ECE ILLINOIS
Department of Electrical and Computer Engineering

A Routing Approach to Reduce Glitches in Low Power FPGAs

Quang Dinh, Deming Chen, Martin Wong

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

This research was supported by Altera and NSF

Outline

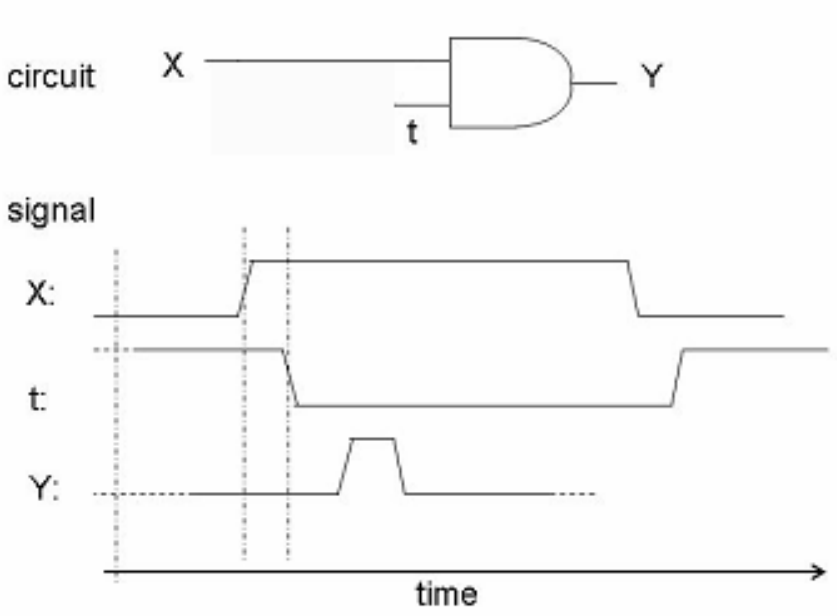
- Introduction
- Background
- Routing Approach
- Path-Finding Algorithm
- Results

Dynamic Power and Glitch

- Dynamic Power
  - 67% of total power (Stratix II FPGAs)

$$P_{Dynamic} = 0.5 \cdot f \cdot {V_{dd}}^2 \cdot \sum_{i=1}^{n} C_i S_i$$

- Glitch
  - Up to 60% of total dynamic power



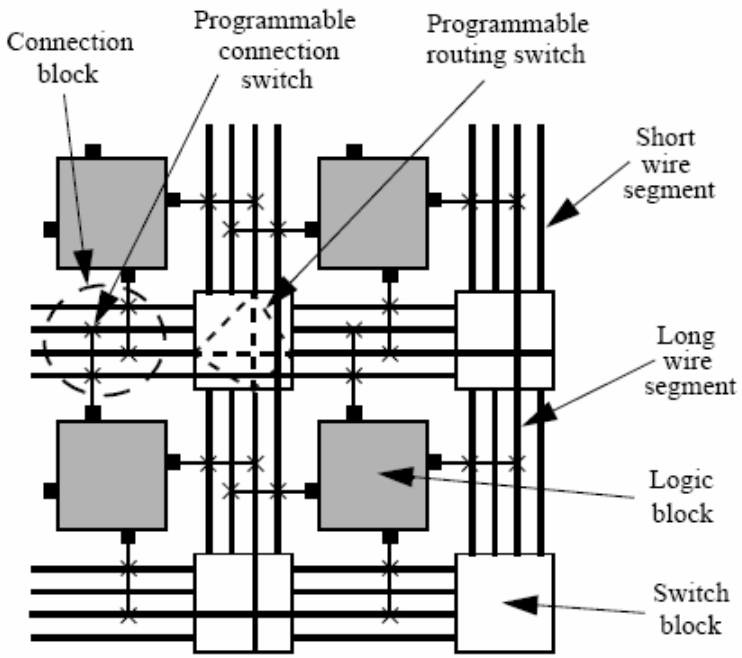
- Glitch Reduction
  - Lower Dynamic Power

Related Works

- High-Level
  - Logic decomposition [Monteiro et al., 1998]
  - RTL synthesis [Raghunathan et al., 1999]
- Not applicable to FPGA
  - Gate freezing [Benini et al., 2000]
  - Delay insertion [Raghunathan et al., 1999]
- FPGA specific
  - GlitchMap [Cheng et al., 2007]
  - GlitchLess [Jamoureux et al., 2007]

FPGA Interconnect

- Island-Style FPGA
- Interconnect delay is more significant than logic delay
- Programmable interconnects with fixed delays
- Rich, under-used interconnect resource

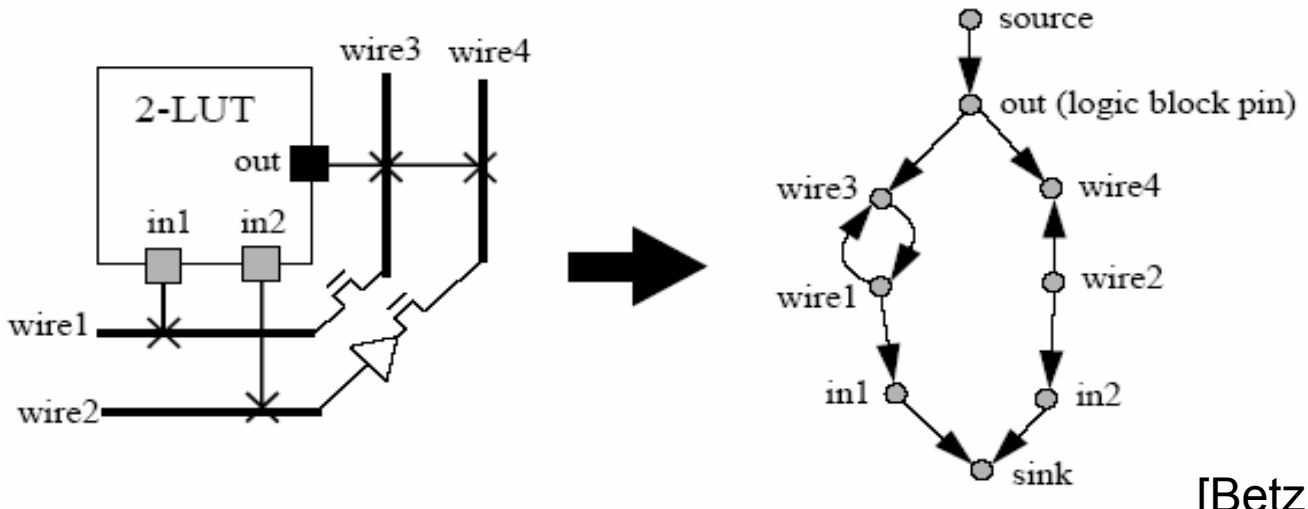


[Betz et al., 1999]

Routing Approach

- Reduce Glitches by balancing input arrival times
  - By routing through paths with desired delays
  - Lengthen paths of early-arriving inputs
- No Architectural Modification
  - Applied to existing FPGAs
- Do not affect critical path delay

FPGA Routing Architecture



[Betz et al., 1999]

- Routing resource graph
- Assumption: Buffered Switches
  - True for commercial FPGAs (boost performance)
  - Linear delay model for paths

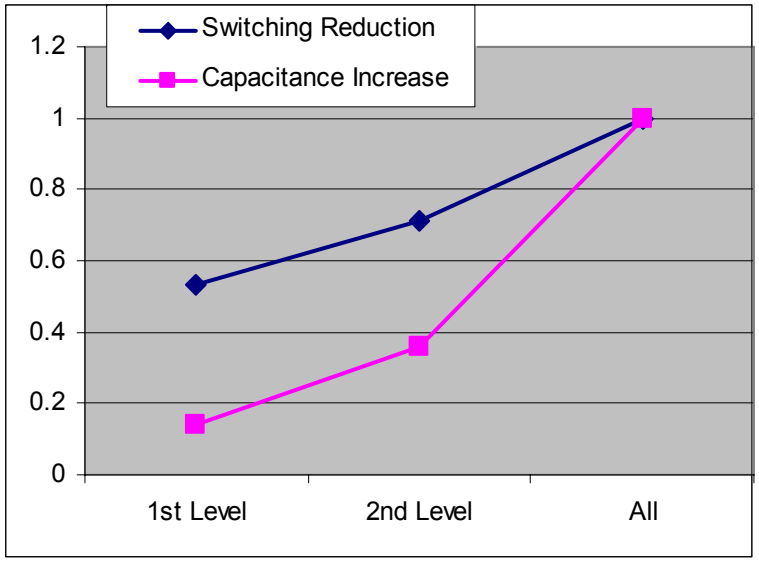
Algorithm Overview

- Get an valid routing solution (VPR router)
- Reduce Glitches by Balancing certain source-sink pairs

- Selection and Ordering of pairs
- Rip-up and re-route each pair
- Reroute with Path-Finding Algorithm
  - Find path with desired delay

Selection Criteria

- Power overhead due to longer path
  - More capacitance, more buffers
- Balance only beneficial inputs
- Heuristic: Balancing inputs of 1st-level clusters



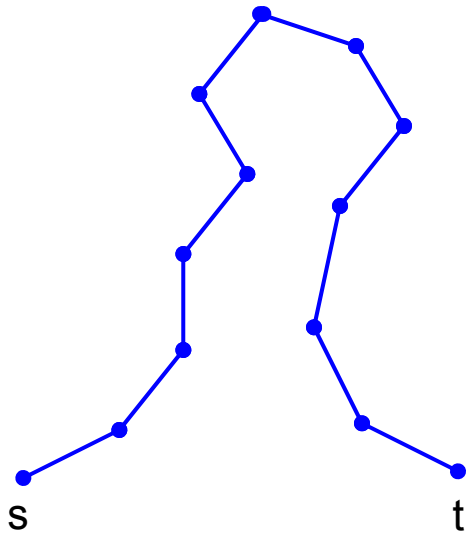
Ordering Criteria

- LUT Input Weighting
  - Likelihood of glitch generation
  - Signal probability of Boolean difference
- Balancing overhead
  - Prefer small increase delays over large increase
- Path ranking

$$Path\_Rank = \frac{LUT\_Input\_Weight}{Increased\_Delay}$$

Path-Finding Algorithm

- Inputs:
  - Routing-resource graph (V, E) with delay information
  - Source-sink pair (*s*, *t*)
  - Desired delay range d±Δ
- Output:
  - Path from *s* to *t* with desired delay
  - Not always has a solution



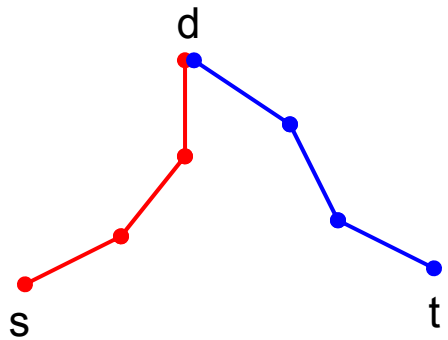
Motivation

- Try every paths: exponential complexity

- Heuristic: Select a manageable subset of paths
  - Polynomial complexity
  - Still provide reasonable quality of results
  - Wide range of path lengths

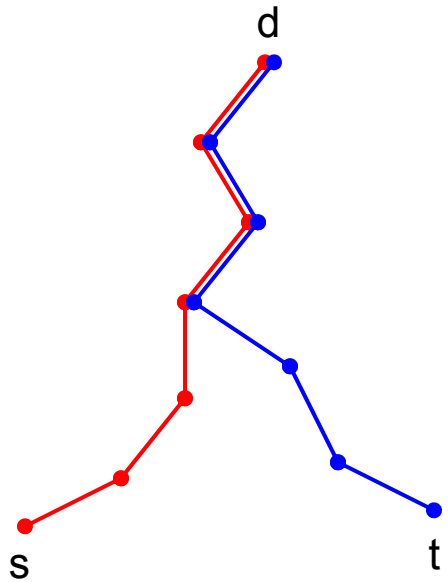
Motivation

- Efficient path algorithm: shortest path
  - Find only path with smallest delay
- Detour
  - Combine shortest paths
  - Increase path delay

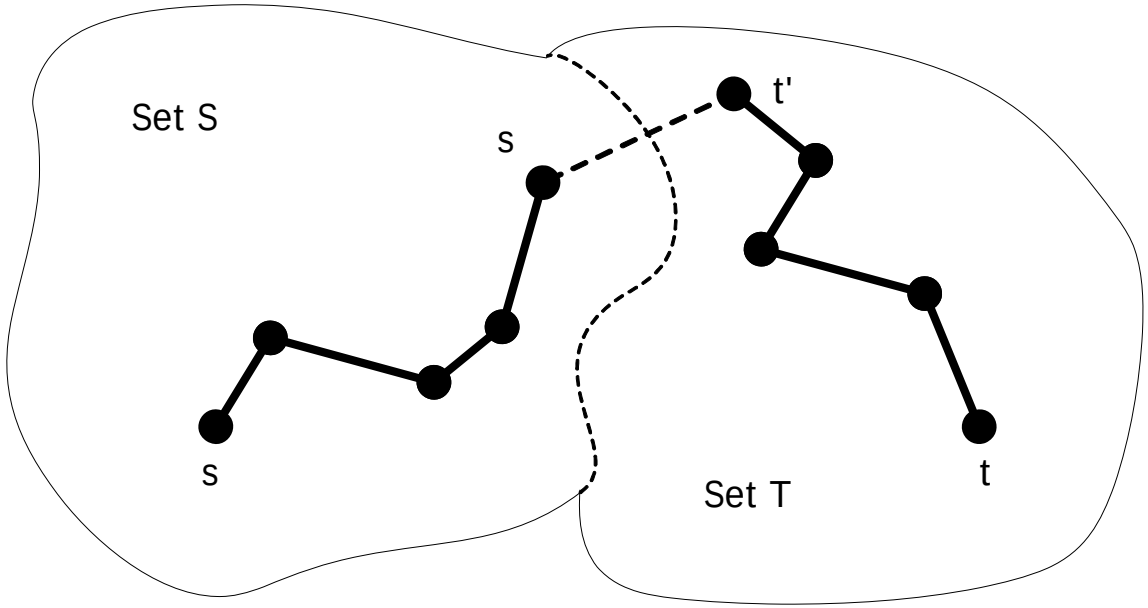


Motivation

- Efficient path algorithm: shortest path
  - Find only path with smallest delay
- Detour
  - Combine shortest paths
  - Increase path delay
- Problem: overlapping



Candidate Paths



- Set S: vertices closer to *s*
- Set T: vertices closer to *t*
- (*s*', *t*'): direct connection (edge)
- Guarantee no overlapping

Glitch-Reducing Framework

Run timing-driven VPR router to get a routing solution

```
for each 1-st level CLB input
    get desired balanced delay
    compute path rank
end
Sort these inputs by their rank
for each input
    rip-up current path
    use path-finding algorithm to find a path with the desired delay
    if can not find such a path
        restore the ripped-up path
    end
end
```

Experiment Settings

- 20 largest circuits from MCNC and ISCAS89
- Timing-driven place and route by VPR
- 4-LUT, cluster size 4
- Power simulator fpgaEVA-LP2 [Li et al., 2005]
  - Dynamic power
  - Glitch power
  - Switching activity

Results

Circuits	Dynamic Power (mW)		Impr. (%)	Runtime (seconds)	
	without	with		without	with
alu4	38.20	35.29	6.45	22.8	42.7
apex2	40.13	39.75	1.26	34.6	82.4
apex4	17.65	16.20	8.10	20.9	22.2
des	67.53	59.95	9.92	25.9	101.9
ex1010	39.04	32.47	**18.60**	187.4	194.3
exp5p	19.78	15.83	16.87	15.1	16.8
misex3	32.33	29.19	7.67	19.2	32.4
pdc	33.11	29.35	**12.28**	212.3	243.4
seq	34.92	32.92	5.42	28.6	41.1
spla	33.80	31.09	8.76	139.2	148.1
C1355	3.71	3.24	12.04	0.6	1.1
C1908	7.82	7.13	8.94	1.0	1.5
C2670	12.64	11.87	5.24	4.0	8.3
C3540	21.54	19.05	11.45	3.5	3.8
C432	2.68	2.44	7.81	0.4	0.6
C499	3.59	3.07	13.78	0.6	0.8
C5315	34.08	31.03	8.82	7.2	14.0
C6288	73.20	59.39	18.84	4.9	6.1
C7552	43.51	38.55	11.40	9.1	17.9
C880	3.49	3.42	3.33	0.7	0.8
average			**9.85**	36.9	49.0

Results

Circuits	Reduction in Glitch Power (%)	Reduction in Switching Activity (%)	Increase in Wire Length (%)
alu4	38.89	8.01	5.51
apex2	7.48	5.23	9.45
apex4	20.35	10.18	7.10
des	20.81	11.94	8.44
ex1010	28.50	22.94	9.41
exp5p	25.79	21.02	8.43
misex3	28.43	9.35	6.36
pdc	31.63	15.26	7.29
seq	26.77	6.96	4.50
spla	16.17	10.96	7.50
C1355	34.76	15.22	7.80
C1908	21.25	11.52	7.62
C2670	19.09	6.57	4.68
C3540	20.92	14.21	9.99
C432	21.82	9.93	6.62
C499	33.45	17.75	8.62
C5315	18.82	11.10	7.42
C6288	18.05	22.84	9.23
C7552	21.32	13.98	9.74
C880	13.82	6.67	8.27
average	**23.41**	**12.58**	**7.70**

Conclusions

- Glitch reduction in FPGA through routing
  - CAD approach, not require architectural modification
  - 9.8% average of dynamic power reduction
- Efficient Path-finding algorithm
  - A small, efficient subset of possible paths
- Can be combined with other techniques
  - GlitchMap
  - GlitchLess