Equal and Unequal Memory Partitioning and Task Scheduling for Multi-Processor System On Chip

Jenitha A [1] Dr. Elumalai R [2]

1. Associate Professor, Dept of ECE, EPCET,
Bidrahalli, Virgonagar Post, Bangalore – 49. Karnataka.
Email : 28jenitha@gmail.com, Phone : 9449680553

2. Professor and Head, Dept of IT , MSRIT,
Mathekeri, MS Nagar , Bangalore. Karnataka.
Email : elumalai.epcet@gmail.com Phone : 9448826813

***Abstract*: - The growing trend in embedded system is to deploy a Multiprocessor system on chip (MPSoC). MPSoC has heterogeneous processing elements, levels of memory hierarchy and input/output component which are linked together by an on chip interconnect structure. Such architectures provide the flexibility to meet the performance requirement of multimedia application while respecting constraint on memory, cost, size, time and power. The MPSoC is an attractive solution for increase in complexity and size of embedded applications. While embedded system becomes increasingly complex, the increase in memory access speed has failed to keep up with processor speed. This makes the memory access latency a major issue in scheduling the task of an embedded application on the processor and partitioning the memory among the processors are two critical issues in such systems. This paper presents an integrated approach to task scheduling and SRAM memory partitioning to reduce the execution time of embedded applications.**

Keyword:- MPSOC, Task scheduling, Memory partitioning, SRAM

I. INTRODUCTION

Due to clock and power constraints, architecture with multiple processor on a single chip have become attractive solution to achieve performance in both high end and low end computing Multiprocessor with large number of different processing cores are now common for variety of reasons, especially in embedded systems. MPSoC consist multiple heterogeneous processing element, memory hierarchy, input/output components which are linked together by an on chip interconnect structure. A critical component of a chip multiprocessor is its memory subsystem. This is because both power and performance behaviour of a chip multiprocessor is largely shaped by its on-chip memory. While it is possible to employ conventional memory designs such as pure private memory or pure shared memory, such designs are very general and rigid, and may not generate the best behaviour for a given embedded application. Execution time predictability is the critical issue for real time embedded applications; this means that data caches are not suitable and hard to model the exact behavior and to predict the execution time of tasks. Software controlled memories allow execution times to be predicted with accuracy.

There is a large number of complex embedded applications consisting of multiple concurrent real Time tasks [1]. The problem of scheduling is weighted directed acyclic graph (DAG), also called a task graph or macro dataflow graph, to a set of homogenous processor to minimize the completion time. The tasks can be scheduled on different processors. To alleviate such problems, many modern MPSoC systems use

SRAM memories to contribute for better timing predictability. Usually tasks are scheduled first and SRAM budget is then partitioned among the processors such a decoupled technique may prevent better scheduling in terms of minimizing the computation time of the whole application, traditionally these two steps are performed separately where tasks are usually scheduled first and the SRAM budget is then partitioning among the processors.

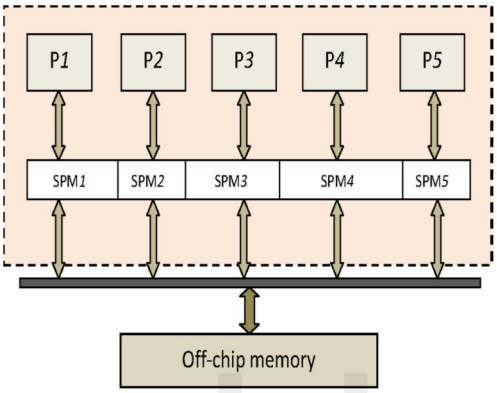

Fig. 1. Architectural model example with five processors, SPM budget, off-chip memory, and interconnection buses.

Organization of the paper has been done with following sections. Section II presents related work Section III presents the problem definition IV presents the implementation of the proposed algorithm.

II Related work

Benini *et al.* [1] proposed a complete approach for solving a complex allocation and scheduling problem for MPSoC. This approach is based on problem decomposition where the allocation is solved through an Integer programming solver, while the scheduling through a Constraint Programming solver.

Kandemir *et al.* [7] Presented a compiler strategy to optimize data accesses in regular array-intensive applications running on embedded multiprocessor environments. Specifically proposed an optimization algorithm that targets the reduction of extra off-chip memory accesses caused by inter-processor communication.

Suhendra *et al.* [9] explored optimization of scratchpad memory (SPM) in the context of embedded chip multiprocessors. In this technique an ILP formulation to capture the interaction between task scheduling and memory optimization is explored. The results are (1) flexible partitioning of the SPM budget among the processors can achieve up to 60% performance improvement compared to equal partitioning and (2) integrating memory optimization with task scheduling can improve performance by up to 80%. The paper also suggests to use the optimal performance limits as guidelines to design effective heuristics as future work.

Hassan Salamy et al. [11] presented an effective heuristic that integrates task scheduling and memory partitioning of embedded applications on multiprocessor systems-on-chip with scratchpad memory. Compared to the widely-used decoupled approach, this integrated approach significantly improved the results since the appropriate partitioning of SPM spaces among different processors depends on the tasks scheduled on each of those processors and vice versa.

III Problem Definition

In a Biomedical Embedded application with MPSoC architectural model and SPM budget

1. Finding better scheduling algorithm for MPSoC
2. Partitioning the SPM during dynamic allocation among the processors.
3. Assigning data variables to private SPM other than allocated to the processor when critical scheduling in carried out.

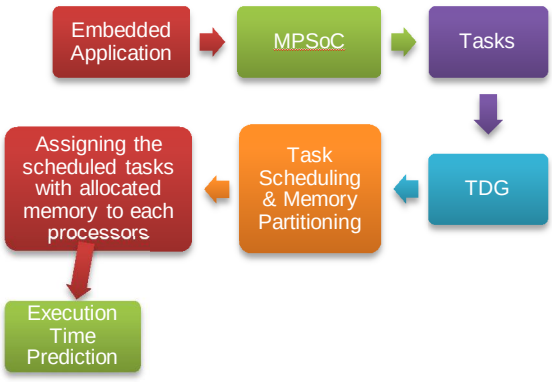

Fig 2: Block diagram of the project

IV IMPLEMENTATION OF THE PROJECT

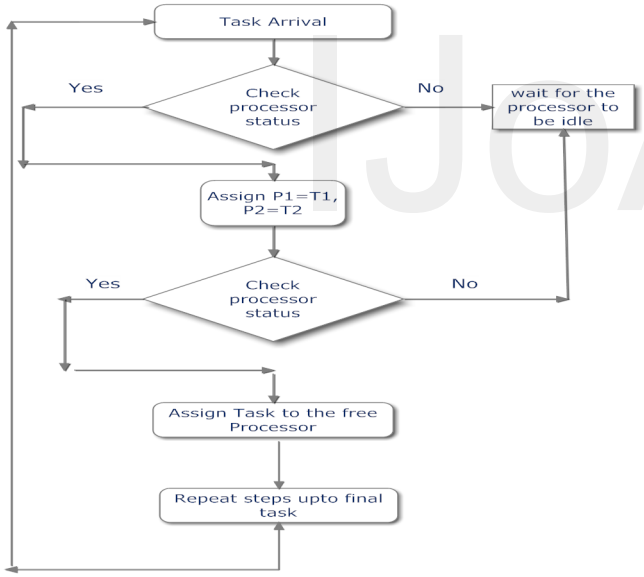

Fig 3: Flow chart of task scheduling

In general, there are 3 states that a task can be in:

1. Active. There can be only 1 active thread on a given processor at a time.
2. Ready. This task is ready to execute, but is not currently executing.
3. Blocked. This task is currently waiting on a lock or a critical section to become free.

Consider the example task graph shown below with six tasks, T1, T2, T3, T4, T5, and T6. Task T4 depends on tasks T1, T2 and T3, and task T6 depends on tasks T4 and T5. Any time there is an edge between two tasks Ti and Tj means that a communication cost should be accounted for, provided that these two tasks are allocated to two different processors. In the example taken tasks T1, T2, T3, and T5 are ready to be scheduled. Task T5 will not be scheduled at this point based on its task dependence value. Thus, first tasks T1 and T2 will be mapped to the two available processors P1 and P2.

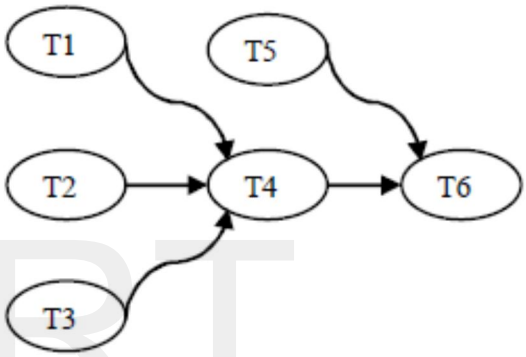

Fig 4:-Task dependence graph

The scheduling algorithm will map T3 to P2 as it is free before P1 since the computation time of T2 is less than that of T1. In a similar fashion, the scheduling algorithm will assign tasks T4 and T6 to processor P1 whereas task T5 will be mapped to processor P2.

- Decoupled task scheduling and memory partitioning assuming equally partitioned SPM among all available processors,

- Decoupled task scheduling and memory partitioning with memory partitioned among different processors with any ratio.

Unlike current approaches that treat task scheduling and memory partitioning as two separate problems, these two problems can be solved in an integrated fashion.

V EXPERIMENTAL RESULTS

The simulation result of task scheduling and memory partitioning for Multiprocessor System on Chip (MPSoC) was done on XILINX ISE Design Suite

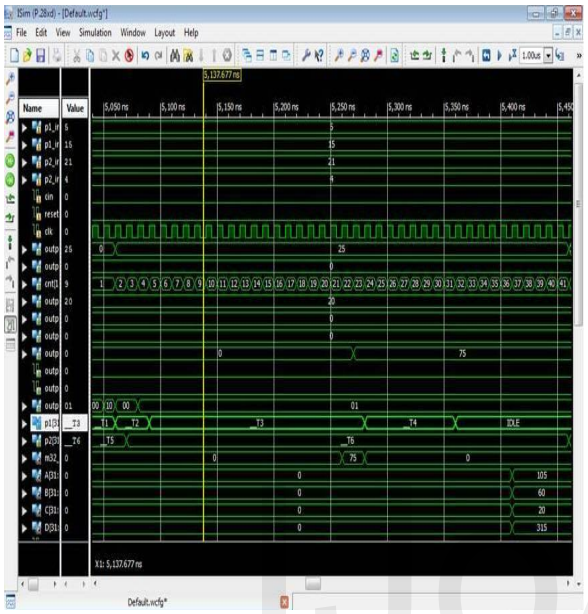

Fig 5**:** Result of task scheduling

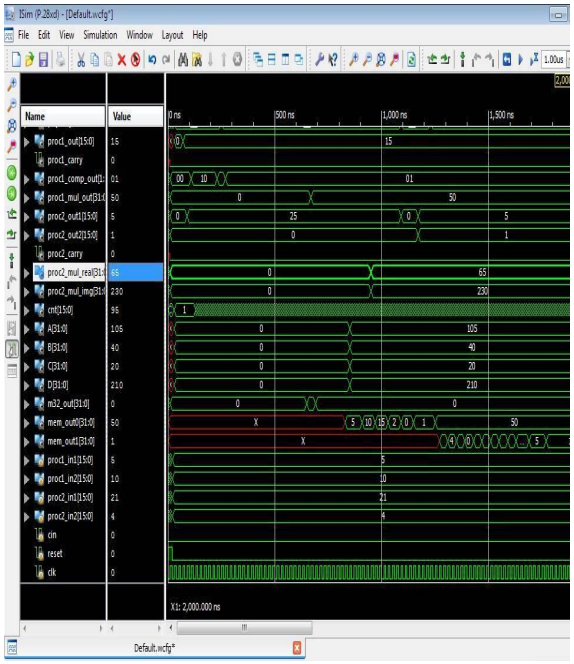

Fig 6: Result of memory partitioning

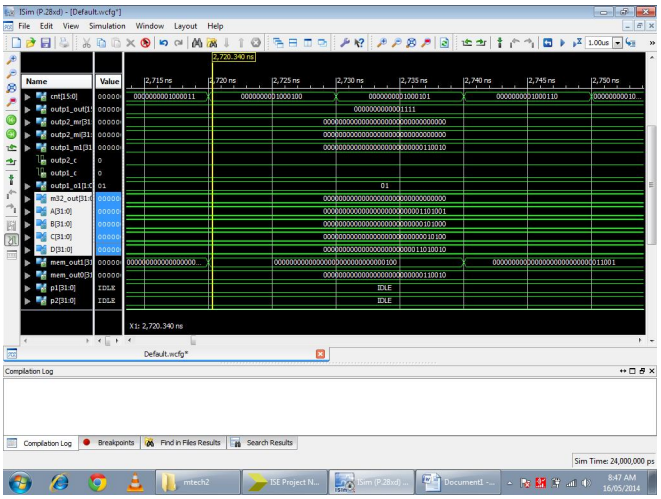

Fig 7 Task scheduling and equal memory partitioning

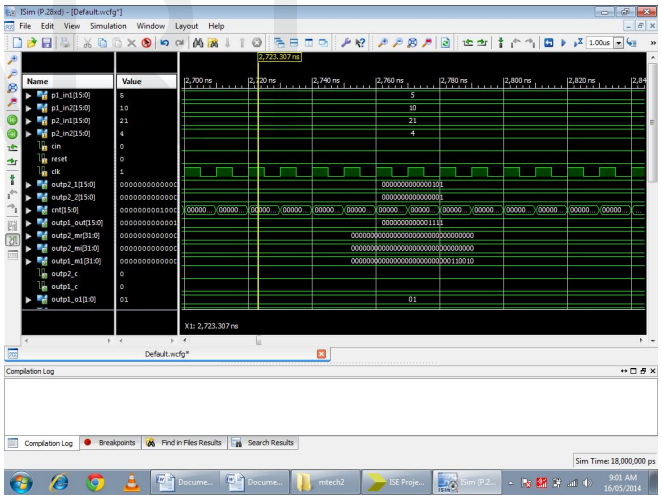

Fig 9 : Task scheduling with unequal memory partitioning

VI CONCLUSION

An effective heuristic was presented that integrates task scheduling and memory partitioning of embedded applications on multiprocessor systems-on-chip with scratchpad memory. Compared to the widely-used decoupled approach, this integrated approach significantly

improved the results, since the appropriate partitioning of SPM spaces among different processors depends on the tasks scheduled on each of those processors and vice versa.

Thus the reduction in the execution time of the tasks scheduled on the processors.

COMPARISON OF THE PREVIOUS RESULTS AND PRESENT RESULTS:-

	Execution time (Previous result)	Execution time (Present result)
1.Task scheduling and Memory partitioning using Cache memory	800ns	700ns
2.Task scheduling and Memory partitioning using Equal memory partitioning	700ns	600ns
3.Task scheduling and Memory partitioning using Non equal memory partitioning	600ns	500ns

References

[1] L. Benini, D. Bertozzi, A. Guerri, and M. Milano, "Allocation and scheduling for MPSOC via decomposition and no-good generation," in *Proc. IJCAI*, 2005, pp. 107–121.

[2] P. Luh, D. Hoitomt, E. Max and K. Pattipati, "Schedule Generation and Reconfiguration for Parallel Machines," *Robotics and Automation*, Vol. 6, No. 6, 1990, pp. 687- 696. http://dx.doi.org/10.1109/70.63271

[3] K. Vivekanandarajah and S. Pilakkat, "Task Mapping in Heterogeneous MPSoCs for System Level Design," 13*th IEEE International Conference on Engineering of Com- plex Computer Systems*, Belfast, 31 March 2008-3 April 2008, pp. 56-65.

[4] T. Adams, K. Chandy and J. Dickson, "A Comparison of List Schedules for Parallel Processing Systems," *Com- munications of the ACM*, Vol. 17, No. 12, 1974, pp. 685- 690. http://dx.doi.org/10.1145/361604.361619

[5] H. Topcuoglu, S. Hariri and M. Wu, "Performance-Ef- fective and Low-Complexity Task Scheduling for Het- erogeneous Computing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 3, 2002, pp. 260- 274.

[6] E. Ilavarasan and P. Thambidurai, "Low Complexity Per- formance Effective Task Scheduling Algorithm for Het- erogeneous Computing Environments," *Journal of Com- puter Sciences*, Vol. 3, 2007, pp. 94-103.

[7] M. Kandemir, J. Ramanujam, and A. Choudhury, "Exploiting sharedscratch pad memory space in embedded multiprocessor systems," in *Proc. DAC*, 2002, pp. 219–224.

[8] P. Paulin and J. Knight, "Scheduling and Binding Algo rithms for High-Level Synthesis," 26th Conference on Design Automation, 25-29 June 1989, pp. 1-6. http://dx.doi.org/10.1145/74382.74383

[9] V. Suhendra, C. Raghavan, and T. Mitra, "Integrated scratchpad memory optimization and task scheduling for MPSOC architecture," in *Proc. Int. Conf. CASES*, 2006, pp. 401–410.

[10] Wu, H. Yu, S. Jin, L. Kuo-Chi and G. Schiavone, "An Incremental Genetic

Algorithm Approach to Multiproc-essor Scheduling," IEEE Transactions on Parallel and Distributed Systems, Vol. 15, No. 9, 2004, pp. 824-834. http://dx.doi.org/10.1109/TPDS.2004.38

[11] Hassan Salamy, *Member, IEEE,* and J. Ramanujam, *Member, IEEE* "An Effective Solution to Task Scheduling and Memory Partitioning for Multiprocessor System-on-Chip," IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 31, NO. 5, MAY 2012

[12] L. Goh, B. Veeravalli and S. Viswanathan, "Design of Fast and Efficient Energy-Aware Gradient-Based Sched-uling Algorithms Heterogeneous Embedded Multiproc-essor Systems," IEEE Transactions on Parallel and Dis-tributed Systems, Vol. 20, No. 1, 2009, pp. 1-12.

[13] Y. Yu and V. Prasanna, "Energy-Balanced Task Alloca-tion for Collaborative Processing in Wireless Sensor Net- works," Mobile Networks and Applications, Vol. 10, 2005, pp. 115-131. http://dx.doi.org/10.1023/B:MONE.0000048550.31717.c5

[14] A. Andrei, M. Schmitz, P. Eles, Z. Peng, B. M. Al- Hashimi, "Overhead-Conscious Voltage Selection for Dy- namic and Leakage Energy Reduction of Time-Cons-trained Systems," IEE Proceedings on Computers and Digital Techniques, Vol. 152, No. 1, 2005, pp. 28-38. http://dx.doi.org/10.1049/ip-cdt:20045055

[15] T. A. AlEnawy and H. Aydin. Energy-Aware Task Allocation for Rate Monotonic Scheduling. In *Proceedings of the 11th Real Time on Embedded Technology and Applications Symposium*, pages 213–223, San Francisco, CA, USA, 7-10 March 2005. IEEE Computer Society.

[16] Y. Wei, C. Yang, T. Kuo, and S. Hung. Energy-Efficient Real-Time Scheduling of Multimedia Tasks on Multi-Core Processors. In *Proceedings of the 25th Sympo-sium on Applied Computing*, pages 258–262, Sierre, Switzerland, 22-26 March 2010. ACM.

Jenitha A received BE in Electronics and communication Engineering in 1998 from Bangalore University and ME in Electronics and Communication Engineering in 2005 from Bangalore University.
She is currently working as an Assoc Prof in Dept. of Electronics and communication Engg., East Point College of Engineering and Technology, Bangalore.
Her research interest includes Embedded Systems, MPSoC architecture, Biomedical Applications.

Dr. Elumalai R received his BE in Electrical and Electronics Engg from Bangalore University, ME in Power Electronics from Bangalore University and PhD from Vinayaka mission University,
He is currently working as professor and Head of the Department of Instrumentation Department, MSRIT, Bangalore.
His research interest includes Embedded System design, Cryptography, MPSoC architecture, Biomedical Applications.