

Controlling Inductive Cross-Talk and Power in Off-chip Buses using CODECs

Brock J. LaMeres

Design Validation Division
Agilent Technologies Inc.
Colorado Springs, CO 80907
brock_lameres@agilent.com

Kanupriya Gulati

Electrical Engineering Department
Texas A&M University
College Station, TX 77843
kanu.gulati@gmail.com

Sunil P. Khatri

Electrical Engineering Department
Texas A&M University
College Station, TX 77843
sunil@ee.tamu.edu

Abstract— The parasitic inductances within IC packaging cause supply bounce as well as glitches on the signal pins, significantly limiting the frequency of high-speed inter-chip communication. Also, off-chip communication contributes a large fraction of the total system power. Until recently, the parasitic inductance problem was addressed by aggressive package design, which is expensive. In this work we present a technique to encode the off-chip data transmission to i) limit bounce on the supplies ii) reduce glitching caused by inductive signal coupling from neighboring signals iii) limit the edge degradation of signals due to mutually inducted voltages from neighboring switching signals and iv) control the total power consumption of the I/O logic. All these factors are modeled in a unified mathematical framework. Our experimental results show that the proposed encoding based techniques result in reduced supply bounce and signal glitching due to inductive cross-talk, closely matching the theoretical predictions. Also, we show that the bus size overhead is reasonable even after stringent power reduction constraints are imposed. We demonstrate that the overall bandwidth of a bus actually increases by 100% over an unencoded bus, using our technique with inductive constraints *only* (even after accounting for the encoding overhead). When the power constraints were added (to limit the power to 20% of worst case switching power) in addition to the inductive constraints, the bandwidth was again 100% improved over the unencoded bus. The asymptotic bus size overhead depends on how stringent the user-specified power and inductive cross-talk parameters are. We have validated our approach by *simulating it in an ASIC setting as well as prototyping and testing it in an FPGA environment*.

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 plus 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 desire 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 (solder bump) technologies in level 1 interconnect for high-end applications. While such technologies decrease the above mentioned inductive effects, they are still prohibitively 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 either. While aggressive package design assists in the problem, it is a slow and expensive process to develop new packages.

Another pressing design issue in modern VLSI design is power [1]. The high power consumption of devices has been a significant stumbling block for designers. Approaches which reduce the power consumption of the I/O structures could therefore contribute significantly to the goal of reducing chip-level and system-level power consumption. Typical off-chip output drivers are rated to drive a typical capacitive load of 5pF. Assuming a supply voltage of 1.5V and a switching frequency of 2Gb/s, each output driver requires consumes 22.5mW of power.

In this paper, we present a technique to avoid the inductive cross-talk in the interconnect, and also bound the I/O power of the IC, by encoding the data being transmitted off-chip. The receiving IC decodes this encoded data to recover the original un-encoded information. The implementation of the interface between the two ICs is unaltered, other than the need to utilize additional bits for the encoding. We construct a set of equations which reflect the constraints that any legal vector sequence must satisfy to avoid supply bounce, signal glitching, and signal edge speed degradation. We also construct equations which reflect the condition that the maximum power consumption of the I/O structure is bounded by some

user-specified quantity. The degree of supply bounce, glitching and edge speed degradation that can be tolerated are expressed by means of user-specified parameters as well. From this set of constraint 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. A Reduced Ordered Binary Decision Diagram [6] (ROBDD) based algorithm is used for this purpose.

We note that the proposed approach is applicable to arbitrary-sized buses. In practice, when a wide off-chip bus is implemented on a VLSI IC, it is decomposed into smaller bus segments as described in Section 4. Typically the size of these segments does not exceed 7 or 8 bits. The analysis which is described in the sequel is performed on bus segments of size up to 14 bits.

We show that the inter-chip bus throughput is increased as much as 100% compared to an unencoded bus, by using our inductive encoding techniques alone. By adding power constraints (limiting the power to 20% of the maximum switching power) the bus throughput was still 100% improved over the unencoded bus, with significantly lowered inductive effects as well. The asymptotic bus overhead varies depending on how aggressive the user-specified inductive cross-talk and power constraints are.

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 bus segments with n bus bits each, with the j^{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 $n = 6$ 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:

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

Note that $v_i^j = 1(-1)$ indicates that the i^{th} signal of the j^{th} bus segment is rising (falling), while $v_i^j = 0$ 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^j such that:

- If b_i^j 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^j 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^j 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^j 's transition is in fact *speed up* due to inductive cross-talk.

The power consumed when a capacitance C is charged at frequency f over a voltage range V is $P = C \cdot V^2 \cdot f$. We assume that our I/O drivers are rated to drive a 5pF load at a frequency of 2Gb/s, and a power supply voltage of 1.5V. This results in a power consumption of 22.5mW per output driver.

3 Previous Work

There has been much work into the reduction of parasitic inductance through package advancement [7, 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 Flat Pack (QFP) with wirebonding, a Ball Grid Array (BGA) with wirebonding, and a flip-chip BGA package). The last approach is an example of solder bump technology. 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. We observe

that while solder bump approaches reduce the parasitic inductances, their significant cost makes them cost-effective only for the highest performance designs.

Bus encoding algorithms have been developed to overcome the capacitive cross-talk for on-chip buses [8, 9, 10]. However, the problem of on-chip capacitive cross-talk minimization for buses 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 [8, 9], we utilize memory-based CODEC solutions.

There has been some recent interest in bus encoding to reduce power in buses [11, 12, 13]. These approaches target on-chip buses, in contrast to our work, and as a result they do not consider inductive effects in the problem formulation.

IO signals are often intentionally skewed to avoid inductive cross-talk effects. However, with increasing process variations [1] in recent technologies, these approaches may incur worst-case inductive cross-talk effects. Further, our approach is able to aid signal transitions by *exploiting* inductive cross-talk effects, something that skewing based techniques are unable to do.

In [14], an approach to encode and decode bus data to avoid inductive cross-talk was presented. In contrast to this approach, our work reduces bus power as well, all under a unified mathematical framework. Further, we employ an *implicit, ROBDD* [6] based formulation to compute the legal vectors on the bus, as opposed to the explicit approach of [14]. Finally, we have simulated our approach in an ASIC setting, and prototyped/tested it in an FPGA framework. The work of [14] did not provide such implementation results.

Techniques have been presented to minimize the inductive problems due to packaging. Pipeline damping was presented in [15]. 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 [16] 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. Our approach is the first to include all the inductive and power effects, and model them 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 power and inductive cross-talk constraints for all bits of the j^{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 en-

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

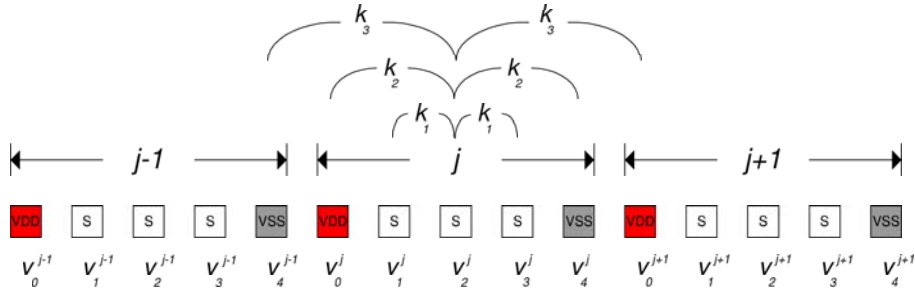


Figure 1: Example Bus Configuration

coding 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 ROBDD [6] which encodes legal transitions between bus vectors. From this ROBDD, 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^{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^{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 4th 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.

We must also guarantee that the total switching power of each segment j is less than the user-specified upper bound p_{max} . Given that the power consumption per pin is p_{pin}^1 , we know that for any segment j :

$$\frac{p_{pin}}{2} \cdot (\# \text{ of } v_i^j \text{ pins that are } -1 \text{ or } 1) \leq p_{max}$$

or, alternately,

$$(\# \text{ of } v_i^j \text{ pins that are } -1 \text{ or } 1) \leq P_{power}$$

Where $P_{power} = \frac{2 \cdot p_{max}}{p_{pin}}$, a user-supplied parameter.

¹For an output pin, $p_{pin} = C \cdot VDD^2 \cdot f$, where C is the trace capacitance (typically 5pF), VDD is the supply voltage (assumed to be 1.5V), and f is the switching frequency (assumed to be 2Gb/s).

- 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} .

- For all signal pins in segment j , ensuring that the power is bounded by p_{max} means that

$$(\# \text{ of } v_i^j \text{ pins that are } -1 \text{ or } 1) \leq P_{power}$$

Where $P_{power} = \frac{2 \cdot p_{max}}{p_{pin}}$, as derived earlier. The factor of 2 arises due to the fact that only one bus transition happens per clock cycle. For example, if $n = 7$ (i.e. there are 5 signal pins per segment) and $p_{max} = 20\%$ of the maximum (i.e. $p_{max} = 0.2 \cdot p_{pin} \cdot 5$) then $P_{power} = 2$. In the sequel we refer to the power constraint as a percentage of the maximum possible value for

ease of exposition.

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:

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

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

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 $\frac{z}{2}$ 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} .

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

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, iii) the bounce on V_{DD} and V_{SS} pins should be bounded and iv) the power in the bus segment is bounded.

4.3 Constructing Legal Vector Sequences

Consider a particular bus configuration (n , k , and α) and user-specified inductive cross-talk constraints (P_1 , P_{-1} , P_0 and P_{bnc}) and power constraint P_{power} . For each signal pin i within the segment j , three constraints equations are written (for $v_i^j = 1, -1, \text{ and } 0$, per Section 4.1). For each power supply pin, one constraint expression is written, per Section 4.2. For each bus segment, we write one power constraint equation as described in Section 4.1. This results in a total of $3n - 3$ 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 $3n - 3$ 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 ($v_i^j = 1, -1, \text{ and } 0$) per signal bit, the total number of vector sequences that need to be tested for legality is $3^{(n+2p-6)}$. Note that the values of n and p for realistic buses is small, so these tests (which need to be done exactly once for a design) can be performed easily. In our experiments, $n = 7$ and $p = 2$, which is reasonable for real-life buses.

After testing the vector sequences for legality modulo inductive cross-talk and power, 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 , P_{bnc} and P_{power} 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 ROBDD [6] 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_{from}^j and w_{to}^j . 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 $v_i^j = 0$.

$G(w_{from}^j, w_{to}^j) = 1$ indicates the legality (from an inductive cross-talk and power 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 and power standpoint, we can construct a ROBDD G whose minterms ($w_{from}^j : w_{to}^j$) are vectors in B^{2n} , such that they indicate a legal transition (from an inductive cross-talk and power 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

```

rest_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) ≥ 2^m
  degrees_changed = 1
  while degrees_changed do
    degrees_changed = 0
    for each v_s ∈ V do
      for each v_d S.T. G(v_s, v_d) = 1 do
        if v_d ∉ V then
          decrement out - degree(v_s) in V
          degrees_changed = 1
        end if
        if out - degree(v_s) < 2^m then
          V ← V \ v_s
          break
        end if
      end for
    end for
  end while
end while
if |V| ≥ 2^m then
  print(m bit bus may be encoded using G)
else
  print(m bit bus cannot be encoded using G)
end if

```

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 ($n = 7$, $k = 3$).

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 segment to avoid inductive cross-talk and limit power consumption. The bus segment configuration is shown in Figure 1, except that our experimental bus segment had 5 signal bits (i.e. $n = 5 + 2$). We used electrical parameters from a standard BGA-wb package in our simulations. This bus segment was encoded using P_0 , P_1 , P_{-1} and P_{bmc} set to 12.5% of V_{DD} . We compared three configurations – i) an unencoded bus segment ii) a bus segment encoded only for inductive constraints and iii) a bus segment encoded for inductive as well as power constraints.

The first step consists of writing the constraint equations for every pin in the bus segment. Example constraints for $n = 5$ (i.e. 3 signal pins), $k = 3$, and $\alpha = 5/2$ are provided below. From the inductive coupling values in Table 1, we set $p = 2$ to ignore inductive coupling with a magnitude less than 0.15. For $p > 2$, the mutual inductive coupling drops off rapidly, justifying our choice. This exercise yields 12 constraint equations, shown below. Note that these constraints have been simplified by removing terms with $v_i^j = 0$.

- 1) $v_0^j = V_{DD} \Rightarrow \frac{k}{2} \cdot (\# \text{ of } v_i^j \text{ (or } v_i^{j-1}) \text{ pins that are 1)} \leq P_{bmc}$
- 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) $v_4^j = V_{SS} \Rightarrow \frac{k}{2} \cdot (\# \text{ of } v_i^j \text{ (or } v_i^{j-1}) \text{ pins that are } -1) \leq P_{bmc}$
- 12) $(\# \text{ of } v_i^j \text{ pins that are } -1 \text{ or } 1) \leq P_{power}$

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., 12.5% of V_{DD}) will remain fixed.

We next find the set of legal vector sequences. We note that the supply bounce and power constraints were violated most frequently. Using the remaining (legal) vector sequences, we construct the ROBDD G as described in Section 4.4. We then find the effective bus width m which can be encoded using the legal transitions in G , as described in Algorithm 1.

We found the value of the effective bus size m as a function of the physical bus size $n - 2$ (since 2 pins are VDD and VSS). The results are shown in Table 2, where we list the effective bus size as a function of $n - 2$. Note that the second column of this table indicates the effective bus size assuming no power constraints are specified. The third, fourth and fifth columns were generated assuming a power constraint of 33%, 20% and 18% of the maximum bus power. These columns include inductive constraints just as in column 2. Note that the effective bus width reported in Table 2 refer to the effective number of *signal* pins in the corresponding bus segment. Also, note that when the value of n and the power constraint are both small, it is impossible to find m , since it is possible that all transitions on the bus segment become illegal.

Table 2 indicates that for bus segments with 7 or more signal pins, the effective bus sizes are comparable when we utilize encoding with or without power constraints. This suggests that if we were to use bus segments with 7 or more signal pins, we can curtail inductive effects and also limit power with a small bus size penalty. In fact the bus throughput increases significantly with encoding, as we will discuss shortly.

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

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

SPICE simulations were conducted to quantify the increased performance of the encoded bus segment. We utilized a TSMC 0.13 μ m process for this purpose. We compared the *original* unencoded bus segment with a *non-aggressive* encoded segment (which represents the case when only inductive constraints were used) and an *aggressive* encoded segment (which represents the case when both inductive and power (limiting the power to 20% of the maximum) constraints were applied).

The simulation results confirm a reduction in the inductive cross-talk on the bus segment, while power is restricted within its specified bound (20% of maximum). SPICE plots are not shown due to lack of space. We observed that the ground bounce magnitude and the glitch magnitude for both versions of the encoded bus are exactly at or below the limit specified (12.5% of V_{DD}), indicating that the *experimental results track closely with the theory*. The aggressive constraints further reduced the glitching and supply bounce magnitudes. We also found that the edge degradation constraints do not play a major part in determining the final solution. This was because satisfying the remaining constraints (particularly the power constraint and the supply bounce constraint) typically ensured that the edge degradation was severely limited.

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 power and inductive cross-talk limits. For this example, we use the same bus configuration as in the previous section. The *original*, *non-aggressive* and *aggressive* conditions are also as described earlier.

The $\frac{di}{dt}$ for the *original* bus and the *non-aggressive* and *aggressive* encoded bus is increased until the coupling limits are reached. The maximum di/dt values are 8.0 MA/s (original), 19.9 MA/s (non-aggressive) and 37.0 MA/s (aggressive). The 5-bit bus without encoding operates at 133 Mb/s (for a total throughput of 665 Mb/s), while our non-aggressive encoded 4-bit (effective) bus operates at 333 Mb/s (for a total throughput of 1332 Mb/s). The aggressive encoded 2-bit (effective) bus operates at 666 Mb/s, for a total throughput of 1332 Mb/s. Hence, encoding the bus increases the total throughput by 100% using the same physical size of the bus, and considering the encoder overhead. The power reduction methodology helps further, allowing us to reduce power to 20% of the worst case, while retaining the throughput of the non-aggressive case.

5.2.1 TSMC 0.13 μ m ASIC Process

The CODECs were implemented using the TSMC 0.13 μ m CMOS IC process to understand their impact on delay and area of the IC. Bus sizes of 2, 4, 6, and 8 were used. For each of these sizes, both the aggressive and the non-aggressive CODECs were synthesized, placed and routed.

$n - 2$	no power constraint	33%	20%	18%
3	2	0	0	0
4	3	2	0	0
5	4	2	2	0
6	5	2	2	2
7	5	4	3	3
8	6	5	3	3
9	7	5	3	3
10	7	6	5	3
11	8	7	6	3
12	8	7	6	6

Table 2: Effective Bus Width for Different Power Constraint Values

	Bus Size (m)	Style	
		aggressive	non-aggressive
Delay (ns)	2	0.170	N/A
	4	0.670	0.503
	6	1.150	0.955
	8	1.310	0.983
Area (μm^2)	2	22	N/A
	4	152	114
	6	614	509
	8	1,181	886

Table 3: Encoder in a TSMC 0.13 μm Process

Table 3 lists the delay and area impact of the CODECs implemented in a TSMC 0.13 μm process. The delays in this table represent the delay of the encoder. *Note that encoding and decoding delays are unimportant for heavily pipelined systems, where these delays can be hidden.* In case of heavily pipelined systems, the maximum data-rate is significantly improved by using our encoding based schemes.

This table illustrates the negligible impact of our approach on a modern VLSI design.

5.2.2 Xilinx 0.35 μm FPGA Experiment

The CODECs were also synthesized, mapped, prototyped and tested for a Xilinx VirtexIIPro, Field Programmable Gate Array (FPGA) which used a 0.35 μm CMOS process.

CODECs for bus sizes of 2, 4, 6, and 8 were implemented using both the aggressive and non-aggressive constraints. Table 4 lists the delay and area impact of our CODECs when implemented in the FPGA environment. In all cases, the CODEC designs occupied less than 1% of the FPGA to be implemented. The CODECs were implemented using standard Function Generators (FGs) within the FPGA which resulted in minimal propagation delay through the circuit. As noted earlier, in the case of pipelined data transfers, the actual delay of the encoding and decoding process can be hidden.

The outputs of the FPGA were monitored using the 16950A Logic Analyzer from Agilent Technologies Inc. The logic analysis measurements verified that the CODECs could be taken from the conception stage to final implementation using standard IC design practices. Logic analyzer measurement results and a photograph of the FPGA test setup are not shown for lack of space.

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. Another important design issue in modern VLSI design is power. Approaches which limit the power consumption of the I/O structures are therefore important to achieve the goal of reduced chip-level and system-level power consumption.

	Bus Size (m)	Style	
		aggressive & 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 4: Bus Expansion Encoder Implementation Results using a 0.35 μm , CMOS FPGA Process

In this work, we presented a technique to encode off-chip bus data to avoid inductive cross-talk effects as well as to limit the power consumption of the I/O. Our technique involves writing constraint equations which express the user-specified bounds on the amount of edge speed degradation, glitch magnitude, supply bounce and power consumption that can be tolerated. We express all these constraints in a common mathematical framework. We construct a set of legal vector sequences with respect to inductive cross-talk and power, and use these to develop a CODEC for inductive cross-talk avoidance. The CODEC is constructed using a ROBDD based computation.

Experimental results track very closely with the theory, and demonstrate an improvement of 100% in the bus throughput for an example 5-bit bus when only inductive constraints are applied. When power constraints (limiting the power of the bus to 20% of the worst case) are applied, the bus throughput is still 100% improved over the unencoded bus. The reduced switching results in improved glitching and supply bounce performance as well. We have validated our approach by simulating it in an ASIC setting as well as prototyping and testing it in an FPGA environment.

References

- [1] "The International Technology Roadmap for Semiconductors," <http://public.itrs.net>, 2003.
- [2] R. Tummalo, *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] R. E. Bryant, "Graph based algorithms for Boolean function representation," *IEEE Transactions on Computers*, vol. C-35, pp. 677–690, August 1986.
- [7] 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.
- [8] 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.
- [9] C. Duan and S. Khatri, "Exploiting crosstalk to speed up on-chip buses," *Design Automation and Test in Europe Conference*, Feb 2004.
- [10] 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.
- [11] M. Stan and W. Burleson, "Bus-invert coding for low-power I/O," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, pp. 49–58, Mar 1995.
- [12] P. Sotiriadis and A. Chandrakasan, "Reducing bus delay in submicron technology using coding," in *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 109–114, Jan-Feb 2001.
- [13] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf, "A dictionary-based en/decoding scheme for low-power data buses," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, pp. 943–951, Oct 2003.
- [14] B. LaMeres and S. Khatri, "Encoding-based minimization of inductive cross-talk for off-chip data transmission," in *Proceedings, Design Automation and Test in Europe (DATE) Conference*, (Munich, Germany), Mar 2005.
- [15] 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.
- [16] 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.
- [17] 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.