# Novel Design Techniques to Reduce Simultaneous Switching Noise in VLSI Packaging

by

**Brock J. LaMeres**

B.S., Montana State University, 1998

M.S., University of Colorado, 2001

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

2005

This thesis entitled:
Novel Design Techniques to Reduce Simultaneous Switching Noise in VLSI Packaging
written by Brock J. LaMeres
has been approved for the Department of Electrical and Computer Engineering

Prof. Sunil P. Khatri

Prof. Fabio Somenzi

Prof. T.S Kalkur

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

LaMeres, Brock J. (Ph.D., Electrical Engineering)

Novel Design Techniques to Reduce Simultaneous Switching Noise in VLSI Packaging

Thesis directed by Prof. Sunil P. Khatri

Advances in VLSI design and fabrication technologies have led to a dramatic increase in the on-chip performance of integrated circuits. The transistor delay in an integrated circuit is no longer the bottleneck to system performance as it has historically been in past decades. System performance is now limited by the electrical parasitics of the packaging interconnect. Noise sources such as supply bounce, signal coupling, and reflections all result in reduced performance. These factors arise due to the parasitic inductance and capacitance of the packaging interconnect. While advanced packaging can aid in reducing the parasitics, the cost and time associated with the design of a new package is often not suited for the majority of VLSI designs. This work presents techniques to model and improve performance the performance of VLSI designs without moving toward advanced packaging. A single, unified mathematical framework is presented that predicts the performance of a given package depending on the package parasitics and bus configuration used. The performance model is shown to be accurate to within 10% of analog simulator results which are much more computationally expensive. Using information about the package, a methodology is presented to select the most cost-effective bus width for a given package. In addition, techniques are presented to encode off-chip data so as to avoid the switching patterns that lead to increased noise. The reduced noise level that results from encoding the off-chip data translates into increased bus performance even after accounting for the encoder overhead. Performance improvements of up to 225% are reported using the encoding techniques. Finally, a compensation technique is presented that matches the impedance of the package interconnect to the impedance of the system, resulting in reduced reflected noise. The compensation technique is shown to reduce reflected noise as much as 400% for broadband frequencies up to 3GHz. The techniques presented in this work are described in general terms so as not to limit the approach to any particular technology. At the same time, the techniques are validated using existing technologies to prove their effectiveness.

## Acknowledgements

The author would like to thank the Design Validation Division of Agilent Technologies in Colorado Springs for their support in this work. Agilent Technologies is the world leader in development of test and measurement equipment. Agilent Technologies provided the necessary funding, instrumentation, EDA tools, and hardware used in the development and analysis of the techniques in this thesis. The author would also like to thank Xilinx Corporation for providing the FPGA devices and design methodologies necessary for prototyping the techniques in this thesis and evaluating their feasibility. Finally, the author would like to express his deepest gratitude to Professor Sunil P. Khatri who directed this research. Dr. Khatri's in-depth knowledge of VLSI CAD, amazing work ethic, and extreme flexibility were all key in the completion of this thesis.

# Contents

**Chapter**

# Tables

**Table**

# Figures

**Figure**

# Chapter 1

# Introduction

Integrated circuit (IC) performance has increased at an exponential rate since the first patent was issued to Robert Noyce of Fairchild Semiconductor in 1961 [88]. Since the advent of the IC, the number of transistors on an integrated circuit has roughly doubled every 18 months. This trend, also known as Moore's Law [63], has been consistently met over the past 45 years. This increase in system performance on the IC is predicted by the International Technology Roadmap for Semiconductors (ITRS) [89] to continue to follow Moore's Law into the foreseeable future. Historically, the gate delay of the digital circuitry has limited IC performance [67]; however, over the past decade the bottleneck of system performance has shifted from the gate delay of the integrated circuit to the package parasitics [63]. While package performance has steadily improved, it has not kept pace with the increases in integrated circuit performance. Package performance is predicted by the ITRS to only double over the next decade. This imbalance in performance expectations between the IC and the package is a major concern for system designers and for the continuation of Moore's Law.

The limitation in package performance comes from the parasitic inductance and capacitance in the electrical interconnect [63, 64, 66]. Package interconnect has historically been designed to meet mechanical, thermal, and cost objectives. In the past, the electrical performance of the interconnect was not an issue since it caused a relatively small performance degradation relative to the gate delay of the IC transistors [67]; however, with the dramatic improvement in transistor gate delay, package performance is now the leading determinant of the overall system performance.

## 1.1 The Role of IC Packaging

The purpose of an IC package is to electrically and mechanically connect the integrated circuit substrate to the system substrate. The most common substrate used in Very Large Scale Integrated (VLSI) circuitry is silicon [67]. Silicon offers a host of electrical advantages that allow the implementation of large numbers of transistors in a cost-effective and easy-to-manufacture manner. In VLSI silicon designs, conductors are most commonly implemented using Polysilicon ($p - Si$), Aluminum ($Al$), and Copper ($Cu$). Insulating layers in VLSI designs are typically implemented using Silicon Oxide ($SiO_2$) and Silicon Nitride ($Si_3N_4$). The photolithography and deposition processes used in IC fabrication allow extremely small feature sizes to be printed on the substrate. Currently, feature sizes as small as $65nm$ are being successfully implemented using VLSI processes [27].

System level interconnect in digital systems is typically constructed using printed circuit board (PCB) technology [64]. PCBs typically use Copper as their conducting layer. Insulating layers are implemented using dielectric materials such as FR4, Nelco-13, $GETEK^{®}$, and Teflon. PCBs are constructed using a lamination process. Modern lamination processes are capable of producing minimum feature sizes between $4\mu m$ and $100\mu m$.

An IC package serves many purposes. The first is to electrically connect the leads on the IC to the corresponding leads on the system PCB. Since the feature sizes on the IC are much smaller than the feature sizes on the PCB, the IC cannot be mounted directly to the system PCB. The package serves as a *density translator* for the electrical signals that will connect extremely fine-pitch signals on the IC to coarser-pitched signals on the PCB.

The second purpose of the package is to protect the substrate of the IC. The IC substrate consists of a very thin crystal of silicon that has transistors and interconnect deposited on it. This substrate is brittle and can be easily fractured, leading to circuit failure [63]. The package serves as a mechanical barrier and hermetic seal between the silicon substrate and the outside environment. In this manner the substrate is isolated from contamination and humidity, which can also lead to IC failures. In addition, the package absorbs thermal expansion mismatches between the silicon substrate and the system PCB. If the thermal mismatches were completely absorbed by the silicon substrate, it would lead to stress fractures and circuit failure.

The third major purpose of IC packaging is to remove heat from the IC substrate. Modern ICs consist of millions of transistors that each consume current and dissipate power. The cumulative power dissipation of the transistors lead to the generation of extreme amounts of heat in a relatively small area. This high thermal density adversely effects circuit performance by increasing the gate delay and shortening the life of the devices on the IC [67]. The typical range of operating junction temperatures for modern VLSI designs is between 80°C and 120°C on the silicon substrate [67, 92]; however, modern microprocessors are projected by the ITRS to surpass the 100 Watt power dissipation mark within the next couple of years [92]. All this power is dissipated by substrates that range from $5mm^2$ to $20mm^2$ in size [20, 82]. The package serves as a heat transfer system that removes the heat from the IC substrate. The heat is delivered to a package surface from which it can ultimately be absorbed by ambient air. By doing this, the package can keep the circuitry on the IC within the acceptable range of junction temperatures. This leads to consistent and predictable performance of the devices on the IC substrate.

Historically, the thermal and mechanical aspects of the package were the main focus of the package design [63]. Since the gate delay of the devices on the IC were the largest source of performance limitation, the electrical interconnect in the package was optimized for mechanical and thermal purposes. This approach has been successful for the past 40 years. Only recently has the electrical interconnect in the package become the bottleneck to system performance. This has occurred due to the dramatic advances in integrated circuit technology that has reduced the gate delay of the on-chip devices to the point where they no longer limit system performance. This shift as resulted in the package (which was originally optimized for mechanical and thermal rather than electrical consideration) becoming the major limiting factor in system level electrical performance.

The electrical interconnect limits performance due to the parasitic resistance, inductance and capacitance that is present in the interconnect structure [64, 65, 66]. Since the interconnect structures were originally created for mechanical properties, their parasitic inductance and capacitance can be considerable relative to the datarates that are currently being transmitted through the package. Package design is a slow and expensive process that has not been able to keep pace with the performance increase in core IC technology. This makes the task of altering the interconnect structures within the package (to increase electrical performance) very difficult. The majority of VLSI designs today cannot afford the cost and time associated with re-engineering a package [92]. This means that in most cases designers must live with the performance limitations in the package. Any technique that can increase system performance without changing the package design is of tremendous value to the VLSI community.

## 1.2 Noise Sources in Packaging

Electrically the package has two objectives. The first is to deliver power to the devices on the integrated circuit. The power resides on the system PCB and must be conducted through the package to the devices on-chip. The second objective is to connect signals on the IC to the signal paths on the system PCB. In both of these cases electrical current and voltage are carried using interconnect structures within the package that were not originally optimized for electrical performance. The parasitic resistance, inductance and capacitance of these structures limits how efficiently the power can be delivered and how fast the signals can be transmitted.

### 1.2.1 Inductive Supply Bounce

When inductance is present in the path that carries power to the devices on the IC, a voltage will form across this inductance based on the relationship:

$$V_L = L \cdot (\frac{di}{dt}) \tag{1.1}$$

In the case of power delivery, the inductance in Equation 1.1 is the parasitic inductance in the interconnect structure that carries current to and from the devices on-chip. $(\frac{di}{dt})$ is the rate of change of the current that is carried through the interconnect. As more and more transistors are integrated on-chip, the total amount of current that is being carried through the interconnect increases. In addition, as the operating frequency of the digital circuits increase, the time in which this current is delivered is reduced. This results in significant increases in the rate of change of current $(\frac{di}{dt})$ through the package interconnect. These two trends are being driven by the increase in IC technology; however, the inductance $(L)$ present in the interconnect is driven by package technology, which is not improving at a rate which would keep $\frac{di}{dt}$ unchanged [89].

When a voltage is induced across the power supply interconnect, the voltage on the chip will be different from the voltage on the system PCB. When delivering a power supply voltage $(V_{DD})$ to a Complimentary Metal Oxide Semiconductor (CMOS) device, there will be a voltage drop between the system PCB and the integrated circuit. This phenomenon is called *supply bounce*. Similarly, when the ground return current for a CMOS device traverses an inductive interconnect, a voltage will be induced across the inductive interconnect (causing the voltage at the ground pin of the device $(V_{SS})$ to be different from the system PCB). This phenomenon is called *ground bounce*. Supply

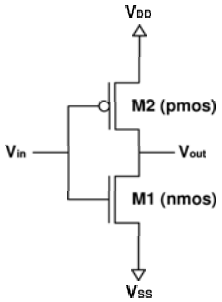


Figure 1.1: Ideal CMOS Inverter Circuit

and ground bounce directly effect the performance of the digital device by increasing its gate delay and causing inadvertent switching [67]. Figure 1.1 shows an ideal CMOS inverter. In this circuit, the on-chip power supplies $V_{DD}$ and $V_{SS}$ are assumed to be the same as on the system PCB. Figure 1.2 shows a CMOS inverter but models the inductance in the supply and ground path. In this case, the voltage from Equation 1.1 (induced across the inductive interconnect) results in a voltage difference between the supply nodes of the device ($V_{DD-IC}$, $V_{SS-IC}$) and the supply voltages on the system PCB ($V_{DD-System}$, $V_{SS-System}$).

In modern VLSI designs, between 25% to 50% of the total $(\frac{di}{dt})$ can be consumed in the off-chip driver circuitry and constitutes the largest single source of current consumption [89, 92]; however, $V_{DD}$ and $V_{SS}$ pins are often shared across multiple signal pins that are drawing and returning current for their off-chip driver circuitry. Supply pins are shared to reduce the cost of the package since cost is proportional to the size of the package. Therefore, in order to accurately compute the magnitude of the supply and ground bounce we must modify Equation 1.1 to consider all of the signal pins that are sharing a supply pin to source or return current. For supply bounce the total number of signals (which share a given $V_{DD}$ pin) that are switching from a 0 to a 1 must be considered. Equation 1.2 expresses the total amount of power supply bounce considering all signals that are drawing current through the supply pin inductance. For ground

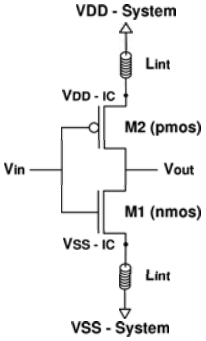


Figure 1.2: CMOS Inverter Circuit with Supply Inductance

bounce, the total number of signals that are switching from a 1 to a 0 and are sharing a particular $V_{SS}$ interconnect to return current must be considered. Equation 1.3 expresses the total amount of ground bounce considering all signals that are returning current through the ground pin inductance.

$$V_{Supply-Bounce} = L_{(V_{DD})} \cdot \sum_{k=0}^{n_{(0 \to 1)}} (\frac{di_k}{dt}) \tag{1.2}$$

$$V_{Ground-Bounce} = L_{(V_{SS})} \cdot \sum_{k=0}^{n_{(1 \to 0)}} (\frac{di_k}{dt}) \tag{1.3}$$

### 1.2.2 Inductive Signal Coupling

When a signal traverses an inductive interconnect in a package it inductively couples to neighboring signals. This is due to the mutual inductive coupling of the magnetic fields for any two current carrying conductors. The magnitude of the mutual inductive coupling voltage between inductive interconnects is governed by the expression:

$$V_M^j = M_{1k} \cdot (\frac{di_k}{dt}) \tag{1.4}$$

$M_{1k}$ is the mutual inductance between conductor $j$ and conductor $k$. This relationship can also be expressed in terms of the mutual inductive coupling coefficient

$K_{ik}$. $K_{1k}$ is a dimensionless quantity. Equation 1.5 describes the relationship between $M_{1k}$ and $K_{1k}$. In this expression, $L_j$ and $L_k$ are the inductances of the two inductively coupled conductors.

$$K_{1k} = \frac{M_{1k}}{[\sqrt{L_j \cdot L_k}]} \tag{1.5}$$

For this work we use the term *victim* for the signal of interest (on which the mutual inductive voltage is induced). We use the term *aggressor* for any neighboring signal that switches and causes a voltage to be induced on the *victim*. Mutual inductive coupling has a cumulative effect in IC packaging. The mutually induced voltage is also dependent on the polarity of the $(\frac{di}{dt})$ in the aggressor's signal path. This means that the total mutual inductive voltage on the victim is the sum of all aggressors' mutually coupled voltage. This unique circumstance creates a situation in which the switching patterns of neighboring signals heavily influence the behavior of the coupled noise on the victim signal. In an extreme case, the aggressors' mutually coupled voltage (of equal and opposite magnitudes) can cancel out and have no effect on the victim. Equation 1.6 describes the total contribution of mutually coupled voltages on a victim signal.

$$V_M^j = \sum_{k=1}^{n} M_{1k} \cdot (\frac{di_k}{dt}) \tag{1.6}$$

We define the term *Glitch* to describe the situation in which the victim signal is not transitioning and neighboring signal pins are coupling voltage onto it. The *Glitch* voltage caused by switching aggressor signals creates an unwatched voltage on the victim signal. This unwanted voltage can lead to inadvertent switching of digital circuitry that use the victim line as an input.

When the victim signal is transitioning, we use the term *Edge Degradation* to describe the effect of voltage that is coupled from neighboring aggressor signals. Since mutual inductive coupling is cumulative, coupled voltage from neighboring signals can

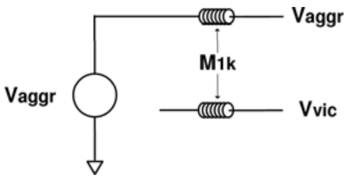


Figure 1.3: Circuit Description of Inductive Signal Coupling

have the effect of aiding, hindering, or not affecting the transition on the victim signal. Figure 1.3 shows the circuit for inductively coupled signal lines in a VLSI package.

### 1.2.3 Capacitive Bandwidth Limiting

When capacitance is present along the path that carries signals to and from the devices on the IC, the frequency at which the signal can switch will be reduced. This capacitance will form a low pass RC filter that will prevent frequency components that are above the filters roll-off frequency from passing through the interconnect. This filtering effect will limit the risetime that the package will allow to pass and in turn reduce the overall datarate and performance of the system. The filter that the package creates in the signal path has a standard RC low pass response. The capacitance in the filter comes from the parasitic capacitance present in the package interconnect. The impedance in the filter comes from the characteristic impedance of the system in which the package is placed. Equation 1.7 gives the 10% to 90% risetime of a single-pole RC filter.

$$t_{rise} = 2.2 \cdot RC = 2.2 \cdot Z_0 \cdot C_{int} \tag{1.7}$$

In this expression $Z_0$ is the characteristic impedance of the system in which the package is placed. $C_{int}$ represents the total capacitance that the package interconnect presents. $C_{int}$ will depend on the parasitic capacitance of the interconnect to ground ($C_0$) in addition to the parasitic capacitance to neighboring signals ($C_{1j}$). The total number of neighboring signals that contribute capacitance to $C_{int}$ must be accounted for. Equation 1.8 gives the total amount of capacitance that an individual package interconnect possesses.

$$C_{total}^{j} = C_0 + \sum_{k=1}^{n} C_{1k} \tag{1.8}$$

The logic level of neighboring signals will affect the amount of capacitance that a victim signal will have to charge or discharge ($C_{1k}$). $C_{1k}$ is defined as the capacitance between two conductors when one conductor is held at zero potential. Capacitance is defined as $C = Q/V$. This means that the variable voltage that is present on neighboring signals will change the effective capacitance of the victim signal. When a neighboring signal has the same voltage change as the victim signal (i.e., they are transitioning), then the total coupling capacitance between signals will be zero because there is no net charge difference between the two conductors. When a neighboring signal has a static voltage value while transitioning, then the total coupling capacitance between the signals will be $C_{1k}$. When a neighboring signal transitions in the opposite direction as the victim signal, then the total effective coupling capacitance will be $2 \cdot C_{1k}$ because the voltage excursion that the capacitor undergoes is $2 \cdot V_{DD}$ [1] . This situation means that the risetime of a signal will change depending on the logic levels and transitions present on neighboring signals. Figure 1.4 shows the circuit diagram for a signal with capacitive bandwidth limitation.

### 1.2.4 Capacitive Signal Coupling

When a signal traverses a capacitive interconnect in a package, it is electrically modified due to coupling from neighboring signals. This is due to the mutual capacitive coupling of the electric fields for any two charge bearing surfaces. The magnitude of the

[1] the voltage across the capacitor is -$V_{DD}$ before the transition and +$V_{DD}$ after the transition (or vice versa).

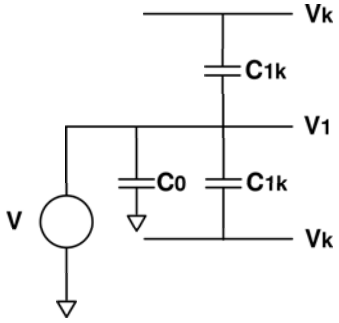


Figure 1.4: Circuit Description of Capacitive Bandwidth Limiting

mutual capacitive coupling voltage between capacitive interconnects will depend on the coupling capacitance between the conductors ($C_{1k}$) in addition to the self-capacitance ($C_0$) of the victim line. The sum of the coupling capacitance, along with the self-capacitance of the victim line will form a capacitive voltage divider between the victim and aggressor lines. Equation 1.9 describes the voltage magnitude that will be coupled onto a victim line ($V_{vic}$) from a voltage change on an aggressor line ($V_{aggr}$).

$$V_{vic} = (\frac{C_{1k}}{C_{1k} + C_0}) \cdot (\Delta V_{aggr}^{k}) \tag{1.9}$$

Capacitive coupling can cause unwanted switching of digital circuitry, which limits system performance. As in the case of mutual inductive coupling, the capacitive coupling has a cumulative effect. Glitches on a victim line that occur due to the transitions on neighboring signals can cause unwanted switching of digital circuitry. Also, the cumulative nature of the coupling will lead to an effect that can either aid, hinder, or not effect the victim signal's risetime. Equation 1.10 models the cumulative nature of the capacitive coupling betweens signals. Figure 1.5 shows the circuit for capacitively coupled signal lines in a VLSI package.

$$V_{vic} = \sum_{k=1}^{n} (\frac{C_{1k}}{C_{1k} + C_0}) \cdot (\Delta V_{aggr}^{k}) \tag{1.10}$$

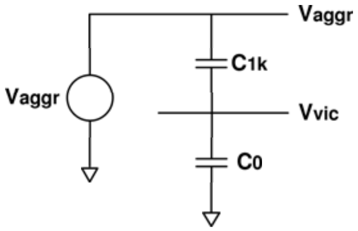


Figure 1.5: Circuit Description of Capacitive Signal Coupling

### 1.2.5 Impedance Discontinuities

Another source of noise in the package interconnect is reflected energy due to impedance discontinuities. As the risetimes of off-chip driver circuits increase, conductors must be considered as distributed elements. Typically when the interconnect being driven has an electrical length [2] that is longer than 20% of the risetime of the source, then the conductor must be treated as a distributed element and modeled as a transmission line [66]. Current off-chip risetimes are entering into the sub 100ps era. For standard packaging technology this means that a structure longer than 1 to 2 cm must be treated as a distributed element. For these electrical lengths almost all modern packages require the use of transmission line theory for accurate modeling of the package interconnect.

When IC packaging interconnect is treated as a distributed element, impedance matching becomes critical. When impedance mismatches are present in the package, reflections and risetime degradation will occur. The reflections can lead to unwanted switching of digital circuitry and intersymbol interference (ISI). In addition, risetime degradation will significantly limit system performance. The characteristic impedance of a loss-less structure is given by:

$$Z_0 = \sqrt{\frac{L}{C}} \tag{1.11}$$

[2] propagation delay of the structure.

In this expression the L and C are the inductance and capacitance of a given structure. When considering the impedance of the package, the switching patterns of neighboring signals will alter the inductance and capacitance in Equation 1.12. Switching signal pins will add or subtract additional amounts of mutual inductance and capacitance to the victim signal. These additional parasitics must be considered in the evaluation of characteristic impedance. This changes the impedance within the package to:

$$Z_0' = \sqrt{\frac{L \pm \sum_1^k M_{1k}}{C \pm \sum_1^k C_{1k}}} \tag{1.12}$$

When a wave front is traveling on a conductor with a particular characteristic impedance ($Z_0$) and encounters a region with a different characteristic impedance ($Z_L$), reflections will occur. Figure 1.6 shows the circuit diagram for a distributed transmission line.

$\Gamma$ is the reflection coefficient which represents the magnitude of the reflected voltage (as a fraction of the incident voltage) and is defined as:

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} \tag{1.13}$$

$\Gamma$ can be used to find the reflected and transmitted voltage of an incident wave front. Equation 1.14 defines the amount of reflected voltage due to an impedance

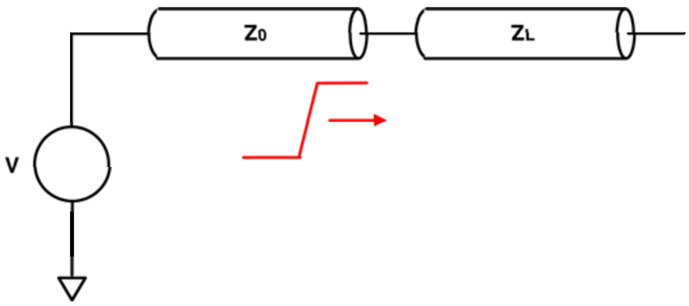


Figure 1.6: Circuit Description of a Distributed Transmission Line

mismatch. This Equation illustrates the relationship between an impedance mismatch and the magnitude of the reflected voltage. Note that the reflected voltage can have either a positive or negative amplitude depending on whether $Z_L$ is greater than or less than $Z_0$.

$$V_{reflected} = V_{incident} \cdot \Gamma \tag{1.14}$$

Equation 1.15 defines the amount of transmitted voltage when encountering an impedance mismatch. In this case the magnitude of the reflected voltage will be subtracted from the forward going wave front. This translates into a slower risetime that is propagated through the impedance mismatch. Since risetimes are directly proportional to the datarate that a package can achieve, any reduction in the risetime due to reflected energy will directly effect system performance.

$$V_{transmitted} = V_{incident} \cdot (1 - \Gamma) \tag{1.15}$$

In digital systems typically the system PCB uses a characteristic impedance close to $50\Omega$. The interconnect structures within the IC package typically do not match to the system PCB impedance and will lead to reflections. When the impedance of a package structure is higher than the characteristic impedance of the system ($50\Omega$), it is considered an inductive interconnect. It is considered to be inductive because in Equation 1.12 a higher than optimal inductance will lead to a higher impedance. In the same manner, when the impedance of a package structure is lower than the characteristic impedance of the system, it is considered a capacitive interconnect. Since current packages were originally designed for mechanical considerations, getting a desired impedance is difficult without using expensive and non-standard processes.

As in the case with capacitive bandwidth limitation, switching neighboring pins will alter the net coupling inductance and capacitance that the victim pin experiences.

The polarity and quantity of switching signals will alter the effective L and C quantities in Equation 1.12 and in turn the magnitude of $\Gamma$.

## 1.3 Performance Modeling and Proposed Techniques

All of the noise sources described in Section 1.2 can be reduced through aggressive package design. We have seen success in Ball Grid Array (BGA) [80] and Flip-chip packaging [81] technologies that have been able to reduce the electrical parasitics of the interconnect structures within the IC package; however, aggressive package design is often too expensive and unpractical for the majority of VLSI designs. In addition, package design is a slow, evolutionary process that is not predicted to keep pace with the performance increases on the IC. As such, any techniques that can assist in improving performance without moving toward advanced packaging are of considerable value to the VLSI community. In addition, such techniques that model the general RLC parasitics of the package can be used to increase the performance of the package, and extend the useful lifetime of the package.

### 1.3.1 Performance Modeling

In all the noise sources in Section 1.2, the amount of noise is ultimately proportional to the rate of change of the current or voltage ($\frac{di}{dt}$ or $\frac{dv}{dt}$). The rate of change of current or voltage can be translated to system performance using standard equations for robust high-speed digital design [66]. Since all of the noise sources are expressed in terms that relate the amount of noise to the rate of change in current or voltage, the acceptable rate of change of voltage or current can be calculated. Using this approach the maximum system performance can be predicted using the electrical package parameters and user-defined noise limits as the two inputs into the model. In this work an analytical model is proposed for system performance which considers each noise source. From this model the maximum rate of change of voltage or current can be found for any given package and set of noise limits. This directly predicts the per-pin datarate and ultimate bus throughput for a given package. This analytical model is also used to predict the performance increase for the various solutions that are presented. The performance model presented in this work is shown to be accurate to within 10% of analog simulator results which are much more computationally expensive.

### 1.3.2 Optimal Bus Sizing

One of the biggest challenges for VLSI designers is selecting the size and configuration for their off-chip busses [92]. The main problem is that simply adding pins to an off-chip bus does not necessarily increase the throughput in a linear manner [87]. As pins are added to a package to increase the bus size and throughput, the noise induced by the signals that are simultaneously switching leads to a significant increase in noise. This means that as signals are added, the per-pin datarate of each pin needs to be decreased to meet the system noise limits. As a result, the overall throughput of an off-chip bus approaches an asymptotic limit as signals are added. As a consequence, multiple bus configurations can meet the same throughput objective. In order to aid in the selection of the optimal bus configuration, the cost of the bus is considered in the model. By including the cost of the signal, power, and ground pins that are needed to expand the size of a bus, the most cost-efficient bus configuration can be found. This allows designers to quickly compare two bus configurations of equal throughput and choose the least expensive solution. The model considers the electrical parameters of the package in addition to the per-pin cost, which allows the model to compare configurations in different packages. The metric of *Bandwidth per Cost* is defined as a means to find the most cost-effective bus configuration. The results presented in this work are supported by the industry trend of moving toward faster narrower busses over slower wider busses to achieve the same throughput at a lower cost.

### 1.3.3 Bus Encoding

When determining the maximum performance of a bus the worst-case noise limits must be considered. In all of the noise sources listed in Section 1.2, a worst-case bus pattern is assumed to be present on the bus. In the case of supply bounce all of the signals in the bus are assumed to be switching in the same direction. In the case of bandwidth limitation and signal coupling, the middle signal in the bus is either switching in one direction while all of the other signals on the bus are switching in the opposite direction, (Edge Degradation) or the middle signal is static while all other signals are switching in the same direction (Glitch). This work presents two bus encoding techniques that encode the off-chip data prior to leaving the IC. The encoding ensures that all patterns that result in noise of a magnitude greater than a user-specified limit are eliminated. By doing this a lower level of noise in the system is achieved which translates into increased performance. The techniques presented in this work show that the performance of the bus is increased dramatically *even after* considering the overhead of the encoder algorithm.

The first encoding technique is called *Bus Expansion* which maps the original set of logic vectors on the IC into an expanded set of vectors prior to leaving the IC. This expanded set of vectors ensures that noise patterns of a magnitude greater than the user-specified limit are never present on the data that is traversing the package.

The second encoding technique is called *Bus Stuttering*, which uses the original number of package pins but avoids the worst-case noise patterns by inserting intermediate logic vectors between outgoing pairs of vectors that induce noise with a magnitude greater than the user-specified limit. This approach allows the algorithm to be tailored for packages that are dominated by interconnect that is inductive, capacitive, or both. Experimental results show that the encoder circuits can be integrated into a modern CMOS process and achieve a minimal delay and area impact on the design. The encoders are shown to improve performance of off-chip busses up to 225% by avoiding data patterns which result in noise violations.

### 1.3.4 Impedance Compensation

Impedance discontinuities are caused by excess inductance or capacitance that is present in the package interconnect. The term *excess* refers to any inductance or capacitance that causes the interconnect impedance to be different from the system impedance. If the interconnect has a higher impedance than the system ($Z_L > Z_0$), then the interconnect is said to have excess inductance. If the interconnect has a lower impedance than the system ($Z_L < Z_0$), then the interconnect is said to have excess capacitance. When this occurs, the impedance discontinuities cause reflections which limit the performance of the system. To address this problem a compensation technique is presented that adds capacitance or inductance near the interconnect to match its impedance to the system impedance. If the spatial location of the compensation inductance/capacitance is near the interconnect, then the interconnect and compensation will be seen as a lumped element and will be governed by Equation 1.12. By using this compensation technique, a forward traveling wave front will see the compensated package interconnect as a matched impedance and reflections will be avoided.

Two compensation techniques are presented in this work. The first technique is a *static compensation* approach in which the pre-defined compensation elements are placed on the package and IC substrate. This technique surrounds the interconnect with the appropriate inductance or capacitance to achieve the impedance match.

The second technique is a *dynamic compensation* approach in which programmable compensation elements are placed on-chip. By placing the compensation on-chip, the interconnect impedance can be adjusted after the IC is packaged. This has the advantage of being able to accommodate any design or manufacturing tolerances that may vary the interconnect's original impedance. Both techniques are shown to be able to compensate for all reasonable ranges of package interconnect impedance and significantly reduce reflections. The compensation techniques are shown to reduce reflected noise as much as 400% for broadband frequencies up to 3GHz.

Figure 1.7 show a summary of the improvement in bus performance that can be achieved using the techniques proposed in this work. The original performance is based on a bus size of 4 bits with one $V_{DD}$ and one $V_{SS}$ pin where supply bounce is the failure mechanism. The performance is improved by encoding the data to avoid data patterns which result in noise limit violations and in turn limit the maximum throughput that can be achieved. The performance improvement considers the overhead of the encoder. For all package technologies studied in this work, the supply bounce due to the inductive nature of the interconnect is the limiting factor. Moving toward advanced packaging reduces the parasitic inductance present in the package which results in faster performance. In addition, when the parasitics of the package are reduced, the improvement techniques presented in this work will have more of a positive impact.

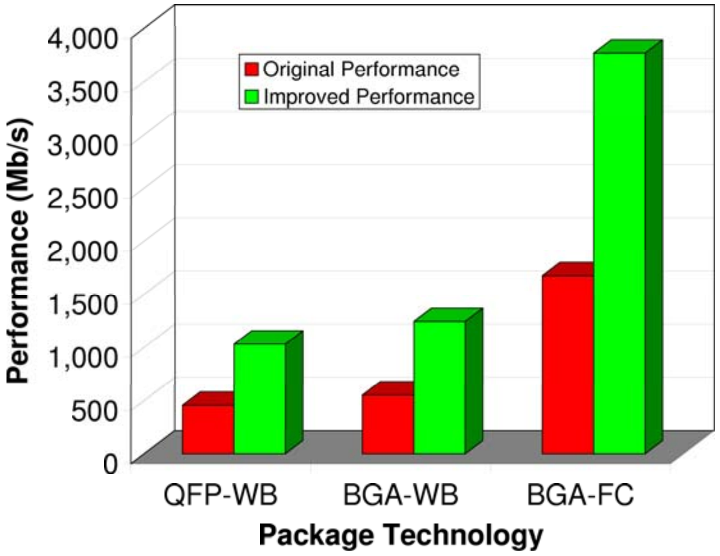


Figure 1.7: Performance Improvement Using Proposed Techniques

## 1.4 Advantages Over Prior Techniques

This section presents previous research and current techniques on the topic of package performance modeling and noise reduction. In each area the contribution of this work is compared and contrasted.

### 1.4.1 Performance Modeling

The most commonly used technique to predict the performance of an IC package is through analog simulation tools such as SPICE [76, 77]. SPICE (**S**imulation **P**rogram for the **I**ntegration **C**ircuit **E**nvironment) is a general purpose circuit simulator. SPICE simulates circuits by simultaneously solving the governing electrical equations for each element in the circuit. By doing this the detailed analog behavior of each node within the circuit can be monitored in either the time or frequency domain. While SPICE is a powerful tool for relatively small circuits, it is typically too computationally expensive to apply to entire VLSI designs [56].

The analytical model presented in this work forms a simple expression that relates a given noise source magnitude to a rate of change in current or voltage ($\frac{di}{dt}$ or $\frac{dv}{dt}$). The noise source magnitude is proportional to the electrical parasitics of the IC package in addition to the rate of change in current or voltage. The rate of change of current or voltage is related to overall system performance by applying a series of design rules for a robust digital system. Using this framework the acceptable noise limits for any noise source can be converted into predicted system performance metric for a given package. This technique is linear resulting in a dramatically lower computation time compared with SPICE with high accuracy. This enables the evaluation of more bus configurations, packages, and noise limits which enhances the decision-making ability of the VLSI designer. In addition, integration of this approach into digital Computer Aided Design (CAD) tools becomes more practical.

### 1.4.2 Optimal Bus Sizing

Traditionally, inter-chip communication is performed using wide parallel busses. The standard approach to achieving the desired system bandwidth is to increase the number of pins on the package until the desired throughput is attained. There are three main problems with this approach:

- Cost of packaging. Package cost scales faster than linearly with the number of I/O pins that are needed, and accounts for a large contribution to the overall chip price [92].

- Performance. Wide parallel busses experience a host of signal integrity issues associated with simultaneous switching of digital signals [56, 58, 61] as outlined in Section 1.2. The noise induced in the package is proportional to the number of off-chip signals that are switching simultaneously. One solution to this problem is to increase the number of power and ground pins in the bus configuration. This reduces the inductance in the power supply current path in addition to reducing the magnitude of the coupled noise between signals; however this increases the cost of the package because the number of I/O pins increases. Another solution to the package parasitic problem is to move toward advanced packaging technologies such as flip-chip bumping to reduce the parasitic inductance and capacitance in the package [81]. While this solution does reduce the noise in the package, advanced package technologies dramatically increases the price of the IC.

- The increases in package bandwidth do not scale at the same rate as on-chip core frequencies [56]. The traditional approach of widening parallel busses to match the inner core's datarate is impractical not only from a cost viewpoint, but also because the signal integrity problems mentioned above limit how wide

> busses can be. The paradox of the wide parallel bus is — intuitively, adding I/O should produce a linear increase in system throughput but in reality suffers an asymptotic limit due to additional noise sources that arise as more signals switch. Parallel busses have to be ran at lower speeds as their width increases, which inherently limits their throughput.

Recently we have seen the emergence of narrower busses that run at higher per-pin datarates [56, 58]. These new busses include Rapid I/O [83], PCI Express [84], and Hyper Transport [85]. All of these busses take advantage of the fact that the same bandwidth can be achieved by using less signals that run faster than a traditional wider parallel bus that uses more signals at a reduced per-pin datarate. By using narrower busses, the desired throughput of the bus is achieved at a lower cost on account of using fewer package pins. Regardless of whether the inter-chip communication uses a slower, parallel bus or a faster, narrow bus, the objective is the same: the inter-chip bus must deliver the highest throughput in the most cost-effective manner. This is a challenging problem due to the faster than linear increase in the cost of adding I/O pins that must be balanced with the fact that there exists an asymptotic limit to how much bandwidth can be attained by widening the bus.

Much work has also been done to increase the throughput of the IC package by moving toward advanced packaging. Research has been performed on both the level 1 (IC to Package Substrate) [53, 81] and level 2 (Package to System PCB) [54, 79, 80] connections with the ultimate goal of reducing the parasitic inductance and capacitance of the interconnect. While these approaches increases performance, they are typically too expensive for the majority of VLSI designs.

Recently there has also been considerable research in the design and characterization of low signal count busses [59, 62]. This approach targets busses that use low channel counts, which run at faster datarates than traditional IC package pins have historically been able to achieve. The main goal of this thread of research has been to increase the performance of an individual off-chip channel.

The bus sizing technique presented in this work aims at finding the optimal configuration of signal, power, and ground pins that yields the most system bandwidth at the least amount of cost. The metric of *Bandwidth per Cost* is introduced which gives VLSI designers a quick method to compare different bus configurations for cost-effectiveness.

### 1.4.3 Bus Encoding

There has been work done in the area of reducing package noise by altering the signals prior to leaving the IC. Pipeline Damping was presented in [1], with the aim of reducing the total $\frac{di}{dt}$ by implementing a multi-valued output circuit. This work reduced the average $\frac{di}{dt}$ in the output stage by limiting the output swing of the driver. While this technique reported significant reduction in average off-chip noise, the peak noise limit of the output stage was not reduced which resulted in the same off-chip datarate.

Other techniques aimed specifically at the design of the output stage to limit the net ground bounce in the package [2]. These techniques were successful in reducing ground bounce; however, other noise sources were not addressed.

On-chip bus coding techniques have been applied to avoid capacitive cross-talk in long busses [3, 4, 5, 6]. These approaches encodes the data prior to driving it on the bus and remove the worst-case capacitive cross-talk patterns. These techniques have been shown to increase the performance of on-chip busses by reducing the interconnect delay. The work in this thesis uses a similar approach but addresses the multiple noise sources present in the inductive off-chip interconnect.

Statistical encoding techniques have been used in audio and video applications to aid in relieving network congestion [28, 31]. These techniques have been successfully applied to MPEG and DVD protocols [33, 34]. The main approach of these encoding algorithms has been to eliminate similar or redundant data packets that occur sequentially on the network. This allows a comparable quality of audio/video (AV) without needing to send each and every data packet associated with the data compression algorithm.

The contribution of the encoding techniques presented in this work differ from previous techniques in that they specifically address the electrical noise sources in the physical interconnect. This makes them ideal for reducing noise within IC packaging. It also makes them suitable for any application in which the switching patterns directly effect performance such as on-chip capacitive busses, power minimization, or in statistical A/V protocols.

### 1.4.4 Impedance Compensation

Much work has been done in the characterization of the electrical parameters of package interconnect [15, 16, 17]. Previous work has demonstrated the severe impedance mismatch that occurs due to the interconnect structures. Altering the impedance of an electrical structure in the package has typically been difficult since adding additional capacitance or inductance near the structure has been impractical due to the relatively large size of the component. The standard approach has been to tolerate the resultant reflections from the package. The magnitude of the reflections was one of the limiting factors to a package's performance. This approach is no longer practical as the rise-times of digital signals continue to increase. Many improvements have been made in integrating on-chip and on-package capacitors and inductors [35, 37, 40]. New materials have allowed placing impedance altering components near the package interconnect, which has enabled the possibility of matching the impedance of the interconnect to the system impedance [38, 51, 52]. This work uses the advances in on-chip and on-package componentry to construct a technique that alters the impedance of the package interconnect [50]. This technique allows a better impedance match through the package, which results in reduced reflections and higher system throughput.

## 1.5 Broader Impact Of This Thesis

This thesis presents techniques that specifically model and improve performance in IC packages. Since all of the modeling and performance techniques are described using a common mathematical framework, the work in this thesis can be easily applied to a wide variety of electronic applications. The encoding techniques are formulated to eliminate data sequences which result in unwanted noise, which makes them suited for application in power reduction, audio/video compression, and internet fabric congestion. In addition, since all the techniques presented in this work are constructed to alleviate the detrimental effects of the inductance and capacitance of the interconnect, this naturally makes them applicable to any noise-prone electrical interconnect in a digital system. These applications include connectors, backplanes, and cables. Since all of the techniques are formulated independent of technology or process, they can be easily implemented as *intellectual property* cores for use in a wide variety of VLSI designs and implementation in Field Programmable Gate Arrays.

## 1.6 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 describes the package technology that is studied in this work. The construction of three industry standard packages are presented. The three packages are the *Quad Flat Pack* (QFP) with *Wire Bonding* (WB); the *Ball Grid Array* (BGA) with *Wire Bonding*; and the BGA with *Flip-Chip Bumping* (FC). In the past decade the QFP-WB package has been the most popular style of packaging. The BGA-WB package is presently the most popular package while the BGA-FC package is predicted to be the most common package in the next decade. Packages construction as well as electrical parameters are presented for all three packages. Chapter 3 presents the terminology and background information that will be used throughout the rest of the thesis. Chapter 4 describes the analytical model for performance and noise level prediction. Chapter 5 presents the optimal bus sizing technique to determine the most cost-effective bus configuration. Chapters 6 and 7 present the two bus encoding techniques that reduce package noise by avoiding switching patterns that result in noise that is greater than a user-specified limit. Chapter 8 presents the impedance compensator methodology. Chapter 9 discusses future trends in VLSI and how this research can be applied to increase digital system performance. Finally, conclusions are drawn in Chapter 10.

# Chapter 2

# Package Construction and Electrical Modeling

The construction of an IC package must accomplish many tasks. The first is to electrically connect the signals on the IC to the signal paths on the system PCB. The second is to electrically conduct power from the system PCB to the devices on the IC. In addition, the package must provide mechanical protection for the IC as well as thermal dissipation. The package typically consists of two levels of interconnect. The first is the connection from the IC to the package (level 1) and the second is from the package to the system PCB (level 2).

## 2.1 Level 1 Interconnect

The level 1 interconnect electrically connects the signals and power on the IC to the package. On the IC the transistors reside on the lowermost layers of the die, above the substrate. The various metal layers contain the electrical interconnect which allows the construction of complex digital circuitry and power distribution grids which supply current to the transistors [20]. The highest metal layer contains the pads to which the level 1 interconnect will contact. In a typical IC, the metal layers utilize Aluminum or Copper [63]. For a modern VLSI IC, the typical sizes of the pads can range from $35\mu m$ x $35\mu m$ to $100\mu m$ x $100\mu m$.

### 2.1.1 Wire Bonding

One of the most common and proven level 1 interconnect structures is the *Wire Bond*. A wire bond can be constructed with Gold, Aluminum, or Copper and is a thin round conducting wire that bonds to the pads on the IC and on the package. The wire bond can range in diameter from $7\mu m$ to $1mm$ depending on the technology of the automated bonding machine. The connection between the wire and the pad is accomplished using thermosonic energy [63] which anneals the two metals into a robust mechanical joint. The contact to the IC is accomplished using a *ball bond* connection. A ball bond is formed when the automated bonding machine approaches the pad in a perpendicular manner. This connection requires a square pad on the IC which minimizes area on the die. Figure 2.1 shows a Scanning Electron Microscope (SEM) photograph of a ball bond joint [86].

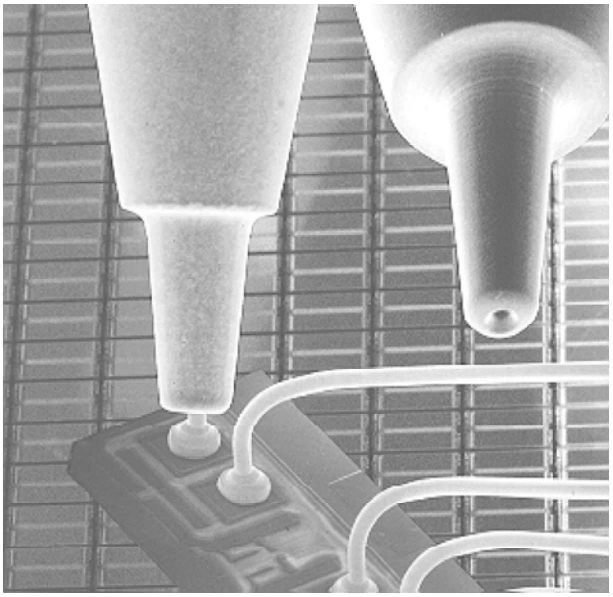


Figure 2.1: SEM Photograph of Ball Bond Connection

The connection between the wire bond and the package pad can be formed using either a ball bond or a *wedge bond.* A wedge bond is accomplished when the automated bonding machine approaches the pad in a parallel sweeping manner. This type of connection requires a rectangular pad on the package, typically on the order of $100\mu m$ x $400\mu m$. The wedge bond is used as a way to increase the speed of the automated bonding process; however, when the area on the package is a constraint, a ball bond can also be used which reduces the package pad size by using a square shaped pad. Figure 2.2 shows an SEM photograph of a wedge bond joint [86].

The automated wire bonding process has been refined over the past decades to increase joint strength and reduce process time. Modern wire bonding machines are capable of making thousands of individual connections per minute. The increased speed of the bonding process reduces the overall cost of the package. The low cost and reliability of the wire bond has made it the most popular choice for the level 1 interconnect within an IC package. Figure 2.3 shows an SEM photograph of the entire wire bond structure for an IC substrate with pads placed on the perimeter of the die [63].

While the wire bond interconnect is the most popular level 1 interconnect due to its low cost and mechanical robustness, its electrical parasitics can be considerable,

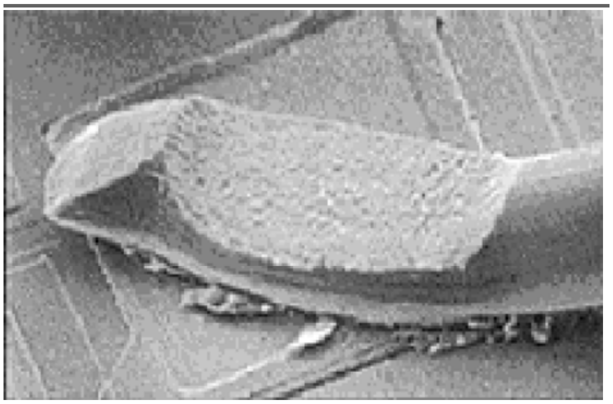


Figure 2.2: SEM Photograph of Wedge Bond Connection

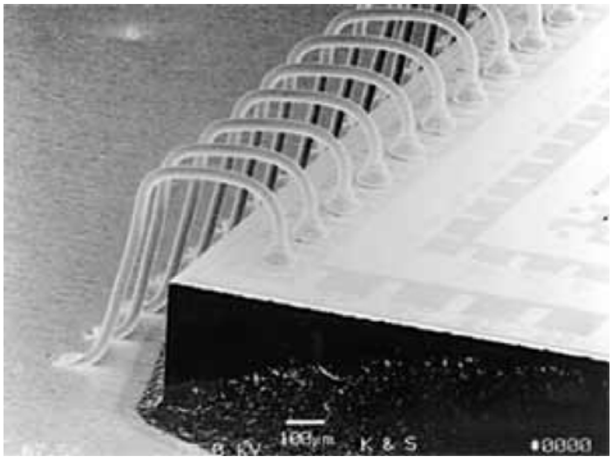


Figure 2.3: SEM Photograph of a Wire Bonded System

especially when running at today's datarates. The wire bond assembly process produces an interconnect structure that is separated from its return current path by a relatively large distance. This has many electrical disadvantages. The first is that the wire bond contains a significant amount of self-inductance ($L_{11}$) due to the loop area of the return current path. This leads to power supply bounce (Equations 1.2 and 1.3), mutual inductive signal coupling (Equation 1.6), and inductive impedance discontinuities (Equation 1.13). The second disadvantage is that since the wire bond is far from its return path, its self capacitance ($C_0$) is reduced and has a similar value as the magnitude as the mutual capacitance ($C_{1k}$) to other signal wires. This leads to increased capacitive signal coupling (Equation 1.9) and capacitive bandwidth limiting (Equation 1.4). While these electrical factors limit the performance of wire bonded packages, wire bonding is still used in more than 95% of all VLSI design starts due to its reliability and cost-effectiveness [63, 89, 92].

### 2.1.2 Flip-Chip Bumping

To address the electrical performance limitations of wire bonding, *Flip-Chip bumping* was introduced. In flip-chip bumping square pads are placed on the top-

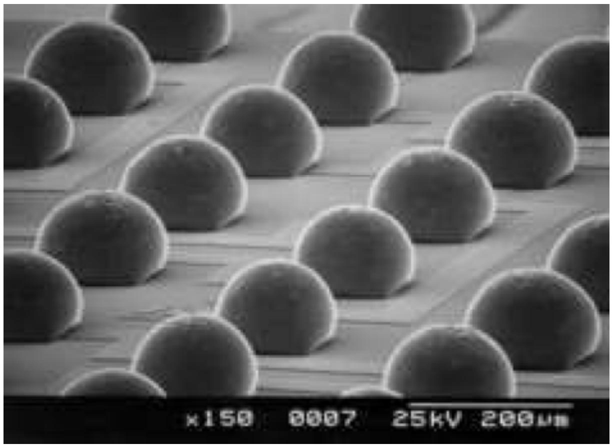


Figure 2.4: SEM Photograph of Flip-Chip Bump Array

most metal layer of the IC substrate. Upon these pads, a soldering compound is placed either through chemical vapor deposition (CVD) or plating. The solder material is then heated to a temperature in which it melts or *reflows*. When the solder material becomes molten, the surface tension of the solder attempts to reduce its surface area and forms a spherical object. Since the bumping is accomplished through a chemical process instead of a mechanical process (as in wire bonding), the IC substrate pads can be arranged in an array pattern and can be much smaller. This allows the entire surface of the IC to be used for bumping pads, which dramatically increases the number of level 1 interconnects that are possible for a given die size. Figure 2.4 shows an SEM photo of flip-chip bumps after reflow [86].

The package substrate contains a complimentary array of pads that matches the array pattern on the IC substrate. The IC substrate is then turned face-down and placed on the package substrate. The IC and package substrates are then subjected to a reflow process in which the solder bumps once again become molten and form an electrical and mechanical connection between the IC and the package pads. The surface tension of the solder bumps prevents the weight of the IC substrate from completely collapsing the solder bump spheres. This leaves an air gap between the IC substrate and the package substrate everywhere there is no flip-chip bump. This gap is then filled with a non-conductive adhesive using an *underfill* process. The underfill process creates a solid mechanical structure that absorbs the thermal expansion mismatches between the IC substrate and package substrate. The underfill process also prevents moisture and contaminants from getting into the bump array, which can cause local thermal expansion mismatches. The underfill process is critical due to the extreme temperature range that an IC will cover during normal operation. A large thermal expansion mismatch can lead to stress fractures in the flip-chip bumps which could sever the electrical connection from the IC to the package.

Electrically, flip-chip bumping has many advantages. First, the flip-chip bumps are significantly smaller than bonding wires due to the use of a chemical process instead of a mechanical process to form the connection. Flip-chip pads can be as small as $35\mu m$ in diameter compared to $100\mu m$ x $100\mu m$ for a wire bond connection. The reduced size of the interconnect leads to reduced parasitic inductance and capacitance associated with the connection. The lower electrical parasitics reduce all of the noise sources described in Section 1.2. The second electrical advantage of flip-chip bumping is that the pads can be placed in an array pattern. This allows more pads to be placed for a given substrate size compared to traditional perimeter placement used in wire bonding. The increased pad count leads to higher signal density and the possibility of large amounts of redundant power and ground connections. When redundant power and ground paths are used it reduces the overall inductance in the power distribution path. This results in less supply bounce (Equations 1.2 and 1.3).

The main disadvantage of flip-chip bumping is that it is more expensive than wire bonding. The cost is higher due to the manufacturing process that is required for flip-chip bumping. The bumping process must be carried out at the wafer level prior to individual IC separation. This means that ICs that have been identified as failures during electrical wafer test still receive the bumping process. As a result, more time is associated with the process, which results in greater cost. In addition, the bumping equipment must be able to accommodate the entire wafer instead of just the individual die as in wire bonding. As wafer sizes continue to increase, the bumping equipment must be continually upgraded. These manufacturing factors have prevented the wide spread adoption of flip-chip bumping despite its electrical advantages.

## 2.2 Level 2 Interconnect

The level 2 interconnect electrically connects the signals and power on the package to the system PCB. The system PCB is constructed using a lamination process in which the outermost layers contain surface mount (SMT) pads, to which the package is soldered. Since a lamination process is used, the pads on the PCB are much larger than that used in the level 1 connection. There are two main styles of level 2 interconnect; lead frame and a ball grid array.

### 2.2.1 Lead Frame

A lead frame is a structure that forms the electrical and mechanical connection between the package and the system PCB. The lead frame begins as a single piece of copper-based alloy sheet metal that contains a die mounting paddle and lead fingers. The features of the lead frame are formed by stamping or etching, which is then followed by a plating finish to reduce oxidation. The die mounting paddle is used to support the die during assembly. The lead fingers are used to form the connection to the system PCB and extend outwards from the perimeter of the die paddle. Lead frames only support wire bonding connections from the die. During assembly, the die resides on the die paddle while the wire bonding takes place to the outward extending lead fingers. The die paddle is then removed and the die is covered with a non-conductive encapsulant that protects the IC substrate and helps dissipate heat. The lead fingers extend out of the encapsulant and are then trimmed and formed to their final shape prior to mounting

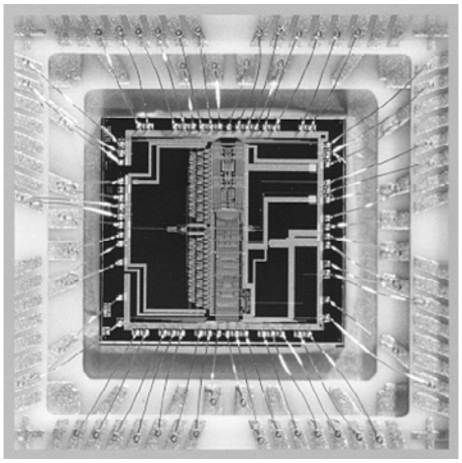


Figure 2.5: Lead Frame Connection to IC Substrate

to the system PCB. At this point the leads can be shaped into through-hole or SMT connections. Figure 2.5 shows the wire bonding from the IC substrate to the fingers of a lead frame [86].

The lead frame assembly process has been refined over the past decades to yield an inexpensive solution to IC packaging; however, the lead frame suffers from the same electrical disadvantages as the wire bond in that the interconnect is relatively long which creates a large return current path. The large return current path increases the self-inductance of the lead and the mutual coupling capacitive between leads, both of which increase package noise that limits the lead frame's performance. Another drawback of the lead frame is that it limits signal from the package to a perimeter egress. This limits the overall signal and power density that can be achieved by this style of interconnect.

### 2.2.2 Array Pattern

To improve upon the lead frame package interconnect, Ball Grid Array packaging was introduced. BGA technology is very similar to flip-chip bumping except that the solder balls reside between the package and the system PCB. Since the laminated construction of the PCB limits the feature sizes that can be achieved, the density of the array pattern that can be achieved is much less than flip-chip bumping. Current BGA packages range from $0.5mm$ to $1.27mm$ contact pitch [63]. Since the level 2 solder balls are much larger than the flip-chip bumps, an underfill encapsulant is not needed since the solder balls can absorb the majority of the thermal expansion mismatch between the package and the system PCB. In addition, the BGA package substrate is typically implemented using a PCB so the thermal expansion difference between the package and the system PCB is minimized. As with the flip-chip bumping interconnect, the BGA interconnect reduces the overall parasitic inductance and capacitance in the interconnect by reducing the electrical length relative to the lead frame approach. Also, the array configuration of the interconnect leads to the easy implementation of redundant power and ground pins which decreases the self-inductance in the power supply path. Figure 2.6 shows an SEM photograph of the underside of a $1mm$ pitch BGA package [86].

Figure 2.7 shows a photograph of an entire BGA package highlighting the array configuration of the interconnect [86].

## 2.3 Modern Packages

This work will focus on three of the most common packages used in modern VLSI designs. These packages represent the past, present, and future technologies of IC

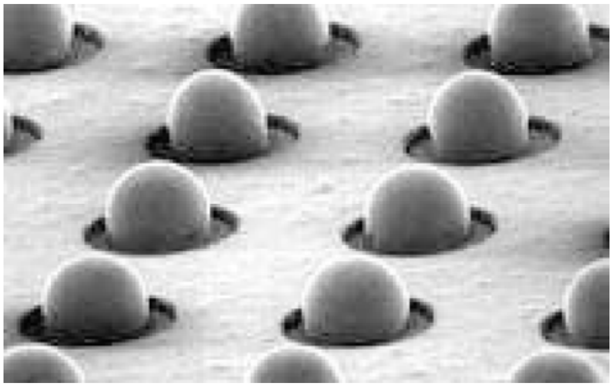


Figure 2.6: SEM Photograph of the Bottom of a $1mm$ Pitch BGA Package

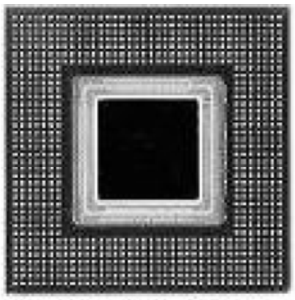


Figure 2.7: Photograph of a 1$mm$ Pitch BGA Package

packaging and illustrate the electrical, mechanical, and cost trade-offs that have been made by VLSI designers.

### 2.3.1 Quad Flat Pack with Wire Bonding

One of the most common packages over the past decade has been the Quad Flat Pack with Wire Bonding (QFP-WB). The QFP-WB uses wire bonding as its level 1 interconnect with ball bonds on the IC substrate and wedge bonds on the lead frame. The QFP lead frame extends from all four sides of the IC substrate, which allows pads to be placed along the entire perimeter of the IC. The QFP has been extremely successful due to its ability to use a common lead frame across many sizes of IC substrates. The flexibility of the wire bond interconnect allows multiple sizes of dies to be connected to the frame. The standardization of the lead frame size allows for the optimization of the plating, encapsulating, and trimming process which has driven the cost out of this package. Figure 2.8 shows a cross-section and top view of a QFP-WB package.

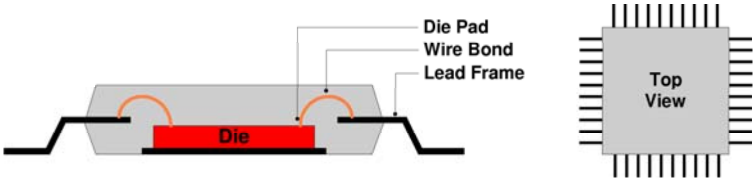


Figure 2.8: Cross-Section of Quad Flat Pack with Wire Bonding Package

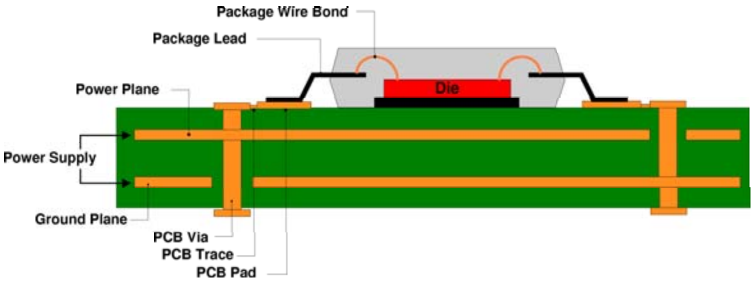


Figure 2.9: Cross-Section of System with QFP Wire Bond Package

The QFP-WB package is mounted to the system PCB using a pattern of perimeter placed pads which align to the lead frame contacts of the package. This package is mounted on the surface of the target PCB which allows placement on both the top and bottom of the system PCB. Figure 2.9 shows a cross-section of the entire assembly of a QFP-WB package to a system PCB. Signals and power make contact to the inner layers of the system PCB using *vias*. While the QFP-WB is a cost-effective package, the electrical limitations in both the level 1 and level 2 interconnect has slowed the use of the QFP-WB package.

### 2.3.2 Ball Grid Array with Wire Bonding

The most popular package in use today is the Ball Grid Array with Wire Bonding (BGA-WB). This type of technology uses a PCB substrate within the package. The level 1 interconnect is implemented with wire bonds that connect the ball bond pads on the IC to the wedge bond pads on the package PCB. The level 2 interconnect is implemented using an array of solder balls on the underside of the package PCB. The construction of the BGA-WB allows either a partial or fully populated array of pads on the bottom side of the package. Figure 2.10 shows the cross-section of a BGA-WB package and the bottom side array of solder balls.

The BGA-WB package addresses the electrical parasitic problem in the level 2

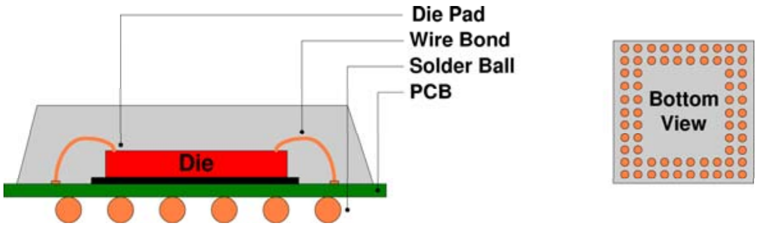


Figure 2.10: Cross-Section of Ball Grid Array with Wire Bonding Package

interconnect by moving away from the lead frame approach. This shift in technology has allowed much higher performance in this style of package. In addition, the array configuration of the solder balls allows much higher pin counts to be realized in the same system PCB area. The BGA-WB is mounted to the system PCB using a reflow process, which creates the bond between the package pads and the system PCB pads through the solder ball. Signals and power of this package make contact to the inner layers of the system PCB using vias. Figure 2.11 shows the cross-section of a BGA-WB package that is mounted to a system PCB.

The BGA-WB has gained wide adoption due to its combination of the inexpensive level 1 wire bonding interconnect with the advantages of an array style level 2 interconnect. The main drawback of the BGA-WB is that it still suffers the electrical parasitics associated with the level 1 wire bond interconnect. The wire bond interconnect also restricts the pad placement on the IC to a perimeter pattern which inherently

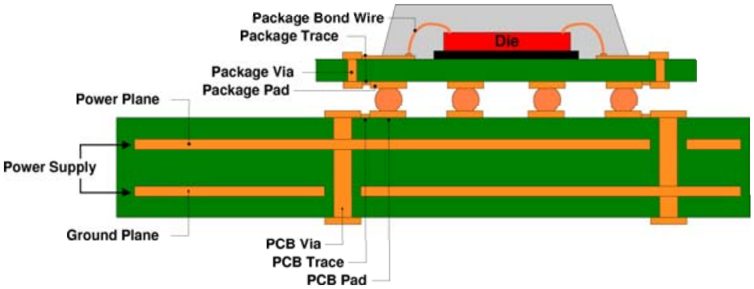


Figure 2.11: Cross-Section of System with BGA Wire Bond Package

limits the number of level 1 connections. This is the main limitation to the number of signals and power contacts that can be brought out to the system PCB. Despite these disadvantages, the BGA wire bond is the most widely used package for current VLSI design starts due to its electrical advantages over the QFP-WB and its cost-effectiveness relative to non-wire bonded packages.

### 2.3.3 Ball Grid Array with Flip-Chip Bumping

The next step in the evolution of VLSI packaging is the Ball Grid Array with Flip-Chip bumping (BGA-FC). This package addresses the electrical parasitics of the wire bond by implementing flip-chip technology as its level 1 connection. When combined with the electrical improvements gained by using a level 2 BGA interconnect, the BGA-FC promises to meet the needs of VLSI systems into the next decade. Figure 2.12 shows the cross-section of a BGA-FC package and the bottom side array of solder balls.

The BGA-FC package can achieve very high signal counts due to its array style pad pattern. In a BGA-WB package the total number of interconnects is limited by the wire bond noise that exists in the level 1 interconnect and the perimeter pad placement on the die. In the BGA-FC package this issue is addressed by implementing an array style interconnect between the IC and the package substrate. This allows higher signal counts and redundant power and ground connections which reduces the package noise and increases system performance. Packages with up to 2000 contacts have been successfully implemented using ball grid array with flip-chip bumping technology [92].

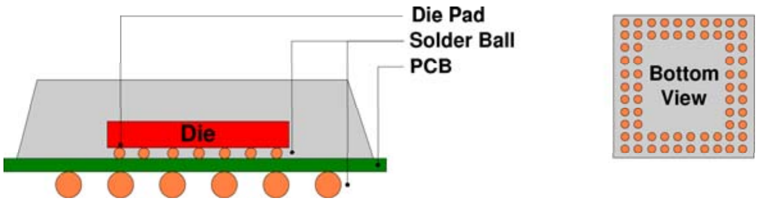


Figure 2.12: Cross-Section of Ball Grid Array with Flip-Chip Package

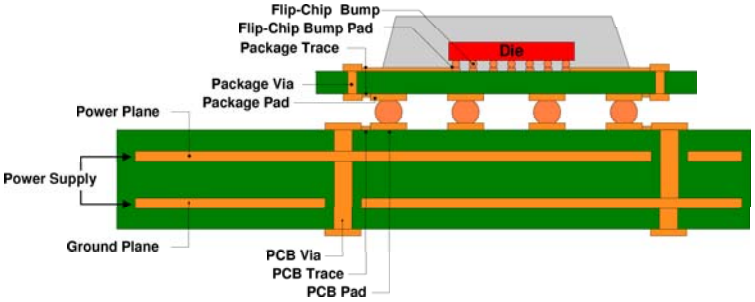


Figure 2.13: Cross-Section of System with BGA Flip-Chip Package

Figure 2.13 shows the cross-section of a BGA-FC package that is mounted to a system PCB.

## 2.4 Electrical Modeling

This section presents the results of the electrical parameter extraction performed to determine the inductance and capacitance magnitudes for the three packages studied in this work. The three dimensional structures within the package were modeled and simulated using electromagnetic field solvers. A combination of *Raphael* from *Avant!* [93] and *Advanced Design System* from *Agilent Technologies, Inc* [75] were used to accomplish the parameter extraction.

### 2.4.1 Quad Flat Pack with Wire Bonding

The QFP-WB package studied in this work uses a 100-pin lead frame with 25 leads per side on 0.8$mm$ pitch. The lead frame is composed of a copper based alloy with a tin plated finish. The IC is implemented on a 5$mm$ x 5$mm$ silicon die. The die is connected to the lead frame with 25$\mu m$ diameter gold wire bonds. The bonding wires attach to the lead frame using wedge bonds onto 100$\mu m$ x 400$\mu m$ pads. The wire connections to the IC are formed using ball bonds onto 100$\mu m$ x 100$\mu m$ pads. The wire bonds range is length from 3$mm$ to 5$mm$.

### 2.4.2 Ball Grid Array with Wire Bonding

The BGA-WB package studied in this work is a 20x20 array of $1mm$ pitch, fully populated with solder balls. Each solder ball has an average diameter of $1mm$ and an average collapsed height of $0.5mm$. The package substrate uses a $1.27mm$ thick $GETEK^{©}$ substrate with traces designed for an impedance of $50\Omega$ using trace widths of $0.1mm$. The wire bond connections to the package substrate are formed using $25\mu m$ diameter gold wires to $100\mu m$ x $400\mu m$ wedge bond pads. The wire connections to the IC are formed using ball bonds onto $100\mu m$ x $100\mu m$ pads. The wire bonds range is length from $1mm$ to $5mm$.

### 2.4.3 Ball Grid Array with Flip-Chip Bumping

The BGA-FC package studied in this work uses the same level 2 construction as the BGA-WB described above. The level 1 interconnect is formed using a 100x100 array of flip-chip bumps that have an average diameter of $125\mu m$ and average collapsed height of $75\mu m$. The flip-chip bumps connect complimentary arrays of pads on the package substrate and on the IC. The pads on each substrate are $100\mu m$ in diameter.

Table 2.1 lists the electrical parameter extraction results for the three packages described above. For each extraction the worst-case interconnect paths are extracted. These values represent the summation of all parasitic sources within the package.

| | QFP-WB | BGA-WB | BGA-FC |
|---|---|---|---|
| $L_{11}$ | 4.550 nH | 3.766 nH | 1.244 nH |
| $K_{12}$ | 0.744 | 0.537 | 0.330 |
| $K_{13}$ | 0.477 | 0.169 | 0.287 |
| $K_{14}$ | 0.352 | 0.123 | 0.230 |
| $K_{15}$ | 0.283 | 0.097 | 0.200 |
| $C_0$ | 300 fF | 288 fF | 197 fF |
| $C_{12}$ | 121 fF | 115 fF | 96 fF |
| $C_{13}$ | 17 fF | 97 fF | 9 fF |

Table 2.1: Electrical Parasitic Magnitudes for Studied Packages

The wire bond is the largest contributor of inductance in a bonded package. This interconnect is the largest source of impedance discontinuity and, in turn, the largest source of reflected noise. Chapter 8 presents an impedance compensation technique for this inductive interconnect so the specific electrical parasitics of a reasonable range of wire bond lengths were also extracted. Table 2.2 lists the specific inductance and capacitance parameters for a reasonable range of wire bond lengths used in the QFP-WB and BGA-WB packages.

| Length | $C_{wb}$ | $L_{wb}$ | $Z_0$ |
|---|---|---|---|
| 1 $mm$ | 26 $fF$ | 0.569 $nH$ | 148 $\Omega$ |
| 2 $mm$ | 52 $fF$ | 1.138 $nH$ | 148 $\Omega$ |
| 3 $mm$ | 78 $fF$ | 1.707 $nH$ | 148 $\Omega$ |
| 4 $mm$ | 104 $fF$ | 2.276 $nH$ | 148 $\Omega$ |
| 5 $mm$ | 130 $fF$ | 2.845 $nH$ | 148 $\Omega$ |

Table 2.2: Electrical Parasitics for Various Wire Bond Lengths

# Chapter 3

# Preliminaries and Terminology

This chapter describes the terminology and notation used throughout the rest of this thesis.

## 3.1 Bus Construction

A bus is defined as a group of signals that transmits data from one circuit to another. When designing a bus that traverses a package, the number of physical interconnect paths must be accounted for. This means that in addition to the total number of signal pins that are needed, the number of power and ground pins associated with the bus must also be considered. In VLSI design there are typically an equal number of power supply pins ($V_{DD}$) as ground pins ($V_{SS}$) [67, 92] for a given bus. For an arbitrarily large bus, individual bus *segments* can be defined that represent a subset of signals within a bus that are associated with at least one $V_{DD}$ and at least one $V_{SS}$ pin. By introducing the notation of a segment, subsets of the bus can be evaluated for performance instead of the entire bus, which leads to faster computation times without any loss of electrical accuracy.

**Definition :** A bus *segment* consists of a group of contiguous signal pins, along with at least one $V_{DD}$ and at least one $GND$ pin. The $V_{DD}$ ($GND$) pins are also contiguous in their placement.

In practice, buses are implemented in the form of concatenated segments, with each segment having their $V_{DD}$ and $GND$ pins on either side of the signal pins.

**Definition :** Let $N_{segment}$ represent the total number of pins within the segment.

**Definition :** Let $W_{bus}$ represent the number of signal pins within an individual segment.

**Definition :** Let $N_g$ represent the number of $V_{SS}$ pins within the bus segment where $N_g \geq 1$.

**Definition :** Let $N_p$ represent the number of $V_{DD}$ pins within the bus segment where $N_p \geq 1$.

Figure 3.1 shows an example segment. In this segment there are three signal pins ($W_{bus}$=3), one $V_{SS}$ pin ($N_g$=1), and one $V_{DD}$ pin ($N_p$=1).

The ratio of signal-to-supply pins are defined to give a quick estimate of the quality of the power and ground distribution is for a given bus segment.

**Definition :** The *Signal-to-Power-Ground* ($SPG$) metric is defined as $W_{bus}$:$N_p$:$N_g$.

**Definition :** The *Signal-to-Power-Ground Ratio* ($SPGR$) is defined as $W_{bus}/N_p$. It assumes that $N_p$=$N_g$=1.

In practice, most packages use $N_p$=$N_g$=1. For the example segment in Figure 3.1, $SPG$=3:1:1 and $SPGR$=3.

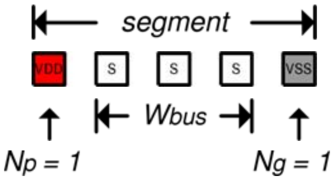


Figure 3.1: Individual Bus Segment Construction

The segment representation above corresponds to the commonly practiced bus construction in the VLSI industry today. We refer to segments by their index $j$, to keep trace of the relative location of a segment with respect to others.

**Definition :** Suppose a given bus has $k$ segments. We represent an arbitrary segment of interest by its index $j$. Segments to the left of the $j^{th}$ segment are assigned indices $j-1$, $j-2$, ..., $j-(k/2)$. Segments to the right of the $j^{th}$ segment are assigned indices $j+1$, $j+2$, ..., $j+(k/2)$.

**Definition :** Let $N_{bus}$ represent the total number of pin in an off-chip bus.

$N_{bus}$ includes all signal, power, and ground pins that are used in the package to construct the off-chip bus. Figure 3.2 shows an example bus construction that consists of 3 segments ($k$=3). In this bus each segment has the same number of signal ($W_{bus}$=3), power ($N_p$=1), and ground ($N_g$=1) pins.

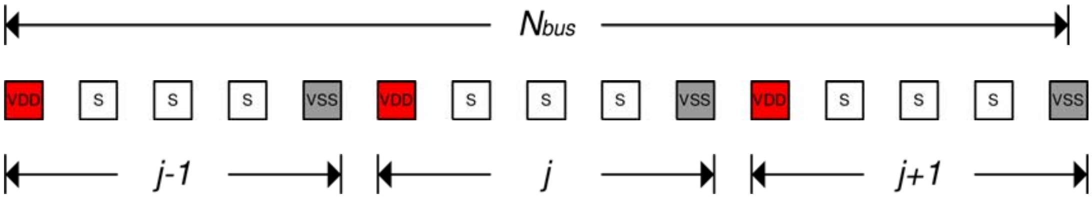


Figure 3.2: Bus Construction Using Multiple Segments

This notation allows a flexible framework that can represent arbitrarily large busses. Further, this notation matches the practical implementation of buses in VLSI designs. The number of pins within a segment is shown in Equation 3.1.

$$N_{segment} = W_{bus} + N_p + N_g \tag{3.1}$$

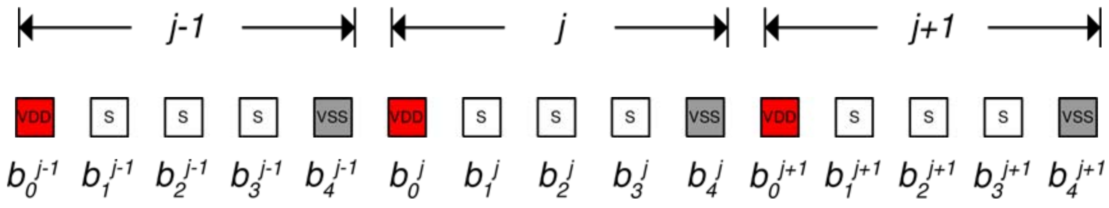
The number of a pins within the entire bus is defined in Equation 3.2.

$$N_{bus} = N_{segment} \cdot k \tag{3.2}$$

In order to specify the exact location within a bus and/or segment, an individual pin notation is defined. Each pin within a bus segment is given the notation $b$. Within a given segment, $i$ is defined as the relative pin location.

**Definition :** The $i^{th}$ pin of the $j^{th}$ segment is denoted $b_i^j$. The indices $i$ and $j$ start with 0 and increase from left to right.

This framework allows the spatial location of any pin $b_i^j$ within a bus to be described. Figure 3.3 shows an example bus using the spatial pin notation $b_i^j$. The arithmetic in the subscript of $b_i^j$ is performed modulo $N_{segment}$. This allows a consistent method to describe relative locations of pins across segment boundaries. For example, if $N_{segment} = 5$, then $b_2^j = b_{2+5}^{j-1} = b_{2-5}^{j+1}$.



j-1
j
j+1
VDD
S
S
S
VSS
VDD
S
S
S
VSS
VDD
S
S
S
VSS
$b_0^{j-1}$ $b_1^{j-1}$ $b_2^{j-1}$ $b_3^{j-1}$ $b_4^{j-1}$
$b_0^{j}$ $b_1^{j}$ $b_2^{j}$ $b_3^{j}$ $b_4^{j}$
$b_0^{j+1}$ $b_1^{j+1}$ $b_2^{j+1}$ $b_3^{j+1}$ $b_4^{j+1}$

Figure 3.3: Pin Representation in a Bus

## 3.2 Logic Values and Transitions

**Definition :** $s_i^j$ represents the logic state of pin $b_i^j$. For a signal pin, $s_i^j$ can take on a value of 0 or 1 which represents the Boolean logic value for pin $b_i^j$. For a $V_{SS}$ pin, $s_i^j$=0, and for a $V_{DD}$ pin, $s_i^j$=1.

**Definition :** $v_i^j$ represents the transition on any pin $b_i^j$. $v_i^j$=0 when no transition occurs on pin $b_i^j$. $v_i^j$=1 when pin $b_i^j$ transitions from a logic 0 to a logic 1. $v_i^j$=-1 when pin $b_i^j$ transitions from a logic 1 to a logic 0. $v_i^j$=0 for $V_{SS}$ or $V_{DD}$ pins.

Figure 3.4 shows an example bus, illustrating the notation for the state and transition of each pin.

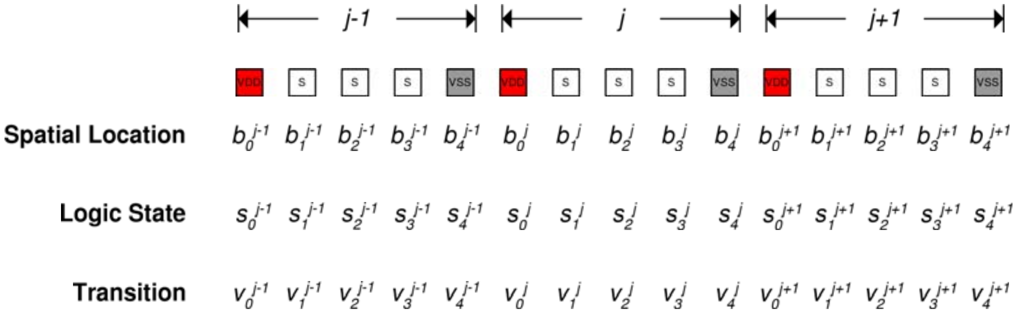

j-1
j
j+1
VDD
S
S
S
VSS
VDD
S
S
S
VSS
VDD
S
S
S
VSS
Spatial Location
$b_0^{j-1}$ $b_1^{j-1}$ $b_2^{j-1}$ $b_3^{j-1}$ $b_4^{j-1}$ $b_0^{j}$ $b_1^{j}$ $b_2^{j}$ $b_3^{j}$ $b_4^{j}$ $b_0^{j+1}$ $b_1^{j+1}$ $b_2^{j+1}$ $b_3^{j+1}$ $b_4^{j+1}$
Logic State
$s_0^{j-1}$ $s_1^{j-1}$ $s_2^{j-1}$ $s_3^{j-1}$ $s_4^{j-1}$ $s_0^{j}$ $s_1^{j}$ $s_2^{j}$ $s_3^{j}$ $s_4^{j}$ $s_0^{j+1}$ $s_1^{j+1}$ $s_2^{j+1}$ $s_3^{j+1}$ $s_4^{j+1}$
Transition
$v_0^{j-1}$ $v_1^{j-1}$ $v_2^{j-1}$ $v_3^{j-1}$ $v_4^{j-1}$ $v_0^{j}$ $v_1^{j}$ $v_2^{j}$ $v_3^{j}$ $v_4^{j}$ $v_0^{j+1}$ $v_1^{j+1}$ $v_2^{j+1}$ $v_3^{j+1}$ $v_4^{j+1}$

Figure 3.4: STATE and Transition Notation

## 3.3 Signal Coupling

### 3.3.1 Mutual Inductive Signal Coupling

The mutual inductive coupling between signals ($M_{ik}$ or $K_{ik}$) represents the inductive coupling magnitude between an arbitrary pin $b_i^j$ to its adjacent neighboring pins. The inductive coupling can span segment boundaries. Mutual inductive coupling exists between any two current carrying interconnects; however, when the magnitude of the coupling coefficient is relatively small the coupling can be ignored. The term $p_L$ represents the number of neighbors on either side of pin $b_i^j$ for which inductive coupling is considered. By considering only $p_L$ neighbors on either side of the signal of interest, we simplify the analysis, without significantly compromising the quality of the results.

Figure 3.5 illustrates this idea. In this Figure, $p_L$=4, which means that we consider 4 neighbors' mutual inductive coupling coefficients (to the left and to the right), and ignore the rest. Reducing $p_L$ decreases the computation time when evaluating package noise, but results in larger error.

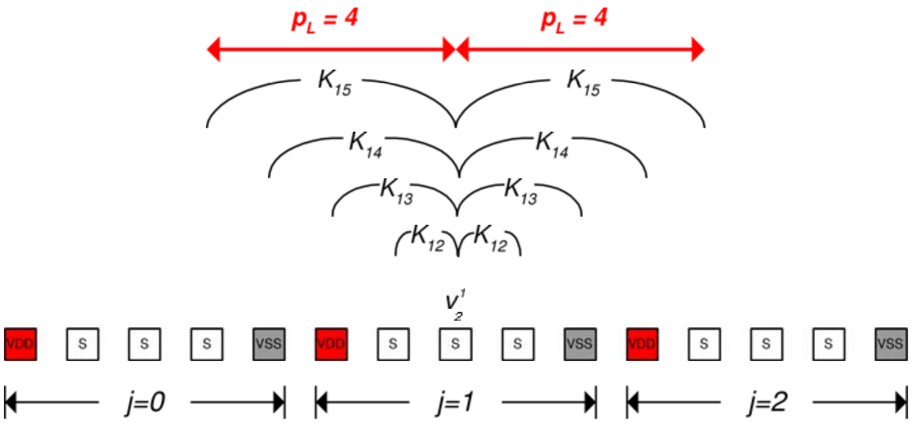


Figure 3.5: Mutual Inductive Coupling Notation

### 3.3.2 Mutual Capacitive Signal Coupling

The mutual capacitive coupling between signals ($C_{ik}$) represents the capacitive coupling magnitude between an arbitrary pin $b_i^j$ to its adjacent neighboring pins. In a similar manner to inductive coupling, mutual capacitive coupling can span segment boundaries. The term $p_C$ is used to describe how many pins away from the pin of interest that capacitive coupling is considered. Typically, $p_C$=1 is adequate for an accurate analysis, since mutual capacitances for subsequent neighbors are significantly smaller on account of capacitive shielding.

Figure 3.6 illustrates this idea. In this figure, $p_C$=2, which means that we consider 2 neighbors' mutual capacitive coupling (to the left and right), and ignore the rest.

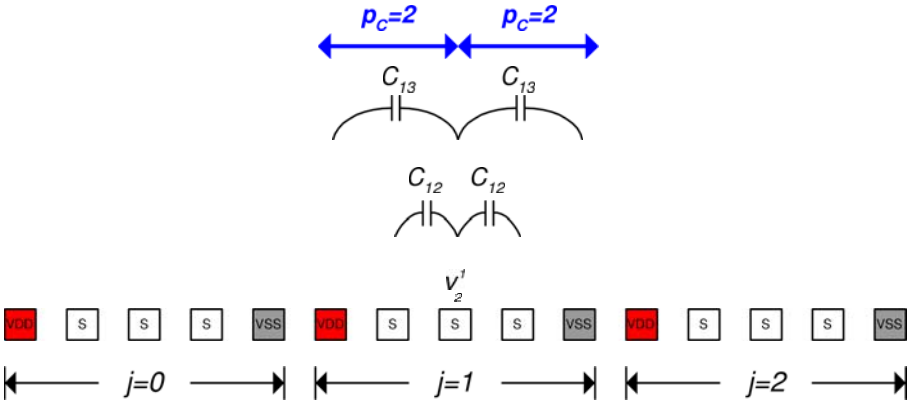


Figure 3.6: Mutual Capacitive Coupling Notation

## 3.4 Return Current

When CMOS output drivers charge or discharge a signal line, return current will flow through the $V_{SS}$ or $V_{DD}$ pins in the package to complete the closed loop circuit path. When inductance is present in the $V_{SS}$ or $V_{DD}$ paths, then supply bounce will result. In VLSI packaging the return current for multiple signal pins will typically share a single supply pin. The return current will always seek the path of least resistance when traversing the package. In the case of VLSI packaging, this means that the return current will seek the supply or ground pin that is the spatially closest to the signal pin that is switching (due to the reduced inductance and resistance in the return path). This behavior may not appear to adhere to the *segment* notation described in Section 3.1. The actual return current for a pin may return current through another segment's supply pin; however, due to the symmetry of the segment notation, the total number of signals that utilize a segment's supply or ground pin is, in the worst-case, $W_{bus}$. This is due to the fact that even though signals within a particular segment may return current through an adjacent segment's supply pin (which is spatially closer), signals from adjacent segments will return current through the supply pins of the segment of interest (because its supply pin is spatially closer to those signals). This allows the evaluation of an individual segment for supply bounce, considering only the signal pins within that segment. This does not lead to a reduction in modeling accuracy. This results in a much simpler analytical model for bus performance modeling and noise evaluation.

Figure 3.7 shows how the return current for a signal can span segment boundaries yet the total amount of return current within a segment will always be $W_{bus} \cdot I$. We assume that all the currents $I_i^j$ are identical, and equal to $I$.

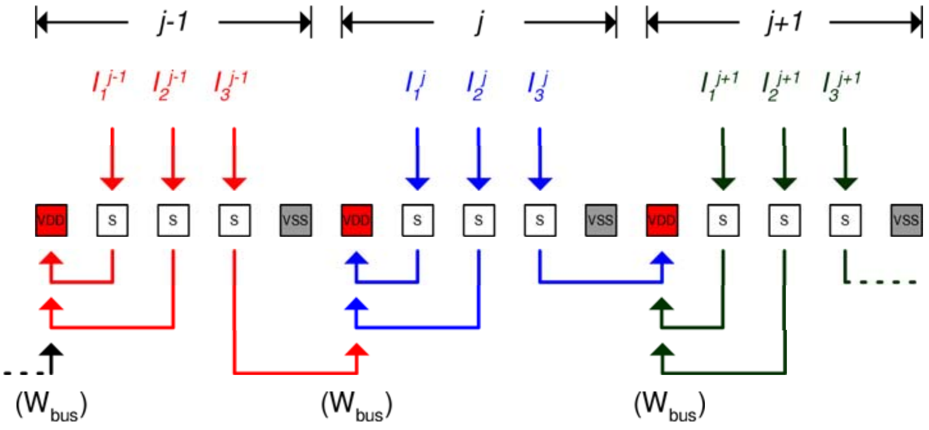


Figure 3.7: Return Current Description

## 3.5 Noise Limits

The maximum allowable noise for any of the noise sources described in Section 1.2 is expressed in terms of user-defined parameters. For supply bounce, signal coupling, and reflected voltage, the noise limits are expressed as a percentage of $V_{DD}$. $P_x \cdot V_{DD}$ is the maximum allowable voltage noise defined by the user, where $0 \leq P_x \leq 1$. Figure 3.8 shows how the user-defined noise limits are expressed in terms of $V_{DD}$.

**Definition :** $P_{gnd}$ represent the maximum allowable amount of ground bounce in the system. $P_{gnd}$ is defined as a fraction of $V_{DD}$ such that the noise due to ground bounce is $\leq P_{gnd} \cdot V_{DD}$.

**Definition :** $N_{-1}$ represent the number of signal pins within a segment that are transitioning from a logic 1 to a logic 0. This number represents the number of signals that are returning current through the $V_{SS}$ pin and resulting in ground bounce.

**Definition :** $P_{supply}$ represent the maximum allowable amount of supply bounce in the system. $P_{supply}$ is defined as a fraction of $V_{DD}$ such that the noise due to supply bounce is $\leq P_{supply} \cdot V_{DD}$.

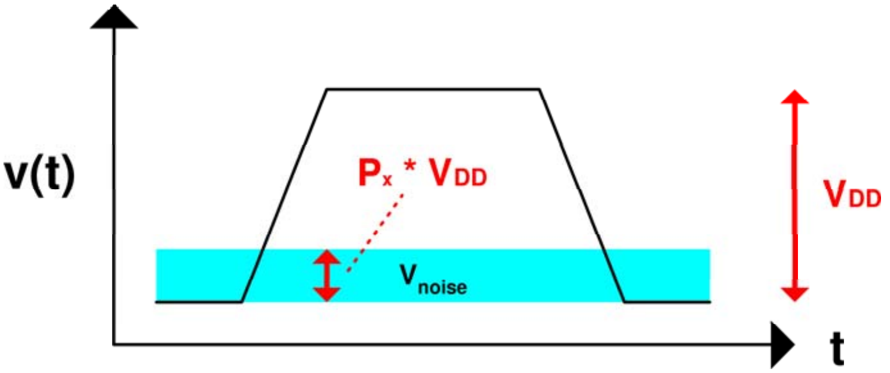


Figure 3.8: Package Noise Notation

**Definition :** $N_1$ represent the number of signal pins within a segment that are transitioning from a logic 0 to a logic 1. This number represents the number of signals that are drawing current through the $V_{DD}$ pin and resulting in supply bounce.

**Definition :** $P_0$ represent the maximum allowable amount of glitching noise in the system. $P_0$ is defined as a fraction of $V_{DD}$ such that the absolute value of the glitching noise is $\leq P_0 \cdot V_{DD}$.

A glitch occurs when a victim signal is static while neighboring signals transition causing mutual inductive and capacitive coupling onto the victim line.

**Definition :** $P_1$ represent the maximum allowable amount of rising edge coupling noise in the system. $P_1$ is defined as a fraction of $V_{DD}$ such that the rising edge coupling noise due to all neighboring signals is $\leq P_1 \cdot V_{DD}$.

Rising edge coupling occurs when a victim signal is transitioning from a logic 0 to a logic 1 at the same time that neighboring signals are transitioning and causing mutual inductive and capacitive coupling onto the victim line. This coupling can either speed up, slow down, or leave unaffected the rising edge on the victim line.

**Definition :** $P_{-1}$ represent the maximum allowable amount of falling edge coupling noise in the system. $P_{-1}$ is defined as a fraction of $V_{DD}$ such that the falling edge coupling noise due to all neighboring signals is $\leq P_{-1} \cdot V_{DD}$.

Falling edge coupling occurs when a victim signal is transitioning from a logic 1 to a logic 0 at the same time that neighboring signals are transitioning and causing mutual inductive and capacitive coupling onto the victim line. This coupling can either speed up, slow down, or leave unaffected the falling edge on the victim line.

**Definition :** $P_{\Gamma}$ represent the maximum allowable amount of reflected noise in the system. $P_{\Gamma}$ is defined as a fraction of $V_{DD}$ such that the absolute value of the reflected noise is $\leq P_{\Gamma} \cdot V_{DD}$.

When expressing the noise limit for capacitive bandwidth limiting, it is not applicable to use a fraction of $V_{DD}$; instead, the fraction by which the risetime of the signal is increased due to switching neighboring pins is used.

**Definition :** $P_{BW}$ represent the maximum amount that the *original* risetime of a signal is altered due to switching neighboring pins. The *original* risetime is defined as the time it takes to switch from 10% of $V_{DD}$ to 90% of $V_{DD}$ while all other signal pins are held at a logic 0.

When neighboring signal pins switch they change the effective capacitance that the victim signal experiences. This can cause the victim signal risetime to speed up, slow down, or remain the same. The fraction of risetime alteration is positive when the original risetime is sped up due to neighboring signal pins that switch. The fraction of risetime alteration is negative when the original risetime is slowed down due to neighboring signal pins that switch. Equation 3.3 gives the expression for how $P_{BW}$ is defined. Figure 3.9 illustrates the mechanism of bandwidth limiting and how it relates to $P_{BW}$.

$$P_{BW} = \frac{t_{orig} - t_{new}}{t_{orig}} \tag{3.3}$$

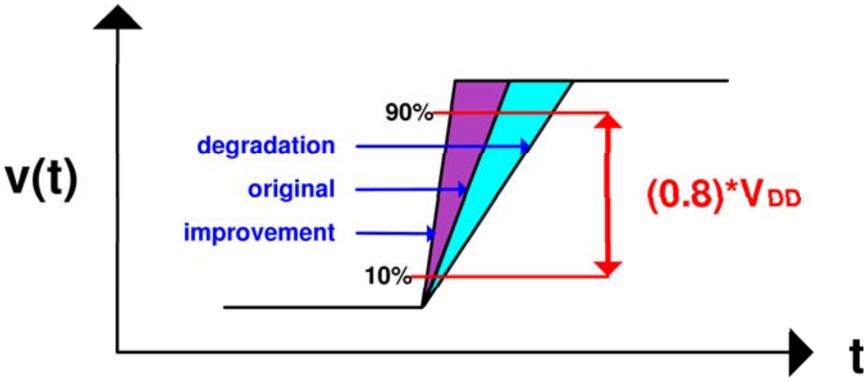


Figure 3.9: Bandwidth Limitation Notation

# Chapter 4

# Analytical Model for Off-Chip Bus Performance

The noise sources described in Section 1.2 limit the maximum performance that an off-chip bus can achieve. This occurs because the package noise sources are proportional to the rate of change in current or voltage ($\frac{di}{dt}$ and $\frac{dv}{dt}$). When any given noise source exceeds the user-defined limits described in Section 3.5, the rate of change of current or voltage must be reduced.

## 4.1 Package Performance Metrics

The performance of an off-chip bus depends on how fast the digital output drivers can switch the output voltage levels. This switching speed relates to how much data that one pin of the bus can transmit per second. The term *Unit Interval* (UI) represents the shortest amount of time that data can be present on an individual pin and still accurately transmit the logic value from the Transmitter (Tx) to the Receiver (Rx) in the off-chip bus. The UI of a pin is directly related to the maximum switching speed that the system can achieve. Figure 4.1 shows a graphical representation of the Unit Interval metric.

The minimum Unit Interval can be translated into the maximum *Datarate* (DR) of an individual pin using Equation 4.1.

$$DR_{max} = \frac{1}{UI_{min}} \tag{4.1}$$

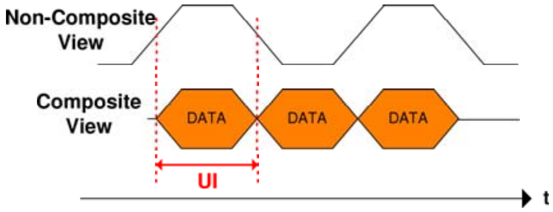


Figure 4.1: Unit Interval Description

Since the off-chip bus segment is constructed using multiple signal pins, the maximum *Throughput* (TP) of the bus segment can be found using Equation 4.2. Figure 4.2 shows a graphical representation of throughput.

$$TP_{max} = DR_{max} \cdot W_{bus} \tag{4.2}$$

## 4.2 Converting Performance to Risetime

To express the package performance metrics *UI*, *DR*, and *TP* in terms of rate of change of current or voltage, they must first be converted into the metric of *risetime.* The risetime ($t_r$) is defined as the time it takes to switch a signal from 10% of $V_{DD}$ to 90% of $V_{DD}$ (a total excursion of $0.8 \cdot V_{DD}$). Figure 4.3 shows the risetime definition.

When a digital signal switches, the signal must be given adequate time to reach its steady state value before the next switching event occurs. The amount of time that

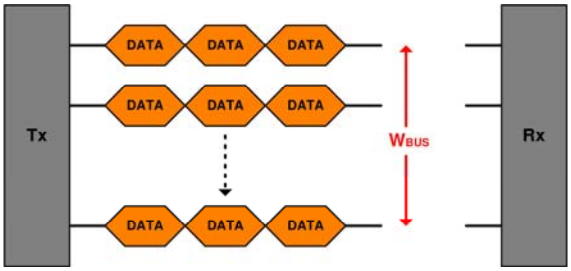


Figure 4.2: Bus Throughput Description

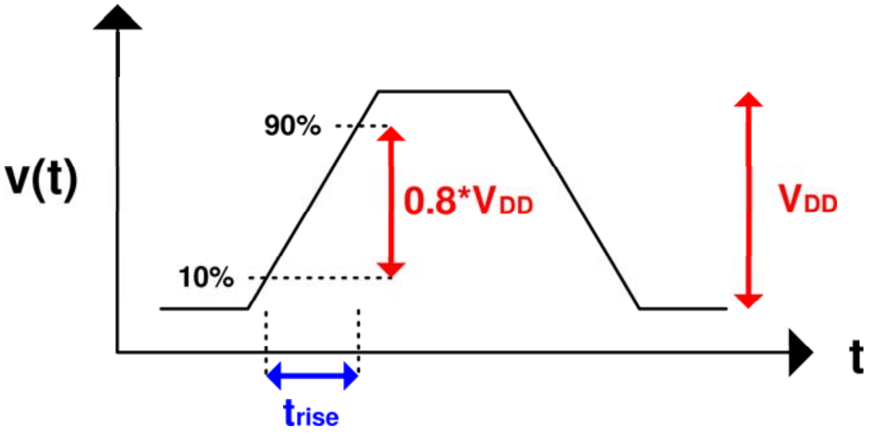


Figure 4.3: Risetime Description

is required is $UI_{min}$ and can be expressed in terms of risetime (Equation 4.3). For a robust digital system, the Unit Interval must be greater than or equal to 1.5 times the risetime [64, 65]. Figure 4.4 shows the graphical representation of how risetime relates to the minimum Unit Interval.

$$UI_{min} = 1.5 \cdot t_{rise} \tag{4.3}$$

## 4.3 Converting Bus Performance to $\frac{di}{dt}$ and $\frac{dv}{dt}$

Once the package performance has been expressed in terms of risetime, it can then be converted into the rate of change of current or voltage. The first step is to convert risetime into *slewrate.* Slewrate is defined as the rate of change in voltage ($\frac{dv}{dt}$) and typically is expressed in the units of $\frac{V}{ns}$. Slewrate can also be converted to the corresponding $\frac{di}{dt}$ by dividing it by the characteristic impedance of the system in which the package resides. Equation 4.4 expresses slewrate in terms of the rate of change in current or voltage. Figure 4.5 shows a graphical representation of slewrate.

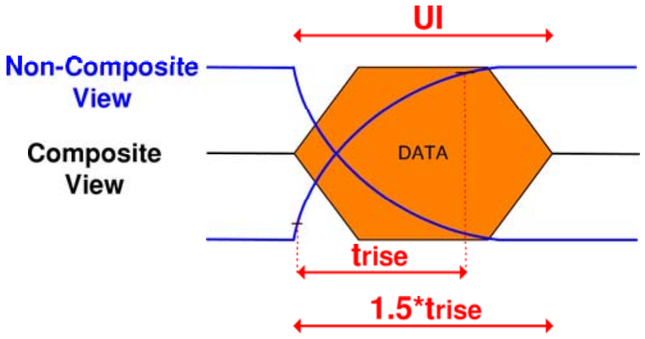


Figure 4.4: Risetime to Unit Interval Conversion

$$slewrate = \frac{dv}{dt} = \frac{di}{dt} \cdot Z_0 \tag{4.4}$$

Using the definition of risetime as the time taken to switch from 10% of $V_{DD}$ to 90% of $V_{DD}$ (an excursion of $0.8 \cdot V_{DD}$), risetime can be expressed in terms of slewrate by Equation 4.5.

$$t_{rise} = \frac{0.8 \cdot V_{DD}}{slewrate} \tag{4.5}$$

This, in turn, can be converted to a relationship between the package performance and the rate of change of current or voltage using Equations 4.6 and 4.7.

$$DR_{max} = \frac{(\frac{dv}{dt})}{(1.5) \cdot (0.8) \cdot (V_{DD})} = \frac{(\frac{di}{dt}) \cdot Z_0}{(1.5) \cdot (0.8) \cdot (V_{DD})} \tag{4.6}$$

$$TP_{max} = \frac{(\frac{dv}{dt}) \cdot W_{bus}}{(1.5) \cdot (0.8) \cdot (V_{DD})} = \frac{(\frac{di}{dt}) \cdot Z_0 \cdot W_{bus}}{(1.5) \cdot (0.8) \cdot (V_{DD})} \tag{4.7}$$

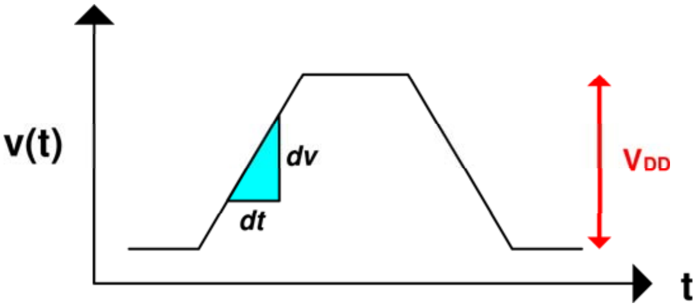


Figure 4.5: Slewrate Description

## 4.4 Translating Noise Limits to Performance

Now that the noise sources described in Section 1.2 and the package performance metrics in Section 4.1 have been expressed in terms of the rate of change in current or voltage, a relationship between the noise limits and package performance can be constructed. For each noise source, the maximum datarate and throughput is found that does not violate any user-defined noise limit parameter. The noise source that approaches its user-defined limit first is the limiting source to performance.

### 4.4.1 Inductive Supply Bounce

When considering the performance limitation due to inductive supply bounce the fastest rate of $\frac{di}{dt}$ is found that results in the user-defined noise limits $P_{supply}$ or $P_{gnd}$ being violated. The first step is to find an expression for the total amount of supply bounce that results as a consequence of signal switching. The total amount of supply bounce is evaluated over one bus *segment*. The evaluation must include all signals within the bus segment ($W_{bus}$) that return current through the supply pin inductance ($L_{11}$). In addition, any mutual inductive and capacitive signal coupling onto the supply pin must be accounted for. Supply bounce and mutual inductive coupling are already in terms of $\frac{di}{dt}$; however, mutual capacitive coupling must first be converted into an expression that considers the rate of change of current or voltage. In order to do this, Equation 1.9 is modified to include how the forcing function ($t_{rise}$) interacts with the natural response ($\tau_{RC}$) of the capacitively coupled circuit [64, 74]. Performing the inverse Laplace Transform on an exponential ramp forcing function and assuming an RC natural response, the capacitive voltage divider is multiplied by ($\tau_{RC}/t_{rise}$) to account for the risetime of the input voltage source. Equation 4.8 expresses the magnitude of the capacitively coupled voltage.

$$V_{vic} = (\frac{C_{1k}}{C_{1k} + C_0}) \cdot \Delta V_{agr} \cdot (\frac{\tau_{RC}}{t_{rise}}) \tag{4.8}$$

The time constant of the capacitively coupled circuit is given by Equation 4.9.

$$\tau_{RC} = Z_0 \cdot (C_{1k} + C_0) \tag{4.9}$$

Using Equation 4.5 to express $t_{rise}$ in terms of $\frac{dv}{dt}$ (and $\frac{di}{dt}$) and substituting $V_{DD}$ for $\Delta V_{agr}$, the magnitude of the capacitively coupled signal can be expressed as in Equation 4.10.

$$V_{vic} = \frac{C_{1k} \cdot Z_0 \cdot (\frac{dv}{dt})}{(0.8)} = \frac{C_{1k} \cdot Z_0^2 \cdot (\frac{di}{dt})}{(0.8)} \tag{4.10}$$

Equation 4.11 [1] expresses the total amount of supply bounce in an off-chip segment. This expression represents the worst-case supply bounce pattern, which occurs when all signals within the segment switch from a logic 0 to a logic 1. This bus pattern means that all of the signal pins will charge their outputs from a 0 to a 1 by drawing current through the $V_{DD}$ pin. In addition, this pattern causes mutual inductive and capacitive coupling to occur on the $V_{DD}$ pin. This coupling adds to the total supply bounce noise in the system.

[1] Note in this equation that when the subscript evaluates to $M_{11}$ or $C_{11}$, these values are set equal to zero.

$$V_{Supply-Bnc} = (\frac{L_{11} \cdot W_{bus}}{N_p}) \cdot (\frac{di}{dt}) + \sum_{k=-p_L}^{p_L} M_{1(|k|+1)} \cdot (\frac{di}{dt}) + \sum_{k=-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)} \cdot (\frac{di}{dt}) \quad (4.11)$$

We set Equation 4.11 to the user-defined maximum allowed supply bounce (which is expressed as a fraction $P_{supply}$ of $V_{DD}$), to get Equation 4.12.

$$P_{supply} \cdot V_{DD} = (\frac{L_{11} \cdot W_{bus}}{N_p}) \cdot (\frac{di}{dt}) + \sum_{k=-p_L}^{p_L} M_{1(|k|+1)} \cdot (\frac{di}{dt}) + \sum_{k=-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)} \cdot (\frac{di}{dt}) \quad (4.12)$$

By factoring $\frac{di}{dt}$ out of Equation 4.12 and substituting into Equation 4.6, the maximum datarate that can be achieved without violating the user-defined noise limit for supply bounce can be found (Equation 4.13).

$$DR_{max-supply} = \frac{P_{supply} \cdot Z_0}{(1.5) \cdot (0.8) \cdot [(\frac{L_{11} \cdot W_{bus}}{N_p}) + \sum_{-p_L}^{p_L} M_{1(|k|+1)} + \sum_{-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}]} \quad (4.13)$$

This, in turn, can be translated into the maximum throughput of the segment by multiplying it by the number of signals within the segment (Equation 4.14).

$$TP_{max-supply} = \frac{P_{supply} \cdot Z_0 \cdot W_{bus}}{(1.5) \cdot (0.8) \cdot [(\frac{L_{11} \cdot W_{bus}}{N_p}) + \sum_{-p_L}^{p_L} M_{1(|k|+1)} + \sum_{-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}]} \quad (4.14)$$

In a similar manner, the maximum datarate and throughput for a system (such that the user-defined noise limit $P_{gnd}$ is not violated) can be derived. These expressions are given in Equations 4.15 and 4.16.

$$DR_{max-gnd} = \frac{P_{gnd} \cdot Z_0}{(1.5) \cdot (0.8) \cdot [(\frac{L_{11} \cdot W_{bus}}{N_g}) + \sum_{-p_L}^{p_L} M_{1(|k|+1)} + \sum_{-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}]} \quad (4.15)$$

$$TP_{max-gnd} = \frac{P_{gnd} \cdot Z_0 \cdot W_{bus}}{(1.5) \cdot (0.8) \cdot [(\frac{L_{11} \cdot W_{bus}}{N_g}) + \sum_{-p_L}^{p_L} M_{1(|k|+1)} + \sum_{-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}]} \quad (4.16)$$

### 4.4.2 Capacitive Bandwidth Limiting

When considering the capacitive bandwidth limitation in the package, the slowest RC time constant is found which corresponds to the worst-case switching pattern on the bus. This time constant is then translated into risetime, which in turn is translated into datarate and throughput as in the previous section. For capacitive bandwidth limitation, the worst-case pattern on the bus occurs when the center pin within the segment transitions from a logic 0 to a 1 while all other signals switch from a 1 to a 0 (or vice versa). This switching pattern results in the highest effective capacitance on the victim signal. This capacitance will result in the slowest risetime since it results in the largest RC time constant for the bus segment. This dictates the maximum datarate and throughput that can be achieved in the package. Combining Equations 1.7 and 1.8 gives the maximum risetime that a given package can achieve when considering capacitive bandwidth limitation.

$$t_{rise-BW} = 2.2 \cdot Z_0 \cdot (C_0 + \sum_{k=-p_C}^{p_C} 2 \cdot C_{1(|k|+1)}) \tag{4.17}$$

This can be directly converted to datarate and throughput using Equation 4.3. Equations 4.18 and 4.19 give the expressions for the maximum datarate and throughput that a package can achieve when considering bandwidth limitation.

$$DR_{max-BW} = \frac{1}{(1.5) \cdot (2.2) \cdot Z_0 \cdot (C_0 + \sum_{-p_C}^{p_C} 2 \cdot C_{1(|k|+1)})} \tag{4.18}$$

$$TP_{max-BW} = \frac{W_{bus}}{(1.5) \cdot (2.2) \cdot Z_0 \cdot (C_0 + \sum_{-p_C}^{p_C} 2 \cdot C_{1(|k|+1)})} \tag{4.19}$$

### 4.4.3 Signal Coupling

The derivation of the signal coupling magnitude is performed exactly as in Section 4.4.1. Equation 4.20 expresses the maximum amount of noise voltage due to mutually inductive and capacitive signal coupling onto a victim line. This expression describes the coupled noise when the victim signal is either static ($P_0$), rising ($P_1$), or falling ($P_{-1}$).

$$V_{coupling} = \sum_{k=-p_L}^{p_L} M_{1(|k|+1)} \cdot (\frac{di}{dt}) + \sum_{k=-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)} \cdot (\frac{di}{dt}) \tag{4.20}$$

Equating this to the user-defined maximum allowed coupled noise (which is expressed as a fraction $P_0$, $P_1$, or $P_{-1}$ of $V_{DD}$), we get Equation 4.21.

$$P_{(0/-1/1)} \cdot V_{DD} = \sum_{k=-p_L}^{p_L} M_{1(|k|+1)} \cdot (\frac{di}{dt}) + \sum_{k=-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)} \cdot (\frac{di}{dt}) \tag{4.21}$$

By factoring $\frac{di}{dt}$ out of Equation 4.21 and substituting into Equation 4.6, the maximum datarate that can be achieved without violating the user-defined noise limits for signal coupling can be found (Equation 4.22).

$$DR_{max-coupling} = \frac{P_{(0/-1/1)} \cdot Z_0}{(1.5) \cdot (0.8) \cdot [\sum_{-p_L}^{p_L} M_{1(|k|+1)} + \sum_{-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}]} \tag{4.22}$$

Equation 4.23 expresses the maximum throughput when considering the user-defined noise limits for signal coupling.

$$TP_{max-coupling} = \frac{P_{(0/-1/1)} \cdot Z_0 \cdot W_{bus}}{(1.5) \cdot (0.8) \cdot [\sum_{-p_L}^{p_L} M_{1(|k|+1)} + \sum_{-p_C}^{p_C} \frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}]} \tag{4.23}$$

### 4.4.4 Impedance Discontinuities

When evaluating a package for transmission line reflections due to impedance discontinuities, the forcing function and natural response of the circuit must be considered in the evaluation of Equations 1.13 and 1.14. To accomplish this we assume that the natural response of the package behaves as a low-pass RC filter. This allows the same derivations made in Section 4.4.1 to be applied to the evaluation of the transmission line reflections. Equation 4.24 expresses the magnitude of the reflection, considering the forcing function and the natural response of the package.

$$V_{reflected} = V_{incident} \cdot (\Gamma) \cdot (\frac{\tau_{RC}}{t_{rise}}) \tag{4.24}$$

Substituting the RC time constant given in Equation 4.9 and replacing $V_{incident}$ with $V_{DD}$, the reflected voltage can be expressed in terms of the user-defined noise limit (which is expressed as a fraction $P_\Gamma$ of $V_{DD}$).

$$P_\Gamma \cdot V_{DD} = V_{DD} \cdot (\Gamma) \cdot (\frac{Z_0 \cdot (\sum_{-p_C}^{p_C} C_{1(|k|+1)} + C_0)}{t_{rise}}) \tag{4.25}$$

This can be substituted into Equation 4.6 to find the maximum datarate that does not violate the user-defined noise limit $P_\Gamma$. Equations 4.26 and 4.27 give the datarate and throughput when considering the maximum allowable noise due to impedance discontinuities.

$$DR_{max-\Gamma} = \frac{P_\Gamma}{(1.5) \cdot \Gamma \cdot Z_0 \cdot (\sum_{-p_C}^{p_C} C_{1(|k|+1)} + C_0)} \tag{4.26}$$

$$TP_{max-\Gamma} = \frac{P_\Gamma \cdot W_{bus}}{(1.5) \cdot \Gamma \cdot Z_0 \cdot (\sum_{-p_C}^{p_C} C_{1(|k|+1)} + C_0)} \tag{4.27}$$

## 4.5 Experimental Results

In order to verify the analytical model, SPICE [76] simulations were performed on a test circuit. The test circuit includes a model for the package parasitics described in Table 2.1. The test circuit is used to monitor noise limit violations due to the package as the performance of the output driver is increased. By monitoring noise limit violations the simulation can indicate the maximum datarate and throughput for a given package and bus configuration. These simulation results are compared to the analytical model predictions to verify the model's accuracy.

### 4.5.1 Test Circuit

The test circuit used in this analysis consists of a standard CMOS inverter as the output driver. The CMOS inverter is implemented using the BPTM $0.1\mu m$ [78] technology using BSIM3 model cards [77]. The CMOS inverter is sized to have an equal drive strength when outputting a logic 0 or a logic 1 by sizing the PMOS and NMOS transistors such that $(\frac{W_P}{W_N}) = (\frac{u_n}{u_p}) = 3.25$ [67]. The CMOS inverter drives through the Tx package model onto a PCB. The PCB contains a series termination resistor ($R_s$) followed by 2" of $50\Omega$ transmission line. The receiver load is modeled using the input to a CMOS inverter that is sized to have a gate capacitance of $3pF$. A package model is also inserted in the signal and power path prior to the Rx inverter. Both the Tx and Rx inverters have $V_{DD}$=1.5v and $V_{SS}$=0v. The output risetime of the transmitter is altered by resizing the CMOS transistors in the inverter. Increasing the size of the transistor will result in a smaller risetime. Figure 4.6 shows the test circuit topology.

For each of the three packages studied in this work (QFP-WB, BGA-WB, and BGA-FC), the number of switching signal pins within a segment are varied from $W_{bus} = 1$ to $W_{bus} = 16$ in the simulation. In addition, three different SPG configurations are used to observe the effect of adding power and ground pins to the segment. The three SPG configurations used are SPG=8:1:1, SPG=4:1:1, and SPG=2:1:1. The user-defined noise limits are set to 5% of $V_{DD}$ so that $P_{supply}$=$P_{gnd}$=$P_0$=$P_1$=$P_{-1}$=-$P_{BW}$=$P_{\Gamma}$=0.05. It was found that for the three packages studied, supply bounce ($P_{supply}$, $P_{gnd}$) was always the limiting factor to performance.

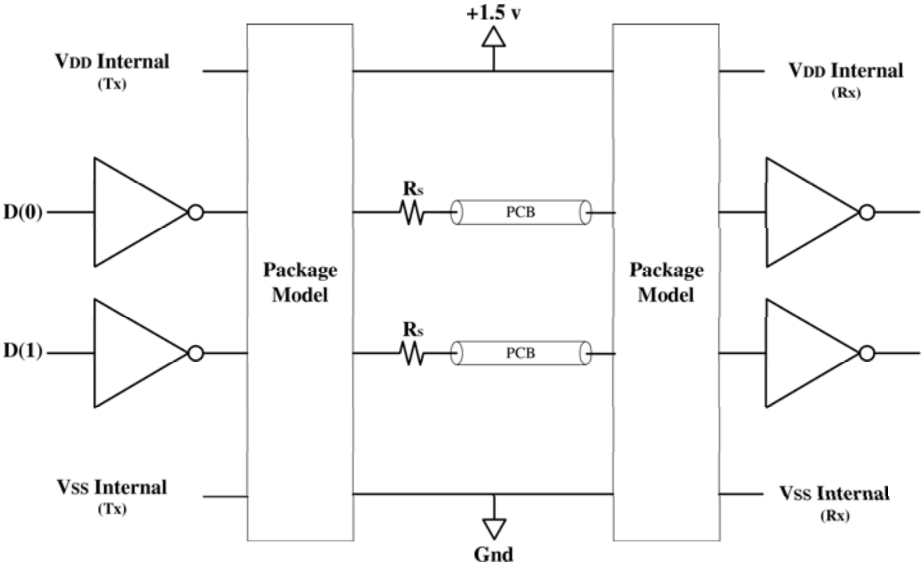


Figure 4.6: Test Circuit used to Verify Analytical Model

### 4.5.2 Quad Flat Pack with Wire Bonding Results

Figure 4.7 shows the per-pin datarate for the QFP-WB package as a function of the number of signal pins within the segment. This plot shows the predicted datarate for the analytical model versus the simulated results. These results illustrate that the per-pin datarate needs to be reduced as channels are added to the bus segment in order to keep the noise below the user-defined limits. This plot also shows the amount of ground and power pins per segment influences the performance of the segment. Segments with a lower SPR perform at higher datarates due to the reduction in the total $\frac{di}{dt}$ that is drawn through each supply pin.

Figure 4.8 shows the bus throughput for the QFP-WB package as a function of the number of signals pins within the segment. Due to the reduction in per-pin datarate as channels are added to the segment, throughput does not increase linearly with the addition of channels; instead, the throughput follows a less than linear increase as pins are added to the segment.

Figure 4.9 shows the error percentage between the simulation results and analytical model prediction for each size of segment evaluated in the QFP-WB package. This plot illustrates the accuracy of the analytical model. In almost all cases the accuracy of the model is within 10% of the simulated results. The analytical model exhibits the largest error for a bus size of 1 channel due to the higher frequency components being present in the forcing function; however, even with an error of 23% for 1 channel, the correlation of the analytical model is still within an acceptable range.

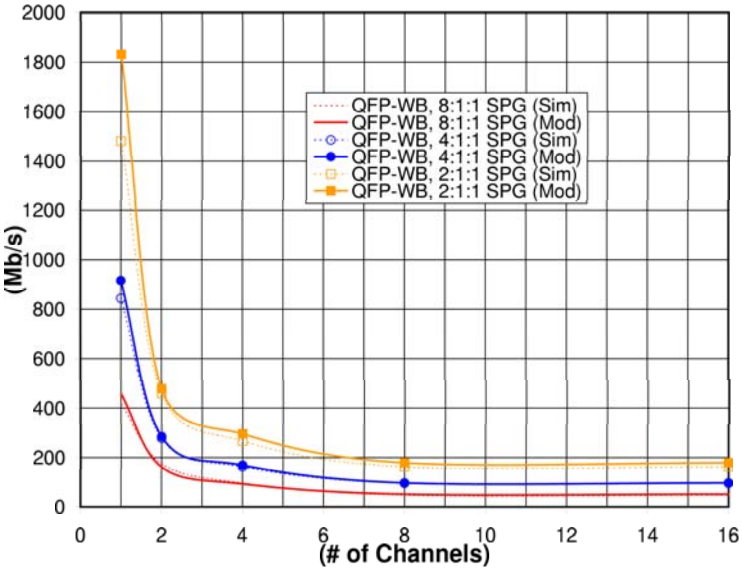


Figure 4.7: Per-Pin Datarate for a QFP-WB Package

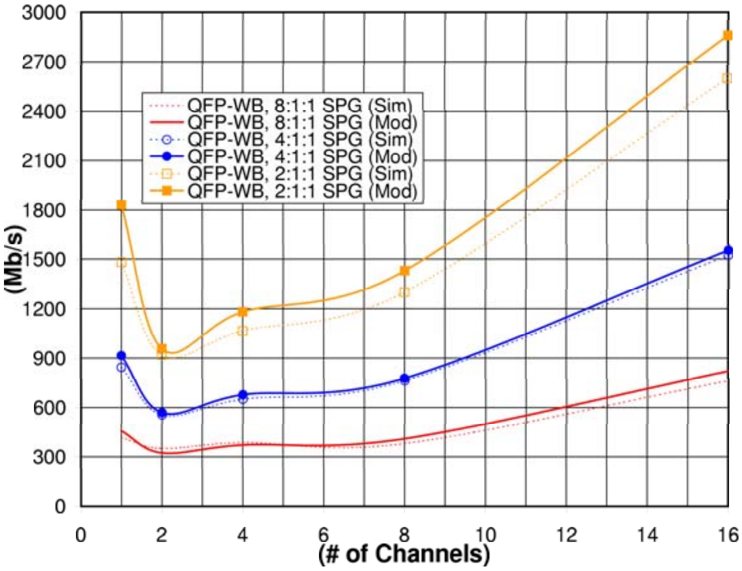


Figure 4.8: Throughput for a QFP-WB Package

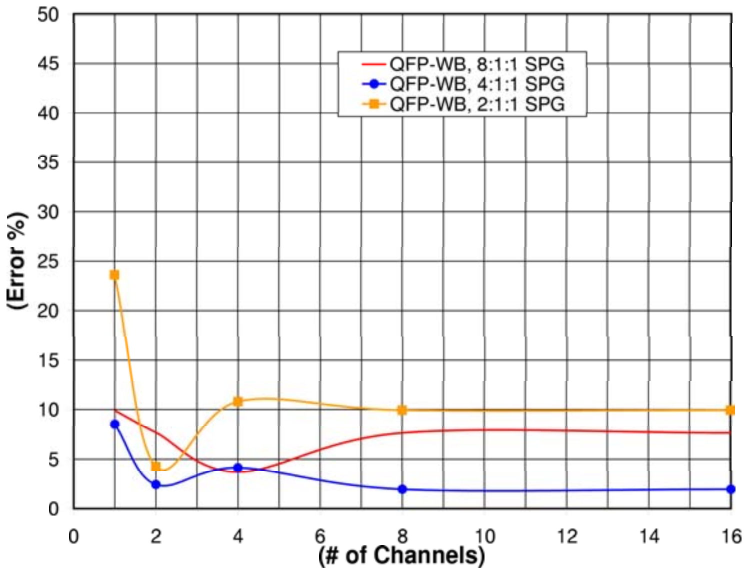


Figure 4.9: Model Accuracy for a QFP-WB Package

### 4.5.3 Ball Grid Array with Wire Bonding Results

Figure 4.10 shows the per-pin datarate results for the BGA-WB package. This figure illustrates how the reduction in the electrical parasitics of the level 2 interconnect translates into increased performance. By moving to the BGA-WB package with less electrical parasitics, the per-pin datarate is increased as much as 25% over the QFP-WB. As with the results for the QFP-WB, performance is also increased by using a lower SPG Ratio. Also, the per-pin datarate decreases with more switching channels as was observed for the QFP-WB package, for the same reasons.

Figure 4.11 shows the throughput for the BGA-WB package. Again, the throughput experiences a less than linear increase in performance as channels are added to the bus segment. In addition, the throughput suffers a plateau at low channel counts at which the throughput is not increased as channels are added. This behavior leads to multiple bus configurations having the same throughput. In these cases the narrowest bus configuration can be chosen to reduce cost.

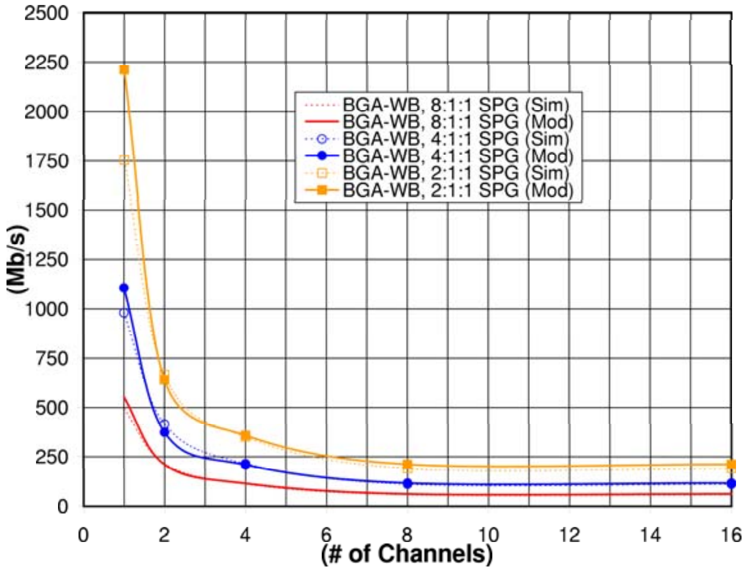


Figure 4.10: Per-Pin Datarate for a BGA-WB Package

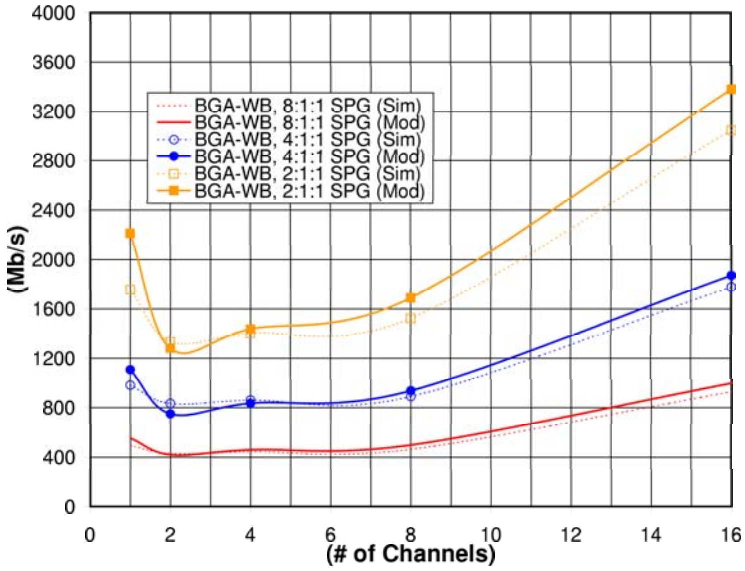


Figure 4.11: Throughput for a BGA-WB Package

Figure 4.12 shows the amount of error between the analytical model prediction and the simulated results for the BGA-WB package. Again, the most error occurs for segments containing only one signal pin. For channel counts greater than 1, the model accuracy is consistently below 11%.

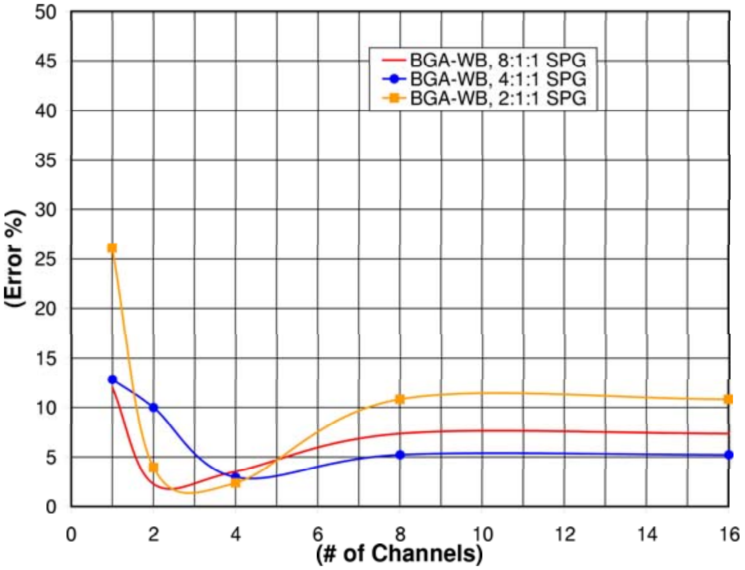


Figure 4.12: Model Accuracy for a BGA-WB Package

### 4.5.4 Ball Grid Array with Flip-Chip Bumping Results

Figure 4.13 shows the per-pin datarate results for the BGA-FC package. The BGA-FC package is the most advanced package studied in this work. By implementing flip-chip bumping in the level 1 interconnect in addition to a BGA in the level 2 interconnect, this package is able to achieve much higher performance over the QFP-WB and BGA-WB. The per-pin datarate achieves a 260% increase in performance over the BGA-WB package and a 333% increase over the QFP-WB. Despite its much higher performance, the BGA-FC still suffers a reduction in per-pin datarate as signals are added to the segment. A lower SPG also results in a higher per-pin datarate, as with the other packages studied.

Figure 4.14 shows the throughput for the BGA-FC package. Again, the throughput increases at a less than linear rate as channels are added to the segment. In addition, the BGA-FC package also exhibits the phenomenon where multiple bus configurations achieve very similar throughputs.

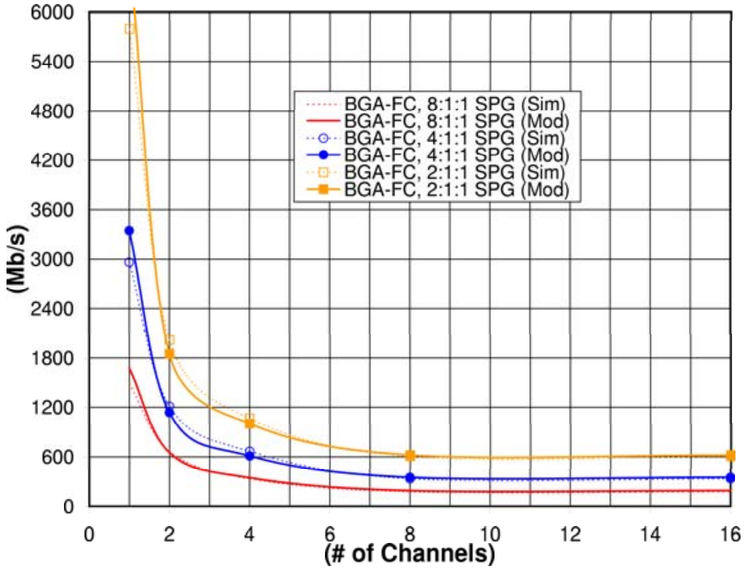


Figure 4.13: Per-Pin Datarate for a BGA-FC Package

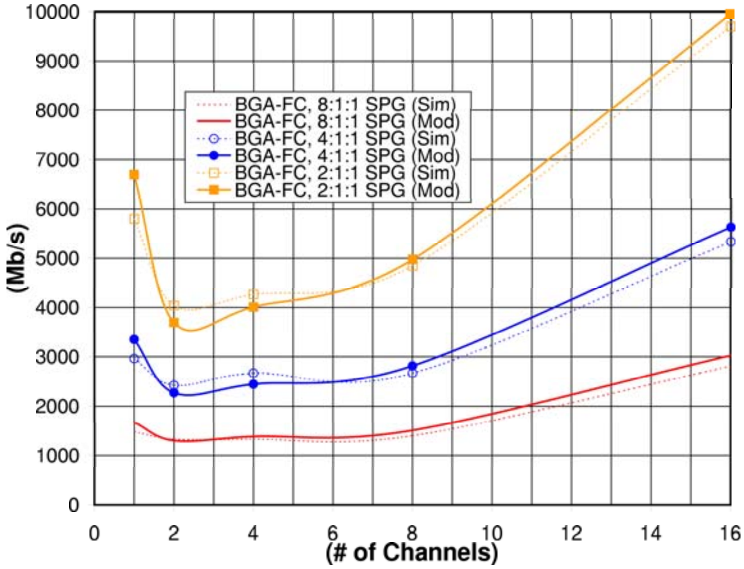


Figure 4.14: Throughput for a BGA-FC Package

Figure 4.15 shows the accuracy of the analytical model for the BGA-FC package. The worst-case model error occurs at a segment size of 1 while all other configurations achieve less than a 10% error.

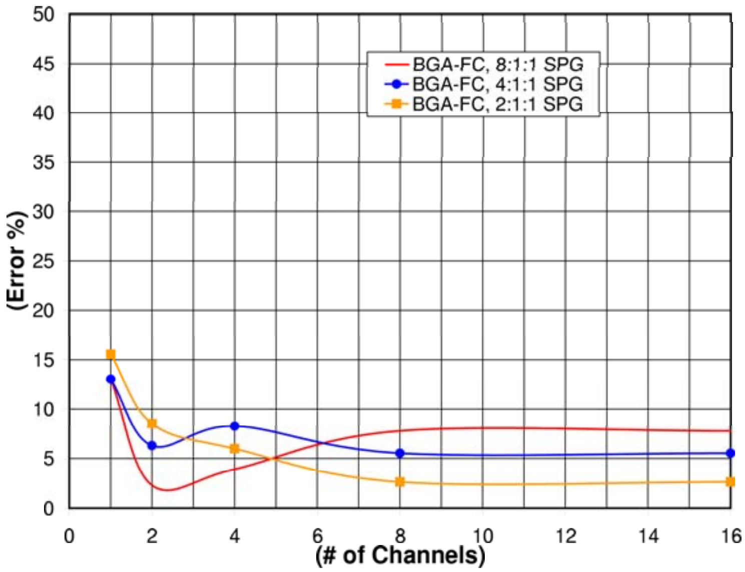


Figure 4.15: Model Accuracy for a BGA-FC Package

### 4.5.5 Discussion

Since the framework of this model is constructed in a parameterized and scalable fashion, it can be applied to differential as well as single ended signaling with minimal alteration. In differential signaling, a portion of the return current for a given signal pin will flow through its complementary pin within the differential pair. In this special case, the amount of supply bounce noise due to current being drawn through the inductance of the $V_{DD}$ or $V_{SS}$ pin is reduced by the amount of current that is inherently returned within each differential pair of the segment. The advantage of differential signaling is that since a portion of the return current is provided within the pair itself, this current does not flow through the inductance of the supply pin and does not result in supply bounce noise. All of the other noise sources that are predicted within this model do not require modification when using differential signaling.

These experimental results illustrate how the electrical parasitics of the IC package limit the overall system performance. The effect of increasing the number of simultaneously switching signals is that the per-pin performance is significantly reduced. In most cases a desired bus throughput can be achieved using multiple package and pin configurations, including different signal counts within a given segment. For these cases we must factor in the cost of each package and pin configuration in order to select the most economical bus design for a required throughput. In the next chapter, a methodology is provided to select the most cost-effective bus configuration for a given throughput requirement.

# Chapter 5

# Optimal Bus Sizing

The package performance results reported in Chapter 4 revealed that simply adding signals to a bus does not necessarily increase the throughput of the system. As more channels are added to the bus, the noise caused by simultaneously switching signals reduces the per-pin datarate. This behavior leads to a less than linear increase in system performance as channels are added. In some cases, the system performance actually decreases as more channels are added. In practice, multiple packages, configurations, and segment sizes are able to achieve the same throughput. In this chapter, a method is described for choosing the package, bus size, and SPG in the most cost-effective manner for a given throughput requirement.

## 5.1 Package Cost

The use of advanced packaging will reduce the electrical parasitics due to the package. This reduces the package noise in the system which leads to increased performance; however, moving toward advanced packaging is often more expensive and can result in a cost prohibitive increase in the majority of modern VLSI designs.

Table 5.1 lists the average cost per-pin ($Cost_{per-pin}$) for the three packages that have been studied in this work [92]. The QFP-WB package has the least cost per pin due to its mature manufacturing process and scalable interconnect; however, this package suffers the worst performance due to the relatively large inductance in the lead frame

| Package | $Cost_{per-pin}$ |
|---|---|
| QFP-WB | \$0.22 |
| BGA-WB | \$0.34 |
| BGA-FC | \$0.63 |

Table 5.1: Package I/O Cost (US Dollars, \$)

and wire bond interconnect. The BGA-WB improves upon the lead frame connection by moving toward a ball grid array. The BGA interconnect reduces the inductance in the level 2 interconnect (Table 2.1) but increases the per pin cost over the QFP-WB by 55%. The BGA-FC package experiences the best electrical performance by using flip-chip bumping for its level 1 interconnect (instead of wire bonding). The inductance in the BGA-FC package is significantly reduced over both the QFP-WB and BGA-WB as shown in Chapter 2; however, the improvement in performance comes at an increase in cost of 85% over the BGA-WB and 286% over the QFP-WB.

Table 5.2 lists the total number of pins that are needed to implement the various SPG configurations used in this work. The number of pins needed to implement a bus increases at a faster-than-linear rate as signals are added due to the need for a pair of $V_{DD}$ and $V_{SS}$ within each segment.

The total cost of the bus configuration is given by Equation 5.1 where the cost is simply the per-pin cost multiplied by the number of pins needed to implement the bus segment. Table 5.3 shows the cost of the various bus configurations. The effect of choosing a better grounding and power scheme (i.e., a lower SPGR) is that the cost increases at a faster-than-linear rate as channels are added. This table shows the relative expense of the different packages considered and illustrates how moving toward advanced packaging can lead to a significant cost increase.

$$Cost_{bus} = (N_{bus}) \cdot (Cost_{per-pin}) \tag{5.1}$$

| | Number of Channels | | | | |
|---|---|---|---|---|---|
| Bus Configuration | 1 | 2 | 4 | 8 | 16 |
| QFP-WB 8:1:1 | 3 | 4 | 6 | 10 | 20 |
| QFP-WB 4:1:1 | 3 | 4 | 6 | 12 | 24 |
| QFP-WB 2:1:1 | 3 | 4 | 8 | 16 | 32 |
| BGA-WB 8:1:1 | 3 | 4 | 6 | 10 | 20 |
| BGA-WB 4:1:1 | 3 | 4 | 6 | 12 | 24 |
| BGA-WB 2:1:1 | 3 | 4 | 8 | 16 | 32 |
| BGA-FC 8:1:1 | 3 | 4 | 6 | 10 | 20 |
| BGA-FC 4:1:1 | 3 | 4 | 6 | 12 | 24 |
| BGA-FC 2:1:1 | 3 | 4 | 8 | 16 | 32 |

Table 5.2: Number of Pins Needed Per Bus Configuration

| | Number of Channels | | | | |
|---|---|---|---|---|---|
| Bus Configuration | 1 | 2 | 4 | 8 | 16 |
| QFP-WB 8:1:1 | 0.66 | 0.88 | 1.32 | 2.20 | 4.40 |
| QFP-WB 4:1:1 | 0.66 | 0.88 | 1.32 | 2.62 | 5.28 |
| QFP-WB 2:1:1 | 0.66 | 0.88 | 1.76 | 3.52 | 7.04 |
| BGA-WB 8:1:1 | 1.02 | 1.36 | 2.04 | 3.40 | 6.80 |
| BGA-WB 4:1:1 | 1.02 | 1.36 | 2.04 | 4.08 | 8.16 |
| BGA-WB 2:1:1 | 1.02 | 1.36 | 2.72 | 5.44 | 10.88 |
| BGA-FC 8:1:1 | 1.89 | 2.52 | 3.78 | 6.30 | 12.60 |
| BGA-FC 4:1:1 | 1.89 | 2.52 | 3.78 | 7.56 | 15.12 |
| BGA-FC 2:1:1 | 1.89 | 2.52 | 5.04 | 10.08 | 20.16 |

Table 5.3: Total Cost for Various Bus Configurations ($)

## 5.2 Bandwidth per Cost

In order to compare the cost-effectiveness of a package configuration, the metric *Bandwidth-per-Cost* $(BPC)$ is introduced. This metric has units of $(\frac{Mb}{\$})$ and takes into account the total bus throughput as well as the cost to implement a given bus configuration. Equation 5.2 gives the definition of the BPC metric. When comparing two busses the configurations, the configuration with a higher BPC indicates that it is able to transmit more data for less cost.

$$BPC = (\frac{TP}{Cost_{bus}}) \tag{5.2}$$

### 5.2.1 Results for Quad Flat Pack with Wire Bonding

Figure 5.1 shows the *Bandwidth-per-Cost* for the QFP-WB package as signals are added to the segment. This figure illustrates that it is more cost-effective to construct a bus that is narrow (and fast) rather than the traditional wider (and slower) configuration. These results match closely with the current industrial trends for high-end computer busses. Busses such as HyperTransport [85] and PCI Express [84] have taken advantage of the fact that narrower busses avoid the noise problems due to simultaneously switching signals, thus enabling higher per-pin datarates. At the same time, these busses use fewer pins which reduces the overall cost of the design. The value of the BPC metric is that it can be used to select the most cost-effective bus configuration for a particular application.

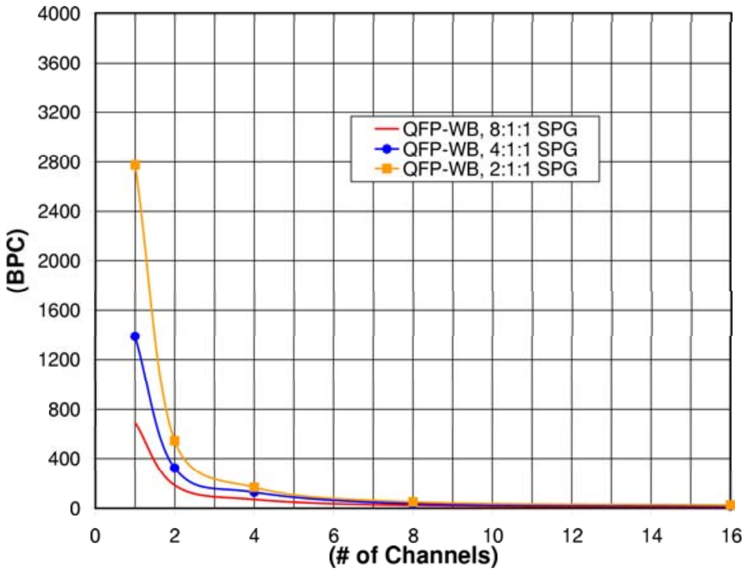


Figure 5.1: Bandwidth-per-Cost for a QFP-WB Package (Mb/$)

### 5.2.2 Results for Ball Grid Array with Wire Bonding

Figure 5.2 shows the *Bandwidth-per-Cost* for the BGA-WB package as signals are added to the segment. This plot again indicates that narrower busses are more cost-effective. This plot also illustrates that the BPC of the BGA-WB is actually less than the QFP-WB package. This is due to the fact that the increase in performance that results from using a BGA (instead of a lead frame) in the level 2 interconnect does not outweigh the increase in cost of the advanced package. This gives rise to the significant conclusion that for certain busses it may be more cost-effective to use a QFP-WB package rather than the more advanced BGA-WB package.

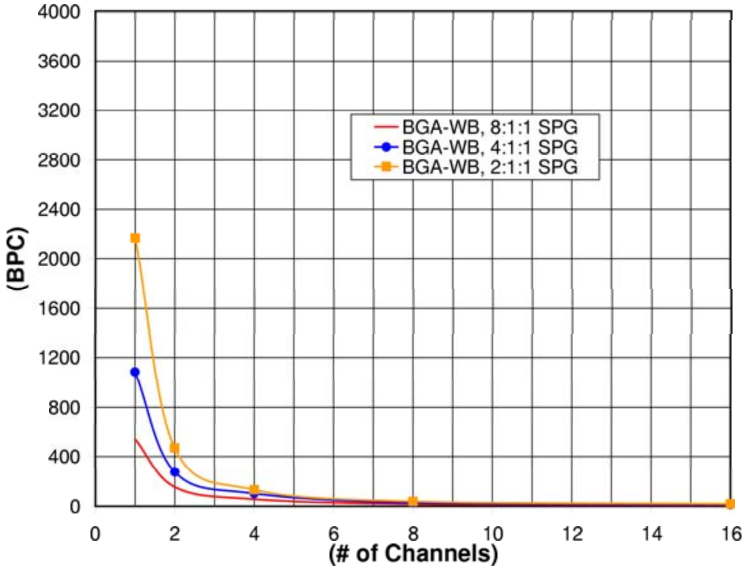


Figure 5.2: Bandwidth-per-Cost for a BGA-WB Package (Mb/$)

### 5.2.3 Results for Ball Grid Array with Flip-Chip Bumping

Figure 5.3 shows the *Bandwidth-per-Cost* for the BGA-FC package as a function of the number of signals in the bus segment. This plot again highlights that narrower busses are more cost-effective. Also, the BPC for the BGA-FC is much greater than for either the QFP-WB or the BGA-WB packages. This is due to the fact that the dramatic increase in performance gained by utilizing flip-chip technology outweighs the corresponding increase in cost.

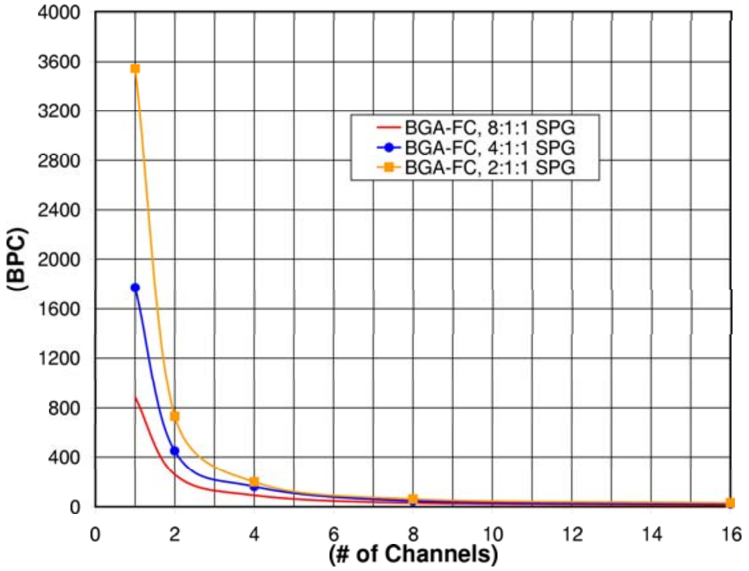


Figure 5.3: Bandwidth-per-Cost for a BGA-FC Package (Mb/$)

For all the packages studied in this work, it was found that faster, narrower busses were more cost-effective. This trend arises from the negative effects of simultaneously switching signals as channels are added to a bus. These effects make it impractical to simply add I/O pins until the desired throughput is reached. Faster and narrower busses offer the advantage of reduced Simultaneous Switching Noise (SSN), which translates into increased per-pin performance. In addition, since fewer package pins are used, the overall cost of such configurations is lower than that of the wider configuration (which has lower per-pin performance). This trend is being exhibited in industry as more and more computer busses move toward faster, narrower busses. This trend is often referred to as moving from *parallel* to *serial* busses. The unique contribution of this chapter is that it provides a quantitative methodology to find the most cost-effective bus configuration for a given application.

Suppose it is desired to find the most cost-effective package and configuration for a given bus. The performance model of Chapter 4, as well as the BPC model of this chapter can be used to determine if a particular configuration meets the desired performance requirements. The DR and TP parameters can be used to test if a particular configuration meets the desired performance requirements. When two or more configurations meet the performance requirements, their BPC values can next be used to select the most cost-effective configuration. Since these models avoid the use of SPICE, the determination of the most cost-effective bus configuration can be done very quickly. In addition, the BPC metric can provide a quick comparison of the cost-effectiveness of the available packaging options. The analytical models and sizing techniques presented can aid VLSI designers to quickly select the optimal off-chip bus configuration.

## 5.3 Bus Sizing Example

In order to illustrate the use of the bus sizing technique, consider the example below.

**Example** Consider an on-chip circuit that needs to transmit $2000\frac{Mb}{s}$ through the package. Table 5.4 lists the modeled throughput results for the three packages studied in this work. From this table, it is clear that multiple configurations can meet the $2000\frac{Mb}{s}$ throughput requirement. The bus configurations that meet the throughput requirement are:

- QFP-WB package with $W_{bus}$=16 and SPG=2:1:1
- BGA-WB package with $W_{bus}$=1 and SPG=2:1:1
- BGA-WB package with $W_{bus}$=16 and SPG=2:1:1
- BGA-FC package with $W_{bus}$=16 and SPG=8:1:1
- BGA-FC package with $W_{bus}$=1 through 16 and SPG=4:1:1
- BGA-FC package with $W_{bus}$=1 through 16 and SPG=2:1:1

Using Table 5.3, it is quickly found that the BGA-WB with $W_{bus}$=1 and SPG=2:1:1 is the least expensive configuration that will meet the throughput requirement. This is also apparent in Figures 5.1, 5.2, and 5.3 which shows that this configuration has a higher BPC than the QFP-WB and BGA-WB configurations. The BGA-FC configurations have higher BPC than the lowest cost BGA-WB configuration. Using the BGA-FC solution has the added advantage that the resulting solution provides additional throughput margin, which is useful in the event that the bus throughput requirement would increase in future revisions of the design.

| | Number of Channels | | | | |
|---|---|---|---|---|---|
| Bus Configuration | 1 | 2 | 4 | 8 | 16 |
| QFP-WB 8:1:1 | 458 | 324 | 372 | 410 | 820 |
| QFP-WB 4:1:1 | 916 | 569 | 677 | 777 | 1554 |
| QFP-WB 2:1:1 | 1832 | 959 | 1182 | 1430 | 2860 |
| BGA-WB 8:1:1 | 553 | 420 | 460 | 498 | 996 |
| BGA-WB 4:1:1 | 1106 | 750 | 835 | 935 | 1871 |
| BGA-WB 2:1:1 | 2213 | 1281 | 1437 | 1689 | 3378 |
| BGA-FC 8:1:1 | 1675 | 1303 | 1386 | 1513 | 3025 |
| BGA-FC 4:1:1 | 3349 | 2272 | 2446 | 2814 | 5628 |
| BGA-FC 2:1:1 | 6699 | 3696 | 4011 | 4976 | 9952 |

Table 5.4: Modeled Throughput Results for Packages Studied ($\frac{Mb}{s}$)

# Chapter 6

## Bus Expansion Encoder

When determining the performance of an off-chip bus, the worst-case noise magnitude must be considered in order to ensure a robust digital system. Chapter 4 presented an analytical model to predict the performance of an off-chip bus using the assumption that each of the noise sources within the package contributes its worst-case noise. Each source of noise within the package (Section 1.2) had a particular set of data sequences that resulted in the worst-case noise. This set of sequences dictates the highest performance that the package can achieve. The sequences that result in noise (from **any** noise source) above a certain magnitude can be avoided by designing an encoder which eliminates such sequences. This increases the performance of the package. By inserting this encoder in the signal path on the IC, it ensures that the off-chip data is encoded before traversing the package interconnect. In this way, data sequences which result in noise above a specified limit can be avoided and the bus performance can be increased.

The encoder is constructed to remove any bus sequence which results in a noise event (regardless of the noise source) above a user-specified value. Experimental results demonstrate that this methodology improves the overall performance of the bus even after considering the overhead of the encoder circuit.

The technique maps each element in the original set of on-chip data sequences to an element in an alternate set of sequences (whose noise is bounded by the user-specified limit). Since the alternate set of sequences has a width greater than the width of the original bus, we refer to the resulting circuit as a **bus expansion** encoder. The construction of the encoder/decoder utilizes an implicit, Reduced Ordered Binary Decision Diagram (ROBDD). If it is desired that the noise due to one or more noise sources (described in Section 1.2) should be reduced, the bus expansion encoding method can by employed.

## 6.1 Constraint Equations

The first step in creating the bus expansion encoder is to create a set of constraint equations. The constraint equations are written so that arbitrary transitions can be evaluated for noise limit violations. When a transition is evaluated using the constraint equations and violates one of the user-defined noise limits, the transition is flagged as *illegal* and is removed from the set of data sequences that are allowed to be driven through the package interconnect. Each of the possible off-chip transitions are evaluated against each of the constraint equations. The enumeration of the off-chip transitions is done implicitly, so as to increase the applicability of the technique. After the evaluation is complete, a subset of *legal* transitions remain which are used in the construction of the encoder.

### 6.1.1 Supply Bounce Constraints

When a pin $i$ in segment $j$ is a $V_{DD}$ pin, it is required that the bounce magnitude due to the electrical parasitics in the package must not exceed the user-defined noise limit $P_{supply}$. When the pin under evaluation is a $V_{DD}$ pin, a constraint equation is written to determine if any transitions that occur on the bus segment will result in a violation of $P_{supply}$. The constraint equation takes into account the voltage noise due to the self-inductance of the $V_{DD}$ pin in addition to any mutual inductive or capacitive coupling that occurs due to switching signals in adjacent pins. By multiplying the coupling magnitude by the transition value $v_i^j$ (which can be 0, 1, or -1), the magnitude and sign of the induced noise value is accounted for. This handles the situation for a static pin a static signal pin ($v_i^j$=0) which has no effect on the noise on the supply pin. The following constraint equation is written for any pin $i$ within a bus segment $j$ that is used as a $V_{DD}$ pin, and is being evaluated for a supply bounce violation:

- $v_i^j = V_{DD} \Rightarrow$
  $P_{supply} \cdot V_{DD} \geq (\frac{di}{dt}) \cdot [(L_{11}) \cdot (N_1) + \sum_{k=-p_L}^{k=p_L} [(M_{1(|k|+1)}) \cdot (v_{i+k}^j)] + \sum_{k=-p_C}^{k=p_C} [(\frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}) \cdot (v_{i+k}^j)]]$

When a pin $i$ in segment $j$ is a $V_{SS}$ pin it is required that the bounce magnitude due to the electrical parasitics in the package must not exceed the user-defined noise limit $P_{gnd}$. When the pin under evaluation is a $V_{SS}$ pin, a constraint equation is written to determine if any transitions that occur on the bus segment will result in a violation of $P_{gnd}$. The following constraint equation is written for any pin within a bus segment $j$ that is used for $V_{SS}$ and is being evaluated for a ground bounce violation:

- $v_i^j = V_{SS} \Rightarrow$
  $P_{gnd} \cdot V_{DD} \geq (\frac{di}{dt}) \cdot [(L_{11}) \cdot (N_{-1}) + \sum_{k=-p_L}^{k=p_L} [(M_{1(|k|+1)}) \cdot (v_{i+k}^j)] + \sum_{k=-p_C}^{k=p_C} [(\frac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}) \cdot (v_{i+k}^j)]]$

### 6.1.2 Signal Coupling Constraints

#### 6.1.2.1 Glitch Magnitude Constraints

When a pin $i$ in segment $j$ is a signal pin, it is required that the coupled voltage onto that pin does not exceed any of the user-defined noise limits for signal coupling. If the signal pin is static ($v_i^j$=0), then the glitch magnitude onto the victim pin must not exceed $P_0$. As in the constraint equations for supply bounce, the magnitude of the coupling contribution of any neighboring pin is multiplied by the transition value $v$ (which can be 0, 1, or -1) of the neighboring pin. This account for the magnitude and sign of the coupling due to the neighboring pin. The following constraint equation is written for any signal pin within a bus segment $j$ that is static ($v_i^j$=0) and being evaluated for a glitch violation:

- $v_i^j = 0 \Rightarrow$

$$-(P_0 \cdot V_{DD}) \leq (\tfrac{di}{dt}) \cdot [\sum_{k=-p_L}^{k=p_L}[(M_{1(|k|+1)}) \cdot (v_{i+k}^j)] + \sum_{k=-p_C}^{k=p_C}[(\tfrac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}) \cdot (v_{i+k}^j)]] \leq (P_0 \cdot V_{DD})$$

#### 6.1.2.2 Risetime Degradation Constraints

When a signal pin $i$ in segment $j$ transitions from a logic 0 to a logic 1 ($v_i^j$=1), it is required that the coupled voltage onto that pin does not hinder its risetime. In this situation the cumulative value of the mutual coupling can be exploited to actually aid the transition on the victim signal pin. By requiring that the cumulative coupled voltage on the victim pin either equals or exceeds the user-defined noise limit $P_1$, it is also possible to improve the risetime of the victim signal (i.e., it is possible to speed up the victim signal). By setting $P_1$ to zero, it is guaranteed that the risetime for the victim signal is not hindered. The following constraint equation is written for any signal pin within a bus segment $j$ that is undergoing a rising transition ($v_i^j$=1) and is being evaluated for a rising edge degradation violation:

- $v_i^j = 1 \Rightarrow$

$$P_1 \cdot V_{DD} \leq (\tfrac{di}{dt}) \cdot [\sum_{k=-p_L}^{k=p_L}[(M_{1(|k|+1)}) \cdot (v_{i+k}^j)] + \sum_{k=-p_C}^{k=p_C}[(\tfrac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}) \cdot (v_{i+k}^j)]]$$

#### 6.1.2.3 Falltime Degradation Constraints

In a similar manner, when a signal pin $i$ in segment $j$ transitions from a logic 1 to a logic 0 ($v_i^j$=-1), it is required that the coupled voltage onto that pin does not hinder its falltime. By requiring that the cumulative coupled voltage on the victim pin either equals or exceeds the user-defined noise limit $P_{-1}$, it is also possible to improve the falltime of the victim signal (i.e., it is possible to speed up the victim signal). By setting $P_{-1}$ to zero, it is guaranteed that the falltime for the victim signal is not hindered. The following constraint equation is written for any signal pin within a bus segment $j$ that is undergoing a falling transition ($v_i^j$=-1) and being evaluated for a falling edge degradation violation:

- $v_i^j = -1 \Rightarrow$

$$P_{-1} \cdot V_{DD} \geq (\tfrac{di}{dt}) \cdot [\textstyle\sum_{k=-p_L}^{k=p_L} [(M_{1(|k|+1)}) \cdot (v_{i+k}^j)] + \textstyle\sum_{k=-p_C}^{k=p_C} [(\tfrac{C_{1(|k|+1)} \cdot Z_0^2}{(0.8)}) \cdot (v_{i+k}^j)]]$$

### 6.1.3 Capacitive Bandwidth Limiting Constraints

When a pin $i$ in segment $j$ is a signal pin, it is required that the capacitive bandwidth limitation due to adjacent switching signal pins does not degrade the *original* risetime by more than $P_{BW}$. As described in Section 1.2.3, the capacitance that a victim signal will experience will depend on the self-capacitance ($C_0$) in addition to mutual coupling capacitance to neighboring signal pins ($C_{1k}$). The original risetime is defined as the risetime achieved when all aggressing signals within the segment are static. In this case, the mutual capacitance to all other signal pins will be $1 \cdot C_{1k}$. When the neighboring aggressor pins switch, the resulting risetime of the victim can be either improved, degraded, or unhindered (depending on the switching pattern of the aggressor pins).

The worst-case transition on the bus (which causes the largest risetime degradation on the victim) will be when all aggressor signal pins switch in the opposite direction as the victim signal. This results in a doubling of the mutual capacitance to all neighboring pins due to the doubling of the voltage excursion between signals. This makes the capacitance to aggressing signal pins $2 \cdot C_{1k}$. The best-case transition on the bus (which causes the largest risetime improvement on the victim) will be when all signal pins switch in the same direction. In this case the coupling capacitance between pins will be zero ($0 \cdot C_{1k}$) because there is no net voltage difference across the pins. Since $P_{BW}$ is described as a percentage of risetime degradation, the constraint equation must account for the *original* risetime in addition to the risetime which results when neighboring signals switch. The following constraint equation is written for any signal pin within a bus segment $j$ that is undergoing a positive transition ($v_i^j$=1) and being evaluated for a capacitive bandwidth limitation violation:

- $v_i^j = 1 \Rightarrow$

$$P_{BW} \leq \frac{[2.2 \cdot (C_0 + \sum_{k=-p_C}^{k=p_C} C_{1(|k|+1)}) \cdot Z_0] - [2.2 \cdot (C_0 + \sum_{k=-p_C}^{k=p_C} C_{1(|k|+1)} \cdot (1 - v_{i+k}^j)) \cdot Z_0]}{[2.2 \cdot (C_0 + \sum_{k=-p_C}^{k=p_C} C_{1(|k|+1)}) \cdot Z_0]}$$

In the above constraint, multiplying the aggressor coupling capacitance by $(1 - v_{i+k}^j)$ handles the contribution of the mutual capacitance to the bandwidth limitation. Since the aggressor signal pin can take on values of $v_{i+k}^j \in \{-1, 0, 1\}$, this evaluates to $(1 - v_{i+k}^j) \in \{2, 1, 0\}$. This quantity is multiplied with $C_{1k}$ and then added to $C_0$ to give the total amount of capacitance that the victim pin charges.

In a similar manner, the bandwidth limitation constraint can be written for a victim signal that undergoes a falling transition ($v_i^j$=-1). In this case, the contribution of the coupling capacitance is handled by multiplying the mutual capacitance by $(|-1 - v_{i+k}^j|)$. Using this expression, the aggressor signal transition values $(v_{i+k}^j \in \{-1, 0, 1\})$ evaluate to $(|-1 - v_{i+k}^j|) \in \{0, 1, 2\}$. The following constraint equation is written for any signal pin $i$ within a bus segment $j$ that undergoes a falling transition ($v_i^j$=-1) and is being evaluated for a capacitive bandwidth limitation violation:

- $v_i^j = -1 \Rightarrow$

$$P_{BW} \geq \frac{[2.2 \cdot (C_0 + \sum_{k=-p_C}^{k=p_C} C_{1(|k|+1)} \cdot (|-1 - v_{i+k}^j|)) \cdot Z_0] - [2.2 \cdot (C_0 + \sum_{k=-p_C}^{k=p_C} C_{1(|k|+1)}) \cdot Z_0]}{[2.2 \cdot (C_0 + \sum_{k=-p_C}^{k=p_C} C_{1(|k|+1)}) \cdot Z_0]}$$

### 6.1.4 Impedance Discontinuity Constraints

When a pin $i$ in segment $j$ is a signal pin it is required that the magnitude of the reflected voltage due to impedance discontinuities within the package must not exceed $P_\Gamma$. As described in Section 1.2.5, the impedance of a given signal path will depend on its self-inductance ($L_{11}$) and self-capacitance ($C_0$) in addition to any mutual inductance ($M_{1k}$) and mutual capacitance ($C_{1k}$). This means that the data sequences present on neighboring signal pins will alter the effective inductance and capacitance of the victim signal. This directly effects the characteristic impedance of the victim pin, which is used in the evaluation of reflected noise. This must be accounted for in the constraint equation. By multiplying this additional mutual inductance and capacitance values by the transition values of the aggressor neighbors ($v_{i+k}^{j}$), the polarity and cumulative effect of the coupling can be handled. The following constraint equation is written for any signal pin $i$ within a bus segment $j$ that undergoes a positive transition ($v_i^j$=1) and is being evaluated for an impedance discontinuity violation:

- $v_i^j = 1 \Rightarrow$

$$P_\Gamma \leq |\frac{\sqrt{\frac{L_{11}+\sum_{k=-p_L}^{k=p_L}(M_{1(|k|+1)}\cdot(v_{i+k}^{j})}{C_0+\sum_{k=-p_C}^{k=p_C}(C_{1(|k|+1)}\cdot(v_{i+k}^{j})}}-Z_0}{\sqrt{\frac{L_{11}+\sum_{k=-p_L}^{k=p_L}(M_{1(|k|+1)}\cdot(v_{i+k}^{j})}{C_0+\sum_{k=-p_C}^{k=p_C}(C_{1(|k|+1)}\cdot(v_{i+k}^{j})}}+Z_0}|$$

In a similar manner, when a pin $i$ in segment $j$ is a signal pin and undergoes a falling transition ($v_i^j$=-1), it is required that the magnitude of the reflected voltage due to impedance discontinuities within the package must not exceed $P_\Gamma$. In this case the contribution of the mutually coupled inductance and capacitance must be inverted to properly represent its cumulative nature. This is accomplished by multiplying the mutual inductance and capacitance values by the negative of the transition value (-$v_{i+k}^{j}$), thereby accounting for the polarity of the mutually coupled voltage. Note that the polarity of $\Gamma$ is reversed when the forcing function is a falling edge [70]. The following constraint equation is written for any signal pin $i$ within a bus segment $j$ that undergoes a falling transition ($v_i^j$=-1) and is being evaluated for an impedance discontinuity violation:

- $v_i^j = -1 \Rightarrow$

$$P_\Gamma \leq |\frac{\sqrt{\frac{L_{11}+\sum_{k=-p_L}^{k=p_L}(M_{1(|k|+1)}\cdot(-v_{i+k}^{j})}{C_0+\sum_{k=-p_C}^{k=p_C}(C_{1(|k|+1)}\cdot(-v_{i+k}^{j})}}-Z_0}{\sqrt{\frac{L_{11}+\sum_{k=-p_L}^{k=p_L}(M_{1(|k|+1)}\cdot(-v_{i+k}^{j})}{C_0+\sum_{k=-p_C}^{k=p_C}(C_{1(|k|+1)}\cdot(-v_{i+k}^{j})}}+Z_0}|$$

### 6.1.5 Number of Constraint Equations

For each $V_{DD}$ pin within a segment, one equation is written that represents the supply bounce constraint ($P_{supply}$). For each $V_{SS}$ pin within a segment, one equation is written that represents the ground bounce constraint ($P_{gnd}$). For each signal pin, constraint equations are written for the three transition values that the pin can have ($v_i^j \in \{-1, 0, 1\}$). When a signal pin is static ($v_i^j = 0$), one equation must be written to constrain the glitching noise due to switching neighbor pins ($P_0$). When a signal pin transitions high ($v_i^j = 1$), three equations must be written that account for rising edge degradation ($P_1$), capacitive bandwidth limitation ($P_{BW}$), and impedance discontinuities ($P_\Gamma$). When a signal pin transitions low ($v_i^j = -1$), three more equations must be written that account for falling edge degradation ($P_{-1}$), capacitive bandwidth limitation ($P_{BW}$), and impedance discontinuities ($P_\Gamma$). This gives the total number of constraint equations that are written as:

$$N_{constraints} = N_p + N_g + (7 \cdot W_{bus}) \tag{6.1}$$

The number of constraints can be reduced for particular applications. For a particular package, some noise sources may be negligible and can be ignored in the constraint evaluation. For the three packages studied in this work, it was found that in all cases the supply bounce and inductive signal coupling were the dominant noise sources. This is a consequence of noise being dominated by the inductance within the package interconnect. If only the inductive noise components are considered, then each signal pin needs only three constraint equations: $P_0$, $P_1$, and $P_{-1}$. This reduces the number of constraint equations to:

$$N_{ind-constraints} = N_p + N_g + (3 \cdot W_{bus}) \tag{6.2}$$

### 6.1.6 Number of Constraint Evaluations

Each of the constraint equations must be evaluated against each of the possible transitions that are to be transmitted through the package. Since each signal pin can take on three unique transition values ($v_i^j \in \{-1, 0, 1\}$), then the total number of possible transitions for a given bus segment is bounded by:

$$N_{transitions} = (W_{bus} + 2 \cdot max(p_L, p_C))^3 \tag{6.3}$$

In practice, however, some of the signals in the neighborhood of the signal of interest may be supply pins, resulting in fewer constraints. This gives the total number of constraint equation evaluations as:

$$N_{evaluations} = (N_{constraints}) \cdot (N_{transitions}) \tag{6.4}$$

Again, the number of evaluations can be reduced by limiting the noise sources that are considered. It should be noted that due to the symmetry of the bus segment representation (which mimics the regular manner in which the busses are implemented in practice), the evaluations need only be performed on the signals within a single bus segment and not on the entire bus. This symmetry allows a dramatic reduction in computation time compared to the case where the entire bus is analyzed monolithically.

## 6.2 Encoder Construction

Once the transitions have been tested against all of the constraining equations, a subset of legal transitions remain from the original set of possible transitions. Assume that the original set of constraints were written for a bus segment with $n$ signals. Using the legal transitions remaining, we next find the maximum effective bus size $m$, such that the transitions on this bus can be encoded using the remaining legal transitions on the physical $n$ bit bus.

### 6.2.1 Encoder Algorithm

From the set of legal vector sequences, we next create a ROBDD [11, 12] $G$, to encode legal bus transitions. We then find the effective size $m$ of the bus that can be encoded using the transitions in $G$, using a ROBDD based algorithm. Note that the ROBDD $G$ has $2n$ variables. The first $n$ variables refer to the **from** vertices and the next $n$ variables refer to the **to** vertices of the vector transition. There is a legal edge between vertices $v_1$ and $v_2$ iff $G(v_1, v_2) = 1$.

Note that for a vector sequence $v^j$, we can construct minterms in $G$ to encode transitions between vectors $w^j_{from}$ and $w^j_{to}$. The end-points of this edge ($w^j_{from}$ and $w^j_{to}$) can be constructed given $v^j$, as follows:

- $w^j_{from,i} = 0$ if $v^j_i = 1$ (i.e. the signal is rising) or if $v^j_i = 0$ (i.e. the signal is static).
- $w^j_{from,i} = 1$ if $v^j_i = -1$ (i.e. the signal is falling) or if $v^j_i = 0$ (i.e. the signal is static).

Similarly, we can write

- $w^j_{to,i} = 1$ if $v^j_i = 1$ or if $v^j_i = 0$.
- $w^j_{to,i} = 0$ if $v^j_i = -1$ or if $v^j_i = 0$.

$G(w^j_{from}, w^j_{to}) = 1$ indicates the legality (from an inductive cross-talk viewpoint) of the transition from vector $w^j_{from}$ to $w^j_{to}$. Therefore, given a set of vector sequences $\{v^j\}$ which are **legal**, we can construct a ROBDD $G$ whose minterms $(w_{from} : w_{to})$ are vectors in $B^{2n}$, such that they indicate a legal transition (from an inductive cross-talk viewpoint) between the source ($w_{from}$) and sink ($w_{to}$) vertices. Note that the ":" symbol above refers to the concatenation operator.

If an $m$-bit bus can be encoded using the legal transitions in $G$, then there must exist a set of vertices $V_c \subseteq B^n$ such that:

- Each $v_s \in V_c$ has at least $2^m$ outgoing edges $e(v_s, v_d)$ (including the self edge), such that the destination vertex $v_d \in V_c$.
- The cardinality of $V_c$ is at least $2^m$.

The resulting encoder is memory based. Note that the physical size of the bus $n$ is obviously greater than or equal to $m$.

Given $G$, we find $m$ using Algorithm 1. The input to the algorithm is $m$ and $G$. We first find the out-degrees (self-edges are counted) of each $v_s \in B^n$. This is done by logically ANDing the ROBDD of the vertex $v_s$ with $G$. We find the cardinality of the resulting ROBDD – it represents the out-degree of $v_s$. If the number of out-edges of any $v_s$ is greater than $2^m$, we add $v_s$ (and its out-degree) into a hash table $V$.

For each $v_s \in V$, we next check if each of its destination nodes $v_d$ are in $V$. If $v_d \notin V$, we decrement the out-degree of $v_s$ by 1. If the out-degree of $v_s$ becomes less than $2^m$, we remove $v_s$ from $V$. These operations are performed until convergence. If at this point, the number of surviving vertices in V is $2^m$ or more, then an $m$-bit memoryless CODEC can be constructed from $G$.

We initially call the algorithm with $m = n - 1$ (where $n$ is the physical bus size). If an $m$ bit bus cannot be encoded using $G$, then we decrement $m$. We repeat this until we find a value of $m$ such that the $m$-bit bus can be encoded by $G$.

**Algorithm 1** Testing if $G$ can encode an $m$-bit bus

```
$test\_encoder(m,\ G)$
$find\ out - degree(v_s)\ of\ each\ node\ v_s$, insert $(v_s, out - degree(v_s))$ in $V$ if $out - degree(v_s) \geq 2^m$
$degrees\_changed = 1$
while $degrees\_changed$ do
  $degrees\_changed = 0$
  for $each\ v_s \in V$ do
    for $each\ v_d$ S.T. $G(v_s, v_d) = 1$ do
      if $v_d \notin V$ then
        $decrement\ out - degree(v_s)$ in $V$
        $degrees\_changed = 1$
      end if
      if $out - degree(v_s) < 2^m$ then
        $V \leftarrow V \setminus v_s$
        $break$
      end if
    end for
  end for
end while
if $|V| \geq 2^m$ then
  print($m$ bit bus may be encoded using $G$)
else
  print($m$ bit bus cannot be encoded using $G$)
end if
```

For a memory-less encoder, the subgraph induced by the edges in $V_c$ should be a clique. Memory-based encoders are explored in this work, due to the fact that they allow effective widths which are higher than memory-less encoders.

Note that this entire analysis needs to be performed for a representative bus segment. In other words, even if the bus is very wide, the analysis is performed for a single segment, which is typically very small. This segment could be part of a much larger bus, and the analysis would be valid for all segments of the bus.

### 6.2.2 Encoder Overhead

The end result of the encoder construction is a mapping between transitions on an $m$-bit on-chip bus to transitions on an $n$-bit off-chip bus. The transitions on the $n$-bit off-chip bus are selected so that the worst-case noise resulting from any of these transitions is less than or equal to the user-specified noise limit. Since the transitions on the $n$-bit bus avoid any noise greater than the user-specified limit, they can be transmitted at a higher datarate The net improvement must also account for the overhead in the number of bits utilized for the bus (which is $n - m$). The computation of the overhead for the bus expansion encoding technique is shown in Equation 6.5.

$$Overhead_{bus-expansion} = (\frac{n-m}{n}) \cdot 100 \tag{6.5}$$

## 6.3 Decoder Construction

The decoder construction is identical to the encoder construction however the process is performed in reverse. The combinational logic for the decoder is simply a reverse permutation of the logic for the encoder. In the case of the memory-less decoder, the logic is induced by that of the encoder. This is also the case for the memory-based decoders.

## 6.4 Experimental Results

This section presents the experimental results that were performed to verify the effectiveness of the bus expansion encoder.

### 6.4.1 3-Bit Fixed $\frac{di}{dt}$ Example

The first example is a 3-bit bus that is evaluated for a fixed $\frac{di}{dt}$. This example illustrates how the encoding technique directly reduces noise within the package. A $\frac{di}{dt}$ of 33 $\frac{MA}{s}$ was chosen that corresponds to a datarate of 550 $\frac{Mb}{s}$ in a 50 $\Omega$ system using Equation 4.6. The electrical parameters for the BGA-WB package are used from Table 2.1. The bus configuration has three signal pins ($W_{bus}$=3), one $V_{DD}$ pin ($N_p$=1), and one $V_{SS}$ pin ($N_g$=1) for a total of 5 pins within the bus segment ($N_{segment}$=5). This configuration yields a *Signal-to-Power-Ground* metric of *SPG*=3:1:1 and a *Signal-to-Power-Ground Ratio* of *SPGR*=3. The mutual inductive and capacitive coupling is considered to the 2 most adjacent pins ($p_L = p_C = 2$). This simplifies the analysis by

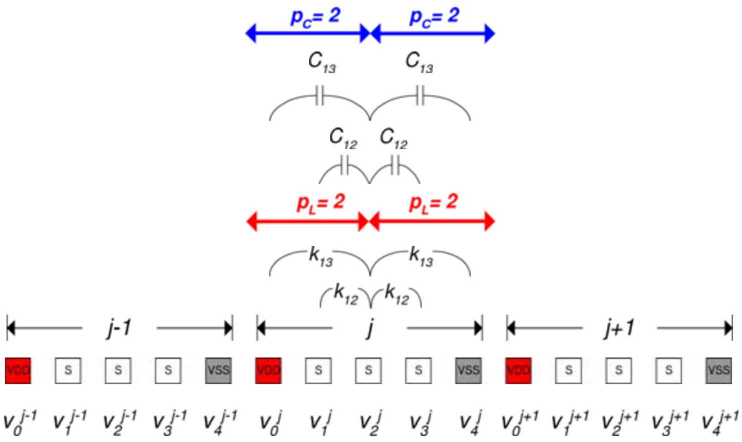

p_C= 2
p_C= 2
C_13
C_13
C_12
C_12
p_L= 2
p_L= 2
k_13
k_13
k_12
k_12
j-1
j
j+1
VDD
S
S
S
VSS
VDD
S
S
S
VSS
VDD
S
S
S
VSS
v_0^j-1 v_1^j-1 v_2^j-1 v_3^j-1 v_4^j-1
v_0^j v_1^j v_2^j v_3^j v_4^j
v_0^j+1 v_1^j+1 v_2^j+1 v_3^j+1 v_4^j+1

Figure 6.1: 3-Bit Bus Example

ignoring inductive coupling less than 15% and capacitive coupling less than $50fF$. The bus configuration is shown in Figure 6.1.

This bus was encoded using two sets of constraints: **aggressive** ($P_{supply}$, $P_{gnd}$, $P_0$, $P_1$, $P_{-1}$, $P_{BW}$, and $P_\Gamma$ set to 5%) and **non-aggressive** ($P_{supply}$, $P_{gnd}$, $P_0$, $P_1$, $P_{-1}$, $P_{BW}$, and $P_\Gamma$ set to 10%). The first step in the bus expansion technique is to write the constraint equations. Equation 6.1 indicates that 24 constraint equations need to be written for this package; however, from the results in Chapter 4, it is known that the BGA-WB performance is dominated by the inductive nature of its interconnect. This allows the number of constraint equations to be reduced to 11 using Equation 6.2. The following lists the 11 constraint equations for this example. These constraints have been simplified by removing terms with $v_i^j = 0$.

1) $v_0^j = V_{DD} \Rightarrow P_{supply} \cdot V_{DD} \leq \frac{di}{dt} \cdot [(L_{11} \cdot N_1) + (M_{13} \cdot v_3^{j-1}) + (M_{12} \cdot v_1^j) + (M_{13} \cdot v_2^j)]$

2) $v_1^j = 1 \Rightarrow P_1 \cdot V_{DD} \leq \frac{di}{dt} \cdot [(M_{12} \cdot v_2^j) + (M_{13} \cdot v_3^j)]$

3) $v_1^j = -1 \Rightarrow P_{-1} \cdot V_{DD} \geq \frac{di}{dt} \cdot [(M_{12} \cdot v_2^j) + (M_{13} \cdot v_3^j)]$

4) $v_1^j = 0 \Rightarrow -(P_0 \cdot V_{DD}) \leq \frac{di}{dt} \cdot [(M_{12} \cdot v_2^j) + (M_{13} \cdot v_3^j)] \leq (P_0 \cdot V_{DD})$

5) $v_2^j = 1 \Rightarrow P_1 \cdot V_{DD} \leq \frac{di}{dt} \cdot [(M_{12} \cdot v_1^j) + (M_{12} \cdot v_3^j)]$

6) $v_2^j = -1 \Rightarrow P_{-1} \cdot V_{DD} \geq \frac{di}{dt} \cdot [(M_{12} \cdot v_1^j) + (M_{12} \cdot v_3^j)]$

7) $v_2^j = 0 \Rightarrow -(P_0 \cdot V_{DD}) \leq \frac{di}{dt} \cdot [(M_{12} \cdot v_1^j) + (M_{12} \cdot v_3^j)] \leq (P_0 \cdot V_{DD})$

8) $v_3^j = 1 \Rightarrow P_1 \cdot V_{DD} \leq \frac{di}{dt} \cdot [(M_{13} \cdot v_1^j) + (M_{12} \cdot v_2^j)]$

9) $v_3^j = -1 \Rightarrow P_{-1} \cdot V_{DD} \geq \frac{di}{dt} \cdot [(M_{13} \cdot v_1^j) + (M_{12} \cdot v_2^j)]$

10) $v_3^j = 0 \Rightarrow -(P_0 \cdot V_{DD}) \leq \frac{di}{dt} \cdot [(M_{13} \cdot v_1^j) + (M_{12} \cdot v_2^j)] \leq (P_0 \cdot V_{DD})$

11) $v_4^j = V_{SS} \Rightarrow P_{gnd} \cdot V_{DD} \leq \frac{di}{dt} \cdot [(L_{11} \cdot N_{-1}) + (M_{13} \cdot v_2^j) + (M_{12} \cdot v_3^j) + (M_{13} \cdot v_1^{j+1})]$

All possible transitions on the bus are then evaluated using the 11 constraint equations. From this evaluation, transitions that violate any of the constraints are flagged as *illegal* and removed from subset of *legal* transitions used in the encoder construction. Table 6.1 shows the results of the constraint evaluations. In this table, if one of the original transitions violates a constraint equation, the number(s) of the constraint equation is listed.

After all transitions have been evaluated against the constraints, the remaining legal transitions are used to create a directed graph. Figure 6.2 shows the directed graph that was created from the non-aggressive constraint evaluation. This figure does not show the self-edges for simplicity.

From the directed graphs for both the non-aggressive and aggressive constraints, the largest effective bus width $m$ was constructed, such that the noise for this bus is within the user-defined limits. From the mapping between the transitions in the $m$-bit bus to those in the legal transitions in the $n$-bit bus, the encoder state machine is constructed. Figure 6.3 shows the overhead for the bus expansion encoder (Equation 6.5) for both the aggressive and non-aggressive constraints. This figure suggests that the overhead of the encoder approaches an asymptotic limit as the number of channels are added to the bus segment.

| Original Transition | Aggressive | Non-Aggressive |
|---|---|---|
| 000 | 000 | 000 |
| 001 | 001 | 001 |
| 00-1 | 00-1 | 00-1 |
| 010 | 010 | 010 |
| 011 | Violates 1,4 | 011 |
| 01-1 | 01-1 | 01-1 |
| 0-10 | 0-10 | 0-10 |
| 0-11 | 0-11 | 0-11 |
| 0-1-1 | Violates 4,11 | 0-1-1 |
| 100 | 100 | 100 |
| 101 | Violates 1,7 | 101 |
| 10-1 | 10-1 | 10-1 |
| 110 | Violates 1,10 | 110 |
| 111 | Violates 1,2,5,8 | Violates 11 |
| 11-1 | Violates 1 | 11-1 |
| 1-10 | 1-10 | 1-10 |
| 1-11 | Violates 1 | 1-11 |
| 1-1-1 | Violates 11 | 1-1-1 |
| -100 | -100 | -100 |
| -101 | -101 | -101 |
| -10-1 | Violates 7,11 | -10-1 |
| -110 | -110 | -110 |
| -111 | Violates 1 | -111 |
| -11-1 | Violates 11 | -11-1 |
| -1-10 | Violates 10,11 | -1-10 |
| -1-11 | Violates 11 | -1-11 |
| -1-1-1 | Violates 3,6,9,11 | Violates 1 |

Table 6.1: Constraint Evaluations for 3-Bit, Fixed $\frac{di}{dt}$ Bus Expansion Example

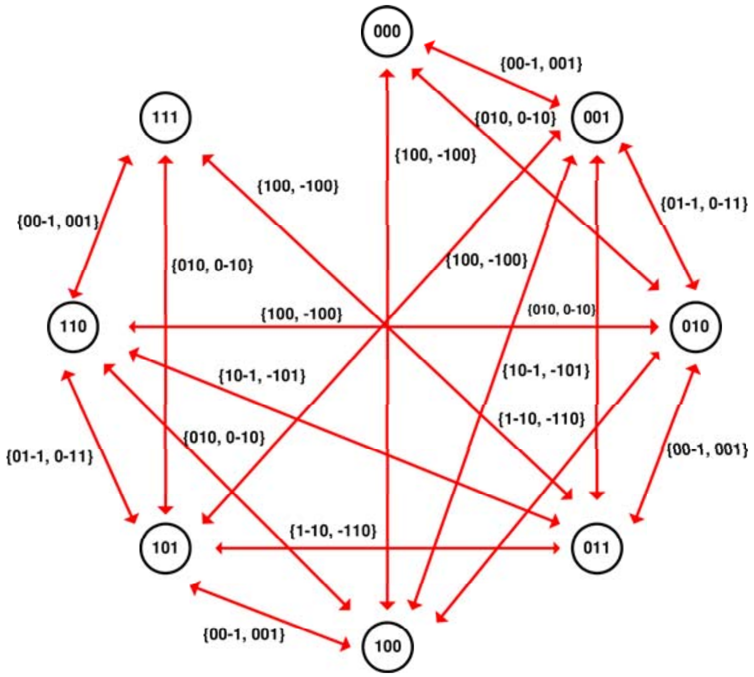


Figure 6.2: Directed Graph for the 3-Bit, Fixed $\frac{di}{dt}$ Bus Expansion Example

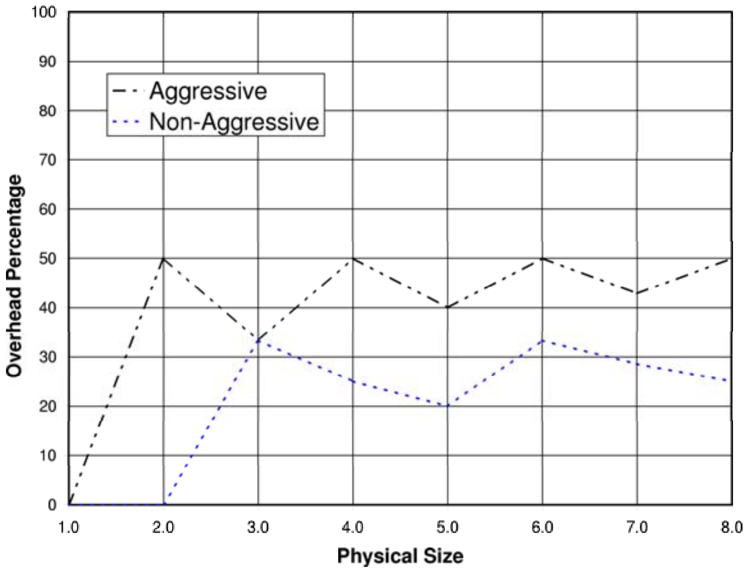


Figure 6.3: Bus Expansion Encoder Overhead for the Fixed $\frac{di}{dt}$ Example

SPICE simulations were performed to quantify the noise reductions achieved by the encoder. Figure 6.4 shows the ground bounce reduction achieved by the encoder for both the aggressive and non-aggressive constraints. This figure shows that the ground bounce was reduced as much as 50% for the non-aggressive encoder and 89% for the aggressive encoder (over the original non-encoded bus). Also, the worst-case noise in the encoded bus is just within the user-specified (aggressive and non-aggressive) noise limits.

Figure 6.5 shows the glitch magnitude reduction achieved by the encoder for both the aggressive and non-aggressive constraints. In this figure the glitch magnitude was reduced 44% for the non-aggressive encoder and 68% for the aggressive encoder over the original non-encoded bus.

Figure 6.6 shows the edge degradation noise reduction achieved by the encoder for both the aggressive and non-aggressive constraints. This figure illustrates the reduction in edge degradation that the encoders achieve relative to the non-encoded configuration.

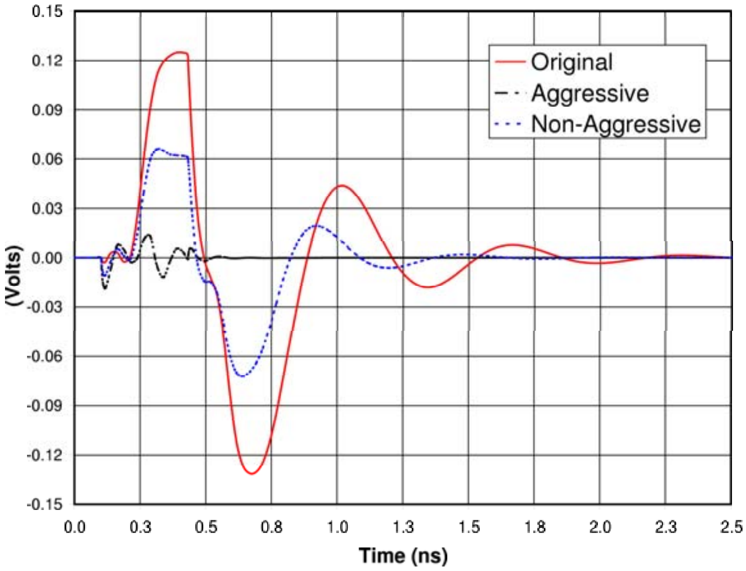


Figure 6.4: SPICE Simulation of Ground Bounce for 3-Bit, Fixed $\frac{di}{dt}$ Example

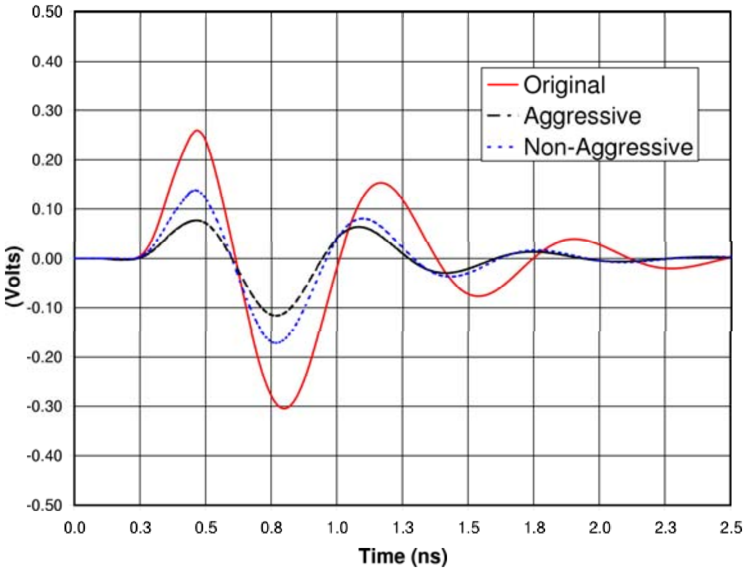


Figure 6.5: SPICE Simulation of Glitching Noise for 3-Bit, Fixed $\frac{di}{dt}$ Example

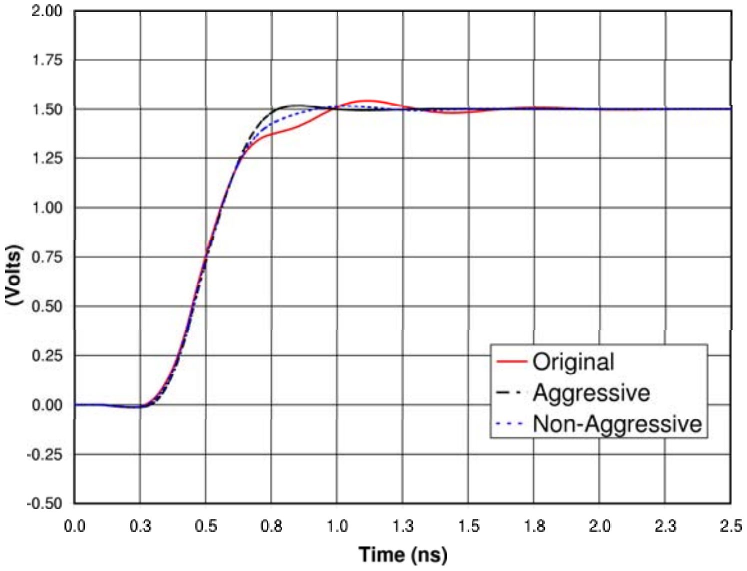


Figure 6.6: SPICE Simulation of Edge Coupling Reduction for 3-Bit, Fixed $\frac{di}{dt}$ Example

### 6.4.2 3-Bit Varying $\frac{di}{dt}$ Example

Using the constraint equations in Section 6.4.1, the maximum $\frac{di}{dt}$ that the package can achieve can be found for both the encoded and non-encoded cases. For this example the original set of transitions is evaluated using the constraint equations to find the maximum $\frac{di}{dt}$ and, in turn, the maximum per-pin datarate for the non-encoded bus.

For the non-encoded evaluation, no vectors are removed from the original set; instead, $\frac{di}{dt}$ is increased to its maximum value without violating any of the constraint equations. At that point, this $\frac{di}{dt}$ indicates the fastest datarate that the non-encoded bus can achieve.

For the encoded evaluation, the reduced transition set from Section 6.4.1 is evaluated in the same manner. In the encoded case, the worst-case transitions have already been removed from the legal transition set and will result in a faster $\frac{di}{dt}$ that can be achieved without violating any of the constraints. Table 6.2 shows the results for this example. In this table the maximum $\frac{di}{dt}$ for the original and encoded cases are reported. This table illustrates that even after considering the overhead of the encoder, the bus throughput is increased as much as 46% **using the same three physical signal pins** on the package.

| | Original | Encoded |
|---|---|---|
| Max $di/dt$ | 13.3 MA/s | 37 MA/s |
| Max Datarate | 222 Mb/s | 617 Mb/s |
| # of Legal Transitions | 27 | 13 |
| Physical Size of Bus | 3 | 3 |
| Effective Size of Bus | 3 | 2 |
| Encoder Overhead | - | 33% |
| Bus Throughput | 666 Mb/s | 1234 Mb/s |
| Throughput Improvement | - | 46% |

Table 6.2: Experimental Results for the 3-Bit, Varying $\frac{di}{dt}$ Example

### 6.4.3 Functional Implementation

The bus expansion encoders were implemented to verify their functionality and feasibility. The Verilog hardware description language (HDL) was used to create the encoder circuitry [68, 69]. The Verilog implementation consisted of a series of *CASE* statement that mapped the on-chip $m$-bit transitions into the $n$-bit transitions that could be transmitted through the package. Algorithm 2 shows the Verilog HDL code for the bus expansion encoder.

**Algorithm 2** Verilog Implementation of $m$-Space to $n$-Space Encoder Mapping

```
module expansion_encoder (expansion_data_out, data_in);
  output reg [n-1:0] expansion_data_out;
  in             [m-1:0] data_in;
   always @ (data_in)
      begin
         case (data_in)
             m'bxx : expansion_data_out = n'bxxx;
               :
               :
             m'bxx : expansion_data_out = n'bxxx;
         endcase
      end
endmodule
```

Figures 6.7 through 6.10 show the Verilog simulation results for aggressively constrained bus expansion encoders for effective sizes of 2, 4, 6, and 8. For each waveform the $m$-bit clock and data are plotted in addition to the encoded $n$-bit data which is transmitted off-chip.

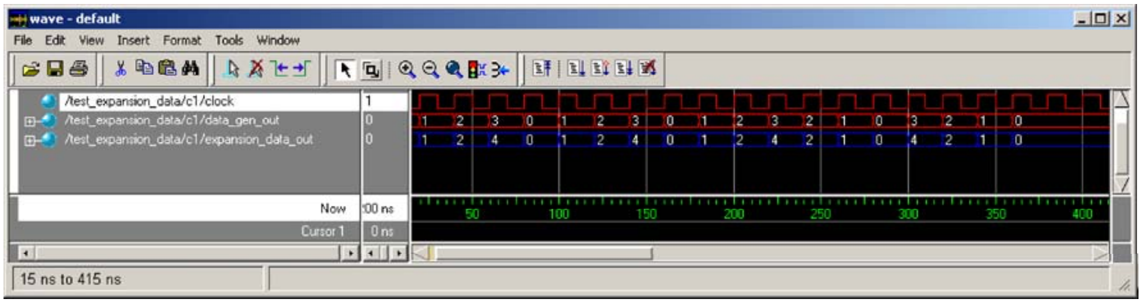

wave - default
File Edit View Insert Format Tools Window
/test_expansion_data/c1/clock 1
/test_expansion_data/c1/data_gen_out 0
/test_expansion_data/c1/expansion_data_out 0
1 2 3 0 1 2 3 0 1 2 3 2 1 0 3 2 1 0
1 2 4 0 1 2 4 0 1 2 4 2 1 0 4 2 1 0
Now 00 ns
Cursor 1 0 ns
50 100 150 200 250 300 350 400
15 ns to 415 ns

Figure 6.7: Verilog Simulation Results for a 2-Bit Bus Expansion Encoder

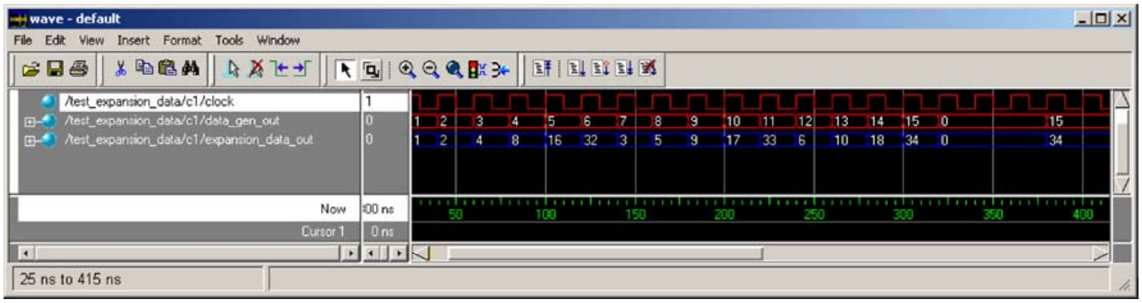
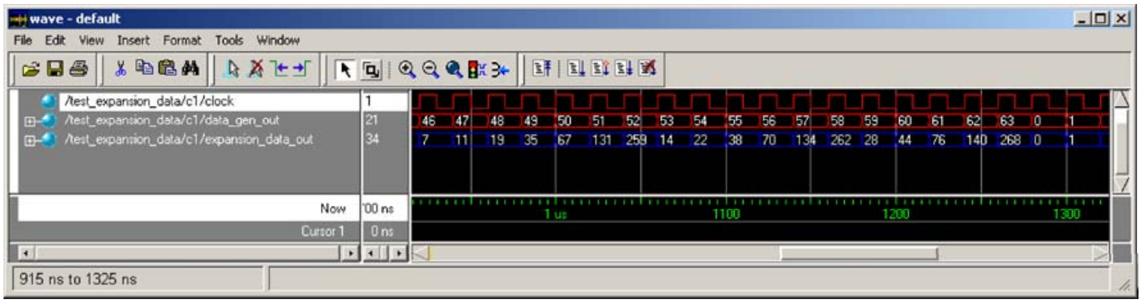
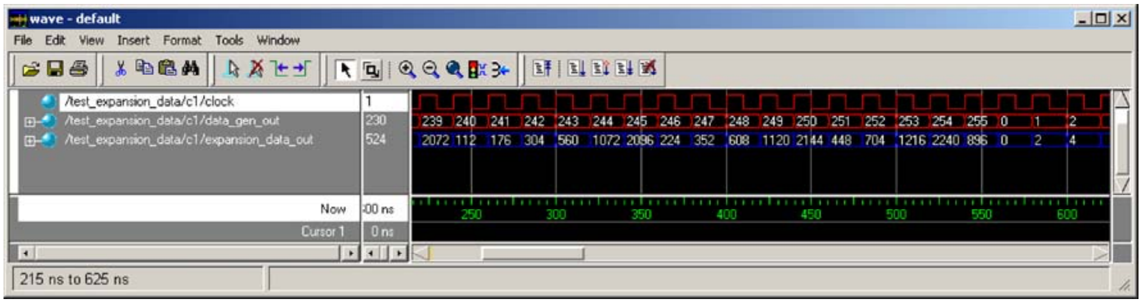


Figure 6.8: Verilog Simulation Results for a 4-Bit Bus Expansion Encoder



```
/test_expansion_data/c1/clock                 1
/test_expansion_data/c1/data_gen_out          21    46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  0  1
/test_expansion_data/c1/expansion_data_out    34    7   11  19  35  67  131 259 14  22  38  70  134 262 28  44  76  140 268 0  1
Now       '00 ns    1 us   1100   1200   1300
Cursor 1  0 ns
915 ns to 1325 ns
```

Figure 6.9: Verilog Simulation Results for a 6-Bit Bus Expansion Encoder



```
/test_expansion_data/c1/clock                 1
/test_expansion_data/c1/data_gen_out          230   239  240 241 242 243 244  245  246 247 248 249  250  251 252 253  254  255 0 1 2
/test_expansion_data/c1/expansion_data_out    524   2072 112 176 304 560 1072 2096 224 352 608 1120 2144 448 704 1216 2240 896 0 2 4
Now       :00 ns    250   300   350   400   450   500   550   600
Cursor 1  0 ns
215 ns to 625 ns
```

Figure 6.10: Verilog Simulation Results for a 8-Bit Bus Expansion Encoder

### 6.4.4 Physical Implementation

In order to evaluate the feasibility of the expansion encoder/decoder, the physical design of the circuitry was performed.

#### 6.4.4.1 TSMC 0.13um ASIC Process

The encoders were synthesized using the TSMC 0.13$\mu m$ CMOS IC process to understand the impact on delay and area if the encoders were integrated on-chip. Encoders for effective bus sizes of 2, 4, 6, and 8 were implemented. For each of these sizes, both the aggressive (5% of $V_{DD}$) and the non-aggressive (10% of $V_{DD}$) encoders were synthesized. Table 6.3 lists the delay and area impact of the bus expansion encoders when implemented in a TSMC 0.13$\mu m$ process. This table illustrates that incorporating the bus expansion encoder in a modern VLSI design results in a negligible area and delay penalty. The delay through the encoders is left unoptimized to illustrate the total combinational delay of the circuit; however, this delay can easily be hidden by pipelining the encoder in order to partition the combinational delay. The decoder implementation results are identical to the encoder results.

| | Bus Size ($m$) | Noise Limit | |
|---|---|---|---|
| | - | 5% (aggressive) | 10% (non-aggressive) |
| Delay ($ns$) | 2 | 0.170 | encoder not required |
| | 4 | 0.670 | 0.503 |
| | 6 | 1.150 | 0.955 |
| | 8 | 1.310 | 0.983 |
| Area ($um^2$) | 2 | 22 | encoder not required |
| | 4 | 152 | 114 |
| | 6 | 614 | 509 |
| | 8 | 1,181 | 886 |

Table 6.3: Bus Expansion Encoder Synthesis Results in a TSMC 0.13$um$ Process

#### 6.4.4.2 Xilinx 0.35um FPGA Process

The encoders were also synthesized, mapped, and implemented for a *Xilinx VirtexIIPro*, Field Programmable Gate Array (FPGA) which used a $0.35\mu m$ CMOS process. Figure 6.11 shows the Xilinx FPGA target that was used for implementation in addition to the test setup for the encoders.

Encoders for effective bus sizes of 2, 4, 6, and 8 were implemented using both the aggressive and non-aggressive constraints. Table 6.4 lists the delay and area impact of the bus expansion encoders when implemented in the FPGA process. In all cases the bus expansion encoder designs took less than 1% of the FPGA resources. The encoders were implemented using standard combinational logic blocks (Function Generators = FG's) within the FPGA, which resulted in minimal propagation delay through the circuit. Again, the unoptimized delay is presented in this table.

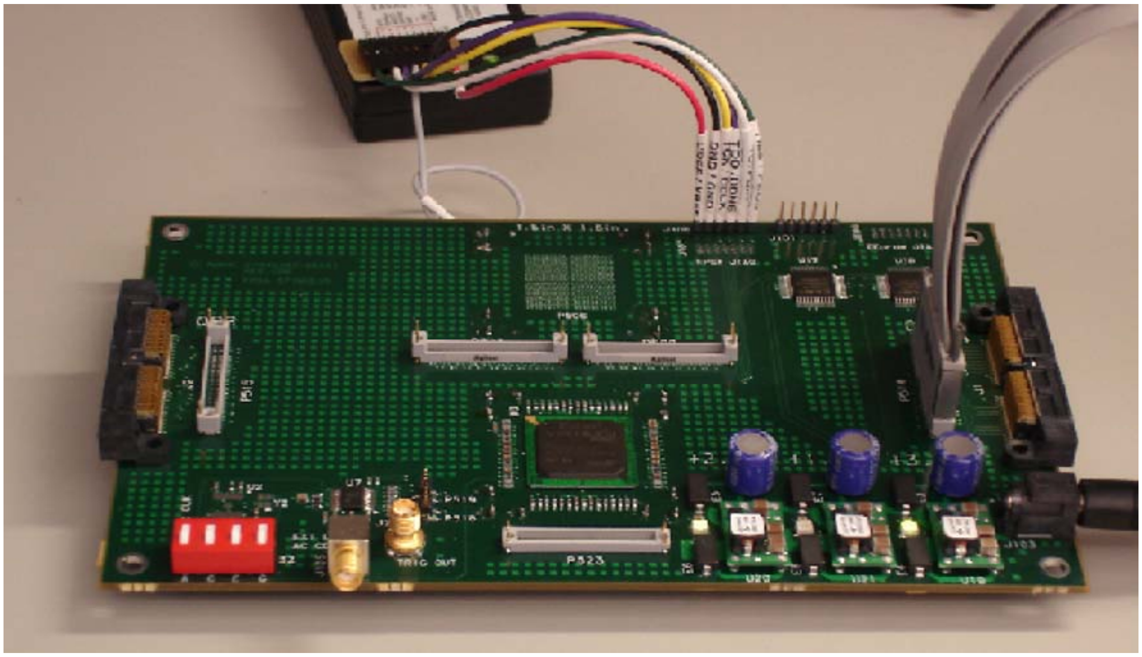


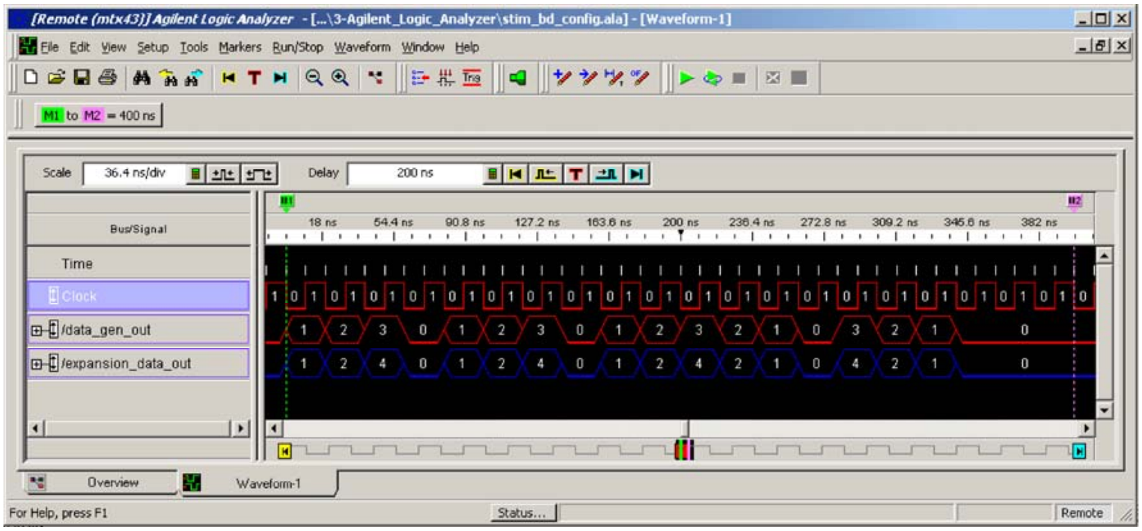
Figure 6.11: Xilinx FPGA Target and Test Setup for Encoder Implementation

| | Bus Size ($m$) | Noise Limit |
|---|---|---|
| | - | 5% (aggressive) & 10% (non-aggressive) |
| Delay ($ns$) | 2 | 0.351 |
| | 4 | 1.020 |
| | 6 | 1.450 |
| | 8 | 1.610 |
| FPGA Usage | 2 | $<1\%$ |
| | 4 | $<1\%$ |
| | 6 | $<1\%$ |
| | 8 | $<1\%$ |
| FPGA Implementation | 2 | 3x, 2-Input FG's |
| | 4 | 6x, 4-Input FG's |
| | 6 | 9x, 6-Input FG's |
| | 8 | 12x, 8-Input FG's |

Table 6.4: Bus Expansion Encoder Synthesis Results in a 0.35$um$, FPGA Process

### 6.4.5 Measurement Results

The outputs of the FPGA were monitored using the 16950A Logic Analyzer from *Agilent Technologies, Inc.* The logic analysis measurements verified that the encoders could be taken from the concept to final implementation using standard IC design practices. Figures 6.12 through 6.15 show the logic analyzer measurement results for the aggressively constrained bus expansion encoders for effective sizes of 2, 4, 6, and 8. For each waveform, the $m$-bit clock and data are plotted in addition to the encoded $n$-bit data which is transmitted off-chip. The $m$-bit vectors are monitored using a debug port that routes the internal nodes of the on-chip bus to the logic analyzer. In practice, this debug port would be removed so not to cause unintentional SSN. These figures show that the actual implementation results of the encoder match the functional simulations. For the bus configuration of this FPGA (SPG=3:1:3), the noise was reduced from 16% to 4% using the bus expansion encoding technique.



[Remote (mtx43)] Agilent Logic Analyzer - [...\3-Agilent_Logic_Analyzer\stim_bd_config.ala] - [Waveform-1]

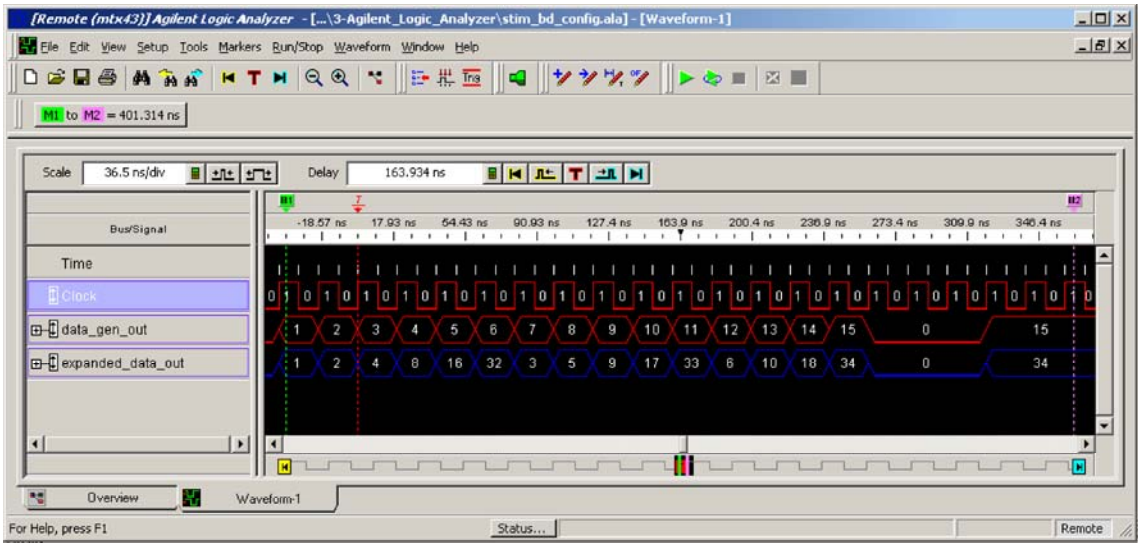
M1 to M2 = 400 ns

Scale 36.4 ns/div

Delay 200 ns

For Help, press F1

Figure 6.12: Logic Analyzer Measurements of a 2-Bit Bus Expansion Encoder



[Remote (mtx43)] Agilent Logic Analyzer - [...\3-Agilent_Logic_Analyzer\stim_bd_config.ala] - [Waveform-1]

M1 to M2 = 401.314 ns

Scale 36.5 ns/div

Delay 163.934 ns

For Help, press F1

Figure 6.13: Logic Analyzer Measurements of a 4-Bit Bus Expansion Encoder

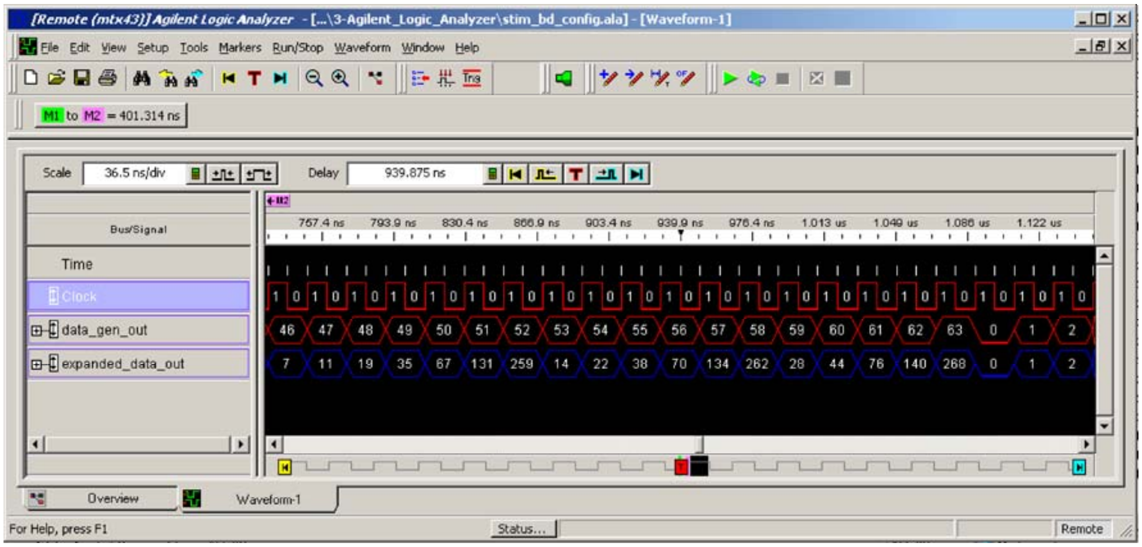


Figure 6.14: Logic Analyzer Measurements of a 6-Bit Bus Expansion Encoder

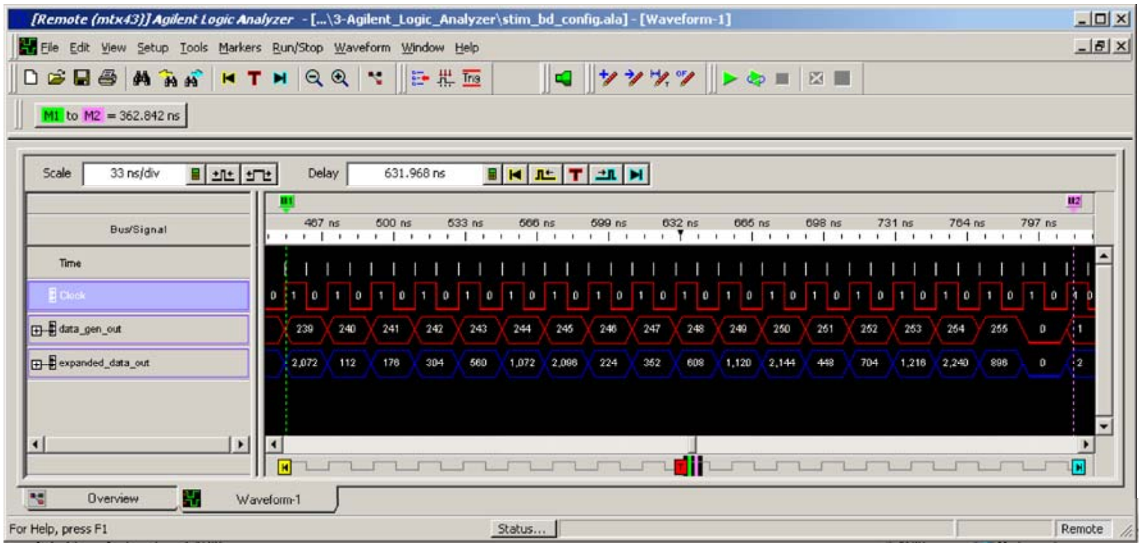


Figure 6.15: Logic Analyzer Measurements of a 8-Bit Bus Expansion Encoder

# Chapter 7

## Bus Stuttering Encoder

It was shown in Chapter 6 that the performance of an off-chip bus could be increased by avoiding a subset of patterns which resulted in noise greater than a specified limit. By eliminating the patterns that create the greatest amount of noise, the remaining patterns can be transmitted at a faster per-pin datarate without exceeding the user-defined noise limits of the system. The bus expansion encoding technique was successful in increasing the overall throughput of the bus even after considering the encoder overhead. The bus expansion encoder is ideal for applications where additional signal pins on the package are available for use when encoding the on-chip vectors; however, in the case where the number of package signal pins equals the on-chip bus width, then a different approach can be used.

In this chapter a *bus stuttering* encoder is presented. In the stutter encoder, if back-to-back vectors (or states) that are being transmitted off-chip induce a transition which results in a noise limit violation, an intermediate state is inserted between them. This intermediate or *stutter* state offers a method to transition between the two original states (without directly switching between them) in such a manner that each of the intermediate transitions do not result in a noise limit violation. It is possible that more than one stutter state is required in some cases. In this fashion, the transition between the original states is performed by inserting one or more stutter states. During the time when the stutter states are being transmitted off-chip, the clock that is used to latch the data on the receiver is gated out. This assumes a source synchronous data transmission scheme. This allows the receiver to be implemented using standard circuitry. The receiver will only acquire valid data since it only latches data when a valid clock edge is present. The number of stutter states that are used in the data transmission will depend on the aggressiveness of the user-defined noise limits.

## 7.1 Encoder Construction

The steps in creating the stutter encoder are similar to those used in constructing the bus expansion encoder up to the point where the directed graph is processed. The user still writes constraint equations which are evaluated for each possible transition on the bus. Transitions which result in a user-defined noise limit violation are removed from the subset of legal transitions. The remaining legal transitions are used to create the directed graph $G$ which represents all of the legal paths between any two vertices. The graph is represented implicitly and efficiently using ROBDDs [11, 12]. At this point, the user can run either the bus expansion algorithm or the bus stuttering algorithm on the directed graph $G$.

### 7.1.1 Encoder Algorithm

The stutter encoder is constructed by evaluating each vertex $v_s$ of $G(V, E)$ and finding the shortest path between $v_s$ and any destination vertex $v_d$ in $G$ using only legal edges of $G$ (including self-edges). In order to be able to construct a stutter encoder, the following conditions must hold:

- There must exist at least two outgoing edges (including the self-edge) for each $v_s \in G$.

- There must exist at least two incoming edges (including the self-edge) for each $v_d \in G$.

These requirements ensure that for the directed graph $G$, each vertex can reach at least one other vertex and can be reached by at least one other vertex. If these requirements are not met, then the user must relax the user-defined noise limits until both conditions are met.

Given $G$, the algorithm first tests whether both of the above mentioned requirements are satisfied. The algorithm next attempts to determine the number of intermediate steps required for a vertex $v_s \in G$ to reach another vertex $v_d \in G$. If $v_d$ can be reached with just one edge, the algorithm records the transition as a **direct** transition (one that requires 0 stutter steps).

For the case where $v_s$ can *not* reach $v_d$ with only one edge, then at least one stutter state is needed to complete the transition. The algorithm then attempts to find a path between $v_s$ and $v_d$ using two edges. Since the set of vertices $V_d$ that can be reached from $v_d$ by means of a direct path is known, the algorithm simply needs to find an edge from $v_s$ to $v \in V_d$. Once such a path between $v_s$ and $v_d$ is found, then the algorithm records the intermediate vertex as a necessary *stutter state* which is required between $v_s$ and $v_d$. This process is repeated for transitions which require more stutter states.

Algorithm 3 contains the pseudo-code for the stutter encoder algorithm. All *transition_path* variables are initially initialized to be empty. The routine $find\_path(v_s, v_d, l)$ returns a shortest path from the source $v_s$ to destination $v_d$, with path_length $l$.

**Algorithm 3** Constructing the Stutter Encoder

```
for $(each\ v_s \in V)$ do
  for $(each\ v_c \in V)$ do
    for $(path\_length = 1;\ path\_length < 2^{W_{bus}-1};\ path\_length++)$ do
      if $(transition\_path[v_s, v_d]\ ==\ \phi)$ then
        $transition\_path[v_s, v_d]\ = find\_path(v_s, v_d, path\_length)$
      end if
    end for
  end for
end for
$find\_edge(v_s, v_d, l)$
if $(\exists v_1, v_2, \cdots v_{(l-1)} S.T.((v_s, v_1) \in E) \wedge ((v_2, v_2) \in E) \wedge \cdots \wedge ((v_{(l-1)}, v_d) \in E))$ then
  $return\ (v_s, v_1, v_2, \cdots, v_{(l-1)}, v_d)$
else
  $return\ \phi$
end if
```

The maximum possible number of stutter states that may be used is $(2^{W_{bus}} - 1)$. This represents the worst-case where in order to transition from $v_s$ to $v_d$, each and every other vertex within $G$ must be used as a stutter state. While this represents the absolute theoretical worst case, experimental results have shown that the number of stutter states is typically between 0 and 3 for bus segments up to 8 bits. In real application of the stutter encoder, the actual number of stutter states that would ever be inserted within any data sequence is $W_{bus} - 1$ since this represents the total number of bits that could ever switch between two data vertices. It should be noted that this analysis is only performed on a representative bus segment (which is typically very small compared to the entire off-chip bus).

The construction of the encoder presented in this thesis is targeted at improving the performance of the off-chip data transmission. For this application, the shortest path between vertices is selected as the optimal choice for achieving maximum throughput. This encoder can be modified for optimization of other constraints such as synthesis complexity, circuit area, or circuit delay by selecting other routes between vertices other than the shortest path.

### 7.1.2 Encoder Overhead

To calculate the overhead of the stutter encoder, it is assumed that each vector $v_s \subseteq V$ has an equal probability of occurring on the bus. Using this assumption, a sequence of data patterns is constructed, in which each and every sequence occurs on the bus at least once. When this sequence is transmitted, the minimum number of stutter states are inserted in the transition between any pair of vectors. The maximum number of stutter states that will be inserted in a sequence for any given encoder is $2^{W_{bus}-1}$. Equation 7.1 gives the overhead of the stutter encoder.

$$Overhead_{bus-stuttering} = \left(\frac{\sum_{k=1}^{2^{(W_{bus}-1)}} (\#\ Transitions\ Requiring\ k\ Stutters) \cdot k}{2^{(2 \cdot W_{bus})}}\right) \cdot 100 \quad (7.1)$$

## 7.2 Decoder Construction

The stutter encoding technique assumes a source synchronous clocking architecture. In source synchronous clocking, the bus clock is generated at the transmitter and synchronized to the off-chip data being transmitted. The clock is then transmitted along with the data in the off-chip bus. By doing this, the timing correlation between the clock and data is extremely tight. This architecture has seen wide adoption in industry as a way to address channel-to-channel skew and common mode noise.

The stutter encoding technique is specifically designed for a source synchronous architecture. The encoding circuitry gates out the source synchronous clock when stutter states are transmitted off-chip. Since the receiving circuitry only aquires data on the rising edge of the source synchronous clock, the stutter states are ignored. In this manner, no special decoding circuitry is needed.

## 7.3 Experimental Results

To validate the feasibility of the stutter encoding technique, experimental results were performed on an example bus segment. For this example the off-chip bus was implemented using a BGA wire bonded package with parasitics given in Table 2.1. The off-chip bus uses a fixed $\frac{di}{dt}$ of 8 $\frac{MA}{s}$. Bus segments with between 2 to 8 signal pins were encoded using the stuttering technique, assuming that each segment had one $V_{DD}$ pin and one $V_{SS}$ pin. This bus was encoded using two sets of constraints – **aggressive** ($P_{supply}$, $P_{gnd}$, $P_0$, $P_1$, $P_{-1}$, -$P_{BW}$, and $P_\Gamma$ set to 5%.) and **non-aggressive** ($P_{supply}$, $P_{gnd}$, $P_0$, $P_1$, $P_{-1}$, -$P_{BW}$, and $P_\Gamma$ set to 10%). For these bus configurations and noise limits, all possible transitions were evaluated using the constraint equations described in Section 6.4. The remaining legal transitions were used in creating the directed graph $G$ which was used to construct the stuttering encoder. Using Algorithm 3, the shortest paths between all possible vertices were found. In addition, the number of stutter states required to complete the intended transition were found. Table 7.1 lists the percentage of transitions that require stutter states for each of the bus sizes in this example. Using Equation 7.1, the overhead of each of the encoders was calculated. Figure 7.1 shows the overhead of the stutter encoders as a function of bus segment size. Observe that for bus segment sizes less than 4 bits (aggressive constraints) and 6 bits (non-aggressive constraints), the overhead is less than 10%.

| Noise Limit | Bus Size | Number of Stutter States | | | |
|---|---|---|---|---|---|
| - | - | 0 | 1 | 2 | 3 |
| 5% Limit | 2 | 100 | 0 | 0 | 0 |
| | 3 | 96.9 | 3.1 | 0 | 0 |
| | 4 | 89.8 | 10.2 | 0 | 0 |
| | 5 | 79.3 | 20.5 | 0.2 | 0 |
| | 6 | 66.6 | 32.5 | 0.9 | 0 |
| | 7 | 53.4 | 44 | 2.6 | 0 |
| | 8 | 41.1 | 53.4 | 5.4 | 0.1 |
| 10% Limit | 2 | 100 | 0 | 0 | 0 |
| | 3 | 100 | 0 | 0 | 0 |
| | 4 | 99.2 | 0.8 | 0 | 0 |
| | 5 | 96.9 | 3.1 | 0 | 0 |
| | 6 | 92.5 | 7.5 | 0 | 0 |
| | 7 | 85.9 | 14.1 | 0 | 0 |
| | 8 | 77.3 | 22.6 | 0.1 | 0 |

Table 7.1: Percentage of Transitions Requiring Stutter States

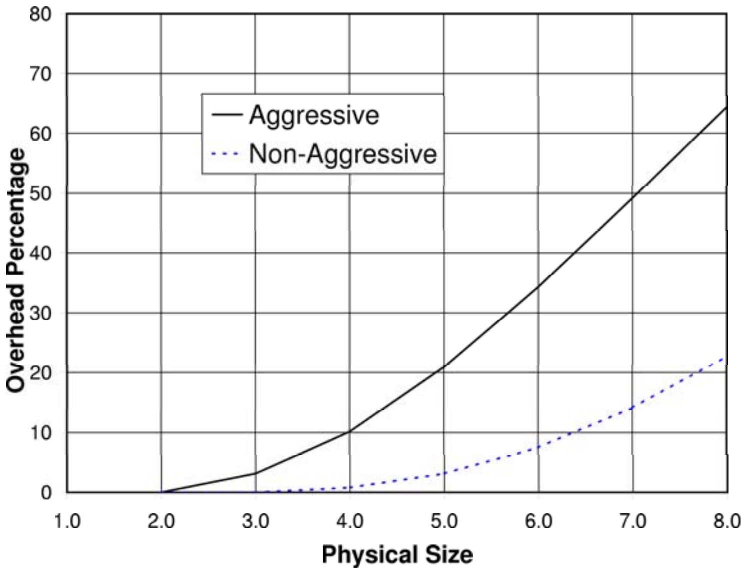


Figure 7.1: Bus Stuttering Encoder Overhead for the Fixed $\frac{di}{dt}$ Example

Figure 7.2 shows the bus throughput improvement when using the stutter encoding technique. This figure shows the percentage of throughput improvement of the bus (including the overhead of the encoder). The improvement of the *aggressive* encoder reaches 225% at a bus size of 6-bits. Beyond 6 bits, the overhead of the stutter encoder begins to outweigh the increased per-pin datarate (due to the increasing number of stutter states needed to avoid the noise limits).

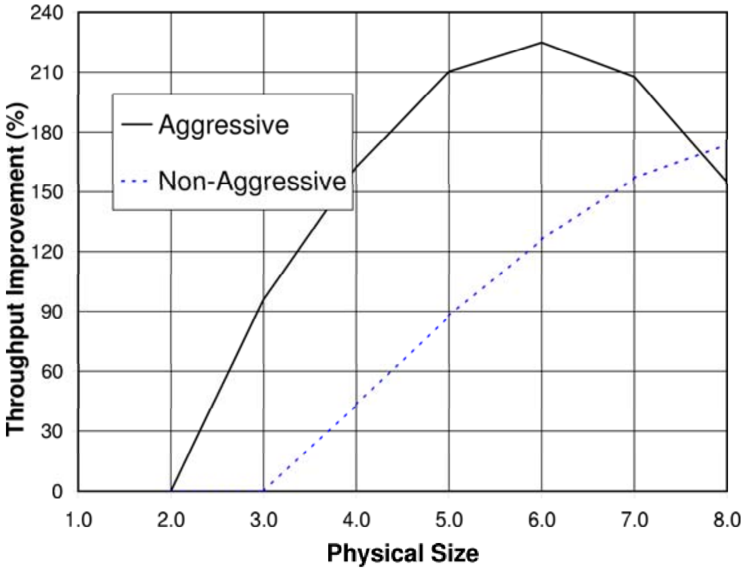


Figure 7.2: Bus Stuttering Encoder Throughput Improvement

### 7.3.1 Functional Implementation

The bus stuttering encoders were implemented to verify their functionality and feasibility. Once again, the Verilog hardware description language was used to implement the encoder circuitry [68, 69]. The implementation consists of a pipeline in which each stage of the pipe is routed to a multiplexer whose output drives the bus patterns off-chip. A state machine monitors the incoming data from the core of the IC to check whether a stutter state is needed in the off-chip transmission. At the beginning of circuit operation, the output of the first pipeline stage is selected to be the output of the multiplexer. When a sequence of vectors occurs which require a stutter state, then the multiplexer is switched to the state machine input where the appropriate stutter state is output. After the stutter state(s) is output, then the state machine switches the multiplexer back to the pipeline but now selects the next stage of the pipe. The state machine continues to monitor the pipeline for illegal consecutive states and inserts the appropriate number of stutter states in the off-chip data transmission. During the time in which a stutter state is being selected as the output of the multiplexer, the control state machine gates out the source synchronous clock. By gating out the clock, it is ensured that the receiver will not latch any of the stutter states.

The pipeline in the stutter encoder is used to compare consecutive states within the data sequence. Each time that a stutter state is output on the bus, the encoded data sequence falls one clock period behind the original data. The depth of the pipeline is dependant on how many stutter states could potentially occur on the bus. The bus protocol dictates the maximum length of consecutive data that could occur on the bus. This, in turn, dictates the maximum depth of the pipeline that is needed to prevent overflow. In practice, bus protocols have scheduled idle periods when the pipeline can be reset [82, 83, 84]. For this work, 32 pipeline stages were used in the implementation of the encoder. Figure 7.3 shows the schematic for the stutter encoder circuit.

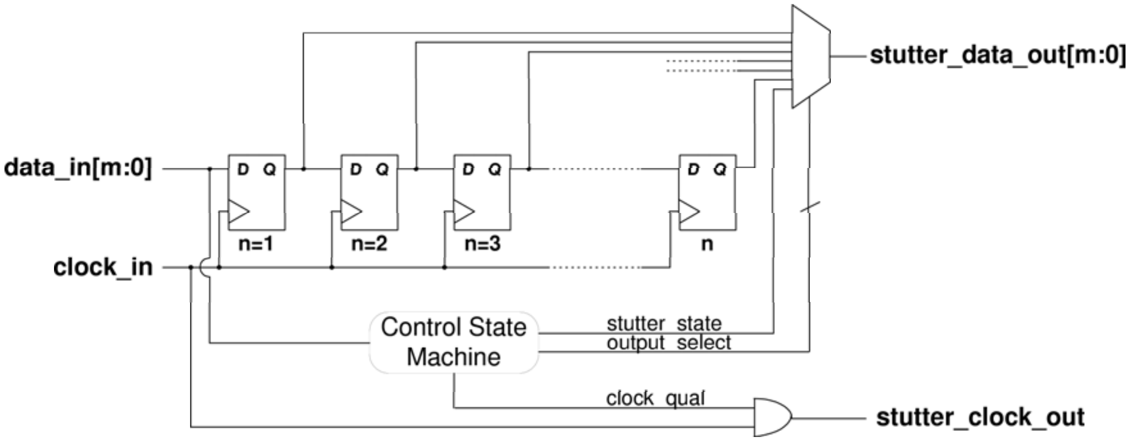
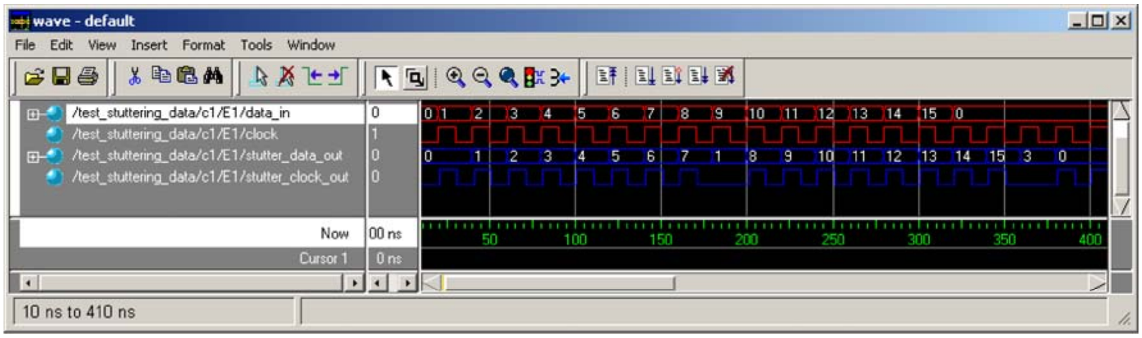


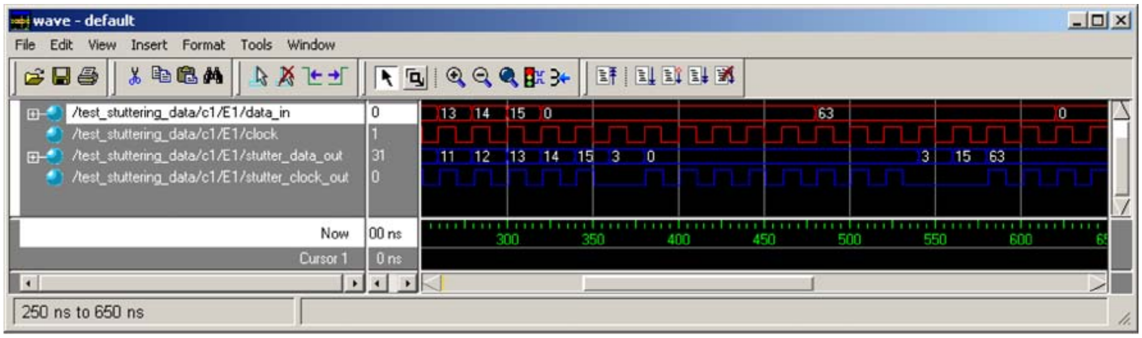
Figure 7.3: Bus Stuttering Encoder Schematic

Figures 7.4 through 7.6 show the Verilog simulation results for aggressively constrained bus stuttering encoders for sizes of 4, 6, and 8 bits. For each waveform the original data and clock are plotted in addition to the encoded data and gated clock which are transmitted off-chip.



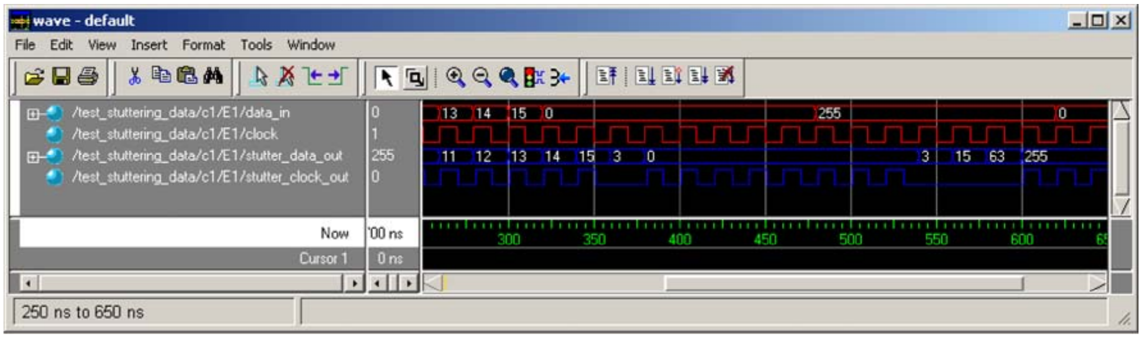
```
/test_stuttering_data/c1/E1/data_in           0
/test_stuttering_data/c1/E1/clock             1
/test_stuttering_data/c1/E1/stutter_data_out  0
/test_stuttering_data/c1/E1/stutter_clock_out 0
Now       00 ns
Cursor 1  0 ns
10 ns to 410 ns
```

Figure 7.4: Verilog Simulation Results for a 4-Bit Bus Stuttering Encoder



```
/test_stuttering_data/c1/E1/data_in           0
/test_stuttering_data/c1/E1/clock             1
/test_stuttering_data/c1/E1/stutter_data_out  31
/test_stuttering_data/c1/E1/stutter_clock_out 0
Now       00 ns
Cursor 1  0 ns
250 ns to 650 ns
```

Figure 7.5: Verilog Simulation Results for a 6-Bit Bus Stuttering Encoder



```
/test_stuttering_data/c1/E1/data_in           0
/test_stuttering_data/c1/E1/clock             1
/test_stuttering_data/c1/E1/stutter_data_out  255
/test_stuttering_data/c1/E1/stutter_clock_out 0
Now       '00 ns
Cursor 1  0 ns
250 ns to 650 ns
```

Figure 7.6: Verilog Simulation Results for a 8-Bit Bus Stuttering Encoder

### 7.3.2 Physical Implementation

In order to validate the feasibility of the stutter encoder, the physical design of the CODEC was performed.

#### 7.3.2.1 TSMC 0.13um ASIC Process

The stutter encoders were synthesized using the TSMC $0.13\mu m$ CMOS IC process to quantify their on-chip delay and area. Encoders for bus sizes of 4, 6, and 8 were implemented. For each of these sizes, both the aggressive (5% of $V_{DD}$) and the non-aggressive (10% of $V_{DD}$) stutter encoders were synthesized. Table 7.2 lists the delay and area impact of the bus stuttering encoders when implemented in a TSMC $0.13\mu m$ process. This table shows that the stuttering encoder takes more area than the bus expansion encoder due to the state machines and pipeline stages associated with the design; however, when analyzing modern off-chip busses such as PCI Express, Rapid I/O, and HyperTransport, it is found that all of these busses already contain pipeline stages and state machines to handle encoding/decoding required by the protocol. This means that the stuttering encoder design could simply be included in the pre-existing state machine and pipeline stages, thereby reducing the associated overhead.

In any case, the physical sizes are still negligible when considering reasonably-sized VLSI die sizes. For the largest stutter encoder listed in this table (8-bit aggressively encoded), the area consumed is still less than 1.5% of a $5mm^2$ die. The un-optimized circuit delay is reported, which could be reduced using more advanced circuit techniques. For the stutter encoder circuit, the combinational logic in the control state machine was the largest source of delay in the system.

| | Bus Size | Noise Limit | |
|---|---|---|---|
| | - | 5% (aggressive) | 10% (non-aggressive) |
| Delay ($ns$) | 4 | 2.02 | 1.99 |
| | 6 | 2.42 | 2.38 |
| | 8 | 2.85 | 2.79 |
| Area ($um^2$) | 4 | 311k | 310k |
| | 6 | 362k | 345k |
| | 8 | 382k | 368k |

Table 7.2: Bus Stuttering Encoder Synthesis Results in a TSMC 0.13$um$ Process

#### 7.3.2.2 Xilinx 0.35um FPGA Process

The encoders were also synthesized, mapped, and implemented for a *Xilinx VirtexIIPro*, Field Programmable Gate Array (FPGA). Stutter encoders for bus sizes of 4, 6, and 8 bits were implemented using both the aggressive and non-aggressive constraints. Table 7.3 lists the delay and area impact of the bus stuttering encoders when implemented in the FPGA process. In all cases, the bus stuttering encoder designs took up less than 1.5% of the total FPGA resources.

| | Bus Size | Noise Limit |
|---|---|---|
| | - | 5% (aggressive) & 10% (non-aggressive) |
| Delay ($ns$) | 4 | 4.78 |
| | 6 | 5.29 |
| | 8 | 5.89 |
| FPGA Usage | 4 | $< 1\%$ |
| | 6 | $< 1\%$ |
| | 8 | $< 1.5\%$ |

Table 7.3: Bus Stuttering Encoder Synthesis Results for a Xilinx VirtexIIPro FPGA

### 7.3.3 Measurement Results

Once again, the test setup in Figure 6.11 was used to verify the actual performance of the encoders. The outputs of the FPGA were again monitored using the 16950A Logic Analyzer from *Agilent Technologies, Inc.* Figures 7.7 through 7.9 show the logic analyzer measurement results for aggressively constrained bus stuttering encoders of sizes of 4, 6, and 8. For each waveform, the original data and clock are plotted in addition to the encoded data and gated clock (which is transmitted off-chip). The measurement results match the functional simulations which validate that the design can be implemented in a real target system. For the bus configuration of this FPGA (SPG=3:1:3), the noise was reduced from 16% to 4% using the stutter encoding technique.

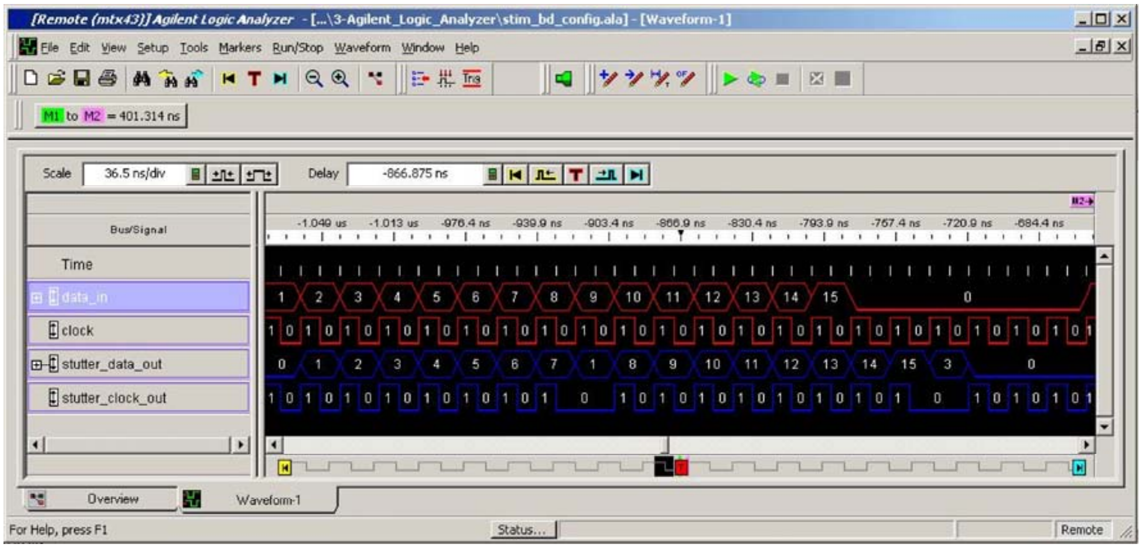
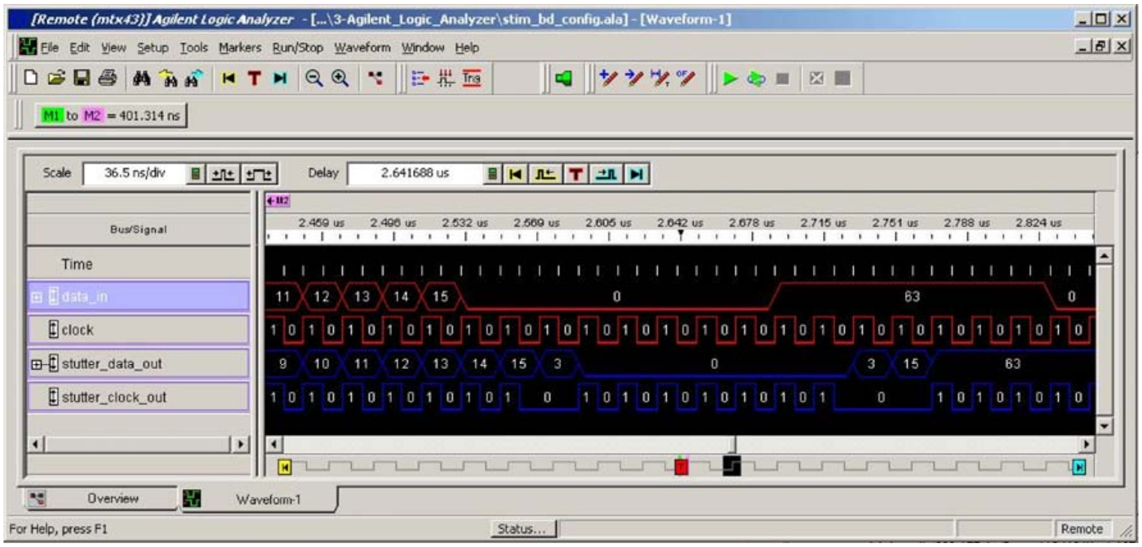

[Remote (mtx43)] Agilent Logic Analyzer - [...\3-Agilent_Logic_Analyzer\stim_bd_config.ala] - [Waveform-1]
File Edit View Setup Tools Markers Run/Stop Waveform Window Help
M1 to M2 = 401.314 ns
Scale 36.5 ns/div
Delay -866.875 ns
-1.049 us -1.013 us -976.4 ns -939.9 ns -903.4 ns -866.9 ns -830.4 ns -793.9 ns -757.4 ns -720.9 ns -684.4 ns
Bus/Signal
Time
data_in
clock
stutter_data_out
stutter_clock_out
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0
0 1 2 3 4 5 6 7 1 8 9 10 11 12 13 14 15 3 0
Overview
Waveform-1
For Help, press F1
Status...
Remote

Figure 7.7: Logic Analyzer Measurements of a 4-Bit Bus Stuttering Encoder


[Remote (mtx43)] Agilent Logic Analyzer - [...\3-Agilent_Logic_Analyzer\stim_bd_config.ala] - [Waveform-1]

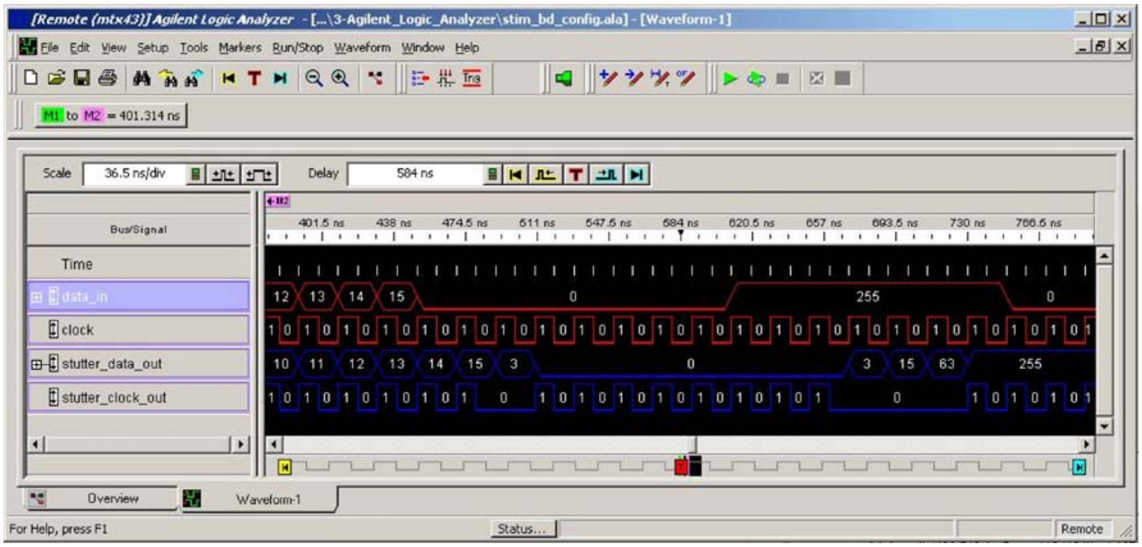
M1 to M2 = 401.314 ns

Scale 36.5 ns/div

Delay 2.641688 us

File Edit View Setup Tools Markers Run/Stop Waveform Window Help
Bus/Signal
Time
data_in
clock
stutter_data_out
stutter_clock_out
2.459 us 2.496 us 2.532 us 2.569 us 2.605 us 2.642 us 2.678 us 2.715 us 2.751 us 2.788 us 2.824 us
11 12 13 14 15 0 63 0
9 10 11 12 13 14 15 3 0 3 15 63
Overview
Waveform-1
For Help, press F1
Status...
Remote

Figure 7.8: Logic Analyzer Measurements of a 6-Bit Bus Stuttering Encoder


[Remote (mtx43)] Agilent Logic Analyzer - [...\3-Agilent_Logic_Analyzer\stim_bd_config.ala] - [Waveform-1]

M1 to M2 = 401.314 ns

Scale 36.5 ns/div

Delay 584 ns

Figure 7.9: Logic Analyzer Measurements of a 8-Bit Bus Stuttering Encoder

### 7.3.4 Discussion

The bus expansion encoding technique presented in Chapter 6 and the stutter encoding technique presented in this chapter both avoid patterns on the off-chip bus which result in noise limit violations. Both encoders are shown to improve performance even after considering the overhead of the encoder. When to use one encoding scheme versus the other depends on the application. The bus expansion encoder is aimed at generic off-chip busses which do not contain any special data processing circuitry for the off-chip transmission. In these cases, the encoder area impact is minimal since it is the only circuitry placed in the path of the off-chip data.

The bus stuttering encoder is targeted at source synchronous, protocol based busses. In these cases, the pipeline and control state machines already exist to handle the protocol of the bus. Adding stutter encoding capability to the preexisting circuitry is incremental and can potentially have less area impact than would the bus expansion encoder depending on how aggressively the noise is constrained.

To select the appropriate encoding technique, the designer must analyze the area impact on the design. For wider, parallel busses the bus expansion encoder is the better choice. For source synchronous, protocol based busses the bus stuttering encoder is the better choice.

# Chapter 8

## Impedance Compensation

Impedance mismatches between the IC package and the system PCB cause reflections that lead to unwanted noise in the system. This noise limits the maximum datarate that a package can achieve (Equation 4.26). As described in Section 1.2, the impedance of the package interconnect is dictated by the self-inductance and self-capacitance of the physical structure in addition to its mutual inductance and mutual capacitance with respect to neighboring structures. Since the package interconnect was originally developed for mechanical robustness, the electrical parasitics of the interconnect structure can be considerable. In the majority of cases the resulting impedance of the package interconnect is not matched to that of the system. In addition, the mechanical structures are difficult to alter due to the complex manufacturing processes required to construct the interconnect. This makes it difficult and expensive to change the physical structure of the package to match the characteristic impedance to that of the system PCB.

Chapters 6 and 7 presented bus encoding techniques that can reduce the effective mutual inductive and mutual capacitive contributions to the impedance of a given signal pin. This is accomplished by avoiding the switching patterns which induce noise of a sufficiently high magnitude. This reduces the worst-case mutual inductive and mutual capacitive coupling the bus. By reducing the coupling contribution for a given signal pin, the variation in the impedance of the structure is reduced, thereby reducing the magnitude of the worst-case reflections from the package. While these encoding techniques address the mutual coupling between signal pins in the package, they do not address the self-inductance and self-capacitance of the package structure. As illustrated in Tables 2.1 and 2.2, the self-inductance and self-capacitance of the interconnect structure contribute the majority of the electrical parasitics to the characteristic impedance of the signal pin. Also illustrated in these tables is that the package interconnect is predominately inductive. The inductive nature of the interconnect leads to a relatively high impedance when compared to the majority of impedances used in system PCBs, which typically range between 50 to 75$\Omega$. The higher impedance of the package pins cause positive reflections according to Equation 1.13.

To reduce the impedance discontinuities introduced by the electrical parasitics of the package interconnect, this chapter introduces an *impedance compensation* technique. In this technique, additional *compensation* capacitance is added near the package interconnect in order to reduce the characteristic impedance of the structure. By inserting the compensation capacitance near the inductive interconnect, the overall impedance can be reduced by increasing the net capacitance in Equation 1.12. Two styles of compensators are presented, *static* and *dynamic*. In the static compensator, predefined capacitance is placed on the package and on the IC to *surround* the level 1 interconnect. In the dynamic compensation approach, programmable capacitors are placed on-chip that can be controlled by software to achieve the best impedance match. For both compensators, two styles of on-chip capacitors are examined. The first on-chip capacitor is a *Metal-Insulator-Metal* (MIM) device. The second is a *Device-Based* capacitor. Each type of capacitor is evaluated for variability across bias voltage values, and for implementation area in the context of impedance matching of the package structure.

## 8.1 Static Compensator

This section describes the static compensation technique to match the characteristic impedance of the package interconnect to that of the system PCB.

### 8.1.1 Methodology

Tables 2.1 and 2.2 illustrate that the interconnect for the packages studied in this work are mostly *inductive* when considering impedance. These tables also illustrate that the level 1 interconnect is the largest source of excess inductance within the package. Specifically, the wire bond interconnect is the largest contributor of inductance within any of the packages. Further, it is assumed that if the compensator is able to address the impedance associated with the wire-bond, it will also be able to address the inductance of any other package structure.

In the static compensator methodology, a predefined capacitance is placed on both sides of the level 1 interconnect. When considering the wire-bond interconnect, the compensation capacitance is placed on-chip as well as on-package. The on-chip capacitance ($C_{comp1}$) is placed near or beneath the ball bond pads of the wire-bond. The on-package capacitance ($C_{comp2}$) is placed near or beneath the wedge bond pads of the wire-bond. Figure 8.1 shows the optimal locations of the static compensation capacitance in a wire-bonded system.

The static compensation capacitors will alter the characteristic impedance of the package by increasing the capacitance value in Equation 1.12. Equations 8.1 and 8.2 provide the total compensation capacitance and characteristic impedance of the wire-

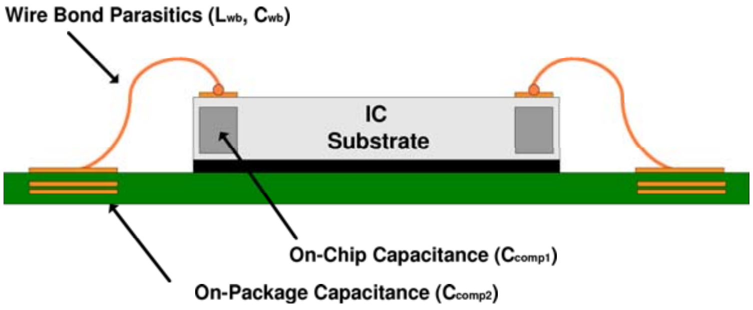

Wire Bond Parasitics ($L_{wb}$, $C_{wb}$)
IC
Substrate
On-Chip Capacitance ($C_{comp1}$)
On-Package Capacitance ($C_{comp2}$)

Figure 8.1: Cross-Section of Wire-Bonded System with Compensation Locations

bond interconnect after applying the static compensation capacitance.

$$C_{comp} = C_{comp1} + C_{comp2} \tag{8.1}$$

$$Z_{0-static} = \sqrt{\frac{L_{wb}}{C_{wb} + C_{comp1} + C_{comp2}}} \tag{8.2}$$

### 8.1.2 Compensator Proximity

In order to model the level 1 inductance and compensation capacitance as a single lumped element, the compensation capacitance must reside within a certain distance of the inductance. The maximum distance of the compensation capacitance from the interconnect inductance depends on the frequency components present in the forcing function of the incident digital signal. If the compensation capacitance resides spatially close to the inductance, the resulting structure can be modeled as a lumped element, and the impedance of the structure is then deterministic. Typically, a structure is considered a lumped element when its electrical length (i.e. its propagation delay) is less than 20% of the highest risetime in the system [66]. Figure 8.2 shows the physical length at which a structure can be treated as either a lumped or distributed element as a function of risetime (for several common package materials). Given a particular risetime, physical lengths less than indicated in this plot are treated as a lumped element. For physical lengths greater than this amount, the structure is treated as a distributed element. This plot considers the propagation delay of four common dielectric materials that are used in VLSI packaging.

For VLSI risetimes less than 1ns, this plot shows that structures with electrical lengths less than 200ps can be treated as lumped elements. Since standard package substrates utilize dielectric constants between Dk=3 to Dk=5, physical lengths less than approximately 1 inch may be considered lumped elements. This means that implementing embedded capacitors within the package or IC substrate can directly alter the characteristic impedance of the package interconnect (using a lumped model to compute the altered characteristic impedance). Specifically, embedding capacitors on-chip or on-package will be utilized to alter the characteristic impedance of the package interconnect.

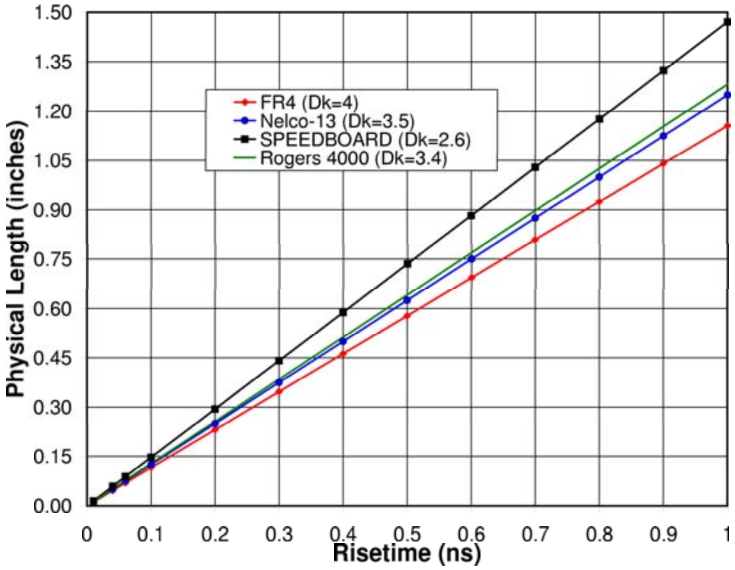


Figure 8.2: Physical Length at Which Structures Become Distributed Elements

### 8.1.3 On-Chip Capacitors

Two styles of on-chip capacitors were evaluated for use to implement $C_{comp1}$ in the static compensator design. The first style of on-chip capacitor is called a *Device-based* capacitor. This capacitor is created using the same process technology utilized in creating an NMOS transistor. The lower conducting plate of the capacitor is the heavily doped, p-type channel of the silicon transistor. The insulating material is the gate oxide of the NMOS transistor, which is a thin layer of silicon-oxide ($SiO_2$) above the channel of the device. The upper plate of the capacitor is the *PolySilicon* (*poly-Si*) gate of the NMOS device. The drain and source terminals are connected together and form one electrode of the capacitor, while the PolySilicon gate forms the other electrode. This style of capacitor has a very high capacitance density due to the thin plate separation which is identical to the gate thickness ($t_{ox}$) of the device. In this work, a $0.1\mu m$, BPTM process [78] is used to implement the on-chip capacitors. Using this technology, oxide layers as thin as $25\AA$ can be realized with a dielectric constant of $D_k = 3.9$. This translates into a capacitance density of $13.7fF/um^2$.

While the device-based capacitor achieves the highest density of any other style of on-chip capacitor, it suffers from two major drawbacks. The first drawback is that the capacitance value will change as a function of bias voltage. This nonlinearity is due to the fact that as bias voltage changes, the device goes from cut-off to weak inversion and strong inversion. The nonlinearity of the capacitance as a function of bias voltage is a problem in digital applications, since the bias voltage typically varies from $V_{SS}$ to $V_{DD}$. The second drawback of this style of capacitor is that it consumes valuable area within the IC that is normally reserved for the implementation of digital circuitry.

The second type of on-chip capacitor studied in this work is called a *Metal-Insulator-Metal* (MIM) capacitor. In a MIM capacitor, an additional process step is used to form an extra metal layer in the upper layers of the IC (typically above metal

3). This extra process step inserts an additional metal layer above the standard metal 3 layer such that the insulator thickness between this new layer and metal 3 is much lower than the inter-layer dielectric between metal 3 and 4. This extra step allows the creation of a parallel plate capacitor that uses a standard metal layer as the lower plate (metal 3) and an additional MIM layer as the upper plate (metal 4'). Since the extra MIM layer is located spatially closer to the lower plate, a higher capacitance density is achieved compared to the capacitor with metal 3 and metal 4 plates. In addition, the extra MIM layer does not take up any routing area on metal 4. MIM technology is able to achieve plate-to-plate separations as thin as $t_{ox} = 100\AA$, which translates into a capacitance density of $1.15fF/um^2$. While MIM capacitors have a lower capacitance density relative to device-based capacitors, their capacitance is constant over the full range of bias voltages. This is because both plates within the MIM capacitor are formed with standard metal with passive materials between the plates.

For the static compensator, both styles of on-chip capacitors are investigated. For each capacitor, the nonlinearity with bias voltage and area utilization are quantified. Figure 8.3 shows the cross-section for both styles of on-chip capacitors.

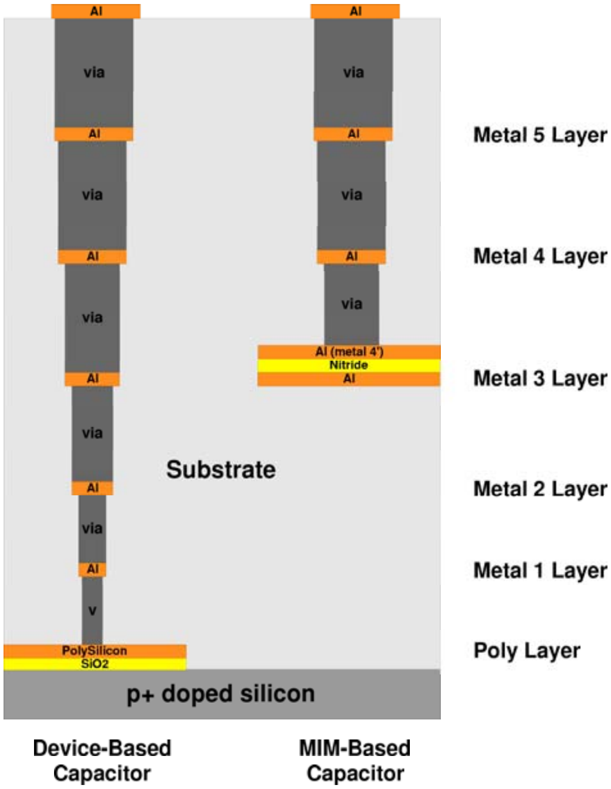


Figure 8.3: On-Chip Capacitor Cross-Section

| Structure | $C_{density}$ | Size | $C_{total}$ | Notes |
|---|---|---|---|---|
| MIM-Based Capacitor | $1.15\ fF/um^2$ | $100um^2$ | $11.5\ pF$ | $t_{ox} = 600\mathring{A}(Si_3N_4), D_k = 7$ |
| Device-Based Capacitor ($V_{GS} = 0v$) | $2.7\ fF/um^2$ | $100um^2$ | $27\ pF$ | $t_{ox} = 25\mathring{A}(SiO_2), D_k = 3.9$ |
| Device-Based Capacitor ($V_{GS} = 1.5v$) | $13.8\ fF/um^2$ | $100um^2$ | $138\ pF$ | $t_{ox} = 25\mathring{A}(SiO_2), D_k = 3.9$ |
| Embedded Capacitor | $420\ fF/mm^2$ | $0.5mm^2$ | $0.105\ pF$ | $t_{ox} = 0.051\text{mm}(FR4), D_k = 4$ |

Table 8.1: Density and Linearity of Capacitors Used for Compensation

### 8.1.4 On-Package Capacitors

Embedded PCB capacitors are evaluated to implement $C_{comp2}$ in the static compensator design. An embedded PCB capacitor (EC) is simply a parallel plate capacitor that is formed using standard metal layers within the PCB lamination. Embedded capacitors are preferred over surface mount (SMT) components due to the simplicity of their implementation. Since the capacitors are constructed using simple metal-to-metal area, no surface mount pads, loading process, or through-hole vias are needed. While ECs are easy to implement, they typically are only used for small values of capacitance (<10pF). Standard PCB lamination processes are able to achieve plane-to-plane separation as low as $0.051mm$. Using a standard package dielectric such as FR4 ($D_k = 4$), this translates into a capacitance density of $420fF/mm^2$. Table 8.1 lists the capacitance densities and variation with bias voltages for all of the capacitors used in the compensator design. For each capacitor, a reasonable size of capacitor is used for illistrative purposes.

### 8.1.5 Static Compensator Design

The static compensator is designed to alter the impedance of the largest inductive component of the package interconnect (i.e., the wire-bond). It is assumed that if this design methodology is successful in compensating for the wire-bond inductance, it will also be successful in compensating for interconnect structures with less inductive parasitics (i.e., flip-chip bumps).

Using the electrical parameters from Table 2.2 and solving Equations 8.1 and 8.2, the values of $C_{comp1}$ and $C_{comp2}$ for the static compensator can be found (assuming $C_{comp1}$=$C_{comp2}$). Table 8.2 lists the optimal capacitor values required to match the wire bond impedances (Table 2.2) to 50Ω. Table 8.3 lists the corresponding sizes of the different types of capacitors required to realize $C_{comp1}$ and $C_{comp2}$. This table lists the sizes for both the MIM-based and Device-based on-chip capacitor implementations of $C_{comp1}$.

| $Length_{wb}$ | $C_{comp1}$ | $C_{comp2}$ |
|---|---|---|
| 1 mm | 102fF | 102fF |
| 2 mm | 208fF | 208fF |
| 3 mm | 325fF | 325fF |
| 4 mm | 450fF | 450fF |
| 5 mm | 575fF | 575fF |

Table 8.2: Static Compensation Capacitor Values

| $Length_{wb}$ | $C_{comp1-MIM}$ | $C_{comp1-Device}$ | $C_{comp2-EC}$ |
|---|---|---|---|
| 1 mm | 10$\mu$m x 10$\mu$m | 2.7$\mu$m x 2.7$\mu$m | 388$\mu$m x 388$\mu$m |
| 2 mm | 14$\mu$m x 14$\mu$m | 3.9$\mu$m x 3.9$\mu$m | 554$\mu$m x 554$\mu$m |
| 3 mm | 18$\mu$m x 18$\mu$m | 4.9$\mu$m x 4.9$\mu$m | 692$\mu$m x 692$\mu$m |
| 4 mm | 21$\mu$m x 21$\mu$m | 5.8$\mu$m x 5.8$\mu$m | 815$\mu$m x 815$\mu$m |
| 5 mm | 24$\mu$m x 24$\mu$m | 6.5$\mu$m x 6.5$\mu$m | 921$\mu$m x 921$\mu$m |

Table 8.3: Static Compensation Capacitor Sizes

### 8.1.6 Experimental Results

In order to evaluate the performance of the static compensator, SPICE simulations were performed on all lengths of wire bonds listed in Table 8.2. Figure 8.4 shows the simulated Time Domain Reflectrometry (TDR) of the static compensator (Equation 1.14). A TDR simulation shows how much of a reflection ($\Gamma$) is caused by the wire bond. For each length of wire bond (1mm to 5mm), a 117ps (3GHz) input step is used to stimulate the wire bond. In Figure 8.4, the TDR waveforms are offset in the voltage axis for view-ability with the 1mm curve on the top and the 5mm curve on bottom. Reflections off of an *inductive* interconnect (i.e., $Z_L \geq Z_0$) will result in the waveform traveling above its steady state value. Reflections off of a *capacitive* interconnect (i.e., $Z_L \leq Z_0$) will result in the waveform traveling below its steady state value. For each length of wire bond the non-compensated, MIM-based, and Device-based static compensation curves are shown. This figure shows the dramatic reduction in wire bond reflections when using a static compensator. Table 8.4 reports the reduction in reflections when using the static compensator(s).

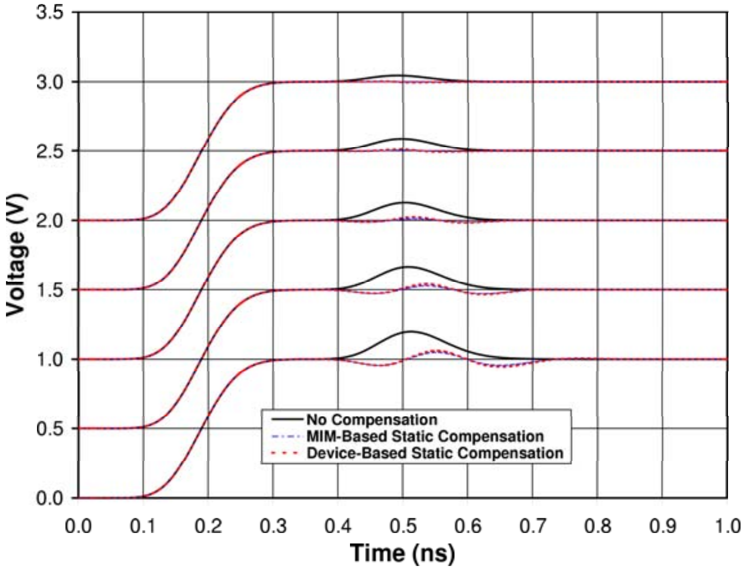


Figure 8.4: Static Compensator TDR Simulation Results

| $Length_{wb}$ | $\Gamma_{No-Comp}$ | $\Gamma_{MIM-Comp}$ | $\Gamma_{Device-Comp}$ |
|---|---|---|---|
| 1 mm | 4.5% | 0.05% | 0.5% |
| 2 mm | 8.7% | 0.4% | 1.2% |
| 3 mm | 12.7% | 1.3% | 2.4% |
| 4 mm | 16.4% | 2.7% | 4.1% |
| 5 mm | 19.8% | 4.8% | 6.0% |

Table 8.4: Reflection Reduction Due to Static Compensator

Another way to observe the effect of the compensator is to observe the input impedance of the structure in the frequency domain. Figure 8.5 shows the input impedance of the wire bond structure versus frequency for the 3mm wire bond. In this figure, the non-compensated, MIM-based, and Device-based static compensation curves are again shown. To compare the performance we record the frequency at which the input impedance deviates by 10$\Omega$ from the target impedance. In this case, the compensators are designed to match the structure to 50$\Omega$. This figure illustrates that adding a static compensator can keep the lumped impedance of the wire bond closer to 50$\Omega$ up to a much higher frequency. For this example, the 3mm wire bond was kept to within 10$\Omega$ of the target impedance (50$\Omega$) up to 4.8GHz when using the MIM-based compensator (compared to only 3.1GHz when considering the uncompensated wire bond).

Table 8.5 lists the frequencies at which the input impedance strays to +/-10$\Omega$ from the target for all of the lengths of wire bonds evaluated.

These results show the dramatic reduction in impedance discontinuities when using a static compensator. In all cases, the MIM-based compensator outperformed the Device-based compensator.

| $Length_{wb}$ | $f_{No-Comp}$ | $f_{MIM-Comp}$ | $f_{Device-Comp}$ |
|---|---|---|---|
| 1 mm | 9.3 GHz | 14 GHz | 12 GHz |
| 2 mm | 4.7 GHz | 7.1 GHz | 5.7 GHz |
| 3 mm | 3.1 GHz | 4.8 GHz | 3.8 GHz |
| 4 mm | 2.4 GHz | 3.7 GHz | 2.9 GHz |
| 5 mm | 1.9 GHz | 3.0 GHz | 2.5 GHz |

Table 8.5: Frequency at Which Static Compensator is +/- 10Ω from Design

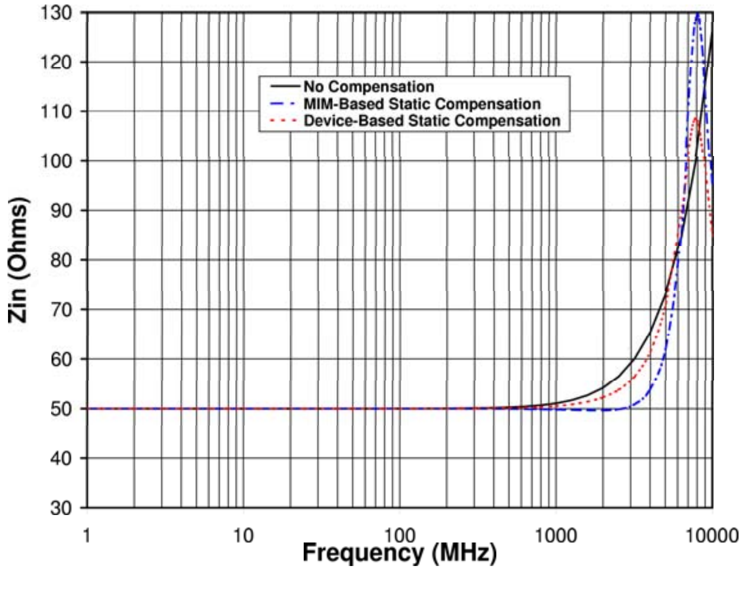


Figure 8.5: Static Compensator Input Impedance Simulation Results

## 8.2 Dynamic Compensator

This section describes the dynamic compensator technique used to match the impedance of the package interconnect to the characteristic impedance of the system.

### 8.2.1 Methodology

In the dynamic compensator methodology, capacitance is placed only on-chip ($C_{comp1}$). The term *dynamic* means that the capacitance is programmable through active circuitry on the chip. The programmability of the compensation capacitance allows the compensator to successfully match impedance across variations in the wire bond inductance. The programmability implies that the designer does not need to know the exact wire bond inductance prior to IC fabrication. This has the advantage that the dynamic compensator can accommodate process and design variation. In the dynamic compensator the net compensation capacitance is given by:

$$C_{comp} = C_{comp1} \tag{8.3}$$

Using Equation 1.12 to describe the impedance of the dynamically compensated structure, the impedance of the wire bond becomes:

$$Z_{0-dynamic} = \sqrt{\frac{L_{wb}}{C_{wb} + C_{comp1}}} \tag{8.4}$$

### 8.2.2 Dynamic Compensator Design

The design of the dynamic compensator consists of CMOS pass gates that connect to integrated binary-weighted, on-chip capacitors. In this design there are three integrated capacitors ($C_1$, $C_2$ and $C_3$) that can be switched in. The number of capacitors can be increased, however experimental results indicate that sufficient resolution can be achieved using just three capacitors. Each of these capacitors uses a pass gate to connect to the wire bond (Pass Gate #1, Pass Gate #2, Pass Gate #3). Each pass gate has a control signal which either connects/isolates the capacitors to/from the on-chip I/O pad, which is connected to one end of the wire bond. Figure 8.6 shows the schematic of the dynamic compensator design.

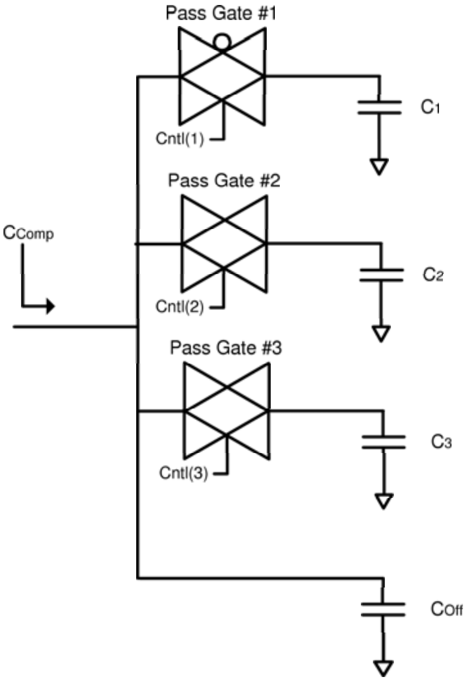


Figure 8.6: Dynamic Compensator Circuit

#### 8.2.2.1 Capacitor Design

The diffusion regions associated with the pass gates contribute an additional capacitance to $C_{comp1}$, which must be considered in the design. For each of the three programmable capacitor banks, the net capacitance will be the sum of the diffusion capacitance of the pass gate and the integrated capacitor (Equations 8.5 through 8.7). If the pass gate of any bank $i$ is turned off, then bank $i$ simply contributes a capacitance $C_{pgi}$ to $C_{comp1}$.

$$C_{Bank1} = C_{pg1} + C_1 \tag{8.5}$$

$$C_{Bank2} = C_{pg2} + C_2 \tag{8.6}$$

$$C_{Bank3} = C_{pg3} + C_3 \tag{8.7}$$

To achieve a larger programming range, the integrated capacitors are binary-weighted such that:

$$C_{Bank3} = 2 \cdot C_{Bank2} = 4 \cdot C_{Bank1} \tag{8.8}$$

Using three control bits, 8 programmable values of capacitance can be connected to the wire bond in increments of $C_{Bank1}$. $C_{Off}$ is a constant, nonprogrammable capacitor which serves as an offset capacitance. It therefore sets the minimum capacitance value of the compensator. Using the conventions just described, the range of the compensator can be described as:

$$C_{min} = C_{Off} = C_{pg1} + C_{pg2} + C_{pg3} \tag{8.9}$$

$$C_{max} = C_{Off} + C_{Bank1} + C_{Bank2} + C_{Bank3} \tag{8.10}$$

$$C_{step} = C_{Bank1} \tag{8.11}$$

In this work the pass gates are implemented using a $0.1\mu m$ CMOS process from BPTM. As with the static compensator, the integrated capacitors ($C_1$,$C_2$, $C_3$, $C_{Off}$) are implemented using the two different on-chip capacitance realization techniques that are evaluated in this work (MIM-based and Device-Based). Both of these capacitor techniques are evaluated for range, area efficiency, and nonlinearity for application in the dynamic compensator design.

### 8.2.2.2 Pass Gate Design

As described in the previous section, the diffusion capacitance of the pass gates will contribute to the total capacitance of each bank. The pass gate must be designed to have sufficient strength to drive the integrated capacitors ($C_1$, $C_2$, $C_3$). Typical CMOS design rules dictate that the pass gate sizing should be 1/3 of the size of an equivalent inverter that represents the integrated capacitor being driven [67]. This rule defines the amount of capacitance that will be present in each bank due to the pass gate and to the integrated capacitor.

$$C_{pg} = \frac{1}{3} \cdot C_{Bank} \tag{8.12}$$

$$C_{int} = \frac{2}{3} \cdot C_{Bank} \tag{8.13}$$

Using the Equations 8.5-8.13 and the values from Table 2.2, the sizing of the resulting compensation circuitry can be determined. Table 8.6 lists the values of the capacitors needed in the dynamic compensator that will match the impedance of the wire bond to 50Ω (Equation 8.4).

| $Length_{wb}$ | $C_{comp1}$ |
|---|---|
| 1 mm | 202fF |
| 2 mm | 403fF |
| 3 mm | 605fF |
| 4 mm | 806fF |
| 5 mm | 1008fF |

Table 8.6: Dynamic Compensation Capacitor Values

Table 8.7 lists the device sizes of the dynamic compensator circuit. The capacitances $C_1$, $C_2$ and $C_3$ are implemented as square devices for minimal area utilization. The total area of the compensator is computed as the size of the smallest enclosing square on the die. It is clear that the MIM-based dynamic compensator occupies more area than the Device-based compensator; however, the nonlinearity of both compensators must be analyzed to compare the applicability of the two circuits. In the next section, experimental results are presented that compare the two compensator designs.

| | MIM-Based | Device-Based |
|---|---|---|
| Component | Area (W $\times$ L) | Area (W $\times$ L) |
| Pass Gate #1 | 32.4$\mu$m x 0.1$\mu$m | 32.4$\mu$m x 0.1$\mu$m |
| Pass Gate #2 | 62.5$\mu$m x 0.1$\mu$m | 62.5$\mu$m x 0.1$\mu$m |
| Pass Gate #3 | 129.6$\mu$m x 0.1$\mu$m | 129.6$\mu$m x 0.1$\mu$m |
| $C_{off}$ | 8.5$\mu$m x 8.5$\mu$m | 2.5$\mu$m x 2.5$\mu$m |
| $C_1$ | 11$\mu$m x 11$\mu$m | 3.3$\mu$m x 3.3$\mu$m |
| $C_2$ | 15.5$\mu$m x 15.5$\mu$m | 4.6$\mu$m x 4.6$\mu$m |
| $C_3$ | 22$\mu$m x 22$\mu$m | 6.6$\mu$m x 6.6$\mu$m |
| Total | 65$\mu$m x 65$\mu$m | 25$\mu$m x 25$\mu$m |

Table 8.7: Dynamic Compensation Capacitor Sizes

### 8.2.3 Experimental Results

The same set of SPICE simulations as in the static compensator were performed on the dynamic compensator to evaluate its performance.

Figure 8.7 shows the simulated TDR of the dynamic compensator (Equation 1.14). Each length of wire bond (1mm to 5mm) is evaluated when stimulated with a 117ps (3GHz) input step. Again, the TDR waveforms are offset in the voltage axis for view-ability with the 1mm curve on the top and the 5mm curve on bottom. As in the case of the static compensator, we can observe that the dynamic compensator results in a dramatic reduction in reflections from the wire bond.

Table 8.8 reports the reduction in reflections when using the dynamic compensator(s), along with the binary control setting used for the compensation.

| $Length_{wb}$ | $\Gamma_{No-Comp}$ | $\Gamma_{MIM-Comp}$ | $\Gamma_{Device-Comp}$ | Setting |
|---|---|---|---|---|
| 1 mm | 4.5% | 1.0% | 1.0% | 001 |
| 2 mm | 8.7% | 1.8% | 1.3% | 011 |
| 3 mm | 12.7% | 3.6% | 3.0% | 100 |
| 4 mm | 16.4% | 4.3% | 3.3% | 110 |
| 5 mm | 19.8% | 6.0% | 5.0% | 111 |

Table 8.8: Reflection Reduction Due to Dynamic Compensator

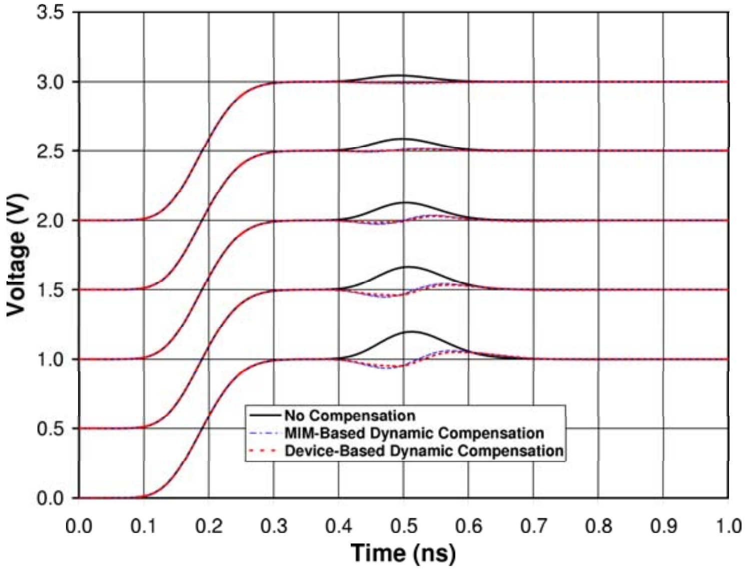


Figure 8.7: Dynamic Compensator TDR Simulation Results

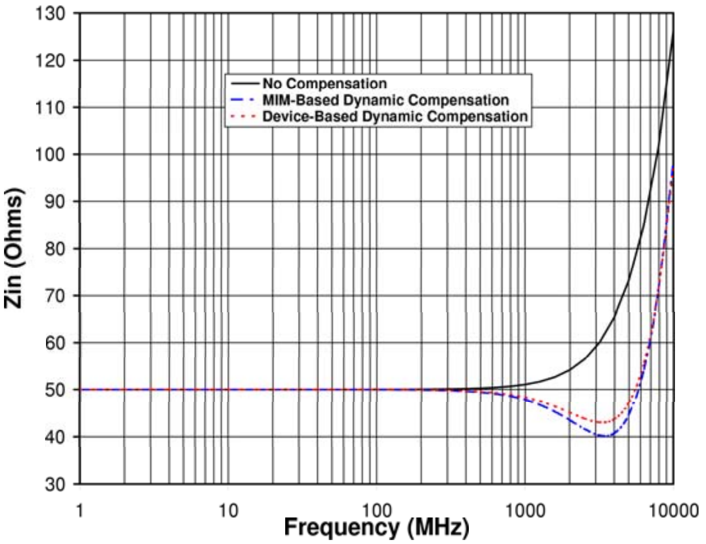


Figure 8.8: Dynamic Compensator Input Impedance Simulation Results

Figure 8.8 shows the input impedance of the wire bond structure versus frequency for the 3mm wire bond using the dynamic compensator. Once again, adding a compensator can keep the lumped impedance of the wire bond closer to 50Ω up to a much higher frequency. In this case, the 3mm wire bond was kept to within 10Ω of design up to 6.8GHz when using the MIM-based compensator (compared to only 3.1GHz when considering the uncompensated wire bond). The corresponding bandwidth of the Device-based compensator is 6.7GHz.

Table 8.9 lists the frequencies at which the package impedance deviates +/- 10Ω from its target impedance of 50Ω for all wire bond lengths, using the dynamic compensator.

Due to the nonlinearity of the Device-based capacitors and active pass gates, the variation of the dynamic compensator as a function of bias voltage was evaluated. For each dynamic compensator setting, the bias voltage was changed for $V_G = 0v$ to $V_G = 1.5v$, and the corresponding capacitance was recorded. Table 8.10 lists the effect of bias voltage on both the MIM-based and Device-based compensator circuits. This

| $Length_{wb}$ | $f_{No-Comp}$ | $f_{MIM-Comp}$ | $f_{Device-Comp}$ | Setting |
|---|---|---|---|---|
| 1 mm | 9.3 GHz | 20 GHz | 20 GHz | 001 |
| 2 mm | 4.7 GHz | 10.1 GHz | 10 GHz | 011 |
| 3 mm | 3.1 GHz | 6.8 GHz | 6.7 GHz | 100 |
| 4 mm | 2.4 GHz | 5.2 GHz | 5.1 GHz | 110 |
| 5 mm | 1.9 GHz | 4.2 GHz | 4.1 GHz | 111 |

Table 8.9: Frequency at Which Dynamic Compensator is +/- 10Ω from Design

clearly shows the nonlinearity of the Device-based compensator which experiences as much as 33% capacitance variation when programmed to its maximum setting. This variation matches the expected variation of standard CMOS PolySilicon gate capacitors [67]. Note that the MIM capacitors also exhibit variability (3.8%), which occurs due to the bias dependence of the pass gate diffusion capacitances [77]. While both dynamic circuits exhibit a bias voltage dependence, both the compensators have sufficient range to cover wire bond lengths from 1mm to 5mm.

These results illustrate that the Device-based compensator outperformed the MIM-based compensator when implemented in a dynamic architecture. Also, both dynamic compensators outperformed their static counterpart. The dynamic compensator has the flexibility to be integrated in all future VLSI designs as part of the standard process to reduce the growing impact of package reflections.

| | | MIM-Based Compensator | | | Device-Based Compensator | | |
|---|---|---|---|---|---|---|---|
| Setting | $C_{(desired)}$ | $C_{(V_{bias}=0v)}$ | $C_{(V_{bias}=1.5v)}$ | $C_{average}$ | $C_{(V_{bias}=0v)}$ | $C_{(V_{bias}=1.5v)}$ | $C_{average}$ |
| 001 | 200 fF | 252 fF | 262 fF | 257 fF | 222 fF | 281 fF | 251 fF |
| 010 | 325 fF | 373 fF | 382 fF | 378 fF | 318 fF | 414 fF | 366 fF |
| 011 | 450 fF | 499 fF | 540 fF | 519 fF | 423 fF | 587 fF | 505 fF |
| 100 | 575 fF | 588 fF | 596 fF | 592 fF | 485 fF | 651 fF | 568 fF |
| 101 | 700 fF | 713 fF | 754 fF | 734 fF | 592 fF | 816 fF | 704 fF |
| 110 | 825 fF | 828 fF | 895 fF | 862 fF | 688 fF | 968 fF | 828 fF |
| 111 | 950 fF | 948 fF | 1041 fF | 994 fF | 788 fF | 1180 fF | 984 fF |

Table 8.10: Dynamic Compensator Range and Linearity

### 8.2.4 Dynamic Compensator Calibration

To program the compensator to the optimal capacitance value, a calibration can be performed at package test.

The circuit used to perform the calibration is inspired by TDR ideas, but is simplified for applicability in a standard, low-cost VLSI tester setting. The calibration circuitry is simple and can be placed in the IC test equipment so that extra circuitry is not needed on the IC. In addition, pre-existing control logic and software interfaces in the tester can be used for calibration. Figure 8.9 shows the calibration circuitry for the compensator.

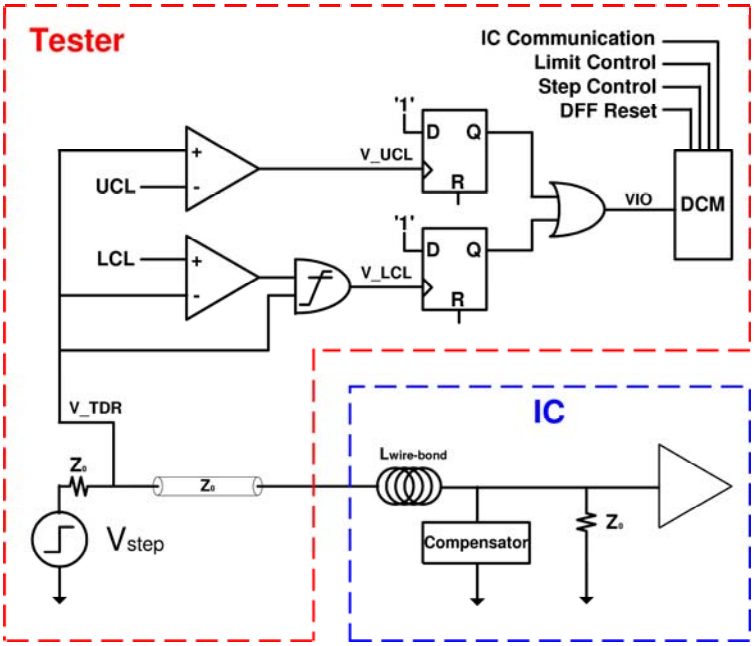


Figure 8.9: Dynamic Compensator Calibration Circuit

To measure the impedance discontinuity from the wire-bond/compensator, the IC tester transmits a voltage step into the IC package. The magnitude of the reflection from the discontinuity is measured in the tester using comparator circuits. The comparator circuits are used instead of a standard A/D converter (as in a true TDR system) to reduce complexity and cost. One comparator monitors for reflections that exceed a user-defined Upper Control Limit (UCL). A second comparator monitors for reflections that exceed a Lower Control Limit (LCL). The programmable voltages (UCL and LCL) are set by a Digital Control Monitor (DCM) in the IC tester.

When a reflection from the wire-bond/compensator element is above/below the violation limits, the comparator(s) will output a glitch signal that indicates a limit violation (V_UCL / V_LCL). The limit violation glitch is fed into the clock input of a D-flip-flop whose data input is tied to a logic 1. The D-flip-flop therefore serves as a trigger element that will switch and remain high when it detects any glitch from the comparators. The D-flip-flop output will remain high until reset by the DCM. The output of the two D-flip-flops are fed into an OR-gate to combine the upper and lower violation signals into one input (VIO) that is monitored by the DCM.

When the DCM detects a violation, it will communicate with the IC under test to change the compensator settings. Once the compensator is adjusted, the D-flip-flops are reset, and another voltage step is launched into the IC package. This process is repeated until the magnitude of the reflections are within the LCL and UCL limits.

The lower control limit violation signal (V_LCL) is qualified using an AND-gate. The purpose of the AND-gate is to prevent glitches on V_LCL when the step voltage is below the LCL due to normal operating conditions such as the beginning of the rising edge of the step. The AND-gate is designed to have a high switch-point such that $V_{switchpoint-AND} >$ LCL. Once the step voltage exceeds the switch-point of the AND-gate, the glitches from the comparator are allowed to pass through to the D-flip-flop. Spurious glitches which may be observed when the step voltage is below the AND gate switch-point are thus filtered out, and the glitches at the output of the AND gate will therefore be solely due to reflections from the package (which violate the LCL).

In the case that the compensator cannot find a setting that meets the reflection control limits, it will relax the UCL and LCL voltages. Since this calibration is only performed on the signal pins of the IC package, the overhead associated with the process is small. The calibration can be sped up further by setting the UCL and LCL to predefined values that are based on the design of the IC and package. If the tester uses reasonable limits to begin with, the convergence of the algorithm will be sped up.

Figure 8.10 shows the signals of the calibration circuit during operation. In this case, the limits are set to indicate violations when reflections exceed 5%. In this figure, a UCL violation is indicated by a glitch on V_UCL. The glitch then triggers the D-flip-flop which in turn sends a static control signal (VIO) to the DCM.

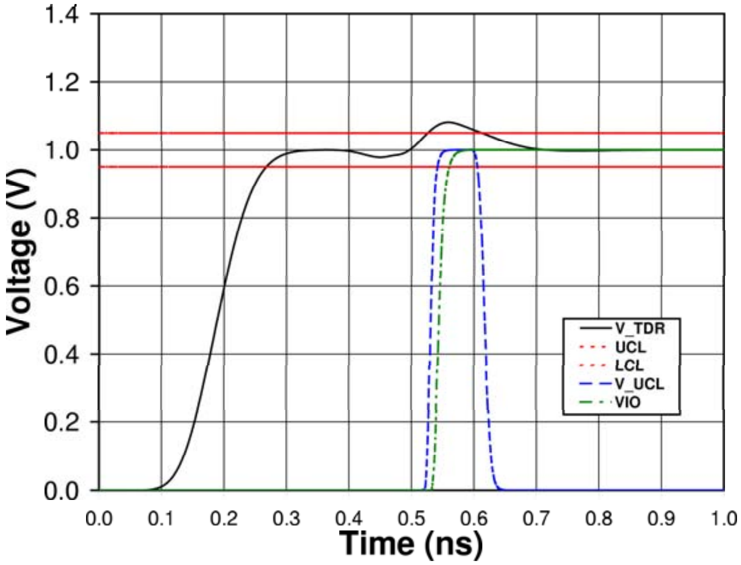


Figure 8.10: Dynamic Compensator Calibration Circuit Operation

# Chapter 9

# Future Trends and Applications

The analytical models and noise reduction techniques presented in Chapters 4 through 8 were analyzed for use with past, present, and future IC packaging in order to predict and improve performance. The experimental results illustrated that the techniques were successful and made significant improvement in the performance of the packaging. While these techniques were demonstrated to have an immediate impact when applied to commonly used VLSI packages, the current trends in IC technology make these techniques even more invaluable. In addition, since all of the modeling and performance techniques were described using a common mathematical framework, the work in this thesis can be easily applied to a wide variety of electronic applications. This chapter presents some of the industry trends and other applications that may benefit from the modeling techniques described in this work.

## 9.1 The Move From ASICs to FPGAs

Application specific integrated circuits (ASICs) have enabled the dramatic increase in computational power that digital systems have enjoyed over the past 20 years; however, within the past 5 years, the cost associated with designing and fabricating an ASIC has increased considerably [92]. This cost stems from the increased complexity of the design process, along with the high cost of manufacturing that accompanies modern fabrication processes. As a consequence, there has been a marked reduction in the number of ASIC design starts per year compared to design starts 10 years ago. At the same time, Field Programmable Gate Arrays (FPGAs) have gained a foothold as one of the most-used building blocks in digital systems. The flexibility of an FPGA allows designers to reduce hardware design cycles times, while adding inherent feature upgradeability in the final product. Traditionally FPGAs have been used solely as a prototyping vehicle due to their higher cost and slower performance compared to ASICs; however, recently FPGAs have experienced a dramatic increase in performance due to the rapid improvement in IC processing technology. FPGA development has kept up with Moore's law, while at the same time providing inherent design flexibility. Figure 9.1 shows that the doubling of transistors in Intel microprocessors has followed the prediction of Moore's Law over the past 25 years [90]. Figure 9.2 shows the recent increase in the amount of logic cells that can be implemented using an FPGA [91]. The rapid growth in FPGA technology illustrates how FPGAs are also tracking Moore's Law prediction.

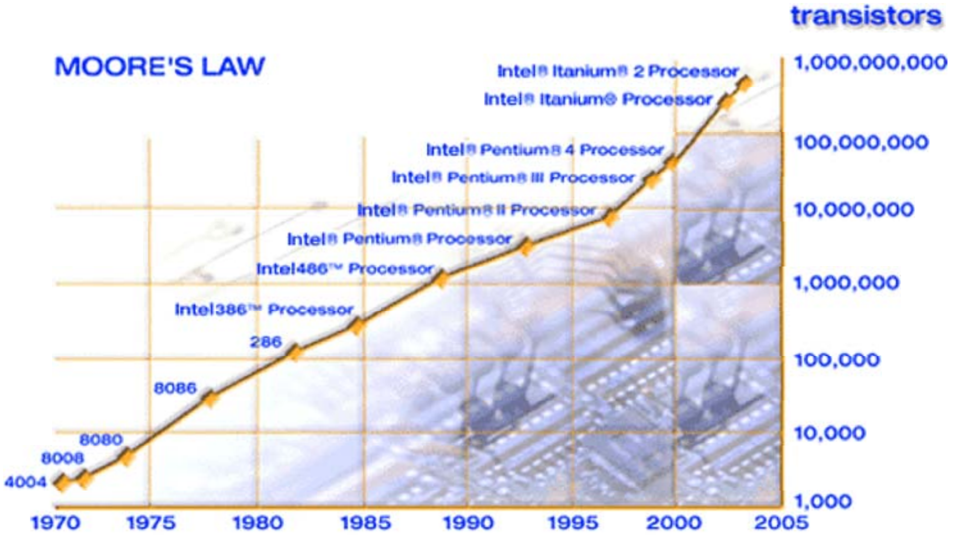


Figure 9.1: Moore's Law Prediction Chart

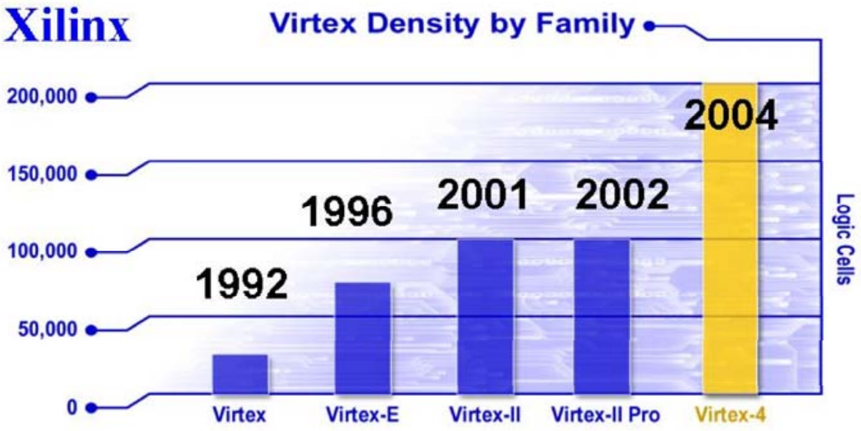


Figure 9.2: Xilinx FPGA Logic Cell Count Evaluation

Combining the increased performance of FPGAs with the dramatic increase in the price of ASICs, FPGAs are now the preferred implementation vehicle for digital systems not requiring extremely high performance. Within the past 5 years the number of digital design starts that use at least one FPGA is four times greater than the number of design starts requiring an ASIC [92]. This trend is important to this work due to the use model of FPGAs. An FPGA is designed for use in a wide variety of applications, which each present different electrical constraints. A single FPGA must operate over various supply voltages, system impedances, bus speeds, and I/O levels. In addition, a particular FPGA circuit design is typically implemented in many styles of packaging to offer a range of performance and cost to the users. This fact means that any technique that can improve system performance yet operate across many styles of packaging and bus configurations will be extremely valuable to the FPGA industry.

All of the techniques presented in this work can be applied to an FPGA. In fact, two of these techniques were prototyped using FPGAs as well. Since the analytical model and bus selection techniques presented in Chapters 4 and 5 were constructed in terms of generic LC models of the package interconnect, they can be directly applied to any style of FPGA as long as the package parasitics are known. This enables system designers to quickly analyze and design off-chip busses as well as understand the impact of moving toward advanced packaging within an FPGA family.

The encoding techniques presented in Chapters 6 and 7 can also be directly applied to FPGAs. Once a given bus configuration is implemented, performance can be further increased by implementing an encoding scheme to avoid the noise induced by bus data traversing the package parasitics. FPGAs lend themselves very well to the encoding methodology since logic can be added after design and fabrication. This allows encoders to be added **during** prototyping if simultaneous switching noise is found to be a problem. In addition, since the encoders can be added at anytime, they present the flexibility to address different noise sources depending on the type of packaging utilized. One FPGA in a system may have an inductive cross-talk noise problem while another FPGA may have an impedance discontinuity issue. Any source of noise that is of concern can be addressed by altering the encoder construction.

Finally, the compensator can also be used to directly improve performance in FPGAs. As mentioned earlier, an FPGA family is usually produced using the same design core, with various packaging, cost, and performance variants. This can be extremely challenging to VLSI designers who are faced with having to deal with multiple package parasitics for a single circuit. An FPGA circuit may exist in a wire bonded package which must be impedance matched to the system impedance. This is difficult to do since the same die may also be placed in a flip-chip package and must be impedance matched to the same system impedance. The dynamic compensator addresses this problem by having a programmable compensation capacitance that can be designed to cover a wide range of package parasitics. The dynamic compensator has the ability to match a wire bond package as well as a flip-chip package to a given system impedance by simply changing the control lines of the capacitive compensation.

All the techniques in this work favorably complement the trend of moving from

ASICs to FPGAs. Since the manufacturing cost of ASICs is not predicted to lower in the foreseeable future, FPGAs will continue to dominate digital design starts. This indicates that the work in this thesis is applicable to a dominant industry trend that may continue for many decades.

## 9.2 IP Cores

Another recent trend in the VLSI industry is the move toward system design through use of *Intellectual Property* (IP) cores. An IP core is a circuit block that is designed and verified such that it can be dropped into a larger design. An IP core differs from standard ASIC designs in that it typically contains large, complex, system level circuitry. Blocks such as DDR memory controllers, PCI interfaces, and PowerPC microprocessors are examples of complex system-level IP cores. IP cores are also gaining acceptance as stand-alone products which are licensed to an end user who is designing an FPGA or ASIC. The move toward the IP core methodology comes from the ever-growing complexity of modern VLSI designs. It is becoming impractical for a single design team to design every block in an ASIC in a reasonable amount of time. As a consequence of time-to-market demands, design teams are turning to IP cores for the more standard portions of the ASIC.

The noise reduction techniques presented in this work lend themselves very elegantly to the IP core design methodology. Since each technique can be implemented using a general purpose block of circuitry (which is relatively independent of the IC design), noise reduction cores can easily be created and incorporated into larger VLSI designs. In fact, these cores can be parameterized, allowing them to be effective for more than one bus or package configuration. Bus encoding/decoding cores can be inserted between the core signals of the IC and package I/O pins. This core insertion would not have any impact on the core circuitry and serves only to improve the speed of the off-chip bus that traverses the package. Integrated circuits that are limited by the package noise can be sped up, so that the core logic on the die may be operated faster as a consequence.

Similarly, the compensator can also be implemented as an IP core. Since the compensator addresses the level 1 interconnect problem, the core circuitry of the die can be designed without the additional constraints that are driven by packaging speed limitations. In addition, since the compensator is shown to have sufficient range to cover a variety of package interconnects, it can be used for designs that go into multiple package technologies.

Figure 9.3 shows a possible scenario in which the techniques presented in this work can be integrated into a larger VLSI design as IP cores. In this example, two styles of off-chip busses are shown. The first is a *PCI* bus, which is implemented using a traditional wider, slower configuration. This style of bus experiences a large amount of supply bounce due to the large number of simultaneously switching signals. For this bus, an encoder is utilized to avoid inductive noise due to data sequences that traverse the package. The second off-chip bus uses a *HyperTransport* core. This bus is implemented using a faster, narrower configuration. In this configuration, the number of signals that simultaneously switch is considerably less than the PCI bus. For this example, the HyperTransport bus will not experience as much supply bounce but will instead be limited by capacitive bandwidth and impedance discontinuity problems. To address the bandwidth limitation, an encoder is utilized to avoid the worst-case bandwidth limiting sequences. To address the impedance discontinuities, a compensator circuit is used to impedance match the package interconnect with the system PCB. Since the noise reduction cores can be implemented in a parameterized manner, variations in the code construction allow the core to be usable in a variety of packaging and bus configurations.

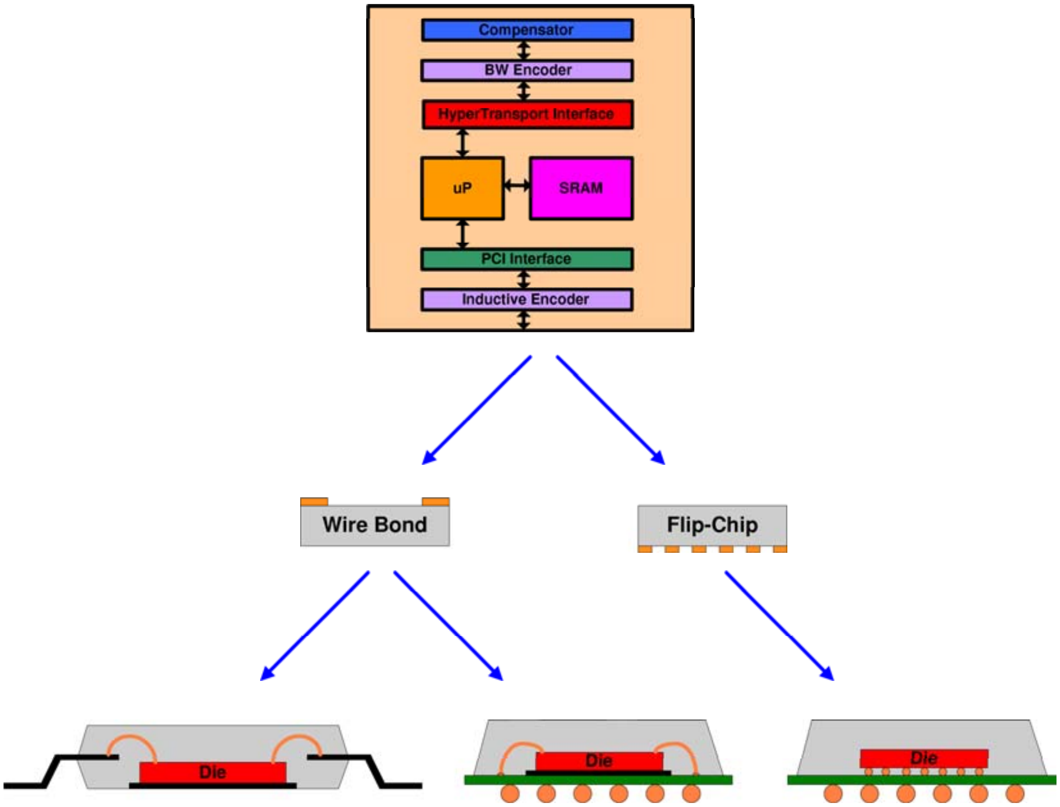


Figure 9.3: IP Core Design Methodology Incorporating Encoder and Compensator

## 9.3 Power Minimization

Another major challenge for VLSI designers in this and the next decade is how to reduce the extreme amount of power that is being consumed as more transistors are integrated on-chip. Advances in IC processing technology enables more transistors to be integrated on a die than can be cooled using modern packaging techniques. Additionally, the ever-diminishing transistor feature sizes enable faster switching times for the core logic. Equation 9.1 expresses the dependence of power consumption on the capacitance ($C_{load}$) being switched, supply voltage ($V_{DD}$), switching frequency ($f_{switch}$), and the activity factor ($\alpha_{switch}$).

$$P_{VLSI} = (C_{load}) \cdot (fswitch) \cdot (V_{DD})^2 \cdot (\alpha_{switch}) \tag{9.1}$$

The switching activity factor, $\alpha_{switch}$, represents the fraction of circuit nodes that switch during any clock cycle. This parameter is typically between 0.2 and 0.35 for modern VLSI designs [67].

The encoders presented in this work were constructed in a general manner so as to eliminate any arbitrary vector sequence from the transmitted data. This methodology can be easily extended to reduce power within the IC. As mentioned earlier, as much as 25% to 50% of the IC power can be consumed in the output drivers [89, 92]. Since the encoders described in this thesis are designed to eliminate vector transitions that result in noise of a magnitude greater than a specified value, these encoders can also be used to avoid data sequences which result in large power consumption. This has the effect of reducing $\alpha_{switch}$ in Equation 9.1, thereby directly reducing power consumption.

A further extension of the encoder can be used to avoid vector sequences in on-chip busses which results in a power consumption above a specified limit. Long on-chip busses suffer not only an increased power consumption due to their typically high switching activity, but also from the fact that the bus capacitance ($C_{load}$) is proportional to bus length. The problem of long on-chip busses consuming considerable power has become more important as more system level blocks are integrated within one die. To incorporate complex system blocks, the die size increases, creating a need for long block-to-block busses. These busses are inherently long therefore consume considerable power, according to Equation 9.1. Implementing encoders that reduce the worst-case power consumption sequences in long on-chip busses will have a direct impact on total power consumption of the IC. In addition, the encoder construction methodology needs minimal modification, with addition constraint equations required to express the user-defined power limits.

Since power is a critical issue in current and future designs, it is often conjectured that power will eventually limit the continued success of Moore's Law. In such a scenario, power reduction techniques like the one described above may play an important part in extending the success of Moore's Law

## 9.4 Connectors and Backplanes

Nearly all digital systems contain connectors which transmit signals and power between various printed circuit boards. Connectors provide both the mechanical and electrical connection between PCBs. Connector construction is similar in nature to that of IC packaging in that the mechanical robustness of the interconnect must be addressed before considering the electrical behavior. Once a stable mechanical structure is designed that can be economically manufactured, then the electrical performance can be evaluated. This results in backplane interconnect that is not optimized for electrical performance. This is the exact situation that exists in IC packaging. Connector leads contain electrical parasitics which degrade the performance of digital signals. Excess inductance and capacitance in the connector leads to voltage drop, capacitive bandwidth limitation, signal-to-signal coupling, and impedance discontinuities.

Since all the techniques presented in this work are constructed to alleviate the detrimental effects of the inductance and capacitance of the interconnect, this naturally makes them applicable to any noise-prone electrical interconnect in a digital system. The analytical modeling, encoding scheme, and compensation methodology presented in this thesis can be applied to any connector without modification. Instead of using the inductance and capacitance parasitics from the IC package, the inductance and capacitance of the connector would be used. From these values, the noise associated with the electrical parasitics of the connector can be directly computed. In addition, the encoding and compensation techniques can be used to bound the maximum amount of connector noise.

The application of the noise reduction techniques to connectors plays well into the *backplane* architecture used throughout industry. The backplane architecture is used widely throughout the computer industry as a way to scale systems. Smaller circuit blocks are implemented which can be plugged into a main PCB to add computational power as needed. Examples of circuit blocks that utilize this architecture are microprocessors, memory, and non-volatile storage elements. The backplane methodology allows the smaller circuit blocks (which are utilized in the larger system) to be designed and verified independent of the larger system. This results in a partitioning of the system functionality and complexity, resulting in reduced design cycle times and more robust system operation. In addition, the smaller circuit blocks can be produced by multiple vendors, which drives down the cost of the design. The advantages of the backplane architecture have resulted in a widespread adoption of this methodology. This architecture has in turn driven the performance of some of the most popular off-chip bus standards. Busses such as PCI Express, Infiniband, and Gigabit Ethernet were designed specifically to improve throughput of backplanes. These busses are predicted to achieve per-pin datarates of 10Gb/s within the next five years [84, 89]. Since the techniques presented in this work apply directly to the connectors used in the backplane methodology, their importance could potentially play a crucial part in the continued increase in performance of backplane-based system.

## 9.5 Internet Fabric

One of the most important developments in the past 30 years has been the internet. This technology has revolutionized communication and sparked one of the longest growth periods for the global economy in recent history. One of the underlying technologies that has enabled the success of the internet is the *internet protocol.* The internet protocol allows communication data to be broken into smaller pieces and transmitted as a series of packets, each of which contains destination address information. The protocol allows data to be sent through multiple infrastructure paths and be reassembled by the receiving computer. Packets may arrive out of order on account of varying delays along different electrical paths. Various alternative infrastructure paths are selected according to the usage on any given path.

Many encoding protocols have been developed over the past decade to address the problem of congestion within the internet infrastructure (or fabric) [28, 30, 31]. These protocols attempt to avoid network congestion by monitoring the sequence of data packets that are being transmitted. In applications such as audio and video, encoding algorithms monitor the data stream for adjacent packets that contain identical information. In an A/V application, sequential data patterns that contain the same information may be eliminated without noticeable distortion in the final reassembled data. These encoding algorithms allow streaming audio and video to be transmitted reliably and efficiently despite congested network fabrics, to be eventually received as a reasonable representation of the original data.

The encoders presented in this work are constructed by creating a directed graph containing transitions that have been deemed *legal* by the constraint equation evaluations. In the package noise constraints within this work, transitions are eliminated from the complete set of encoder transitions if they violate any of the user-defined noise limits for the system; however, the constraint equations can be easily modified to detect sequences of data patterns that can be eliminated in an A/V application when faced with network congestion. This modification would consist of a new series of constraint equations that are written specifically for the A/V application. With the increasing use of streaming audio and video over the internet, the encoding techniques presented in this work could potentially provide significant improvement in the throughput of A/V data over congested network paths.

This chapter presented potential applications of the techniques in this work to current and future industry trends. In each case, the general manner in which the noise reduction techniques are formulated enables their use in a variety of applications. Any system which experiences unwanted noise due to the electrical parasitics of the interconnect or simply desires the elimination of specific transitions can benefit from the techniques presented in this thesis.

# Chapter 10

# Conclusion

This thesis has presented a comprehensive look at the noise problems within IC packaging. Today's integrated circuits are experiencing a dramatic increase in performance due to significant advances in the IC design and fabrication processes. IC technology has followed Moore's Laws for the past 30 years and is expected to continue at this rate sometime into the next decade. At the same time, IC packaging technology has evolved at a much slower pace. This mismatch in performance between the IC and the package is now the leading limitation to system performance. Inter-chip busses that transfer data between ICs within the system need to be slowed so as to avoid unwanted noise from the package. Off-chip communication is now the largest bottleneck in modern digital systems design.

Unwanted noise occurs in the package due to the parasitic inductance and capacitance within the package interconnect. The parasitics of the package can cause supply bounce, signal-to-signal coupling, capacitive bandwidth limiting, and impedance discontinuities. Since the IC packaging was originally developed for robust mechanical performance and manufacturability, the interconnect was not optimized for electrical performance. In addition, the complex manufacturing processes used to create the package make altering the interconnect to improve electrical performance very difficult and expensive. The move toward advanced packaging can reduce the electrical parasitics in the package but is often too expensive for the majority of VLSI designs. As such, any technique that can aid VLSI designers in the prediction of performance and reduction of noise within the package is of great value.

Chapter 4 presented an analytical model to predict the performance of an IC package. This was accomplished by finding the fastest rate of change in current or voltage that could be tolerated without violating any of the user-defined noise limits. The amount of noise in the package is dependant on how large the parasitic inductance and capacitance are within the package interconnect. The electrical rate of change of current/voltage was translated a risetime figure and, in turn, into per-pin datarate and bus throughput. For each of the sources of noise, equations were derived that considered the package parasitics and the user-defined noise limits. To verify the accuracy of the models, SPICE simulations were performed on a test circuit. It was found that the analytical model matched simulated results within 10% for bus segments up to 16 bits. It was also found that for the three packages studied in this work, the inductive supply bounce and inductive signal coupling were the dominant noise sources. The analytical models derived in this chapter provide a method for VLSI designers to quickly analyze different packaging technologies and bus configurations in order to meet their throughput requirements.

Chapter 5 presented a technique to select the most cost-effective bus configuration. Using the analytical models from Chapter 4 and including the per-pin cost of the bus configuration, an algorithm was developed that could easily determine the most cost-effective package selection and configuration. In addition, the metric of Bandwidth-per-Cost was introduced, which quantified the cost-effectiveness of a given bus. Using this metric, the total cost of different bus and package configurations (all of which meet the desired throughput) can be compared efficiently. The algorithm gives VLSI designers a powerful deterministic tool to find the most cost-effective off-chip bus design.

Chapter 6 presented a bus expansion encoding technique that was able to reduce package noise by encoding the data prior to it traversing the package interconnect. The encoder algorithm consisted of a series of constraint equations which were written to model noise violations for arbitrary bus transitions. These equations were evaluated to indicate if a given transition would result in a noise limit violation when traversing the package interconnect. Using the remaining legal transitions, a directed graph was created that was used to construct the encoder. Each vertex within the graph was evaluated in turn to find if its remaining outgoing edges were able to encode the transitions of an $m$-bit bus. The output of the encoder is a mapping of transitions in the original $m$-bit bus to transitions in an expanded $n$-bit bus, such that the transitions in the expanded bus enable data transmission without noise violations.

Since the encoded data introduces less package noise, it can be transmitted at a higher per-pin datarate. This speed increase results in throughput improvement even after accounting for the encoder overhead. Experimental results performed on the encoder illustrated that for a fixed $\frac{di}{dt}$ of $33\frac{MA}{s}$, package noise was reduced as much as 89% for an aggressively encoded bus. It was also shown that for a 3-bit bus which was encoded and evaluated using a varying $\frac{di}{dt}$, the bus throughput was increased up to 46% compared to the original unencoded data. The bus expansion encoding technique presented in this work provides VLSI designers a complete methodology to reduce the noise within the IC package and increase off-chip bus throughput.

Chapter 7 presented a bus stuttering encoder which was also successful in reducing the amount of package noise. This technique was similar to the bus expansion encoder in terms of the creation and evaluation of the constraints and legal transitions of the bus; however, a stuttering algorithm was invoked on the directed graph of legal transitions, which inserted intermediate states between pairs of vertices which resulted in noise limit violations. The stutter states were inserted such that each vertex of the directed graph could transition to any other vertex using only the remaining legal vectors. This encoder technique had the advantage that no additional package pins were needed to encode the $m$-bit bus. This technique did require more area on the IC than the bus expansion encoder, but was deemed to be incremental to existing protocol-based busses that already contain state machines in their output circuitry. Experimental results illustrated that throughput could be increased as much as 225% for a 6-bit, aggressively encoded bus. The bus stuttering encoder presented in this work provides VLSI designers with a technique to improve throughput in their off-chip busses without adding pins to the package.

Chapter 8 presented a compensation technique that was able to match the impedance of the package interconnect to that of the system PCB. This was accomplished by adding additional capacitance near the wire bonds or flip-chips within the package. This has the effect of lowering the typically high interconnect impedance to a level that matches the system PCB impedance. By matching the impedance of the package to the system PCB, reflections are avoided and performance can be increased. Static and dynamic compensation techniques were presented. In the static approach, predefined capacitance was placed on-chip and on the package to surround the inductive level 1 interconnect. The static compensator was able to reduce reflections in a 5mm wire bond from 20% to 5% for a risetime of 117ps. In the dynamic approach, a programmable capacitance was placed on-chip that could be altered after IC fabrication. The dynamic technique allowed the same circuit to compensate for various interconnect inductances which may result from design or process variations in the interconnect. It was demonstrated that the dynamic technique was able to reduce reflections of a 5mm wire bond from 20% to 6% for a risetime of 117ps. The compensators presented in this work offer VLSI designers a simple technique to impedance match the package interconnect to the system PCB without noticeable area utilization within the package or on the die.

Chapter 9 gave an overview of other applications to which the techniques presented in this work are well suited. The move from ASICs to FPGAs was described, along with a discussion of how the noise reduction techniques reported in this work apply directly to the FPGA design methodology. In addition, the utility of implementing the techniques as IP cores was discussed. The application of the encoders to the problem of power reduction within the IC was also presented. Finally, the possible use of these techniques to connectors, backplanes, and the internet fabric was pointed out. While only a subset of applications were presented in Chapter 9, it is believed that the techniques in this thesis can be applied to a wide variety of electronic applications due to their flexible formulation.

With the continued advancement in IC technology, the performance limitation of the package will continue to dominate system performance. While advanced packaging can aid in reducing the noise associated with the parasitics in the package, it is often too expensive for deployment in the majority of VLSI designs. This thesis presented noise analysis and noise reduction techniques that can be directly applied to current and future packaging technologies. With the use of the techniques presented in this thesis, the VLSI community can continue to experience the dramatic increases in the computational power of digital systems.

# Bibliography

[1] M.D. Powell and T.N. Vijaykumar. Pipeline damping: a microarchitectural technique to reduce inductive noise in supply voltage. In Proceedings of the 30th International Symposium on Computer Architecture, pages 72–83, June 2003.

[2] E. Mejia-Motta, F. Sandoval-Ibarra, and J. Santana. Design of cmos buffers using the settling time of the ground bounce voltage as a key parameter. In Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, volume 2, pages 718–721, August 2000.

[3] C.L Chen and B.W Curran. Switching codes for delta-i noise reduction. In IEEE Transactions of the 43rd IEEE Midwest Symposium on Circuits and Systems, volume 45, pages 1017 – 1021, September 1996.

[4] C. Duan and S.P. Khatri. Exploiting crosstalk to speed up on-chip buses. Design Automation and Test in Europe Conference, February 2004.

[5] C. Duan, A Tirumala, and S.P. Khatri. Analysis and avoidance of cross-talk in on-chip buses. IEEE Symposium on High-Performance Interconnects (HOT Interconnects), pages 133–138, August 2001.

[6] B Victor and K Keutze. Bus encoding to prevent crosstalk delay. In Proceedings of the IEEE/ACM International Conference on Computer Aided Design, pages 57–63, San Jose, CA, November 2001.

[7] B.J. LaMeres and S.P. Khatri. Encoding-based minimization of inductive cross-talk for off-chip data transmission. In Proceedings of Design, Automation, and Test in Europe (DATA) Conference, pages 1318–1323, Germany, March 2005.

[8] M.D. Powell and T.N. Vijaykumar. Pipeline muffling and a priori current ramping: architectural techniques to reduce high-frequency inductive noise. In Proceedings of the 2003 International Symposium on Low Power Electronics and Design (ISLPED), pages 223–228, August 2003.

[9] W. El-Essawy and D.H Albonesi. Mitigating inductive noise in smt processors. In Proceedings of the 2004 International Symposium on Low Power Electronics and Design (ISLPED), pages 332–337, August 2004.

[10] M.D. Powell and T.N. Vijaykumar. Exploiting resonant behavior to reduce inductive noise. In Proceedings of the 2003 International Symposium on Computer Architecture, pages 288–299, June 2004.

[11] R. E. Bryant. Graph based algorithms for Boolean function representation. In IEEE Transactions on Computers, volume C-35, pages 677–690, August 1990.

[12] K. S. Brace, R. L. Rudell, and R. E. Bryant. Efficient implementation of a BDD package. In Design Automation Conference (DAC), pages 40–45, June 1990.

[13] N. Hirano, M. Miura, Y. Hiruta, and T. Sudo. Characterization and reduction of simultaneous switching noise for a multilayer package. In Proceedings of the 44th Conference on Electronic Components and Technology, pages 949–956.

[14] B.J. LaMeres and S.P. Khatri. Performance model for inter-chip busses considering bandwidth and cost. In DesignCon, pages 5–WA1, Santa Clara, CA, January 2005.

[15] M. Miura, N. Hirano, Y. Hiruta, and T. Sudo. Electrical characterization and modeling of simultaneous switching noise for leadframe packages. In Proceedings of the 45th Conference on Electronic Components and Technology, pages 857–864, May 1995.

[16] B. Young. Return path inductance in measurements of package inductance matrixes. In IEEE Transactions on Components, Packaging, and Manufacturing Technology, volume 20, pages 50–55, February 1997.

[17] M. Lopez, J.L. Prince, and A.C. Cangellaris. Influence of a floating plane on effective ground plane inductance in multilayer and coplanar packages. In IEEE Transactions on Advanced Packaging, volume 22, pages 182–188, May 1999.

[18] B.J. LaMeres and S.P. Khatri. Performance model for inter-chip conmunication considering inductive cross-talk and cost. In IEEE International Symposium on Circuits and Systems (ISCAS), pages 4130–4133, Kobe, Japan, May 2005.

[19] B.J. LaMeres. Fpga i/o - when to go serial. In IEE Magazine on Electronic Systems and Software, volume 2, pages 14–18, July 2004.

[20] S Khatri, A Mehrotra, R Brayton, A Sangiovanni-Vincentelli, and R Otten. A novel vlsi layout fabric for deep sub-micron applications. 36th Design Automation Conference (DAC-99), pages 491–496, 1999.

[21] V. Adler and E. Friedman. Repeater design to reduce delay and power in resistive interconnect. IEEE Transactions on Circuits and Systems II, 45:607–616, June 1997.

[22] I. Bouras, Y. Liaperdos, and A. Arapoyanni. A high speed low power CMOS clock driver using charge recycling technique. Proceedings of the IEEE Int. Symposium on Circuits and Systems (ISCAS 2000), pages 657–660, May 2000.

[23] E.D. Kyriakis-Bitzaros and S.S. Nikolaidis. Design of low power CMOS drivers based on charge recycling. Proceedings of the IEEE Int. Symposium on Circuits and Systems (ISCAS), pages 1924–1927, May 1997.

[24] X. Wang and W. Porod. A low power charge-recycling CMOS clock driver. Proceedings of the Ninth Great Lakes Symposium on VLSI, pages 238–239, June 1998.

[25] M. Purandare, A. Sung, and S. Khatri. A differential amplifier based technique to reduce delay in long interconnect. International Conference on VLSI Design, 2004.

[26] H. Kawaguchi and T. Sakurai. A reduced clock swing flip-flop (rcff) for 63% power reduction. In IEEE Journal of Solid-State Circuits, volume 33, pages 807–811, 1998.

[27] T. Sakurai. Design challenges for 0.1um and beyond. Proceedings of the Asia South Pacific Design Automation Conference (ASP-DAC), pages 553–558, 2000.

[28] A.A.M. Ibrahim. Statistical rate control for efficient admission control of mpeg-2 vbr video sources. In IEEE Proceedings of the ATM Workshop, pages 26–29, May 1998.

[29] W. Lee and H. Mehrpour. Tsmr - a new statistical model for mpeg-coded video. In 10th IEEE International Conference on Networks, pages 27–30, August 2002.

[30] A.S. Abraham, Ju Wang, and J.C.L. Liu. Bandwidth-aware video encoding with adaptive image scaling. IEEE International Conference on Multimedia and Expo (ICME), pages 157 – 160, June 2004.

[31] A. Sugiura, M. Kamata, and A.T. Hayashi. Mpeg video encoding based on assigning a high information priority to the focused region. Asia-Pacific Conference on Circuits and Systems (APCCAS), pages 545 – 548, October 2002.

[32] S.E. El-Nahas, E.S. Youssef, M. Moussa, and I. Ahmed. Statistical characterization of variable bit rate mpeg-1 encoded video streams. Proceedings of the Sixteenth National Radio Science Conference (NRSC), pages C32/1 – C32/7, February 1999.

[33] Y. Iraqi, R. Boutaba, and R. Dssouli. Statistical properties of mpeg video traffic and their impact on bandwidth allocation in wireless atm networks. IEEE Wireless Communications and Networking Conference (WCNC), pages 998 – 1002, September 1999.

[34] O. Rose. Statistical properties of mpeg video traffic and their impact on traffic modeling in atm systems. Proceedings from the 20th Conference on Local Computer Networks, pages 397 – 406, October 1995.

[35] H. Kim, B.K Sun, and J. Kim. Suppression of ghz range power/ground inductive impedance and simultaneous switching noise using embedded film capacitors in multilayer packages and pcbs. In IEEE Microwave and Wireless Components Letters, volume 14, pages 71–73, February 2004.

[36] B.J. LaMeres and T.S. Kalkur. The effect of ground vias on changing signal layers in a multilayered pcb. In Microwave and Optical Technology Letters, volume 28, pages 257–260, February 2001.

[37] H. Kim, Y. Jeong, J. Park, S. Lee, and J. Hong. Significant reduction of power/ground inductive impedance and simultaneous switching noise by using embedded film capacitor. Transactions on Electrical Performance of Electronic Packaging, pages 129–132, October 2003.

[38] D. Balaraman, J. Choi, V. Patel, P.M. Raj, I.R. Abothu, S. Bhattacharya, L. Wan, M. Swaminathan, and R. Tunimala. Simultaneous switching noise suppression using hydrothermal barium titanate thin film capacitors. Proceedings of Electronic Components and Technology (ECTC), pages 282–288, June 2004.

[39] B.J. LaMeres and S.P. Khatri. Design of a low-power differential repeater using low voltage swing and charge recycling. In DesignCon, pages 3–WP1, Santa Clara, CA, January 2005.

[40] J.M. Hobbs, H. Windlass, V. Sundaram, S. Chun, G.E. White, M. Swaminathan, and R.R. Tummala. Simultaneous switching noise suppression for high speed systems using embedded decoupling. Proceedings of the 51st Electronic Components and Technology Conference, pages 339–343, June 2001.

[41] Jay Pajagopalan and Haris Basit. Optimization of metal-metal comb-capacitors for rf applications. OEA International Inc, 2004.

[42] B.J. LaMeres and T.S. Kalkur. Time domain analysis of a pcb via. In Microwave Journal, volume 43, pages 76–84, November 2001.

[43] H.H. Jhuang and T.W. Huang. Design for electrical performance of wideband multilayer ltcc microstrip-to-stripline transition. In Proceedings of the 6th Electronics Packaging Technology Conference (EPTC), pages 506–509, 2004.

[44] S. Luan, J. Fan, J.L. Knighten, and N.W. Smith. The design of a lumped element impedance-matching network with reduced parasitic effects obtained from numerical modeling. In International Symposium on Electromagnetic Compatibility, volume 3, pages 984–987, 2004.

[45] A.R. Lopez. Review of narrowband impedance matching limitations. In IEEE Magazine on Antennas and Propagation, volume 46, pages 88–90, August 2004.

[46] K.M. Hock. Impedance matching for the multilayer medium - toward a design methodology. In IEEE Transactions on Microwave Theory and Techniques, volume 51, pages 908–914, March 2003.

[47] A.D. Yalcinkaya, S. Jensen, and O. Hansen. High aspect ratio mems capacitor for high frequency impedance matching applications. In Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems (ICECS), volume 2, pages 14–17, December 2003.

[48] M.A.K. Schwan and J. LoCicero. Data rate enhancement of vdsl via termination impedance matching. In The 2002 45th Midwest Symposium on Circuits and Systems (MWSCAS), volume 3, pages 4–7, August 2002.

[49] R. Aparicio and A. Hajimiri. Capacity limits and matching properties of integrated capacitors. In IEEE Journal of Solid-State Circuits, volume 37, pages 384–387, March 2002.

[50] Kuo-Yu Chou and Ming-Jer Chen. Active circuits under wire bonding i/o pads in 0.13 m eight-level cu metal, fsg low-k inter-metal dielectric cmos technology. IEEE Electron Device Letters, pages 466–468, October 2001.

[51] J.H. Ahm, K.T. Lee, M.K. Jung, Y.J. Lee, B.J. Oh, S.H. Liu, Y.H. Kim, Y.W. Kim, and K.P. Suh. Integration of mim capacitors with low-k/cu process for 90 nm analog circuit applications. Proceedings of the IEEE International Interconnect Technology Conference, pages 183–185, June 2003.

[52] J. Prasad, M. Anser, and M. Thomason. Electrical characterization of dielectrics (oxide, nitride, oxy-nitride) for use in mim capacitors for mixed signal applications. International Semiconductor Device Research Symposium, pages 326–327, December 2003.

[53] A.C.W Lu, W. Fan, L. Wai, C.K. Wang, and H.G. Low. Design optimization of wire bonding for advanced packaging. Proceedings of the 53rd Electronic Components and Technology Conference, pages 1364–1372, May 2003.

[54] C. Mattei and A.P. Agrawal. Electrical characterization of bga packages. Proceedings of the 47th Electronic Components and Technology Conference, pages 1087–1093, May 1997.

[55] B.J. LaMeres and S.P. Khatri. Broadband impedance matching for inductive interconnect in vlsi packages. In IEEE International Conference on Computer Design (ICCD), San Jose, CA, October 2005.

[56] M. Horowitz, C. Yang, and S. Sidiropoulos. High-speed electrical signaling: Overview and limitations. In IEEE Micro., volume 18, pages 12–24, January 1998.

[57] M.E. Lee, W.J. Dally, and P. Chiang. Low-power area-efficient high-speed i/o circuit techniques. In IEEE Journal of Solid-State Circuits, volume 35, pages 1591–1599, November 2000.

[58] B. Casper. An accurate and efficient analysis method for multi gb/s chip-to-chip signaling schemes. Symposium of VLSI Circuits Digest of Technical Papers, pages 54–57, June 2002.

[59] B. Casper, A. Martin, J.E. Jaussi, J. Kennedy, and R. Mooney. An 8-gb/s simultaneous bidirectional link with on-die waveform capture. In IEEE Journal of Solid-State Circuits, volume 38, pages 2111–2120, December 2003.

[60] P.K. Hanumolu, B. Casper, and R Mooney. Analysis of pll clock jitter in high-speed serial links. In IEEE Transactions On Circuits and Systems II, volume 50, pages 879–886, November 2003.

[61] C. Kinnaird. Standards are key to optimizing high-speed data bus communications. Planet Analog (planetanalog.com), October 2002.

[62] W.H. Dally and J. Poulton. Transmitter equalization for 4-gbps signaling. In IEEE Micro, volume 17, pages 48–56, February 1997.

[63] R.R. Tummalo. Fundamentals of Microsystem Packaging. McGraw-Hill, 2001.

[64] W. Dally and J. Poulton. Digital Systems Engineering. Cambridge University Press, Cambridge, U.K., 1998.

[65] H. Johnson and M. Graham. High-Speed Signal Propagation. Prentice Hall PTR, 2003.

[66] H. Johnson and M. Graham. High-Speed Digital Design. Prentice Hall PTR, 2003.

[67] S. Kang and Y. Lebledici. CMOS Digital Integrated Circuits, 2nd edition. McGraw-Hill Companies, 1999.

[68] S. Palnitkar. Verilog HDL - A Guide to Digital Design and Synthesis. SunSoft Press, 1996.

[69] M.D. Ciletti. Modeling, Synthesis, and Rapid Prototyping with the Verilog HDL. Prentice Hall, 1999.

[70] F.T. Ulaby. Fundamentals of Applied Electromagnetics. Prentice Hall, 2002.

[71] T.H. Lee. The Design of CMOS Radio Frequency Integrated Circuits. Cambridge University Press, 2000.

[72] D.A. Johns and K. Matrin. Analog Integrated Circuit Design. John Wiley and Sons, 1997.

[73] M.E. Valkenburg. Analog Filter Design. Oxford University Press, 1982.

[74] R.N. Bracewell. The Fourier Transform and its Applications. McGraw-Hill, 2000.

[75] Agilent Technologies Inc. Advanced design systems user's manual. `www.agilent.com/ads`, 2000.

[76] L. Nagel. Spice: A computer program to simulate computer circuits.

[77] Bsim3 homepage. `www-device.eecs.berkeley.edu/bsim3`.

[78] Berkeley predictive technology modeling homepage. `www-device.eecs.berkeley.edu/bptm/`.

[79] ASAT Inc. Peak performance enhanced lead packages. Technical report, `www.asat.com`, 2004.

[80] ASAT Inc. Peak performance array packages. Technical report, `www.asat.com`, 2004.

[81] ASAT Inc. Peak performance flip chip packages. Technical report, `www.asat.com`, 2004.

[82] Intel Inc. Intel itanium 2 processor. Technical report, `www.intel.com`, 2003.

[83] Rapid IO Trade Association. Technical report, `www.rapidio.org/`, 2004.

[84] PCI-SIG Trade Organization. Technical report, `www.pcisig.com/`, 2004.

[85] HyperTransport Consortium. Technical report, `www.hypertransport.org`, 2005.

[86] SEMICON Far East. Technical report, `www.semiconfarest.com`, 2005.

[87] Actel Inc. Simultaneous switching noise and signal integrity. Technical report, `www.actel.com`, 2004.

[88] 1958 : The invention of the integrated circuit. Technical report, `www.pcb.org`, 2002.

[89] The international technology roadmap for semiconductors (itrs). Technical report, `public.itrs.net`, 2003.

[90] Intel. Prediction to reality. Technical report, `www.intel.com`, 2005.

[91] Xilinx. Virtex fpga family specifications. Technical report, `www.xilinx.com`, 2005.

[92] Agilent Packaging Group. Personal Communication, 2004. Ft. Collins, CO.

[93] Avant! Raphael : Interconnect analysis program user's guide. `www.avant!.com/raphael`, 2000.