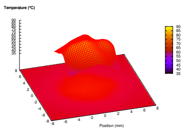
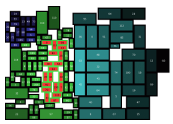
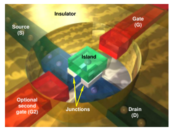
Temperature-Aware and Low-Power Design and Synthesis of Integrated Circuits and Systems

Robert P. Dick

http://robertdick.org/talp
L477 Tech
847–467–2298
Department of Electrical Engineering and Computer Science
Northwestern University

Thermal effects now critical in IC design

Performance impact
- Transistors
- Interconnect

Power impact

Subthreshold leakage depends on temperature

System cost

Cooling and packaging cost

Reliability impact
- Electromigration
- Dielectric breakdown
- Thermal cycling
- Etc.

Problem decomposition

If

$$\left.\begin{array}{c}\text{Interconnect length}\\ \text{Temperature}\end{array}\right\} \text{independent from} \left\{\begin{array}{c}\text{Power consumption}\\ \text{Performance}\\ \text{Reliability}\end{array}\right.$$

Then, this simplifies the problem — decompose into subproblems
- Architectural synthesis or design
- Physical synthesis

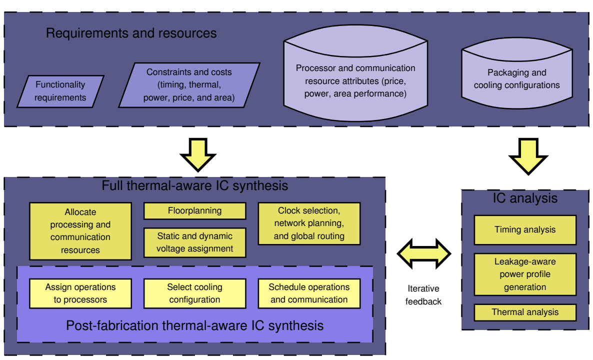
Problem decomposition cannot always be used

Reality

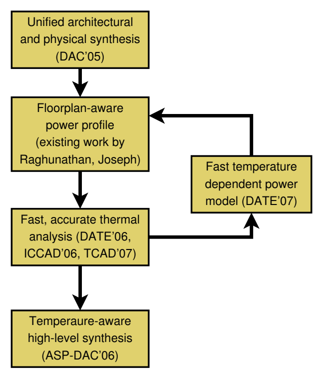
$$\left.\begin{array}{c}\text{Interconnect length}\\ \text{Temperature}\end{array}\right\} \text{interdependent with} \left\{\begin{array}{c}\text{Power consumption}\\ \text{Performance}\\ \text{Reliability}\end{array}\right.$$

Complicates system-level design
- Architectural decisions impact physical design
- Physical design impacts architectural decisions

Temperature-aware IC synthesis overview



Research prerequisites



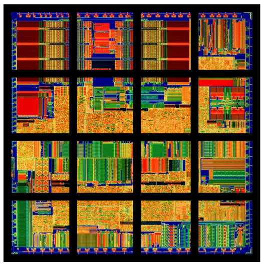
Thermal analysis requirements

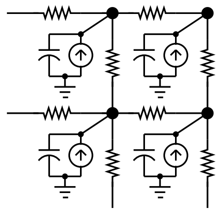
- Non-linear impact on reliability and other design characteristics
- Must be accurate
- Speed necessary for thorough design exploration
- Architectural design or synthesis require many thousands of invocations
- Must be extremely fast

Overview of thermal analysis section

- Problem definition
- Steady-state thermal analysis
- Short time scale dynamic thermal analysis
- Long time scale dynamic thermal analysis

R(C) model





Partition into 3-D elements (diagram 2-D for simplicity)

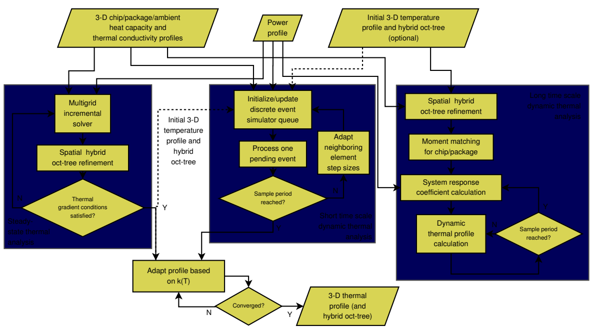
Thermal resistance	↔	Resistance
Heat flow	↔	Current
For dynamic: Heat capacity	↔	Capacitance

Problem definition

$$\mathbf{C}\frac{d\mathbf{T}(t)}{dt} = \mathbf{A}\mathbf{T}(t) + \mathbf{P}U(t)$$

- **A** is the thermal conductivity matrix
- Steady-state: Initial temperature and **C** unnecessary
- Dynamic: Transient temperature analysis, must also consider heat capacity

Thermal analysis infrastructure overview



Steady-state thermal analysis

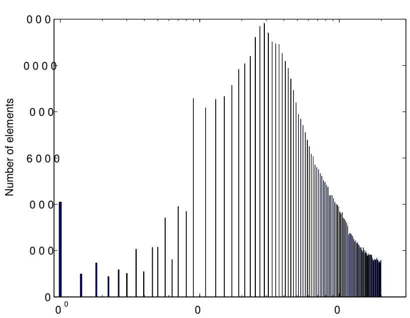
Basis: Multigrid analysis

Fast, multi-resolution relaxation method for matrix solving

- Iterative solver (relaxation) on fine grid
- Coarsen and propagate residual upward
- Iterative solver for error at coarser level
- Correct fine-grained solution based on coarse-grained error
- Iterative solver for error at fine level

- Main challenge: Too slow for repeated use on large structures, especially 3-D chip-package modeling
- Observation: Steepness of thermal gradients vary across IC

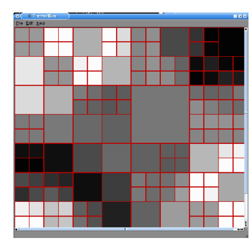
Neighbor temperature difference histogram



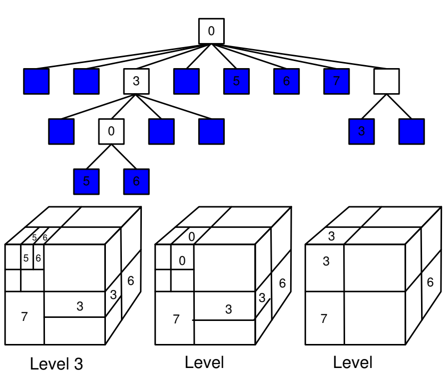
Spatial adaptation can improve performance w.o. loss of accuracy

Hybrid oct-tree

- Reduce element count by merging when $\Delta_T < \epsilon$
- Conventional oct-tree inefficient for chip-package model
- Anisotropic thermal gradients
- We generalize to hybrid oct-tree
- Arbitrary partitioning on each axis



Hybrid oct-tree

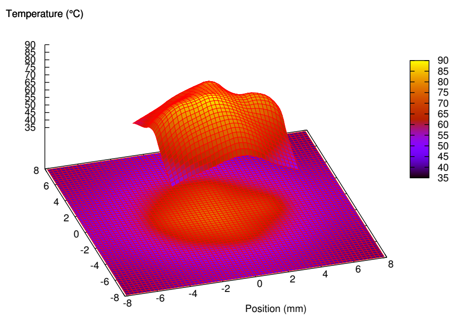


Steady-state thermal analysis evaluation

Test cases	Multigrid_HM CPU time (s)	Multigrid_HM Memory use (KB)	Multigrid_HM Elements	Const. k Peak temp. (°C)	Const. k Average temp. (°C)
IBM chip	2.2	4,506	32,768	85.2	53.8
MIT Raw	6.9	4,506	32,768	83.1	77.5

Test cases	Spatial adaptation Peak temp. (°C)	Average temp. (°C)	Error (%)	CPU time (s)	Speedup (×)	Memory use (KB)	Elements
IBM chip	90.7	54.8	1.7	0.08	27.50	252	1,800
MIT Raw	88.0	81.3	0.7	0.01	690.00	108	888

Example ASIC thermal profile



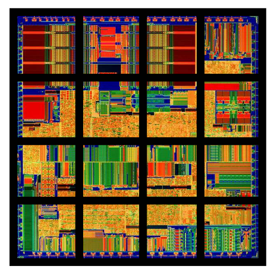
Steady-state thermal analysis in IC Synthesis

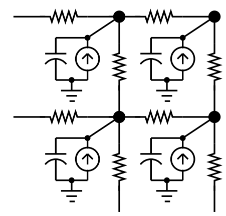
Problem	Spatial adaptation				Multigrid_HM		HLS
	CPU time (s)	Speedup (×)	Mem. (KB)	Error (%)	CPU time (s)	Mem. (KB)	CPU time (s)
chemical	0.78	53.06	265	0.35	41.39	4,506	40.02
dct_wang	2.52	37.08	264	0.24	93.43	4,506	301.37
dct_dif	2.4	37.63	266	1.5	90.31	4,506	71.60
dct_lee	6.1	27.64	268	0.5	168.6	4,506	132.15
elliptic	2.31	32.38	267	0.43	74.79	4,506	38.07
iir77	3.35	29.27	265	0.2	98.06	4,506	77.93
jcb_sm	1.63	21.64	277	0.13	35.27	4,506	151.95
mac	0.26	79.08	264	0.12	20.56	4,506	12.32
paulin	0.13	202.85	264	0.25	26.37	4,506	4.06
pr2	8.29	22.53	285	0.55	186.75	4,506	220.81

Time-domain thermal analysis

- Conventional time marching slow
- Adaptive Runge–Kutta faster
- However, synchronous
  - Still not fast enough
- Observation: Most ideal step sizes far greater than minimum

R(C) model

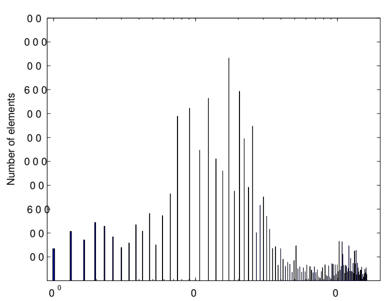




Partition into 3-D elements (diagram 2-D for simplicity)

Thermal resistance ↔ Resistance
Heat flow ↔ Current
For dynamic: Heat capacity ↔ Capacitance

Histogram of minimum safe step sizes



Asynchronous temporal adaptation can improve performance w.o. loss of accuracy

Formulation

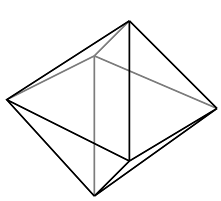
- Allow step sizes to differ in space and time
- This eliminates local time synchronization
- How to handle steps when neighbors at different time?
  - Extrapolate?

Asynchronous elements

- Extrapolation expensive for higher-order methods
- For each element, compute partial results based on n neighbors

$$n = (4d^3/3 + 2d^2 + 8d/3)$$

- Discretized octahedron
- $|E|$ is the number of elements
- d is the transitive neighbor depth



Asynchronous elements

- For higher-order Runge–Kutta, most computations can be reused/amortized
- However, asynchronous operation makes this impractical
- Asynchronous temporal adaptation will favor lower-order methods
- Try it anyway

Example asynchronous temporal method

$$0 = \sum_{i=1}^{6} \frac{T(t) - T_i \cdot u(t)}{R_i} + C\frac{dT}{dt} - P \cdot u(t)$$

By Laplace transform, linearity theorem, and inverse Laplace transform.

$$\frac{dT}{dt} = \left(\frac{\sum_{i=1}^{6} T_i/R_i + P - T(0^-) \cdot \sum_{i=1}^{6} 1/R_i}{C}\right) \cdot e^{-t/C \sum_{i=1}^{6} 1/R_i}$$

Allows computation of temperature after time step.

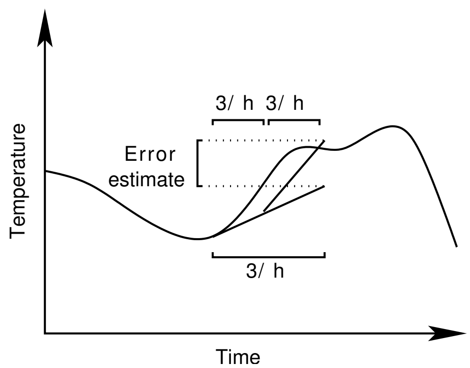
Asynchronous temporal adaptation

- Neighbors at different times
- Extrapolate neighbor temperatures to take step
- Adapt step size by taking two $3/4\ h$, one $3/2\ h$ steps and comparing

$$s_i(t_i) = u \cdot \sqrt[v]{\frac{y}{\left|\frac{dT_i}{dt}(t_i) \cdot \frac{3}{2} \cdot h_i - \frac{3}{4} \cdot h_i \left(\frac{dT_i}{dt}(t_i) + \frac{dT_i}{dt}(t_i + \frac{3}{4} \cdot h_i)\right)\right|}}$$

where v is the order of the method in use

Step size adaptation



Wave propagation and update order

- Bound neighbor difference to prevent wave propagation problem
$$h'_i = \min\left(s_i(t_i), \min_{n \in N_i}(w \cdot (t_n + h_n - t_i))\right)$$
- w a small constant, e.g., 3
- Asynchronous times, which element to update?
- Discrete event simulator
- Used event queue ordered by earliest step target time $t_i + h_i$

Asynchronous time marching validation

Problem	ISAC				GARK4	
	CPU time (s)	Speedup (×)	Mem. (KB)	Error (%)	CPU time (s)	Mem. (KB)
chemical	1.35	1354	463.47	0.13	1827.41	4,506
dct_wang	0.39	1457	312.64	0.09	568.22	4,506
dct_dif	0.40	1807	332.91	0.05	722.64	4,506
dct_lee	0.85	1071	439.22	0.04	910.88	4,506
elliptic	2.24	1361	412.23	0.02	3042.61	4,506
iir77	0.86	1521	803.09	0.08	1305.25	4,506
jcb_sm	0.58	1890	357.30	0.11	1092.98	4,506
mac	1.65	1105	403.47	0.45	1817.71	4,506
paulin	0.77	1439	354.28	0.18	1111.68	4,506
pr2	1.06	1831	489.36	0.35	1932.95	4,506

Long time scale thermal analysis

Time-domain method

- Integration-based technique
- Accurate early in time period
- Little start-up cost
- Accumulate estimation error for long time scales

Moment matching frequency-domain technique

- Uses approximate analytical expression for direct temperature calculation
- Accurate for long time scales: Doesn't accumulate error
- Higher start-up cost
- Very low cost for each new time point

Moment matching technique

$$\mathbf{C}\frac{d\mathbf{T}(t)}{dt} = \mathbf{AT}(t) + \mathbf{P}U(t)$$

Laplace Transform

$$\mathbf{T}(s) = -\mathbf{A}^{-1}(\mathbf{I} - s\mathbf{CA}^{-1})^{-1}(\mathbf{P}/s + \mathbf{CT}(0^-))$$

Expand $(\mathbf{I} - s\mathbf{CA}^{-1})^{-1}$ about $s = 0$

$$\mathbf{T}(s) = (\mathbf{m_0} + \mathbf{m_1}s^1 + \mathbf{m_2}s^2 + \dots)(\mathbf{P}/s + \mathbf{CT}(0^-))$$

Moment matching technique

Keep only first q moments

$$\mathbf{T}(s) = (\mathbf{m_0} + \mathbf{m_1}s^1 + \mathbf{m_2}s^2 + \cdots + \mathbf{m_{q-1}}s^{q-1})(\mathbf{P}/s + \mathbf{CT}(0^-))$$

Transform back to time domain, for element j

$$T_j(t) = l_{0,j} + k_{0,j}e^{p_{0,j}t} + k_{1,j}e^{p_{1,j}t} + \ldots + k_{q-1,j}e^{p_{q-1,j}t}$$

Temperatures computed directly without time-domain integration

Moment matching technique

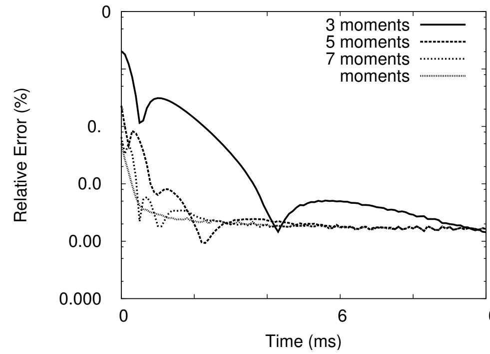
Challenges

- Matrix operations, including inversion and multiplication, are computation and memory intensive
- Model granularity critical

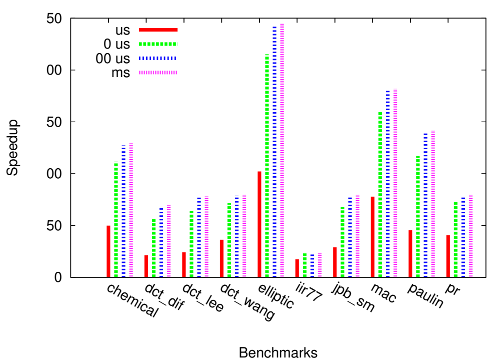
Solution

- Spatial adaptation based on predicted thermal gradient to maximize modeling efficiency
- Efficient numerical solvers, i.e., multigrid matrix inversion, LAPACK for many matrix operations

Accuracy of long time scale thermal analysis



Frequency and time domain techniques



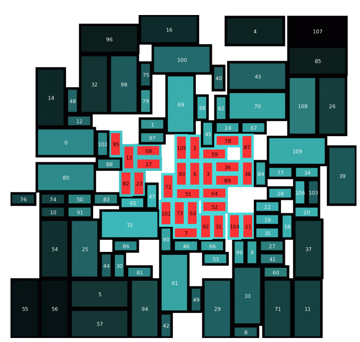
Efficiency of time of moment matching technique

Problem	Elts.	Static $\mathbf{A}^{-1}$ (s)	Static **M** mul. (s)	Static **H** coeff. (s)	Periodic (ms)	Dynamic (μs)
chemical	3,383	93.44	16.53	0.80	104.35	0.26
dct_dif	2,282	32.28	8.54	0.42	55.37	0.20
dct_lee	2,430	42.23	7.78	0.40	50.91	0.18
dct_wang	3,206	371.31	23.39	0.84	106.25	0.20
elliptic	3,009	194.44	19.46	0.74	91.31	0.20
iir77	5,862	509.74	214.84	19.58	359.65	0.21
jcb_sm	2,608	125.93	12.63	0.58	72.45	0.20
mac	2,945	221.84	17.93	0.72	90.51	0.19
paulin	2,586	66.21	8.47	0.42	54.25	0.18
pr2	3,572	287.97	31.98	1.06	132.92	0.20

Thermal optimization

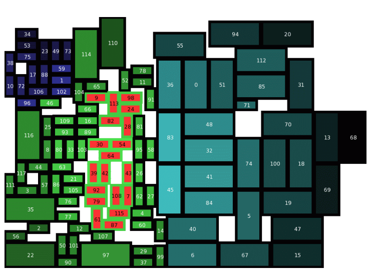
- Used ideas in thermal-aware architectural synthesis system
- Optimize area, power, temperature under performance constraints
- Required architectural and physical optimizations
- Relies on high-performance unified incremental high-level synthesis and floorplanning

Example floorplan



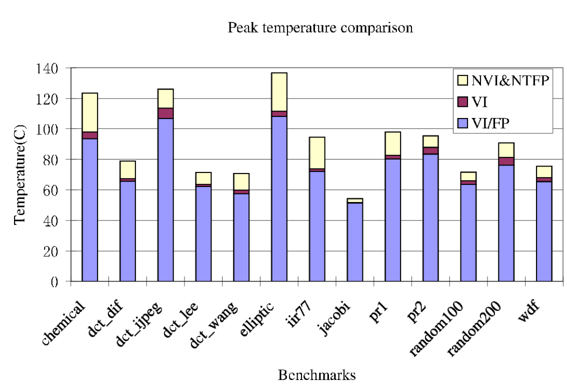
29 functional units have temperatures higher than 85°C

Floorplan with voltage islands



- Improvement: Using three voltage islands improves things
- Still have 19 functional units $> 85°\mathrm{C}$
- Use thermal-aware binding and floorplanning moves to further improve

Thermal optimization results



Thermal optimization summary

- Thermal-aware binding of operations to functional units
- Voltage island generation to reduce temperature
- Thermal-aware floorplanning
- Together, result is 12.5 °C temperature reduction
- 9.9% area reduction under peak temperature constraint

Related work I

Thermal modeling

- P. Li, L. T. Pileggi, M. Ashghi, and R. Chandra. Efficient full-chip thermal modeling and analysis. In *Proc. Int. Conf. Computer-Aided Design*, pages 319–326, November 2004
- Kevin Skadron, Mircea R. Stan, Wei Huang, Sivakumar Velusamy, Karthik Sankaranarayanan, and David Tarjan. Temperature-aware microarchitecture. In *Proc. Int. Symp. Computer Architecture*, pages 2–13, June 2003
- COMSOL Multiphysics (FEMLAB)

Related work II

Thermal modeling

Lorenzo Codecasa, Dario D'Amore, and Paolo Maffezzoni. An Arnoldi based thermal network reduction method for electro-thermal analysis. *Trans. Components and Packaging Technologies*, 26(1):168–192, March 2003

Thermal-aware synthesis

- Rajarshi Mukherjee, Seda Ogrenci Memik, and Gokhan Memik. Temperature-aware resource allocation and binding in high-level synthesis. In *Proc. Design Automation Conf.*, June 2005
- W.-L. Hung, G. Link, Y. Xie, N. Vijaykrishnan, N. Dhanwada, and J. Conner. Temperature-aware voltage islands architecting in system-on-chip design. In *Proc. Int. Conf. Computer Design*, October 2005