

Timing-Driven Steiner Trees are (Practically) Free

Charles J. Alpert - *IBM Austin Research Lab*

Andrew B. Kahng - *Blaze DFM Inc.*

C. N. Sze - *IBM Austin Research Lab*

Qinke Wang – *UCSD*

for Timing Estimation

Timing-Driven Steiner Trees are (Practically) Free

during Physical Synthesis

Charles J. Alpert - *IBM Austin Research Lab*

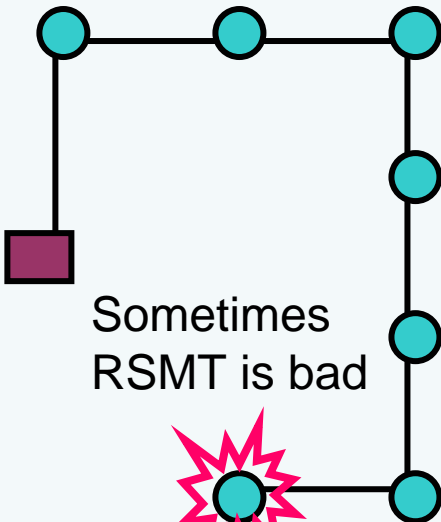
Andrew B. Kahng - *Blaze DFM Inc.*

C. N. Sze - *IBM Austin Research Lab*

Qinke Wang – *UCSD*

The Big Picture

- **RSMT**
- Minimum wirelength
- Widely used by PD tools



Bounded Radius
Bounded Cost
(BRBC)

Prim-Dijkstra
(PD)

Shallow-light
spanning trees

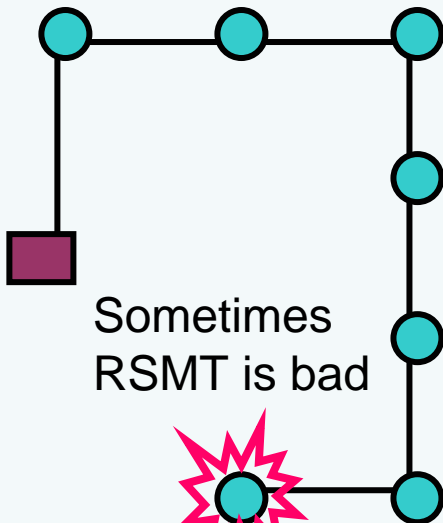
Minimum k-moment
Steiner trees

Shortest-weighted-path
Steiner trees

- **MRSA**
- Minimum source-sink distance
- Shortest routing distance for any critical sink

The Big Picture

- **RSMT**
- Minimum wirelength
- Widely used by PD tools



Timing Driven Steiner trees

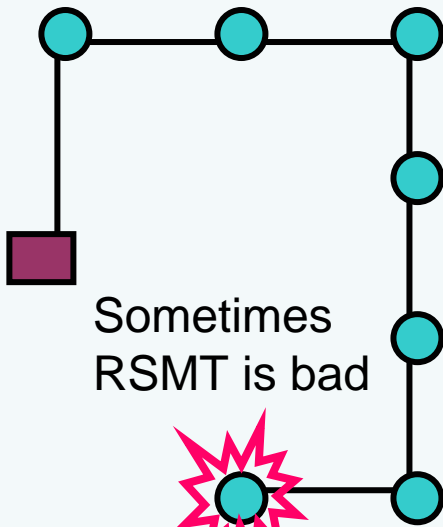
- Extra wirelength is bad
 - Increase load cap and in turn increase delay
 - Hurt routability

- **They are actually not very bad!**
 - Average 2-4% increase for real ASIC nets
 - Negligible increase for post P&R results

New metric
for evaluation

The Big Picture

- **RMST**
- Minimum wirelength
- Widely used by PD tools



What is a
"good"
timing-
driven
Steiner tree
for timing
estimation?

- **MRSA**
- Minimum source-sink distance
- Shortest routing distance for any critical sink

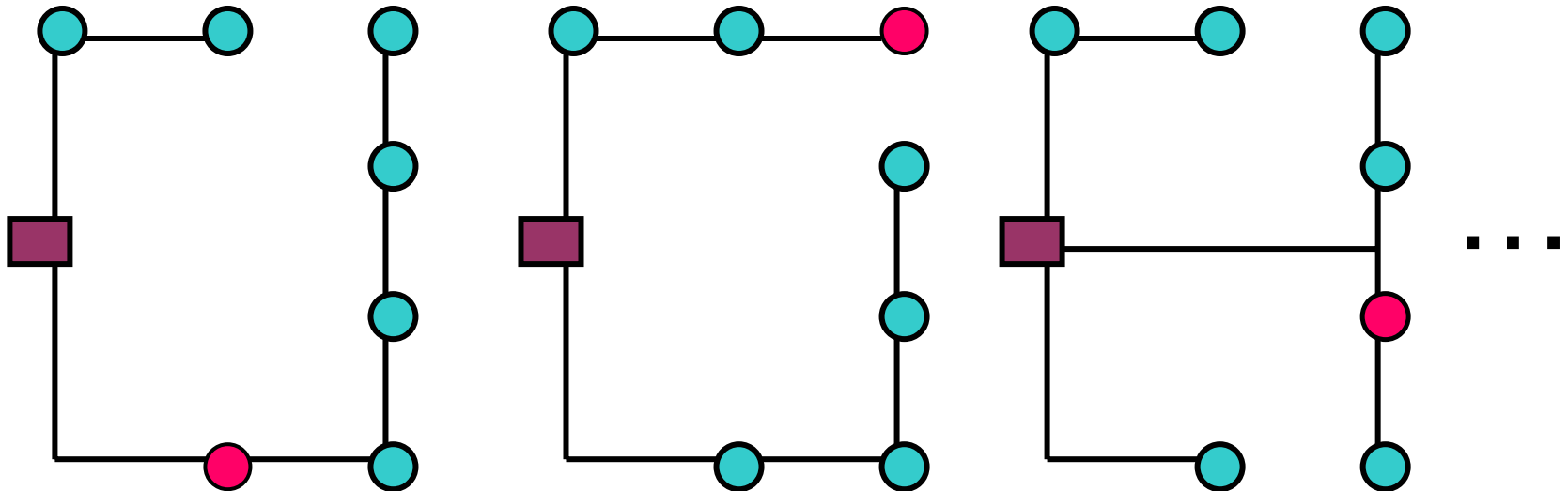
Rectilinear Steiner Minimum Tree

- Real problem of RSMT
 - Unmatched with the final routing
 - Wrong timing estimation
 - Bad information to optimization steps



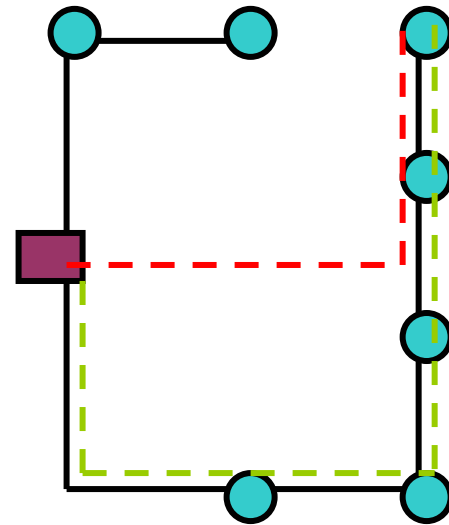
Timing-driven Steiner Trees

- Real problem
 - Timing assertions change over optimization steps



New Metric : Radius-ratio

- Radius-ratio of a sink
 - $\frac{\text{Path length along tree}}{\text{source-sink distance}}$
- Radius-ratio of the routing tree
 - Maximum radius-ratio for all sinks
- Better than bounded-radius trees
 - Dominated by the farthest sink

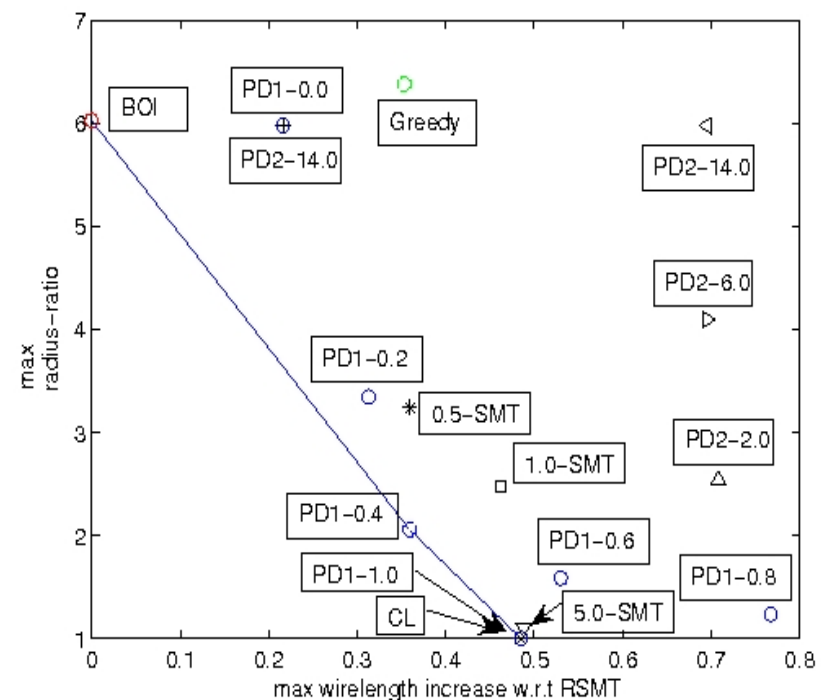
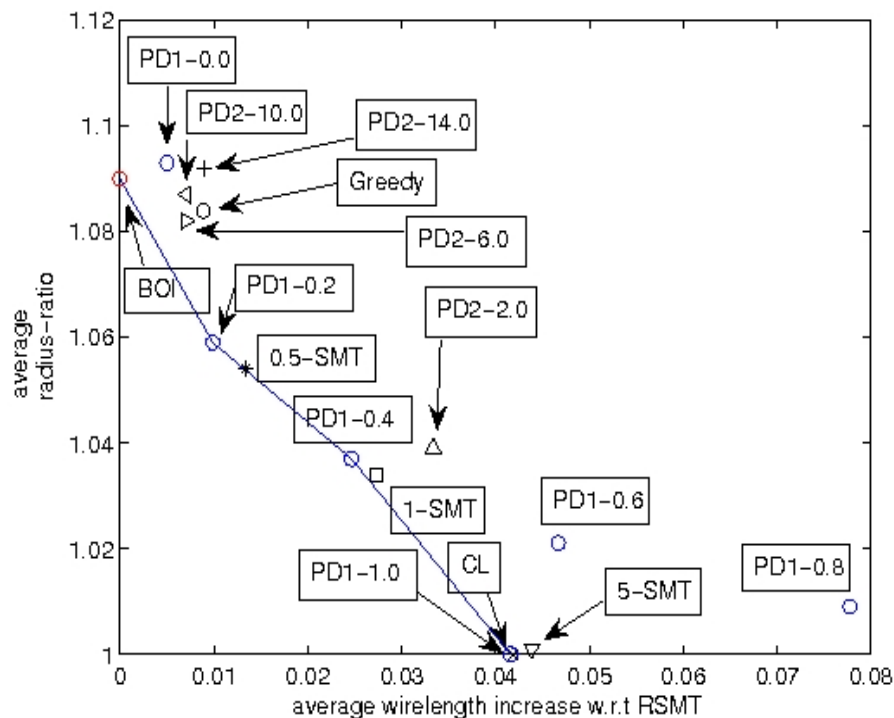


Comparison of Steiner trees

- Fast Steiner trees in test
 - MRSA heuristic (Rao et al, Algorithmica 92)
 - RSMT heuristic (Borah et al, TCAD 94)
 - SERT (Boese et al, TCAD 95)
 - Prim-Dijkstra (Alpert et al, TCAD 95)
 - k-th moment SMT (Qiu & Shi, SODA 04)
- Benchmarks
 - 4000 nets from IBM ASIC designs

Experiment Results

- CL : MRSA heuristic
- BOI : RSMT heuristic
- k-SMT : k-th moment SMT
- SERT : Steiner Elmore routing tree
- PD1-c : Prim-Dijkstra alg-1 with parm c
- PD2-p : Prim-Dijkstra alg-2 with parm p

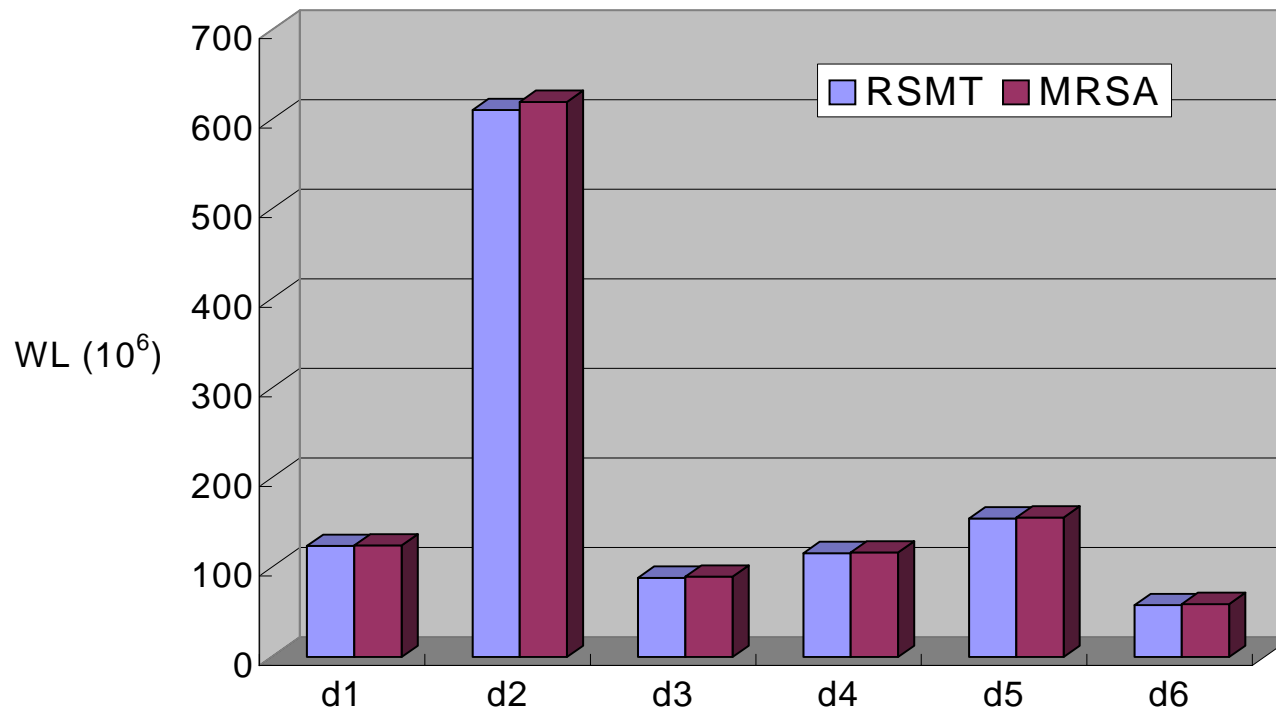


Summary of Results

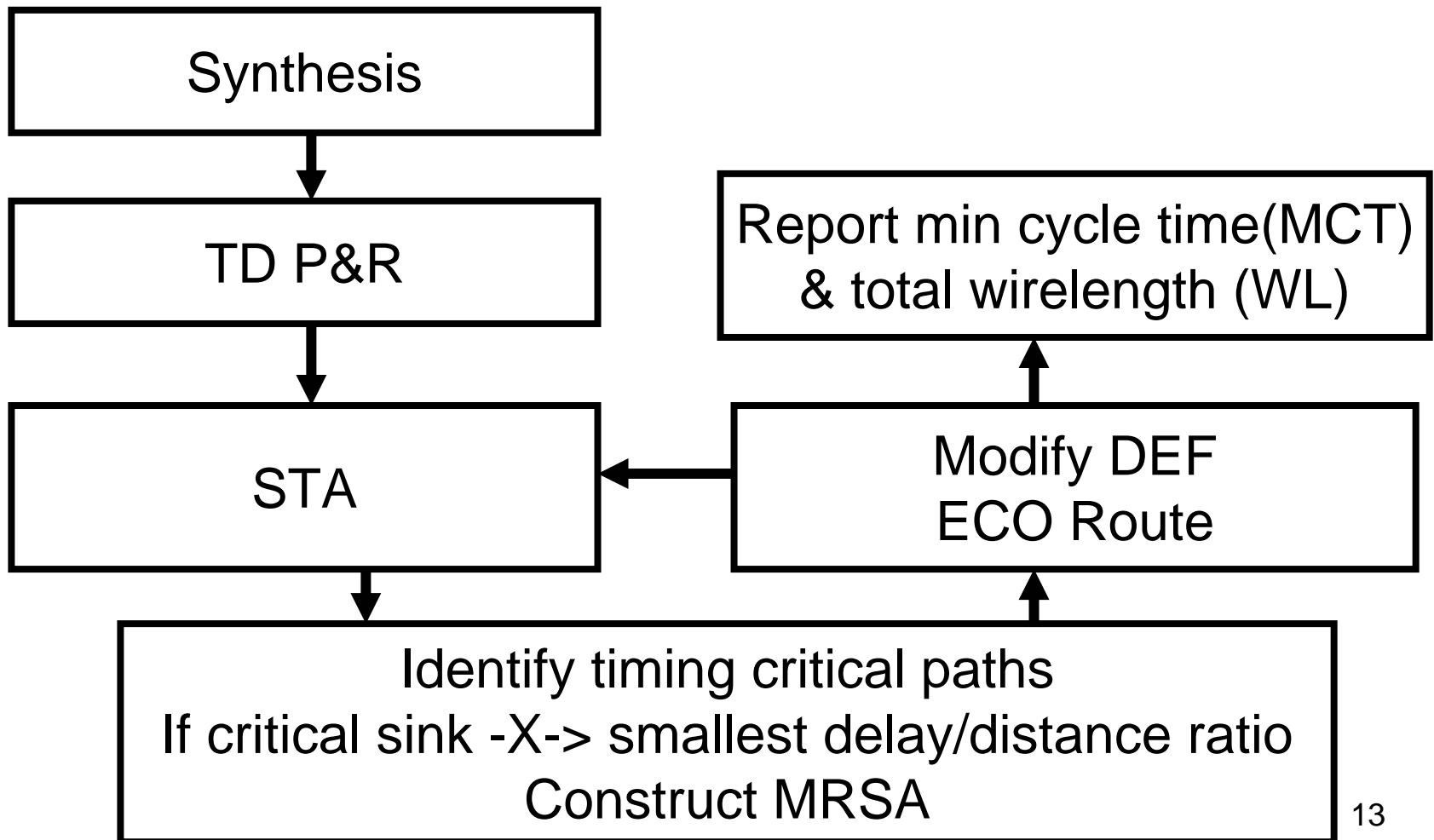
- RMST : radius-ratio as most 6x the optimal
- PD1-1.0 : very close to a MRSA
- MRSA : 4% WL penalty on average
- SERT : highest WL penalty
- PD1 : better trade-off between WL and radius-ratio
- k-SMT and PD2 : also provide a good trade-off, but not as good as PD1

MRSA for Real Designs

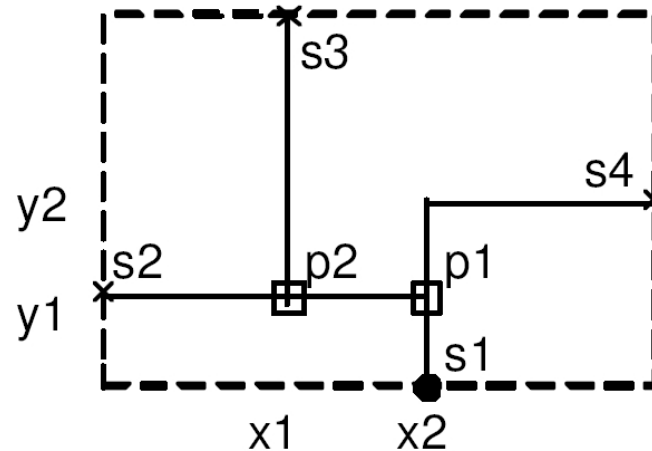
Designs	design1	design2	design3	design4	design5	design6
#Nets	146k	330k	223k	168k	342k	342k
RSMT-Total WL	124057096	610531613	88485247	116000653	154572475	57887506
MRSA-Total WL	124453400	619629806	89660129	116631947	155663525	59050751
WL-Increase (%)	0.32	1.49	1.33	0.54	0.71	2.01



MRSA for Physical Synthesis Flow



Force the Router to Use MRSA



```
+ VPIN P1 ( -h -w ) ( h w ) FIXED ( x2 y1 ) S
+ VPIN P2 ( -h -w ) ( h w ) FIXED ( x1 y1 ) S
+ SUBNET N1 ( PIN s1 ) ( VPIN p1 )
+ SUBNET N2 ( VPIN p1 ) ( PIN s4 ) ( VPIN p2 )
+ SUBNET N3 ( VPIN p2 ) ( PIN s2 ) ( PIN s3 )
```

Experiment Settings

- Two designs from OpenCores
 - AES, JPEG

Design	Utilization	Minimum Cycle Time	WL	#Vias	#Cells	#Nets
AES	70%	2.490ns	8.784	186068	25187	25446
JPEG	70%	3.605ns	27.85	836543	135652	135672

Results

- Cycle time improvement (4% ~ 5%)
- Insignificant WL increase (-2.41% ~ +0.01%)

Design	g	#Nets	MCT-reduction(%)	WL-change(%)	#Vias	#Vios
AES	0.01	14	4.68	0.01%	186133	2
	0.05	18	4.87	0.01%	186147	0
	0.09	22	5.05	0.01%	186149	0
JPEG	0.01	131	0.38	0.04%	836686	0
	0.03	915	2.5	-0.97%	828663	0
	0.05	1786	2.62	-1.47%	823711	0
	0.07	3151	3.59	-2.41%	814497	0

-
- Due to space limit, not all results are shown in paper
 - For a full version of the paper, please send me an email

Thank you

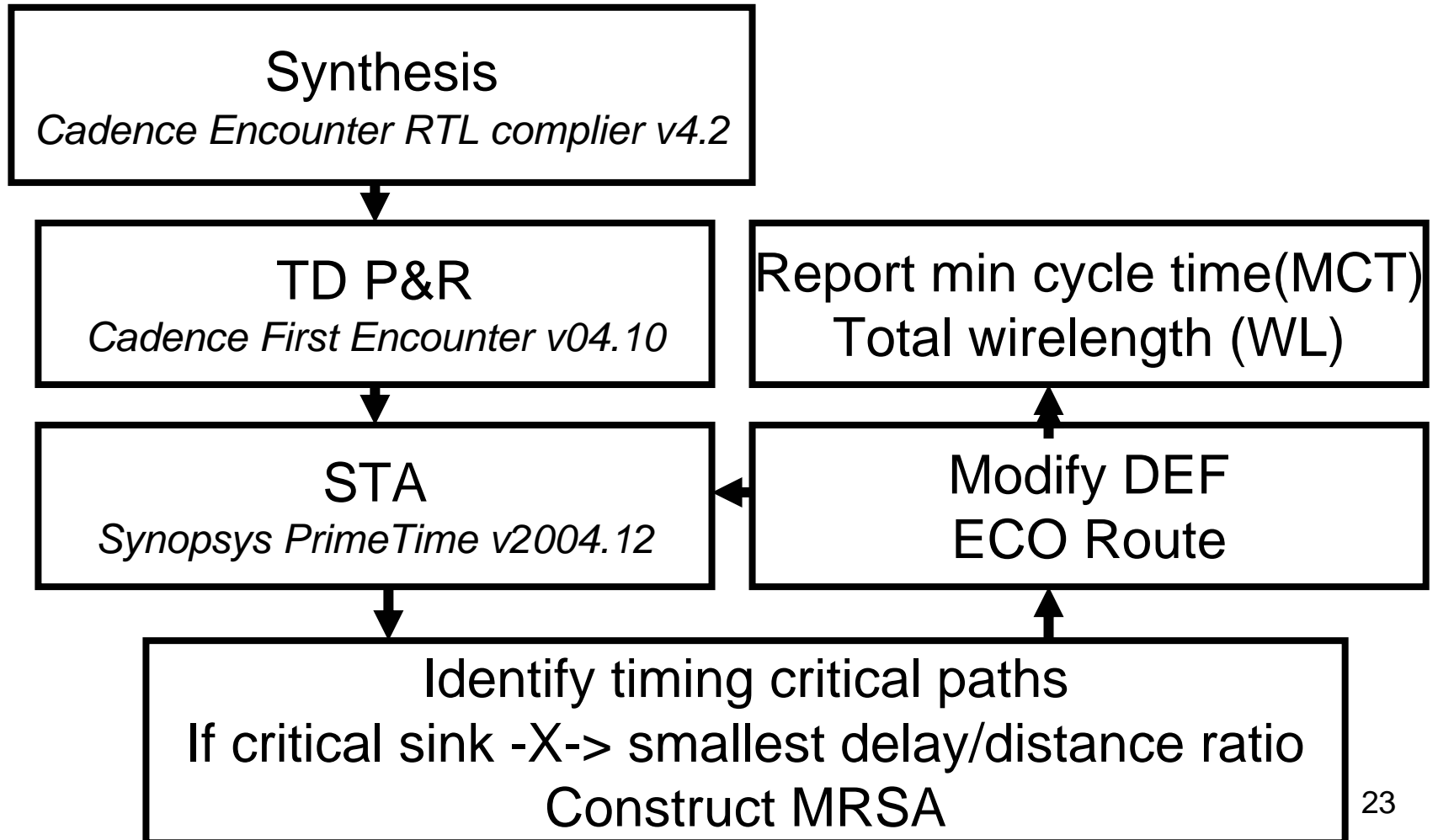
Q&A

Two Simple Heuristics

- H1
 - Starts from an RMST
 - For each sink, if the radius-ratio is bigger than a threshold, disconnect and reconnect it to the tree with shortest path length
- H2
 - Starts from an MRSA
 - Relaxes the radius-ratio
 - Improves wirelength with BOI-style edge addition and removal

Experiment Results

MRSA for Physical Synthesis Flow

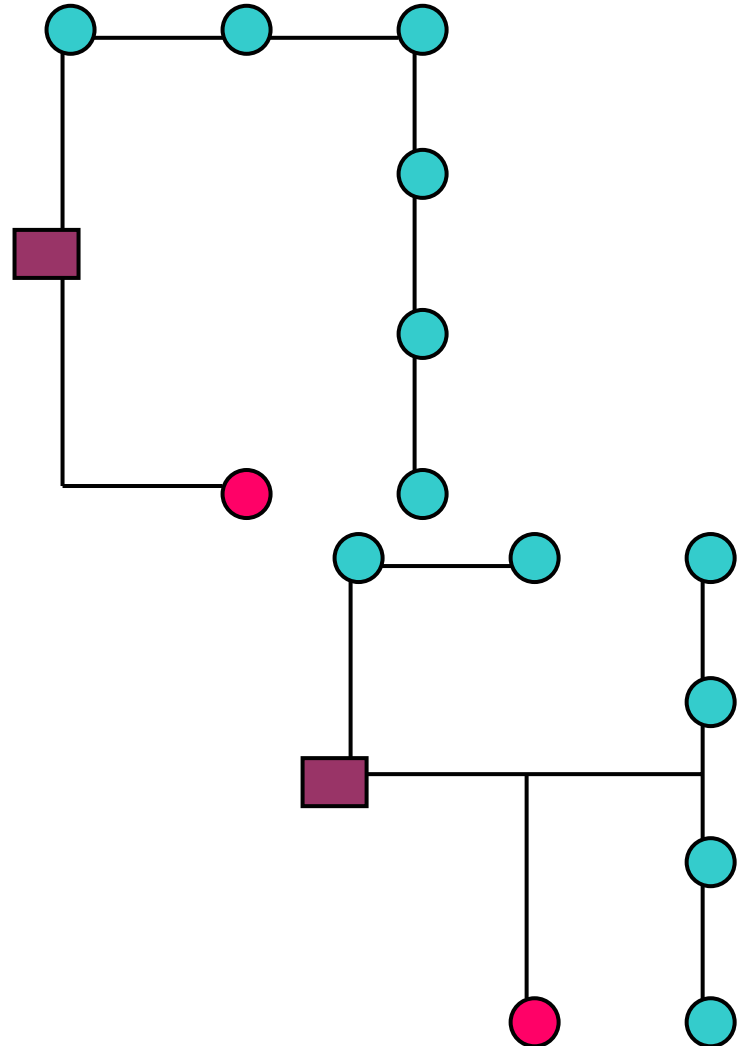


MRSA for Physical Synthesis Flow

- Synthesis : UMC 90nm library
- Physical Design : Cadence First Encounter v04.10
- STA : Synopsys PrimeTime v2004.12
- Identify timing-critical paths
- For all sinks in a tree, if the sink with smallest slack does not route with smallest Generate a MRSA for the tree

Timing-driven RST Algorithms

- Steiner Elmore delay tree (SERT, SERT/C)
 - Greedy algorithm to minimize Elmore delay to all sinks (or the critical sink)
 - The timing assertion varies a lot during physical optimization
 - WL penalty
- Rectilinear Steiner arborescence
 - All shortest source-sink path
 - WL penalty



The Big Picture

- Why most people uses RSMT as routing estimation?
- What is the problem in other timing-driven trees?
- How bad is timing-driven trees?
- What if we try timing-driven trees for routing estimation?