International Conference on Information Technology, Electronics and Communications (ICITEC – 2011)
Hyderabad, India, November 29-30, 2011

# Real Time Scheduler Simulator Framework for Multi-Processor Embedded Application: A Review

P .Joel Josephson, Dr. R. Ramesh and C. Yaashuwanth
Department of Electrical and Electronics Engineering
Anna University Chennai, Chennai 600025.

**Abstract:** *A Real Time System (RTS) is one which controls an environment by receiving data, processing it, and returning the results quickly enough to affect the functioning of the environment at that time. The main objective of this research is to develop an architectural model for the simulation of real time tasks in multi-processor environment and to implement a new scheduling algorithm. The proposed model can be used for real time scheduling algorithms for multi-processor systems. A Framework is developed for explaining the functionality of the Scheduler in allocating the tasks for the available processors. The results in the Framework are explained with the help of timing diagrams.*

*Key words: Real time, Scheduler, Frame work*

## 1. NTRODUCTION

A Real-time system is any information processing system which has to respond to externally generated input stimuli within a finite and specified period. The correctness depends not only on the logical result but also the time it was delivered. A deadline is given time after triggering an event, by which response has to be completed. A Hard Real time system is a system in which an overrun in response time leads to potential loss of life and/or big financial damage. It is a system which should meet the timing requirements of the system, computations must always be met or the system will fail. A Soft Real time system is a system in which deadline overruns are tolerable, but not desired. It is a system that has timing requirements, but occasionally missing the task deadlines have negligible effects. (Shin and Krishna, 1997) Given these characteristics and the relative priorities of tasks and interrupts in the system, it is possible to analyse the worst-case performance of the software and the real-time characteristics of the system.

(Jian Huang and Jooheung Lee 2011) Explained about the Reconfigurable Architecture to support Zero quantized discrete cosine transform(ZQDCT) is proposed and a hybrid model based quality priority algorithm is developed to reduce power consumption, required hardware resources, and consumption time with a small quality degradation.

(Ashish Shenoy, Jeff Hiner, Susan Lysecky, Roman Lysecky, Ann Gorgon Ross 2010) Described about Several Profiling methods for dynamically monitoring sensor based platforms and analyze the associated network traffic, energy and code impacts.

(Xi Jin, Nan Guan, MinSong Lv,and Quinxu Deng 2010) Impulse C is a high level language widely used in Hardware/Software Co-Design and provides users with various Hardware Software Communication mechanisms. Here a improved implementation of Shared memory communication in Impulse C is presented.

(Mohammed N Sabry, Jose.L.Ayala, and David Atienza 2010) A Compilation flow that utilizes the register windows to optimize the thermal profile and to reduce the hot spots is proposed.

(Serge Middonet, Damien Masson,and Remie Lassalle 2010) An approximate slack stealer Algorithm like the minimal approximate slack stealer(MASS) is a good solution for real time Embedded System, this algorithm is demonstrated here.

(Arnab Sarkar, Amit Shanker, Sujoy Ghose, P.P.Chakrabarthi 2011) A mechanism on Partition-Oriented ERfair Scheduler achieves lower overhead using an online partitioning/merging mechanism that retains the optimal schedulability of a fully global scheduler by merging processor groups as resources become critical while using partitioning for fast scheduling at other times

(Arshdeep Bahga and Vijay K Madisetti 2011) The Design of Resource Manager and Master Scheduler for the open CLose environment that allowa efficient realization of multiple applications within a multi tasked platform is described here.

(J. Venkataramanan and Xiaojun Lin) The overflow metric is appropriately modified, the minimum-cost-to-overflow under the algorithm can be achieved by a simple linear path, and it can be written as the solution of a vector-optimization problem. Finally, this result enables us to design scheduling algorithms that are both close to optimal in terms of the asymptotic decay rate of the overflow probability and empirically shown to maintain small queue-overflow probabilities over queue-length ranges of practical interest

(Zhe Yang and Lin Cai and Wu-Sheng Lu 2010) Extensive simulations are conducted to demonstrate the effectiveness and efficiency of the proposed scheduling algorithms which are found to achieve throughputs close to the exhaustive searching results with much lower computational complexity.

(Li Jie and Guo Ruifeng and Shao Zhixiang 2010) Distributed real-time scheduling algorithms such as GRMS and DSr are analysed. Finally, it points out the future direction of real-time scheduling research.

(Sai Rahul Chalamalasetti, Sohan Purohit and Martin Margala and Wim Vanderbauwhede 2010) We also present the communication management scheme between the processors in the presence of varying degrees of single event upsets. The impact on throughput while evaluating an 8*8 discrete wavelet transform (DWT) and 8*8 discrete cosine transform (DCT) are also presented

(Jiayin Li Meikang Qiu Jianwei Niu Wenzhong Gao Ziliang Zong Xiao Qin 2010 ) we present a preemptable job scheduling mechanism in cloud system. We propose two feedback dynamic scheduling algorithms for this scheduling mechanism. We compare these two scheduling algorithms in simulations. The results show that the feedback procedure in our algorithms works well in the situation where resource contentions are fierce.

(Karthik Shankar,Roman Lysecky 2010) We present an area-efficient control focused soft error detector (CNFSED) capable of nonintrusively detecting soft errors within the execution of a software application without modifications to the software application or the target processor. This soft error detector achieves an error detection rate greater than 90% for control errors and 85% of unmasked errors while incurring minimal area overhead.

(Raphael Kunis and Gudula Rünger 2010) Especially, a block identification algorithm is proposed, which identifies suitable blocks of tasks in arbitrary layer-schedules and, thus, allows the application of the movement of blocks to a wide range of layer-based scheduling algorithms. The block identification and the movement algorithm are applied to two scheduling algorithms and show good performance improvements.

(Yi He, Zili Shao, Bin Xiao, Qingfeng Zhuge, Edwin Sha ) To deploy such heterogeneous embedded systems in critical applications, an important research problem is how to maximize system reliability while satisfying the required time constraint. .In this paper, we study the heterogeneous reliability scheduling problem is studied.

(Muthucumaru Maheswaran and Howard Jay Siegel) A heterogeneous computing system provides a variety of different machines, orchestrated to perform an application whose subtasks have diverse execution requirements.

(Yu-Kwong Kwok and Ishfaq Ahmad) In this paper, we propose a static scheduling algorithm for allocating task graphs to fully connected multiprocessors.

(Paulo Baltarejo Sousa, Bj □ rn Andersson, and Eduardo Tovar) In this paper, the challenges and design principles for implementing slot-based task-splitting algorithms on multiprocessor systems running the Linux kernel.

(Greg Levin, Caitlin Sadowski, Ian Pye, Scott Brandt) In this paper, we examine exactly how fluid scheduling techniques overcome inherent problems faced by greedy scheduling algorithms such as EDF. Based on these foundations, we describe a simple and clear optimal scheduling algorithm which serves as a least common ancestor to other recent algorithms.

(A. Burns, R.I. Davis, P. Wang and F. Zhang) An EDF-based task-splitting scheme for scheduling multiprocessor systems is introduced in this paper.

## 2. Significance of Multi-Processor Environment

2.1 Multi-Processor Environment for Embedded Systems: In an embedded system the multi-processors are necessary for executing individual tasks to develop an application. The type of programmable multi-processor platform is representative for signal-processing architectures where low power, and support for many features and standards is imperative.

2.2 Task Scheduling in Multi-Processor Environment: In a Multi-Processor environment a scheduler plays a major role of assigning a task to the processor. In other words scheduler synchronizes the execution of different tasks with the different processors.

2.3 Developing a simulator for task scheduling in Multi Processor Environment: Simulators try to model the behavior of the complete microcontroller in software. Some simulators go even a step further and include the whole system (simulation of peripherals outside of the microcontroller). Simulating external events can become a time-consuming exercise, as you have to manually create "stimulus" files that tell the simulator what external waveforms to expect on which microcontroller pin.

## 3. Study and Simulation of Scheduling Algorithms for Multi-Processors

3.1. Simulator: A simulator can also not talk to the target system, so functions that rely on external components are difficult to verify. For that reason simulators are best suited to test algorithms that run completely within the microcontroller (like a math routine for example).

3.2. Scheduling Algorithms for Multi-Processors: The Functionality of the Scheduler completely depends on the Scheduling Algorithm written. There are few scheduling algorithms to be considered for Multi-Processor Environment. They are

3.2.1. LPT (Longest Processing time) Algorithm: The longest processing time rule orders the jobs in the order of decreasing processing times. This algorithm is a heuristic used for finding the minimum make span of a schedule. It schedules the longest jobs first so that no large job will "stick out" at the end of the schedule and dramatically lengthen the completion time of the last job.

3.2.2. Johnson Algorithm: This is a heuristic algorithm used to solve the case of 2 – machine N job problem when all the jobs are processing in the same order.

3.2.3 Genetic Algorithm: It is a simple algorithm which runs on a local workstation which evaluates two shared memory multiple processing systems in real time, and the resulting performance data are then returned to the local work station for incorporation.

3.3. Existing Simulators vs Proposed Simulators

| Features | Real TSS | Cheddar | YASA | Proposed Simulator |
|---|---|---|---|---|
| Support Platforms | Windows, Linux, all TCL supported platforms. | Solaris, Linux, Windows, all GNAT/GTKAda supported platforms | Windows, Linux, Mac, OS X, FreeBSD and most of the commercial UNIX versions like Solaris, HP-UX, AIX | All Operating Systems |
| Programmin g Environmen t | Active TCL | Not available | QT, Cygwin, VC++ | C |
| Language Used | TCL | Ada/GTKAda | C | C |
| Output Resolutions | Timing diagrams, processor utilization, | Timing diagrams, statistics of missed deadlines, response | Timing diagram, statistics of missed deadlines, | Timing diagrams status Reports |

| | | | | |
|---|---|---|---|---|
| - | response time, task waiting times, missed deadlines, overruns | time, processor utilization | utilization, response time etc | |

3.3.1 Need for Simulators in Multi-Processor Environment: A Simulator is needed in Multi-Processor Environment to have a clear picture of the Hardware where the code written need to be implemented.

## 4. Development of Scheduler Framework for Multi-Processor environment

The Scheduler schedules the tasks for processing and the tasks processing to the assigned processor can be viewed on the Framework.

The Research work can be achieved by implementing the hardware with the help of the block schematic shown in the following figure 1

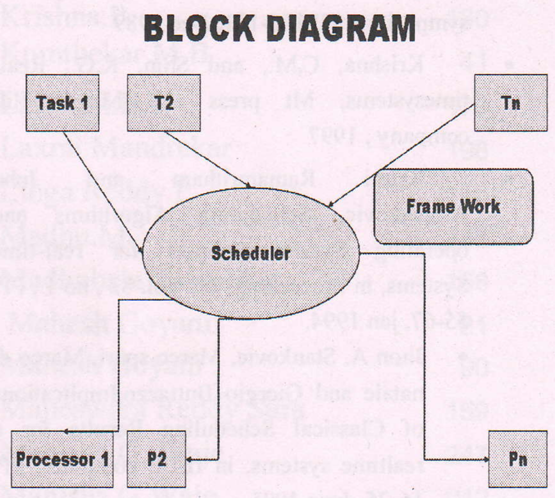

Figure1

## 5. Conclusion and Future Scope

Previously a framework has been implemented for a single processor. This work gives a facility to know the performance of processors. Scheduling is an essential one in real time systems to achieve the deadlines.

A new deterministic algorithm can be implemented for meeting the deadlines.

## 6. References

- Reinder J.Brill and Pieter J.L. Cuijpers, Analysis of hierchial fixed priority pre-emptive Scheduling revisitedin Technische university
- Sergio Saez, Joan vila, Alfons Crespo and Angel Garcia A Hardware Scheduler for complex real time Syatems in isie;99 ,pp 43-48,Bled,Slovenia, 1999
- Sha L., Abdelzaher, T, Arzen, k.,Cervin, a., Baker, T.P., Burns, AButtazzoG., caccamo, M.,Lehoczky,J., and Mok, A., real time scheduling theory . a historical prespective in real time system 28(2) 46-61 Nov.2004
- National Instruments Corpration, Labview Fundamentals Aug 2005
- Liu, c.l and James .Layland Scheduling Algorithm for Multiprogramming in hard real timeenvironmentin Journal of the ACM, vol. 20, no.1, PP. 46-6, Jan1973.
- Melissa Vetromille, Luciano ost,cesar,M.Marcon,Carlos Reif, Febianno Hessel, RTOS Schedular Implementation in Hardware abd Software for Real Time Applications in Proceedings uf 17th IEEE international workshop on Rapid System Prototyping (RSP'06), June 2006
- Paul Kahout, Brinda Ganesh and Bruce Jocob, hardware support for realtime operating systems, in codes-isss;03, pages 45-51, Newport Beach, Californea usa, oct., 2003
- Jhon A. Stankovic, Marco spuri, Marco di natale and Giorgio Buttazzo,Implications of Classical Scheduling Results for a realtime

systems, in IEEE computer, PP. 16-25, June 1995

- John Lehoczky. Lui Sha and Ye Ding, The Rate Monotonic Scheduling Algorithm: Exact Charecterization and Average Case Behaviour, in proceedings of real time systems symposium. PP.166-171, Dec 1989
- Krishna, C,M., and Shin, K.G., Real-timesystems, Mt press and Mcgraw-Hill company , 1997
- Krithi Ramamritham and John A.Stankovic, Scheduling Algorithms and operating System Support for real-time Systems, in Proceedings of, Vol. 82 , no 1 , PP. 55-67, jan 1994.
- Jhon A. Stankovic, Marco spuri, Marco di natale and Giorgio Buttazzo,Implications of Classical Scheduling Results for a realtime systems, in IEEE computer, PP. 16-25, June 1995

- John Lehoczky. Lui Sha and Ye Ding, The Rate Monotonic Scheduling Algorithm: Exact Charecterization and Average Case Behaviour, in proceedings of real time systems symposium. PP.166-171, Dec 1989
- Krishna, C,M., and Shin, K.G., Real-timesystems, Mt press and Mcgraw-Hill company , 1997
- Krithi Ramamritham and John A.Stankovic, Scheduling Algorithms and operating System Support for real-time Systems, in Proceedings of, Vol. 82 , no 1 , PP. 55-67, jan 1994.