Timing Driven Placement for Quasi Delay-Insensitive Circuits

Robert Karmazin, Stephen Longfield Jr.,
Carlos Tadeo Ortega Otero, Rajit Manohar

May 5th, 2015

CSL

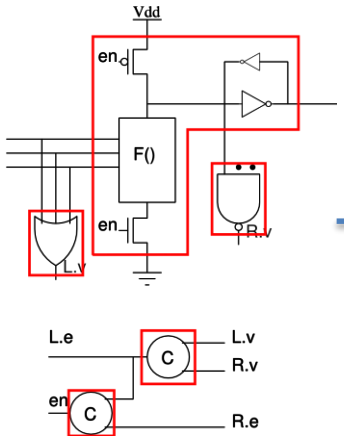
AVLSI

Introduction

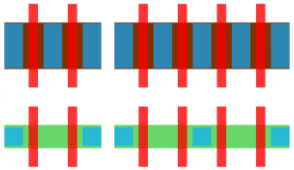
- Chip complexity has been scaling with transistor count
- Designer productivity not keeping up
- Automation key to leveraging device scaling potential
- Bring synchronous productivity to asynchronous flows

cellTK Overview

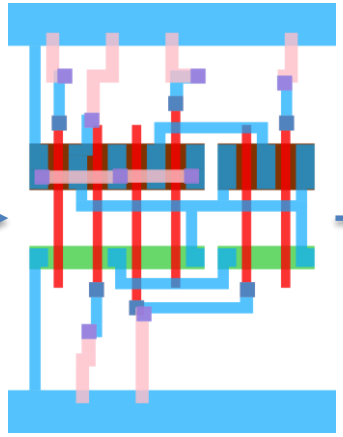
Partition



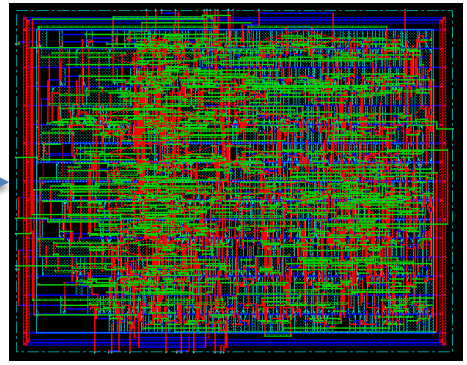
Transistor Place



Transistor Route



Cell P&R

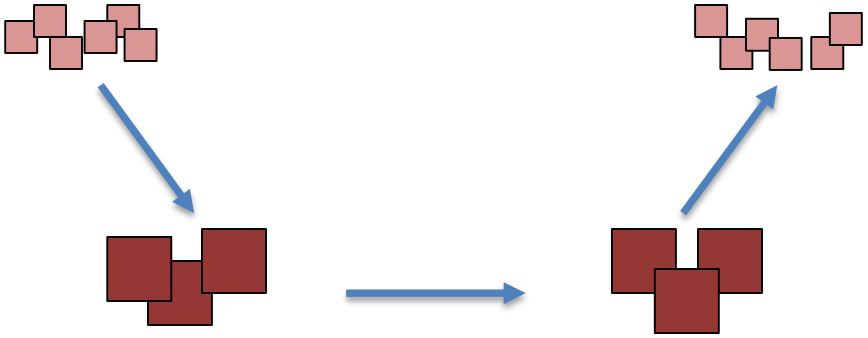


Summary

1. Introduction
2. Placer Background
3. Timing Driven Placement for QDI Circuits
4. Evaluation
5. Conclusion

Analytical Placer Stages

1. Global Placement



2. Legalization
3. Detailed Placement
   - Local window-based approach

Placement Formulation

- Objective Function:

$$\min_{\boldsymbol{x},\boldsymbol{y}} W(\boldsymbol{x},\boldsymbol{y})$$

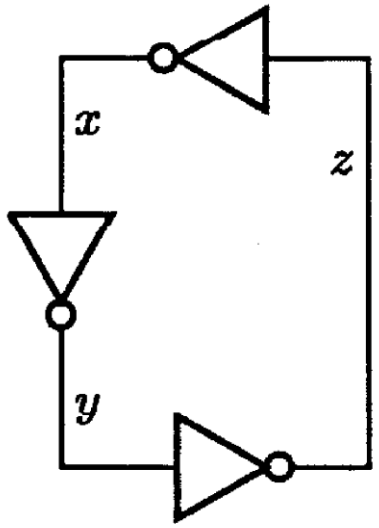
- Constraints:

$$D_b(\boldsymbol{x},\boldsymbol{y}) \leq M_b$$

- Reformulate using the penalty method

Timing Model

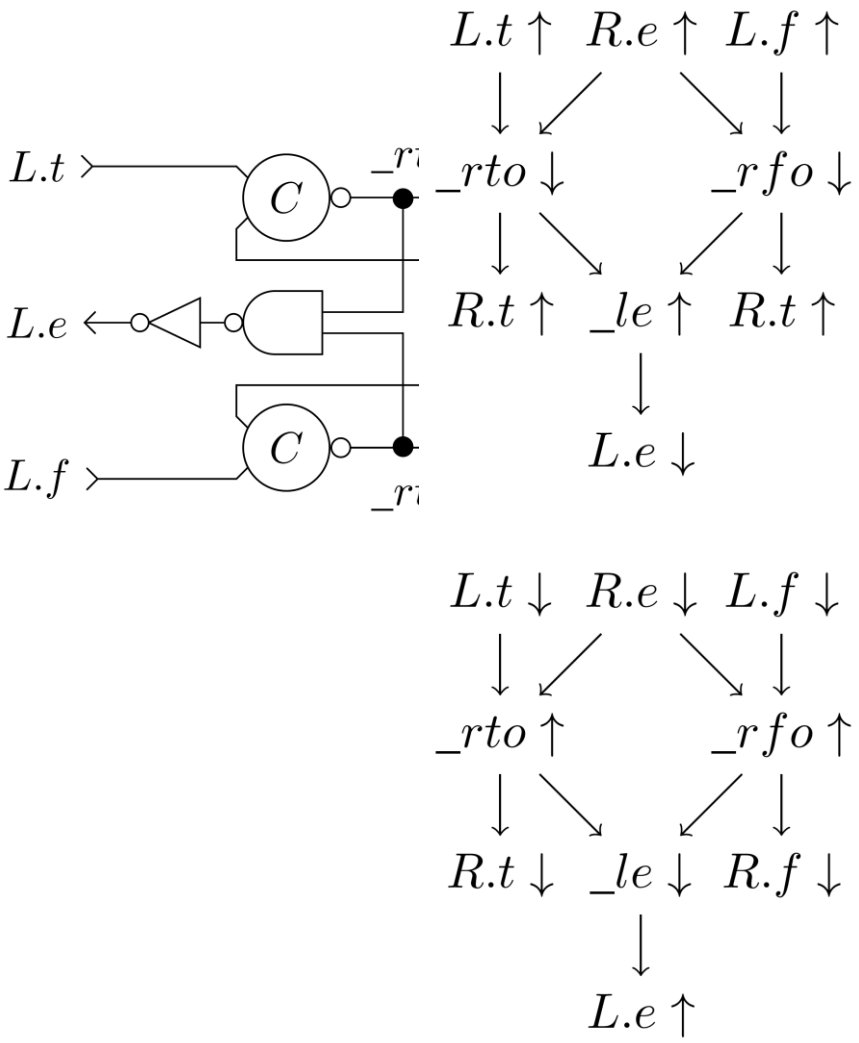
- Repetitive Event Rule (RER) Systems

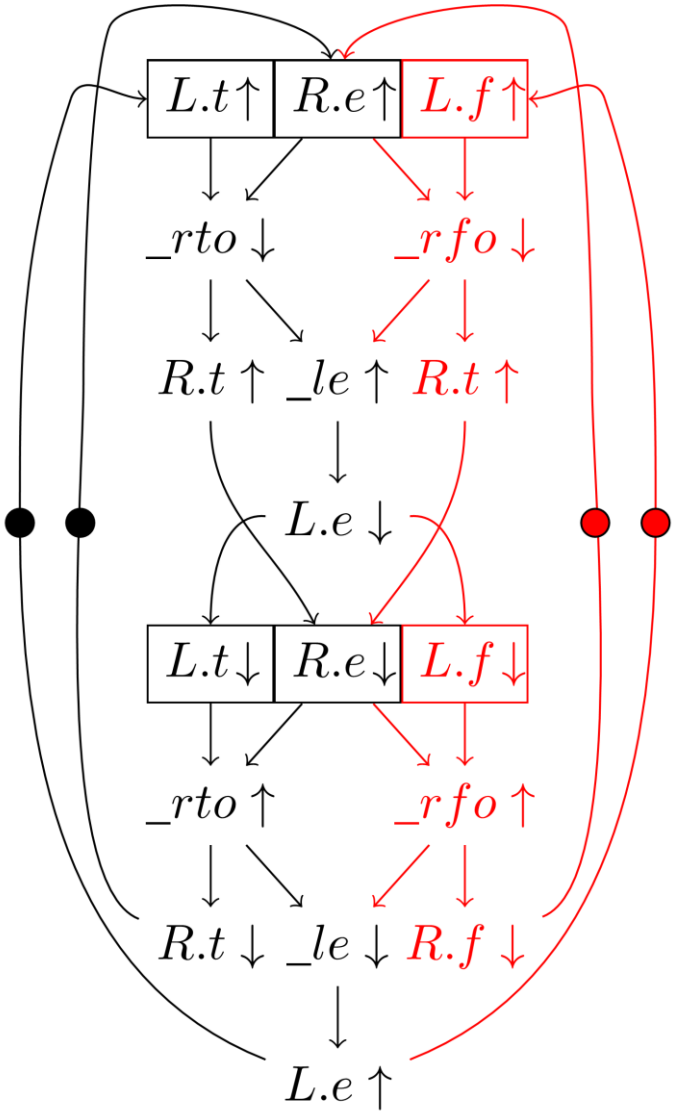


$$
\begin{aligned}
E' &= \{x\uparrow, y\uparrow, z\uparrow, x\downarrow, y\downarrow, z\downarrow\} \\
R' &= \{\ \langle z\downarrow, i-1\rangle \overset{\alpha_{x\uparrow}}{\mapsto} \langle x\uparrow, i\rangle, \\
&\quad\ \langle x\uparrow, i\rangle \overset{\alpha_{y\downarrow}}{\mapsto} \langle y\downarrow, i\rangle, \\
&\quad\ \langle y\downarrow, i\rangle \overset{\alpha_{z\uparrow}}{\mapsto} \langle z\uparrow, i\rangle, \\
&\quad\ \langle z\uparrow, i\rangle \overset{\alpha_{x\downarrow}}{\mapsto} \langle x\downarrow, i\rangle, \\
&\quad\ \langle x\downarrow, i\rangle \overset{\alpha_{y\uparrow}}{\mapsto} \langle y\uparrow, i\rangle, \\
&\quad\ \langle y\uparrow, i\rangle \overset{\alpha_{z\downarrow}}{\mapsto} \langle z\downarrow, i\rangle \\
&\quad\ \}\ .
\end{aligned}
$$

- Built from Index-Priority Simulation

Timing Model





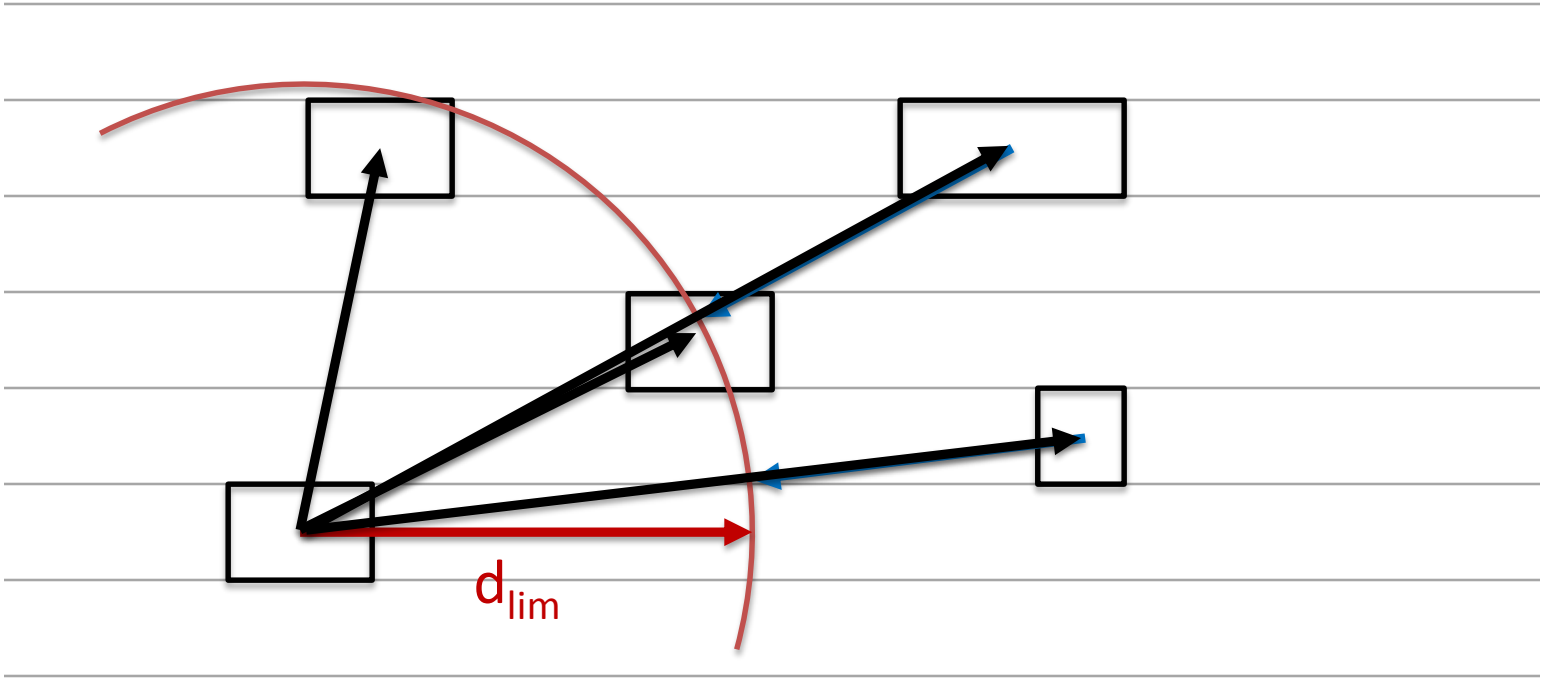
A-NTUPlace: Net Weights

1. Clustering
   A. Based on affinity value (connectivity, cell area)
   B. Net weight make cells more likely to be clustered
2. LSE Wire Model

$$\gamma \sum_{e \in E} w_e \left(\ln \sum_{v_k \in e} \exp \frac{x_k}{\gamma} + \cdots \right)$$

3. Detailed Placement
   A. Adjust cell-swapping cost matrix

A-NTUplace: Isochronic Forks



$$\min(\max d_n - \min d_n) \qquad \min d_n \leq d_{lim} < \max d_n$$

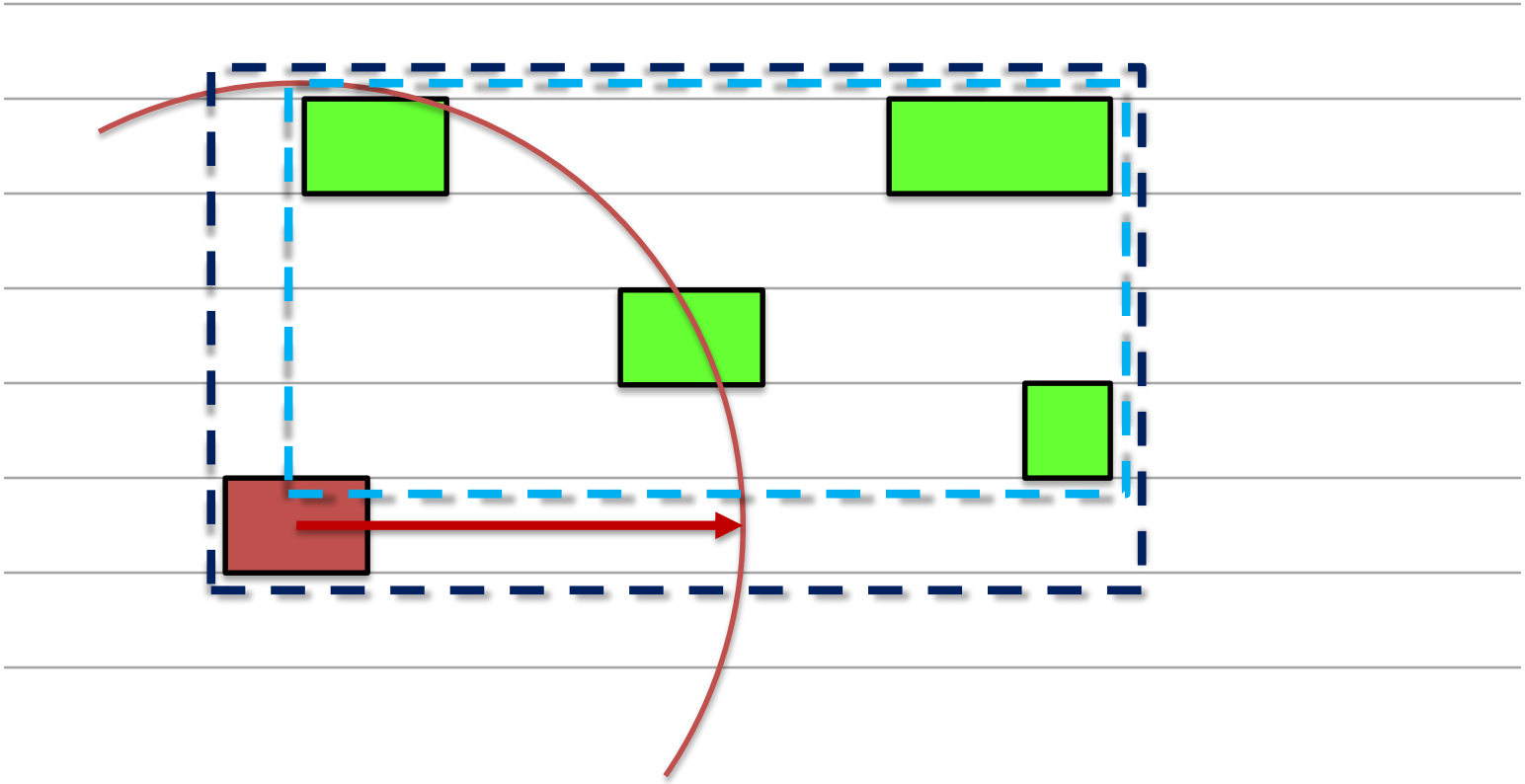
A-NTUplace: Isochronic Forks

- Global Placement Formulation

$$\min_{\boldsymbol{x},\boldsymbol{y}} W(\boldsymbol{x},\boldsymbol{y})$$
$$D_b(\boldsymbol{x},\boldsymbol{y}) \leq M_b$$
$$\forall n \in FO(e), d_n \leq d_{lim,e}$$

- Detailed Placement Formulation

A-NTUplace: Isochronic Forks



A-NTUplace: Isochronic Forks

- Global Placement Formulation

$$\min_{x,y} W(\boldsymbol{x}, \boldsymbol{y})$$
$$D_b(\boldsymbol{x}, \boldsymbol{y}) \leq M_b$$
$$\forall n \in FO(e), d_n \leq d_{lim,e}$$

- Detailed Placement Formulation

$$P_{n,e} = (d_n - d_{lim,e})^2$$

$$P_{m,e} = \frac{HPWL_{FO(e)}}{HPWL} \sum_{n \in FO(e)} (d_n - d_{lim,e})^2$$

Evaluation

- Four benchmarks from ULSNAP

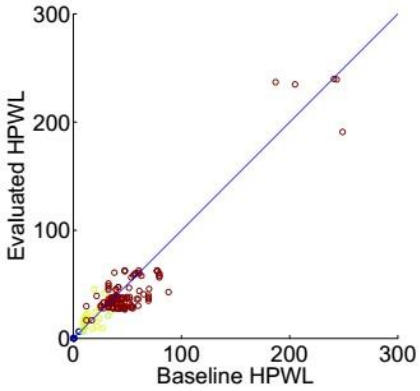
Benchmark	Unique Cells	Total Cells	Total Nets	Isochronic Forks	Commercial Baseline
Logic	13	103	178	71	8437
Shift	51	1065	1132	174	66712
Decode	56	282	329	54	26048
Fetch	100	571	636	184	39975

Results: Timing Model

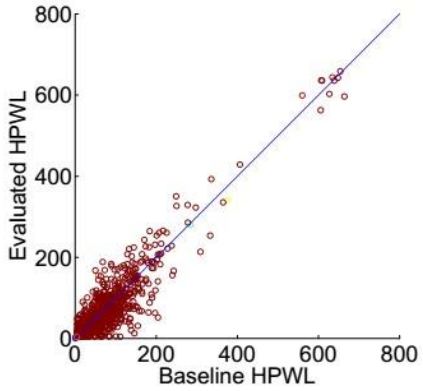
- Accuracy

Benchmark	Cycle Time Error	False Pos Critical	False Neg Critical	RMSD Slack
Logic	0.00%	0.00%	8.33%	1.20%
Shift	67.62%	0.00%	1.60%	0.90%
Decode	0.07%	17.76%	4.63%	7.30%
Fetch	8.60%	3.34%	3.45%	3.30%

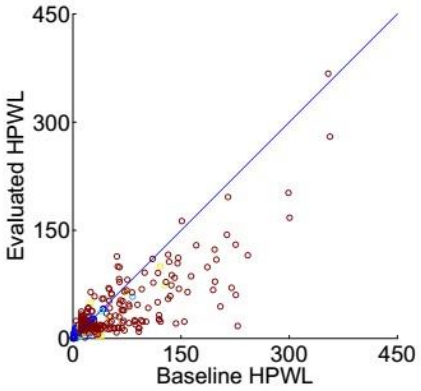
Results: HPWL with Net Weights



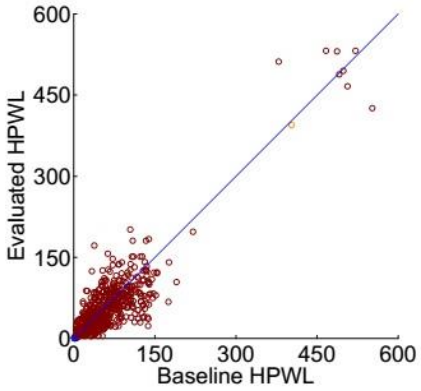
(a) Logic



(b) Shift



(c) Decode



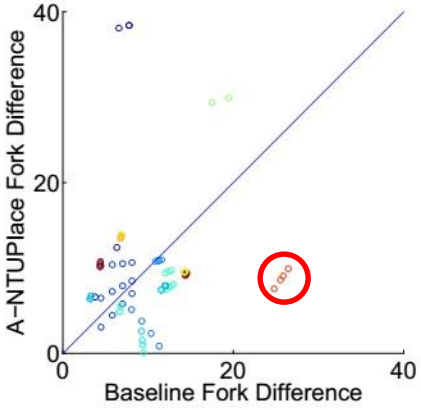
(d) Fetch

Legend: More Critical
Less Critical

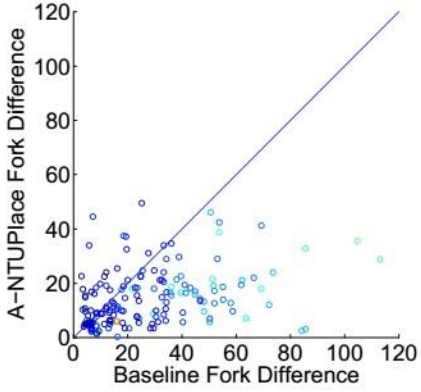
Results: Isochronic Fork Balancing

Benchmark	Total Forks	Improvement	
		Global	Detailed
Logic	71	52%	62%
Shift	174	70%	76%
Decode	54	65%	76%
Fetch	184	46%	42%

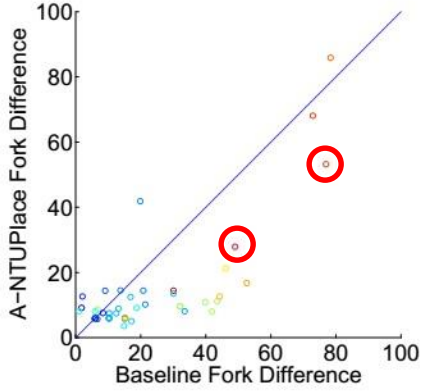
Results: Isochronic Fork Balancing



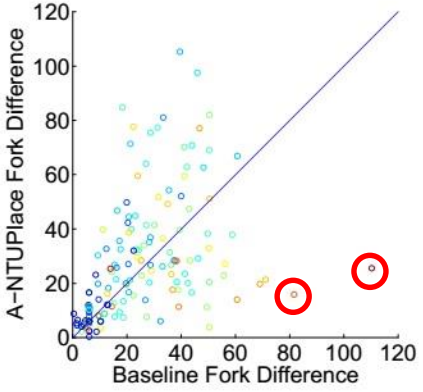
(a) Logic



(b) Shift



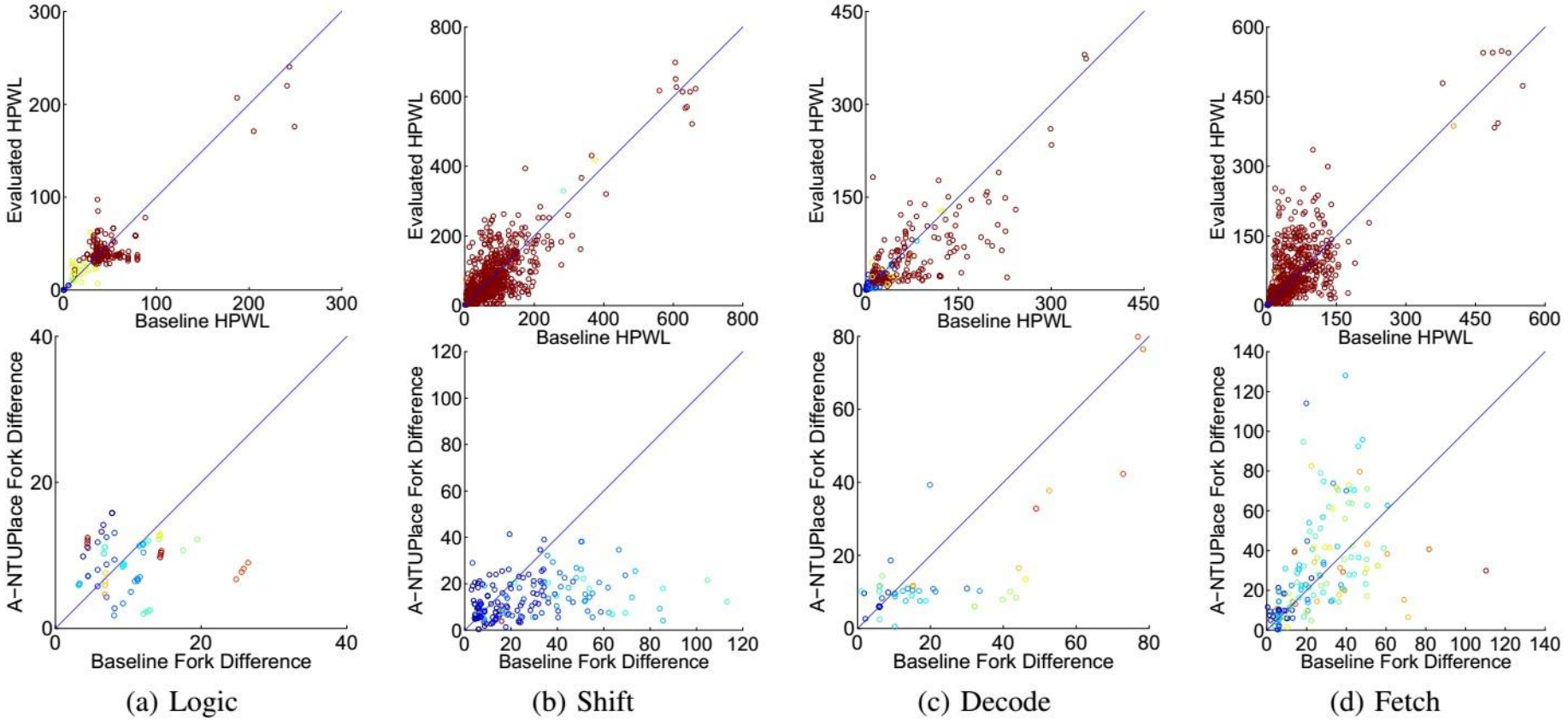
(c) Decode



(d) Fetch

Legend: Large Baseline HPWL
Small Baseline HPWL

Results: Net Weights and Balancing



(a) Logic (b) Shift (c) Decode (d) Fetch

Conclusion

- Timing Driven Placer for QDI Circuits
  - Created fast yet accurate timing model
  - Adapted leading academic placer to respect net criticality and isochronic constraints
- Embedded timing model and placer into *cellTK* flow
- Results are comparable to commercial placement solutions

Timing Driven Placement for Quasi Delay-Insensitive Circuits

Robert Karmazin, Stephen Longfield,
Carlos Tadeo Ortega Otero, Rajit Manohar

May 5th, 2015

CSL

AVLSI

Backup: Placer Categories

1. Stochastic
   - Simulated Annealing
   - Ex. Timberwolf
2. Partitioning
   - Min-cut Heuristic
   - Ex. Capo, Dragon
3. Analytical
   - Mathematical Programming
   - Ex. Kaftwerk2, Gordian, FastPlace, APlace, NTUPlace

Backup: Wire Models

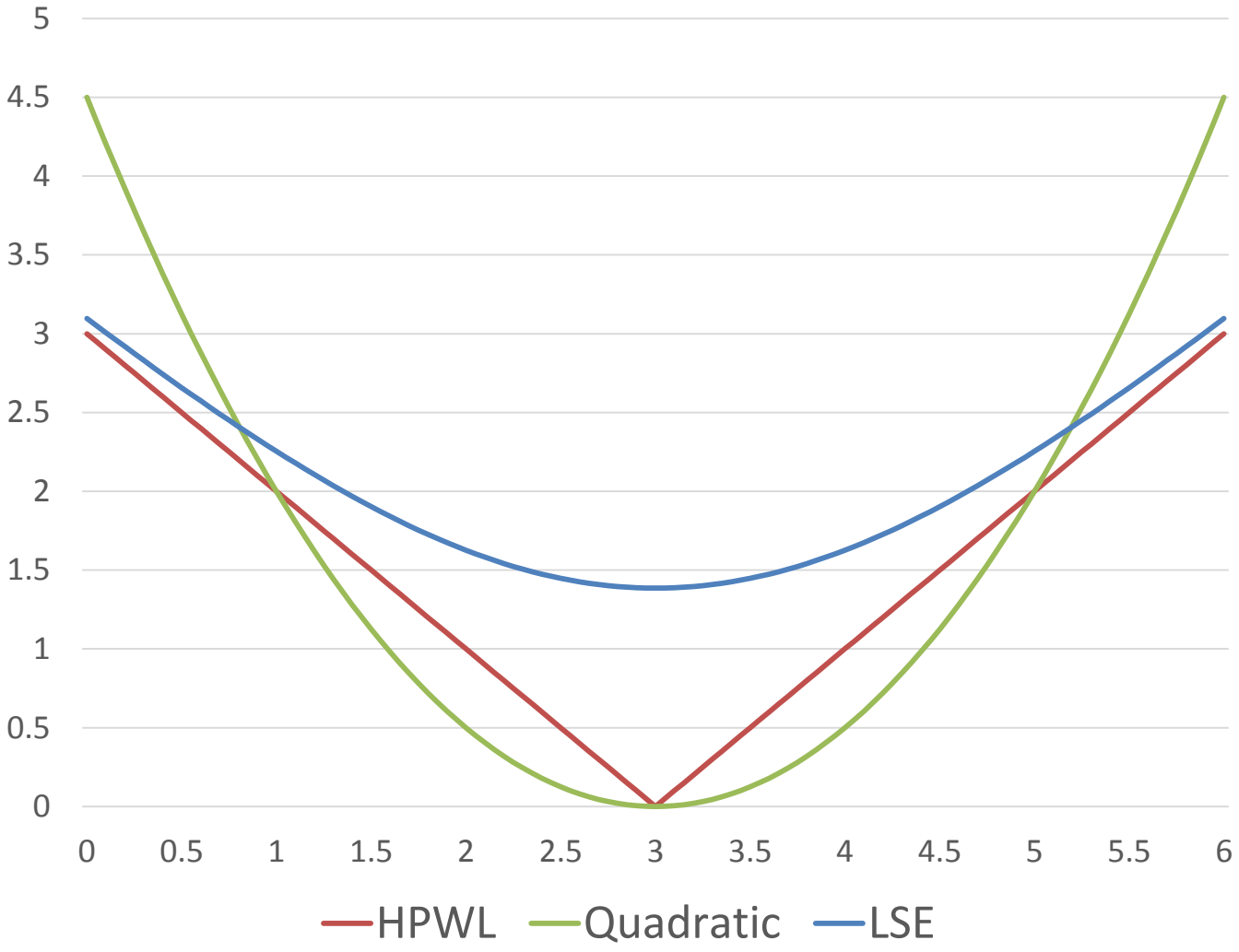
- Wire Model:

$$HPWL(\boldsymbol{x},\boldsymbol{y}) = \sum_{e\in E}\left(\max\left|x_i - x_j\right| + \max\left|y_i - y_j\right|\right)$$

- Log-Sum-Exp Wire Model

$$\gamma \sum_{e\in E}\left(\begin{array}{l}\ln \sum_{v_k\in e} \exp\frac{x_k}{\gamma} + \ln \sum_{v_k\in e} \exp\frac{-x_k}{\gamma} + \\ \ln \sum_{v_k\in e} \exp\frac{y_k}{\gamma} + \ln \sum_{v_k\in e} \exp\frac{-y_k}{\gamma}\end{array}\right)$$

Wire Models


5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
HPWL
Quadratic
LSE