

Enhancing Energy Efficiency in Temperature Controlled Dynamic Scheduling Technique for Multi Processing System on Chip

Hamayun Khan¹, Muhammad Yousaf Ali khan¹, Qaiser Bashir², Muhammad Usman Hashmi³, irfan Ud Din³, Shahid khan⁴

Abstract:

Microprocessors designs consist of many micro level chips that reaches to a state where thermal upsurge occurs due to rapid processing of data and effect (reduce) their efficiency in many different aspects. That production of heat can cause disintegration which makes the chips disable of doing many functions they are assigned to perform. Embedded devices are designed to combine hardware and software, software integration can insert to hardware to perform some specific function. Multicore embedded devices are in different shapes and dimension. It has various applications on larger scale in networking and nuclear powerhouses to small multimedia players printers, automobiles, cameras mobile handset due to higher demand of power the energy becomes the major concern of the multicore devices for this a thermal aware scheduling algorithm has been proposed that consider the migration of load from higher state to that of lower state and considers all type of tasks and forecast them according to the priority by maintaining the previous history. The proposed technique also considers various thermal values by consulting the previous priorities of task on multicore systems. Migration policy is used to share load from one core to another the algorithm efficiently decreases almost 3°C temperatures at 40% utilization and the energy utilization is 221.3 J which is 3.12 % improved as compared to the global EDF.

Keywords: *Storm, Atmi, Dynamic Power Management, Dynamic Thermal Management*

1. Introduction

Now a day's systems are getting advanced and they are characterized as they have "real time" necessities to operate efficiently. In such embedded machines, the performance of the system does not consider those results that come at the end of complete

simulation of a task in a multi-core system, but consider the duration during which the required results can be achieved [1]. Such kind of systems is generally recognized as "Real Time Systems". These systems are usually linked to such applications that need

¹ Department of Electrical Engineering, Gomal University, D.I.Khan, KPK, Pakistan

² Department of Electrical Engineering, The University of Lahore, Lahore, Pakistan

³ Department of Computer Science, Superior University, Lahore, Pakistan

⁴ United Consulting Services, D.I.Khan, KPK, Pakistan

to have a smooth operation for all the applications that required reliability and cannot afford any erroneous results in such real time systems all the deadlines meet at the accurate time and no chances for the system to miss its deadline. If a deadline is missed a failure as response can occur. These devices are called hard real-time systems a very common examples on larger scale for hard real time systems are avionics systems, textile industries, nuclear power plants, and also those devices that operates with wire systems established by present advanced automobiles [2, 3]. 'Figure 1 illustrates the block diagram of real time system'.

While on other side systems in which the real time limit exists, but the system has the ability to function whether a multi-core system missed the deadlines or either releases late. Such type of system is recognized as "Soft real time embedded system". These embedded devices are safer as compared to hard. If the time limit for each task is missed and yet the system is functional such systems are known as "Soft" real time systems. The performance of the system can be slightly disturbed and can affect the total performance of the system. Such systems can be used in networking of cellular systems [4]. While there is another system in which the missing of deadline is acceptable. In this case, the deadline of the task takes place. Then there is no improvement in the overall results either such systems are known as "firm" real time such systems that can afford delay of few seconds like a server of network that can afford delay. A usual Real time system can behave as hybrid. Sometime it behaves as "hard" real time, for which all the deadlines can meet according to their arrangements for the efficient and safe functionality of the system and some other embedded aspects like "firm" and "soft" real time where the infrequent failure is acceptable. There are few characteristics of RTES are as follow [5]. In real-time embedded systems the overall accuracy of

the system depends on both the useful accuracy of the system and also on time constraints. Real-time systems know about significant information of the application running on the system. Real time system depends on deadlines predictability and consistency. Fault acceptance characteristics. Hard, soft and firm are those real time systems that are commonly used everywhere and they are linked with a conventional multi-core system. These are further categorized in few other types like interactive real time that gives response to a user while the application is in running mode these types of embedded devices depend on time so the user can expect the response of the system well on time. In such cases the user can have a better system's speed and have a continuous process and have very less chances of missing a deadline such systems have almost the same characteristics like soft real time embedded systems. All through in this research work, the word "real time embedded system" is concerned with properties of an embedded systems and the real time. Those devices that are in real time embedded system category have a larger number of common characteristics. Real time systems have characteristics like embedded while embedded systems usually have characteristics like Real time on larger level or either on smaller extent [6].

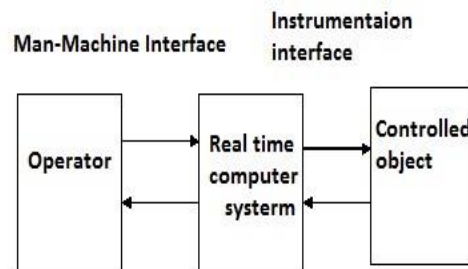


Fig 1: Block diagram of real time system.

A. Events in Real-time Systems

In Real time systems events are categorize in two forms.

Synchronous Events: The release time assigned to all the tasks of an application is same.

Asynchronous Events: The release time assigned to all the tasks of an application is random. Figure 2 describes how a real time system works with synchronous and asynchronous events.

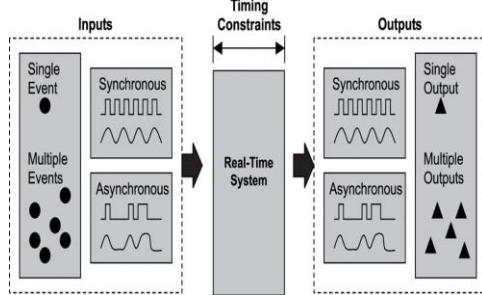


Fig 2: A simple view of real-time systems [7]

B. Function of RTES

There are few functions of real time embedded systems. RTS, Nuclear power plants monitoring and scheduling, EMB, Printer, MP3 player and cell phone. RTSEMB Cardiac pacemaker.

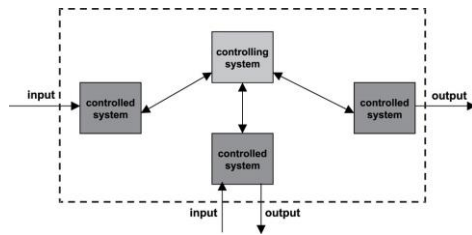


Fig 3: An Input/output Structure of Real time systems

RTES are those systems for which it is necessary to execute the set of jobs allotted to it firmly within the pre-set limit restrictions. Which is guaranteeing that all time-critical tasks are handled within time is called real time system with hard deadlines.

These tasks trust deeply on computational analysis and data plays an important role for

achievement of the specific task to avoid scheduling issues all the resources that are distributed try to meet the time constraints for the task not to miss any deadlines [8].

2. Literature Review

Real-time systems are divided into hard and soft. The scheduling mechanism for hard are functional for soft real-time scheduling. Static and dynamic are the two further types of hard real time scheduling. The scheduling choices are usually occupied during the compilation time of task that depends on previous information about task set constraints [9]. Static scheduling occurs when the completing time for all the tasks are in the condition where scheduling by the scheduler finds earlier to the start of execution.

Tasks in which the information has some deadlines and the time for their execution is required are called static scheduling. The scheduling techniques occur during offline execution of tasks are static. In difference to this, dynamic scheduling makes quick decisions when the task is in running state by finding the set of tasks currently executing. This type of scheduling affords flexibility.

Dynamic scheduling represents such scheduling where the execution time for the task is not continuous and changes with time according to scheduler instructions.

During execution the process is completed dynamically when there is a limited priority assign to a task. In such order task are executing according to the plan settled during run time and deviates by the instructions provided from the scheduler. Executions of tasks with or without any issue are likely with both static and dynamic scheduling. Real time operating system proposed various scheduling mechanisms.

All these methods for scheduling have that issue that they allow prompt tasks to achieve execution by disturbing the execution of running tasks. All of them fall in the

classification of non-preemptive scheduling [10].

There are few Scheduling characteristics that are necessary for scheduling of a multi core real time system that are as follow [11, 12, and 13].

- Guaranteeing that system timing restraints are meeting.
- Avoid synchronized installment of devices and mutual resources.
- Attaining a state where utilization is very high.
- Confirms that the dispatch cost of a real-time systems must be low

Fundamentally, the scheduling problem includes expressing a plan for the implementation, execution and completion of all the tasks [14]. Due to advancement in embedded technology periodic activities shows more computational demand on a larger scale in a multi-core system these activities are categories in three types in a task model that are Periodic, A Periodic and Sporadic.

Figure 2-9 describes the various categories of jobs that are used in a processor' [15].

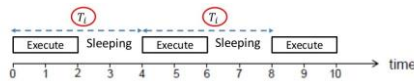


Fig 4: Categories of Task Model

C. Periodic Task

In system monitoring periodic scheduling plays very important role periodic tasks usually occurs during control loops. These kinds of activities are continuously executing on definite rates. Periodic task repeats itself at specific periodic time interval in a continuous manner e.g. Control loops, Sensor reading [16]. Figure 5 illustrates how periodic tasks are scheduled for the completion of their jobs in a specified time.

For better Scheduling of task few notations are introduced that are as follow. For generic task sets i represents a periodic task while T_i

represents the time period of a general periodic tasks and ϕ_i represents the phase of task i . T_{in} represents the n th occurrence to a task "i" and R_{ia} represents the release discharge duration for n th instance of task i in periodic task there are few major parameters that are used are written below. Reaction duration of an occurrence of task is the time in which the instance is finished. The time starts and calculates from the start of an instance to the discharge of a task at that time instance it is finished.

$$R_{ak} = f_{ak} - T_{ak}$$

Significant instant of a task:

The time when larger response is created at the release of a job.

Relation when release of a task occurs:

Among two continuous instants the major deviation of change is known as relative release of a task.

$$RRJ_a = \max | (2i, f_c - r_a, f_c) - (2i, f_c - a - r_a, f_c - a) |$$

Complete discharge jitter of a Job

The instantaneous change in the duration between a single point instance and the maximum instant in which the deviation of a task occurs [43].

$$ARJ_a = \max (2a, f_c - r_a) - \min (sa, f_c - r_a)$$

There are three basic categories of scheduling algorithms that uses periodic tasks are Rate monotonic scheduling, earliest deadline first Scheduling and deadline monotonic scheduling.

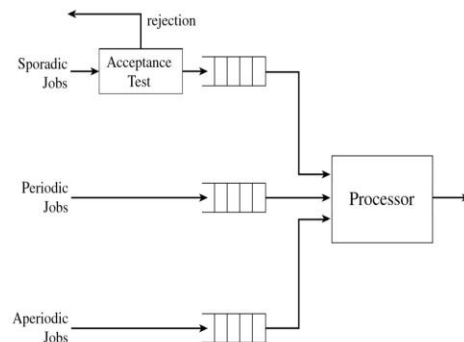


Fig 5: Periodic Task Model

D. A Periodic Task

A periodic task can occur at any time interval when the task is in running mode. There is no specific instant for the occurrence of aperiodic task. Aperiodic task can have very elastic deadline and sometime aperiodic task doesn't have any deadlines e.g. Alarm clocks and all emergency embedded devices that can arrive at any time instant during emergency like robotic cars and obstacle avoidance system [17]. Figure 6 illustrates how aperiodic tasks are scheduled and jobs occur at any time instant.

3. Problem Statement

The main objective and focus of this research work is chip (reliability) that is the most important issues in multi core embedded technologies. High temperature causes multiple performances and reliability issues.

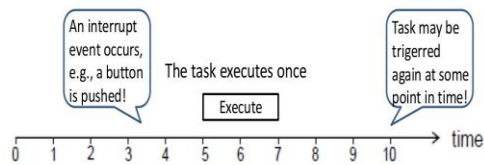


Fig 6: A Periodic Task Model [18]

Task migration is the way to avoid high temperature and improve performance. However accurate prediction about the coolest core and the task which needs to migrate to control the chip temperature is the major design issue in task migration techniques. Our technique considers all types of tasks including hot and cold task to accurately predict the temperature of the core in running mode and also use precisely thermal history to precisely check the chip temperature for the selection of cold core. A task migration technique based on previous history by considering workload of running core to predict the future temperature and coolest cores. This technique improves the reliability by avoiding thermal issues [19].

4. Experimental Technique

The major components include a task producer where user can create random task sets in XML file that contains different parameter. This XML input file is used as an input for the simulation tool. The simulation tool for real-time multiprocessor scheduling can simulate the input file according to the scheduling policy and developed the power profiles according to the set of parameters given in XML file. These Power profiles generated from STORM can be saved in a text output file. The XML file contains all the information of task sets. A temperature model can also analyze the thermal responses of different scheduling algorithms. A statistical testing mechanism is used to test a many data set. These data sets are already stored produces thermal profile when used with some hardware constraint in thermal model ATMI and a power model is used for each core comprises of static and dynamic power.

Total power = dynamic power + static power
 $Dynamic\ power = C \times F \times V$
 Therefore, C is the switching capacitance. F is the frequency and V is the supply voltage
 Static power = Leakage current \times V

Flow chart

We have proposed technique for octa core multiprocessing Least processor in which the scheduler will first check the number of task jobs as shown in figure 7. That is running in an application. Primarily the value of counter is zero for all 8-cores in a multiprocessing least processor. The scheduler is able to determine the constraint and values of every individual task. So, the counter for all the cores is initially zero. The configuration of core will be selected on the basis of least temperature and lower Power of consumption among all other configurations once the core with least temperature is

selected its counter in running mode is rapidly rising. In such case when the temperature of the core in running mode is less than the maximum allowable temperature then the scheduler will perform normal process of task execution.

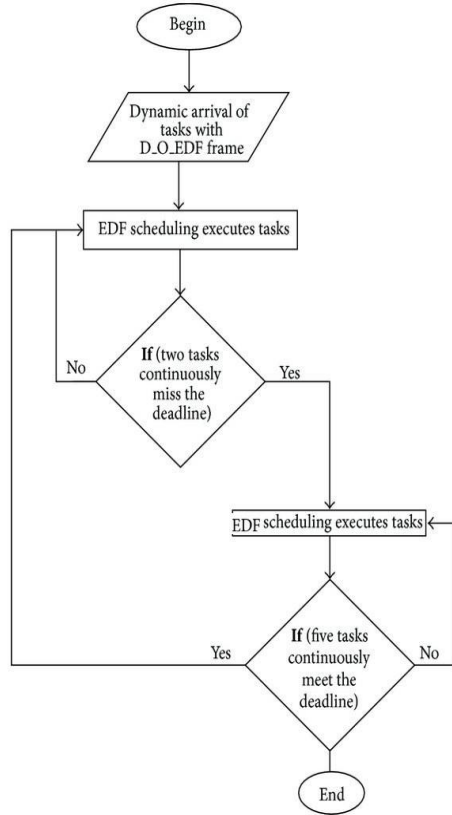
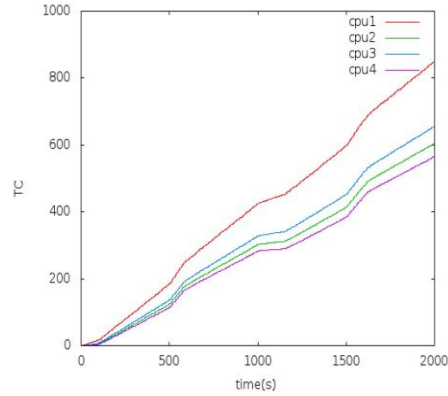


Fig 7: Flow Chart

5. Experimental Results

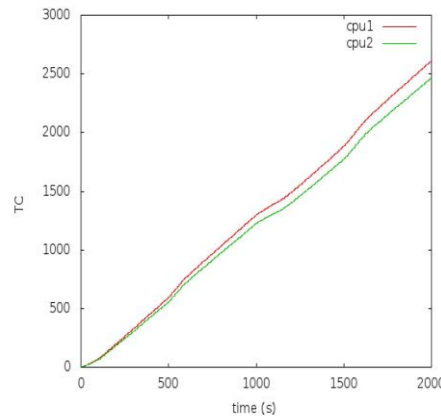
In this section we discuss our experimental results that illustrates the temperature variation between the curves of proposed EDF and Global EDF. Proposed EDF considers reliability and performance

parameter so only those cores are in running state that is in working condition. In the beginning exponentially, temperature on chip rises and then arrive at a steady state level. At 13% utilization factor the global EDF has 5°C more temperature on chip as



(c)

Fig 8: Thermal Cycling under 80Mhz for CPU1 & CPU2 at low Utilization Factor



(a)

Fig 9: Thermal cycling under 80Mhz for CPU1 & CPU2 at high Utilization Factor

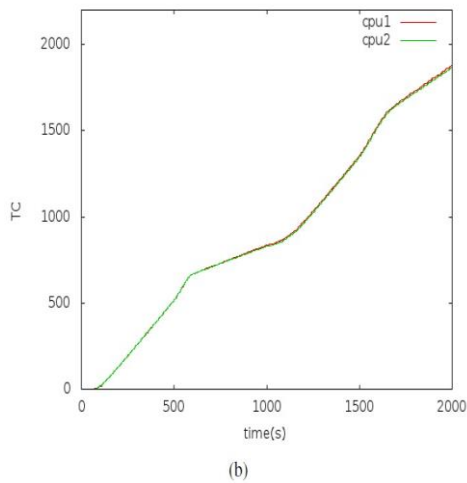


Fig 10: Thermal cycling under 80Mhz for CPU1, CPU2, CPU3 and CPU4 at low Utilization Factor

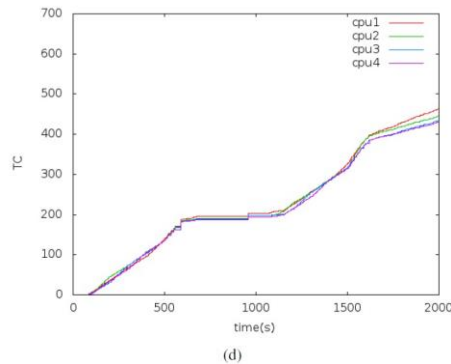


Fig 11: Thermal Cycling under 80Mhz for CPU1, CPU2, CPU3 and CPU4 at High Utilization Factor

6. Conclusion and Future Work

In this research work functional simulation on STORM and thermal model and proved its practicability in the context of MPSoC. A large amount of techniques now a days doesn't consider the affect of environmental temperature which can be investigated in future work. Majority of the current algorithms dsefines only for homogeneous multi-core systems which can be extensive to heterogeneous multi-core systems.

REFERENCES

- [1] Matt Bach. How ambient temperatures affect your pc. Temperatures-Affect-Your-PC-158/, Written on August 15, 2012.
- [2] Shekhar Borkar. Design challenges of technology scaling. *IEEE Micro*, 19(4):23–29, July 1999.
- [3] Thidapat Chantem, Xiaobo Sharon Hu, and Robert P. Dick. Temperature-aware scheduling and assignment for hard real-time applications on mpsoCs. *IEEE Trans. VLSI Syst.*, 19(10):1884–1897, 2011.
- [4] Jian-Jia Chen, Shengquan Wang, and Lothar Thiele. Proactive speed scheduling for real-time tasks under thermal constraints. In *Proceedings of the 2009 15th IEEE Symposium on Real-Time and Embedded Technology and Applications, RTAS '09*, pages 141–150, Washington,DC, USA, 2009. IEEE Computer Society.
- [5] H. Khan, M. U. Hashmi, Z. Khan, R. Ahmad, and A. Saleem, "Performance Evaluation for Secure DES-Algorithm Based Authentication & Counter Measures for Internet Mobile Host Protocol," *IJCSNS Int. J. Comput. Sci. Netw. Secur.* VOL.18 No.12, December 2018, vol. 18, no. 12, pp. 181–185, 2018.
- [6] Farzan Fallah and Massoud Pedram. Standby and active leakage current control and minimization in cmos vlsi circuits. *IEICE Transactions*, 88- C(4):509–519, 2005.
- [7] Stephen H. Gunther, Frank Binns, Douglas M. Carmean, and Jonathan C. Hall. Managing the impacincreasing microprocessor power consumption. (Q1):9, February 2001.
- [8] H. Khan, M. U. Hashmi, Z. Khan, and R. Ahmad, "Offline Earliest Deadline first Scheduling based Technique for Optimization of Energy using STORM in Homogeneous Multi- core Systems," *IJCSNS Int. J. Comput. Sci. Netw. Secur.* VOL.18 No.12, December 2018, vol. 18, no. 12, pp. 125–130, 2018.
- [9] Khan, H., Bashir, Q., & Hashmi, M. U. (2018). Scheduling based energy optimization technique in multiprocessor embedded systems. 2018 International Conference on Engineering and Emerging Technologies (ICEET).doi:10.1109/iceet1.2018.8338643
- [10] Joohno Kong, Sung Woo Chung, Kevin Skadron, "Recent Thermal Management Techniques for Microprocessors" *ACM Computing Surveys*, Vol. 44, No. 3, Article 13, pp. 13:1-13:42, June 2012

- [11] Mehdi Kamal, A. Iranfar, A. Afzali-Kusha, M. Pedram, "A Thermal Stress-aware Algorithm for Power and Temperature Management of MPSoCs" In EDAA, 2015.
- [12] Q. Bashir, H. Khan, M. U. Hashmi, and S. Ali zamin, "A Survey on Scheduling Based Optimization Techniques in Multi-Processor Systems," in Proceedings of the 3rd International Conference on Engineering & Emerging Technologies (ICEET), Superior University, Lahore, PK, 7-8 April, 2016., 2016.
- [13] T. S. Rosing, K. Mihic, G. De Micheli, "Power and Reliability Management of SoCs," In IEEE Transaction Very Large Scale Integrated System (VLSI), vol. 15. no.4, pp.391-403, 2007.
- [14] Pratyush Kumar, Lothar Thiele, "Thermally Optimal Stop-GO Scheduling of Task Graphs with Real Time Constraints," In ASP-DAC, IEEE Press, pp. 123-128, 2011.
- [15] Alexandru Andrei, Petru Eles, Zebo Peng, Marcus T. Schmitz, Bashir M. Al Hashimi, "Energy optimization of multiprocessor systems on chip by voltage selection," IEEE Transaction on VLSI, vol 50, no.3, 2007.
- [16] "1965 – "Moore's Law" Predicts the Future of Integrated Circuits". Computer History Museum, 2007
- [17] C. J. Lasance, "Thermally Driven Reliability Issues in Microelectronic Systems: Status-quo and Challenges". Microelectronics Reliability, pp. 1969–1974, December 2003.
- [18] H. Khan, Q. Bashir, and M. U. Hashmi, "Scheduling based Energy Optimization Technique in multiprocessor Embedded Systems," in 2018 International Conference on Engineering and Emerging Technologies (ICEET). doi:10.1109/iceet1.2018.8338643, 2018.
- [19] H. Khan, S. Ahmad, N. Saleem, M. U. Hashmi, and Q. Bashir, "Scheduling Based Dynamic Power Management Technique for offline Optimization of Energy in Multi Core Processors," Int. J. Sci. Eng. Res. Vol. 9, Issue 12, December-2018, vol. 9, no. 12, pp. 6–10, 2018.
- [20] G. Petrone, G. Seagnuolo, R. Teodorescu, "Reliability Issues In Photovoltaic Power Processing Systems", IEEE Transactions, July, 2008.