elfPlace: Electrostatics-based Placement for Large-Scale Heterogeneous FPGAs

Wuxi Li, Yibo Lin, and David Z. Pan

Department of Electrical & Computer Engineering
University of Texas at Austin

wuxi.li@utexas.edu
http://wuxili.net

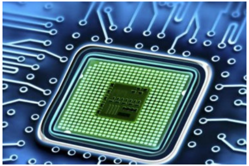
TEXAS
The University of Texas at Austin

UTDA

Outline

Outline

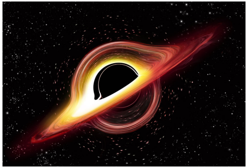
Modern FPGA Applications


Emulation and Prototyping


Deep Learning


Data Center


Industrial IoT


Scientific Computing


Cloud Computing

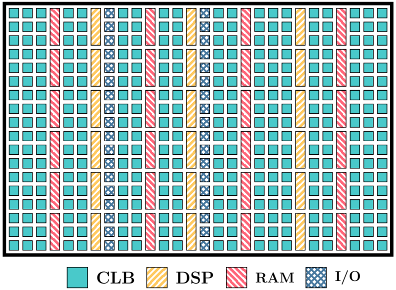

Automotive


Wireless Communication

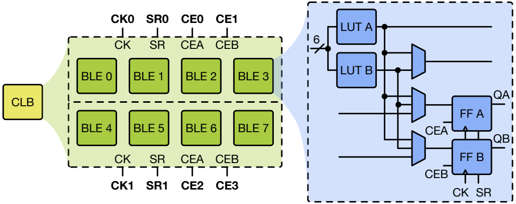

High-Frequency Trading

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	CLB clustering rules, . . .

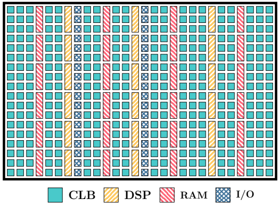


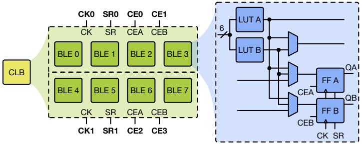
FPGA Layout



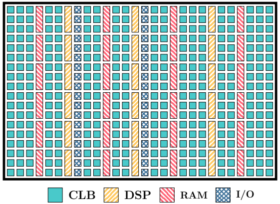
Configurable Logic Block (CLB) Architecture

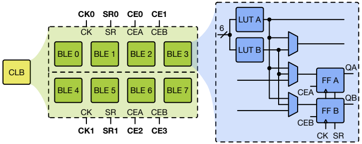
Modern FPGA Placement Challenges





Modern FPGA Placement Challenges

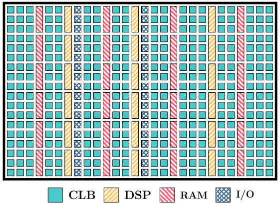


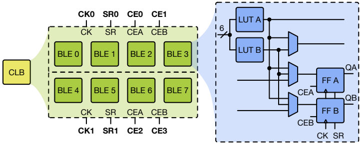


Highly Heterogeneous

- Convert heterogeneous netlists to homogeneous ones by clustering, [Betz, FPL'97]
- Homogeneous placement with heuristics, [Chen+, TCAD'18], [Abuowaimer+, TODAES'18]
- Handle a single cell type at a time, [Darav+, FPGA'19]

Modern FPGA Placement Challenges





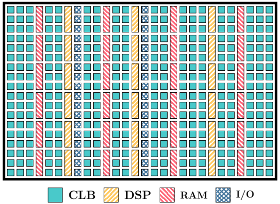
Highly Heterogeneous

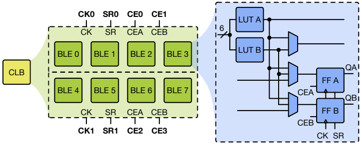
- Convert heterogeneous netlists to homogeneous ones by clustering, [Betz, FPL'97]
- Homogeneous placement with heuristics, [Chen+, TCAD'18], [Abuowaimer+, TODAES'18]
- Handle a single cell type at a time, [Darav+, FPGA'19]

Highly Discrete

- Handle highly discrete cells separately, [Li+, TCAD'18], [Chen+, TCAD'18]
- Add extra cost to objective functions, [Chen+, ICCAD'14], [Kuo+, ICCAD'17]

Modern FPGA Placement Challenges





Highly Heterogeneous

- Convert heterogeneous netlists to homogeneous ones by clustering, [Betz, FPL'97]
- Homogeneous placement with heuristics, [Chen+, TCAD'18], [Abuowaimer+, TODAES'18]
- Handle a single cell type at a time, [Darav+, FPGA'19]

Highly Discrete

- Handle highly discrete cells separately, [Li+, TCAD'18], [Chen+, TCAD'18]
- Add extra cost to objective functions, [Chen+, ICCAD'14], [Kuo+, ICCAD'17]

Downstream Clustering Dependent

- Adjust cell area based on a local clustering estimation, [Li+, TCAD'19]

Our Contributions

elfPlace: a general, flat, nonlinear placement algorithm for large-scale heterogeneous FPGAs

Our Contributions

elfPlace: a general, flat, nonlinear placement algorithm for large-scale heterogeneous FPGAs

- We extend the `ePlace` algorithm [Lu+, TCAD'15] for ASICs to deal with heterogeneity and discreteness issues in FPGAs in a unified manner

Our Contributions

elfPlace: a general, flat, nonlinear placement algorithm for large-scale heterogeneous FPGAs

- We extend the `ePlace` algorithm [Lu+, TCAD'15] for ASICs to deal with heterogeneity and discreteness issues in FPGAs in a unified manner
- We improve the numeric optimization formulation and preconditioning approahces in `ePlace` for FPGAs

Our Contributions

elfPlace: a general, flat, nonlinear placement algorithm for large-scale heterogeneous FPGAs

- We extend the `ePlace` algorithm [Lu+, TCAD'15] for ASICs to deal with heterogeneity and discreteness issues in FPGAs in a unified manner
- We improve the numeric optimization formulation and preconditioning approahces in `ePlace` for FPGAs
- We propose a normalized subgradient method to spread heterogeneous cell types in a self-adaptive manner.

Our Contributions

elfPlace: a general, flat, nonlinear placement algorithm for large-scale heterogeneous FPGAs

- We extend the `ePlace` algorithm [Lu+, TCAD'15] for ASICs to deal with heterogeneity and discreteness issues in FPGAs in a unified manner
- We improve the numeric optimization formulation and preconditioning approahces in `ePlace` for FPGAs
- We propose a normalized subgradient method to spread heterogeneous cell types in a self-adaptive manner.
- We improve the clustering-aware area adjustment technique in [Li+, TCAD'19] and integrate it, together with routability and pin density optimizations.

Our Contributions

elfPlace: a general, flat, nonlinear placement algorithm for large-scale heterogeneous FPGAs

- We extend the `ePlace` algorithm [Lu+, TCAD'15] for ASICs to deal with heterogeneity and discreteness issues in FPGAs in a unified manner
- We improve the numeric optimization formulation and preconditioning approahces in `ePlace` for FPGAs
- We propose a normalized subgradient method to spread heterogeneous cell types in a self-adaptive manner.
- We improve the clustering-aware area adjustment technique in [Li+, TCAD'19] and integrate it, together with routability and pin density optimizations.
- We demonstrate more than 7% improvement in routed wirelength, on ISPD 2016 benchmark suite, over 4 cutting-edge placers with very competitive runtime.

Outline

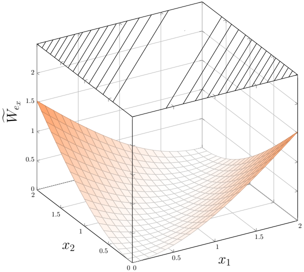
Weighted-Average (WA) Wirelength Model

Weighted-average (WA) wirelength model approximates half-perimeter wirelength (HPWL),

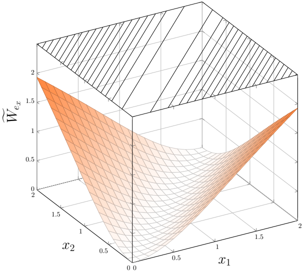
$$W(\boldsymbol{x},\boldsymbol{y}) = \sum_{e\in\mathcal{E}} W_e(\boldsymbol{x},\boldsymbol{y}) = \sum_{e\in\mathcal{E}} \left(\max_{i,j\in e} |x_i - x_j| + \max_{i,j\in e} |y_i - y_j| \right),$$

using soft min/max functions,

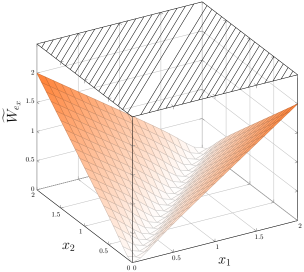
$$\widetilde{W}_{e_x}(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_{i\in e} x_i \exp(x_i/\gamma)}{\sum_{i\in e} \exp(x_i/\gamma)} - \frac{\sum_{i\in e} x_i \exp(-x_i/\gamma)}{\sum_{i\in e} \exp(-x_i/\gamma)}.$$



$\gamma = 1.0$



$\gamma = 0.5$

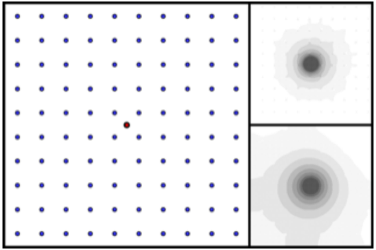


$\gamma = 0.1$

Electrostatics-Based Density Model

Analogy between placement and electrostatic system, ePlace [Lu+,TCAD'15]

Placement	Electrostatic System	
Cell	Positive Charge	q
Site	Fixed Negative Charge	
Utilization	Charge Density	ρ
Utilization Penalty	Electric Potential Energy	Φ
Utilization Gradient	Electric Field	ξ



Electrostatics-Based Density Model

Poisson's equation of the electrostatic system

$$\begin{cases} \nabla \cdot \nabla \psi(x,y) = -\rho(x,y),\ (x,y) \in R, \\ \hat{\mathbf{n}} \cdot \nabla \psi(x,y) = \mathbf{0},\ (x,y) \in \partial R, \\ \iint\limits_R \rho(x,y) = \iint\limits_R \psi(x,y) = 0,\ (x,y) \in R. \end{cases}$$

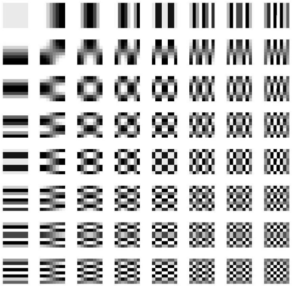
The numerical solution using spectral method

$$a_{u,v} = \frac{1}{m^2} \sum_{x=0}^{m-1} \sum_{y=0}^{m-1} \rho(x,y) \cos(\omega_u x) \cos(\omega_v y),$$

$$\psi(x,y) = \sum_{u=0}^{m-1} \sum_{v=0}^{m-1} \frac{a_{u,v}}{\omega_u^2 + \omega_v^2} \cos(\omega_u x) \cos(\omega_v y),$$

$$\xi_x(x,y) = \sum_{u=0}^{m-1} \sum_{v=0}^{m-1} \frac{a_{u,v}\omega_u}{\omega_u^2 + \omega_v^2} \sin(\omega_u x) \cos(\omega_v y),$$

$$\xi_y(x,y) = \sum_{u=0}^{m-1} \sum_{v=0}^{m-1} \frac{a_{u,v}\omega_v}{\omega_u^2 + \omega_v^2} \cos(\omega_u x) \sin(\omega_v y).$$



8 × 8 2D-DCT

Outline

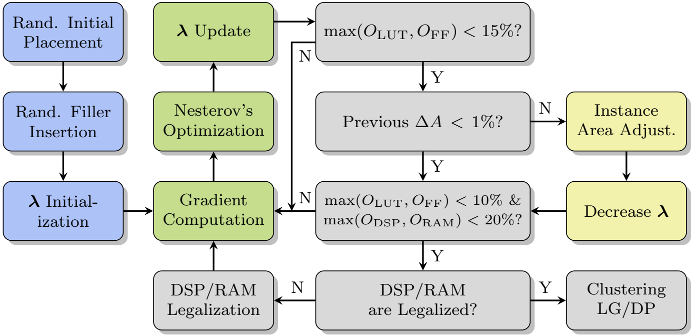
Problem Formulation

Each resource type has a separate electrostatic system

$$\min_{\boldsymbol{x},\boldsymbol{y}} \ \widetilde{W}(\boldsymbol{x},\boldsymbol{y}) \ \text{ s.t. } \Phi_s(\boldsymbol{x},\boldsymbol{y}) = 0, \forall s \in \mathcal{S} = \{\text{LUT, FF, DSP, RAM}\}.$$

Relax the constraints using Augmented Lagrangian Method (ALM)

$$\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x},\boldsymbol{y}) = \widetilde{W}(\boldsymbol{x},\boldsymbol{y}) + \sum_{s\in\mathcal{S}} \lambda_s \left(\Phi_s(\boldsymbol{x},\boldsymbol{y}) + \frac{c_s}{2}\Phi_s(\boldsymbol{x},\boldsymbol{y})^2\right).$$



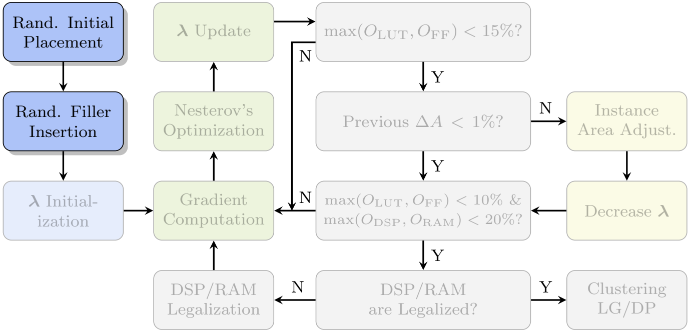
Initial Placement

Randomly place physical instances by

$$(X, Y) \sim \mathcal{N}\left(\frac{1}{2}\begin{bmatrix} W_R \\ H_R \end{bmatrix}, 10^{-3}\begin{bmatrix} W_R & 0 \\ 0 & H_R \end{bmatrix}\right)$$

Create fillers to achieve charge neutrality and randomly place them by

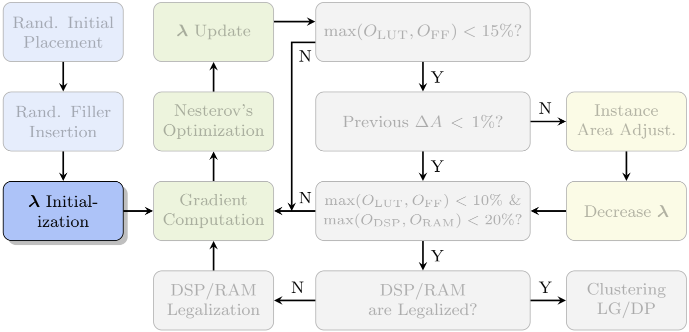
$$(X, Y) \sim \text{Resource Distribution}$$



Density Weight Initialization

Initialize λ based on the wirelength and energy gradient norm ratio

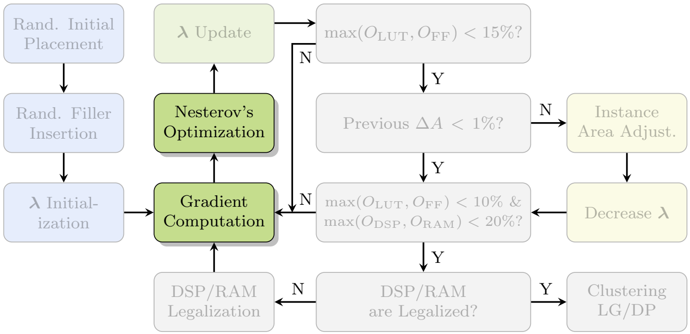
$$\boldsymbol{\lambda}^{(0)} = \eta \frac{\|\nabla \widetilde{W}^{(0)}\|_1}{\sum_{s\in\mathcal{S}} \|\nabla \Phi_s^{(0)}\|_1} \left(1, 1, \cdots, 1\right)^T.$$



Gradient Computation

Gradient ∇f of the ALM-based formulation

$$\begin{aligned}\frac{\partial f}{\partial x_i} &= \frac{\partial \widetilde{W}}{\partial x_i} + \lambda_s\Big(\frac{\partial \Phi_s}{\partial x_i} + c_s\Phi_s\frac{\partial \Phi_s}{\partial x_i}\Big)\\ &= \frac{\partial \widetilde{W}}{\partial x_i} - \lambda_s q_i \xi_{x_i}\Big(1 + c_s\Phi_s\Big), \quad \forall i \in \mathcal{V}_s.\end{aligned}$$

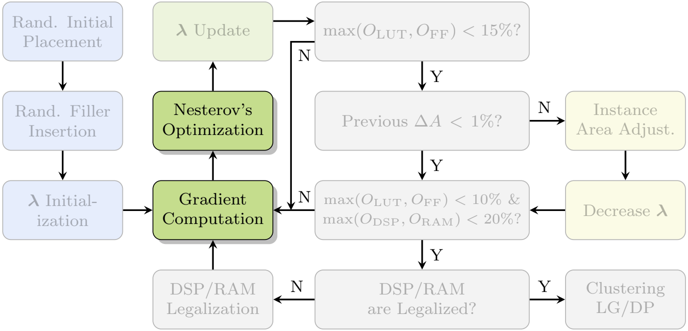


Preconditioning

Precondition ∇f by $\boldsymbol{H}_f^{-1}\nabla f$, where $\boldsymbol{H}_f$ is a diagonal matrix with each diagonal entry defined as

$$\frac{\partial^2 \widetilde{W}}{\partial x_i^2} \sim h_{x_i} = \max\left(\sum_{e\in\mathcal{E}_i}\frac{1}{|e|-1} + \lambda_s q_i,\ 1\right), \forall i \in \mathcal{V}_s, \forall s \in \mathcal{S}.$$

Update placement along $-\boldsymbol{H}_f^{-1}\nabla f$ by Nesterov's method, ePlace [Lu+, TCAD'15]

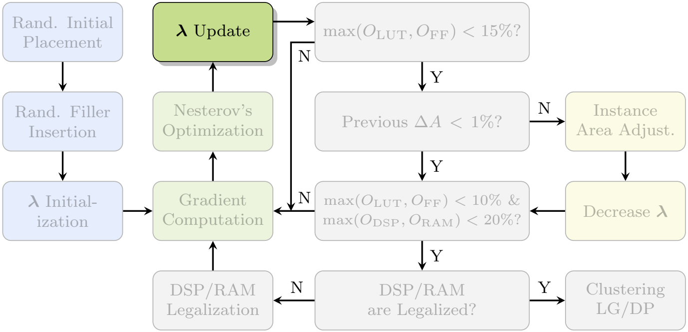


Density Weight Updating

Update λ using normalized subgradient method

$$\widehat{\nabla}_{\text{sub}} \boldsymbol{\lambda}^{(k)} = \left(\cdots, \frac{1}{\Phi_s^{(0)}} \left(\Phi_s^{(k)} + \frac{c_s}{2} {\Phi_s^{(k)}}^2 \right), \cdots \right)^T.$$

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + t^{(k)} \frac{\widehat{\nabla}_{\text{sub}} \boldsymbol{\lambda}^{(k)}}{\|\widehat{\nabla}_{\text{sub}} \boldsymbol{\lambda}^{(k)}\|_2}.$$

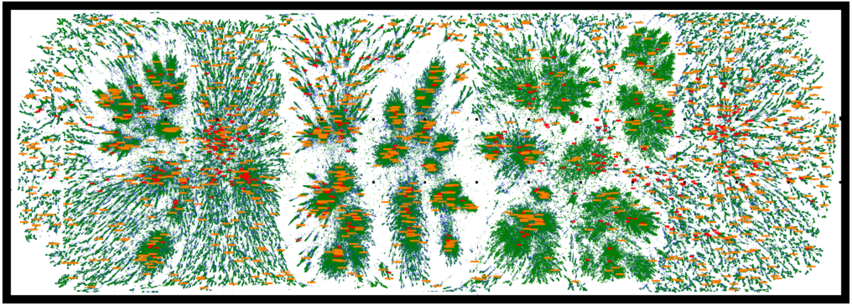


Density Weight Updating

Update λ using normalized subgradient method

$$\widehat{\nabla}_{\text{sub}}\boldsymbol{\lambda}^{(k)} = \left(\cdots, \frac{1}{\Phi_s^{(0)}}\left(\Phi_s^{(k)} + \frac{c_s}{2}\Phi_s^{(k)\,2}\right), \cdots\right)^T.$$

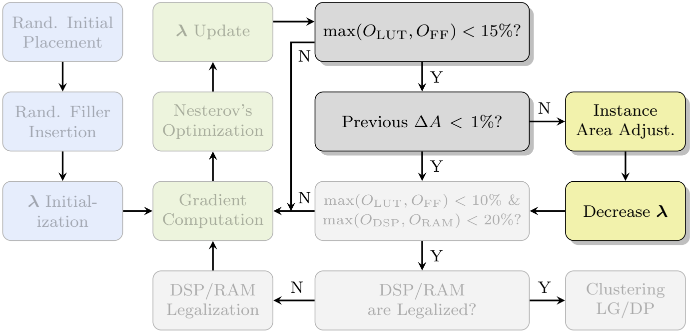
$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + t^{(k)} \frac{\widehat{\nabla}_{\text{sub}}\boldsymbol{\lambda}^{(k)}}{\|\widehat{\nabla}_{\text{sub}}\boldsymbol{\lambda}^{(k)}\|_2}.$$



Density Weight Redirecting

- Adjust instance areas to optimize routability, pin density, and clustering compatibility
- Redirect λ to adapt the perturbation

$$\boldsymbol{\lambda}' = \eta' \frac{\|\nabla \widetilde{W}\|_1}{\langle(\cdots, \|\nabla \Phi_s\|_1, \cdots)^T, \widehat{\nabla}_{\text{sub}}\boldsymbol{\lambda}\rangle} \widehat{\nabla}_{\text{sub}}\boldsymbol{\lambda},$$

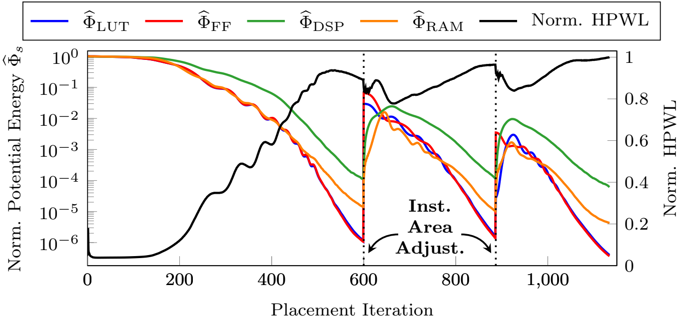


Density Weight Redirecting

Adjust instance areas to optimize routability, pin density, and clustering compatibility

Redirect λ to adapt the perturbation

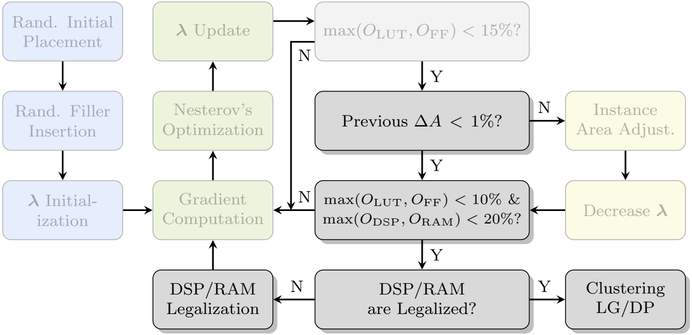
$$\boldsymbol{\lambda}' = \eta' \frac{\|\nabla \widetilde{W}\|_1}{\langle(\cdots, \|\nabla \Phi_s\|_1, \cdots)^T, \widehat{\nabla}_{\text{sub}}\boldsymbol{\lambda}\rangle} \widehat{\nabla}_{\text{sub}}\boldsymbol{\lambda},$$



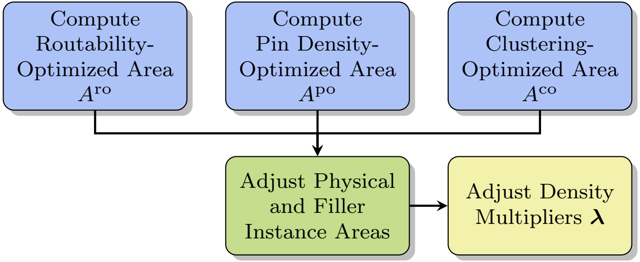
Post-GP Placement

Legalize highly-discrete and large DSP and RAM blocks

Finish the flow by clustering, legalization, and detailed placement



Area Adjustment Scheme



- Non-filler cells: $A_i = \max(A_i^{\text{ro}}, A_i^{\text{po}}, A_i^{\text{co}}, A_i), \forall i \in \mathcal{V}$
- Filler cells: Reduce areas to maintain electrostatic neutrality
- Traditional cell inflation-based routability and pin density optimization

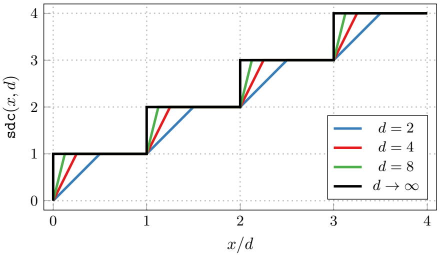
Clustering Compatibility-Optimized Area

Areas of LUTs and FFs are clustering-dependent

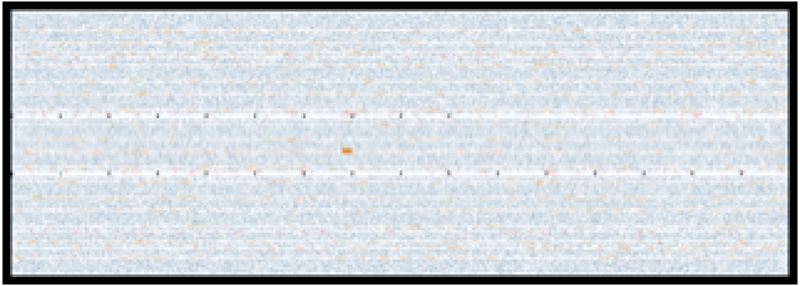
`UTPlaceF-DL` [Li+, TCAD'19] adjusts areas based on a local clustering estimation

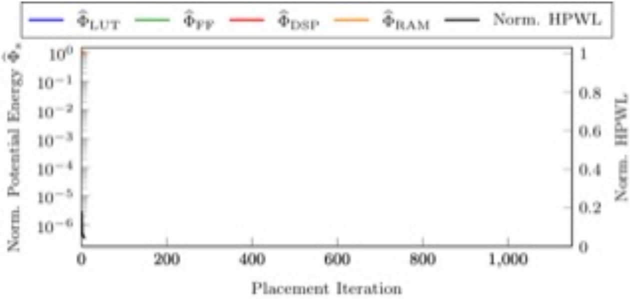
`elfPlace` improves `UTPlaceF-DL`'s approaches using various smoothing techniques

- `UTPlaceF-DL`: $A_i^{\text{co}} = \frac{1}{2n_{i,\theta_i}} \frac{\lceil n_{i,\theta_i}/4 \rceil}{\sum_{\theta \in \Theta_i} \lceil n_{i,\theta}/4 \rceil} \left\lceil \frac{\sum_{\theta \in \Theta_i} \lceil n_{i,\theta}/4 \rceil}{2} \right\rceil, \forall i \in \mathcal{V}$
- `elfPlace`: $A_i^{\text{co}} = \frac{1}{2\mathbb{E}_{i,\theta_i}} \frac{\mathtt{sdc}(\mathbb{E}_{i,\theta_i}, 4)}{\sum_{\theta \in \Theta_i} \mathtt{sdc}(\mathbb{E}_{i,\theta}, 4)} \, \mathtt{sdc}\Big(\sum_{\theta \in \Theta_i} \mathtt{sdc}(\mathbb{E}_{i,\theta}, 4),\ 2 \Big), \forall i \in \mathcal{V}$



elfPlace Animation





Outline

Introduction

Preliminaries

elfPlace Algorithm

Experimental Results

Conclusion and Future Work

Experimental Setup

Machine

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

ISPD 2016 contest benchmark suite

- Released by Xilinx
- 0.1M - 1.1M cells

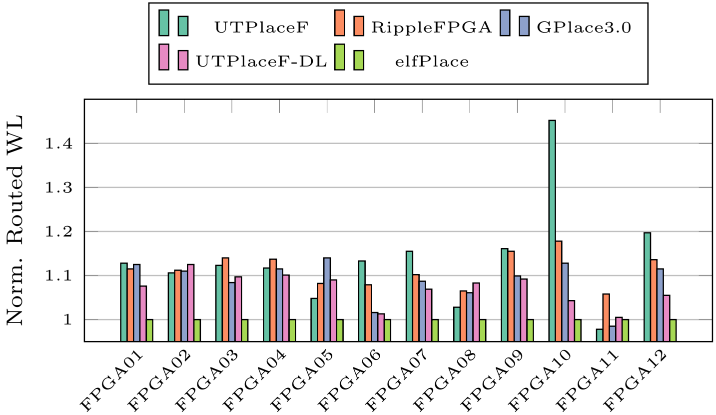
Placers for comparison

- `UTPlaceF` [Li+, TCAD'18]
- `RippleFPGA` [Chen+, TCAD'18]
- `GPlace3.0` [Abuowaimer+, TODAES'18]
- `UTPlaceF-DL` [Li+, TCAD'19]
- `elfPlace`

Routed Wirelength Comparison

`elfPlace` significantly outperforms other placers in routed wirelength

- 13.6% better than `UTPlaceF`
- 11.3% better than `RippleFPGA`
- 8.9% better than `GPlace3.0`
- 7.1% better than `UTPlaceF-DL`



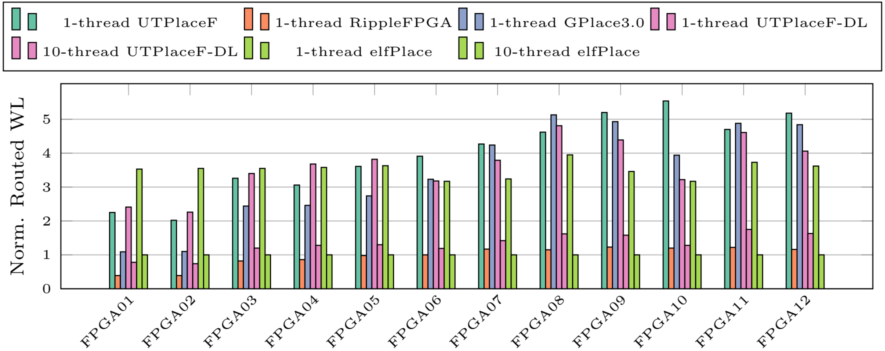
Runtime Comparison

1-thread elfPlace is

- 1.13× faster than 1-thread UTPlaceF
- 3.65× slower than 1-thread RippleFPGA
- 1.03× slower than 1-thread GPlace3.0
- 1.03× faster than 1-thread UTPlaceF-DL

10-thread elfPlace is

- 3.51× faster than 1-thread elfPlace
- 1.31× faster than 10-thread UTPlaceF-DL



Individual Technique Validation

w/ `ePlace`'s Multiplier Method

- +1.2% routed wirelength
- +1.0% runtime

w/o preconditioning

- 11 out of 12 designs fail to converge

w/ `ePlace`'s preconditioning

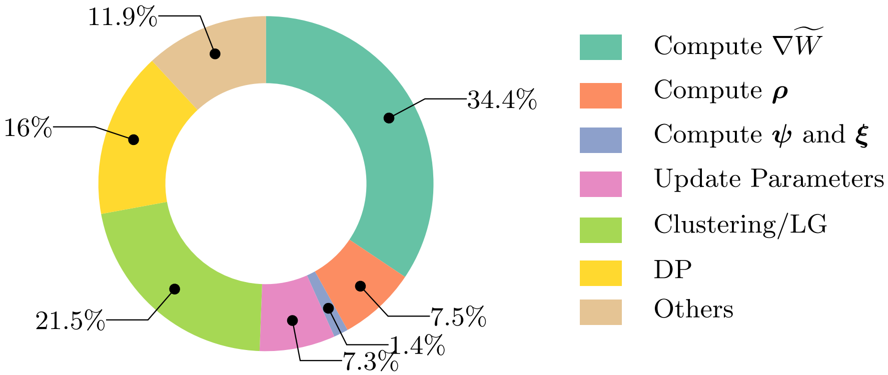
- 2 out of 12 designs fail to converge
- +0.1% routed wirelength
- +3.0% runtime

w/o Gaussian smoothing and `sdc` function

- Same routed wirelength
- +15.0% runtime

Runtime Breakdown

Based on `FPGA-12` (1.1M cells) using 10 threads



Outline

Conclusion and Future Work

Conclusion

- **elfPlace**: a general, flat, nonlinear placement algorithm for large-scale heterogeneous FPGAs
- We extend the `ePlace` algorithm [Lu+, TCAD'15] for ASICs to deal with heterogeneity and discreteness issues in FPGAs in a unified manner
- We improve the numeric optimization formulation and preconditioning approahces in `ePlace` for FPGAs
- We propose a normalized subgradient method to spread heterogeneous cell types in a self-adaptive manner.
- We improve the clustering-aware area adjustment technique in [Li+, TCAD'19] and integrate it, together with routability and pin density optimizations.
- We demonstrate more than 7% improvement in routed wirelength, on ISPD 2016 benchmark suite, over 4 cutting-edge placers with very competitive runtime.

Future Work

- Other optimization algorithms
- Timing-driven placement
- FPGA/GPU acceleration

Thank You!