

New Power Budgeting and Thermal Management Scheme for Multi-Core Systems in Dark Silicon

Hai Wang*, Ming Zhang*, Sheldon X.-D. Tan[†], Chi Zhang*, Yuan Yuan[‡], Keheng Huang[§], and Zhenghong Zhang[§]

*School of Microelectronics & Solid-State Electronics,

University of Electronic Science & Technology of China, Chengdu, 610054 China

[†]Department of Electrical Engineering, University of California, Riverside, CA 92521 USA

[‡]School of Automation Engineering, University of Electronic Science & Technology of China, Chengdu, 610054 China

[§]Southwest China Research Institute of Electronic Equipment, Chengdu, 610036 China

Abstract—As integrated circuit technology advances, we are now facing the utilization wall of the chip, which prevents us from turning on all the cores at the same time and leads to dark silicon. In order to maximize performance without violating thermal threshold for dark silicon chips, we propose a new power budgeting and dynamic thermal management scheme. Utilizing model predictive control, the new method automatically determines power budget values as well as active core positions considering transient thermal effects. Then, a dynamic thermal management method with task migration and DVFS is used to assign appropriate task loads to active cores according to the previously determined power budget. Experiments on multi-core systems with different number of cores and different dark silicon ratios have shown the new method is able to calculate the power budget accurately, keep the chip in safe temperature range, and retain the system in high performance.

I. INTRODUCTION

As CMOS scaling continues according to Moore's law, more and more transistors are integrated on a chip, leading to elevated heat dissipation problems. Recently, as the transistor density increases further, we are facing a utilization wall of the chip: only a portion of the transistors can be turned on at the same time in order to satisfy the heat dissipation constraint. For multi/many-core chips, this means that only a limited number of cores can operate simultaneously while the other cores should be turned off, resulting in the so called dark silicon areas [1], [2].

There are dark silicon control techniques proposed to determine how many cores can be turned on to maximize the performance without violating the thermal constraints [1]. However, only determining the number of active cores is not enough to guarantee the safety and performance of chip. One reason is that the running application itself plays a vital role in power dissipation: with the same operating voltage and frequency, some applications consume more power than others in average. Even for one application itself, its power

consumption changes over time. Another important reason is that besides determining the number of active cores, activating the correct cores is equally important. All these reasons make power budgeting very important after the number of active cores have been determined. There is a power budgeting work [3] which uses thermal design power (TDP) to determine the power budget for dark silicon chips, which balances the power and performance using control based method. Recently, thermal safe power (TSP) [4] method has been proposed, which improves the TDP based methods by providing power budget as a function of the number of active cores. However, TSP method computes the power budget based on static information without transient thermal/power considerations. As a result, it is unable to adjust the power budget to accommodate the transient thermal/power changes. In this work, we solve this problem by proposing a transient state aware power budgeting method.

After power budgeting process, how to assign and adjust tasks according to the power budget is also important. In this work, we introduce a task migration and DVFS based dynamic thermal management method (DTM) to do the job. Task migration based DTM method [5], [6], [7], [8], [9], [10], which switches tasks among cores, is able to distribute the current available tasks/powers according to the calculated power budget. On the other hand, the DVFS based DTM method [11], [12], [13], which reduces the power by lowering the operating voltage and frequency, is able to assist the task migration in dealing with the tasks/powers which cannot be assigned properly according to power budget. We will show that by combining both task migration and DVFS, the new method is able to handle tasks effectively according to the given power budget.

In this work, we solve the dark silicon control problem with a new power budgeting technology combined with dynamic thermal management (DTM). We propose a power budget strategy with the number of active cores provided. The new strategy is able to consider the transient power/thermal behaviors, and automatically determine which cores should be activated in order to maximize the performance of the chip.

This work is supported in part by National Natural Science Foundation of China under grant No. 61404024, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, in part by the Open Foundation of State Key Laboratory of Electronic Thin Films and Integrated Devices under grant No. KFJJ201409.

After the budgeted powers are acquired, a task migration and DVFS based DTM method is applied to allocate and adjust the tasks, in order to make sure no temperature constraint is violated and the budgeted powers are properly and effectively used.

II. BASICS OF THERMAL MODELING AND DYNAMIC THERMAL MANAGEMENT

In this section, we will first introduce the thermal model used in this paper. Next, dynamic thermal management techniques [14], such as task migration and dynamic voltage and frequency scaling (DVFS) will be briefly presented.

A. Thermal modeling basics

Based on the well known similarity between electronic system and thermal system, we can analyze the thermal system using similar ordinary differential equations for RC networks. It is even more convenient for us to choose a fixed time step, and use the Euler's method to discretize the differential equation into the difference equation, which will be used in this paper as

$$\begin{aligned} t(k+1) &= A_m t(k) + B_m P(k), \\ t_c(k) &= C_c t(k). \end{aligned} \quad (1)$$

In (1), we assume the chip has m cores and n thermal parts (thermal parts include all the cores and some other parts). Then, $t \in \mathbb{R}^{n \times 1}$ represents temperatures of all thermal parts. $A_m \in \mathbb{R}^{n \times n}$ and $B_m \in \mathbb{R}^{n \times m}$ are the parameter matrices which contain information of thermal conductance, thermal capacitance and the topology of the chip. $t_c(k) \in \mathbb{R}^{m \times 1}$ is the vector containing cores' temperatures at time k . $P(k) \in \mathbb{R}^{m \times 1}$ denotes the power input at time step k , while $C_c \in \mathbb{R}^{n \times m}$ is the selection matrix to select the cores' temperatures from t . Please note that for dark silicon problems, not all cores can be turned on at the same time. Assume there are only q active cores at time step k , the power vector $P(k)$ must have $m - q$ zero entries.

B. Introduction to dynamic Thermal Management

DTM methods such as DVFS and task migration have been used to adjust the trade-off ratio between performance and reliability of the chip. It is quite effective in traditional single-core or multi/many-core chips, and we will show that it can also be integrated into the power/thermal regulation framework in dark silicon chips.

Many DTM methods are based on task migration technique. For a multi/many-core chip, there are thermal sensors monitoring the temperatures of the cores. If a high temperature is detected, the corresponding task, which causes the high temperature, will be moved to a low temperature core, in order to minimize the temperature difference across the chip. As shown in Fig. 1, the sensor detects that core 5 reaches the dangerous temperature, then task migration takes effect to exchange its task with the one in core 9, which has a relatively lower temperature. After the task migration process, both temperatures of core 5 and core 9 may converge to the

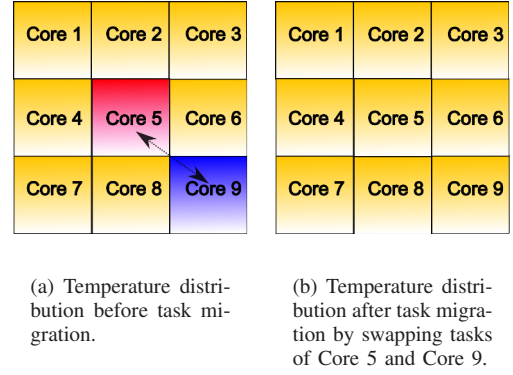


Fig. 1. Example of task migration. Where red core means the core has higher temperature than average, and blue core indicates the core has lower temperature.

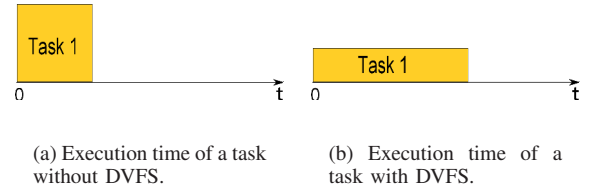


Fig. 2. Execution time of a one-task and one-core system. The area of the rectangle means the task, and the height of the rectangle means the power consumed, and the width represents the execution time.

average temperature. However, task migration alone is unable to avoid the high temperature problem, especially when all cores are processing heavy loaded tasks under extremely high temperatures.

DVFS is a different DTM technique from task migration, which can be used in both single-core and multi/many-core systems. Once a high temperature is detected, DVFS reduces the voltage and frequency of the corresponding core to lower its temperature. However, the performance of the cores under DVFS will suffer a great loss. As Fig. 2 shows, if we use the area of the rectangle to represent the task, the height of the rectangle to indicate the power consumed, and the length to show the time consumed, it is obvious that once DVFS is performed, the task execution time will become longer.

III. NEW POWER BUDGETING METHOD FOR DARK SILICON

In this section, we will present the new power budgeting method. In dark silicon problems, the cores are divided into two categories: active cores and idle cores. With the given number of active cores, the new power budgeting method determines which cores should be active ones, and also determines the maximum power can be consumed by each active core with consideration of both thermal constraint and transient thermal behavior.

The goal of our power budgeting method is to maximize the total power of all active cores, considering both transient thermal behavior and thermal constraint. However, considering

transient thermal behavior and avoid violating the thermal constraint is not an easy task, since it requires the new method to be aware of current power/thermal status and its influence in the future. In order to solve the problem, we utilize model prediction method (MPC), and modify it to adapt the special requirements in dark silicon problem. In this paper, we ignore the detailed MPC derivation process due to page limitation. Interested readers are referred to [15] for a comprehensive discussion of MPC.

The MPC based power budgeting method extends the transient thermal model in (1) into the predictive form with the ability to look into the future and compute the *future-aware* power budget. The most important property of such power budget is that it is calculated in MPC with the ability to *track* a user defined temperature profile. In order to maximize the budget power under a thermal threshold, we simply set the thermal threshold for all core $t_t \in \mathbb{R}^{m \times 1}$ as the user defined temperature profile to be tracked. We further expand the thermal threshold vector into the future n_p time steps, and generate a vector

$$T_t = [t_t^T, t_t^T, \dots, t_t^T]^T \in \mathbb{R}^{mn_p \times 1}, \quad (2)$$

where n_p is called the *prediction horizon*. In this way, MPC is going to compute the power which leads to the given temperature profile T_t (already set to be as high as possible without violating the thermal constraint) from current step to the future n_p steps.

Similar to T_t , we further introduce the predicted core temperatures in MPC model within the prediction horizon as

$$T_c = [t_c(k+1|k)^T, t_c(k+2|k)^T, \dots, t_c(k+n_p|k)^T]^T, \quad (3)$$

where $t_c(k+j|k)$ contains the predicted core temperatures at the time $(k+j)$ using the power at current time k . T_c can be expressed in the following way by MPC model using power information from current and previous steps:

$$T_c(k) = Mt_r(k) + N(P(k) - P(k-1)), \quad (4)$$

where M , N and t_r are shown in (5) on top of the next page, and n_c in (5) represents *control horizon* in MPC, similar to n_p for prediction horizon.

In order to make the predicted temperature T_c to track the thermal threshold T_t , it is straightforward to minimize the difference between them, and formulate the following optimization problem, with the constraints modified to adapt the dark silicon requirements:

$$\begin{aligned} & \text{minimize} && \| T_t - T_c \|_2 \\ & \text{subject to} && \text{card}(P(k)) = n_a, \\ & && T_c \leq T_t, \end{aligned} \quad (6)$$

where n_a represents the number of active cores, $\text{card}(P(k))$ is the cardinality function and its output is the number of nonzero components of $P(k)$. Please also note that T_c is a function of $P(k)$ as shown in (4). The first constraint in (6) is to guarantee the number of active cores follow the dark

silicon requirement, by forcing $P(k)$ to be sparse with only n_a nonzero elements. The second constraint is to make sure that each core's temperature will not go above the threshold. Since $P(k)$ is the only variable in (6), we will get the power budget $P(k)$ at current time by solving the optimization problem (6).

There is a good property of (6): by solving it and obtain the corresponding $P(k)$, we can *automatically determine* which cores should be activated and which cores should be deactivated in order to maximize the power budget. It is very intuitive and also verified in [4] that separating active cores helps the chip to dissipate heat and increases the power budget of the whole chip. Minimizing the cost function in (6) also tends to spatially separate the active cores: please note that T_c contains all core's temperature information, including both active cores (or more accurately, to be determined to be active) and idle cores. The active cores lead to higher temperatures at their positions comparing to the idle cores. In order to minimize $\| T_t - T_c \|_2$ with the two constraints, T_c and T_t should have exactly the same temperatures at the elements corresponding to active cores. While for the elements representing the idle cores, the values of T_c should be lower than T_t . For the same number of active cores, separating them spatially leads to higher average temperature (thus smaller valued $\| T_t - T_c \|_2$) than making them clustered together [4]. As a result, solving the optimization problem (6) automatically determines which cores should be active.

In order to solve the optimization problem (6), we rewrite it as

$$\begin{aligned} & \text{minimize} && \| NP(k) - e \|_2 \\ & \text{subject to} && \text{card}(P(k)) = n_a, \\ & && NP(k) - e \leq 0, \end{aligned} \quad (7)$$

where $e = T - Mt_r(k) + NP(k-1)$. It is clearly a regressor selection problem. But solving this problem directly is very time consuming. Instead, we find an approximated solution by solving the following alternative optimization problem [16]

$$\begin{aligned} & \text{minimize} && \| NP(k) - e \|_2 \\ & \text{subject to} && \| P(k) \|_1 < \alpha, \\ & && NP(k) - e \leq 0, \end{aligned} \quad (8)$$

where α is a positive number. We change the value of α in a bisection search way, and solve the convex optimization problem for $P(k)$. The searching of α stops at finding the correct $P(k)$ with $\text{card}(P(k)) = n_a$. The resulted $P(k)$ serves as the computed power budget for the chip.

IV. DYNAMIC THERMAL MANAGEMENT AFTER POWER BUDGETING

In the previous section, we have shown the power budgeting method. But we still need to develop the algorithm to use the budget correctly and efficiently. In this section, the corresponding dynamic thermal management is proposed for that reason.

$$\begin{aligned}
M &= \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{n_p} \end{bmatrix}, & N &= \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{n_p-1}B & CA^{n_p-2}B & CA^{n_p-3}B & \cdots & CA^{n_p-n_c}B \end{bmatrix}, \\
A &= \begin{bmatrix} A_m & 0_m \\ C_c A_m & I \end{bmatrix}, & B &= \begin{bmatrix} B_m \\ C_c B_m \end{bmatrix}, & C &= [0_m \quad I], & t_r(k) &= \begin{bmatrix} t_c(k) - t_c(k-1) \\ t_c(k) \end{bmatrix}.
\end{aligned} \tag{5}$$

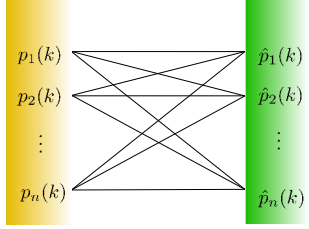


Fig. 3. Example of the bipartite figure. $p_i(k)$ means the predicted power and $\hat{p}_j(k)$ denotes the original power. The solid lines are the edges.

There are two popular dynamic thermal management techniques available: task migration and DVFS as already introduced in Section II-B. Since we have the power budget in hand for each core, it is intuitive to use task migration to assign the active cores with tasks suitable for their power budget. However, because there may exist imperfect assignment in the task migration process and the number of active cores could also be optimistic (i.e., total real power could be higher than the total budgeted power), DVFS is used after the task migration process to make sure every core is following its power budget.

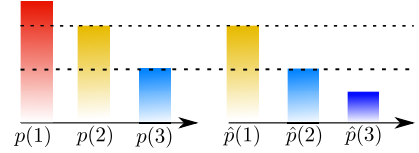
A. Power assignment by task migration

At current time, we have both power budget $P(k) = \{p_1(k), p_2(k), \dots, p_m(k)\}$ for all m cores, and real power distribution $\hat{P}(k) = \{\hat{p}_1(k), \hat{p}_2(k), \dots, \hat{p}_m(k)\}$ for all cores. We can simply assign $\hat{P}(k)$ to the correct cores according to the budget $P(k)$ as discussed before.

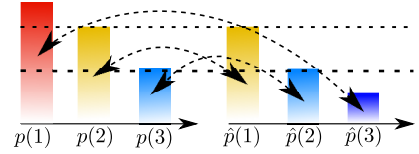
This is an assignment problem. At first, we build a weighted complete bipartite graph $G = (P(k), \hat{P}(k), E)$. One example of the graph is shown in Fig. 3. For each edge $e_{ij} \in E$, its weight is calculated as

$$w_{ij} = \begin{cases} p_i(k) - \hat{p}_j(k) & 0 \leq p_i(k) - \hat{p}_j(k) < w_{th}, \\ \infty & \text{else,} \end{cases} \tag{9}$$

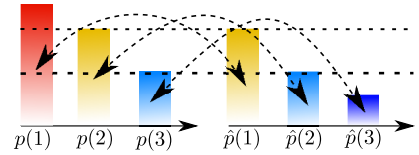
where the infinity weight means that we never assign a power to a core whose power budget is smaller nor the power budget is much larger (larger than threshold w_{th}). The goal of the assignment problem is to find the matching pairs between $P(k)$ and $\hat{P}(k)$ with the smallest total weights of the linking edges in all pairs. It can be solved by the Hungarian algorithm [17], which is a polynomial time algorithm.



(a) $p(1)$, $p(2)$ and $p(3)$ are budgeted powers. $\hat{p}(1)$, $\hat{p}(2)$ and $\hat{p}(3)$ are real powers by current running tasks.



(b) The first matching method.



(c) The second matching method.

Fig. 4. Example of two different matching decisions.

We also consider one special matching case in Fig. 4, where we use the double-headed arrow to represent the matching. There exists two possible matchings: one is $(p(2), \hat{p}(1))$, $(p(3), \hat{p}(2))$, $(p(1), \hat{p}(3))$, and the other one is $(p(1), \hat{p}(1))$, $(p(2), \hat{p}(2))$, $(p(3), \hat{p}(3))$. Both are correct in the view of the assignment problem, and they even have the same cost. However, the first matching will cause one very low temperature on core 1, but relatively high temperatures on core 2 and 3. While the second matching may lead to similar temperatures for all three cores. In terms of reliability, we prefer the second matching, and we can realize this preference by magnifying the large weights with a tuning parameter.

B. DVFS for final power adjustment

Task migration cannot guarantee perfect power assignment according to power budget, i.e., there should have some

unmatched powers left, especially there may have large powers which cannot be assigned to any active core. We are able to handle these powers using DVFS technique.

Assume there are q unmatched powers for both real side and budget side. First, we collect the unmatched powers into the sets $P_l(k) = \{p_{l1}, p_{l2}, \dots, p_{lq}\}$ and $\hat{P}_l(k) = \{\hat{p}_{l1}, \hat{p}_{l2}, \dots, \hat{p}_{lq}\}$. We can categorize the unmatched powers into two cases, and use different ways to handle them.

Case one: The unmatched real powers are much larger in average value than the budget powers. Thus, we perform DVFS on the unmatched real powers, with the power scale ratio

$$\gamma = \frac{\text{avg}(P_l(k))}{\text{avg}(\hat{P}_l(k))}. \quad (10)$$

Next, another round of assignment can be performed with the scaled real powers.

Case two: The unmatched real powers have similar average value comparing with the budget powers. In this case, we simply remove the infinity condition in (9), and get all the remaining powers matched. Then, for a matched pair $(p_{li}(k), \hat{p}_{lj}(k))$, if there is $p_{li}(k) < \hat{p}_{lj}(k)$, DVFS is performed on the real power with scaling ratio

$$\gamma = \frac{p_{li}(k)}{\hat{p}_{lj}(k)}. \quad (11)$$

Otherwise, we do not perform DVFS, because the real power is smaller than the budget.

V. EXPERIMENTAL RESULTS

The experiments are performed on the Dell T620 workstation with two 2.90GHz 8-cores 16-thread CPUs and 64 GB memory. We use the HostSpot [18] software to get the thermal model of different systems, with the number of the cores ranging from 9 to 100. The size of the chip is $9mm \times 9mm \times 0.15mm$. Wattch [19] is employed to get the power and instruction information with SPEC [20] benchmarks. The ambient temperature is set as 20°C . In order to avoid frequent task migration and DVFS, we set the power budgeting and DTM activating cycle to be every 10s. In order to show the effectiveness of the new method, we have also compared it with the recently proposed power budgeting method TSP [4].

The 9-core microprocessor is used as illustration because its simple structure makes the figures easy to read.

Fig. 5 shows the results of the new power budgeting method at two different time steps, allowing 4 cores to be active at the same time. As shown in the figure, the active cores are automatically chosen by the power budgeting method, and they are chosen to be spatially separated.

Next, we test the DTM method together with the power budgeting method. In Fig. 6 (a), the transient temperature of the 9-core chip without any power budgeting and DTM methods are given. The four active cores in this test are randomly chosen. It can be seen that temperatures at certain active cores are high, which may harm the reliability of the chip. At the same time, there are active cores with low temperatures, not

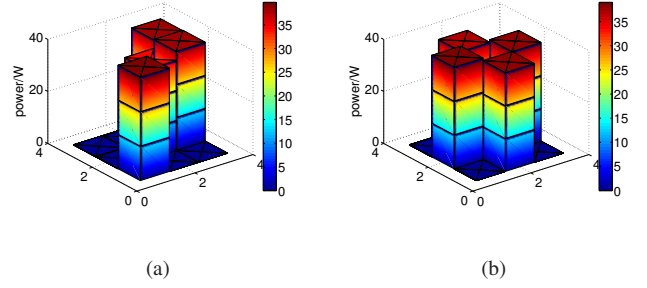


Fig. 5. The power budget generated by the new method for the 9-core microprocessor with 4 cores active at two different times.

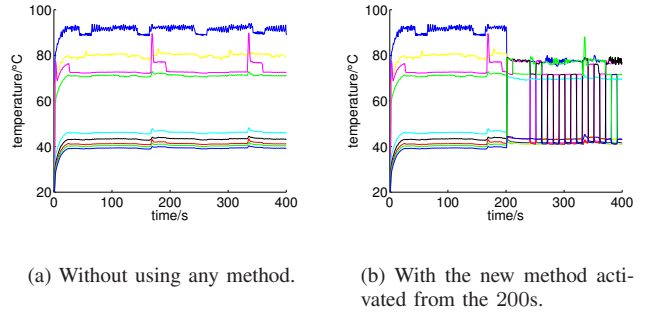


Fig. 6. Transient temperature of the 9-core microprocessor with 4 active cores (temperatures of idle cores is identified as the ones around 40°C).

to mention the idle cores, which means there are performance boost potentials on these cores. Then, we test our new power budgeting method combined with DTM. We set the threshold temperature to be 80°C , and activate our method at the time of 200s. The result is shown in Fig. 6 (b). For the first 200s, the transient temperature lines are the same as in Fig. 6 (a), because there is no power budgeting and DTM method used. Beginning from 200s, the active cores and idle cores begin to switch as automatically determined by the power budgeting method. The tasks also begin to migrate among cores every 10s as set in the experiment, and DVFS is also performed

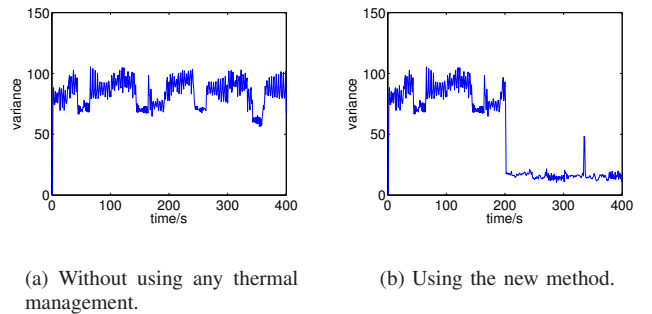


Fig. 7. Transient temperature variances of the active cores of the 9-core system with 4 active cores.

TABLE I
SYSTEM PERFORMANCE COMPARISON WITH DIFFERENT NUMBER OF CORES AND DIFFERENT DARK SILICON RATIOS.

Core #	Active #	M_n	M_d	M_t	Active #	M_n	M_d	M_t	Active #	M_n	M_d	M_t
9	2	174.72	146.11	170.77	4	323.26	287.40	288.14	6	415.12	374.99	348.93
16	3	130.89	115.05	125.77	6	234.40	211.05	227.51	9	408.32	371.91	360.29
25	5	108.32	98.28	100.62	10	215.26	193.39	215.26	15	328.04	297.81	320.60
36	7	90.23	82.99	85.69	14	186.51	168.88	170.32	22	290.38	263.77	276.76
49	10	90.97	80.81	88.73	20	174.53	157.26	166.75	30	265.40	242.23	237.44
64	13	76.39	68.36	73.93	26	174.53	157.26	159.85	38	233.30	213.45	221.28
81	16	71.04	64.17	66.39	32	138.72	125.33	118.79	49	216.14	194.26	183.80
100	20	65.15	58.53	58.91	40	136.03	123.52	130.08	60	204.49	186.11	195.81

according to the power budget. The temperatures at the four active cores stay around the temperature threshold all the time, showing the effectiveness of the new method.

We have also recorded the temperature variance data in Fig. 7. The variance of the active cores reveals the differences in temperatures of the active cores. We can see from the figure that after 200s, the variance of active cores' temperatures are significantly lowered comparing to the first 200s and the one without DTM method.

Finally, we test the performances of the systems with our new method, comparing with DVFS only in DTM, and the TSP power budgeting method [4]. The performance of the systems is measured as million instructions per second (MIPS). We also collect data with different ratio of cores set to be active, including 20%, 40%, and 60%. The results are collected in Table I, where M_n refers to MIPS of our new method, M_d refers to MIPS of DVFS only method, and M_t denotes MIPS of TSP method. From the table, it is clear that our new method is always better than DVFS. Because by using task migration, we have avoided a lot of unnecessary DVFS actions. Our method also performs better than the TSP method in performance. The major reason is that TSP is a static method which cannot consider transient thermal information, while the new method is able to adjust power budget according to current and predicted thermal/power information.

VI. CONCLUSION

In this paper, we have proposed a new power budgeting and dynamic thermal management scheme. Integrated with the model predictive control, the new method automatically determines power budget values as well as active core positions considering transient thermal effects. Then, a dynamic thermal management method with task migration and DVFS is used to assign appropriate task loads to active cores according to the previously determined power budget. Experiments on multi-core systems with different number of cores and different dark silicon ratios have shown the new method is able to calculate the power budget accurately, keep the chip in safe temperature range, and retain the system in high performance. It also outperforms the recently proposed static power budgeting method TSP.

REFERENCES

[1] M. Shafique *et al.*, "The EDA challenges in the dark silicon era," in *Proc. Design Automation Conf. (DAC)*, June, 2014, pp. 1–6.

[2] H. Esmaeilzadeh *et al.*, "Dark silicon and the end of multicore scaling," *IEEE Micro*, vol. 32, no. 3, pp. 122–134, May 2012.

[3] T. S. Muthukaruppan *et al.*, "Hierarchical power management for asymmetric multi-core in dark silicon era," in *Proc. Design Automation Conf. (DAC)*, May 2013.

[4] S. Pagani *et al.*, "TSP: Thermal safe power - efficient power budgeting for many-core systems in dark silicon," in *Proc. International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2014.

[5] M. Powell, M. Goma, and T. Vijaykumar, "Heat-and-Run: Leveraging SMT and CMP to manage power density through the operating system," in *Proc. of International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, October 2004, pp. 260–270.

[6] Y. Ge, P. Malani, and Q. Qiu, "Distributed task migration for thermal management in many-core systems," in *Proc. Design Automation Conf. (DAC)*, June 2010, pp. 579–584.

[7] T. Chantem, S. Hu, and R. Dick, "Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 10, pp. 1884–1897, October 2011.

[8] G. Liu, M. Fan, and G. Quan, "Neighbor-aware dynamic thermal management for multi-core platform," in *Proc. European Design and Test Conf. (DATE)*, March 2012, pp. 187–192.

[9] R. Ayoub and T. Rosing, "Predict and act: Dynamic thermal management for multi-core processor," in *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, August 2009, pp. 99–104.

[10] T. Ebi, M. Al Faruque, and J. Henkel, "TAPE: Thermal-aware agent-based power economy for multi/many-core architectures," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2009, pp. 302–309.

[11] K. Skadron *et al.*, "Temperature-aware microarchitecture," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, 2003, pp. 2–13.

[12] R. Jayaseelan and T. Mitra, "A hybrid local-global approach for multi-core thermal management," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2009, pp. 314–320.

[13] A. Mutapcic *et al.*, "Processor speed control with thermal constraints," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 56, no. 9, pp. 1994–2007, September 2009.

[14] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, June 2006, pp. 78–88.

[15] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, 2009.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2006.

[17] H. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[18] W. Huang *et al.*, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 501–513, May 2006.

[19] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, 2000, pp. 83–94.

[20] J. L. Henning, "SPEC CPU 2000: Measuring CPU performance in the new millennium," *IEEE computer*, vol. 1, no. 7, pp. 28–35, July 2000.