Incremental Placement Algorithm for Field Programmable Gate Arrays

David Leong

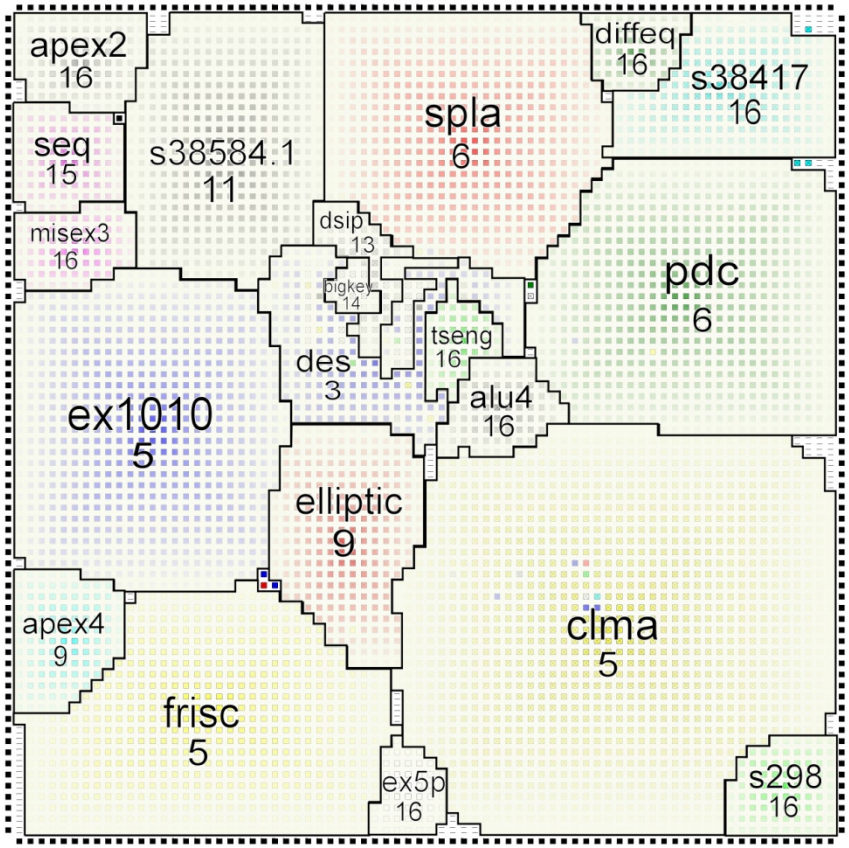
Advisor: Guy Lemieux
University of British Columbia
Department of Electrical and Computer Engineering
Vancouver, BC, Canada

Contributions

- Incremental Placement Algorithm
  - RePlace
  - Based on Placement Locality and Floor-planning
  - Multi-Region Algorithm
- Incremental Benchmark Circuits
  - Synthetic Benchmark Set
  - Physical Re-Synthesis Benchmark Set
- Findings
  - 50-260x speed up for Single-Region Benchmarks
  - 50-70x speed up for Multi-Region Benchmarks

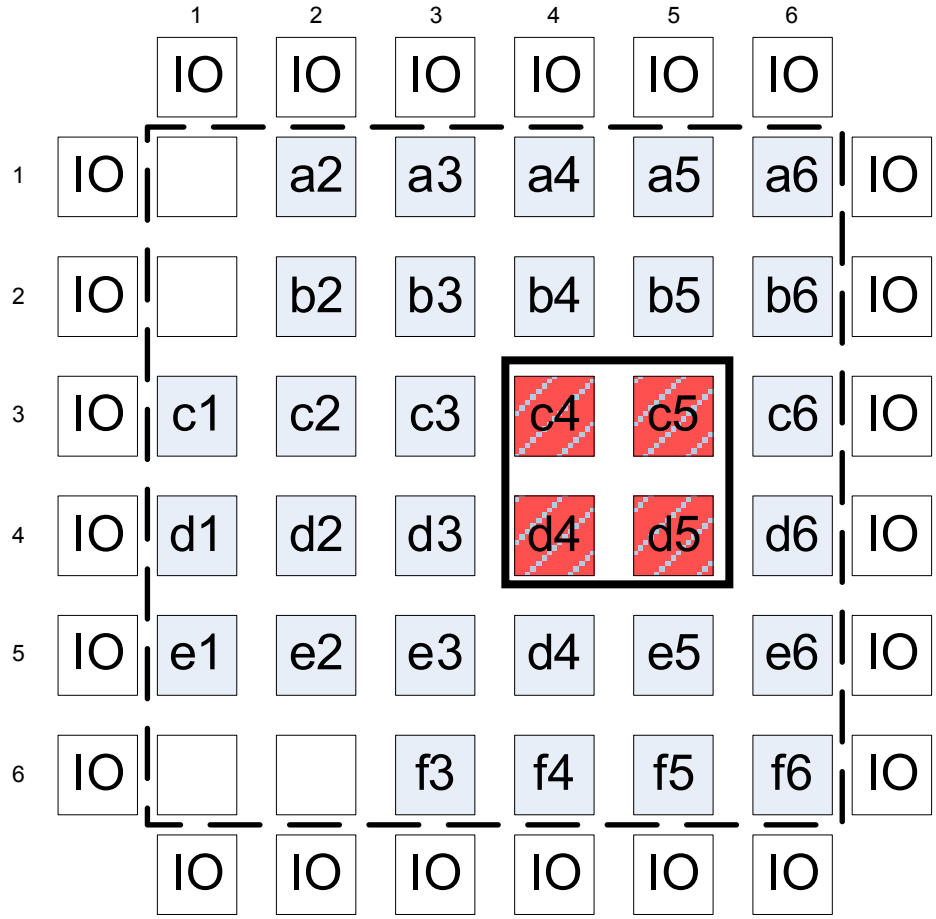
Motivation

- Runtime for placement is increasing with increasing FPGA sizes
  - 6 hours for 50,000 LUT circuit
  - Time == Engineering Cost
- System-on-Chip circuits are made of many subcomponents
- What if one part is modified?
  - Component reuse and hierarchy
  - Multiple regions of the circuit need incremental placement in order to support reuse
  - Floor-planning for each circuit
- Physical Re-Synthesis Flows



Incremental Placement Challenges

- Example:
  - C4,C5,D4,D5 is a modified sub-circuit
  - Can only fit <=4 CLBs in the previous location
  - Free space is far away!
  - How do we fit >4 CLBs quickly?



RePlace Formulation

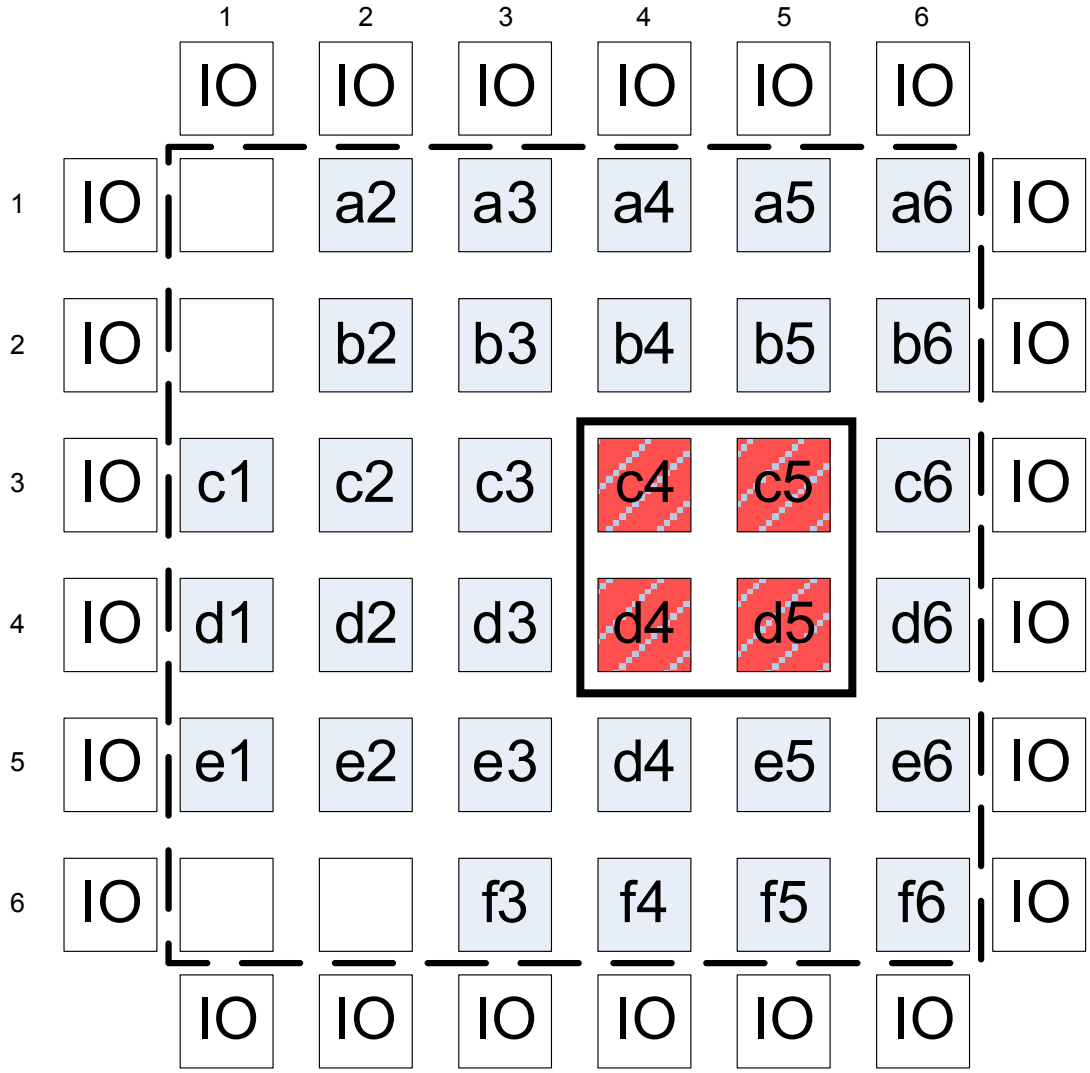
- Placement Locality
  - Modified Sub-Circuits should be “close” to previous location
  - Floorplans
- Expanded “Virtual” Placement Grid
  - Literally thinking outside of the box!
- Efficient Shifting Algorithm
  - No CPU Intensive LE/CLB swapping and cost evaluation

RePlace Algorithm

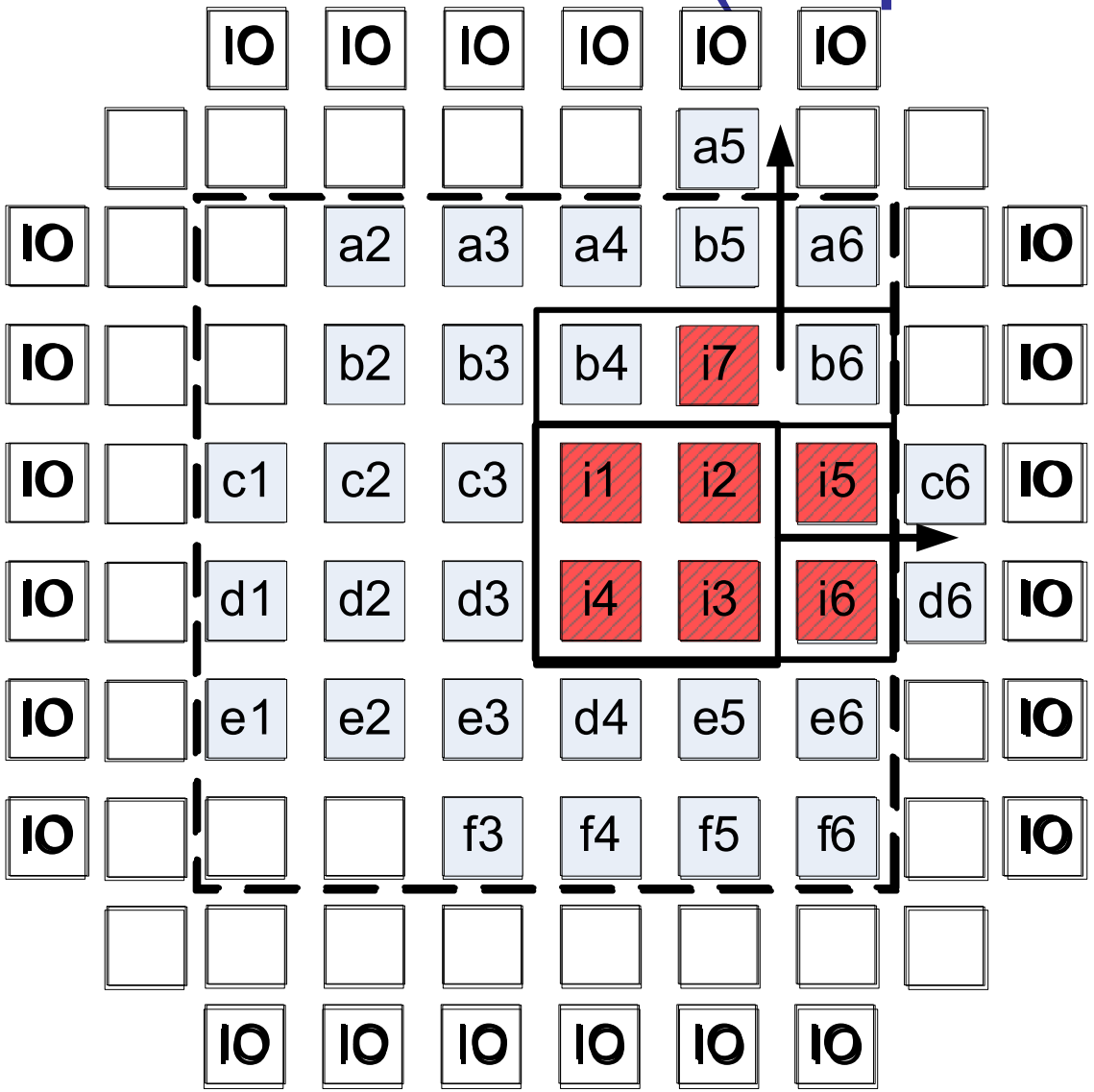
Four Steps to Incremental Placement

1. Previous Placement and Floor-planning
2. Expanded “SuperGrid” Placement
3. Compaction Re-legalization
4. Simulated Annealing Refinement

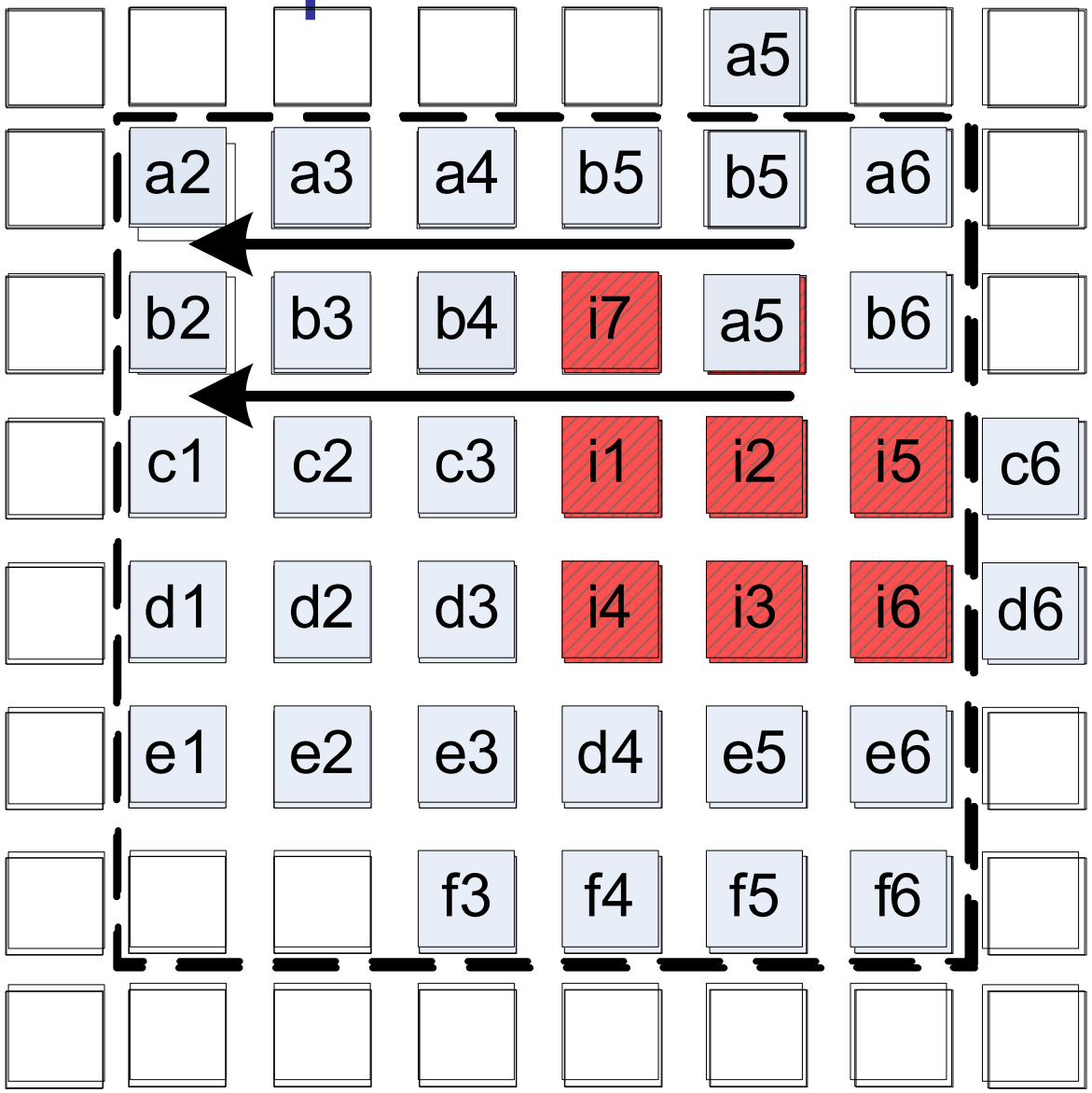
Previous Placement and Floor-planning



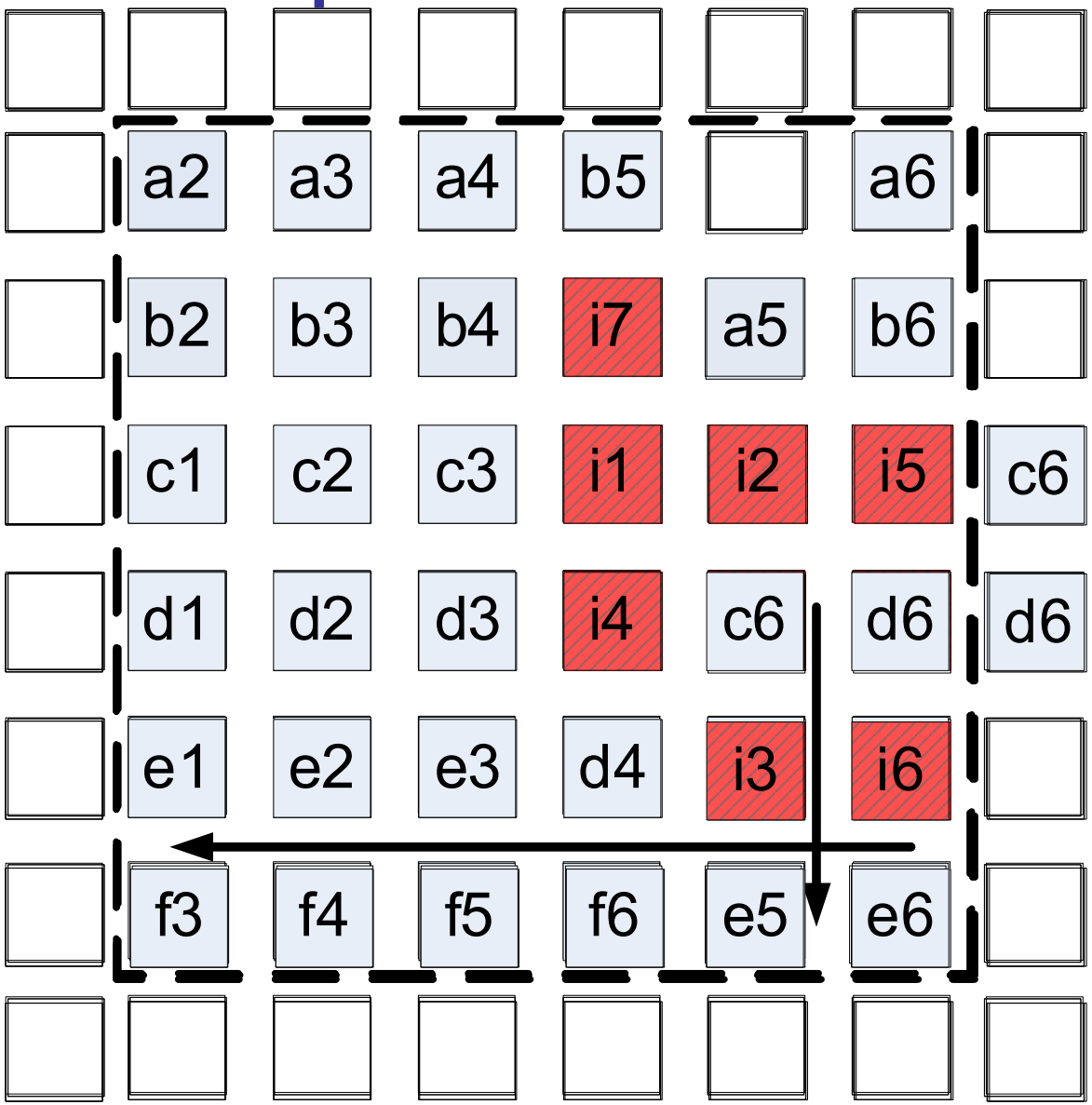
Expansion Phase (SuperGrid)



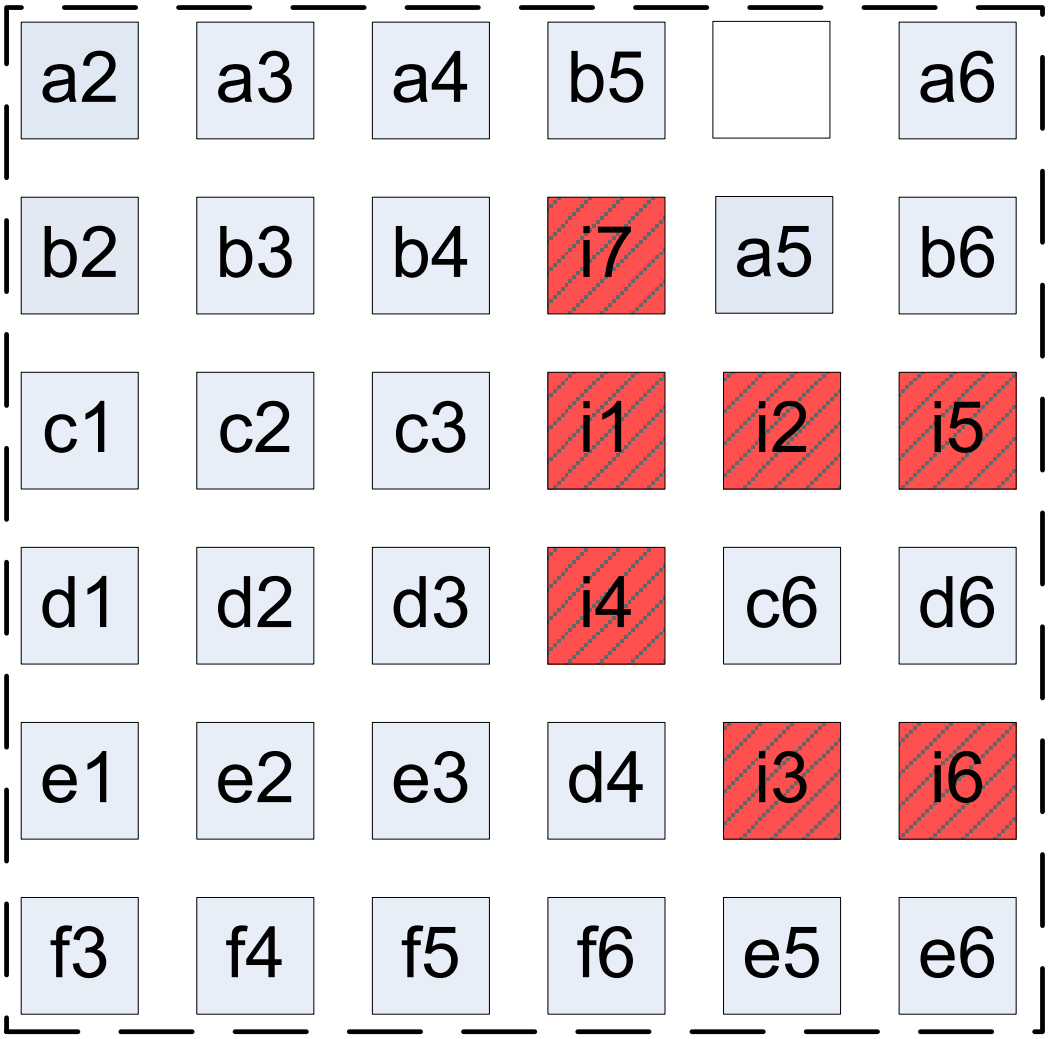
Compaction Phase



Compaction Phase



Intermediate Solution



Simulated Annealing Refinement

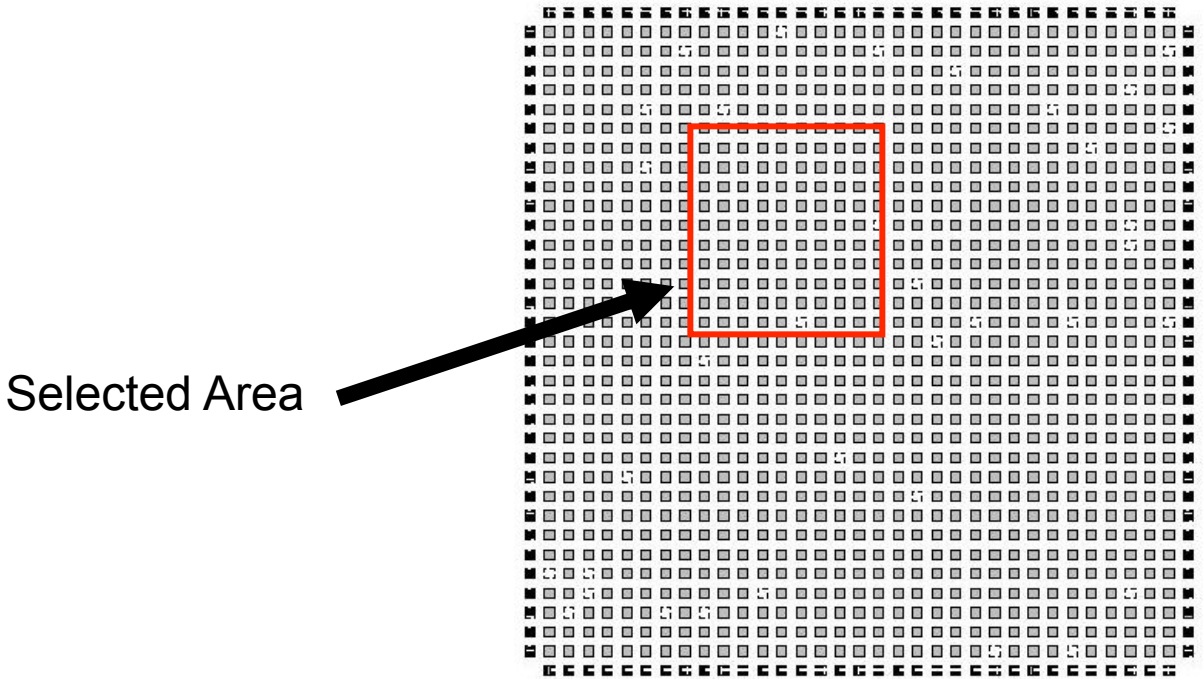
- Retuned VPR Simulated Annealing Algorithm
  - Lower Initial Temperature (44% temp)
  - Smaller Range Window
  - Lower Temperature degradation factor (Alpha)
  - Variable number of swaps per temperature range, 1-3x number of CLBs

Benchmarking

- Two Benchmark Sets
- Single Region:
  - Synthetic Flow
    - Simulates a design change
- Multi Region:
  - Physical Re-Synthesis flow
    - Circuit unchanged, clustering or tech-mapping changed for optimizations
    - Scaled to select multiple regions

Single Region Synthetic Benchmark Set (SR)

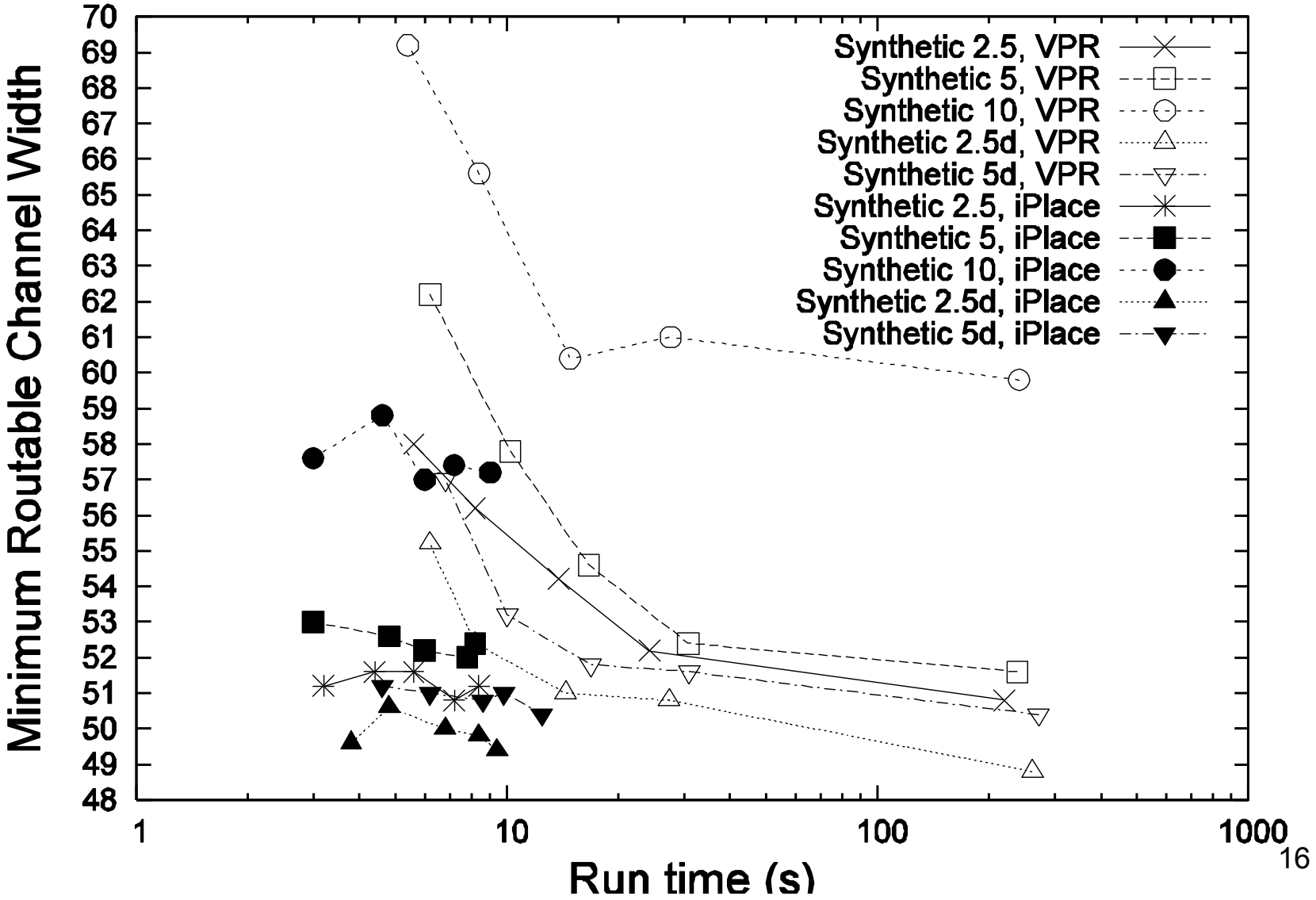
- Developed by Dave Grant
- Select a rectangular region on the placement grid
  - Replace it with a synthetic clone
  - Clone can be same size, smaller, or larger (double size)
- Selection region of 2.5%, 5%, 10% of array size



SR: Run-Time Results

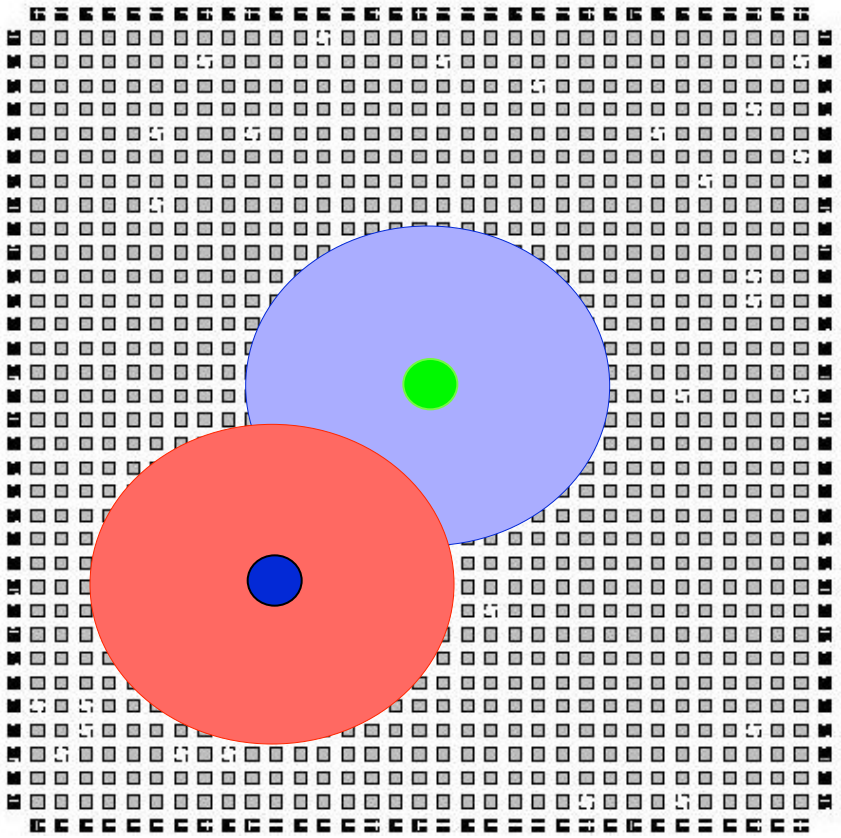
Synthetic Circuit	Syn - 2.5	Syn - 5	Syn - 10	Syn - 2.5d	Syn - 5d
CLMA	72.0	70.8	73.5	80.3	70.0
EX1010	77.6	75.0	77.0	69.0	76.2
MISEX3	-	-	-	-	-
PDC	80.6	64.0	68.7	84.4	68.1
SPLA	75.7	55.5	44.8	84.0	51.2
				Geometric Mean:	**70.1**

SR: Channel Width Results



Multi-Region Physical Resynthesis Benchmark (MR)

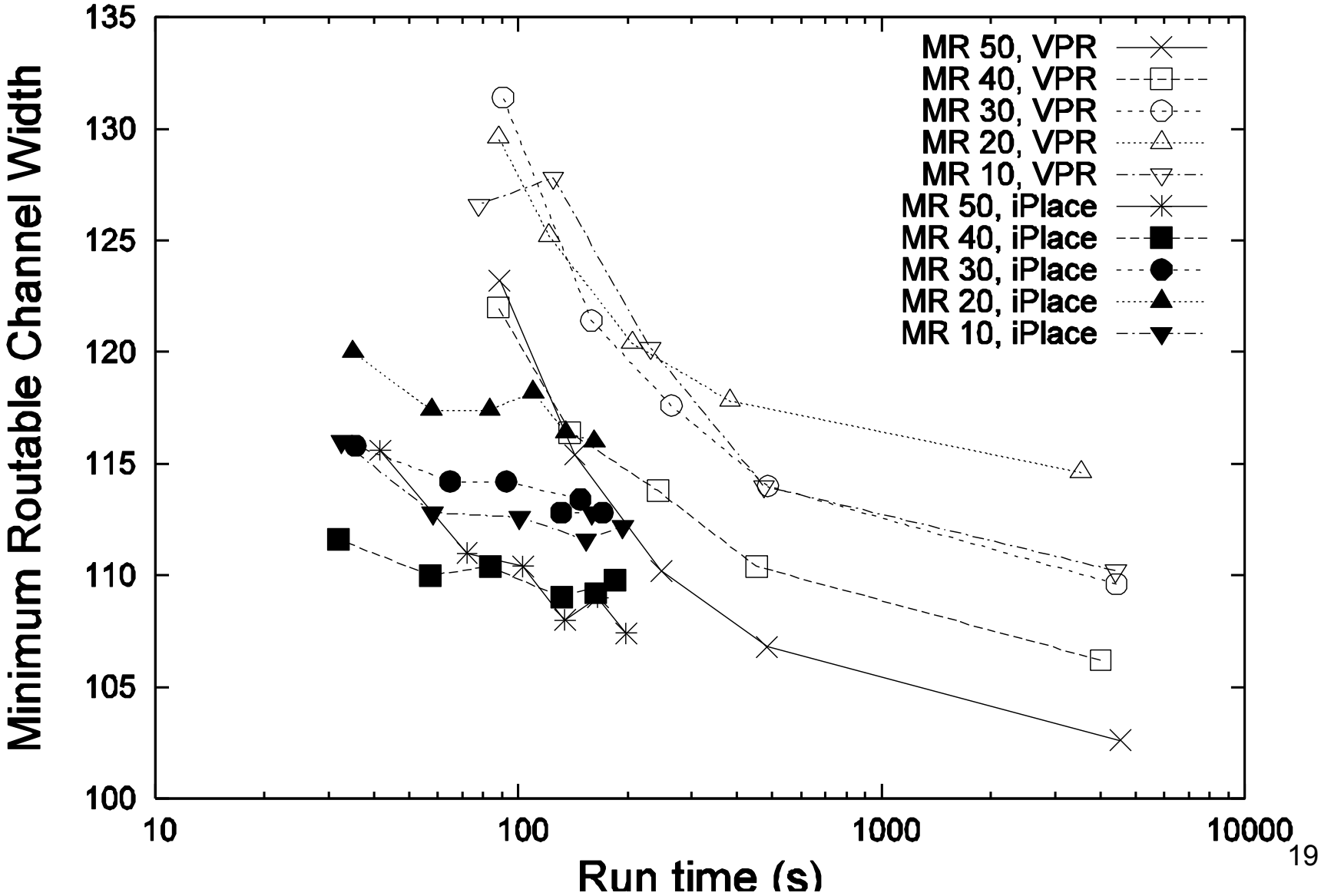
- Un/DoPack developed by Marvin, Guy and myself
- Iterative Congestion Reduction Algorithm
  - Select congested region
  - Spread out LUTs by adding whitespace to each CLB
- Multiple Regions
- Select all regions needed to reduce CW by 10%, 20%, 30%, 40%, 50%
- 1/3 to 2/3 of the entire FPGA re-synthesized



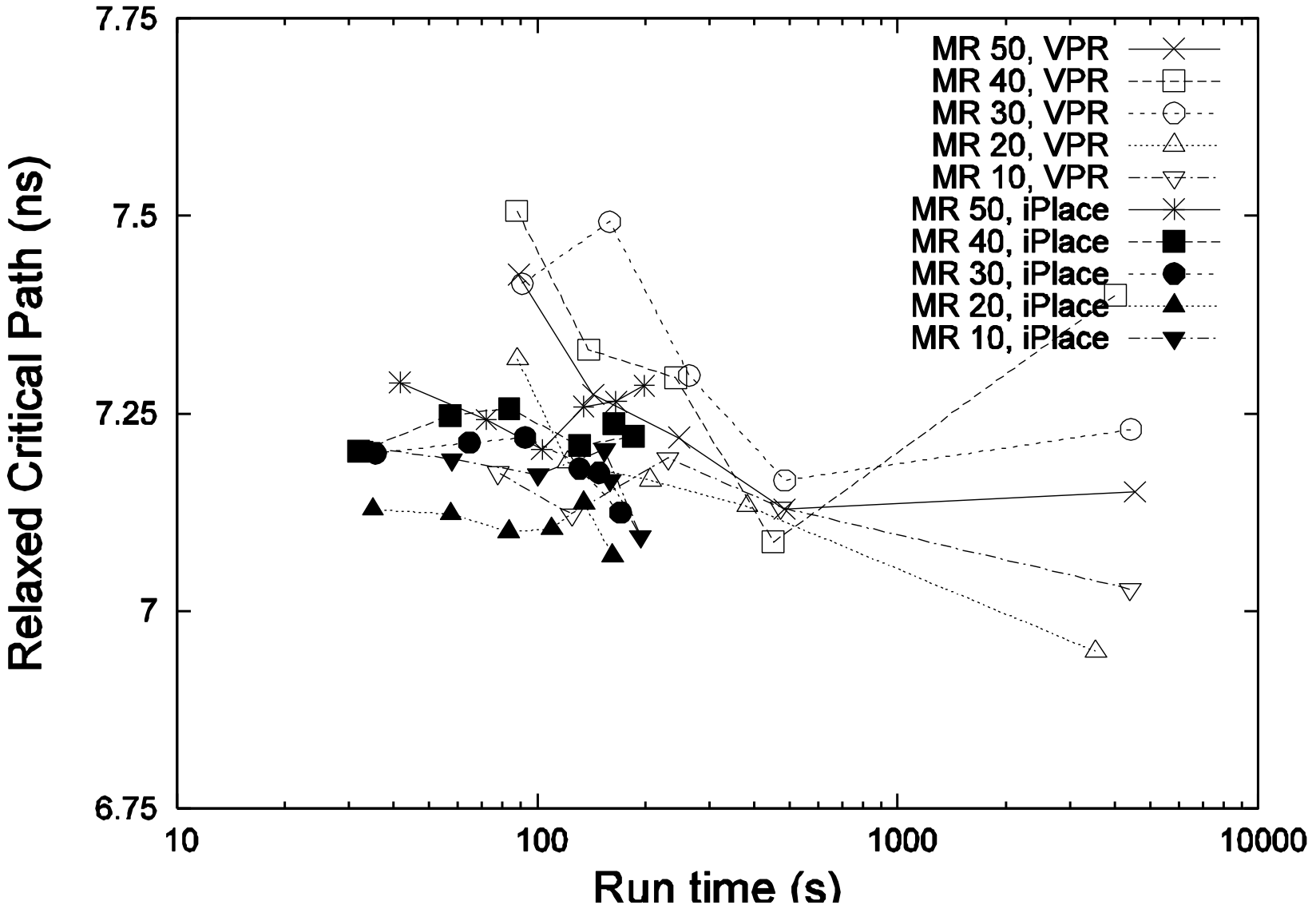
MR: Run-Time Results

Multi-Region Circuit	MR - 50	MR - 40	MR - 30	MR - 20	MR -10
CLONE	62.8	70.0	68.3	61.5	75.9
STDEV0	67.3	66.8	55.1	56.0	66.3
STDEV010	47.1	59.7	68.6	66.5	58.5
			Geometric Mean		**63.0**

MR: Channel Width Results



MR: Critical Path Results



Contributions

- Incremental Placement Algorithm
  - RePlace
  - Based on Placement Locality and Floor-planning
  - Multi-Region Algorithm
- Incremental Benchmark Circuits
  - Synthetic Benchmark Set
  - Physical Re-Synthesis Benchmark Set
- Findings
  - 50-260x speed up for Single-Region Benchmarks
  - 50-70x speed up for Multi-Region Benchmarks

Future Works

- Support for Macro Blocks
- Support for Carry Chains
- More Intelligent Shifting
  - Still keep it simple
- Integration with Commercial flows
  - EG: QUIP

End of Talk,

Thank you!!

Questions?