Clock Buffer Polarity Assignment Utilizing Useful Clock Skews for Power Noise Reduction

Deokjin Joo and Taewhan Kim

SNU CAD Laboratory,
Seoul National University, Korea

Outline

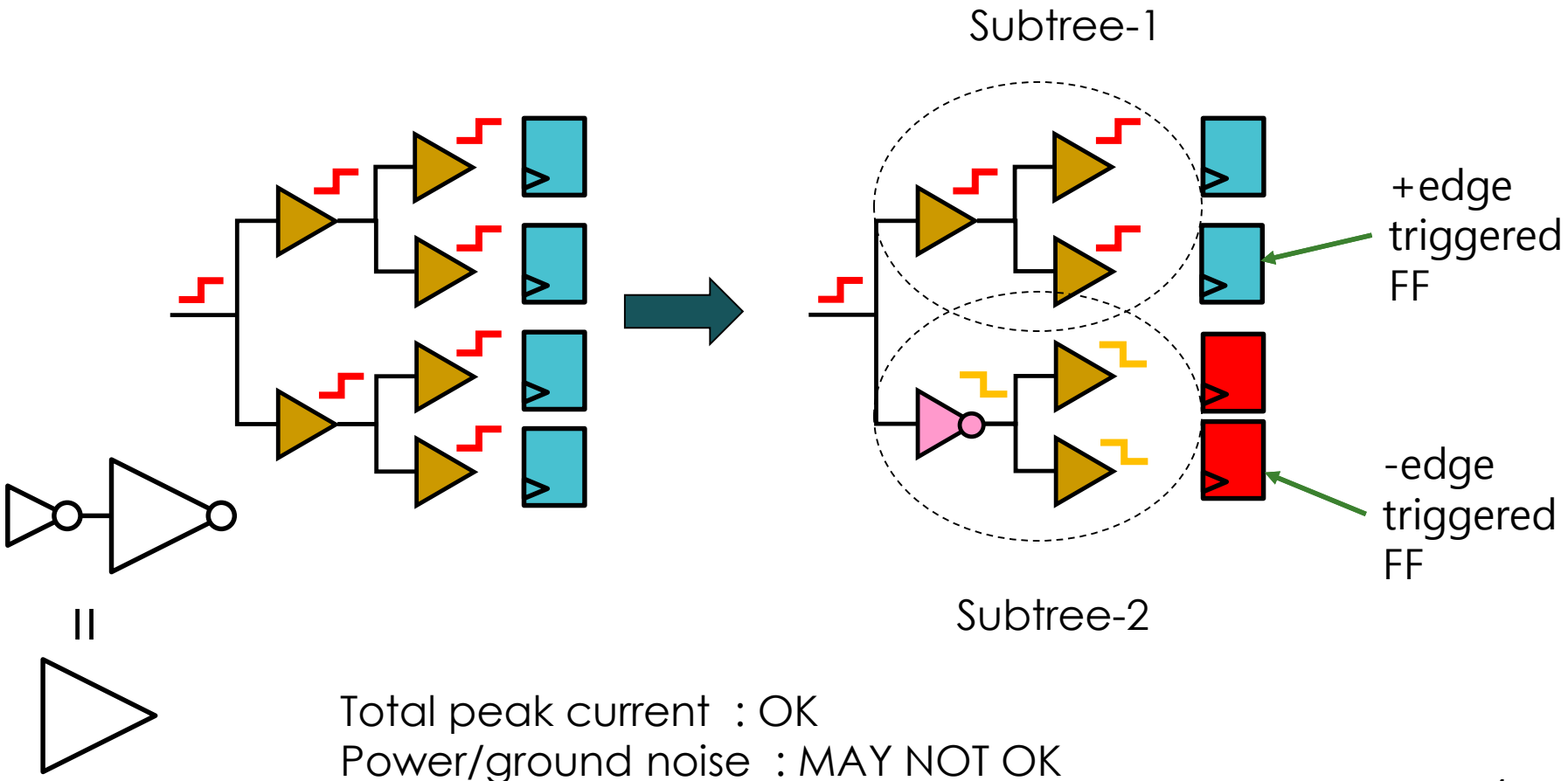
- Introduction
- Motivational Example
- Proposed Method
- Experimental Results
- Extension: Coping with Process Variations

Introduction

- **Introduction**
- Motivational Example
- Proposed Method
- Experimental Results
- Extension: Coping with Process Variations

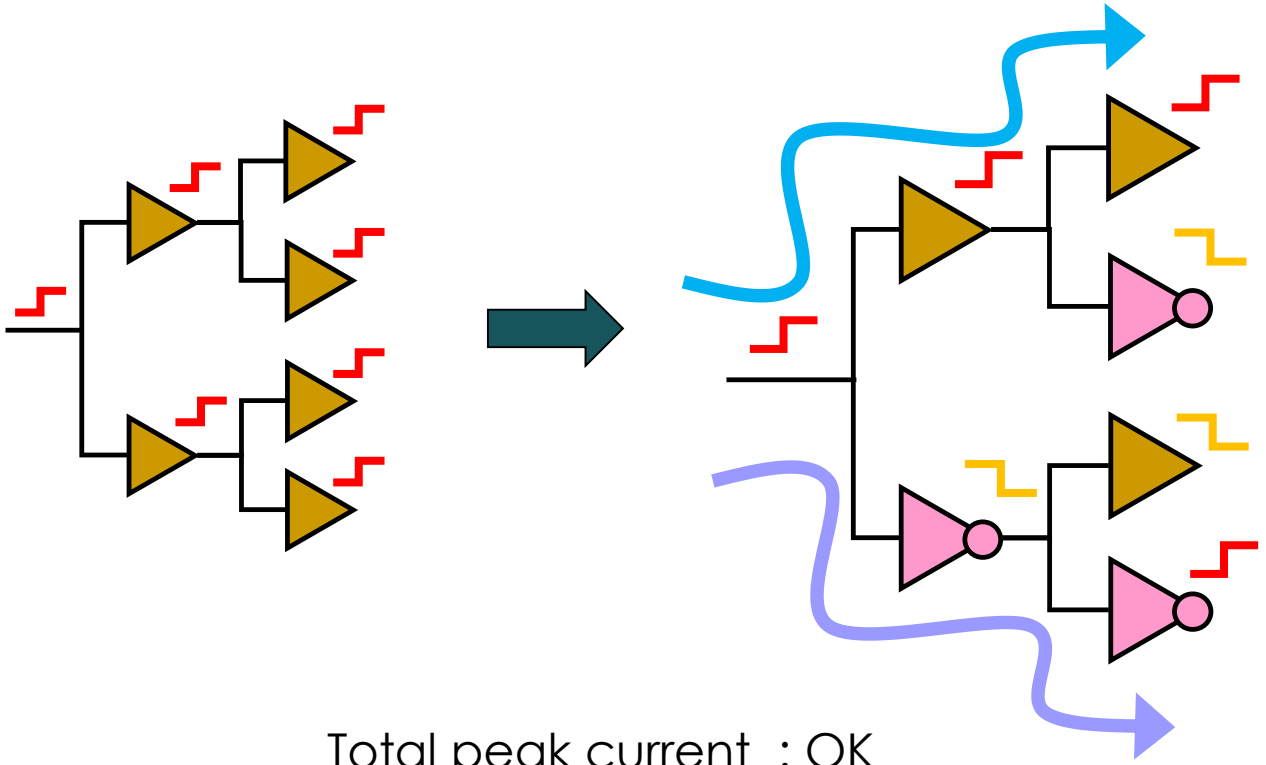
Introduction: Clock Polarity Assignment

- Y.-T. Nieh, S.-H. Huang, and S.-Y. Hsu
  "Minimizing peak current via opposite-phase clock tree"



Introduction: Clock Polarity Assignment

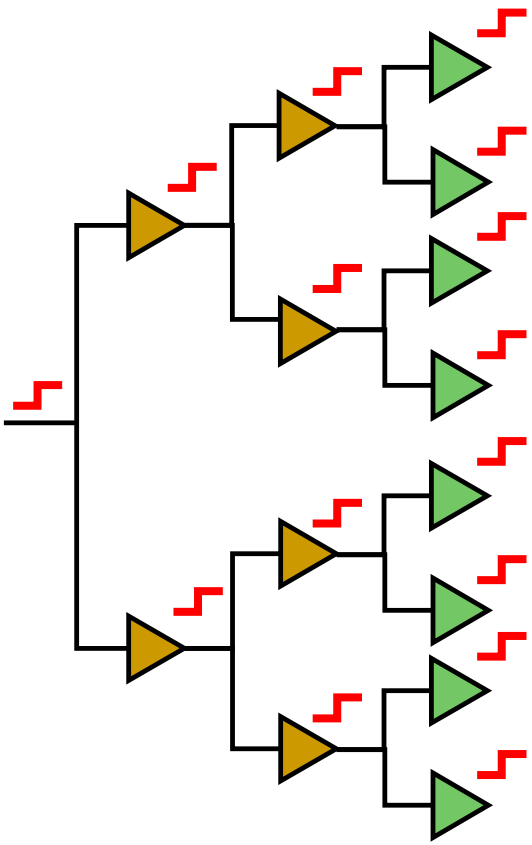
- R. Samanta, G. Venkataraman, and J. Hu,
  "Clock buffer polarity assignment for power noise reduction"

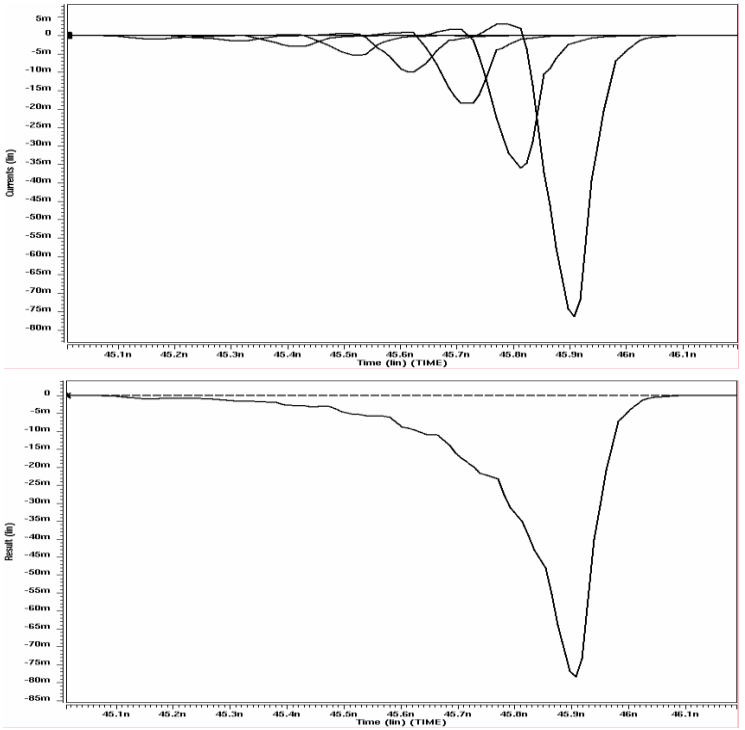


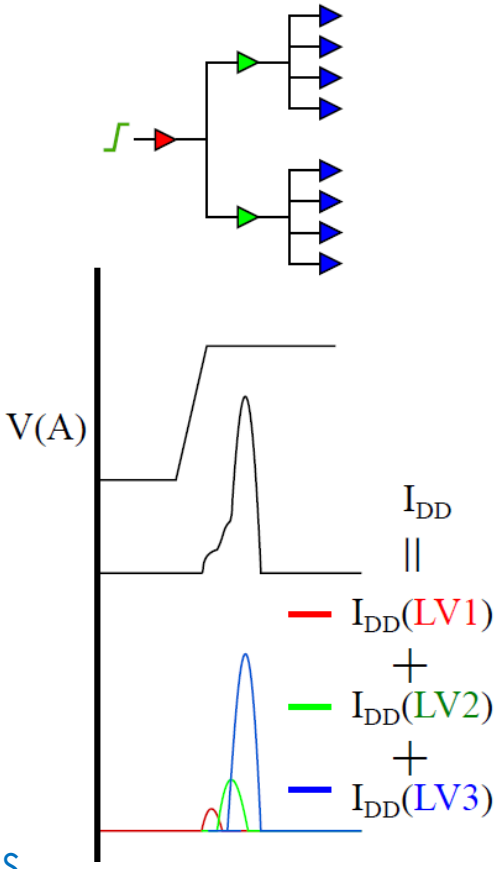
Total peak current : OK
Power/ground noise : OK
Clock skew : MAY NOT BE OK

Introduction: Leaves are Dominant

- P.-Y. Chen, K.-H. Ho, and T. Hwang
  "Skew Aware Polarity Assignment in Clock Tree"
- # buffers ∝ Noise

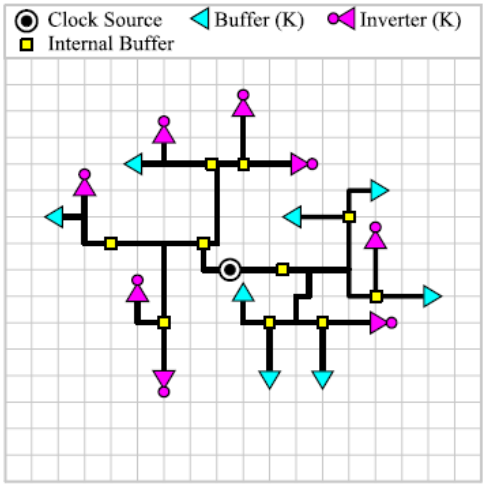


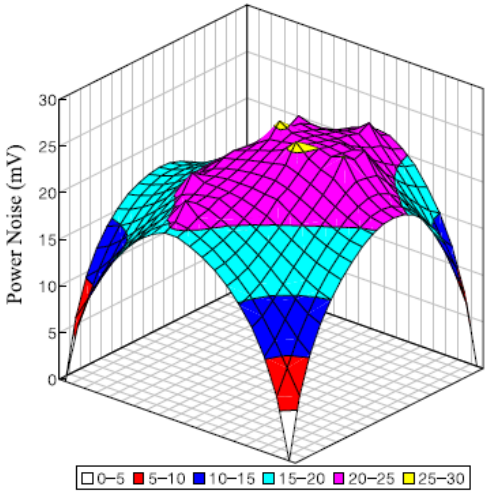




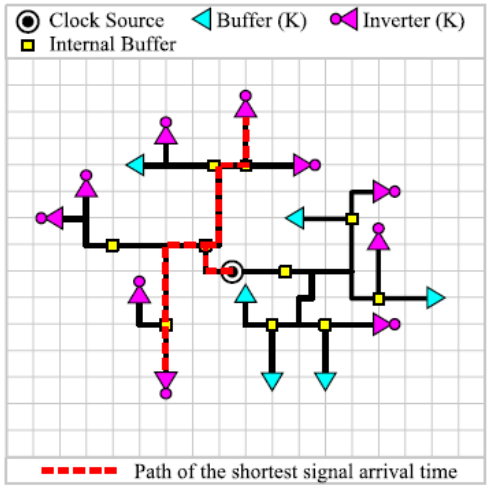
Still, tries to make #bufs = #invs,
NOT OPTIMAL, as will be shown

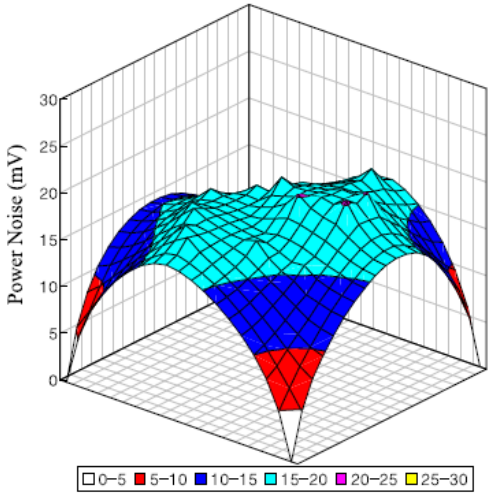
BUFs ≠ # INVs for Better Results

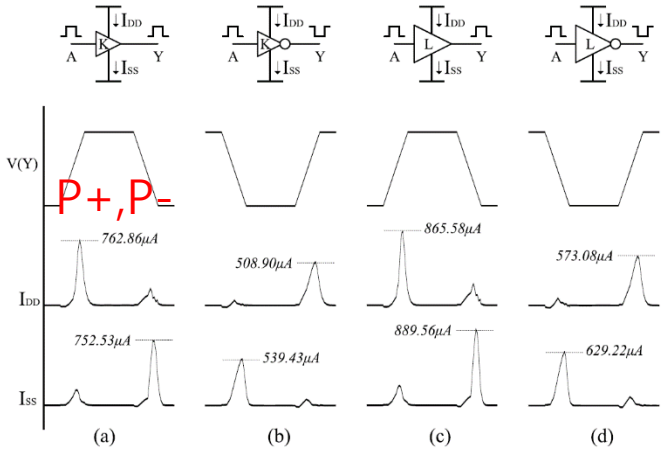




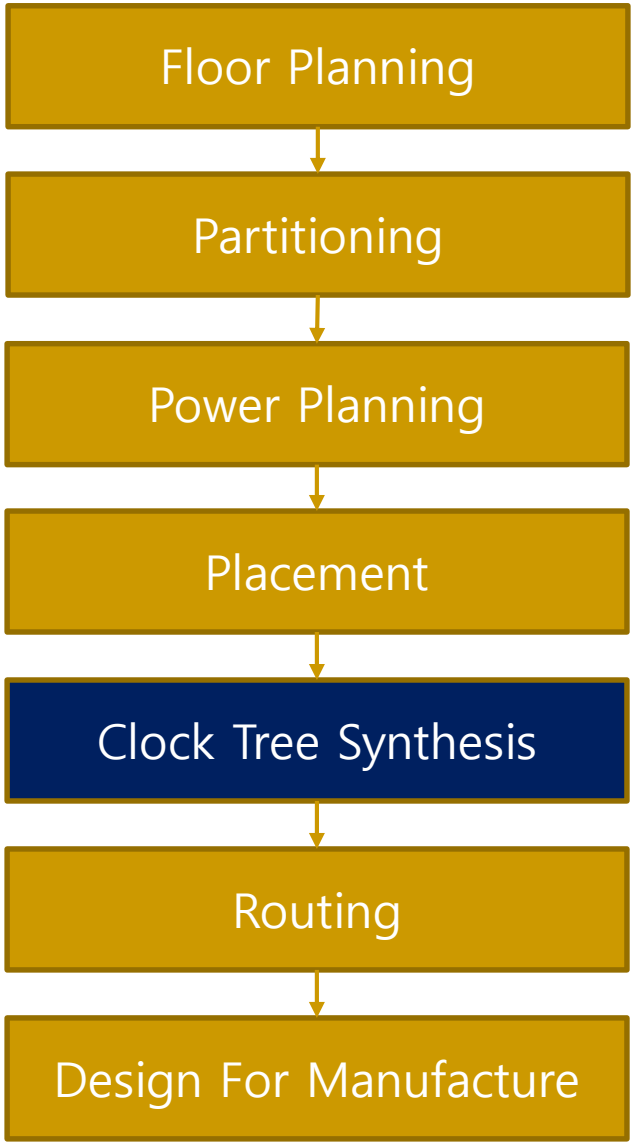
- H. Jang, T. Kim, "Simultaneous Clock Buffer Sizing and Polarity Assignment for Power/Ground Noise Minimization"
- Took peak noise into consideration







Introduction: Clock Polarity Assignment



Only affects CTS stage, minimal impact on design flow!

Motivational Example

- Introduction
- **Motivational Example**
- Proposed Method
- Experimental Results
- Extension: Coping with Process Variations

Limitations of Previous Works

- **Either focused on #BUFs and #INVs, or**
- ***Bounded clock skew* based**

- **Leads to suboptimal results (higher peak current noise)**

Useful Clock Skew

- **Setup and Hold times considered *individually***
  - High performance

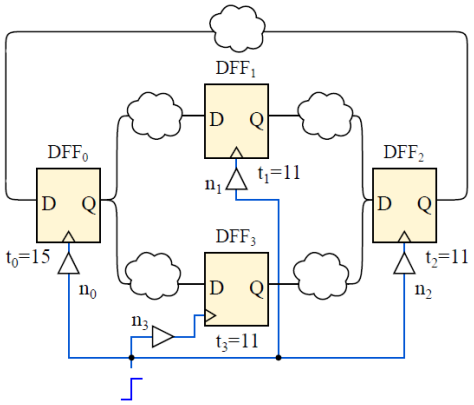
-3	$\leq$	$t_0 - t_1$	$\leq$	2
-5	$\leq$	$t_1 - t_2$	$\leq$	4
-3	$\leq$	$t_0 - t_3$	$\leq$	3
-4	$\leq$	$t_3 - t_2$	$\leq$	2
-3	$\leq$	$t_2 - t_0$	$\leq$	2

(b) Timing constraints

Type	Δt	Noise	
		P+	P-
B1	0	10	3
B2	+2	12	3
I1	0	3	9
I2	+1	3	11

(c) Library

Optimal mapping



(a) Input tree with 4 leaf nodes

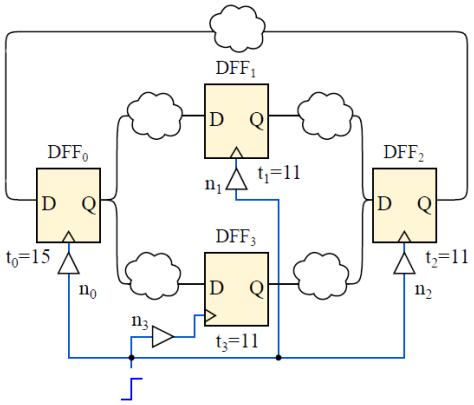
#	Type Selection				Skew	Noise	
	n_0	n_1	n_2	n_3		P+	P-
1	B1	B2	B2	B2	2	46	12
2	B1	B2	B2	I2	3	37	20
3	B1	B2	I2	B2	3	37	20
4	**B1**	**B2**	**I2**	**I2**	**3**	**28**	**28**
5	I1	B2	B2	B2	2	39	18
6	I1	B2	B2	I2	3	30	26
7	I1	B2	I2	B2	3	30	26
8	I1	B2	I2	I2	3	21	34

(d) 8 feasible mappings, found in 4^4 search space

Useful Clock Skew

- **Setup and Hold times considered individually**

Bounded skew based algorithms use the tightest constraint, ±2



(a) Input tree with 4 leaf nodes

-2	-3	≤	$t_0 - t_1$	≤	2	2
-2	-5	≤	$t_1 - t_2$	≤	4	2
-2	-3	≤	$t_0 - t_3$	≤	3	2
-2	-4	≤	$t_3 - t_2$	≤	2	2
-2	-3	≤	$t_2 - t_0$	≤	2	2

(b) Timing constraints

Type	Δt	Noise	
		P+	P-
B1	0	10	3
B2	+2	12	3
I1	0	3	9
I2	+1	3	11

(c) Library

#	Type Selection				Skew	Noise	
	n_0	n_1	n_2	n_3		P+	P-
1	B1	B2	B2	B2	2	46	12
5	I1	B2	B2	B2	2	39	18

(d) 8 feasible mappings, found in 4^4 search space

Proposed Method

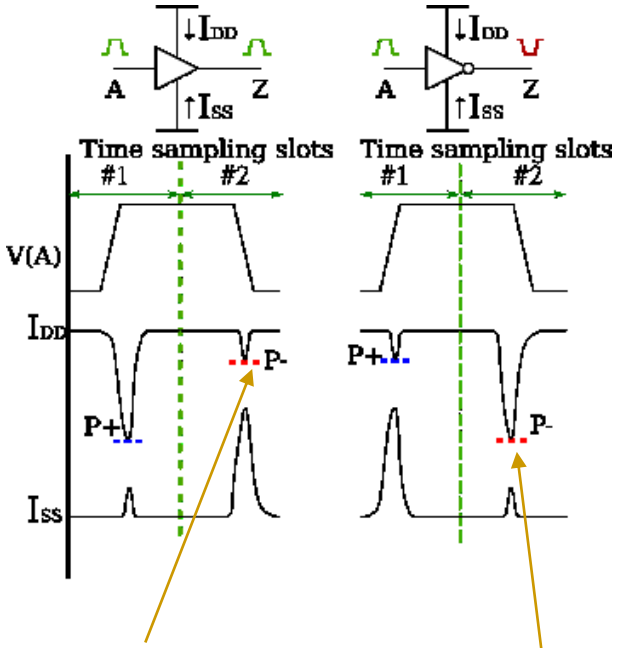
- Introduction
- Motivational Example
- **Proposed Method**
- Experimental Results
- Extension: Coping with Process Variations

Problem Definition: UsefulMin

- **Given:**
  - B: Buffer library
  - I: Inverter library
  - L: Leaf buffering elements
  - S: Time sampling slots
- **Find:**
  - ϕ: $L \rightarrow \{B \cup I\}$ (Polarity assignment / Sizing) that minimizes

$$\max_{s \in S} \left\{ \sum_{e_i \in L} noise(\phi(e_i), s) \right\}$$

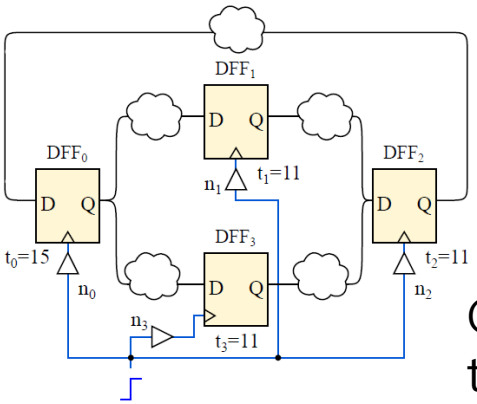
- **Subject to**
  - A set of useful clock skew constraints
- **Map leaf buffers to BUF/INV of good size**
- **Minimize noise**
- **Under a set of *useful clock skew* constraints (individual pair-wise clock skew constraints)**
- **NP-Complete (decision version)**



$noise(\phi(B_1), s_2)$

$noise(\phi(I_1), s_2)$

Clique Based Approach

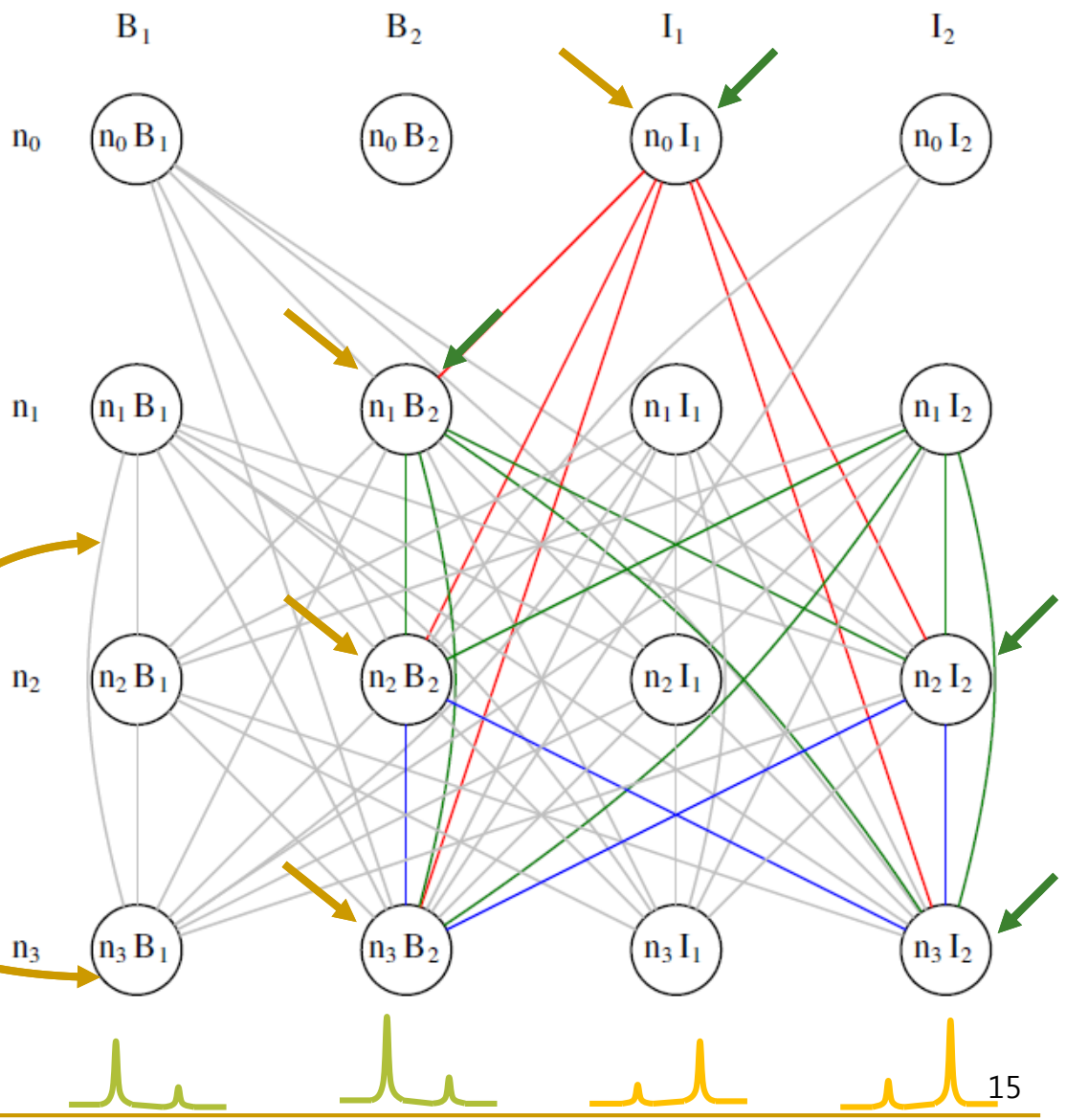


Graph topology

$$
\begin{array}{rcccl}
-3 & \leq & t_0 - t_1 & \leq & 2 \\
-5 & \leq & t_1 - t_2 & \leq & 4 \\
-3 & \leq & t_0 - t_3 & \leq & 3 \\
-4 & \leq & t_3 - t_2 & \leq & 2 \\
-3 & \leq & t_2 - t_0 & \leq & 2
\end{array}
$$

Type	Δt	Noise	
		P+	P-
B1	0	10	3
B2	+2	12	3
I1	0	3	9
I2	+1	3	11

Weights bound to nodes



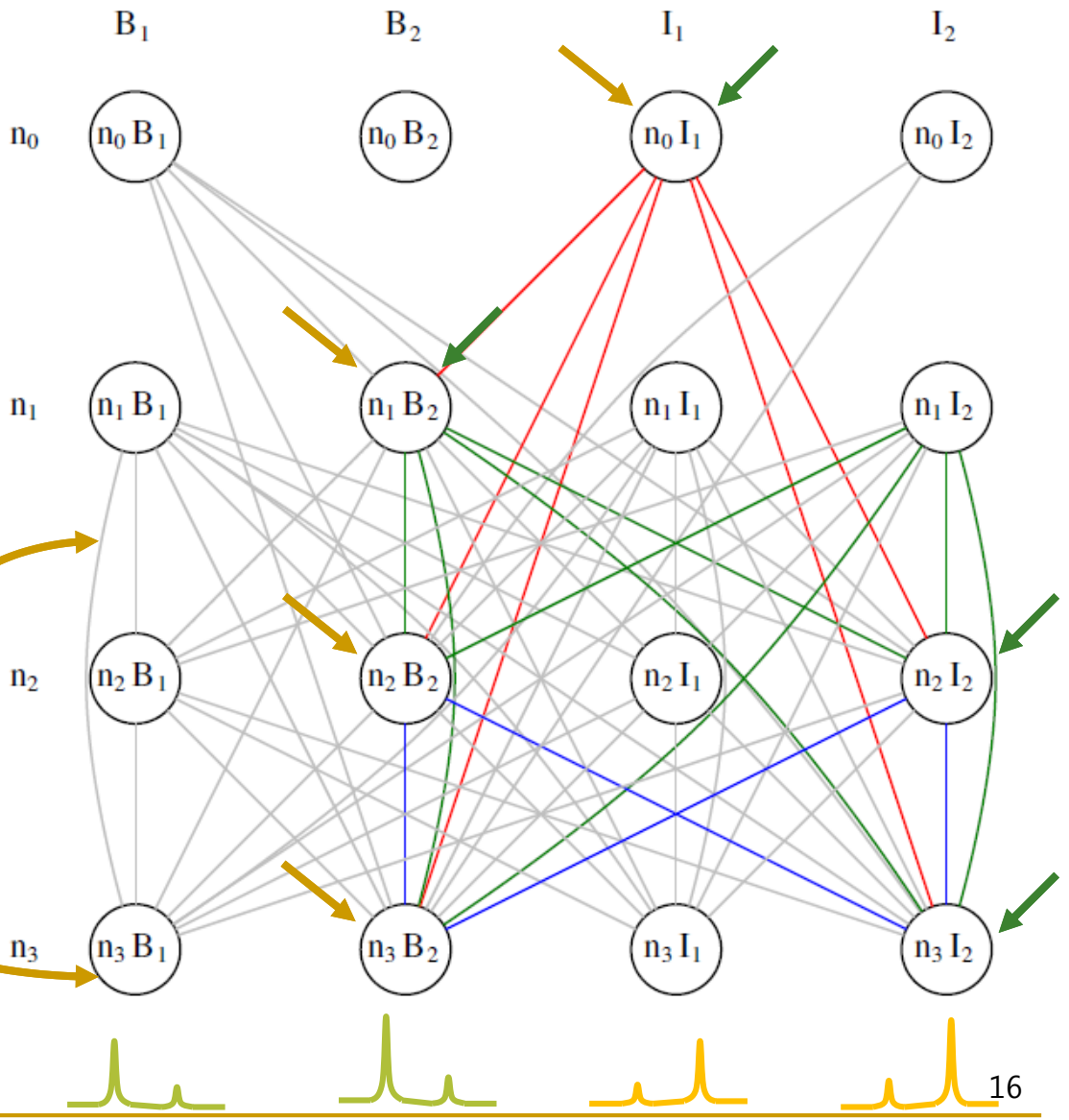
Clique Based Approach

#	Type Selection			
	n_0	n_1	n_2	n_3
1	B1	B2	B2	B2
2	B1	B2	B2	I2
3	B1	B2	I2	B2
4	**B1**	**B2**	**I2**	**I2**
5	I1	B2	B2	B2
6	I1	B2	B2	I2
7	I1	B2	I2	B2
8	I1	B2	I2	I2

-3	$\leq$	$t_0 - t_1$	$\leq$	2
-5	$\leq$	$t_1 - t_2$	$\leq$	4
-3	$\leq$	$t_0 - t_3$	$\leq$	3
-4	$\leq$	$t_3 - t_2$	$\leq$	2
-3	$\leq$	$t_2 - t_0$	$\leq$	2

Type	Δt	Noise	
		P+	P-
B1	0	10	3
B2	+2	12	3
I1	0	3	9
I2	+1	3	11

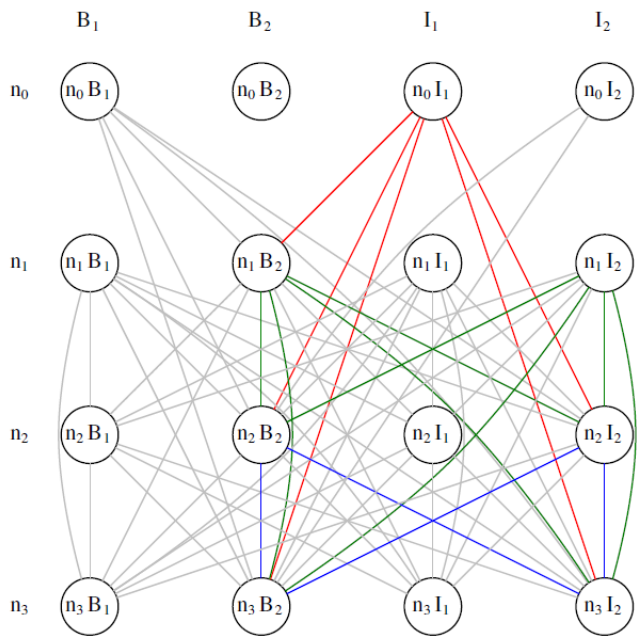
Weights bound to nodes



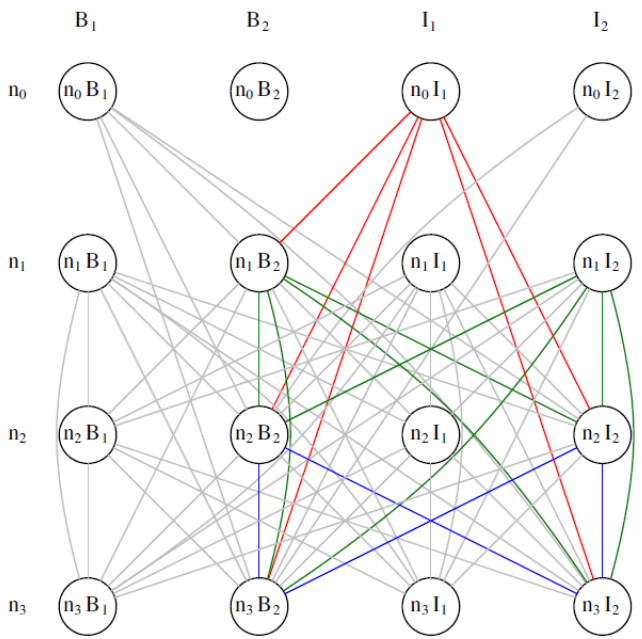
Integer Linear Programming Approach

- **ILP approach possible**
  - 1, when choice in clique, 0 when not
  - Unsuitable for larger problems
- **LP relaxation infeasible (problem property)**

e0= 1 0 0 0



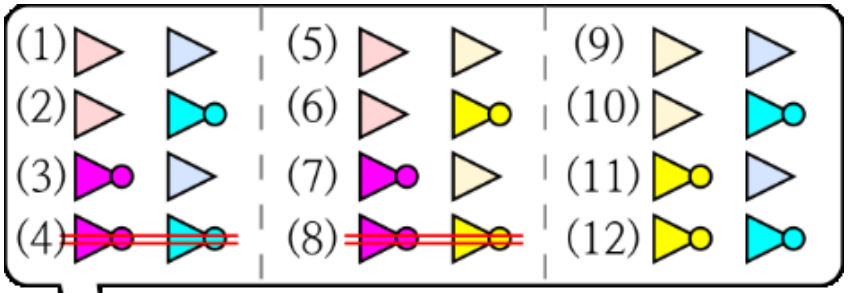
e0= 0.25 0.25 0.25 0.25



K-neighbor Search

- **There are too many cliques in the graph**
- **Complete search impossible**
- **Local search: find a clique, and explore K-neighbor cliques.**
  - Clique Y is an K-neighbor of X,
  - If Y is obtainable from X by replacing at most K vertices of X.
- **Still, too many nodes in the graph.**
  Not easy to increase K
  - 100 leaves and K=5 → $\binom{100}{K} = 75.29 \times 10^6$

Zone-by-zone K-neighbor Search

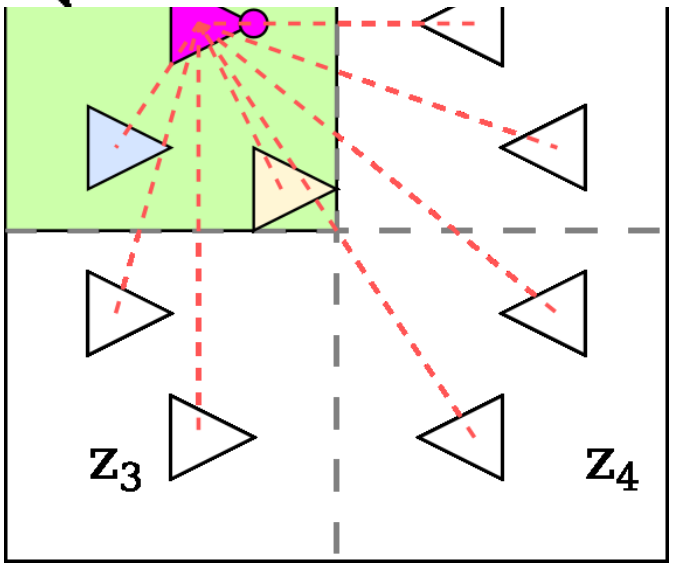


K=2

$\binom{3}{2} 2^2 = 12$

neighbor cliques

- Group leaf buffers into zones by locations
- Zone-by-zone Iteration repeats, until no improvement
- Scalable: |Zone| and K.



Experimental Results

- Introduction
- Motivational Example
- Proposed Method
- **Experimental Results**
- Coping with Process Variations

Experimental Results

- **10.9% Noise reduction, compared to previous (bounded skew) algorithm**

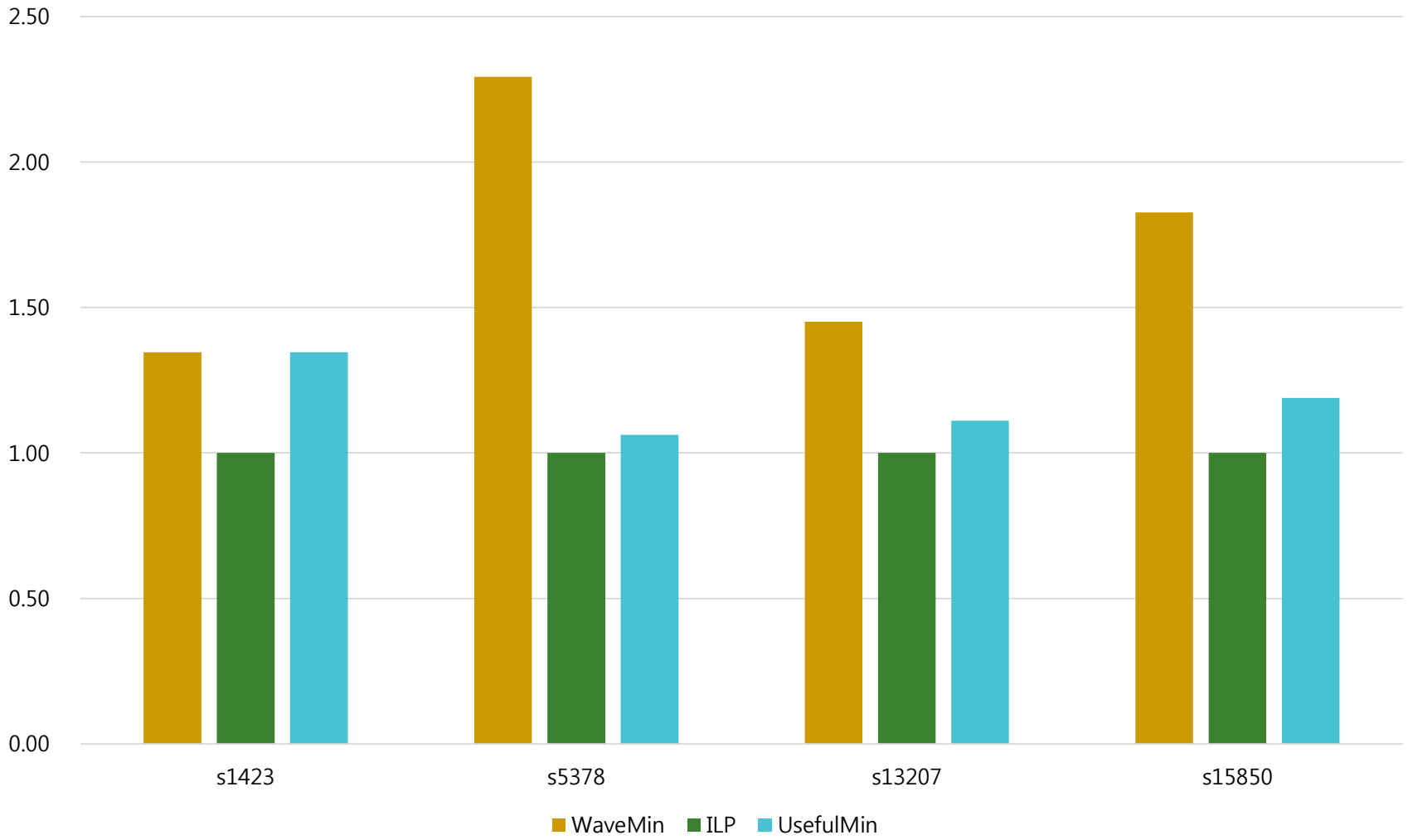
60-90 ps

Nangate 45nm Tech.

ISPD 2010 Benchmark Info			Base		WaveMin*		UsefulMin		Improvement	
Benchmk Circuits	# Cnstraints	# Lvs	Noise (mA)	Area (um^2)	Noise (mA)	Area (um^2)	Noise (mA)	Area (um^2)	Noise (%)	Area (%)
01	11070	221	235.1	22.38	155.3	21.24	122.1	13.73	21.38	35.38
02	22490	454	433.9	45.97	281.9	43.83	242.4	29.62	14.01	32.42
03	12000	113	106.1	11.44	45.58	6.00	42.56	6.49	6.63	-8.26
04	18450	116	115.3	11.75	52.85	8.58	48.21	6.60	8.78	23.07
05	10160	49	51.83	4.96	35.12	4.84	27.78	3.09	20.90	36.05
06	9810	77	74.26	7.80	43	6.28	40.25	4.88	6.40	22.22
07	19150	131	125.4	13.26	73.26	13.20	67.13	8.51	8.37	35.55
08	11340	89	90.88	9.01	51.43	7.05	51.05	5.88	0.74	16.59
Average									**10.90**	**24.13**

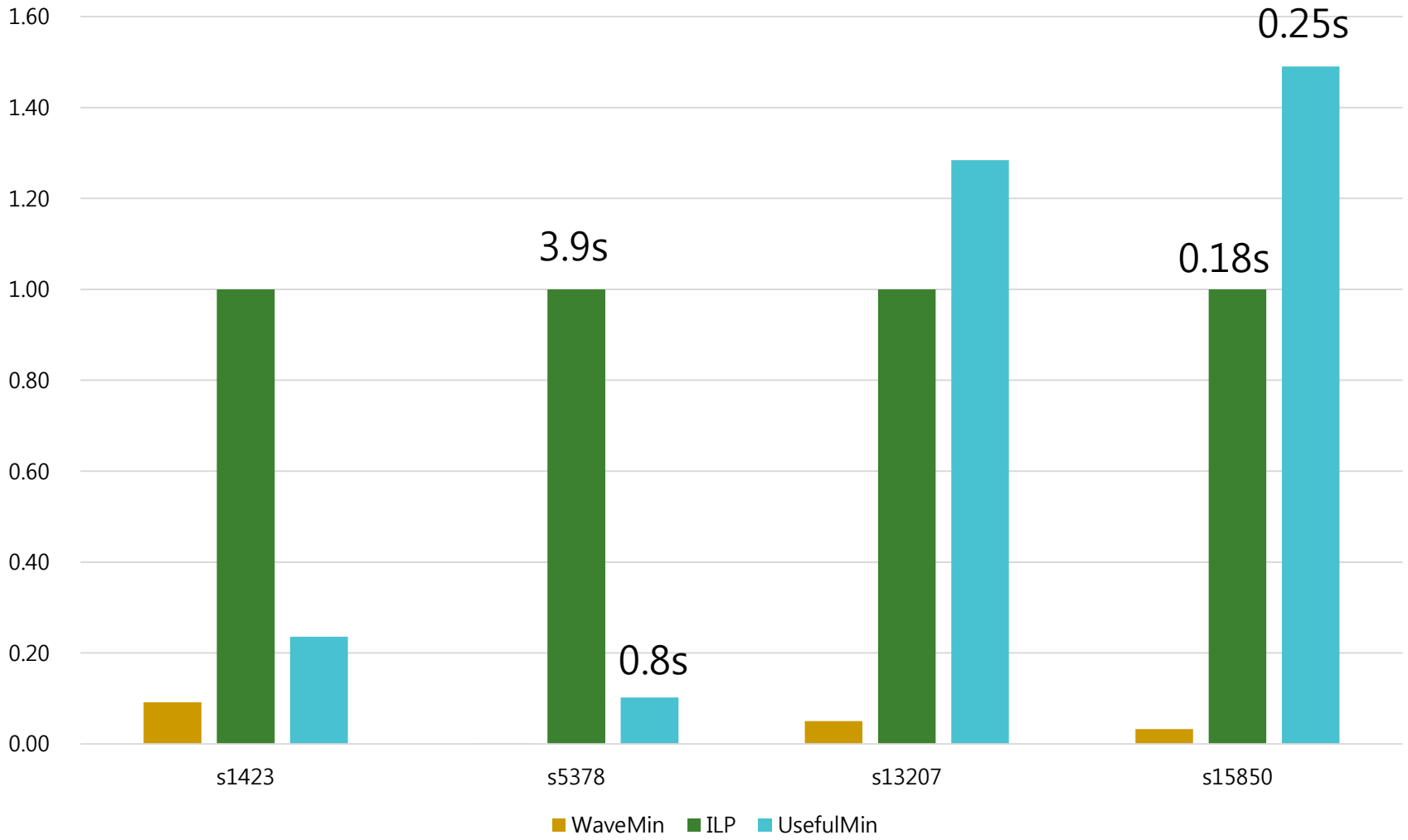
* "A Fine-Grained Clock Buffer Polarity Assignment for High-Speed and Low-Power Digital Systems", D. Joo and T. Kim, *IEEE TCAD*, Mar. 2014

Experimental Results: Normalized Peak Current



ISCAS'89 Benchmarks

Experimental Results: Normalized Exec. Time



ISCAS'89 Benchmarks

The largest circuit = ISCAS'89 s5378

Coping with Process Variations

- Introduction
- Motivational Example
- Proposed Method
- Experimental Results
- **Extension: Coping with Process Variations**

Background

- **Technology scaling (<45nm), higher process variations**
- **Propagation delays are random variables**

1. **Worst-case timing analysis**
   - Excess margin, lower performance
2. **Statistical Static Timing Analysis (SSTA)**

Background

- **J. Lu and B. Taskin, "Clock buffer polarity assignment with skew tuning," TODAES 2011.**
  - Incrementally reduce clock skew to increase yield
  - Noise increases

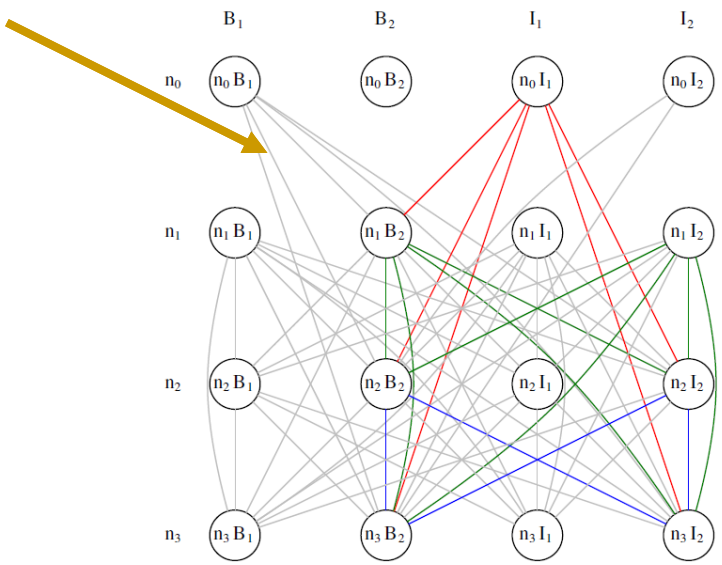
- **M. Kang and T. Kim, "Clock buffer polarity assignment considering the effect of delay variations," ISQED 2010**
  - For *each pair* of leaves, yield is satisfied
  - No direct control over design yield

Proposed Method

- **Construct graph, considering γ**
  - Statistical Static Timing Analysis (SSTA)
- **Given yield constraint γ,**

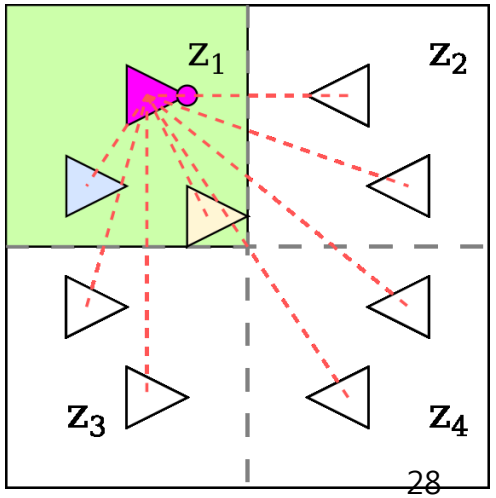
Create edge, if pairwise yield satisfies γ.

(Satisfies imposed useful skew constraint with probability γ)



Proposed Method

- **Compute design yield, during K-neighbor exploration,**
- **Given yield constraint γ,**
  - Suppose the current best clique has noise n_{best}, *design* yield γ_{best}
  - For candidate clique, compute noise n_c and *design* yield γ_c
  - Select candidate clique if…
    - $\gamma_c > \gamma_{best}$, when $\gamma_{best} < \gamma$
      - Select higher yield choice if neither satisfy γ
    - $n_c < n_{best}$, when $\gamma_{best} > \gamma$ and $\gamma_c > \gamma$
      - Select lower noise choice if both satisfy γ



Experimental Results Under Process Variations

- **Gained yield improvement,**
- **without sacrificing noise**

Benchmark Info		Average Peak Current (mA)			Yield (%)			
Circuit	γ	Skew Tuning	Pairwise	UsefulMin-VA	Base	Skew Tuning	Pairwise	UsefulMin-VA
01	0.83	120.10	123.68	92.93	84.70	76.4	73.1	81.1
02	0.39	222.75	230.50	195.54	40.70	28.6	27.2	39.4
03	0.98	50.40	51.10	51.47	99.60	94.9	93.8	98.6
04	0.98	54.79	55.43	58.77	99.70	94	94.4	98.9
05	0.98	27.93	27.78	27.92	99.99	98.8	98.6	99.7
06	0.98	39.54	39.65	40.12	100.00	99.4	99.7	100
07	0.98	59.76	62.09	66.45	99.90	96.3	96.5	99.1
08	0.98	47.59	47.45	45.40	100.00	99.7	99.8	99.8

Conclusions

- Proposed a scalable solution to the Polarity Assignment problem under Useful Clock Skew constraints
- Average 10.9% noise reduction over the best known method
- Extension: PA under delay variations

Questions?

THANK YOU