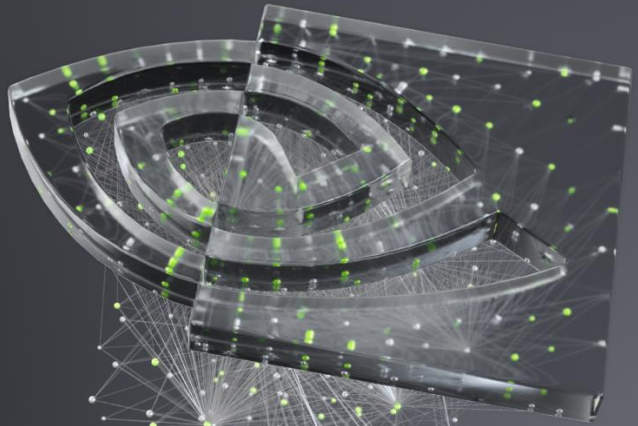


Mixed-Cell-Height Legalization on CPU-GPU Heterogeneous Systems

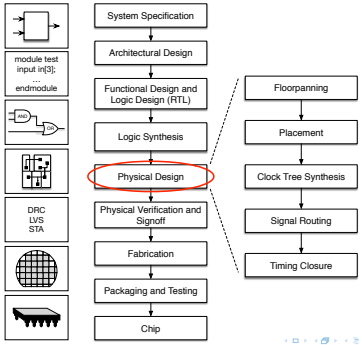
Haoyu Yang (NVIDIA Research), Kit Fung (CUHK), Yuxuan Zhao (CUHK), Yibo Lin (PKU), Bei Yu (CUHK)

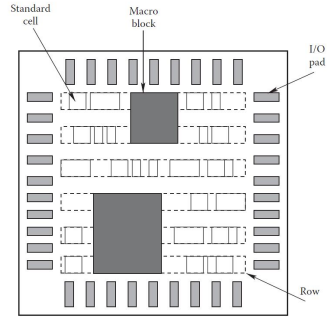
Mar. 2022

Outline

Outline

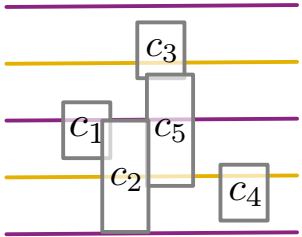
Placement Overview



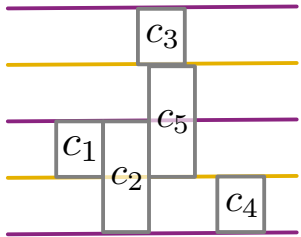


- Placement is an essential step in the physical design flow, it assigns exact locations for various circuit components within the chip's core area
- Typical placement objectives include total wirelength, timing, congestion, and power

Global Placement and Legalization



(a) Global Placement



(b) Legalization

Mixed-Cell-Height Legalization

Objectives

- No overlapped cells.
- Cells aligned with layout power lines.
- Cells entirely placed within assigned fences.

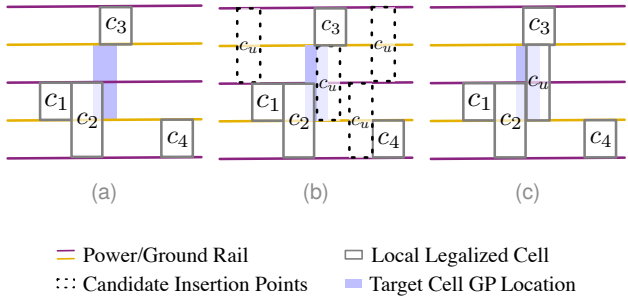
Preserve Global Placement Quality

- Minimize cell displacement.

$$\delta_i = |x_i - x_i'| + |y_i - y_i'|,$$

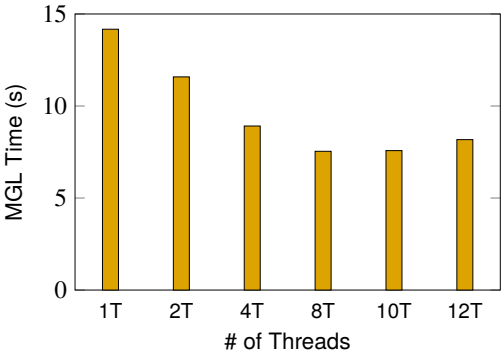
$$S_{\text{am}} = \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} \frac{1}{|\mathcal{C}_h|} \sum_{c_i \in \mathcal{C}_h} \delta_i.$$

Multi-row Global Legalization [Li+, DAC'2018]

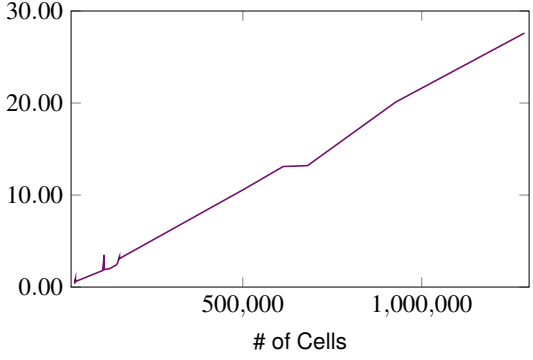


- Local Cell Search: Find a target cell (unlegalized) and its surrounding legalized cells within a window.
- Insertion Interval Evaluation: Find possible intervals to insert the target cell and caused displacement.
- Target Cell Legalization: Place the target cell at the interval with min displacement.

Runtime Scales of MGL

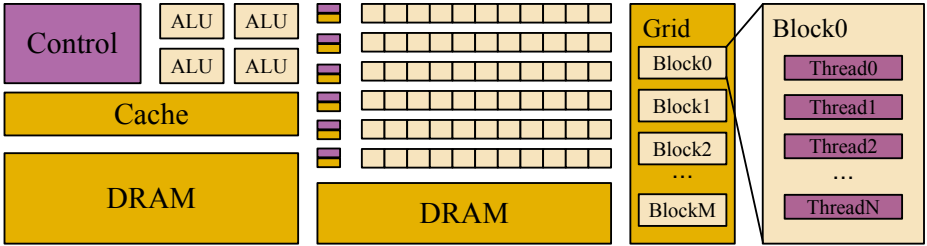


(a)



(b)

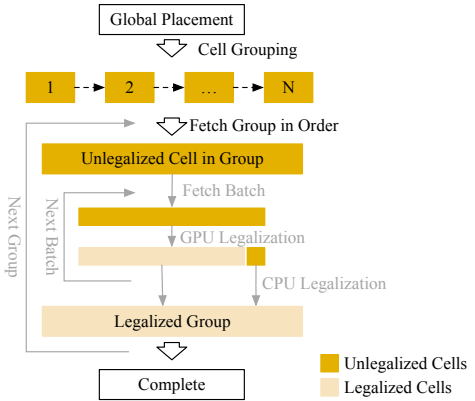
GPU vs. CPU



- CPU is talented at complicated control logic and instruction set.
- GPU comes with massive computing units and simple control units.
- GPU also support multi-level thread hierarchy.

Outline

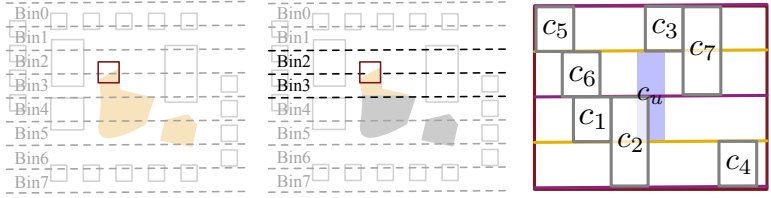
Our Proposal: Heterogeneous Global Legalization



- We observe large fraction of cells ($> 90\%$) can be easily legalized.
- Minor cells are hard to legalize that requires iterative search, lift and replace.
- GPU handles massive easy-to-place cells and leave few stubborn cells to CPU.

Key Kernel Functions

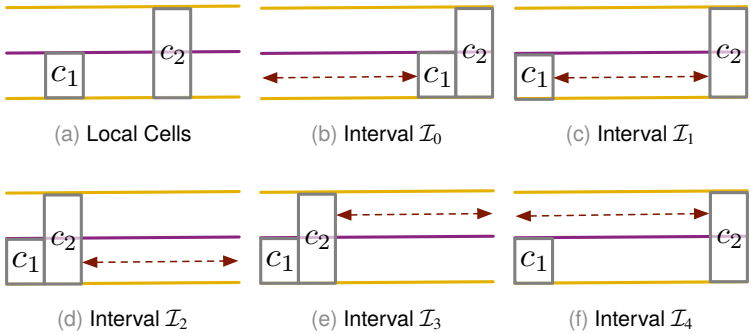
Local Region Search



(a) Bin extraction; (b) Local cell search on bins that touch the given search window; (c) Extracted region with local cells.

Key Kernel Functions

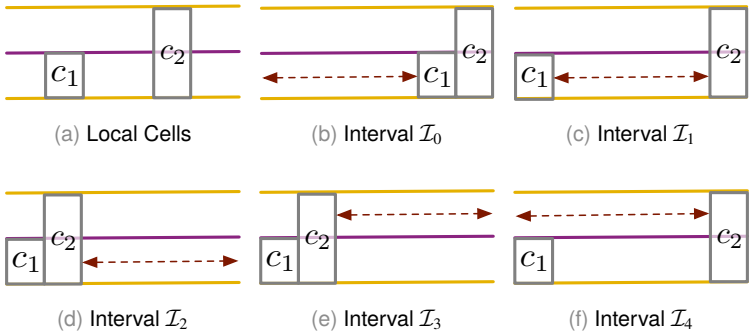
Interval Evaluation



(a) An example search window with 2 local cells; (b) Bottom row interval with c_1 and c_2 pushed to right; (c) Bottom row interval with c_1 pushed to left and c_2 pushed to right; (d) Bottom row interval with c_1 and c_2 pushed to left; (e) Top row interval with c_1 and c_2 pushed to left; (f) Top row interval with c_2 pushed to right.

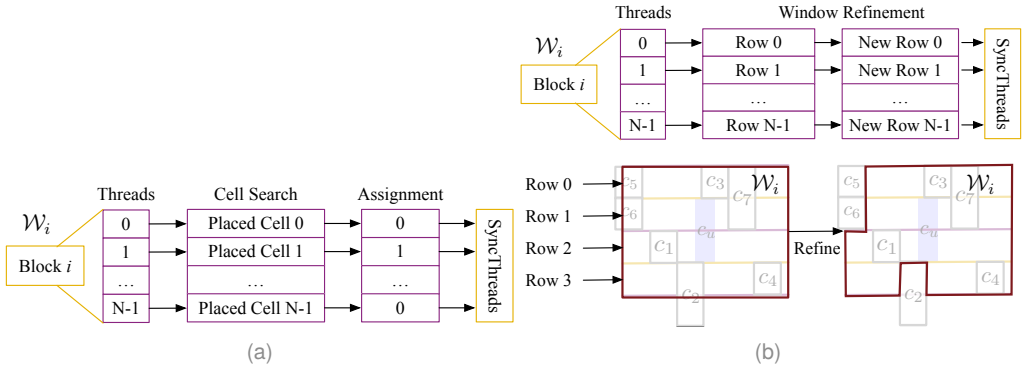
Key Kernel Functions

Interval Evaluation



(a) An example search window with 2 local cells; (b) Bottom row interval with c_1 and c_2 pushed to right; (c) Bottom row interval with c_1 pushed to left and c_2 pushed to right; (d) Bottom row interval with c_1 and c_2 pushed to left; (e) Top row interval with c_1 and c_2 pushed to left; (f) Top row interval with c_2 pushed to right.

GPU Local Region Search



GPU local region $\mathcal{W}_i$ extraction. CUDA thread management for (a) local cell search and (b) search window refinement, where each thread block is responsible for the search window of one target cell.

GPU Global Legalization

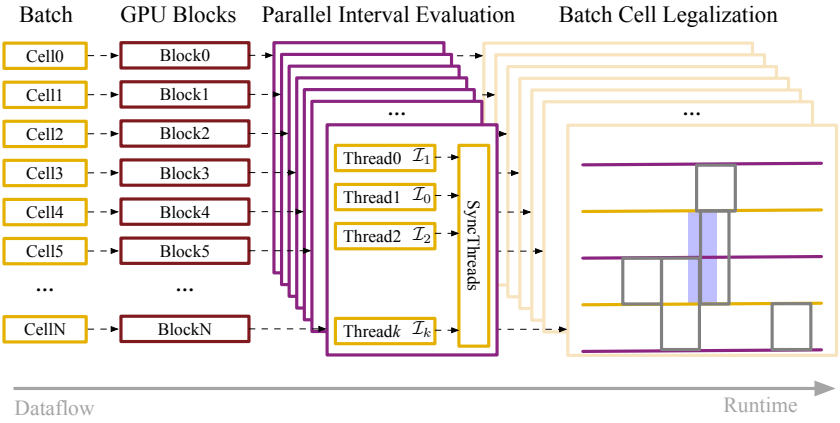


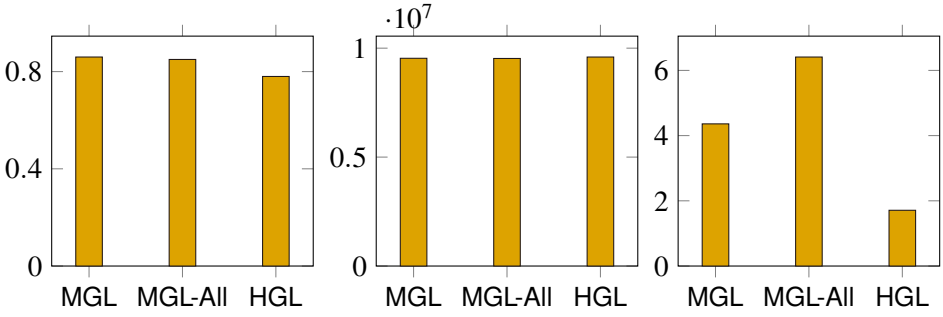
Illustration of GPU legalization kernel.

Discussion: Why CPU Do Better on Stubborn Cells?

- In CPU, insertion interval candidates are handled in Queue structure such that each possible interval will be evaluated.
- Complicated data structure is not available on GPU, we therefore limit the max # of possible intervals.
- When there are no available interval to place a cell, CPU implementation can increase the window size and perform legalization iteratively, which is extremely costly for GPU.

Outline

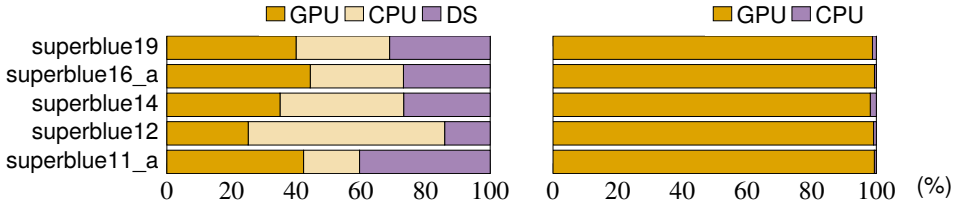
Performance Evaluation of HGL



(a) Average Disp. (b) HPWL (c) Runtime (s)

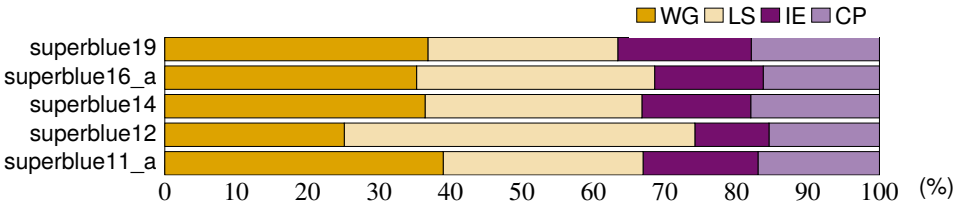
- MGL evaluated on CPU with 8 Thread.
- HGL evaluated on GeForce GTX1660Ti.
- HGL achieves comparable legalization performance with $2\times$ speedup over 8T CPU even we are only using very low end GPU.

Efficiency Study



(a) Runtime Breakdown-Device

(b) GPU CPU Workload



(c) Runtime Breakdown-Kernel

Outline

Summary and Future Work

- Discuss the opportunity of high performance legalization on GPU-CPU heterogeneous platforms.
- Proposed HGL algorithm with GPU legalization kernels, CPU global legalization and task scheduling.
- Reduce CPU-GPU communication overhead.
- Optimize memory usage.
- Smarter scheduler for concurrent CPU GPU execution.

Thank You