# GPU Acceleration in Physical Synthesis

*Evangeline Young, The Chinese University of Hong Kong*

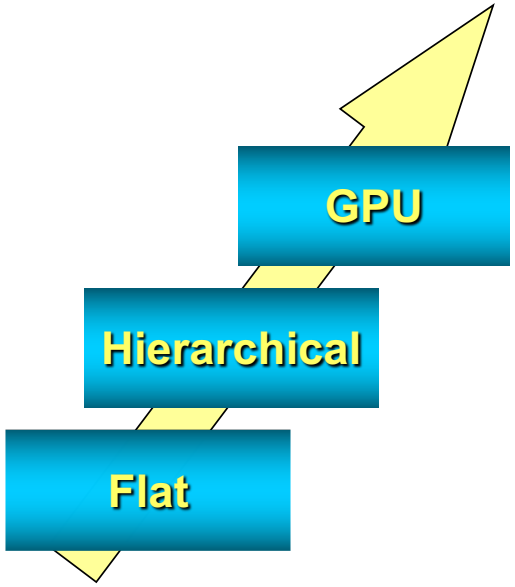
March 28, 2023

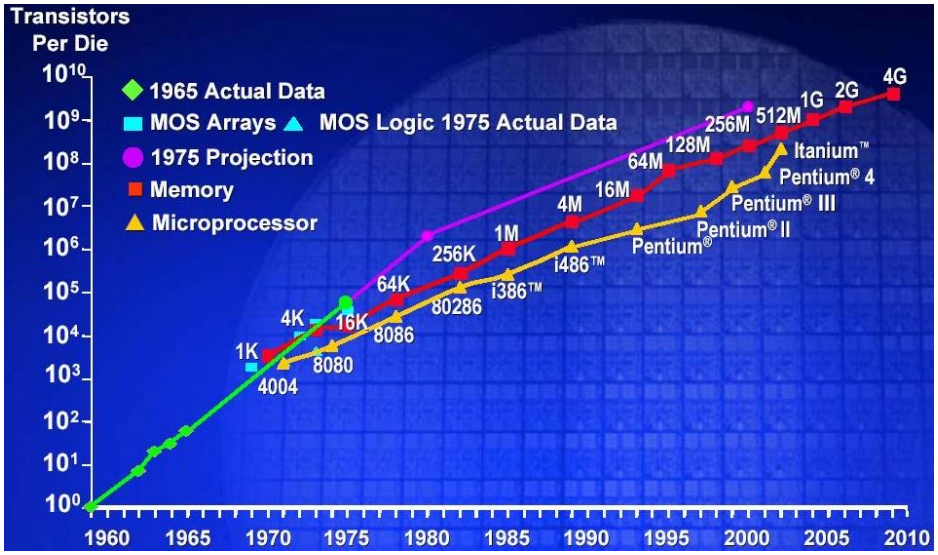
# Agenda

- GPU Acceleration on:
  - Placement
  - Routing
  - Routability-driven Placement

# Framework Evolution

- Billions of transistors fabricated in a single chip.
- Need frameworks for very large-scale designs.
- Framework evolution for EDA tools: Flat ➔ Hierarchical ➔ GPU Accelerated



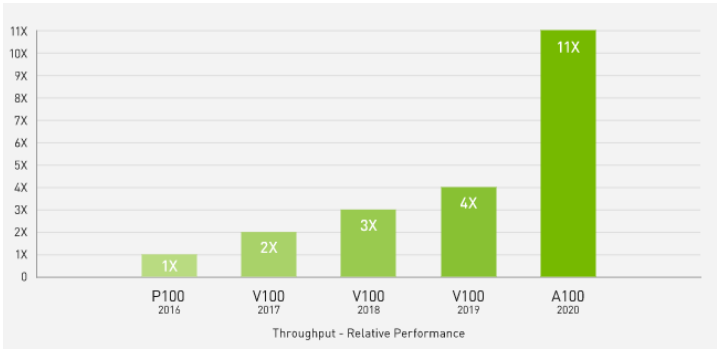

Source: Intel (ISSCC-03)

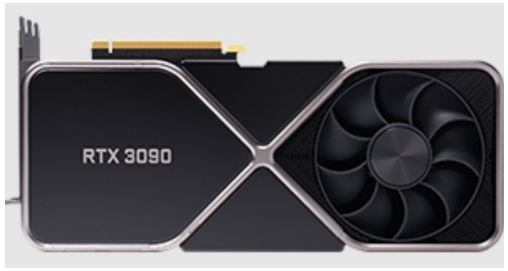
# Agenda

- GPU Acceleration on:
  - Placement
  - Routing
  - Routability-driven Placement

# GPU-accelerated Global Placers

- Rapid development of GPU's computational power. GPU acceleration becomes an important direction
- Recently, DREAMPlace[1]:
  - Implemented the approach of ePlace[2] on GPU
  - Produced the SOTA solution quality and performance
- **Xplace** further improve on DREAMPlace's performance.




[1] Y. Lin, Z. Jiang, J. Gu, W. Li, S. Dhar, H. Ren, B. Khailany, and D. Z. Pan, "DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement," IEEE TCAD 2020
[2] J. Lu, H. Zhuang, P. Chen, H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. Huang, Y. Luo, C.-C. Teng, et al., "ePlace-MS: Electrostatics-based placement for mixed-size circuits," IEEE TCAD 2015

# ePlace Methodology

- Cell density is modeled as an electrostatic system (Poisson's Equation):

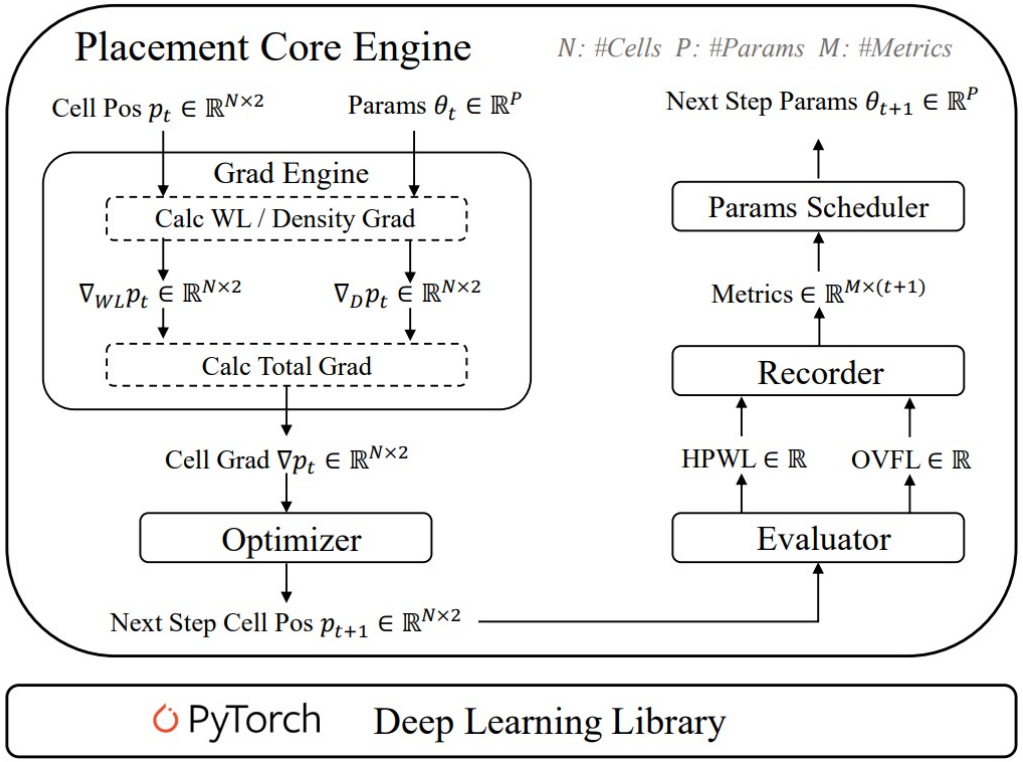
$$\begin{cases} \nabla \cdot \nabla\psi(x,y) = -\rho(x,y), \\ \hat{\mathbf{n}} \cdot \nabla\psi(x,y) = 0, (x,y) \in \partial R, \\ \iint_R \rho(x,y) = \iint_R \psi(x,y) = 0, \end{cases}$$

- $\partial R$ is the boundary
- $\rho(x,y)$ is the electron distribution
- $\psi(x,y)$ is the potential distribution
- $\nabla\psi(x,y)$ is the Electric Field

- Electron Distribution $\rho \rightarrow$ 2D Density map $D$ of placement
- Electric Field $\nabla\psi_x, \nabla\psi_y \rightarrow$ moving force on *x* and *y*-axis
- Solve this PDE problem by **Discrete Cosine Transformation (DCT)**

J. Lu, H. Zhuang, P. Chen, H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. Huang, Y. Luo, C.-C. Teng, et al., "ePlace-MS: Electrostatics-based placement for mixed-size circuits," IEEE TCAD 2015

# Xplace Framework




L. Liu, B. Fu, Martin D.F. Wong and Evangeline F.Y. Young, "Xplace: an Extremely Fast and Extensible Global Placement Framework" DAC 2022: 1309-1314.

# Operator-Level Optimization in Xplace

- Wirelength Operator Combination (OC):
  - **Observation**: Both the HPWL function and the stable WA wirelength function need the min and max cell positions in a net.
  - **Method**: combining the three operators with heavy wirelength-related workload, ***WA wirelength***, ***WA gradient*** and ***HPWL***, into one operator to avoid redundant computation

$$HPWL_e(p) = \left(\max_{i \in e} x_i - \min_{i \in e} x_i\right) + \left(\max_{i \in e} y_i - \min_{i \in e} y_i\right)$$

HPWL function

$$WL_e(x) = \frac{\sum_{i \in e} x_i e^{\frac{x_i - \max_{j \in e} x_j}{\gamma}}}{\sum_{i \in e} e^{\frac{x_i - \max_{j \in e} x_j}{\gamma}}} - \frac{\sum_{i \in e} x_i e^{\frac{\min_{j \in e} x_j - x_i}{\gamma}}}{\sum_{i \in e} e^{\frac{\min_{j \in e} x_j - x_i}{\gamma}}}$$

$$WL_e(y) = \frac{\sum_{i \in e} y_i e^{\frac{y_i - \max_{j \in e} y_j}{\gamma}}}{\sum_{i \in e} e^{\frac{y_i - \max_{j \in e} y_j}{\gamma}}} - \frac{\sum_{i \in e} y_i e^{\frac{\min_{j \in e} y_j - y_i}{\gamma}}}{\sum_{i \in e} e^{\frac{\min_{j \in e} y_j - y_i}{\gamma}}}$$

Stable WA wirelength

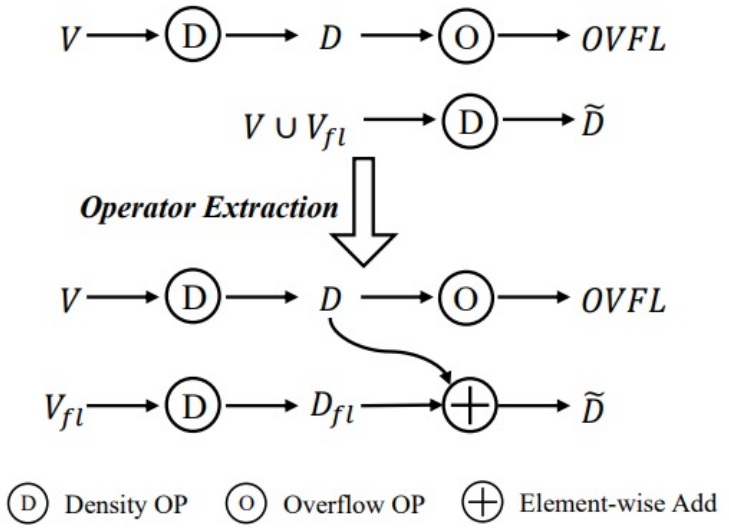
# Operator-Level Optimization in Xplace

- Density Operator Extraction (OE)
  - **Observation**: Both the calculation of $OVFL$ and total density map $\widetilde{D}$ need the cell density map $D$.

$$OVFL = \frac{\sum_{b \in B} \max(D_b - D_t, 0) A_b}{\sum_{i \in V_{mov}} A_i}$$

$$\tilde{D} = D + D_{fl}$$

  - **Method**: common sub-operator $D$ extraction, compute the cell density map $D$ and the filler density map $D_{fl}$ separately



# Operator-Level Optimization in Xplace

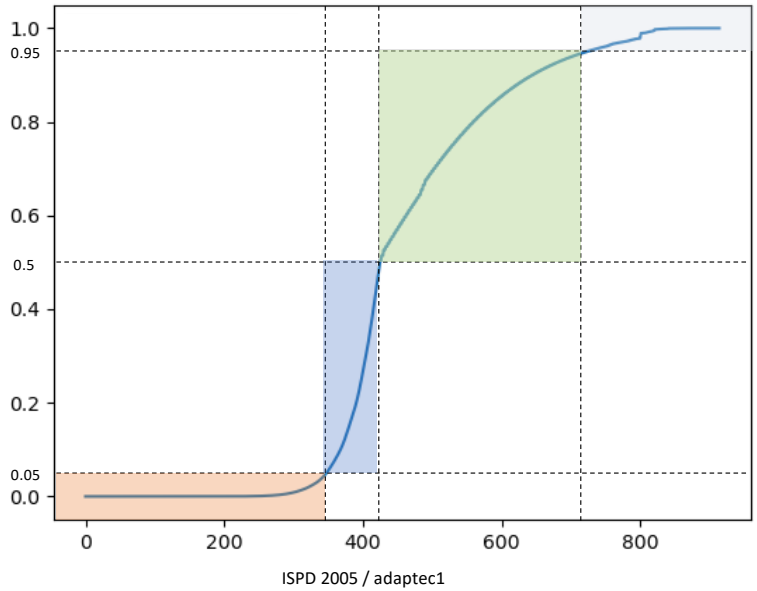
- Operator Reduction (OR)
  - Avoid invoking the heavy **autograd engine.** Directly derive the numerical solutions of the WL / density grad
  - Use in-place operators as much as possible to avoid redundant copying
  - Reorder the operators that need sync to the end of the execution queue to reduce the frequency of interrupting the GPU pipeline
  - Skip some density grad calculation in early placement stage as their values are very small at that stage. (OS)

# Placement-Stage-Aware Parameters Scheduling

- Precondition weighted ratio $\omega = \frac{\lambda|\mathbf{H_D}|}{|\mathbf{H_W}|+\lambda|\mathbf{H_D}|} \in [0,1]$
- To fully exploit the optimization space

**Algorithm 1** Placement-Stage-Aware Parameters Scheduling

$\gamma \leftarrow \gamma_0$ ▷ wirelength coefficient
$\lambda \leftarrow \lambda_0$ ▷ density weight
**while** $iteration$ < ITER and NOT Convergence **do**
  **if** $0.5 < \omega < 0.95$ and $iteration\%3 \neq 0$ **then**
    SKIP_UPDATE
  **else**
    $\gamma \leftarrow \gamma \times coef(overflow)$
    $\lambda \leftarrow \lambda \times \mu(\Delta hpwl)$ ▷ Both $\gamma$ and $\lambda$ are derived from [10]
  $\omega \leftarrow \frac{\lambda|\mathbf{H_D}|}{|\mathbf{H_W}|+\lambda|\mathbf{H_D}|}$



ISPD 2005 / adaptec1

# Comparisons

- Validation on Contest Benchmarks

| Benchmarks | DREAMPlace[1] | | | Xplace | | |
|---|---|---|---|---|---|---|
| | HPWL | GP/s | DP/s | HPWL | GP/s | DP/s |
| adaptec1 | 72.89 | 4.15 | 34.9 | 72.93 | 1.35 | 35.8 |
| adaptec2 | 81.84 | 3.73 | 46.2 | 81.04 | 1.58 | 45.4 |
| adaptec3 | 191.68 | 4.54 | 88.1 | 190.94 | 2.38 | 89.6 |
| adaptec4 | 173.45 | 4.90 | 95.4 | 172.41 | 2.85 | 96.1 |
| bigblue1 | 89.39 | 4.03 | 42.3 | 89.12 | 1.47 | 42.1 |
| bigblue2 | 136.57 | 4.68 | 129.3 | 136.56 | 2.41 | 127.2 |
| bigblue3 | 302.58 | 8.05 | 207.9 | 301.36 | 5.49 | 209.8 |
| bigblue4 | 742.95 | 13.38 | 459.7 | 741.18 | 11.65 | 463.1 |
| Sum | 1791.36 | 47.46 | 1103.6 | 1785.6 | 29.18 | 1109.0 |
| Ratio | 1.003 | 1.626 | 0.995 | 1.000 | 1.000 | 1.000 |

ISPD 2005

| Benchmarks | DREAMPlace [14] | | | | Xplace | | | |
|---|---|---|---|---|---|---|---|---|
| | HPWL | OVFL-5 | GP/s | DP/s | HPWL | OVFL-5 | GP/s | DP/s |
| des_perf_1 | 1107.5 | 65.28 | 3.71 | 1.31 | **1106.7** | 64.35 | 1.14 | 1.23 |
| fft_1 | 411.7 | 56.19 | 3.59 | 0.67 | **411.3** | 56.34 | 1.17 | 0.61 |
| fft_2 | **374.0** | 47.72 | 4.28 | 0.69 | 374.3 | 47.49 | 1.18 | 0.64 |
| fft_a | 627.6 | 35.12 | 3.60 | 0.61 | **625.6** | 34.7 | 1.29 | 0.70 |
| fft_b | **845.7** | 51.82 | 3.57 | 0.74 | 846.2 | 52.02 | 1.28 | 0.73 |
| matrix_mult_1 | 2129.2 | 81.02 | 3.71 | 1.61 | **2116.4** | 81.69 | 1.29 | 1.44 |
| matrix_mult_2 | 2163.3 | 77.61 | 3.97 | 1.63 | **2152.9** | 77.95 | 1.23 | 1.48 |
| matrix_mult_a | 3036.8 | 48.10 | 4.04 | 2.79 | **3031.7** | 48.34 | 1.29 | 3.68 |
| superblue12 | 25803.0 | 92.45 | 8.91 | 16.37 | **25783.8** | 93.18 | 4.64 | 17.29 |
| superblue14 | **23015.5** | 63.56 | 4.63 | 11.49 | 23017.1 | 64.34 | 1.60 | 13.74 |
| superblue19 | 15633.1 | 61.82 | 4.56 | 8.26 | **15544.1** | 62.39 | 1.46 | 6.60 |
| des_perf_a† | 2020.5 | 53.27 | 3.66 | 2.04 | **1998.6** | 52.32 | 1.18 | 1.67 |
| des_perf_b† | **1610.3** | 54.65 | 3.66 | 1.70 | 1612.6 | 53.64 | 1.27 | 1.58 |
| edit_dist_a† | 4217.9 | 80.30 | 3.97 | 2.31 | **4198.7** | 80.10 | 1.45 | 2.13 |
| matrix_mult_b† | 2786.7 | 44.86 | 3.82 | 1.98 | **2765.7** | 44.98 | 1.29 | 1.89 |
| matrix_mult_c† | **2672.9** | 42.13 | 4.07 | 2.07 | 2675.2 | 42.20 | 1.29 | 1.89 |
| pci_bridge32_a† | 361.8 | 30.55 | 3.54 | 0.82 | **356.0** | 30.36 | 1.08 | 0.69 |
| pci_bridge32_b† | 741.1 | 22.89 | 6.77 | 1.04 | **714.2** | 22.75 | 1.12 | 1.04 |
| superblue11_a† | **33411.2** | 54.51 | 5.59 | 13.33 | 33528.3 | 54.78 | 2.87 | 12.73 |
| superblue16_a† | 25600.9 | 65.85 | 4.38 | 10.60 | **25505.1** | 65.85 | 1.91 | 11.08 |
| Sum | 148571 | 1129.70 | 88.03 | 82.06 | 148364 | 1129.77 | 31.03 | 82.84 |
| Ratio | 1.001 | 1.000 | 2.837 | 0.991 | 1.000 | 1.000 | 1.000 | 1.000 |

ISPD 2015

# Ablation Study

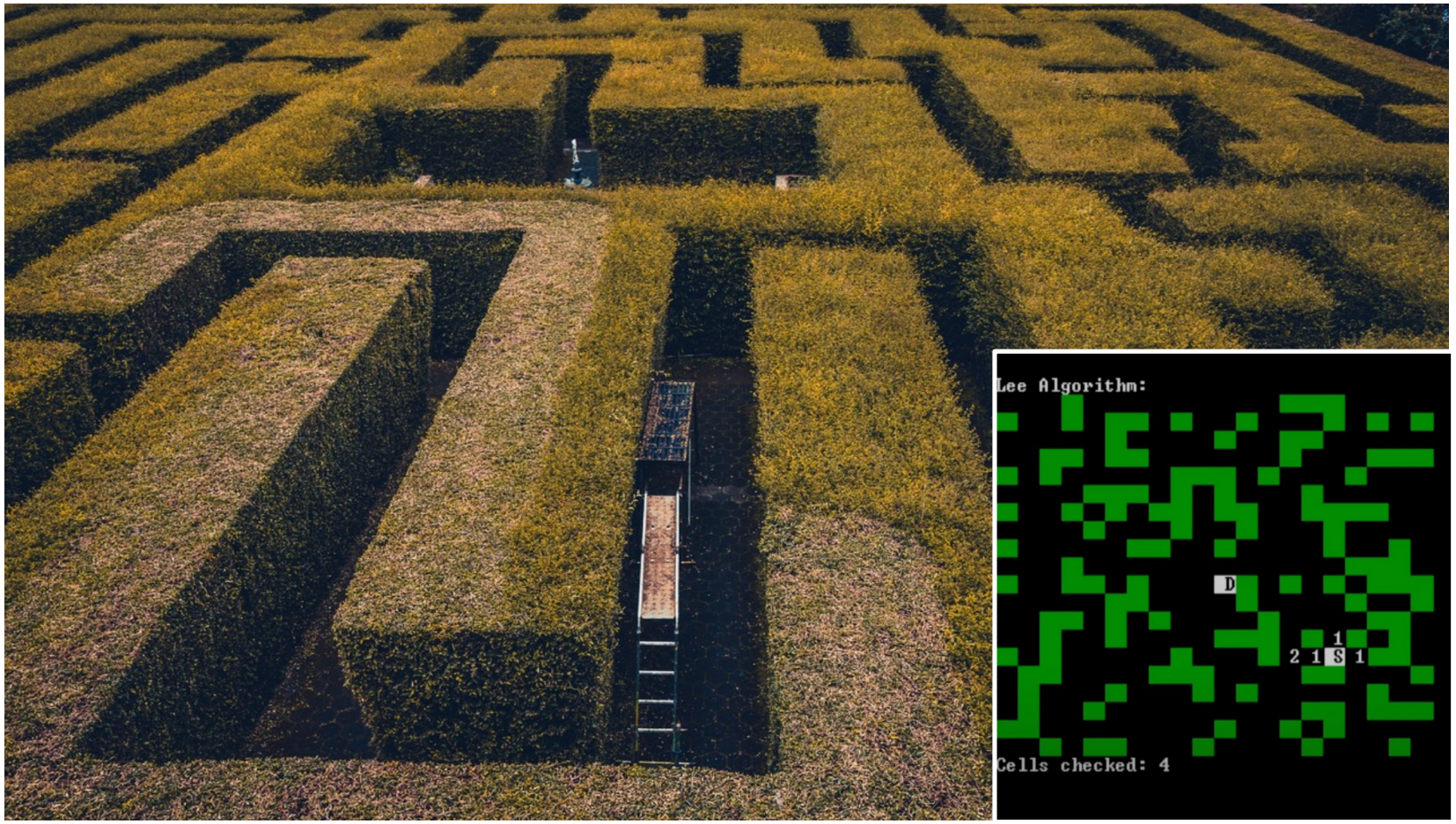
- Ablation Study of the Operator-Level Optimization Techniques

| Methods | OR | OC | OE | OS | adaptec1 | adaptec2 | adaptec3 | adaptec4 | bigblue1 | bigblue2 | bigblue3 | bigblue4 | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ratio | - | - | - | - | 234% | 194% | 136% | 124% | 198% | 140% | 123% | 121% | 159% |
| | ✓ | - | - | - | 110% | 109% | 113% | 115% | 105% | 115% | 119% | 118% | 113% |
| | ✓ | ✓ | - | - | 107% | 107% | 107% | 108% | 104% | 108% | 113% | 112% | 108% |
| | ✓ | ✓ | ✓ | - | 104% | 102% | 104% | 104% | 102% | 104% | 106% | 105% | 104% |
| Xplace | Ratio | | | | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | GP / Iter Time (ms) | | | | 1.478 | 1.671 | 2.325 | 2.688 | 1.572 | 2.441 | 4.974 | 10.018 | - |
| DREAMPlace | Ratio | | | | 462% | 345% | 288% | 254% | 376% | 288% | 199% | 158% | 296% |
| | GP / Iter Time (ms) | | | | 6.832 | 5.769 | 6.699 | 6.840 | 5.915 | 7.023 | 9.904 | 15.831 | - |

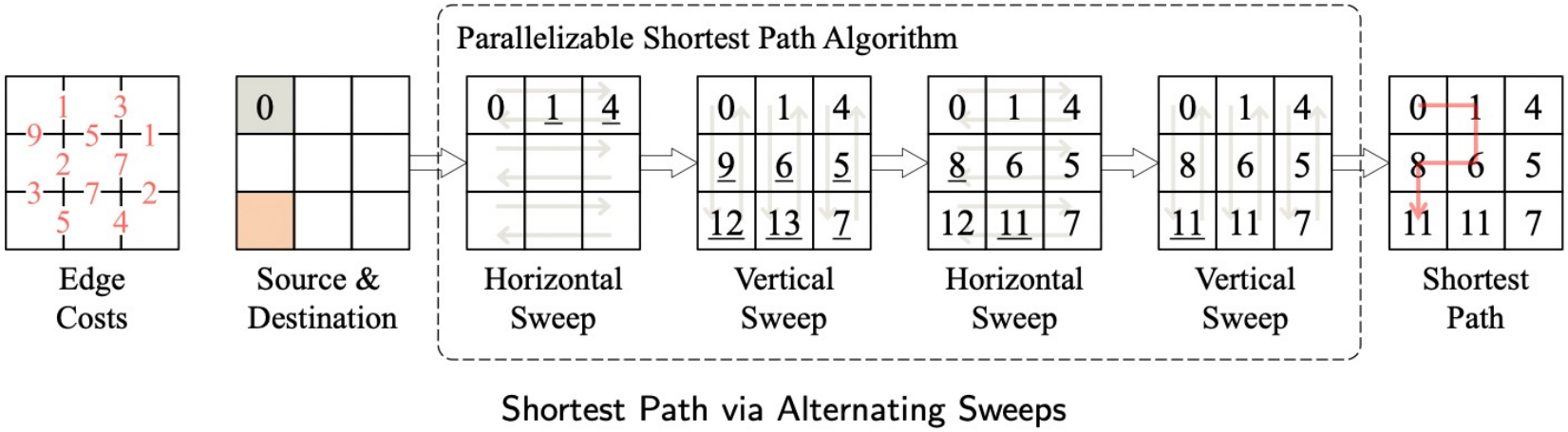
# Agenda

- GPU Acceleration on:
  - Placement
  - Routing
  - Routability-driven Placement

# GPU Accelerated Maze Routing



# GPU Accelerated Maze Routing




Shortest Path via Alternating Sweeps

$d_i$: current shortest distance to G-cell $i$

$c_i$: wire cost between G-cell $i-1$ and G-cell $i$

$$d_i^* = \min_{0 \le j \le i} \left( d_j + \sum_{k=j+1}^{i} c_k \right)$$

# Sweep Operations

- Notations
  - $d_i$: current shortest distance
  - $c_i$: wire cost
  - $s_i = \sum_{j=0}^{i} c_j$
  - Original formulation

$$d_i^* = \min_{0 \le j \le i} \left( d_j + \sum_{k=j+1}^{i} c_k \right)$$

- New formulation

$$d_i^* - s_i = \min_{0 \le j \le i} (d_j - s_j)$$

- This new formulation is a prefix sum problem and a prefix min problem.

# Maze Routing Results

- Sequential maze routing is implemented with priority queues.
- 11 alternations of sweeps in GAMER
- Directional change cost: $50$
- Congestion cost:
  - $91\% : \{1, 2, 3\}$
  - $1\% : 3 + 2^1$
  - $1\% : 3 + 2^2$
  - ...
  - $1\% : 3 + 2^9$

| Grid Graph Size | #Pins | Runtime (s) | | Speedup |
|---|---|---|---|---|
| | | GAMER | Maze Routing | |
| 256×256 | 4 | 0.07 | 1.29 | 17.36× |
| 256×256 | 8 | 0.14 | 2.51 | 17.71× |
| 256×256 | 16 | 0.29 | 4.93 | 17.25× |
| 512×512 | 4 | 0.16 | 5.57 | 33.92× |
| 512×512 | 8 | 0.25 | 11.44 | 45.14× |
| 512×512 | 16 | 0.40 | 24.60 | 61.53× |
| 1024×1024 | 4 | 0.34 | 30.40 | 90.02× |
| 1024×1024 | 8 | 0.65 | 74.51 | 115.05× |
| 1024×1024 | 16 | 1.25 | 152.52 | 121.78× |

S. Lin, J. Liu, Evangeline F.Y. Young, Martin D.F. Wong, "GAMER: GPU Accelerated Maze Routing", TCAD, 42(2):583-593, 2023

# Global Routing Results

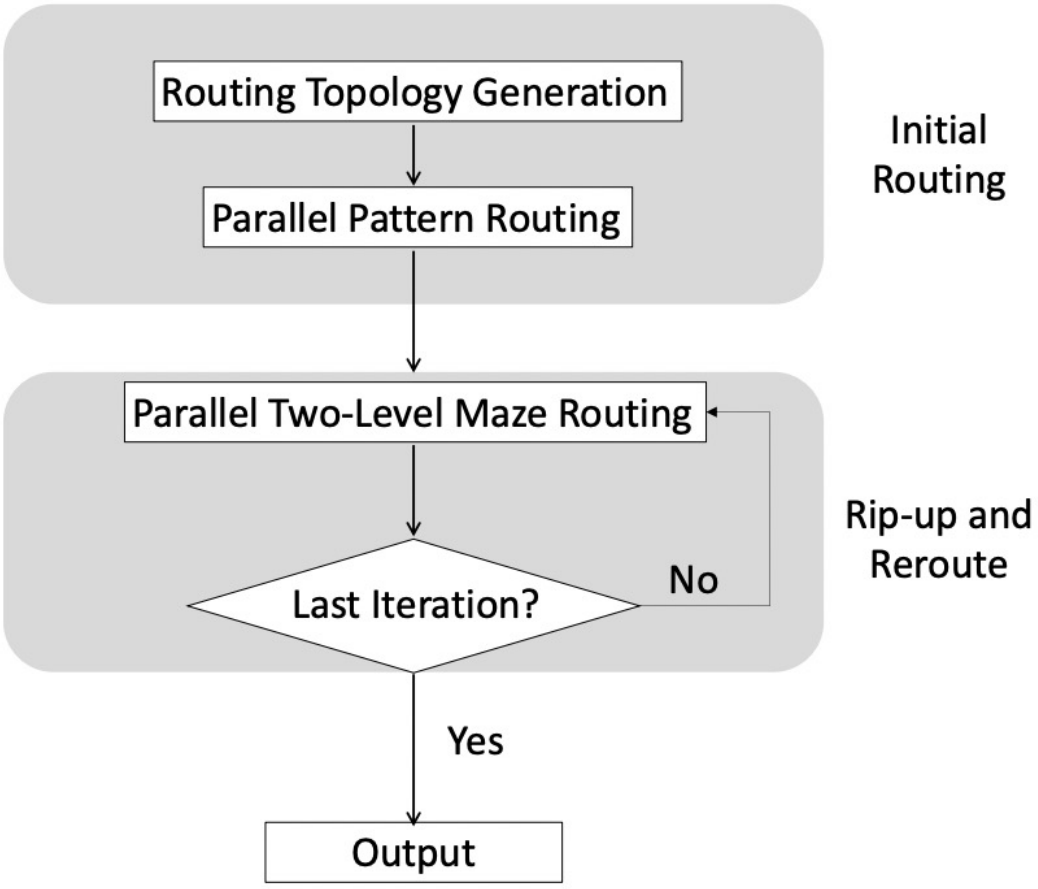
| Benchmarks | Coarse-Grained Time (s) | | | Fine-Grained Time (s) | | | Total Time (s) | | | Quality Score | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | CUGR | GAMER | Speedup | CUGR | GAMER | Speedup | CUGR | GAMER | Speedup | CUGR | GAMER |
| ispd18_test5 | 29.15 | 1.08 | 26.98 | 1.91 | 1.17 | 1.63 | 73.08 | 47.70 | 1.53 | 16104127 | 16146848 |
| ispd18_test8 | 185.21 | 6.83 | 27.13 | 8.22 | 5.58 | 1.47 | 282.90 | 99.72 | 2.84 | 37937622 | 37982628 |
| ispd18_test10 | 257.22 | 8.98 | 28.66 | 18.41 | 5.17 | 3.56 | 373.25 | 118.59 | 3.15 | 40593601 | 40942190 |
| ispd19_test7 | 488.70 | 11.45 | 42.67 | 11.63 | 7.84 | 1.48 | 652.42 | 170.61 | 3.82 | 88481579 | 86843860 |
| ispd19_test8 | 207.28 | 9.22 | 22.48 | 7.07 | 9.02 | 0.78 | 431.06 | 226.44 | 1.90 | 128287651 | 127394338 |
| ispd19_test9 | 308.81 | 11.72 | 26.34 | 7.29 | 7.09 | 1.03 | 620.88 | 326.84 | 1.90 | 201500802 | 199734785 |
| ispd18_test5_metal5 | 41.96 | 7.49 | 5.60 | 20.36 | 6.70 | 3.04 | 93.48 | 40.49 | 2.31 | 16206355 | 16258160 |
| ispd18_test8_metal5 | 147.38 | 22.09 | 6.67 | 60.89 | 14.61 | 4.17 | 289.95 | 100.72 | 2.88 | 37313105 | 37334467 |
| ispd18_test10_metal5 | 226.19 | 27.18 | 8.32 | 76.63 | 15.68 | 4.89 | 399.13 | 114.33 | 3.49 | 46068410 | 45991227 |
| ispd19_test7_metal5 | 297.36 | 15.72 | 18.92 | 30.29 | 12.89 | 2.35 | 434.59 | 136.02 | 3.20 | 82368279 | 81140446 |
| ispd19_test8_metal5 | 442.85 | 29.04 | 15.25 | 61.31 | 22.34 | 2.74 | 670.22 | 224.17 | 2.99 | 126219722 | 126706049 |
| ispd19_test9_metal5 | 427.64 | 46.91 | 9.12 | 72.96 | 18.48 | 3.95 | 732.86 | 304.05 | 2.41 | 197686583 | 196779131 |
| Average | | | **19.85**× | | | **2.59**× | | | **2.70**× | 84897320 | **84437844** |

# Modern Parallel Global Routers

| Router | Dimension | Pattern Routing | Maze Routing |
|---|---|---|---|
| SPRoute 2.0[5] | 2D | Multi-Threading | |
| CUGR[9] | 3D | | |
| GAMER[8] | | Multi-Threading | Partially GPU |
| FastGR[10] | | GPU | Multi-Threading |
| Ours | | GPU | |

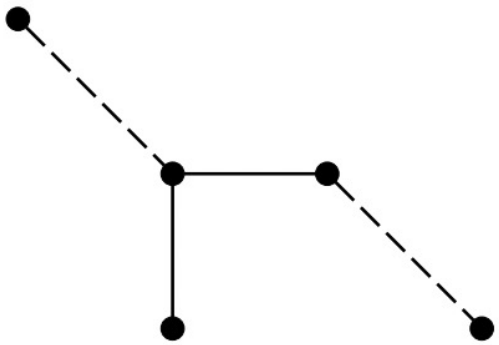
[5] “SPRoute 2.0: A Detailed-Routability-Driven Deterministic Parallel Global Router with Soft Capacity”, ASP-DAC 2022.
[9] “CUGR: Detailed-Routability-Driven 3D Global Routing with Probabilistic Resource Model”, DAC 2020.
[10] ”FastGR: Global Routing on CPU-GPU with Heterogeneous Task Graph Scheduler”, DATE 2022.
[11] “Superfast Full-Scale GPU-Accelerated Global Routing”, ICCAD 2022.

# An Overview of GUGR



# GPU Accelerate Pattern Routing

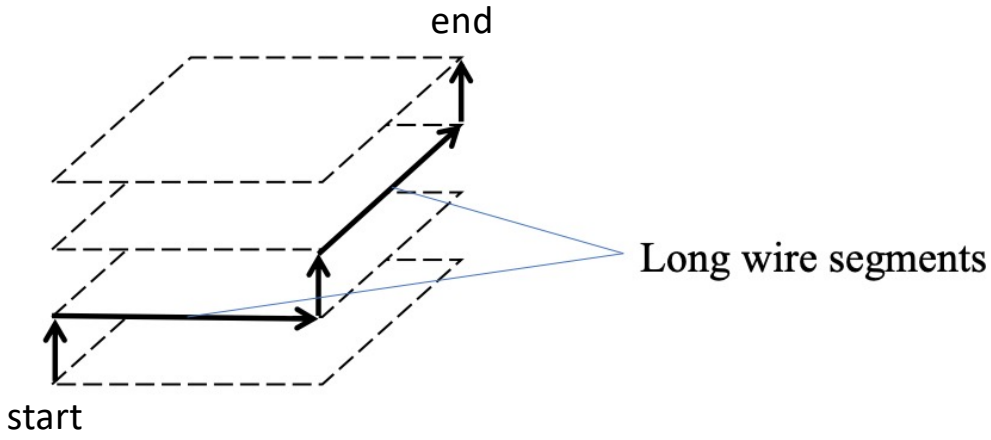
- RSMT construction with FLUTE
- L/Z-shape routing and layer assignment
- Jointly optimized by dynamic programming on GPU



2D Topology

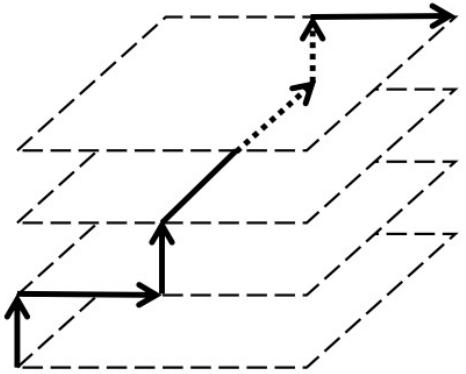
# Parallel L-Shaped Routing

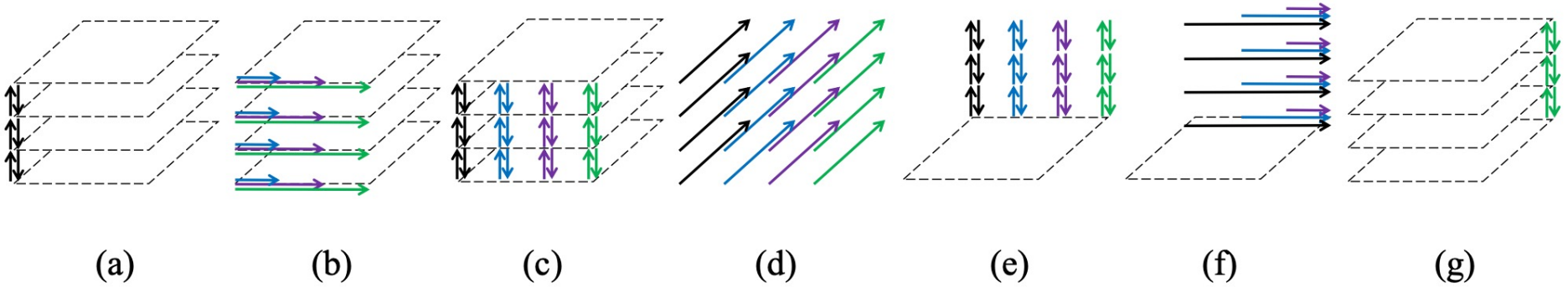
- L-shape routing can finish in O(K) time with one thread:
- Five steps:
  - from starting point to every layer
  - first long wire on every layer
  - bending on every layer
  - second long wire on every layer
  - from every layer to end point



# Parallel Z-Shaped Routing

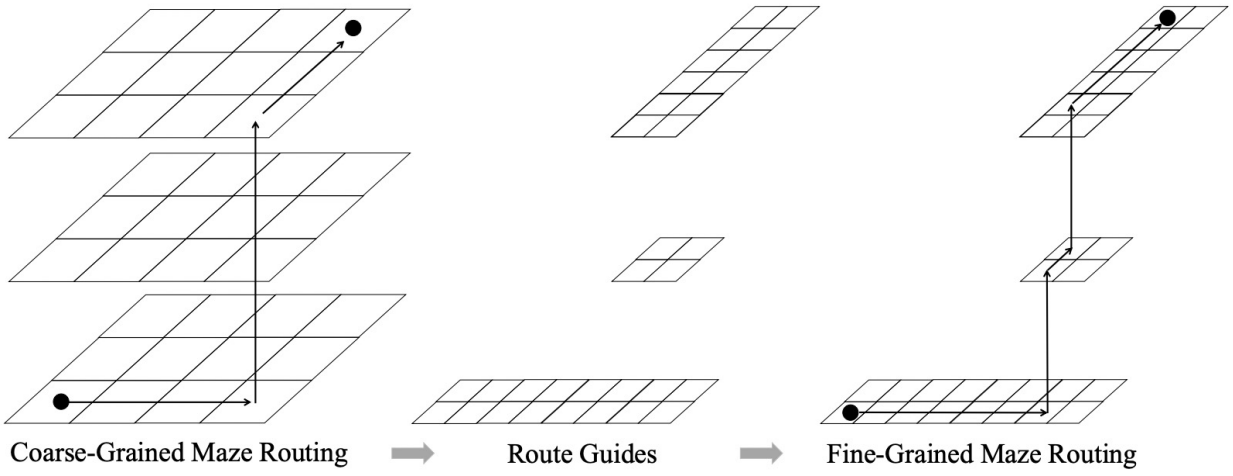
- $O(K)$ time with $n$ threads
- Seven steps as shown below. Each color represents one thread





# Other Parallelization Techniques

- Inter-net parallelism:
  - Route a batch of nets that do not have overlapping bounding boxes in parallel
  - Many nets are small and local, resulting in large batch size and small batch number
- Adapted GAMER to handle fine-grained maze routing with irregular routing grid graph



# Experimental Results

- All the routers are run on our machine

| Benchmark | CUGR[9] | | FastGR[10] | | GAMER[8] | | SPRoute 2.0[5] | | Ours | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Time | Score | Time | Score | Time | Score | Time | Score | Time | Score |
| ispd18_test5 | 73.1 | 16104127 | 34.5 | 16144948 | 48.0 | 16099658 | 9.5 | 16266324 | 5.6 | 16157416 |
| ispd18_test8 | 282.9 | 37937622 | 123.8 | 37899025 | 108.0 | 37936493 | 26.0 | 38303840 | 15.4 | 38123108 |
| ispd18_test10 | 373.2 | 40593601 | 151.5 | 40674135 | 126.0 | 40523120 | 33.0 | 42865920 | 17.5 | 41257617 |
| ispd19_test7 | 652.4 | 88481579 | 225.3 | 88502557 | 179.0 | 88478185 | 46.6 | 87406430 | 30.6 | 88686088 |
| ispd19_test8 | 431.1 | 128287651 | 222.2 | 128446998 | 253.0 | 128411260 | 65.5 | 127723613 | 37.0 | 127403748 |
| ispd19_test9 | 620.9 | 201500802 | 321.1 | 201490357 | 355.0 | 201161603 | 101.3 | 199900738 | 59.2 | 199574710 |
| ispd18_test5_metal5 | 93.5 | 16206355 | 46.5 | 16224663 | 55.0 | 16206853 | 11.1 | 16101826 | 7.8 | 16559899 |
| ispd18_test8_metal5 | 289.9 | 37313105 | 181.2 | 37240290 | 149.0 | 37262694 | 45.9 | 39604236 | 24.1 | 37738787 |
| ispd18_test10_metal5 | 399.1 | 46068410 | 212.1 | 48513161 | 192.0 | 45685561 | 97.2 | 47267127 | 22.0 | 47909903 |
| ispd19_test7_metal5 | 434.6 | 82368279 | 232.4 | 82243597 | 164.0 | 82349960 | 51.5 | 81104527 | 32.8 | 81344536 |
| ispd19_test8_metal5 | 670.2 | 126219722 | 472.5 | 126674690 | 265.0 | 126175581 | 128.3 | 126389940 | 48.4 | 126366645 |
| ispd19_test9_metal5 | 732.9 | 197686583 | 652.9 | 197866328 | 367.0 | 197778170 | 128.3 | 197033254 | 80.1 | 195588171 |
| Average | 421.2 | 84897320 | 239.7 | 85160062 | 188.4 | 84839095 | 62.0 | 84997315 | 31.7 | 84725886 |
| Ratio | 13.3 | 1.002 | 7.6 | 1.005 | 5.9 | 1.001 | 2.0 | 1.003 | **1.0** | **1.000** |


[9] “CUGR: Detailed-Routability-Driven 3D Global Routing with Probabilistic Resource Model”, DAC 2020.
[10] ”FastGR: Global Routing on CPU-GPU with Heterogeneous Task Graph Scheduler”, DATE 2022.
[5] “SPRoute 2.0: A Detailed-Routability-Driven Deterministic Parallel Global Router with Soft Capacity”, ASP-DAC 2022.
[11] “Superfast Full-Scale GPU-Accelerated Global Routing”, ICCAD 2022.

# Agenda

- GPU Acceleration on:
  - Placement
  - Routing
  - Routability-driven Placement

# Routability Xplace (All on GPU)

| DR | | | Xplace + GGR | | | | | | DREAMPlace | | | | | | DREAMPlace + CUGR | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| design | #cells | #nets | DR WL (um) | #DR Vias | #DRC | Place Time (s) | DR Time (s) | PnR Total Time ( | DR WL (um) | #DR Vias | #DRC | Place Time (s) | DR Time (s) | PnR Total Time ( | DR WL (um) | #DR Vias | #DRC | Place Time (s) | DR Time (s) | PnR Total Time ( |
| des_perf_1 | 113k | 113k | 1439037 | 569154 | 19642 | 16 | 2594 | 2610 | 1450914 | 571104 | 20104 | 16 | 2727 | 2743 | 1581152 | 612600 | 21358 | 208 | 3412 | 3620 |
| des_perf_a | 108k | 115k | 1906755 | 564605 | 21599 | 20 | 3716 | 3736 | 2442153 | 548296 | 39528 | 16 | 737 | 753 | 2439715 | 546586 | 37663 | 201 | 763 | 964 |
| des_perf_b | 113k | 113k | 1456675 | 546994 | 19278 | 18 | 2118 | 2136 | 1877233 | 538765 | 14923 | 16 | 2430 | 2446 | 1875447 | 525621 | 14021 | 253 | 2380 | 2633 |
| edit_dist_a | 127k | 134k | 5261914 | 961245 | 468215 | 23 | 2743 | 2766 | 5792993 | 1015546 | 463347 | 18 | 2773 | 2791 | 5786429 | 1015029 | 463634 | 195 | 2892 | 3087 |
| fft_1 | 35k | 33k | 512157 | 186784 | 7020 | 12 | 1286 | 1298 | 518943 | 187982 | 7822 | 25 | 1453 | 1478 | 526801 | 184942 | 8235 | 89 | 265 | 354 |
| fft_2 | 35k | 33k | 621183 | 196145 | 4787 | 11 | 1262 | 1273 | 601729 | 187979 | 9445 | 10 | 428 | 438 | 606512 | 187448 | 7665 | 72 | 353 | 425 |
| fft_a | 34k | 32k | 1123576 | 195799 | 5110 | 12 | 1348 | 1360 | 1079919 | 192806 | 5426 | 11 | 1151 | 1162 | 1082475 | 193604 | 5586 | 81 | 1218 | 1299 |
| fft_b | 34k | 32k | 1254629 | 203870 | 35161 | 13 | 773 | 786 | 1254097 | 212459 | 21824 | 12 | 846 | 858 | 1249536 | 211817 | 22050 | 82 | 876 | 958 |
| matrix_mult_1 | 160k | 159k | 2645899 | 829772 | 28676 | 21 | 6567 | 6588 | 2717619 | 811317 | 79404 | 21 | 1383 | 1404 | 2713886 | 810909 | 81273 | 155 | 1286 | 1441 |
| matrix_mult_2 | 160k | 159k | 2684676 | 864351 | 30878 | 22 | 7186 | 7208 | 2724075 | 842557 | 69226 | 16 | 1516 | 1532 | 2713664 | 835190 | 55544 | 142 | 1655 | 1797 |
| matrix_mult_a | 154k | 154k | 3897297 | 863779 | 27597 | 5 | 6061 | 6066 | 3880868 | 863304 | 27369 | 17 | 5010 | 5027 | 3881274 | 865086 | 27455 | 135 | 5171 | 5306 |
| matrix_mult_b | 146k | 152k | 3714463 | 783806 | 97327 | 17 | 1643 | 1660 | 3763409 | 753502 | 78694 | 19 | 1437 | 1456 | 3749777 | 743892 | 57641 | 129 | 1259 | 1388 |
| matrix_mult_c | 146k | 152k | 3854406 | 793071 | 63158 | 15 | 1265 | 1280 | 3783339 | 785381 | 30185 | 21 | 7778 | 7799 | 3779358 | 785276 | 30140 | 138 | 8187 | 8325 |
| pci_bridge32_a | 30k | 34k | 356213 | 127447 | 4939 | 12 | 857 | 869 | 668228 | 143073 | 5959 | 10 | 2603 | 2613 | 667505 | 143367 | 6388 | 100 | 2844 | 2944 |
| pci_bridge32_b | 29k | 33k | 799554 | 134161 | 5526 | 12 | 954 | 966 | 1032143 | 145702 | 2872 | 14 | 380 | 394 | 1052680 | 146910 | 2791 | 328 | 349 | 677 |
| superblue11_a | 926k | 936k | 39819167 | 6229072 | 4175 | 97 | 8819 | 8916 | 39989373 | 5659125 | 2476 | 92 | 7565 | 7657 | 41233052 | 5594255 | 2331 | 1270 | 7353 | 8623 |
| superblue12 | 1293k | 1293k | 42991473 | 10294135 | 22463 | 277 | 20982 | 21259 | 42643015 | 11289783 | 3362900 | 101 | 30045 | 30146 | 48380447 | 11537300 | 36723 | 2589 | 29528 | 32117 |
| superblue14 | 634k | 620k | 27894811 | 4248981 | 309 | 55 | 11237 | 11292 | 28368425 | 4415917 | 356 | 56 | 14321 | 14377 | 28667576 | 4224657 | 335 | 1163 | 7029 | 8192 |
| superblue16_a | 680k | 697k | 29968257 | 4918597 | 4127 | 174 | 19790 | 19964 | 31468535 | 4734294 | 3422 | 62 | 15843 | 15905 | 31330051 | 4612795 | 2465 | 823 | 12866 | 13689 |
| superblue19 | 522k | 512k | 20384423 | 3501561 | 8071 | 41 | 8088 | 8129 | 20568915 | 3608958 | 7147 | 54 | 7767 | 7821 | 23358871 | 3839686 | 14518 | 1373 | 12148 | 13521 |
| mean | | | **9629328** | **1850666** | **43903** | 44 | 5464 | 5508 | 9831296 | 1875393 | 212621 | 30 | 5410 | 5440 | 10333810 | 1880849 | 44891 | 476 | 5092 | 5568 |
| geo mean | | | **3346652** | **783682** | **14063** | 23 | 3230 | 3258 | 3613371 | 788832 | 18469 | 23 | 2686 | 2715 | 3687014 | 789453 | 14462 | 246 | 2421 | 2770 |
| trim mean (outlier) | | | **6806722** | 1264282 | **19255** | 25 | 4180 | 4205 | 7054554 | 1266885 | 26459 | 24 | 3844 | 3870 | 7245587 | **1262196** | 21890 | 338 | 3677 | 4049 |

Place Time (s) includes IO, GP, DP, and internal GR

# Thank YOU