

Realization of Embedded Automotive System Based On Dual Core Processor

G. Murugeswari Kavitha*, C. Rajasekaran**

*Student of K.S.R college technology, Tamilnadu, India

**Faculty of Engineering, K.S.R college technology, Tamilnadu, India

Summary

The power Consumption of modern high-performance processors is becoming a major concern because it leads to increased heat dissipation and decreased reliability. A heterogeneous multiprocessor (HeMP) system consists of several heterogeneous processors each would have the capability of producing better energy saving performance. The problem of minimizing the energy consumption without missing deadlines has become an important issue in constructing low-power real-time HeMP systems. Each processor may have its own instruction-set architecture, specially designed to provide the best performance for a particular category of applications. Here pruning-based algorithm is used to obtain the optimal solution and a heuristic algorithm is also proposed to derive an approximate solution. Each processor's speed is determined by its final workload. In simulations, we model more than a couple dozens of off-the-shelf embedded processors including ARM processor and TI DSP. The results show that the pruning-based algorithm reduces the time needed to derive the optimal solution by at least 98%, compared with the exhaustive search.

Key words:

Energy-aware systems, Heterogeneous multiprocessor (HeMPs), Open Multimedia Application Platform (OMAP).

1. Introduction

The HeMP architecture is commonly adopted by real-time embedded systems, in which each task must complete before its deadline. Examples are embedded control [1]–[3] and multimedia systems [4], [5]. Energy consumption is one of the critical factors in designing battery-operated systems, such as portable personal computing and communication devices.

To reduce system energy consumption, supply voltage reduction is the most powerful technique since power is a quadratic function of the voltage. Recent advances in power supply circuits have enabled systems to operate under dynamically varying supply voltages. In such an environment, the speed of the system can be dynamically controlled. In recent years, processor performance has increased at the expense of drastically increased power consumption. Thus heat dissipation has become a major problem because it requires more expensive packaging and cooling technology and decreases reliability especially for multi-processor systems. In order to reduce

heat dissipation and to increase reliability, many hardware and software techniques have been proposed to lower processor power consumption. Processors running on multiple supply voltages (i.e., multiple power levels) have become available in recent years making power management at the processor level possible. Using this feature, several software techniques have been proposed to adjust the supply voltage, especially in the area of mobile computing where devices are battery operated and have a restricted power budget.

1.1 Models and Power Management

Our HeMP system consists of m heterogeneous processors. They are c_1, c_2, \dots, c_m . Adopt a commonly used energy model in which the power consumption of c_j at speed y_j , denoted by $P_j(y_j)$, is determined by

$$P_j(y_j) = k_j y_j^3 + q_j \quad (1)$$

Goal is to develop a feasible schedule that minimizes the total energy consumption of this system. The task model includes a set of periodic real-time tasks

$$p = \min \sum_{j=1}^m E_j(y_j) \quad (2)$$

Let T be the set of n periodic tasks, $\{\tau_1, \tau_2, \dots, \tau_n\}$. Each periodic task τ_i is a sequence of jobs released at constant intervals called periods.

Moreover, the job of each task must be completed before the next job of the task is released. All tasks are independent and preemptible.

2. Related Works

2.1 .Dynamic Reclaiming Algorithm

Though the static scheme can be shown to be optimal under a worst-case workload, it is known that, in many cases, the instances of real-time tasks complete earlier than under the worst-case scenario [4]. A trivial remedy would be to shutdown the processor when there are no ready tasks. However, this technique is clearly suboptimal

because of the convexity of the power consumption function: It is always more energy-efficient to transfer unused CPU time to other tasks by reducing their speeds whenever possible. The dynamic reclaiming algorithm we present in this section is based on detecting early completions and adjusting (reducing) the speed of other tasks on-the-fly in order to provide additional power savings while still meeting the deadlines. To this aim, we perform comparisons between the actual execution history and the canonical schedule *Scan*, which is the static optimal schedule in which every instance presents its worst-case workload to the processor and runs at the constant speed *S*. The CPU speed is adjusted only at task dispatch times: Thus, we should be able to say whether the task is being dispatched earlier than under *Scan* and, if so, determine the amount of additional CPU time the dispatched task can safely use to slow down its execution; we will refer to this additional CPU time as the earliness of the dispatched task.

2.2 Overview of PBA

Initially, the current best task partition is empty, and its energy consumption is infinite. PBA first does a variable-based pruning to remove some binary variables. It next traverses the pruned tree from the root node v_0 . At each node v_i , PBA estimates the energy lower bound of each v_i 's children and first visits the node with the lowest lower bound. Two pruning rules are used to speed up the tree traverse. One is energy-based pruning, which prunes the branch whose energy lower bound is higher than the current best solution. The other is speed-based pruning, which removes the branch when a processor's speed exceeds its maximum speed. When reaching a leaf node, a local optimization algorithm, named Max Reduction, is adopted to find the best task partition x in the neighborhood of the obtained task partition x .

workload of processor can be calculated by using this formula

$$w_\phi = \sum_{j=1}^m r_j + \sum_{\tau_i \in \theta_\phi} l_i \quad (3)$$

Table 1: Utilization Factor Of Processor

Task Number	6	8	10	12	14	16
2 processors	3	3	3	3	3	3
4 processors	5	5	5	6	6	5
6 processors	7	8	9	8	9	11
8 processors	10	11	12	14	13	12

There exist task sets with a total utilization slightly greater than and arbitrarily close to 1 that are unschedulable on a system with m processor cores

2.3 Optimal Voltage Schedule

The basic idea of this approach is to transform the complicated problem of determining the optimal voltage schedule to an easier problem: finding the consumption among the optimal voltage schedules for a number of primary job sets.

2.4 Scheduling Algorithm

A number of real-time scheduling algorithms have been proposed for a HoMP system [2, 4]. The Proportionatefair (Pfair) algorithm, provides an optimal real-time schedule for periodic tasks. This algorithm, however, considers no energy consumption and cannot be used in a low-power system. It proposed a method for finding the optimal number of processors on which a given set of periodic tasks the minimum energy consumption. Chen et al. [4] optimally bounds the energy consumption for a set of frame-based tasks, each of which has different power characteristics. All these algorithms focused their discussion on HoMP systems. Without considering that a task may have different execution times on heterogeneous processors, these algorithms cannot be directly applied on HeMP systems.

3. Experimental Results

In this section, we compare the performance of our work with the proposed work in [12] and [13]. We conducted a series of simulations to demonstrate the effectiveness of our algorithms in delivering the optimal energy-saving performance. There are nearly 30 processors modelled in our simulations. These processors include general-purpose embedded processors, such as ARM9, ARMI10, and ARMI11, and DSP processors, such as TMS320C and TMS320D. The adjusted switched capacitance of each processor is obtained from the official website of ARM and TI and is summarized in Table I. We evaluate our algorithms on a simulated HeMP system consisting of 2, 4, 6, and 8 processors, each of which is randomly from the list of modelled processors.

We vary the workload by changing the number of tasks and each task has an execution cycle count between 1,000 and 3,000. For each configuration, we ran simulations for 30 times and took the average value for comparison. We use kX3 to denote the kX3-Partition algorithm. We use List to denote a commonly-used HoMP low-power scheduling algorithm that dispatches a task to a least loaded processor [1, 5].

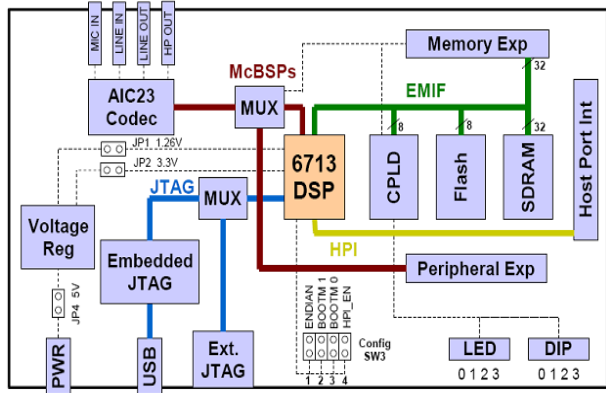


Fig.1.TMS320C613 DSK Block Diagram

3.1. DSK Hardware

The above figure shows the block diagram of the TMS320C613 DSK hardware. The heart of the DSK is the TMS320C613 DSP chip which runs at 225 MHz. The DSP is in the center of the block diagram and connects to external memory through the EMIF interface. There are several devices connected to this interface. One device is a 16 Mbyte SDRAM chip. This memory, along with the internal DSP memory, will be where code and data are stored. On the DSK board there is a TLV320AIC23 (AIC23) 16-bit stereo audio CODEC (coder/decoder).

The chip has a mono microphone input, stereo line input, stereo line output and stereo headphone output. These outputs are accessible on the DSK board. The AIC23 figure 1 shows a simplified block diagram of the AIC23 and its interfaces. The CODEC interfaces to the DSP through its McBSP serial interface. The CODEC is a 16-bit device and will be set up to deliver 16-bit signed 2's complement samples packed into a 32-bit word. Each 32-bit word will contain a sample from the left and right channel in that order. The data range is from - to (-1) or -32768 to 32767.

3.2. Simulation setup

There are various versions of CCStudio in current circulation, each has difference between the other versions of CCStudio. These differences affect what processors are available to debug, how code is to be run, and how projects are compiled. Depending on your version of CCStudio there may be additional steps you will have to take to ensure your own projects will run correctly. The latest version (CCStudio v3.1 Platinum) merges support for all DSP platforms (C6000, C5000, C2000, and OMAP) into one application.

Table 2: Versions of Ccstudio

CCStudio Version	DSP Family Supports			
V3.1 Platinum	C6000	C5000	C2000	OMAP
V3.0 for c6000	C6000			
V3.0 for OMAP		C5000		OMAP
V3.0 for c2000			C2000	
V2.21 for c6000	C6000			
V2.21 for OMAP		C5000		OMAP
V2.21 for c2000			C2000	

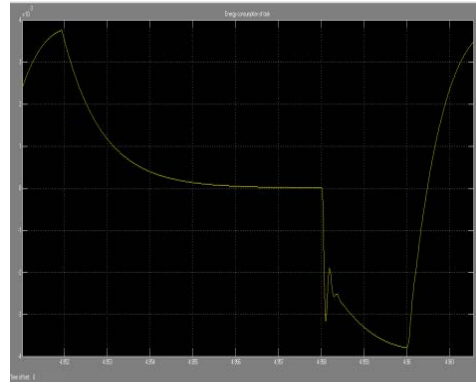


Fig.2. Simulation result of c2000 processor

3.3. DSP PLATFORMS

The most obvious change in CCStudio, is that the latest version supports all the DSP platforms, while earlier versions of CCStudio were developed to support only one DSP platform. Table II is a list of the previous CCStudio versions and what each supported.

The main function will be executed and the program will leave the main function. After the main function executes the TSK0 object is schedule and the tsk0 function is executed. Both functions print to the trace LOG object. The output can be viewed in the Printf Logs window. The profile statistics of C2000 processor is shown in figure 4.

4. CONCLUSION

In this paper, we explored the scheduling of real-time tasks on a heterogeneous platform with energy minimization as a goal. Our heuristic offers a high success rate and significantly improves the state of the art heuristics, especially for small task sets. Our algorithm is stable in its results when exploring different task sets and platforms of different heterogeneities.

Our experiments also evaluate the importance of the order in which the tasks are selected for scheduling. Furthermore, we have shown how to improve the results by combining two heuristics and how to improve the success rate by using the sensitivity of the scheduler to the tightness of the constraint. Our scheduling strategy relies on a kernel composed of independent tasks and use

software-pipelining to build this kernel of independent tasks. However, this can create a problem because it increases the pressure on cache and memory traffic. because now several items are in-flight at the same time. For hard real-time systems missed deadlines are unacceptable. In this case, the worst-case computation time can be used, resulting in a schedule that meets timing constraints but is not always energy-optimal.

For the voltage setup problem, existing work focused on a one-processor system and has provided a couple of solutions for both the single-level and the multi-level problems. Our work is the first one that addresses the multi-processor voltage setup problem. We started with the HeMP single level problem in this paper. Currently, we are extending our discussion to solve the HeMP multi-level problem.

Location	Count	Average	Total	Maximum	Minimum
Volume.c line 92	0	0.0	0	0	0
Volume.c line 89	0	0.0	0	0	0
Volume.c line 91	0	0.0	0	0	0

Fig.3.Profile Statistics of c2000 processor

time	seqID	formattedMsg	logger
0	0	In main function	trace
1	1	In TSK0	trace

Fig.4.Printf Logs output window

Acknowledgement

The authors would like to thank all of the anonymous reviewers for their suggestions

References

[1] Edward T.-H. Chu, Tai-Yi Huang, and Yu-Che Tsai “An Optimal Solution for the Heterogeneous Multiprocessor Single-Level Voltage-Setup Problem” IEEE transaction on

computer-aided design of integrated circuits and systems, Vol. 28, No. 11, November 2009.

[2] C. Li, Y. Jiang, Z. Wu, and T. Watanabe, “A multiprocessor system for a small size soccer robot control system,” in Proc. 4th IEEE Int. Symp. DELTA, 2008, pp. 115–118.

[3] S. Szabo, V. Singule, V. Oplustil, and R. Kral, “Autonomous mobile robot with multiprocessor control system,” in Proc. 7th Int. Workshop Adv. Motion Control, 2002, pp. 467–471.

[4] Gozde Bozdagi Akar “Applied Study Based On OMAP Digital Fingerprint Encryption Method” Proceedings of the IEEE 2010

[5] Mursel Yildiz, “User Directed View Synthesis On Omap Processors”, Proceedings of the IEEE,2010

[6] Jing Liang Yinqin Wu “Wireless ECG Monitoring System Based on OMAP” International Conference on Computational Science and Engineering,2009

[7] Yu-Lin Wang, Sung-Yen Chang, Shih-Chieh Chen, Yueh-Min Huang”Implementation and Optimization of Multimedia” IEEE Globecom Workshop on Multimedia Communications,2010

[8] Gozde Bozdagi Akar “Applied Study Based On OMAP Digital Fingerprint Encryption Method” Proceedings of the IEEE 2010

[9] Mursel Yildiz, “User Directed View Synthesis On Omap Processors”, Proceedings of the IEEE,2010

[10] James Song, Thomas Shepherd, Minh Chau, Ayesha Huq, Ikram Syed, Somdipta Roy, Achuta Thippana, Kaijian Shi+, Uming KO. “A Low Power Open Multimedia Application Platform” Proceedings of the IEEE,2009

[11] Jun Yu, Peihuang Lou, Xing Wu”A Dual-core Real-time Embedded System for Vision-based Automated Guided Vehicle” IITA International Conference on Control, Automation and Systems Engineering,2009

[12] Edward T.-H. Chu, Tai-Yi Huang, and Yu-Che Tsai An “Optimal Solution for the Heterogeneous Multiprocessor Single-Level Voltage-Setup Problem”, IEEE transactions on computer-aided design of integrated circuits and systems, vol. 28, no. 11, november 2009

[13] Kun-Yuan Hsieh, Yen-Chih Liu, Po-Wen Wu, Shou-Wei Chang, Jenq Kuen Lee “Enabling Streaming Remoting on Embedded Dual-core Processors” 37th International Conference on Parallel Processing 2008 IEEE

[14] K. Mankodiya1, S. Vogt1, A. Kundu1,4, M. Klostermann1, J. Pohl1, A. Ayoub2, H. Gehring3, U.G.Hofmann1”Portable Electrophysiologic Monitoring Based On The Omap-Family Processor From A Beginner Prospective”2009,IEEE

[15] Dinesh.P.M1, Dr.R.S.Sabeenian”Multi Format Scalable Media Decoder Implementation Using OMAP3530”,2010 IEEE

[16] Pramod Poudel and Mukul Shirvaikar.”Optimization of Computer Vision Algorithms for Real Time Platforms” 42nd South Eastern Symposium on System Theory 2010,IEEE

[17] Atanas Gotchev', Antti Tikanmaki', Atanas Boev 1, Karen Egiazarian' Ivan Pushkarov', Nikolai Daskalov2”Mobile 3dtv Technology Demonstrator Based On Omap 3430”2009,IEEE

- [18] Michael G. Morrow Cameron H. G. Wright Thad B. Welch”
Winds8: The Continuing Saga Of Winds8” 2011,IEEE
- [19] Devang P. Soni “Autonomous Arecanut Tree Climbing
And Pruning Robot”2010,IEEE
- [20] Wolfgang Schriebl, Thomas Winkler, “A Pervasive Smart
Camera Network Architecture applied for Multi-Camera
Object Classification”2009,IEEE



Murugeswari (ME AE), K.S.R. College of Technology, Tiruchengode. This is the project work done for the fulfillment of ME degree under anna university, Coimbatore during the period 2011-2012.



C.Rajasekaran Associate professor, ECE Department, K.S.R. College of Technology, Tiruchengode. This project work is supervised for the fulfillment of ME degree under anna university by murugeswari@kavitha, Coimbatore during the period 2011-2012.