QUALCOMM®

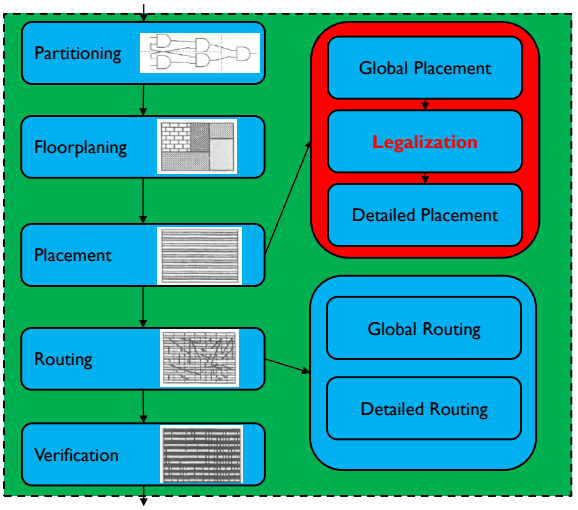
ABACUS2: A Fast, Minimum Displacement Standard-Cell Legalizer

Nikolaos Sketopoulos, Christos Sotiriou

Contributors: Xanthos Vlachos, Michalis Giaourtas

Contents

Floorplaning/Place and Route Flow



Analytical Placement Problem

- Placement Problem
  - **NP-Hard**
- Analytical Formulation
  - Convex Model
  - Treats Cells as Points
  - **Cells will Overlap!**
  - **Will not be Grid Aligned!**

Figure 4.1 – WL Solution Space

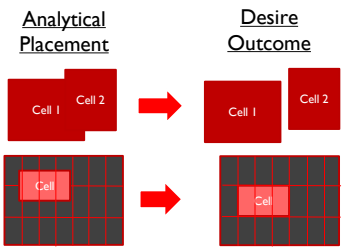


Figure 4.2 – Overlap and Alignment

- How to address the problem?

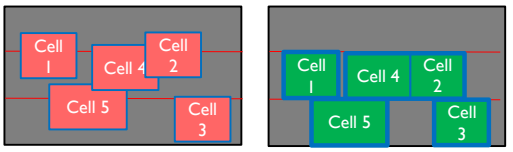


Figure 4. 3 – Global and Legal Placement

Floorplanning and Placement

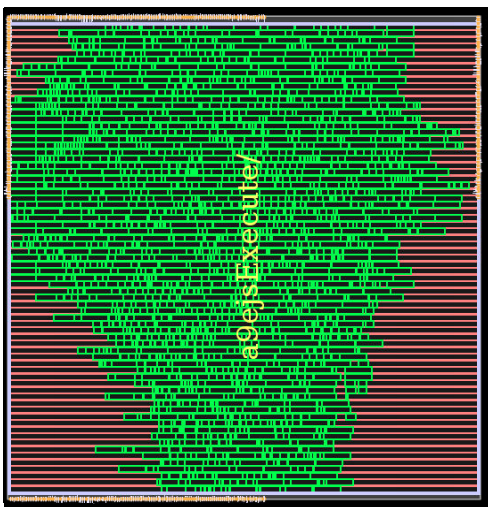
- Placement
  - **Global Placement (GP)**
    - Place Circuit Cells
  - GP Cost Function Minimization
    - Wire Length (WL)
    - Density
    - Slack, *etc.*
  - **Legalized Placement**
    - Minimum Displacement from GP Locations
  - Placement Algorithms
    - **Combinational**
    - **Analytical**
      - Globally Optimal Locations
      - Fast

Placer	Wirelength Model	Overlap Model	Integration	Optimization
APlace	LSE	Density	Penalty Method	Nonlinear
BonnPlace	Quadratic	Partitioning	Region Constraints	Quadratic
DPlace	Quadratic	Diffusion	Fixed Point	Quadratic
FastPlace	Quadratic	Cell Shifting	Fixed Point	Quadratic
FDP	Quadratic	Density	Fixed Point	Quadratic
Gordian	Quadratic	Partitioning	Region Constraint	Quadratic
hATP	Quadratic	Partitioning	Region Constraint	Quadratic
Kraftwerk2	Bound2Bound	Density	Fixed Point	Quadratic
mFAR	Quadratic	Density	Fixed Point	Quadratic
mPL6	LSE	Density	Penalty Method	Nonlinear
NTUPlace3	LSE/WA	Density	Penalty Method	Nonlinear
RQL	Quadratic	Cell Shifting	Fixed Point	Quadratic
SimPL	Bound2Bound	Partitioning	Region Constraint	Quadratic
Vasstu	LSE	Fixed Point	Fixed Point	Nonlinear
ePlace	WA	Density	Fixed Point	Nonlinear

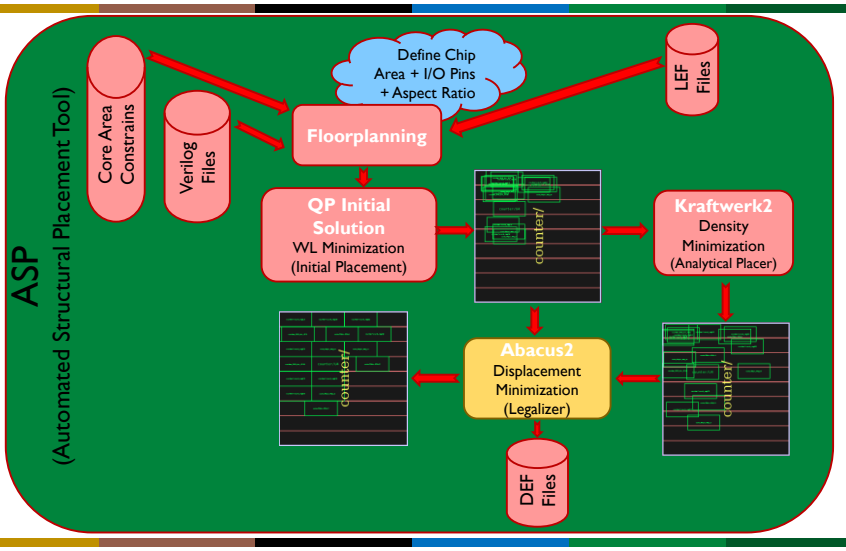
Figure 5.1 – Placers and their Characteristics

Analytical Placement Example

- 70% Utilization Floorplan
- Top level I/O Pin Placement
- QP (Quadratic Placement) produces Minimum WL Solution
  - **Significant amount of Cell Overlaps**
- Density Minimization
  - Spread Cells ➔ reduce overlaps
- Legalization Process
  - Reducing Overlaps induces significant WL overhead



Our Placement Flow

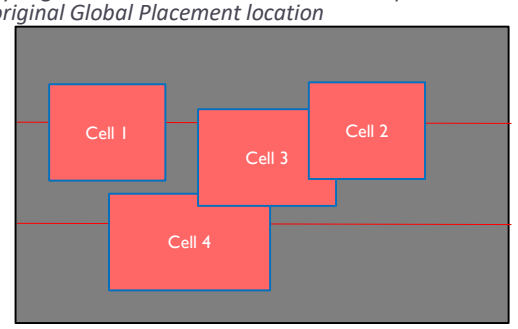


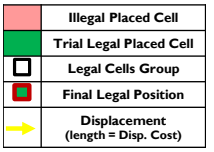
Contents

- Introduction
- **Abacus Algorithm Basics**
- Abacus2
- Results
- Conclusion and Future Work

Abacus Algorithm Example

- Legalizes on cell by cell basis (sequential)
- **Goal (per cell)**
  - Identify Legal location so as to *minimize cell displacement from original Global Placement location*



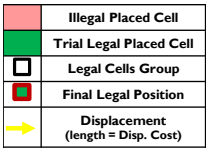


- Sort cells according to x-coordinate:

  Increasing Order: {Cell 1, Cell 4, Cell 3, Cell 2}

Spindler, Peter, Ulf Schlichtmann, and Frank M. Johannes. "Abacus: fast legalization of standard cell circuits with minimal movement." *Proceedings of the 2008 international symposium on Physical design*. ACM, 2008.

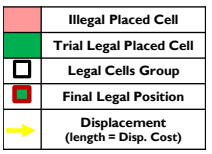
Abacus Algorithm Example (cont.)



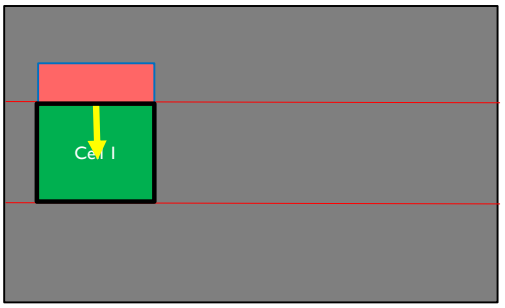
- Placement of Cell 1 in first row

- Cell 1 displacement = 1.52

Abacus Algorithm Example (cont.)

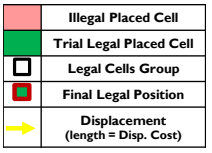


- Placement of Cell 1 in second row

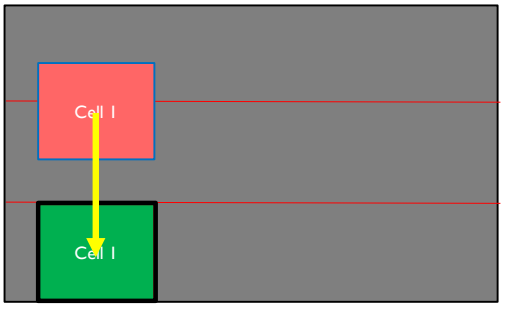


- Cell 1 displacement = 1.46

Abacus Algorithm Example (cont.)



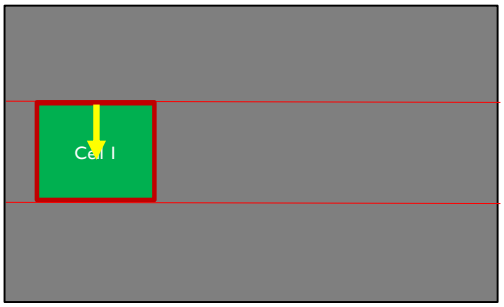
- Placement of Cell 1 in third row



- Cell 1 displacement = 3.82

Abacus Algorithm Example (cont.)

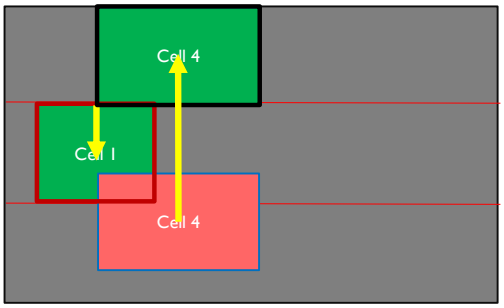
- Cell 1 is placed in second row, the row with the least displacement from original global placement location



Illegal Placed Cell
Trial Legal Placed Cell
Legal Cells Group
Final Legal Position
Displacement (length = Disp. Cost)

Abacus Algorithm Example (cont.)

- Placement of Cell 4 in first row

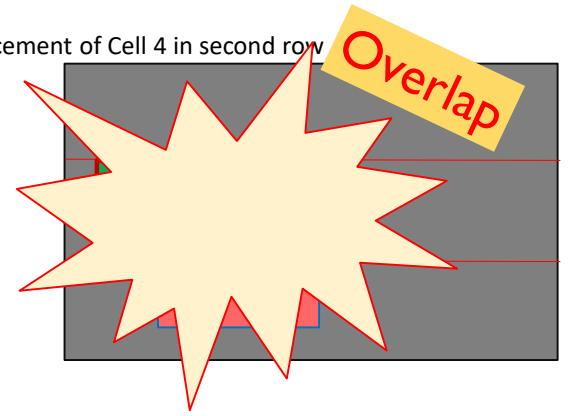


Illegal Placed Cell
Trial Legal Placed Cell
Legal Cells Group
Final Legal Position
Displacement (length = Disp. Cost)

- Cell 4 displacement = 4.44

Abacus Algorithm Example (cont.)

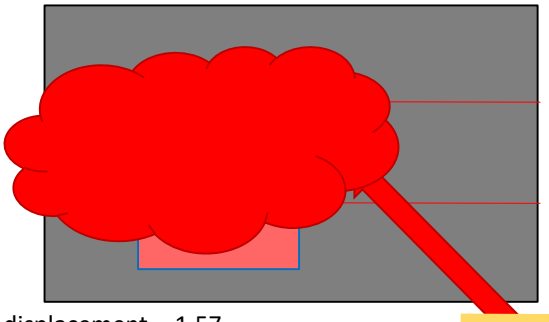
- Placement of Cell 4 in second row



Illegal Placed Cell
Trial Legal Placed Cell
Legal Cells Group
Final Legal Position
Displacement (length = Disp. Cost)

Abacus Algorithm Example (cont.)

- Placement of Cell 4 in second row



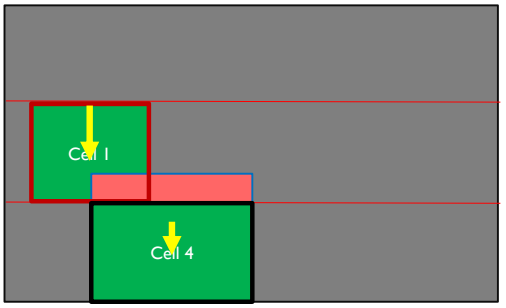
Illegal Placed Cell
Trial Legal Placed Cell
Legal Cells Group
Final Legal Position
Displacement (length = Disp. Cost)

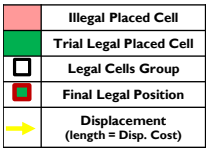
- Cell 4 displacement = 1.57

Cell Group

Abacus Algorithm Example (cont.)

- Placement of Cell 4 the third row

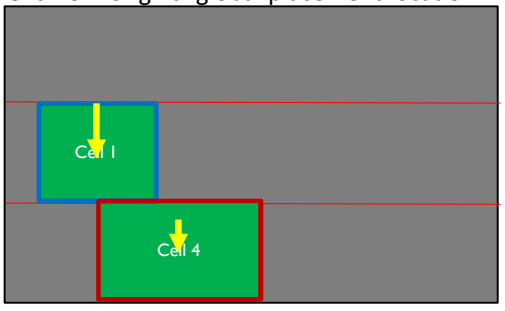


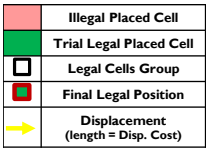


- Cell 4 displacement = 0.89

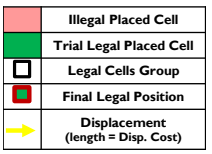
Abacus Algorithm Example (cont.)

- Cell 4 is placed in third row, the row with the least displacement from original global placement location

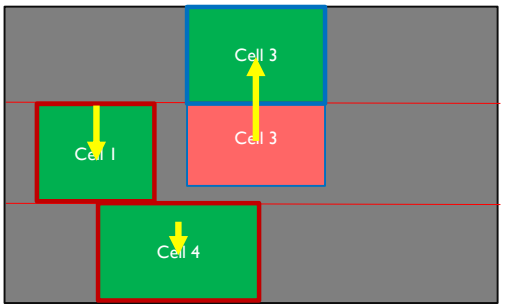




Abacus Algorithm Example (cont.)

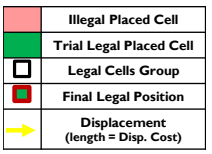


- Placement of Cell 3 in first row

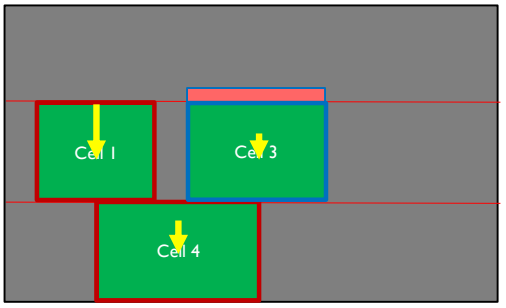


- Cell 3 displacement = 2.22

Abacus Algorithm Example (cont.)



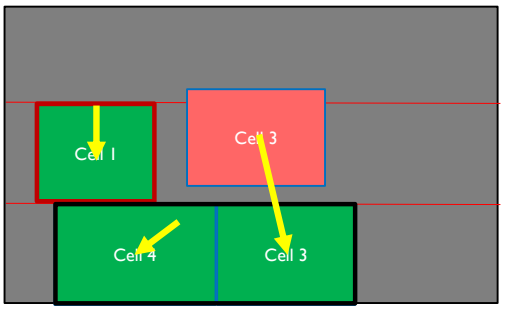
- Placement of Cell 3 in second row

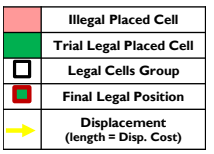


- Cell 3 displacement = 0.69

Abacus Algorithm Example (cont.)

- Placement of Cell 3 in third row

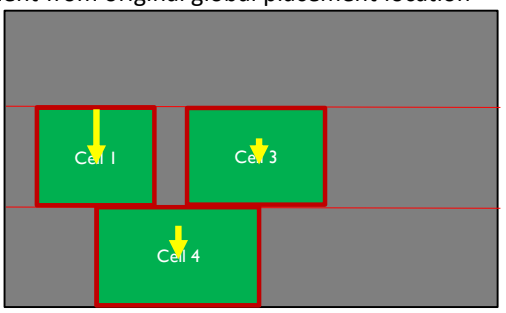


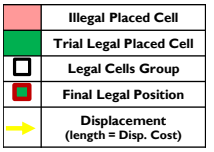


- Cell 3 displacement = 3.1

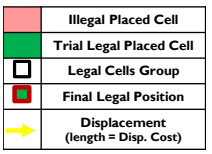
Abacus Algorithm Example (cont.)

- Cell 3 is placed in second row, the row with the least displacement from original global placement location

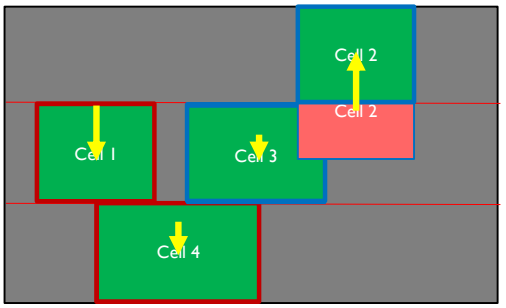




Abacus Algorithm Example (cont.)

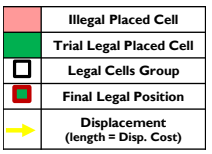


- Placement of Cell 2 in first row

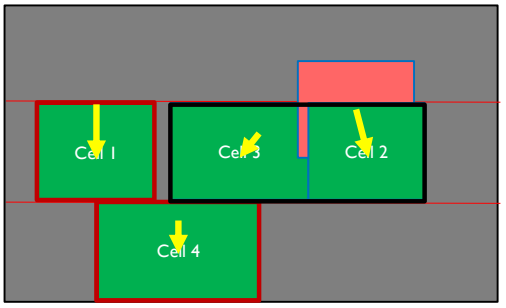


- Cell 2 displacement = 1.9

Abacus Algorithm Example (cont.)



- Placement of Cell 2 in second row

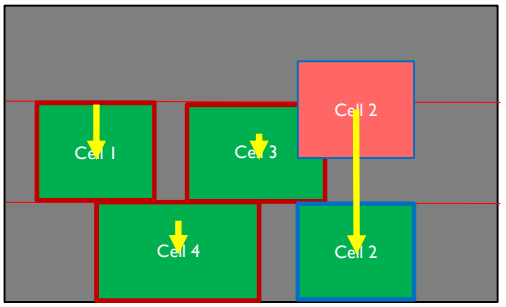


- Cell 2 displacement = 1.8

Abacus Algorithm Example (cont.)

	Illegal Placed Cell
	Trial Legal Placed Cell
	Legal Cells Group
	Final Legal Position
	Displacement (length = Disp. Cost)

- Placement of Cell 2 in third row

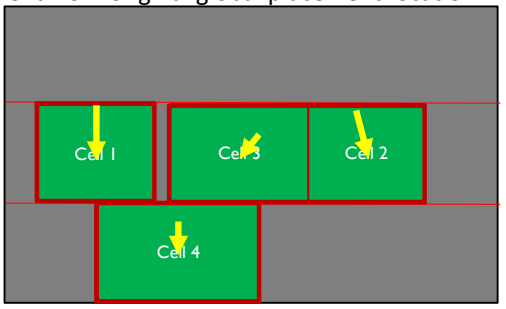


- Cell 2 displacement = 3.83

Abacus Algorithm Example (cont.)

	Illegal Placed Cell
	Trial Legal Placed Cell
	Legal Cells Group
	Final Legal Position
	Displacement (length = Disp. Cost)

- Cell 2 is placed in second row, the row with the least displacement from original global placement location



- This completes the Legalization process for this example!

Contents

- Introduction
- Abacus Algorithm Basics
- **Abacus2**
- Results
- Conclusion and Future Work

Abacus2 - Our Work

Feature Set Comparison

Features			Abacus	Abacus2
Cell Ordering Support	Increasing		✓	✓
	Decreasing		✓	✓
	Center-outwards		✗	✓
Displacement Cost Functions Support	Single-cell		✓	✓
	Multi-cell Mean		✗	✓
	Multi-cell Total		✗	✓
Search Bounding Methods	Exhaustive (Slow)		✓	✓
	Bounded	Single-Cell	✓	✓
		Multi-Cell Total	✗	✓
		Multi-Cell Mean	✗	✓
Blockages/Hard Macros Support	Sub-Row Assign Algorithm		✗	✓
	Sub-Row Re-Assign Algorithm		✗	✓
Row Overflow Checks			✗	✓
Multi-Row Height Cell			✗	✓ (in progress)
Better Quality of Results (QoR)			✗	✓

Feature 1: Cell Selection Ordering

- Abacus2 is a sequential Algorithm
  - Cell order affects QOR
    - Displacement Cost, WL, *etc.*
- Three possible Selection Orders:
  - **Increasing Order (IO)**
  - **Decreasing Order (DO)**
  - **Center-outwards Order (CO)**
- Observed Displacement Cost difference w.r.t. Cell Order
  - **~30%**

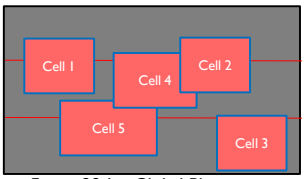


Figure 29.1 – Global Placement

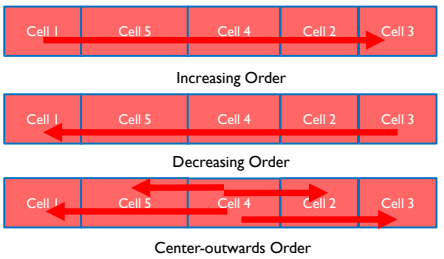


Figure 29.2 – Selection Orderings

Feature 2: Displacement Cost Function (DCF)

- Cell Displacement Functions
  - Based on Euclidean Distance

1) **Single-cell Displacement – F_{SD}**
   - **Current** cell displacement
2) **Multi-cell Total Displacement – F_{TD}**
   - Displacement **sum** of perturbed cells
3) **Multi-cell Mean Displacement – F_{MD}**
   - **Mean** displacement of perturbed cells

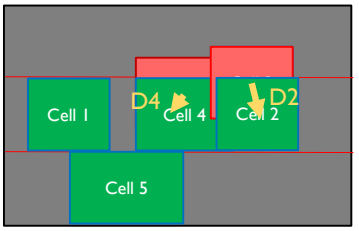
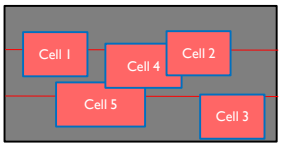


Figure 30.1 – Assign Cell 2 in second row

- $|F_{SD}| = D2$
- $|F_{TD}| = D2 + D4$
- $|F_{MD}| = \frac{D2 + D4}{2}$

Feature 3: Row Search

- Row Search Approaches
  - **Exhaustive Search (ES)**
    - Explore all rows
  - **Bounded Search (BS)**
    - Row search range is related to the ***Displacement Cost Function***
  1. Single-Cell
     - Fast
     - Range is Over-Constrained and Suboptimal
  2. Multi-Cell Total
     - Slow
  3. Multi-Cell Mean
     - Best Overall!



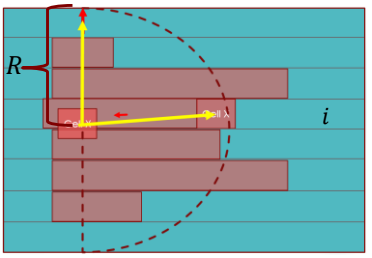
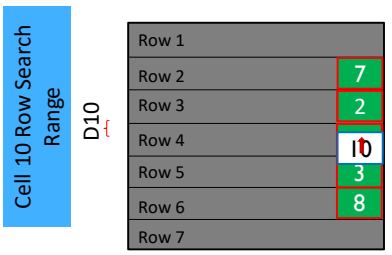


Figure 31.1 – Exploration Range for BS

Feature 3: Row Search (cont.)

Single-Cell Displacement Bound Example

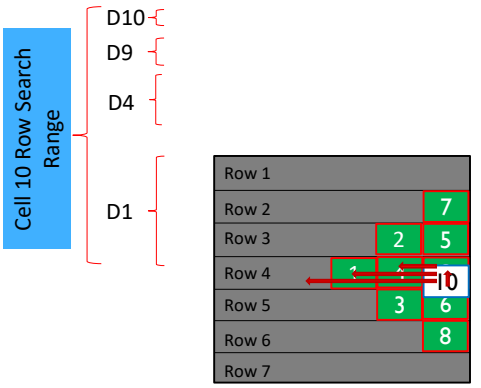
- Over-Constrained Range
  - Leads to sub-optimal solutions
- Fast



Feature 3: Row Search (cont.)

Multi-Cell Total Displacement Bound Example

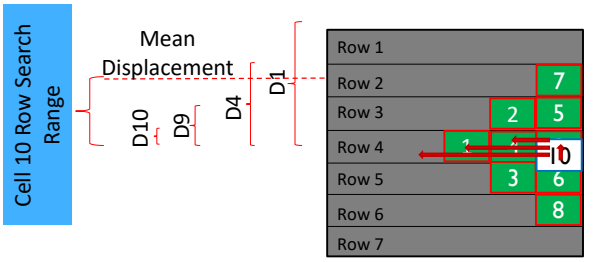
- Dense Areas
  - Range is Large
- Very slow
  - Approaches Exhaustive Search



Feature 3: Row Search (cont.)

Multi-Cell Mean Displacement Bound Example

- Best Overall
  - Uniform Legalizations
    - No horizontal, or vertical artifacts (*explained later*)
  - Faster than Multi-Cell Total
- Cons
  - Slower than Single-Cell



Feature 4: Abacus2 Blockage Handling

- Blockages
  - Memories, Analog IP, Hard Macros, *etc.*
  - Fixed, not movable
- Standard-Cell Placement
  - In the remaining areas
  - Analytical Placement cannot directly support blockages!
- Our Approach
  - Legalize cells in Free Space between blockages
- Two Algorithms for Blockage Support
  - **SRA** (Sub-Row Assignment) Algorithm
  - **SRR** (Sub-Row Re-assignment) Algorithm

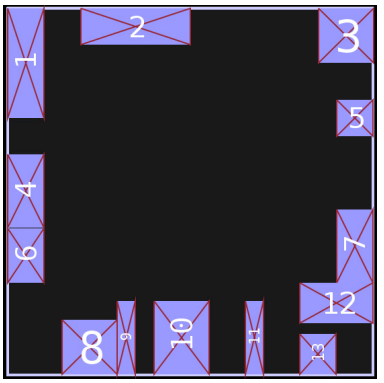


Figure 35.1 – Design with Blockages

Feature 4: Abacus2 Blockage Handling

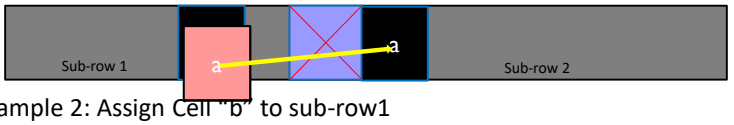
SRA Algorithm (Sub-Row Assignment)

Components are placed in the sub-row with minimum displacement and **become fixed**.

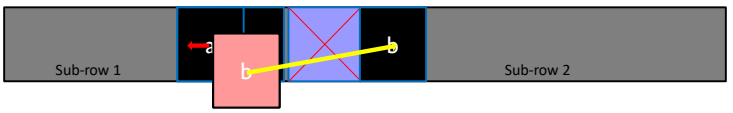
Feature 4: Abacus2 Blockage Handling (cont.)

SRA Algorithm (Sub-Row Assignment)

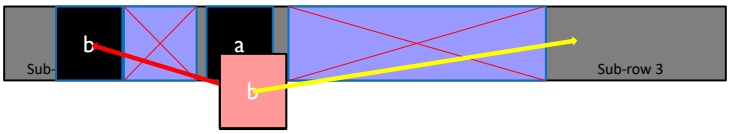
- Example 1: Assign Cell "a" to sub-row1, as its displacement cost is smaller than that of sub-row2



- Example 2: Assign Cell "b" to sub-row1



- Example 3: Assign "b" to sub-row1
- **Initial Cell Order is NOT maintained within the row**



Feature 4: Abacus2 Blockage Handling

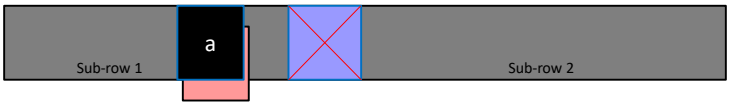
SRR Algorithm (Sub-Row Re-Assignment)

Cells are **recursively reassigned** to neighboring sub-rows

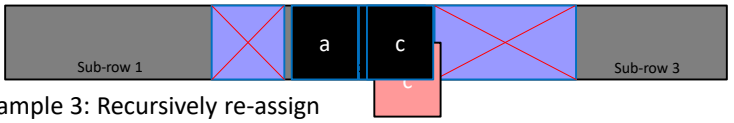
Feature 4: Abacus2 Blockage Handling (cont.)

SRR Algorithm (Sub-Row Re-Assign)

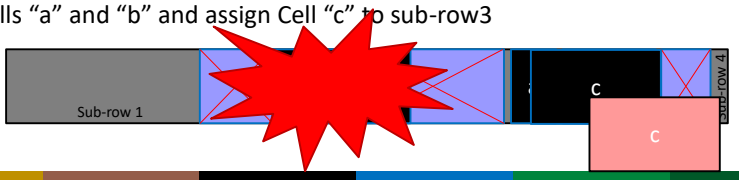
- Example 1: Assign Cell “a” to sub-row1



- Example 2: Re-assign Cell “a” to sub-row1 and then assign Cell “c” to sub-row2



- Example 3: Recursively re-assign Cells “a” and “b” and assign Cell “c” to sub-row3



Feature 5: Row Overflow Check

- Fundamental Algorithm Omission
  - Row Overflow

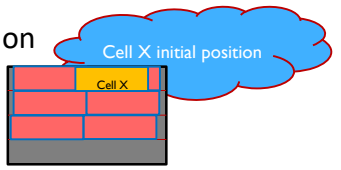


Figure 40.1

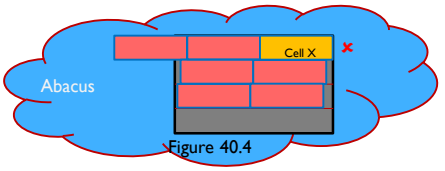


Figure 40.4

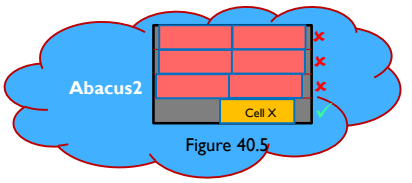


Figure 40.5

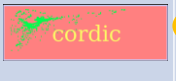
Contents

- Introduction
- Abacus Algorithm Basics
- Abacus2
- **Results**
- Conclusion and Future Work

Displacement Cost Function Feature

Abacus vs Abacus2

- **Horizontal and Vertical Cell Spreading Artifacts**

Core Aspect Ratio	Global Placement WL:	Single-Cell Function Legalization:	Multi-Cell Total Function Legalization:	Multi-Cell Mean Function Legalization:
1:1				
2:1				
3:1				

Results

Benchmark Set

- 8 Benchmarks
- Global Components' Positions
  - QP Placement Minimization
  - No Spreading Minimization
  - 50% Core Utilization
  - 1:1 Aspect Ratio

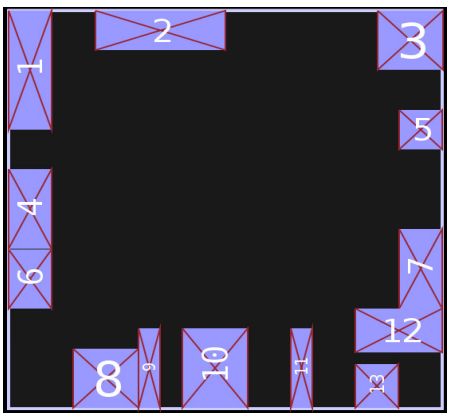
	Design Name	# Components	Type
Small Industrial Benchmarks	execute	2346	Hierarchical
	a53_fp_alu	18796	Hierarchical
Larger Academic Benchmarks (ICCAD'14 & ISPD'14)	bridge32_1	30675	Flat
	fft	32281	Flat
	cordic_I4	41601	Flat
	edit_dist_1	130661	Flat
	matrix_mult	155325	Flat
	b19	219268	Flat

Figure 43.1 - Benchmarks

Results

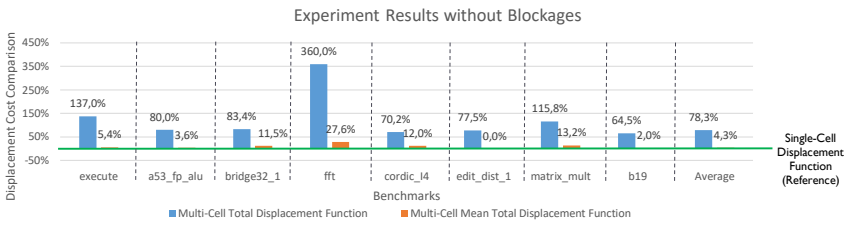
Blockages / Hard Macro Analysis

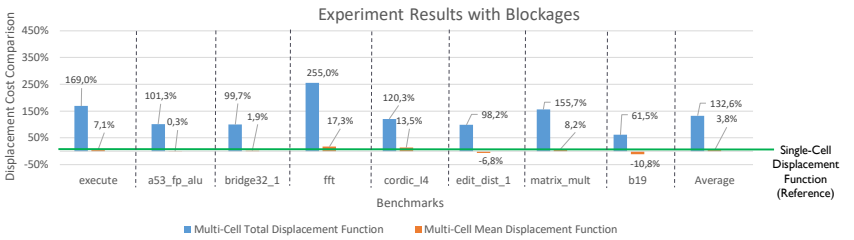
- Blockage Pattern Occupy ***20%*** of core area



Results

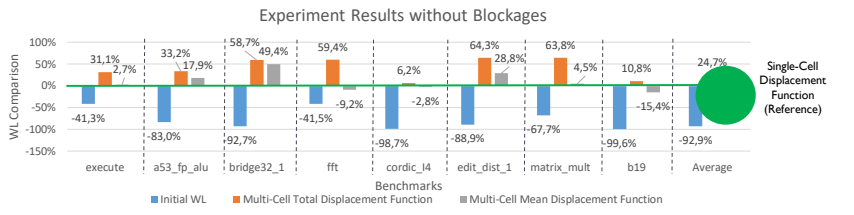
Total Cell Displacement Comparison

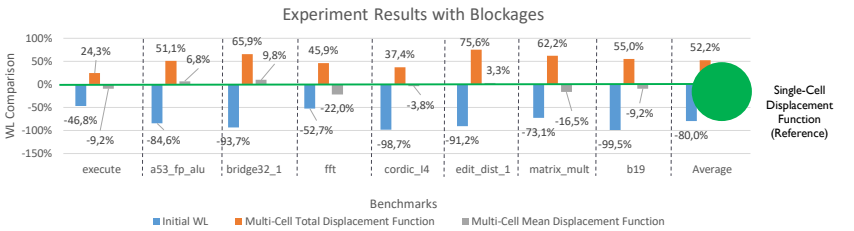




Results

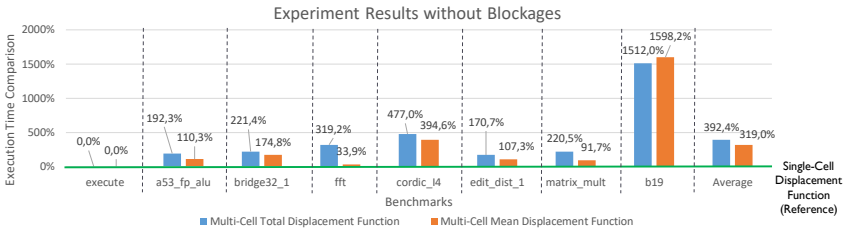
WL Comparison

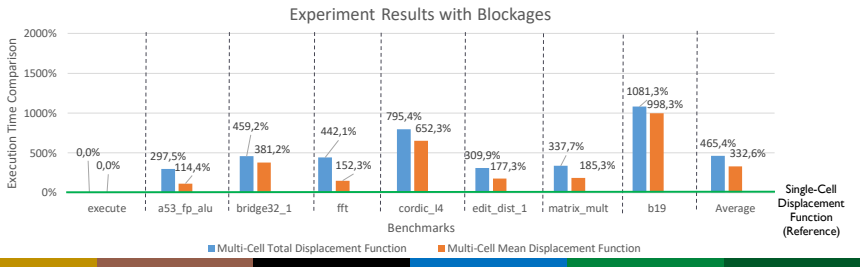




Results

Execution Time Comparison



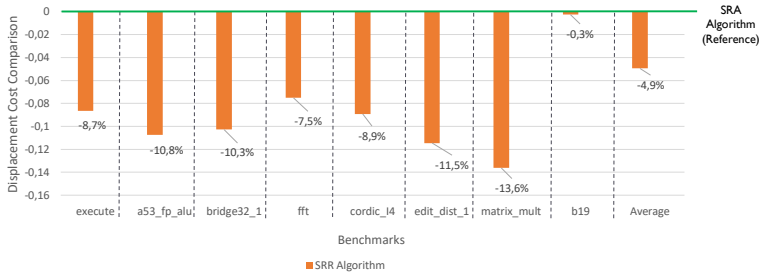


Results

SRA vs SRR Comparison

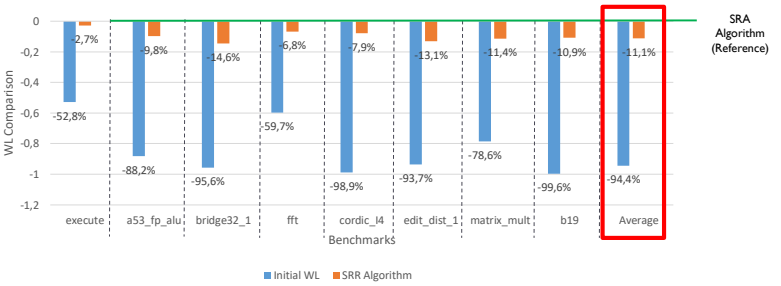
Results

SRA vs SRR Displacement Cost Comparison



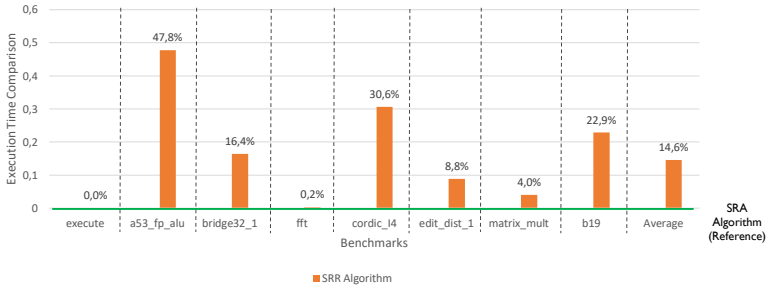
Results

SRA vs SRR WL Comparison



Results

SRA vs SRR Execution Time Comparison



Contents

- Introduction
- Abacus Algorithm Basics
- Abacus2
- Results
- **Conclusions and Future Work**

Conclusions

Abacus2 Legalizer

- Abacus2 Legalizer
  - Can handle small to large Designs
  - Cell Ordering Analysis
    - Observed Displacement Cost difference of ~**30%** based on different Cell Orders
- Multi-Cell Mean Displacement Cost Function
  - Best QOR
  - Homogeneous Cell Spreading, no Horizontal or Vertical Artifacts
  - **6% Better WL**
- Row Search Bounds
  - Abacus Single Cell Bound is over-constrained and suboptimal as illustrated
  - Multi-Cell Mean Displacement Bound produces best QOR overall
- Blockage Support Legalization Algorithms:
  SRR vs **SRA** (Re-assign vs. Assign and Fix)
  - **SRR produces 5% less Displacement Cost** to SRA
  - **SRR achieves 11% less WL**
  - SRR is **14%** slower

Future Work

- Implement an Analytical Placer (In progress)
- Expand Legalizer for 3D circuits
- Multi-Row Height Cells Bipartite Matching Legalization
- Explore Other Heuristic Parameters and Methodologies

Any Questions?