

Simultaneous Placement and Clock Tree Construction for Modern FPGAs

Wuxi Li¹ Mehrdad E. Dehkordi² Stephen Yang² David Z. Pan¹

¹Electrical & Computer Engineering, University of Texas at Austin

²Vivado Implementation Team, Xilinx Inc.





Introduction

Proposed Algorithms

Experimental Results

Conclusion



Introduction

Proposed Algorithms

Experimental Results

Conclusion

Placement for Modern FPGAs

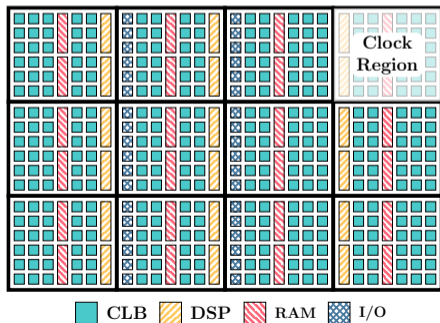


Input A netlist of cells (LUT, FF, DSP, RAM, ...)

Output Cell physical locations in the FPGA layout

Objectives Wirelength, timing, power, routability, ...

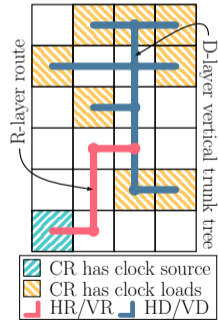
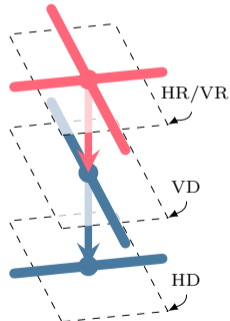
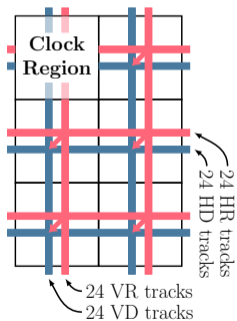
Constraints **Clock network feasibility**, ...



Xilinx UltraScale Clocking Architecture



- ▶ Layout is divided into a grid of clock regions (CRs)
- ▶ Clock network consists of routing Layer (HR/VR) and distribution Layer (HD/VD)
- ▶ 24 HR/VR/HD/VD tracks in each CR
- ▶ Clock tree consists of D-layer vertical trunk tree + R-layer 2-pin route





Simultaneous Placement and Clock Tree Construction Problem

Input A netlist of cells

Output A global placement solution
A clock routing solution

Objectives Min. wirelength

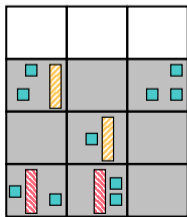
Constraints No logic resource overflow
No clock routing overflow

Simulated annealing-based approach

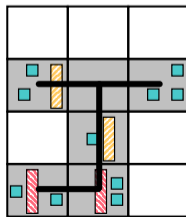
- ▶ [Lamoureux+, TRETS'08]
- ▶ Incorporating clock cost in objective
- ▶ Generic to any clocking architecture
- ▶ Slow convergence

Bounding box-based approach

- ▶ [Kuo+, ICCAD'17], [Pui+, ICCAD'17], [Li+, TODAES'18]
- ▶ Greedily shrinking clock net bounding boxes to reduce overflow
- ▶ Cheap computation and fast convergence
- ▶ Often overestimates clock routing demand



Bounding box



Clock tree



- ▶ Explicit clock tree construction
- ▶ Solution space of clock routing → tree space
Clock routing → tree space exploration process
- ▶ Inspired by branch-and-bound idea, an iterative algorithm is proposed to efficiently explore the tree space
- ▶ A Lagrangian relaxation-based clock tree construction technique is also proposed to achieve feasible clock routing solutions
- ▶ Experiments demonstrate the effectiveness/efficiency of our approach over previous works.



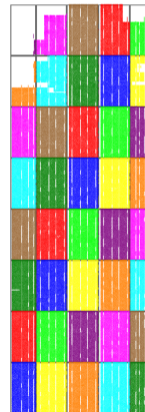
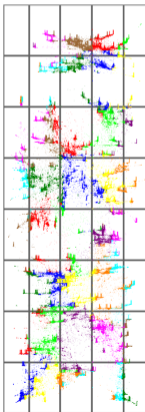
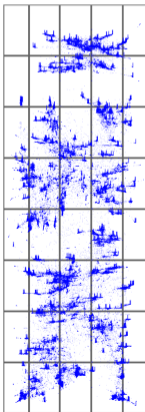
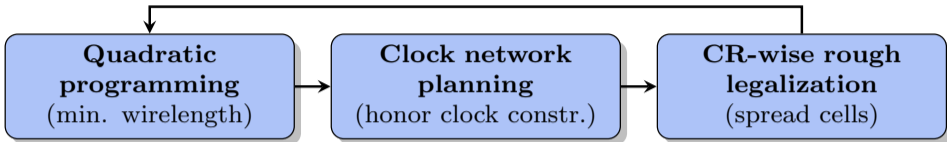
Introduction

Proposed Algorithms

Experimental Results

Conclusion

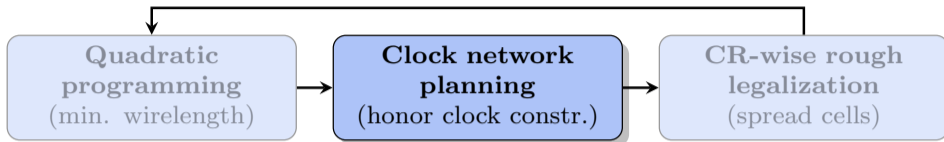
Overall Flow



Overall Flow



Clock Network Planning Problem



Problem Statement

Input A placement produced by quadratic programming

Output A cell-to-CR assignment
A clock routing solution

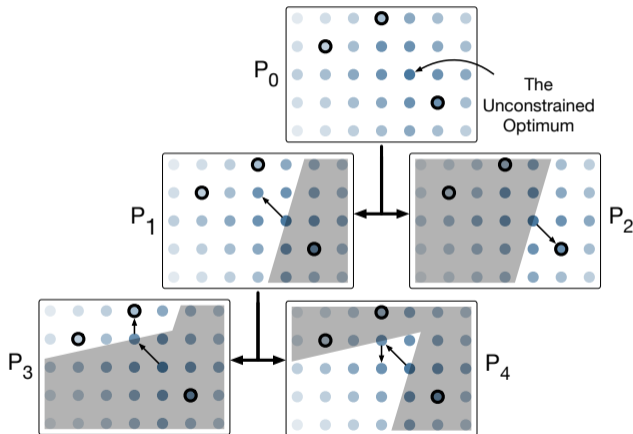
Objectives Min. cell displacement

Constraints No logic resource overflow
No clock routing overflow

Mathematical Formulation

$$\begin{aligned} \min_x \quad & \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}} D_{v,r} \cdot x_{v,r}, \\ \text{s.t.} \quad & x_{v,r} \in \{0, 1\}, \forall v \in \mathcal{V}, \forall r \in \mathcal{R}, \\ & \sum_{r \in \mathcal{R}} x_{v,r} = 1, \forall v \in \mathcal{V}, \\ & \sum_{v \in \mathcal{V}} A_v \cdot x_{v,r} \leq C_r, \forall r \in \mathcal{R}, \\ & \text{Exist a legal clock routing w.r.t. } x. \end{aligned}$$

Branch-and-Bound Idea



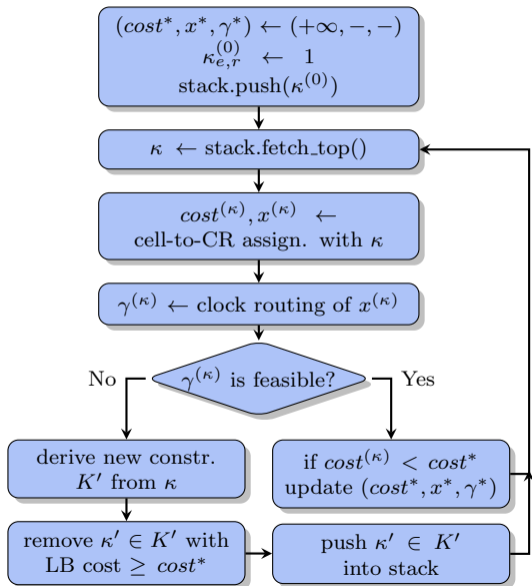
Problem Properties

- ▶ Integer minimization problem with complex constraints
- ▶ Hard to solve directly
- ▶ Can be efficiently solved by relaxing some constraints

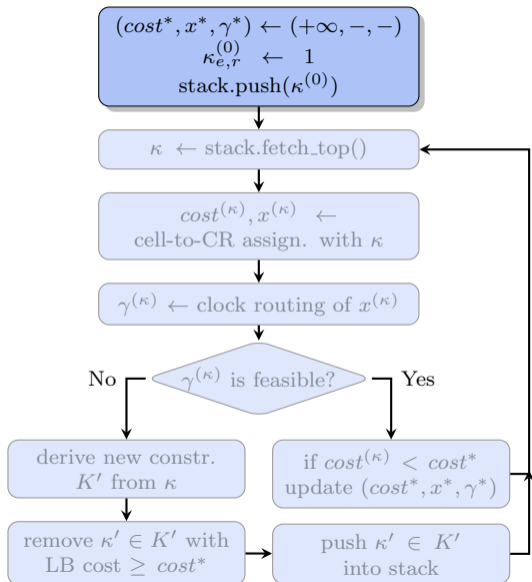
B&B Algorithm

- ▶ Keeping solving the relaxed problem in iteratively branching spaces
- ▶ Tracking the lowest cost of feasible solutions found as the upper bound of optimum
- ▶ Pruning branches with lower bound costs worse than this upper bound

Clock Network Planning Algorithm



Clock Network Planning Algorithm

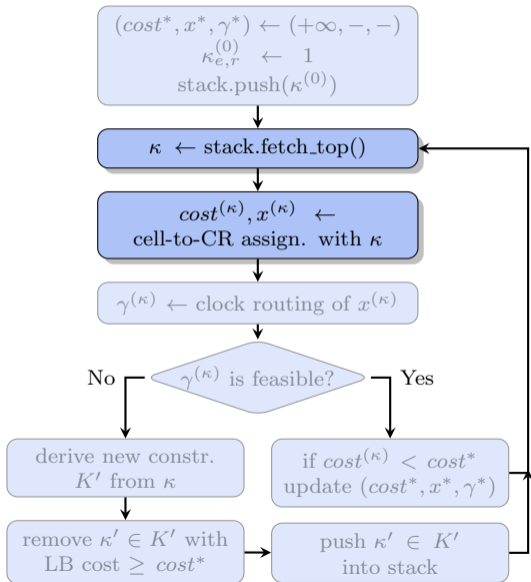


$cost^*$ The best feasible cost found
 x^* The best feasible cell-to-CR assignment
 γ^* The best feasible clock routing
 $\kappa_{e,r}$ Binary values to represent whether cells in clock net e can be assigned to CR r

Initialization

- ▶ Set the best solution found as NONE
- ▶ Allow any cell-to-CR assignment ($\kappa^{(0)}$)
- ▶ Initialize the stack with only $\kappa^{(0)}$

Clock Network Planning Algorithm



Cell-to-CR assignment problem

- ▶ Relax the clock constraint
- ▶ Solve the clock-unconstrained version of the original problem in subspace κ

$$\min_x \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}} D_{v,r} \cdot x_{v,r},$$

$$\text{s.t. } x_{v,r} \in \{0, 1\}, \forall v \in \mathcal{V}, \forall r \in \mathcal{R},$$

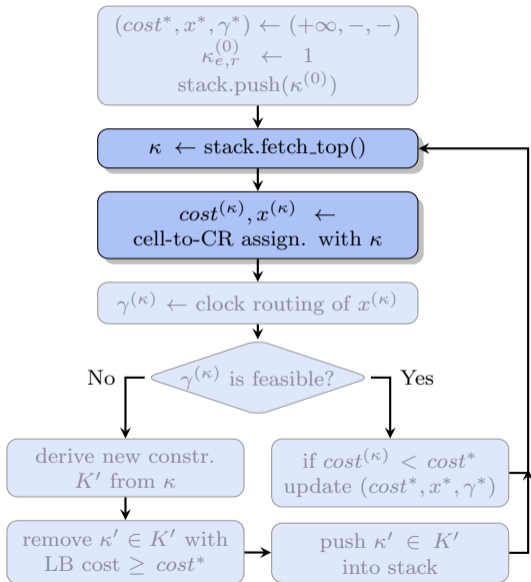
$$\sum_{r \in \mathcal{R}} x_{v,r} = 1, \forall v \in \mathcal{V},$$

$$\sum_{v \in \mathcal{V}} A_v \cdot x_{v,r} \leq C_r, \forall r \in \mathcal{R},$$

~~Exist a legal clock routing w.r.t. x .~~

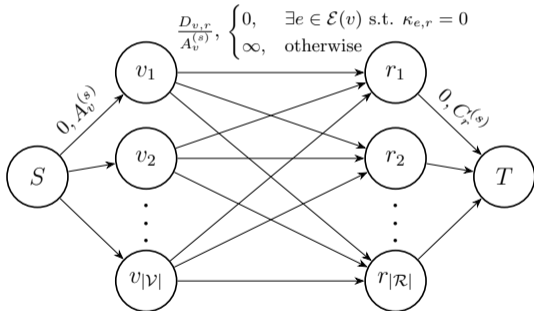
$$x_{v,r} = 0, \forall (v, r) \in \{\exists e \in \mathcal{E}(v) \text{ s.t. } \kappa_{e,r} = 0\}.$$

Clock Network Planning Algorithm

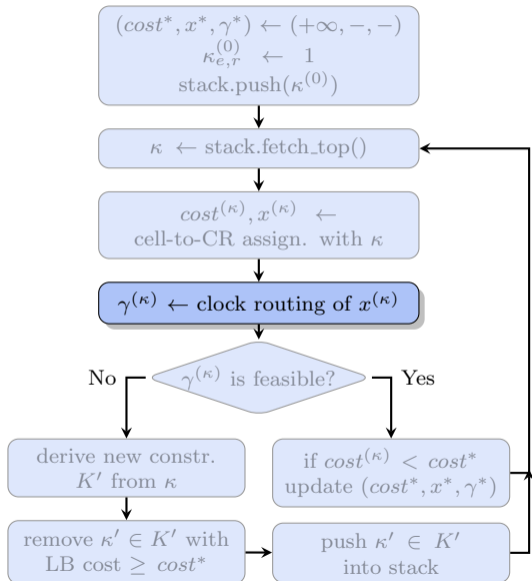


Cell-to-CR assignment problem

- Can be nearly optimally solved by a minimum-cost flow approximation

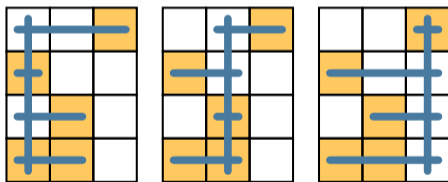


Clock Network Planning Algorithm

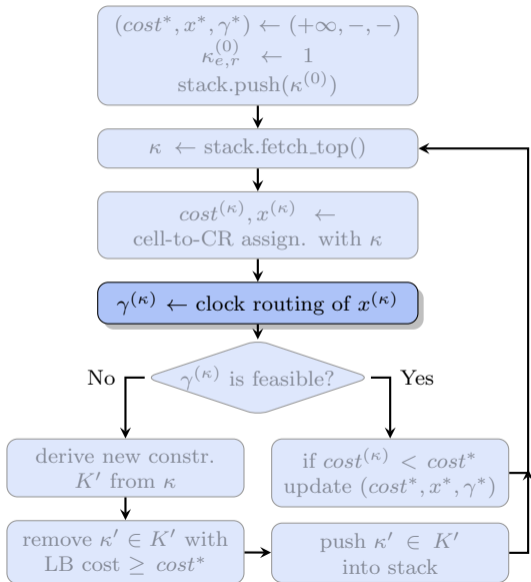


D-layer clock tree candidates generation

- ▶ Each D-layer clock is a vertical trunk tree
- ▶ There are m candidates for each clock on a CR grid with m columns
- ▶ Total of $m|\mathcal{E}|$ clock tree candidates



Clock Network Planning Algorithm

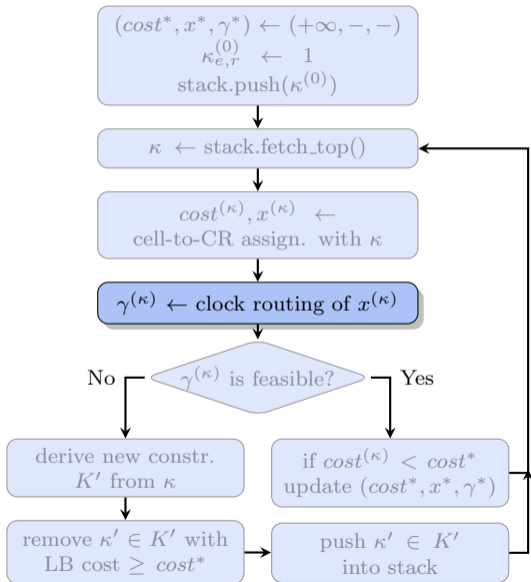


Clock tree candidate selection problem

- ▶ Minimize a topology-dependent cost
- ▶ Capacity constraints make the problem intractable
- ▶ Feasible solution may not exist

$$\begin{aligned} \min_x \quad & \sum_{t \in \mathcal{T}} \phi_t \cdot z_t, \\ \text{s.t.} \quad & z_t \in \{0, 1\}, \forall t \in \mathcal{T}, \\ & \sum_{t \in \mathcal{T}(e)} z_t = 1, \forall e \in \mathcal{E}, \\ & \sum_{t \in \mathcal{T}} H_{t,r} \cdot z_t \leq 24, \forall r \in \mathcal{R}, \\ & \sum_{t \in \mathcal{T}} V_{t,r} \cdot z_t \leq 24, \forall r \in \mathcal{R}. \end{aligned}$$

Clock Network Planning Algorithm

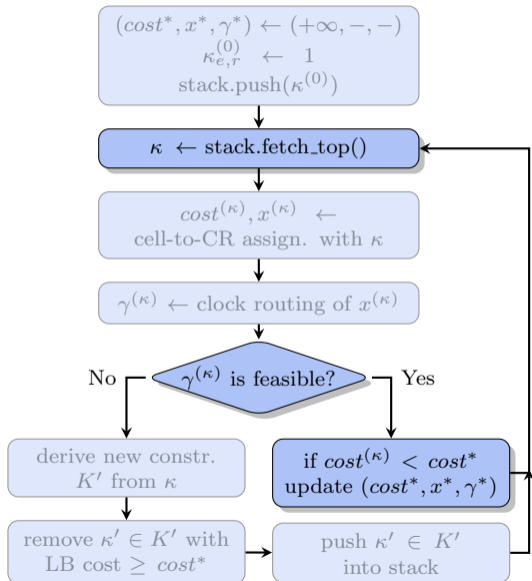


Lagrangian relaxation of the candidate selection problem

- ▶ Relax the capacity constraints and introduce Lagrangian multipliers λ
- ▶ Iteratively solve the relaxed problem and update λ until it converges

$$\begin{aligned} \min_x \quad & \sum_{t \in \mathcal{T}} (\phi_t + \lambda_t) \cdot z_t, \\ \text{s.t.} \quad & z_t \in \{0, 1\}, \forall t \in \mathcal{T}, \\ & \sum_{t \in \mathcal{T}(e)} z_t = 1, \forall e \in \mathcal{E}, \\ & \sum_{t \in \mathcal{T}} H_{t,r} \cdot z_t \leq 24, \forall r \in \mathcal{R}, \\ & \sum_{t \in \mathcal{T}} V_{t,r} \cdot z_t \leq 24, \forall r \in \mathcal{R}. \end{aligned}$$

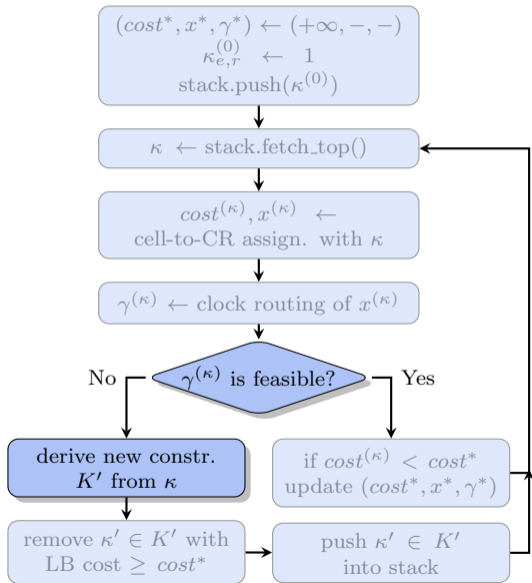
Clock Network Planning Algorithm



If clock routing $\gamma^{(\kappa)}$ is feasible

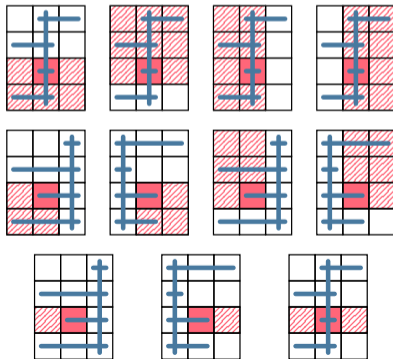
- ▶ Update the best solution if $cost^{(\kappa)}$ is better than the previous best $cost^*$
- ▶ Fetch the next κ in the stack to explore other subspaces (branches)

Clock Network Planning Algorithm

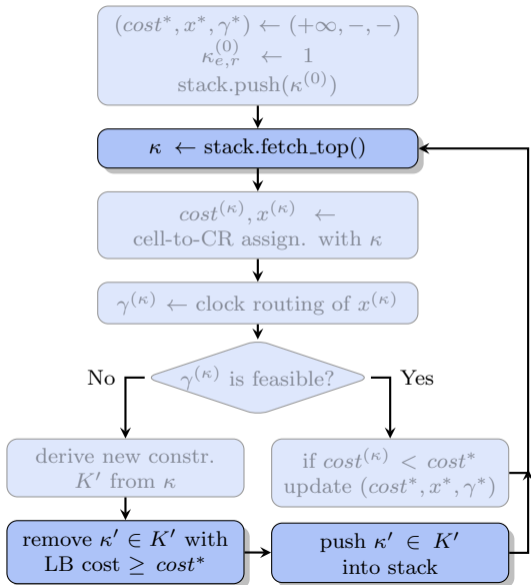


Derive new constraints $\kappa' \in K'$ from κ

- ▶ Each κ' is a subspace of κ
- ▶ Want $\kappa' \in K'$ can encourage more clock-friendly cell-to-CR assignment
- ▶ Forbid some cell-to-CR assign. that can potentially reduce clock overflow on top of κ



Clock Network Planning Algorithm



Remove suboptimal subspaces $\kappa' \in K'$

- We can safely prune subspaces with LB costs no better than $cost^*$

$$cost_{LB}(\kappa) = \sum_{v \in \mathcal{V}} \min_{\{r \in \mathcal{R} | \kappa_{e,r} = 1, \forall e \in \mathcal{E}(v)\}} D_{v,r}.$$

$$\min_x \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}} D_{v,r} \cdot x_{v,r},$$

$$\text{s.t. } x_{v,r} \in \{0, 1\}, \forall v \in \mathcal{V}, \forall r \in \mathcal{R},$$

$$\sum_{r \in \mathcal{R}} x_{v,r} = 1, \forall v \in \mathcal{V},$$

~~$$\sum_{v \in \mathcal{V}} A_v \cdot x_{v,r} \leq C_r, \forall r \in \mathcal{R},$$~~

$$x_{v,r} = 0, \forall (v, r) \in \{\exists e \in \mathcal{E}(v) \text{ s.t. } \kappa_{e,r} = 0\}.$$



Introduction

Proposed Algorithms

Experimental Results

Conclusion

Experimental Setup



Implemented in C++ on top of placement framework [Li+, TCAD'18]

Linux, Intel Core i9-7900X CPUs (3.30 GHz and 10 cores) and 128 GB RAM

Routed by Vivado v2016.4

ISPD 2017 contest benchmark

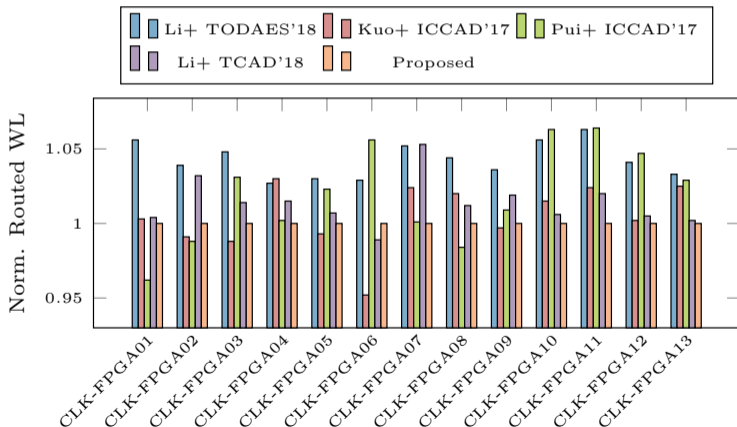
- ▶ Xilinx UltraScale architecture
- ▶ 0.5M - 1.0M cells
- ▶ 32 - 58 clock nets

Comparison with Other State-of-the-Art Placers



Achieved the best routed WL with feasible clock routing

- ▶ On average, outperforms [Li+, TODAES'18] / [Kuo+, ICCAD'17] / [Pui+, ICCAD'17] / [Li+, TCAD'18] by 4.3% / 0.5% / 2.0% / 1.4% in routed WL

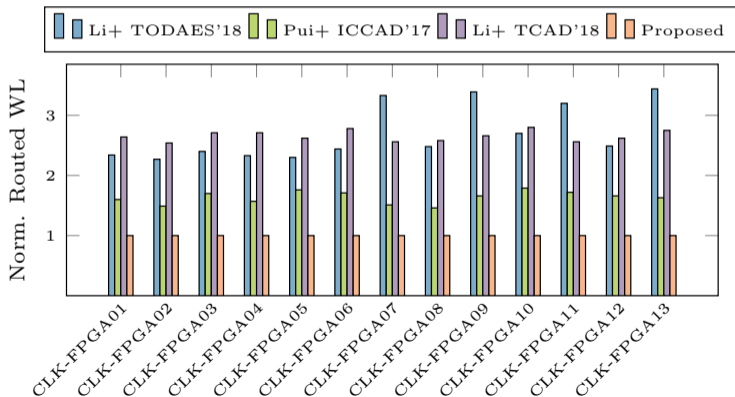


Comparison with Other State-of-the-Art Placers



Achieved the best runtime

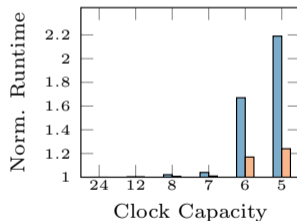
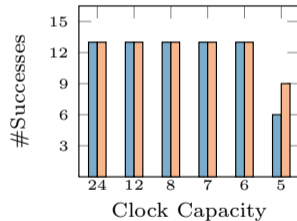
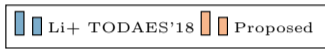
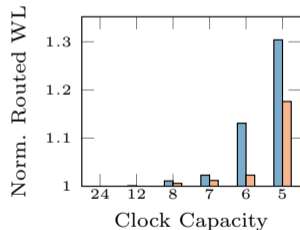
- ▶ On average, runs $2.70\times$ / $1.64\times$ / $2.66\times$ faster than [Li+, TODAES'18] / [Pui+, ICCAD'17] / [Li+, TCAD'18]
- ▶ [Kuo+, ICCAD'17] did not report placement runtime



Comparison Under Different Clock Capacities (CC)



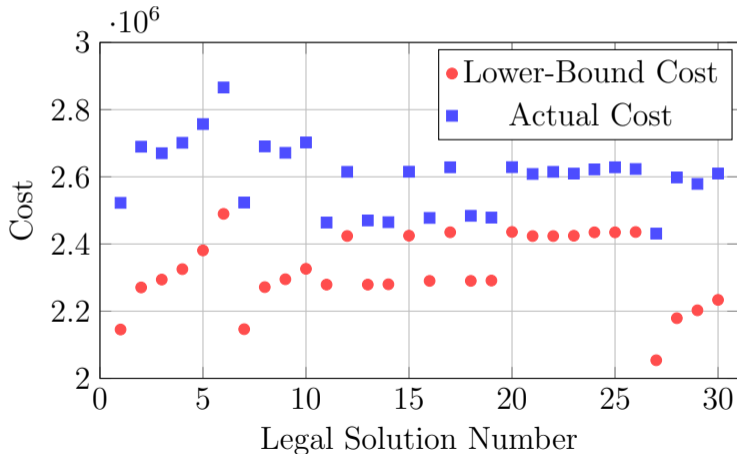
Apple-to-apple comparison with [Li+, TODAES'18] under different CC



Branch-and-Bound Tree Exploration



- ▶ First 30 feasible solutions found in clock network planning algorithm
- ▶ The best solution among them is achieved at #27, which is 4% better than #1





Introduction

Proposed Algorithms

Experimental Results

Conclusion



- ▶ A generic FPGA placement framework that simultaneously optimizes placement quality and ensures clock feasibility by explicit clock tree construction
- ▶ The proposed framework significantly reduces the placement quality degradation while honoring the clock feasibility for designs with high clock utilization
- ▶ A branch-and-bound-inspired clock network planning algorithm and a Lagrangian relaxation-based clock tree construction technique are proposed
- ▶ The proposed approach outperforms other state-of-the-art approaches in routed wirelength with competitive runtime

Thank You!