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

Authors: Brock J. LaMeres, Kanupriya Gulati, Sunil P. Khatri

Agilent Technologies, Texas A&M University, Texas A&M University
Motivation

• **Power delivery is the biggest challenge facing designers entering DSM**
  - The IC core current continues to increases ($P_4 = 80$Amps).
  - The package interconnect inductance limits instantaneous current delivery.
  - The inductance leads to ground and power supply bounce.

• **SSN on signal pins is the leading cause of inter-chip bus failure**
  - Ground/power supply bounce causes unwanted switching.
  - Mutual Inductive cross-talk causes edge degradation which limits speed.
  - Mutual Inductive cross-talk causes glitches which results in unwanted switching.

• **Further, power in off-chip buses can be significant.**
  - Large percentage of power may be consumed in the output stages

• **Aggressive package design helps, but is too expensive:**
  - Flip-Chip technology can reduce the interconnect inductance.
  - Flip-Chip requires a unique package design for each ASIC.
  - This leads to longer process time which equals cost.
  - 90% of ASIC design starts use wire-bonding due to its low cost.
  - Wire-bonding has large parasitic inductance that must be addressed.
Our Solution

“Encode Off-Chip Data to Avoid Inductive Cross-talk & Power Consumption”

• Avoid the following cases:

1) Excessive switching in the same direction  = reduce ground/power bounce
2) Excessive X-talk on a signal when switching  = reduce edge degradation
3) Excessive X-talk on signal when static  = reduce glitching
4) At the same time, limit the number of transitions  = reduce power
Our Solution

• This results in:

1) A subset of vectors is transmitted that avoids inductive X-talk & power.

2) The off-chip bus can now be ran at a higher data rate.

3) The subset of vectors running faster can achieve a higher throughput over the original set of vectors running slower.

\[
\text{Throughput of less vectors at higher data-rate} > \text{Throughput of more vectors at lower data-rate}
\]
Agenda

1) Inductive X-talk & Power
2) Terminology
3) Methodology
4) Experimental Results
5) Conclusion
1) Inductive X-Talk

Supply Bounce

• The instantaneous current that flows when signals switch induces a voltage across the inductance of the power supply interconnect following:

\[ V_{bnc} = L \cdot \left( \frac{di}{dt} \right) \]

• When more than one signal returns current through one supply pin, the expression becomes:

\[ V_{bnc} = L \cdot \sum_j \left( \frac{di}{dt} \right) \]

NOTE: Reducing the number of signals switching in the same direction at the same time will reduce the supply bounce.
1) Inductive X-Talk

Glitching

• Mutual inductive coupling from neighboring signals that are switching cause a voltage to induce on the victim that is static:

\[ V_{\text{glitch}}^i = \pm M_{ik} \cdot \left( \frac{di_k}{dt} \right) \]

• The net coupling is the summation from all neighboring signals that are switching:

\[ V_{\text{glitch}}^i = \sum_{k=1}^{m} \pm M_{ik} \cdot \left( \frac{di_k}{dt} \right) \]

\[ M_{ik} = K_{ik} \cdot \sqrt{L_i \cdot L_k} \]

NOTE: The mutual inductive coupling can be canceled out when two neighbors of equal \( K_{ik} \) switch in opposite directions. Also, \( K_{ik} \) is the mutual inductive coupling coefficient.
1) Inductive X-Talk

**Edge Degradation**

- Mutual inductive coupling from neighboring signals that are switching cause a voltage to be induced on the victim that is also switching. This follows the same expression as glitch coupling:

\[
V_{glitch} = \sum_{k=1}^{k} \pm M_{1k} \cdot \left( \frac{di_k}{dt} \right)
\]

- The mutual inductive coupling can be manipulated to cause a positive (negative) glitch for a rising (falling) signal.
- Mutual coupling can thus be exploited so as to *help* the transition resulting in a faster rise-time or fall-time (alternately, to *not hinder* the risetime of the transition)
1) Power

## Power Consumption

- The power consumed in the output stage is proportional to the capacitance being driven, the output voltage swing, and the switching frequency.

\[ P_{pin} = C \cdot V_{DD}^2 \cdot f \]

**NOTE**: Power is proportional to the number of switching pins.
2) Terminology

Define the following:

\[ n = \text{width of the bus segment} \]
\[ \text{where each bus segment consists of } n-2 \text{ signals and 1 } V_{DD} \text{ and 1 } V_{SS}. \]

\[ j = \text{the segment consisting of an n-bit bus.} \]
\[ j \text{ is the segment under consideration.} \]
\[ j-1 \text{ is the segment to the immediate left.} \]
\[ j+1 \text{ is the segment to the immediate right.} \]
\[ \text{each segment has the same } V_{DD}/V_{SS} \text{ placement.} \]
2) Terminology

Define the following:

\[ v_i^j = \begin{cases} 
1 & \text{rising edge} \\
-1 & \text{falling edge} \\
0 & \text{signal is static} 
\end{cases} \]

This 3-valued algebra enables us to model mutual inductive coupling of any sign.
2) Terminology

Define the following coding constraints:

**Supply Bounce**

if \( v_i^j \) is a supply pin, the total bounce on this pin is bounded by \( P_{bnc} \).

\( P_{bnc} \) is a user defined constant.

**Glitching**

if \( v_i^j \) is a signal pin and is static (\( v_i^j = 0 \)), the total magnitude of the glitch from switching neighbors should be less than \( P_0 \). \( P_0 \) is a user defined constant.

**Edge Degradation**

if \( v_i^j \) is a signal pin and is switching (\( v_i^j = 1/-1 \)), the total magnitude of the coupling from switching neighbors should be greater than \( P_1/P_{-1} \). This coupling should not hurt (should aid) the transition. \( P_1/P_{-1} \) is a user defined constant.
2) Terminology - Power

Define the following coding constraints:

**Power**

for a given segment $j$, the total power consumption on that segment is bounded by $P_{\text{power}}$. 
$P_{\text{power}}$ is a user defined constant.
2) Terminology

Also define the following:

\[ p = \text{how far away to consider coupling} \]
(ex., \( p = 3 \), consider \( K_{11}, K_{12}, \) and \( K_{13} \) on each side of the victim)

\[ k_q = \text{Magnitude of coupled voltage on pin } i \text{ when its } q^{th} \text{ neighbor } p \text{ switches:} \]

\[ k_q = \left| M_{ip} \cdot \left( \frac{d i_p}{dt} \right) \right| \]
3) Methodology

• For each pin $v_i^j$ within segment $j$, we will write a series of constraints that will bound the inductive cross-talk magnitude.

• The constraints will differ depending on whether $v_i^j$ is a signal or power pin.

• The coupling constraints will consider signals in adjacent segments $(j+1, j-1)$ depending on $p$. 
3) Methodology – Signal Pin Constraints

**Glitching**: coupling is bounded by $P_0$

Example:

$v_2^j = 0$, and $p=3$. This means the three adjacent neighbors on either side of $v_2^j$ need to be considered ($v_4^{j-1}$, $v_0^j$, $v_f^j$, $v_3^j$, $v_4^j$, $v_0^{j+1}$).

Note we use **modulo n** arithmetic (and consider adjacent segments as required).

\[
v_2^j = 0 \text{ (static)}
\]

\[
-P_0 \leq k_3 \cdot (v_4^{j-1}) + k_2 \cdot (v_0^j) + k_f \cdot (v_f^j) + k_1 \cdot (v_3^j) + k_2 \cdot (v_4^j) + k_3 \cdot (v_0^{j+1}) \leq P_0
\]

The constraint equation is tested against each possible transition and the transitions that violate the constraint are eliminated.
3) Methodology – Signal Pin Constraints

**Edge Degradation**: coupling is bounded by $P_1$ and $P_{-1}$

**Example:**

$v_j^2 = 1$ or $-1$, and $p = 3$. This means the three adjacent neighbors on either side of $v_j^2$ need to be considered ($v_{j-1}^4$, $v_j^0$, $v_j^1$, $v_j^3$, $v_{j+1}^4$, $v_{j+1}^0$).

\[ v_j^2 = 1 \text{ (rising)} \]

\[
\begin{align*}
&k_3 \cdot (v_{j-1}^4) + k_2 \cdot (v_j^0) + k_1 \cdot (v_j^1) + k_1 \cdot (v_j^3) + k_2 \cdot (v_{j+1}^4) + k_3 \cdot (v_{j+1}^0) \geq P_1 \\
&k_3 \cdot (v_{j-1}^4) + k_2 \cdot (v_j^0) + k_1 \cdot (v_j^1) + k_1 \cdot (v_j^3) + k_2 \cdot (v_{j+1}^4) + k_3 \cdot (v_{j+1}^0) \leq P_{-1}
\end{align*}
\]

Again, the constraint equations are tested against each possible transition and the transitions that violate the constraints are eliminated.
3) Methodology – Power Pin Constraints

**Supply Bounce**: coupling is bounded by \( P_{bnc} \)

Example:
\( v_o^j = V_{DD} \) or \( V_{SS} \). The total number of switching signals that use \( v_o^j \) to return current must be considered. Due to symmetry of the bus arrangement, signal pins will always return current through two supply pins. i.e., \((v_o^{j-1} \text{ and } v_o^j)\) or \((v_d^{j} \text{ and } v_d^{j+1})\). This results in the self inductance of the return path being divided by 2. Let \( z = |L \frac{di}{dt}| \) for any pin. Then,

\[ (z/2) \cdot (\# \text{ of } v_i^j \text{ pins that are 1}) \leq P_{bnc} \]

\[ v_d^j = V_{SS} \]

\[ (z/2) \cdot (\# \text{ of } v_i^j \text{ pins that are -1}) \leq P_{bnc} \]
3) Methodology – Power Constraints

**Power Consumption**: consumption is bounded by $P_{\text{power}}$

**Example:**

*For segment $j$. The total number of switching signals can be constrained to reduce power.*

Segment $j$

$(\# \text{ of } v_i^j \text{ pins that are 1 or -1}) \leq P_{\text{power}}$
3) Methodology – Constructing Legal Vectors Sequences

• For each bit in the $j^{th}$ segment bus, constraints are written.

• If the pin is a signal, 3 constraint equations are written;
  - $v_j = 0$, the bit is static and a *glitching constraint* is written
  - $v_j = 1$, the bit is rising and an *edge degradation* constraint is written.
  - $v_j = -1$, the bit is falling and an *edge degradation* constraint is written.

• If the pin is $V_{DD}$, 1 constraint equation is written to avoid *supply bounce*.

• If the pin is $V_{SS}$, 1 constraint equation is written to avoid *ground bounce*.

• For the segment, 1 constraint equation is written to constrain *power*.
3) Methodology – Constructing Legal Vectors Sequences

• This results in the total number of constraint equations written is:

\[(3 \cdot n - 3)\]

• Each equation must be evaluated for each possible transition to verify if the transition meets the constraints. The total number of transitions that are evaluated depends on \(n\) and \(p\):

\[3(n+2p - 6)\]

• This follows since there are \(n-2\) signal pins in the segment \(j\), and \(2p-4\) signal pins in neighboring segments.

• **The values of \(n\) and \(p\) are small in practice, hence this is tractable.**
3) Methodology – Constructing the CODEC

- The remaining legal transitions are used to create the CODEC.
- The total number of remaining legal transitions will depend on how aggressive the user-defined constants are chosen \((P_0, P_1, P_{-1}, P_{bnc}, P_{power})\).
- From the remaining legal transitions, find the effective bus width \(m\) that can be encoded using a physical bus of width \(n\), using a memory-based CODEC.
  - Utilize a fixpoint computation
3) Methodology – Constructing the CODEC

- Represent remaining legal transitions in a digraph
- Algorithm to find CODEC:
  - Let $n =$ size of physical bus
  - Let $m =$ size of effective bus
  - Then the digraph of legal transitions of the $n$ bit bus can encode an $m$ bit bus ($m < n$) iff
    - We can find a closed set $S$ of nodes such that
      - $|S| \geq 2^m$
      - Each vertex $s$ in $S$ has at least $2^m$ out-edges (including self-edges) to vertices $s'$ in $S$
  - Now we can synthesize the encoder and decoder (memory based).
4) Experimental Results – 5 Signal Pins

Example Bus: \( n=7, p=2 \)

\[ P_0, P_1, P_{-1}, P_{bnc} \]

- Aggressive Encoding: 5% of \( V_{DD} \)
- Non-Aggressive Encoding: 12.5% of \( V_{DD} \)
- Power Encoding: 20% of Max
4) Experimental Results – Constraint Equations

# of Constraints = (3n – 3) = 12

1) \( v_0^j = \text{VDD} \) → \((L/2) \cdot (\text{# of } v_1^j \text{ pins that are 1}) \leq P_{\text{bnc}}\)

2) \( v_1^j = 1 \) → \( k_1 \cdot (v_2^j) + k_2 \cdot (v_3^j) \geq P_1 \)

3) \( v_1^j = -1 \) → \( k_1 \cdot (v_2^j) + k_2 \cdot (v_3^j) \leq P_{-1} \)

4) \( v_1^j = 0 \) → \( -P_0 \leq k_1 \cdot (v_2^j) + k_2 \cdot (v_3^j) \leq P_0 \)

5) \( v_2^j = 1 \) → \( k_1 \cdot (v_1^j) + k_1 \cdot (v_3^j) \geq P_1 \)

6) \( v_2^j = -1 \) → \( k_1 \cdot (v_1^j) + k_1 \cdot (v_3^j) \leq P_{-1} \)

7) \( v_2^j = 0 \) → \( -P_0 \leq k_1 \cdot (v_1^j) + k_1 \cdot (v_3^j) \leq P_0 \)

8) \( v_3^j = 1 \) → \( k_2 \cdot (v_1^j) + k_1 \cdot (v_2^j) \geq P_1 \)

9) \( v_3^j = -1 \) → \( k_2 \cdot (v_1^j) + k_1 \cdot (v_2^j) \leq P_{-1} \)

10) \( v_3^j = 0 \) → \( -P_0 \leq k_2 \cdot (v_1^j) + k_1 \cdot (v_2^j) \leq P_0 \)

11) \( v_4^j = \text{VSS} \) → \((L/2) \cdot (\text{# of } v_1^j \text{ pins that are -1}) \leq P_{\text{bnc}}\)

12) \((\text{# of } v_1^j \text{ pins that are -1 or 1}) \leq P_{\text{power}}\)
4) Experimental Results – CASE 1: Fixed di/dt

Transitions Eliminated due to Rule Violations

<table>
<thead>
<tr>
<th>Transition</th>
<th>Aggressive</th>
<th>Non Aggressive</th>
</tr>
</thead>
<tbody>
<tr>
<td>011</td>
<td>violates 1,4</td>
<td>-</td>
</tr>
<tr>
<td>0-1-1</td>
<td>violates 4,11</td>
<td>-</td>
</tr>
<tr>
<td>101</td>
<td>violates 1,7</td>
<td>-</td>
</tr>
<tr>
<td>110</td>
<td>violates 1,10</td>
<td>-</td>
</tr>
<tr>
<td>111</td>
<td>violates 1,2,5,8</td>
<td>violates 11</td>
</tr>
<tr>
<td>11-1</td>
<td>violates 1</td>
<td>-</td>
</tr>
<tr>
<td>1-11</td>
<td>violates 1</td>
<td>-</td>
</tr>
<tr>
<td>1-1-1</td>
<td>violates 11</td>
<td>-</td>
</tr>
<tr>
<td>-10-1</td>
<td>violates 7,11</td>
<td>-</td>
</tr>
<tr>
<td>-111</td>
<td>violates 1</td>
<td>-</td>
</tr>
<tr>
<td>-11-1</td>
<td>violates 11</td>
<td>-</td>
</tr>
<tr>
<td>-1-10</td>
<td>violates 10,11</td>
<td>-</td>
</tr>
<tr>
<td>-1-11</td>
<td>violates 11</td>
<td>-</td>
</tr>
<tr>
<td>-1-1-1</td>
<td>violates 3,6,9,11</td>
<td>violates 1</td>
</tr>
</tbody>
</table>
4) Experimental Results – CASE 1: Fixed $di/dt$

- **Encoded data avoids Inductive X-talk pattern**

\[
\text{Overhead} = 1 - \frac{\text{Effective}}{\text{Physical}} = \frac{n - m}{m}
\]

- **Bus can be ran faster**
4) Experimental Results – CASE 1: Fixed di/dt

Ground Bounce Simulation
4) Experimental Results – CASE 1: Fixed di/dt

Glitch Simulation

![Graph showing Glitch Simulation for different cases: Original, Aggressive, Non-Aggressive. The graph plots voltage (Volts) against time (ns).]
4) Experimental Results – CASE 1: Fixed di/dt

Edge Degradation Simulation
4) Experimental Results – CASE 2: Variable di/dt

- di/dt was swept for both the non-encoded and encoded configuration.
- the maximum di/dt was recorded that resulted in a failure.
- Failure: 5% of VDD (Aggressive) and 12.5% of VDD (Non-Aggressive)
- the maximum di/dt was converted to data rate and throughput.

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Aggressive</th>
<th>Non-Aggr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum di/dt:</td>
<td>8 MA/s</td>
<td>19.9 MA/s</td>
<td>37 MA/s</td>
</tr>
<tr>
<td>Maximum data-rate per pin:</td>
<td>133 Mb/s</td>
<td>333 Mb/s</td>
<td>667 Mb/s</td>
</tr>
<tr>
<td>Effective bus width:</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Total Throughput:</td>
<td>667 Mb/s</td>
<td>1332 Mb/s</td>
<td>1332 Mb/s</td>
</tr>
<tr>
<td>Improvement</td>
<td>-</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Power Constraint (% of Max)</td>
<td>100%</td>
<td>20%</td>
<td>20%</td>
</tr>
</tbody>
</table>
4) Experimental Results – ASIC Synthesis

- A 0.13um, TSMC ASIC process was used.
- Delay and Area Extracted

<table>
<thead>
<tr>
<th>Bus Size (m)</th>
<th>Style</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>aggressive</td>
<td>N/A</td>
</tr>
<tr>
<td>Delay ((ns))</td>
<td>2</td>
<td>0.170</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.670</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1.150</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1.310</td>
</tr>
<tr>
<td>Area ((um^2))</td>
<td>2</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>152</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>614</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1,181</td>
</tr>
</tbody>
</table>
4) Experimental Results – FPGA Implementation

- A Xilinx, Virtex-II, 0.35um, FPGA was used.
- Delay and Area Extracted

<table>
<thead>
<tr>
<th></th>
<th>Bus Size ((m))</th>
<th>Style</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-</td>
<td>aggressive &amp; non-aggressive</td>
</tr>
<tr>
<td>Delay ((ns))</td>
<td>2</td>
<td>0.351</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.020</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1.450</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1.610</td>
</tr>
<tr>
<td>FPGA Usage</td>
<td>2</td>
<td>&lt; 1%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>&lt; 1%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>&lt; 1%</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>&lt; 1%</td>
</tr>
<tr>
<td>FPGA Implementation</td>
<td>2</td>
<td>3x, 2-Input FG’s</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6x, 4-Input FG’s</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>9x, 6-Input FG’s</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>12x, 8-Input FG’s</td>
</tr>
</tbody>
</table>
5) Conclusion

• Using a single mathematical framework, inductive X-talk & power constraints can be written that consider supply bounce, glitching, and edge degradation.

• This technique can be used to encode off-chip data transmission to reduce inductive X-talk & power to acceptable levels.

• It was demonstrated that even after reducing the effective bus size, the improvement in per pin data-rate resulted in an increase in throughput compared to a non-encoded bus.
Thank you!