

# **LOW-POWER FFT VIA REDUCED PRECISION REDUNDANCY**

**Srinivasa Raghavan Sridhara**

*Coordinated Science Laboratory*  
1308 West Main Street, Urbana, IL 61801  
University of Illinois at Urbana-Champaign

---

SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS None		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILLU-ENG-02-2220 (DAC 90)			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois		6b. OFFICE SYMBOL (If applicable) N/A		7a. NAME OF MONITORING ORGANIZATION NSF	
6c. ADDRESS (City, State, and ZIP Code) 1308 W Main St Urbana, IL 61801			7b. ADDRESS (City, State, and ZIP Code) 4201 Wilson Blvd Arlington, VA 22230		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION NSF		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code) 4201 Wilson Blvd Arlington, VA 22230			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
11. TITLE (Include Security Classification) Low-Power FFT Via Reduced Precision Redundancy					
12. PERSONAL AUTHOR(S) Sridhara, Srinivasa Raghavan					
13a. TYPE OF REPORT Technical		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Year, Month, Day) 02 Aug 05	
15. PAGE COUNT 43					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)  DSP, low-power, FFT		
FIELD	GROUP	SUB-GROUP			
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Low-power design of digital signal processing (DSP) systems is critical in supporting next generation wireless applications. Voltage overscaling and algorithmic noise tolerance (ANT) techniques are a means of achieving significant power savings over and beyond that achievable through voltage scaling alone. In this thesis, we propose a novel ANT scheme referred to as reduced precision redundancy for low-power fast Fourier transform (FFT) processors. We employ arithmetic unit architectures that are the most suitable for voltage overscaling and show that significant power savings can be achieved in butterfly functional unit of FFT processors through the proposed ANT technique.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS				21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL				22b. TELEPHONE (Include Area Code)	
				22c. OFFICE SYMBOL	

LOW-POWER FFT VIA REDUCED PRECISION REDUNDANCY

BY

SRINIVASA RAGHAVAN SRIDHARA

B. Tech., Indian Institute of Technology, Kharagpur, 1999

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Electrical Engineering  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2002

Urbana, Illinois

© Copyright by Srinivasa Raghavan Sridhara, 2002

## ACKNOWLEDGMENTS

I would like to thank Prof. Naresh Shanbhag for his advice and support. I would like to thank all members of the ViPS group for providing a stimulating environment for research. I am indebted to my parents and my sisters, Manjula and Sandhya, for their love and constant encouragement.

# TABLE OF CONTENTS

CHAPTER	PAGE
<b>1 INTRODUCTION</b> . . . . .	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Algorithmic Noise Tolerance . . . . .	2
1.3 Arithmetic Units . . . . .	4
1.4 Thesis Organization . . . . .	4
<b>2 LOW-POWER FFT</b> . . . . .	<b>6</b>
2.1 Reduced Precision Redundancy . . . . .	7
2.1.1 VOS error characteristics . . . . .	7
2.1.2 Reduced precision system . . . . .	8
2.2 FFT Algorithm and Architecture . . . . .	9
2.3 Multiplier with Reduced Precision Redundancy . . . . .	11
2.3.1 The FFT multiplier . . . . .	11
2.3.2 Reduced precision multiplier . . . . .	12
2.3.3 Decision threshold and comparator . . . . .	13
2.4 Simulation Results . . . . .	15
2.4.1 Performance metrics . . . . .	15
2.4.2 Performance of RPR . . . . .	17
2.5 Conclusions . . . . .	18
<b>3 SOFT DSP AWARE ARITHMETIC UNITS</b> . . . . .	<b>20</b>
3.1 Delay Balancing vs. VOS . . . . .	21
3.2 Inherent Delay Imbalance . . . . .	24
3.3 Adders with Voltage Overscaling . . . . .	27
3.4 Multipliers with Voltage Overscaling . . . . .	29
3.5 Conclusions . . . . .	32
<b>4 CONCLUSIONS AND FUTURE WORK</b> . . . . .	<b>33</b>
4.1 Summary . . . . .	33
4.2 Future Work . . . . .	34
<b>REFERENCES</b> . . . . .	<b>35</b>

# LIST OF TABLES

Table	Page
3.1 Power and delay for 16-bit adders from [4] relative to ripple carry adder. . . . .	27
3.2 Power and delay for $16 \times 16$ multipliers relative to Baugh-Wooley multiplier. . . .	30

# LIST OF FIGURES

Figure	Page
1.1 A soft DSP with a generic algorithmic noise tolerance (ANT) scheme. . . . .	3
2.1 Overview of the proposed algorithmic noise tolerance (ANT) scheme using reduced precision redundancy. . . . .	7
2.2 Decimation-in-time butterfly. . . . .	10
2.3 Path delay distribution of an $8 \times 8$ Baugh-Wooley multiplier. . . . .	11
2.4 Multiplier with reduced precision redundancy. . . . .	12
2.5 Comparator $ Y_o - Y_r  > Th$ when $Th = 2^{n+r-t}$ . . . . .	14
2.6 $SNR$ vs. $K_{vos}$ for $0.25\text{-}\mu\text{m}$ CMOS technology with $\alpha = 1.2$ . The corresponding power savings are shown in parentheses. . . . .	17
2.7 Variation of power savings with FFT precision $n$ and FFT block length $N$ . . . . .	19
3.1 Delay balancing to reduce glitches: (a) delay imbalanced circuit and (b) circuit with delay balancing. . . . .	22
3.2 (a) Circuit I with delay imbalance and (b) circuit II with inherent delay balance. . . . .	24
3.3 Cumulative path delay distribution of 16-bit adders. . . . .	28
3.4 Power-delay product vs. error rate. . . . .	29
3.5 Cumulative path delay distribution of $16 \times 16$ multipliers. . . . .	31
3.6 Power-delay product vs. error rate. . . . .	31



# CHAPTER 1

## INTRODUCTION

### 1.1 Motivation

Low-power design of digital signal processing (DSP) systems is critical in supporting next generation wireless applications. Energy efficient implementation of VLSI systems is also important to reduce packaging costs, improve reliability, and increase operational life of the circuits. The three components of power consumption in digital VLSI circuits are dynamic power dissipation, short circuit power dissipation, and leakage power dissipation. The dynamic component is the dominant source of power dissipation and depends on the circuit parameters as follows [1]:

$$P = tC_L V_{dd}^2 f_s, \quad (1.1)$$

where  $t$  is the transition activity,  $C_L$  is effective load capacitance,  $V_{dd}$  is the operating supply voltage, and  $f_s$  is the clock frequency.

Since the dynamic power dissipation depends on the square of the supply voltage, aggressive scaling of supply voltage has been widely used to reduce the power consumption in VLSI circuits. However, scaling of the supply voltage  $V_{dd}$  increases the circuit delay

$\tau_d$  as shown below:

$$\tau_d = \frac{C_L V_{dd}}{\beta(V_{dd} - V_t)^\alpha}, \quad (1.2)$$

where  $\alpha$  is the velocity saturation index,  $\beta$  is the gate transconductance, and  $V_t$  is the device threshold voltage. The increased circuit delay results in loss of system throughput. Hence, power reduction in conventional voltage-scaled systems is limited by a minimum supply voltage  $V_{dd-crit}$  at which the throughput requirements of the system are just met.

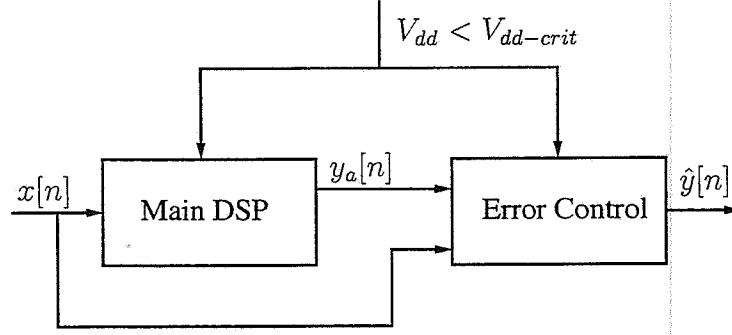
## 1.2 Algorithmic Noise Tolerance

Recently, voltage overscaling (VOS) [2] in combination with algorithmic noise-tolerance (ANT) has been proposed as a means of achieving significant power savings over and beyond that achievable by voltage scaling alone. VOS refers to the scaling of supply voltage beyond  $V_{dd-crit}$ , without sacrificing the throughput, i.e.,

$$V_{dd} = K_{vos} V_{dd-crit}, \quad 0 \leq K_{vos} < 1, \quad (1.3)$$

where  $K_{vos}$  is the VOS factor. VOS makes the critical path delay  $T_{cp} > T_s$ , where  $T_s$  is the clock period. This results in input-dependent (hence intermittent or *soft*) errors whenever paths with delays greater than  $T_s$  are excited, leading to a severe signal-to-noise ratio (*SNR*) loss.

Energy-efficient means of compensating for these soft errors is achieved via ANT [2], [3]. The ANT technique uses an error control block that mitigates the effect of soft errors



**Figure 1.1** A soft DSP with a generic algorithmic noise tolerance (ANT) scheme.

occurring at the output of a DSP block with VOS. Figure 1.1 shows a *soft DSP* system using a generic ANT scheme. In prediction based ANT schemes [2], a linear predictor is used as the error control block to restore the *SNR* in a narrowband frequency-selective filter. The adaptive error-cancellation [3] based ANT scheme models the error using an adaptive filter and cancels the estimated error from the output of main DSP.

The prediction and adaptive error cancellation based techniques are suitable for frequency-selective filtering. In this thesis, we employ a novel ANT scheme referred to as reduced precision redundancy (RPR) to significantly reduce the power dissipation in the butterfly functional unit of a fast Fourier transform (FFT) processor.

As supply voltage  $V_{dd}$  approaches the threshold voltage  $V_t$  in future technologies, voltage scaling and, in particular, VOS, may not remain a viable means of power reduction due to the resulting severe delay penalty. However, the soft errors also occur if the clock frequency is increased beyond a critical frequency  $f_{crit}$  (frequency overscaling). Therefore, ANT schemes can be used in the design of high-speed DSP systems with marginal penalty in power. Further, ANT schemes can be used to mitigate the effects of deep submicron noise and process variations [2] resulting in robust DSP systems.

## 1.3 Arithmetic Units

Power consumption in a given DSP datapath is determined by the power dissipated in the arithmetic units present in the datapath. Therefore, minimizing the power consumption of arithmetic units is of great importance. Since performance is often limited by arithmetic units' speed, it is also important to maximize speed. Frequently, the compromise between these conflicting demands of low power dissipation and high speed can be accomplished by selecting the arithmetic unit that has the minimum power-delay product.

Extensive studies have been done to compare a variety of adders [4] and multipliers [5] based on their power-delay product. However, it was shown in [2] that the frequency of soft errors due to voltage overscaling mainly depends on the path delay distribution of the arithmetic units. Specifically, it was shown that delay imbalance in arithmetic units increases the achievable power savings. Therefore, to select a suitable arithmetic unit for a soft DSP system, we not only need to consider the power-delay product but also study the delay imbalance of the arithmetic unit. In this thesis, we study the delay imbalance of various adders and multipliers and determine the soft DSP friendly architectures.

## 1.4 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we introduce the novel reduced precision redundancy (RPR) technique for mitigating the degradation in  $SNR$  due to VOS. An FFT multiplier architecture that employs the proposed RPR scheme

is presented. Extensive simulation results show the achievable power savings using the proposed scheme.

In Chapter 3, we show how delay imbalance helps voltage overscaling. We study the delay imbalance present in a variety of adders and multipliers and show that the arithmetic units with higher delay imbalance perform better with VOS. We use the result of the comparison to further improve the power savings achieved in the FFT architecture with the RPR scheme.

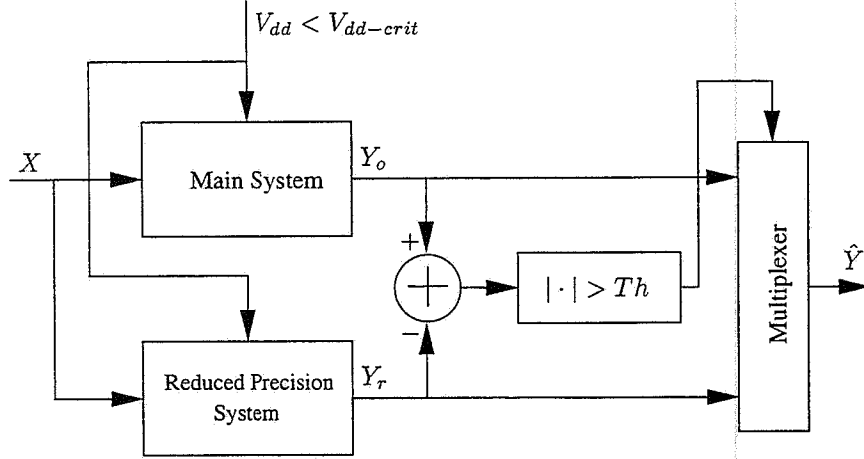
In Chapter 4, we present the conclusions and outline possible future work.

## CHAPTER 2

# LOW-POWER FFT

A low-power implementation of the fast Fourier transform (FFT) is critical in supporting next generation wireless LAN and wireless access systems. These systems are based upon orthogonal frequency division multiplexing (OFDM) [6], [7] requiring FFT processors in the transceivers. In the past, aggressive scaling of the supply voltage has been employed to reduce the power consumption in FFT processors [7], [8] so as to meet the stringent power budget of such systems. However, as described in Chapter 1, scaling of the supply voltage  $V_{dd}$  increases the circuit delay  $\tau_d$ , resulting in processor throughput loss. Hence, we apply VOS to achieve power savings over and beyond that achievable by voltage scaling alone.

In this chapter, we introduce a novel ANT scheme referred to as reduced precision redundancy (RPR) [9], [10] to restore the performance degradation at the output of the FFT processor due to VOS. We present the proposed technique in Section 2.1. We describe the architecture of a typical FFT processor in Section 2.2 and identify multipliers as the major sources of power consumption in an FFT butterfly. In Section 2.3, we describe how RPR can be applied to FFT multipliers. Simulation results are presented in Section 2.4 and conclusions are in Section 2.5.



**Figure 2.1** Overview of the proposed algorithmic noise tolerance (ANT) scheme using reduced precision redundancy.

## 2.1 Reduced Precision Redundancy

In this section, we describe the reduced precision redundancy (RPR) based ANT scheme. The proposed soft DSP system is shown in Figure 2.1, where a reduced precision replica operates in parallel with the main system with VOS in order to detect and correct errors. We first describe the error characteristics of the output of the main system due to VOS and then present the proposed scheme.

### 2.1.1 VOS error characteristics

Voltage overscaling introduces input-dependent soft errors whenever a path with delay greater than the clock period  $T_s$  is excited. Since most of the arithmetic units employed in DSP systems are based on least significant bit (LSB) first computation, soft errors appear first in the most significant bits (MSBs), resulting in errors of large magnitude.

These errors, though they degrade the performance severely, are desirable as they are easier to detect if the correct MSB information is known.

### 2.1.2 Reduced precision system

The reduced precision system is a replica of the main system but with reduced precision operands. For a given set of inputs  $X$ , the output of reduced precision system  $Y_r$  is generally not equal to the correct output  $Y$  (the output when the main system operates with  $V_{dd} = V_{dd-crit}$ ) due to truncation. However, since VOS causes large errors, we can employ  $Y_r$  to detect errors in  $Y_o$ . For example, an error in  $Y_o$  is said to have occurred if the difference  $|Y_o - Y_r|$  is greater than a threshold  $Th$ . In the event an error is detected, the output of the reduced precision system is declared as the actual output.

$$\hat{Y} = \begin{cases} Y_o & \text{if } |Y_o - Y_r| \leq Th \\ Y_r & \text{if } |Y_o - Y_r| > Th. \end{cases} \quad (2.1)$$

To ensure that  $\hat{Y} = Y_o$  when  $Y_o = Y$ , i.e., the output of the main system is chosen as the final output when there is no soft error,  $Th$  is chosen as

$$Th = \max_{all X} |Y - Y_r|. \quad (2.2)$$

The analysis above assumes that the reduced precision system does not suffer from soft errors. This assumption is valid as long as the critical path delay of the reduced precision system is smaller than the clock period  $T_s$ . This is guaranteed if the delays

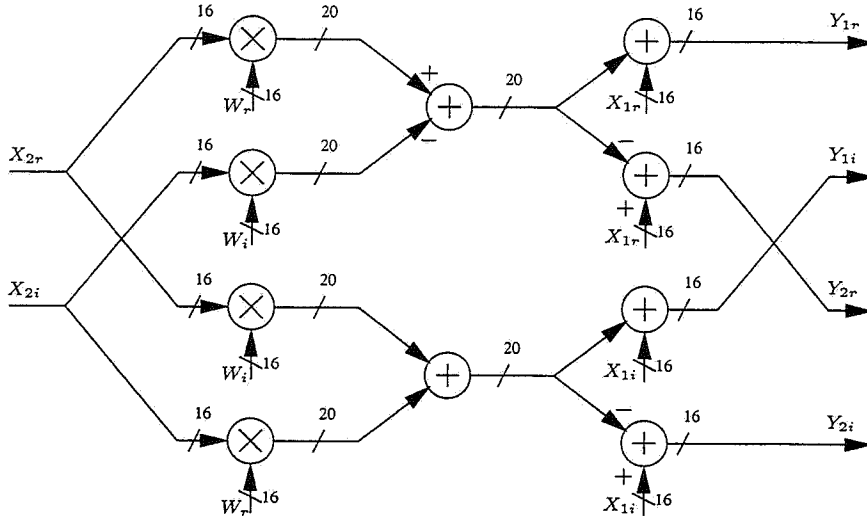


of adders and multipliers decrease linearly with decrease in precision, which is indeed the case for array architectures. For example, if the precision is halved in the standard 0.25- $\mu\text{m}$  CMOS technology, then using a velocity saturation index of  $\alpha = 1.2$  in (1.2), we see that the supply voltage can be reduced up to  $K_{vos} = 0.44$  before the critical path of the reduced precision system is violated. Further, to achieve the goal of reducing the power and maintaining the  $SNR$ , the reduced precision system should consume minimal power. Therefore, for a given value of  $K_{vos}$ , we choose the smallest precision that achieves the specified  $SNR$ .

Note that precision reduction has been employed to reduce power [11], [12] in DSP systems before. In these systems, power is traded off with  $SNR$ . Our technique differs from those in [11], [12] in that the  $SNR$  of the original system is maintained.

## 2.2 FFT Algorithm and Architecture

We consider an FFT processor that implements a radix-2 decimation-in-time (DIT) algorithm. The processor's datapath computes one complex radix-2 DIT butterfly per cycle [8]. The DIT butterfly calculates two outputs,  $Y_1 = X_1 + WX_2$  and  $Y_2 = X_1 - WX_2$ , from two inputs  $X_1$  and  $X_2$ , and a twiddle factor  $W$  as shown in Figure 2.2, where subscripts  $r$  and  $i$  stand for real and imaginary components, respectively. It is assumed that appropriate pipelining has been employed to route data between the memory and the functional units in order to maximize throughput. A VLSI implementation of such a processor [8] shows that, apart from the main memory, multipliers are the largest functional



**Figure 2.2** Decimation-in-time butterfly.

units. Therefore, we employ VOS in the multipliers while exploiting the algorithmic properties of an FFT computation to correct the errors.

In this paper, we consider a 64-point FFT processor with 16-bit precision operating on 10-bit fixed-point complex inputs. These parameters are typical for an FFT in a wireless local area network (WLAN) orthogonal frequency division multiplexing (OFDM) modem [6]. Four Baugh-Wooley multipliers along with two adders multiply  $W$  and  $X_2$ . Then, four adders generate  $Y_1$  and  $Y_2$ . Since, the fixed-point data format requires the eventual truncation of butterfly outputs when writing the outputs back to memory, it was shown in [8] that calculating all 32 bits of the multiplier output is unnecessary. Therefore, only the partial products needed to compute 20 MSBs are calculated to reduce area and power consumption.

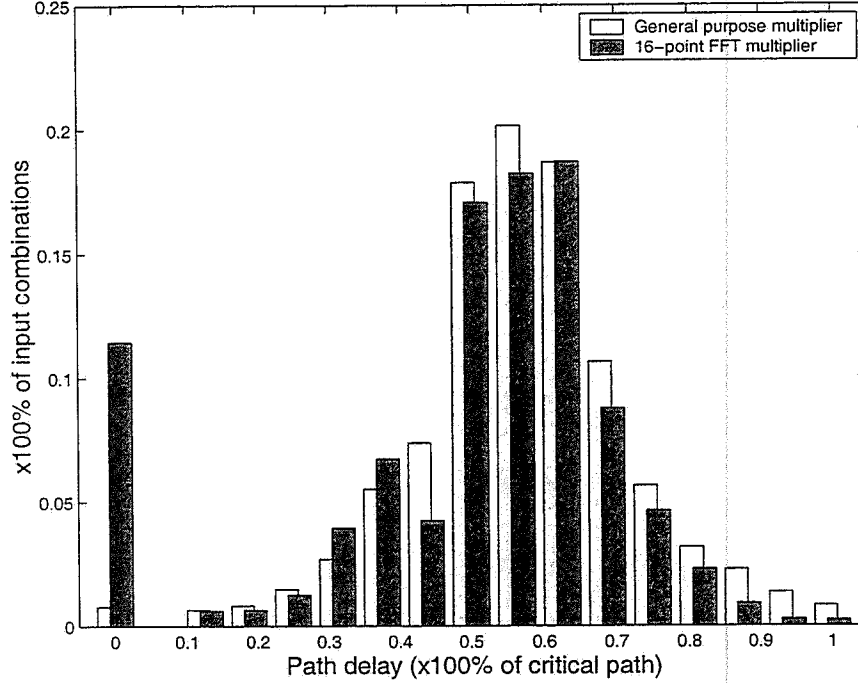


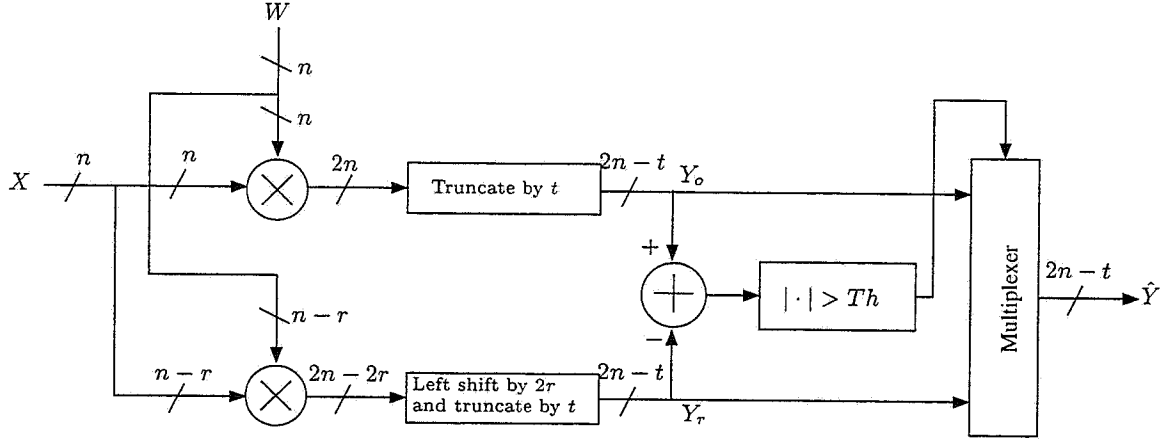
Figure 2.3 Path delay distribution of an  $8 \times 8$  Baugh-Wooley multiplier.

## 2.3 Multiplier with Reduced Precision Redundancy

### 2.3.1 The FFT multiplier

It was shown in [2] that the frequency of errors in a voltage overscaled system depends upon  $K_{\text{TC}}$  and on the path delay distribution of the arithmetic units. The path delay histogram for all possible input combinations of an  $8 \times 8$  Baugh-Wooley multiplier is shown in Figure 2.3. It is seen that about 14% of the input combinations excite paths with delays greater than 75% of the critical path delay. Therefore, VOS can introduce severe *SNR* degradation that may not be efficiently compensated for by ANT techniques.

The multiplier used in an FFT processor is more amenable to voltage overscaling than a general purpose array multiplier due to the fact that in FFTs the real and imaginary



**Figure 2.4** Multiplier with reduced precision redundancy.

parts of the twiddle factor take only  $N/2 + 1$  distinct values where  $N$  is the length of FFT. For example, in a 16-point FFT with 8-bit precision, the twiddle factor components take only  $16/2 + 1 = 9$  distinct values of the possible  $2^8 = 256$  values. Figure 2.3 also shows the path delay histogram of the multiplier with one operand taking only the nine possible twiddle factor values. It is easily observed that there is a significant reduction in the percentage of longer paths. In particular, now only 8% of the input combinations excite paths with delays longer than 75% of the critical path delay.

### 2.3.2 Reduced precision multiplier

In order to compensate for the loss in  $SNR$  in a voltage overscaled  $n \times n$  multiplier, we introduce reduced precision redundancy in the form of  $(n - r) \times (n - r)$  multiplier as shown in Figure 2.4. The operands to the reduced precision multiplier are the same as the operands to the main multiplier but truncated by  $r$  bits. The output  $Y_r$  of the reduced precision multiplier is  $2n - 2r$  bits wide and provides the MSB information required for

error control. Therefore, in the subtractor, these bits are used as the MSB bits and this is denoted as left shift in the figure. As discussed in Section 2.2, the output of the multiplier needs to be truncated by  $t$  bits to save area and power. Therefore, outputs of both the multipliers are truncated by  $t$  bits and the final output  $\hat{Y}$  is  $2n - t$  bits wide. Note that the left shift and the truncation operations do not require any hardware.

The output  $Y_r$  of the reduced precision multiplier is subtracted from the output  $Y_o$  the main multiplier with VOS and the absolute value of the result  $|Y_o - Y_r|$  is compared with the decision threshold  $Th$ . One of  $Y_o$  and  $Y_r$  is selected as the final output  $\hat{Y}$  according to (2.1). From (2.2), the maximum difference between the reduced precision output and full precision output at  $V_{dd-crit}$  is used as the decision threshold which is computed next.

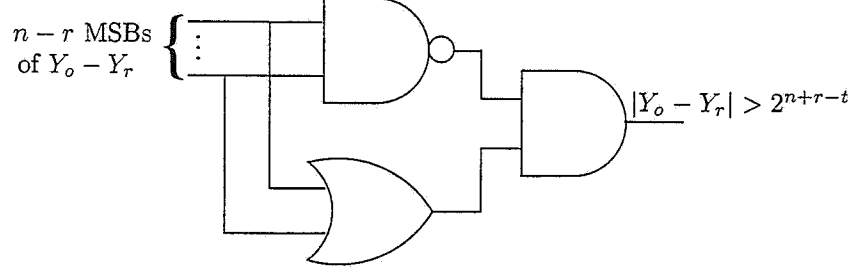
### 2.3.3 Decision threshold and comparator

The input to the reduced precision multiplier suffers maximum truncation when all of the  $r$  truncated bits are 1 and the amount of maximum truncation is equal to  $2^r - 1$ . Since the largest number in magnitude that can be represented using  $n$  bits in two's complement is  $-2^{n-1}$ , the maximum difference occurs when both the input operands have a value equal to  $(-2^{n-1} + 2^r - 1)$ . Therefore, the decision threshold  $Th$  is given by

$$Th = |(-2^{n-1} - 2^r + 1)(-2^{n-1} - 2^r + 1) - (-2^{n-1})(-2^{n-1})|, \quad (2.3)$$

which simplifies to

$$Th = (2^{n+r} - 2^n - 2^{2r} + 2^{r+1} - 1). \quad (2.4)$$



**Figure 2.5** Comparator  $|Y_o - Y_r| > Th$  when  $Th = 2^{n+r-t}$

Since the outputs of the multipliers are truncated by  $t$  bits,

$$Th = (2^{n+r} - 2^n - 2^{2r} + 2^{r+1} - 1) \cdot 2^{-t}. \quad (2.5)$$

For large  $n$  and  $r \approx n/2$ ,  $2^{n+r} \gg (2^n + 2^{2r} - 2^{r+1} + 1)$ . Therefore,  $Th$  can be approximated as

$$Th = 2^{n+r-t}. \quad (2.6)$$

The advantage of choosing  $Th$  according to (2.6) is that the comparator can be realized with the simple circuit shown in Figure 2.5 instead of a  $2n - t$  bit full adder.

Based on simulation of an FFT processor with  $n = 16$ , we find that  $r = 7$  provides the best trade-off between the ability to restore  $SNR$  and the area overhead due to additional hardware. This choice of  $r$  also satisfies the assumption  $r \approx n/2$  made above. The proposed technique involves a hardware overhead of about 40%. However, in Section 2.4.2, we show that the power savings achieved through VOS more than compensate for this overhead.

## 2.4 Simulation Results

In this section, we describe the metrics for algorithmic performance and power consumption used for evaluating the proposed FFT butterfly architecture; then we show the power savings achievable through the proposed RPR technique.

### 2.4.1 Performance metrics

The output  $SNR$  of the conventional FFT processor is defined as

$$SNR_{REF} = 10 \log \left( \frac{\sigma_s^2}{\sigma_{trunc1}^2} \right), \quad (2.7)$$

where  $\sigma_{trunc1}^2$  is the variance of truncation error due to fixed-point data format and  $\sigma_s^2$  is variance of the FFT outputs in the absence of truncation error. The output  $SNR$  of the voltage overscaled FFT processor is given by

$$SNR_{VOS} = 10 \log \left( \frac{\sigma_s^2}{\sigma_{trunc2}^2 + \sigma_{vos}^2} \right), \quad (2.8)$$

where  $\sigma_{trunc2}^2$  is the variance of truncation error due to fixed-point data format and  $\sigma_{vos}^2$  is the variance of the soft error introduced at the output due to VOS. The output  $SNR$  of the voltage overscaled FFT processor with reduced precision redundancy is defined as

$$SNR_{ANT} = 10 \log \left( \frac{\sigma_s^2}{\sigma_{trunc2}^2 + \sigma_{ANT}^2} \right), \quad (2.9)$$

where  $\sigma_{trunc2}^2$  is the variance of truncation error and  $\sigma_{ANT}^2$  is the variance of the residual error at the output. Since we wish to meet the  $SNR$  specification of the reference system,  $SNR_{ANT} \geq SNR_{REF}$ . Therefore,

$$\sigma_{trunc2}^2 + \sigma_{vos}^2 \leq \sigma_{trunc1}^2. \quad (2.10)$$

We achieve this by increasing the internal datapath width to 23 in the DIT butterfly of Figure 2.2.

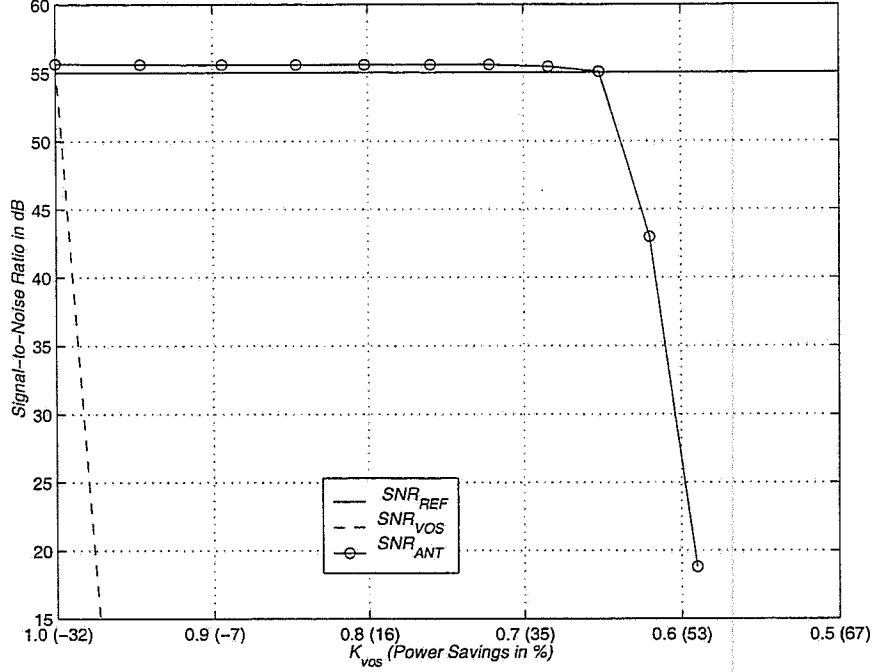
The power dissipation in the butterfly functional unit viz. four multipliers and six adders, is estimated through the gate-level power simulator MED [13]. If  $P_{REF}$  is the power dissipation of conventional butterfly at  $V_{dd-crit}$  and  $P_{ANT}$  is the power dissipation of the voltage overscaled butterfly with the proposed ANT technique, then the power savings  $P_{sav}$  is given by

$$P_{sav} = \frac{P_{REF} - P_{ANT}}{P_{REF}} \times 100\%. \quad (2.11)$$

Since dynamic power dissipation depends on the square of the supply voltage,  $P_{ANT} = K_{vos}^2 (P_{REF} + P_{overhead})$ , where  $P_{overhead}$  is the power dissipation overhead due to the ANT scheme. Therefore,

$$P_{sav} = \left[ 1 - K_{vos}^2 \left( 1 + \frac{P_{overhead}}{P_{REF}} \right) \right] \times 100\%. \quad (2.12)$$





**Figure 2.6**  $SNR$  vs.  $K_{vos}$  for 0.25- $\mu\text{m}$  CMOS technology with  $\alpha = 1.2$ . The corresponding power savings are shown in parentheses.

## 2.4.2 Performance of RPR

The FFT processor used in simulations has typical WLAN OFDM parameters [6] with FFT length  $N = 64$ , FFT precision  $n = 16$  bits and input precision of 10 bits. The inputs to reduced precision multipliers are truncated by  $r = 7$  and the internal truncation  $t$  is set to 9 bits for the voltage overscaled butterfly with ANT while the reference butterfly has internal truncation of 12 bits.

Simulations are performed with 1000 random complex inputs of length 64 using 0.25- $\mu\text{m}$  CMOS technology and velocity saturation index  $\alpha = 1.2$ . Figure 2.6 plots  $SNR$  vs.  $K_{vos}$  for the simulation with the corresponding power savings shown in parentheses. The  $SNR$  of the original system when  $V_{dd} = V_{dd-crit}$  (i.e., VOS is not applied yet) equals 55 dB. It is seen that the requirement  $SNR_{ANT} \geq SNR_{REF}$  is satisfied

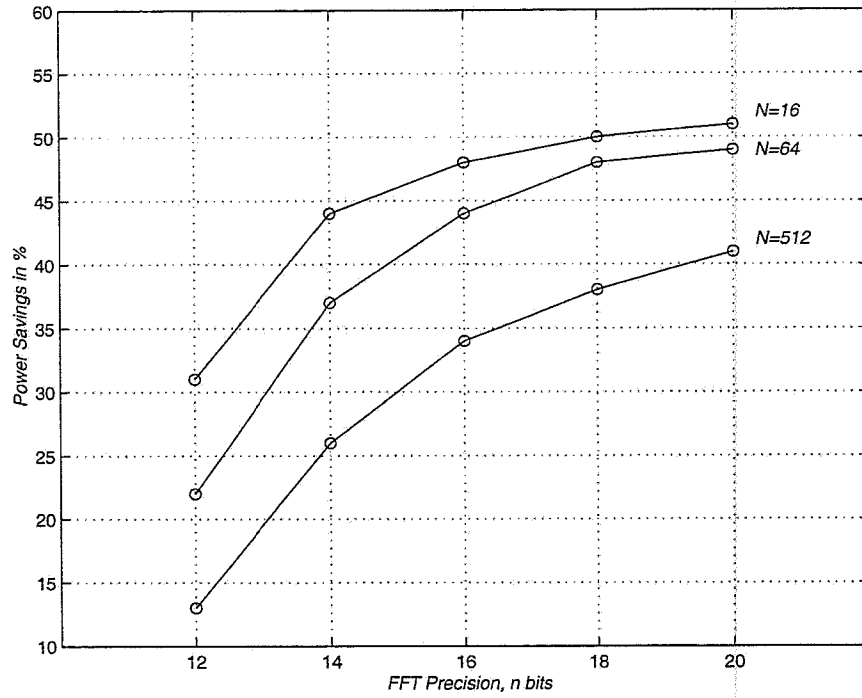
until  $K_{vos} = 0.65$ , which corresponds to approximately 44% power savings. Note that  $SNR_{VOS}$ , the  $SNR$  of the reference system under VOS but without ANT, falls off rapidly with decrease in  $K_{vos}$ .

As a comparison, simulation results show that about 27% power savings can be achieved with respect to  $P_{REF}$  by directly reducing FFT precision of the original system by 1 bit without VOS. This, however, results in an  $SNR$  loss of 3 dB, which fails to meet the  $SNR$  requirements. Thus, the proposed scheme achieves substantially more power savings without any loss in  $SNR$ .

The achievable power savings through the proposed scheme depend on the length and the precision of FFT. This is mainly due to the variation in the path delay distribution of the FFT multiplier with the variation in length and precision as described in Section 2.3. Figure 2.7 shows the variation of power savings with FFT precision  $n$  and FFT length  $N$ . For a given precision, the power savings decrease with increase in FFT length as this leads to an increase in the percentage of longer paths that fail to compute at a specific value of  $K_{vos}$ . Similarly, for a given FFT length, the power savings improve with increase in FFT precision.

## 2.5 Conclusions

In this chapter, we have shown that VOS and reduced precision redundancy reduce the power dissipation of the butterfly functional unit of an FFT processor by 44% over



**Figure 2.7** Variation of power savings with FFT precision  $n$  and FFT block length  $N$ .

present day voltage scaling. Reduced precision redundancy in multipliers has been shown to effectively compensate for the soft errors introduced due to VOS.

# CHAPTER 3

## SOFT DSP AWARE ARITHMETIC UNITS

The effectiveness of an ANT scheme depends, to a significant extent, on the path delay distribution of the architecture. Specifically, the achievable power savings improve with decrease in the number of input combinations exciting long paths of the circuit. In other words, delay imbalance helps voltage overscaling. However, past work on low-power arithmetic unit design has focused on delay balancing by adding additional delay circuits to reduce glitching power [14], [15]. In this chapter, we derive the conditions under which VOS provides higher power savings than reduction of glitches by adding delay circuits. In order to derive the conditions, we assume that ANT techniques exist for effectively controlling the errors due to VOS and that the ANT schemes consume relatively small amounts of power.

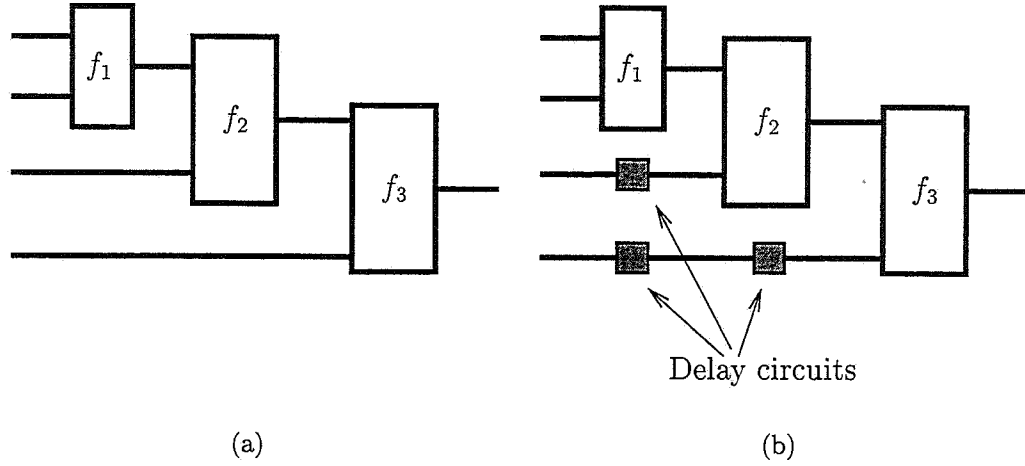
Further, there are several arithmetic unit architectures that have *inherent* delay balance. Note that we use the term “inherent” to distinguish the delay balanced architectures that result from delay optimization or other optimization techniques from those architectures that are delay balanced by adding delay circuits. Therefore, to compare various arithmetic unit architectures, we explore the path delay distribution of the architectures. We use this information to obtain the power-delay product for each of the

architectures in the presence of voltage overscaling and, hence, in the presence of errors. Note that the path delay distribution also depends on the input statistics which, in turn, are dependent on the application. To make comparisons independent of the application, we assume that all inputs are equally likely to appear. This assumption is not valid for many applications; however, it is needed for the ease of comparison. Further, we assume that only the frequency, and not magnitude, of errors introduced by voltage overscaling is important. This is justified as the currently known ANT schemes do not make use of the erroneous value in any way and the architectures use LSB-first computations leading to MSB errors.

Section 3.1 compares delay balancing with VOS and derives the condition under which VOS provides higher power savings. Section 3.2 derives the conditions under which delay imbalanced architectures are better than inherently delay balanced architectures. Section 3.3 presents the comparison between three types of adders. Section 3.4 compares three types of multipliers and uses the result in the FFT processor, described in Chapter 2, to achieve additional power savings. Section 3.5 presents the conclusions.

### **3.1 Delay Balancing vs. VOS**

Delay balancing is a popular technique for reducing the glitching power. In this section, we derive the conditions under which voltage overscaling provides better power savings than delay balancing.



**Figure 3.1** Delay balancing to reduce glitches: (a) delay imbalanced circuit and (b) circuit with delay balancing.

Delay imbalance of the signals in a circuit causes spurious transitions or glitches and, hence, unnecessary power dissipation. Most of these glitches can be eliminated by introducing delay circuits in faster paths to balance the delays. Figure 3.1(a) shows a typical delay imbalanced circuit, in which one of inputs to blocks  $f_2$  and  $f_3$  arrives earlier than the other. Figure 3.1(b) shows the corresponding delay balanced circuit where appropriate delay circuits have been added to make the arrival times of inputs to  $f_2$  and  $f_3$  similar.

Let  $P_{glitch1}$  and  $P_{useful}$  denote the power dissipations in the delay imbalanced circuit due to useless and useful transitions, respectively. Further, let  $m_1$  denote the ratio of useless transitions to useful transitions in the delay imbalanced circuit. Then,

$$P_{glitch1} = m_1 P_{useful}. \quad (3.1)$$

The total power dissipation is the sum of the glitching power and the useful power.

Therefore,

$$P_{total1} = (1 + m_1)P_{useful}. \quad (3.2)$$

If the delay imbalanced circuit is voltage overscaled such that the voltage overscaling factor is  $K_{vos}$ , then the power dissipation is

$$P_{total1vos} = K_{vos}^2(1 + m_1)P_{useful}, \quad (3.3)$$

where  $K_{vos}$  is assumed to include the overhead due to the ANT scheme.

After the addition of delay circuits as shown in Figure 3.1, let  $m_2$  be the ratio of useless transitions to useful transitions. Delay balancing only reduces useless transitions; therefore,  $m_2 < m_1$ . The total power dissipation in the circuit with delay balancing is

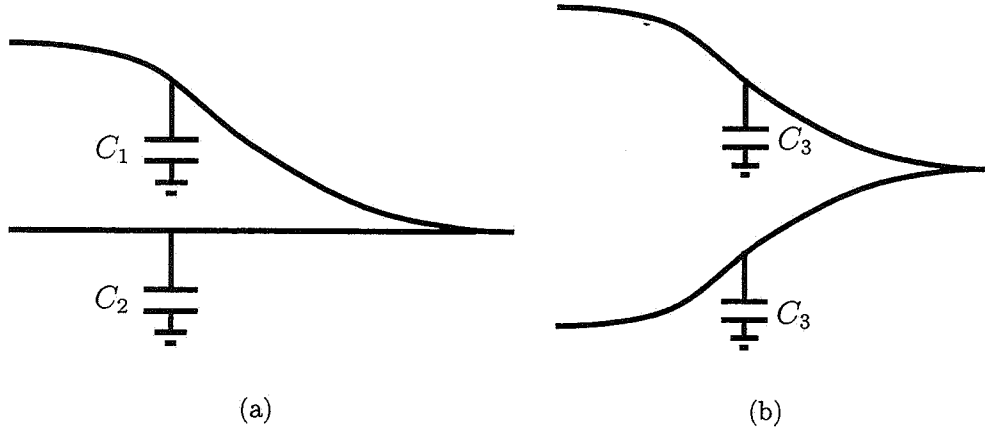
$$P_{total2} = (1 + m_2)P_{useful}. \quad (3.4)$$

Voltage overscaling is an useful option if

$$P_{total1vos} < P_{total2}. \quad (3.5)$$

Substituting (3.3) and (3.4) into (3.5), we get

$$K_{vos}^2 < \frac{1 + m_2}{1 + m_1}. \quad (3.6)$$



**Figure 3.2** (a) Circuit I with delay imbalance and (b) circuit II with inherent delay balance.

Typical values of  $K_{vos}^2$  reported in [2], [3], [9] and [10] range from 0.4 to 0.6, while up to 36% power savings has been reported using delay balancing [15]. Therefore,  $\frac{1+m_2}{1+m_1} = 0.64$  which is greater than typically reported value of  $K_{vos}^2$ . Therefore, in general, VOS provides higher power savings than reduction of glitching power through delay balancing.

### 3.2 Inherent Delay Imbalance

In this section, we derive the conditions under which delay imbalanced architectures perform better in terms of power-delay product than inherently delay balanced architectures. Consider the two circuits shown in Figure 3.2. Circuit I is delay imbalanced in the sense that  $C_1$  is excited with probability  $p \ll 1$  and  $C_1 > C_2$ . Circuit II is delay balanced with both paths excited equally often and, for simplicity, the two paths have the same capacitance  $C_3$ . Further, circuit II is assumed to have better power-delay product than circuit I.



$$P_{II}\tau_{II} < P_I\tau_I \quad (3.7)$$

Since circuit II is inherently delay balanced, it is faster than circuit I at the same supply voltage  $V_{dd}$ , i.e.,

$$\tau_I > \tau_{II}. \quad (3.8)$$

Using the delay equation given in (1.2), it can be inferred that

$$C_1 > C_3 \quad (3.9)$$

Now, given (3.7) and (3.8), it is possible that one of the following two conditions holds:  $P_I < P_{II}$  or  $P_I > P_{II}$ . Let us first consider the case where  $P_I < P_{II}$ . Using the power equation given in (1.1), it can be inferred that

$$C_1 + C_2 < 2C_3. \quad (3.10)$$

Inequalities (3.9) and (3.10) jointly imply that

$$C_1 > C_3 > C_2. \quad (3.11)$$

Let  $V_{dd1}$ ,  $V_{dd2}$ , and  $V_{dd3}$  represent the supply voltages at which the paths with capacitances  $C_1$ ,  $C_2$ , and  $C_3$ , respectively, have the same delay  $\tau_{cp}$ . Then,

$$V_{dd1} > V_{dd3} > V_{dd2}. \quad (3.12)$$

If circuit I is operated at  $V_{dd2}$  with a clock having period  $\tau_{cp}$ , then soft errors appear whenever the path with  $C_1$  is excited, i.e., with probability  $p$ . However, using (3.10) and (3.12), it can be shown that

$$\begin{aligned} (C_1 + C_2)V_{dd2}^2 &< 2C_3V_{dd3}^2 \\ \Rightarrow P_{Ivos} &< P_{II} \\ \Rightarrow P_{Ivos}\tau_{cp} &< P_{II}\tau_{cp} \end{aligned} \quad (3.13)$$

Now, if an ANT scheme, which provides adequate error control when error probability is  $p$ , can be designed for circuit I, then voltage overscaling circuit I to  $V_{dd2}$  makes it better, in terms of power-delay product, than circuit II.

If, instead,  $P_I > P_{II}$ , then  $C_1 + C_2 > 2C_3$  and it is not possible to infer whether VOS is useful. In this case, whether circuit I can be made better than circuit II will depend on the actual values of  $C_1$ ,  $C_2$  and  $C_3$ .

To summarize, the conditions under which VOS is useful are:

- (1) At same  $V_{dd}$ , delay imbalanced circuit consumes less power than inherently delay balanced circuit, i.e.,  $P_I < P_{II}$ .

(2) ANT scheme provides adequate error control when error probability is  $p$ .

In Sections 3.3 and 3.4, we show that both the conditions are satisfied by typical delay imbalanced adders and multipliers.

### 3.3 Adders with Voltage Overscaling

The following types of adders [16] are investigated:

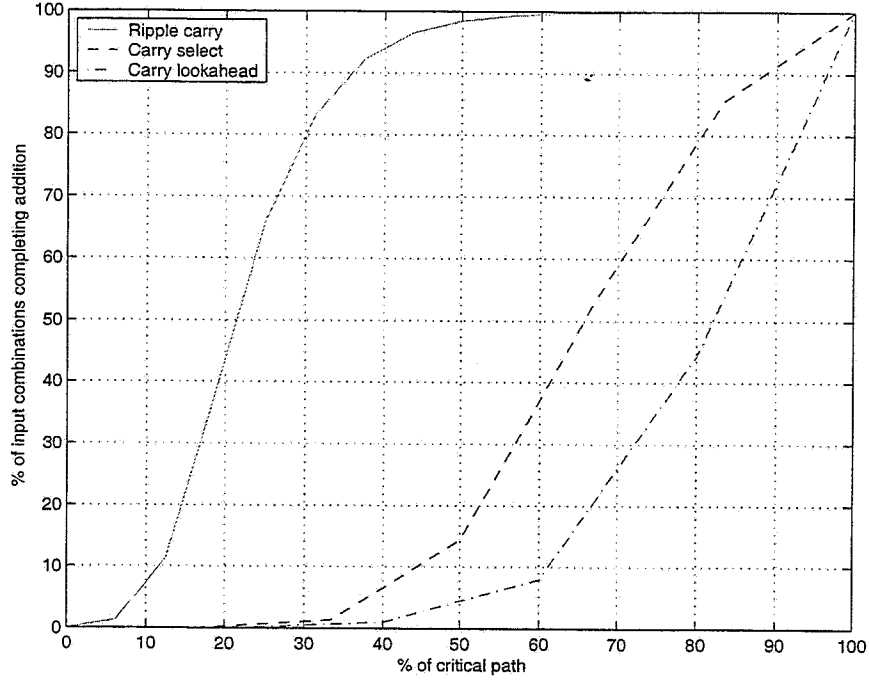
- Ripple carry adder
- Carry select adder
- Carry lookahead adder

All the adders used are 16-bit wide. The critical path delays for the adders are obtained in terms of full adder delays [16]. Table 3.1 shows the delay of the adders relative to the delay of the ripple carry adder. The carry lookahead adder has significantly less critical path delay than the ripple carry adder due to faster carry propagation.

The power dissipation for the adders is obtained from [4]. Table 3.1 shows the relative power consumption for each of the adders with respect to the ripple carry adder. Both

**Table 3.1** Power and delay for 16-bit adders from [4] relative to ripple carry adder.

Adder type	Power	Delay	Power $\times$ Delay
Ripple carry adder (RCA)	1.00	1.00	1.00
Carry select adder (CSA)	1.85	0.39	0.73
Carry lookahead adder (CLA)	1.46	0.28	0.41

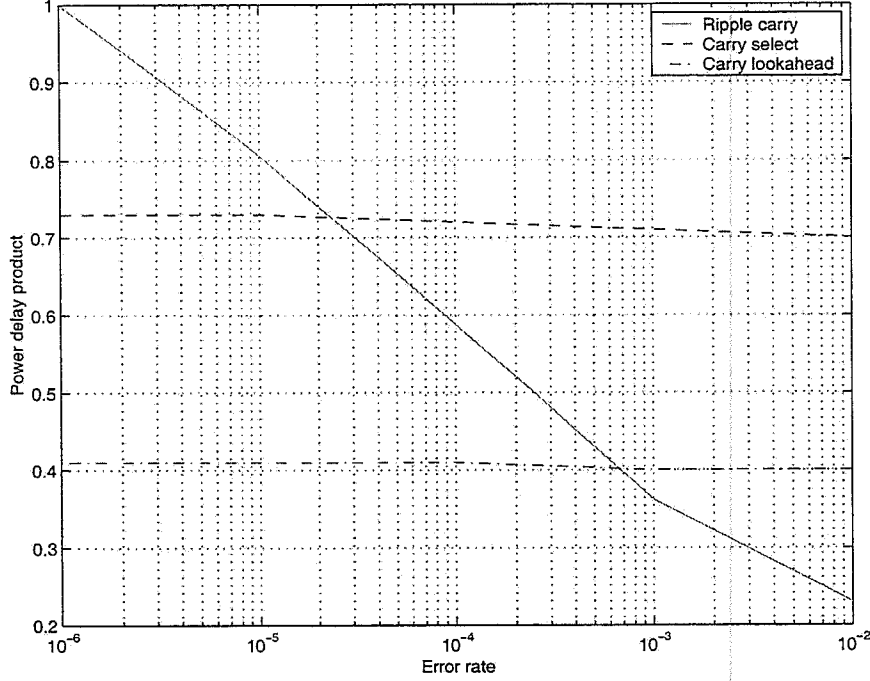


**Figure 3.3** Cumulative path delay distribution of 16-bit adders.

the carry select adder and the carry lookahead adder consume more power due to the additional fast carry propagation circuits. Table 3.1 also shows the power-delay product. It is seen that the carry lookahead adder is, by far, the best adder among the adders considered.

The cumulative path delay distribution for each of the adders is obtained by simulation with a large number of input combinations. With voltage overscaling, the critical path delay  $T_{cp}$  increases as governed by (1.2) and becomes greater than the clock period  $T_s$ . Figure 3.3 plots the total number of input combinations that are able to complete the addition within the given clock period against  $100 \times (T_s/T_{cp})$ .

Figure 3.3 enables us to obtain the frequency of errors or the error rate if the VOS factor is known. Further, from (1.1), we can obtain the power dissipation for a given  $V_{dd}$ .



**Figure 3.4** Power-delay product vs. error rate.

Figure 3.4 plots the power-delay product against the error rates due to VOS. It is seen that the carry select adder and the carry lookahead adder are not suitable for voltage overscaling as the error rates increase rapidly with VOS. However, the ripple carry adder is amenable to VOS and performs better than the carry lookahead adder if the error rate  $> 7 \times 10^{-4}$ . Therefore, if we have an ANT scheme at the algorithmic level that can tolerate an error rate greater than  $7 \times 10^{-4}$ , then the ripple carry adder is the most energy-efficient adder for the application.

### 3.4 Multipliers with Voltage Overscaling

The following multipliers are investigated:

- Baugh-Wooley multiplier [16]

- Booth-encoded carry save array multiplier (BCSA) [17]
- Booth-encoded carry save array with carry lookahead adder as vector merging adder (BCSA+CLA)

The BCSA has a ripple carry adder as the vector merging adder while the BCSA+CLA is delay optimized by using a carry lookahead adder as the vector merging adder. We consider  $16 \times 16$  multipliers in this thesis. The power dissipation is estimated using the gate level simulator MED [13]. The critical path delay is obtained in terms of full adder delays. The power and delay for the multipliers relative to the Baugh-Wooley multiplier are shown in Table 3.2. The BCSA+CLA multiplier has the lowest power-delay product and, hence, is the fastest, most efficient multiplier among the multipliers considered.

The cumulative path delay distributions for the multipliers are plotted in Figure 3.5. It is seen that BCSA is the most amenable multiplier to VOS followed by Baugh-Wooley and BCSA+CLA multipliers. The corresponding plot of power-delay product vs. error rate is shown in Figure 3.6. It is seen that the power-delay product of BCSA drops rapidly as the error rate is increased. For error rates greater than  $4 \times 10^{-5}$ , BCSA has lower power-delay product than BCSA+CLA. The Baugh-Wooley multiplier does not perform as well as BCSA because it has a higher percentage of longer paths. Now, we

**Table 3.2** Power and delay for  $16 \times 16$  multipliers relative to Baugh-Wooley multiplier.

Multiplier type	Power	Delay	Power $\times$ Delay
Baugh-Wooley multiplier	1.00	1.00	1.00
Booth-encoded carry save array (BCSA)	0.85	1.03	0.86
BCSA with carry lookahead adder (BCSA+CLA)	1.16	0.44	0.51

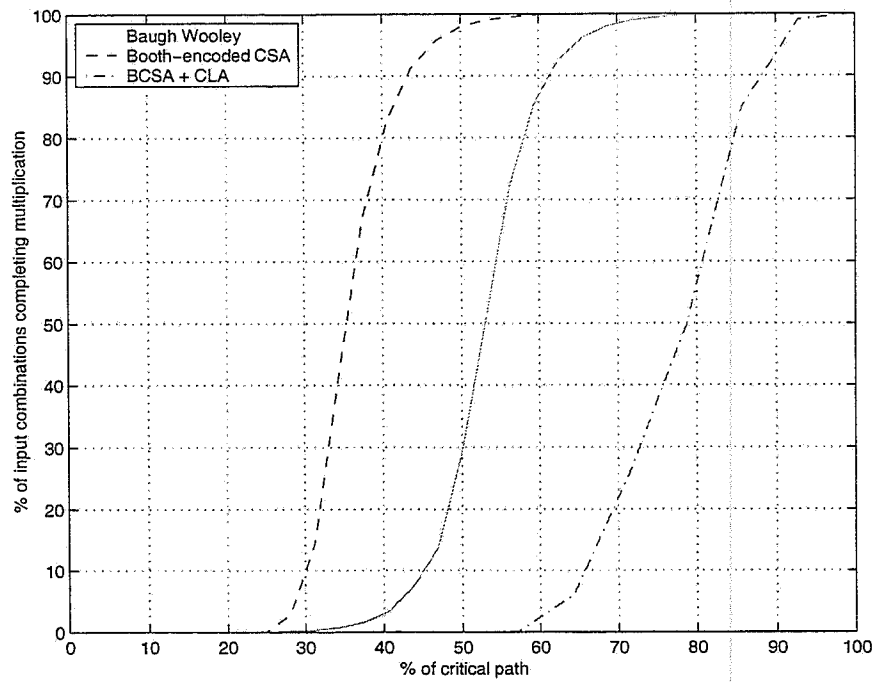


Figure 3.5 Cumulative path delay distribution of  $16 \times 16$  multipliers.

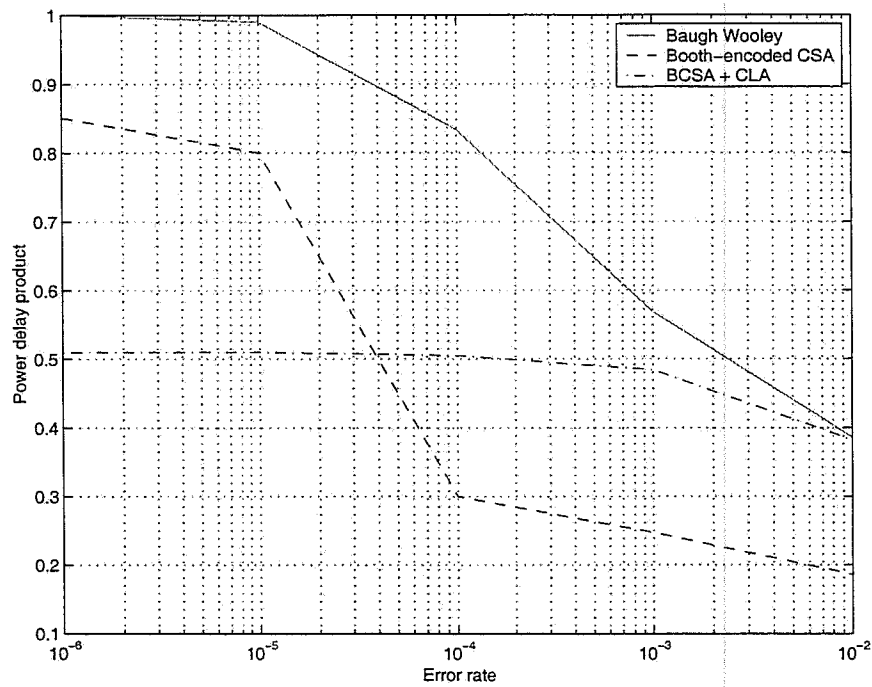


Figure 3.6 Power-delay product vs. error rate.

utilize this result to further improve the energy efficiency of the FFT processor described in Chapter 2.

We showed that up to 44% power savings can be obtained by using RPR technique in Baugh-Wooley multipliers of an FFT butterfly. As we have seen from Figure 3.6, BCSA performs better than Baugh-Wooley multipliers in the context of soft DSP. Therefore, we replace the Baugh-Wooley multipliers in the FFT processor by BCSA multipliers. Simulation results show that an additional 18% power savings can be obtained.

## 3.5 Conclusions

In this chapter, we have derived the conditions under which VOS provides higher power savings than delay balancing, and we have shown that typical arithmetic units satisfy these conditions. We have compared the performance of several adders and multipliers in the context of soft DSP. We find that if certain error rates can be tolerated by algorithmic noise tolerance techniques, then the ripple carry adder performs better than the carry lookahead adder and the Booth-encoded carry save array multiplier performs better than the delay optimized Booth-encoded carry save array multiplier with carry lookahead adder. The main conclusion of this chapter is that if we have a sufficiently powerful ANT scheme, then the power-delay product alone cannot be used to choose arithmetic unit architectures, but the error rates have to be considered. The translation of error control capability of the ANT scheme to the required reliability at the arithmetic unit level is an open problem that needs to be addressed.



# CHAPTER 4

## CONCLUSIONS AND FUTURE WORK

### 4.1 Summary

The contributions of this thesis are as follows:

- We have introduced a novel algorithmic noise tolerance scheme referred to as reduced precision redundancy.
- Reduced precision redundancy provides 44% power savings in the butterfly functional unit of an FFT processor.
- We have derived conditions under which VOS provides higher power savings than delay balancing.
- We have shown that power-delay product along with error rates provides a suitable metric for comparison of arithmetic units in the context of soft DSP.
- The highly delay imbalanced ripple carry adder has the best performance when error rates greater than  $7 \times 10^{-4}$  can be tolerated.
- The Booth-encoded carry save array multiplier has the best performance when error rates greater than  $4 \times 10^{-5}$  can be tolerated.

- Reduced precision redundancy provides 18% additional power savings when the Baugh-Wooley multiplier is replaced by the Booth-encoded carry save array multiplier.

## 4.2 Future Work

Reduced precision redundancy has been used in frequency selective digital filters [9] to achieve significant power savings. This technique is applicable to both narrowband and wideband filters, unlike prediction and adaptive error cancellation techniques that are applicable only to narrowband and wideband filters, respectively. Therefore, RPR has the potential to be adapted to several other DSP algorithms. Designing novel DSP architectures using RPR can be the focus of future research.

Algorithmic noise tolerance techniques mitigate the effect of hardware errors on the system performance. In addition to VOS, there are other sources of intermittent errors such as process variations, deep submicron (DSM) noise, and delay faults. ANT techniques can be used to combat process variations and, hence, improve process yield. They can be used to design robust systems by reducing the effect of DSM noise and delay faults.

## REFERENCES

- [1] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. New Jersey: Prentice-Hall, 1996.
- [2] R. Hegde and N. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *Proceedings of International Symposium on Low Power Electronics and Design*, 1999, pp. 30-35.
- [3] L. Wang and N. Shanbhag, "Low-power signal processing via error-cancellation," in *IEEE Workshop on Signal Processing Systems*, 2000, pp. 553-562.
- [4] T. K. Callaway and E. E. Swartzlander Jr., "Estimating the power consumption of CMOS adders," in *Proceedings of 11th Symposium on Computer Arithmetic*, 1993, pp. 210-216.
- [5] T. K. Callaway and E. E. Swartzlander Jr., "Power-delay characteristics of CMOS multipliers" in *Proceedings of 13th IEEE Symposium on Computer Arithmetic*, 1997, pp. 26-32.
- [6] N. Weste and D. J. Skellern, "VLSI for OFDM," *IEEE Communications Magazine*, vol. 36, pp. 127-131, Oct. 1998.
- [7] S. Hong, S. Kim, M. C. Papaefthymiou, and W. E. Stark, "Power-complexity analysis of pipelined VLSI FFT architectures for low energy wireless communication applications," in *Proceedings of 42nd Midwest Symposium on Circuits and Systems*, 2000, pp. 313-316.
- [8] B. M. Baas, "A low-power, high-performance 1024-point FFT processor," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 380-387, March 1999.
- [9] B. Shim and N. R. Shanbhag, "Reduced precision redundancy for low-power digital filtering," in *Proceedings of Thirty Fifth Asilomar Conference on Signals, Systems and Computers*, 2001, pp. 148-152.
- [10] S. R. Sridhara and N. R. Shanbhag, "Low-power FFT via reduced precision redundancy," in *IEEE Workshop on Signal Processing Systems*, 2001, pp. 117-124.
- [11] P. Larsson and C. Nicol, "Self-adjusting bit-precision for low power digital filters," in *1997 Symposium on VLSI Circuits*, 1997, pp. 123-124.
- [12] R. Amirtharajah, T. Xanthopoulos, and A. Chandrakasan, "Power scalable processing using distributed arithmetic," in *Proceedings of International Symposium on Low Power Electronics and Design*, 1999, pp. 170-175.

- [13] F. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Transactions on VLSI Systems*, vol. 2, pp. 446-455, Dec. 1994.
- [14] J. Leijten, J. van Meerbergen, and J. Jess, "Analysis and reduction of glitches in synchronous networks," in *Proceedings of European Design and Test Conference*, 1995, pp. 398-403.
- [15] T. Sakuta, W. Lee, and P. T. Balsara, "Delay balanced multipliers for low power/low voltage DSP core," in *Proceedings of IEEE Symposium on Low Power Electronics*, 1995, pp. 36-37.
- [16] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Design*. New York: Oxford University Press, 2000.
- [17] Y. Zhan, L. Wasserman, and A. N. Willson Jr., "A painless way to reduce power dissipation by over 18% in Booth-encoded carry-save array multipliers for DSP," in *IEEE Workshop on Signal Processing Systems*, 2000, pp. 571-580.

