Switching Activity Analysis and Pre-Layout Activity Prediction for FPGAs

Jason H. Anderson and Farid N. Najm

ACM/IEEE Int'l Workshop on System Level Interconnect Prediction (SLIP)
Monterey, CA, April 5-6, 2003

Dept. of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario, Canada

Motivation

- Today’s largest FPGAs are “hot”
  - consume watts of power
    - Xilinx Virtex-II CLB: 5.9 μW/MHz [Shang02]
    - Modest design: 2500 CLBs, 100 MHz → 1.5 W
- Optimize FPGA power consumption: reduce cooling/packaging costs, new apps, better reliability
- Characterize, then optimize

FPGA Power Dissipation

- Power breakdown:
  - – Majority is dynamic
  - – Interconnect dominates:
    - Xilinx Virtex-II: 50-70% of power dissipated in interconnect [Shang02];
      similar results: [Poon02, Kusse98]
  - – Average dynamic power:

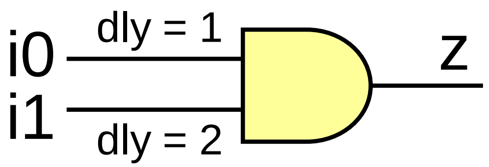
$$P_{avg} = \frac{1}{2}\sum_{i\in nets} C_{i} \cdot f_i \cdot V^2$$

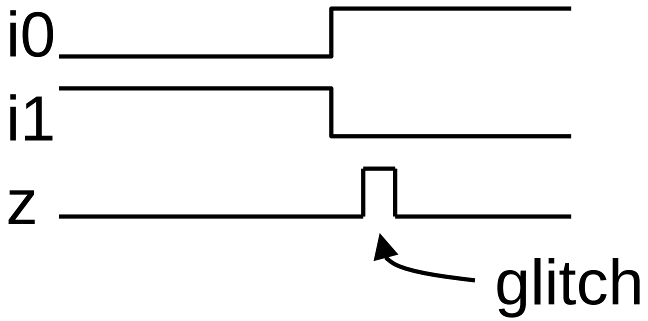
supply voltage

capacitance toggle rate (switching activity)

Switching Activity

- Different views:
  - – zero delay activity: all dlys are zero
  - – logic delay activity: logic dlys only
  - – routed delay activity: both logic/routing dlys
- Delays introduce glitches: spurious transitions that consume power





Motivation

- Activity analysis (**part 1 of this work**):
  - Study extent of activity change due to glitches
  - FPGA delays dominated by interconnect → severe glitching in this technology?
  - Low-power CAD based on zero delay activities → valid for FPGAs?

Motivation

- State-of-the-art FPGAs can implement complex systems with millions of gates
  - Design *teams*, not just individuals
  - Increasingly long design cycle
- Early, high-level power estimation: minimize design time & cost

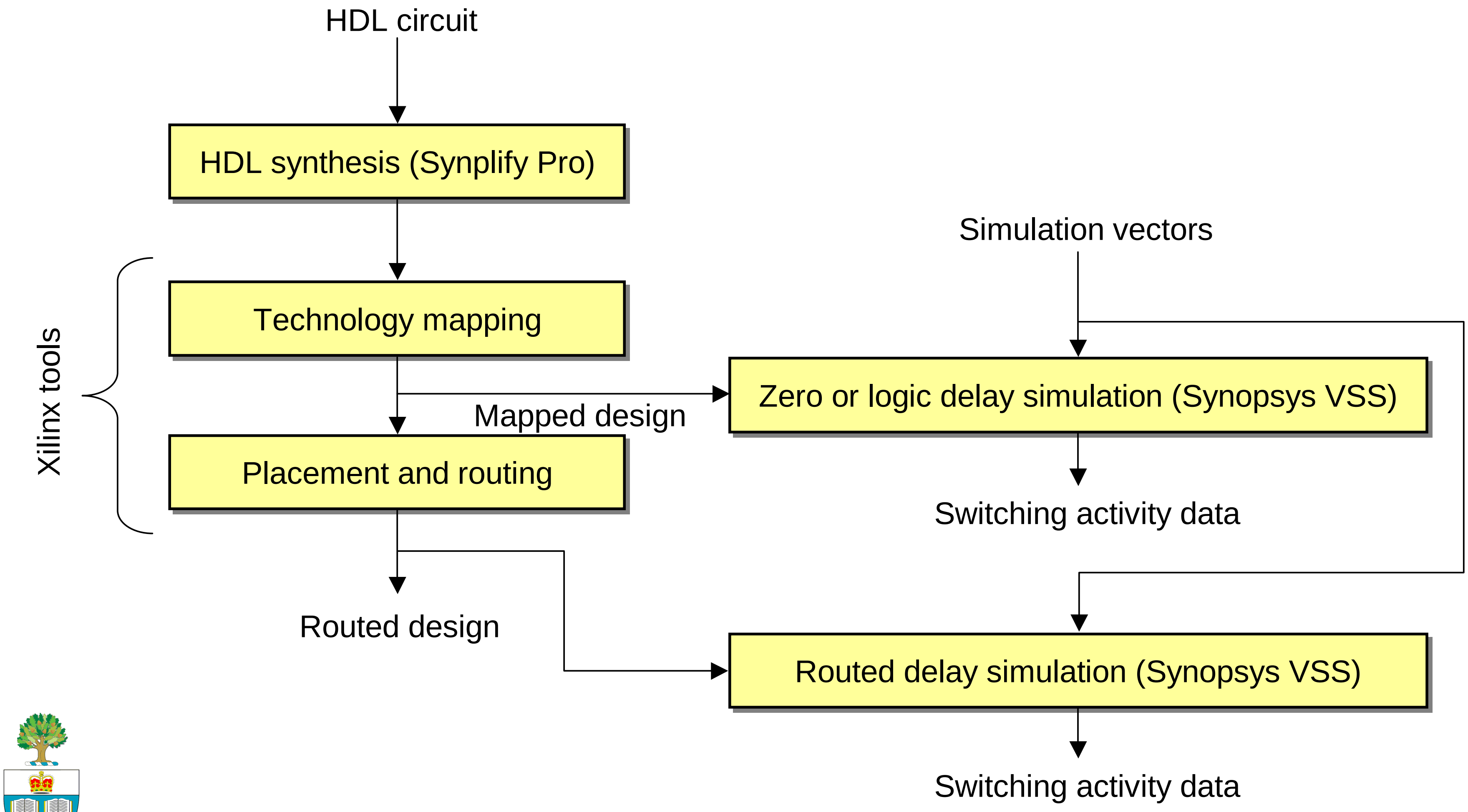
Motivation

- Layout is most time-consuming part of FPGA CAD flow.
- Pre-layout power estimation requires:
  - Net capacitance prediction
  - Net activity prediction (**part 2 of this work**)

Activity Analysis

- Simulation-based approach
- Map 14 circuits into Xilinx Virtex-II
- Simulate with zero'ed delays,
  logic delays, routed delays
- 2 vectors sets: high or low input activity
  - high (low) activity vector set:
    each input has 50% (25%) probability of toggling between vectors

Activity Analysis Flow



HDL circuit

HDL synthesis (Synplify Pro)

Xilinx tools

Technology mapping

Mapped design

Placement and routing

Routed design

Simulation vectors

Zero or logic delay simulation (Synopsys VSS)

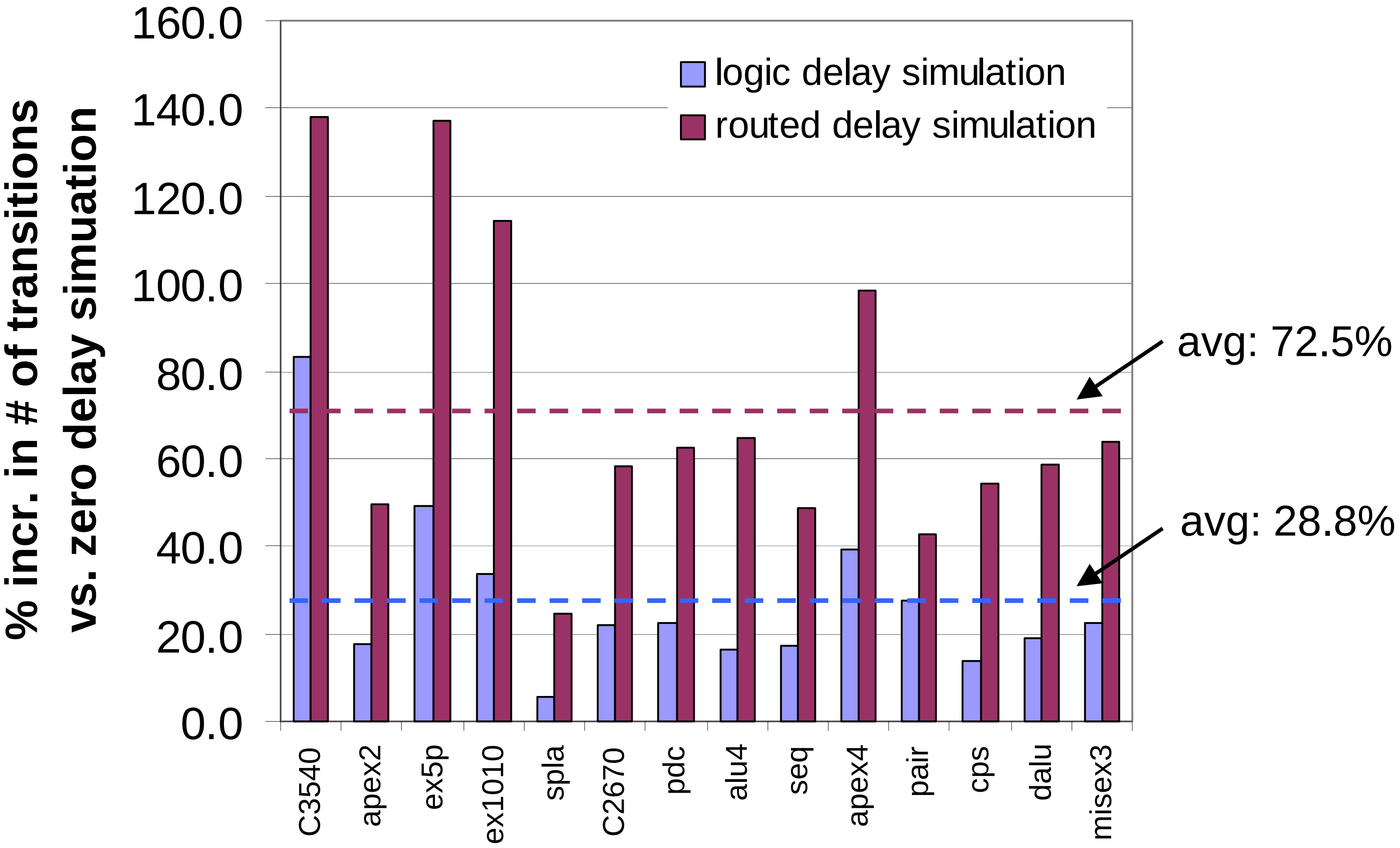
Switching activity data

Routed delay simulation (Synopsys VSS)

Switching activity data

Effect of Glitching on Transition Count

High activity vector set results:

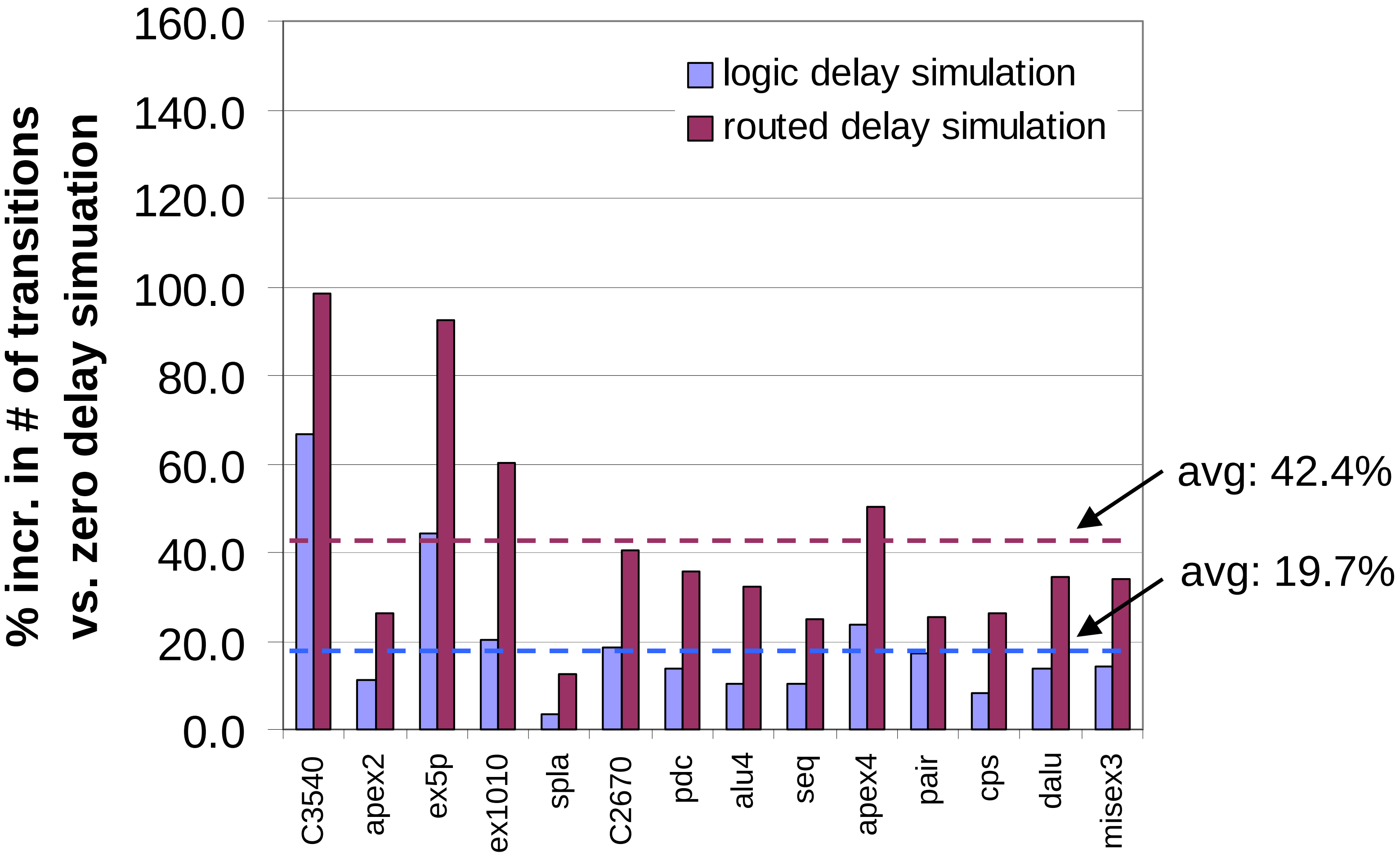


Activity Analysis

- Substantial activity increase when routing delays are accounted for
  - Accounting for logic delays is not enough -- interconnect dominates delay
  - High activity vector set:
    - act. incr. zero → logic: avg: 28%, max: 84%
    - act. incr. logic → routing: avg: 34%, max: 61%

Effect of Glitching on Transition Count

Low activity vector set results:



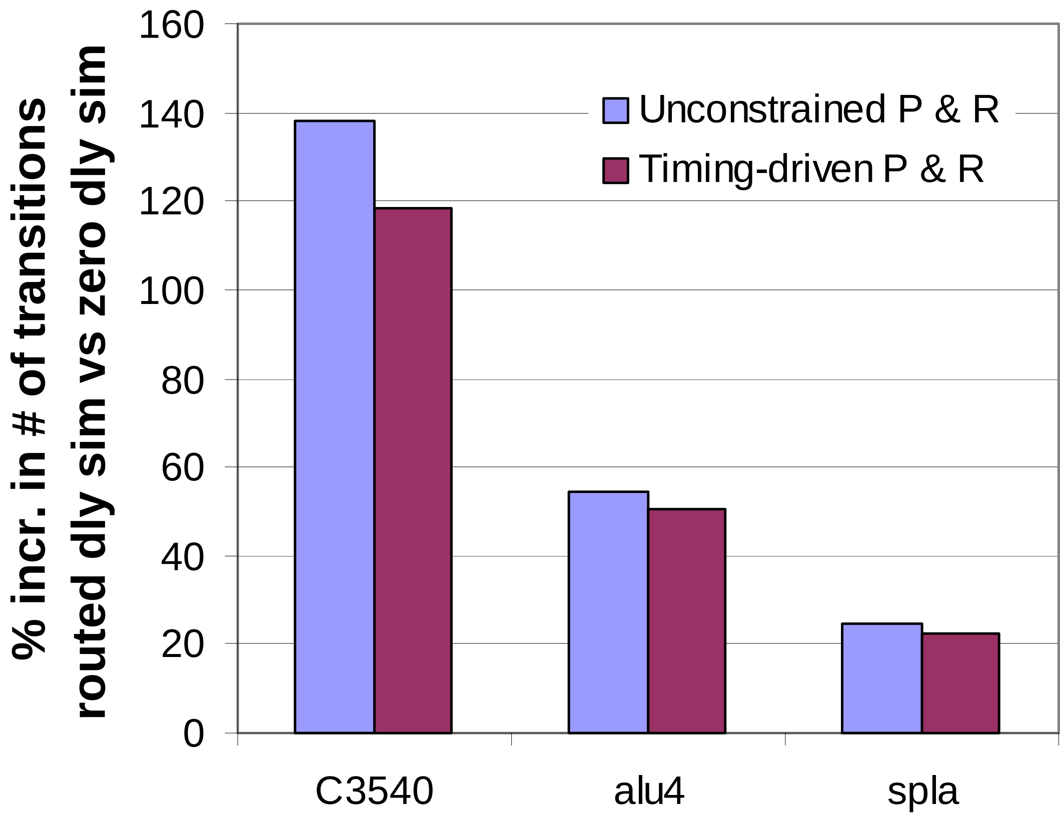
Activity Analysis

- Low activity vector set glitching 1/2 to 2/3 as severe as high activity vector set
  - Fewer inputs switch simultaneously → fewer simultaneous transitions on different paths to net
  - act. incr. zero → logic: avg: 20%, max: 66%
  - act. incr. logic → routing: avg: 19%, max 35%

Effect of Delay Optimization

- Previous results: P & R run without performance constraints
- Timing-driven P & R may lead to smaller interconnect delays → less glitching?

Effect of Delay Optimization



- Glitching reduction from timing-driven P & R is not that substantial

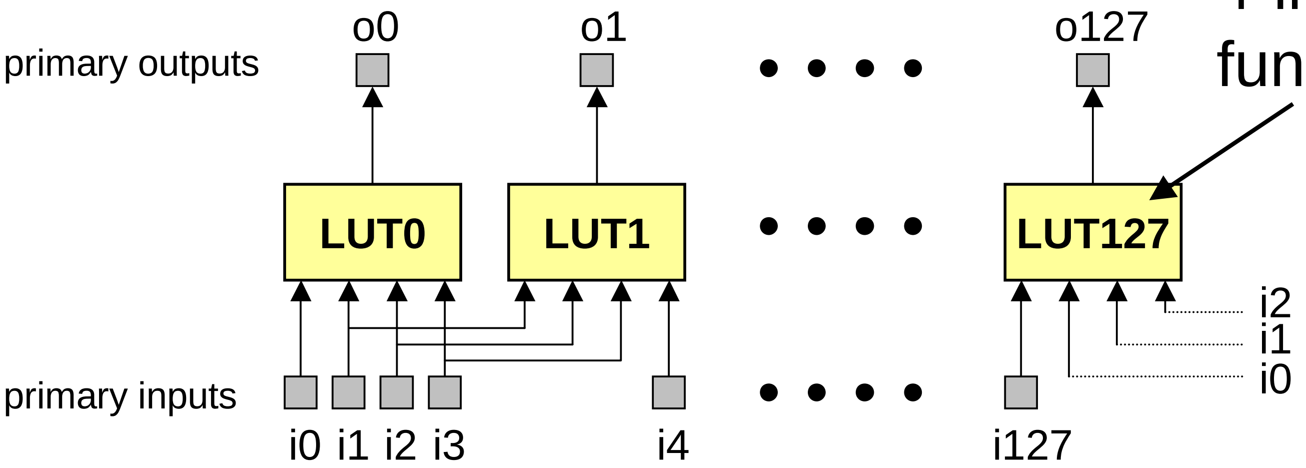
Activity Prediction

- Problem difficulty:
  - How “hard” is the prediction problem?
  - What degree of accuracy can be expected?

- Gauge “noise” in the prediction problem using a specially-designed circuit

Problem Difficulty

- Regular circuit:



each LUT implements 4-input AND function

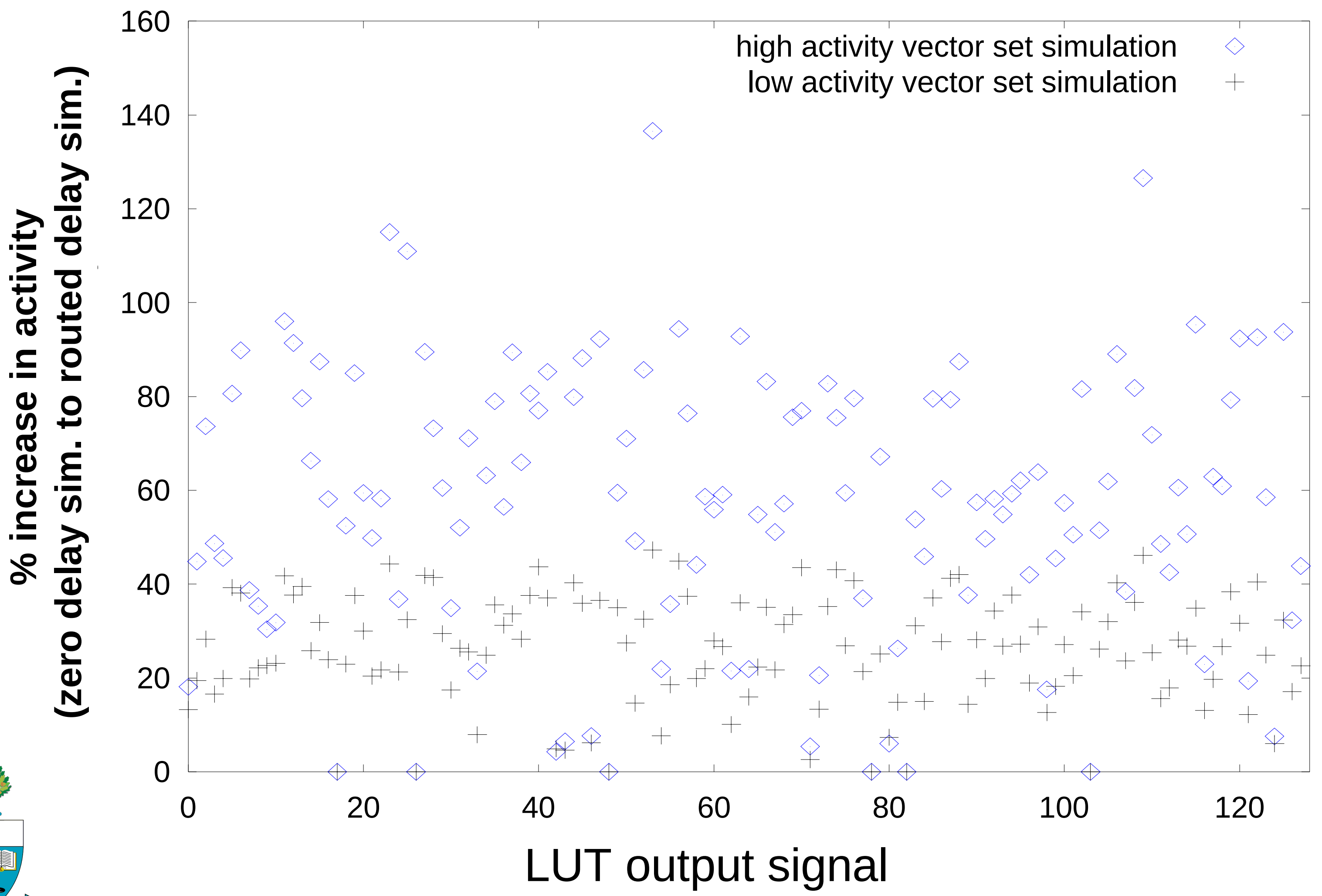
- Has structural & functional regularity

Problem Difficulty

- Implement “regular” circuit in Virtex-II
- Analyze activity increase on LUT output signals from zero to routed delay sim.

- Variability in increase (across LUTs) due to delays known only after layout:
  - routing delays
  - different input-to-output LUT delays
- Represents noise we cannot predict

Problem Difficulty



Problem Difficulty

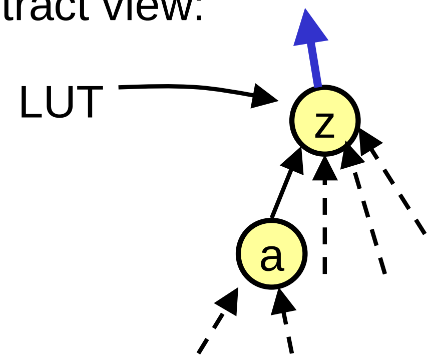
- Variability in activity increase significant:
  - 0-40% (low activity vector set)
  - 0-100% (high activity vector set)

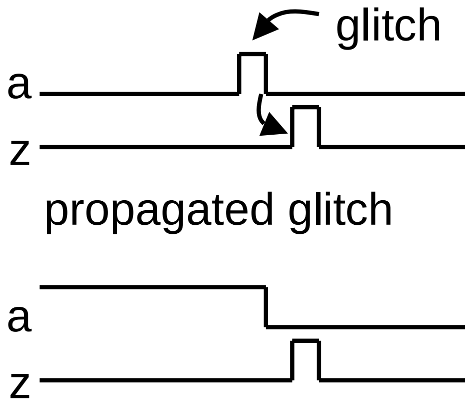
- Accurate pre-layout activity prediction for FPGAs is a difficult problem

Activity Prediction

- Predict net glitching using zero (or logic) delay activity, circuit properties
- Idea: glitches propagated or generated

Abstract view:





propagated glitch

generated glitch

$$predict(z) = \alpha \cdot gen(z) + \beta \cdot prop(z) + \phi$$

Generated Glitches

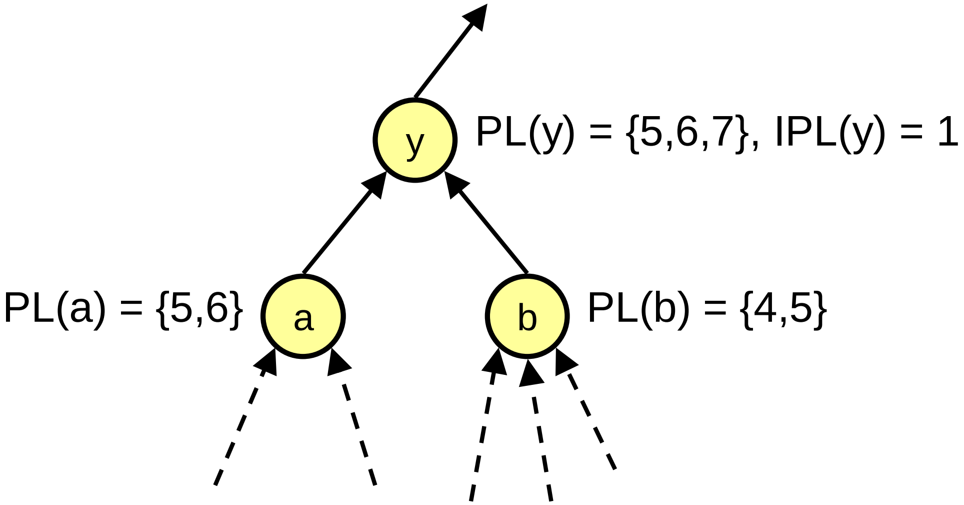
- FPGA logic elements are uniform, have equal drive capability
- Buffered routing switches → connection delay approx. fanout independent
- Predict pre-layout path delay using path length (# of LUTs)

- Unequal path delays lead to glitches

Generated Glitches

- Let PL(x) = set of path lengths to node x
- Define # of path lengths introduced by node y:

$$IPL(y) = \min_{x_i \in inputs(y)} \{|PL(y)| - |PL(x_i)|\}$$



Generated Glitches

$$gen(y) = IPL(y) + \gamma \cdot \underset{\uparrow}{depth(y)}$$

depth of node driving net y

- Depth term included since glitching likely to be worse for “deeper” nodes

Propagated Glitches

- Propagate term uses notions of Boolean difference & static probability
- Consider logic function: $y = f(x_1, x_2, \ldots, x_n)$
- Boolean difference of *y* w.r.t. x_i =

$$\frac{\partial y}{\partial x_i} = f_{x_i} \oplus f_{\overline{x_i}}$$

function *f*(…) with x_i replaced by 1

function *f*(…) with x_i replaced by 0

Propagated Glitches

- Key: $\frac{\partial y}{\partial x_i} = 1 \rightarrow$ transition on x_i will cause transition on y

- Static probability: fraction of time logic signal is in “1” state

$P\left(\frac{\partial y}{\partial x_i}\right)$: probability a transition on x_i will result in transition on y

- Relevant to whether a glitch on x_i will become a glitch on y

Propagated Glitches

$$prop(y) = \frac{\sum_{x_i \in inputs(y)} P\left(\frac{\partial y}{\partial x_i}\right) \cdot predict(x_i) \cdot za(x_i)}{\sum_{x_i \in inputs(y)} P\left(\frac{\partial y}{\partial x_i}\right) \cdot za(x_i)}$$

- $za(x_i)$ = zero delay activity of x_i
  - – replace with logic dly activity (if available)

Experimental Methodology

- Divide 14 circuits into 2 groups: characterization circuits and test circuits

1) Tune model for specific CAD flow & device using characterization circuits
2) Apply model to predict activity in test circuits

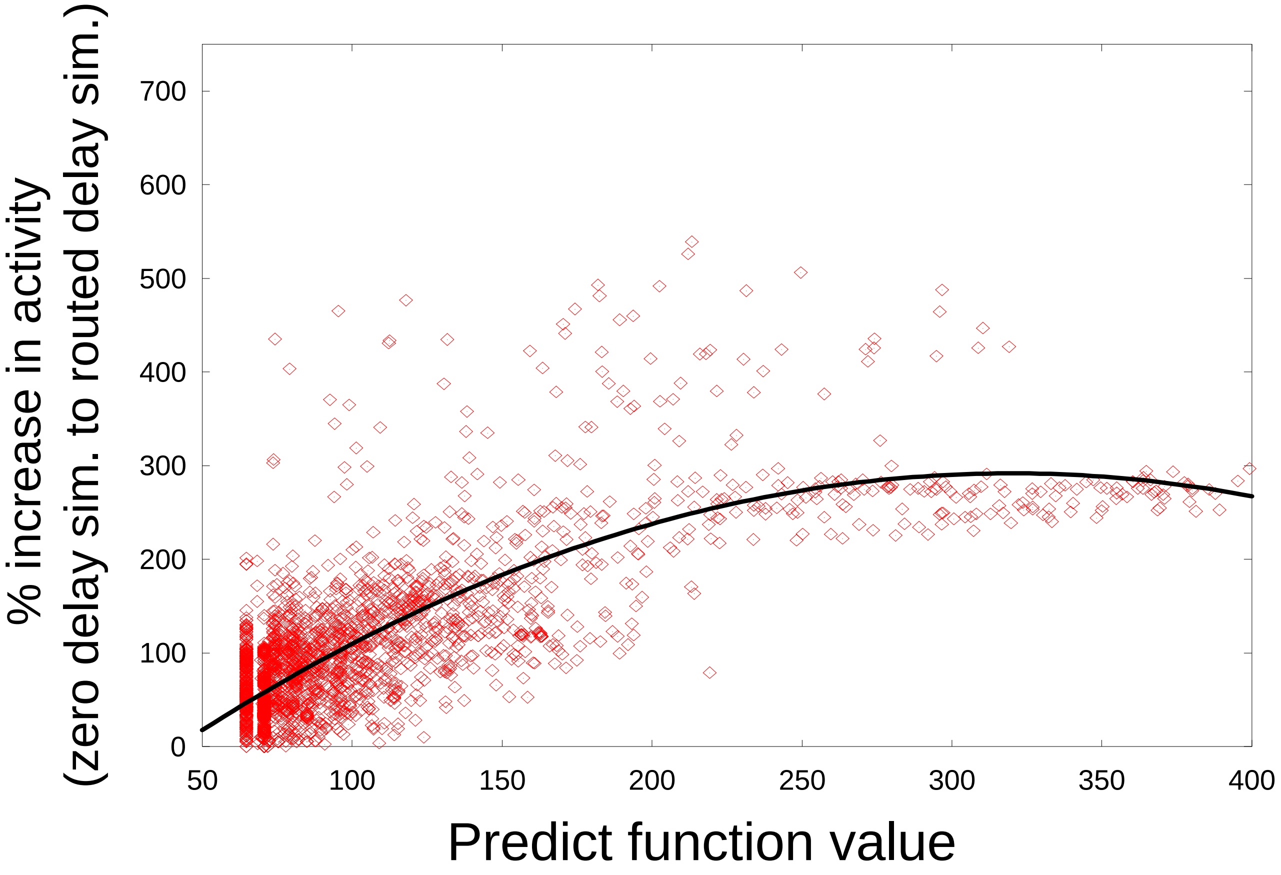
Experimental Methodology

- Two prediction scenarios:
  - predict routed dly activity from zero dly act.
  - predict routed dly activity from logic dly act.

- Static probability, zero/logic activity extracted from simulation
  - parameters can also be computed using probabilistic approaches

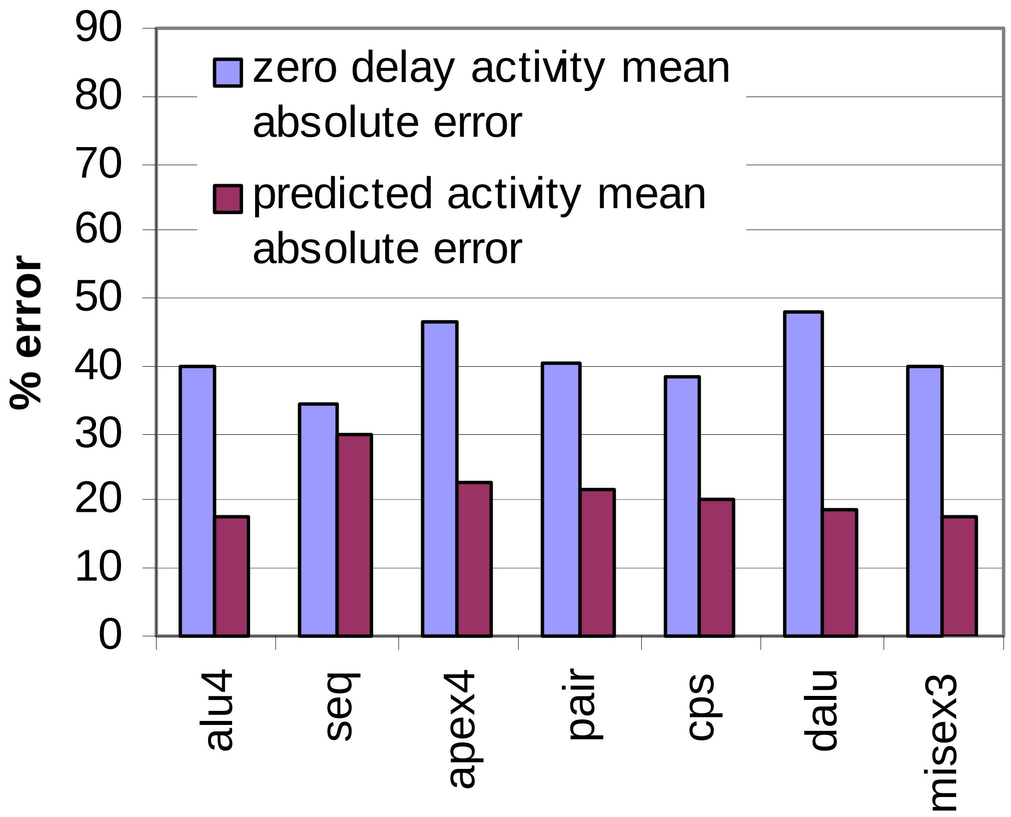
Model Tuning

- High activity vector set simulation of characterization circuits



Results

Prediction from *zero* delay activity data:



zero dly act routed dly act

🟦 $$err = \sum_{n \in nets} \left| \frac{za(n) - ra(n)}{ra(n)} \right|$$

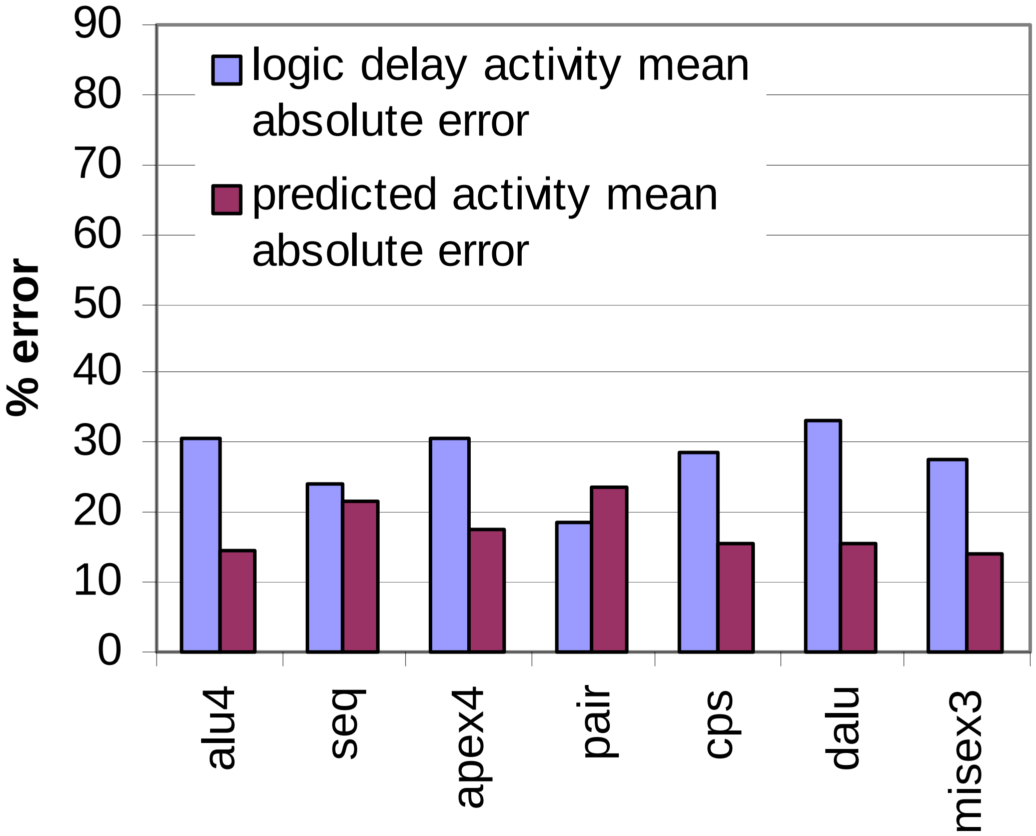
🟪 $$err = \sum_{n \in nets} \left| \frac{pa(n) - ra(n)}{ra(n)} \right|$$

predicted act

- mean error reduced by factor of 2 for most circuits

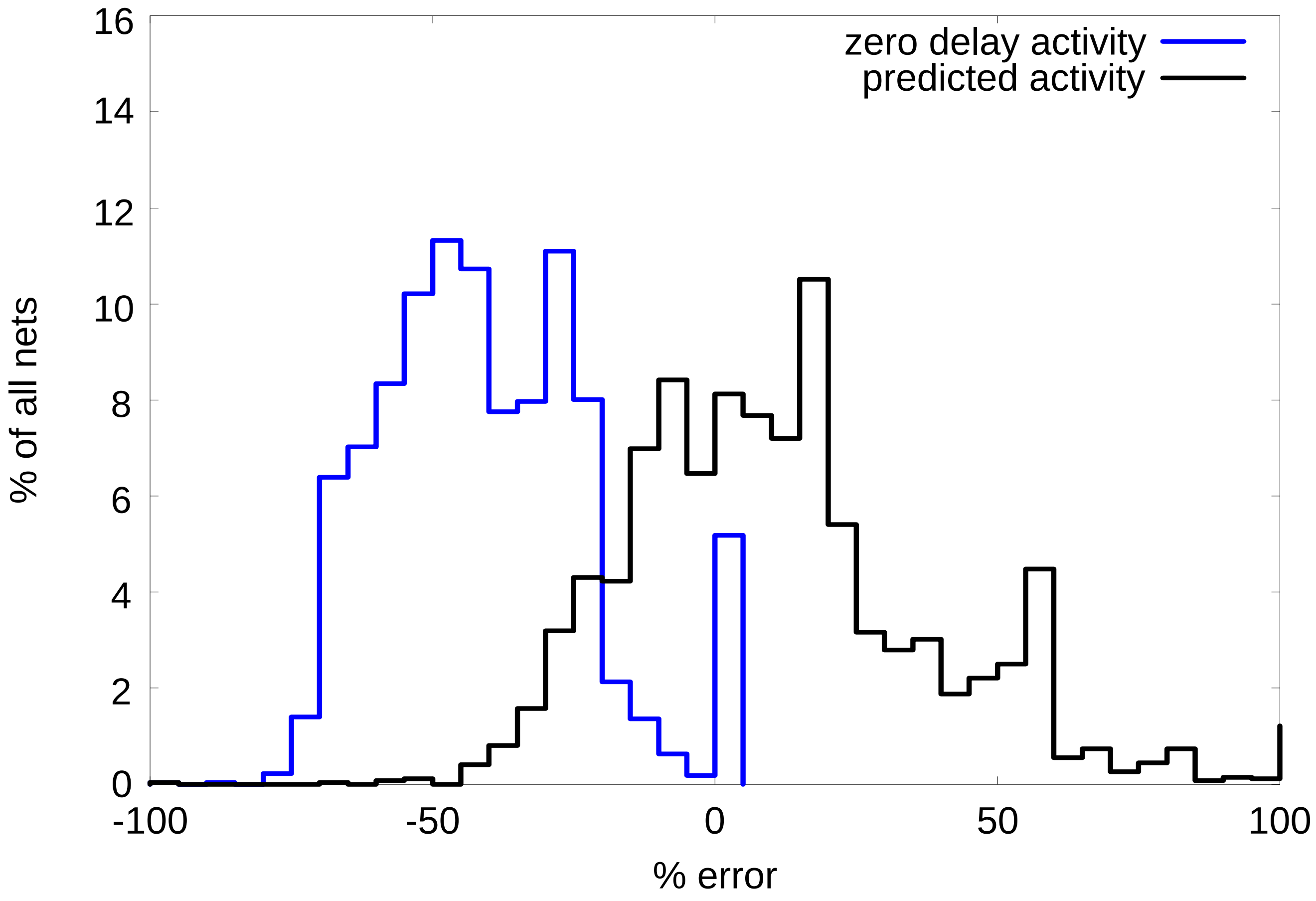
Results

Prediction from *logic* delay activity data:



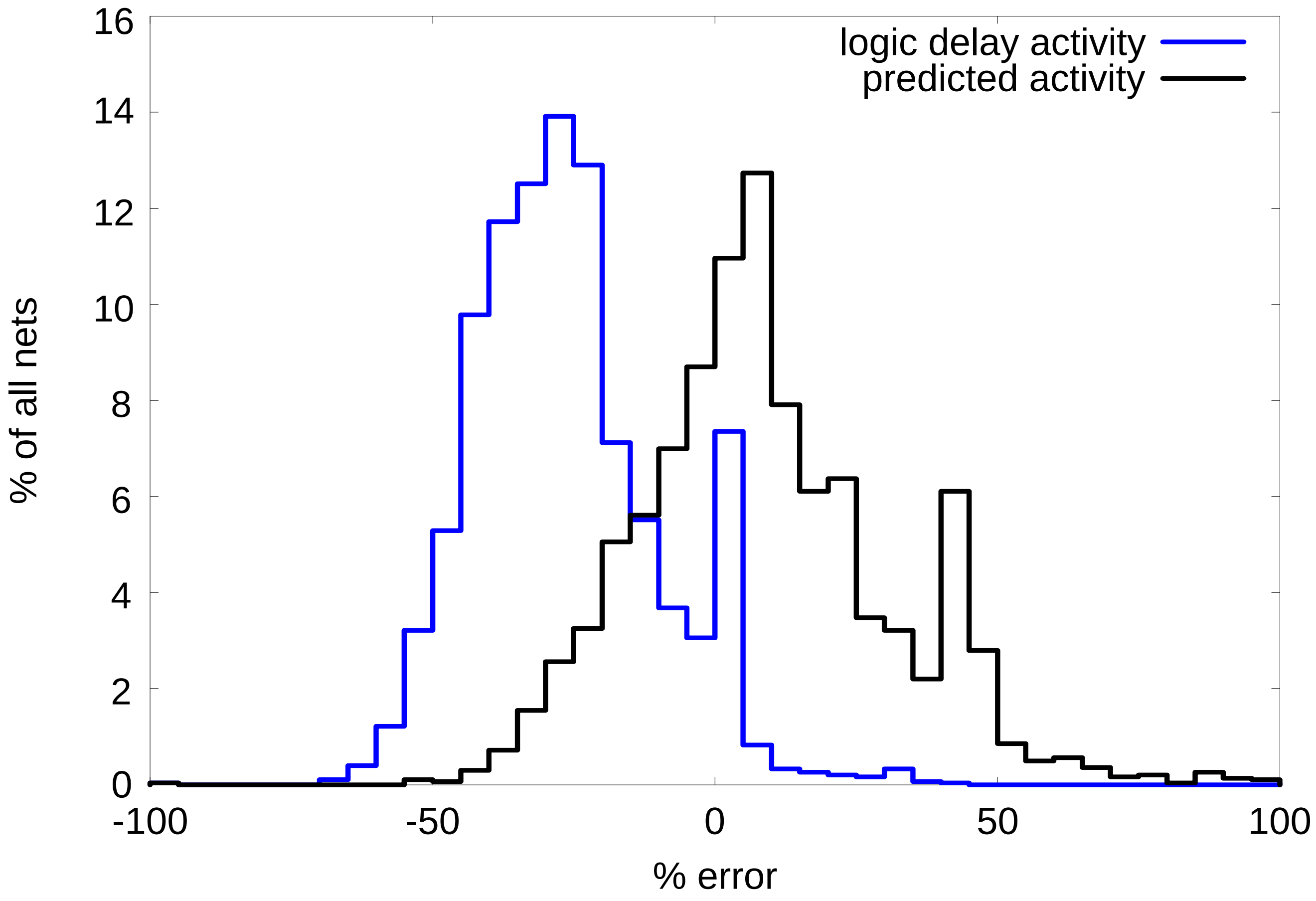
Results

Error histogram for zero delay activity, predicted activity:



Results

Error histogram for logic delay activity, predicted activity:



Results Summary

- Mean absolute error in activity reduced by factor of 2 for many circuits
- Zero/logic delay activities have one-sided error bias
  - Will consistently underestimate power
- Prediction model: one-sided error bias is eliminated
  - Better avg. power estimates

Summary

- Switching activity analysis:
  - Differences between zero, logic, routed delay activity can be significant
  - Glitching severity depends on input activity
- Pre-layout activity prediction:
  - A difficult problem
  - Demonstrated prediction approach based on circuit structure/functionality
  - Mean activity error reduced,
    one-sided error bias eliminated