Technical reports in Computer and Information Science

Report number 2009:4

An Energy Efficient Technique for Temperature-Aware Voltage Selection

by

Min Bao, Alexandru Andrei, Petru Eles and Zebo Peng
minba@ida.liu.se, alean@ida.liu.se, petel@ida.liu.se, zebpe@ida.liu.se

Linköping University
INSTITUTE OF TECHNOLOGY

Department of Computer and Information Science
Linköping University
SE-581 83 Linköping, Sweden

Technical reports in Computer and Information Science are available online at
Linköping Electronic Press: http://www.ep.liu.se/ea/trcis/

An Energy Efficient Technique for Temperature-Aware Voltage Selection

M. Bao, A. Andrei, P. Eles, Z. Peng
Embedded Systems Laboratory (ESLAB)
Department of Computer and Information Science, Linköping University, Sweden
{minba, alean, petel, zebpe}@ida.liu.se

Abstract

High power densities in current SoCs result in both huge energy consumption and increased chip temperature. This paper proposes a temperature-aware dynamic voltage selection technique for energy minimization and presents a thorough analysis of the parameters that influence the potential gains that can be expected from such a technique, compared to a voltage selection approach that ignores temperature. In addition to demonstrating the actual percentages of energy that can be saved by being temperature aware, we explore some significant issues in this context, such as the relevance of taking into consideration transient temperature effects at optimization, the impact of the percentage of leakage power relative to the total power consumed and of the degree to which leakage depends on temperature.

1. Introduction

Energy efficiency has become a major concern for the designers of embedded systems. Due to increasing demands on performance, embedded applications are frequently implemented on multiprocessor systems on chip (SoC). Very often they are required to satisfy strict timing constraints and are functioning with a limited energy budget. One of the preferred approaches for reducing the overall energy consumption is dynamic voltage selection (DVS). This technique exploits the available slack times by reducing the voltage and frequency at which the processors operate and, thus, achieves energy efficiency.

Several DVS techniques have been proposed in literature [7], [9], [10], [21]. All these approaches neglect leakage power and only scale the supply voltage. However, recent trends in submicron CMOS technology are resulting in the fact that a substantial portion of the power dissipation is due to leakage. Therefore, adaptive body biasing (ABB) techniques have been proposed in order to reduce leakage power by increasing the threshold voltage through body biasing [8], [14]. The combined dynamic supply voltage selection and adaptive body biasing problem has been studied in [1] [23].

The high power densities achieved in current SoCs do not only result in huge energy consumption but also lead to increased chip temperatures. High temperatures can impact reliability as well as cooling and package cost. One aspect, of particular interest for this paper, is that growing temperature leads to an increase in leakage power and, consequently, energy, which, again, produces higher temperatures. Several approaches to thermal aware system-level design have been proposed in recent years. Of particular importance in this context is the development of adequate temperature modeling and analysis tools. The approach proposed in HotSpot [6] is based on elaborating an equivalent circuit of thermal resistances and capacitances corresponding to the architecture blocks and to the elements of the thermal package. HotSpot performs both static analysis (producing steady state temperature) and dynamic analysis (producing temperature profiles). A similar approach is proposed in [22] where dynamic adaptation of the resolution is performed, in order to speed up the analysis. Simpler, analytical temperature models, which are much less accurate, have been proposed in [3], [19].

Based on the available temperature models, several system-level design problems have been approached. Thermal aware task allocation and scheduling have been addressed in [20]. In [19] an approach to task scheduling under peak temperature constraints is presented. Design space exploration for multiprocessor SoC architectures under area and thermal constraints is presented in [12], while in [17] thermal aware floorplanning is advocated. As highlighted above, there exists a cyclic dependency between consumed energy and temperature, which is particularly strong in current high leakage technologies. Since DVS techniques are supposed to reduce energy consumption by adapting voltage levels, it could be assumed that temperature is an important parameter to be taken into consideration at voltage selection. Nevertheless, the temperature issue has been completely ignored in the proposed DVS techniques for real-time embedded systems. One exception is [13] which takes into consideration the effect of temperature on leakage at voltage scaling, in the context of a design process aimed at reducing peak temperature. The approach does not consider ABB techniques for leakage optimization and ignores dynamic temperature analysis.

In this paper we propose a technique for temperature aware energy minimization by DVS, considering both supply voltage selection and ABB. We consider both static and dynamic temperature analysis in our optimization process. Furthermore, we perform, for the first time, a thorough analysis of the parameters that influence the potential gains that can be expected from a thermal aware DVS technique, compared to an approach that ignores temperature. In addition to demonstrating the actual percentages of energy that can be saved by being temperature aware, we explore some significant issues in this context, such as the relevance of taking into consideration transient temperature effects, the impact of the percentage of leakage power relative to the total power consumed and of the degree to which leakage depends on temperature.

The paper is organized as follows: In Section 2 we introduce the application and power model as well as the voltage selection technique used in the paper. Section 3 describes our temperature analysis method, while in Section 4 we present the proposed temperature aware voltage selection approach. Experimental results and conclusions are presented in Sections 5 and 6, respectively.

2. Preliminaries

2.1. System and Application Model

We consider systems realized as heterogeneous multiprocessor architectures on chip. We assume that the processors can operate in several discrete execution modes. An execution mode is characterized by a pair of supply and body bias voltages: (V_{dd}, V_{bs}). As a result, an execution mode has an associated frequency and power consumption (dynamic and leakage) that can be calculated using equ. (1) and (2).

The functionality of the application is captured as a set of task graphs. In a task graph $G(\Pi, \Gamma)$, nodes $\tau \in \Pi$ represent computational tasks, while edges $\eta \in \Gamma$ indicate data dependencies between tasks (communication). Tasks are annotated with deadlines that have to be met at run-time. We assume that the task graphs are mapped and scheduled on the target architecture, i.e., it is known where and in which order tasks and communications take place. For each task the worst case number of cycles to be executed is given.

2.2. Power Model and Voltage Selection

For dynamic power we use the following equation [4], [14]:

$$\mathrm{P}_{dyn} = C_{eff} * f * V_{dd}^2 \tag{1}$$

where C_{eff}, V_{dd}, and f denote the effective switched capacitance, supply voltage, and frequency, respectively.

The leakage power is expressed as follows [11], [13], [14]:

$$P_{leak} = I_{sr} * T^2 * e^{(\frac{a*V_{dd} + \beta*V_{bs} + \gamma}{T})} * V_{dd} + |V_{bs}| * I_{Ju} \tag{2}$$

where I_{sr} is the reference leakage current at reference temperature. T is the current temperature, V_{bs} is the body bias voltage, and I_{ju} is the junction leakage current. a, β and γ are curve fitting circuit technology dependent coefficients.

Circuit delay and operational frequency are depending on the supply and body bias voltage [14]:

$$f = \frac{1}{d} = \frac{((1 + K_1) * V_{dd} + K_2 * V_{bs} - V_{th1})^{\alpha}}{K_6 * Ld * V_{dd}} \tag{3}$$

where Ld is the logic depth. K_1, K_2, K_6, and V_{th1} are technology dependent coefficients. α reflects the velocity saturation imposed by the used technology (common values $1.4 < \alpha < 2$).

In [1] we have presented an approach to combined supply voltage selection and adaptive body biasing. Given a multiprocessor architecture and a mapped and scheduled application, as presented in Section 2.1, the DVS algorithm calculates the appropriate execution modes (V_{dd} and V_{bs}) for each task, such that the total energy consumption is minimized. Another input to the algorithm is the dynamic power profile of the application, which is captured by the average switched capacitance of each task. This information will be used for calculating the dynamic energy consumed by the task in a certain execution mode, according to equ. (1). Leakage energy, during the optimization process, is calculated based on equ. (2). However, since leakage strongly depends on temperature, an obvious question is which temperature to use for leakage calculation. Ideally, it should be the temperature at which the chip will work when executing the application. This temperature, however, is not known, since the algorithm is just calculating the voltages at which to run the system and these voltages are influencing the energy dissipation which, again, is determining the temperature. As mentioned in Section 1, this issue has been ignored for the voltage scaling algorithms proposed in literature (if they, at all, consider leakage). The algorithm in [1] requires the designer to introduce an assumed temperature which is used at energy optimization. This, of course, leads to suboptimal results, since the temperature used for energy calculation during voltage selection is different from the actual temperature at which the chip works. Therefore, using the calculated voltages, the chip will dissipate more energy than with voltages that would be obtained knowing the real temperature at which the chip is going to function.

In the following, based on our approach in [1], we will develop a temperature aware voltage selection technique. Then, based on extensive experiments, we will investigate the effectiveness of temperature aware voltage selection.

3. Temperature Analysis

Temperature analysis in our proposed DVS technique is based on HotSpot [6]. The basic idea of HotSpot is to build an equivalent circuit of thermal resistances and capacitances capturing both the architecture blocks and the elements of the thermal package. HotSpot can be used both for static analysis, in which case it produces a temperature at which the circuit is supposed to function in steady state, as well as for dynamic analysis, producing temperature profiles. For our purposes, the architecture is modeled at core level. Thus, from the architecture point of view, the actual blocks whose temperature is analyzed are the processor cores on which the tasks are executed. When provided with the physical/thermal parameters (size and placement of blocks, thermal capacitances and resistances, parameters of packaging elements) and the power profile capturing the power dissipation of each core, HotSpot produces the steady state temperature or the temperature profile of the cores. However, the temperature analysis does not support the case in which power dissipation is dependent on the temperature, which, obviously, is the situation with leakage.

Solutions to overcome the above problem, in the case of static temperature analysis, have been proposed in [5], [16], [18]. A similar solution is used by us and is outlined in Fig. 1. As mentioned, corresponding to an input power profile for each core, HotSpot will produce a steady state temperature at which the core is supposed to work. However, to input the leakage component of the power profile, the working temperature has to be known. In order to overcome this deadlock situation, the process is started with an "assumed" temperature and then continued iteratively until the produced temperature converges. At this steady state temperature the dissipated heat is in balance with the heat removal capacity of the package. However, it can happen that such a balance is not achieved, due to insufficient heat removal, and the temperature is increasing, potentially, to infinite. In such a case, the iterations in Fig. 1will not converge. This phenomenon, called *thermal runaway*, is detected and indicates that the design is incorrect from the thermal point of view. Detecting thermal runaway is an important part of a thermal aware design process.

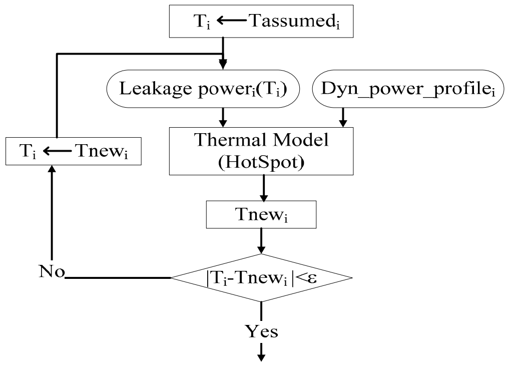
$T_i \longleftarrow Tassumed_i$
Leakage power$_i$(T_i)
Dyn_power_profile$_i$
$T_i \longleftarrow Tnew_i$
Thermal Model
(HotSpot)
$Tnew_i$
No
$|T_i\text{-}Tnew_i|<\varepsilon$
Yes

Fig. 1 Static thermal analysis with leakage (the index i indicates that the particular item is introduced/produced for each core)

Static analysis assumes that, eventually, the chip will function at one constant temperature. This, however, is not necessarily the case in reality. In the context of a variable power profile, the chip will not reach a constant steady state temperature but a steady state in which temperature is varying according to a certain pattern. In order to obtain the steady state temperature profile, we need to use dynamic thermal analysis. For dynamic analysis, HotSpot is calculating temperatures at successive time steps [6]. At each step a new temperature is calculated for each block by solving the equations describing the thermal model, based on a fourth-order Runge-Kutta method. The power consumption during the time interval between two steps is extracted from the power profile for the respective block. However, leakage power is a function of the temperature and, thus, cannot be delivered as an input to the analysis.

In order to solve the above problem we have extended the thermal analysis such that the power consumption during a time step is calculated as the sum of two components: (1) the dynamic power extracted from the input power profile and (2) the leakage power calculated at the temperature level of the previous step. The process is illustrated in Fig. 2. Temperature analysis is repeated for successive periods of the application. In order to detect convergence, temperature values at corresponding time steps of these successive periods are compared.

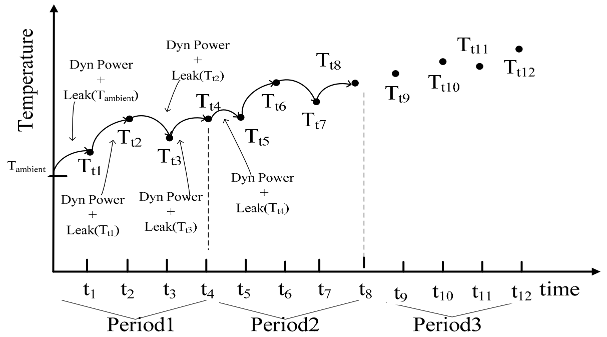
Temperature
$T_{ambient}$
Dyn Power + Leak($T_{ambient}$)
Dyn Power + Leak(T_{t2})
Dyn Power + Leak(T_{t1})
Dyn Power + Leak(T_{t3})
Dyn Power + Leak(T_{t4})
T_{t1}
T_{t2}
T_{t3}
T_{t4}
T_{t5}
T_{t6}
T_{t7}
T_{t8}
T_{t9}
T_{t10}
T_{t11}
T_{t12}
t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} time
Period1
Period2
Period3

Fig. 2 Dynamic thermal analysis with leakage

For both static and dynamic analysis, convergence is reached efficiently except, of course, if thermal runaway occurs. Since dynamic thermal analysis itself is much more time consuming than static analysis, obtaining a steady state temperature profile is much slower than calculating a constant steady state temperature. Our experimental results will further elaborate on this aspect.

4. Temperature-Aware Voltage Selection

In Fig. 3. we show the overall flow of our temperature-aware voltage selection approach. Given is a task graph mapped and scheduled on a multicore SoC, and the average switched capacitance for each task, as discussed in Section 2. A so called "assumed" temperature, at which each core is supposed to run, is also fixed as input. The voltage selection algorithm (see Section 2.2 and [1]) will determine, for each task, the voltage modes (V_{dd} and V_{bs}) such that energy consumption is minimized. Based on the determined voltage modes (and the switched capacitances known for each task) the dynamic power profiles are calculated and the thermal analysis is performed as discussed in Section 3. Depending on what the designer selects, a unique temperature or a dynamic temperature profile is determined for each core in the steady state. This new temperature/temperature-profile is now used again for voltage selection and the process is repeated until the temperature/temperature-profile converges. Convergence means that the actual temperature values used at voltage selection correspond to the temperature at which the chip will function when running with the calculated voltages. Once convergence has been reached, based on the determined voltage modes and temperatures, the minimized energy consumption E_{ta} (temperature aware energy) is calculated.

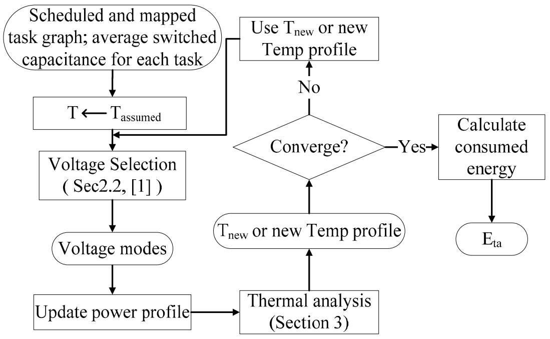
Scheduled and mapped task graph; average switched capacitance for each task
T ⟵ T$_{assumed}$
Voltage Selection (Sec2.2, [1])
Voltage modes
Update power profile
Use T$_{new}$ or new Temp profile
No
Converge?
Yes
T$_{new}$ or new Temp profile
Thermal analysis (Section 3)
Calculate consumed energy
E$_{ta}$

Fig. 3 Temperature-aware voltage selection

It is also important to notice that during thermal analysis potential thermal runaway is detected.

Fig. 4 shows a typical temperature convergence curve for the process described in Fig. 3. The circles indicate the temperature produced after each iteration (for this experiment static temperature analysis has been used and the curve illustrates the convergence for one core). As a basic technique, this new temperature (in the case of dynamic analysis, this new temperature profile) is used as input to the voltage selection in the next iteration. The dots represent successive temperatures in the inner iteration loop for temperature analysis (Fig. 1). As convergence criterion, a temperature difference of 0.2° has been used. Based on our experiments (Section 5) up to 90% of the cases reach convergence after less than five iterations (both for static and dynamic temperature analysis).

However, there are situations in which the temperature oscillates and the simple temperature updating technique described above leads to an infinite loop. This has happened in 2.5% of our experiments (Section 5). Such a situation is illustrated in Fig. 5. Oscillations are detected and are solved by changing the temperature update rule: Instead of using the just produced temperature for the next iteration, a middle value between the new temperature and the one produced in the previous iteration is used (in the case of dynamic temperature analysis, the points on the temperature profile are recalculated accordingly). By using this technique, all infinite loops occurring in our experiments have been solved.

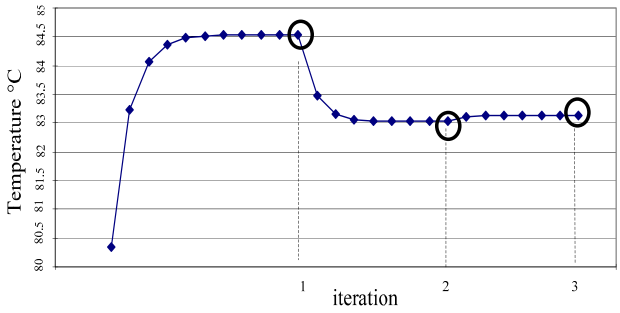

Fig. 4 Typical temperature convergence curve

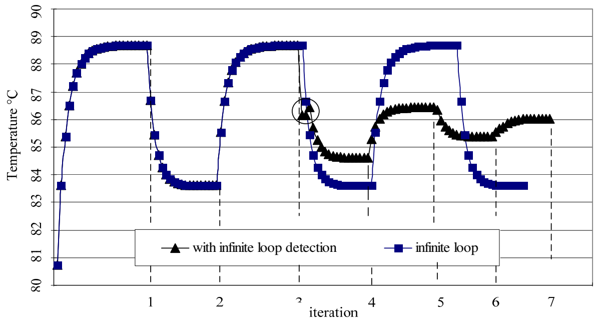

Fig. 5 Temperature convergence with infinite loop detection

5. Experimental Results and Discussion

Experimental results presented in this section are aimed at exploring the efficiency of temperature-aware voltage selection, compared to a voltage selection technique which ignores the temperature issue. For our experiments we have randomly generated applications consisting of 16 to 100 tasks. The size of each task is between 10^6 and $9{\cdot}10^6$ cycles, randomly distributed. The task graphs are mapped on SoC architectures consisting of 2 to 9 cores. Furthermore, in Section 5.4 we will present experimental results based on two real-life examples.

We have run experiments both considering only dynamic supply voltage selection and combined supply voltage and body bias selection. In the first case, 10 voltage levels were considered for V_{dd}, in the interval [0.6V, 1.8V]; for the second case 10 voltage modes (V_{dd}, V_{bs}) are used with V_{dd} and V_{bs} in the interval [0.6V, 1.8V] and [-0.1V, -1V], respectively.

The temperature model related coefficients for the SoC are given in Table 1. The parameters for leakage and frequency calculation (Equ. (1), (2), (3)) are the same as in [13] and [14].

Given a certain application and architecture, we run the temperature aware voltage selection algorithm illustrated in Fig. 3 and obtain the optimized energy consumption E_{ta}. For the same application and setting we run the voltage selection algorithm ignoring temperature, resulting in energy consumption E_{nta}. The temperature unaware voltage selection is realized by running one single iteration of the process in Fig. 3. The assumed temperature is used for voltage selection which produces the voltage modes; temperature analysis gives the real temperature at which the chip will run using those voltages and, finally, we calculate the consumed energy E_{nta}. By comparing E_{ta} with E_{nta} we can appreciate the efficiency of using a temperature aware voltage selection scheme.

Table 1 Temperature model settings

Chip thickness	0.00025m
Chip size	0.001m*0.001m~0.009m*0.009m per processor
Ambient temperature	313.15K
Convection capacitance	140.4J/K
Convection resistance	0.1~0.6 K/W
Heat sink area	0.02m* 0.02m~0.03m* 0.03m
Heat sink thickness	0.005m~ 0.008m
Heat spreader area	0.01m* 0.01m~0.02m* 0.02m
Heat spreader thickness	0.001m~0.002m

5.1. Optimization Time

To start with, we have investigated the execution time needed for the proposed temperature aware voltage selection approach. This time is composed of the time needed for thermal analysis and the one consumed for actual voltage selection (see Fig. 3).

The diagram in Fig. 6 shows the execution time for temperature aware voltage selection with static temperature analysis. In this case, since static analysis is very fast, the total optimization time is dominated by the actual voltage selection algorithm. The primary parameter in this case, as can be seen in Fig. 6, is the number of tasks (for this experiment we considered applications consisting of up to 800 tasks), while the number of processors is less significant.

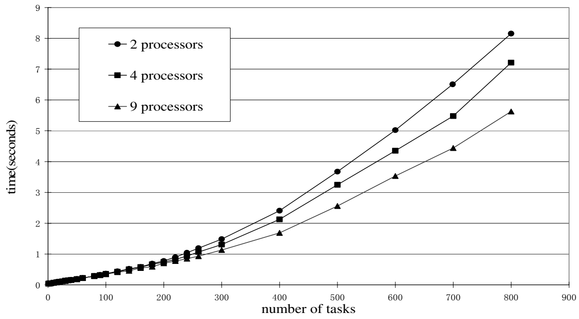

Fig. 6 Optimization time with static analysis

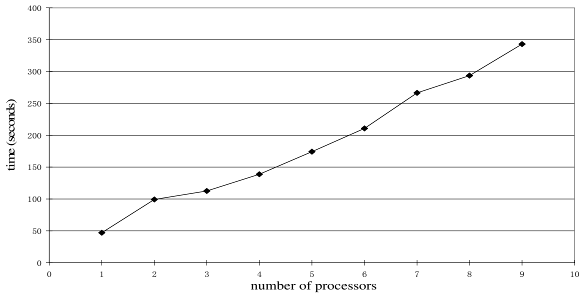

Fig. 7 Optimization time with dynamic analysis

The situation is different with dynamic temperature analysis, which is significantly more time consuming than the static one. In this case, the energy optimization time is dominated by the temperature analysis which is sensible to the number of analyzed blocks (in our case, the processors) but is unaware of the number of tasks. The diagram in Fig. 7 illustrates the optimization time as a function of the number of cores. Obviously, the optimization time with dynamic analysis is considerably larger than the one needed with the static one.

5.2. Static vs. Dynamic Temperature Analysis

As shown before, voltage selection based on dynamic temperature analysis is more time consuming than the alternative using static analysis. However, dynamic analysis is more accurate and, thus, potentially can lead to huger energy savings.

We have applied both the static and the dynamic temperature analysis – based voltage selection to a set of applications and have compared the energy consumption produced by the two methods. Two categories of applications were investigated: 100 applications were such that the temperature oscillation on the cores, in steady state, was less than 5°; 250 applications produced temperature oscillations larger than 5°.

Table 2 shows the results for the first category of applications. For 69% of the cases there was no difference between the static and dynamic analysis – based methods and there was no single case in which the energy saving due to the dynamic method was larger than 1%. Table 3 gives the results produced by the second category, for which the gains with dynamic analysis are expected to be larger. However, even in this case, only 2.8% of the applications produced an energy saving larger than 1% with an average gain of 0.12%.

Table 2 Static vs. dynamic analysis(small temperature oscillation)

Average improvement	Largest improvement	more than 1% difference	No improvement
0.01%	0.40%	0	69%

Table 3 Static vs. dynamic analysis(high temperature oscillation)

Average improvement	Largest improvement	more than 1% difference	No improvement
0.12%	5.11%	2.80%	30.1%

The obvious conclusion that can be drawn from the above experiments is that static thermal analysis is sufficiently accurate for the purposes of thermal aware voltage selection. This is the alternative we have used in the following experiments.

5.3. Temperature-Aware vs. Unaware DVS

In this section we compare, in terms of energy efficiency, the temperature aware DVS approach with the approach ignoring temperature. Given a certain application, we define the *energy efficiency factor of the temperature aware DVS*, compared to the unaware one, as $G = (E_{nta}-E_{ta})/E_{nta}*100\%$. It is expected that the energy E_{ta} produced by the temperature aware approach is smaller than E_{nta}. Obviously, E_{nta} depends on the *assumed temperature* provided by the designer. If the designer's guess is correct (equal to the temperature at which the chip functions with the selected voltages), a situation which is very unlikely, then $E_{nta} = E_{ta}$. As further away the designer's guess is, as larger E_{nta} is compared to E_{ta}.

We have used 150 applications for these experiments. They are running at temperatures in the range 40°C to 100°C. It is assumed that the circuit cannot work above 140°C and, thus, the possible range of the designer's guess is between ambient temperature (35°C) and 140°C. The diagrams in Fig. 8 show the average value of the energy efficiency factor G as a function of how far the temperature guess is from the actual temperature at which the application runs. The experiments were run with combined V_{dd} and V_{bs} scaling, and they were performed considering two different cases for the dependency of leakage current on the temperature. For the first case we used the value $\gamma = -2223.7$ for the coefficient in equ. (2) (this is one typical value indicated in [13]). For the second case we considered

$\gamma = -3223.7$, which indicates a higher degree of dependency of the leakage current on temperature (For all other experiments in the paper we considered $\gamma = -2223.7$). Fig. 9 shows the same experiments in the context of V_{dd} only scaling[1].

As can be seen, important energy savings can be achieved by thermal aware DVS. In the context in which it is practically impossible to predict at which temperature the circuit will function, since the actual voltages are not known before voltage selection, a thermal aware approach is a safe solution if energy losses are to be avoided. This is the case both for V_{dd} only and for combined V_{dd} and V_{bs} scaling. It is interesting to observe that, in the case of V_{dd} only, when temperatures are underestimated, the energy losses are smaller. The explanation is the following: When temperatures are overestimated, the temperature unaware approach assumes that leakage currents are very high (due to the high assumed temperature). Thus, the voltage selection algorithm will tend to select high supply voltages so that tasks are terminated early and slack time is used to put the circuit into low leakage modes. Since, in reality, the circuit will work at lower temperature and leakage currents will be considerably smaller (due to the exponential dependency of leakage on temperature, which at high temperature values leads to larger errors than at low temperatures), the temperature aware approach will produce smaller supply voltages, which explains the energy differences at overestimated temperature. In the case of temperature underestimations, the V_{dd} only approach will produce lower voltages (which extend the execution time in the limits of available slack) and, by this, find solutions that are close to those produced by the temperature aware approach. This is confirmed by following the voltages produced for our experiments.

In the case of the combined approach, however, which, in addition to the V_{dd} only technique, has the opportunity to control leakage by adapting the body bias voltage V_{bs}, the temperature aware approach makes a considerable difference, both in the case of temperature under - and overestimations.

The diagrams in Fig. 8 and Fig. 9 also indicate that, as the dependency of leakage on temperature grows, the difference made by the temperature aware scaling technique becomes more significant.

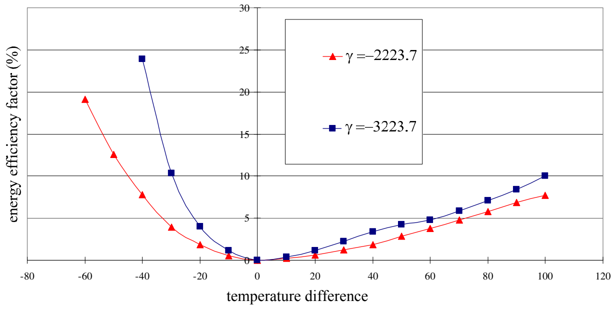

Fig. 8 Energy efficiency factor with combined V_{bs} and V_{dd} scaling

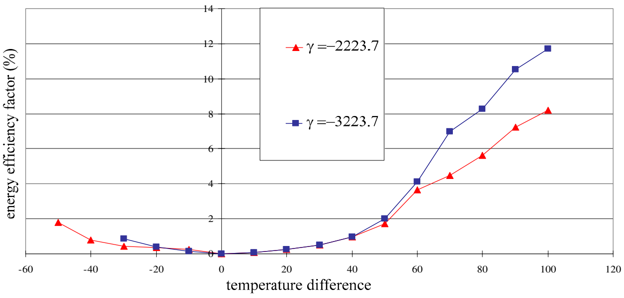

Fig. 9 Energy efficiency factor with V_{dd} only scaling

For the above experiments, the amount of leakage power (calculated at 70°C) was, on average, 50% of the total power. In a next set of experiments we have investigated the dependency of the factor *G* on the amount of leakage consumed by the circuit. We have performed our experiments with three different leakage percentage levels. The dependency is illustrated in Fig. 10 (V_{dd} only) and, Fig. 11 (combined V_{dd} and V_{bs} scaling). As expected, for a higher leakage percentage the temperature aware approach makes a larger difference. We have an interesting exception for combined V_{dd} and V_{bs} scaling in the case of temperature overestimations. The explanation is the following: In the case of high leakage percentage and if the assumed and real temperature are both high, both the temperature aware and the unaware scaling assume a very high leakage power and, thus, come close to producing that V_{dd} and V_{bs} combinations that forces down the leakage as much as possible and, by this, the produced voltage levels are becoming relatively similar. This similarity is as stronger as fewer execution modes are available on the processor.

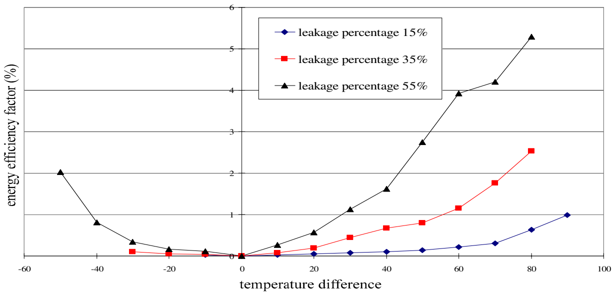

Fig. 10 Dependence on leakage percentage, V_{dd} only scaling

[1] In all experiments, the total amount of energy is significantly smaller with combined V_{dd} and V_{bs} scaling than with V_{dd} scaling only. However, not this is the issue that we investigate in this paper but, what we are interested in, is the impact of temperature awareness.

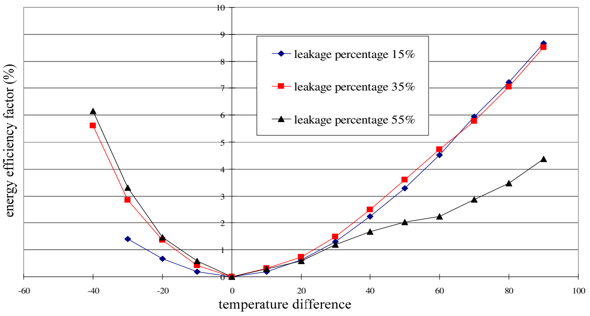

Fig. 11 Dependence on leakage percentage, V_{bs} and V_{dd} scaling

5.4. Real Life Examples

We have investigated the efficiency of temperature aware voltage selection using two real-life examples: A GSM voice codec in and a multimedia MPEG4 audio-video encoder. Details regarding the two applications can be found in [15] and [2], respectively. The GSM voice codec is composed of an encoder and a decoder of GSM frames and consists of 87 tasks considered to run on an architecture composed of 3 cores with 13 voltage modes. The MPEG4 consists of 109 tasks and is considered to run on 2 cores with 13 voltage modes.

The results are presented in Fig. 12 and they confirm the trends outlined by our previous experiments.

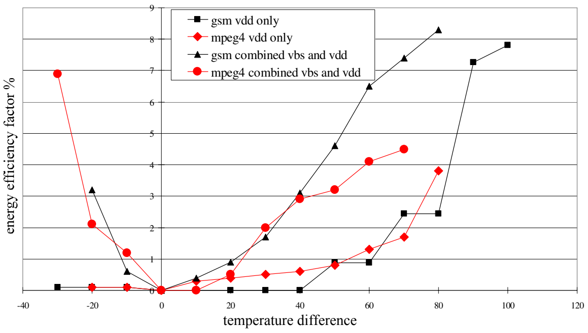

Fig. 12 Real life examples

6. Conclusions

We have presented an approach to thermal-aware voltage selection for energy minimization. The approach can be applied when only supply voltage selection is available as well as with combined supply voltage and body bias selection. We have shown that, besides having the potential to detect possible thermal runaway, a thermal aware approach can produce energy savings which can reach above 15%. The amount of energy savings depends on several factors, such as the percentage of leakage power, the dependency of leakage to temperature, and the availability of adaptive body biasing. We have also shown that using static thermal analysis inside the energy optimization loop produces results which, practically, are identical with those produced using dynamic analysis. Thus, the proposed heuristic is very efficient in terms of optimization time and can be applied to large applications both in terms of number tasks and processor cores.

References

[1]. A. Andrei, P. Eles, Z. Peng, M. Schmitz, B. M. Al-Hashimi, Energy Optimization of Multiprocessor Systems on Chip by Voltage Selection, IEEE Transactions on Very Large Scale Integration Systems, 15(3):262-275, 2007

[2]. http://ffmpeg.mplayerhq.hu/.

[3]. N. Bansal, T Kimbrel, Speed Scaling to Manage Energy and Temperature, Journal of the ACM, V54(1), Article 3, 2007.

[4]. A. P. Chandrakasan, R. W. Brodersen, Low Power Digital CMOS Design. Norwell, MA: Kluwer, 1995.

[5]. J. Choi, A. Bansal, M. Meterelliyoz, J. Murthy, K. Roy, Leakage power dependent temperature estimation to predict thermal runaway in FinFET circuits, ICCAD, pp. 583-586, 2006.

[6]. W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, M. Stan, HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design, IEEE onVLSI Systems, 14(5):501-513, 2006.

[7]. Tohru Ishihara, Hiroto Yasuura, Voltage Scheduling Problem for Dynamically Variable Voltage Processors, ISLPED, pp. 197-202, 1998.

[8]. C. Kim, K. Roy, Dynamic Vth Scaling Scheme for Active Leakage Power Reduction, DATE02, pp. 163-167, 2002

[9]. W. Kwon, T. Kim, Opt. Volt. Allocation Techniques for Dynamically Variable Voltage Processors, ACM TECS, V4, 1, '05, pp. 211-230.

[10]. J. Luo, N. Jha, Power-profile Driven Variable Voltage Scaling for Heterogeneous Distributed Real-time Embedded Systems, Int. Conf. on VLSI Design, 2003. pp 369-375.

[11]. W. P. Liao, L. He, and K. M. Lepak, "Temperature and supply voltage aware performance and power modeling at micro-architecture level," IEEE TonCAD, V24, no. 7, pp. 1042–1053, July 2005.

[12]. Y. Li, B. C. Lee, D. Brooks, Z. Hu, K. Skadron, CMP Design Space Exploration Subject to Physical Constraints, HPCA06, pp. 15-26, 2006.

[13]. Y. Liu, H. Yang, R.P. Dick, H. Wang, L. Shang, Thermal vs Energy Optimization for DVFS-enabled Processors in Embedded Systems, Int. Symp. on Quality Electronic Design (ISQED07), pp. 204 - 209, 2007.

[14]. S. Martin, K. Flautner, T. Mudge, D. Blaauw, Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Lower Power Microprocessors under Dynamic Workloads, ICCAD, pp. 721-725, 2002.

[15]. M. Schmitz, B. Al-Hashimi, P. Eles, System-Level Design Techniques for Energy-Efficient Embedded Systems, Kluwer Ac. Publ., 2004.

[16]. H. Su, F. Liu, A. Acar, S. Nassif, Full Chip Leakage Estimation Considering Power Supply and Temperature Variations, ISLPED, 2003.

[17]. K. Sankaranarayanan, S. Velusamy, M.R. Stan, K. Skadron, A Case for Thermal-Aware Floorplanning at the Microarchitectural Level, The Journal of Instruction-Level Parallelism, V7, Oct. 2005, pp. 1-16.

[18]. Y. Tsai, A. Ankadi, N. Vijaykrishnan, ChipPower : An Architecture-Level Leakage Simulator, Int. SOC Conference, 2004.

[19]. S. Wang, R Bettati, Delay Analysis in Temp.-Constrained Hard Real-Time Systems with General Task Arrivals, RTSS06, pp. 323-334.

[20]. Yuan Xie, Wei-Lun Hung, Temperature-Aware Task Allocation and Scheduling for Embedded Multiprocessor Systems-on-Chip Design, Journal of VLSI Signal Processing, 45(3), pp. 177-189, 2006.

[21]. F. Yao, A. Demers, S. Shenker, A Scheduling Model for Reduced CPU Energy, Symp. on Found. of Comp. Sience, pp. 374-382, 1995.

[22]. Y. Yang, Z. Gu, C. Zhu, R. Dick, L. Shang, ISAC: Integrated Space-and-Time-Adaptive Chip-Package Thermal Analysis, IEEE TonCAD, 26(1), pp. 86-99, 2007.

[23]. L. Yan, J. Luo, N. Jha, Joint Dynamic Voltage Scaling and Adaptive Body Biasing for Heterogeneous Distributed Real-time Embedded Systems, IEEE TonCAD, 24(7), pp. 1030-1041, 2005.