

Efficient Energy Aware Task Scheduling In Heterogeneous Network On Chip

Sathis Kumar.K¹, Dr. Paramasivam.K², Janani T³

¹Assistant Professor, Department of Computer Science and Engineering, Bannari Amman Institute of Technology

²Professor, Department of Electrical and Electronics Engineering, Kumaraguru College of Technology, Coimbatore

³Assistant Professor, Department of Information Technology, Bannari Amman Institute of Technology

Email: ¹Erode, sathiscse7@gmail.com, ²kpsivam5@gmail.com, ³Erode, janani8904@gmail.com

Abstract: *High performance, Quality of Service and energy efficiency are the main factors in real-time embedded applications. Network-on-Chip (NoC) based Multiprocessor System-on-Chips (MPSoC) computing platform performs better than superscalar uniprocessor architectures. The operating cost, lifetime and readability of the embedded system are based on the energy consumption. Reduction of energy consumption can be accomplished by efficiently scheduling the tasks with parameters such as priority and constraints. Ordering of tasks, assignment of voltage for task mapping need to be considered in Energy aware scheduling. Contention between the communications plays a major role in Energy consumption. Ordering of tasks, tasks mapping, assignment of voltage and contention should be combined for energy optimization. Energy consumption can be reduced by mapping each task to processor and minimizing the energy consumed in communication by allocating distinct levels of voltage to the Links in NoC. Scaling of frequency, dynamic voltage and power management can be used for optimization of energy consumption. The proposed approach maps contention-aware tasks and manages the energy by ordering the task and assigning the priority to the tasks.*

Keyword: *Network-on-Chip, Scheduling, Multiprocessor System-on-Chips, Task*

1. INTRODUCTION

Chips in NoC architecture consisted with general-purpose processor cores, memory subsystem and on-chip components. Efficient and fast routing service is provided by on-chip routers. A small single sized chip is integrated with input output peripherals, functionalities of multi-core and power management circuits in System-on-Chip (SoC). Power consumption is less and computational capability is higher in SoC [3][4]. Demands for higher consumption of power and throughput leads to Multi-processor SoC [9].

Multi-processor SoC meets the high throughput and power consumption. MPSoC can be classified based on the nature of utilized processors as homogeneous or heterogeneous. Different kind of processor or cores is employed in heterogeneous MPSoC. This heterogeneous system improves performance and energy efficiency by implementing the

dissimilar processor type. It incorporates specialized processing capabilities to handle specialized tasks. Thus, MPSoC drives embedded systems for real-time applications.

Performance and power usage are extremely important aspects in MPSoC. Performance of an application can be defined in terms of control and dependencies in data. Dependencies in control make a task not to start executing till the previous task in execution is completed, whereas data dependencies indicate the communication between the tasks [2].

Scheduling the tasks for computation and transactions in communication is an area of concern. This scheduling includes in choosing the tasks for assignment in the core and transactions onto computation and communications tasks, and finalizing the order in which the tasks are executed on the resources shared in the environment. Energy consumption will be different when assigning same task to different processing elements. Volume of communication among the inter task and the path will vary significantly in assignments for tasks. This leads to dissimilar consumption values for the consumption of energy. Hence, scheduling of tasks and transactions has important impact in the consumption of energy [1].

Various heuristics based on mathematical foundation is proposed to accomplish the mapping of tasks and scheduling the MPSoC architecture. Among the available search-based heuristic algorithms, Genetic Algorithm (GA) is mainly used for the scheduling of tasks. This algorithm is based on exploring the available search space and utilization of the information available in the search space.

In this paper, mapping of tasks and scheduling these tasks on heterogeneous MPSoC architecture are investigated. Mapping of tasks, scheduling tasks and voltage scaling is integrated. Energy consumption can be minimized by making each node to use the available slack between tasks and communication [6].

The remaining part of the paper is organised as follows. Inside the segment II we address relevant mapping of tasks and scheduling function. Section III discusses a heterogeneous application model and Platform for NoC-MPSoC. The experimental section IV demonstrates Results and, finally, this paper is concluded in Section V.

2. RELATED WORK

The set of tasks can be allocated to processors by specific commitment as reduced execution time and consumption of energy. The performance and reliability of the embedded system depends on the approach followed in mapping and scheduling the tasks [7]. Scheduling of tasks in MPSoC is a Non-deterministic Polynomial-time (NP) hard problem. Scheduling is done with various search based and heuristics algorithms. Task scheduling can be static, before the embedded system executes and dynamic scheduling, during the run-time

Stulova et al. proposed that by using Genetic Algorithm to optimise the performance of the heterogeneous MPSoC platform for multiple applications executing in a core by maximizing parallelism in data tasks [8]. Haider Ali et al proposed Integrated Task Mapping by concentrating the energy consumption, Scheduling the tasks and Voltage Scaling algorithm. Ordering of tasks and its timely communication is performed using this algorithm [1].

Tosun[9] mapped independent periodic tasks to minimise computational energy consumption on heterogeneous MPSoC architectures. An Earliest Deadline First algorithm based on heuristic is deployed for mapping of tasks and based on tasks; the voltage levels are optimized and assigned. The used task replication has an adverse impact in the MPSoC platform's memory use[18].

Energy optimization in a Multi Integer Linear Programming using NoC based homogeneous MPSoC architecture, Directed Acyclic Graph (DAG) is used for representing the dependent

tasks in real-time is proposed by Chen et al. [10] Energy consumption is optimized by scheduling the tasks in a non-pre-emptive manner and each task are assigned with a discrete voltage level[19].

In a memory-shared homogenous MPSoC, priority is assigned for the tasks and energy-aware scheduling of tasks with its corresponding priority was investigated by Tariq and Wu [11]. First, the tasks are mapped to processors and the scaling of voltage is performed. Algorithm based on Natural Language Processing is used to assign optimal level of voltage to each task. Chou and Marculescu [12] use DAG to optimize the contention in NoC architecture for MPSoC. The mapping of tasks is performed to develop the ability of control congestions in the network there by providing best-effort communication. The drop of energy in computation by the processors in not considered in this work.

In DAG, the dependencies in intra-data are distorted into inter-data. Optimal task scheduling is carried out by the implementation of Heuristic Memory-Aware Task Scheduling. This methodology of minimizing the overhead in inter-process communication and improvement in usage of memory and throughput with the conventional bus-based interconnect homogenous MPSoC platform is proposed by Wang et al. [13].

In a heterogeneous NoC MPSoC architecture, Sing et al.[14] investigates contention in the network, energy efficient based on replication mixed integer programming. The tasks are duplicates and allotted to the processor to avoid the congestion in traffic and to minimize the energy consumption in inter-processor communication. This energy model ignores the effect of overall energy consumed in duplication of tasks.

In homogenous MPSoC architecture, the energy consumed in computation of tasks and communication is optimized for real-time DAG task based bus communication [15]. Mapping of tasks and scheduling is devised as ILP. Inter-process communication overhead is reduced to minimize the schedule length. In two phase energy optimization algorithm, re-timing is used to transform DAG into task independent model. During the second phase, GeneS – an algorithm based on genetic is used to minimize the computation and energy consumption in inter-process communication. NoC based on the architecture of heterogeneous MPSoC with distributed memory is considered in this work. DAG represents real-time dependent task. The processing and energy consumption is reduced by considering the contention.

3. SYSTEM MODEL

DAG is used to model the dependent tasks in a real-time application as shown in Figure 1. The representation is made as $DAG(V, E, W)$. V is the set of nodes in the graph which symbolize the dependent task. Each task on execution requires processor clock cycles with a time to execute. E signifies the set of edges which connects the nodes i.e., tasks. Data dependency relation between the tasks is represented by the edges. The data volume in bits transferred from a node to another is represented by W which is the edge weight.

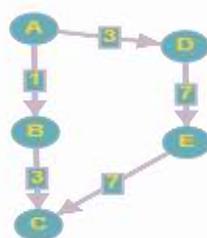


Figure 1 Directed Acyclic Graph

Let k be the number of tiles in the heterogeneous NoC based MPSoC. Each of it includes processor, interface for network and a local memory. Let $P = \{p_1, p_2, \dots, p_n\}$ heterogeneous processors be interconnected via 2D mesh NoC. P , the processor will have their local memory. This is illustrated in Figure 2. Processor may have unique instruction set architecture [17]. Dynamic Voltage and Scaling of frequency are enabled in the processor and its communication link. The set of voltage and frequency levels are assigned for processor and communication links.

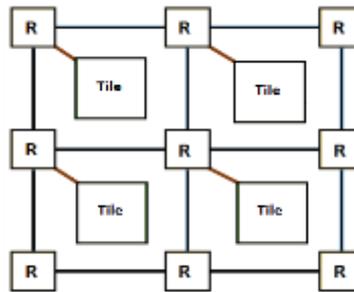


Figure 2 NoC-MPSoC architecture

The genetically operators namely reproduction, crossover and mutation are used to manipulate the population. Initial population is randomly generated that includes chromosomes. Fitness function is used to evaluate the integer chromosomes. A parent of next generation is produced by selecting the best fittest chromosomes. Crossover and mutation operations are applied to produce children chromosome from the random selected parent chromosome. These above steps are continued for a set of iterations or till exit criteria are met.

Chromosome: Includes a series of genes and each one represents the task. AN integer value is assigned which symbolizes IP core.

Crossover: Random selection is adopted for choosing parent chromosomes. A random crossover point is selected for any two selected chromosomes. Each core will be assigned to only one task.

Mutation: A single child chromosome is randomly chosen from the population. Random numbers are generated and allocated for each gene. It is compared with the mutation probability and replaced with a random gene in the same chromosome.

Fitness Function: The required optimization of task mapping by minimizing the energy consumption and contention is represented with the fitness function.

Consumption of energy by the task that executes on a processor at a voltage is computed by multiplying the power dissipated by a processor with the task execution time. [16]

Algorithm

- 1 **while** 1 **do**
- 2 **for** For every chromosome in the population **do**
- 3 Tasks are mapped and ordered using Earliest Latest Finish Time First (ELFTF) strategy [1]
- 4 Calculate energy consumption using Delay-based Max Slow Down Policy DVFS
- 5 Fitness value is computed

```

6      if it meets the stopping criteria then
7          Choose the chromosome with highest value and perform the mapping
8          build the schedule for the tasks to run and produce the schedule
9          break
10     if stagnation occurs then
11         choose chromosome with smallest value of fitness
12         perform crossover to generate new chromosome
13         repeat 2 to 5 for new set of chromosome
14     The chromosomes are sorted in a non-decreasing value of fitness
15     Remove chromosome with smaller value of fitness
16     Make a random selection of chromosome
17     For each random selected chromosome do
18         For each element of chromosome do
19             Perform the remapping of takes that are represented by chromosome
20     Perform a merge of chromosome for next generation
    
```

4. EXPERIMENTAL SETUP AND RESULTS

DAG is constructed and the tasks are mapped in MPSoC architecture. Task nodes and communication nodes are assigned with discrete voltage. This will reduce the energy consumption. GA is applied in performing this task. Mapping of tasks and voltage scaling are carried out separately. Energy is consumed by the processor during its idle period and to make it ON even if no tasks are allocated to run on it.

Matlab R2016a is used to build the simulation environment. Hardware platforms such as Intel(R) Xeon(R), i5 processor with 3.5 GHz clock frequency, memory of 16 GB and cache size of 10 MB is used to carry out the experiment. The HMPSoC architecture which is based on NoC is deployed. The processors are enabled with DVFS. In a heterogeneous platform, each tasks execution time varies with its processor.

An embedded application which performs tasks such as JPEG compression and decompression, Fast Fourier Transform, pattern recognition with real-time streaming from embedded systems Synthesis is considered to find the energy and contention in scheduling approach. Table 1 shows the comparison results

Real-world Benchmarks	Energy-efficient Scheduling	Efficient energy aware task scheduling
Consumer	0.32	0.30
Auto-industry	0.62	0.35
MP3 Decoder	1.7	1.01
Pattern Recognition	4.8	3.12

Table 1 Results comparison

5. CONCLUSION

Reducing the energy consumption in MPSoC improves the embedded system's lifetime. This is achieved by efficiently scheduling the tasks allocated to the processor. The proposed task scheduling significantly reduces the energy consumption. Dynamic Power Management can be adopted in future by which the processors can be switched into power saving low-power mode when it's possible.

6. REFERENCES

- [1] Haider Ali, Xiaojun Zhai, Lu Liu and Umair Ullah Tariq, "Energy Efficient Task Mapping & Scheduling on Heterogeneous NoC-MPSoCs in IoT based Smart City", 2018 IEEE 20th International Conference on High Performance Computing and Communications; pp. 1305-1313, 2018
- [2] Jingcao Hu and R. Marculescu, "Energy-aware communication and task scheduling for network-on-chip architectures under real-time constraints," Proceedings Design, Automation and Test in Europe Conference and Exhibition, Paris, France, 2004
- [3] S. Mishra, N. K. Singh, and V. Rousseau, System on Chip Interfaces for Low Power Design. Morgan Kaufmann, 2015.
- [4] X. Zhai, A. A. S. Ali, A. Amira, and F. Bensaali, "Mlp neural network based gas classification system on zynq soc," IEEE Access, vol. 4, pp. 8138–8146, 2016.
- [5] H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran, "Energy-efficient adaptive pipelined mpsocs for multimedia applications," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 33, no. 5, pp. 663–676, 2014
- [6] U. U. Tariq, H. Wu, and S. Abd Ishak, "Energy-aware scheduling of conditional task graphs on noc-based mpsocs," in Proceedings of the 51st Hawaii International Conference on System Sciences, 2018.
- [7] E. L. de Souza Carvalho, N. L. V. Calazans, and F. G. Moraes, "Dynamic task mapping for mpsocs," IEEE Design & Test of Computers, vol. 27, no. 5, pp. 26–35, 2010.
- [8] A. Stulova, R. Leupers, and G. Ascheid, "Throughput driven transformations of synchronous data flows for mapping to heterogeneous mpsocs," in 2012 International Conference on Embedded Computer Systems (SAMOS), IEEE, 2012, pp. 144–151.
- [9] S. Tosun, "Energy- and reliability-aware task scheduling onto heterogeneous MPSoC architectures," J. Supercomput., vol. 62, no. 1, pp. 265_289, 2012.
- [10] G. Chen, K. Huang, and A. Knoll, "Energy optimization for real-time multiprocessor system-on-chip with optimal DVFS and DPM combination," ACM Trans. Embedded Comput. Syst., vol. 13, no. 3s, p. 111, 2014.
- [11] U. U. Tariq and H. Wu, "Energy-aware scheduling of conditional task graphs with deadlines on MPSoCs," in Proc. IEEE 34th Int. Conf. Comput. Design (ICCD), Oct. 2016, pp. 265_272.
- [12] C.-L. Chou and R. Marculescu, "Contention-aware application mapping for network-on-chip communication architectures," in Proc. IEEE Int. Conf. Comput. Design (ICCD), Oct. 2008, pp. 164_169.
- [13] Y. Wang, Z. Shao, H. C. B. Chan, D. Liu, and Y. Guan, "Memory-aware task scheduling with communication overhead minimization for streaming applications on bus-based multiprocessor system-on-chips," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 7, pp. 1797_1807, Jul. 2014.
- [14] J. Singh, S. Betha, B. Mangipudi, and N. Auluck, "Contention aware energy efficient scheduling on heterogeneous multiprocessors," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1251_1264, May 2015.
- [15] Y. Wang, H. Liu, D. Liu, Z. Qin, Z. Shao, and E. H.-M. Sha, "Overhead aware energy optimization for real-time streaming applications on multiprocessor system-on-chip," ACM Trans. Des. Autom. Electron. Syst., vol. 16, no. 2, p. 14, 2011.

- [16] U. U. Tariq, H. Wu, and S. Abd Ishak, “Energy-aware scheduling of conditional task graphs on noc-based mpsocs,” in Proceedings of the 51st Hawaii International Conference on System Sciences, 2018.
- [17] Haider Ali, Umair Ullah Tariq, Yongjun Zheng Xiaojun Zhai And Lu Liu, “Contention & Energy-Aware Real-Time Task Mapping on NoC Based Heterogeneous MPSoCs”, IEEE, 2018
- [18] Sujatha, K DS Punithavathani, Analysis and Epigrammatic Study of Various Tone Mapping Operators for High Dynamic Range Images, Indian Journal of Science and Technology, 8 (36) A-II.2015(Scopus)
- [19] Venkatachalam K, Karthikeyan NK (2018) A framework for constraint based web service discovery with natural language user queries. J Adv Res Dyn Control Syst, Elsevier Publication 05-Special Issue, 1310–1316