



**HAL**  
open science

# Encoding-Based Minimization of Inductive Cross-Talk for Off-Chip Data Transmission

Brock J. Lamerer, Sunil P. Khatri

► **To cite this version:**

Brock J. Lamerer, Sunil P. Khatri. Encoding-Based Minimization of Inductive Cross-Talk for Off-Chip Data Transmission. DATE'05, Mar 2005, Munich, Germany. pp.1318-1323. <hal-00181309>

**HAL Id: hal-00181309**

**<https://hal.science/hal-00181309v1>**

Submitted on 23 Oct 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Encoding-based Minimization of Inductive Cross-talk for Off-chip Data Transmission

Brock J. LaMeres  
University of Colorado  
Dept. of Electrical and Computer Engineering  
Boulder, CO 80309  
brock\_lameres@agilent.com

Sunil P. Khatri  
Texas A&M University  
Dept. of Electrical Engineering  
College Station, TX 77843  
sunil@ee.tamu.edu

## Abstract

Inductive cross-talk within IC packaging is becoming a significant bottleneck in high-speed inter-chip communication. The parasitic inductance within IC packaging causes bounce on the power supply pins in addition to glitches and rise-time degradation on the signal pins. Until recently, the parasitic inductance problem was addressed by aggressive package design. In this work we present a technique to encode the off-chip data transmission to limit bounce on the supplies and reduce inductive signal coupling due to transitions on neighboring signal lines. Both these performance limiting factors are modeled in a common mathematical framework. Our experimental results show that the proposed encoding based techniques result in reduced supply bounce and signal degradation due to inductive cross-talk, closely matching the theoretical predictions. We demonstrate that the overall bandwidth of a bus actually increases by 85% using our technique, even after accounting for the encoding overhead. The asymptotic bus size overhead is between 30% and 50%, depending on how stringent the user-specified inductive cross-talk parameters are.

## 1 Introduction

Advances in VLSI fabrication technologies have led to a dramatic increase in the on-chip performance of integrated circuits. The increase in IC performance is predicted by the International Technology Roadmap for Semiconductors (ITRS) [1] to continue doubling every 18 months, following Moore's Law, for at least the next several years [2]. However, 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. The main limitation of the package performance is the parasitic inductance present in the level 1 (from IC die to package) and level 2 (from package to board) interconnects [3, 4, 5]. The inductance factors that affect signal speed and integrity are as follows:

- Supply bounce. Typically supply ( $V_{SS}$  and  $V_{DD}$ ) pins are interspersed at regular intervals between signal pins. Every  $n^{th}$  pin is a  $V_{SS}$  or  $V_{DD}$ . The supply bounce is proportional to the number of pins switching low or high. Ground bounce is expressed as:

$$V_{bnc} = L \sum_i \left( \frac{di}{dt} \right) \quad (1)$$

Where  $L$  is the self-inductance of the  $V_{SS}$  pin, and  $\sum_i \left( \frac{di}{dt} \right)$  is evaluated over the number of signal pins switching low. Since the placement of power and signal pins is regular, we can compute this quantity as half the number of signal pins switching low to the immediate right of the  $V_{SS}$  pin and half the number of signal pins switching low to the immediate left of the  $V_{SS}$  pin. Since each signal always has a  $V_{SS}$  pin to the left and to the right, we assume that if it switches high, then half the switching current is supplied by the  $V_{SS}$  pin to its left, and the other half by the  $V_{SS}$  pin to its right.

In a similar manner, a supply voltage droop is encountered on  $V_{DD}$  pins as well.

- Glitching. If a signal pin  $j$  is static, then a glitch may be induced in its voltage due to neighboring pins which switch. This is governed by the expression

$$V_{glitch}^j = \sum_k \pm (M_{jk} \frac{di_k}{dt}) \quad (2)$$

where  $i_k$  is the current in the  $k^{th}$  pin, and  $M_{jk}$  is the mutual inductance between the  $j^{th}$  pin being considered and the  $k^{th}$  pin. The sign of the coupled voltage is positive or negative depending on whether the  $k^{th}$  neighboring pin undergoes a rising or falling transition.

- Switching speed. When a signal is switching, its transition can be sped up if the coupled voltage induced by its neighbors' mutual inductance aids the transition. We would like that a signal is not slowed down (i.e. either sped up, or unhindered) in its transitions due to this effect. We would like that when a signal  $j$  is rising (falling), the coupled voltage on this signal (Equation 2) due to its neighbors' transitions is zero or positive (negative). In this way, the transitions of signals are not slowed down due to inductive cross-talk.

The traditional approach to reducing the parasitic inductance within the package has been through aggressive package design. We are currently seeing success in the application of chip-scale and flip-chip technologies in level 1 interconnect for high-end applications. While such technologies decrease the above

mentioned inductive effects, they are still relatively expensive for the majority of ICs. Further, they do not completely eliminate the inductive problems. Level 2 interconnect has been improved by moving toward surface mount and grid array style packaging. While these technologies are becoming affordable due to process improvements, they do not completely eliminate the inductance problem. While aggressive package design assists in the problem, it is a slow and expensive process to develop new packages. In this paper, we present a technique to avoid the inductive cross-talk in the interconnect by encoding the data being transmitted off-chip. We construct a set of equations which encode the constraints that any legal vector sequence must satisfy to avoid supply bounce, signal glitching, and signal edge speed degradation. The degree of supply bounce, glitching and edge speed degradation that can be tolerated are expressed by means of user-specified parameters. From this set of equations, we construct a set of legal vector sequences for the bus. We use this set to find the largest effective size of the bus that can be achieved by encoding, for a given physical size of the bus.

We show that the inter-chip bus throughput is increased as much as 85% by using our encoding techniques. The bus size overhead is as low as 20%, and can be reduced further by using less aggressive user-specified inductive cross-talk constraints. This compares very favorably with the 100% overhead associated with differential signaling.

The rest of this paper is organized as follows. Section 2 provides the definitions used in the rest of this paper. Section 3 describes previous work on this topic. Section 4 presents our encoding scheme to reduce inductive cross-talk. Experimental results are presented in Section 5, and conclusions are drawn in Section 6.

## 2 Preliminaries and Terminology

Consider  $k$  segments of  $n$  bus bits, with the  $j^{\text{th}}$  segment consisting of signals  $b_0^j, b_1^j, b_2^j \dots b_{n-1}^j$ . Let the vector sequence on segment  $j$  be denoted as  $v^j$ .

For example, if we had a  $V_{SS}$  and  $V_{DD}$  pin repeating after every 4 signal pins, the segments would consist of 6 pins. If the bus consisted of 20 signal pins, then we would implement it using 5 such segments.

- **Definition 1 :** A Vector Sequence  $v^j$  is an assignment of values to the signals  $b_i^j$  as follows:

$$b_i^j = v_i^j, \text{ (where } 0 \leq i \leq n-1 \text{ and } v_i^j \in \{0, 1, -1\} \text{).}$$

Note that  $v_i^j = 1(-1)$  indicates that the  $i^{\text{th}}$  signal of the  $j^{\text{th}}$  bus segment is rising (falling), while  $v_i = 0$  indicates that it is either statically low or high.

- **Definition 2 :** A Legal Vector Sequence (modulo inductive cross-talk)  $v$  is an assignment to the signals  $b_i$  such that:

- If  $b_i$  is a supply pin, the total bounce on this pin is bounded by  $P_{bnc}$  volts, where  $P_{bnc}$  is a user-specified constant.
- if  $b_i$  is a signal pin which is static during the vector sequence, the glitch on this pin has a magnitude bounded by  $P_0$  volts, where  $P_0$  is a user-specified constant.
- if  $b_i$  is a signal pin which is switching during the vector

sequence, the switching speed of this pin is **not degraded** due to the effect of inductive cross-talk. Note that we can make this restriction stricter – by specifying that  $b_i$ 's transition is in fact *speed up* due to inductive cross-talk.

## 3 Previous Work

There has been much work into the reduction of parasitic inductance through package advancement [3, 6, 5]. Since the performance limitation is caused by the parasitic inductance in the level 1 and level 2 interconnects of the IC package, many packaging technologies have been developed. Table 1 shows the parasitic inductance values for three industry standard packages (a Quad FlatPak (QFP) with wirebonding, a Ball Grid Array (BGA) with wirebonding, and a flip-chip BGA package). In this table,  $L_{self}$  is the self-inductance of a pin, and the columns to its right are the mutual inductive coupling coefficients of successive neighbors of this pin.

Package	$L_{self}$	$K_1$	$K_2$	$K_3$	$K_4$	$K_5$
QFP-wb	4.550nH	0.744	0.477	0.352	0.283	0.263
BGA-wb	3.766nH	0.537	0.169	0.123	0.097	0.078
BGA-fc	1.244nH	0.630	0.287	0.230	0.200	0.175

Table 1: Self and Mutual Inductance Values for Modern Packages

Bus encoding algorithms have been developed to overcome the capacitive cross-talk for on-chip busses [7, 8, 9]. However, the problem of on-chip capacitive cross-talk minimization for busses is very different from that of off-chip inductive cross-talk minimization. Although our approach also constructs (inductive) cross-talk resistant CODECs algorithmically, in contrast to [7, 8], we utilize memory-based CODEC solutions.

Techniques have been presented to minimize the inductive problems due to packaging. Pipeline damping was presented in [10]. In this approach, the authors attempt to minimize peak current levels by using a multi-valued output driver. While this approach improves performance by reducing the inductive ringing, it requires complex circuitry to implement the multi-valued output driver. CODECs have also been presented [11] that limit the total number of simultaneously switching signals with the same transition direction. This has the effect of reducing the power supply bounce by limiting the total amount of current flowing through the power supply pins at any given time. This technique reported performance improvements but only considered the supply bounce and not the signal-to-signal cross-talk. Our work improves upon previous techniques by additionally considering signal rise-time degradation and glitching due to inductive cross-talk. In our approach, all the inductive effects are captured in a common mathematical framework.

## 4 Our Approach

Consider a bus consisting of  $k$  identical segments, each of width  $n$ . For any segment  $j$ , let  $j-1$  represent the segment to the immediate left of  $j$ , and let  $j+1$  represent the segment to its immediate right. Let us also denote the values of the  $n$  bits of segment  $j$  as  $v_i^j$  ( $0 \leq i \leq n-1$ ). Figure 1 shows an example of a bus configuration with  $k=3$  and

$n = 5$ . The signal-to-power ratio for this bus configuration is  $\alpha = \frac{\# \text{ of pins in each segment}}{\# \text{ of supply pins in each segment}} = \frac{5}{2}$ .

In general, when assigning package pins for an off-chip bus,  $V_{DD}$  and  $V_{SS}$  pins are interspersed among the signal pins in a regular fashion. The overall bus arrangement consists of a repetitive pattern of segments, each with their  $V_{DD}$  and  $V_{SS}$  pins in the same relative position within the segment (as shown in Figure 1).

In our approach, we write equations to encode the inductive cross-talk constraints for all bits of the  $j^{\text{th}}$  bus segment. The constraints are different for signal,  $V_{DD}$ , and  $V_{SS}$  pins. Depending on the number of neighboring pins whose mutual inductance effects we want to model, the constraint equations will include pins belonging to neighboring segments as well. Since the segments are arranged in a repetitive manner, the encoding obtained for any segment will be valid for all  $k$  segments within the bus.

Having written these constraints, we then determine the vector sequences which satisfy these constraints. The valid sequences are used to construct a directed graph which encodes legal transitions between bus vectors. From this digraph, we construct a memory-based CODEC which is used during the bus data transfer.

#### 4.1 Signal Pin Constraints

Consider the coupled voltage on a pin  $i$  (in bus segment  $j$ ), due to a transition on its neighbor  $p$  (which is  $q$  pins away from  $i$ , and called the  $q^{\text{th}}$  neighbor of  $i$ ). This voltage is expressed as  $v_i = \pm M_{ip} \frac{di_p}{dt}$ . The sign of the coupled voltage depends on the direction of the transition on the  $q^{\text{th}}$  neighbor  $p$ . Since output drivers in a bus all have the same drive strength (i.e.  $\frac{di_p}{dt} = \frac{di_q}{dt}$  for any pair of bus signal pins  $p$  and  $q$ ), let  $k_q = |M_{ip} \frac{di_p}{dt}|$ . As a result, we can write  $v_i = k_q \cdot v_{i+p}^j$ , where  $v_{i+p}^j \in \{-1, 0, 1\}$  as per Definition 1. Also, the arithmetic in the subscript of  $v_{i+p}^j$  is performed modulo  $n$ . For example, if  $n = 5$ ,  $j = 4$ , and  $i = 0$ , then  $v_{i-3}^j$  is the same as  $v_2^j$  (i.e. the second bit of the adjacent bus segment to the left). Using this notation allows us to write the inductive cross-talk constraints very compactly.

We can write the mutual inductive coupling of any signal pin to its immediate neighbor signal pin as  $k_1$ . Further, let the mutual inductive coupling of a signal pin to its neighbor's neighbor be expressed as  $k_2$  (likewise  $k_3, k_4$ , etc.). We assume that  $k_x = 0$  for  $x > p$ . In other words, if  $p = 3$ , then we ignore the inductive cross-talk due to the  $4^{\text{th}}$  neighbor and beyond, by setting  $k_4 = k_5 = \dots = k_n = 0$ . As a consequence, we include the mutual inductive contributions of three neighboring pins on either side of the pin under consideration. The  $k_i$  labels in Figure 1 illustrate the mutual inductive signal coupling for  $p = 3$ . Note that each signal pin within the bus will experience coupling from pins on either side. This symmetry allows for encoding to reduce or cancel out the net mutual inductive effect experienced on a victim signal. For this work, any  $K_j$  value less than 0.15 is ignored, and the corresponding  $k_j$  values are set to 0.

The polarity of the mutual inductive coupling on the victim signal will depend on whether the neighboring signals are rising ( $v_i^j = 1$ ) or falling ( $v_i^j = -1$ ). Constraints for the victim signal

are written for all three possible transitions, those being rising ( $v_i^j = 1$ ), falling ( $v_i^j = -1$ ), or static ( $v_i^j = 0$ ). Using the notation described above, a constraint equation can be written for each victim signal, to limit the mutual inductive coupling effect. The inductive cross-talk requirements for a signal pin  $i$  in segment  $j$  are expressed below.

- If signal  $i$  rises in segment  $j$ , then the cumulative inductive cross-talk on this signal should not deter (or should aid) its transition by inducing a mutually coupled voltage which is greater than or equal to a user-specified quantity  $P_1$ :

$$v_i^j = 1 \Rightarrow$$

$$k_1 \cdot (v_{i-1}^j + v_{i+1}^j) + k_2 \cdot (v_{i-2}^j + v_{i+2}^j) + \dots + k_p \cdot (v_{i-p}^j + v_{i+p}^j) \geq P_1$$

Note that  $P_1$  has units of voltage and represents the minimum amount of inductive signal coupling allowed for the pin  $i$  in segment  $j$ . If  $P_1 = 0$  and the inequality in the above expression is changed to an equality, then all the mutual inductive cross-talk is canceled out (i.e.  $v_{i-1}^j = -v_{i+1}^j$ , etc.). If we wish to speed up the transition of pin  $i$  in segment  $j$ , then we simply set  $P_1 > 0$ . This would force the mutually induced voltage on pin  $i$  of segment  $j$  to speed up its rising transition.

Also note that by definition  $v_i^j$  for any supply pin is 0. This eliminates any mutual induced voltage on a victim signal pin  $i$ , due to  $V_{SS}$  and  $V_{DD}$  pins, as required. Likewise, any signal pin which remains static will also have  $v_i^j = 0$  and hence will not cause in any mutually induced voltage on any neighboring victim pins.

- If signal  $i$  falls in segment  $j$ , then the cumulative inductive cross-talk on this signal should not deter (or should aid) its transition by inducing a mutually coupled inductive voltage which is less than or equal to a user-specified quantity  $P_{-1}$ :

$$v_i^j = -1 \Rightarrow$$

$$k_1 \cdot (v_{i-1}^j + v_{i+1}^j) + k_2 \cdot (v_{i-2}^j + v_{i+2}^j) + \dots + k_p \cdot (v_{i-p}^j + v_{i+p}^j) \leq P_{-1}$$

Again,  $P_{-1}$  has units of voltage, and  $P_{-1} \leq 0$ . Note that for symmetric rise and fall times we set  $|P_1| = |P_{-1}|$ . However,  $|P_1|$  and  $|P_{-1}|$  can be set to different values, to aid in only a rising or falling transition. In this way, the designer could compensate for differences in the rise and fall times of off-chip drivers.

- If signal  $i$  is static in segment  $j$ , then the cumulative inductive cross-talk on this signal should not result in a glitch greater than  $P_0$ .

$$v_i^j = 0 \Rightarrow$$

$$-P_0 \leq k_1 \cdot (v_{i-1}^j + v_{i+1}^j) + k_2 \cdot (v_{i-2}^j + v_{i+2}^j) + \dots + k_p \cdot (v_{i-p}^j + v_{i+p}^j) \leq P_0$$

Again,  $P_0$  has units of voltage, just like  $P_1$  and  $P_{-1}$ .

#### 4.2 Power Pin Constraints

If a pin  $i$  in segment  $j$  is a  $V_{SS}$  ( $V_{DD}$ ) pin, we require that the bounce due to its self inductance be limited by  $P_{bnc}$ , the absolute bounce (droop) voltage that can be tolerated.  $P_{bnc}$  is a user-specified quantity.

Let  $z = |L \frac{di}{dt}|$  in Equation 1. Note that since all output drivers of the bus are identically sized,  $\frac{di}{dt}$  is identical for all drivers. Using this notation, we can write the constraint equation for  $V_{DD}$  and  $V_{SS}$  pins as follows:

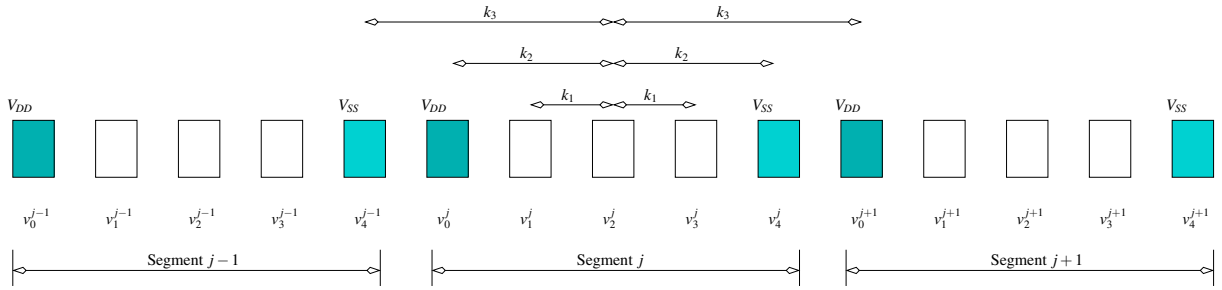


Figure 1: Example Bus Configuration

- If signal  $i$  is  $V_{DD}$  in segment  $j$ , then the cumulative supply bounce should be less than  $P_{bnc}$ .

$$v_i^j = V_{DD} \Rightarrow \frac{\xi}{2} \cdot (\# \text{ of } v_i^j \text{ and } v_i^{j-1} \text{ pins that are } 1) \leq P_{bnc}$$

Note that this assumes that any  $V_{DD}$  pin supplies switching current for half the signal pins in its segment  $j$ , and half the signal pins in the segment to its left. Since each signal always has a  $V_{DD}$  pin to the left and to the right, we assume that if it switches high, then half the switching current is supplied by the  $V_{DD}$  pin to its left, and the other half by the  $V_{DD}$  pin to its right. This explains the presence of the  $\frac{\xi}{2}$  term in the constraint equation above.

- If signal  $i$  is  $V_{SS}$  in segment  $j$ , then the cumulative ground bounce should be less than  $P_{bnc}$ .

$$v_i^j = V_{SS} \Rightarrow \frac{\xi}{2} \cdot (\# \text{ of } v_i^j \text{ and } v_i^{j-1} \text{ pins that are } -1) \leq P_{bnc}$$

It should be noted that the constraints for supply pins are solved to find the maximum number of signals that are allowed to transition in the same direction at once.

Once the configuration of  $V_{DD}$ ,  $V_{SS}$  and signal pins is known for the bus, the above constraints can be greatly simplified. For example, in Figure 1, setting  $v_0^{j-1} = v_4^{j-1} = v_0^j = v_4^j = v_0^{j+1} = v_4^{j+1} = 0$  would encode the supply constraints. In this manner, a single mathematical framework encodes all the required inductive cross-talk constraints, which are i) that switching signals should not have their slew-rates degraded, ii) that the glitch magnitude on static signal pins should be limited, and iii) the bounce on  $V_{DD}$  and  $V_{SS}$  pins should be bounded.

### 4.3 Constructing Legal Vector Sequences

Consider a particular bus configuration ( $n$ ,  $k$ , and  $\alpha$ ) and user-specified inductive cross-talk constraints ( $P_1$ ,  $P_{-1}$ ,  $P_0$ ,  $P_{bnc}$ , and  $p$ ). For each signal pin  $i$  within the segment  $j$ , three constraints equations are written (for  $v_i^j = 1, -1$ , and  $0$ , per Section 4.1). For each power supply pin, one constraint expression is written, per Section 4.2. This results in a total of  $3n - 4$  constraint equations for an  $n$ -bit bus segment. These equations may refer to  $v_i^j$  values from neighboring bus segments as well.

Each possible vector sequence is evaluated for legality by testing if it satisfies each of the  $3n - 4$  constraint equations. The total number of signal pins that need to be considered depends on  $p$ . Since the  $v_i^j$  values for  $V_{DD}$  and  $V_{SS}$  pins are always zero, the number of evaluations is significantly reduced. Since there are three possible signal transitions ( $v_i^j = 1, -1$ , and  $0$ ) per signal bit, the total number of vector sequences that need to be tested for legality is  $3^{(n+2 \cdot p - 6)}$ . Note that the values of  $n$  and

$p$  for realistic busses is small, so these tests (which need to be done exactly once for a design) can be performed easily. In our experiments,  $n = 5$  and  $p = 2$ , which is reasonable for real-life busses.

After testing the vector sequences for legality modulo inductive cross-talk, we create a set of legal vector sequences for the segment  $j$ . The size of this subset depends on how aggressively the parameters  $P_1$ ,  $P_{-1}$ ,  $P_0$  and  $P_{bnc}$  are selected. The final list of legal vector sequences refers to  $n + 2p - 6$  signal pins ( $n - 2$  pins within the segment being considered, and  $2p - 4$  pins on either side of the segment under consideration).

### 4.4 Constructing the CODEC

From the set of legal vector sequences, we next create a directed graph  $G(V, E)$ , of legal bus transitions. We next find the effective size  $n$  of the bus that can be encoded using the transitions in  $G$ .

Note that for a vector sequence  $v^j$ , we can construct a directed edge in  $G$  between vectors  $w_{from}^j$  and  $w_{to}^j$  (which are vertices of  $G$ ). The end-points of this edge ( $w_{from}^j$  and  $w_{to}^j$ ) can be constructed given  $v^j$ , as follows:

$w_{from,i}^j = 1$  if  $v_i^j = -1$  (i.e. the signal is falling) or if  $v_i^j = 0$  (i.e. the signal is static).

$w_{from,i}^j = 0$  if  $v_i^j = 1$  (i.e. the signal is rising) or if  $v_i^j = 0$  (i.e. the signal is static).

Similarly, we can write

$w_{to,i}^j = 0$  if  $v_i^j = -1$  or if  $v_i^j = 0$ .

$w_{to,i}^j = 1$  if  $v_i^j = 1$  or if  $v_i^j = 0$ .

A directed edge between vertices  $w_{from}^j$  and  $w_{to}^j$  in  $G$  indicates the legality (from an inductive cross-talk viewpoint) of the transition from vector  $w_{from}^j$  to  $w_{to}^j$ . Therefore, given a set of vector sequences  $\{v^j\}$  which are legal from an inductive cross-talk standpoint, we can construct a directed graph whose vertices are vectors in  $B^n$ , and whose edges indicate a legal transition (from an inductive cross-talk viewpoint) between the source and sink vectors of the edge.

If an  $m$ -bit bus can be encoded using the legal transitions in  $G$ , then there must exist a set of vertices  $V_c \subseteq V$  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 above 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  $G(V, E)$ . We first find the out-degrees (self-edges are counted) of each  $v \in V$ . For each vertex  $v \in V$ , if the out-degree of  $v$  is less than  $2^m$ , we assign  $V \leftarrow V \setminus v$  (i.e. we delete  $v$ ) and delete all out-going edges rooted at  $v$ , as well as all in-coming edges incident on  $v$ . Given the updated digraph  $G$ , we repeat these steps until convergence. If, after convergence, the cardinality of  $V$  is greater than  $2^m$ , we can construct a memory-based encoder using the legal transitions of  $G$ . The effective bus size that can be encoded in this case is  $m$ .

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  $G(V, E)$  can encode an  $n$ -bit bus

---

```

test_encoder(m, G(V, E))
find out-degree of each node v in V
degrees_changed = 1
while degrees_changed do
degrees_changed = 0
for each v in V do
if out-degree(v) < 2^m then
V ← V \ v
E ← E \ out-edges(v)
E ← E \ in-edges(v)
degrees_changed = 1
end if
end for
end while

```

---

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). The experimental results we report next consider a typical bus segment ( $n = 5$ ,  $k = 3$ ). This segment could be part of a much larger bus, and the analysis would be valid for all segments of the bus.

## 5 Experimental Results

To validate the technique presented, we encoded an example bus configuration to avoid inductive cross-talk. The bus configuration is shown in Figure 1. We used the BGA-wb electrical parameters from Table 1. This bus was encoded using two sets of constraints – *aggressive* ( $P_0$ ,  $P_1$ ,  $P_{-1}$  and  $P_{bnc}$  set to 5% of  $V_{DD}$ ) and *non-aggressive* ( $P_0$ ,  $P_1$ ,  $P_{-1}$  and  $P_{bnc}$  set to 10% of  $V_{DD}$ ).

The first step consists of writing the constraint equations for every pin in the bus. In this bus,  $n = 5$ ,  $k = 3$ , and  $\alpha = 5/2$ . For the inductive coupling values in Table 1, we set  $p = 2$  to ignore inductive coupling with a magnitude less than 0.15. This exercise yields 11 constraint equations, shown below. Note that these constraints have been simplified by removing terms with  $v_i^j = 0$ .

- 1)  $v_0^j = V_{DD} \Rightarrow \frac{L}{2} \cdot (\# \text{ of } v_i^j \text{ pins that are } 1) \leq P_{bnc}$
- 2)  $v_1^j = 1 \Rightarrow k_1 \cdot (v_2^j) + k_2 \cdot (v_3^j) \geq P_1$
- 3)  $v_1^j = -1 \Rightarrow k_1 \cdot (v_2^j) + k_2 \cdot (v_3^j) \leq P_{-1}$
- 4)  $v_1^j = 0 \Rightarrow -P_0 \leq k_1 \cdot (v_2^j) + k_2 \cdot (v_3^j) \leq P_0$
- 5)  $v_2^j = 1 \Rightarrow k_1 \cdot (v_1^j) + k_1 \cdot (v_3^j) \geq P_1$
- 6)  $v_2^j = -1 \Rightarrow k_1 \cdot (v_1^j) + k_1 \cdot (v_3^j) \leq P_{-1}$

- 7)  $v_2^j = 0 \Rightarrow -P_0 \leq k_1 \cdot (v_1^j) + k_1 \cdot (v_3^j) \leq P_0$
- 8)  $v_3^j = 1 \Rightarrow k_2 \cdot (v_1^j) + k_1 \cdot (v_2^j) \geq P_1$
- 9)  $v_3^j = -1 \Rightarrow k_2 \cdot (v_1^j) + k_1 \cdot (v_2^j) \leq P_{-1}$
- 10)  $v_3^j = 0 \Rightarrow -P_0 \leq k_2 \cdot (v_1^j) + k_1 \cdot (v_2^j) \leq P_0$
- 11)  $v_4^j = V_{SS} \Rightarrow \frac{L}{2} \cdot (\# \text{ of } v_i^j \text{ pins that are } -1) \leq P_{bnc}$

### 5.1 Case 1: Fixed $\frac{di}{dt}$

The first bus considered has a fixed  $\frac{di}{dt} = 33 \frac{MA}{s}$ . This corresponds to a data rate of 550 Mb/s in a 50  $\Omega$  system using the rule of thumb that  $data\ rate = \frac{1}{3 \cdot rise\ time}$ .

Note that the  $k_i$  values depend on the magnitude of  $\frac{di}{dt}$ . This means that as  $\frac{di}{dt}$  is changed, the  $k_i$  parameters will also change. However, the absolute voltage that the  $P_x$  parameters represent (i.e., 5% or 10% of  $V_{DD}$ ) will remain fixed.

We next find the set of legal vector sequences. The illegal vector sequences (along with the number of the constraint equation that they violate) are listed in Table 2. Note that the supply bounce constraints are violated frequently. Using the remaining (legal) vector sequences, we construct the digraph  $G$  as described in Section 4.4. We then find the effective bus width  $n$  which can be encoded using the legal transitions in  $G$ , as described in Algorithm 1.

We found the value of the effective bus size  $n$  as a function of the physical bus size  $m$ . The results are shown in Figure 2, where we plot the bus size overhead (i.e.  $\frac{n-m}{m}$ ) as a function of  $n$ . Note that the asymptotic overhead is about 50% (using aggressive inductive cross-talk parameters) and about 30% (using non-aggressive inductive cross-talk parameters).

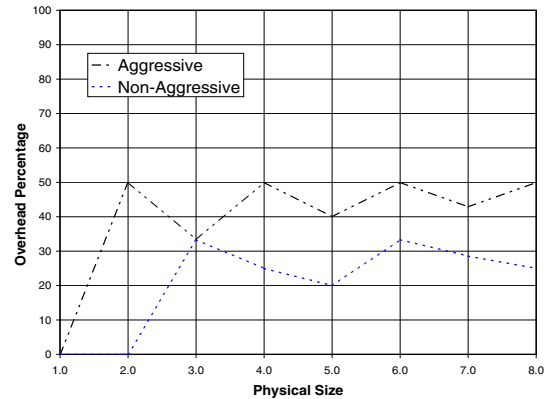


Figure 2: Encoding Efficiency

SPICE simulations were conducted to quantify the increased performance of the encoded bus. The simulation results confirm a reduction in the inductive cross-talk on the bus. Figures 3, 4, and 5 shows the ground bounce, edge degradation, and glitch magnitudes for the three bus configurations. These plots correspond to the worst case inductive cross-talk among all bus pins. Note that the ground bounce magnitude (Figure 3) and the glitch magnitude (Figure 5) for both versions of the encoded bus are exactly at or below the limit specified (5% and 10% of  $V_{DD}$ ), indicating that the experimental results track closely with the theory.

Original	Aggressive	Non-Aggressive
011	Violates 1,4	-
0-1-1	Violates 4,11	-
101	Violates 1,7	-
110	Violates 1,10	-
111	Violates 1,2,5,8	Violates 11
11-1	Violates 1	-
1-11	Violates 1	-
1-1-1	Violates 11	-
-10-1	Violates 7,11	-
-111	Violates 1	-
-11-1	Violates 11	-
-1-10	Violates 10,11	-
-1-11	Violates 11	-
-1-1-1	Violates 3,6,9,11	Violates 1

Table 2: Transition Violations for Fixed  $\frac{di}{dt}$  Example

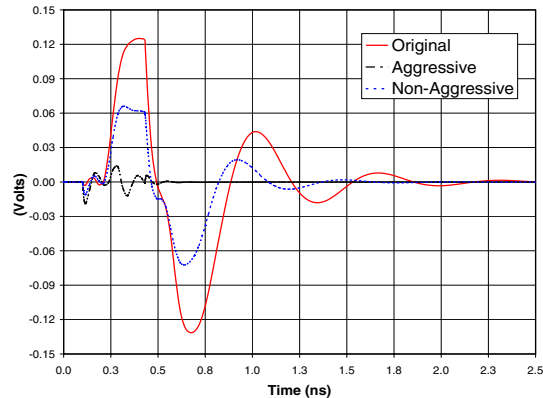


Figure 3: Simulation of Ground Bounce

## 5.2 Case 2: Varying $\frac{di}{dt}$

Using the same analysis technique described in Case 1, we can sweep  $\frac{di}{dt}$  to find the data rate at which the bus reaches the inductive cross-talk limits. For this example, we use the same bus configuration but the constraints are set to limit supply bounce, signal coupling, and glitching to 5% of  $V_{DD}$ . The  $\frac{di}{dt}$  for the original un-coded bus and the encoded bus is increased until the coupling limits are reached. The maximum  $di/dt$  values are 13.3 MA/s (un-encoded) and 37 MA/s (encoded). The 3-bit bus without encoding operates at 222 Mb/s (for a total throughput of 666 Mb/s), while our encoded 2-bit (effective) bus operates at 617 Mb/s (for a total throughput of 1234 Mb/s). Hence, encoding the bus increases the total throughput by 85% using the same physical size and considering the 33% bus size overhead. By relaxing the inductive crosstalk constraints, this overhead can be reduced further. The delay overhead in implementing the encoder is less than 200ps (using a 0.1 $\mu$ m process technology to implement the encoder logic).

## 6 Conclusions

Inductive cross-talk within IC packages is an important factor limiting off-chip I/O throughput. Addressing this issue with aggressive package design is slow and often too expensive for a majority of applications.

In this work, we presented a technique to encode off-chip bus data to avoid inductive cross-talk effects. The technique involves writing constraint equations which express the user-specified bounds on the amount of edge speed degradation, glitch magnitude, and supply bounce that can be tolerated. We

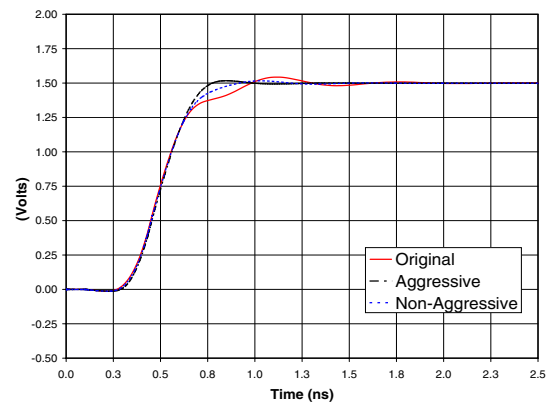


Figure 4: Simulation of Edge Degradation

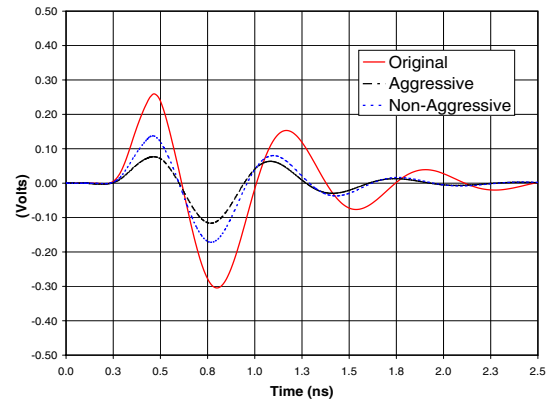


Figure 5: Simulation of Glitch Magnitude

incorporate all these inductive cross-talk effects in a common mathematical framework. We construct a set of legal vector sequences with respect to inductive cross-talk, and use these to develop a CODEC for inductive cross-talk avoidance.

Experimental results track very closely with the theory, and demonstrate an improvement of 85% in the bus throughput for an example bus. Additionally, the asymptotic bus size overhead for our technique is less than 50%.

## References

- [1] "The International Technology Roadmap for Semiconductors." <http://public.itrs.net>, 2003.
- [2] R. Tummala, *Fundamentals of Microsystem Packaging*. McGraw-Hill, 2001.
- [3] M. Miura, N. Hirano, Y. Hiruta, and T. Sudo, "Electrical characterization and modeling of simultaneous switching noise for leadframe packages," in *Proceedings of 45th Electronic Components and Technology Conference*, pp. 857-864, May 1995.
- [4] B. Young, "Return path inductance in measurements of package inductance matrices," in *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, vol. 20, Feb 1997.
- [5] N. Hirano, M. Miura, Y. Hiruta, and T. Sudo, "Characterization and reduction of simultaneous switching noise for a multilayer package,"
- [6] M. Lopez, J. Prince, and A. Cangelaris, "Influence of a floating plane on effective ground plane inductance in multilayer and coplanar packages," in *IEEE Transactions on Advanced Packaging*, vol. 22, pp. 182-188, May 1999.
- [7] C. Duan, A. Tirumala, and S. Khatri, "Analysis and avoidance of cross-talk in on-chip buses," *IEEE Symposium on High-Performance Interconnects (HOT Interconnects)*, pp. 133-138, Aug 2001.
- [8] C. Duan and S. Khatri, "Exploiting crosstalk to speed up on-chip buses," *Design Automation and Test in Europe Conference*, Feb 2004.
- [9] B. Victor and K. Keutzer, "Bus encoding to prevent crosstalk delay," in *Proceedings, IEEE/ACM International Conference on Computer Aided Design*, (San Jose, CA), pp. 57-63, Nov 2001.
- [10] M. Powell and T. Vijaykumar, "Pipeline damping: a microarchitectural technique to reduce inductive noise in supply voltage," in *Proceedings of 30th International Symposium on Computer Architecture*, pp. 72-83, June 2003.
- [11] C. Chen and B. Curran, "Switching codes for delta-i noise reduction," in *IEEE Transactions of the 43rd IEEE Midwest Symposium on Circuits and Systems*, vol. 45, pp. 1017 - 1021, Sept 1996.
- [12] 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 43rd IEEE Midwest Symposium on Circuits and Systems*, vol. 2, pp. 718-721, Aug 2000.