Fast Algorithms for Thermal Analysis of 3-D VLSI Chips

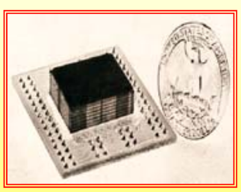
Prof. Pinaki Mazumder
GSRA: Baohua Wang
The University of Michigan
Ann Arbor, MI 48109-2122

International Symposium on Circuits and Systems

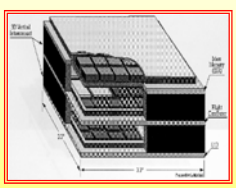
Earlier Generation 3-D System Integration

In Mission Critical (Space and Military) Applications
High Performance and Light Weight are Paramount

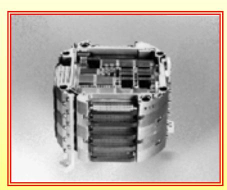
Examples of Earlier Generation 3-D Wafer Scale Integration



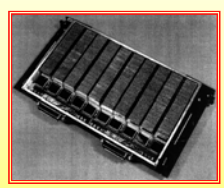
Harris Semiconductors 3-D Memory Module



Jet Propulsion Laboratory Advanced Flight Computer (AFC)



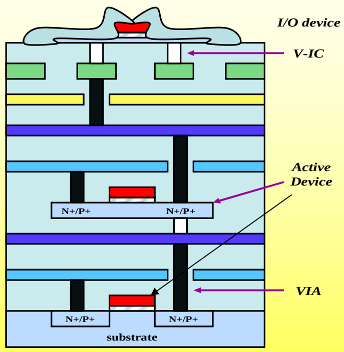
Raytheon Aladdin Parallel Processors to the Night Vision Electronic Sensors Directorate (NVESD)



Texas Instruments High density 1.2 Giga bit solid state recorder

3-D VLSI Chip: Problems and Challenges

Goal: To Design Commercial High-Performance, Low-Power Hand-Held Systems



3-D VLSI System on a Multi-Active Layered Chip (Advanced SoC)

Problems and Challenges

1. **Degradation of Reliability due to Thermal Rundowns**
2. **Identification of Hot Spots in Active Layers in 3-D Chips**
3. **Avoidance of Hot Spots by Thermal-aware Block Placement**
4. **Fast computation of Thermal maps due to power dissipation in multiple active layers of transistors and interconnects**
5. **Development of appropriate Heat Sinking Technology**

Key Requirement: Thermal Engine having High Accuracy v. Low Computation Time

Thermal Problem Definition

- **Time Variant Thermal Equation**

$$\nabla^2 T(x,y,z,t) + \frac{1}{\alpha(x,y,z)} g(x,y,z,t) = \frac{1}{\alpha(x,y,z)} \frac{\partial T(x,y,z,t)}{\partial t}$$

Boundary Condition: $k(x,y,z)\frac{\partial T(x,y,z,t)}{\partial n_i} + hT(x,y,z,t) = f_i(x,y,z)$

- **Steady-State Thermal Analysis**

$$\nabla \cdot (\nabla T(x,y,z)) = -\frac{g(x,y,z)}{k(z)}$$

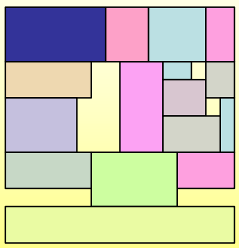
Poisson's Eqn.

$$\nabla^2 \mathbf{V} = -\rho / \varepsilon$$

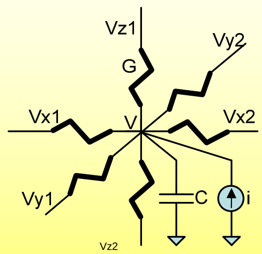
- T: temperature [K], k: thermal conductivity [W/(mK)], g: power density [W/m^3], α: thermal diffusivity[m^2/s], h: heat transfer coefficient. $\alpha = k/\rho c$: ρ: density [kg/m^3] and c: specific heat [J].

Equivalent Thermal Network Approach

- **Modeling Procedure for Circuit Network Method**



Chip Structure



Equiv. Circuit Node

$$Elec.: C\frac{dV}{dt} = i + G\left[\sum_{m=x,y,z}(V_{m1}+V_{m2}-2V)\right]$$

$$= i + G\nabla^2 V$$

$$Thermal: \rho c\frac{\partial T}{dt} = g + k\nabla^2 T$$

$$T \equiv V, \rho c \equiv C, g(x,y,z,t) \equiv i, k(x,y,z) \equiv G$$

Circuit System v. Thermal System

- **Electric Current (I) ←→ Heat flow (Q); Voltage (V) ←→ Temperature (T); Electrical Resistance (R)←→ Thermal Resistance (Θ); Electrical Conductivity (G) ←→ Thermal Conductivity (κ)**
- **Model Order Reduction for the Resulting Equivalent Thermal Network using Krylov subspace method, Arnoldi algorithm, or PVL**

Thermal Analysis Methods

- **Two most efficient approaches:**
  - **Reduced Order Modeling of Equivalent Thermal Network for Chip Structures**
  - **Free Space Green's Function Method**

Free-Space Green's Function Approach

- Temperature can be obtained by spatial and temporal convolution of heat sources with Green's function since in reality the present type of thermal analysis leads to a linear system

$$T(x,y,z,t)=\int_0^t\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}G(x,y,z,t\,|\,x',y',z',\tau)g(x',y',z',\tau)dx'dy'dz'd\tau$$

- Green's function is the response of original system to Dirac Delta source, i.e., the transfer function of the original system

$$\nabla^2 G+\frac{1}{\alpha}\delta(x-x',y-y',z-z',t-\tau)=\frac{1}{\alpha}\frac{\partial G}{\partial t}$$

- $G(x,y,z,t\,|\,x',y',z',\tau)$ $\longrightarrow$ can be represented by

$$G(x,t\,|\,x',\tau)\cdot G(y,t\,|\,y',\tau)\cdot G(z,t\,|\,z',\tau)$$

Free-Space Green's Function Approach

- First use Fourier Transform in the spatial domain, then Laplace Transform in the time domain:

$$G(x\,|\,x',s)=\frac{e^{s\tau}}{\sqrt{2\pi\alpha s}}\exp\left[-\sqrt{\frac{s}{a}}|x-x'|\right]$$

$$G(y\,|\,y',s)=\frac{e^{s\tau}}{\sqrt{2\pi\alpha s}}\exp\left[-\sqrt{\frac{s}{a}}|y-y'|\right]$$

- Using <u>image method</u> further to account for the effect of heat insulation at z=0, it can be obtained that

$$G(z\,|\,z',s)=\frac{e^{s\tau}}{\sqrt{2\pi\alpha s}}\left[\exp\left(-\sqrt{\frac{s}{a}}|z-z'|\right)+\exp\left(-\sqrt{\frac{s}{a}}|z+z'|\right)\right]$$

Free-Space Green's Function Approach

$$G(x,a,\tau)=\frac{1}{2}[\mathrm{erf}(\frac{a/2+x}{2\sqrt{\alpha(t-\tau)}})+\mathrm{erf}(\frac{a/2-x}{2\sqrt{\alpha(t-\tau)}})]$$

$$G(y,b,\tau)=\frac{1}{2}[\mathrm{erf}(\frac{b/2+y}{2\sqrt{\alpha(t-\tau)}})+\mathrm{erf}(\frac{b/2-y}{2\sqrt{\alpha(t-\tau)}})]$$

$$G(0,c,\tau)=\mathrm{erf}(\frac{c}{2\sqrt{a(t-\tau)}})$$

PWL approximation of $\mathrm{erf}(x)$

$$\mathrm{erf}(x)\approx\begin{cases}2x/\sqrt{\pi} & \text{if } x\le 0.5\sqrt{\pi}\\ 1 & \text{if } x>0.5\sqrt{\pi}\end{cases}$$

$$\Delta T(x,y,0,t)=\frac{\alpha P_0}{kabc}\int_0^t G(x,a,\tau).G(y,b,\tau).G(0,c,\tau)d\tau$$

$$\Rightarrow \Delta T(x,y,0,t)=\frac{\alpha P_0}{kabc}\int_0^t[\mathrm{erf}(\frac{A_1}{4\sqrt{\alpha\tau}})+\mathrm{erf}(\frac{A_2}{4\sqrt{\alpha\tau}})].[\mathrm{erf}(\frac{B_1}{4\sqrt{\alpha\tau}})+\mathrm{erf}(\frac{B_2}{4\sqrt{\alpha\tau}})].\mathrm{erf}(\frac{C}{4\sqrt{\alpha\tau}})d\tau$$

The Framework for Free-Space Green's Function Approach

- Evaluation of time-domain free-space Green's Function
- Evaluation of Convolution Integral: Approximation of Error Function
- Problems:
  - Cannot handle Multiple active layer heterogeneous heat conduction materials.
  - In reality, it uses time-domain free-space Green's function to analyze the equilibrium states

The Proposed Approach Based on Layered Green's Function in Multilayer 3-D VLSI

- Difficulties
  - Multi-layered Green's function itself is more complex than free space Green's function
  - How to reduce the speed penalty of evaluating Green's function itself
  - How to ensure the fast evaluation of temperature distribution when specific functions and transforms are presented.

Layered Green's Function in Thermal Problem

- Thermal Problem Description $\nabla \cdot (\nabla T(x,y,z)) = -\frac{g(x,y,z)}{k(z)}$
- *K* is only function of *z*. The *x-y* plane can be assumed as infinite. Thus cylindrical coordinate system can be used

$$\frac{\partial^2 G(\rho,z)}{\partial \rho^2} + \frac{1}{\rho}\frac{\partial G(\rho,z)}{\partial \rho} + \frac{\partial G(\rho,z)}{\partial z} = -\frac{\delta(\rho')}{2\pi\rho}\frac{\delta(z')}{k(z)}$$

- Hankel transform $\mathcal{G}(s,z) \equiv \int_0^\infty G(\rho,z)\rho J_0(\rho s)d\rho$ for further simplification. J_0 : zero-order Bessel function Spectral domain representation:

$$\frac{\partial^2 \mathcal{G}(s,z)}{\partial z^2} - s^2\mathcal{G}(s,z) = -\frac{\delta(z')}{2\pi k(z)}$$

- Mapping the infinite layered 3-D space into a 1-D infinite long rod.

Solution Using Transmission Line Analogies

- **Analogies**

Green's Function Formulation Of Temperature Distribution v. Transmission Line Equation

$$\frac{\partial^2 \mathcal{G}(s,z)}{\partial z^2} - s^2 \mathcal{G}(s,z) = -\frac{\delta(z')}{2\pi k(z)}$$

$$\frac{\partial^2 V}{\partial z^2} - r^2 V = -r Z_0 \delta(z')$$

$$r \equiv s, \frac{1}{2\pi}\frac{V(r)}{r} \equiv \mathcal{G}(s), Z_0 \equiv \frac{1}{k(z)}$$

- **Evaluation of Layered Green's Function**
  - **Evaluation of transmission line impedance *Z(s)***
  - **From *Z*(*s*) obtain Green's function by taking inverse Hankel transform**

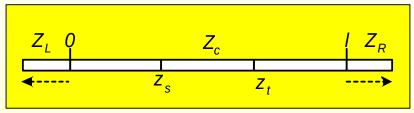
$$G(\rho, z) = \mathcal{H}[\rho, Z(s)] \equiv \frac{1}{2\pi}\int_0^\infty Z(s) J_0(\rho s) ds$$

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- **Properties and Merits**
  - **Leads to fast evaluation of Green's Function**
  - **Leads to fast evaluation of temperature distribution**
  - **Does not incur sampling overhead of Green's Function for interpolation**
  - **Considers both far-field and near-field calculations**

Problems using Interpolation Method

- Evaluation of Z(s) and Green's Function: Interpolation method



  - Given specific *(z_s, z_t)*, Obtain Z(S), function of complex frequency *S*, and Evaluate *G(ρ, Z(S))* and Construct library of *G(ρ, Z(S))* by sampling with respect to *(s, t)* and *ρ*.

  - Problems:

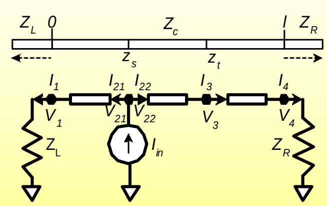
    How to decide the number of sampling points and the position of the sampling points.

    Use of Fast Hankel transform method to calculate *G(ρ, Z(S))* from *Z(S)* is still time-consuming if sampling points are too large.

    Temperature evaluation is costly since numerical integration might involve too many space partitions.

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- Analytical reformulation of *Z(S)* for (*s*, *t*) in the same layer



$$Z(s) = \frac{V_3(s)}{I_{in}(s)} = \sum_{i,j\in\{1,2\}} B_{ij}(s) K^s_{ij}(z_s, l - z_t)$$

- Two parts broken into:
  - Independent of *(s, t)*,

$$B_{ij}(s) = \frac{Z_c[Z_L(s)+(-1)^i Z_c][Z_R(s)+(-1)^j Z_c]}{C(s)}$$
$$C(s) = Z_c[Z_L(s)+Z_R(s)]\cosh[sl] + [Z_L(s)Z_R(s)+Z_c^2]\sinh[sl]$$

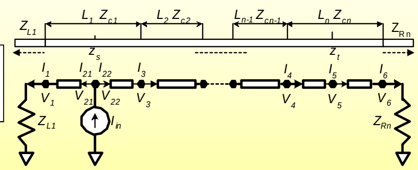
  - Independent of transmission line properties: Kernel

$$K^s_{ij}(p,q) \equiv \frac{\exp\{s[(-1)^i p + (-1)^j q]\}}{4}$$

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- **Analytical reformulation of *Z(S)* for (*s*, *t*) in multiple layers**

$$Z(s)=\frac{V_5(s)}{I_{in}(s)}=\sum_{i,j\in\{1,2\}} D_{ij}(s)K_{ij}^{s}(z_s,l_n-z_t)$$



- **Two parts broken into:**
  - **Independent of *(s, t)***

$$E(s)=F(s)\{Z_{c1}[Z_{L1}(s)+Z_{R1}(s)]\cosh[sl_1]+[Z_{c1}^2+Z_{L1}(s)Z_{R1}(s)]\sinh[sl_1]\}\cdot\{Z_{Rn}(s)\cosh[sl_n]+Z_{cn}(s)\sinh[sl_n]\}$$

$$F(s)=\prod_{2\le k\le n-1}\left(\cosh[sl_k]+\frac{Z_{ck}}{Z_{Lk}}\sinh[sl_k]\right)$$

  - **Independent of transmission line properties: Kernel**

$$K_{ij}^{s}(p,q)\equiv\frac{\exp\{s[(-1)^i p+(-1)^j q)]\}}{4}$$

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- **Summary for Formulations**
  - **Formulations separate the items which are dependent on positions of source and target points from the intrinsic transmission line impedance properties at different sections.**
  - **It avoids sampling of *Z(S)* with respect to *(s, t)* since the transmission impedance properties between different sections can be pre-characterized, either entirely analytically or approximately analytically.**

    Then combine with Kernels to get the complete impedance representation

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- The object is to pursue fast thermal simulation
  Spatial convolution with Green's function to obtain temperature distribution

$$\Delta T = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(x, y, z | x', y', z')\, dens(x, y, z)\, dxdydz$$

- The spatial convolution is usually done by numerical integration of Green's function.

- Problems
  - Accuracy is related to number of spatial partitions.
    Large number of partitions increase the computation time.
  - The effect of self heating cannot be considered accurately due to the singularity of Green's function at that point.

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- Moments matching technique leads to analytical temperature evaluation procedure.
  Representing transmission line impedance by $B_{ij}(s)$ and $D_{ij}(s)$ with exponential functions using modified Prony's algorithm, which is usually used in statistics for estimating the distribution of data.

$$Z(s) \approx \sum_{i,j\in\{1,2\}} K_{ij}^{s}(z_s, l_t - z_t) \sum_{k=1}^{m} c_{ijk} \exp\left[s d_{ijk}\right]$$

$$f(s) \approx \sum_{k=1}^{m} c_k e^{s d_k}$$

- Inverse Hankel Transform can be done analytically.

$$\mathcal{H}[\rho, e^{sd} K_{ij}^{s}(p,q)] = \frac{1}{\sqrt{[d+(-1)^i p + (-1)^j q]^2 + \rho^2}}$$
$$\text{for } d \le (-1)^i p + (-1)^j q$$

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- **We have introduced weighted coefficients to account for the square-root decreasing rate of Bessel function w.r.t. to the radius in order to improve the accuracy in specific regions during the moments matching procedure by using modified Prony's algorithm. For instance, near/far field can be discriminated.**
- **To improve accuracy, the weighted version of modified Prony's algorithm, known as pre-conditioned modified Prony's algorithm, is designed.**
- **The modification is made for the B-matrix as given below**

$$B_{ij} = y^T X_i A X_j^T y - y^T X A X_i^T W^{-2} X_j A X^T y$$
$$A = X^T W^{-2} X$$

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- **Temperature Evaluation**
  - **Superposition principle for such a linear system**

$$\Delta T(x_0, y_0, z_0) = \sum_{hs \in U} dens(hs) \Delta T^{hs}(x_0, y_0, z_0)$$
$$\Delta T^{hs}(x_0, y_0, z_0) = \sum_{i,j \in \{1,2\}} \sum_{d \in d_{ij}} T_{ij}^d$$

  - **Simple implication of integration than general numerical integration**

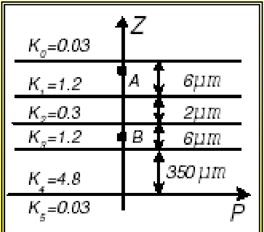
$$T_{ij}^d = I_{i,j}^d(z_1, z_2, l_0 - z_0)$$

$$T_{ij}^d = \begin{cases} I_{i,j}^d(z_1, z_0, l_0 - z_0) \\ +I_{j+1,i}^d(z_0, z_2, z_0 + (-1)^{(j-i)} l_0), & z_1 \le z_0 \le z_2 \\ I_{j+1,i}^d(z_1, z_2, z_0 + (-1)^{(j-i)} l_0), & otherwise \end{cases}$$

$$I_{p,q}^d(z_a, z_b, z_c) = \int_{x_1}^{x_2} \int_{y_1}^{y_2} \int_{z_a}^{z_b} \frac{dx\,dy\,dz}{\sqrt{[d + (-1)^p z + (-1)^q z_c]^2 + (x - x_0)^2 + (y - y_0)^2}}$$

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- **Validation of Semi-analytical Green's function**

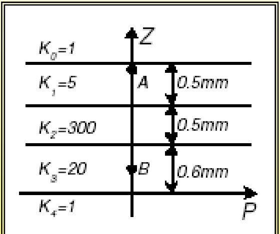


(b) Structure of 4 layers

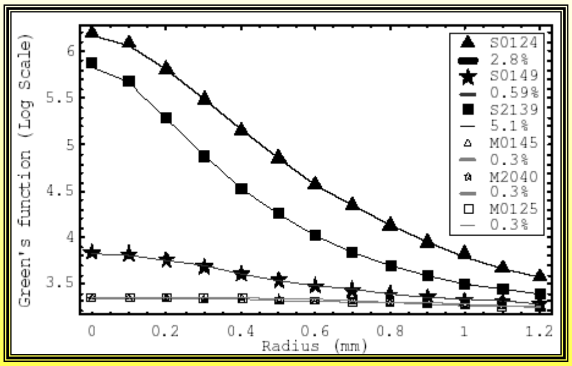
ρ mm	S11	2148	M13	2148	S11	0135	M13	0135
0	70.5	69.4	8.23	8.4	72.2	72.4	10.1	10.4
0.1	67.1	66.1	8.2	8.36	69.6	69.6	10.1	10.8
0.2	59.4	58.6	8.13	8.26	63.0	62.8	10.0	10.2
0.3	51.1	50.6	8.01	8.11	55.2	54.7	9.82	10.0
0.4	43.8	43.6	7.84	7.9	47.6	47.2	9.59	9.73
0.5	37.8	37.6	7.64	7.66	41.1	40.6	9.31	9.38
0.6	32.9	32.7	7.41	7.39	35.6	35.2	8.99	9.0
0.7	28.8	28.6	7.16	7.11	31.0	30.7	8.64	8.6
0.8	25.4	25.2	6.9	6.82	27.2	26.9	8.28	8.2
0.9	22.6	22.3	6.64	6.53	24.1	23.8	7.9	7.79
1.	20.2	19.9	6.37	6.25	21.4	21.1	7.53	7.4

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- **Validation of Semi-analytical Green's function**



(a) Structure of 3 layers



Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- Thermal Simulation Results

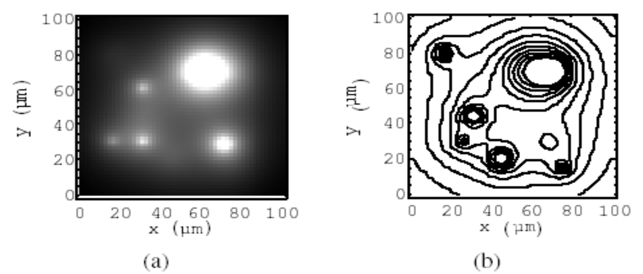


Fig. 5. Temperature distribution in layer 1 and layer 3 for the structure in Fig. 3(b): (a) Plot of temperature intensity in layer 1 at z=3μm; (b) Isothermal contour plot of temperature in layer 3 at z=3μm.

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- **Discussion**
  - **Semi-analytical Green's Function is <u>space-efficient</u> without incurring sampling overhead of Green's Function**
  - **Thermal analysis using semi-analytical Green's Function is <u>time-efficient</u>.**
  - **Self-heating effect due to singularity of Green's Function is addressed by semi-analytical Green's Function.**

Proposed Semi-analytical Green's Function for 3-D VLSI Chips

- **Conclusion and Future Work**
  - **Fast thermal analysis approach using Green's Function is proposed.**
  - **The thermal analysis approach is shown to be efficient.**
  - **Efficient semi-analytical Green's Function technique is proposed. This is shown to be accurate vis a vis conventional numerical techniques.**

Multivariate Normal Distribution Based Statistical Timing Analysis Using Global Projection and Local Expansion

Prof. Pinaki Mazumder
University of Michigan
Ann Arbor, MI 48109

VLSI Design, January 3-7, 2005

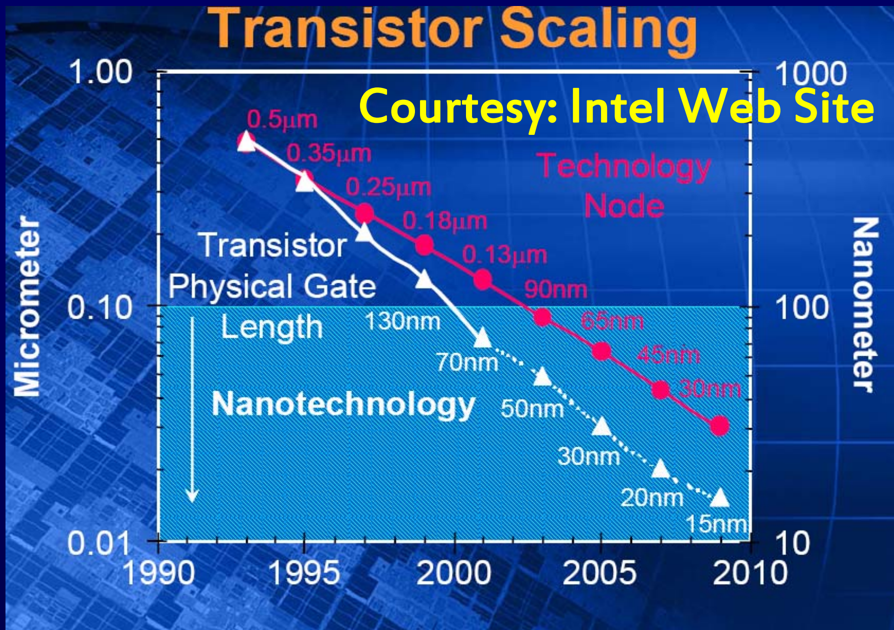
GSRA: Baohua Wang

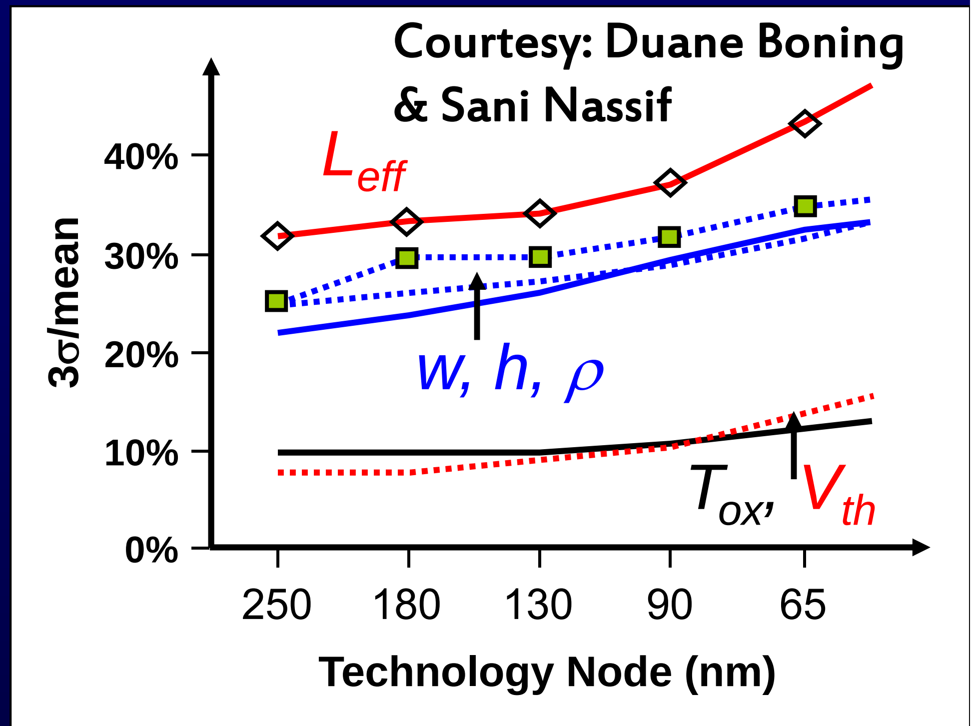
Solutions to Sub-90 nm VLSI Problems Involve Advanced Mathematics

There is no branch of mathematics, however abstract, which may not some day be applied to phenomena of the real world. --- Nikolai Lobatchevsky, N. Rose Mathematical Maxims and Minims

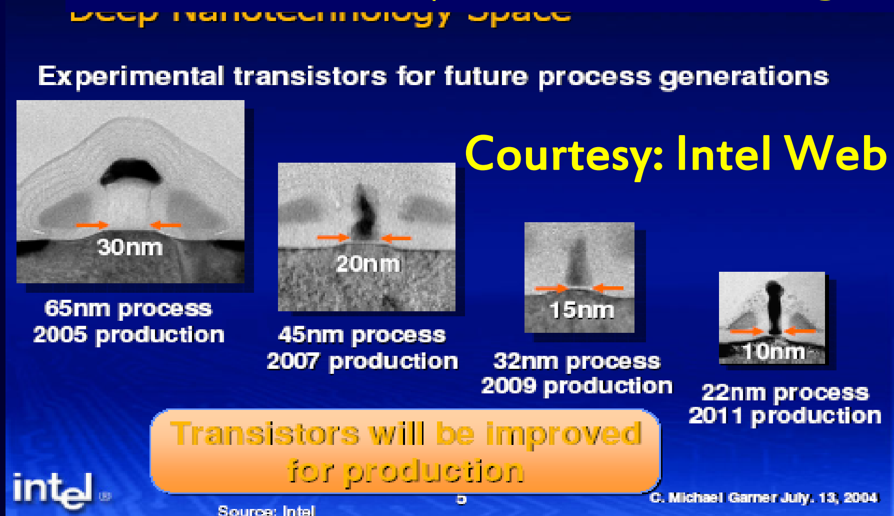
To create a good philosophy you should renounce metaphysics but be a good mathematician.
- Bertrand Russell

Manufacturing Challenges at Sub-90 nm



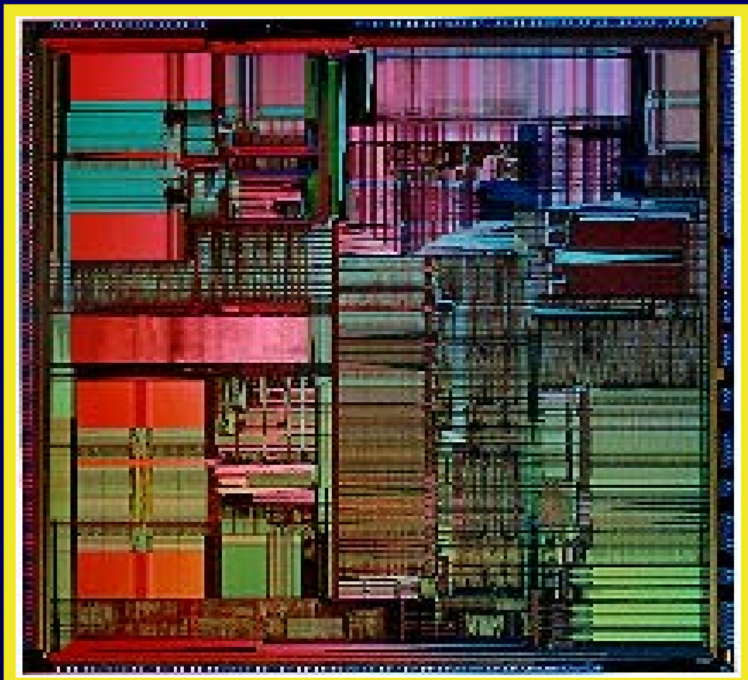


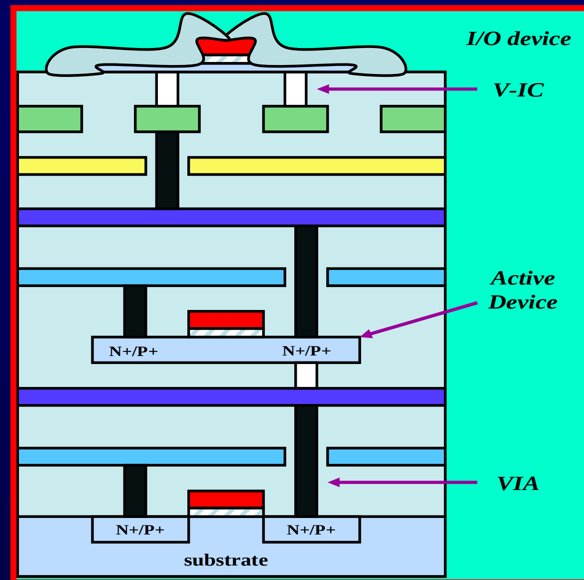
Advancement by Device Shrinking

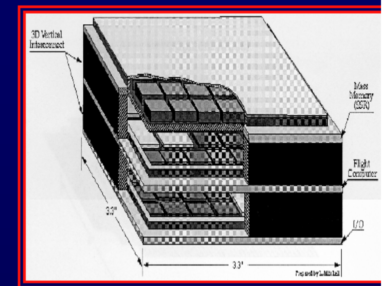


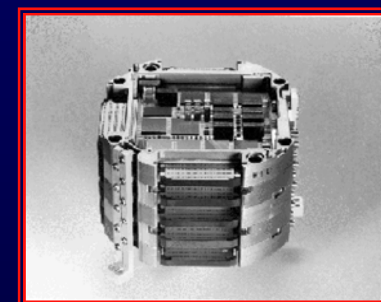
- **Process Variations**
  - **Lithography: L_{gate} (±15%)**
  - **Doping: V_{th} (±30%), L_{eff} (±15%)**
  - **Gate oxide: T_{ox} (±4%)**
  - **Metal definition: line size (±20%)**
- **Circuit Operation**
  - **Power supply: V_{dd} (±10%)**
  - **Crosstalk noise: ΔT_d / T_d (>50%)**
  - **Temperature fluctuation**

Trends in Chip Integration Technology









2-D Lateral Integration

Example: DEC EV Microprocessor

500 Million transistors

Goal: 3.5 billion transistors

3-D Vertical Integration

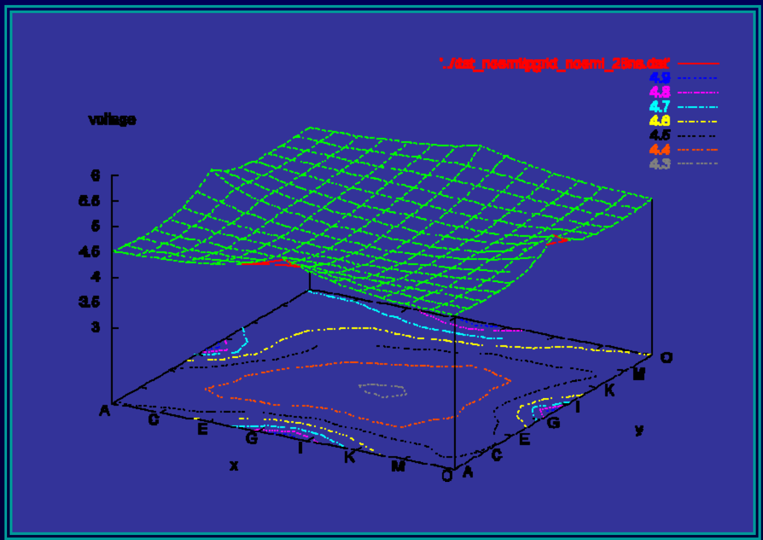
Multiple Active Layers

Future Goal: 200 Active Layers

Environmental and Operating Conditional Variations

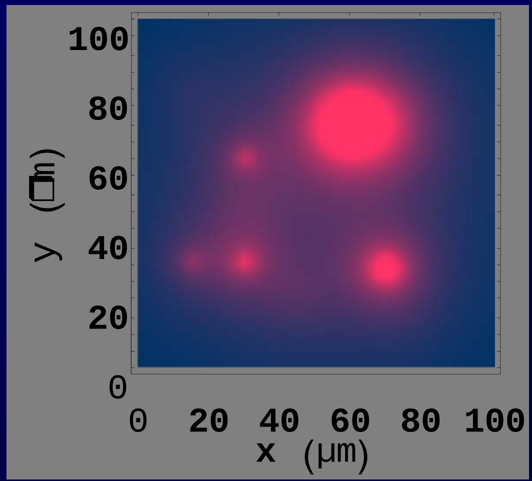
Formidable Mathematical Challenges

Accurate Estimation of Full-Chip IR-drop at Sub 1 V Supply Voltage

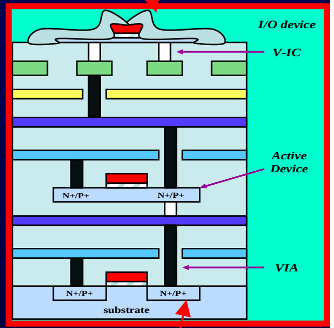


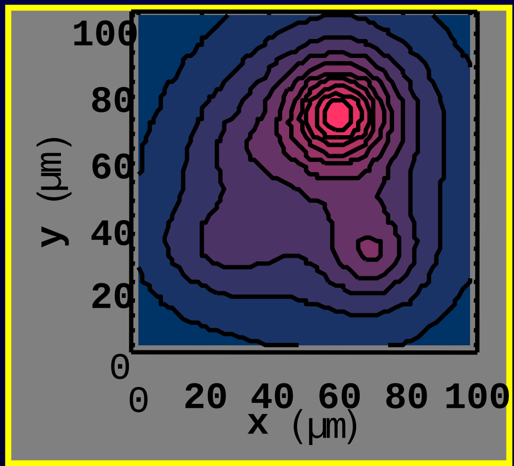
Wang and Mazumder, ISCAS 2004,
Fast Thermal Analysis via Semi-analytic Green's function in Multi-layer Materials

Thermal Distribution in a 3-D Chip by Layered Green's Functions



Thermal Plot on Third layer





Thermal Plot on First layer

Layered Green's Function in 3-D Thermal Modeling

- **Thermal Equation in Cylindrical coordinate system**

$$\boxed{\frac{\partial^2 G(\rho,z)}{\partial \rho^2}+\frac{1}{\rho}\frac{\partial G(\rho,z)}{\partial \rho}+\frac{\partial G(\rho,z)}{\partial z}=-\frac{\delta(\rho')}{2\pi\rho}\frac{\delta(z')}{k(z)}}$$

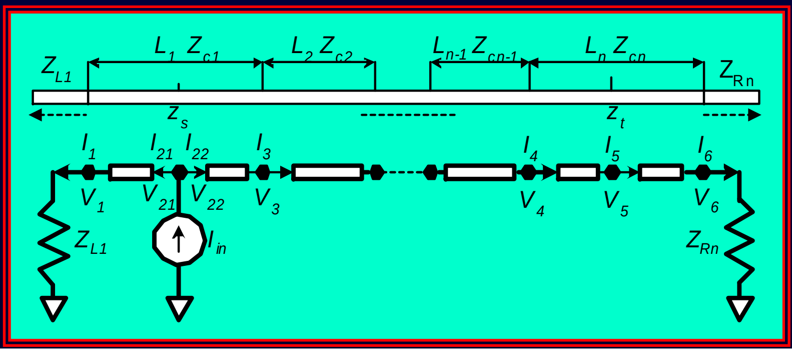
- **Apply Hankel transform**

$$\boxed{\mathcal{G}(s,z)\equiv\int_0^{\infty} G(\rho,z)\rho J_0(\rho s)d\rho}$$

- **Map the infinite layered 3-D space into a 1-D infinite long rod.**

$$\boxed{\frac{\partial^2 \mathcal{G}(s,z)}{\partial z^2}-s^2\mathcal{G}(s,z)=-\frac{\delta(z')}{2\pi k(z)}}$$

$$\boxed{\frac{\partial^2 V}{\partial z^2}-r^2 V=-rZ_0\delta(z')}$$



$$\boxed{Z(s)=\frac{V_5(s)}{I_{in}(s)}=\sum_{i,j\in\{1,2\}} D_{ij}(s)K_{ij}^{s}(z_s,l_n-z_t)}$$

Green's Function based Thermal Mapping in a 3-D VLSI Chip

- Obtain Green's function values and then the steady state temperature profile on each layer

$$
\begin{aligned}
E(s) = & F(s)\,\{Z_{c1}[Z_{L1}(s) + Z_{R1}(s)]\cosh[sl_1] \\
& +[Z_{c1}^2 + Z_{L1}(s)Z_{R1}(s)]\sinh[sl_1]\} \\
& \cdot \{Z_{Rn}(s)\cosh[sl_n] + Z_{cn}(s)\sinh[sl_n]\}
\end{aligned}
$$

$$
F(s) = \prod_{2 \le k \le n-1} \left(\cosh[sl_k] + \frac{Z_{ck}}{Z_{Lk}} \sinh[sl_k] \right)
$$

Moment Matching By Modified Prony's Algorithm

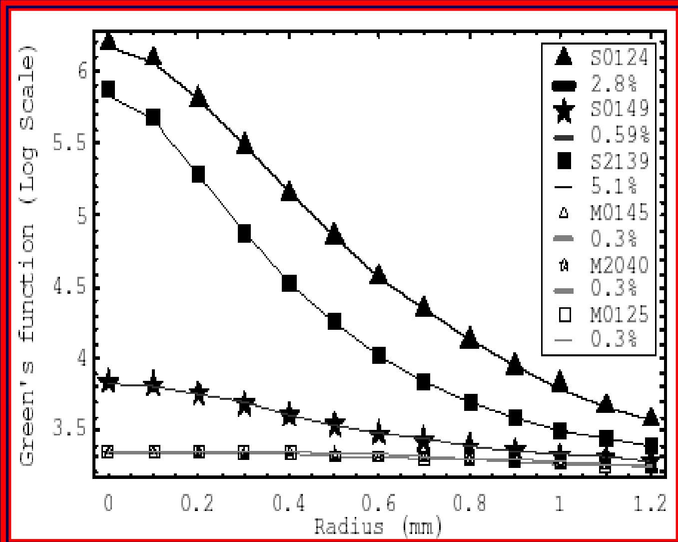
$$
G(\rho, z) = \mathcal{H}[\rho, Z(s)] \equiv \frac{1}{2\pi} \int_0^\infty Z(s) J_0(\rho s)\, ds
$$

$$
\mathcal{H}[\rho, e^{sd} K_{ij}^{s}(p, q)] = \frac{1}{\sqrt{[d+(-1)^i p+(-1)^j q]^2+\rho^2}} \quad \text{for } d \le (-1)^i p + (-1)^j q
$$

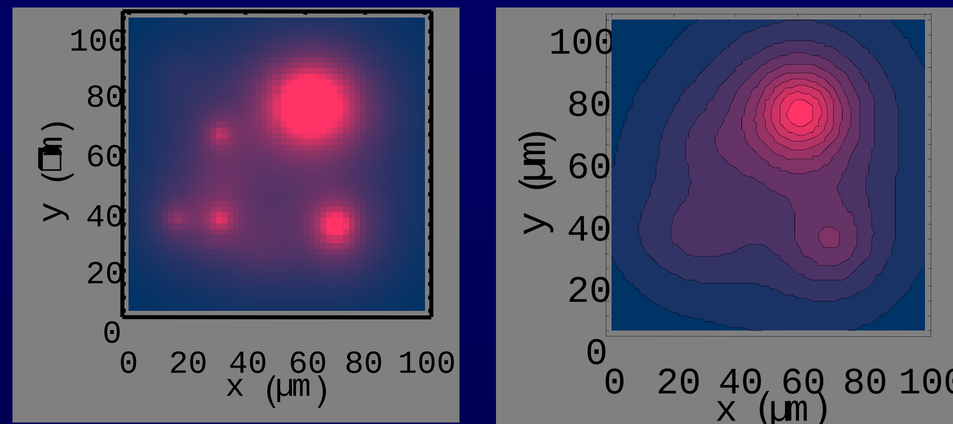
$$
T_{ij}^{d} = I_{i,j}^{d}(z_1, z_2, l_0 - z_0)
$$

$$
I_{p,q}^{d}(z_a, z_b, z_c) = \int_{x_1}^{x_2}\int_{y_1}^{y_2}\int_{z_a}^{z_b} \frac{dx\,dy\,dz}{\sqrt{[d+(-1)^p z+(-1)^q z_c]^2+(x-x_0)^2+(y-y_0)^2}}
$$

Thermal Modeling for 3-D VLSI Chip



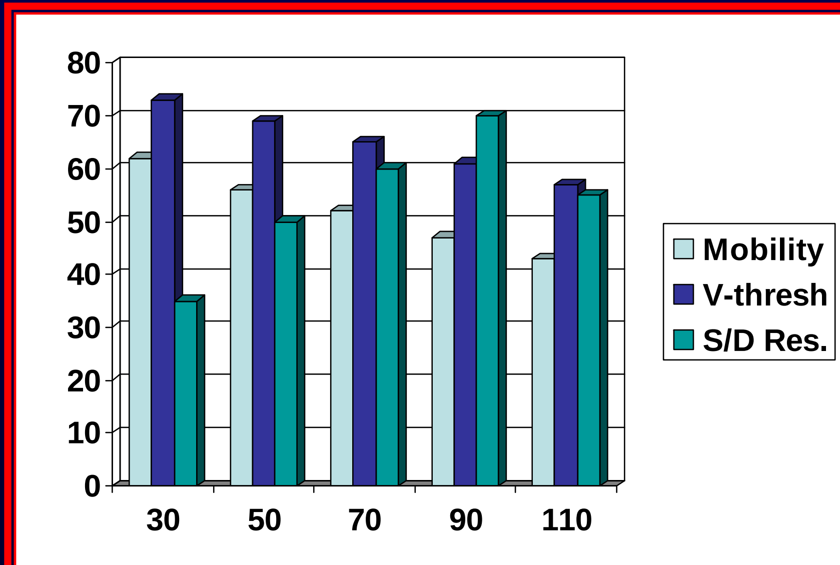
Pre-computed Green's Function



Thermal Maps: Layer 1 and Layer 3

On Circuit Parameters

- Mobility Degradation
- Threshold Voltage Shift
- Source/Drain Sheet Resistance Variation



CMOS Manufacturing Tolerances

Principal Component Analysis (PCA)

BSIM 4.0 has 184 model parameters; however, using PCA it has been found that only 21 Independent Process Parameters account for 82% of variance for the original correlated parameter sets.

Dominant parameters are: Leff (20.5%), Tox (19.1%), K1, Bias coefficient (16.0%), U0, Mobility of PMOS (13.3%), and Weff of NMOS (12.8%).

PCA is used for selecting the basis function in Global Projection of Global Variables

Gate Delay Scattering

Canonical Gate Delay Model:

$$D_M = \mu_M + \sum_{j=1}^{n} \sigma_M^j X_g^j + \sigma_M^l X_M^l$$

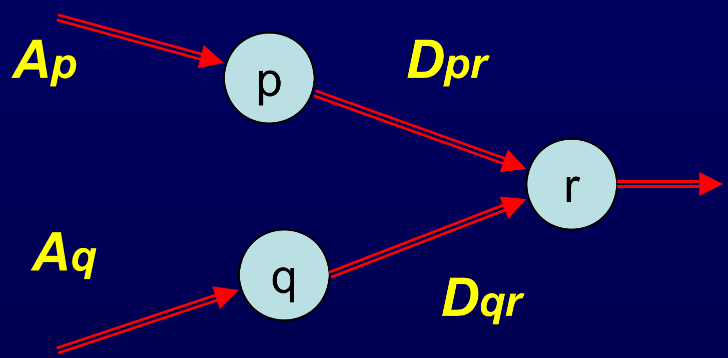
- **Global** Sources of variations *Xg*: Leff, Vdd, Tempr.
- Global Sensitivities are σ_M
- **Local** Gate Delay Variation, X_M

Gate	Node Variation		Global Sensitivities		
	Mean	Std	Length	Vdd	Temp
not	24.0ps	3.4ps	2.8ps	3.3ps	0.7ps
nand2	43.0ps	5.8ps	5.1ps	5.8ps	2.0ps
nand3	50.0ps	6.7ps	5.4ps	6.0ps	2.5ps
nor2	53.0ps	8.0ps	8.0ps	7.9ps	1.8ps
nor3	72.0ps	10.3ps	8.8ps	9.5ps	2.7ps
x(n)or	78.0ps	10.7ps	8.6ps	9.9ps	3.3ps
inpt	0.0ps	0.0ps	0.0ps	0.0ps	0.0ps

Data Source: Li Zheng Zhang, et al., University of Wisconsin

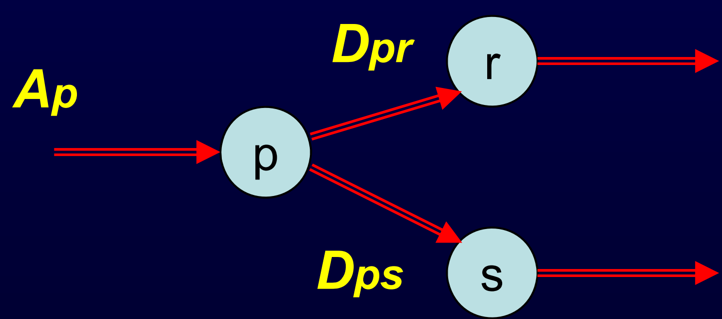
Canonical Delay Models at Fan-in and Fan-out Nodes

Fan-in Node Delay Model:



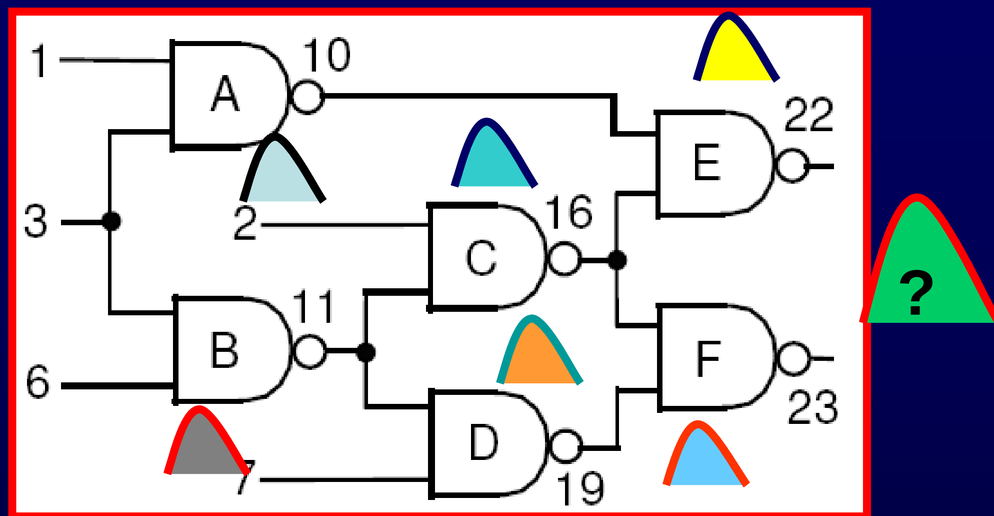
$$\mathrm{Cov}_{merge}(A_p + D_{pr}, A_q + D_{qr})$$
$$= \mathrm{Cov}(A_p, A_q) + \sum_{j=1}^{n} [\sigma_{pr}^{j} \,\mathrm{cov}(A_p, X_g^{j}) +$$
$$\sigma_{pr}^{j} \,\mathrm{cov}(A_q, X_g^{j}) + \sigma_{pr}^{j} \sigma_{qr}^{j}]$$

Fan-out Node Delay Model:



$$\mathrm{Cov}_{split}(A_p + D_{pr}, A_p + D_{ps})$$
$$= \mathrm{Var}(A_p) + \sum_{j=1}^{n} [(\sigma_{pr}^{j} + \sigma_{ps}^{j})$$
$$\mathrm{cov}(A_p, X_g^{j}) + \sigma_{pr}^{j} \sigma_{ps}^{j}]$$

Propagation of Covariance Through Gated Network



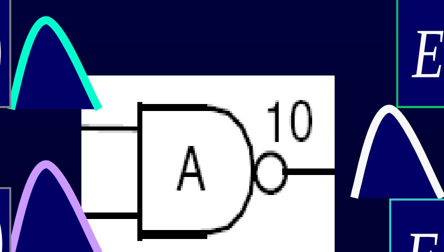
Question:

How to Derive The Output Covariance by propagating Arrival times From inputs To output?

Derivation of <u>Sum</u>, <u>Min</u> and <u>Max</u> Values from Input Variables and Gate Model

$A\left(\mu_A,\Sigma_A\right)$

$B\left(\mu_B,\Sigma_B\right)$



$E\left(\mu_{A+M},\Sigma_{A+M}\right)$

$F\left(\mu_{B+M},\Sigma_{B+M}\right)$

?

$X\left(\mu_{\min(E,F)},\Sigma_{\min(E,F)}\right)$

$Y\left(\mu_{\max(E,F)},\Sigma_{\max(E,F)}\right)$

Multivariate Normal Distribution

Existing techniques:

Can Handle Only Bi-variate Normal Distributions

→ Difficult to compute covariance of Min/Max of more than 2 Variables

Objectives of Wang-Mazumder's Paper:

To handle multiple input gates

To improve accuracy (using Global Projection)

To improve speed of analysis by unifying smaller gates into a multi-input larger block

To handle re-convergence paths problem (using Local Expansion)

To develop mathematical framework for general class of circuits using k-normal multivariate statistics

Theory of Covariance

- Generalized Siegel's formula

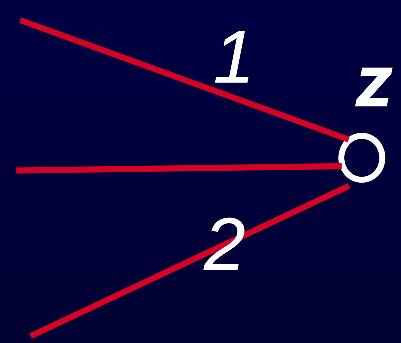
There is no branch of mathematics, however abstract, which may not some day be applied to phenomena of the real world. --- Nikolai Lobatchevsky, N. Rose Mathematical Maxims and Minims

Example:

A_{z1}, arrival time at path 1,

A_{z2}, arrival time at path 2

…



Cov (Y, Max(A_{z1}, A_{z2},…))=

Cov (A_{z1}, Y) P (A_{z1} is the max) +

Cov (A_{z2}, Y) P (A_{z2} is the max) +

…

Theory of Moments

- Mean/variance of the Min/Max of arrival times using derived recursive moment functions.

Here, the w-th Moment of

the Standard k-Normal Ditribution : $m_R^{\Sigma}(x_1^w) \equiv \int_{-\infty}^{\infty} x_1^w \int_{a_2}^{b_2} \cdots \int_{a_k}^{b_k} \phi_k^{\Sigma}(x)dx_1 \ldots dx_k$.

with $\phi_k^B(x_1,\cdots,x_k) = \dfrac{\exp[-0.5X^T B^{-1} X]}{\sqrt{|B|(2\pi)^n}}$, where B is the Covariance/Correlation Matrix.

$$
\begin{aligned}
m_R^{\Sigma}(x_1^w) = (w-1)m_R^{\Sigma}(x_1^{(w-2)}) &+ \\
\sum_{j=2}^{k} \Bigg[\rho_{1j}\phi(a_j) \cdot \sum_{r=0}^{w-1} \binom{\hat{w}}{r} c(a,j,r) m_{\hat{R}_j^a}^{\hat{\Sigma}_j}\left(x_1^{(\hat{w}-r)}\right) \\
- \rho_{1j}\phi(b_j) \cdot \sum_{r=0}^{w-1} \binom{\hat{w}}{r} c(b,j,r) m_{\hat{R}_j^b}^{\hat{\Sigma}_j}\left(x_1^{(\hat{w}-r)}\right) \Bigg]
\end{aligned}
$$

Theory of Moments

- Mean/Variance formulas

$$\mu\left[f(A_{1r},\ldots,A_{kr}\right] = \sum_{i=1}^{k}\left[\mu(A_{ir})\cdot\Phi_{k}^{\Sigma^{Y\hat{i}}}(y\in R_f)+\sigma(A_{ir})\cdot m_{R_f}^{\Sigma^{Y\hat{i}}}(y_1)\right]$$

f → Min or Max function

$$var\left[f(Ar_{1r},\ldots,Ar_{kr}\right] = \sum_{i=1}^{k}\left[[\sigma(A_{ir})]^2\cdot m_{R_f}^{\Sigma^{Y\hat{i}}}(y_1^2) + 2\mu(A_{ir})\sigma(A_{ir})\cdot m_{R_f}^{\Sigma^{Y\hat{i}}}(y_1) + [\mu(A_{ir})]^2\cdot\Phi_{k}^{\Sigma^{Y\hat{i}}}(y\in R_f)\right] - \{\mu[f(A_{1r},\ldots,A_{kr})]\}^2,$$

The General SSTA Algorithm

Arrival Time at a node p,

denoted by A_p, is given by a 4 - tuple :

$$SA_p\{A_p; \mu, \sigma, \mathrm{T}, \perp\}, \text{ where}$$

$$\mathrm{T} \rightarrow \text{Global Projection; } \perp \rightarrow \text{Local Expansion}$$

Global Projection is given as a vector of Cov :

$$\mathrm{T}(A_p) = (\mathrm{Cov}(A_p, X_g^1), \ldots, \mathrm{Cov}(A_p, X_g^n))$$

$$\Rightarrow \text{Covariance between } A_p \text{ and arbitrary } D_{qr} =$$

$$\mathrm{Cov}(A_p, D_{qr}) = \sum_{j=1}^{n} \sigma_{qr}^{j} \mathrm{T}(A_p)$$

Local Expansion is given by

$$\perp(A_p) = \{\mathrm{Cov}(A_p, A_x) \mid x \in LS_p\}$$

• Bl … is

–

–

ased

h:

-first

on the

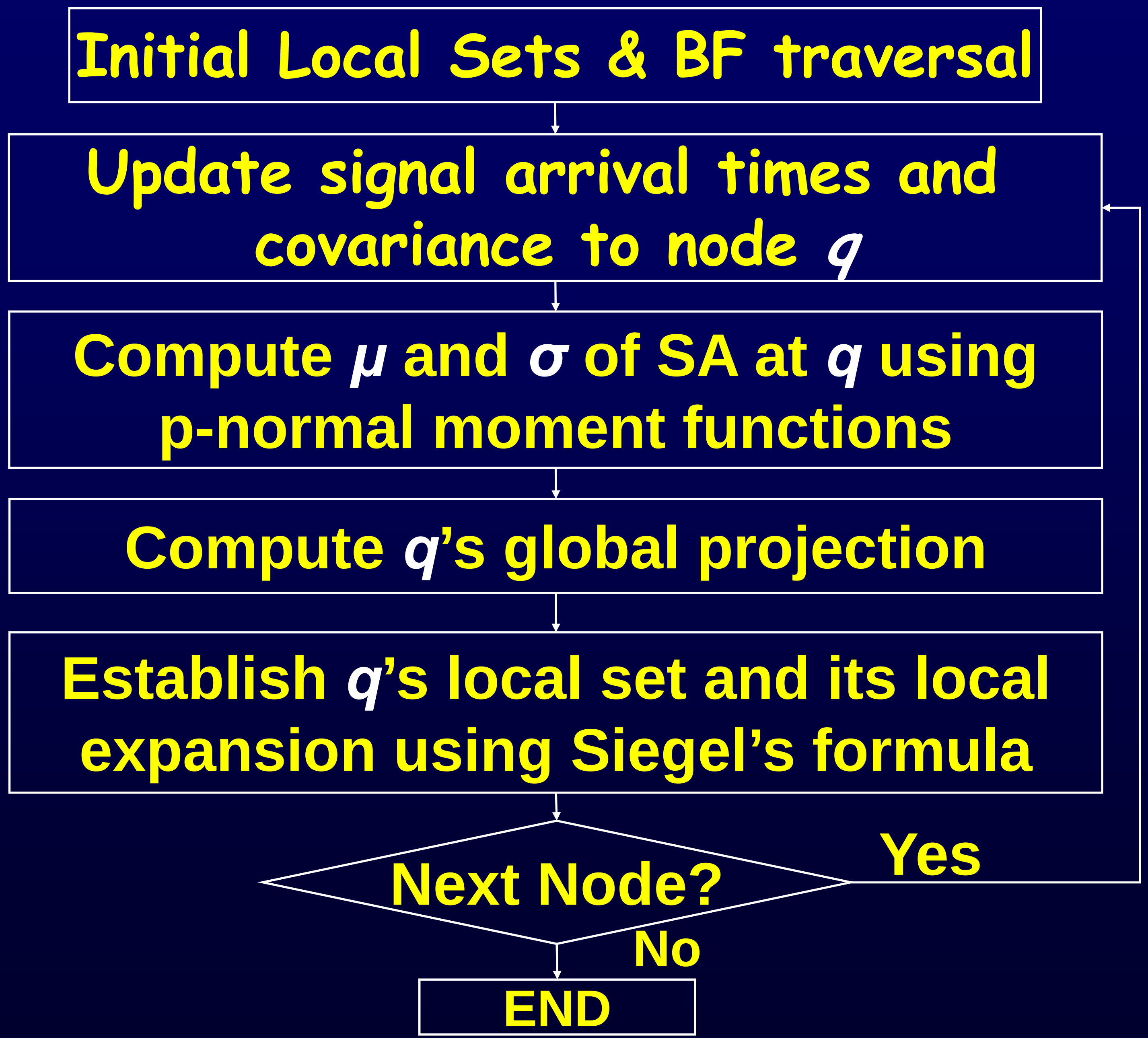
Graph

Property of Local Set

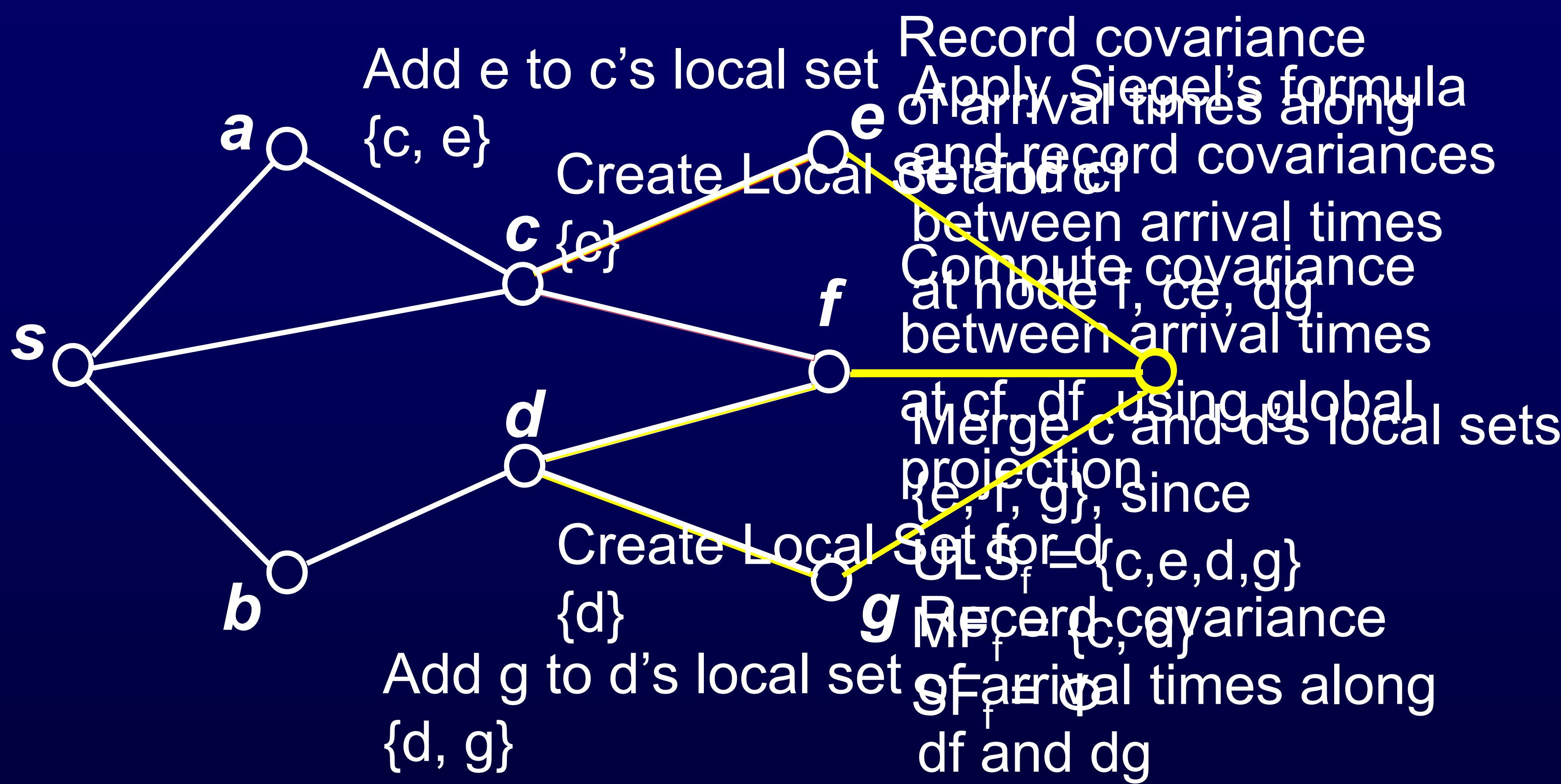
Set Theoretic Model for Identifying Re-convergent Paths

- Each node can be at most in one local set.
- Merge multiple local sets when arrival times meet at one node.
- Add a multiple fan-out node to its parents' local set, or create a new local set.
- The total number of elements in all the local sets doesn't exceed the maximum number of nodes in one BFT level, space ~$O(N/d)$.
- Local expansion is defined as the covariance structure of a local set.

Flow Chart of the SSTA algorithm



Example



$ULS_p = U_{i=1..k}\{LS_{ip}\} - SF_p - MF_p$

$LS_p = ULS_p$ U {p} if ULS_p is non-empty or p is a multiple fan-out node

Experimental Results

- **ISCAS 85 benchmark to verify the algorithm**
  50% global, 50% local variations, respectively

COMPARISON OF MAXIMUM CIRCUIT DELAY BETWEEN THE PROPOSED METHOD AND RANDOM SIMULATION

Circuit Example	Random Simulation			The Proposed Method			Mean Err.	SD. Err.
	Mean(ps)	SD.(ps)	CPU Time(s)	Mean(ps)	SD.(ps)	CPU Time(s)		
C432	2032.9	128.6	50.1	2031.3	121.5	8.0	0.1%	5.5%
C499	1324.4	98.6	64.6	1326.1	89.7	1.1	0.1%	9.0%
C880	2569.9	175.1	146.5	2570.5	174.7	0.3	0.1%	0.2%
C1355	2813.4	149.9	211.5	2821.7	156.1	0.6	0.3%	4.1%
C1908	4284.3	221.9	313.4	4325.4	209.5	7.8	1.0%	5.6%
C2670	3556.6	207.8	441.7	3556.1	201.0	1.3	0.1%	3.3%
C3540	5069.1	240.6	593.2	5086.8	223.0	18.6	0.3%	7.3%
C5315	5153.5	275.6	880.7	5157.6	271.9	6.2	0.1%	1.3%
C6288	1367.1	580.6	937.8	1367.0	577.5	0.2	0.1%	0.5%
C7552	449.0	26.0	1271.3	449.9	26.0	7.7	0.2%	0.0%

Speed up Achieved is between 6 and 4,500 for circuits up to 2,000 gates while accuracy is between 0.1% and 0.2%.

Conclusion

- Global Projection handles global variations which cause correlations across the chip
- General moments, mean/variance, covariance formulas for k-normal distribution
- Local Expansion handles correlation from local variations due to re-convergent paths
- Development of a <u>fast</u> Statistical Timing Analysis Tool that utilizes Multivariate Normal Distributions

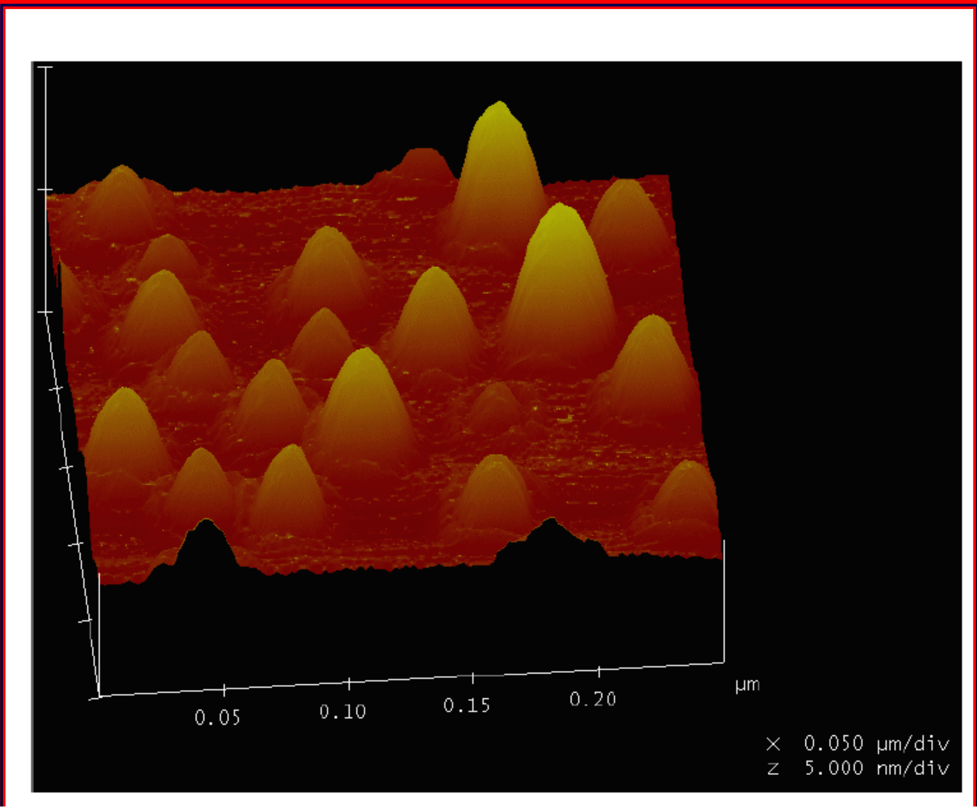
To create a good philosophy you should renounce metaphysics but be a good mathematician.

- Bertrand Russell

Alternate Robust Solution?

Can we Solve the Manufacturing Problem by Adopting Novel Nanotechnologies?

Self-assembled Quantum Dot structures are intrinsically more Robust because of their collective neuromorphic computation



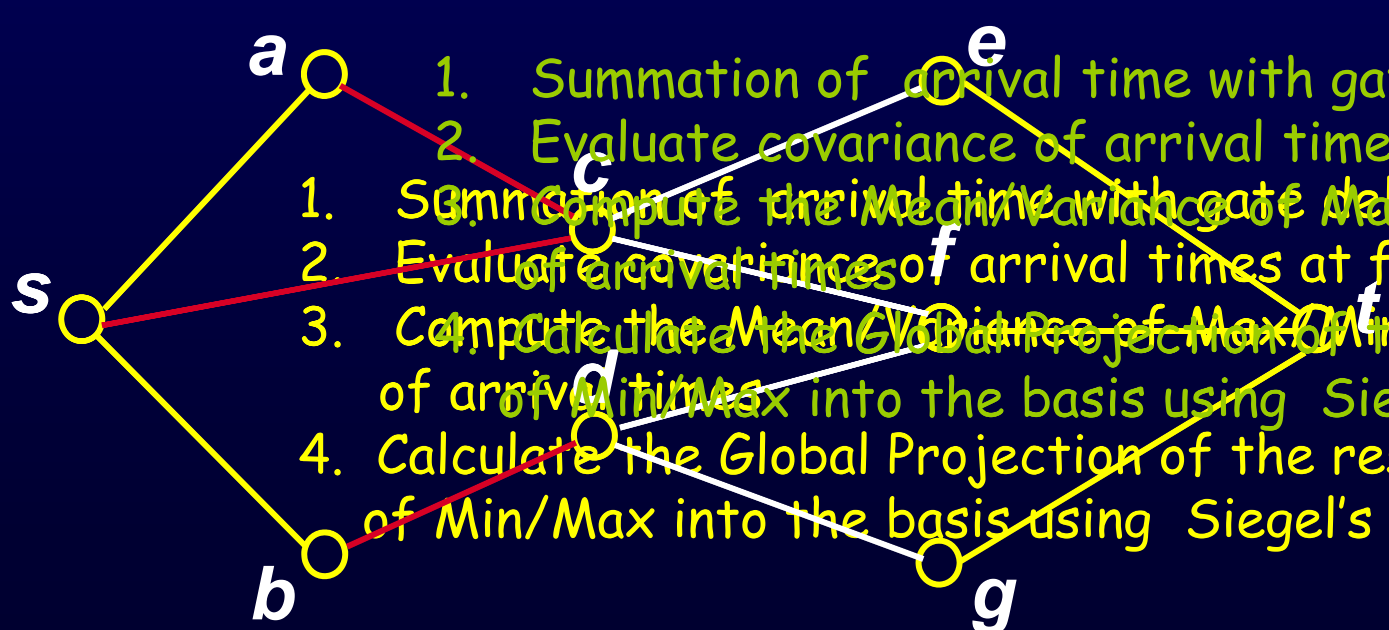
Intrinsically Robust in Presence Of Many Faults

Edge Extraction

Global Projection

- Independent normal V.V.'s as a 'basis', derived by **Principal Component Analysis**
- Project timing variables onto the basis $X=(X_1, X_2, \ldots, X_n)$: $A_r = a_r X$, $A_q = a_q X$, $D_g = d_g X$

$$A_r + D_g = (a_r + d_g)X \quad Cov(A_r, A_q) = Cov(a_r X X^T a_g^T)$$



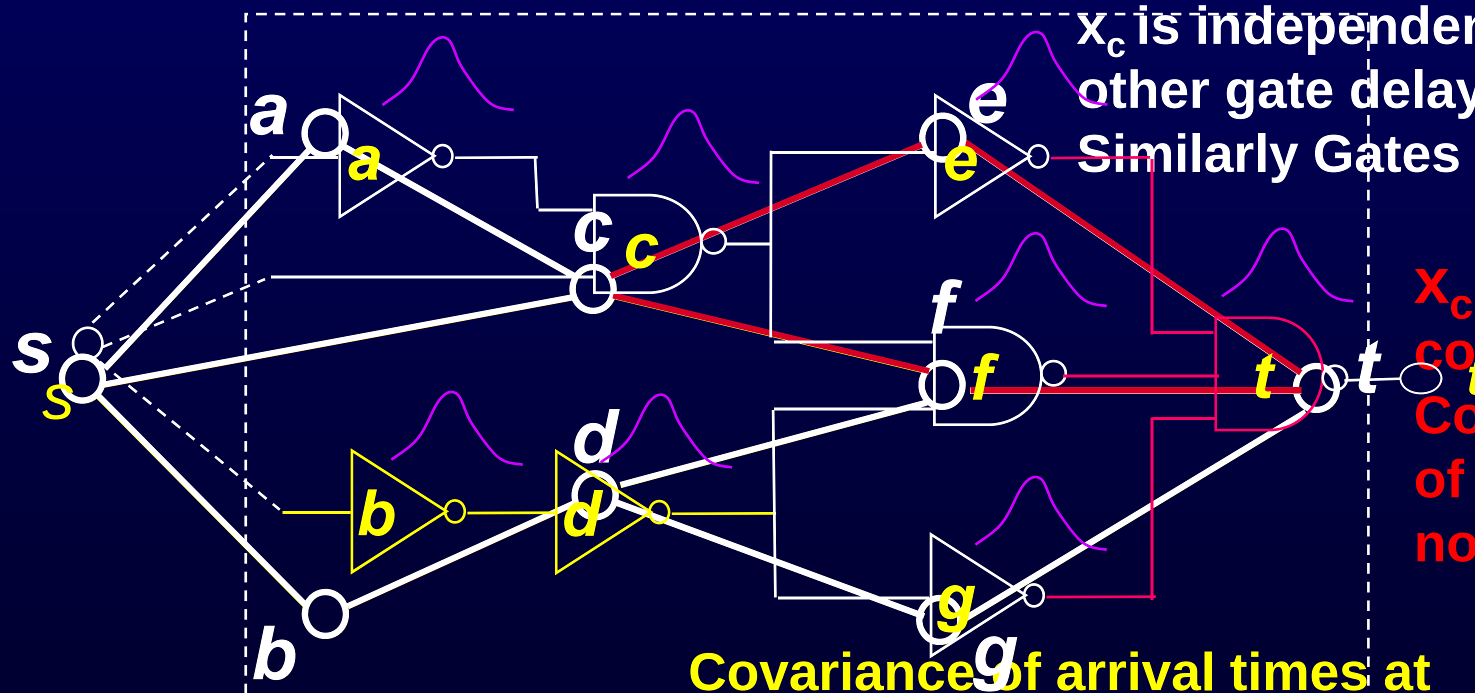
1. Summation of arrival time with gate delay
2. Evaluate covariance of arrival times at fan-ins
3. Compute the Mean/Variance of Max/Min operation of arrival times
4. Calculate the Global Projection of the results of Min/Max into the basis using Siegel's formula

1. Summation of arrival time with gate delay
2. Evaluate covariance of arrival times at fan-ins
3. Compute the Mean/Variance of Max/Min operation of arrival times
4. Calculate the Global Projection of the results of Min/Max into the basis using Siegel's formula

Correlation by Local Variations

- Independent local variations can cause path delay correlation due to re-convergent paths

Gate c has local variation x_c
x_c is independent of any other gate delays.
Similarly Gates *b*, *d*, …



x_c must be considered into Covariance of arrival times at node t

Covariance of arrival times at path *scf* and path *sbdf* only requires global projection

Properties of Local Set

- Each node can be at most in one local set.
- Merge multiple local sets when arrival times meet at one node.
- Add a multiple fan-out node to its parents' local set, or create a new local set.
- The total number of elements in all the local sets doesn't exceed the maximum number of nodes in one BFT level, space ~O(N/d).
- Local expansion is defined as the covariance structure of a local set.

Local Expansion

- Using local expansion
  - Local set, *LS*: for each node *a* in its local set, there at least is another node which shares one multiple fanout node with *a*.

  $LS_p = ULS_p \cup \{p\}$, if ULS_p is non-empty or *p* is a multiple fan-out node

  $ULS_p = \cup_{i=1..k}\{LS_{ip}\} - SF_p - MF_p$

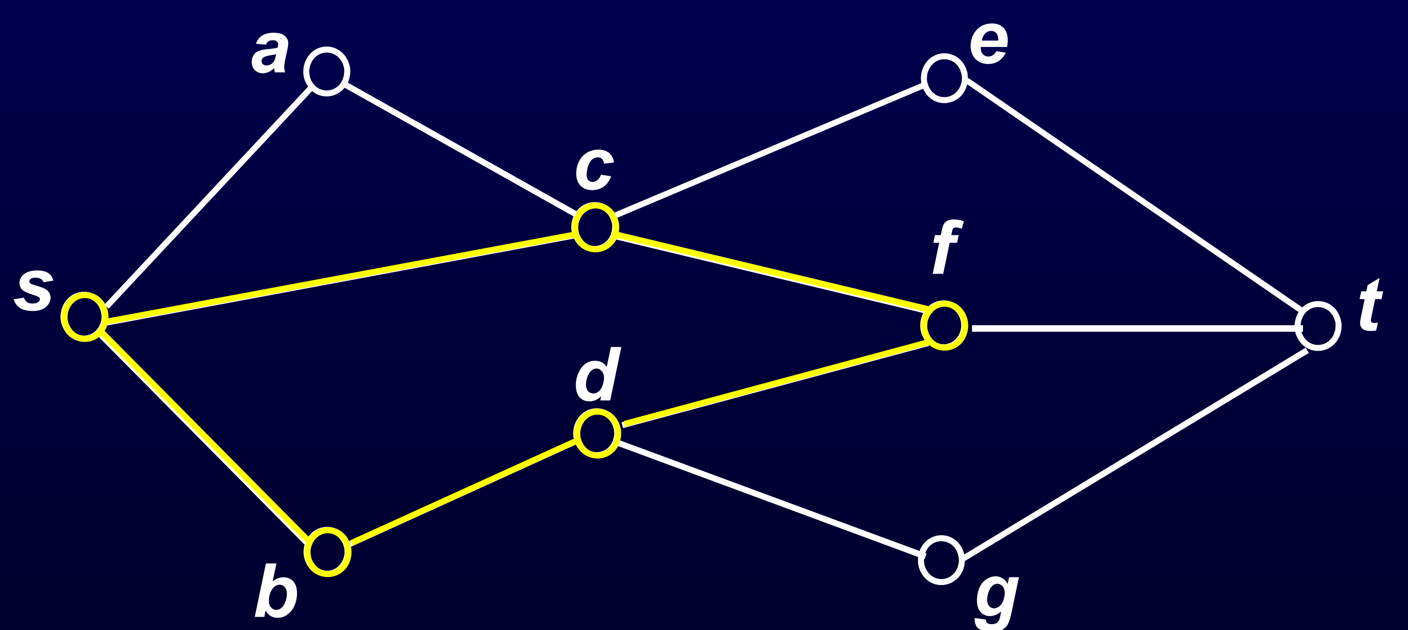
  SF_p the set of nodes only fan out to *p*.

  MF_p the set of multiple fan-out nodes which has *p* as their last visited fan-out node.
- Local expansion is defined as the covariance structure of a local set.

Some Theoretical Results

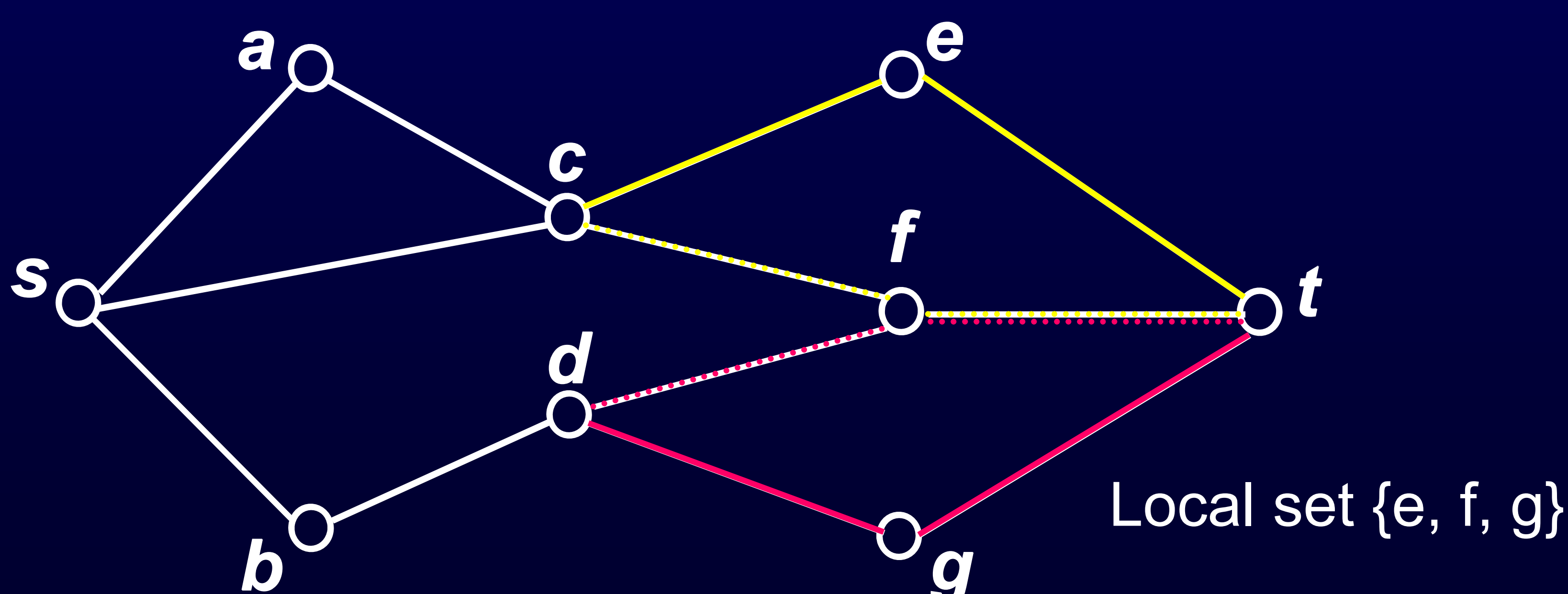
- ***Lemma 1*: for two nodes where no shared paths from primary inputs, covariance of arrival times at the two nodes equals to the inner product of their two global projections**

$$Cov\ (A_{fc},\ A_{fd}) = Cov\ (a_{fc} X X^T a_{fd}^T)$$



Some Theoretical Results

- *Lemma 3*: In the breadth-first traversal of timing graph, if a re-convergent node is visited, then those of its fan-in nodes in the re-convergent paths must already be in the same local set.



Experimental Results

- Moments and mean/variance formulas are verified using MATLAB

TABLE I

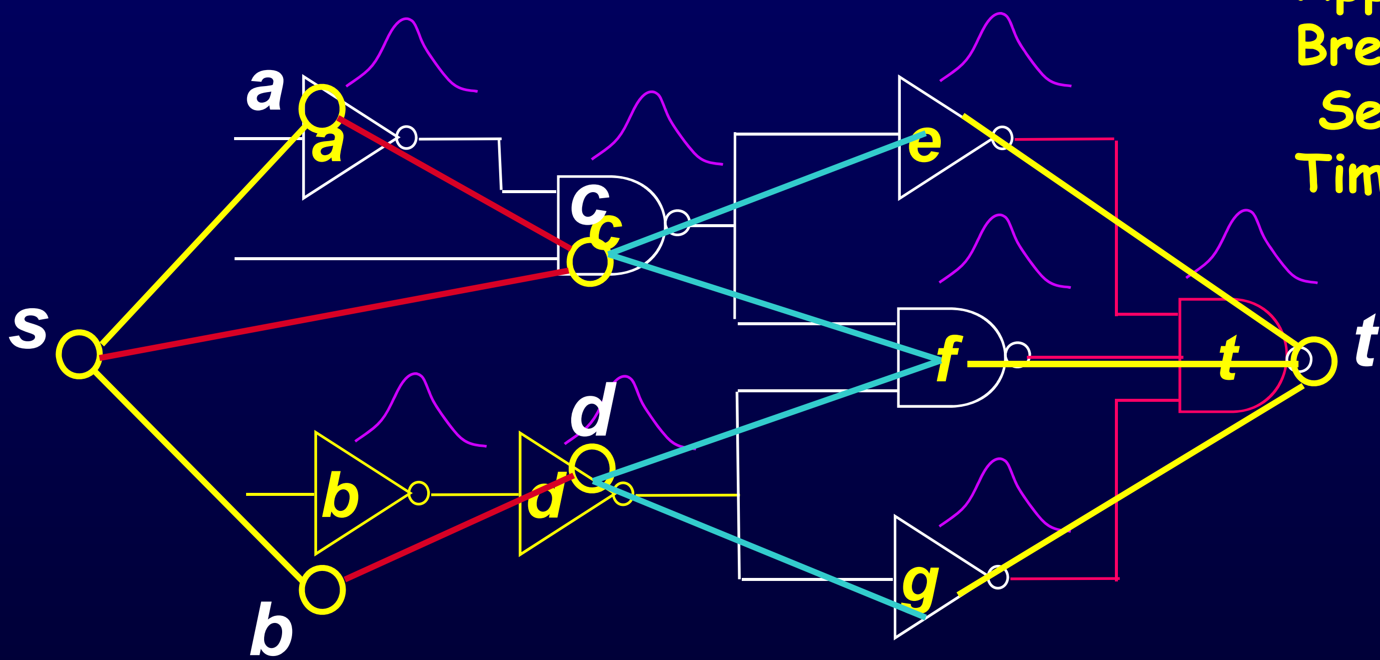
COMPARISONS OF MEAN/VARIANCE OF MAX/MIN USING FORMULAS IN SECTION III-C AND RANDOM SIMULATION USING MATLAB

Example	SAP	DEG	SOL	PWD	GMX	GEN	LAG
Dimen.	3	4	5	6	5	6	6
EQ. μ	0.999	1.412	1.374	1.517	1.426	1.515	1.853
Var.	0.623	0.457	0.691	0.735	0.628	0.738	0.448
RD. μ	1.008	1.412	1.374	1.517	1.416	1.514	1.854
Var.	0.624	0.456	0.691	0.739	0.641	0.738	0.448
EQ.$-\mu$	0.611	0.812	0.574	0.514	0.614	0.514	0.853
var.	0.598	0.457	0.691	0.738	0.654	0.737	0.448
RD.$-\mu$	0.602	0.812	0.574	0.514	0.616	0.515	0.853
Var.	0.627	0.456	0.691	0.737	0.642	0.738	0.448

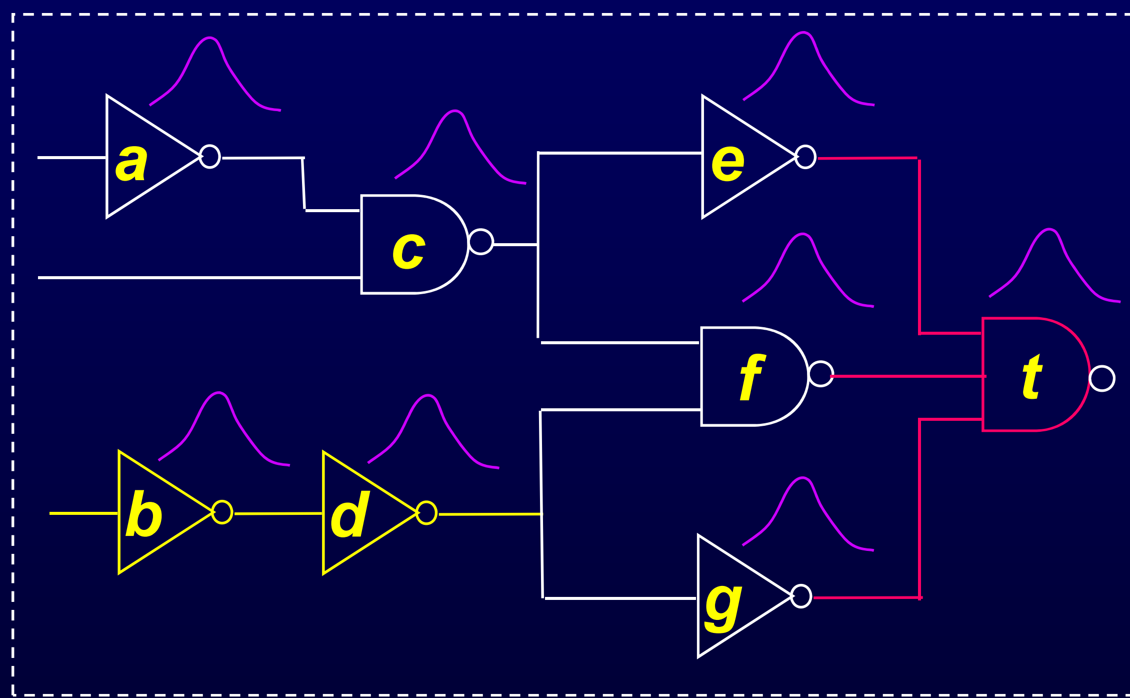
General Framework of SSTA

- Basic operations required in Statistical Static Timing Analyzer (SSTA)
  - **Sum and Min/Max**

Block-based Approach: Breadth-first Search on the Timing Graph



- **Normal distribution approximations of timing**


a
c
e
f
t
b
d
g

Interconnect Modeling using Differential Quadrature Method

Prof. P. Mazumder and Q. Xu

University of Michigan

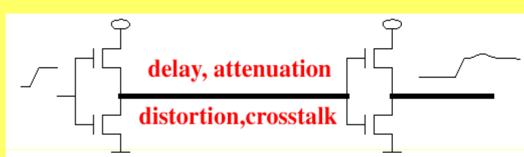
ELECTRICAL ENGINEERING and COMPUTER SCIENCE

12/16/99 NDR 1

ELECTRICAL ENGINEERING and COMPUTER SCIENCE NDR

Why modeling interconnects

- Chip size and operating speed increasing
  - ◆ System-on-a-chip (SOC)
  - ◆ Multi-chip module (MCM)
  - ◆ GHz signals

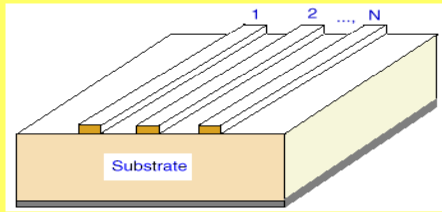


- Interconnect effects
  - ◆ interconnection delay dominates
  - ◆ crosstalk, dispersion, attenuation and reflection

2

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

How to model interconnects



- 1. TEM or quasi-TEM assumptions
- 2. Parameter extraction of metal interconnect
- 3. Multiport models of interconnects
- 4. Incorporation of multiport into simulator frame

3

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Basis of interconnect modeling

- Telegrapher's equations
  - Time-domain

$$\frac{\partial}{\partial x} i(x,t) = -C(x)\frac{\partial}{\partial t} v(x,t) - G(x)v(x,t)$$

$$\frac{\partial}{\partial x} v(x,t) = -L(x)\frac{\partial}{\partial t} i(x,t) - R(x)i(x,t)$$

  - s-domain

$$-\frac{\partial}{\partial x} I(x,s) = [sC(x) + G(x)]V(x,s)$$

$$-\frac{\partial}{\partial x} V(x,s) = [sL(x) + R(x)]I(x,s)$$

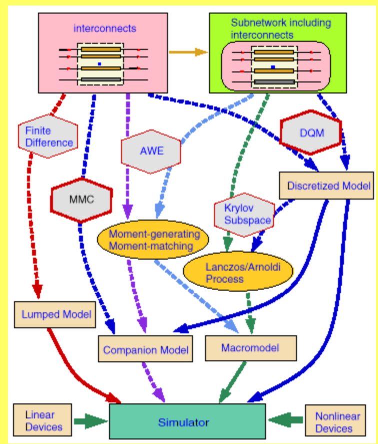
- Common goal: time-domain model
  - ready to be incorporated into SPICE frame

4

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Existing modeling methods

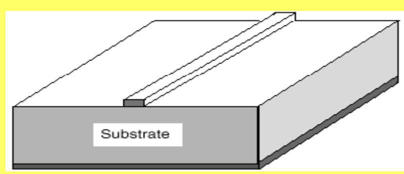
- Finite difference
  - ◆ lumped elements
- MC
  - ◆ special cases
- AWE
  - ◆ reduced order
- Reduction
  - ◆ Krylov subspace
- DQM

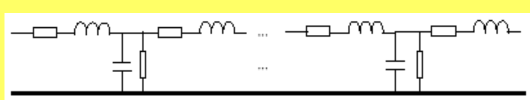


5

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Finite difference method





- SPICE early version
- Short interconnect

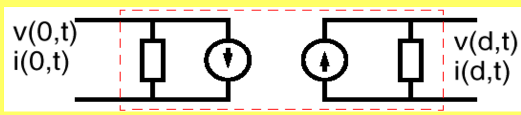
6

Method of Characteristics

- Basis: uniform interconnect terminals

$$YV(0,s) - I(0,s) = \exp(\gamma d)[YV(d,s) - I(d,s)]$$
$$YV(d,s) + I(d,s) = \exp(\gamma d)[YV(0,s) + I(0,s)]$$
$$Y = \sqrt{\frac{sC+G}{sL+R}}, \gamma = d\sqrt{(sC+G)(sL+R)}$$

- Most useful for lossless interconnect



$$yv(0,t) - i(0,t) = yv(d,t-\tau) - i(d,t-\tau)$$
$$yv(d,t) + i(d,t) = yv(0,t-\tau) + i(0,t-\tau)$$

7

Asymptotic waveform evaluation

- Pade approximation

$$H(s) \cong \hat{H}(s) = \frac{P(s)}{Q(s)} = \left(\sum_{i=1}^{n} a_n s^n\right) \Big/ \left(\sum_{i=1}^{m} b_m s^m\right) = \sum_{i=1}^{m} \frac{q_m}{s+p_m}$$

- Recursive convolution
  - turn convolution to time linear operation

$$y(t)) = x(t)*h(t) = x(t)*\sum_{i=1}^{m} q_m \exp(-p_m t)$$

- Stability problem
  - right half plane poles

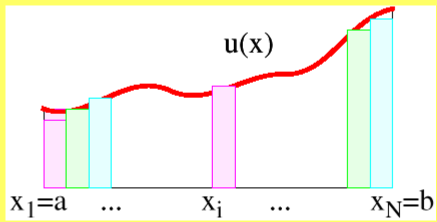
8

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Quadrature method

- Integral

$$\int_a^b u(x)dx = \sum_{i=1}^{N} a_i u(x_i)$$



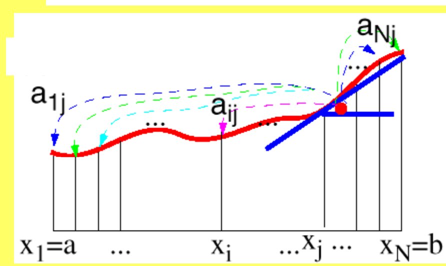
9

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Differential quadrature method

- Derivative

$$\frac{\partial}{\partial x}u(x), \frac{d}{dx}u(x)|_{x=x_j} = \sum_{i=1}^{N} a_{ij} u(x_i)$$



10

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Determination of DQM coefficient

- Select test functions *u*(*x*) to fit

$$\frac{\partial}{\partial x}u(x), \frac{d}{dx}u(x)|_{x=x_j} = \sum_{i=1}^{N} a_{ij}u(x_i)$$

- ◆ PDQ: Power functions

$$g(x) = \{1, x, x^2, x^3, \ldots, x^{N-1}\}$$

- ◆ CDQ: Chebyshev polynomials

$$g(x) = \{T_0(x), T_1(x), T_2(x), \ldots, T_N\}$$

- ◆ HDQ: Harmonic functions

$$g(x) = \left\{1, \sin\pi x, \cos\pi x, \sin 2\pi x, \cos 2\pi x, \ldots, \sin\frac{N-1}{2}\pi x, \cos\frac{N-1}{2}\pi x\right\}$$

11

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

DQM operator

- Operator

$$\frac{\partial}{\partial x}u(x), \frac{d}{dx}u(x)|_{x=x_j} = \sum_{i=1}^{N} a_{ij}u(x_i)$$

$$\frac{d}{dx}u(x) \Rightarrow \mathbf{Au}$$

- Accuracy example: derivative of **f(x)=(sin x)^2**
  - ◆ average squared error

N	5	10	15	20	25	50	75	100
FD	0.0181	0.0043	0.0019	0.0010	6.6571e-4	1.6430e-4	2.2705e-5	4.0807e-5

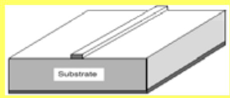
N	5	7	9	11	13	15	17	19
PDQ	4.3367e-5	7.5297e-9	4.6001e-13	1.3445e-13	1.4850e-13	1.4085e-13	1.2589e-13	1.9603e-13
CDQ	0.0169	0.0342	0.0596	0.0909	0.1266	0.1659	0.2080	0.2525
HQM	0.0101	6.9302e-4	5.2277e-5	4.1760e-6	3.4524e-7	2.9192e-8	2.5292e-9	2.1988e-10

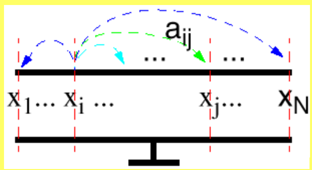
12

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Discretization of interconnect

- DQM operation





$$\frac{\partial}{\partial x}V(x_i, s) = \sum_{j=1}^{N} a_{ij}V(x_j, s)$$

$$\frac{\partial}{\partial x}I(x_i, s) = \sum_{j=1}^{N} a_{ij}I(x_j, s)$$

- Single line example: step 1
  - L'=360nH/m, C'=100pF/m
  - G'=0.01S/m, R'=50Ω/m

$$-\frac{\partial}{\partial x}I(x, s) = [sC(x) + G(x)]V(x, s)$$

$$-\frac{\partial}{\partial x}V(x, s) = [sL(x) + R(x)]I(x, s)$$

  - normalize parameters
    - L=dL', C=dC', G=dG', R=dR'
  - let 5 grid points equally space
  - 5th order HDQ operator

$$\mathbf{A} = \begin{bmatrix} -6.014 & 10.726 & -7.584 & 4.443 & -1.571 \\ -1.571 & -2.221 & 5.363 & -2.221 & 0.651 \\ 0.651 & -3.142 & 0 & 3.142 & -0.651 \\ -0.651 & 2.221 & -5.363 & 2.221 & 1.571 \\ 1.571 & -4.443 & 7.584 & -10.726 & 6.014 \end{bmatrix}$$

13

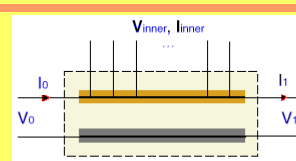
ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Discretized model---single line

- Single line

$$\mathbf{AV} = -(s\mathbf{L} + \mathbf{R})\mathbf{I}$$
$$\mathbf{AI} = -(s\mathbf{C} + \mathbf{G})\mathbf{V}$$

$$\begin{aligned}
\mathbf{A} &= [a_{ij}] \in R^{N \times N}, \\
\mathbf{V} &= [V(x_1, s), V(x_2, s), \dots, V(x_N, s)]^T, \\
\mathbf{I} &= [I(x_1, s), I(x_2, s), \dots, I(x_N, s)]^T, \\
\mathbf{L} &= diag\{L(x_1), L(x_2), \dots, L(x_N),\} \\
\mathbf{R} &= diag\{R(x_1), R(x_2), \dots, R(x_N),\} \\
\mathbf{C} &= diag\{C(x_1), C(x_2), \dots, C(x_N),\} \\
\mathbf{G} &= diag\{G(x_1), G(x_2), \dots, G(x_N),\}
\end{aligned}$$



- Single line example: Step 2

$$\cancel{\mathbf{A}}\begin{bmatrix} V(x_1,s) \\ V(x_2,s) \\ V(x_3,s) \\ V(x_4,s) \\ V(x_5,s) \end{bmatrix} = -\begin{bmatrix} sL+R & 0 & 0 & 0 & 0 \\ 0 & sL+R & 0 & 0 & 0 \\ \cancel{0} & \cancel{0} & \cancel{sL+R} & \cancel{0} & \cancel{0} \\ 0 & 0 & 0 & sL+R & 0 \\ 0 & 0 & 0 & 0 & sL+R \end{bmatrix}\begin{bmatrix} I(x_1,s) \\ I(x_2,s) \\ I(x_3,s) \\ I(x_4,s) \\ I(x_5,s) \end{bmatrix}$$

$$\cancel{\mathbf{A}}\begin{bmatrix} I(x_0,s) \\ I(x_2,s) \\ I(x_3,s) \\ I(x_4,s) \\ I(x_5,s) \end{bmatrix} = -\begin{bmatrix} sC+G & 0 & 0 & 0 & 0 \\ 0 & sC+G & 0 & 0 & 0 \\ \cancel{0} & \cancel{0} & \cancel{sC+G} & \cancel{0} & \cancel{0} \\ 0 & 0 & 0 & sC+G & 0 \\ 0 & 0 & 0 & 0 & sC+G \end{bmatrix}\begin{bmatrix} V(x_1,s) \\ V(x_2,s) \\ V(x_3,s) \\ V(x_4,s) \\ V(x_5,s) \end{bmatrix}$$

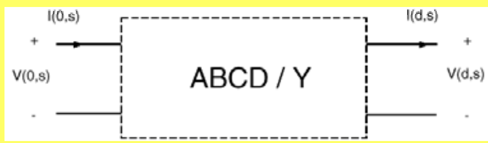
  - with equations at mid-point removed
  - (adding boundary conditions)

14

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Compact model---single line

- Eliminate internal variables



  - ABCD model

$$\begin{bmatrix} V(0,s) \\ I(0,s) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V(d,s) \\ I(d,s) \end{bmatrix}$$

  - Admittance (Y) model

$$\begin{bmatrix} I(0,s) \\ I(d,s) \end{bmatrix} = \begin{bmatrix} y11 & y12 \\ y13 & y14 \end{bmatrix} \begin{bmatrix} V(0,s) \\ V(d,s) \end{bmatrix}$$

- Single line example: step 3
  - eliminate $V(x_2,s)$, $V(x_3,s)$, $V(x_4,s)$, $I(x_2,s)$, $I(x_3,s)$, $I(x_4,s)$

$$\begin{bmatrix} V(0,s) \\ I(0,s) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V(d,s) \\ I(d,s) \end{bmatrix}$$

$$\begin{bmatrix} I(0,s) \\ I(d,s) \end{bmatrix} = \begin{bmatrix} y11 & y12 \\ y13 & y14 \end{bmatrix} \begin{bmatrix} V(0,s) \\ V(d,s) \end{bmatrix}$$

  - A, B, C, D, y11, y12, y13, and y14 are rational approximation

$$\frac{a_0 + a_1 s + \ldots + a_6 s^6}{b_0 + b_1 s + \ldots + b_6 s^6} = \sum_{n=1}^{6} \frac{q_n}{s + p_n}$$

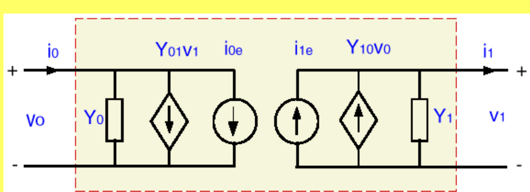
  - order of denominator/numerator is 2*N-4 (N: DQM order)

15

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Companion model---single line

- Take inverse Laplace transform
- Apply recursive convolution



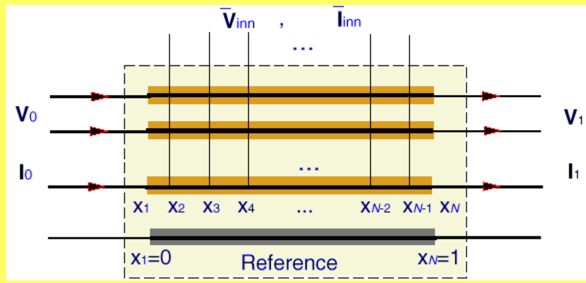
- Single line example: step 4
  - companion model
  - add boundary conditions
  - or incorporate into simulator frame
  - time-stepping
- transient results

16

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Discretized model---multi-line

- Multiconductor line

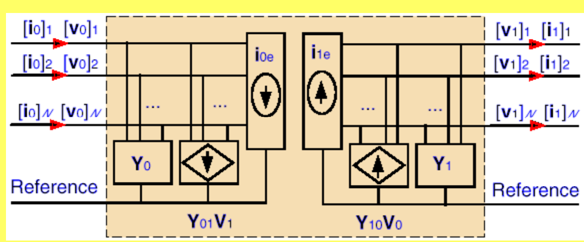


17

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Companion model---multi-line

- Eliminate internal variables
- Take inverse Laplace transform
- Apply recursive convolution



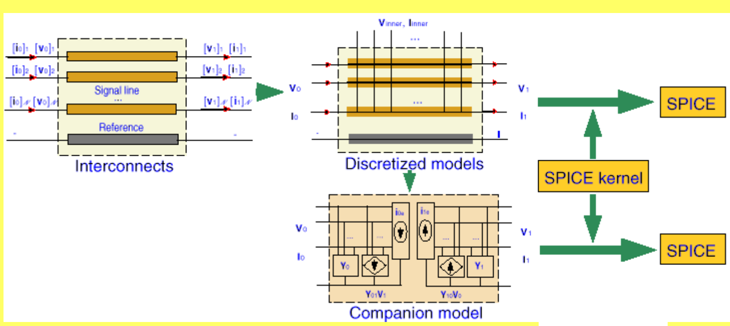
18

DQM application: step by step

- Preparation
  - Telegrapher's equations
  - DQ coefficient matrices: PDQ, CDQ, HDQ
  - Select a kind of DQM
- Discretization of Telegrapher's equations
- Compact *s*-domain models
- Companion models
- Incorporation into MNA or simulator frame
- Transient simulation results

Two different incorporations

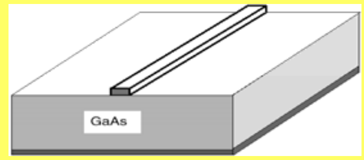
- Discretized model
- companion model

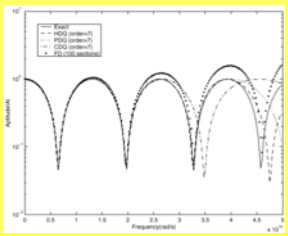


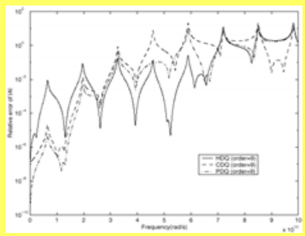
ELECTRICAL ENGINEERING and COMPUTER SCIENCE

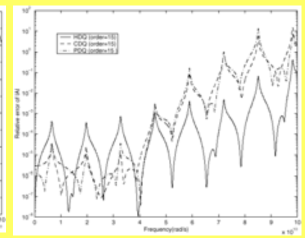
Example 1---frequency response

- A of ABCD matrix
- Error of 9th order DQM
- Error of 15th order DQM







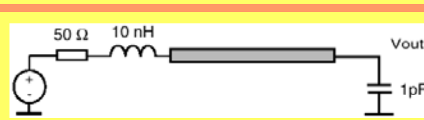


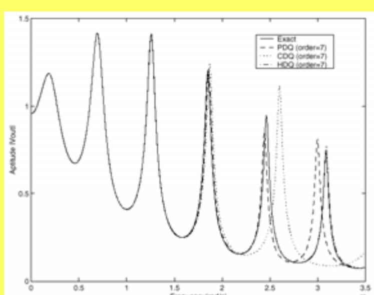
21

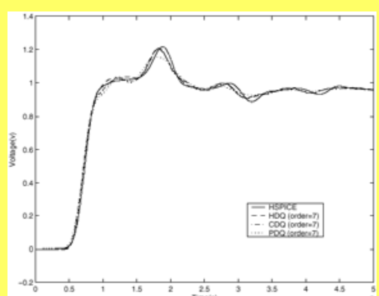
ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Example 2---single line

- Frequency response
- Transient response





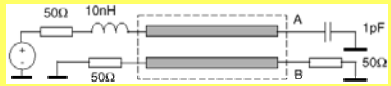


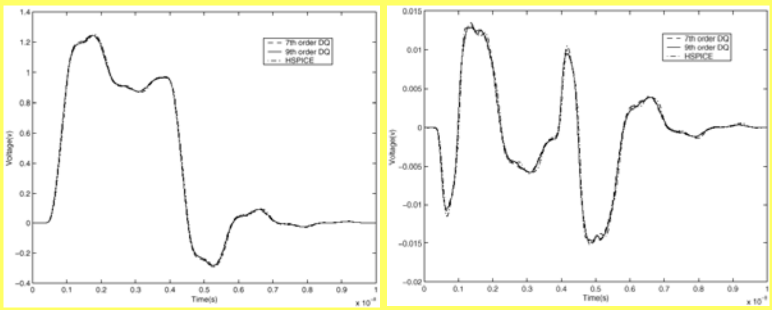
22

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Example 3---coupled line

- Transient response



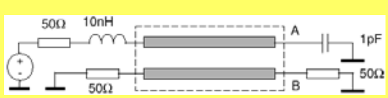


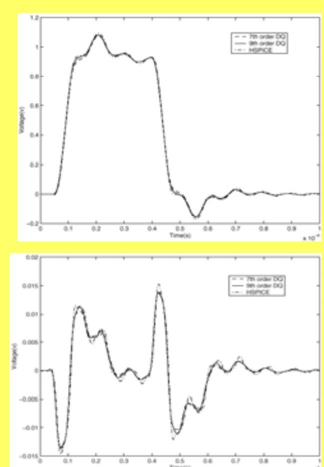
23

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Example 4---nonuniform line

- Two coupled line
- Transient responses
- Running time on Ultra-1
  - HSPICE 3.6s
  - 7th-order DQM 0.95s
  - 9th-order DQM 1.2s



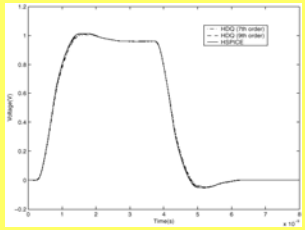


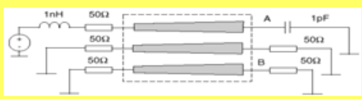
24

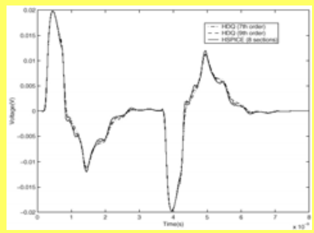
ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Example 5---nonuniform line

- Three coupled line
- Transient response
- Running time on Ultra-1
  - HSPICE 5.527s
  - 9th-order DQM 1.56s







25

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Comparison of existing methods

	Properties					
Existing methods	Feature	Applicability	Accuracy	Efficiency	Stability	Passivity
Finite Difference	Direct numerical	uniform/nonuniform	fair	poor	fair	fair
MMC	Device model	uniform	good	good	good	good
AWE	Reduced-order (explicit)	uniform/subnetwork	fair	good	poor	poor
Krylov subspace	Reduced-order (implicit)	subnetwork	good	good	fair	fair
DQM	global numerical	uniform/nonuniform	good	good	good	fair

26

ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Conclusions

- Derivative of a function can be approximated by a weighted linear sum of all the function values at every mesh point. DQM
  - a simple direct numerical technique
  - less sample points needs (1/10 of FD grid points)
  - considerable accuracy
  - high efficiency
  - a general formulation of such kind of methods
  - applicable to both uniform and nonuniform line

27

n1

Compact Finite Difference Quadrature Method and Its Numerical Dispersion

Presented by: Prof. Pinaki Mazumder
GSRA: Q. Xu
University of Michigan, EECS Dept.,
1301 Beal Ave
Ann Arbor, MI 48109 USA

Slide 1

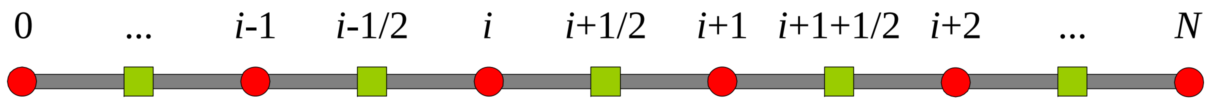
n1 UMEECS

ndrgrp, 5/23/2003

Topics

- General Finite Difference Quadrature Method
- Compact FDQ method
- Numerical dispersion analysis
- Resolution heuristic

FDQ Method Formulation



- General Approximation framework for FDQ
  - Integer grid points and half grid points
- Finite difference of integer grid points

$$f(x_{i+1})-f(x_i)=\Delta x\sum_{j=0}^{N-1}a_{ij}w_{ij}f'(x_{j+1/2})$$

- Finite differences of half grid points

$$f(x_{i+1/2})-f(x_{i-1/2})=\Delta x\sum_{j=0}^{N}a_{ij}w_{ij}f'(x_j)$$

Finite differences

Quadratures: Summations of weighted derivative

Matrix Representation

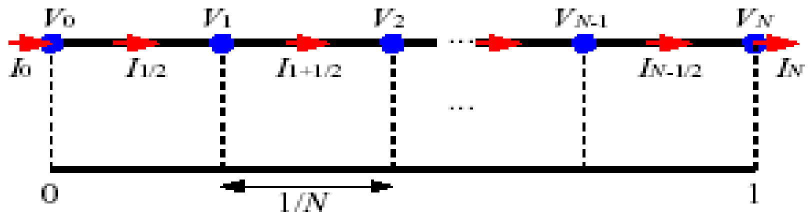
- General Approximation framework for FDQ

$$f(x_{i+1}) - f(x_i) = \Delta x \sum_{j=0}^{N-1} a_{ij} f'(x_{j+1/2})$$

$$\begin{bmatrix} f_1 - f_0 \\ f_2 - f_1 \\ \cdots \\ f_N - f_{N-1} \end{bmatrix} = \Delta x \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ & & \ddots & \\ a_{N1} & a_{N2} & & a_{NN} \end{bmatrix} \begin{bmatrix} f'_{1/2} \\ f'_{1+1/2} \\ \cdots \\ f'_{N-1/2} \end{bmatrix}$$

- Determine the coefficients by using testing functions
- Accuracy: O(Δx^{N-1})

Application to Transmission Lines



- S-domain Telegrapher's equation

$$\boxed{\begin{aligned} V' &= (sL+R)I \\ I' &= (sC+G)V \end{aligned}}$$

- Voltage grid points: integers $\{x_i\}$, i=0..N
- Current grid points: halves $\{x_{i+1/2}\}$, i=0..N-1
- Approximation framework

$$\boxed{\begin{aligned} V(x_{k+1},s) - V(x_k,s) &= \Delta x \sum_{j=1}^{N-1} a_{kj} V'(x_{j+1/2},s) \\ I(x_{k+1/2},s) - I(x_{k-1/2},s) &= \Delta x \sum_{j=0}^{N} b_{kj} I'(x_j,s) \end{aligned}}$$

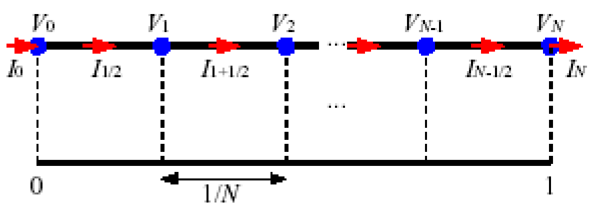
Matrix Representation

- Discrete Telegrapher's equations

$$\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \cdots & \cdots & \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_N \end{bmatrix} = (sL+R) \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ & & \ddots & \\ a_{N1} & a_{N2} & & a_{NN} \end{bmatrix} \begin{bmatrix} I_{1/2} \\ I_{1+1/2} \\ \vdots \\ I_{N-1/2} \end{bmatrix}$$

$$\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} I_0 \\ I_{1/2} \\ \vdots \\ I_N \end{bmatrix} = (sC+G) \begin{bmatrix} b_{00} & b_{01} & \cdots & b_{0N} \\ b_{10} & b_{11} & \cdots & b_{1N} \\ & & \ddots & \\ b_{N1} & b_{N2} & & b_{NN} \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_N \end{bmatrix}$$

Original FDQ Scheme



$$\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \cdots & \cdots & \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_N \end{bmatrix} = (sL + R) \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ & & \ddots & \\ a_{N1} & a_{N2} & & a_{NN} \end{bmatrix} \begin{bmatrix} I_{1/2} \\ I_{1+1/2} \\ \vdots \\ I_{N-1/2} \end{bmatrix}$$

$$\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} I_0 \\ I_{1/2} \\ \vdots \\ I_N \end{bmatrix} = (sC + G) \begin{bmatrix} b_{00} & b_{01} & \cdots & b_{0N} \\ b_{10} & b_{11} & \cdots & b_{1N} \\ & & \ddots & \\ b_{N1} & b_{N2} & & b_{NN} \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_N \end{bmatrix}$$

- Matrices A and B are dense

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} B = \begin{bmatrix} b_{00} & \cdots & b_{0N} \\ \vdots & \ddots & \vdots \\ b_{N0} & \cdots & b_{NN} \end{bmatrix}$$

- Problem: dense matrices, computationally expensive

Dense Matrices: Computationally Expensive

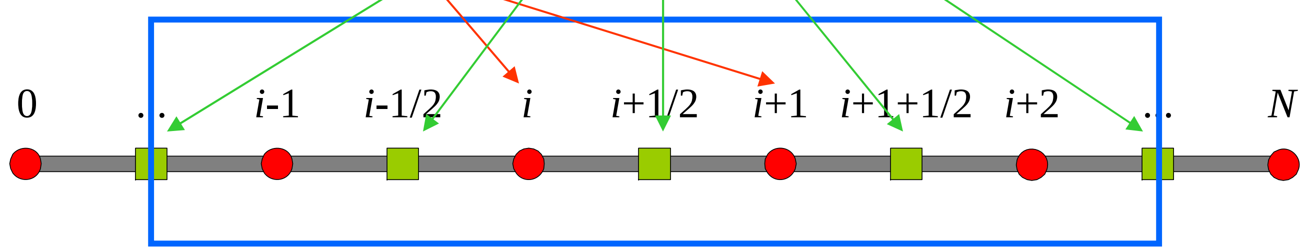
$$A = \begin{bmatrix} a_{11} & \dots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} B = \begin{bmatrix} b_{00} & \dots & b_{0N} \\ \vdots & \ddots & \vdots \\ b_{N0} & \cdots & b_{NN} \end{bmatrix}$$

- Suitable for small scale problems (< 3~5 wavelengths)
  - N~10
- Expensive for larger scale problems
  - N~100

Solution: Compact FDQ Scheme

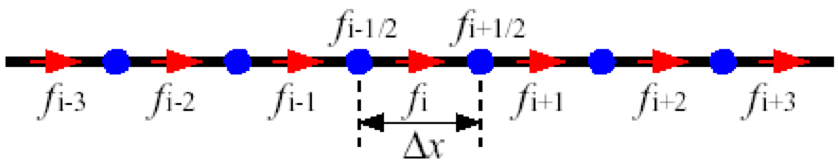
- Compact approximation framework for FDQ
  - A Sliding window with length of n for central grid points

$$f(x_{i+1}) - f(x_i) = \Delta x \sum_{j=-(n-1)/2}^{(n-1)/2} c_j f'(x_{i+j+1/2})$$



  - For boundary grid points, use the original FDQ schemes

Improvement: Keep Large Matrices Sparse

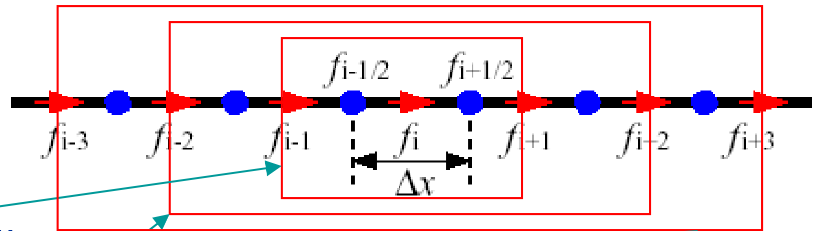


- Use a scheme with fixed bandwidth

$$A=\begin{bmatrix} X & X & X & & & & \\ X & X & X & X & & & \\ X & X & X & X & X & & \\ & \dots & \dots & \dots & \dots & \dots & \\ & & X & X & X & X & X \\ & & & X & X & X & X \\ & & & & X & X & X \end{bmatrix} B=\begin{bmatrix} X & X & X & & & & & & \\ X & X & X & X & & & & & \\ X & X & X & X & X & & & & \\ & X & X & X & X & X & & & \\ & & \dots & \dots & \dots & \dots & \dots & & \\ & & & X & X & X & X & X & \\ & & & & X & X & X & X & X \\ & & & & & X & X & X & X \\ & & & & & & X & X & X \end{bmatrix}$$

- Result in sparse diagonal matrices

Formulation of Compact FDQ Schemes



- FDQ4 (4th order accuracy)

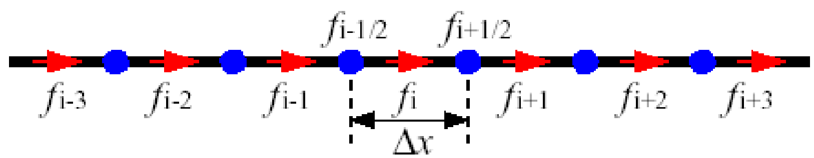
$$f(x_{i+1/2}) - f(x_{i-1/2}) =$$
$$\Delta x[af'(x_{i-1}) + bf'(x_i) + af'(x_{i+1})]$$

- FDQ6 (6th order accuracy)

$$f(x_{i+1/2}) - f(x_{i-1/2}) =$$
$$\Delta x[af'(x_{i-2}) + bf'(x_{i-1}) + cf'(x_i) + bf'(x_{i+1}) + af'(x_{i+2})]$$

- FDQ8 (8th order accuracy)

How Accurate: Fourier Analysis for Numerical Dispersion



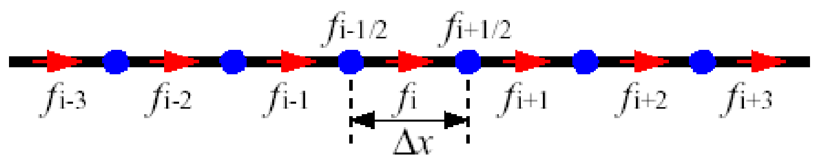
- Assume a wave $f(x,t) = e^{j(\omega t - kx)}$
- FD2 $f(x_{i+1/2}) - f(x_{i-1/2}) = \Delta x f'(x_i)$
- Plug in and simplify, compare the wave-numbers

$$\frac{\sin(k\Delta x/2)}{\Delta x/2} \leftrightarrow k$$

- LHS is numerical dispersion caused by discretization
- Normalized wavenumber of FD2:

$$\frac{\sin(k\Delta x/2)}{k\Delta x/2}$$

On Numerical Dispersion: FDQ4, etc



- FDQ4 $f(x_{i+1/2}) - f(x_{i-1/2}) = af'(x_{i-1}) + bf'(x_i) + af'(x_{i+1})$
- Plug in wave $f(x,t) = e^{j(\omega t - kx)}$

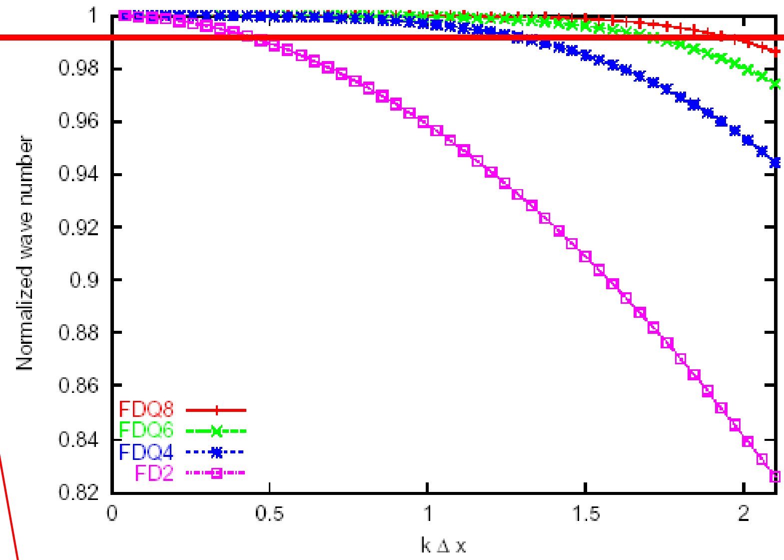
$$\frac{\sin(k\Delta x/2)}{\Delta x/2} \leftrightarrow k(2a\cos(k\Delta x) + b)$$

- Normalized wave-number of FDQ4:

$$\frac{\sin(k\Delta x/2)}{k\Delta x/2(2a\cos(k\Delta x) + b)}$$

Comparison of Numerical Dispersion

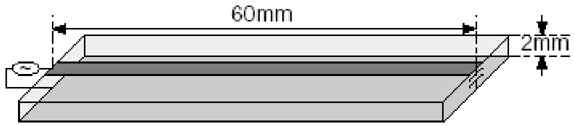
- Idea normalized wave number = 1



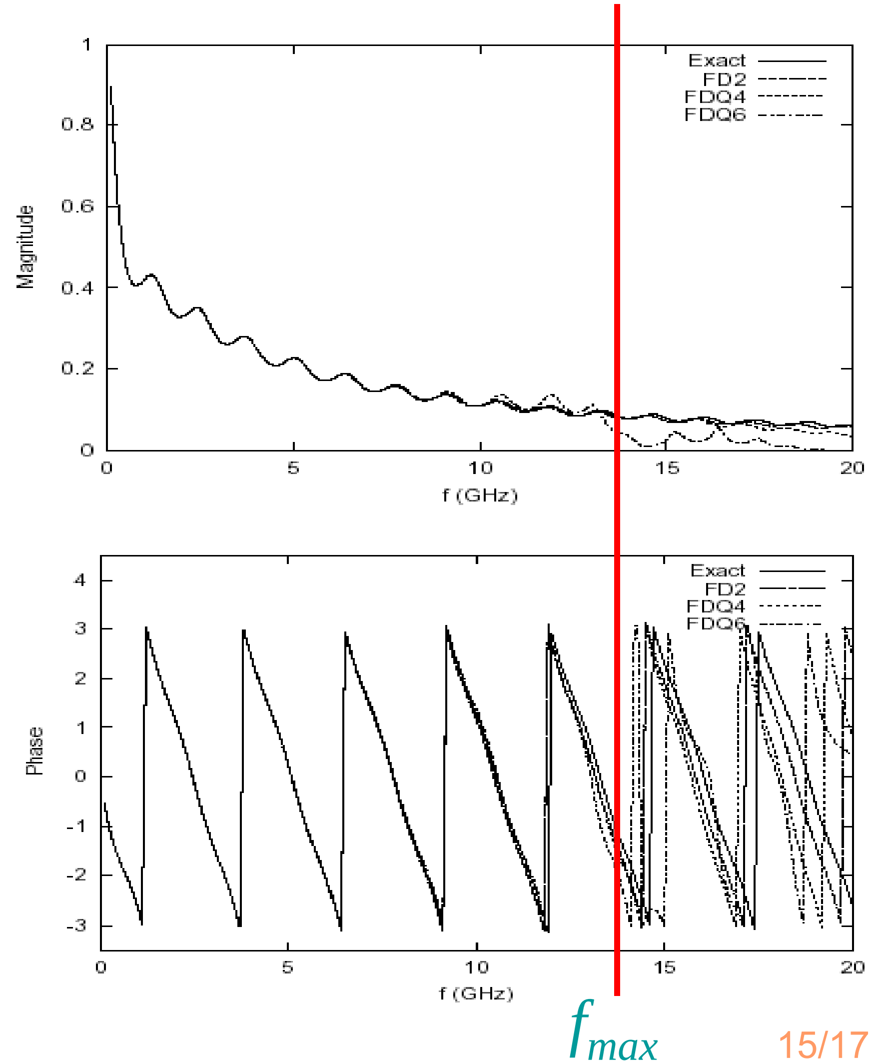
- Number of cells per wavelength needed given phase accuracy

error (%)	FD2	FDQ4	FDQ6	FDQ8
1	12.8	4.6	3.5	3.1
0.1	40.8	8.2	5.3	4.3

Numerical Results: 1

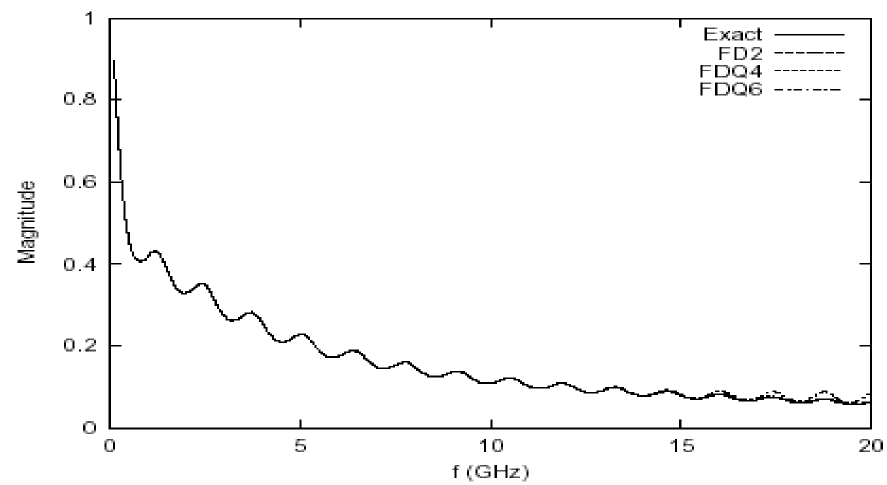


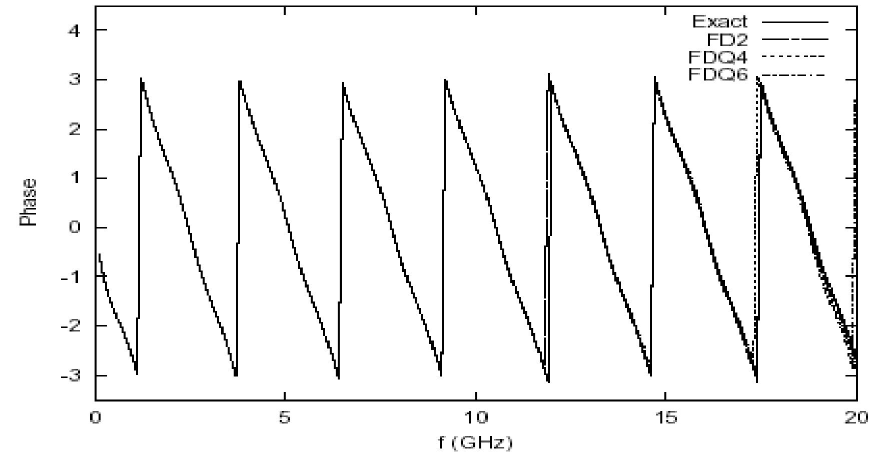
- L=360nH/m, C=100pf/m
  R=1 kOhm/m
  f_{max}=14 GHz
  λmin=1.2 cm
- Phase error: 1%
- Numbers of cells:
  - FD2: 64
  - FDQ4: 23
  - FDQ6: 17



Numerical Results: 2

- Phase error: 0.1%
- Numbers of cells
  - FD2: 204
  - FDQ4: 41
  - FDQ6: 27
- Heuristics for resolution:
  - 1% phase error is OK
  - 0.1% is good





Summary

- Modified FDQ method to reduce the matrix density and therefore computational expense
- Accuracy comparison by using Fourier analysis
- Heuristic for Number of cells per wavelength needed

  Suggested resolutions at 1% of phase error, CPW:

  - FD2: 12.8
  - FDQ4: 4.6
  - FDQ6: 3.5
  - FDQ8: 3.1