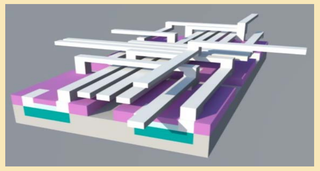
Parallel VLSI CAD Algorithms for Energy Efficient Heterogeneous Computing Platforms

Xueqian Zhao, Zhuo Feng
Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI

Michigan Tech — Create the Future

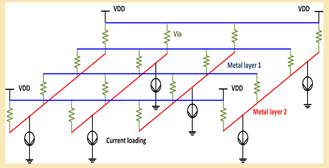
1. INTRODUCTION

3D Interconnections



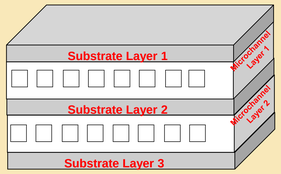
3D Parasitic Extraction: Capacitance extraction for delay, noise and power estimation computes charge distributions for discretized panels

Power Delivery Network



Large Power Grid Simulation: Power grids DC and transient simulations involve more than hundreds of millions of nodes

3D-ICs w/ Microchannels



Full Chip Thermal Simulation: Thermal simulation for 3-D dense mesh structures requires solving many millions of unknowns

a) GPU Has Evolved into A Very Flexible and Powerful Processor

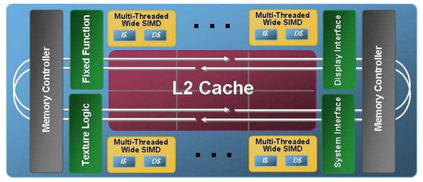
- 180 GB/s (GPU) vs. 30 GB/s (CPU)
- 1350 GFLOPS (GPU) vs. 200 GFLOPS (CPU)

b) Heterogeneous CPU-GPU Computing Achieves Much Better Energy Efficiency

- Achieve highly energy efficient computing, and CPU-GPU communications
- Need reinventing CAD algorithms: circuit simulations, interconnect modeling, performance optimizations,…

NVIDIA Tegra 2

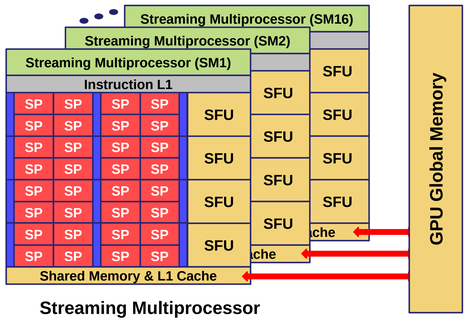
AMD APU



Intel Larrabee

c) Latest NVIDIA Fermi GPU Architecture

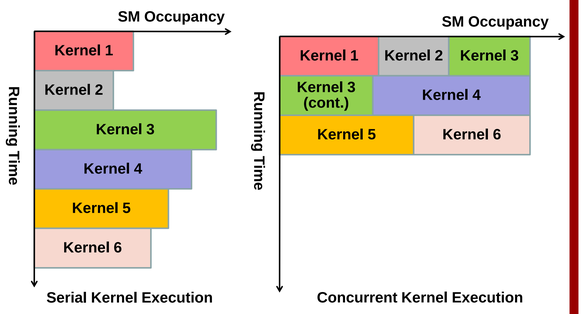
- All processors do the same job on different data sets (SIMD)
- More transistors are devoted to data processing
- Less transistors are used for data caching and control flow



Streaming Multiprocessor

d) Concurrent Kernel Executions

- GPU concurrent kernel execution can improve the SM occupancy.
- Fermi GPUs allow at most *16* different kernels executed at the same time

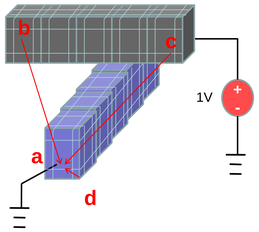


2. CAPACITANCE EXTRACTION

a) Problem Formulation

$$V_{b\to a} = \int_{\text{panel } b} \frac{\varepsilon_b}{R_{ba}} dA$$

$$\varepsilon_b = \frac{q_b}{A_b}$$



Summation: $V_a = \sum_{i=b}^{\cdots} \frac{q_i}{A_i} \int_{\text{panel } i} \frac{1}{R_{ia}} dA_i = \sum \Phi_i q_i$

b) FMM Turns the O(n²) into O(n) via Approximations

$$p_k = \sum_{x_i \in \Omega(y_k)} \Phi^{(k,i)} q_i + \sum_{x_i \notin \Omega(y_k)} \Phi^{(k,i)} q_i, \quad x_i, y_k \in \mathbb{R}^d$$

Directly Solve — Approximately Solve

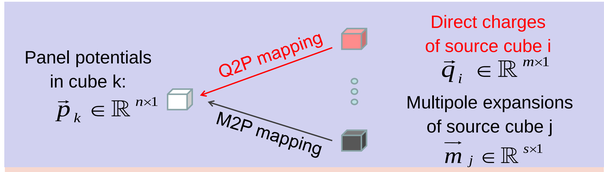
Key Steps:

- Direct Pass — Q2P
- Upward Pass — Q2M, M2M
- Downward Pass — M2L, L2L
- Evaluation Pass — M2P, L2P

Multipole expansion: Approximate panel charges within a small sphere

Local expansion: Approximate panel potentials within a small sphere

c) FMMGpu Data Structure and Algorithm Flow



$$\vec{p}_k = \Phi^{(k,i)}_{Q2P} \times \vec{q}_i + \cdots + \Phi^{(k,j)}_{M2P} \times \vec{m}_j$$

$$\Rightarrow \vec{p}_k = \left[\Phi^{(k,i)}_{Q2P} \cdots \Phi^{(k,j)}_{M2P}\right] \times \begin{bmatrix} \vec{q}_i \\ \vdots \\ \vec{m}_j \end{bmatrix}$$

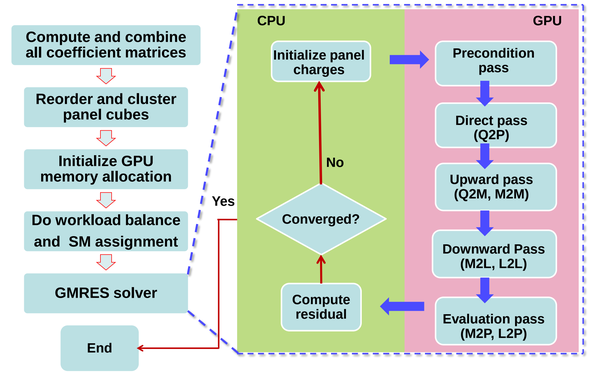
Q2P coefficient matrix: $\Phi^{(k,i)}_{Q2P} \in \mathbb{R}^{n\times m}$

M2P coefficient matrix: $\Phi^{(k,j)}_{M2P} \in \mathbb{R}^{n\times s}$

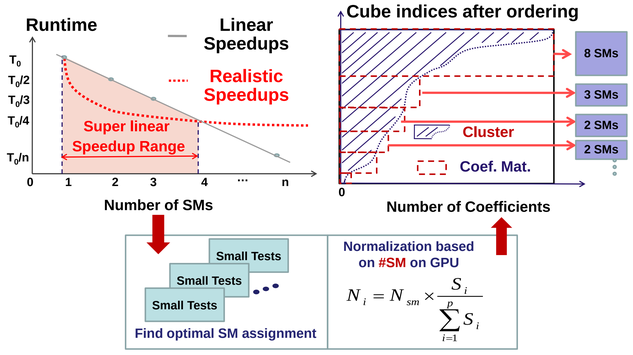
$$\underbrace{\begin{bmatrix} \vec{p}_k \\ \vdots \\ \vec{p}_l \end{bmatrix}}_{\vec{p}} = \underbrace{\begin{bmatrix} \Phi^{(k,i)}_{Q2P} & \cdots & \Phi^{(k,j)}_{M2P} \\ \vdots & \cdots & \vdots \\ \Phi^{(l,k)}_{Q2P} & \cdots & \Phi^{r\times d} \end{bmatrix}}_{\Phi_{mix}} \otimes \underbrace{\begin{bmatrix} I^{(k,i)}_{Q2P} & \cdots & I^{(k,j)}_{M2P} \\ \vdots & \cdots & \vdots \\ I^{(l,k)}_{Q2P} & \cdots & I^{r\times d} \end{bmatrix}}_{I_{mix}}$$

Q2P index matrix: $I^{(k,i)}_{Q2P} \in \mathbb{R}^{n\times m}$ M2P index matrix: $I^{(k,j)}_{M2P} \in \mathbb{R}^{n\times s}$

Global vector: $\overline{Vec}_{global} = \left[0 \;\; \vec{q}_1^T \;\; \cdots \;\; \vec{q}_x^T \;\; \vec{m}_1^T \;\; \cdots \;\; \vec{m}_y^T \;\; \vec{l}_1^T \;\; \cdots \;\; \vec{l}_z^T\right]^T$

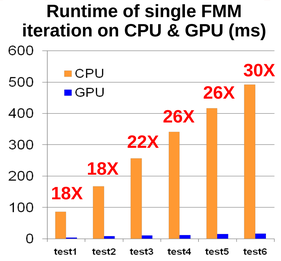


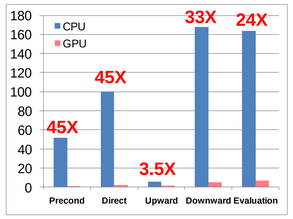
d) GPU Performance Modeling and Workload Balancing



$$N_i = N_{sm} \times \frac{S_i}{\sum_{i=1}^{p} S_i}$$

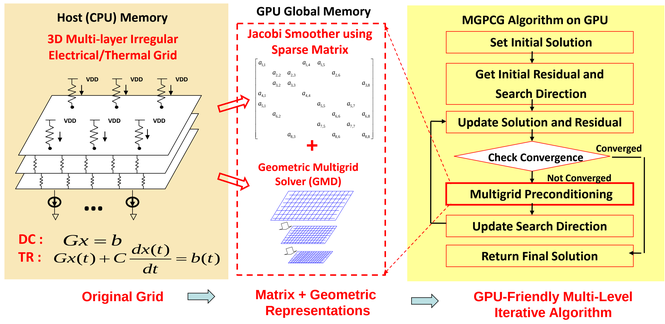
e) Experimental Results [4]





3. PERFORMANCE MODELING & OPTIMIZATION FOR ELECTRICAL & THERMAL SIMULATIONS

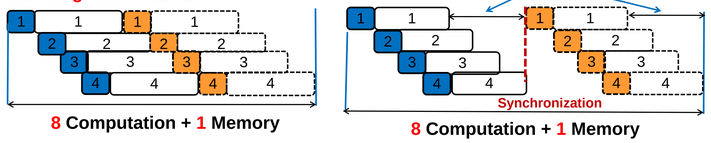
a) Multigrid Preconditioned Krylov Subspace Iterative Solver



b) Analytical GPU Runtime Performance Model (ISCA'09)

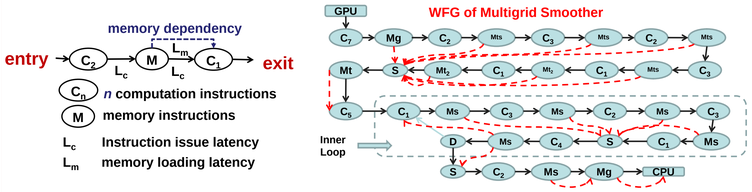
- **MWP(Memory Warp Parallelism):** the max. num. of warps per SM that can access the memory simultaneously
- **CWP(Computation Warp Parallelism):** the num. of warps the SM processor can execute during one memory warp waiting period plus *one*

MWP is greater than CWP



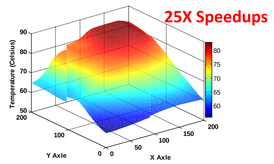
c) Adaptive Performance Modeling (PPoPP'10)

- **Work Flow Graph (WFG):** an extension of the control flow graph of a GPU kernel

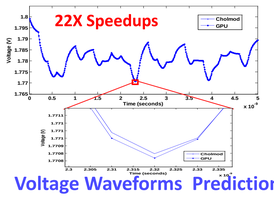


d) Experimental Results

Chip-level Electrical [1,2] and Thermal Modeling & Simulation [3]

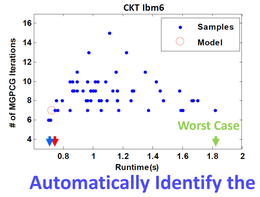


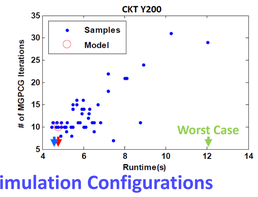
IC Temperature Prediction



Voltage Waveforms Prediction

Performance Modeling & Optimization [1]





Automatically Identify the Optimal Simulation Configurations

4. REFERENCE

[1] Z. Feng, X. Zhao, and Z. Zeng. Robust Parallel Preconditioned Power Grid Simulation on GPU with Adaptive Runtime Performance Modeling and Optimization. In IEEE*TCAD*, 2011.
[2] Z. Feng and Z. Zeng. Parallel multigrid preconditioning on graphics processing units (GPUs) for robust power grid analysis. In *Proc. IEEE/ACM DAC*, 2010.
[3] Z. Feng and P. Li. Fast thermal analysis on GPU for 3D-ICs with integrated microchannel cooling. In *Proc. IEEE/ACM ICCAD*, 2010.
[4] X. Zhao and Z. Feng. Fast Multipole Method on GPU: Tackling 3-D Capacitance Extraction on Massively Parallel SIMD Platforms. In *Proc. IEEE/ACM DAC*, 2011.