Delivering *Success*.

Hardware accelerated interconnect capacitance extractor for VLSI design

Narain D. Arora
Silterra Malaysia Sdn. Bhd.
Email: narain@silterra.com
Steve Worley
Nanosim Laboratories, USA

Outline

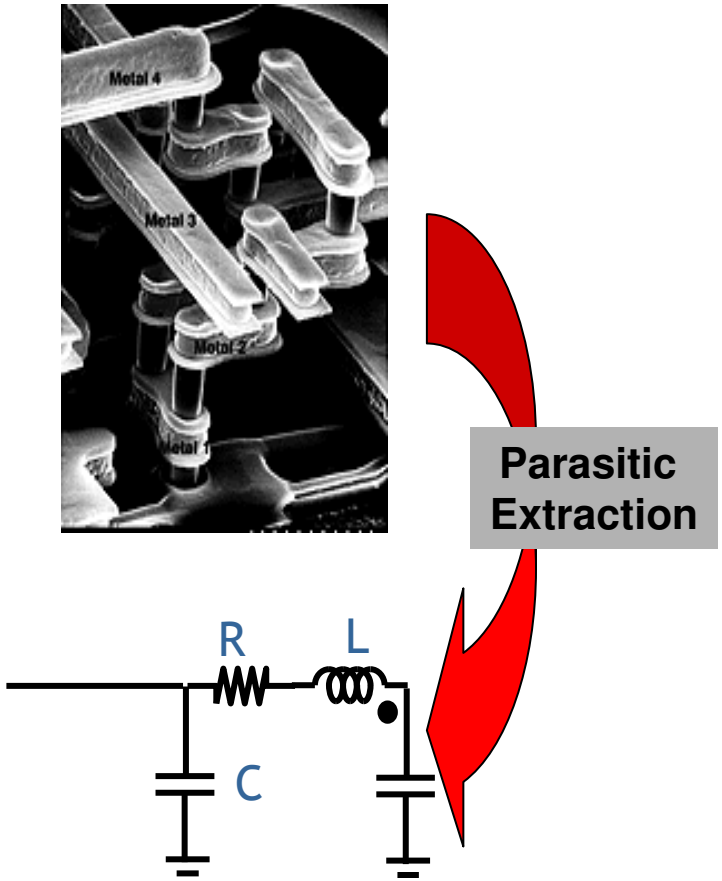
SILTERRA
Delivering *Success*.

- Interconnect (wire) capacitance – a Parasitic element
  - an overview
- Methods of modeling Capacitance – full chip
  - Modeling Approach
  - Field Solver Approach
- Monte Carlo solver as enabler for full chip wire
` Capacitance estimation
  - CPU – Distributed processing
  - GPU – increases speed many folds
- Conclusion

Interconnect - Parasitic Element

SILTERRA Delivering Success.

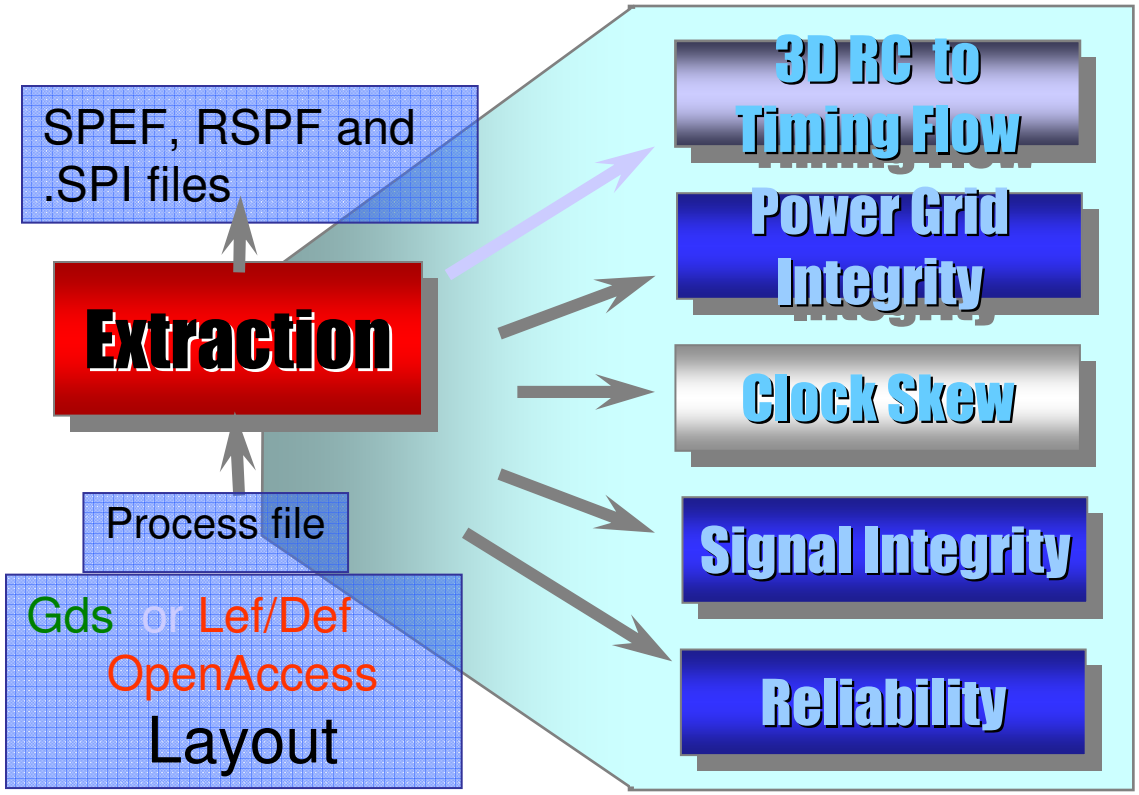
- Interconnects/Wires are critical to SoC design as it affects
  - Timing (clock frequency)
  - Signal Integrity (noise)
  - Power consumption
  - Yield
- As we scaled the devices, so are the interconnects that are represented by parasitic elements:
  - Resistance, R
  - Capacitance, C
  - Inductance, L
- Modeling wire R,C, and L is very important as inaccurate extraction result in
  - Chip failure
  - Low performance
  - High power consumption



Parasitic Extractor

SILTERRA
Delivering Success.

Tools used to extract R,C, L at chip level are called Parasitic Extractors



Speed and accuracy of parasitic extraction are critical to the success of multi-million gate designs

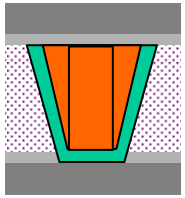
Wire Modeling Complexity

SILTERRA
Delivering Success.

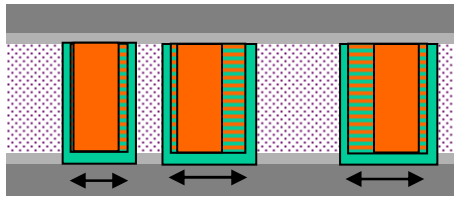
➢ What You Draw Isn't What you see in Silicon (DFM effects

- variations from drawn are much more significant in Cu than Al

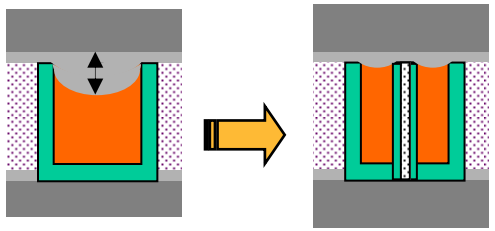
- Non-rectangular wires
  - Wire shape is more pronounced <150nm



- Wire Edge Effects (optical)
  - Actual wire widths vary depending on spacing to neighboring wires on each side
  - Varies wire resistance and capacitance



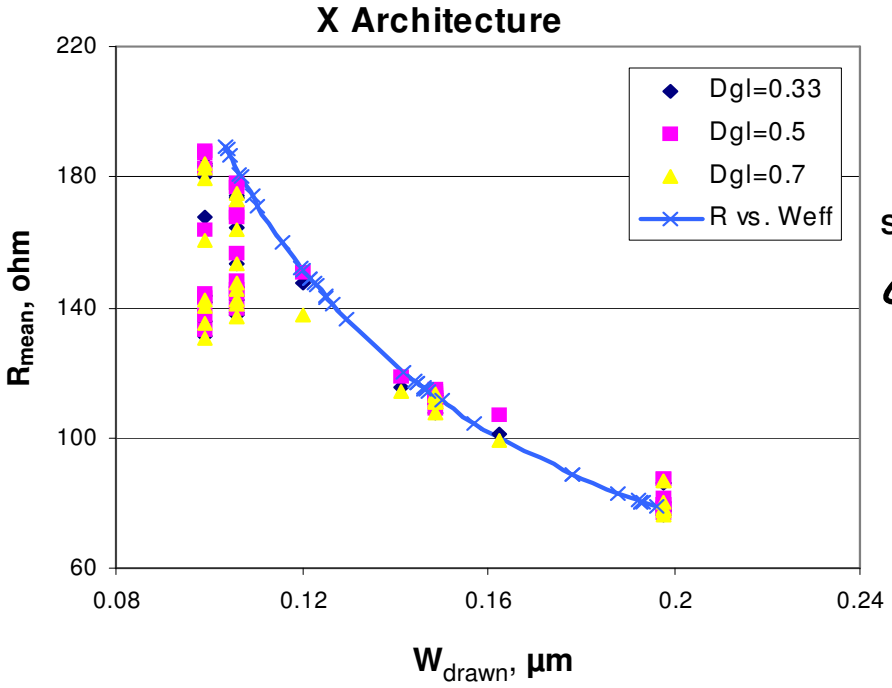
- Dishing, slotting
  - Dishing: CMP rubs away metal
  - Slotting/cheesing: slots or holes are placed in wires to reduce dishing

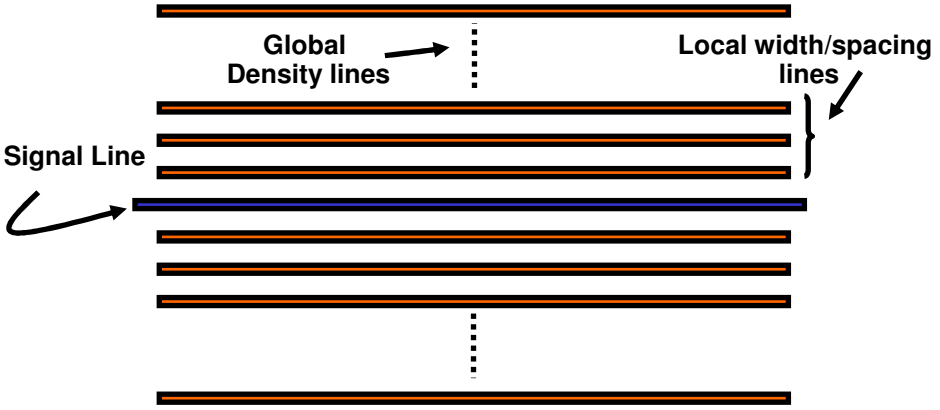


- Wire R, C function of wire width, spacing, thickness, and wire density

Influence of local and global environment on the line Resistance

Measurement data at 65 nm for metal M3



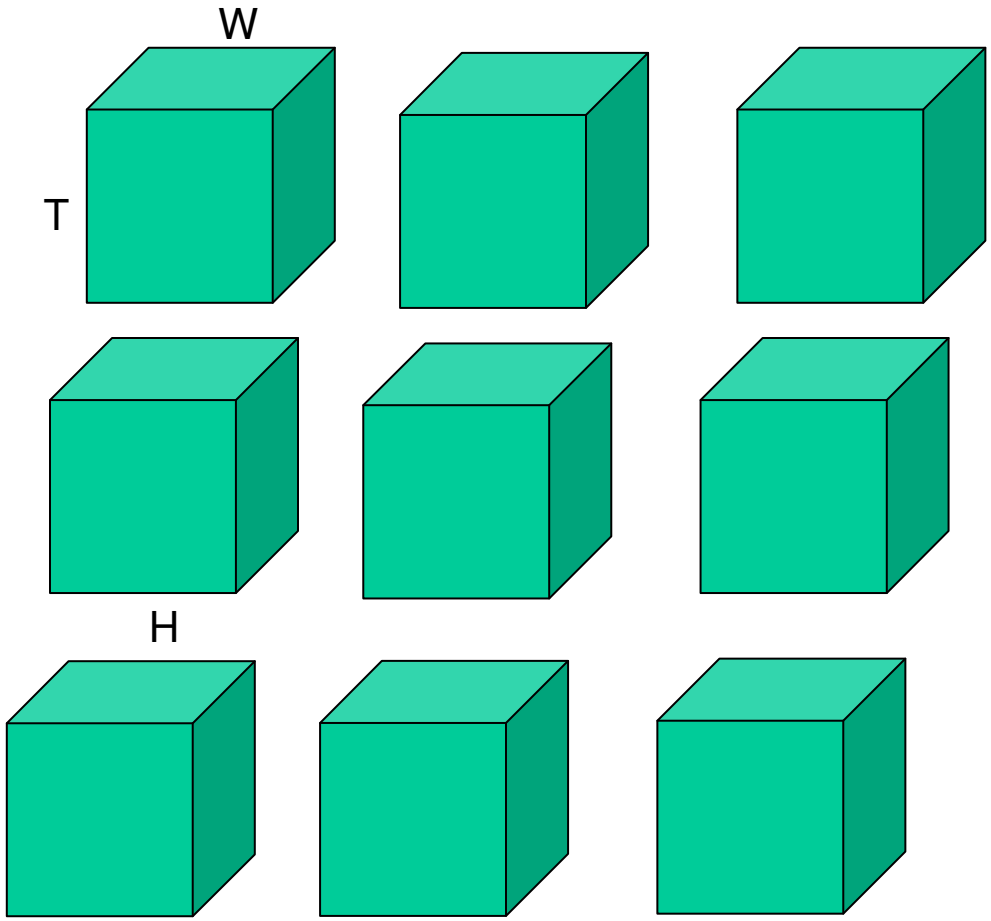


Manhattan Resistance test structure

Arora et al IEEE Tran SM 2005

1. **Wire resistance is strongly influenced by local environment (local lines width and spacing) and is pronounced for line width less than 0.15µm.**
2. **Influence of global density in area ±50µm around the line is the second order effect.**
3. **Resistance variations disappear when resistance is plotted as a function of the extracted effective width (continuous line)**

Variability



Need to Address variability in wire:

1. Thickness T due to CMP
2. Width W due to Litho
3. Height H due to Etch

- Variation across the length need also be specified

Numerical Methods

SILTERRA
Delivering Success.

- Tools that solves Maxwell equations numerically to calculate fields and potentials, hence R, L and C, in an interconnect is called the Field Solver (Numerical Methods).
  - Integral, also known as Boundary-Element method (BEM), method of moments, Green function method) -Fastcap (MIT)
  - Finite Difference (FD) – Raphel (Synopsis)
  - Finite Element (FE) – Maxwell (Ansoft)
  - Monte Carlo (including Random Walk) – QuickCap (Magma), Rapid3d (Synopsis)
- FE and FD methods together are also know as Differential methods.
- Other methods:
  - **Analytical**
    - Conformal transformation
    - Empirical

Today's Conventional (Modeling) Approach

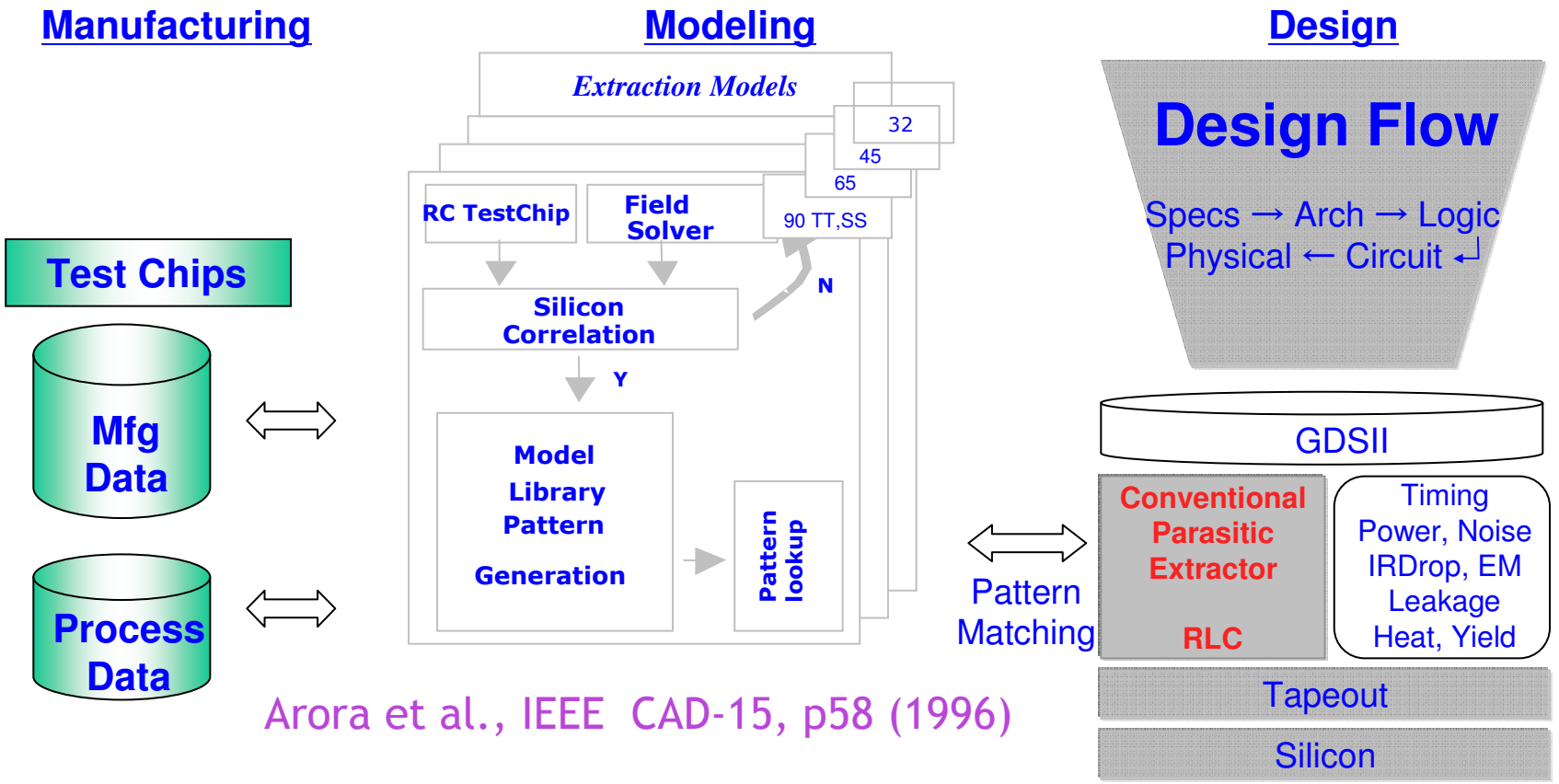
SILTERRA Delivering Success.

- Uses a predefined library of cells with capacitance models generated for each technology node and process corners using predefined wire width (W), wire thickness (T), and dielectric thickness (H) etc. Library must be regenerated, if any process parameters change, which takes at least a day to complete.
- Even for advanced modeling approaches, being Quasi-3D, accuracy is poor particularly for coupling capacitances. The situation is worse for sub-45nm processes.
- Existing EDA tools (QRC, StarRC, xCalibre) differ from each other only in the number of various types (shapes) of cells in the library and their models.
- IC manufacturing/process effects (DFM) are *included in* a convoluted and approximate way causing error in sub-45nm process nodes.
- Uncertainty in accuracy needs large guard banding resulting in poor chip performance and yield.

Conventional Approach

Breaks down at 65/45nm designs

SILTERRA
Delivering Success.



Arora et al., IEEE CAD-15, p58 (1996)

Inaccurate Modeling and Pattern-matching Approximations Leads to Poor Yield and Manufacturability of Designs

Challenges at Sub-45nm and below

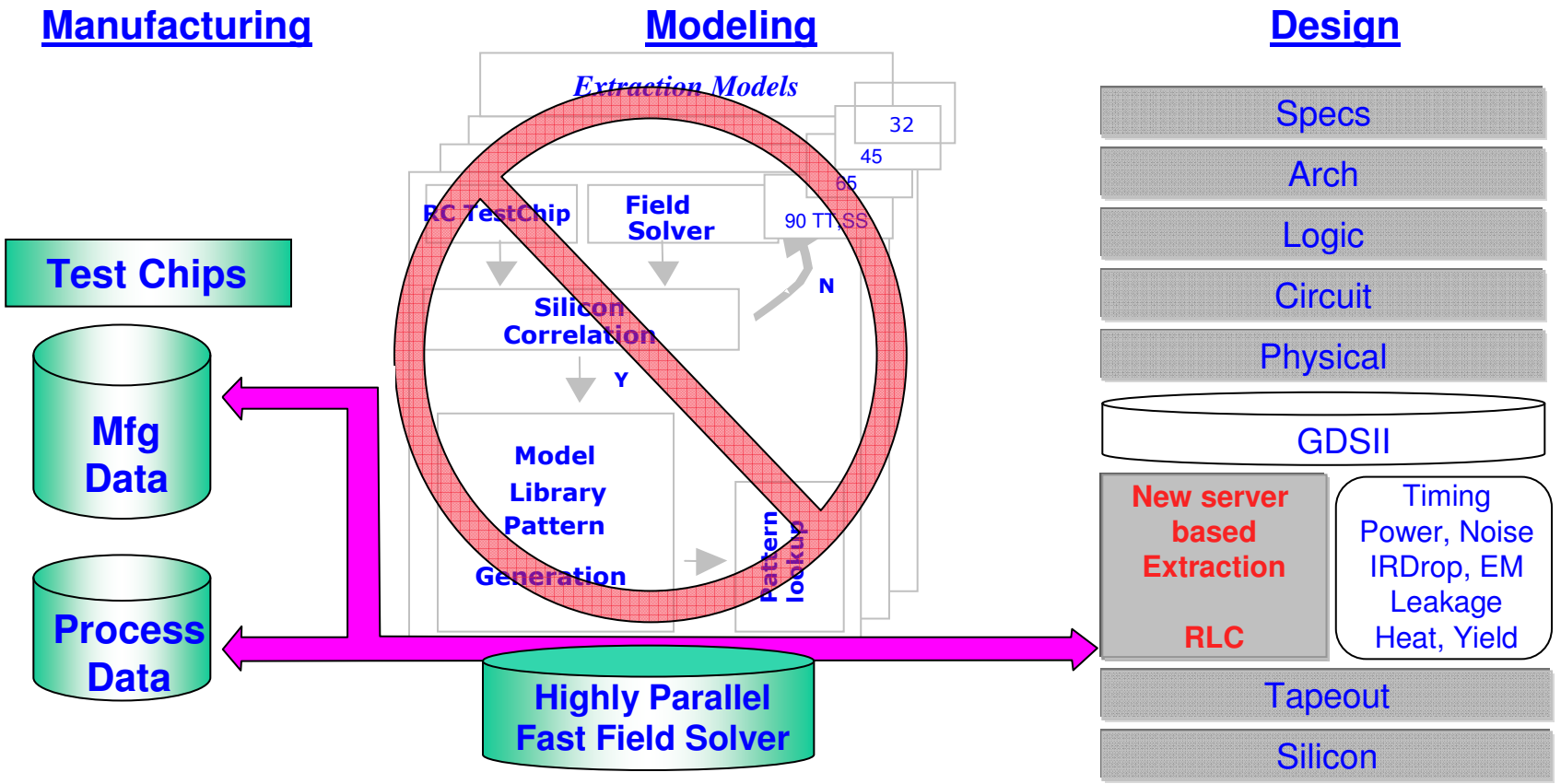
SILTERRA
Delivering *Success.*

- **Transition to sub 45 nm requires direct use of DFM aware 3D Field solver that enables**
  - **Accurate coupling capacitance calculation for each net**
  - **Avoid regeneration of costly and time consuming model libraries**
  - **Reduce guard banding**
  - **Improve performance and yield**
- **Conventional method applies DFM effects inaccurately, after the effect as capacitance is precalculated without DFM effect.**
- **Conventional wisdom**
  - **Field solver based extraction is extremely slow and can only be employed for critical nets.**
  - **QuickCap, a Monte Carlo (MC) based 3D capacitance extractor, is regarded as the Gold standard for capacitance extraction but its use limited to critical cells and small blocks net extraction**

New Solver Based Approach – Full 3D

Must-have for 40/32/22nm designs

SILTERRA Delivering Success.



Accurate Parasitics with Integrated Statistical Sigma and PVT Variance Eliminates Modeling and Pattern-matching Approximations for High Yield and Manufacturability

Advantages of Monte Carlo

SILTERRA
Delivering Success.

- For full chip extraction Monte Carlo (MC) algorithm has advantages over other field solvers:
  - Unbiased estimate of average E strength (C!) with any desired accuracy
  - Unlike BEM, FE or FD based field solvers that requires meshing of the wire/medium, MC does not require meshing, so is inherently faster, hence can handle larger designs.
  - **Inherently parallelizable**
    - **Enabling fast computation, using multiple processors that results in almost linear speed improvements, or multiple machines**
    - **Hardware acceleration through Field Programmable Gate Array (FPGA) or Graphical Processor Units (GPU) that could exploit parallelization giving speed comparable to Model based extractors like, Star-RCX, QRC, xCalibre**
    - **FPGA, though fast, is difficult to implement and debug with VHDL design language. GPU has lower cost ($500 off-the-shelf board) and ease of implementation (C++), we have developed GPU solution for hardware acceleration.**

Capacitance is integrated E

SILTERRA
Delivering *Success.*

- $$C = Q\,V$$

  Fix V at 1V, and find Q via Gauss's Law.

- Gauss's Law relates capacitance (total charge on a conductor) to integrated electric field around the conductor

  $$C = \int E \cdot ds$$

- Green's function relates E strength (derivative of potential) to potentials over a boundary
- We can measure electric field by random point samples of potentials.
- Random sampling gives an unbiased estimate of average E strength (C)
- Potentials themselves can be computed by random walk.

Potential Estimates

SILTERRA
Delivering *Success*.

- For potentials, the Mean Value Theorem of electrostatics states:
  The potential of any point is equal to the mean potential of all points on a sphere surrounding that point.
- This theorem can be extended to weighted samples over any closed geometry, like a cube.
- Therefore a *random* sample of this cube is a noisy but unbiased estimate of the potential. We can "hop" to a new point. We repeat this until a known potential is reached. This is the so-called Monte Carlo Floating Random Walk method of estimating capacitance.
- Based on these concepts we developed MC based solver called iSPACE (Interconnect **S**tatistical **Para**sitic **E**xtractor) for full chip C extraction.

Monte Carlo method of C computation

The Dirichlet boundary value problem for a potential distribution in a volume *V* with a boundary S is

$$\begin{cases} \nabla^2 \varphi(\mathbf{r}) = 0 & (r \in V) \\ \varphi(\mathbf{r}) = \mu(r) & (r \in S) \end{cases} \quad (1)$$

The potential $\varphi(\mathbf{r})$ at the point *r* inside the volume can be written as

$$\varphi(\mathbf{r}) = \oint G_S(r, \theta) \cdot \mu(\theta) dS \quad (2)$$

where $G_S(r, \theta)$ is a Green function that is satisfying the following condition

$$\oint G_S(r, \theta) dS = 1 \quad (3)$$

The electrical field ***E(r)*** at the point *r* can be written as

$$\mathbf{E}(\mathbf{r}) = -\nabla\varphi(\mathbf{r}) = -\oint \nabla G_S(r, \theta) \cdot \mu(\theta) dS \quad (4)$$

If one takes a cube V_0 located inside the volume V, the equation (4) can be rewritten as

$$\mathbf{E}(\mathbf{r}_0) = -\oint \nabla G_S(r_0, \theta) \cdot f_0(\theta) dS_0 \quad (5)$$

Where $f_0(\theta)$ is the potential at point θ on the surface of the cube V_0.

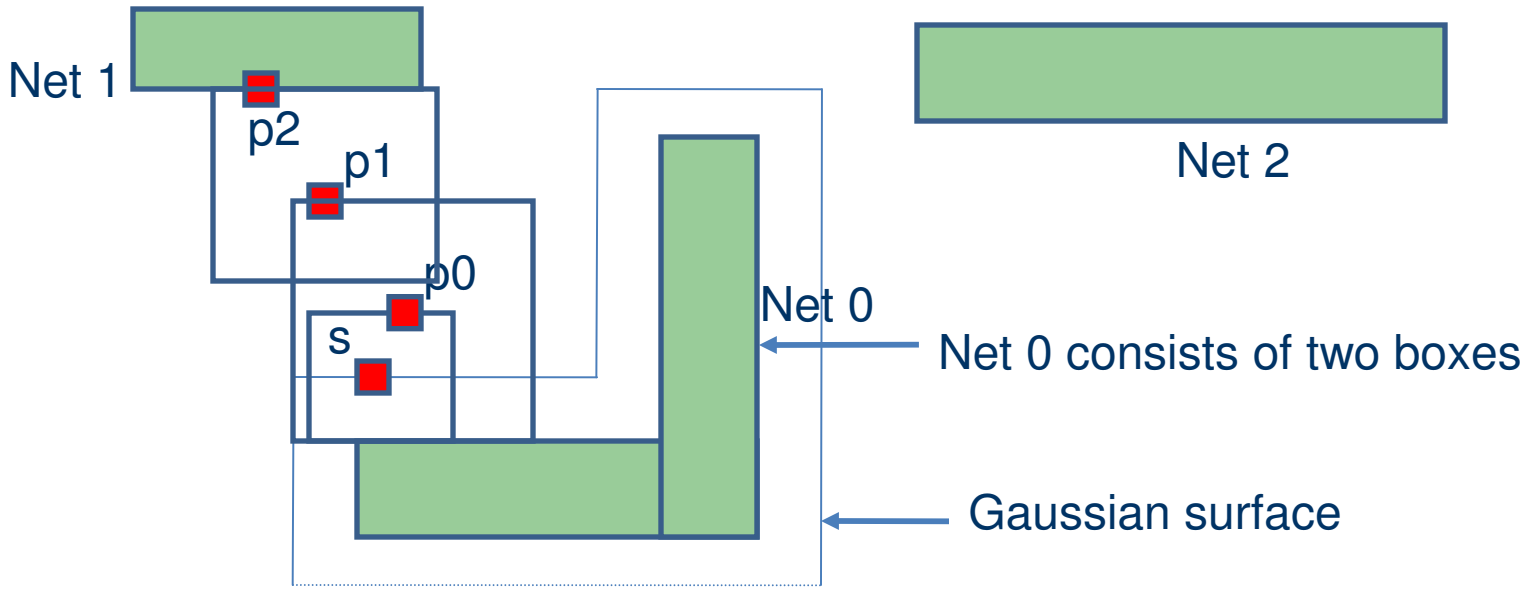
Substituting the equation (2) into (5) N times (a random walk) and assuming that the potential on the very small last N cube surface is equal 1 and taking into account equation (3) the equation (5) can be written as

$$\mathbf{E}(\mathbf{r}_0) = -\oint \nabla G_S(r_0, \theta) \cdot \oint G_{S1}(\theta, \theta_1) \cdot \ldots \cdot \oint G_{SN}(\theta_{N-1}, \theta_N) dS_N \ldots dS_1 dS_0 = -\oint \nabla G_S(r_0, \theta) dS_0$$

Floating Random Walk Algorithm for Extracting Net0.

SILTERRA Delivering Success.

- Wires in a design are simply collection of 3D boxes.
- A net is a collection of electrically connected boxes.
- A Gaussian surface is drawn around a net whose C to be calculated
- Start with a random point (s) on the Gaussian surface of net0.
- Construct a Maximum Empty Cube (MEC) at **s**. Pick a random point **p0** on MEC. The algorithm starts at **p0** and keeps hopping (picking the next point according to a distribution) till the next point is incident on some Conductor. The Hops start at **p0** and terminate at **p2**.

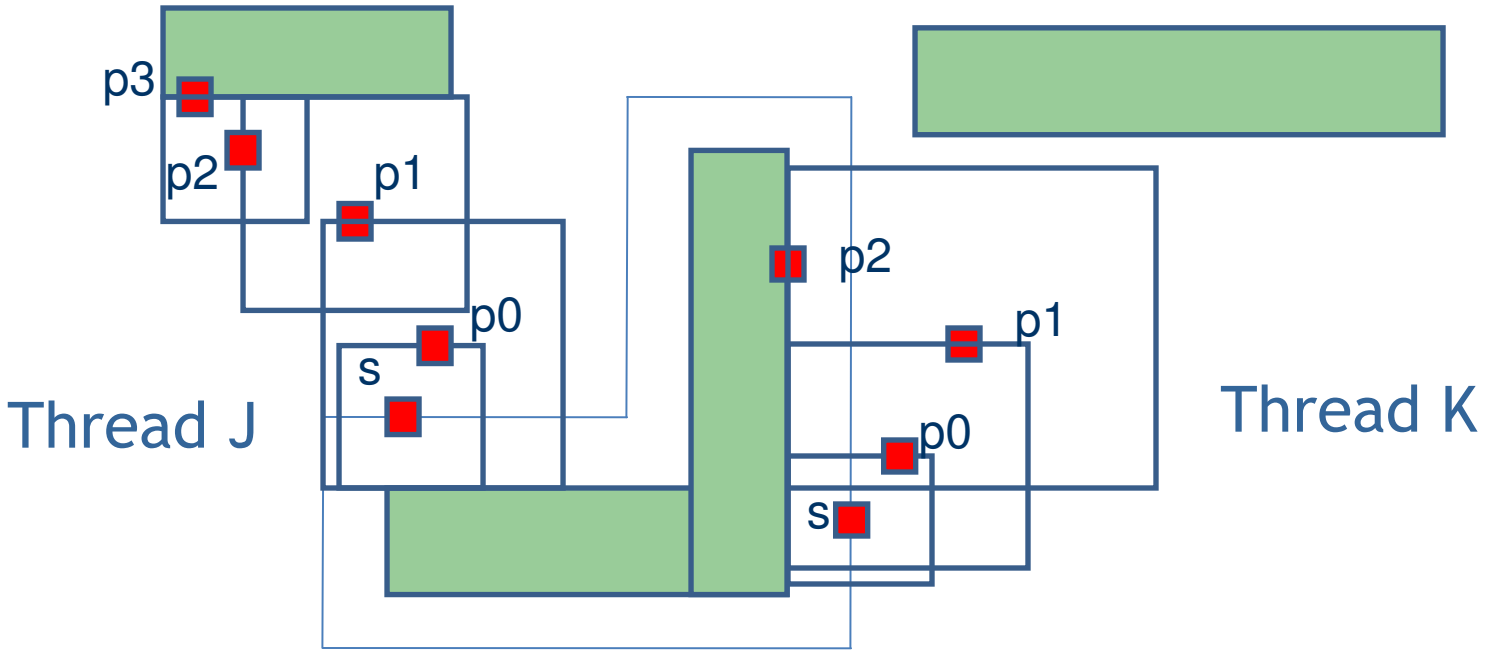


Structure of a walk: **thread J**.

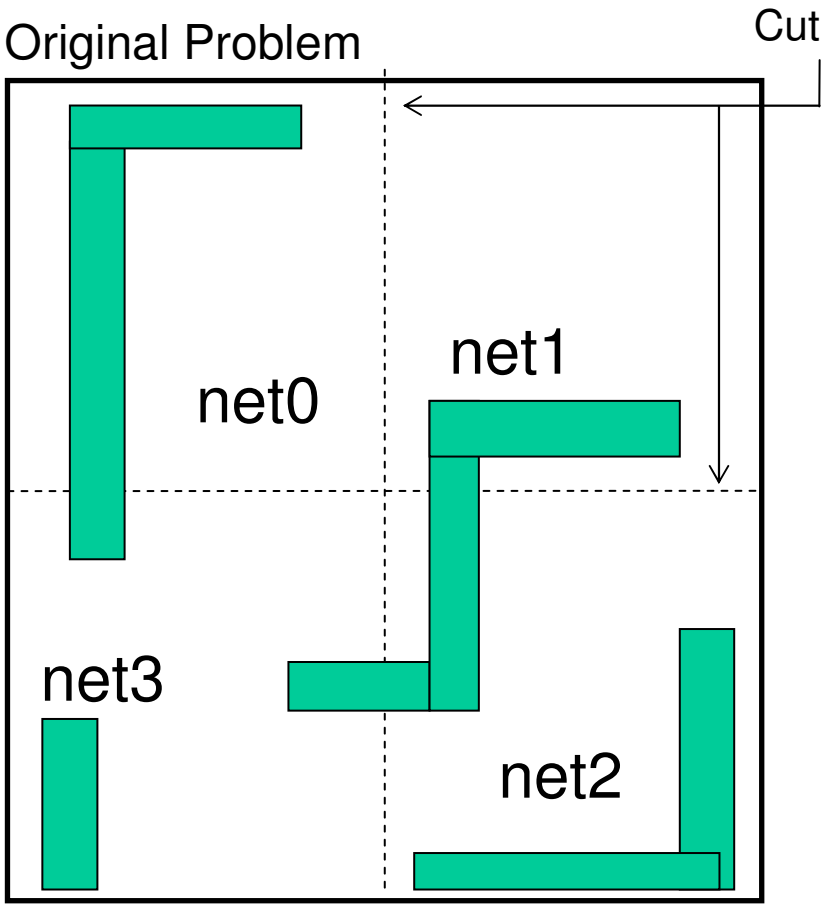
SILTERRA
Delivering Success.

To compute P_{i+1} from the previous point P_i, we need access to:

1. Layout 3D representation for calculating max cube.
2. Dielectric stack
3. Thread K performs walk, which is different compared to Thread J



Extracting Large Designs using ispace

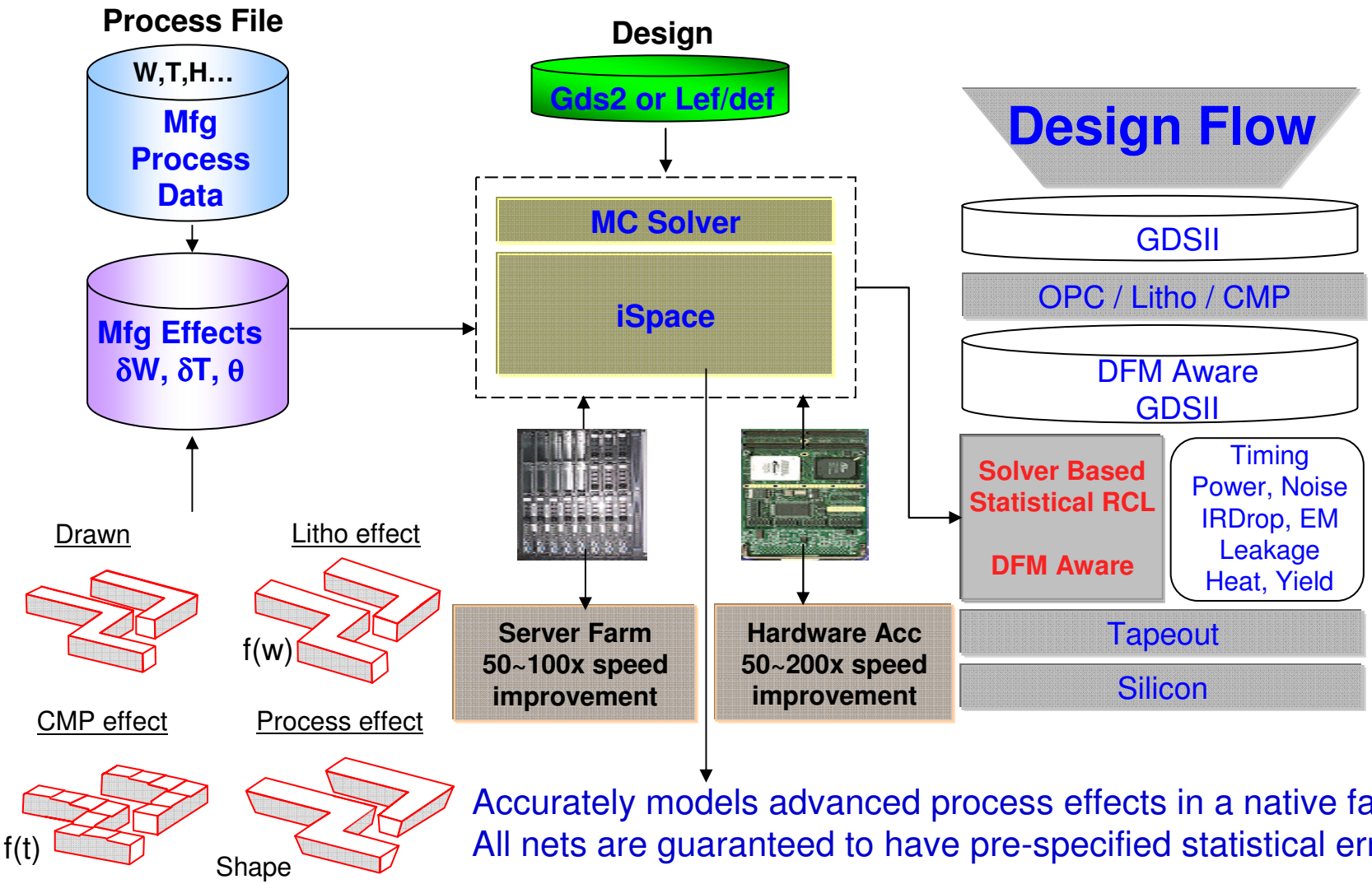


- ispace can extract very large designs through distributed computing.
- Original design is carved into tiles, with each tile processed independently on a separate machine.
- The results from individual tiles are stitched back to obtain global results.
- Tiling done using the notion of HALO regions. No impact on accuracy from tiling.

iSpace - Parasitic Extractor

Sign-off Tool 45/32/22nm

SILTERRA
Delivering Success.

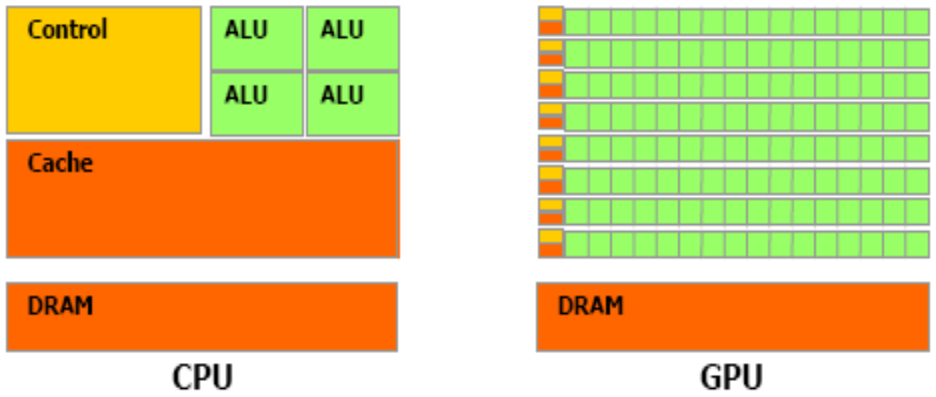


Accurately models advanced process effects in a native fashion
All nets are guaranteed to have pre-specified statistical error

Hardware Acceleration - CPU versus GPU

SILTERRA Delivering *Success*.

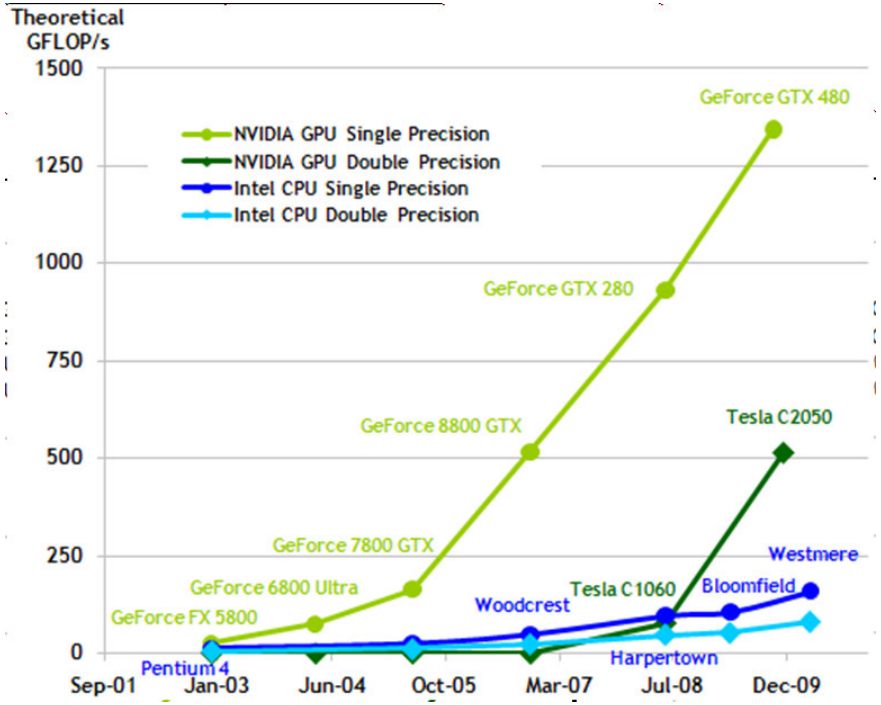
- FRW walk algorithm is proven and effective in CPU
- Using multiple cores and/multiple machine (Form) compute *speed* is still an issue.
- To achieve speed comparable to model based approach, hardware acceleration through GPU is used
- CPU designed for **lowest latency** compute
  - Memory speed limited, so lots of cach
- GPU designed for **highest throughput**
  - Lots of compute power, with high latency



Future GPU vs CPU performance

SILTERRA Delivering *Success.*

- GPU performance is improving faster than CPU's modest gains
- GPU development driven and financed by **gaming** computational demands. Billions of GPUs sold for gamers!
- Future GPUs will have even higher speed ratio over future CPUs



GPU Advantages

SILTERRA
Delivering Success.

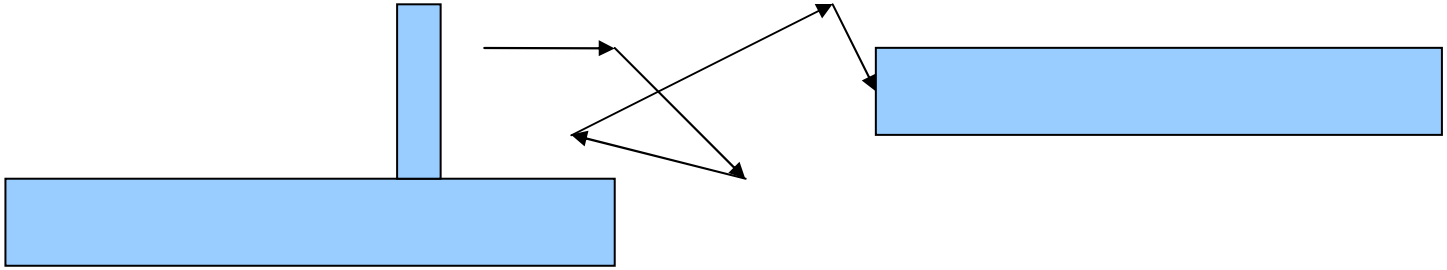
- One GPU = $500
  - 512 cores!
- Extraction scales linearly with multiple GPUs
- Host PC's CPU is idle while GPU works!
- $1500 PC can host 3 GPUs.
- Even many laptops have fast GPUs (yes, they can do capacitance extraction!)

$3,000 3-GPU tower used for ispace

CPU Floating Random Walk

SILTERRA
Delivering *Success*.

- Floating Random Walk applied one walker at a time

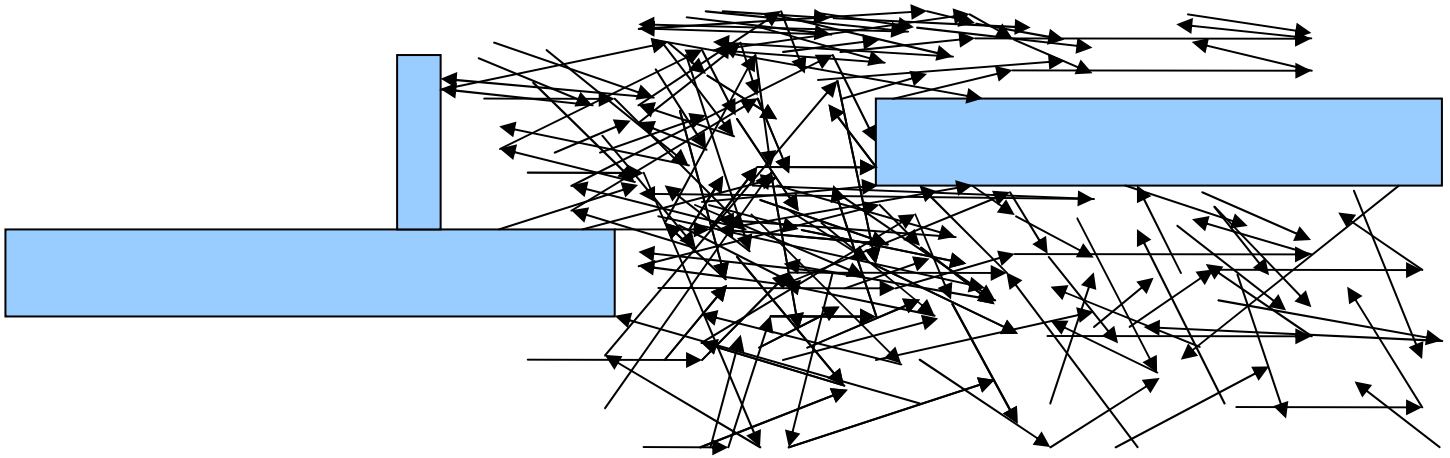


- CPU creates and simulates each walker in turn
- Repeated **millions** of times sequentially
- Multicore CPUs run 2-8 independent threads
  - No intercommunication needed

GPU Floating Random Walk

SILTERRA
Delivering *Success*.

- Same FRW algorithm: just with **20,000** walkers **at once.**



- GPU has **512** simple cores on one chip
- Unlike CPU, cores coordinate with each other
- Local box geometry is shared with all cores

Porting Algorithms to GPU

SILTERRA
Delivering Success.

- GPU isn't a big CPU, so code isn't compatable
- CPU Algorithms have been modified, but mostly rewritten for GPU implementtaion.
- Still programmed in C, with GPU specific extensions
- CUDA language on NVIDIA GPUs most efficient, and our target tool.
- OpenCL is an alternative language, similar to CUDA. (We don't use it though!)

GPU Algorithm details

SILTERRA
Delivering Success.

- Layout boxes stored in 2D voxels
- Cores create 1,000-25,000 walkers
- Voxel with highest number of walkers chosen
- Voxel is loaded, cores compute walkers inside that cell until they've left the cell or finished
- Repeat until all walkers have finished.

ispace Comparison Run time

SILTERRA
Delivering Success.

ispace runtimes from 4 representative designs at 40/45 nm process nodes

Design	#Net	2 GPU ($1K) Runtime (minutes)	CPU 1 Thread (minutes)	CPU 8 Threads (minutes)	Confidence
A (GDS, NO TILING, 40nm)	154K	25.7	3058	446	1 %
B (GDS, NO TILING, 40nm)	238K	40.5	4413	643	1 %
C (LEF/DEF, TILED, 45nm)	323K	33.5	3654	542	1 %
D (GDS, TILED, 40nm)	883K	245.2	11032	1471	1 %

Conclusions

SILTERRA
Delivering *Success.*

- A *manufacturability and yield aware 3D field solver based statistical parasitic extractor that is*
  - **accurate as any other industry standard 3D field solver, but has higher capacity - can run large blocks of 200K nets in few hours on a single processor with less than 2GB memory.**
- **Easy to Adopt and use**
  - ***Modeling independent***: No need for generating lengthy time consuming technology files. No need to maintain models for various process nodes.
- **Yield Aware**
  - ***Provides 'sigma' for each net to reflect process variations, enabling*** statistical timing analysis in a single run. This eliminates the need to generate worst-case corners.
- **Inherently Parallelizable**
  - Fast turnaround time using multiple processors that result in linear speed improvements.
- **Fast Performance – speed comparable to modeling based approach** musing ***low cost hardware acceleration (*off the shelf GPU plugged into PC parallel port *)***

Publication

SILTERRA
Delivering *Success*.

- Work reported here has been submitted for possible publication in IEEE Trans ICCAD on Jan 20, 2012
- Others who contributed
  - Steve Worley
  - Rimma Pirogova
  - Nataraj Akkiraju

Thank You