



Difficulty of Predicting Interconnect Delay in a Timing Driven FPGA CAD Flow

*Valavan Manohararajah, Gordon R. Chiu,
Deshanand P. Singh & Stephen D. Brown*



Outline

- Motivation
- Previous work
- Preliminaries
- Placement self prediction
- A simple early timing model
- Prediction for physical synthesis
- Upper bound on physical synthesis gains
- Conclusions

Motivation

- Most of the delay in an FPGA is in its interconnect
 - Academic estimates 60%
 - Real FPGAs even higher.
- Prediction of delay before P & R is a difficult yet important problem.
- Two potential uses:
 - Guide early timing driven restructuring ops.
 - Early user feedback for design changes.

Previous Work

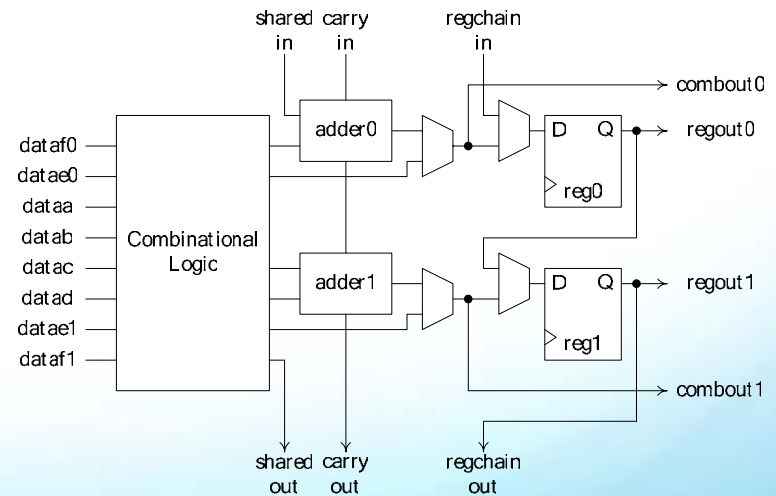
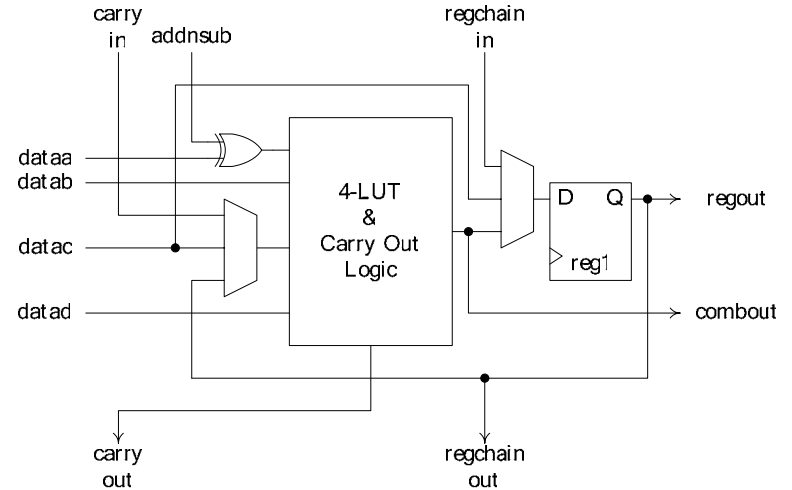
- Number of methods exploring wirelength prediction:
 - C. Sechen. *Average Interconnection Length Estimation for Random and Optimized Placements*. ICCAD 1987.
 - T. Hamada et al. *A Wire Length Estimation Technique Utilizing Neighborhood Density Equations*. TCAD August 1986.
 - S. Bodpati et al. *Pre-Layout Estimation of Individual Wire Lengths*. SLIP 2000.
 - S. Balachandran et al. *A-Priori Wirelength and Interconnect Estimation Based on Circuit Characteristics*. SLIP 2003.
- Might argue that wirelength \sim delay.
- But timing driven tools are focused on the optimization of timing critical connections.

Our Work

1. Show that state-of-the-art placement has a large amount of inherent randomness.
2. Show that a simple delay model is able to predict some aspects of timing with almost as much accuracy as placement itself.
3. A method of overcoming the limited predictability for a physical synthesis tool is presented.
4. Establish an upper bound on the gains possible by using an ideal predictor.

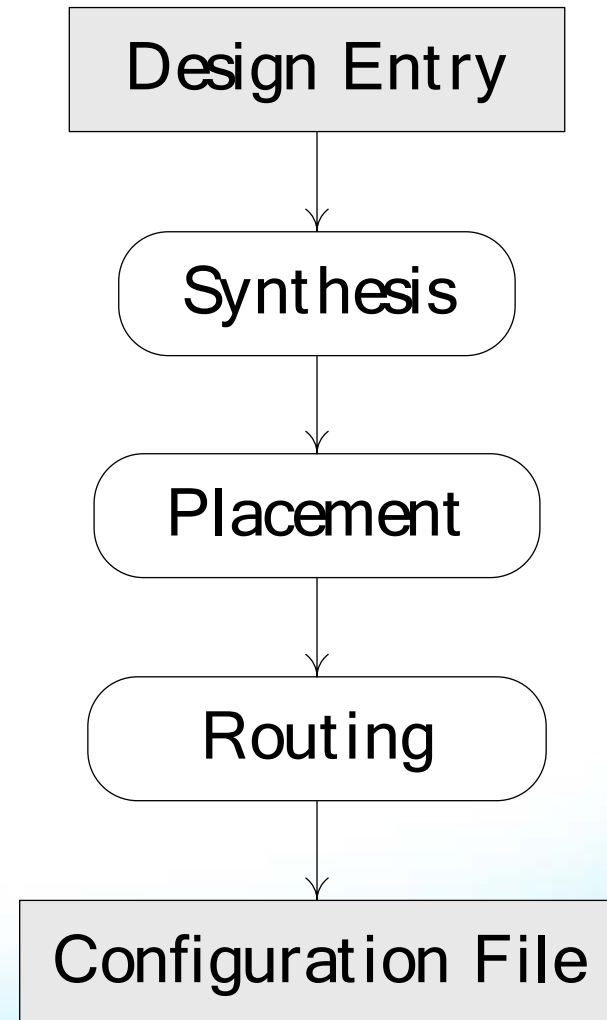
Architecture

- Used Stratix & StratixII chips.
- Contains:
 - I/O elements
 - Logic array blocks
 - DSP blocks
 - Memory elements



CAD Flow

- Design entry:
 - HDL, Schematics
- Synthesis:
 - Optimize gate level network.
- Placement:
 - Determine a location for each logic element.
- Routing:
 - Determine wires needed to hookup the logic elements.
- Configuration file:
 - Used to program the FPGA.



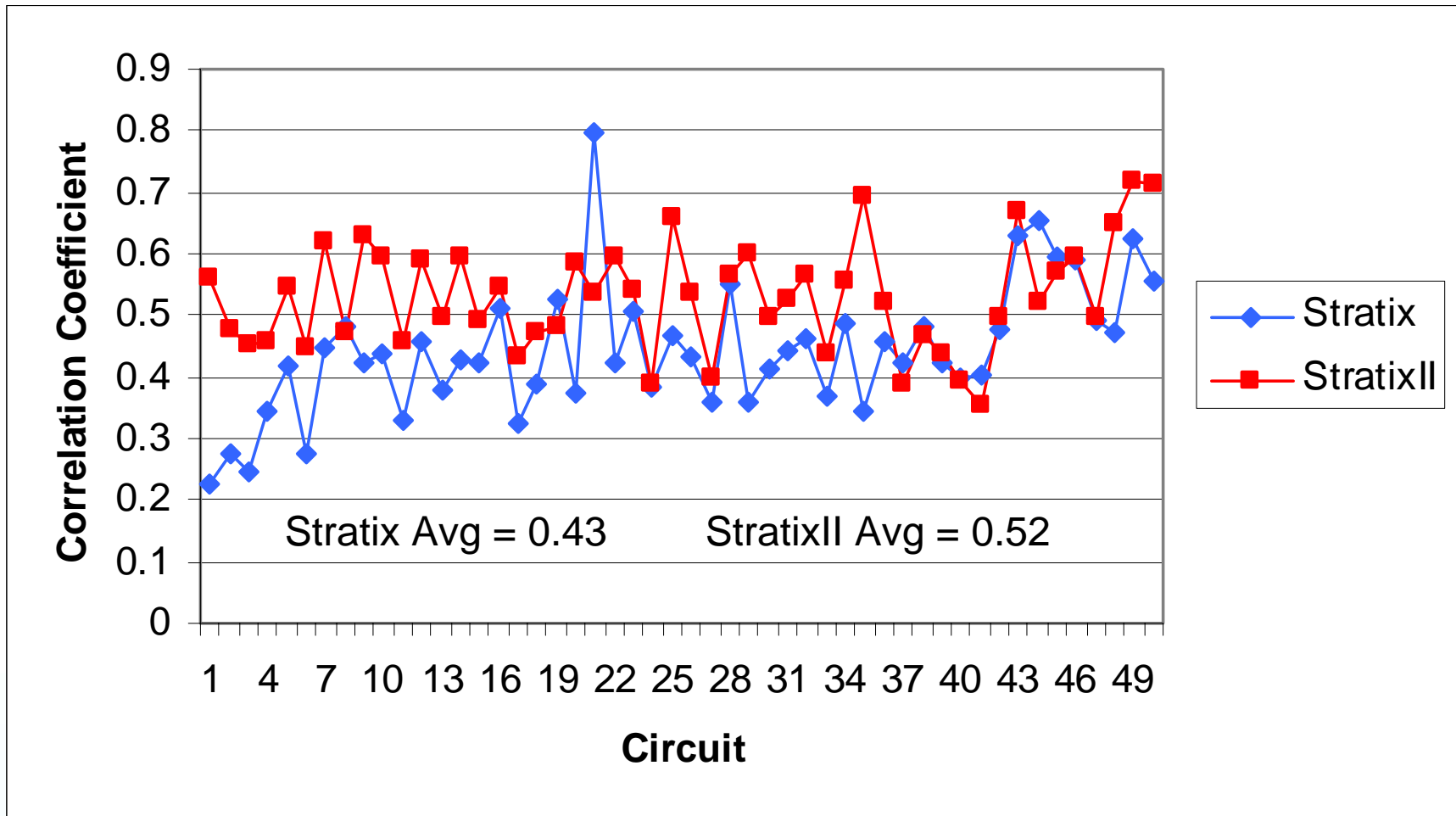
Timing Analysis

- Given delay within elements and delay of connections between elements, timing analysis establishes the **slack** of each connection.
- **Slack** = amount of delay that can be added before the connection becomes critical.
- **Critical** = the connection is on path that is equal to the longest path in the circuit.
- **Criticality** = A value between 0 and 1 that indicates the relative importance of each connection to overall circuit timing.

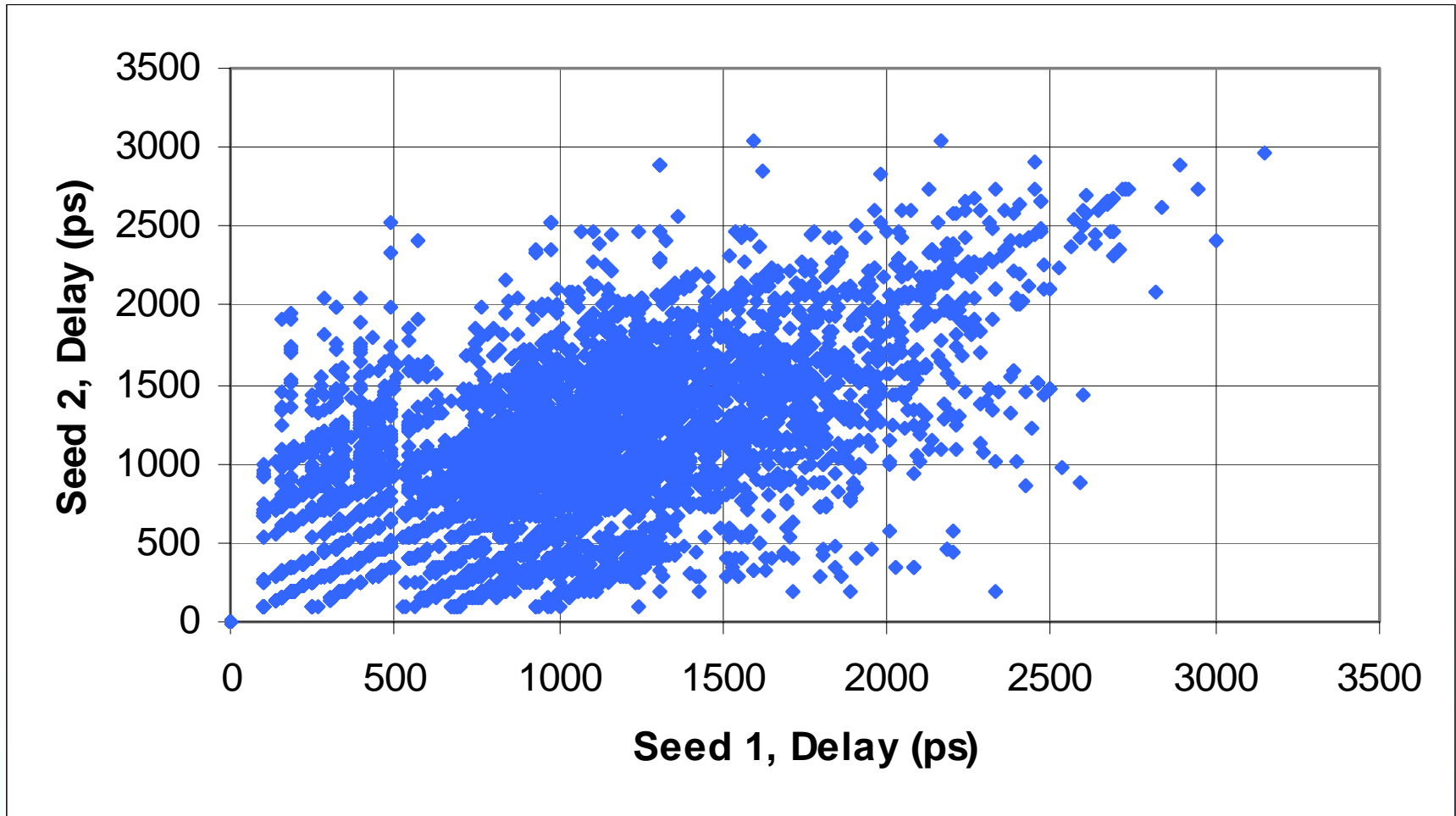
Placement, Self Prediction

- Behavior of the QuartusII placer can be changed by choosing a different seed.
- Determine an upper bound on the predictability:
 - Run placer with two different seeds.
 - First seed is to be a predictor for the second.
- Measure predictability using correlation coefficient (r^2).
- Focus on the critical connections:
 - Fitter pays more attention to the critical connections.
 - Observe connections with a criticality of 0.5 or higher.

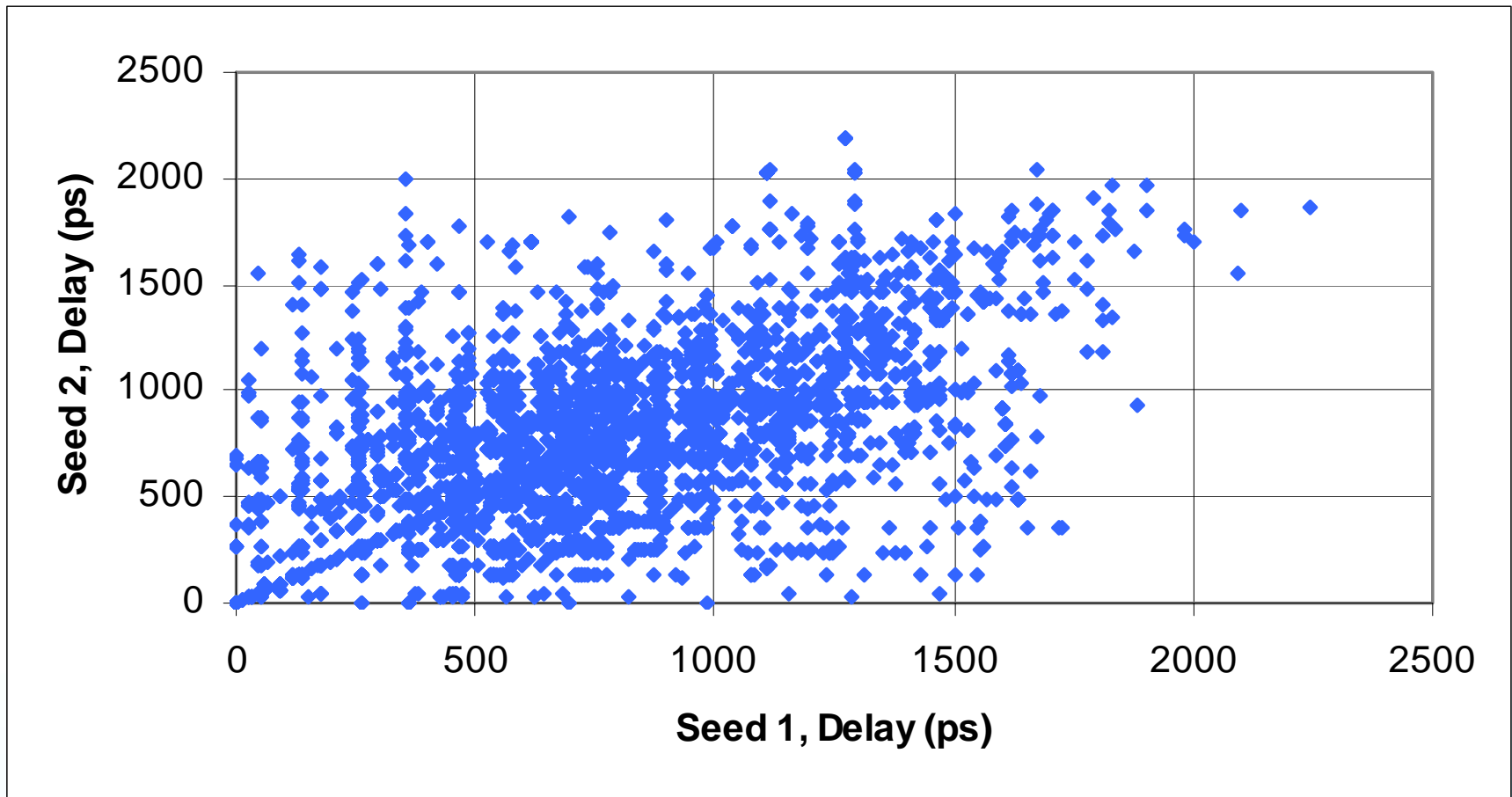
r² of Delays



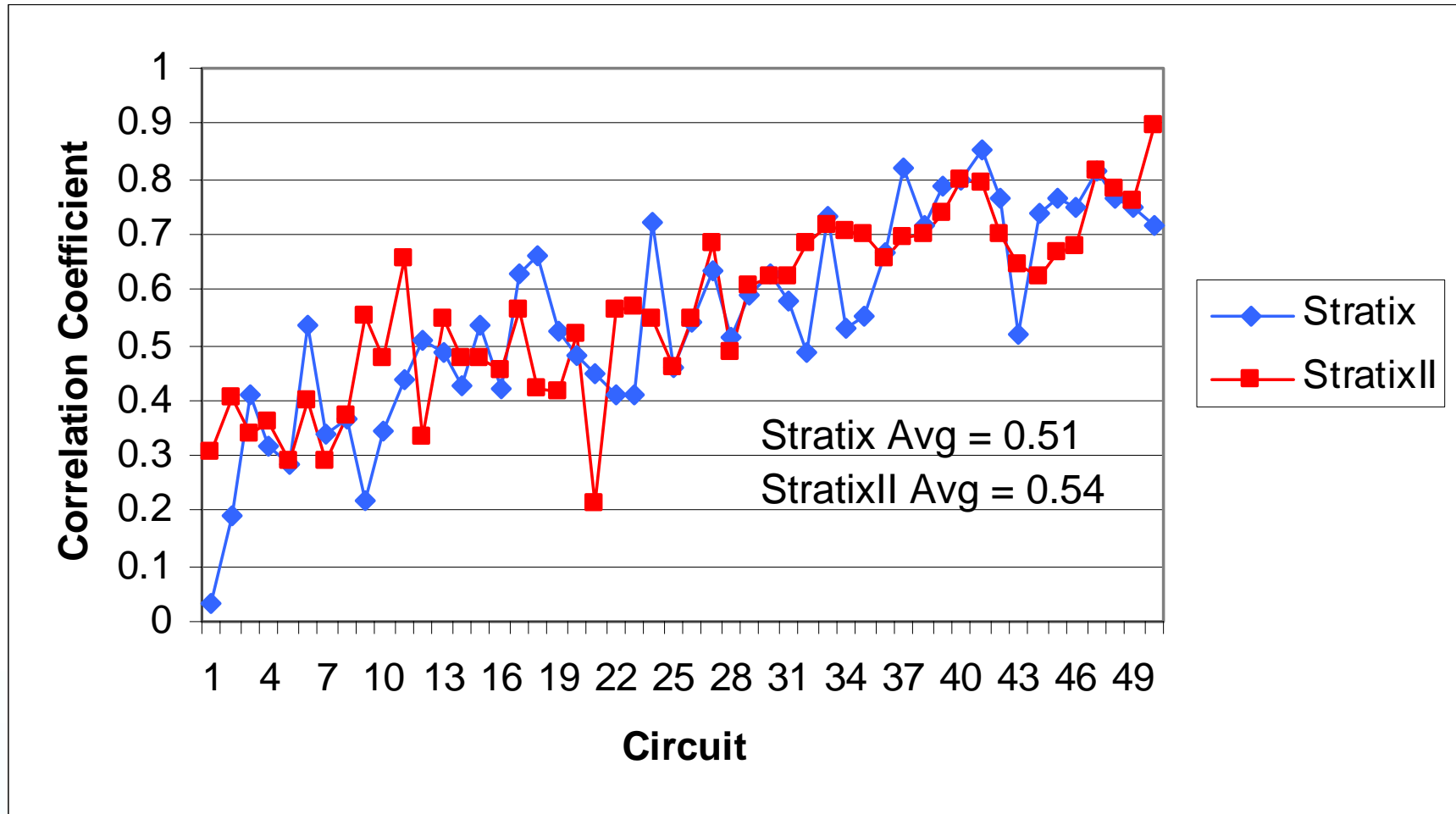
Delays, Seed #1 vs Seed #2



Delays, Seed #1 vs Seed #2



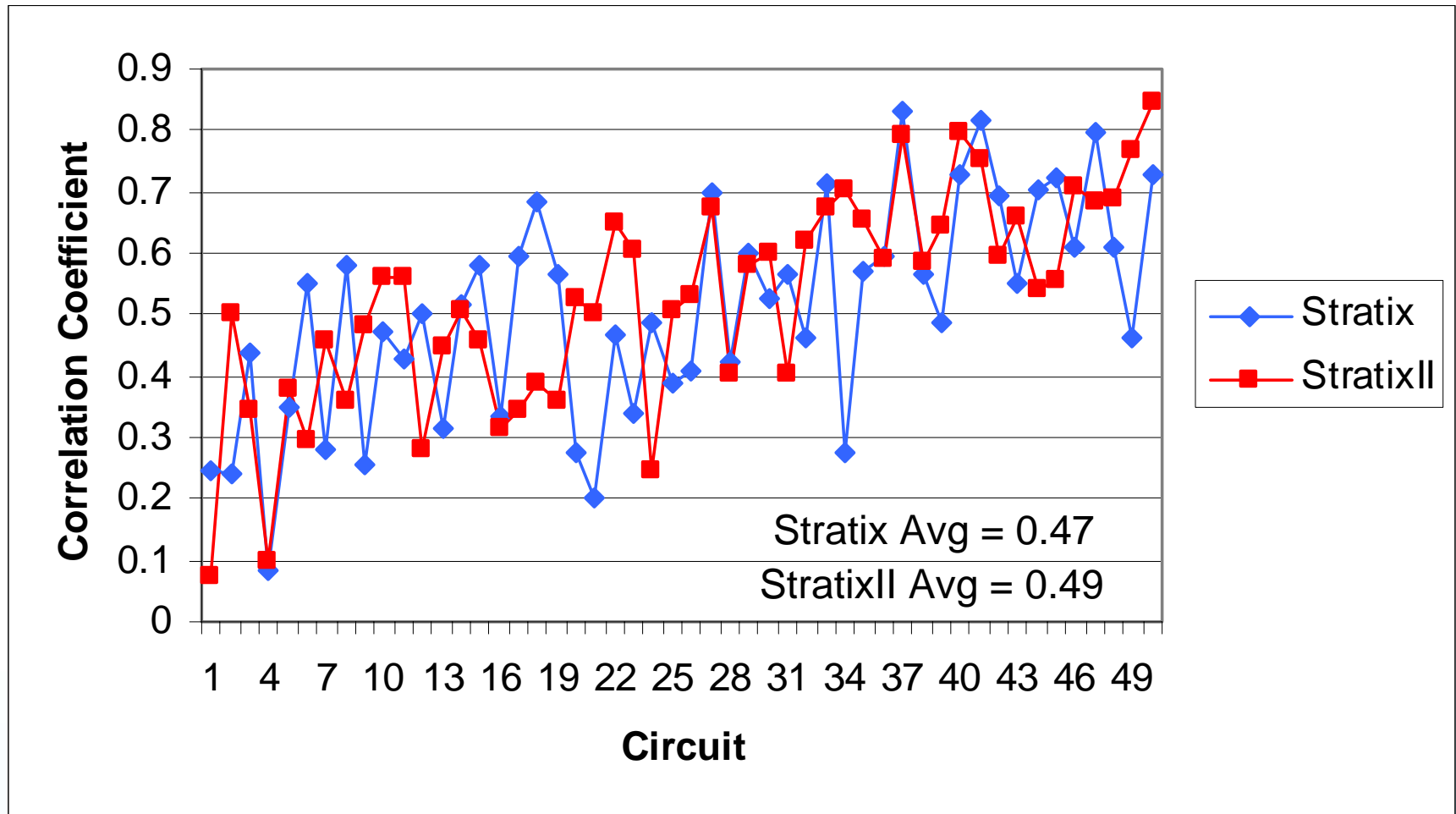
r² of Criticalities



A Simple Early Timing Model

- For each **connection type** gather observed delays over several circuits.
- Connection type identified by:
 - Driving node type.
 - Driving port type.
 - Driven node type.
 - Driven port type.
- Compute a weighted delay for each connection type, placing more emphasis on on faster observed delays.

Simple Timing Model Correlation



The Need for Early Physical Synthesis

- Physical Synthesis consists of post-placement timing driven optimizations.
 - retiming, rewiring, decomposition, replication
- Substantial compile time overhead
 - need to legalize post-placement netlist (incremental placement)
- Can we reduce the compile time overhead?
 - With good delay prediction, can perform many optimizations pre-placement.

Early and Late Physical Synthesis

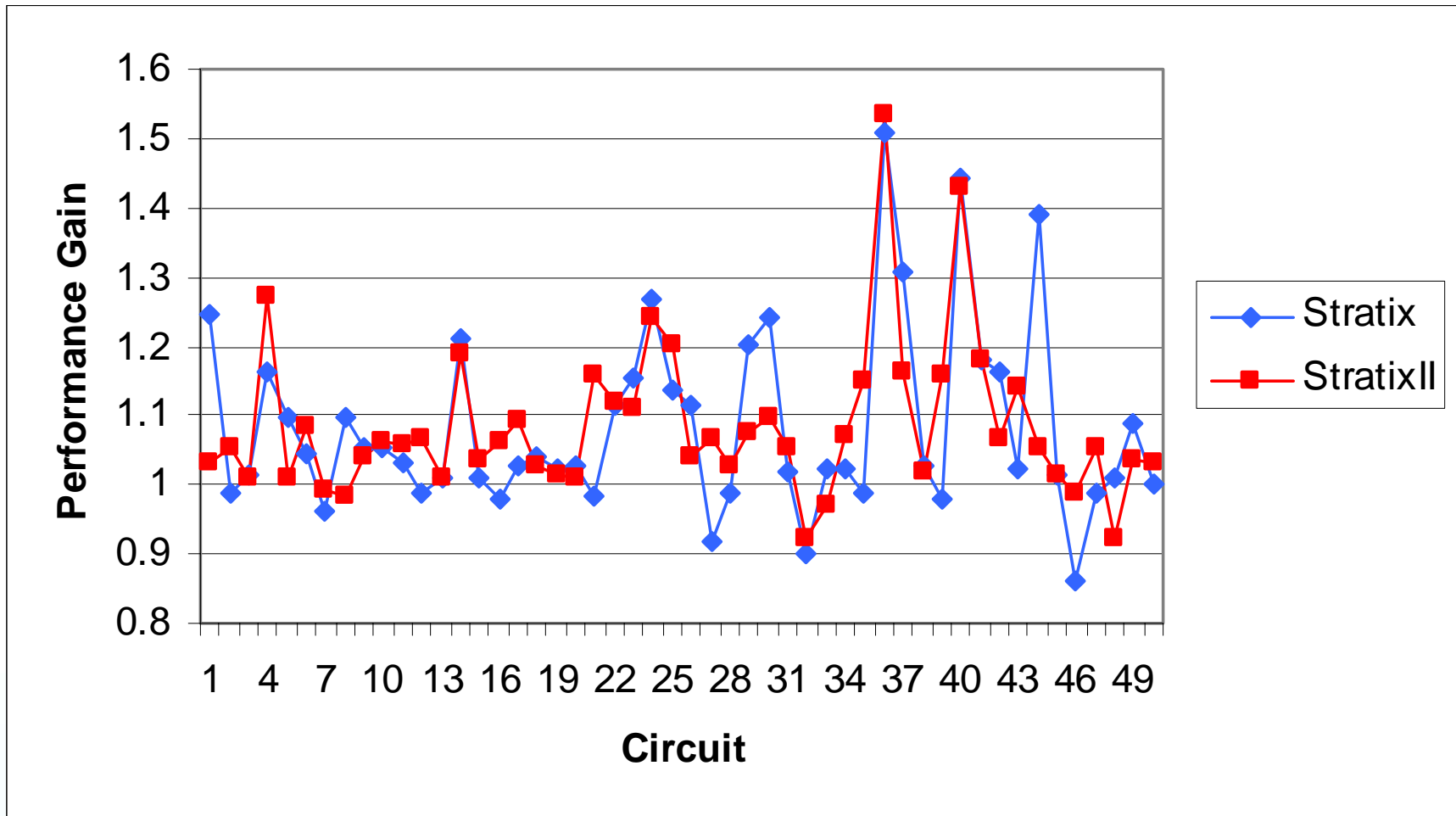
- Split physical synthesis into *early* and *late*.
- *early*: pre-placement optimizations
 - Use simple delay model to predict delays
 - Allow more optimizations.
- *late*: post-placement optimizations
 - Use real post-placement delays
 - Restrict optimizations to minimize need for costly legalization.

Results from Early and Late Physical Synthesis

Physical Synthesis Flow	% Performance Gain	% Compile Time Gain
Early Only	8%	60%
Early + Late*	13%	150%
Late Only	11%	200%

Addition of Early Physical Synthesis Increases Performance and Decreases Compile Time

Early Physical Synthesis Results

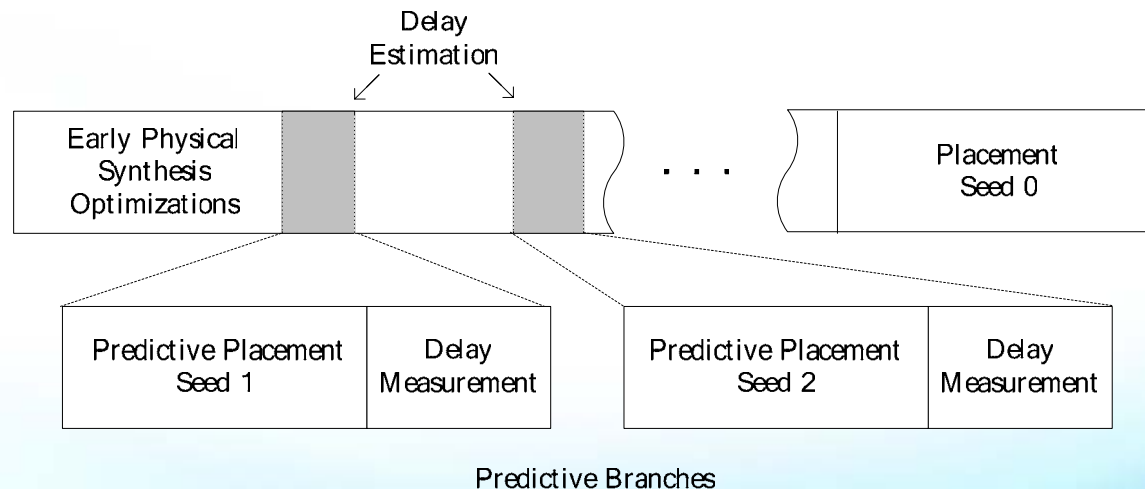


An Upper Bound on Early Physical Synthesis Gain

- Upper bound on predictability previously established.
 - Best possible predictability is seed-to-seed prediction.
- Determine performance gains if an "upper bound" predictor is used
- Devise experiment to measure gains from "upper bound" predictor
 - Compare gains with those achieved with the simple early timing model.

An Upper Bound Predictor for Interconnect Delay

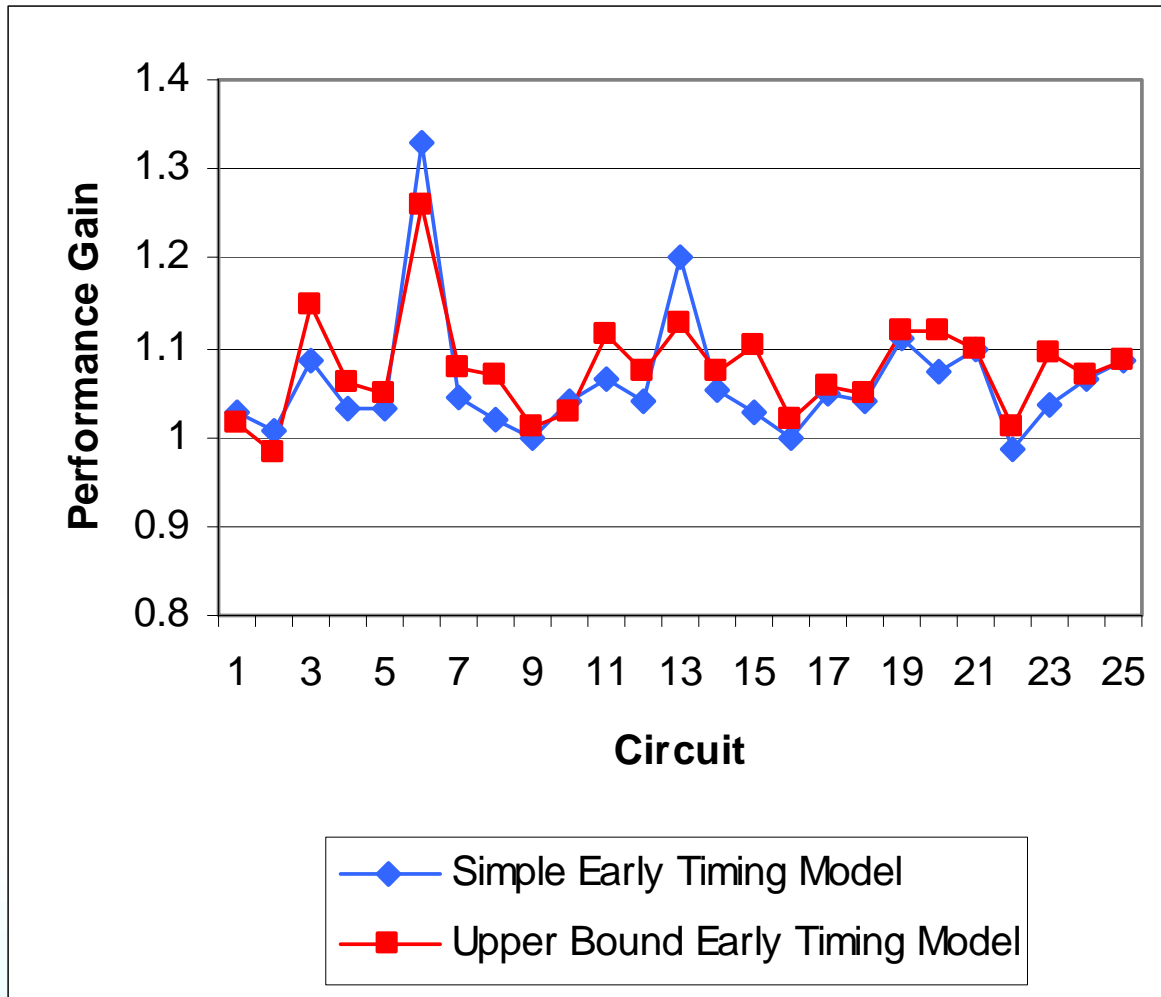
- Use placement (with different seed) as predictor model
 - Predictor is as good as seed-to-seed prediction:
the upper bound
- Periodic timing analyses during early physical synthesis
 - For each, spawn prediction branch and perform placement
 - Use post-placement delays to drive future optimizations



Upper Bound Predictor Results

- Subset of designs tested with new flow
 - Flow very slow due to repeated placements
- 7.6% performance gain from Upper Bound flow
 - compare with 6.0% with simple early timing model
 - 76% of circuits improve with Upper Bound flow
- Limited gains available from improving early timing model

Physical Synthesis using the Upper Bound Predictor



Conclusions

- Large amount of inherent randomness in an FPGA placement algorithm
- Simple delay model can predict interconnect criticalities well
 - Almost as well as running placement itself
 - With sufficient accuracy to perform timing-driven optimizations
- Even with predictor achieving the upper bound in predictability, only small additional performance gains are available

ALTERA®

Thank you. Questions?

