International Conference on New Horizons in Science Engineering Technology (NHSET-2018)
International Journal of Scientific Research in Computer Science, Engineering and Information Technolo
© 2018IJSRCSEIT | Volume 4 | Issue 5 | ISSN :2456-3307

Genetic Algorithm Implementation In MPSoC

[1]Jenitha A, [2]Dr. R. Elumalai, [3]Dr. S. Sujitha
[1] Research Scholar, Vishvesvaraya Technological University, Belagavi, India
[2,3] Department of Electrical and Electronics Engineering, New Horizon College of Engineering Bangalore, India

ABSTRACT

Multiprocessor designs are the best substitute for single-core designs, but the new architecture has any kind of architectural problems associated with it. The main problems are the tools and techniques needed to maximize multiprocessors and develop new techniques to produce powerful architecture associated. To overcome the above problems, one of the best techniques is to combine the techniques of planning and management of memory in computer systems. Here, we introduce a genetic algorithm to do the same. This algorithm finds the best solution by performing three operations, namely, mutation, crossover and the fitness function for which the planning of activities on multiple processors is done with the use of adequate memory. By implementing this algorithm in different tasks, the total delay is reduced and an increase is also obtained in terms of performance. The implementation was made with Xilinx.

Keywords : Multiprocessor, Crossover, Mutation, Fitness Function.

I. INTRODUCTION

In the history of personal computers with high performance for existing embedded systems. The method of increasing the speed frequency systems and single core processor is not a viable solution because of the high power requirements. Therefore, the current MPSoC systems implemented to increase speed. In embedded applications, if a multiprocessor system used integrated circuit, there are several advantages, for example, the power to run complex applications is significantly reduced. Since the programmable clock SoC processors can be programmed to allow the design later watertight compartments, longer life compared to SoC systems, MPSoC use specialized software controlled is known as memory scraper memories (SPM). It provides the running time with precision. SPM is the array of data together with the decoder memory. And power limits, architectures with multiple processors on a single chip have become a more attractive solution for high performance in the calculation. For real-time embedded applications, the predictability of the execution time is a critical issue. Multiprocessors consist of heterogeneous processing elements, NOC (Network-on-chip), memory, communication platforms for input and output. If an integrated application is programmed correctly in MPSoC, we are able to optimize the appropriate use of system resources. Processing elements and memory chips refer to system resources in this document.

The embedded application can be divided into several tasks. The activity can have a different processing time depending on the type of processor performing that particular task and the amount of memory allocated to it. The access to the data elements to be part of a task from the memory is off-chip

Comparatively slower than accessing memory on the chip. For this both programming activities such as partitioning pattern memory must be always integrated, so that an efficient algorithm for this is done by the genetic algorithm. MPSoC is an on-chip system with multiple sets of instructions processors (CPUs). These are typically heterogeneous multiprocessor, consisting of different types of memory elements of heterogeneous processing the whole machine and the interconnection between the processing elements and memory and require a large memory. They are regularly structured and in its most preferred form because it is easy to program. The specialization and the balance programming of multiple processor system. Programmable processors can be programmed to allow SoC after being manufactured. MPSoC sometimes called as platforms, because in it the implementation of a given type of system is allowed. The advantages of the integrated circuit programming are reusable, the reduction of the costs of the product

II. LITERATURE SURVEY

The main objective is to reduce the execution time of the cycle and the preservation of power. The literature review is provided based on different surveys for partitioning algorithm and activity planning methods in order to meet our main goal

V.Suhendra, C.Raghavan and T.Mitra proposed a linear programming model Entire making interaction between programming and system partition. The ILP is a program of integration, and the assignment of tasks to processors available in the system. In this application, the tasks are taken and are at the program. They are then pipelined for simultaneous processing.

M.Kandemir, J.Ramanujam and A. Choudhury presented the compiler-based technique. This method is used to improve the data access to the multiprocessor system to increase the usefulness of resources and reduce energy consumption. Here you can access the data using parallel arrays. It focuses on two main issues, namely: access to the requested memory off-chip and reduce the backlog of energy. This technique based compiler is built with the help of matrices and relationship-oriented software residing on the internal system.

The crosstalk that occurs in the cache problem can be solved with the help of this partitioning strategy. These characteristics are related to the program codes that are considered based partitions. The characteristics of this strategy are as follows:

- Variables and constants.
- Array size
- Cycle time.
- Computational cost.
- Loop conflicts.

The above algorithm uses the functionality to assign the data to SPM Budget for the fulfillment of the task processors.

Zhang Lei, Meikang Qiu, Wei-Che Tseng focuses on two main issues are interdependent. These problems are partitioning and programming that are resolved with the help of the decoupled method. Decoupling approach uses two methods to solve the above problems. High frequency of access (HAFF). Predictive partitioning Global View (GVP) variables.

III. THE GENETIC ALGORITHM

The genetic algorithm can be used to solve problems by coding a possible solution to any problem and a path that must be found. We take this bit string as an example of the chromosome

10010101110101001010011101101110111111101

To begin the genetic algorithm they are created random chromosomes or chromosome number of random population. When each decoded will lead to different solutions.

Consider that there are N chromosomes in the initial state of the population. So you need to repeat the following steps until you find a solution

- Each chromosome test to see how good it is to solve the problem in question and therefore assign a fitness score. The result of the fitness score is a measure of how well the chromosome to solve the problem.
- Then select two members of the current population. The probability that these chromosomes are selected result is proportional to the fitness of chromosomes. The roulette wheel is a commonly used method for the selection of the chromosomes.
- Depending on the crosshead speed at a point chosen at random, they are selected the bits of each intersection chosen chromosome.
- After chromosomes and bit flip chosen it depends on the mutation rate.
- Repeat the above steps until you have created a new population of members N.

Crosshead speed:

This simply selected the two chromosomes exchange their tips. For that around 0.7 is a good relationship. In the process of traversal of a random gene and along the length of the chromosomes are exchanged genes after selecting this point.

Given for example two chromosomes

10001001110010010
01010001001000011

Select a bit 'random throughout the length, say at 7-position, and share all bits after that point so that the above becomes:

10001001001000011
01010001110010010

Mutation frequency

There is the possibility that a bit within a chromosome tipping (0 becomes 1, 1 becomes 0). This is usually a very low value for the genes encoded binary, say about 0.001.

Therefore, whenever you are chosen the chromosomes of the population available, the algorithm first checks if the crossover is to be applied or not, and then algorithm iterates along each chromosome mutating the bits if applicable

IV. IMPLEMENTATION

Processor 2: is assigned with a task that computes the Radix – 2 FFT algorithm.

Processor 1: is assigned with three tasks SIPO, Ripple carry Adder and Booth Multiplier.

Delay Element: is used to calculate the number of clock cycles required by the tasks in both the processor for the complete execution of the tasks.

Genetic algorithm : it performs a mathematical calculation to determine the best suited solution by performing mutation, crossover and fitness function to schedule the task with proper utilization of memory. It reduces the delay in producing the output and results in high throughput. For our design the best suited fitness function is 2 clock cycles.

Shared Memory: once the task is completed the

result is stored in shared memory in FIFO manner.

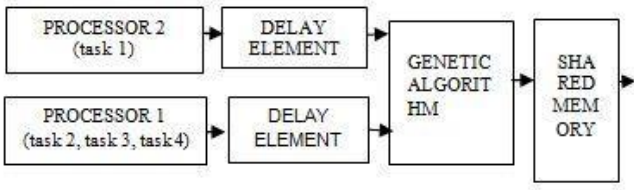

Fig 1 Block diagram of the proposed design.

V. RESULT

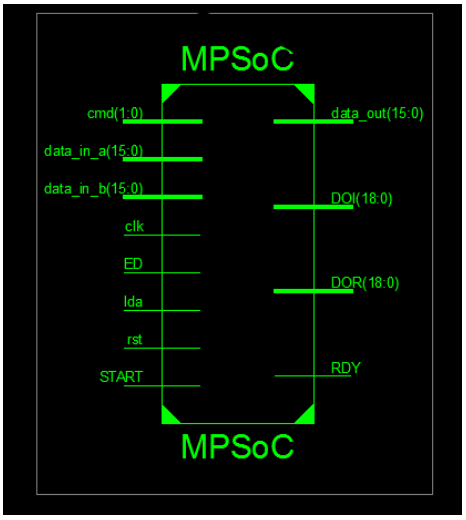

Fig 2 Schematic diagram of MPSoC

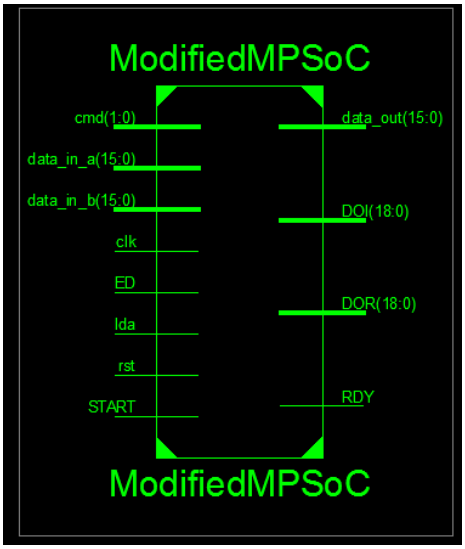

Fig 3 Schematic diagram of Modified MPSoC

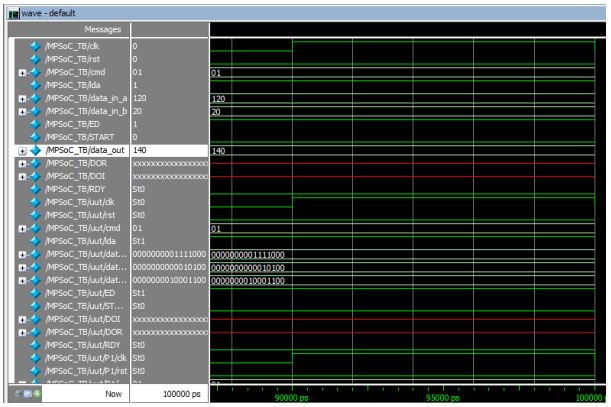

Fig 4 simulation result of MPSoC

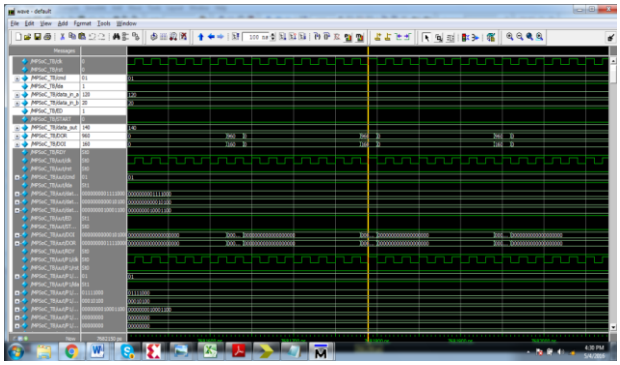

Fig 5 simulation result of modified MPSoC

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	923	960	96%
Number of Slice Flip Flops	981	1920	51%
Number of 4 input LUTs	1369	1920	71%
Number of bonded IOBs	93	66	140%
Number of MULT18X18SIOs	4	4	100%
Number of GCLKs	1	24	4%

Fig 6 Device utilization memory

VI. CONCLUSION

A genetic algorithm method is used to find out the best solution to schedule the task with proper utilization of memory. This implementation leads to reduction in delay to produce the output and results in high throughput. And in our design we have done a comparison study between normal MPSoC and modified MPSoC, even though the normal MPSoC covers less area, the delay is more which is approximately 14 clock cycles to get output whereas modified MPSoC require 2 clock cycles.

VII. REFERENCES

[1]. Hassan salamy, semih aslan,"A genetic algorithm based approach to pipelined memory- ware scheduling on an MPSoC," in Proc. IEEE 2015. (Base paper).

[2]. Poorani.A, Anuradha.B, Dr. C. Vivekanadhan,"An Effectual Elucidation of Task Scheduling and Memory Partitioning for MPSoC," in Proc. International conference on intelligent systems and control (ISCO), 2014.

[3]. L. Benini, D. Bertozzi, A. Guerri, and M. Milano, "Allocation and scheduling for MPSoC via decomposition and no-good generation," in Proc. International Joint conferences on Artificial Intelligence (IJCAI),2005.

[4]. Suhendra, C. Raghavan, and T. Mitra, "Integrated scratchpad memory optimization and task scheduling for MPSoC architecture,"in proc. International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES), 2006.

[5]. Anuradha.B, hemalatha M, vivekanadhan C," task allocation and memory partitioning for MPSOC in embedded systems, "in proc. international journal of engineering science and innovation technology (IJESIT), 2013.

[6]. Kanoun, N. Mastronarde, D. Atienza, and M. V. D. Schaar, "On line energy-efficient task-graph scheduling for multicore platforms," IEEE transactions on Computer Aided Design, vol. 33, no. 8, 2014.

[7]. P.-H. Tseng, P.-C. Hsiu, C.-C. Pan, and T.-W. Kuo, "User-centric energy-efficient scheduling on multi- core mobile devices," in Design Automation Conference.

[8]. S.Sujitha, Vivek, C.Venkatesh, "Fuzzy Logic Based Speed Control of DC Motor drive", International Conference on Emerging Trends in Engineering and Technology, 2015.

[9]. S.Sujitha, "Investigation of Standalone PV Fed Switched Reluctance Motor Drives Using C Dump Converter", Global Journal of Pure and Applied Mathematics, Volume 13, Number 9, pp. 6317-6326, ISSN 0973-1768, 2017

[10]. S.Sujitha, "Exploration of Hybrid Switched Reluctance Motor Drives Using H Bridge Converter", Middle-East Journal of Scientific Research 25 (6): 1298-1302, ISSN 1990-9233, 2017