Performance Modeling and Noise Reduction in VLSI Packaging

Ph.D. Defense

Brock J. LaMeres
University of Colorado
October 7, 2005
Problem Statement

• VLSI Packaging Limits System Performance

  1) Supply Bounce
  2) Signal Coupling
  3) Bandwidth Limitation
  4) Impedance Discontinuities
  5) Cost & Scaling
Agenda

1) Problem Motivation

2) Research Overview

3) Advantages Over Prior Techniques

4) Broader Impact of this Work
1) Problem Motivation
Why is packaging limiting performance?

- **IC Design/Fabrication is Outpacing Package Technology**
  - We’re seeing exponential increase in IC transistor performance
  - >1.3 Billion transistors on 1 die [Fall IDF-05]
Why is packaging limiting performance?

• Packages Have Been Designed for Mechanical Performance
  - Electrical performance was not primary consideration
  - IC’s limited electrical performance
  - Package performance was not the bottleneck
Why is packaging limiting performance?

• VLSI Performance Exceeds Package Performance
  - Packages optimized for mechanical reliability, but still used due to cost
  - IC performance far exceeds package performance

On-Chip
- $f_{IC} > 4\text{GHz}$
- large signal counts
- exponential scaling

Package
- $f_{pkg} < 2\text{GHz}$
- limited signal counts
- linear scaling
Why is packaging limiting performance?

• **Package Interconnect Contains Parasitic Inductance and Capacitance**

- Long interconnect paths

- Large return loops

\[
L = \frac{\Phi}{I}
\]

\[
C = \frac{Q}{V} = \frac{A\varepsilon}{t}
\]
Why is packaging limiting performance?

• **Package Parasitics Limit Performance**

  - Excess L and C causes package noise
  - Noise limits how fast the package can transmit data

1) Supply Bounce
2) Signal Coupling
3) Bandwidth Limiting
4) Impedance Discontinuities
Why is packaging limiting performance?

- **Aggressive Package Design Helps, but is expensive…**
  - 95% of VLSI design-starts are wire bonded
  - Goal: Extend the life of current packages

  QFP – Wire Bond : 4.5nH → $0.22 / pin

  BGA – Wire Bond : 3.7nH → $0.34 / pin ***

  BGA – Flip-Chip : 1.2nH → $0.63 / pin
2) Research Overview
Research Overview

• **Performance Modeling & Bus Sizing**
  - algebraic model to predict performance and cost-effectiveness

• **Bus Expansion CODEC**
  - encoding data to avoid patterns on bus which cause excessive noise

• **Bus Stuttering CODEC**
  - encoding data to avoid patterns on bus which cause excessive noise

• **Impedance Compensation**
  - adding C or L near package to match impedance to system
Publications: Performance Modeling and Bus Sizing

- “FPGA I/O – When to go serial”,
  *IEE Electronic Systems and Software*, 2004

- “Performance Model for Inter-Chip Busses Considering Bandwidth and Cost”
  *DesignCon*, 2005
  • *Best Paper Award*

- “Performance Model for Inter-chip Com Considering Inductive Cross-talk and Cost”,
  *ISCAS*, 2005

- “Performance Model for Inter-Chip Busses Considering Bandwidth and Cost”,
  *DesignConEast*, 2005

- “Package Performance Model for Off-chip Busses Considering Bandwidth and Cost”,
  *IEE Journal on Computers and Digital Techniques* (accepted for publication)
Publications: Bus CODECs to Avoid Package Noise

- “Encoding-based Minimization of Inductive Cross-talk for Off-chip Data Transmission”,
  DATE, 2005

- “Controlling Inductive Cross-talk and Power in Off-chip Buses using CODECS”,
  ASP-DAC 2006 (accepted for publication)

- “Bus Stuttering: An Encoding Technique to Reduce Inductive Noise in Off-Chip Data”,
  DATE 2006 (submitted)
Publications: Impedance Compensation

- “Time Domain Analysis of a Printed Circuit Board Via”,
  *Microwave Journal*, 2000

- “The Effect of Ground Vias on Changing Signal Layers in Multi-Layered PCBs”,
  *Microwave and Optical Technology Letters*, 2001

- “Broadband Impedance Matching for Inductive Interconnect in VLSI Packages”,
  *ICCD*, 2005
  - *Best Paper Award*

- “Impedance Matching Techniques for VLSI Packaging”,
  *DesignCon, 2006* (accepted for publication)
Performance Modeling

• Analytical Model To Predict Bus Performance
  – VLSI/CAD integration
  – Quick hand calculations

• We Can Use Package Noise As the Failure Parameter
  – Any noise source can be used as limit
  – Max \( \frac{di}{dt} \) or \( \frac{dv}{dt} \) is extracted and converted to bus throughput
Performance Modeling

• **Bus Notation**
  - Analysis performed on repetitive segment, reducing computation time
  - A scalable framework is used to represent the bus configuration

![Diagram showing bus segments and connectivity with labels for Mutual Capacitance, Mutual Inductance, and Self Inductance, Capacitance]
Performance Modeling

- Use Ground Bounce as Failure Mechanism

\[
V_{gnd-bnc} = \left( \frac{L_{11} \cdot W_{bus}}{N_g} \right) \left( \frac{di}{dt} \right)_k + \sum_{k=-P_L}^{P_L} \left( M_{1(|k|+1)} \right) \left( \frac{di}{dt} \right)_k + \sum_{k=-P_C}^{P_C} \left( \frac{C_{1(|k|+1)} \cdot Z_0^2}{0.8} \right) \left( \frac{di}{dt} \right)_k = p \cdot V_{DD}
\]

- Self Contribution
- Mutual Inductive Contribution
- Mutual Capacitive Contribution
- Noise Limit
Performance Modeling

\[ \text{slewrate} = \left( \frac{dv}{dt} \right) = \left( \frac{di}{dt} \right) \cdot Z_{\text{load}} \]

\[ t_{\text{rise}} = \frac{(0.8) \cdot V_{DD}}{\text{slewrate}} \]
Performance Modeling

:. Datarate

\[ DR_{\text{max}} = \frac{p \cdot Z_0}{(1.5) \cdot (0.8) \cdot \left [ \left( \frac{L_{11} \cdot W_{\text{bus}}}{N_g} \right) + \sum_{k=p_l}^{p_l} M_{1(k|l+1)} + \sum_{k=p_c}^{p_c} C_{1(k|l+1)} \cdot Z_0^2 \right]} \]

\[ UI = (1.5)(\text{trise}) = 1/DR \]

:. Throughput

\[ TP_{\text{max}} = W_{\text{BUS}} \cdot DR_{\text{max}} \]
Performance Modeling

• BGA Wire-Bond Package Simulations

- Model Matches Simulations to 11% for segments greater than 1 bit
- Throughput does not increase linearly as channels are added
Bus Expansion CODEC

- **Encode the Data To Avoid Noise Causing Vector Sequences**
  - Reducing noise allows faster per-pin datarate
  - Throughput is increased even after considering Overhead
  - Bus Expansion CODEC maps on-chip bus size ($m$) into off-chip bus size ($n$)
Bus Expansion CODEC - Constraints

• For Each Possible Noise Source on the Bus, a Constraint is written

1) \( v^j_0 = VDD \rightarrow - P_{bnc} \geq (L/2) \cdot (\# \text{ of } v^j_i \text{ pins} = 1) \leq P_{bnc} \)
2) \( v^j_1 = 1 \rightarrow k_1 \cdot (v^j_2) + k_2 \cdot (v^j_3) \geq P_1 \)
3) \( v^j_1 = -1 \rightarrow k_1 \cdot (v^j_2) + k_2 \cdot (v^j_3) \leq P_{-1} \)
4) \( v^j_1 = 0 \rightarrow - P_0 \leq k_1 \cdot (v^j_2) + k_2 \cdot (v^j_3) \leq P_0 \)

• Each Constraint is Evaluated to Find Illegal Transitions:

\( v^j_1 = 1 = \text{rising} \)  \( v^j_1 = 0 = \text{static} \)  \( v^j_1 = -1 = \text{falling} \)

\begin{align*}
v^j_1 & \quad v^j_2 & \quad v^j_3 \\
1 & \quad 0 & \quad 1 \\
1 & \quad -1 & \quad 0 \\
1 & \quad 1 & \quad 1 \\
1 & \quad -1 & \quad 1 \\
1 & \quad 1 & \quad 0 \\
1 & \quad -1 & \quad -1 \\& \quad 1 & \quad 1 & \quad 1 \\
\end{align*}

violates user-defined “glitch” parameter

violates user-defined “supply” bounce parameter
Bus Expansion CODEC - Algorithm

• The Remaining *Legal* Transitions Construct a Directed Graph

- A closed set of nodes \( S \) must exist such that:
  - \( |S| \geq 2^m \)
  - each vertex \( s \) in \( S \) has at least \( 2^m \) outgoing edges to vertices \( s' \) in \( S \)
Bus Expansion CODEC – Physical Results

• **TSMC 0.13um Synthesis Results**
  - RTL design, synthesized and mapped
  - Segment sizes 2 → 8 implemented
  - Logic, delay, and area evaluated

<table>
<thead>
<tr>
<th>Bus Size (m)</th>
<th>5% (aggressive)</th>
<th>10% (non-aggressive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay (ns)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.170</td>
<td>encoder not required</td>
</tr>
<tr>
<td>4</td>
<td>0.670</td>
<td>0.503</td>
</tr>
<tr>
<td>6</td>
<td>1.150</td>
<td>0.955</td>
</tr>
<tr>
<td>8</td>
<td>1.310</td>
<td>0.983</td>
</tr>
<tr>
<td>Area (um²)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>encoder not required</td>
</tr>
<tr>
<td>4</td>
<td>152</td>
<td>114</td>
</tr>
<tr>
<td>6</td>
<td>614</td>
<td>509</td>
</tr>
<tr>
<td>8</td>
<td>1,181</td>
<td>886</td>
</tr>
</tbody>
</table>
Bus Expansion CODEC – Physical Results

• Xilinx FPGA, 0.35um Implementation Results
  - RTL design implemented
  - Xilinx, VirtexIIPro, FPGA
Bus Expansion CODEC – Physical Results

- Xilinx FPGA, 0.35um Implementation Results
  - RTL design, implemented
  - Segment sizes 2 → 8 measured
  - Logic operation verified
  - Noise Reduced from 16% to 4%

(3 bit, SPG=4:1:1)

<table>
<thead>
<tr>
<th>Bus Size (m)</th>
<th>Noise Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>5% (aggressive) &amp; 10% (non-aggressive)</td>
</tr>
<tr>
<td>2</td>
<td>0.351</td>
</tr>
<tr>
<td>4</td>
<td>1.020</td>
</tr>
<tr>
<td>6</td>
<td>1.450</td>
</tr>
<tr>
<td>8</td>
<td>1.610</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FPGA Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FPGA Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>
Bus Stuttering CODEC

**Intermediate States are Inserted Between Noise Causing Transitions**

- *Stutter* states limit the number of simultaneously switching signals
- The source synchronous clock is gated during stutter state

---

**No Encoding**

**Un-encoded:**

**B→C Vector Sequence Causes Noise Limit Violation**

**w/ Encoding**

**Encoded:**

**B→C Vector Sequence is eliminated using Stutter**
Bus Stuttering CODEC - Algorithm

- Constraints are Evaluated and a Legal Directed Graph is Created

- Directed Graph is Used to Map Transitions Between any Two Vectors
  - A transition path (which may include stutters) exists between any two vectors if:
    - There exists at least two outgoing edges for each vector $v_s \in G$ (including self-edge)
    - There exists at least two incoming edges for each vector $v_d \in G$ (including self-edge)
Bus Stuttering CODEC - Construction

- **Multiple Stutter States can be used**
  - between 0 and $2^{(W_{bus}-1)}$ stutters can be inserted between any two vectors
  - experimental results show that for segments up to 8 bits, more than 3 stutters is rare

- **Overhead**
  - Overhead increases as segments sizes increase
  - Still useful since segments greater than 8 bits are rarely used (SPG=8:1:1)

\[
Overhead = \sum_{k=1}^{2^{(W_{bus}-1)}} \frac{\text{(#_Trans Requiring _k_stutters)} \cdot k}{2^{(2W_{bus})}}
\]
Bus Stuttering CODEC – Physical Results

• **Circuit Implementation**
  - 32 pipeline stages used
  - pipeline reset after 32 idle states (similar to SRIO, HT, and PCI Express)
  - protocol inherently handles pipeline overflow
Bus Stuttering CODEC – Physical Results

• TSMC 0.13um Synthesis Results
  - RTL design, synthesized and mapped
  - Segment sizes 2 → 8 implemented
  - Logic, delay, and area evaluated

<table>
<thead>
<tr>
<th>Bus Size</th>
<th>Noise Limit</th>
<th>Noise Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5% (aggressive)</td>
<td>10% (non-aggressive)</td>
</tr>
<tr>
<td>Delay (ns)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.02</td>
<td>1.99</td>
</tr>
<tr>
<td>6</td>
<td>2.42</td>
<td>2.38</td>
</tr>
<tr>
<td>8</td>
<td>2.85</td>
<td>2.79</td>
</tr>
<tr>
<td>Area (um²)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>311k</td>
<td>310k</td>
</tr>
<tr>
<td>6</td>
<td>362k</td>
<td>345k</td>
</tr>
<tr>
<td>8</td>
<td>382k</td>
<td>368k</td>
</tr>
</tbody>
</table>
Bus Stuttering CODEC – Physical Results

- Xilinx FPGA, 0.35um Implementation Results
  - RTL design, implemented
  - Segment sizes 2 → 8 measured
  - Logic operation verified
  - Noise Reduced from 16% to 4%

<table>
<thead>
<tr>
<th>Bus Size</th>
<th>Noise Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>5% (aggressive) &amp; 10% (non-aggressive)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delay (ns)</th>
<th>Noise Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4.78</td>
</tr>
<tr>
<td>6</td>
<td>5.29</td>
</tr>
<tr>
<td>8</td>
<td>5.89</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FPGA Usage</th>
<th>Noise Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>&lt; 1%</td>
</tr>
<tr>
<td>6</td>
<td>&lt; 1%</td>
</tr>
<tr>
<td>8</td>
<td>&lt; 1.5%</td>
</tr>
</tbody>
</table>

(3 bit, SPG=4:1:1)
Impedance Compensation

- Add Capacitance Near Bond Wire to Reduce Impedance

- Adding additional capacitance lowers the wire bond impedance
- Impedance can be matched to system, reducing reflections

\[ Z_{WireBond} = \sqrt{\frac{L_{WireBond}}{C_{WireBond}}} \]

\[ \Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} \]

Add Capacitance to lower Z

Better Impedance Match results in less reflections
Impedance Compensation

• If the capacitance is close to the wire bond, it will alter its impedance

- Electrical lengths less than 20% of risetime are treated as lumped elements
- For modern dielectrics, anything within 0.15” of wire bond is lumped
Static Impedance Compensator

- Capacitor values chosen prior to fabrication

- Equal amounts of capacitance are used on-chip and on-package

\[ Z_{\text{WireBond}} = \sqrt{\frac{L_{WB}}{C_{WB} + C_{pkg} + C_{MIM}}} = 50\Omega' S \]
Static Impedance Compensator

- Time Domain Analysis (TDR)

<table>
<thead>
<tr>
<th>Length $L_{wb}$</th>
<th>$\Gamma_{No-Comp}$</th>
<th>$\Gamma_{MIM-Comp}$</th>
<th>$\Gamma_{Device-Comp}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mm</td>
<td>4.5%</td>
<td>0.05%</td>
<td>0.5%</td>
</tr>
<tr>
<td>2 mm</td>
<td>8.7%</td>
<td>0.4%</td>
<td>1.2%</td>
</tr>
<tr>
<td>3 mm</td>
<td>12.7%</td>
<td>1.3%</td>
<td>2.4%</td>
</tr>
<tr>
<td>4 mm</td>
<td>16.4%</td>
<td>2.7%</td>
<td>4.1%</td>
</tr>
<tr>
<td>5 mm</td>
<td>19.8%</td>
<td>4.8%</td>
<td>6.0%</td>
</tr>
</tbody>
</table>

Worst Case: 5mm

No Static Capacitance $= 19.8\%$

w/ Static Capacitance $= 4.8\%$
Dynamic Impedance Compensator

- **Pass Gates are used to switch in on-chip capacitors**
  
  - Pass gates connect on-chip capacitance to the wire bond inductance
  - Pass gates have control signals which can be programmed after fabrication

\[
Z_{WireBond} = \sqrt{\frac{L_{WB}}{C_{WB} + C_{Comp}}} = 50\Omega' s
\]
Dynamic Impedance Compensator

- **Time Domain Analysis (TDR)**

<table>
<thead>
<tr>
<th>Length (mm)</th>
<th>( \Gamma_{No-Comp} )</th>
<th>( \Gamma_{MIM-Comp} )</th>
<th>( \Gamma_{Device-Comp} )</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mm</td>
<td>4.5%</td>
<td>1.0%</td>
<td>1.0%</td>
<td>001</td>
</tr>
<tr>
<td>2 mm</td>
<td>8.7%</td>
<td>1.8%</td>
<td>1.3%</td>
<td>011</td>
</tr>
<tr>
<td>3 mm</td>
<td>12.7%</td>
<td>3.6%</td>
<td>3.0%</td>
<td>100</td>
</tr>
<tr>
<td>4 mm</td>
<td>16.4%</td>
<td>4.3%</td>
<td>3.3%</td>
<td>110</td>
</tr>
<tr>
<td>5 mm</td>
<td>19.8%</td>
<td>6.0%</td>
<td>5.0%</td>
<td>111</td>
</tr>
</tbody>
</table>

**Worst Case: 5mm**

- No Dynamic Capacitance = 19.8%
- w/ Dynamic Capacitance = 6.0%
3) Advantages Over Prior Techniques
Performance Modeling and Bus Sizing

• **Currently Packages are Modeled Using SPICE**
  - Analog simulators are computationally expensive\[BSIM, BPTM]\)
  - Time of simulation reduces the number of configurations to be evaluated [Agilent Ft. Collins]

• **Model is Linear in the size of the bus**
  - Fast computation is enabled using key assumptions
  - More configurations can be evaluated, which expands usefulness
  - Narrows hundreds of configurations into 2 or 3 for SPICE evaluation

• **Cost is Considered**
  - Analog simulators do not account for cost
  - This adds even more time to analysis
Bus CODECs to Avoid Package Noise

• Current Approaches Have Physical Limitations
  - Operate by reducing \( \frac{di}{dt} \) or skewing transitions [pipeline_damping, Multi-Level]
  - Reducing \( \frac{di}{dt} \) will ultimately limit performance
  - Skewing data increases data invalid window, will ultimately limit performance

• Our CODECs operate above the physical layer
  - Only data vectors are altered
  - Off-chip drivers are left unchanged, no skewing is necessary
  - This allows usefulness up to higher frequencies
  - This also allows implementation in various process and package technologies
Impedance Compensation

• Currently, Package Interconnect is Not Addressed
  - Only primary impedance is terminated (i.e., the PCB T-line) [HS_Design, MGT]
  - No broadband solution exists

• Our Techniques Target Package Directly
  - Impedance of wire bond or bumping can be addressed
  - Broadband operation suited well for digital VLSI

• Static Compensator
  - Developed using embedded construction, no cost
  - Simple and requires no active circuitry

• Dynamic Compensator
  - Accounts for process variation by allowing programmability after fabrication
4) Broader Applications of this Work
The Move Toward FPGAs

• 80% of Design Starts Have FPGAs
The Move Toward FPGAs

- **FPGA Business Model**
  - Single design is packaged in multiple technologies
  - This enables multiple performance price-points
  - Designer cannot optimize for particular package

- Performance Modeling
- Noise Reduction CODECs
- Impedance Compensators

Cost / Performance
Power Minimization

• **Power is Predicted to Limit Moore’s Law**

- Large amounts of power are consumed in the off-chip drivers
- CODECs can remove patterns which result in noise violations
- CODECs can also remove patterns with high power consumption

> 100W [ITRS]
Internet Fabric

- **Network Congestion Slows Internet Performance**
  - CODECs can remove patterns which result in noise violations
  - Can extend CODECs to remove redundant patterns in streaming A/V
Backplanes and Connectors

- **All Interconnect Has Parasitic Inductance and Capacitance**
  - Backplanes are popular to provide design segmentation and scalability
  - Connectors are present in all digital designs
  - Modeling, CODECs, and Compensation can be applied to backplanes/connectors
Questions?