ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 3, Special Issue 3, April 2014*

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,

Paiyanoor-603 104, Tamil Nadu, India

Energy Efficient Framework for heterogeneous Multicore Parallel Processors

A.S.Radhamani, E.Baburaj

Research Scholar, Department of Computer Science and Engineering, Manonmanium Sundaranar University,Tirunelveli, India

Professor, Department of Computer Science and Engineering, Sun College of Engineering and Technology, Nagercoil, India

Abstract: Recently, energy-efficient computing has become a major interest, both in the mobile and IT sectors. With the advent of multi-core processors and their energy-saving mechanisms, there is a necessity to model their power consumption. The existing models for multi-core processors are based on the assumption that the power consumption of multiple cores performing parallel computations is equal to the sum of the power of each of those active cores. In this paper, we analyze this assumption and show that it leads to lack of accuracy when applied to modern processors such as quad-core. Based on our analysis, we present a methodology for estimating the energy consumption of multi-core processors. Unlike existing models, we take into account resource sharing and power saving mechanisms. We show that our approach provides accuracy with varying task sets.

I.INTRODUCTION

Industry has successfully continued to innovate and increase performance. These performance gains can be accomplished in several ways including more sophisticated process technology, innovative architecture or micro-architecture. The architecture of a processor refers to the instruction set, registers, and data structures that are public to the programmer and are maintained and enhanced from one generation to the next. The micro-architecture of a processor refers to an implementation of processor's architecture in silicon, the micro-architecture typically changes from one processor generation to the next, while implementing the same public processor architecture.

Lately, the power consumption of processors has become a key concern for energy-efficient computing systems. It was shown in [11, 18, 19] that processors contribute between 23-40% to the total server's power draw. Furthermore, the power drained by a processor mostly depends on its energy aware mechanisms (e.g. Intel SpeedStep) and load. In 2005, Barroso et al. [5] analyzed Google servers during peak utilization and showed that processors consumed about 57% of the total server's power consumption. However, this percentage in 2007 dropped to 43% thanks to the emergence of energy-aware mechanisms. This variation, which is highly related to the load as well as energy-aware features, demands a thorough understanding of the power consumption behavior in relation to these factors. Given the importance of the topic, several power consumption models for single- [8] and multi-core [7, 15] processors have been proposed. However, these models have three key limitations: i) they take into account only processors with at most two cores (e.g. dual-core processors), ii) the impact of energy-saving techniques such as Intel Speed- Step [20] and AMD Cool'n'Quiet [1] have not been considered, and iii) for a given same load on cores, it is assumed that the power consumption of each active core is identical due to their similar behavior [23]. Consequently, the overall power consumption of a multi-core processor is considered in the above mentioned models as the sum of power consumption of its constituent cores. However, when several cores are active (e.g. performing computations), they can share resources such as off-

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 3, Special Issue 3, April 2014*

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,

Paiyanoor-603 104, Tamil Nadu, India

chip cache. With such sharing, cores reduce their access to memory. Hence, cores accessing the memory require different power than the ones which do not. As a matter of fact, all the above-mentioned power estimation models suffer from an inaccuracy of up to 62% when I/O bound jobs are executed . In this paper, we circumvent the above-mentioned drawbacks by proposing a model that estimates the dynamic power consumption of multi-core processors. Due to the variable behavior of the different components of a processor, we decompose the modeling process into the following three component levels: i) processor's chip: these are power consumption is modeled using capacitance methods [10] based on the core's utilization. When several cores are active, inter-core (cores on the same die) and -die (cores on different dies) communications occur. In this regard, the power consumption of each communication is modelled. With the emergence of energy-saving mechanisms, the behavior of power dissipation is different than that without such mechanisms. In order to reflect this aspect, we provide a model that estimates the power consumption with and without energy-efficient mechanisms. die: these are components within a die (e.g. off-chip cache) and iii) core: these are components within a core (e.g. control unit and on-chip cache). For a single core, the power consumption is modeled using capacitance methods [10] based on the core's utilization. When several cores are active, inter-core (cores on the same die) and -die (cores on different dies) communications occur. In this regard, the power consumption of each communication is modeled.

II.Related Work

A variety of power consumption models both for single and multi-core processors have been proposed in the literature. For single-core processors, the power consumption is measured directly at hardware-level such as CPU cycles [8], circuit [16] and register-transfer-level (RTL) [9, 14]. The main advantage is that these models provide a high level of accuracy. However, monitoring the activities of a processor at low (transistor) level is complex since a processor has millions (billions) of transistors and monitoring each transistor is not trivial. To overcome this complexity, software-level models have been developed. The power dissipation of the underlying hardware (i.e. CPU) is predicted based on the power consumed by each instruction [22] or function [21] it executes. One key issue is that software-level models depend upon tracing tools that parse an application to determine all its constituent instructions or functions. In case tracing tools are unable to extract the complete information regarding instructions, software-level models suffer from inaccuracy in power estimation.

In order to prevent the dependency on tracing tools, models based on the performance monitoring counters (PMC) [17, 7] have been proposed. Basically, power dissipation during application execution is highly related to the amount of accesses to cache and switching activities within processors.

Such activities (events) have been monitored through embedded programmable event counters [6] to calculate the total power consumption of a processor.

The major disadvantage of the above-mentioned models is that they do not take into account modern energy-saving techniques. Furthermore, they don't differentiate the variable behavior of cores having parallel or stand-alone computations. To overcome these problems, our model takes into account the behavior of individual and multiple cores as well as energy-saving mechanisms.

III.EXISTING METHODOLOGY

The existing models [15, 7] for multi-core processors assume that the overall power of such processors is the sum of Figure 1: an abstract architecture of a multi-core processor and power of their constituent cores. Based on this assumption, the overall power of multi-core processors is given by:

$$P_n = \sum_{j=1}^{n} Pc(j) \quad \text{_____} \quad (1)$$

where Pn denotes the power consumption of n cores and Pc(j) represents the power dissipation of a core j. The key concern is that such models assume that the power consumption behavior of a core remains identical regardless it performs computations either (1) alone and the others stay idle or (2) in parallel with other cores. This assumption is considered due to the similar behavior of cores [23], which is not always adequate. One major counterexample is the sharing of resources. For instance, when several cores share off-chip cache and one core fetches data from the main

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 3, Special Issue 3, April 2014*

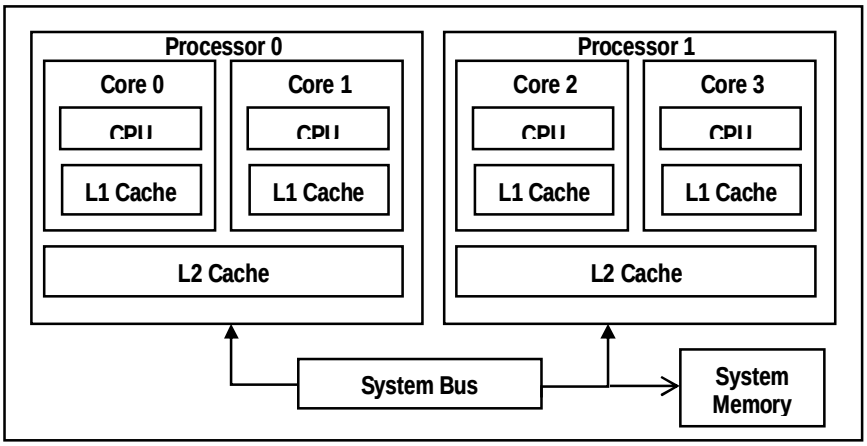
International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,

Paiyanoor-603 104, Tamil Nadu, India

memory (RAM), the others may not need to further access the memory, if the required data has already been extracted. Instead, they can fetch data directly from the cache.



All the components that lie within core level rectangle are limited (exclusive) to a specific core, and cannot be shared with other cores. Components outside the core-level rectangle are the non-exclusive ones, which can be shared between cores and dies. Some shared components are mandatory, which can be at chip-level (e.g. on-chip voltage regulator) and at die-level. On the other hand, some shared components (e.g. off-chip cache) can be optional. With these aspects, the most relevant components of a multi-core processor can be classified into three generic categories: i) mandatory components (chip- and die-level), ii) exclusive components and iii) optional components. In this section, Equation (2) is evaluated from the perspective of above three components' categories.On the other hand, if each single core performs computation alone and the others remain idle, each such core has to access the memory. In other words, the frequency of accessing memory decreases due to sharing. Consequently, the power consumption of several cores becomes less than the sum of their individual powers as given in Equation (2).Figure 1 shows an abstract architecture of multi-core processors, which may consist of several dies, and each one can have several cores. Components within multi-core processors can be lied within chip-, die- and core-level as illustrated in Figure 1 in the form of enclosed rectangles.

IV.PROPOSED METHODOLOGY

4.1 SYSTEM MODEL

4.1.1 Task and Processor Model

We consider a set of n periodic real-time tasks $\varphi = \{\gamma_{1,}, \gamma_{2,\ldots.} \gamma_{n}\}$, that are partitioned upon m heterogeneous processing cores C1 . . .Cm. We use φi, to denote the subset of tasks allocated to core Ci. Each periodic task γi is characterized by a worst-case workload of wcci cycles and a period of Pi, assumed to be equal to the relative deadline of its jobs. We assume the Global DVS feature and the voltage can be adjusted for all active cores uniformly, along with the frequency (up to an upper bound fmax). The worst case execution time of task γi under frequency f, is given by wcci f . We use the symbol Wi to denote the worst-case execution time of task γi under maximum frequency; that is, Wi = wcci/ fmax . The base utilization of task γi (under maximum frequency) is Ui = Wi/Pi ≤ 1.0. Hence, the total utilization of the task set Ã is given by Utot = Pn ,i=1 Ui ≤ m. Finally, the load on core Ci is given by the

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 3, Special Issue 3, April 2014

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,

Paiyanoor-603 104, Tamil Nadu, India

total utilization of tasks allocated to Ci, On each core, the preemptive Earliest Deadline First (EDF) scheduling policy is adopted.

4.12. Power Model

Advanced Configuration and Power Interface (ACPI) is a unified and open power management standard introduced and endorsed by major hardware and software manufacturers such as Intel, Microsoft, HP and Toshiba. ACPI defines an active state in which the core executes instructions. The
exact power profile in active state (defined as state C0 in ACPI) will consist of static and dynamic power figures. In the active state, by using the power model from we model the power consumption of a core Ci executing task γi as:

$Pi(t) = Pstatic + ajV^2f + Pj\,ind$ ---------(2)

where ajV^2f and Pjind represent the frequency-dependent and frequency-independent components of active power, respectively. V denotes the supply voltage and f denotes the CPU clock frequency. aj is the effective switching capacitance of task γi . Note that the values of aj and Pj ind depend on the characteristics of the task γi executing on core Ci at a given time . Pstatic represents the static power. In Global DVS settings, all active cores are inherently constrained to operate at the same supply voltage and frequency level . Given the almost linear relationship between supply voltage and frequency, the power consumption of the active core Ci at time t is given as:

$Pi(t) = Pstatic + ajf^3 + Pj\,ind$ _____(3).

The aggregate power consumption of all the cores varies with time and is a function of individual core states and the global operating frequency of all active cores. Let H be the hyper period of the task set φ. The energy consumption of the voltage island over the interval [0,H] is given as:

$E= \sum_{i=1}^{m} Pi(t)dt$ _____(4)

When a core is not executing any instructions, it may be put in one of the various idle states [34]. Each idle state has a different power consumption characteristic; as a general rule, the lower power consumption in a given idle state, the higher the time and energy overheads involved in returning to the active state. While the exact number of idle states varies from architecture, in this work, we assume the existence of at least the following three fundamental states that are supported by most modern multicore systems:

• Halt state: In this state, the execution of instructions is halted and the core clocks are gated, resulting in significant reduction in dynamic power. The core can return to active state almost instantaneously (≈ 10ns) . We model the power consumption on core Ci in the halt state as
Pi = Pstatic + P0, where P0 is the reduced dynamic power.

•Sleep state: Here, further, the Phase Locked Loops (PLLs) are gated and L1 cache contents are invalidated. In this state, the dynamic power is practically eliminated thus making Pstatic the only component of power consumption. However, this saving in power consumption comes at the cost of addition overheads compared to the halt state. Returning to active state may require a few hundred microseconds and involves non-trivial energy overheads

• Off state: Here, the core voltage is reduced to very low levels, to make even the static power consumption negligible. CPU context is not preserved and returning to active state involves significant time and energy overheads . Intel's new i7 architecture achieves this very low energy consumption through power gating feature.

V.IMPLEMENTATION

5.1Energy-efficient Core Activation and Task Allocation:

In general, the number of available processing cores (m) may be greater than the minimum number of cores upon which the given real-time workload can be scheduled in feasible manner. While the early studies that exclusively focused on dynamic power using all processing elements in parallel whenever possible, ever increasing static power figures renders such an approach infeasible. The power consumption of a given core can be minimized (in fact, effectively

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 3, Special Issue 3, April 2014*

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,

Paiyanoor-603 104, Tamil Nadu, India

eliminated through techniques such as power gating in Intel i7 architecture when it is put to off state .In active, halt and sleep states, the static power would be consumed continuously. This is because the periodic nature of the real-time application and significant time/energy overheads associated with transitions to/from off state make dynamically putting a core to off state at runtime an unrealistic option. As a result, instead of activating a core with light workload (with corresponding static energy consumption), it would be preferable to move that workload to other cores when possible. Obviously, a correlated and major issue is to perform task allocation on the selected cores to preserve feasibility and prepare favorable initial conditions for run-time management of dynamic energy. Thus, the offline phase can be seen as an integrated component that decides on task-to-core allocations while keeping an eye on total (i.e. static+dynamic) potential energy consumption. The $k \leq$ m cores selected by this phase will
be activated and then will be managed by the run-time component. The remaining (m−k) cores are put to off state with negligible power consumption.

5.2 Run-time Power Management of Active Cores:

The run-time management of the selected $k \leq m$ cores involves the use of Global Voltage Scaling as well as selectively putting some cores to halt and sleep states to reduce dynamic energy. To start with, the global frequency level that determines the dynamic power consumption at time t is decided by the highest performance level required by any core in active state at time t (Equation (2). This requires both closely monitoring the workload conditions on all cores and exploiting the available idle states whenever possible. As an example, if the core that requires highest performance level (to guarantee the feasibility of its workload) is put to halt or sleep state temporarily, the frequency can be reduced to the next highest performance level required by any of the remaining active cores during that interval. In addition, putting any core to halt and in particular sleep states have the potential of reducing dynamic energy consumption for all the cores through reducing the global energy-efficient frequency

V. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of our algorithms through the help of a MATLAB simulator. For 2-64 core systems, we generated synthetic task sets each with 20 and 50 tasks, respectively. The effective switching capacitance ai of tasks was set to 1. Pi ind values were randomly chosen in the range [0, 0.2]. Task periods were generated randomly in the interval [63ms, 1300ms] which are comparable to those seen in practice. Figure 2 shows the energy consumption of cores for varying task sets.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

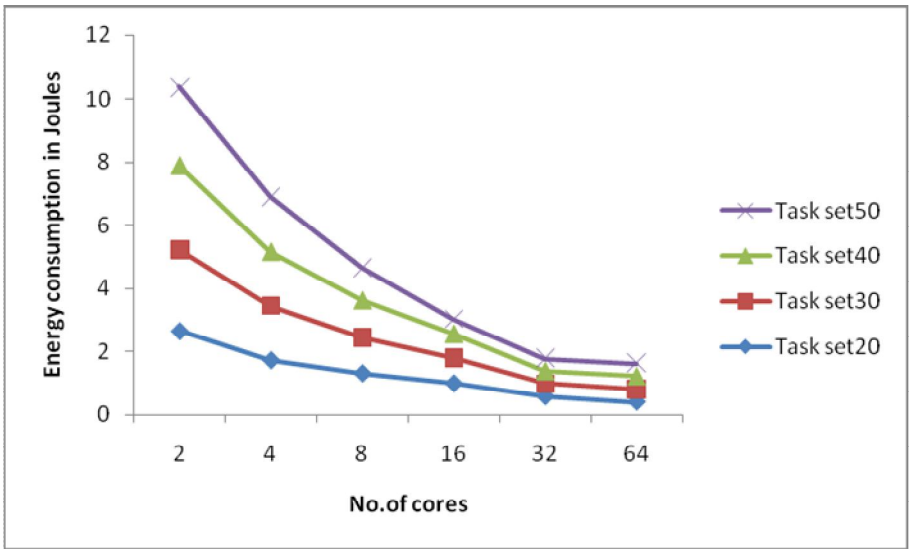
An ISO 3297: 2007 Certified Organization *Vol. 3, Special Issue 3, April 2014*

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,

Paiyanoor-603 104, Tamil Nadu, India



For a target total utilization value Utot, we generated individual task utilizations randomly in such a way that each task utilization is no greater than a pre-defined threshold ® ≤ 1.0. Previous studies dealing with energy minimization on multi-processor systems showed that the maximum task utilization (denoted as ®) is an important parameter for performance. As a result, we also investigated the impact of this task utilization factor ®. In the experiments, we refer to normalized utilization as the quantity Utotm , where m is the number of cores on which the workload is executed. For each normalized utilization and ® pair, we generated 1000 task sets; the data points in the plots reflect the average of these runs. The reported energy consumption values are normalized with respect to the base scheme that executes all tasks at fmax at all times (no power management).

REFERENCES

[1] http://www.amd.com/us/products/technologies/ cool-n-quiet/Pages/cool-n-quiet.aspx.
[2] http://www.zes.com/english/products/ one-to-eight-channel-precision-power-analyzer-lmg500.html.
[3] http://ark.intel.com/Product.aspx?id=33929.
[4] http://www.devin.com/lookbusy/.
[5] L. Barroso and U. Holzle. The case for energy-proportional computing. Computer, 40(12):33 –37, 2007.
[6] R. Berrendorf and B. Mohr. PCL - The Performance Counter Library Version 2.2, Jan. 2003.
[7] R. Bertran, M. Gonzalez, X. Martorell, N. Navarro, and E. Ayguade. Decomposable and responsive power models for multicore processors using performance counters. In Proceedings of 24th ACM Int'l Conf. On Supercomputing, ICS '10, pages 147–158. ACM, 2010.
[8] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In Proceedings of the 27th Int'l Symp. on Computer Architecture, pages 83 – 94, 2000.
[9] A. Chandrakasan and R. Brodersen. Minimizing power consumption in digital CMOS circuits. Proceedings of the IEEE, 83(4):498 –523, Apr. 1995.
[10] A. P. Chandrakasan and R. W. Brodersen. Minimizing power consumption in cmos circuits. Technical report,University of California at Berkeley, 1995.
[11] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In Proceedings of the 34th annual Int'l Symposium on Computer Architecture, pages 13–23. ACM, 2007.
[12] S. Herbert and D. Marculescu. Analysis of dynamic voltage/frequency scaling in chip-multiprocessors. In Proceedings of Int'l Symp. on Low Power Electronics and Design, pages 38 –43. ACM/ IEEE, 2007.
[13] C. Hewlett-Packard, C. Intel, C. Microsoft, L. Phoenix Technologies, and C. Toshiba. Advanced configuration and power interface specification, 2010.
[14] C.-T. Hsieh, Q. Wu, C.-S. Ding, and M. Pedram. Statistical sampling and regression analysis for RT-Level power evaluation. In Proceedings of Int'l Conf. on Computer-Aided Design, pages 583 – 588.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 3, Special Issue 3, April 2014*

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,

Paiyanoor-603 104, Tamil Nadu, India

[15] C.-H. Hsu, J. J. Chen, and S.-L. Tsao. Evaluation and modeling of power consumption of a heterogeneous dual-core processor. In Proceedings of Int'l Conf. On Parallel and Distributed Systems, pages 1–8, 2007.
[16] C. X. Huang, B. Zhang, A.-C. Deng, and B. Swirski. The design and implementation of PowerMill. In Proceedings of the Int'l Symp. on Low Power Design, pages 105–110. ACM, 1995.
[17] R. Joseph and M. Martonosi. Run-time power estimation in high performance microprocessors, 2001.
[18] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. Keller. Energy management for commercial servers. Computer, 36(12):39 – 48, 2003.
[19] D. Meisner, B. T. Gold, and T. F. Wenisch. PowerNap: eliminating server idle power. In Proceeding of the 14th Int'l Conf. on Architectural Support for Programming Languages and Operating Systems, pages 205–216. ACM, 2009.
[20] V. Pallipadi. Enhanced Intel SpeedStep Technology and Demand-Based Switching on Linux, Feb 2009.
[21] G. Qu, N. Kawabe, K. Usarni, and M. Potkonjak. Function-level power estimation methodology for microprocessors. In Proceedings of Design Automation Conference, pages 810–813, 2000.
[22] J. Russell and M. Jacome. Software power estimation and optimization for high performance, 32-bit embedded processors. In Proceedings of Int'l Conf. On Computer Design, pages 328 –333, 1998.
[23] K. Singh, M. Bhadauria, and S. A. McKee. Real time power estimation and thread scheduling via performance counters. SIGARCH Comput. Archit.News, 37:46–55, July 2009.