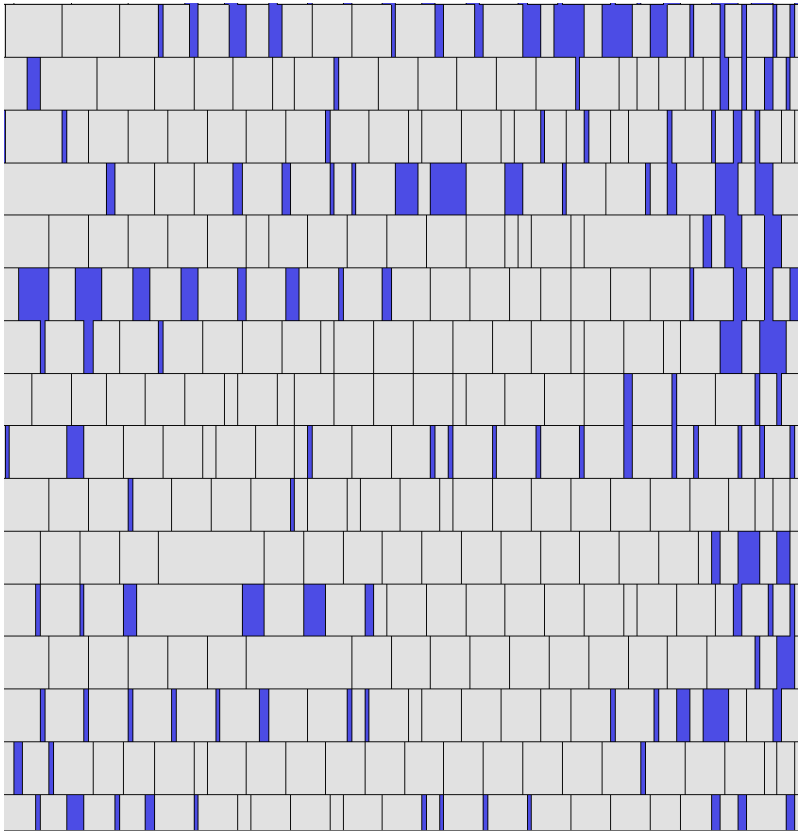
Detailed Placement Refinement for Complex Manufacturing Constraints

Introduction

- Traditionally, detailed placement puts standard cells to placement sites in rows without overlap to optimize timing and routability



Introduction

- Nowadays, various manufacturing constraints means that a traditionally legal detailed placement solution may not be feasible
- Manufacturing constraints e.g. abutment constraints, implant area constraints, must be decomposable into desired number of masks, etc.

Detailed Placement Refinement

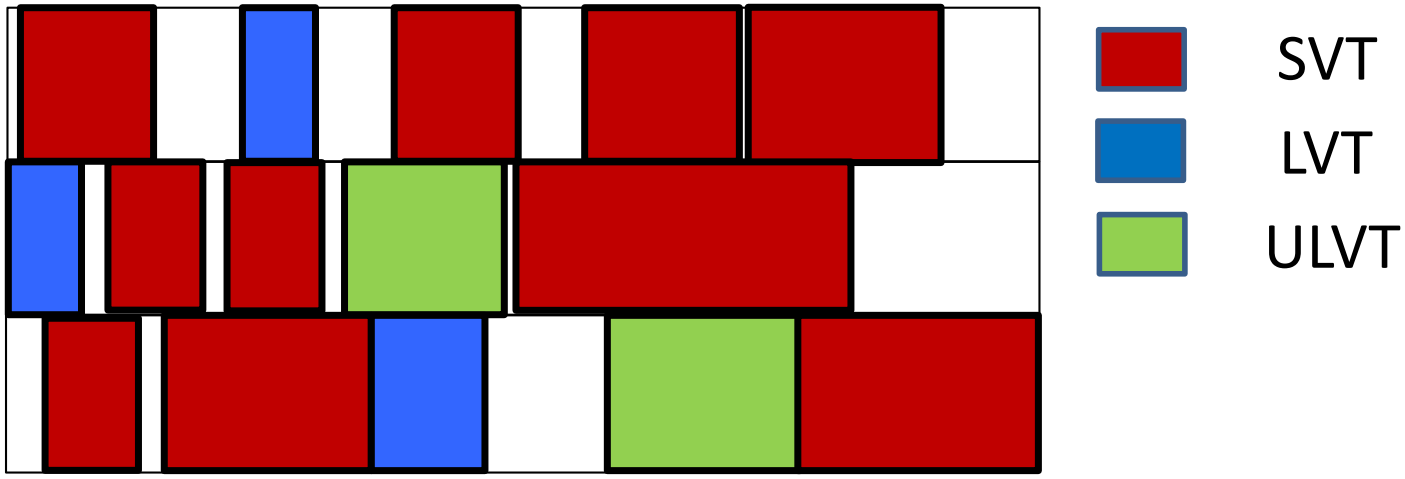
- Inputs:
  - An initial routability and timing-optimized detailed placement of a netlist
  - Manufacturing constraints
- Output:
  - A refined detailed placement with as little perturbation as possible to preserve routability and timing while satisfying the manufacturing constraints

Minimum Implant Area-Aware Placement and Threshold Voltage Refinement

W.K.Mak W.S.Kuo S.H.Zhang
S.I. Lei C.Chu

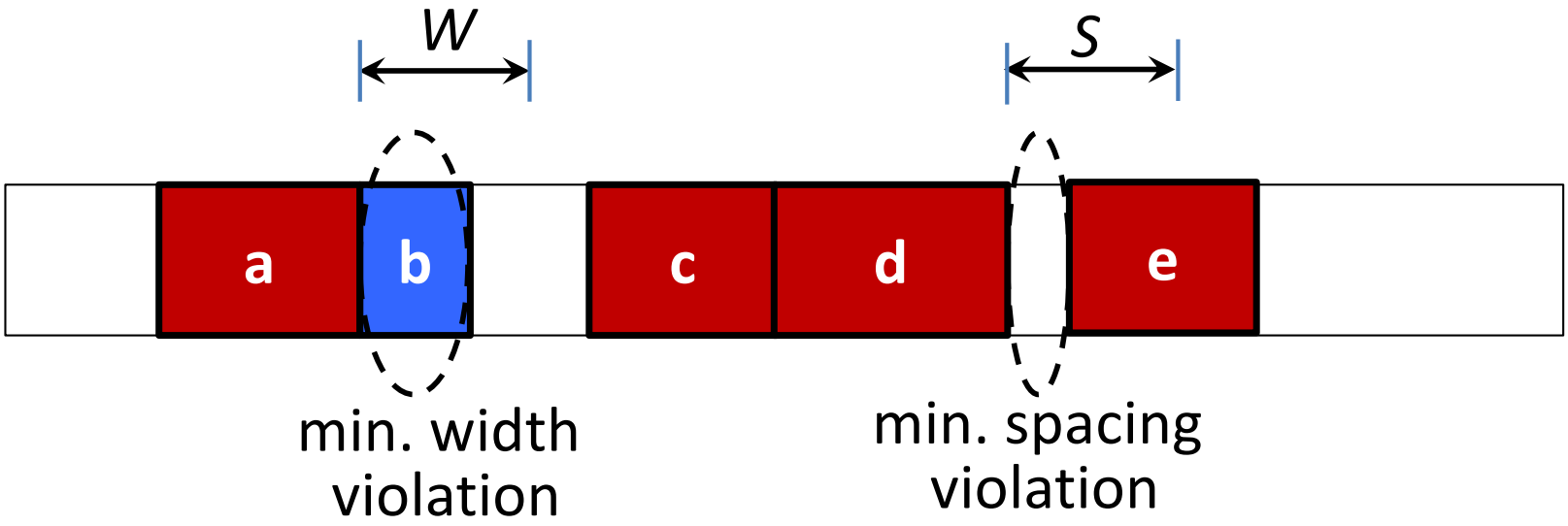
Introduction

- Multi-V_t (multiple threshold voltage) is common for power-aware high-performance chip design.
- E.g. mixing standard V_t (SVT), low V_t (LVT) and ultra-low V_t (ULVT)
  - Lower V_t : faster but more leaky
  - Higher V_t : slower but less leaky



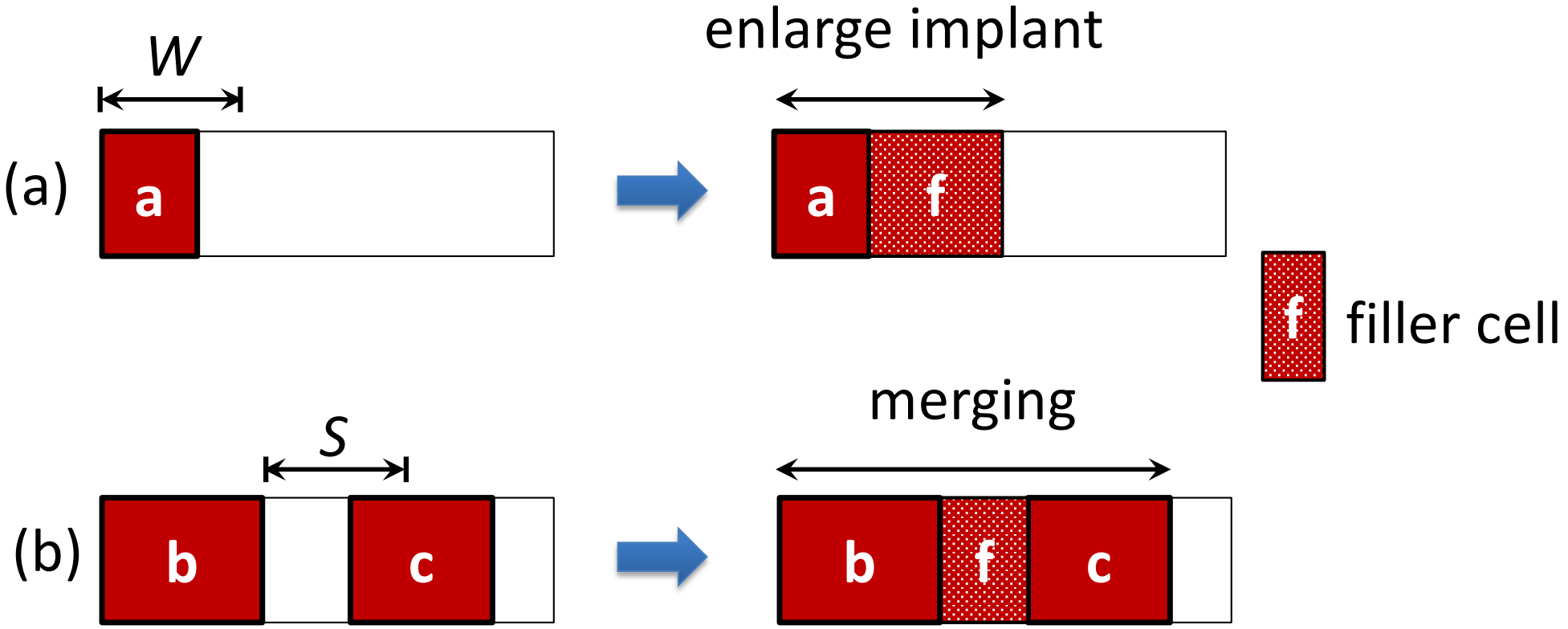
Minimum Implant Area Constraints

- V_t of a transistor is controlled by ion implantation of substrate.
- Manufacturing constrains the
  - minimum implant width *W*
  - minimum spacing *S* between the same type of implant regions



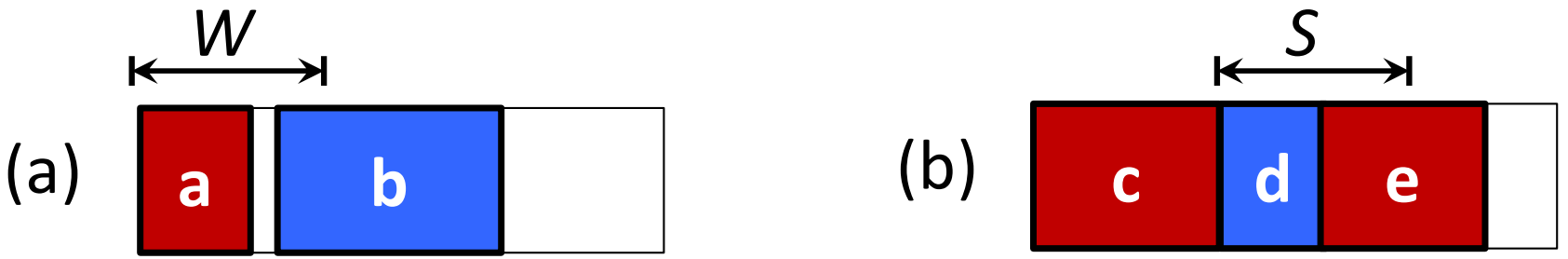
Filler Cell Insertion

- Filler cell insertion is a common way to fix MinIA violations.



New Challenge

- Today, some cell's width has shrunk below *W*.
- It may cause MinIA violations not fixable by filler cell insertion alone.



- Solutions
  1. Move cells further apart (e.g. a and b, or c and e) before inserting filler cell.
  2. Change V_t of some cells.

Heuristic for Fixing MinIA Violations [Kahng+ GLSVLSI14]

- Given an initial placement, [Kahng+ GLSVLSI14] presented an iterative heuristic to fix MinIA violations
  - Fix MinIA violations one at a time (local view)
  - Sequentially apply (1) filler cell insertion, (2) V_t re-assignment, (3) placement perturbation, (4) gate sizing, in four different phases
  - No guarantee that all violations can be fixed
  - Only consider forming a larger implant area for a narrow cell with its immediate left and right whitespace/neighbors, so success rate will reduce when avg. cell size reduces further

Fixing MinIA Violations by ILP-based Approach [Lei+ TCAD17]

- Given an initial placement, [Lei+ TCAD17] proposed an optimal algorithm to fix all MinIA violations.
  - Consider all MinIA violations simultaneously (global view)
  - Concurrently apply (1) filler cell insertion, (2) V_t re-assignment, (3) placement perturbation (no gate sizing)
  - Guarantee that all violations can be fixed
  - Based on mixed integer linear programming w/ practical run time

Problem Formulation [Lei+ TCAD17]

- MinIA-aware Detailed Placement and V_t Refinement
  - Given
    - an initial detailed placement
    - initial power-optimized V_t assignment of each standard cell
    - minimum implant width *W* and spacing *S*
    - allowable displacement range of each cell from its initial location based on its timing criticality
    - power penalty for reducing the V_t of each cell
  - Compute
    - a legal placement with no overlap and no MinIA constraint violation
    - minimizing total cell displacement and power overhead

Useful Observations

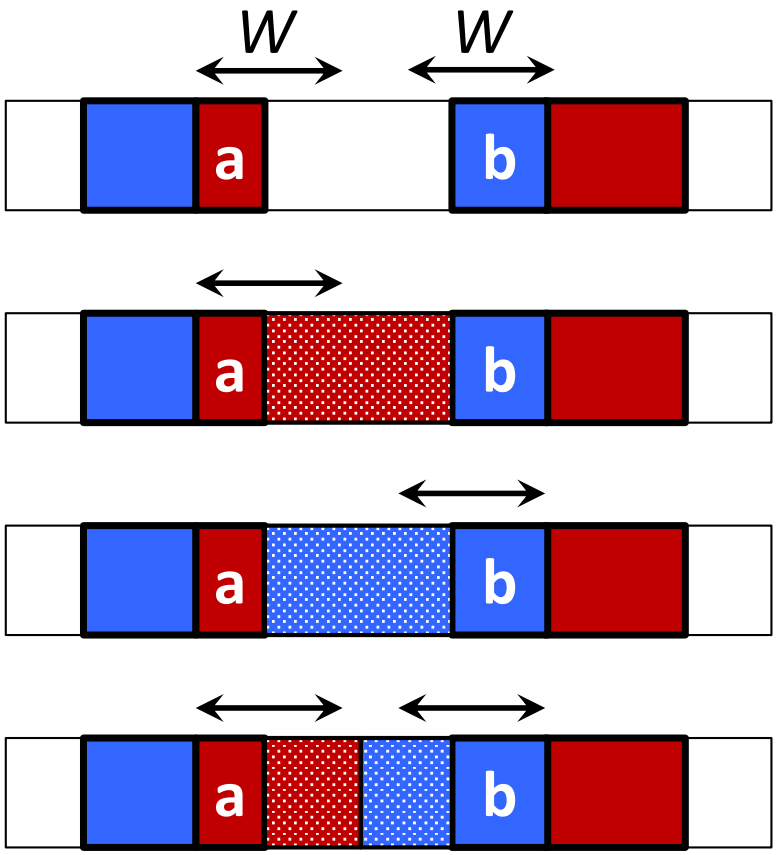
1. Division of a whitespace between two cells into left and right subspaces is helpful.

(a) Original configuration

(b) Fix one violation

(c) Fix one violation

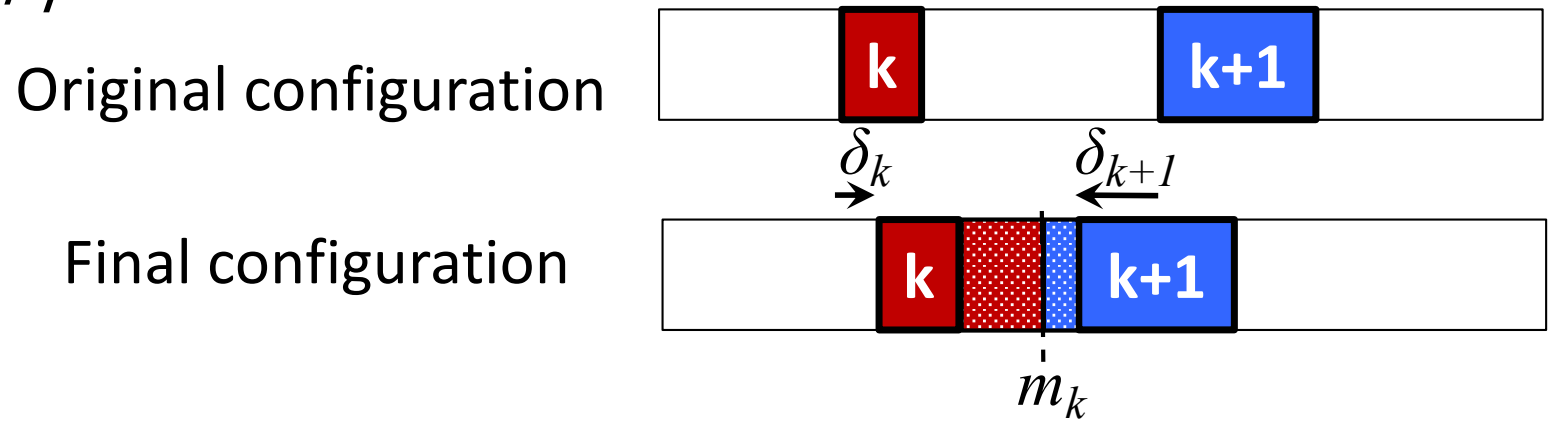
(d) Fix both violations



2. Suffice to consider min width constraint only if all whitespaces are inserted with filler cells since $S \leq W$.

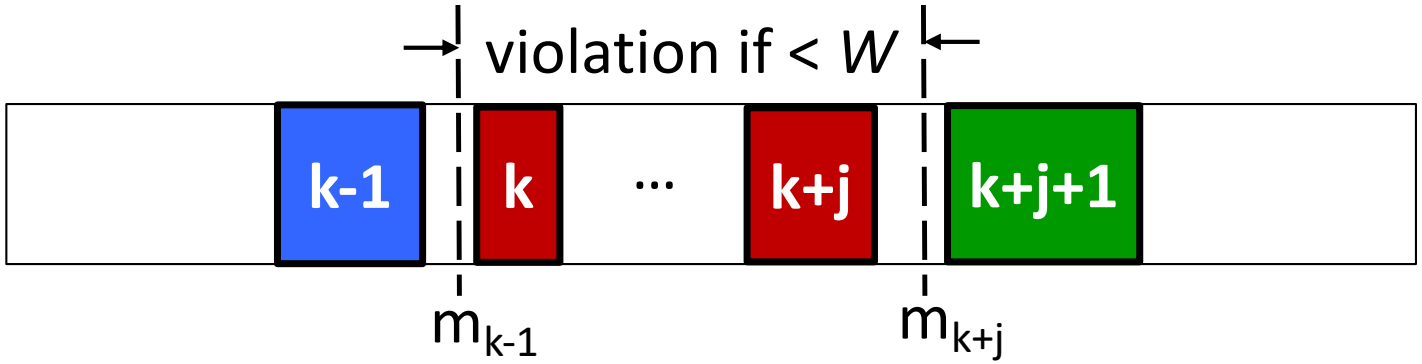
Mixed Integer Linear Programming Approach

- Row based approach
- Binary variables
  - a_k^S , a_k^L , a_k^{UL} : 1 iff cell *k*'s V_t is re-assigned to SVT, LVT, ULVT, resp.
- Continuous variables
  - δ_k : final displacement of cell *k* from its original location
  - m_k : division point of whitespace between cells *k* and *k+1*



Mixed Integer Linear Programming Approach

- Ensure no narrow V_t island is formed



  - If V_t of cells *k-1* and *k* differ, and V_t of cells *k+j* and *k+j+1* differ, then $m_{k+j} - m_{k-1}$ must be at least *W* or a MinIA violation occurs, i.e.,

$$m_{k+j} - m_{k-1} \geq (d_{k-1,k} + d_{k+j,k+j+1} - 1)W$$

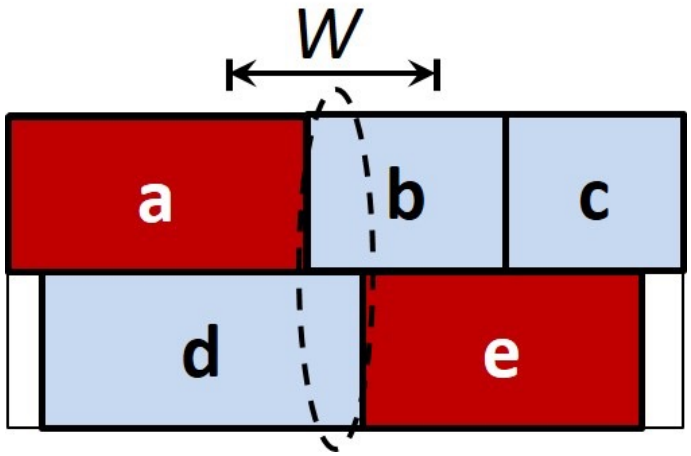
    where vars *d*'s can be expressed in terms of vars *a*'s.

Remarks

- Special case
  - If V_t re-reassignment is not considered, it becomes a linear program only.
- Placement sites
  - Initial placement of all cells align with placement sites. If ILP solver returns a non-integral δ_k or m_k, we may round it down to align with placement site without hurting feasibility.
- Fixed macros
  - Fixed macros can be handled by dividing each row into sub-rows separated by the fixed macros.

Extension: Inter-Row Constraint

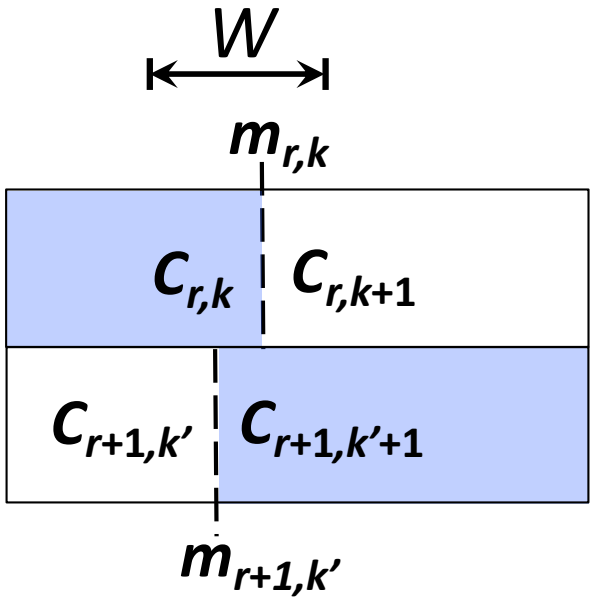
- Inter-row min implant width constraint for sub-10nm nodes
- V_t islands of the same type forbidden to form a narrow staircase



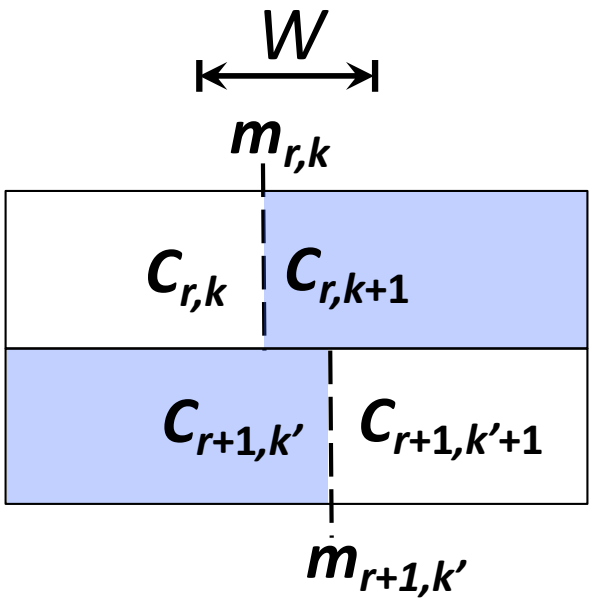
Inter-row minIA constraint violation

Observation

- Two ways of forming a narrow staircase



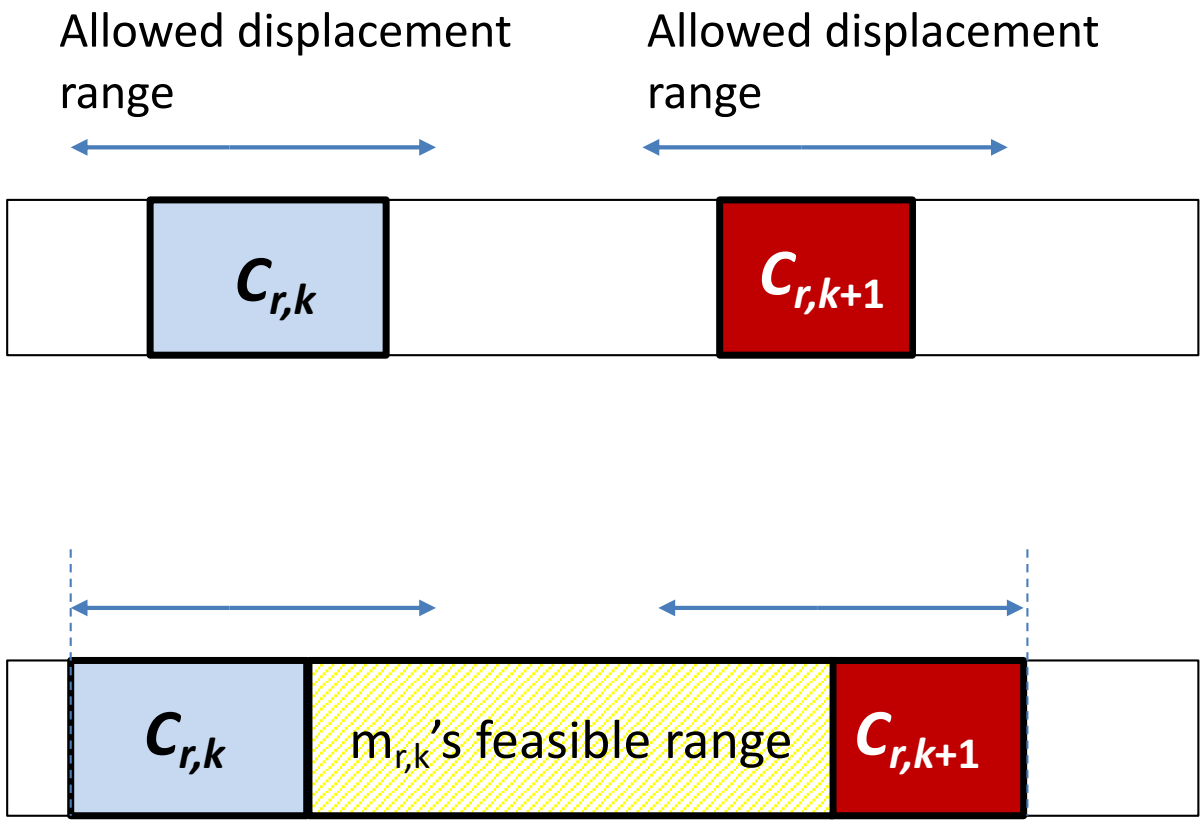
Type A



Type B

Addressing Inter-Row Constraints

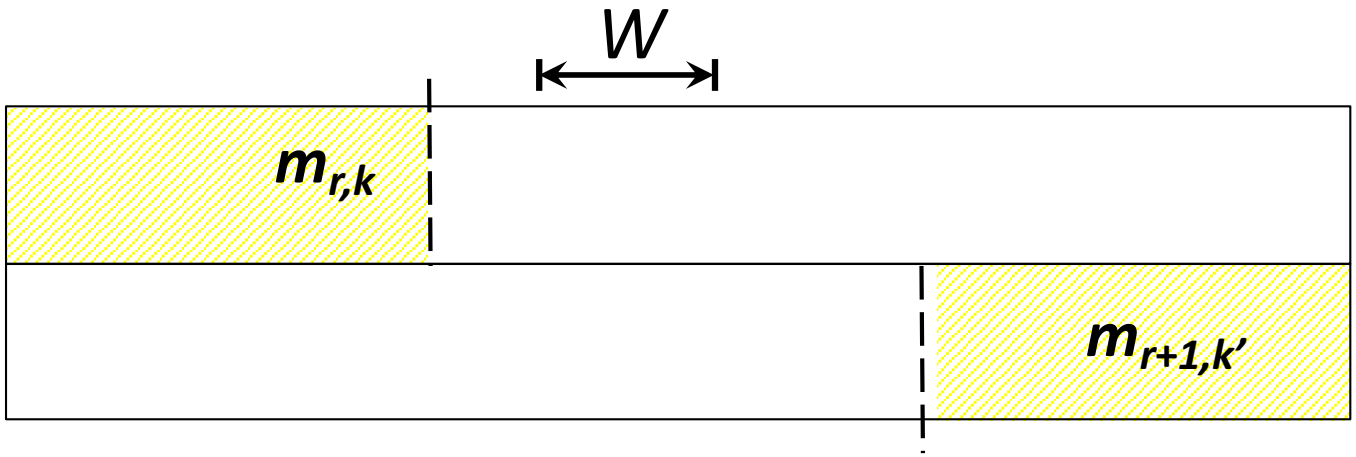
- Feasible range for variable $m_{r,k}$



Addressing Inter-Row Constraints

- Check feasible ranges for $m_{r,k}$ and $m_{r+1,k'}$, consider three different cases

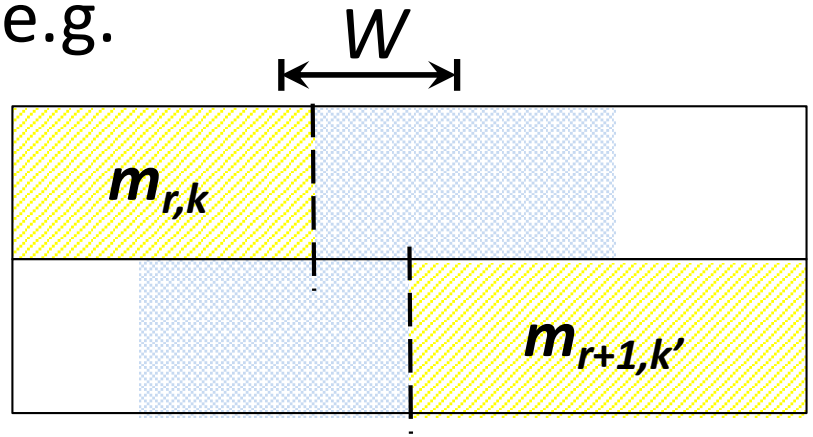
- Case 1. Their ranges do not overlap and are at least W apart
  - Neither type A/B staircase can be formed



Addressing Inter-Row Constraints

- Case 2. Their ranges do not overlap but are less than W apart
  - If $m_{r,k} > m_{r+1,k'}$, add constraint to avoid type A staircase
  - Otherwise, add constraint to avoid type B staircase

e.g.



Add

$$m_{r+1,k'} - m_{r,k} \geq (a^{\tau}_{r,k+1} + a^{\tau}_{r+1,k'} - a^{\tau}_{r,k} - a^{\tau}_{r+1,k'+1} - 1)W$$

for τ = S/L/UL

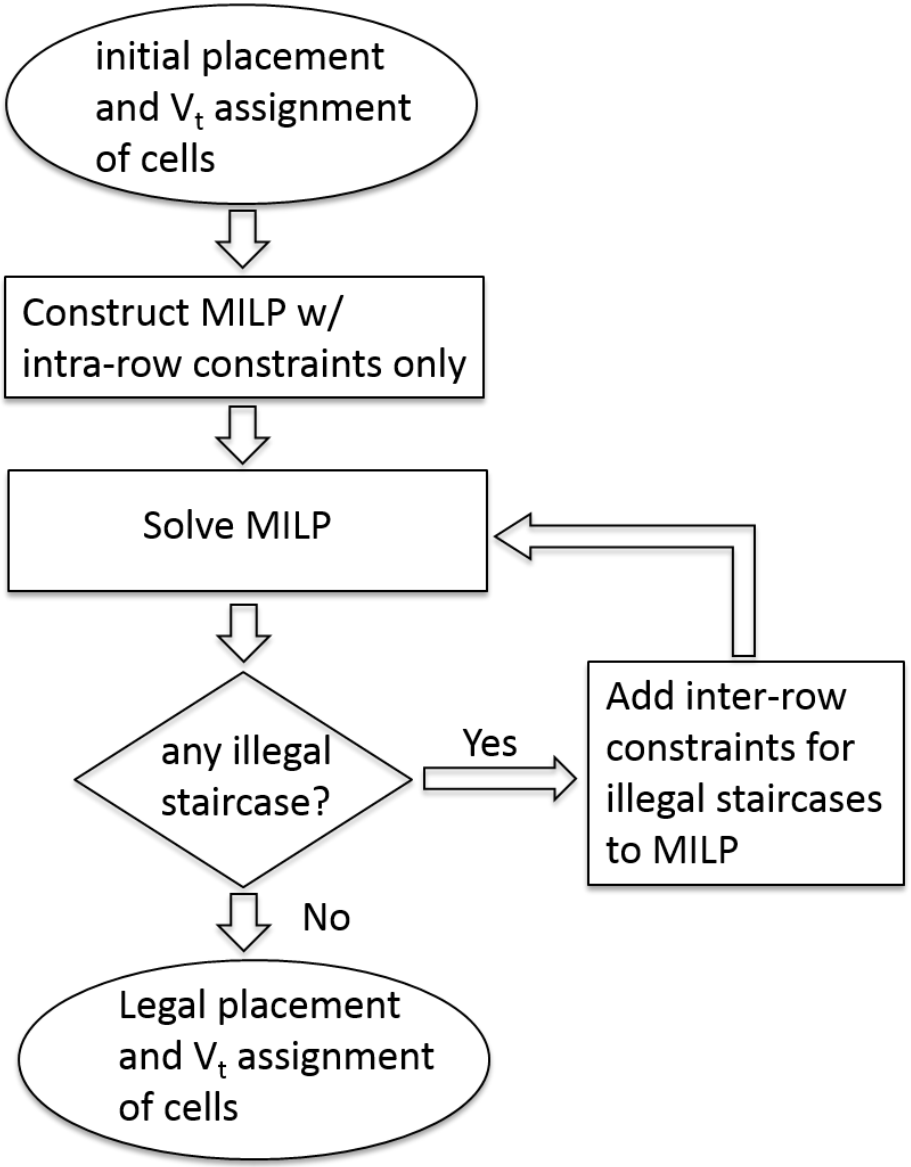
Addressing Inter-Row Constraints

- Case 3. Their ranges overlap
  - So, both $m_{r,k} > m_{r+1,k'}$ or $m_{r,k} < m_{r+1,k'}$ are possible
  - Add constraints as in case 2 with the introduction of binary variables $g_{r,k,r+1,k'}$ and $l_{r,k,r+1,k'}$ s.t. $g_{r,k,r+1,k'}$ must be 1 if $m_{r,k} > m_{r+1,k'}$ and $l_{r,k,r+1,k'}$ must be 1 if $m_{r,k} < m_{r+1,k'}$ (How?)

Speedup Techniques

- The resultant ILP of whole layout with all inter-row constraints is huge
- Divide into subproblems
  - Divide layout into multiple strips horizontally and process strip sequentially
  - Treat last row in the strip above the current strip as fixed (trade optimality for speed)
- Add inter-row constraints on an as-needed basis
  - This may require a few iterations to eliminate all illegal staircases but is usually faster than including all inter-row constraints upfront

Add Inter-Row Constraints on an As-Needed Basis



Conclusions

- For intra-row MinIA constraints, developed an optimal approach for minimum implant-aware placement and threshold voltage refinement
- With
  - cell movement
  - intelligent whitespace division
  - filler cell insertion
  - threshold voltage re-assignment
- Extended to consider inter-row MinIA constraints
- Subsequently, [Jeong+ TVLSI22] proposed breaking the MILP formulation into two simpler MILPs but added some heuristic steps in-between.

References

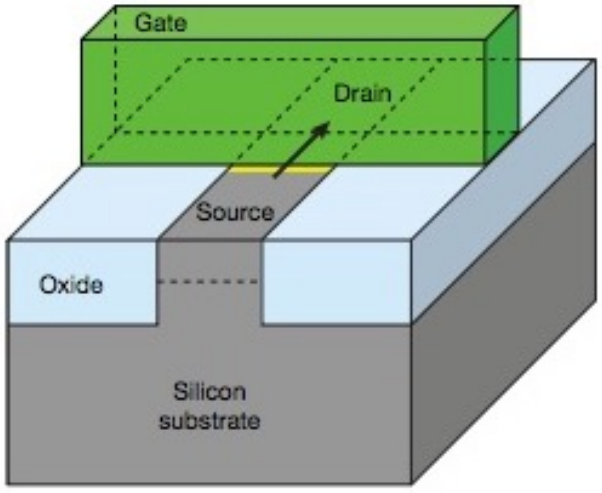
- [Kahng+ GLSVLSI14] Minimum Implant Area-Aware Gate Sizing and Placement
- [Mak+ TCAD17] Minimum Implant Area-Aware Placement and Threshold Voltage Refinement
- [Jeong+ TVLSI 22] Eliminating Minimum Implant Area Violations with Design Quality Preservation

Optimization of Standard Cell Based Detailed Placement for FinFET Process

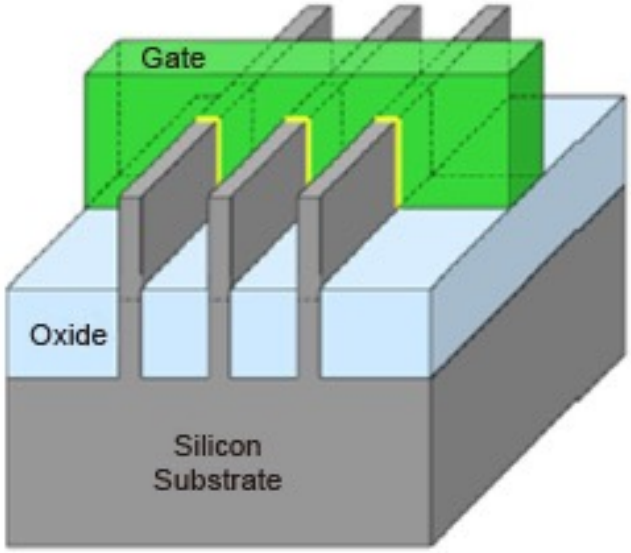
Y. Du and M.D.F. Wong

FinFET

- FinFET (fin based multiple-gate field effect transistors) was introduced to replace traditional planar transistors
  - faster
  - less leakage



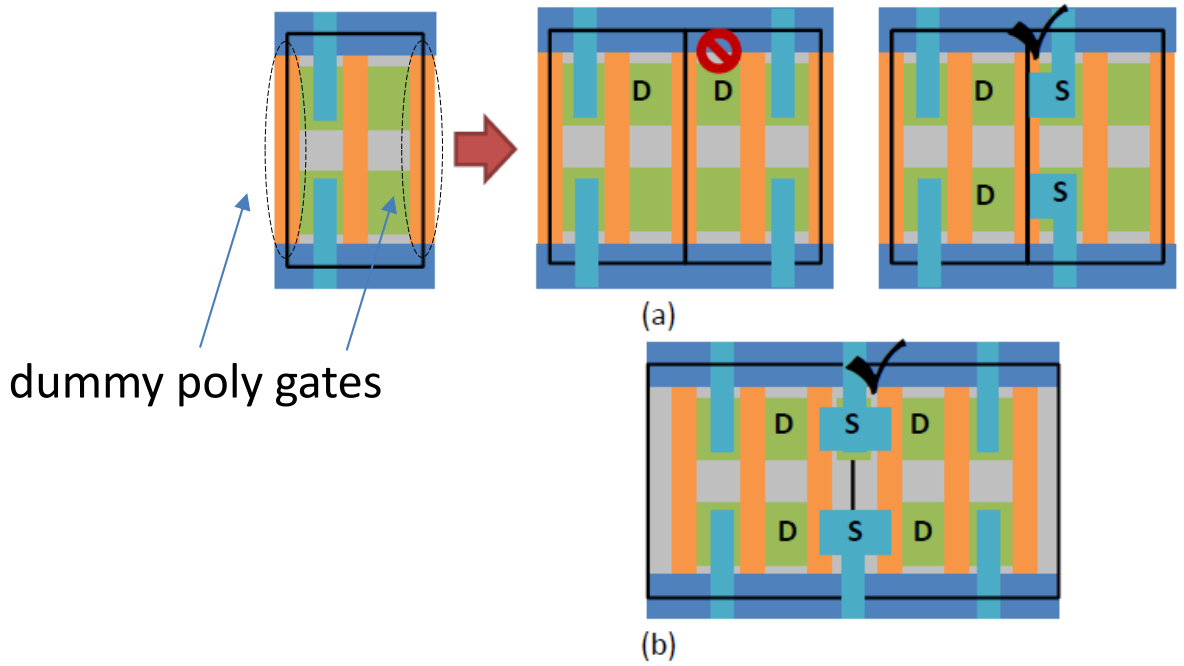
Planar FET



FinFET

Placement Constraint for Drain-Drain Abutment

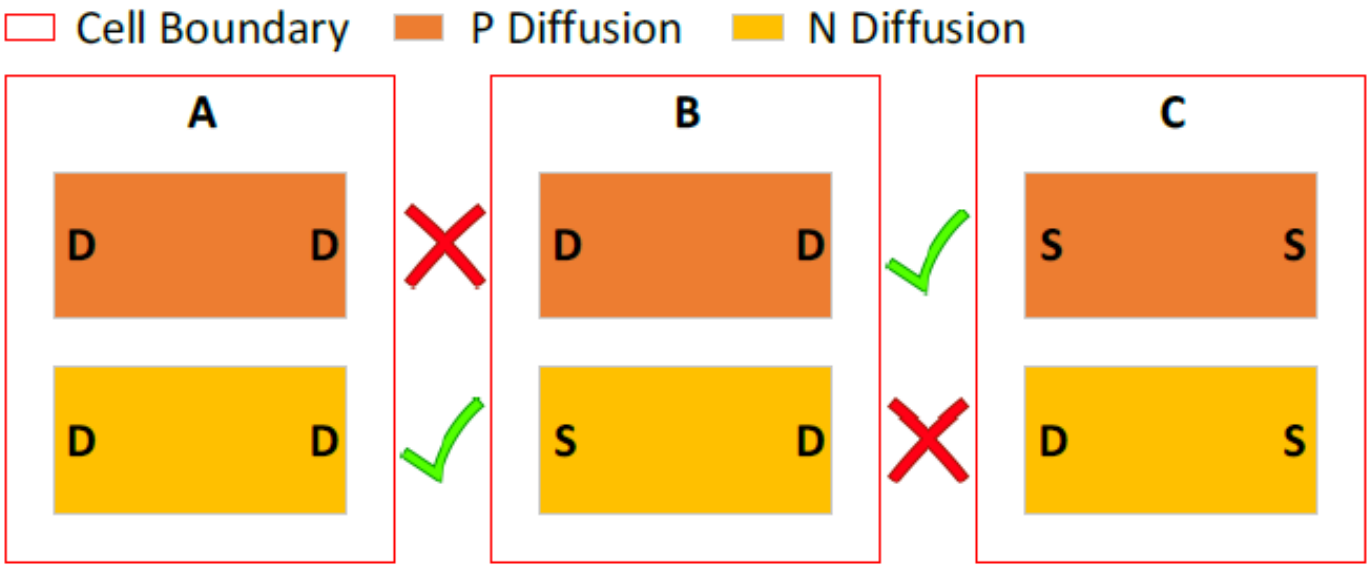
- For FinFET, dummy poly gates exist at vertical cell boundaries to avoid edge device variability
- Dummy poly gates need to be tied to power/ground rail



- Drain-drain abutment of adjacent cells require additional spacing in order to create an extra source node to be tied with power/ground rail as in (b)

Drain-Drain Abutment

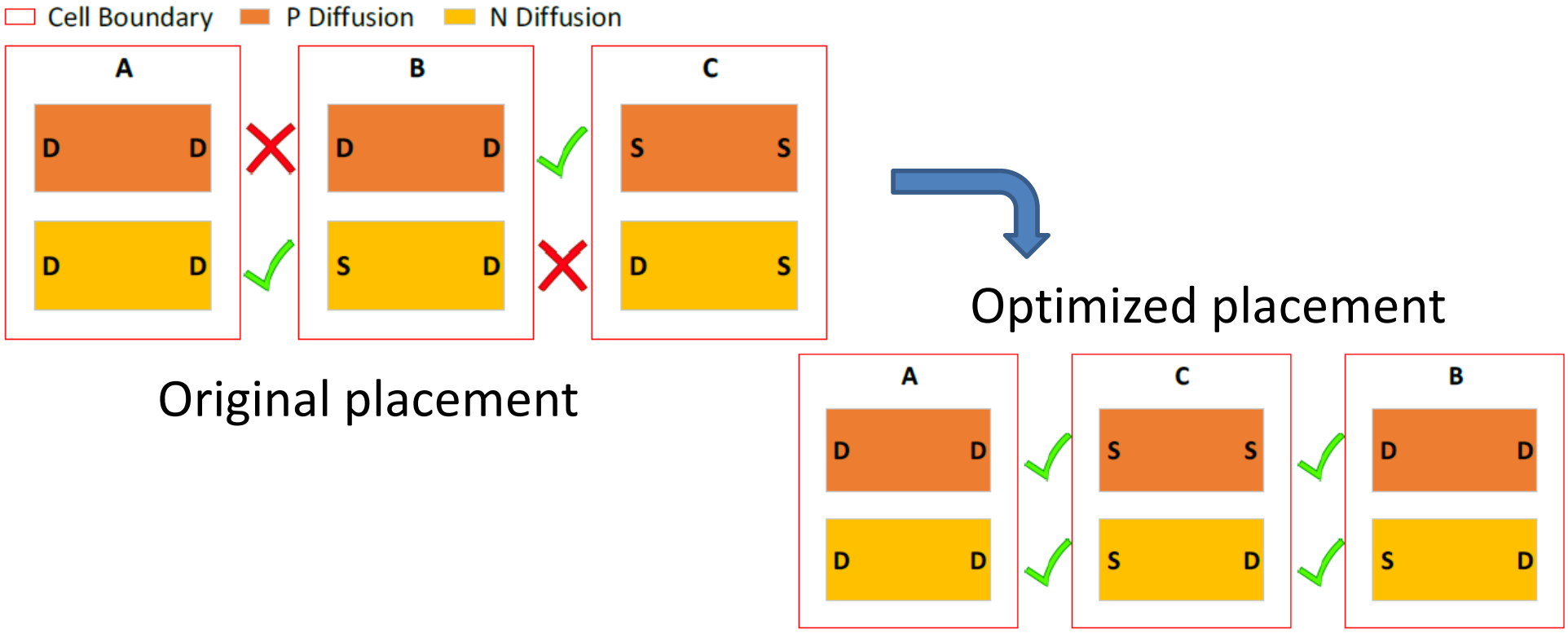
- Drain-drain abutment (DDA) exists between two adjacent cells if either the P-diffusions or the N-diffusions have a DDA problem.



A row placement of 3 standard cells

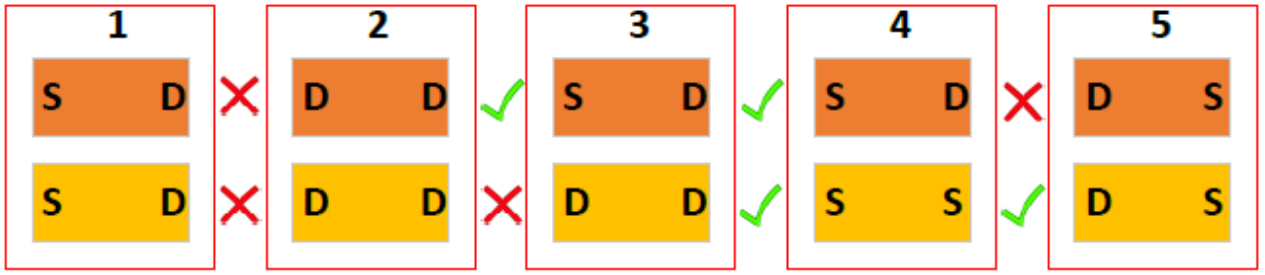
DDA Minimization Problem [Du+ DATE14]

- Given a row of standard cells, horizontally flip some cells and/or switch the positions of adjacent cells such that the total number of DDA between adjacent cells is minimized.



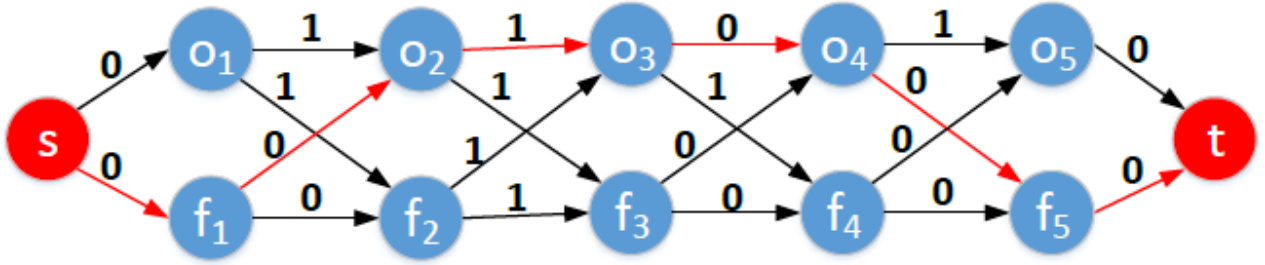
Cell Flipping

- 2 orientations for each cell (original or flipped)

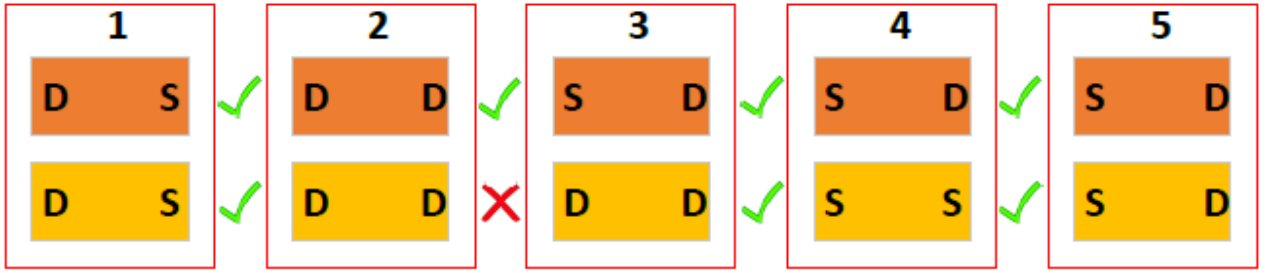


(a) The original placement of five cells.

Model as shortest path problem
- edge cost of 1 for a DDA



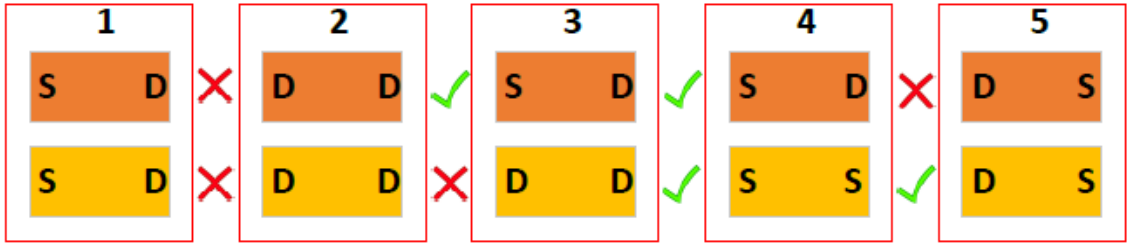
(b) The CFP graph model and a shortest path from s to t.



(c) The optimized placement via cell flipping.

Cell Switching

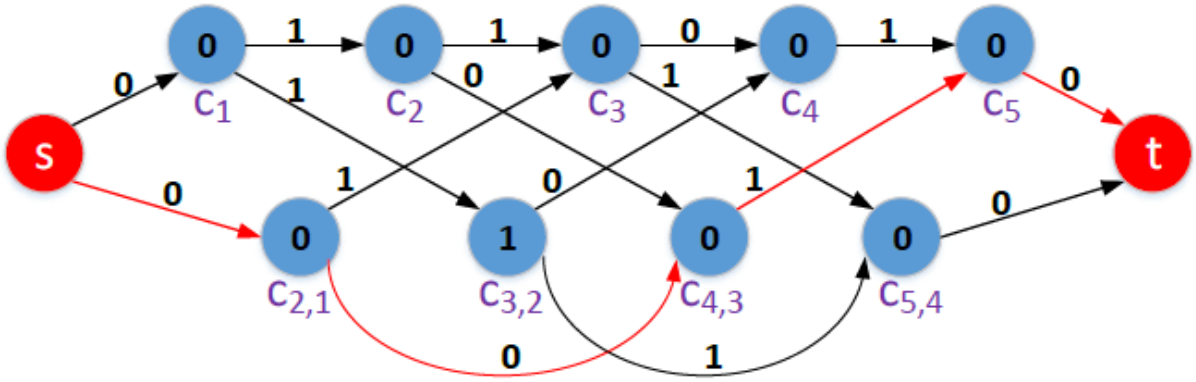
- Allow switching adjacent cells only



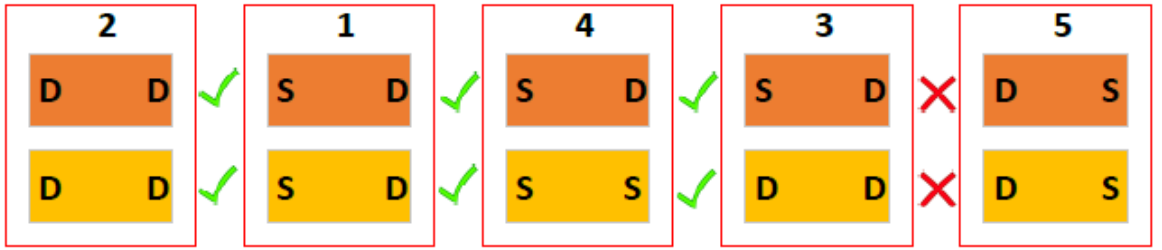
(a) Original placement

Model as shortest path problem

- node $c_{i+1,i}$ denote switching cells i and i+1
- node/edge cost of 1 for a DDA

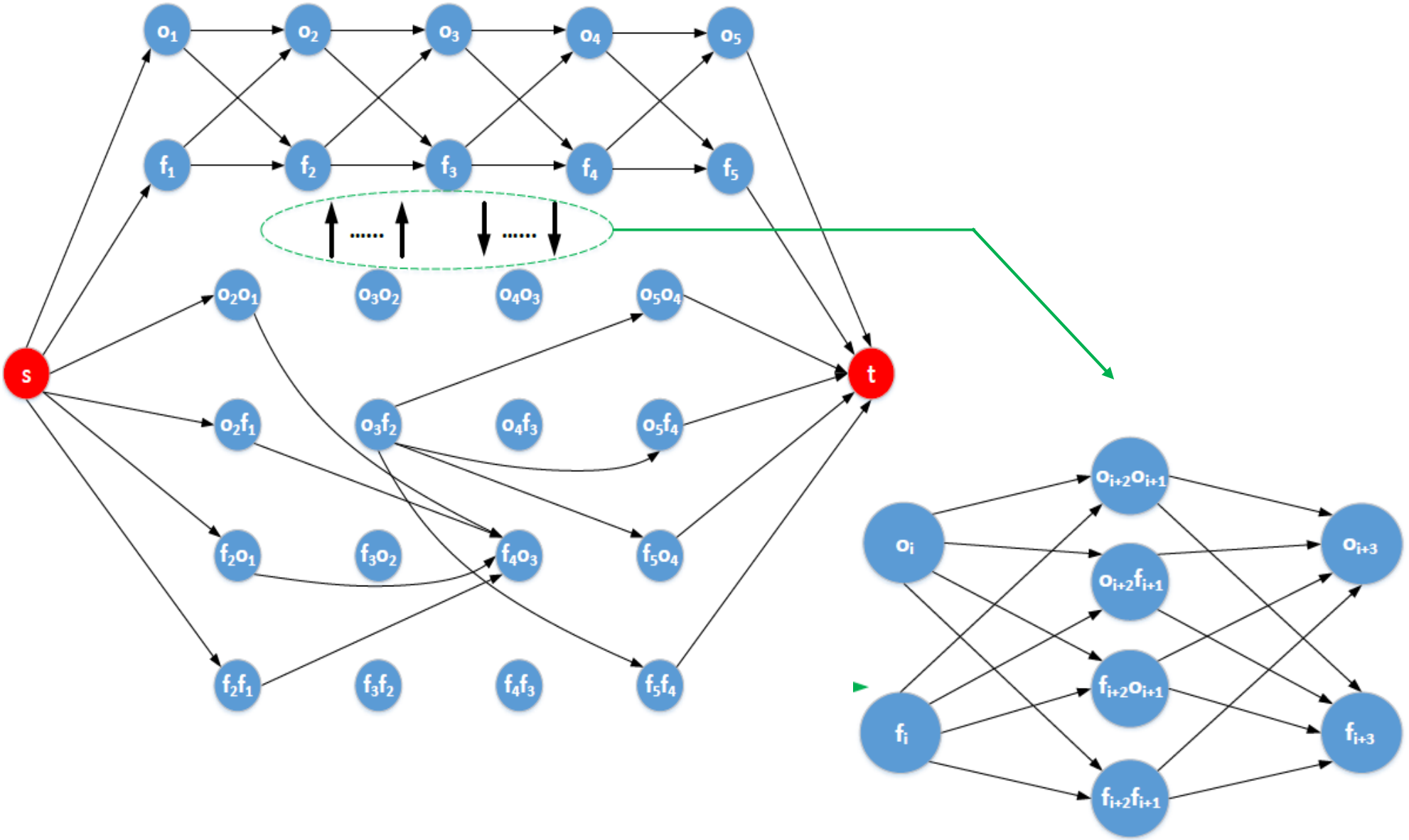


(b) The CSP graph and a shortest path



(c) Optimized placement via cell switching

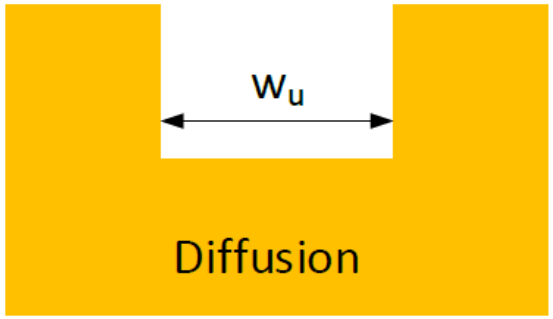
Simultaneous Cell Flipping and Switching



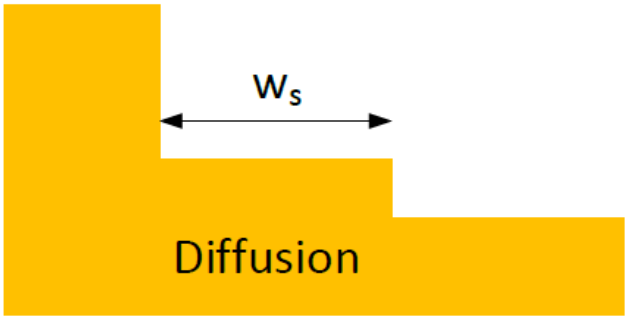
(b) The interactive edges between the first two rows and the last four rows.

Extensions

e.g. Minimum diffusion jog widths

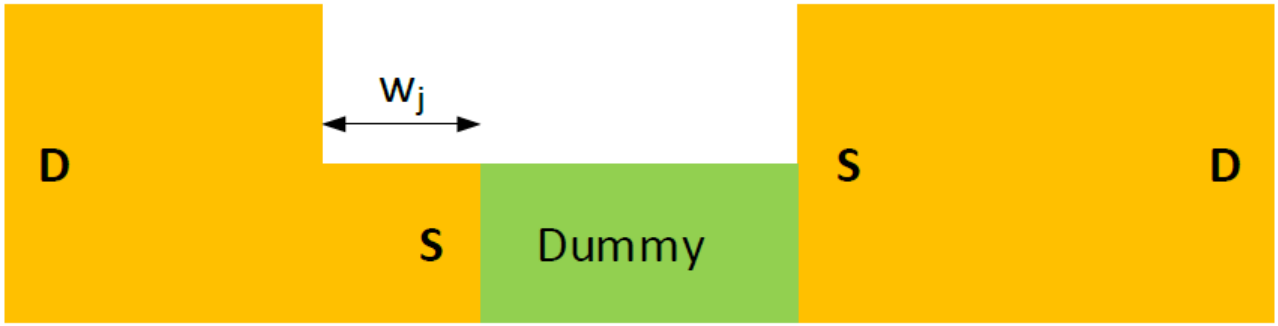


Minimum width of U-shaped jog

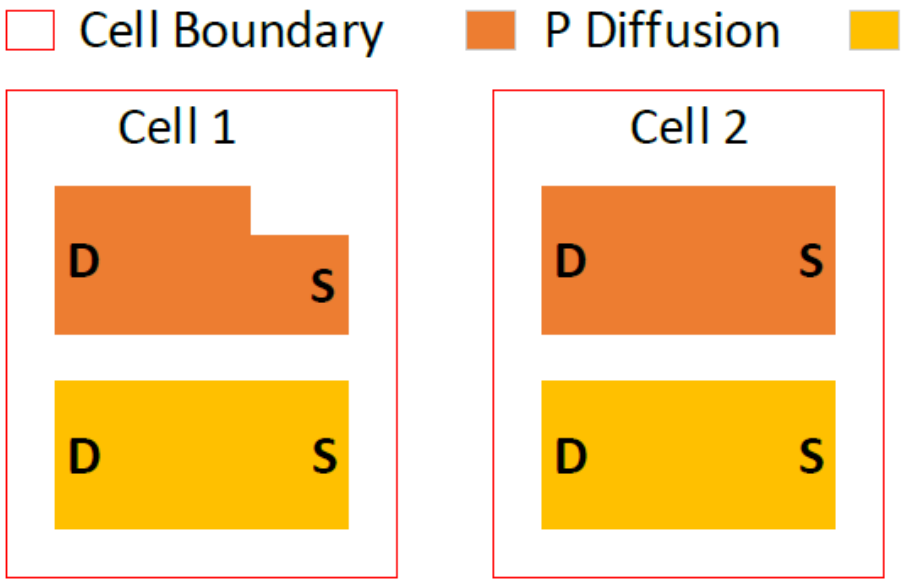


Minimum width of stair-shaped jog

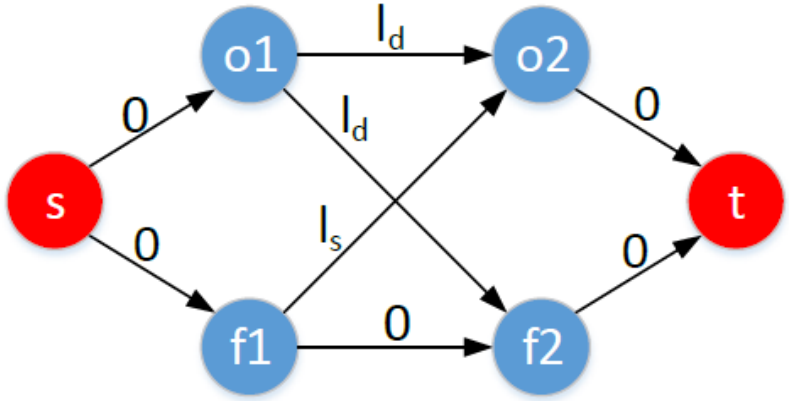
- When cell abutment causes min jog width violation, dummy diffusion has to be inserted



Extensions



Placement of two cells



Modified cell flipping graph (l_s= cost to resolve DDA, l_d = cost to meet min jog width)

Experiments

- Randomly arrange cells in a row with required spacing for each DDA

♯ Cells	Operations	Org. Len. (mm)	Opt. Len. (mm)	Saved Len. (mm)	Runtime (ms)
10k	flip only	9.70	9.53	0.17	2
	flip&switch	9.70	9.24	0.46	7
20k	flip only	19.43	19.07	0.36	8
	flip&switch	19.43	18.50	0.93	20
40k	flip only	38.67	37.96	0.71	12
	flip&switch	38.67	36.81	1.86	28
60k	flip only	58.28	57.21	1.07	14
	flip&switch	58.28	55.51	2.77	44
80k	flip only	78.02	76.58	1.44	20
	flip&switch	78.02	74.27	3.75	59
100k	flip only	96.88	95.12	1.76	23
	flip&switch	96.88	92.24	4.64	75

Reference

- [Du+ DATE14] Optimization of Standard Cell Based Detailed Placement for 16nm FinFET Process