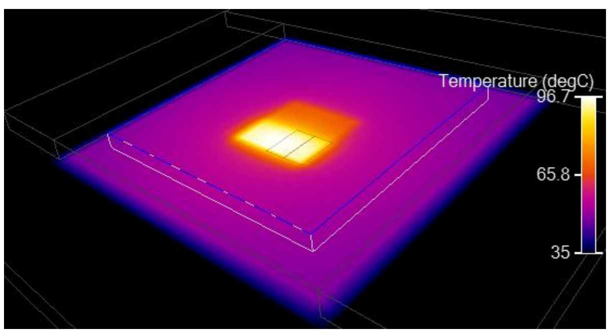
Behavioral thermal modeling for quad-core microprocessors



Duo Li and Sheldon X.-D. Tan

Department of Electrical Engineering
University of California, Riverside, CA

Murli Tirumala
Intel Corporation

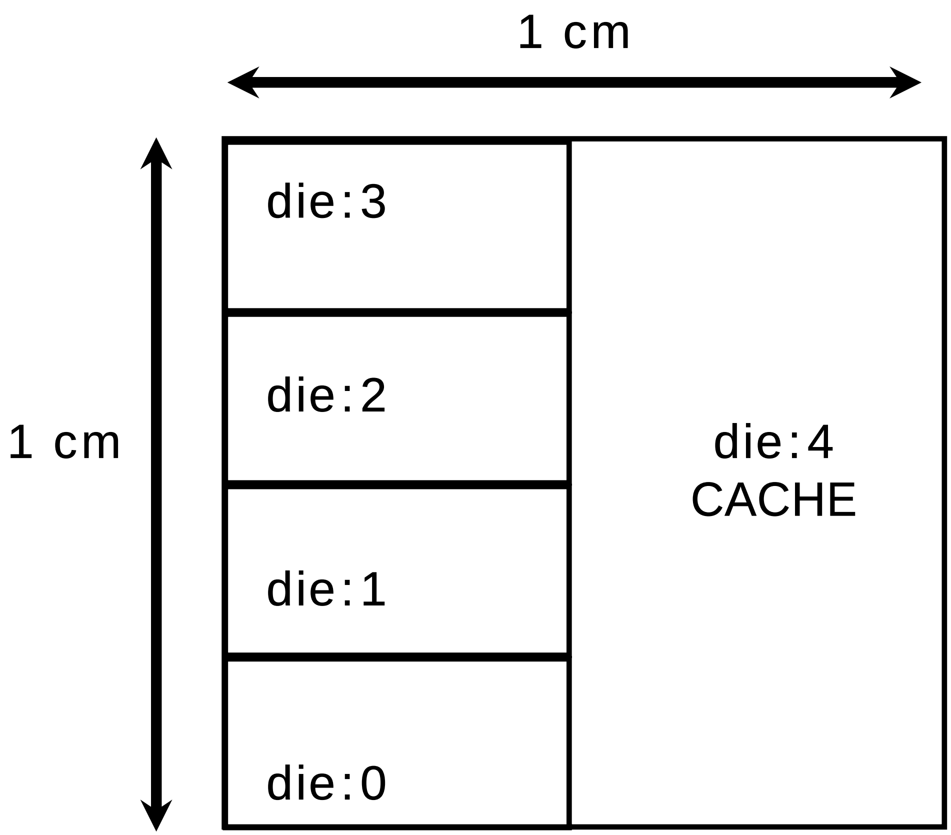
Outline

- Introduction and Motivation
  - The need for dynamic thermal management (DTM)
  - Why software thermal sensors
- Power estimation for functional units
- Architecture level thermal modeling
- Summary

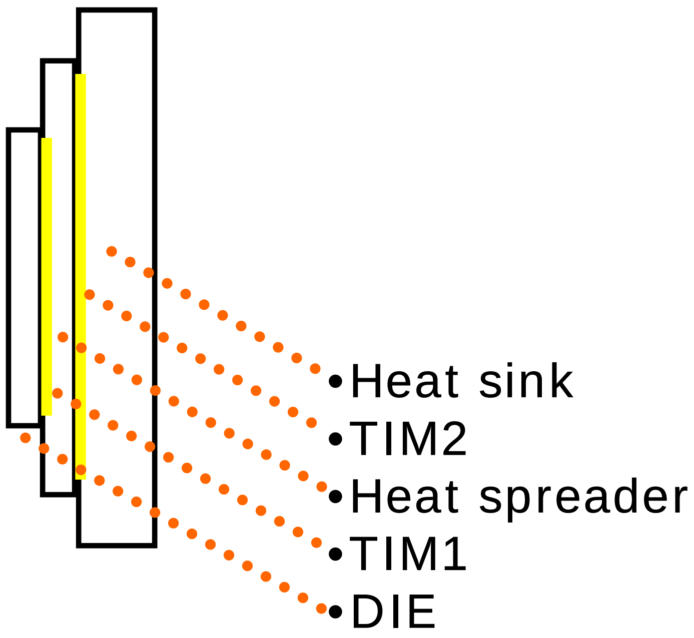
Outline

- Introduction and Motivation
- Architecture level thermal modeling
  - Intel quad-core structure
  - Transfer function
  - General pencil of function method
  - Log-sale sampling and stabilization
  - Simulation results
- Summary

Top view: quad core

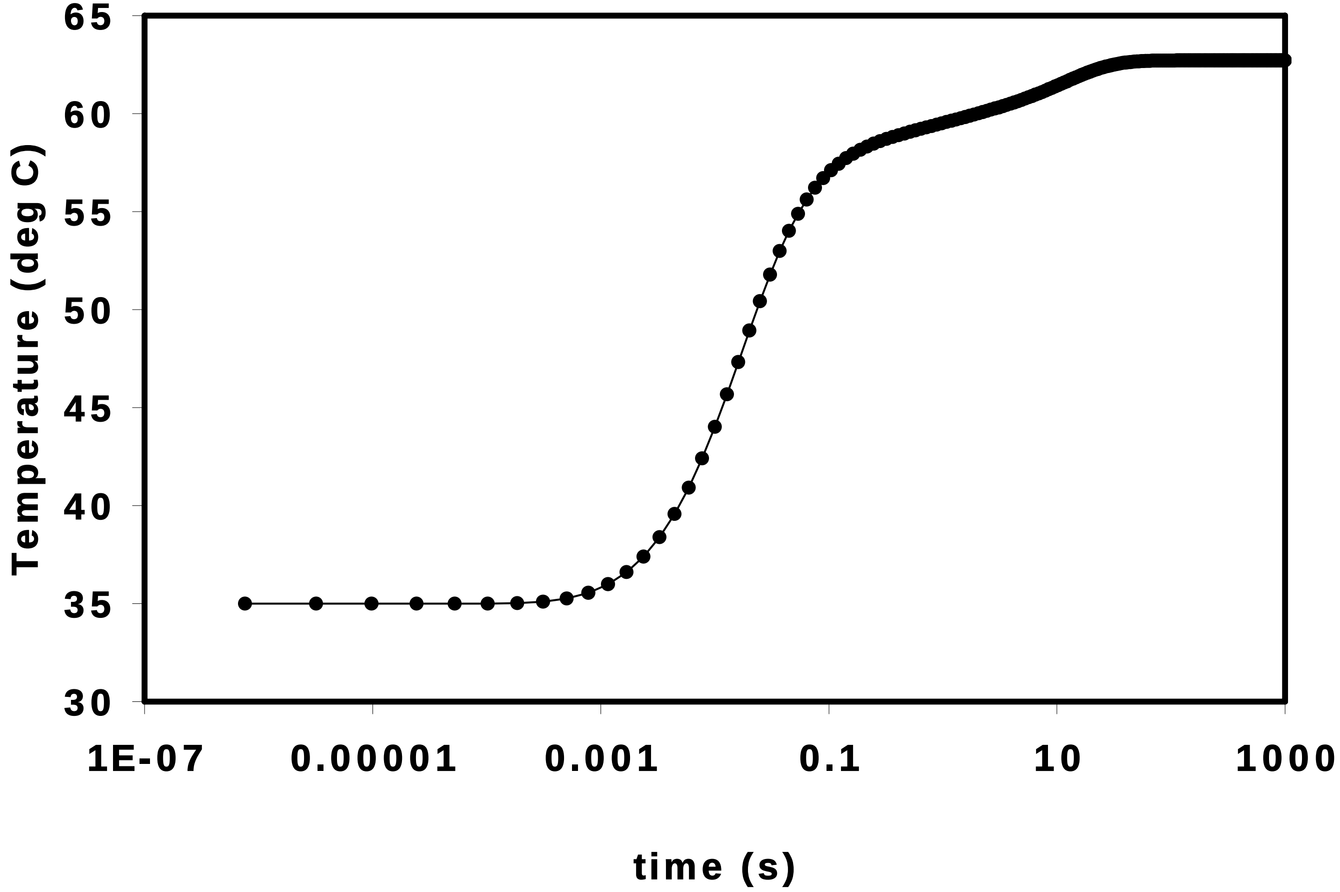


Lateral view

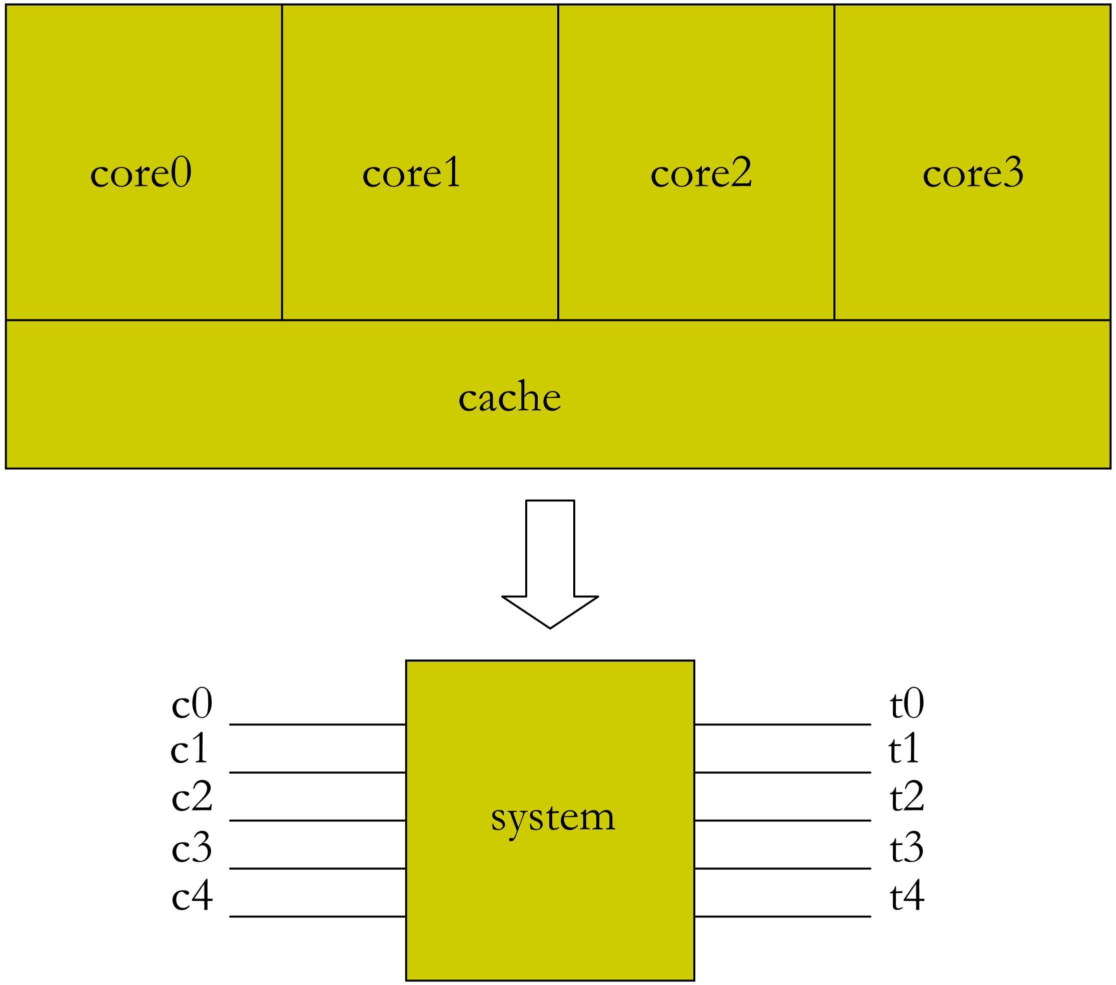


Temperatures reported are on the die bottom face and centered with each die region

Active core 0 at 20 W: T distribution



Quad-core


core0
core1
core2
core3
cache
c0
c1
c2
c3
c4
system
t0
t1
t2
t3
t4

Transfer function

LTI (linear, time-invariant systems)

input signal $x(t)$ and output $y(t)$

$$Y(s) = H(s)X(s)$$

or

$$H(s) = \frac{Y(s)}{X(s)}$$

where $H(s)$ is the transfer function of the LTI system

Impulse responses and pole-residue representation

- Pole-zero

$$H(s) = \frac{b_0 + b_1 s + \ldots + b_m s^m}{1 + a_1 s + \ldots + a_n s^n}$$

$$H(s) = K \frac{(s - z_1)\cdot(s - z_2)\ldots(s - z_m)}{(s - p_1)\cdot(s - p_2)\ldots(s - p_n)}$$

- Pole-residue and impulse response in time domain

$$H(s) = \sum_{i=1}^{n} \frac{k_i}{s - p_i}$$

$$h(t) = k_i \exp(t p_i)$$

Concepts of Matrix Pencil

- Matrix pencil

$$M(z) = Y_1 - zY_2$$

where z is a scalar valuable, Y1 and Y2 are two (square or rectangular) matrices.
M(z) decreases its rank by one if only if z is the generalized eigenvalues of M, which contain the desired information about the system like directions of the wave arrivals and the signal poles (thus the poles of the system, which generates the signals).

- Pencil-of-function

$$f(t,z) = g(t) + zh(t)$$

General pencil-of-function method

- Used for extracting poles and residues from transient signals.

$$y_k = \sum_{i=1}^{M} r_i \exp(p_i \Delta t k)$$

- k = 0, 1, …, N-1,
- r_i are the complex residues,
- p_i are the complex poles,
- Δt is the sampling interval.

N: # of samples
L: window size for GPOF,. i.e. number of samples used in GPOF.
M: # of poles used in the model.

General pencil-of-function method

- Define following pof as

$$y_0 - \lambda y_1,\ y_1 - \lambda y_2\ ,\ldots,\ y_{M-1} - \lambda y_M$$

- Define Y_1 and Y_2 as

$$Y_1 = \begin{bmatrix} x(1) & x(2) & \cdots & x(L) \\ x(2) & x(3) & \cdots & x(L+1) \\ \vdots & \vdots & & \vdots \\ x(N-L) & x(N-L+1) & \cdots & x(N-1) \end{bmatrix},$$

$$Y_2 = \begin{bmatrix} x(0) & x(1) & \cdots & x(L-1) \\ x(1) & x(2) & \cdots & x(L) \\ \vdots & \vdots & & \vdots \\ x(N-L-1) & x(N-L) & \cdots & x(N-2) \end{bmatrix}, \quad (3.3)$$

$$Z_0 = \mathrm{diag}[z_1, z_2, \ldots, z_M],$$

$$R = \mathrm{diag}[R_1, R_2, \ldots, R_M].$$

Then, we have

$$Y_1 - \lambda Y_2 = Z_1 R (Z_0 - \lambda I) Z_2.$$

So, the rank of matrix pencil reduces one when λ becomes z_i, which is the poles in the z-domain ($z_i = \exp(\Delta t p_i)$) as Y1 and Y2 span the same Subspaces of sampled signals.

General Pencil of Function Method

ALGORITHM: GPOF
Input: sampling vectors $\mathbf{y}_i = [y_i, y_{i+1}, \ldots, y_{i+N-L-1}]^T$
Output: poles vector $\mathbf{p}$ and residues vector $\mathbf{r}$

1. Construct matrices Y_1 and Y_2.

   $Y_1 = [\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{L-1}] \quad Y_2 = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L]$

2. Singular value decomposition (SVD) of Y_1. $Y_1 = UDV^H$

3. Construct matrix Z. $Z = D^{-1}U^H Y_2 V$

4. Eigen-decomposition of Z. $Z_0 = eig(Z)$

   find poles vector: $p_i = \frac{log(z_i)}{\Delta t}$

5. Solve R_1 and R_2 from $Y_1 = Z_1 R Z_2$ and $Y_2 = Z_1 R Z_0 Z_2$.

$$Z_1 = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ z_1 & z_2 & \ldots & z_M \\ \vdots & \vdots & \ldots & \vdots \\ z_1^{N-L-1} & z_2^{N-L-1} & \ldots & z_M^{N-L-1} \end{bmatrix}$$

$$Z_2 = \begin{bmatrix} 1 & z_1 & \ldots & z_1^{L-1} \\ & & \ldots & \\ \vdots & \vdots & \ldots & \vdots \\ 1 & z_M & \ldots & z_M^{L-1} \end{bmatrix}$$

find residues vector: $\mathbf{r} = \frac{R_1 + R_2}{2}$

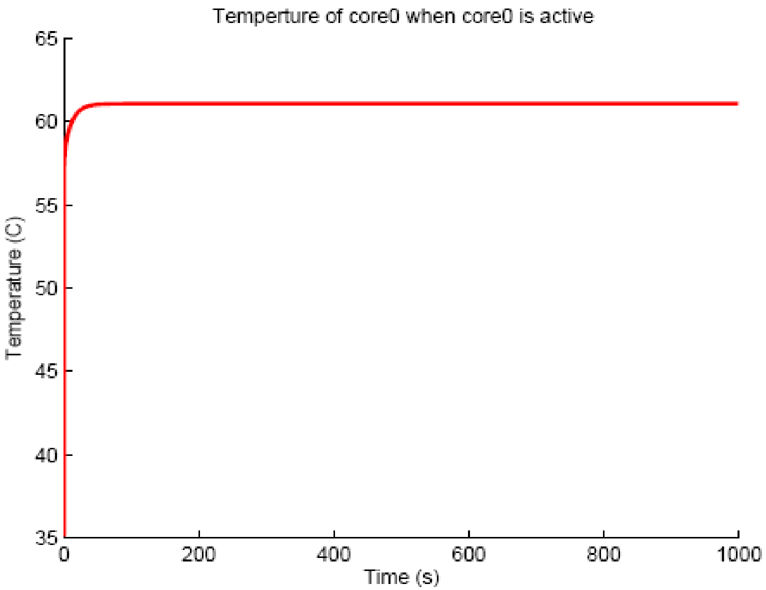
How to choose M and L

- M is model order number.
- L is sampling window size.
- N is the number of total sampled points.
- For GPOF, $M \le L \le N-M$. Allow different window sizes and pole numbers.
- Typically, choosing $L = N/2$ and $M = L$ can yield better results.

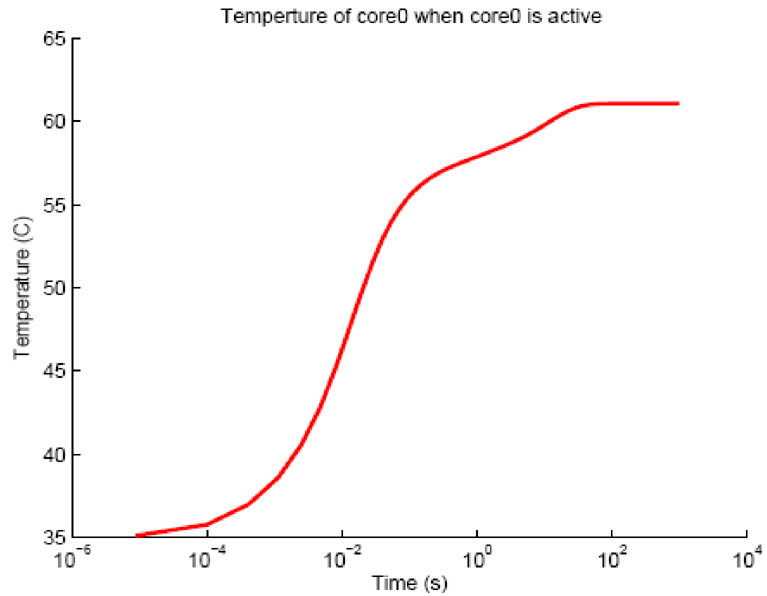
Sampling issue

- Traditional MP using constant interval time for sampling.
  - Temperature increase dramatically fast in the first few seconds.
- Log-scale sampling is a good way.
- Numerical differentiation for computing impulse response.
  - Need to compute the impulse response instead of step responses, which are given.

Linear vs Log-scale



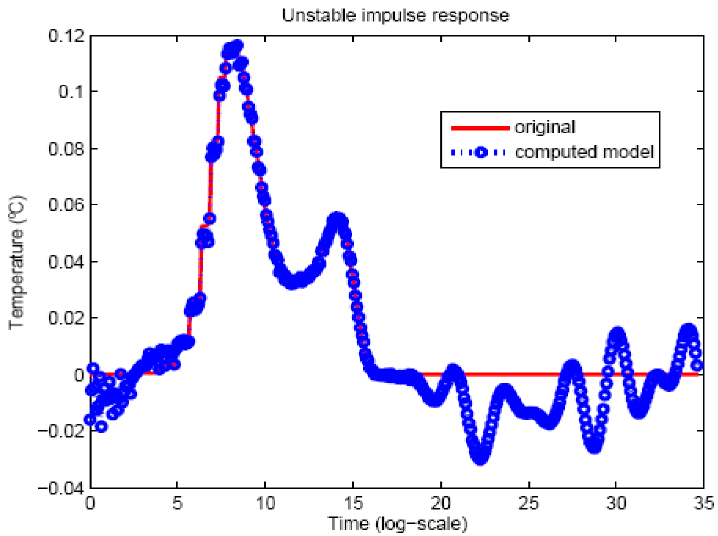
(a) Linear time scale thermal step response.



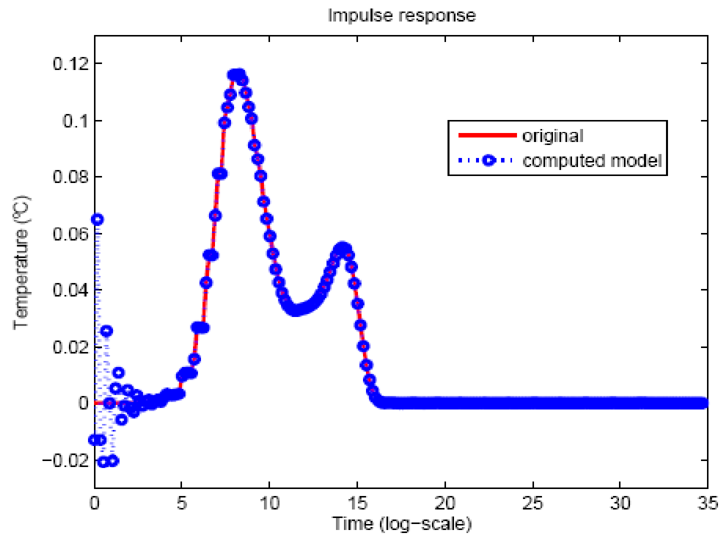
(b) Logarithmic time scale thermal step response.

Numerical Differential and Stabilization (1)

- Stable pole extraction



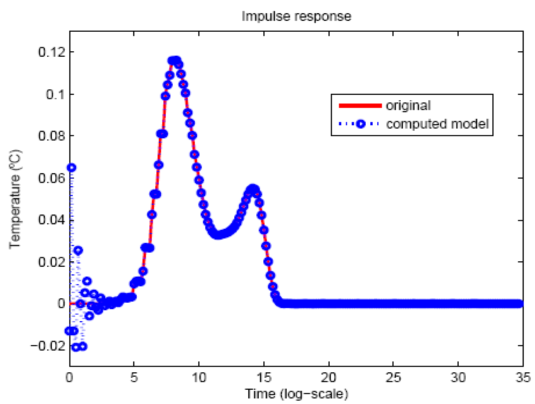
(a) Extracted impulse response with positive poles

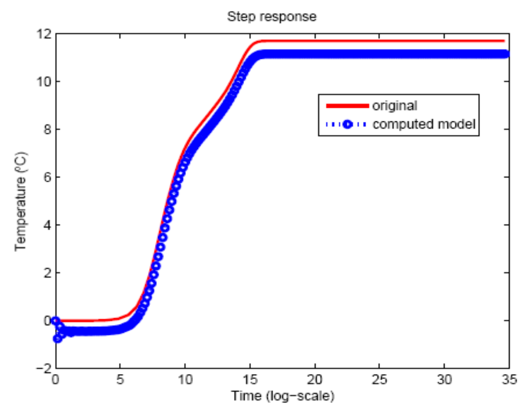


(b) Extracted impulse response with only negative poles

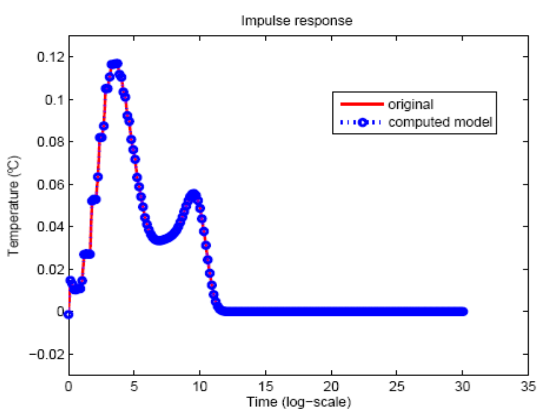
Numerical Differential and Stabilization (2)

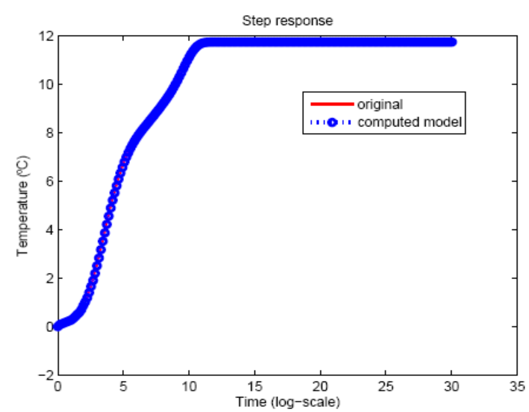
- Stabilizing the starting response





Impulse and step responses without starting time truncation.



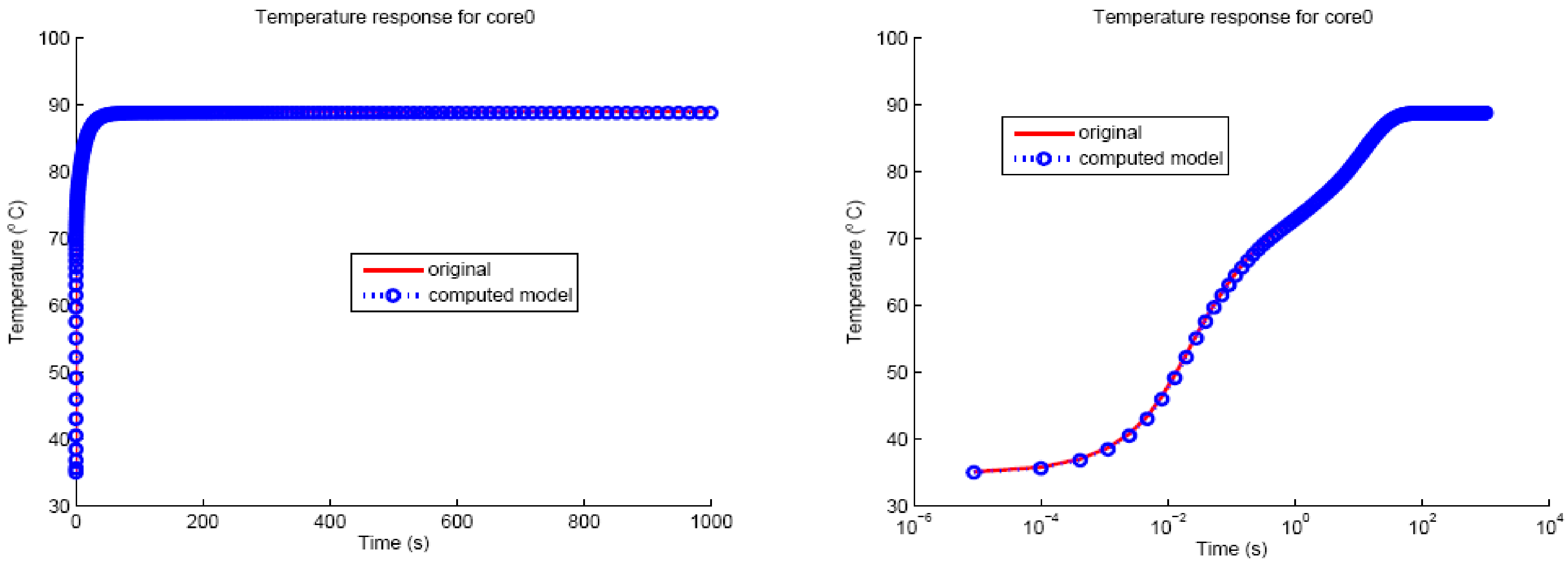


Impulse and step responses with starting time truncation.

Training

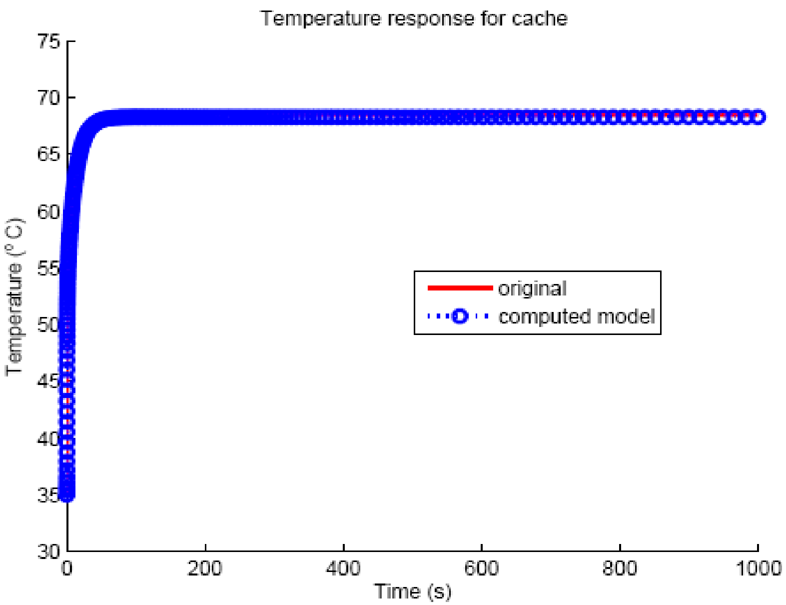
- Extracting 5 groups of poles and residues using matrix pencil method.
- Obtaining the transfer function of the system.
- Simulating the output of the system (thermal simulation).
- Linear combination

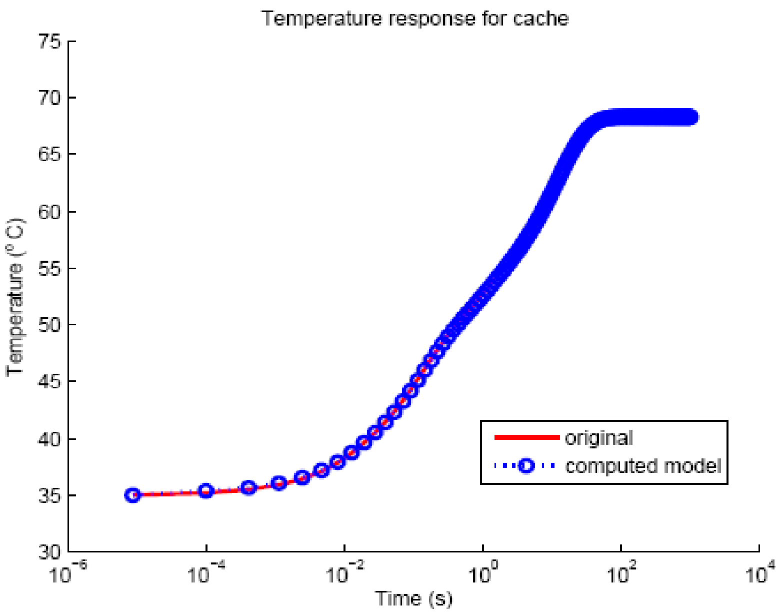
Simulation result (1)



Core0's temperature increase curve, when all the cores and cache are active (driven by 20W powers).

Simulation result (2)





Cache's temperature increase curve, when all the cores and cache are active (driven by 20W powers).

Simulation result (3)

Table 1: Difference when temperatures achieve the steady state

	Measured Temp. ($^\circ C$)	Computed Temp. ($^\circ C$)	Difference percentage
Core0	88.96	88.78	0.22%
Core1	90.60	90.52	0.08%
Core2	90.04	88.94	0.11%
Core3	88.96	88.78	0.20%
Cache	68.46	68.32	0.20%

Simulation result (4)

Table 2: Features of the difference between measured and computed temperatures

	Difference ($^\circ C$)			Difference percentage	
	Maximum	Mean	Std. deviation	Maximum	On average
Core0	0.46	0.25	0.08	0.89%	0.32%
Core1	0.27	0.18	0.07	0.42%	0.15%
Core2	0.37	0.16	0.08	0.73%	0.20%
Core3	0.46	0.24	0.08	0.88%	0.31%
Cache	0.31	0.16	0.08	0.51%	0.26%

The maximum difference is less than 0.5 ^{o}C and 1 % for all the cores.

The average difference is less than 0.3 ^{o}C and 0.3 % for all the cores.

Conclusion

- Efficient on-chip thermal analysis technique is required for on-chip dynamic thermal management study and run-timing DTM.
- Developed a new estimation method to compute real microprocessor Function Units' power.
- Developed behavioral thermal modeling techniques based on general pencil-of-function method.