Keep It Straight: Teaching Placement how to Better Handle Designs with Datapaths

Samuel I. Ward, Myung-Chul Kim*, Natarajan Viswanathan*,
Zhuo Li*, Charles Alpert*, Earl E. Swartzlander, Jr., David Z. Pan

ECE Dept. The University of Texas at Austin, Austin, TX 78712
* IBM Austin Research Laboratory, 11501 Burnet Road, Austin, TX, 78758
{wardsi}@utexas.edu, {mckima}@umich.edu, {nviswan, lizhuo, alpert}@us.ibm.com,{eswartzla}@aol.com, {dpan}@cerc.utexas.edu

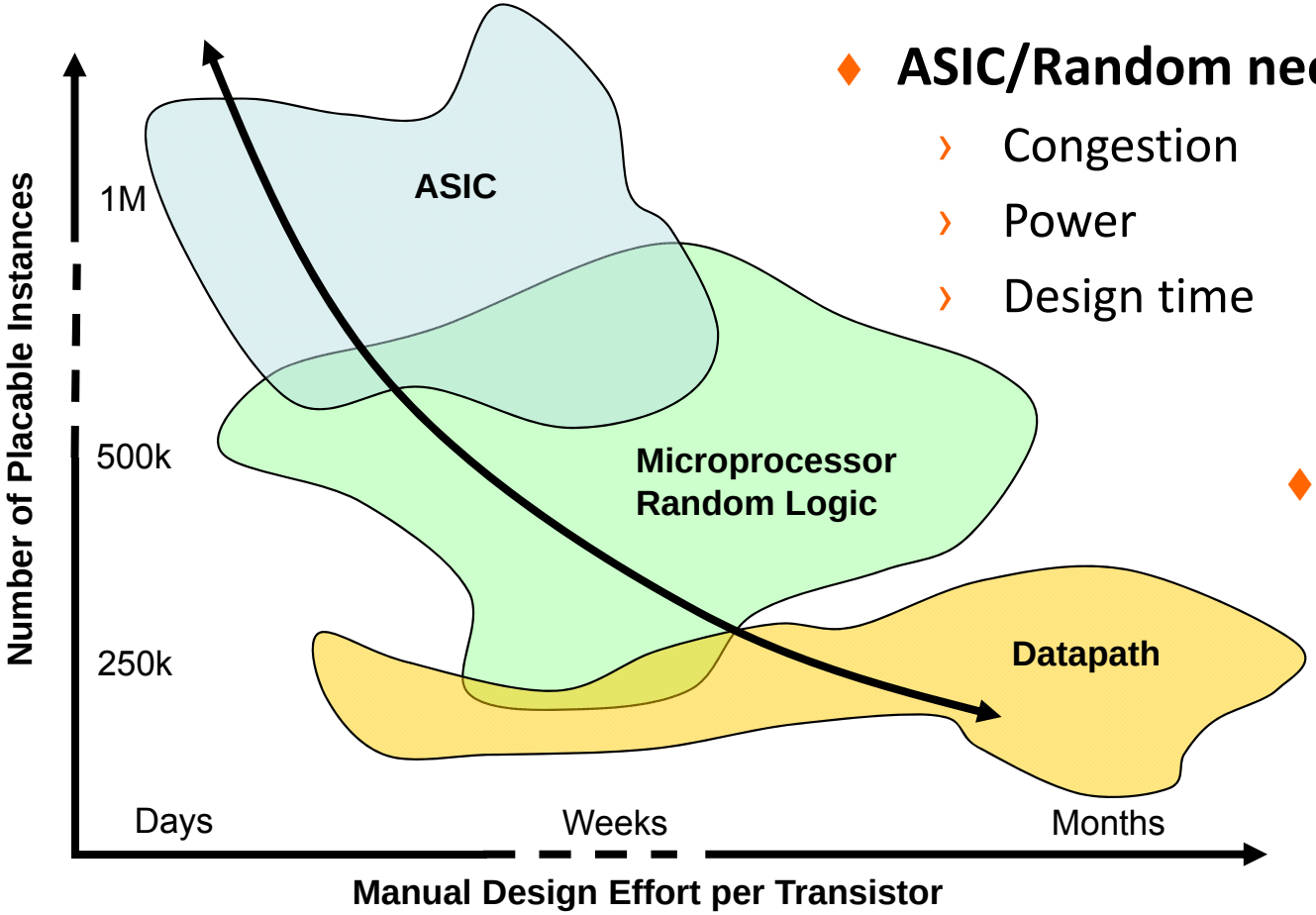
Dept. of Electrical and Computer Engineering
The University of Texas at Austin

Outline

- **General Placement Overview and Motivation**
  - Why is the current formulation a problem?
  - Key Contributions
- **Structure Aware Placement Techniques (SAPT)**
  - **Global Placement Techniques**
    - Skewed net weighting with step size scheduling
    - Fixed-point and pseudo net alignment constraint
  - **Detailed Placement Techniques**
    - Bit-stack aligned cell swapping
    - Datapath group repartitioning
- **Experimental Results**
- **Future Work**
  - Placement
  - Congestion

Why is There A Big Difference?

- **Datapath Needs to <u>Increase</u>**
  - **Circuit Performance:** Timing, congestion, and power
  - **Manpower Performance:** Design time, controllability
  - **Stability:** Drives design closure

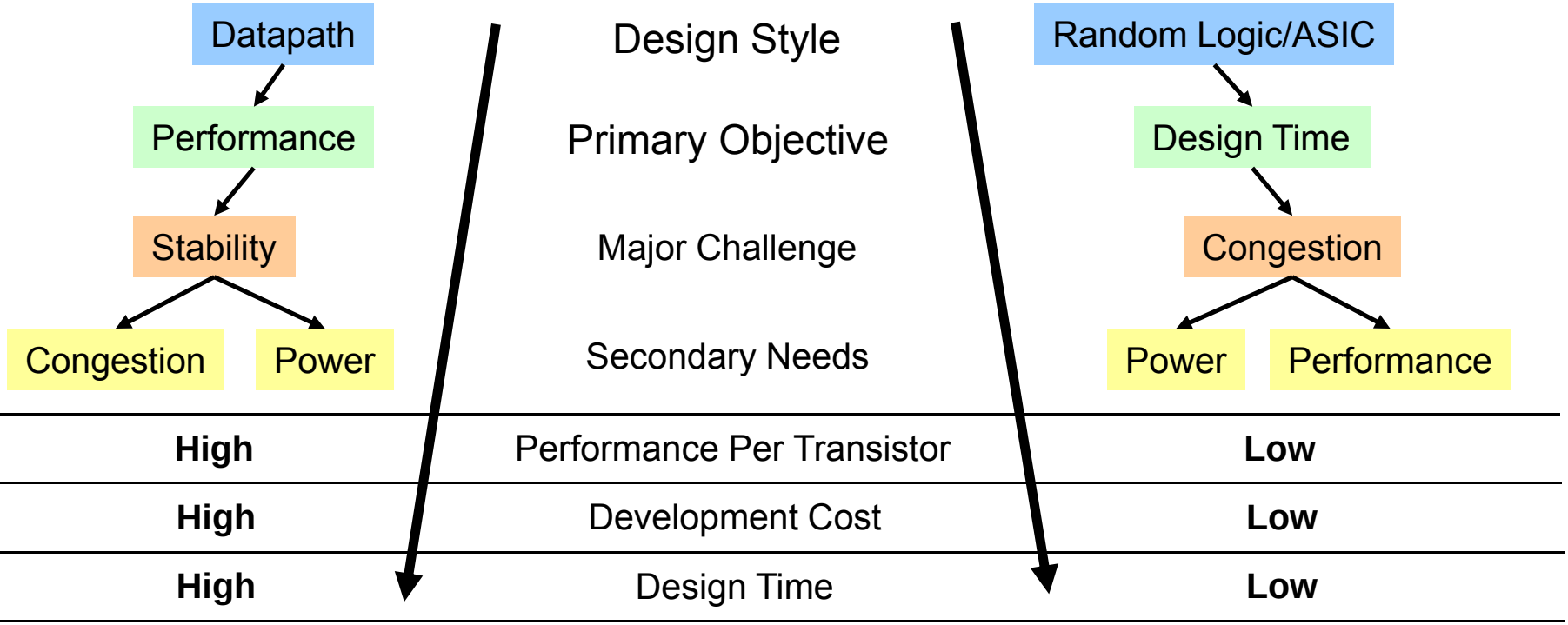


- **ASIC/Random needs to <u>Lower</u>**
  - Congestion
  - Power
  - Design time

- **Where does this lead?**

Two Worlds: Samuel's Hierarchy of Design Needs

- **Modern industrial designs have two flows…why?**
  - Different needs -> primary objective is different
  - Different styles -> tools tuned differently
- **With different objectives can we unify the placement flow?**
  - Which flow should we use?

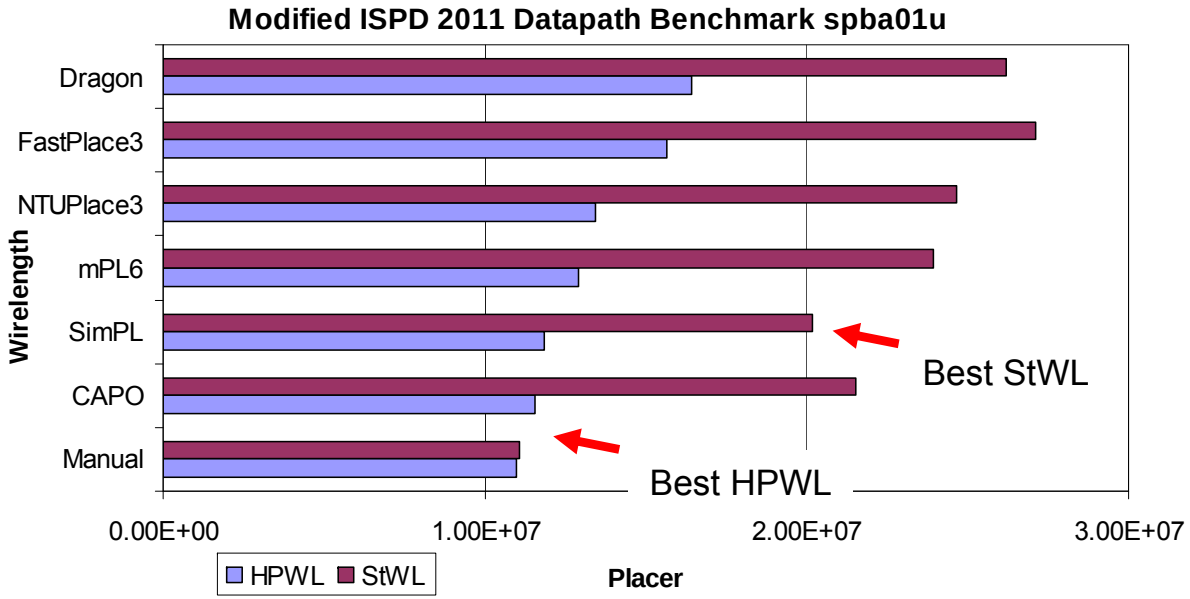


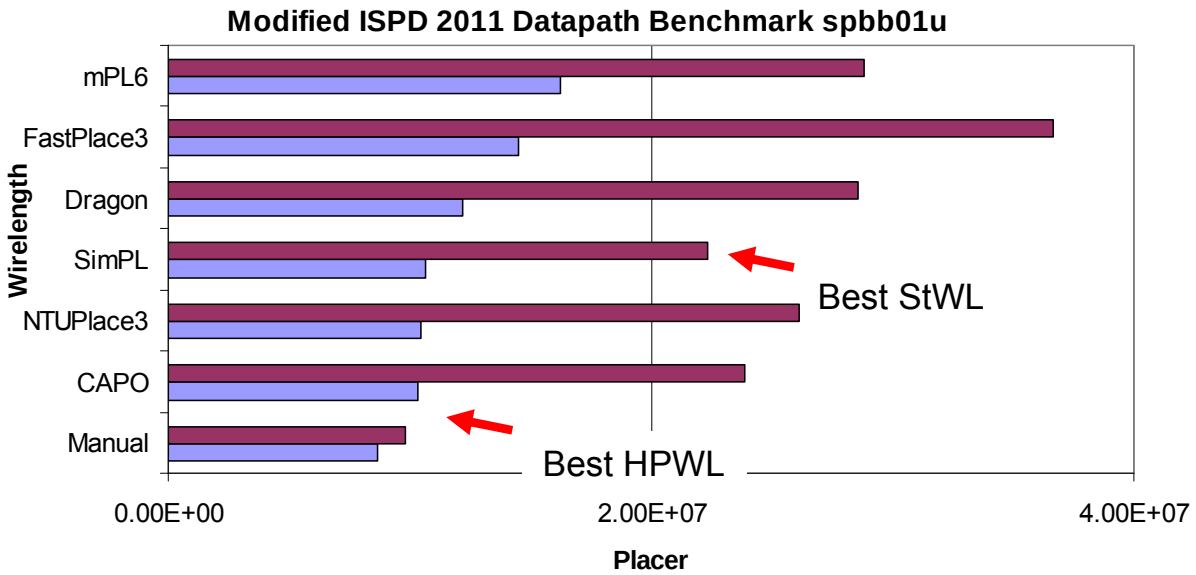
How Do We Unify the Placement Flow?

- Should we:
  - Develop a datapath placer able to place random logic?
  - Enhance current placers to place datapath logic?
- Wide industry acceptance of the random placer
  - Speed is impressive
  - Quality is impressive
- BUT, can we enhance placers for datapath?

HPWL: Does the Model Hold for Datapath?

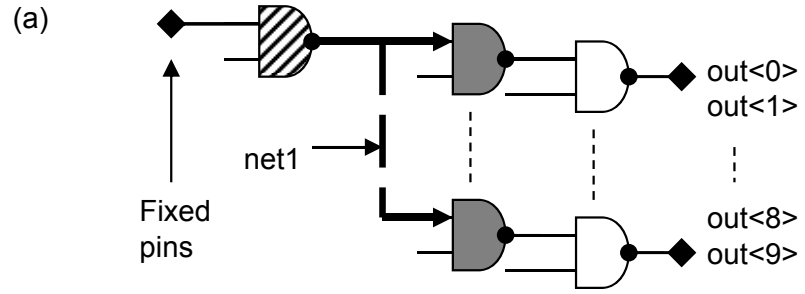
- **Major observations:**
  - HPWL Accuracy
  - HPWL Fidelity





- **Surprising questions:**
  - Is HPWL the right model for dp placement evaluation?
  - Are there specific structures causing this issue?

Datapath HPWL Fidelity Example

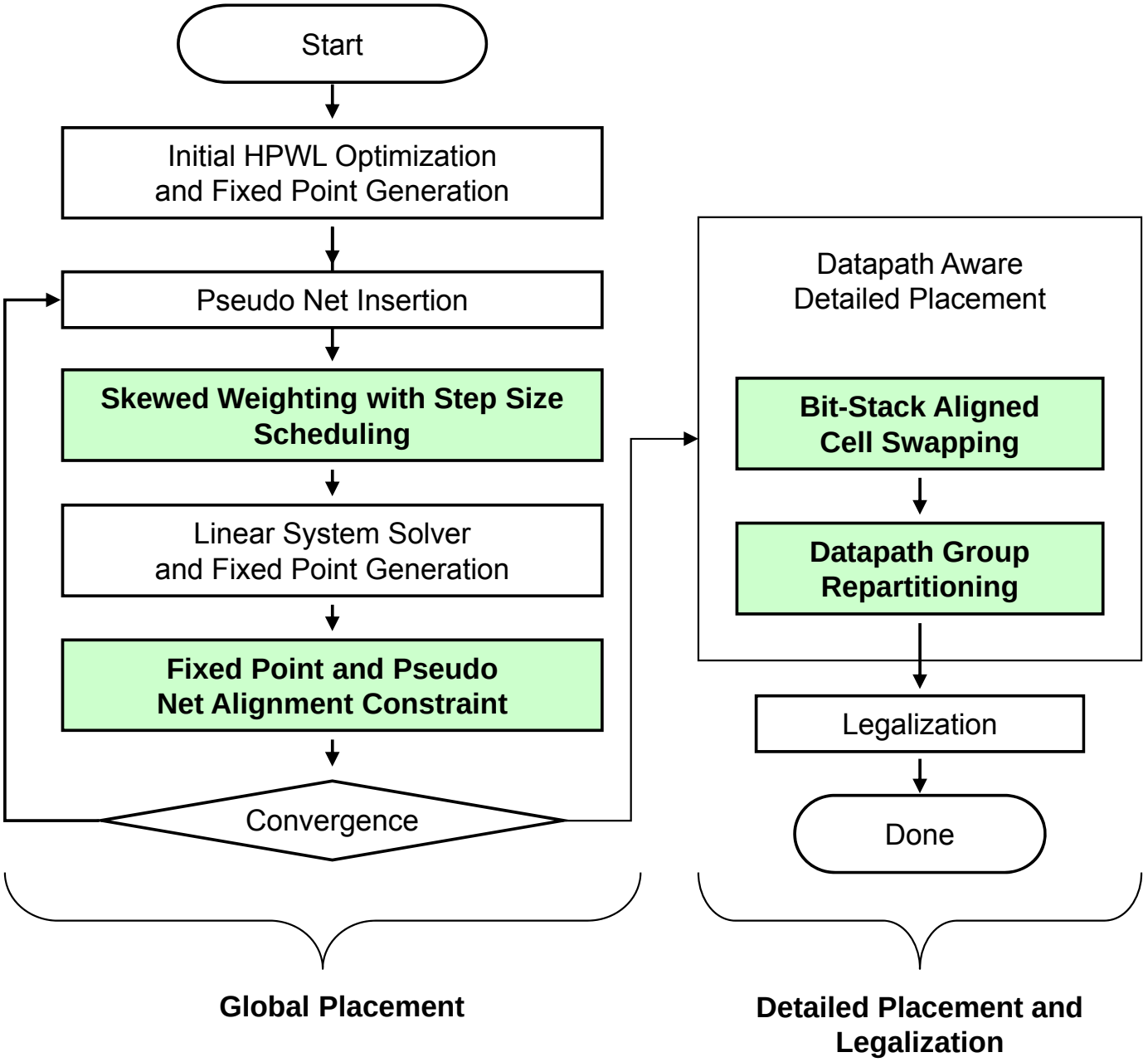


- **Why exactly are the placement solutions bad?**
- **HPWL model is:**
  - exact for 2-pin and 3-pin nets
  - underestimate for > 3-pin nets
- **StWL more accurately represents routed wirelength (RWL)**
- **Manually placed circuit:**
  - HPWL: 2% worse
  - StWL: 9% better
- **Based on this, can we:**
  - Integrate alignment constraint instead of optimizing StWL directly?

Key Contributions of this Work

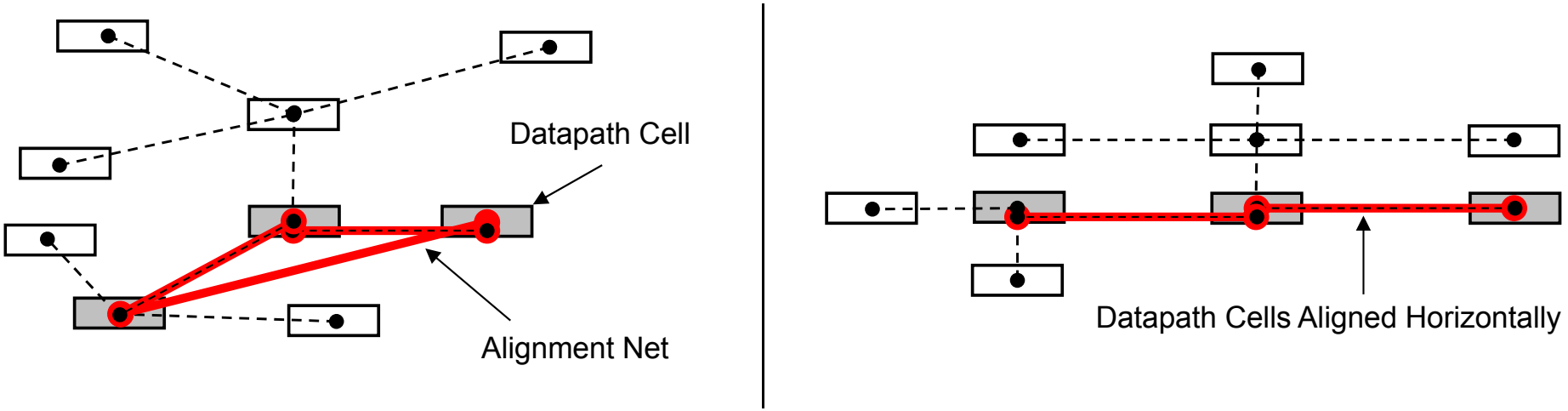
- **Goals:**
  - **Integrate alignment constraint into force-directed placement**
  - **Simultaneously place datapath and random logic**
- **Key Contributions**
  - **Study of obstacles to current academic placers:**
    - Inadequacies of the HPWL model for datapath logic
  - **Key insight to StWL improvement through bit-stack alignment:**
    - Alignment of the bitstack guides indirect StWL optimization
    - Significantly improves total StWL and routing congestion
    - Causes other cells to align
  - **Novel placement techniques:**
    - Skewed Weighting with Step Size Scheduling
    - Fixed-Point Alignment Constraint
    - Bit-Stack Aligned Cell Swapping
    - Datapath Group Repartitioning

Overall Flow

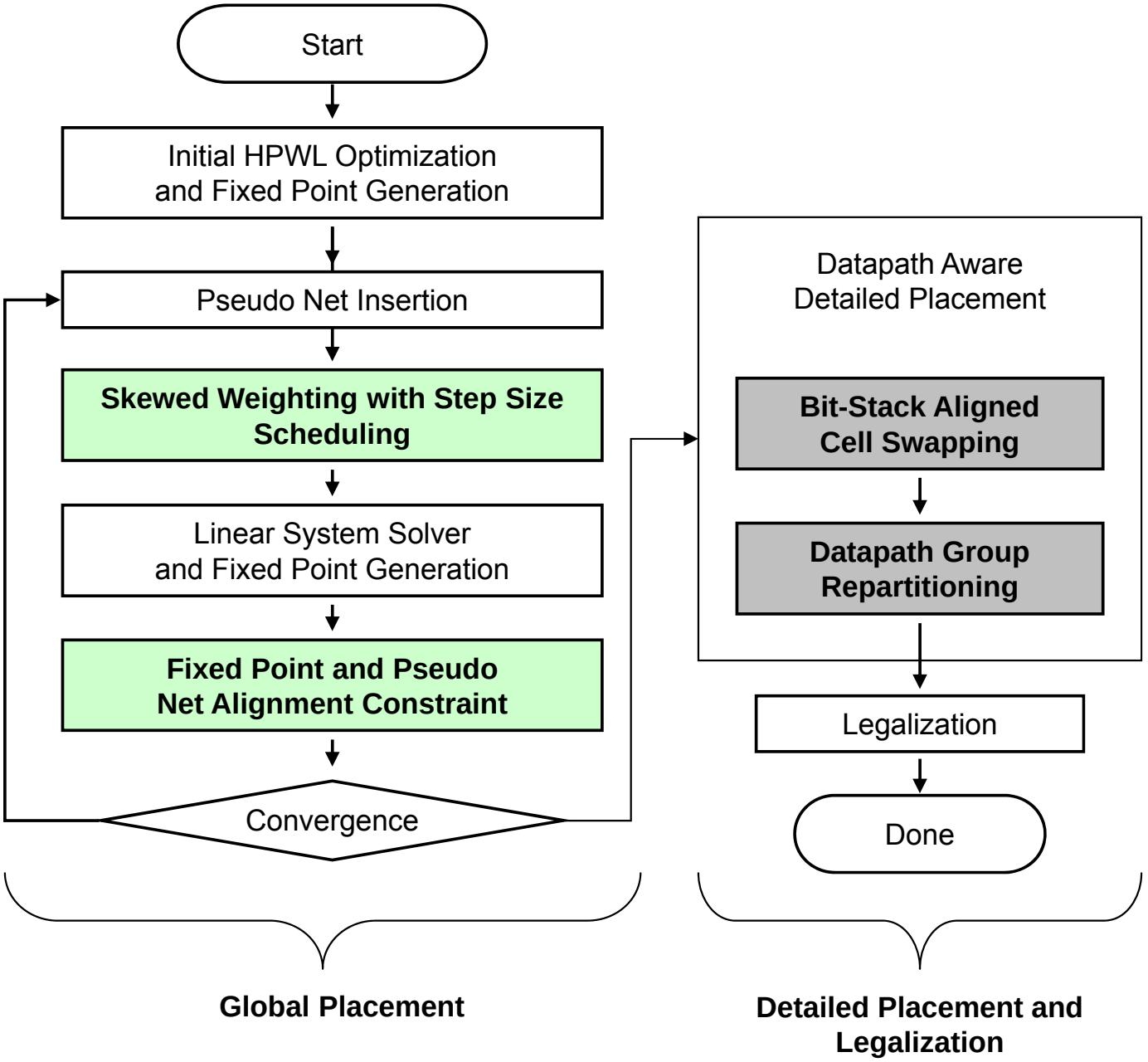


Alignment Net

- **Example of an alignment net**
- **A weighted multi-pin connection**
- **Connects between cells in a datapath group**
- **Modeled using the Bound2Bound model**



Skewed Weighting with Step Size Scheduling



Skewed Weighting with Step Size Scheduling

- **Method for creating an alignment constraint during global placement**
  - Skew net weighting along datapath direction
  - Cells align that are connected to the alignment net
  - Gradually increase the weighting

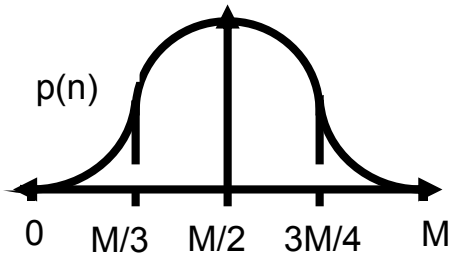
- **Manipulate the skewed weighting**
  - n Global placement iteration number
  - d_k Datapath Direction
  - β Scaling factor
  - $\delta_{i,j}$, $\gamma_{i,j}$ Horizontal and vertical alignment net weight
  - p(n) Step function
  - $\sigma^2(n)$ Cell position variance
  - w_{ij} User Net weighting

$$\gamma^n = \gamma^{n-1} + \hat{\mathbf{y}} \cdot \overrightarrow{\mathbf{d_k}} * \beta * p(n) * \sigma_x^2(n) \quad \text{where } \gamma^0 = 1$$

$$\delta^n = \delta^{n-1} + \hat{\mathbf{x}} \cdot \overrightarrow{\mathbf{d_k}} * \beta * p(n) * \sigma_y^2(n) \quad \text{where } \delta^0 = 1$$

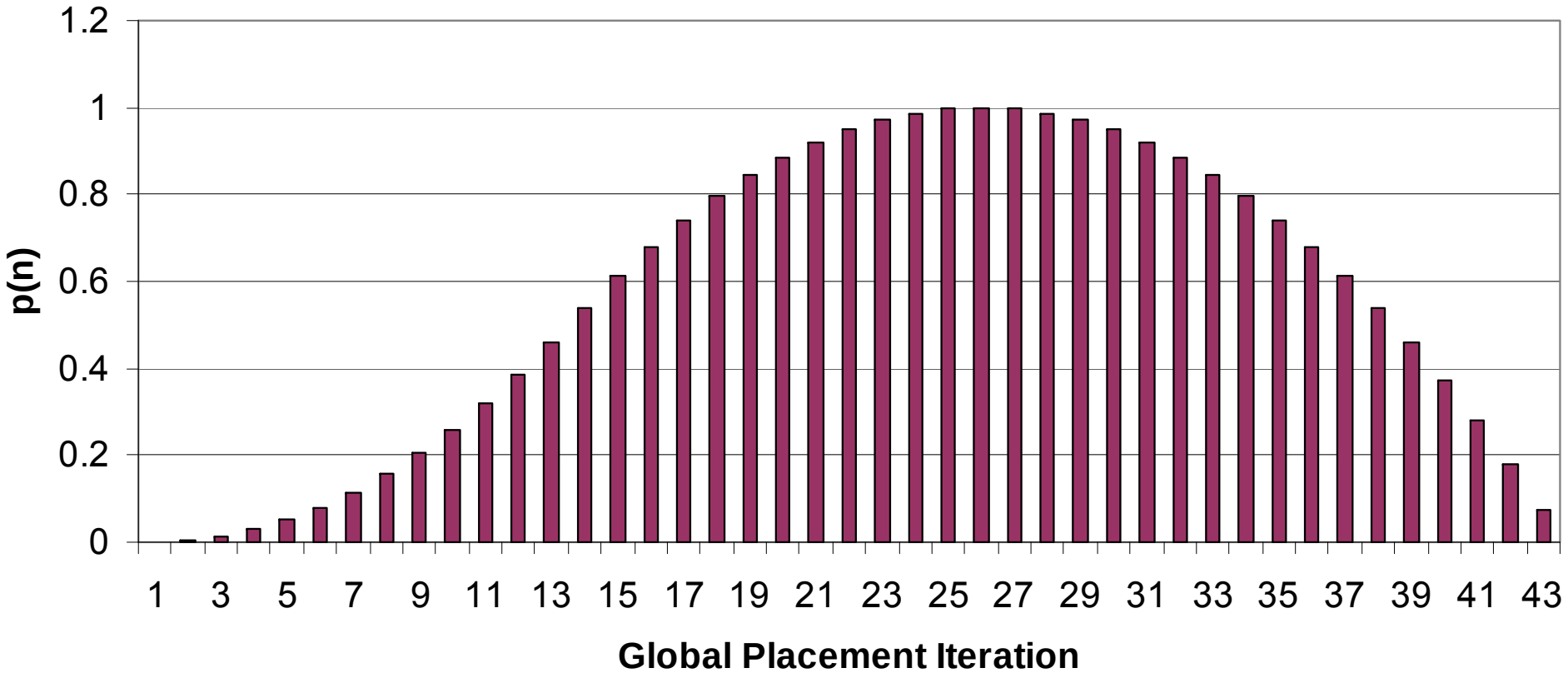
$$\Phi_G(\vec{x}, \vec{y})^n = \sum_{i,j} [(\gamma_{i,j}^n + w_{i,j})(x_i - x_j)^2 + (\delta_{i,j}^n + w_{i,j})(y_i - y_j)^2]$$

Step Size Scheduling



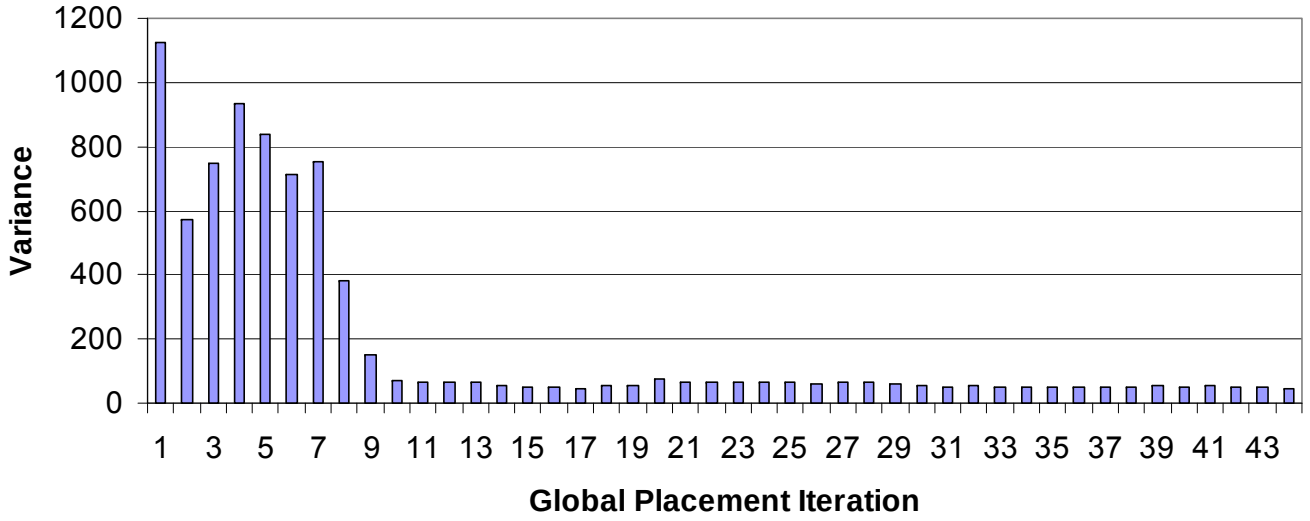
$$p(n) = \begin{cases} \frac{8n^2}{M^2} & 0 \le n < \frac{M}{4} \\ 1 - \frac{8(n-\frac{M}{2})^2}{M^2} & \frac{M}{4} \le n \le \frac{3M}{4} \\ \frac{8(n-M)^2}{M^2} & \frac{3M}{4} < n \le M \end{cases}$$

Weighting Step Function



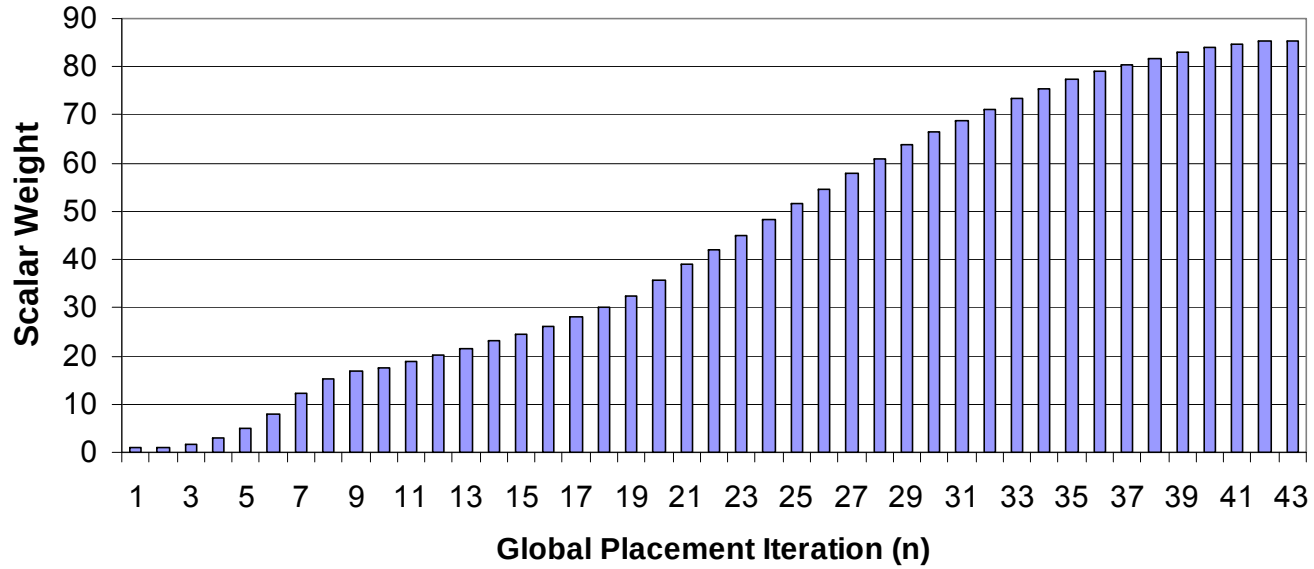
Skewed Weighting Results

$\sigma^2_x(n)$



- Low initial weight allows movement of the bit-stack
- Weigh tapers off near the end of global placement

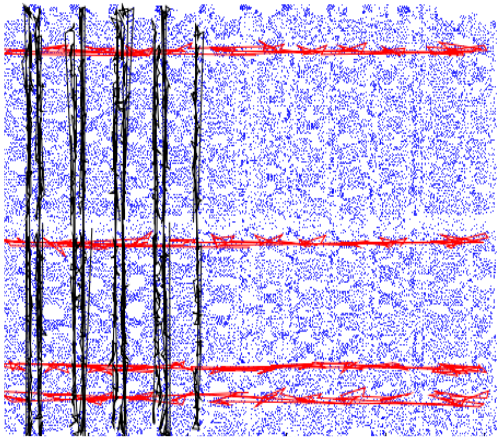
δ^n

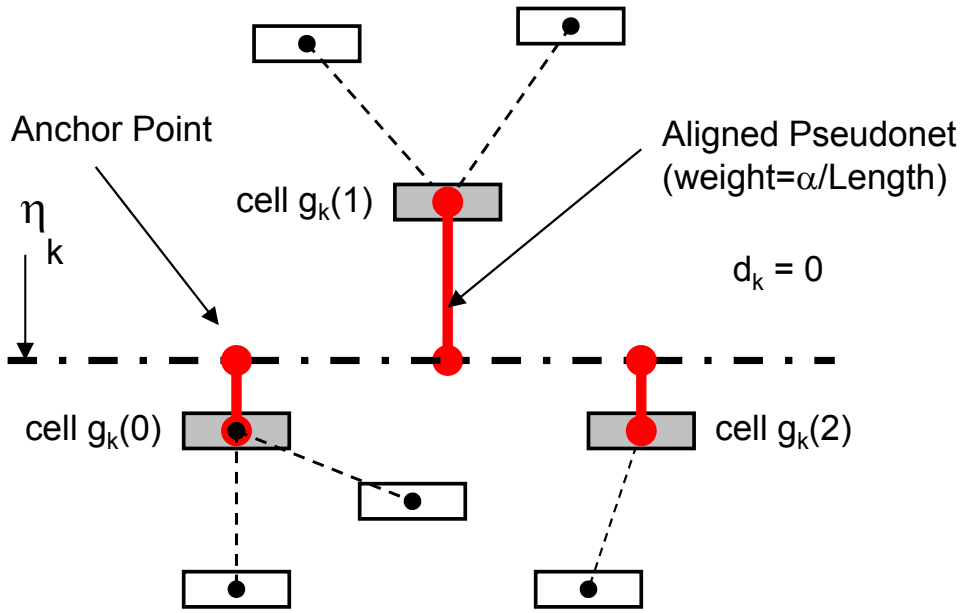


Fixed-Point Alignment Constraint

$$\eta^n_{k,i} = (x_i, \sqrt[|g_k|]{\prod_{j=1,\ldots,|g_k|} y_j}), \quad \text{if } d_k = 0$$

$$\eta^n_{k,i} = (\sqrt[|g_k|]{\prod_{j=1,\ldots,|g_k|} x_j}, y_i), \quad \text{if } d_k = 90$$

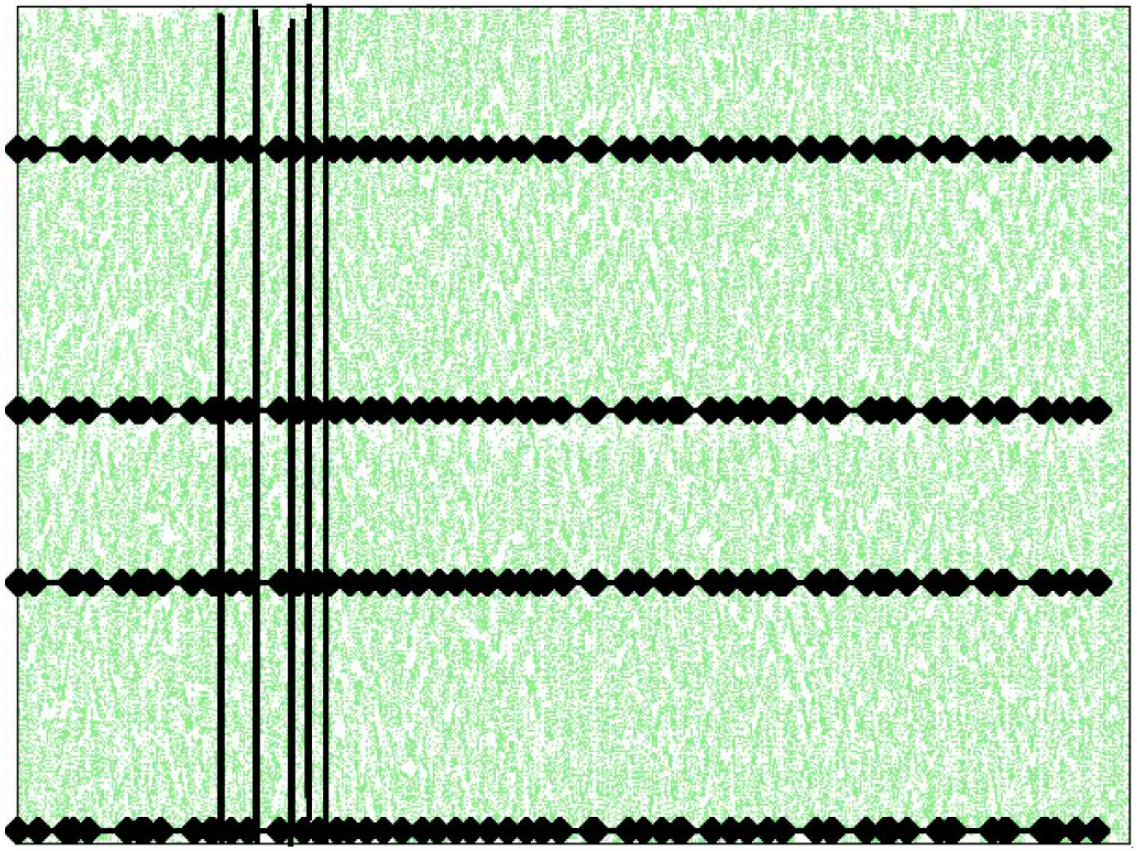




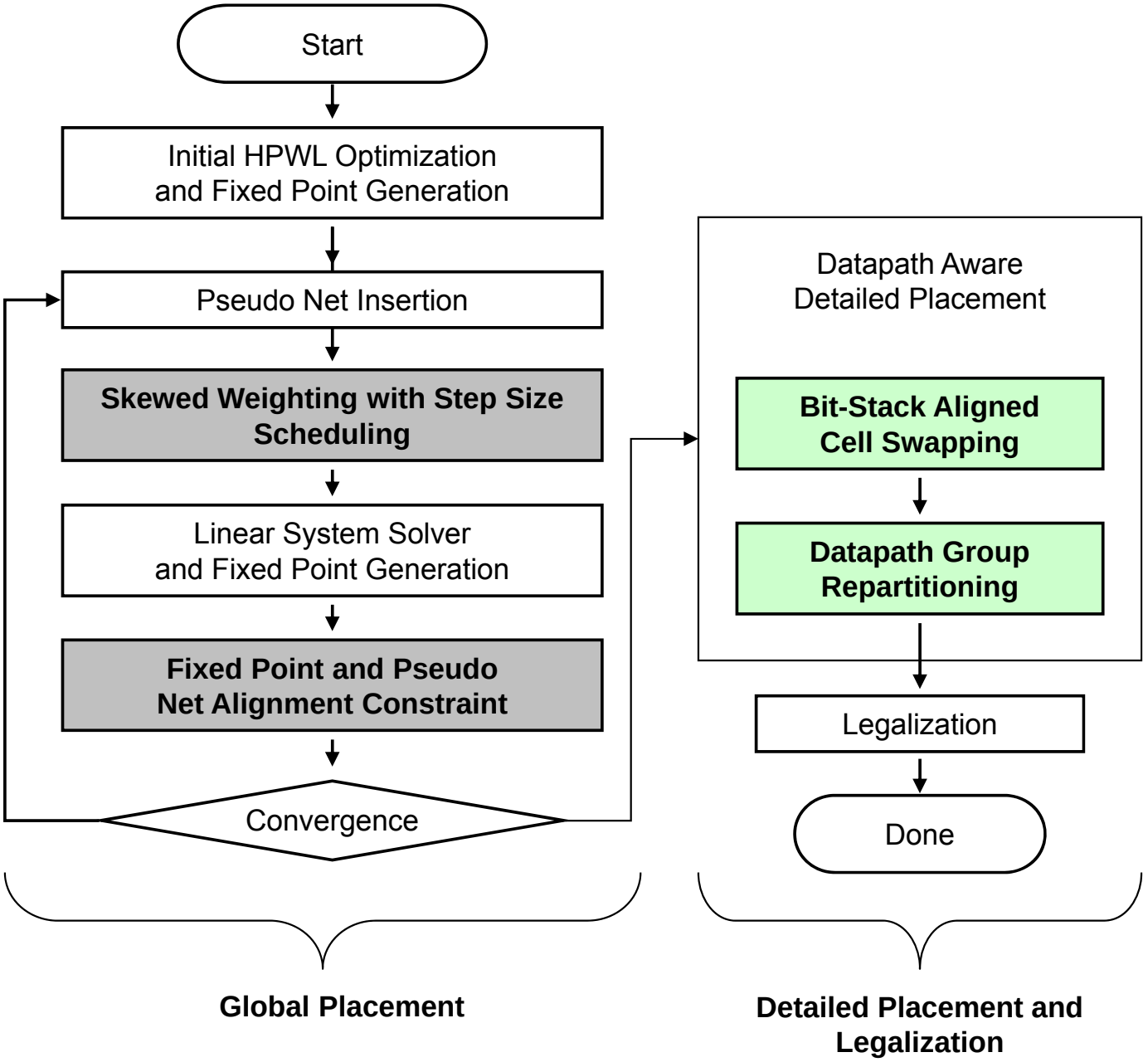
- Datapath cell shown in grey
- Directional weighting alone does not force alignment
- Modify fixed-point location for alignment nets
- During the next global placement iteration:
  - Cells are “pulled” into alignment my modifying fixed-point locations
  - Use the geometric mean to identify the position

Fixed-Point Alignment Results

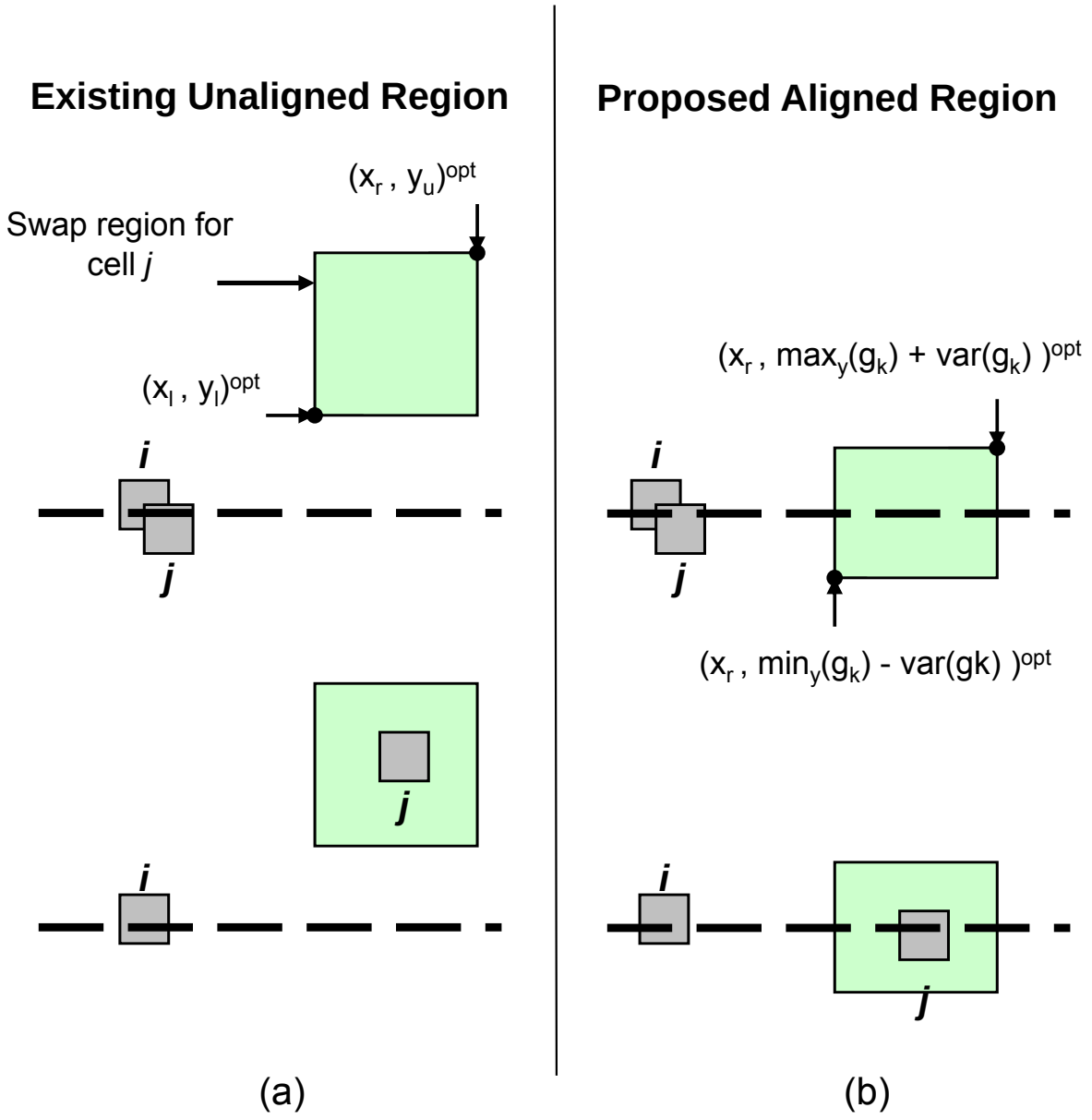
- **Fixed-Point alignment causes cells to be aligned almost perfectly**
- **Bit-stack cells are aligned horizontally**
- **Nets are aligned vertically**



Bit-Stack Aligned Cell Swapping



Bit-Stack Aligned Cell Swapping



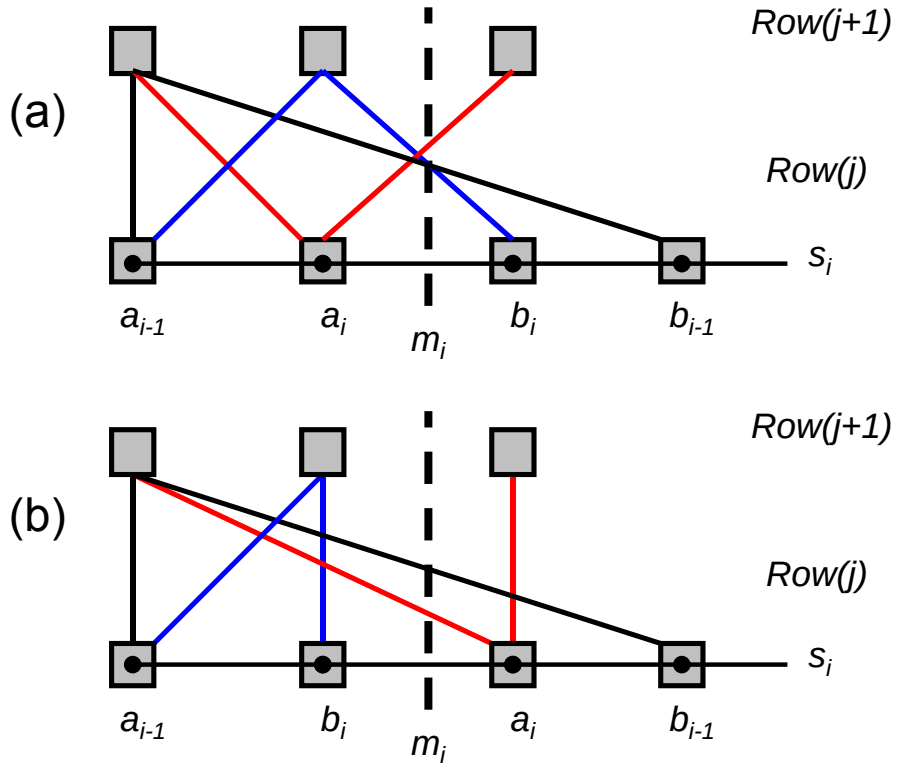
$$x_l^{opt}, min_y(g_k) - var_y(g_k)$$
$$x_r^{opt}, max_y(g_k) + var_y(g_k)$$
$$when \quad (\overrightarrow{\mathbf{d_k}} = 0)$$

$$min_x(g_k) - var_x(g_k), y_l^{opt},$$
$$max_x(g_k) + var_x(g_k), y_u^{opt}$$
$$when \quad (\overrightarrow{\mathbf{d_k}} = 90)$$

- **Maintain alignment during detailed placement (DP)**
- **Minimize wrong direction "global moves"**

Datapath Group Repartitioning

- **Use greedy moves to improve bitstack alignment**
- **Bipartition each alignment net**
- **Swap cells along the median if cut count improves**
- **Discard move if HPWL degrades**
- **Median point m_i is the median of the cells connected to the alignment net**



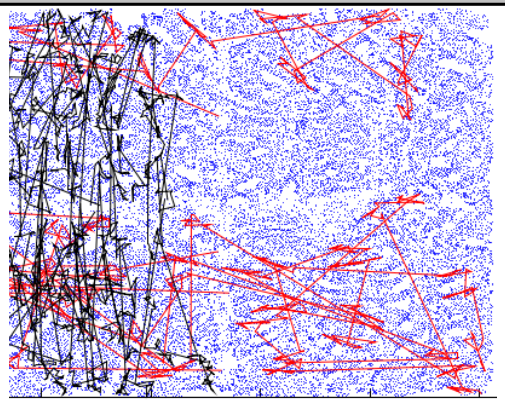
Outline

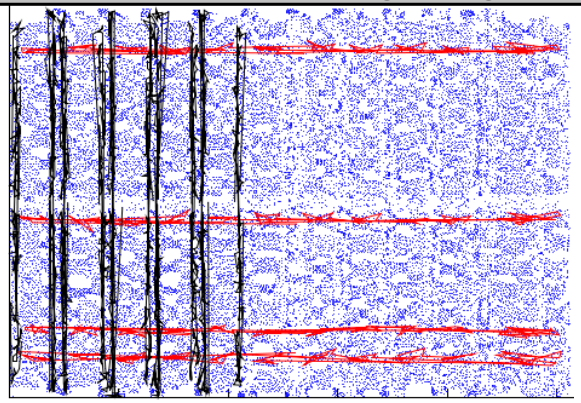
- General Placement Overview and Motivation
  - Why is the current formulation a problem?
- Key Contributions
- Global Placement Techniques
- Detailed Placement Techniques
- **Experimental Results**
- Future Work
  - Placement
  - Congestion

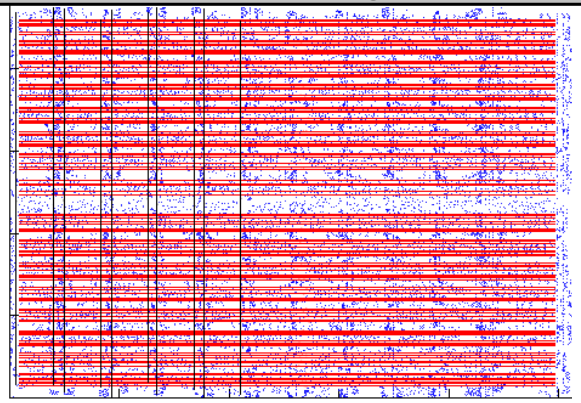
SAPT Experimental Results: GP

- **Plots of the vertical and horizontal alignments**
- **Base run shows the significant misalignment**
- **Skewed weighting allows for improved alignment: some jogging**
- **Fixed-point constraint forces almost exact alignment**

Base Run	Skewed Weighting	Fixed-Point Alignment

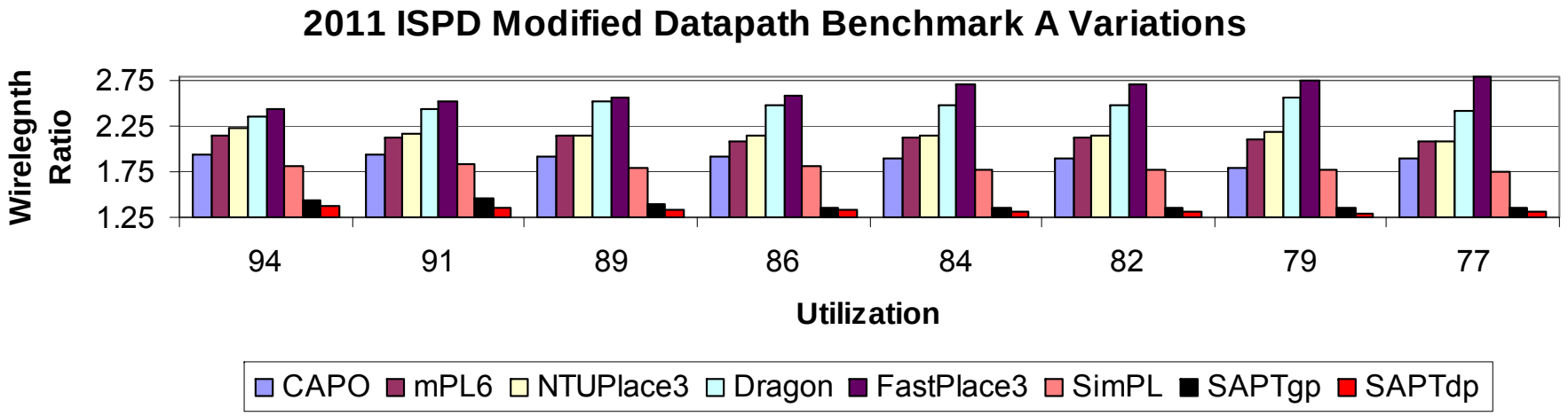


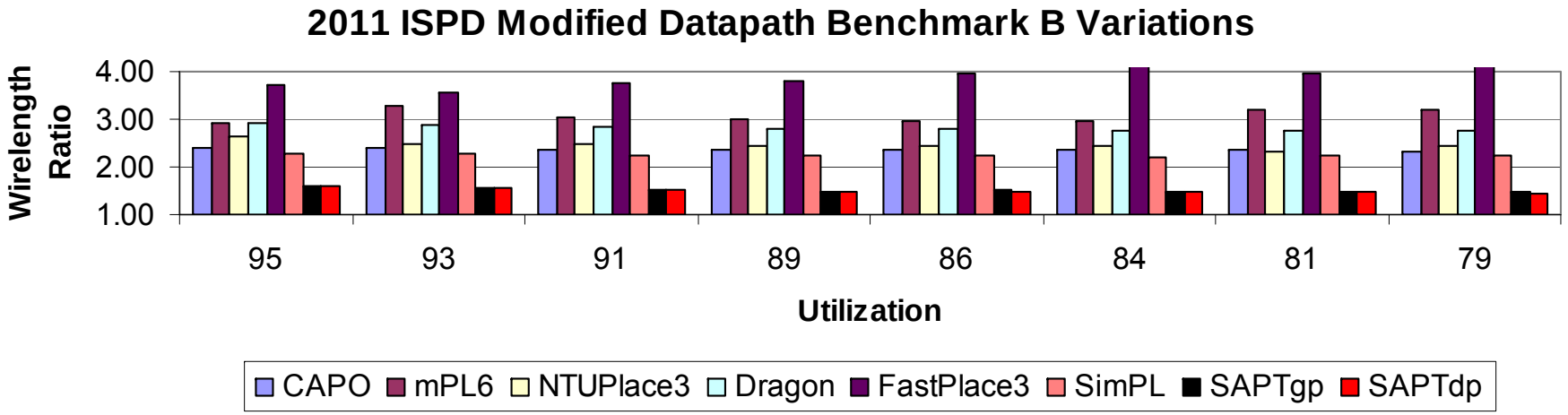




SAPT Experimental Results: Wirelength

- Total StWL ratio comparison on the modified ISPD 2011 Datapath Benchmark A and B variants
- Benchmarks are modified with unfixed latches
- All wirelength reported for legalized placement
- The ratios are computed with respect to the manually placed solution





SAPT Experimental Results: Hybrids

- **What is a hybrid?**
  - Some datapath
  - Lots of random logic
- **This is the future (really the present) design style**
- **Placers need to be able to handle both!**
- **Results highlight the HPWL fidelity issue**
- **Table shows:**
  - **Tatio of total wirelength (both random and datapath wirelength) compared to the wirelength of SAPTdp**
  - **Datapath percentage: < 1.2% for all designs**

HPWL	Hybrid C	Hybrid D	Hybrid E	Hybrid F	StWL	Hybrid C	Hybrid D	Hybrid E	Hybrid F
CAPO	1.13	1.17	1.12	1.19	CAPO	1.26	1.32	1.27	1.17
mPL6	1.05	1.02	1.20	1.37	mPL6	1.15	1.14	1.32	1.30
NTUPlace3	**0.95**	**0.95**	**0.99**	1.30	NTUPlace3	1.10	1.13	1.19	1.30
Dragon	1.10	2.11	1.32	1.29	Dragon	1.20	2.04	1.38	1.24
FastPlace3	**0.95**	0.96	1.22	1.17	FastPlace3	1.04	1.16	1.30	1.14
SimPL	1.02	0.97	1.03	1.04	SimPL	1.10	1.16	1.12	1.04
SAPTdp	1.00	1.00	1.00	**1.00**	SAPTdp	**1.00**	**1.00**	**1.00**	**1.00**

SAPT Experimental Results: Congestion

- **The Total Overflow on Datapath Benchmarks**
- **How do we measure congestion?**
  - Used the router and evaluation script from the ISPD 2011 routability-driven placement contest
  - Results after legalized placement
- **Results:**
  - Overflow reduced to zero on six of the benchmark A variants
  - Overflow reduced by at least 6.7x for all benchmark B variants

ISPD 2011 Datapath Benchmark A: Routing Overflow

	94	91	89	86	84	82	79	77
CAPO	2.29E+05	2.17E+05	1.72E+05	1.83E+05	1.84E+05	1.68E+05	1.10E+05	2.18E+05
mPL6	4.66E+05	4.38E+05	4.44E+05	3.40E+05	3.38E+05	3.65E+05	6.03E+05	5.02E+05
NTUPlace3	5.54E+05	5.12E+05	4.63E+05	5.19E+05	4.92E+05	5.63E+05	6.03E+05	5.02E+05
Dragon	-	-	-	-	-	-	-	-
FastPlace3	7.23E+05	8.10E+05	8.72E+05	9.08E+05	8.80E+05	1.04E+06	1.18E+06	1.21E+06
SimPL	1.28E+05	1.28E+05	1.22E+05	9.80E+03	8.70E+04	8.70E+04	8.50E+04	7.70E+04
SAPTgp	1.20E+02	3.20E+03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
SAPTdp	1.40E+02	3.80E+03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

ISPD 2011 Datapath Benchmark B: Routing Overflow

	95	93	91	89	86	84	81	79
CAPO	9.16E+05	7.28E+05	7.05E+05	6.68E+05	7.17E+05	7.01E+05	7.13E+05	6.98E+05
mPL6	1.27E+06	1.64E+06	1.40E+06	1.36E+06	1.28E+06	1.26E+06	1.53E+06	1.53E+06
NTUPlace3	1.02E+06	8.41E+05	8.30E+05	8.09E+05	8.92E+05	9.07E+05	8.21E+05	9.92E+05
Dragon	1.28E+06	1.27E+06	1.25E+06	1.24E+06	1.26E+06	1.27E+06	1.28E+06	1.29E+06
FastPlace3	2.08E+06	1.93E+06	2.16E+06	2.17E+06	2.37E+06	2.55E+06	2.35E+06	2.56E+06
SimPL	5.98E+05	6.24E+05	5.65E+05	5.49E+05	5.26E+05	4.85E+05	5.21E+05	5.25E+05
SAPTgp	9.00E+04	7.00E+04	5.60E+04	4.50E+04	4.80E+04	5.90E+04	6.20E+04	5.90E+04
SAPTdp	8.80E+04	7.00E+04	5.50E+04	4.30E+04	6.70E+04	5.80E+08	6.00E+04	5.80E+04

Future Work

- Upcoming work:
  - Will show method for the automatic datapath extraction of:
    - Datapath groups
    - Datapath direction
  - Will quantify:
    - Routing improvements on industrial designs
    - Timing improvements on industrial designs
    - Wirelength improvements across wider range of designs

Additional Slides

Total Horizontal Congestion for Benchmark A08

- Purple: > 100%