FEKIS: A Fast Architecture-level Thermal Analyzer for Online Thermal Regulation *

Pu Liu†, Sheldon X.-D. Tan†, Wei Wu$^\delta$ and Murli Tirumala$^\delta$
†Department of Electrical Engineering, University of California, Riverside, CA 92521
$^\delta$Intel Corporation, 2200 Mission College Blvd, Santa Clara, CA 95052

ABSTRACT

Owning to increasing power consumption and the corresponding heat dissipated on die, efficient on-chip temperature regulation becomes imperative for today's high performance microprocessors. Temperature tracking based on the on-chip thermal sensors is not sufficient as the temperature hot spots keep changing with the load. One way to mitigate this problem is by means of software sensors, where temperature of any location is computed based on realtime power information and calibrated with the physical sensors. In this paper, we present a very efficient numerical thermal analyzer, which is suitable for fast temperature tracking and online thermal regulation. The proposed method, called FEKIS, combines two existing numerical techniques: extended Krylov subspace reduction technique to reduce the thermal circuit complexity and large-step integration method to exploits the piecewise constant power input traces, which is typical in the power traces at the architecture level. Experimental results show that FEKIS runs $10X$ faster than the precise time-step integration method only and $1000X$ faster than the traditional numerical integration method with high accuracy.

Categories and Subject Descriptors

J.6 [**Computer-Aided Engineering**]: Computer-aided design(CAD); I.6.5 [**Simulation and Modeling**]: Model Development

General Terms

Algorithms, Management

Keywords

Thermal simulation, Model reduction, Architecture

*This work is supported in part by NSF grant under CCF-0448534, NSF Grant CCF-0541456, UC Micro Program #07-101 via Intel Corp.

GLSVLSI'08, May 4–6, 2008, Orlando, Florida, USA.

1. INTRODUCTION

Higher temperature has significant adverse impacts on chip performance and reliability. This problem becomes more severe as VLSI technology scales to the nanometer ranges. The exponential power density increase will in turn lead to average chip temperature to raise rapidly [2]. Excessive on-chip temperature can cause many severe problems such as reduced reliability of chips, elevated cool cost of the packaging. One way to mitigate this problem is by means of online temperature regulation or dynamic thermal management (DTM), which dynamically reduces the temperature of some hot units in a chip via a suite of techniques such as activity migration, local toggling, dynamic voltage/frequency scaling [3, 14]. Furthermore, recent studies show that architecture level thermal management at small performance degradation cost can significantly reduce the packaging costs typically designed for worst cases [14, 15, 7].

One of the most critical aspects of thermal modeling and simulation for DTM is to efficiently capture the temperature profile changes at regular execution intervals caused by the variations of the power consumption from runtime applications at the chip architecture level or the operating system level. Physical thermal diode-based sensor has problems ranging from imprecision, delay and space overhead for hardware implementation [3, 4, 8]. These sensor noises could degrade DTM performance significantly from conservative triggering of DTM [14].

One viable alternative solution to this problem is by means of fast on-chip thermal estimation technique in software form with help of physical sensors for effective DTM application. An architecture level thermal modeling and simulation tool HotSpot [14] exploited and studied different DTM techniques in regulating microprocessor operating temperature for representative benchmark programs. However, HotSpot uses Runge-Kutta (RK) method to solve the linear differential equation and it can be slow for very long power traces from modern benchmark programs which have tens to hundreds of billions of instructions. Recently more efficient thermal simulation methods based moment matching techniques have been proposed [9, 18]. These methods compute the transient temperature changes based on frequency domain moment matching concept. But those approaches require the evaluation of exponential functions from a closed form response expressions at each simulation intervals, which can be slower at runtime simulation.

In addition to online temperature tracking, fast thermal estimation can be used for architecture level physical designs, testing during the early stages of the chip design pro-

cesses. In those applications, fast thermal evaluation for long power traces are required for many thermal-aware optimization applications such as thermal floorplaning [12], thermal-safe test scheduling [11] or thermal-aware testing [10]. Typically thermal simulation is a core function, and will be called thousands of times. So the time efficiency becomes very critical.

In this paper, we propose a new, fast extended Krylov subspace integration simulation method, FEKIS, for runtime temperature monitoring and fast off-line estimation. The new method combines two existing numerical techniques, Krylov subspace reduction technique and precise time-step integration method. It exploits the fact that the typical power trace pattern shown in Figure 1, can be approximated by the average power of each step to accurately determine the temperature tread [9]. Our new contributions are as follows: (1) We apply a precise integration method (PIM), which can be very fast for linear systems with a very long fixed time step and is driven by constant inputs in each fixed time step. This is typical power input patterns at the architecture level as many programs exhibit long instruction execution intervals, which has the same power consumptions owning to loops and regular program patterns [13], to save computation cost at runtime. (2) We apply Krylov subspace method to reduce the circuit matrices into smaller ones to speed up the simulation by roughly one order of magnitude (3) We test FEKIS on the architecture thermal circuit of a Pentium 4 Northwood core under 22 SPEC2K [1] benchmarks. We show that FEKIS becomes more efficient when finer granularity thermal models are used.

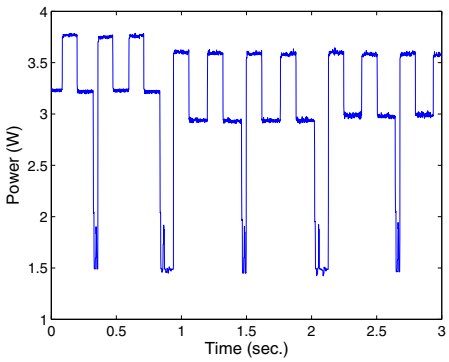

Figure 1: The power trace under SPEC2K *wupwise* **benchmark at the integer register file unit**

The rest of the paper is organized as follows: In section 2 we introduce the PIM algorithm for thermal simulation. Section 3 describes the model order reduction technique for special pulse input. We summarize the whole FEKIS algorithm in section 4. The experimental results are presented in section 5 to validate our method. Finally, section 6 concludes this paper.

2. FAST THERMAL PRECISE INTEGRATION METHOD

The PIM algorithm computes the complete response of a linear system including zero-input and zero-state responses by taking second order Taylor expansion and using so called 2^N algorithm to compute the exponential matrix required in the evaluation.

2.1 PIM for linear circuit system

For the equivalent thermal RC circuit, we can use modified nodal analysis (MNA) to formulate the thermal circuit ordinary differential equation in matrix form as:

$$G\mathbf{x}(t) + C\dot{\mathbf{x}}(t) = B\mathbf{u}(t), \tag{1}$$

where $G \in R^{n\times n}$ is the circuit conductance matrix which is stamped with thermal resistors, $C \in R^{n\times n}$ is the circuit capacitance matrix which is stamped with thermal capacitors, $\mathbf{x} \in R^n$ is the vector of node temperature, $\mathbf{u} \in R^p$ is the vector of independent power sources, and $B \in R^{n\times p}$ is the input select matrix mapping the input power sources to the internal nodes. n and p are the number of nodes and power sources.

For the architecture level thermal model, matrix C in (1) usually satisfy with two conditions, nonsingular and diagonal. So we can easily get its inverse matrix C^{-1} and transpose (1) into standard formulation:

$$\dot{\mathbf{x}}(t) = -C^{-1}G\mathbf{x}(t) + C^{-1}B\mathbf{u}(t). \tag{2}$$

The general solution of (2) can be written as:

$$\mathbf{x}(t) = e^{-C^{-1}Gt}\ \mathbf{x}(0) + \int_0^t e^{-C^{-1}G(t-\tau)}C^{-1}B\ \mathbf{u}(\tau)d\tau. \tag{3}$$

If we assume that the input power $\mathbf{u}$ becomes a constant during the fixed sampling interval Δt, the integration equation(3) can be represented as:

$$\mathbf{x}(i+1) = A\mathbf{x}(i) + AT\mathbf{u}(i) - T\mathbf{u}(i), \tag{4}$$

where

$$A = e^{-C^{-1}G\Delta t} \quad \text{and} \quad T = -G^{-1}B. \tag{5}$$

Furthermore, equation(4) can be written as:

$$\mathbf{x}(i+1) = A[\mathbf{x}(i) + T\mathbf{u}(i)] - T\mathbf{u}(i). \tag{6}$$

Therefore we can calculate the transit response at any time step by using the previous time step value.

2.2 2^N algorithm for exponential matrix approximation

In order to get the accurate results, the key is to calculate the exponential matrix A. There are many algorithms to compute the exponential matrix. Here the so called 2^N algorithm [19, 20] is applied and cooperated with second order Taylor expansion.

First we rewrite the matrix A as following form.

$$A = e^{-C^{-1}G\Delta t} = (e^{-C^{-1}G\Delta t/m})^m = (e^{H\tau})^m \tag{7}$$

where

$$H = -C^{-1}G \quad \text{and} \quad \tau = \Delta t/m. \tag{8}$$

We select m as an integer to power of 2. If $N = 20$ and $m = 2^N = 1048576$, τ becomes a very small number. After second order Taylor expansion, we have

$$e^{H\tau} \approx I + H\tau + \frac{(H\tau)^2}{2!} = I + A_a, \tag{9}$$

where $A_a = H\tau + (H\tau)^2/2!$. Since τ is selected to be a very small number, second order approximation can achieve very accurate approximation.

Because the elements of matrix A_a is very small compared with the identity matrix I, we can not directly compute A by computing $(I + A_a)$ to the power of m. However, since we select m as an integer to the power of 2, we can easily derive the following relationship.

$$\begin{aligned} A &= (I + A_a)^m \\ &= (I + A_a)^{2^N} \\ &= [(I + A_a)(I + A_a)]^{2^{N-1}} \\ &= [I + (2A_a + A_a A_a)]^{2^{N-1}} \end{aligned} \quad (10)$$

Then by using a variable replacement process, $A_a = 2A_a + A_a A_a$, we can compute A recursively. We calculate matrix A by using only N times of matrix multiplication as following:

$$\begin{array}{l} \text{for } i \leftarrow 1 \text{ to } N \text{ do} \\ \quad A_a \leftarrow 2A_a + A_a A_a \\ A \leftarrow I + A_a \end{array} \quad (11)$$

We summarize the PIM algorithm in Figure 2.

PIM ALGORITHM:
Input: G, C, B, $\mathbf{u}(t)$, and initial state $\mathbf{x}(0)$
Output: $\mathbf{x}(t)$

1. Solve T, H and τ using (5) and (8);
2. Compute A_a using (9);
3. Compute A using (11);
4. Do transient simulation using (6).

Figure 2: Fast precise integration method (PIM)

3. EXTENDED KRYLOV SUBSPACE FOR MODEL ORDER REDUCTION

The PIM algorithm we described above can achieve faster simulation than some transient integration algorithms, such as first order Backward Euler (BE), by taking advantage of the fixed time step. It is very suitable for architecture level thermal model simulation because of its small size (around 100 nodes). Original thermal model from HotSpot is quite basic, which only capture 1D heat flow into and within the thermal package, and 2D lateral heat flow among pipeline units. Recently more detailed thermal models are proposed in [5, 7], which are partition based grid models. The grid models can provide more detail information but also result more circuit nodes. When the matrices dimension becomes larger, the efficiency of the PIM algorithm is degraded. The reason is that it lose the sparsity of the thermal matrices. The matrix A we get is a dense matrix, where all elements are nonzero generally.

In our first example, there are only 488 nonzero elements and 82 nonzero elements in thermal matrix G and C with dimension of 82. As we know, the computational complexity of sparse operations is proportional to the number of nonzero elements in the matrix. Computational complexity also depends linearly on the row size m and column size n of the matrix, but is independent of the product $m*n$, the total number of zero and nonzero elements. However, the matrix A for PIM simulation has $82^2 = 6724$ nonzero elements which may not more efficient than directly solving sparse format differential equation. Fortunately this problem can be mitigated by taking model order reduction(MOR) to the thermal model. For instance, we reduce the model dimension from 82 to 10, and reduced matrix $\hat{A}$ at most has $10^2 = 100$ nonzero elements compared with 6724 in original matrix A.

3.1 Review of the extended Krylov subspace method

In this section, we are going to review moment matching based MOR technique [16], the extended Krylov subspace method (EKS) [17], for thermal simulation.

In frequency domain, (1) can be written as

$$(G + sC)X(s) = BU(s). \quad (12)$$

If we can represent the input power signal $U(s)$ as

$$U(s) = \mathbf{u}_0 + \mathbf{u}_1 s + \mathbf{u}_2 s^2 + \cdots + \mathbf{u}_i s^i + \cdots, \quad (13)$$

by taking Taylor expansion at $s = 0$ for $X(s)$, we have

$$(G+sC)(\mathbf{m}_0+\mathbf{m}_1 s+\mathbf{m}_2 s^2+\cdots) = B(\mathbf{u}_0+\mathbf{u}_1 s+\mathbf{u}_2 s^2+\cdots). \quad (14)$$

So we can derive the recursive relationship for moments generation in (15).

$$\begin{aligned} G\mathbf{m}_0 &= B\mathbf{u}_0 \\ G\mathbf{m}_1 + C\mathbf{m}_0 &= B\mathbf{u}_1 \\ &\cdots \\ G\mathbf{m}_{r-1} + C\mathbf{m}_{r-2} &= B\mathbf{u}_{r-1} \end{aligned} \quad (15)$$

Then r-order Krylov subspace cab be constructed as

$$K_r(G, C, B, U) \doteq span\{\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{r-1}\} \quad (16)$$

Those moments generated in (15) are explicit moments which are easily suffering the well known numerical stability problem when higher order moments are required for better accuracy of reduced order model [16]. Fortunately EKS presented an explicit moments generation method by applying Arnoldi algorithm to orthonomalize the basis of the Krylov subspace K_r.

Then we use matrix V, the matrix representation of Krylov subspace K_r, to perform model order reduction by taking congruence transforation to thermal matrices G, C, and B.

$$\begin{aligned} \hat{G} &= V^T G V \\ \hat{C} &= V^T C V \\ \hat{B} &= V^T B \end{aligned} \quad (17)$$

Finally we get the reduced thermal system

$$(\hat{G} + s\hat{C})\hat{X}(s) = \hat{B}U(s). \quad (18)$$

Now we can do the transient simulation in time domain on the order reduced model (18) instead of the original model (12) to save the computation cost. And the original simulation results can be obtained by taking projection from r dimension space $\hat{\mathbf{x}}$ to n dimension space $\tilde{\mathbf{x}}$ shown in (19).

$$\tilde{\mathbf{x}}(t) = V\hat{\mathbf{x}}(t). \quad (19)$$

$\tilde{\mathbf{x}}$ is the approximation of $\mathbf{x}$.

3.2 Input moments generation

In order to perform model order reduction, we need to get the right-hand-side moments for input power traces, which

means we need transform the input power signal from time domain to frequency domain.

For the thermal transient simulation algorithm described in section 2, the input power trace looks like the waveform shown in Figure 3.

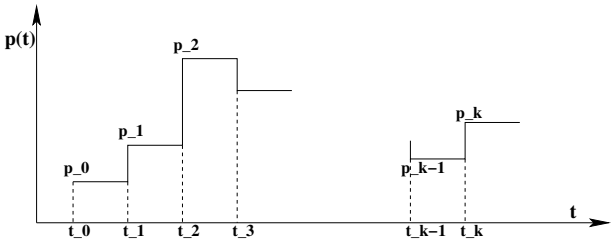

Figure 3: The waveform of input power trace for thermal transient simulation

For this special pulse input, we can describe it in time domain as

$$\mathbf{u}(t) = \sum_{i=0}^{k} \mathbf{p}_i(E(t-t_i) - E(t-t_{i+1})), \tag{20}$$

where $E(t)$ is an unit step function. After Laplace transformation, the s domain representation of input signal becomes

$$U(s) = \sum_{i=0}^{k} \mathbf{p}_i(\frac{e^{-t_i s}}{s} - \frac{e^{-t_{i+1} s}}{s}). \tag{21}$$

By taken Taylor expansion, (21) can be described as

$$U(s) = \sum_{i=0}^{k} \mathbf{p}_i\{[(-t_i) - (-t_{i+1})] + \frac{1}{2!}[(-t_i)^2 - (-t_{i+1})^2]s + \frac{1}{3!}[(-t_i)^3 - (-t_{i+1})^3]s^2 + \cdots\}. \tag{22}$$

Then we get the input moment representation as (13), where

$$\begin{aligned} \mathbf{u}_0 &= \sum_{i=0}^{k} \mathbf{p}_i\{[(-t_i) - (-t_{i+1})]\} \\ \mathbf{u}_1 &= \sum_{i=0}^{k} \mathbf{p}_i\{\frac{1}{2!}[(-t_i)^2 - (-t_{i+1})^2]\} \\ \mathbf{u}_2 &= \sum_{i=0}^{k} \mathbf{p}_i\{\frac{1}{3!}[(-t_i)^3 - (-t_{i+1})^3]\} \\ &\vdots \end{aligned} \tag{23}$$

4. FAST EXTENDED KRYLOV SUBSPACE INTEGRATION SIMULATION METHOD – FEKIS

In this section, we first summarize our FEKIS algorithm flow. Then we present a grid-based fine granularity thermal models for better accuracy.

4.1 FEKIS algorithm flow

The FEKIS algorithm flow is shown in Figure 4.

The FEKIS algorithm computes the average powers by using the sliding window based method such that a number of long fixed time steps are used. Then it compute the powers in terms of its moment forms. After the EKS method is used to reduce the original large thermal circuit matrices into small ones, those reduced circuits typically are very dense, which is suitable for the PIM method to compute the responses over a number of time internals with fixed time steps. Finally the transient responses are computed by mapping back to the original circuit nodes.

FEKIS ALGORITHM
Input: G, C, B, $\mathbf{u}(t)$, and reduced order r
Output: $\mathbf{x}(t)$

1. Compute the window-based power average for input power patterns;
2. Generate input moments $U(s)$ using (23);
3. Compute reduced order model using EKS algorithm;
4. Perform PIM for fast precise integration on reduced order model;
5. Compute original states approximation using (19).

Figure 4: FEKIS algorithm flow

For online thermal circuits simulation, we can start with any time instance with known initial temperature and compute the temperature during the given time for the computed or predicted power traces. In this way thermal profiles can be computed in a incremental way, which is suitable for online temperature estimation.

4.2 Finer granularity thermal models for functional units

Existing architecture thermal simulation assumes that each functional unit (FU) block has the uniform temperature and is treated as one node in the thermal circuit. However such model is not very accurate, since we treat the whole function block as a thermal uniform subject and ignore the gradient within the block. The other downside of block-based model is that temperature discrepancy would occur when there're blocks with radically different sizes [6].

To obtain more accurate thermal profiles within each FU, we need to have more accurate thermal models. One way is to generate more detailed thermal models by using grid-based method. The idea is that we further divided each FU into hundreds of sub-blocks. The cost is the larger thermal RC circuits, and longer simulation time.

5. EXPERIMENTAL RESULTS

We compare the experimental results by running proposed algorithm FEKIS in terms of speedup and accuracy with PIM which is without reducing model order, and first order Backward Euler(BE) method that only can handle small time-step simulation. In this way, we know the speedup contribution breakdown regarding different algorithms. we implemented these three algorithms in MATLAB 7.0.

The experimental parameters are set to $N = 20$ for algorithms PIM and FEKIS, and the reduced order is 20 for FEKIS. Since the integer register file is typically hottest [14], we show all experimental results at this FU.

5.1 Original FU block model simulation

The first example's floor-plan is shown in Figure 5. We

run 22 SPEC2K [1] benchmarks on it and sample each FU's power at $4\mu s$ time interval. For the proposed algorithms PIM and FEKIS, we set the sliding window size to 1000, which yields $4ms$ fixed simulation time interval.

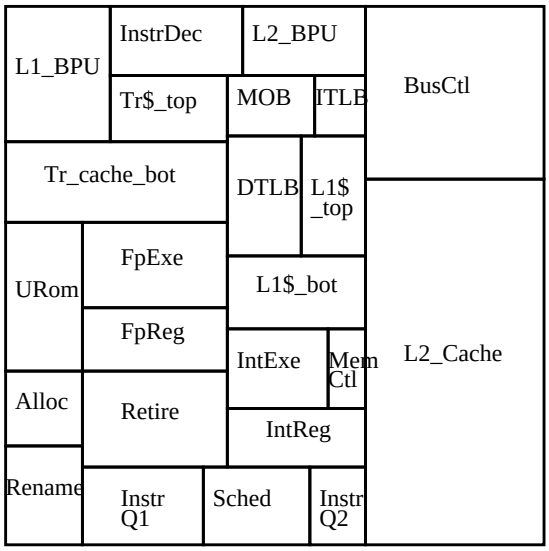

Figure 5: Pentium 4 Northwood floor-plan

In Figure 6, we show the accuracy of both proposed algorithms with or without MOR under *mcf* benchmark. Both algorithms achieve high accuracy. FEKIS has maximum error of 0.61°C and average error of 0.08°C for the whole execute time simulation, and the running time is 0.52 second compared with 1.61 second and 1861 second by PIM and BE algorithms.

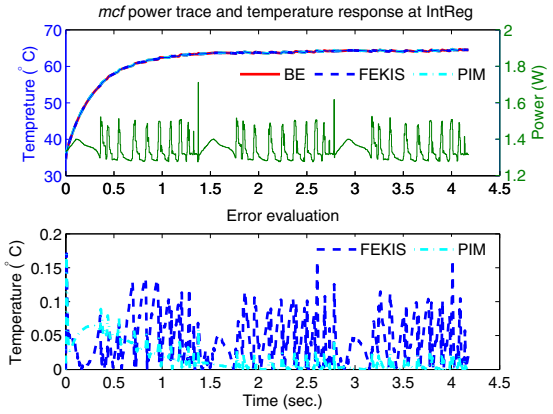

Figure 6: Performance and error evaluation on FU block model under *mcf* benchmark

We test our proposed algorithms under other benchmarks and summarize results in Table 1. For all 22 benchmarks, the average speedup of FEKIS over BE algorithm is $3958X$, which is very significant. But FEKIS only achieves to average $2.9X$ speedup over PIM because of the small thermal model size. Although there are some maximum errors beyond 1°C, we observe that the maximum errors mostly happen at the beginning of simulation and last for very short time due to the initial dramatic input power change.

5.2 Finer granularity grid thermal model simulation

The second example uses the same microprocessor as the first example. However, we do the regular partition for each FU block. In our experiment, we evenly partition each FU block into 3×3 grids. Therefore, the node number of new thermal model changes from 82 to 658. Then the FEKIS algorithm shows better efficiency than the PIM algorithm.

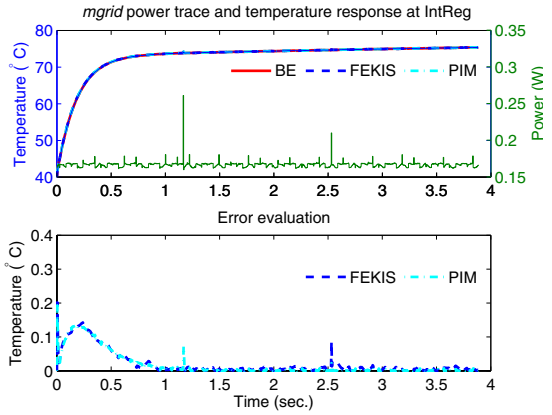

Figure 7: Performance and error evaluation on FU grid model under *mgrid* benchmark

First we present a temperature response shown in Figure 7 under *mgrid* benchmark. The proposed method FEKIS still obtain accurate results with maximum error 0.20°C and average error 0.02°C. The speedup over the PIM algorithm is more obvious, which is $14X$ vs. $3X$.

Similarly we run all 22 benchmarks on the 3×3 grid model. From Table 1, we can see the speedup of the MOR aided precise integration method, FEKIS, is more advantageous than the PIM algorithm which is without model reduction, and achieves averagely $25X$ speedup. The speedup coming from the PIM method is still very impressive. As a result, the proposed FEKIS method is a very efficient architecture level thermal simulation method. We believe that FEKIS will deliver more speedup for very detailed thermal circuits.

6. CONCLUSION

In this paper, we have presented a novel architecture thermal analysis algorithm, FEKIS, for fast temperature estimation of high performance microprocessors and platforms. The new efficient algorithms can be used as software sensors for dynamic thermal management and architecture level thermal estimation for early stages of the microprocessor design. The new algorithm combines the Krylov subspace for reduction and the large-step integration, which exploits the long input power traces, to speed up the simulation. Experimental results showed that FEKIS can lead to about $10X$ speedup over its precise time-step integration algorithm without model reduction and about $1000X$ faster than the numerical integration method BE with high accuracy.

7. REFERENCES

[1] http://www.spec.org/cpu2000/CFP2000/.
[2] International technology roadmap for semiconductors(itrs), 2004 update, 2004. http://public.itrs.net.

Table 1: Performance evaluation of FEKIS

SPEC2K	Execute Time(s)	Original block FU model					Finer granularity grid model				
		Sim. (s)	vs. BE	vs. PIM	Max(°C)	Ave(°C)	Sim. (s)	vs. BE	vs. PIM	Max(°C)	Ave(°C)
ammp	6.26	1.24	3289	2.7	0.36	0.05	9.59	3450	7.4	0.24	0.03
applu	3.70	0.34	4732	3.3	0.18	0.05	3.52	4177	16.3	0.27	0.04
apsi	6.08	0.83	4724	4.0	1.19	0.15	10.39	3190	6.8	0.35	0.07
art	14.16	7.09	2572	2.4	0.10	0.02	49.87	2838	3.7	0.33	0.03
bzip	2.72	0.28	3125	2.0	0.56	0.08	2.08	4455	27.0	0.28	0.06
crafty	1.42	0.09	3656	3.0	0.21	0.06	0.64	5590	65.0	0.31	0.09
equake	1.64	0.11	3691	2.3	0.69	0.05	1.14	4578	38.2	0.24	0.07
facerec	4.54	0.48	4623	4.0	0.40	0.13	5.24	3757	11.0	0.24	0.05
fma3d	5.15	0.84	3423	2.7	0.91	0.08	6.77	3556	9.2	0.66	0.11
gap	1.56	0.09	3989	3.0	0.21	0.06	0.97	4865	49.0	0.26	0.06
gcc	1.64	0.13	3608	2.4	1.21	0.27	1.05	5626	63.1	0.42	0.08
gzip	2.02	0.13	4331	3.0	0.94	0.17	1.61	5376	40.9	0.40	0.05
lucas	3.77	0.27	5615	4.4	0.61	0.08	4.98	3517	14.7	0.40	0.05
mcf	4.18	0.52	3635	3.1	0.61	0.08	5.88	3667	12.2	0.14	0.03
mesa	2.44	0.19	3868	2.6	0.51	0.05	1.89	5514	34.9	0.34	0.04
mgrid	4.05	0.50	3532	2.9	0.60	0.03	5.63	3796	14.1	0.20	0.02
parser	3.14	0.36	3428	2.5	0.41	0.09	3.41	4719	20.4	0.26	0.04
swim	6.48	1.52	2827	2.4	1.18	0.05	13.70	3715	6.7	0.14	0.01
twolf	1.38	0.56	5468	5.0	0.80	0.02	9.47	3679	8.9	0.40	0.01
vortex	2.30	0.16	5469	2.4	0.64	0.10	1.87	5005	32.4	0.27	0.06
vpr	3.12	0.28	4132	2.6	0.62	0.07	3.31	4451	20.2	0.24	0.04
wupwise	1.85	0.14	3343	2.1	0.42	0.09	1.19	5133	47.9	0.24	0.04
Average	-	-	**3958X**	**2.9X**	**0.61**	**0.08**	-	**4302X**	**25.0X**	**0.30**	**0.05**

[3] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *Proc. of Intl. Symp. on High-Performance Comp. Architecture*, pages 171–182, 2001.

[4] S. Gunther, F. Binns, D. Carmean, and J. Hall. Managing the impact of increasing microprocessor power consumption. In *Intel Technology Journal*, First Quarter 2001.

[5] L. He, W. Liao, and M. R. Stan. System level leakage reduction considering the interdependence of temperature and leakage. In *Proc. Design Automation Conf. (DAC)*, pages 12–17, 2004.

[6] W. Huang, S. Ghosh, K. Sankaranarayanan, K. Skadron, and M. R. Stan. Hotspot: Thermal modeling for cmos vlsi systems. *IEEE Transactions on Component Packaging and Manufacturing Technology*, to appear.

[7] W. Huang, M. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy. Compact thermal modeling for temperature-aware design. In *Proc. Design Automation Conf. (DAC)*, pages 878–883, 2004.

[8] K. Lee and K. Skadron. Using performance counters for runtime temperature sensing in high performance processors. In *the Workshop on High-Performance, Power-aware Computing(HP-PAC), in conjunction with the 2005 International Parallel and Distributed Processing Symposium*, Apr. 2005.

[9] H. Li, P. Liu, Z. Qi, L. Jin, W. Wu, S. X.-D. Tan, and J. Yang. Efficient thermal simulation for run-time temperature tracking and management. In *Proc. IEEE Int. Conf. on Computer Design (ICCD)*, pages 130–133, Oct. 2005.

[10] C. Liu, V. Iyengar, and D. K. Pradhan. Thermal-aware testing of network-on-chip using multiple-frequency clocking. In *Procedings of the 24th IEEE VLSI Test Symposium*, pages 46–51, 2006.

[11] P. Rosinger, B. M. Al-Hashimi, and K. Chakrabarty. Thermal-safe test scheduling for core-based system-on-chip integrated circuits. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 25:2502–2512, Nov. 2006.

[12] K. Sankaranarayanan, S. Velusamy, M. R. Stan, and K. Skadron. A case for thermal-aware floorplanning at the microarchitectural level. *The Journal of Instruction-Level Parallelism*, to appear.

[13] T. Sherwood, E. Perelman, and B. Calder. Basic block distribution analysis to find periodic behavior and simulation points in applications. In *the International Conference on Parallel Architectures and Compilation Techniques(PACT2001)*, pages 3–14, 2001.

[14] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature aware microarchitecture. In *Proc. IEEE International Symposium on Computer Architecture (ISCA)*, pages 2–13, 2003.

[15] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature aware microarchitecture: Extended discussion and results. In *University of Virginia, Dept. of Computer Science, Technical Report CS-2003-08*, Apr. 2003.

[16] S. X.-D. Tan and L. He. *Advanced Model Order Reduction Techniques in VLSI Design*. Cambridge University Press, 2007.

[17] J. M. Wang and T. V. Nguyen. Extended Krylov subspace method for reduced order analysis of linear circuit with multiple sources. In *Proc. Design Automation Conf. (DAC)*, pages 247–252, 2000.

[18] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan. Efficient power modeling and software thermal sensing for runtime temperature monitoring. *ACM Trans. on Design Automation of Electronics Systems*, 12(3):26, 2007.

[19] W. X. Zhong and F. W. Williams. A high precise time step integration method. *Journal of Mechanical Engineering Science*, 208:427–430, 1994.

[20] W. X. Zhong, J. Zhu, and X. X. Zhong. On a new time integration method for solving time dependent partial differential equations. *Computer Methods in Applied Mechanics Engineering*, 130:163–178, 1996.