ECO-system: Embracing the Change in Placement

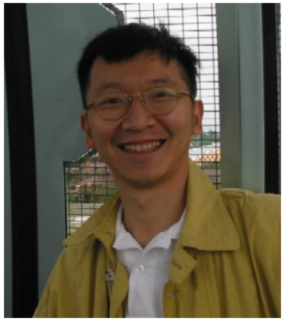
Jarrod A. Roy and Igor L. Markov

University of Michigan at Ann Arbor

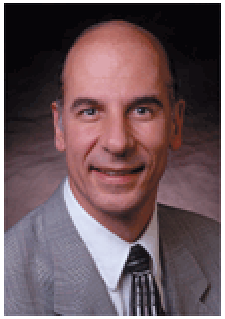
MichiganEngineering

Motivation

- Cong and Sarrafzadeh: state-of-the-art incremental placement techniques "unfocused and incomplete" (ISPD 2000)



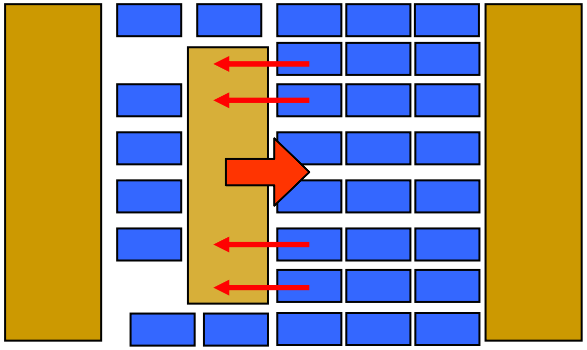
- Kahng and Mantik: CAD tools *"may not be correctly designed for ECO-dominated design processes"* (ICCAD 2000)



- Cadence CTO Ted Vucurevich: need *"re-entrant, heterogeneous, incremental, and hierarchical"* tools for next-generation designs (ISPD 2006 keynote)

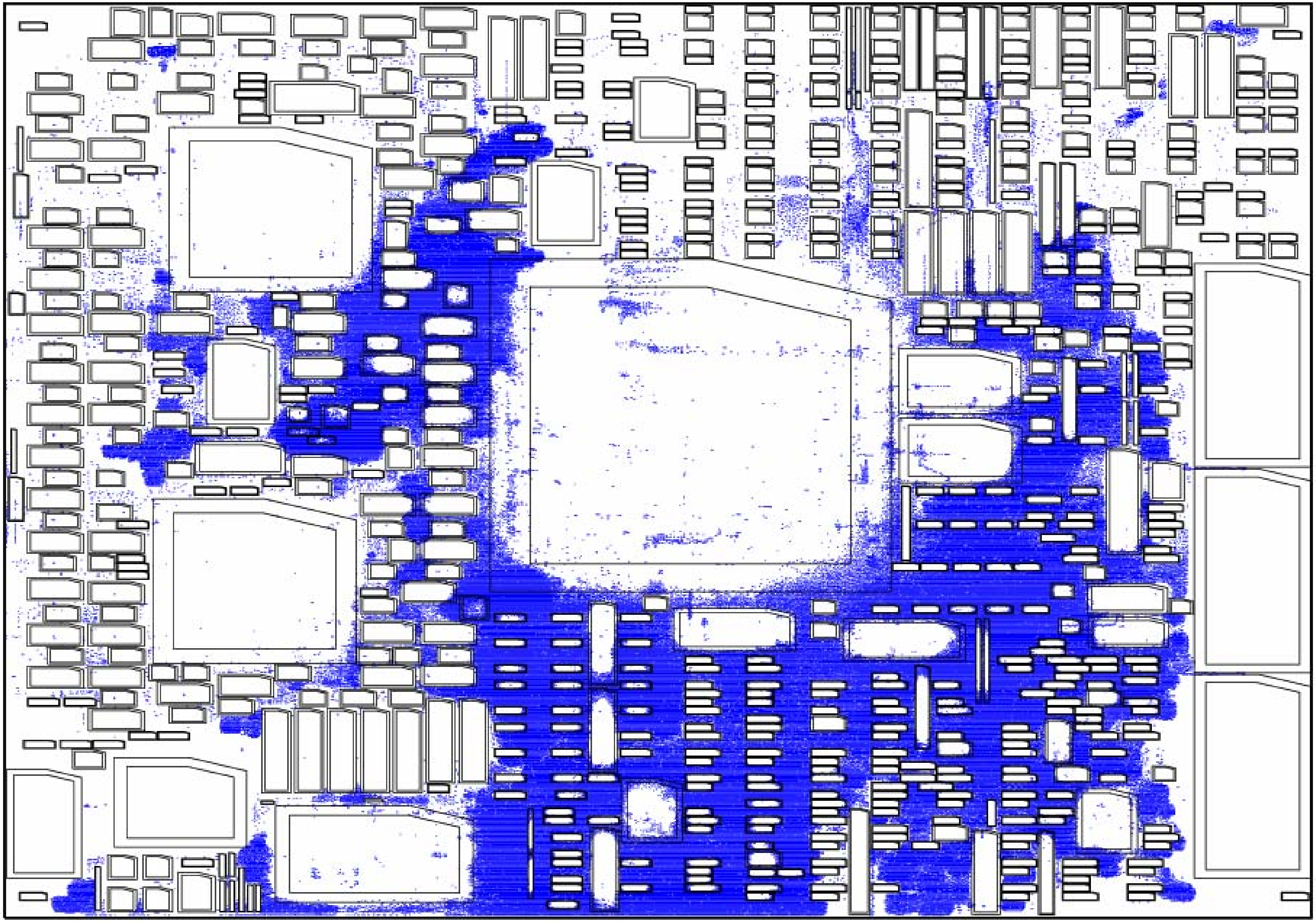
- Synplicity CTO Ken McElvain: "**Our focus in this flow is to produce similar output for small design changes**, …" (EE Times, Jan. 16, 2007)

Limitations of Prior Work



- Instead of preserving design metrics, seeks to preserve geometry
  - Seeks minimum movement (ISPD'05)
  - Seeks to preserve relative ordering (DAC'05, ICCAD'05)
  - **Cannot place new logic**
  - **Cannot accommodate dramatic changes**
- Assume generous whitespace
  - In dense regions, cell-reordering dominates cell-shifting
    - If not the legalizer, the detail placer will reorder cells
  - **Fixed obstacles** complicate cell-shifting
  - Handling of **chunky macros** becomes difficult
    - **Puzzle-solving cannot be performed with PDEs** or non-linear optimization

APlace 2.04 Global



Contexts for ECO Placement

- Connecting global and detail placement
  - Analytical placers produce significant overlap, cells do not align to site and row boundaries
- Physical Synthesis
  - Buffering, sizing and resynthesis require legalization
  - “Safe Delay Optimization for Physical Synthesis”, K.-H. Chang et al., in session 6C
- High-level Synthesis
  - Restructuring multipliers
  - Adding new IP blocks
- Functional bug-fixing and other modifications
  - “Fixing Design Errors with Counterexamples and Resynthesis”, K.-H. Chang et al., in session 9C

Requirements for ECO Placement

- Changing cell dimensions
- Updating net weights/criticalities
- Adding/Removing various constraints:
  - Density (to promote routability)
  - Regions (to address timing)
- Adding/Removing nets
- Adding cells or macros
  - With or w/o initial locations
- Adding/Moving obstacles
  - Memories, IP blocks, RTL macros, etc.

Illustration 1: Moving a Macro

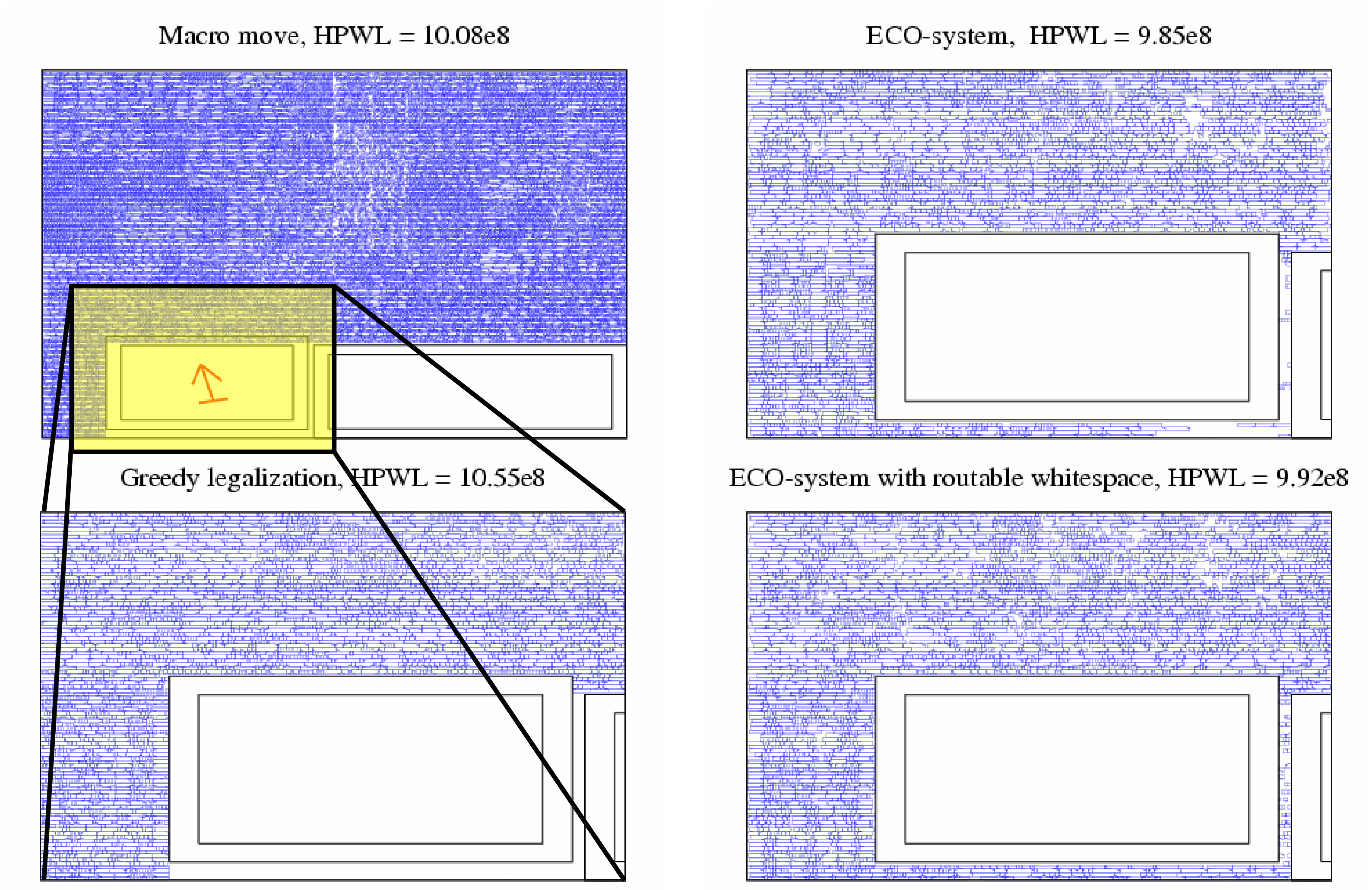
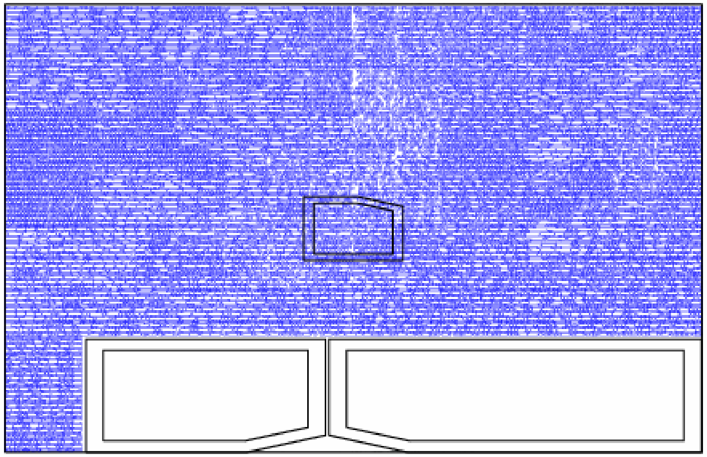
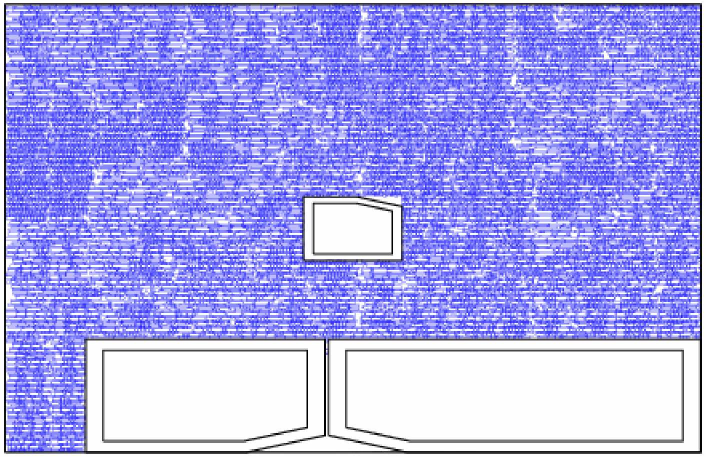
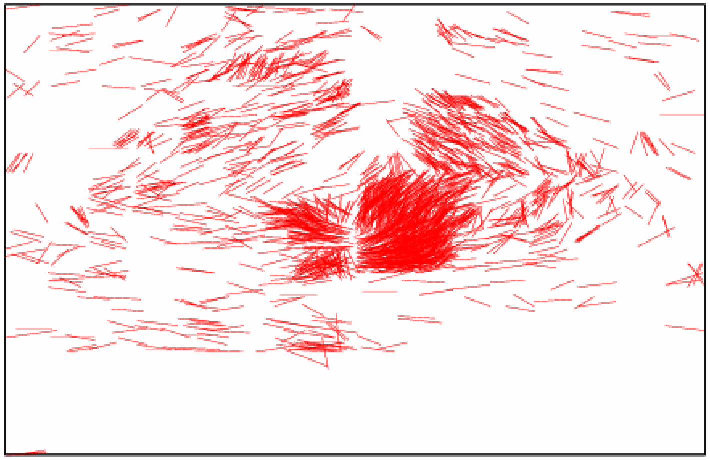

Macro move, HPWL = 10.08e8
ECO-system, HPWL = 9.85e8
Greedy legalization, HPWL = 10.55e8
ECO-system with routable whitespace, HPWL = 9.92e8

Illustration 2: Adding a New Macro


New macro at center, HPWL = 10.08e8

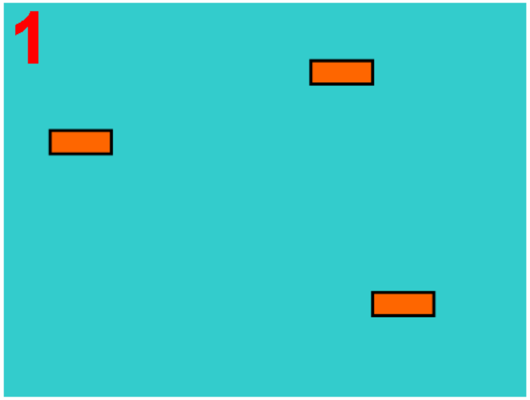

Post-processed by ECO-system, HPWL = 9.73e8


Cell Displacements >2.5% of Core Semi-perimeter

Our Solution: ECO-system

- Zooms in on regions that require change
- Applies adequate effort
  - Is capable of replacing whole regions
  - Can call a black-box global placer in regions
  - Can legalize even dramatic overlap
  - Can handle new logic modules, new obstacles
- Handles macros and fixed obstacles natively
- Includes all detail placement from Capo

ECO-system in Action



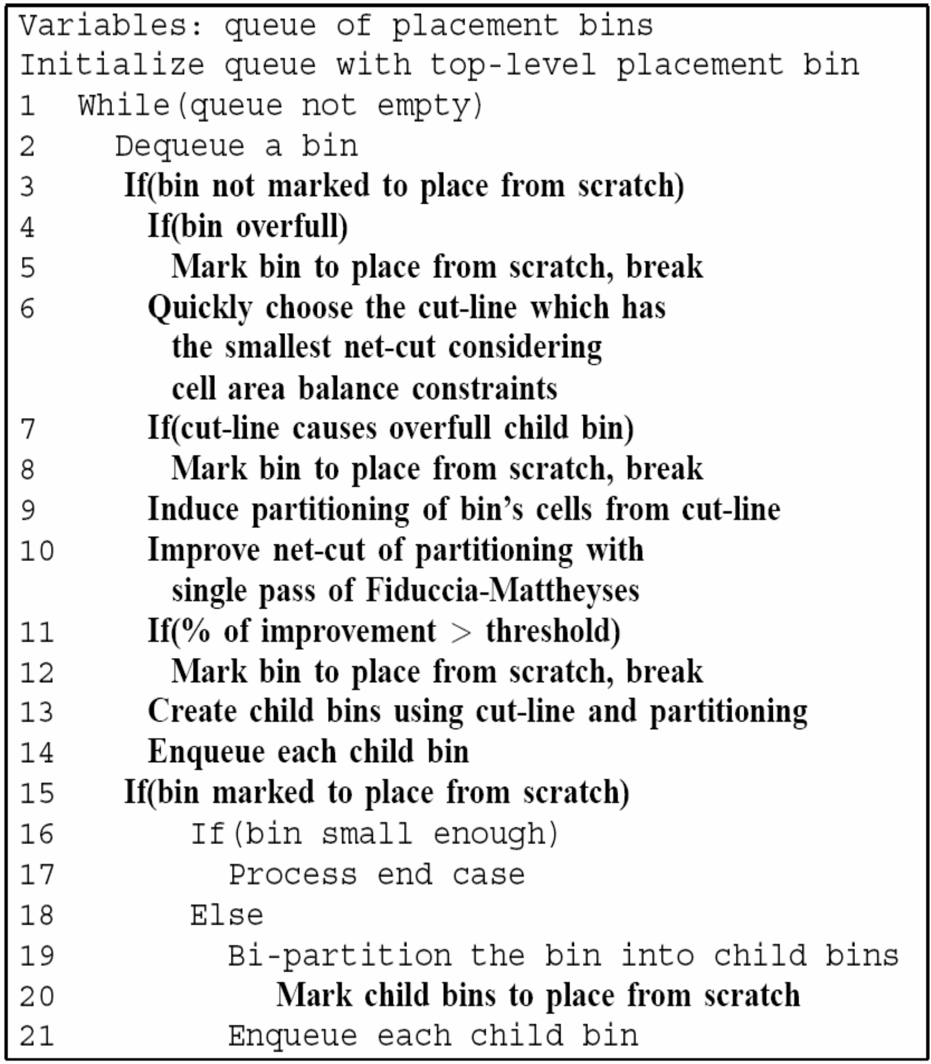
▭ = Original Placement   🟧 = Overlap

🟦 = Replaced from scratch   🟩 = Untouched by legalizer

ECO-system Flow

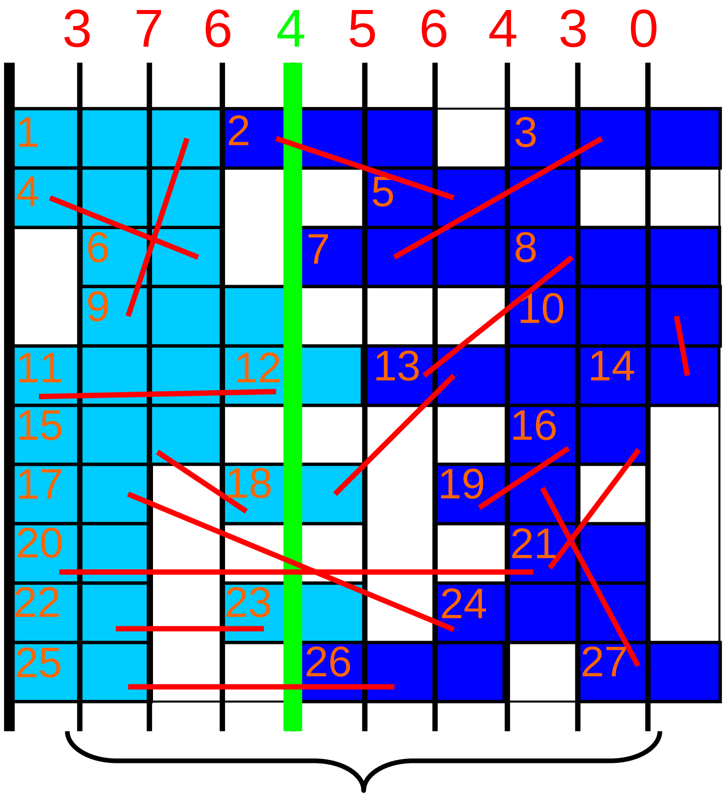
- Start with existing placmnt, proceed top-down
- Partition layout, not netlist
  - Unlike min-cut placers
- Fast geometric sweep
  - Minimize net-cut
  - Linear time, next slide
- Check quality of partitioning
  - Also linear time
- If cut-line bad or illegal, replace region from scratch

Details in proceedings



```
Variables: queue of placement bins
Initialize queue with top-level placement bin
While(queue not empty)
  Dequeue a bin
  If(bin not marked to place from scratch)
    If(bin overfull)
      Mark bin to place from scratch, break
    Quickly choose the cut-line which has
      the smallest net-cut considering
      cell area balance constraints
    If(cut-line causes overfull child bin)
      Mark bin to place from scratch, break
    Induce partitioning of bin's cells from cut-line
    Improve net-cut of partitioning with
      single pass of Fiduccia-Mattheyses
    If(% of improvement > threshold)
      Mark bin to place from scratch, break
    Create child bins using cut-line and partitioning
    Enqueue each child bin
  If(bin marked to place from scratch)
    If(bin small enough)
      Process end case
    Else
      Bi-partition the bin into child bins
      Mark child bins to place from scratch
      Enqueue each child bin
```

Linear-time Cut-line Selection



Potential Vertical Cut-lines

- Proceed left to right (or bottom to top)
- Maintain area and net-cut per cut-line
- Choose balanced cut-line with least cut
- Runs in linear time w.r.t. # of pins

Evaluating a Partition

- A geometric cut-line and a placement determine a netlist partition
  - E.g., for a min-cut placement this may be an original placement found by hgraph partitioner
  - We make no assumptions about the placement
- We can either accept or reject a partition
  - Accept: the above algorithm continues
  - Reject: region is replaced from scratch using any placer
- Rejection criterion
  - If (best found) partition is unbalanced, then reject
  - Run a single pass of Fidducia-Mattheyses (linear time)
  - If cut improvement >90%, then reject (tolerance represents aggressiveness)
  - If several additional checks pass, then accept

Interface with High-level and Physical Synthesis

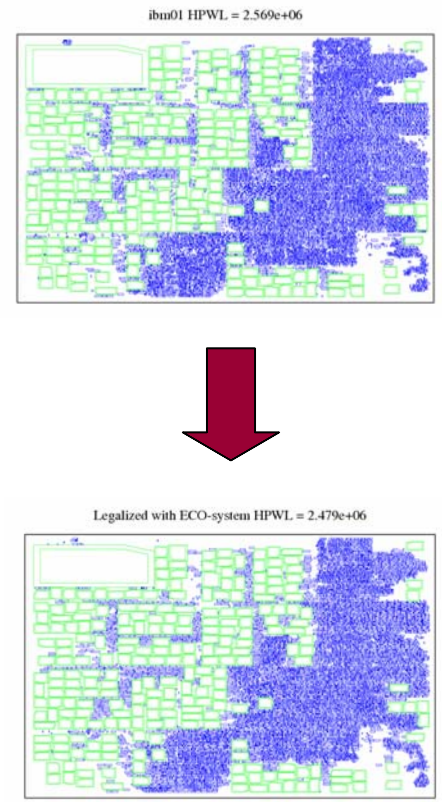
- Additional user controls
  - Specify areas for refinement
  - Tune **ECO-system**'s aggressiveness
  - Update net weights for TD placement
  - Redistribute whitespace
- Placing new cells and macros
  - With or without initial locations

Experimental Results

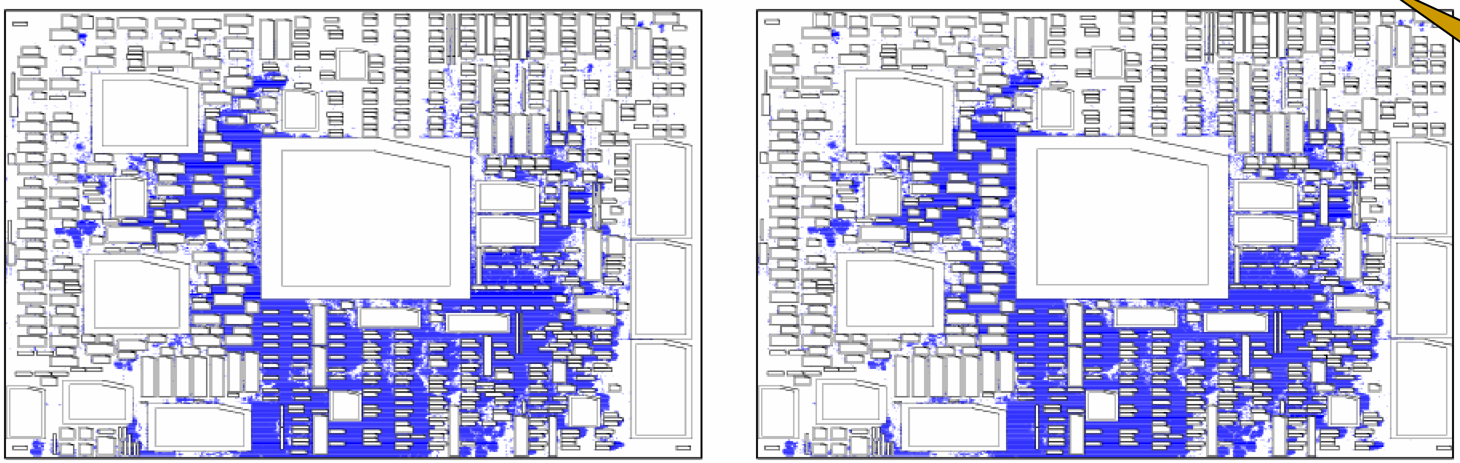
- **ECO-system** tested in several contexts
  - Cell resizing
  - Legalization of analytical global placements
  - Improving routability
- Tested on a wide range of publicly available benchmark suites
  - ISPD'02 IBMv2 benchmarks
  - ICCAD'04 IBM-MixedSizewPins benchmarks
  - ICCAD'04 Faraday benchmarks
  - ISPD'05 placement contest benchmarks
  - IWLS'05 OpenCores benchmarks

Cell Resizing Experiments

- Experiment 1
  - Start with Capo placements
    of IBM-MixedSizewPins benchmarks
  - Randomly resize each cell but maintain total area
  - Compare **ECO-system** with Capo 10 legalizer
- Experiment 2
  - Start with APlace placements of ISPD’05 benchmarks
  - Randomly resize each cell but maintain total area
  - Compare **ECO-system** with Capo 10 legalizer
- Experiment 3
  - Start with Capo placements of IWLS’05 benchmarks
  - Resize standard cells based on wire load
  - Upsize cells that drive longer wires
    according to fixed delay methodology
  - Compare **ECO-system** with Capo 10 legalizer



Resizing on ISPD'05

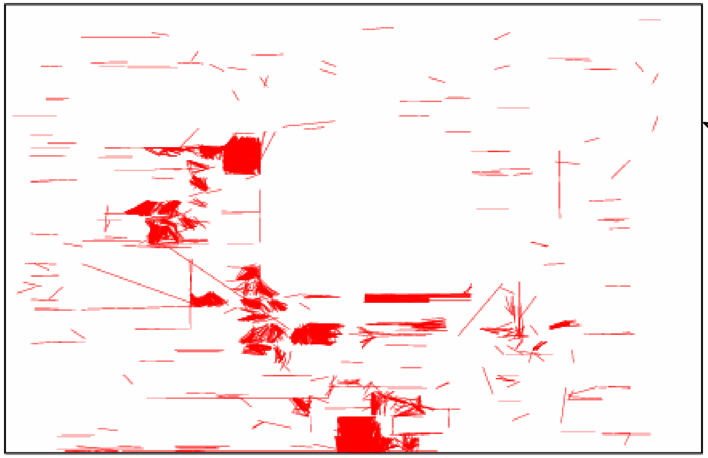


Before Resizing HPWL = 231.2e6

Postprocessed by ECO-system HPWL= 226.6e6

ECO-system *reduces* HPWL 2%

Placed by APlace 2.04



Cell Displacements >1.5% of Core Semi-perimeter (1.93% of all cells)

Average displacement 0.28% of core semi-perimeter, 10x <u>smaller</u> than previous work

Cell Resizing: Results

- Experiment 1: IBM-MSwPins benchmarks
  - Capo 10 legalizer takes 1% original place time, increases HPWL by 3.93%
  - **ECO-system** takes 16% original place time, increases HPWL by 0.61%
- Experiment 2: ISPD’05 benchmarks
  - Capo 10: 4% place time, increases HPWL 4.28%
  - **ECO-system**: 12% place time, decreases HPWL 1.00%
- Experiment 3: IWLS’05 benchmarks
  - Capo 10: negligible runtime, increases HPWL 1.85%
  - **ECO-system**: 6% place time, decreases HPWL 1.81%

ECO-system’s Impact on Timing

- Measure timing before and after **ECO-system** on resized IWLS’05 BMs
  - Timer uses D2M delay model with FLUTE Steiner trees
- **ECO-system** largely preserves timing
  - On average, critical path delay changes 1%
  - Worst case increases delay 8.07%
  - Best case decreases delay 7.37%
- Results not specific to our STA engine
  - In this experiment **ECO-system** is completely independent of timer
  - Average cell movement <1%, therefore **most design metrics should be largely unchanged**
- Using STA in **ECO-system** can further improve results

Experiments with Legalization of Analytical Global Placements

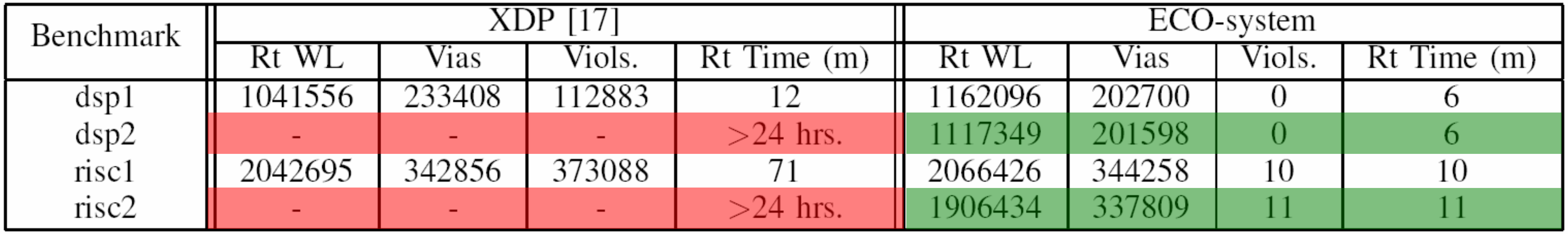
- Run APlace 2.04 on ISPD'05 benchmarks
  - Save global placements (overlap 28-47% by area)
  - Save final placements
- Legalize global placements using **ECO-system**
  - Compare the two sets of final placements
- Empirical results:
  - APlace legalizer increases HPWL 4.91%
  - **ECO-system** increases HPWL 3.68%, runs 3x faster than APlace legalizer

Routability Improvements

- Place IBMv2 benchmarks with mPL6
  - Save global placements
  - Save final placements
- Legalize global placements using **ECO-system**
- Route two sets of final placements with Cadence WarpRoute
  - Compare final routed designs
- Empirical results:
  - **ECO-system** placements route without violation
  - **ECO-system** *reduces* routed wirelength by 1.1%, vias by 7.8% and routing time by 50%

Routability with Fixed Obstacles

- Place Faraday (ICCAD'04) benchmarks with mPL6
  - Design "dma" omitted as it's obstacle-free
  - Save global placements
  - Save final placements
- Legalize global placements using **ECO-system**
- Route two sets of final placements with Cadence WarpRoute
  - Compare final routed designs
  - mPL6's detail placer is XDP (ASPDAC'06)



Benchmark	XDP [17]				ECO-system			
	Rt WL	Vias	Viols.	Rt Time (m)	Rt WL	Vias	Viols.	Rt Time (m)
dsp1	1041556	233408	112883	12	1162096	202700	0	6
dsp2	-	-	-	>24 hrs.	1117349	201598	0	6
risc1	2042695	342856	373088	71	2066426	344258	10	10
risc2	-	-	-	>24 hrs.	1906434	337809	11	11

Conclusions

- **ECO-system**: A robust and efficient placement recycler
  - Preserves the original placement but *has the power* to replace from scratch
  - Outperforms other incremental tools in runtime, HPWL and routability
  - *Minimal* impact on timing
- **ECO-system** provides reliable legalization with the ability to replace regions from scratch
  - Can resort to full-fledged placement
  - Lowers the barriers to research in global placement and physical synthesis
- **ECO-system** is included in Capo 10.5
  - Free for all uses
  - http://vlsicad.eecs.umich.edu/BK/PDtools/