. The additional clock period lets every stage recompute its result using the Razor shadow latch as input. Consequently, the correct value from the Razor shadow latch will replace any previously forwarded errant values. Because all stages reevaluate their result with the Razor shadow latch input, a single cycle can tolerate any number of errors, guaranteeing forward progress. If all stages produce an error in each cycle, the pipeline will continue to run, but at half the normal speed. Thus Razor is applicable to systems that allow dynamic voltage/frequency scaling for power reduction. This results in small performance loss and also error detection and recovery logic itself can consume a significant portion of energy savings obtained by over-scaling.

### 2.2.2 Adaptive Body Bias (ABB) Technique

A post-Si healing technique based on Adaptive Body Bias(ABB)and it's impacts are in proposed in [11] and [12] respectively. Bidirectional ABB allows each die on a wafer to have the optimum threshold voltage which maximizes the die frequency subject to the power constraint. The threshold voltage of each die is controlled not only by process but also by the application of the appropriate amount of Forward Body Bias(FBB) or Reverser Body Bias(RBB). Dies which are too slow receive FBB, increasing the die frequency as well as the die leakage. Dies that violate the leakage constraint receive RBB, reducing the leakage as well as the frequency. In this way, the combined effect of parameter variations is compensated by changing the $V_t$ of the devices. The pMOS and nMOS body bias voltages which result in this optimum threshold voltage is applied by an external source or by an on-chip body bias generator. Similarly, the control circuitry which generally includes a phase detector that determines the optimum body bias voltage is implemented off-chip or integrated in the die.

To compensate for inter die variations a chip is divided into circuit blocks which are identified as equally critical. One of these circuit blocks contains a replica critical path and a phase detector which communicates with a central body bias generator. Based on the frequency of this critical path, the central bias generator determines the body bias which must be applied to meet a target frequency and this body bias is applied to all circuit blocks on the die. But this method doesn't take intra die variations into account.

To compensate for intra-die variations an improved ABB technique is used determining the best pMOS/nMOS bias combination per die. The technique is similar to technique used to compensate for inter die variations except that each circuit block requires its own phase detector structure and the central bias generator takes into account each phase detector output when determining the appropriate body bias voltage. This is accomplished by combining the individual phase detector signals with

an "OR" structure so that the bias generator counter is updated if any circuit block does not meet the target frequency.

However ABB needs separate power distribution network, additional routing resources with shielding. Also RBB might not work very well in the future because it increases within die Vt variation. Thus ABB might lead to considerable area and power overhead.

### 2.2.3 Adaptive Supply Voltage (ASV) and Adaptive Body Bias (ABB) Technique

In another technique [13], for post silicon process compensation, the frequency and leakage of the processor dies can be controlled through adaptive change of supply voltage $V_{CC}$. Because both switching and leakage components of power consumption have a super linear relation to $V_{CC}$, changing the supply voltage has a significant impact on the total power consumption. Using adaptive $V_{CC}$ in conjunction with adaptive $V_{BS}$ (adaptive $V_{CC}$ + $V_{BS}$) is more effective than using either of them individually. Adaptive $V_{CC}$ + $V_{BS}$ recovers the dies above the active power limit by: 1) first lowering $V_{CC}$ and natural operating frequency together to bring the sum total of their switching and leakage powers well below the active power limit and 2) then applying FBB to speed them up and move them to the highest frequency bin allowed by the active power limit. As a result, more dies use lower values than adaptive $V_{CC}$ (Fig. 2.3). In addition, more dies use FBB, instead of RBB, compared to adaptive $V_{BS}$. Since the effectiveness of RBB for leakage power reduction diminishes with technology scaling, adaptive $V_{CC}$ + $V_{BS}$ will be more effective in future technology generations than adaptive $V_{BS}$ alone.

### 2.2.4 Voltage over Scaling (VoS)

In this technique [24] supply voltage is reduced beyond that limited by the critical path delay of a given DSP architecture. The degradation in the output quality

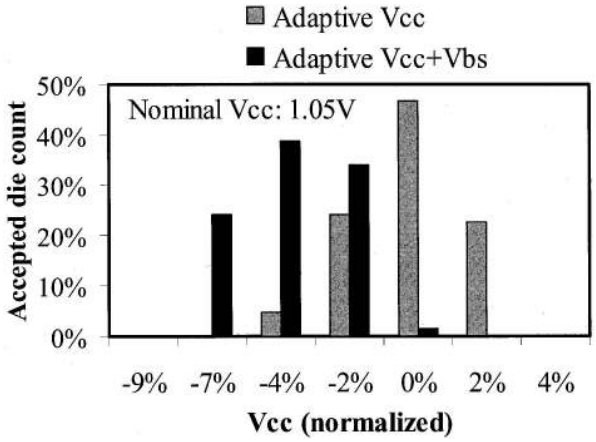



Fig. 2.3. Distribution of $V_{CC}$ values for adaptive $V_{CC}$ and adaptive $V_{CC} + V_{BS}$ [13]

is restored via algorithmic noise tolerance, where the signal statistics are exploited to develop low complexity error-control schemes. But VoS in its current form cannot dynamically switch between different voltages depending the amount of error. As voltage is set to a pre-defined value, during process variations where the delays of the paths vary randomly in different circuits there will be unpredictable energy consumption and quality deterioration.

## 2.3 Avoidance of Variation-Induced Failure

In this category the circuits are designed such that they sense process variations during runtime and switch to the appropriate operations as discussed below.

### 2.3.1 CRitical path ISolation for Timing Adaptiveness (CRISTA) Technique

The proposed technique [25], based on the concept of critical path isolation, makes a circuit amenable to aggressive voltage scaling, while being robust to parametric failures. This is accomplished by a synthesis technique that 1) isolates and predicts the set of possible paths that may become critical under process variations, 2) ensures

they are activated rarely, and 3) tolerates any delay failures in the set of critical paths by adaptively operating in two-cycles (assuming all standard operations are single cycle). This allows us to operate the synthesized circuit at reduced supply voltage while achieving the required yield. The delay margin between critical and non-critical blocks helps to avoid delay failure and achieve voltage scaling. The notion of critical path isolation indicates confinement of critical paths of a synthesized design to a known logic block or cofactor. This is accomplished by partitioning a circuit into multiple cofactors using Shannon decomposition and then using gate-sizing to create timing margin between cofactors. Any delay errors (that may occur under a single cycle operation) are predicted dynamically by decoding a small set of inputs and are adaptively avoided with two cycle operations. It thus comes with both performance and area overhead.

### 2.3.2 Process Variation Tolerant Low Power DCT Architecture

An other process variation tolerant low power design for DCT architecture has been proposed in [9]. This technique explores the fact that not all intermediate computations are equally important in a DCT system to obtain "good" image quality with Peak Signal to Noise Ratio(PSNR) $>$ 30 dB. Based on this fact a DCT architecture has been proposed where the signal paths that are less contributive to PSNR improvement are designed to be longer than the paths that are more contributive to PSNR improvement. First the computational paths that are vital in maintaining the high image quality are identified and then an algorithm/architecture is developed to make more important computations (in terms of image quality) to have shorter paths than less important ones. This architecture is then utilized to make any path-delay errors predictable under a single scaled supply voltage and process parameter variations, and to tolerate delay failures in such paths with minimal PSNR degradation of image.

### 2.3.3 A Process Variation Aware Low Power Synthesis Methodology for Fixed-point FIR filters

In this work [10] a novel FIR filter synthesis technique is proposed to tolerate process variations and voltage scaling with a graceful degradation in the filter response. This technique exploits the fact that all the filter coefficients are not equally important to obtain a reasonably accurate filter response. The technique implements a Level Constrained Common Subexpression Elimination (LCCSE) algorithm, where the number of adder levels required to compute each of the coefficient outputs is constrained. Tighter constraints are specified on the important coefficients such that the later computation steps compute only less important coefficient outputs. Thus in case of delay variations due to voltage scaling and/or process variations, only the less important outputs are affected, resulting in graceful degradation of filter quality.

However the above two techniques also incur significant power and area overhead. Also since changes to the design have to be made to compensate for process variations the design complexity increases.

# 3. MOTIVATION AND METHODOLOGY

Process imperfections during manufacturing of ICs introduce variations in the circuit parameters, particularly in the threshold voltage ($V_{th}$) ( [4], [5], [6] ). Threshold voltage is a strong determinant of circuit speed i.e. the maximum delay at which the circuit operates. Increase in threshold voltage due to process variations increase the delay inside the circuit which can lead to delay failures, where all the output bits are not computed correctly within the clock constraint. Usually the ICs which do not meet the clock constraint are discarded post-manufacturing. To increase the manufacturing yield, people tend to opt for worst-case design or statistical design where process variations are accounted for during the design process as discussed in chapter 2and the ICs are designed with tight delay constraints. Some other variation induced delay avoidance healing techniques have also been proposed during design time to account for process variations but all these techniques result in large area and power overhead as seen in chapter 2. On the other hand, people can go for nominal design and use post-manufacturing healing techniques as discussed in chapter 2, by frequency scaling, supply voltage scaling or application of proper body bias to reduce the delay of ICs which fail to meet the delay constraint. Frequency scaling degrades the performance. Scaling up the supply voltage though reduces the delay it results in increase in the power consumption of the IC which is not desired. Application of body bias reduces the delay but results in increase in area and power overhead. Also the effectiveness of these techniques is reduced at scaled technologies.

Thus we started investigating new healing techniques and the main objective was to find a technique which would not result in large area and power overhead. In our attempt to find a new healing technique we exploited the fact that certain DSP applications can tolerate some error in their outputs as long as the error is within an acceptable margin, determined by the Quality of Service (QoS) of the application.

This is possible if the failing bits are not the most significant bits (MSBs) of the outputs and they can be computed correctly within the clock constraint if they do not fall on the critical path. This motivated us to investigate techniques for healing the ICs without affecting the power by on-demand removal of the critical path instead of reducing its delay. In order to "remove" the critical path or prevent it from being excited, we truncate its input so that the critical delay shifts to the next-critical path. However, this delay reduction could be at the cost of incorrect computations at some of the output bits and this might affect the output quality. We then exploited the fact that in DSP circuits truncating least significant input bits have a minimum impact on the output quality and it is much better than the effect on output quality due to process variation induced delay failures, which typically affect the MSBs. So if the critical paths in the design are made to originate from least significant bits (LSBs) then the failing ICs are healed by means of truncating their LSBs. Thus we can heal the ICs failing to meet the delay target at the cost of graceful degradation in QoS.

Effect of truncation on critical path delay and output quality is discussed below through the simulation results of 2 bit adder and 8 bit adder respectively.

### 3.0.4 Effect of Truncation on Critical Path

We designed and synthesized a 2 bit adder using Synopsys Design Compiler and IBM 90nm standard cell library. The gate level circuit is shown in Fig. 3.1. The critical path is initially from A[0]→O[2] passing through the gates u1, u2 and u3 and the delay is 170ps. Now input A[0] is truncated to constant value 0 and hence there will be no transition at the outputs of the gates u1 and u2. So the critical path shifts to the next highest path starting from A[1]→O[1] passing through the gates u4, u5 and the delay is reduced to 140ps. Thus truncation reduces the circuit delay by shifting the critical path to the next highest path.

### 3.0.5 Effect of Truncation on Quality

Truncation of least significant input bits will ensure that most significant output bits are computed correctly within the clock period while failures are restricted to least significant output bits. This is explained for an 8 bit adder circuit using HSPICE simulations at the PTM45nm technology [26]. Consider an 8 bit adder with two inputs A and B designed with a clock period of 145ps. Now consider three cases as shown in Table 3.1. In Case 1 there are no process variations. At the first clock edge the initial input vectors for A and B are applied and the vectors are 10101011 and 01010100 respectively. At the next clock edge a transition at B[0] is applied, exciting the critical path B[0]→O[8]. The outputs are computed correctly meeting the target delay as shown in Table 3.1 under no process variations. The decimal output value is 256. Now consider Case 2 where 10% inter-die variations are introduced which lead to an increment in critical path delay as shown in Table 3.1 . This leads to a delay failure as the values of the output bits at the end of the specified clock bound are observed to have a decimal value of 128. This is because the most significant input bits A[7] and B[7] failed to latch the correct output bit within 145ps and retained their previous values, as shown in Table1. Thus from Case 1 and Case 2 it is observed that there is a significant impact on the quality due to process variations. Now consider Case 3 with process variations as in Case 2. In this case the least significant input bits i.e.

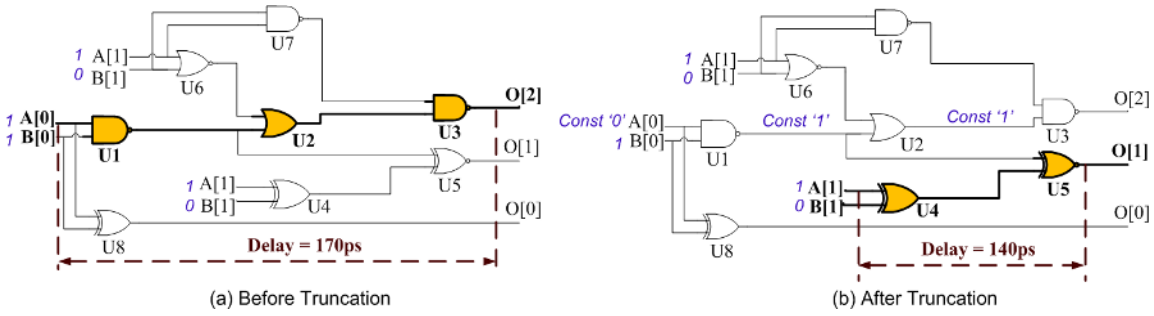



Fig. 3.1. Effect of truncation on critical path delay for 2 bit adder.

A[0] and B[0] are truncated to zero preventing the critical path from being excited. The output bits are again checked and it is found that all the other output bits are computed correctly as shown in Table 3.1. The output decimal value in this case with truncation is 255 which is much better than 128 in Case 2 with process variations and is very close to original value 256 in Case 1 without process variations. Thus truncation has a minimum impact on the output quality allowing the most significant output bits to compute correctly within the specified clock constraint.

Table 3.1
Effect of variation with and without truncation on 8 bit adder

| **Critical Path** | Without Process Variations **(Case 1)** | | With Process Variations **(Case 2)** | | With Process Variations and Truncation **(Case 3)** | |
|---|---|---|---|---|---|---|
| | Delay(ps) | Addition Computation | Delay(ps) | Addition Computation | Delay(ps) | Addition Computation |
| B[0]→O[8] | 145 | Carry 1111111 | 154 | Carry 1111111 | Truncated A[0], | Carry 0000000 |
| | | A 10101011 | | A 10101011 | B[0].So critical | A 10101010 |
| | | B 01010101 | | B 01010101 | path is not | B 01010100 |
| Output Binary Value | | 100000000 | | 10000000 | excited and delay | 011111110 |
| Output Decimal Value | | **256** | | **128**(at 145ps) | is less than 145 | **255** |

We synthesized an 8 bit adder and 8 bit multiplier circuit using Synopsys Design Compiler and IBM 90nm standard cell library and applied truncation to the least significant input bits. The truncation results for 8 bit adder and 8 bit multiplier are listed in Table 3.2 and Table 3.3. From the results it is observed that as number of bits to truncate increase critical path keeps on shifting to the next highest path thus reducing the overall delay of the circuit. So bit width truncation result in more and more percentage decrease in delay. In other words, to accommodate for higher increase in delay due to process variations we can truncate more number of input bits.

Coming to the power overhead by comparing the proposed technique with voltage scaling and body biasing it is observed that the healed ICs doesn't consume extra power for healing thus resulting in significant power savings. Thus there is no power

Table 3.2
Truncation results for 8 bit adder

| # of truncated bits | % decrease in delay |
|---|---|
| 2 | 12.3 |
| 3 | 25.8 |
| 4 | 39.3 |
| 5 | 51.6 |

Table 3.3
Truncation results for 8 bit multiplier

| # of truncated bits | % decrease in delay |
|---|---|
| 2 | 2.6 |
| 4 | 9.6 |
| 6 | 30.8 |
| 8 | 37.1 |

overhead unlike other techniques. Also as a byproduct there are power savings by truncating the input bits since we are preventing the transition/activity to happen at those nodes.

The methodology for this technique is discussed in detail in the following section.

## 3.1 Methodology

The truncation-based healing methodology can be applied to heal any DSP circuit where we can trade-off QoS to increase manufacturing yield with minimal power overhead. The two main features of this technique are:

1. Truncation of least significant input bits has less impact on QoS and allows the circuit to meet the delay target.

2. Truncation also helps in saving some switching power as it eliminates switching activity at the truncated nodes.

The proposed methodology is shown in Fig. 3.2 using a flow chart. It is primarily classified into two phases:

1. **Design Phase**
2. **Manufacturing Test Phase**

For a given design, the inputs are the target delay constraint ($D_{max}$) and a set S of different frequency bins into which the manufactured ICs can be classified, post-manufacturing. The output is healed chips meeting the target delay which are sorted into different bins based on their QoS. Note that the proposed technique can be used as alternative or complementary to existing design-time approaches [9] where the most significant coefficients are computed with higher delay margins. Next, we describe each of the steps in detail.

### 3.1.1 Design Phase

- **Perform Timing analysis and Sizing**

  To perform timing analysis and sizing we need a netlist with a desired clock constraint or target delay. As seen in the flowchart in Fig. 3.2 there is also another input S with a set of frequencies. This set S contains frequencies used in the frequency binning step during post manufacturing. Once all the ICs are manufactured, they are distributed in to different bins based on their frequencies and this process is called frequency binning.

  The approach is motivated by the fact that truncation of least significant input bits will have minimum impact on the output quality. So if the longest paths in the circuit originate from the least significant input bits then truncating them result in critical path shifting to the next highest path along with reduction in

delay which helps to compensate for increment in delay due to process variations. Static timing analysis is performed on a given netlist to find the delays of the paths originating from all the input bits . Tighter timing delay constraints are set on the paths originating from most significant input bits and relaxed timing constraints are set on the paths originating from least significant input bits. This makes the longest paths in the design originate from the least significant input bits. Skewing the design by applying constraints is to make sure that even after introduction of process variations post manufacturing, the critical paths originate from the least significant input bits. The paths originating from most significant input bits will not be longer than paths originating from least significant input bits though there will be a delay increment on all the paths due to process variations. There should also be more slack between the paths originating from most significant and least significant input bits so as to make sure that critical paths never originate from most significant input bits in the presence of process variations. Also it is desired to have more slack even between the longest paths originating from the least significant input bits so that truncation of each path lead to a maximum delay reduction thus meeting the target delay with a minimum effect on the output quality.

The tightest timing constraint met by the paths is achieved by starting with a relaxed delay constraint and then re-synthesize to check if the paths meet the assigned timing constraint. If the paths meet the constraint then it is made even tighter and this process is repeated until the paths meet the tightest timing constraint. Similarly to have more slack between the longest paths, constraints are initially set such that paths have less delay difference and if this difference is met after synthesizing the design, the difference is gradually increased. This process is repeated until the paths meeting the timing constraints with a longest possible delay difference is found. The path distribution should be skewed such that when a path with more delay is truncated we get more delay reduction.

Once the design is ready each frequency bin in the input set S is considered. The amount of delay to be tolerated by each bin has to be calculated. The number of input bits to truncate to account for that percentage increase in delay have to be found. For example lets say for a particular frequency bin the percentage increase in delay might be 2% which might require 2 bits of truncation and for an other frequency bin the percentage increase in delay might be 4% which might require 4 bits of truncation. Now different truncation values have to be applied to the input bits to find out a combination which has a minimal effect on the QoS. The details of how different values can be applied to the input bits are discussed in the following step.

- **Choice of number of truncation bits and their values**

  In this stage, truncation values are assigned to the input bits to truncate the paths and the impact on the output quality is seen by simulating the netlist. Either truncating to 0 or truncating to 1 might give us less impact on the output quality by meeting the required delay tolerance. For example, if two input bits have to be truncated to shift the critical path to the next highest path then combination of all truncation values for those input bits i.e. 00, 01, 10, 11 are applied and the optimal combination which has a minimal impact on the output quality, while meeting the required delay tolerance, is selected. For instance, let us consider that for a particular frequency bin, the desired delay tolerance is 5% and this can be achieved by truncation of 2 input bits. All possible truncation combinations have to be applied. Let us consider that combination '00' gives 7% delay tolerance with 2% quality loss, combination '11' gives 6% delay tolerance with 3% quality loss, combination '10' gives 5% delay tolerance with 4% quality loss and combination '01' gives 4% delay tolerance with 1% quality loss. The truncation combination '01' though has only 1% quality loss cannot be selected because the desired delay reduction is not achieved. Instead '00' is chosen as the best truncation combination as it has the least impact on the output quality

while meeting the required delay tolerance. Thus, the optimal truncation values are determined for each frequency bin. There is no use in determining truncation values for all the frequency bins. The designer has to determine an acceptable QoS margin and has to continuously check the QoS of each frequency bin. If the impact of truncation values on QoS of a particular frequency bin exceeds an acceptable QoS margin, then truncation has to be stopped and no more frequency bins will be considered. The QoS can be computed during the design phase by simulating the netlist and applying truncation to bits manually. If QoS of a particular frequency bin falls below the acceptable QoS margin then next frequency bin in set S has to be considered and again the process of applying truncation is repeated. The process is repeated until we find a frequency bin whose QoS after finding optimal truncation values exceed an acceptable QoS margin. Once we encounter that frequency bin, no more frequency bins will be considered from set S and truncation of further bits is stopped. Next we get to the implementation of truncation bits through a truncation circuit. The details of truncation circuit are discussed in the following section.

- **Choice of Truncation Circuit**

  Truncation Circuit is designed with a minimum overhead and it can be turned on to truncate different number of bits, which is nothing but dynamic truncation, with different stages of quality loss depending on the variation in the critical path delay due to process variations. To prevent the critical path excitation we can either truncate the inputs of the first level gates or the outputs of those gates. One obvious way is truncation can be achieved by passing the input bit of a circuit, which has to be truncated, to the 2 input NAND/NOR gate keeping the other input 0/1 so that there will not be any excitation at the output of the gate. But inserting a NAND/NOR gate at every input bit that has to be truncated result in area and also delay overhead due to insertion of an extra NAND/NOR gate. The other way is setting and resetting the flipflops,

providing inputs to the design, but this approach would require multiplexers. However, both schemes incur huge area and delay overhead. Thus both schemes result in huge area and delay overhead. So we decided to truncate the outputs of the gates. As truncation values are already determined, by using a single pull-down or pull-up transistor the output of the gate can be wired to 0/1. However, these pulldown/ pull-up transistors at the output of the gate can result in large leakage current when those transistors are on. So to avoid these leakage paths, the first-level gates can be supply gated when the pull down transistors are turned on and ground gated when the pull up transistors are turned on for truncating the gate outputs [27]. If a particular input bit going to any inverting gate has to be truncated to 1, $V_{DD}$ gating is applied at the output of that gate and a pull down transistor is used to force the output to GND as shown in Fig. 3.2. Similarly if a particular input bit going to any inverting gate has to be truncated to 0, GND gating is applied at the output of the gate and pull up transistor is used to force the output to $V_{DD}$ as shown in Fig. 3.2. Each input bit can be provided with both the provisions of truncating it to 0 and also 1. The gating control signals for all these transistors are supplied by a decoder which is inserted during the design phase. Each input combination of the decoder corresponds to single level of truncation. One input combination of the decoder corresponds to no truncation and is applied to the chips which already meet the delay constraint. For example if the decoder has 2 input bits then four input combinations are possible with one input combination corresponding to no truncation and other three input combinations corresponding to 2 bit, 3 bit, 4 bit truncations respectively.

A small non volatile memory is also provided for each IC which stores the input combination that has to be applied to the IC as soon as it starts running. This input combination will be fed to the memory after the IC is fabricated and tested to determine the level of truncation that has to be applied to get back that IC to the nominal delay. Thus the truncation circuit is turned on

after manufacturing the IC and through dynamic truncation i.e. by applying appropriate inputs to the decoder by fetching from the non volatile memory an IC is healed.

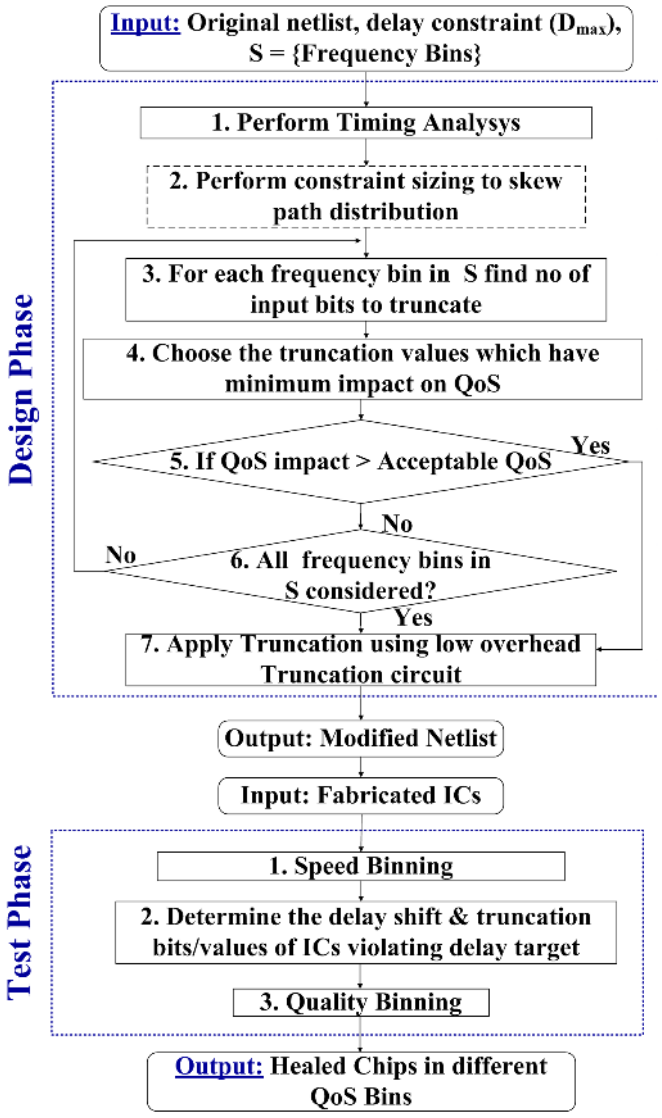



Fig. 3.2. Flow chart of the design and test methodology for the proposed truncation approach.

### 3.1.2 Manufacturing Test Phase

The final step during design phase is insertion of truncation circuit. After this step ICs will be manufactured. Process imperfections during manufacturing introduce variations and effect the delay inside the circuits and the delay follows a statistical distribution as discussed in chapter 1.

Post manufacturing, the ICs are subject to testing and speed binning [28] to estimate the amount of delay increment caused in different ICs due to process variations. Large spread in the frequency distribution due to increasing uncertainties has led to concept of speed binning and it is used during manufacturing test to qualitatively sort the working ICs based on their highest allowable frequency of operation. So ICs are tested and are sorted into different frequency bins. Now each frequency bin is considered and depending on the amount of delay exceeding the nominal delay, truncation has to be applied. To compensate for the delay increment, appropriate input combination of bits is then stored in the configuration ROM which provide inputs to the decoder. Different output combinations will turn on the respective amount of truncation required to heal that IC in each frequency bin.

However, it should be noted that after applying truncation the healed ICs now fall into nominal frequency bin but provide different QoS levels depending on the number of truncated bits. So in the final step as shown in step 3 in the manufacturing test phase of the flowchart in Fig. 3.2, quality binning is performed and the healed ICs are distributed in different bins based on the amount of quality degradation. Thus truncation heal the chips by making the ICs meet the timing constraint with a low impact on the quality and improves the overall yield.

It should be noted that truncation is not applied at the design time, since due to the nature of process variations, around 50% of the ICs will have nominal or better delays, hence these chips will not suffer from any delay failures and can be used as high-quality DSP chips. However, many of the chips which originally would have been discarded because of failure to meet the delay constraint, can now be used as nominal

performance chips with slight degradation in quality. Another use of the truncation is to compensate for aging-induced temporal variations which can cause delay failures in chips which were initially meeting the delay target. In order to compensate for the delay increase in these chips, the ICs need to be periodically tested and characterized. Any chip which can be healed by applying appropriate truncation or by changing the level of truncation need not be discarded, as long as the quality degradation can be tolerated.

We simulated the DCT architecture and applied the proposed technique by following the steps in the flowchart. From the results it is observed that there is not much area and power overhead when compared to the existing techniques. There is a significant increase in the manufacturing yield with almost 5X power savings compared to voltage scaling and body biasing. The details of the DCT architecture and implementation of the proposed technique on DCT are discussed in the following chapter.

# 4. IMPLEMENTATION OF *VAROT* ON DCT ARCHITECTURE

Transform coding constitutes an integral component of contemporary image/video processing applications. Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighboring pixels. Similarly in a video transmission system, adjacent pixels in consecutive frames show very high correlation. Consequently, these correlations can be exploited to predict the value of a pixel from its respective neighbors. A transformation is, therefore, defined to map this spatial (correlated) data into transformed (uncorrelated) coefficients. Clearly, the transformation should utilize the fact that the information content of an individual pixel is relatively small i.e., to a large extent visual contribution of a pixel can be predicted using its neighbors.

## 4.1 The Discrete Cosine Transform

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency.

### 4.1.1 The One-Dimensional DCT

The most common DCT definition of a 1-D sequence of length N is

$$c(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{\pi(2x+1)u}{2N}\right) \text{ for } u = 0, 1, 2 \cdots, N-1 \tag{4.1}$$

$$where\ \alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u{=}0 \\ \sqrt{\frac{2}{N}} & \text{for } u{\neq}0 \end{cases}$$

It is clear from equation 4.1 that for u = 0 the first transform coefficient is the average value of the sample sequence. In literature, this value is referred to as the DC Coefficient. All other transform coefficients are called the AC Coefficients.

### 4.1.2 The Two-Dimensional DCT

Our main objective is to study the efficacy of DCT on images since the impact of truncation is measured in terms of quality degradation in the image. This necessitates the extension of ideas presented in the last section to a two-dimensional space. The 2-D DCT is a direct extension of the 1-D case and is given by

$$c(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left(\frac{\pi(2x+1)u}{2N}\right) \cos\left(\frac{\pi(2y+1)v}{2N}\right) \qquad (4.2)$$
$$\text{for } u, v = 0, 1, 2 \cdots, N-1$$

$\alpha$(u) and $\alpha$(v) are defined in equation 4.1. The inverse transform is defined as

$$f(x,y) = \alpha(u)\alpha(v) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u,v) \cos\left(\frac{\pi(2x+1)u}{2N}\right) \cos\left(\frac{\pi(2y+1)v}{2N}\right) \qquad (4.3)$$
$$\text{for } x, y = 0, 1, 2 \cdots, N-1$$

Thus Discrete Cosine Transform (DCT) is an efficient way of transform coding. The use of cosine rather than sine functions is critical in these applications: for compression, it turns out that cosine functions are much more efficient (fewer are needed to approximate a typical signal), whereas for differential equations the cosines express a particular choice of boundary conditions. The DCT is fast. It can be quickly calculated and is best for images with smooth edges like photos with human subjects. The DCT coefficients are all real numbers unlike the Fourier Transform. The Inverse Discrete Cosine Transform (IDCT) can be used to retrieve the image from its transform representation.

### 4.1.3 Applications of DCT

The DCT is widely used in JPEG image compression, MJPEG, MPEG, DV and Theora video compression. Here 2 dimensional DCT is used. The following steps give a general overview of JPEG process

**JPEG**

1. The image is broken into 8x8 blocks of pixels.
2. Working from left to right top to bottom DCT is applied to each block.
3. Each block is compressed through quantization.
4. The array of compressed blocks that constitute the image are stored in a drastically reduced amount of space.
5. When desired the image is reconstructed through decompression, a process that uses inverse discrete cosine transform.

## 4.2 Implementation of the proposed technique

The proposed technique is implemented on DCT design intended to use for Image compression. However referring to standard JPEG standard compression steps we did not implement quantization step and so this is lossless image compression. 2d-DCT is applied on 256×256 and 512×512 standard images and then proposed technique is applied. To retrieve the image back Inverse Discrete Cosine Transform (IDCT) is performed on the 2d- DCT output matrix.

The design of a 2-D Discrete Cosine Transform was obtained from [29]. It takes as its input an 8×8 block of 10-bit pixels from an image and outputs sixty-four 12-bit DCT coefficients. The DCT architecture used in this work has 64 MAC units which compute each DCT coefficient in parallel. A MAC unit consists of a 24-bit multiplier followed by a 27-bit adder in different pipeline stages. As all MAC units

run in parallel the critical path inside the DCT circuit is effectively through a single MAC unit and since it is a pipelined circuit, the critical path is either through adder or multiplier. The DCT architecture used in this work is shown in Fig. 4.1.

The DCT design is synthesized using Synopsys Design Compiler and mapped to IBM 90nm standard cell library. The DCT design is synthesized with a clock constraint of 3.5ns. The target is to improve the manufacturing yield by healing the bins with input set S = 3.6ns, 3.71ns, 3.85ns, 3.95ns, 4.06ns, 4.16ns. By following the design flow as shown in Fig. 3.2, static timing analysis is performed on the gate-level netlist and the critical paths are found to be lying within the multiplier or adder block of each MAC unit. After identifying the longest paths coming through both adder and multiplier units we determined two sets of timing constraints where one set excites the longest paths only from the adder unit and the other set of excites the longest paths only from the multiplier unit of MAC unit.

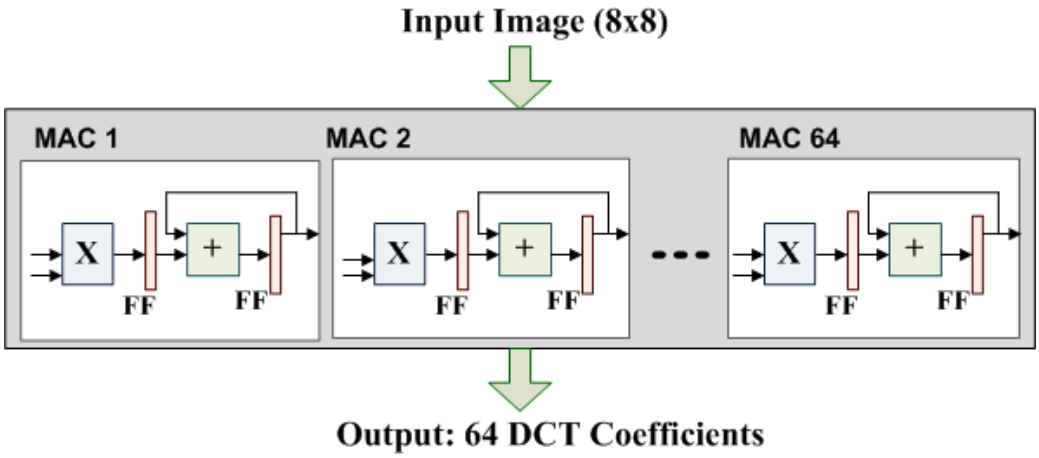



Fig. 4.1. DCT Architecture.

We also made sure that the paths originate from the least significant input bits and have a maximum possible slack between the longest paths by trying different timing constraints as explained in subsection 3.1.1. Lets first discuss about the design in which sizing constraints are applied such that critical paths originate from the least significant bits of the 27 bit adder. For each frequency bin in S, the amount of delay

increment is calculated. In this example, the first bin exceeds the nominal delay by 3% and it is observed from timing analysis that to compensate for 3% delay tolerance, three input bits (A[0], A[1], B[1] assuming A and B as inputs of the adder) have to be truncated so that the critical path shifts to the next highest path (originating from A[2]). Different combinations of truncation values are applied to these three input bits and the optimal truncation combination is found to be '000' which has a minimum impact on QoS. The truncation bits and their corresponding values are determined for all frequency bins in set S. Simultaneously for every level of truncation the impact on the QoS is computed for 256×256 and 512×512 standard test images by manually truncating the input bits in the netlist and simulating it to get the DCT output coefficients. We then took the DCT output matrix and applied Inverse Discrete Cosine Transform (IDCT) in matlab to retrieve the image. The output quality of an image is measured in terms of Peak Signal to Noise Ratio (PSNR). So PSNR is calculated to see the impact on the output quality due to truncation. Table 4.1 lists the percentage decrease in delay reduction for every truncation level, output quality measured in terms of PSNR on different images like Lena, Kiel, Barbara, Lake, Clown, Aero and House and percentage decrease in switching power. Thus Table 4.1 serves as a reference for the designer, after the chips are manufactured, to see how many input bits have to be truncated to compensate for a particular delay increment in order to heal the failing ICs.

Finally, the selected truncation values to the input bits are implemented using truncation circuit. A 3-to-8 decoder is used to apply different levels of truncation up to 9 bits and one of the input combinations is designed to cause no truncation. One of the decoder's output combinations performs truncation of input bits A[0], A[1] and B[1] by assigning truncation values 0, 0 and 0 respectively. The truncation circuit is shown in Fig. 4.2. The truncation of input bit A[2] to constant '0' is performed by applying ground-gating and pull-up transistor at the gate output. Similarly for an input bit A[4] whose value has to be truncated to one, supply-gating is applied and the output of that gate is forced to GND using a pull-down transistor as shown in Fig.

Table 4.1
Truncation Results for DCT Design when the critical paths are through the adder of MAC unit

| # of truncated bits | % of delay reduction | % of power reduction | PSNR(db) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Lena | Keil | Barbara | Lake | Clown | Aero | House |
| Original | — | — | 50.83 | 52.31 | 51.16 | 48.71 | 48.84 | 50.78 | 51.08 |
| 3(Trunc1) | 3.50 | 0.28 | 50.81 | 52.27 | 51.13 | 48.69 | 48.82 | 50.71 | 50.99 |
| 4(Trunc2) | 6.70 | 1.40 | 50.77 | 52.20 | 51.08 | 48.67 | 48.80 | 50.69 | 50.94 |
| 5(Trunc3) | 10.06 | 1.92 | 50.69 | 52.01 | 50.94 | 48.59 | 48.71 | 50.70 | 50.80 |
| 6(Trunc4) | 13.41 | 2.66 | 50.5 | 51.72 | 50.63 | 48.41 | 48.54 | 50.42 | 50.47 |
| 7(Trunc5) | 16.46 | 3.33 | 50.04 | 51.15 | 49.97 | 48.01 | 48.09 | 49.89 | 49.81 |
| 8(Trunc6) | 19.81 | 4.03 | 48.93 | 49.68 | 48.54 | 46.99 | 47.07 | 48.72 | 48.28 |
| 9(Trunc7) | 22.86 | 4.76 | 46.45 | 46.66 | 45.61 | 44.73 | 44.80 | 46.04 | 45.37 |

4.2. The gating, pull-up and pull-down transistors are controlled by the gating control (GC) signals whose value will be high only in the truncation mode. The gating control signals for achieving different levels of truncation are generated using a decoder which is driven by some configuration bits stored in a non-volatile memory. These bits can be fixed during manufacturing test phase for different ICs which will be truncated at different levels. One of the input combinations of the decoder corresponds to no truncation, and will be applied to the chips which already meet the target delay, post manufacturing. This scheme ensures minimal area overhead, caused by the 3:8 decoder circuit and 2 extra transistors for the first-level gates which need to be truncated. In other words we will need two extra transistors to truncate every input bit.

By estimating the delay and area, comparison is made between original architecture and the architecture with truncation circuit and an other low power DCT architecture [9] designed to account for process variations as shown in Table 4.2. The values show that critical path delay of the architecture with truncation circuit has only 1.2% overhead. The area overhead due to pull-up, pull-down and gating transistors and the decoder circuit is only 0.96%.

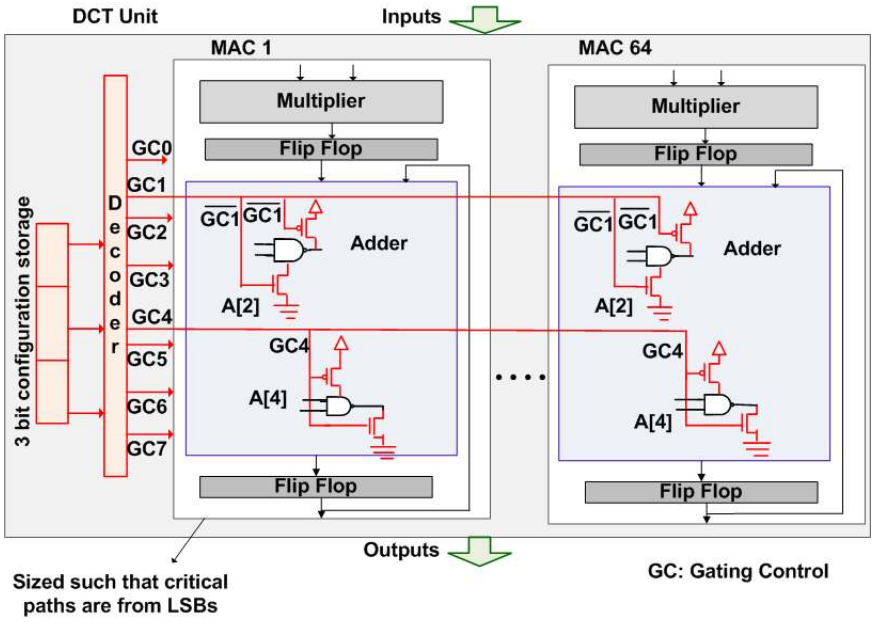



Fig. 4.2. DCT Hardware with Truncation Scheme

Table 4.2
Comparison of area and delay

| | Original | VaROT | Overhead | |
|---|---|---|---|---|
| | | | VaROT | Existing Method [9] |
| **Delay(ps)** | 407 | 412 | 1.20% | 11.56% |
| **Area($mm^2$)** | 9.92 | 10.02 | 0.96% | 12.23% |

Area overhead is calculated by taking in to account all 64 MAC units. For example in case of 3 bit truncation, as single bit would require 2 extra transistors, we need total of 6 transistors for truncating 3 bits and that accounts for 1 MAC unit. So for 64 MACs we will need 64x6 = 384 transistors. Also transistors for decoder should be accounted while calculating area overhead. Coming to power overhead, the truncation circuit is turned on as soon as the IC starts working. So the decoder, pull-up, pull-down and gating transistors will only switch once and hence there will be no dynamic power overhead. In fact the switching power decreases due to decrease in input switching activity as more input bits are truncated, as shown in Table 4.1. Also

due to first-level supply gating, there will be significant savings in the overall leakage power [27]. However, for a chip with no truncation applied, the power overhead is due to the extra leakage caused by the decoder and extra truncation transistors. It is to be noted that the healed ICs originally fall in high delay and hence, low-power process corners and after healing, they will have lower power compared to the nominal chips at high quality, even though their performance has been brought to nominal by healing.

The images in Fig. 4.3 show the impact of truncating different no of input bits on the output quality as listed in Table 4.1. We observe from Table 4.1 that truncating 9 bits is giving a delay reduction of 22% with a power reduction of 4.7% and the PSNR is still maintained at 46.45dB for Lena image, which though shows 9% quality decrement compared to the original PSNR value, there is no visual distortion at all in the output image.


Fig. 4.3. Images resulting after applying different levels of truncation when the critical paths originate through the adder of the MAC unit

Now consider the design where constraints are applied such that critical paths originate through the least significant input bits of the multiplier. Table 4.3 lists the percentage decrease in delay by truncating different number of input bits with truncation values as 0.

Table 4.3
Truncation Results for DCT Design when the critical paths are through the multiplier of MAC unit

| # of truncated bits | % of delay reduction | PSNR(db) | | | |
|---|---|---|---|---|---|
| | | Lena | Barbara | Clown | House |
| original | — | 50.83 | 51.16 | 48.84 | 51.08 |
| 2(Trunc1) | 1.20 | 48.99 | 49.22 | 47.55 | 49.03 |
| 4(Trunc2) | 6.46 | 45.46 | 45.64 | 44.58 | 45.55 |
| 6(Trunc3) | 7.82 | 40.11 | 40.43 | 39.66 | 40.75 |
| 8(Trunc4) | 14.28 | 34.26 | 34.75 | 33.89 | 35.66 |
| 10(Trunc5) | 17.00 | 28.38 | 28.98 | 28.42 | 30.02 |
| 12(Trunc6) | 28.20 | 22.35 | 22.86 | 22.44 | 23.90 |

The images in Fig. 4.4 show the impact of truncating input bits on the output quality of the Lena image when the critical paths originate through the multiplier.


Fig. 4.4. Images resulting after applying different levels of truncation when the critical paths originate through the multiplier of the MAC unit

It is observed from Table 4.3 and Fig. 4.4 that the impact on the output quality for different truncation levels is more for design with critical paths originating from the least significant input bits of the multiplier compared to the design with critical

paths originating from least significant bits of the adder. But the former design still gives a graceful degradation in quality. For example from Table 4.3 and Fig. 4.4, it is observed that by truncating 8 bits we get a delay reduction of 14.28% with PSNR still being at 34.26 db and there isn't much visual distortion in the image.

## 4.3 Effect of Process Variations

The effect of process variations without truncation and with truncation on the output image quality is shown in Fig. 4.5. We chose the design with critical paths originating from the least significant input bits of the adder since this design is giving better PSNR results.

Fig. 4.5(a) shows the output Lena image of the DCT architecture without process variations. Fig. 4.5(b) shows the images with 10%, 20% and 30% process variations. To compensate for increase in delay due to these process variations for the above three cases appropriate truncation has been applied with reference to Table 4.1 and Fig. 4.5(c) shows its impact on output quality of the image. From Case 1 and Case 2 it is observed that quality of the images is much better and even close to the original image after healing by means of truncating bits. But in Case 3, to compensate delay for extreme process variations no of bits to truncate increase and the effect on the output quality is significant.

## 4.4 Impact on Manufacturing Yield

As mentioned in Chapter 1, parameter variations are becoming an increasing concern with technology scaling. The variations increase the delay inside the circuits due to which computational paths fail to meet the target delay. Conventional wisdom dictates a healing approach of scaling up the supply voltage which makes circuits faster improving the delay and hence the manufacturing yield. But, such healing techniques come at a huge power overhead. Design techniques which tend to overcompensate for process variations also suffer from huge area and power overheads. By using our

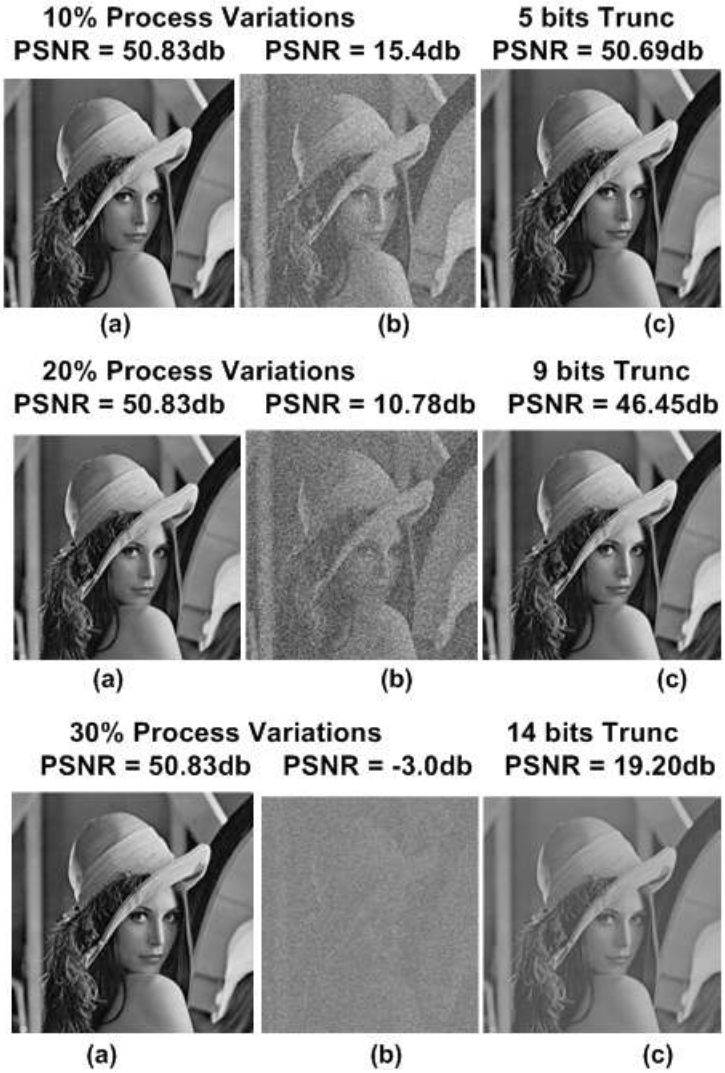



Fig. 4.5. a) Original Image; b) Output image with process variations; c) Output image with process variations and truncation

proposed technique, we heal an IC by determining the amount of delay increment post manufacturing and then apply appropriate truncation which does not increase the power consumption. Thus yield can be increased significantly with minimal area and power overhead at the cost of slight degradation in output QoS.

We performed Monte Carlo simulations in HSPICE for the DCT circuit using PTM 45 nm technology [14]. Monte Carlo simulations are performed for 10,000 process corners with inter-die variation of 20% and intra-die variation of 15%. The

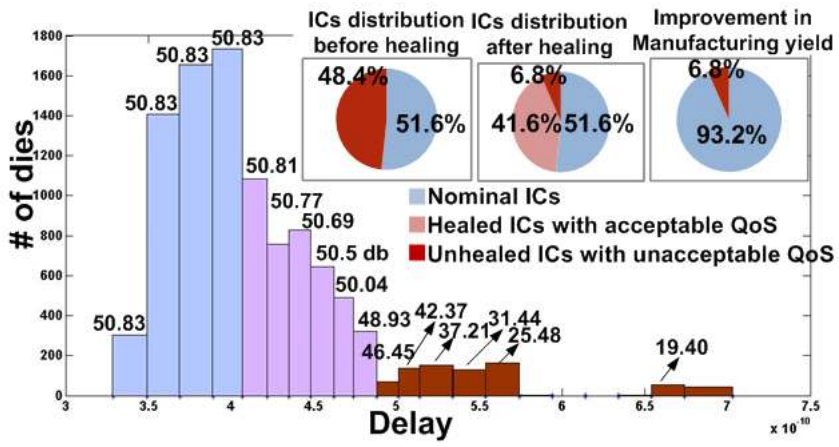



Fig. 4.6. Post-manufacturing delay distribution of 10,000 dies. By using truncation, chips in different frequency bins can be healed leading to increased yield. However, these healed ICs fall into degraded but acceptable QoS bins. The chips which cannot be healed within the acceptable QoS margin still lead to yield loss of 7%.

resulting delay distribution histogram is shown in Fig. 4.6. By defining the QoS margin of the healed ICs to be less than 3 dB from that of the nominal ICs QoS, truncation till 8 bits is performed and is found that yield is significantly improved from 51.6% without truncation to 93.2% after truncation. The corresponding quality bins are also shown in Fig. 4.6. Thus the healed ICs fall in the bins with different degrees of quality loss and depending on the customer requirement of acceptable QoS margin, the healed ICs can be salvaged.

## 4.5 Power Savings with VaROT

Next we compare the power savings achieved with our technique when compared to healing techniques such as supply voltage scaling and body biasing. Let us consider that a multimedia IC is designed with a target yield of 50% without considering any healing technique. After the ICs are manufactured, the designers try to improve the yield by 30% by compensating for the process variation-induced delay increments. Now we have three options for post-Si healing. Designer 1 decides to increase the

yield by increasing the supply voltage. This gives 30% yield improvement but at the cost of high power consumption. Designer 2 decides to increase the yield by applying Forward Body bias (FBB), which affects the threshold voltage of the transistors and bring the ICs within the nominal delay target. Designer 3 opts for truncation. This also results in 30% yield but truncation does not require any extra power consumption and in fact there is some reduction in switching power due to truncation of some input bits. Also it is known that high $V_{th}$ chips have low power. Process variations increase the threshold voltage of transistors and so ICs which need healing consume low power compared to the nominal ICs. This fact is exploited by the truncation technique to achieve low power healed ICs compared to nominal ICs where as other techniques require extra power consumption for healing. Thus we will have significant power savings by truncation. Though there is a little impact on the output quality, the designer depending on the demand for output quality can always limit the number of truncation bits.

We calculated the power savings by simulating the DCT design in HSPICE and applied voltage scaling and body biasing techniques. Different process corners which makes the IC's run at slower speeds are selected. The power for healing the ICs at those process corners using voltage scaling, body biasing and the proposed technique is measured. Table 4.4 lists the percentage increment in power consumption (compared to the nominal power) due to scaling up the $V_{dd}$ and body biasing to compensate for different delay increments under process variations to meet the target delay. The table also lists the percentage increase in power savings that can be achieved with our technique for the same improvement in yield over voltage scaling and body biasing techniques, the number of bits to be truncated to compensate for the delay and loss in QoS at every truncation level.

In case of body biasing we use Forward Body Bias(FBB) since it is the most effective way to reduce both active and leakage power and improve the performance of the circuit. In general the highest speed of the NMOS is when the bulk is connected to $V_{dd}$. This leads to decrease in the threshold voltage $V_t$ of the NMOS and increase

Table 4.4
Power Savings with VaROT

| % Delay increment | $V_{dd}$ Scaling (V) | Power $V_{dd}$ Scaling (mW) | Optimum body bias | | Power FBB (mW) | VaROT # of bits to truncate | Power VaROT (mW) | % increase in power ($V_{dd}$ Scaling) | % increase in power (FBB) |
|---|---|---|---|---|---|---|---|---|---|
| | | | $V_b$ pmos (V) | $V_b$ nmos (V) | | | | | |
| 0 | 1.0 | 19.22 | 1.0 | 0.0 | 19.22 | 0 | 19.22 | 0 | 0 |
| 3.22 | 1.11 | 26.55 | 0.6 | 0.2 | 20.90 | 3 | 17.53 | 51 | 19 |
| 6.52 | 1.13 | 26.21 | 0.5 | 0.5 | 21.83 | 4 | 16.06 | 63 | 35 |
| 9.95 | 1.19 | 28.39 | 0.3 | 51.72 | 22.39 | 5 | 14.68 | 93 | 52 |
| 12.60 | 1.24 | 31.61 | 0.25 | 51.15 | 23.45 | 6 | 13.69 | 131 | 71 |
| 15.57 | 1.29 | 34.46 | 0.22 | 49.68 | 27.25 | 7 | 12.73 | 171 | 114 |
| 19.67 | 1.38 | 41.58 | 0.20 | 46.66 | 33.26 | 8 | 11.52 | 261 | 185 |
| 22.82 | 1.44 | 47.65 | 0.19 | 46.66 | 37.96 | 9 | 10.66 | 347 | 256 |
| 23.95 | 1.46 | 50.00 | 0.18 | 46.66 | 46.10 | 10 | 10.35 | 382 | 344 |

in the speed. In the case of PMOS, when the bulk voltage is connected to ground the threshold voltage decreases and current through the PMOS device increases. Here we applied optimum body bias voltages ($V_b$) to PMOS and NMOS transistors to obtain a combination of lesser power and delay.

Thus the table shows large power savings(almost 5X) can be achieved through VaROT when compared to voltage scaling and FBB techniques at the cost of slight loss in QoS with significant increase in manufacturing yield with a minimum area and power overhead.

The proposed technique is also implemented on Finite Impulse Response(FIR) circuit. The details and the results are discussed in the following chapter.

# 5. IMPLEMENTATION OF *VAROT* ON FIR

In this chapter we present the implementation of the proposed methodology on Finite Impulse Response(FIR).

An FIR filter is used to perform filtering and it can have different shapes in its transfer function, which is the Fourier transform of its impulse response. An FIR filter is usually implemented by using building blocks like delay elements, multipliers, and adders to create the filter's output.

The difference equation that defines the output of an FIR filter in terms of its input is:

$$y[n] = coef(0)x(n) + coef(1)x(n-1) + \cdots\cdots + coef(N)x(n-N) \tag{5.1}$$

where x[n] is the input signal, y[n] is the output signal, coef(0) till coef(N) are the filter coefficients, also known as tap weights, and N is the filter order - an Nth-order filter has (N + 1) terms on the right-hand side. These are commonly referred to as taps (the number of inputs).

The transpose form of an FIR filter is generally used since it has a shorter critical path. The block diagram is shown in Fig. 5.1. The critical path for this design is through an adder and multiplier unit.

In this work we used transposed form of a pipelined 31 tap Low Pass Filter designed with Sampling frequency($F_s$) as 200hz; Pass band frequency($F_{pass}$) as 40hz and Stop band frequency($F_{stop}$) as 50hz. The block diagram is shown in Fig. 5.2. Extra delay elements/flipflops are inserted to pipeline the design. The critical path is now effectively through an adder unit or multiplier unit.

This filter is designed using Matlab FDA tool and the coefficient set coef0 till coef30 is generated. The width of the coefficients is 16 bits each. The filter has the following coefficient values: 216, 538, -287, -444, 84, 726, 131, -985, -566, 1240, 1349,

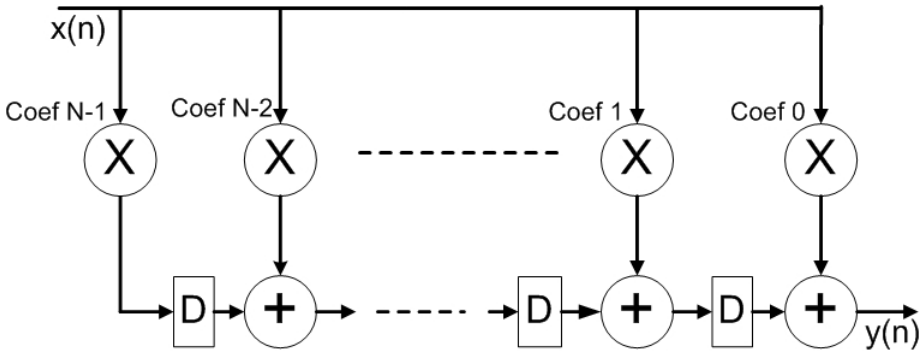



Fig. 5.1. Transpose form of an FIR Filter

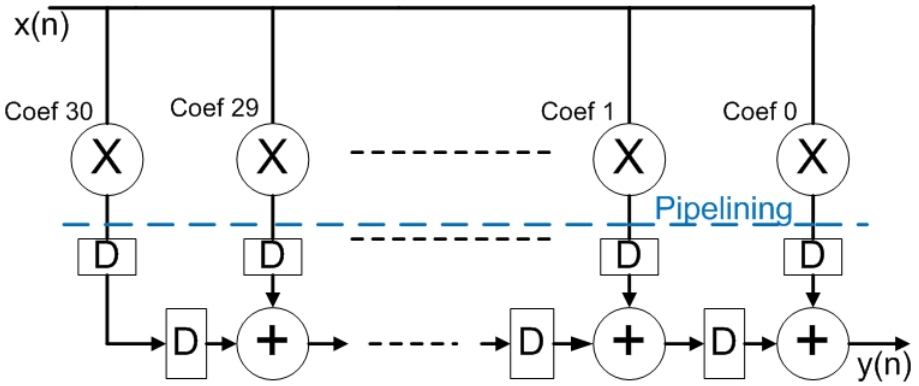



Fig. 5.2. Pipelined FIR

-1451, -3001, 1592, 10266, 14743, 10266, 1592, -3001, -1451, 1349, 1240, -566, -985, 131, 726, 84, -444, -287, 538, 216. Developed the filter design in Verilog. The data width of the inputs is 8 bits each and the data width of the outputs is 24 bits each. The FIR design is then synthesized using Synopsys Design Compiler with a clock constraint of 3.5ns and mapped to IBM 90nm standard cell library. As the design is pipelined all the addition and multiplication units run in parallel and critical path originates either from the adder unit or the multiplier unit.

By following the design flow as shown in Fig. 3.2 static timing analysis is performed on the gate level netlist. Sizing constraints are applied such that critical paths originate from the least significant input bits of the adder since truncating these bits are resulting in maximum delay reduction with minimum impact on the frequency

response. For every level of truncation where different no of input bits are truncated, the netlist is simulated and the quality impact on the filter response is measured in terms of pass band ripple and stop band ripple. Different combinations of truncation values are applied to these input bits for each level of truncation and the optimal truncation combination which has a minimum impact on the output quality is applied. Table 5.1 lists the percentage reduction in delay for each truncation level, the impact on the output quality measured in terms of pass band ripple and stop band ripple and the percentage decrease in switching power for each truncation as switching activity at some nodes is reduced.

Table 5.1
Truncation Results for FIR Design when the critical paths are through the adder unit

| **# of Truncated bits** | **Delay** | | **Pass Band Ripple** $\delta$ | | **Stop Band Ripple** $\delta$ | | **Power % reduction** |
|---|---|---|---|---|---|---|---|
| | **Absolute Value** | **% reduction** | **Absolute Value** | **in db** | **Absolute Value** | **in db** | |
| original | 3.41 | 0 | 0.0241 | 0.2219 | 0.0246 | 32.3022 | 0 |
| 1 | 3.33 | 2.3 | 0.0241 | 0.2119 | 0.0243 | 32.2879 | 0.2 |
| 2 | 3.20 | 6.1 | 0.0242 | 0.2128 | 0.02431 | 32.2843 | 0.4 |
| 3 | 3.02 | 11.4 | 0.025 | 0.2145 | 0.02499 | 32.0447 | 0.7 |
| 4 | 2.85 | 16.4 | 0.025 | 0.2145 | 0.02426 | 32.3022 | 0.7 |
| 5 | 2.17 | 20.5 | 0.0257 | 0.2261 | 0.02469 | 32.1496 | 0.8 |
| 6 | 2.54 | 25.5 | 0.028 | 0.2399 | 0.0275 | 31.2927 | 1.0 |
| 7 | 2.40 | 29.6 | 0.0349 | 0.3086 | 0.03029 | 30.3740 | 1.1 |
| 8 | 2.23 | 34.6 | 0.0793 | 0.7176 | 0.05216 | 25.6532 | 1.2 |
| 9 | 2.15 | 36.9 | 0.1184 | 1.0946 | 0.05986 | 24.4573 | 1.4 |

It is observed from Table 5.1 that as we truncate more input bits the deviation from the original frequency response is increasing. Also the critical path delay is reducing and there is a slight reduction in power as more input bits are truncated. The frequency response curves for design without truncation and with 1 to 9 bits truncation are shown in Fig. 5.3. Fig. 5.4 zooms into stop band region of Fig. 5.3

where the deviation from the original frequency response curve is clearly visible as we increase the number of truncation bits. As we can see from Fig. 5.3 and Fig. 5.4 truncation from 1 to 5 bits has very slight impact on the frequency response curves compared to the original and we get a significant delay reduction of 20%. As we do more truncation the amount of deviation from the original frequency response curve is increasing with more delay reduction. However depending on the demand for output quality the designer can always limit of the number of truncation bits and improve the manufacturing yield.

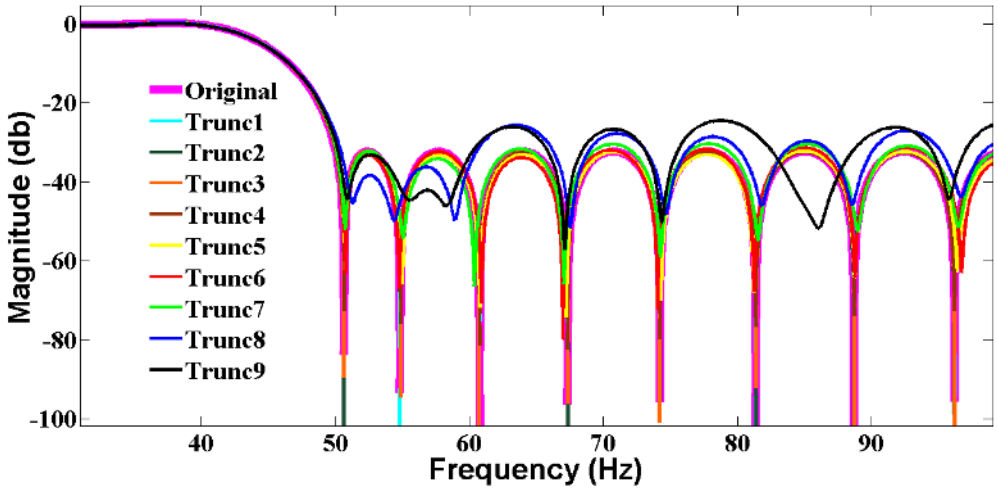



Fig. 5.3. Filter response for original and after truncating different no of input bits

Truncation circuit will be inserted into the design as shown in Fig. 4.2 to perform the desired truncation of the bits. The area overhead due to pull-up, pull-down and gating transistors and the decoder circuit is very less since truncation of each bit would require only 2 extra transistors as discussed in Chapter 4. Coming to power overhead, the truncation circuit is turned on as soon as the IC starts working. So the decoder, pull-up, pull-down and gating transistors will only switch once and hence there will be no dynamic power overhead. During the manufacturing test phase the ICs are tested and the truncation circuit is turned on. Each input combination of a decoder produces an output which truncates appropriate number of input bits in the

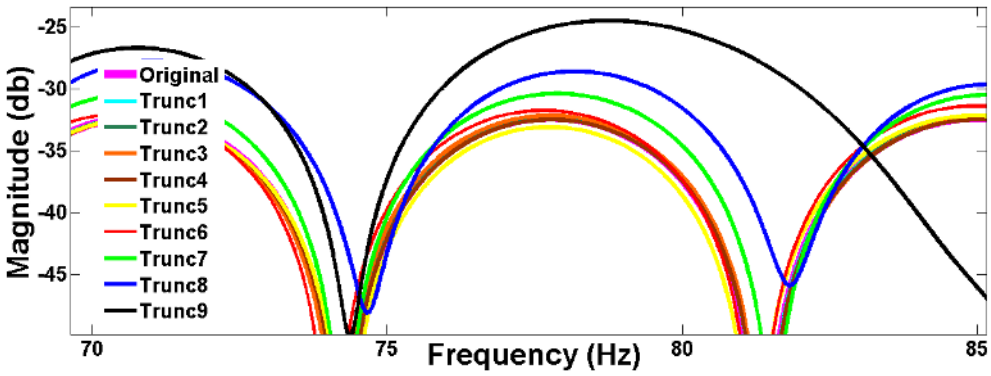


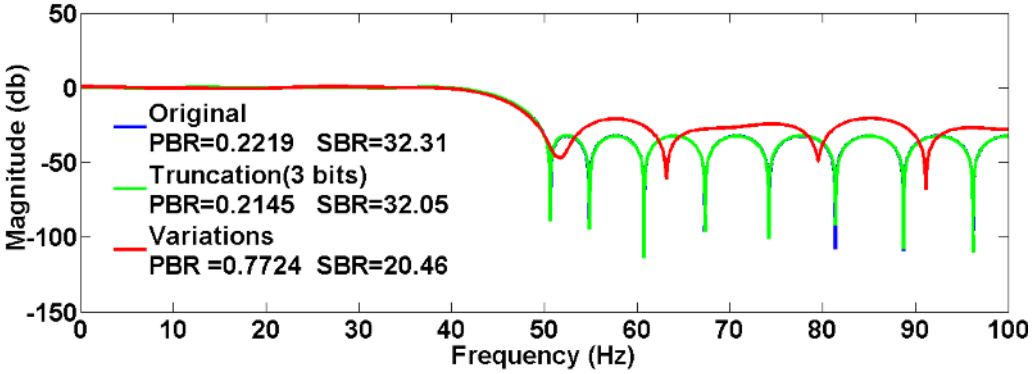

Fig. 5.4. Zooming into the stop band region of Fig. 5.3 where change in the ripple is more as more input bits are truncated

circuit. Depending on the sensed process corner of the chip the appropriate input combination to tolerate the process variation is stored in an non volatile memory attached to the chip so that as soon the IC starts working the input combination is always restored from the memory and is applied to the decoder. Thus it is made sure that the IC always meets the clock constraints and the manufacturing yield is improved at the cost of slight degradation in quality.

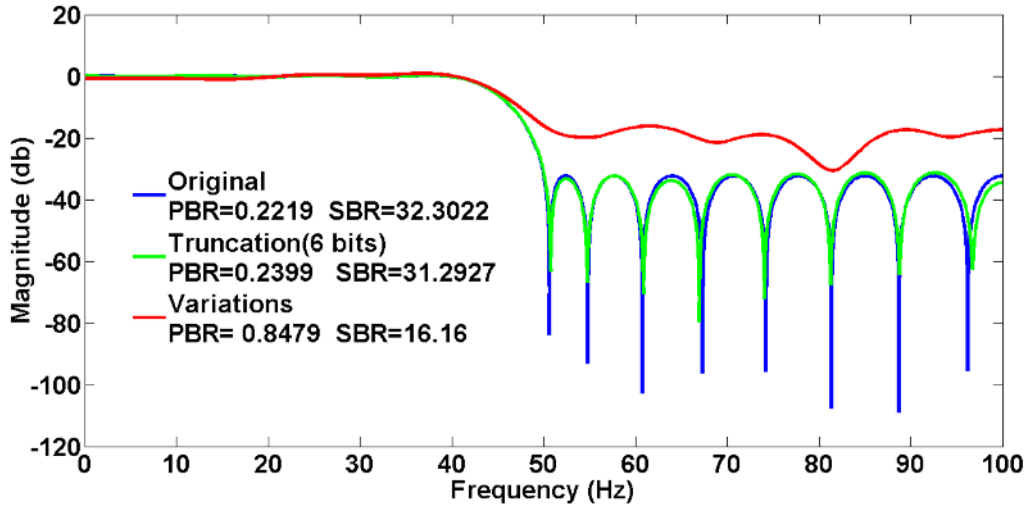
## 5.1 Effect of Process Variations

The effect of process variations with truncation on the output response of Low pass filter is shown in Fig.5.5for 10%, 20% and 30% variations. From Fig. 5.5(a) and Fig. 5.5(b) it is observed that the filter response after healing, by truncating appropriate number of input bits to compensate for the delay increment due to process variations, is very close to the original response. For extreme process variations as in Fig. 5.5(c) the filter response even after healing slightly degrades compared to the original but is better than the effect on the filter response due to process variations.

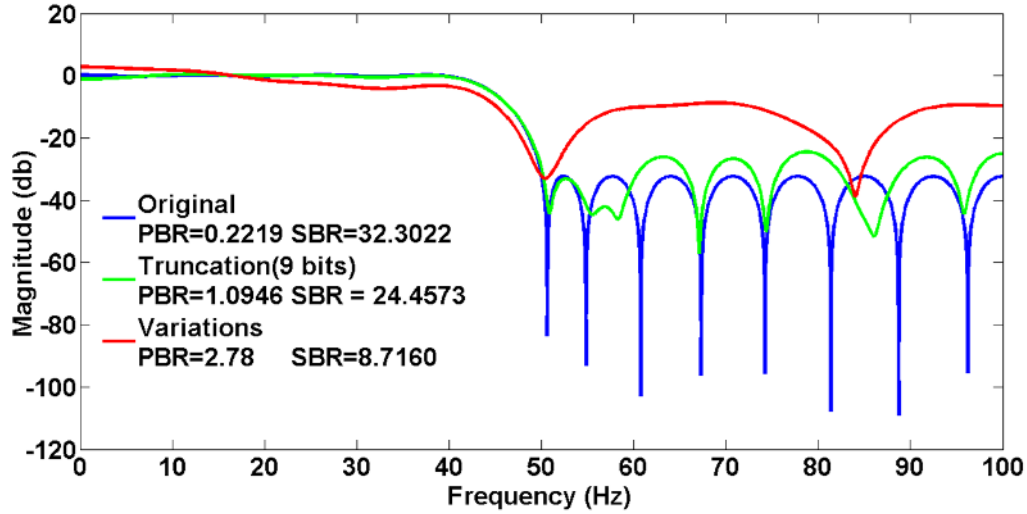
Thus the proposed technique VaROT proves to be effective on the FIR ciruit.




(a) 10%




(b) 20%




(c) 30%

Fig. 5.5. Frequency response of a Low Pass Filter with different amounts of process variations and truncation.

# 6. EXTENSIONS AND FUTURE DIRECTIONS

Although we have used truncation for process compensation in DSP hardware, it can also be effective for dynamic adaptation to temporal parameter variations - e.g. aging or environment induced delay variations. High-performance DSP circuits experience increased junction temperature during high activity. Activity-dependent temperature fluctuation can cause considerable variations in circuit delay [30]. Hence, unless sufficient delay margin is built into a design to account for worst-case temperature fluctuation, a DSP datapath can encounter delay failure with temperature shift leading to large degradation in QoS. The proposed approach can be used to truncate appropriate number of bits when the temperature goes beyond a pre-determined threshold, allowing a graceful degradation in quality. In this case the configuration bits for truncation need to be determined (based on design-time knowledge) and set dynamically. The entire operating range of temperature can be divided into multiple regions and required number of bits to be truncated can be pre-determined based on estimated delay shift in a specific temperature region. Similarly, periodic calibration of aging effect such as bias temperature instability (BTI) and hot carrier injection (HCI) can be combined with the proposed healing step. Hence, such an adaptation approach can also prevent pessimistic design for worst-case aging condition.

The proposed approach can also be effective for power saving with voltage scaling. Dynamic scaling of operating voltage has emerged as an effective approach for low-power operation due to quadratic impact of supply voltage on switching power. Voltage scaling also results large reduction in active leakage. However, unless the operating frequency is scaled in a commensurate manner at the cost of large impact on performance, voltage scaling leads to delay failures in DSP datapaths resulting in large degradation in QoS, as noted in [9]. The proposed approach can be effective to prevent large degradation in QoS at scaled supply via appropriate operand

truncation. Note that graceful degradation in QoS under voltage scaling can also be achieved with a skewed design approach as in [9] [10], where critical (in terms of QoS) components in a DSP unit are designed with higher delay margin than non-critical ones. However, the proposed design approach can be used as a lower overhead alternative (1% area overhead in VaROT over 12% in [9]). It can also be used as a complementary approach to that in [9] to minimize the impact on QoS. In this case, VaROT can be applied to the datapaths in less-critical components.

# 7. CONCLUSION

We have presented *VaROT* - a low-overhead post-silicon compensation approach for DSP hardware using dynamic truncation of operand width. The proposed approach can improve the profit with minimal impact on Quality of Sevice (QoS). It exploits the fact that critical paths in DSP datapaths typically originate from the input LSBs and truncation of these bits by setting them to fixed values results in shortening of the timing paths, leading to avoidance of delay failures in slow process corners without affecting the QoS considerably. Unlike the existing healing approaches using voltage/frequency scaling or body biasing, the proposed approach does not affect the performance and power of the DSP chips.

We have presented a design methodology to minimize the overhead due to truncation hardware and a gate sizing step, which makes the paths originating from the least significant bits longer to maximize delay improvement with truncation. Simulation results for two example applications, namely DCT and FIR, demonstrate the effectiveness of the approach in improving parametric yield by repairing the chips which fail to meet the target delay due to process variations. The healed ICs however suffer from slight degradation in QoS over nominal value. The proposed approach, hence, can benefit from a quality binning step, which sorts the repaired chips in bins with acceptable but slightly degraded QoS. The truncation method provides some opportunistic power saving in the healed ICs due to absence of switching in some input bits. The proposed healing approach can be easily combined with statistical design or other variation-tolerant design approach to maximize yield improvement under variations. Finally, the proposed healing approach for DSP datapaths can be combined with healing of embedded memory array and analog/mixed-signal cores to produce system-level self-healing [31] approach for complex system-on-chips.

# REFERENCES


[1] G. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 8, pp. 114–117, Apr. 1965.

[2] W. Maly, "Ic design in high-cost nanometer-technologies era," in *DAC '01: Proceedings of the 38th annual Design Automation Conference*, pp. 9–14, 2001.

[3] S. G. Duvall, "Statistical circuit modeling and optimization," in *Statistical Metrology, 5th International Workshop on*, pp. 56–63, 2000.

[4] K. A. Bowman, S. G. Duvall, and J. D. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 183 –190, feb 2002.

[5] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and D. Vivek, "Parameter variations and impact on circuits and microarchitecture," *Design Automation Conference*, p. 338, 2003.

[6] K. A. Bowman, X. Tang, J. C. Eble, and J. D. Meindl, "Impact of extrinsic and intrinsic parameter fluctuations on CMOS circuit performance," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1186–1193, feb 2000.

[7] A. Agarwal, K. Chopra, D. Blaauw, and V. Zolotov, "Circuit optimization using statistical static timing analysis," in *42nd Design Automation Conference*, pp. 321–324, june 2005.

[8] S. Bhunia, S. Mukhopadhyay, and K. Roy, "Process variations and process-tolerant design," in *20th International Conference on VLSI Design, held jointly with 6th International Conference on Embedded Systems*, pp. 699–704, jan 2007.

[9] N. Banerjee, G. Karakonstantis, and K. Roy, "Process variation tolerant low power dct architecture," *Design Automation Test in Europe Conference Exhibition*, pp. 1–6, apr 2007.

[10] J. H.Choi, N. Banerjee, and K. Roy, "Variation-aware low-power synthesis methodology for fixed-point fir filters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, pp. 87–97, jan 2009.

[11] J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," in *IEEE International Solid-State Circuits Conference*, pp. 422–478, 2002.

[12] S. Narendra, D. Antoniadis, and V. De, "Impact of using adaptive body bias to compensate die-to-die vt variation on within-die vt variation," in *ISLPED*, pp. 229–232, 1999.

[13] J. Tschanz, S. Narendra, R. Nair, and V. De, "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," *2002. Symposium on VLSI Circuits Digest of Technical Papers*, pp. 310–311, 2002.

[14] R. C. Gonzalez and R. E. Woods, "Digital image processing." Prentice Hall, 2002.

[15] F. Fang, T. Chen, and R. A.Rutenbar, "Floating-point bit-width optimization for low-power signal processing applications," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1993*, april 1993.

[16] D. U. Lee, A. A. Gaffar, R. C. Cheung, O. Mencer, W. Luk, and G. A. Constantinides, "Accuracy-guaranteed bit-width optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, pp. 1990 – 2000, oct 2006.

[17] T. Xanthopoulos and A. Chandrakasan, "A low-power dct core using adaptive bitwidth and arithmetic activity exploring signal correlations and quantization," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 740 – 750, May 2000.

[18] J. Park, J. H. Choi, and K. Roy, "Dynamic bit-width adaptation in dct: An approach to trade off image quality and computation energy," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, pp. 787 – 793, May 2010.

[19] Y. Liu, J. Liu, and T. Zhang, "Design of low-power variation tolerant signal processing systems with adaptive finite word-length configuration," in *11th International Symposium on Quality Electronic Design (ISQED), 2010*, pp. 372–379, march 2010.

[20] A. Nardi, A. Neviani, E. Zanoni, M. Quarantelli, and C. Guardiani, "Impact of unrealistic worst case modeling on the performance of VLSI circuits in deep submicron cmos technologies," *IEEE Transactions on Semiconductor Manufacturing*, vol. 12, pp. 396–402, nov 1999.

[21] W. Maly, "Computer-aided design for vlsi circuit manufacturability," *Proceedings of the IEEE*, vol. 78, pp. 356–392, feb 1990.

[22] H. Chang and S. S.Sapatnekar, "Statistical timing analysis considering spatial correlations using a single pert-like traversal," in *International Conference on Computer Aided Design, ICCAD*, pp. 621–625, 2003.

[23] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation," *36th Annual IEEE/ACM International Symposium on Microarchitecture 2003. MICRO-36. Proceedings*, pp. 7–18, dec 2003.

[24] R. Hegde and N. R.Shanbhag, "Soft digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, pp. 813–823, dec 2001.

[25] S. Ghosh, S. Bhunia, and K. Roy, "Crista: A new paradigm for low-power, variation-tolerant, and adaptive circuit synthesis using critical path isolation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, pp. 1947–1956, nov 2007.

[26] PTM[Online]. http://www.eas.asu.edu/~ptm/.

[27] S. Bhunia, H. Mahmoodi, D. Ghosh, S. Mukhopadhyay, and K. Roy, "Low-power scan design using first-level supply gating," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, pp. 384 – 395, mar 2005.

[28] A. Datta, S. Bhunia, J. H. Choi, S. Mukhopadhyay, and K. Roy, "Speed binning aware design methodology to improve profit under parameter variations," *ASP-DAC '06: Proceedings of the 2006 Asia and South Pacific Design Automation Conference*, pp. 712–717, 2006.

[29] [Online]. http://www.opencores.org.

[30] S. Krishnamurthy, S. Paul, and S. Bhunia, "Adaptation to temperature-induced delay variations in logic circuits using low-overhead online delay calibration," in *IEEE International Symposium on Quality Electronic Design*, March 2007.

[31] S. Narasimhan, S. Paul, R. S.Chakraborty, F. Wolff, C. Papachristou, D. Weyer, and S. Bhunia, "System level self-healing for parametric yield and reliability improvement under power bound," in *NASA/ESA Conference on Adaptive Hardware and System*, 2010.