Fuzzy logic based adaptive MPSoC for balanced energy and throughput

Adeel Ahmad[1], Muhammad Yasir Qadri[2,*], Nadia N. Qadri[3], Jameel Ahmed[1]

[1]Dept. of Electrical Engineering, HITEC University, Taxila Cantt, Pakistan

[2]School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, United Kingdom

[3]COMSATS Institute of Information Technology, Wah Cantt, Pakistan

[*]Corresponding author: yasirqadri@acm.org

Abstract

In this paper a Fuzzy logic based Controller has been proposed to reconfigure multi-processor system on chip (MPSoC) architecture according to the workload requirements. Mamdani and Sugeno Inference Engines are compared and analyzed to enhance the smooth regulation of reconfiguration in design space. The proposed MPSoC platform consists of 16 cores, where each core has private L1 cache and a shared L2 cache. The design space exploration parameters evaluated are CPUs operating frequency, number of cores, throughput and energy consumption. Marssx86 simulator is used to build the target machine and SPLASH-2 benchmarks are executed on this configured architecture. Mamdani Inference Engine for the said implementation has shown up to 14%-23% decrease in Energy-Delay^{2}Product (ED^{2}P) for various benchmark applications. Hence the performance of Mamdani inference engine was found to be better than the Sugeno one for this particular implementation.

Keywords: Design space exploration; Fuzzy logic; MPSoC; reconfigurable architecture.

1. Introduction

MPSoC performance can be improved by reconfiguring its architecture. Design Space Exploration (DSE) is a process of finding an optimum objective or feasible architectural design to gain in performance. DSE is composed of two parts; problem space and solution space (Cant´o, 2013). Problem space comprises of available resources of a system, which are reconfigurable (e.g. number of cores, clock frequency, cache size and associativity, chip area, issue width, interconnects schemes and peripherals etc.), while Solution space is an objective need to be fulfilled (e.g. power, heat density, energy, throughput, performance and quality of service). Problem space is usually reconfigured using search algorithms.

Various search algorithms are used for exploration in design space. DSE searching algorithms can be divided into three main classes i.e. exhaustive search, random search, and heuristic search. In Exhaustive Search, all possible solutions of problem domain are explored to find the feasible solution, which satisfies the desired statement of problem. This method is quite useful when the design space is small, otherwise with large design space the computation time is too long. DSE related examples for exhaustive search are discussed in Baghdadi (2000); Blythe & Walker (2000); Lahiri (2001). On the other hand, in Random Search, the problem is explored on the basis of randomness to reach the possible solution. Tabu search (Kreutz, 2005), simulated annealing (Gajski, 1998; Orsila, 2006) and Monte Carlo approximation (Bruni, 2001) belongs to this class of search algorithms. However, in heuristic search, a guided search through knowledge of search space is used to reach the possible solution. Genetic algorithms (Kang & Kumar, 2008), Hybrid Ant Colony Optimization (Sedighpour 2014), Markov Decision Process (MDP) (Beltrame, 2010), Evolutionary Algorithms (Erbas, 2006) and Fuzzy Based Rule(s) (Qadri *et al.*, 2014) are few techniques of heuristic search. In this paper, fuzzy logic, which is based on heuristic search is implemented in the DSE process to achieve the optimum solution. This paper is a continuation of a novel work proposed by Qadri *et al.* (2014).

A fuzzy logic based controller had been proposed to reconfigure architecture according to the workload requirements and a balance had been made between throughput and energy consumption. The reconfigurable architecture was divided into two levels i.e., core and SoC. At core level, solution space contained L1 miss ratio, energy consumption and throughput, while problem space had L1 cache associativity, L1 cache size and clock frequency. At SoC level, solution space contained L2 miss ratio, energy consumption and throughput, while problem space has L2 cache associativity, L2 cache size and number of cores. The controller was devised on Mamdani inference engine with triangular membership function, while this paper shows comparison of Mamdani and Sugeno inference engines using Gaussian membership function.

This paper has been organized as follows. Section 2, provides details about previous methodologies and strategies opted in the design space exploration process. Section 3, elaborates the proposed fuzzy logic based design space exploration engine. Section 4 reports about, the proposed framework setup and simulation environment. In Section 5, results are compared and discussed followed by Section 6 that concludes the paper and discusses the future work.

2. Related work

DSE attracted many multicore architecture designers to devise a system which achieve better performance with low energy consumption. Different methodologies have been explored to attain optimum solution. In this section, initially traditional techniques have been discussed, followed by Qadri *et al.* (2014) the proposed technique,which has been implemented in this paperis eloborated.

Baghdadi *et al.* (2000) shows the performance estimation and design space exploration problems in context of both hardware and software. A system's architecture is explored before the start of any process. Neither architecture configuration can be selected, nor additional decision can be made to change workload during processing. In performance estimation, timing accuracy is achieved using another block model; back-annotation and time-annotation is used to determine the pre performance of all the architectures. Because of its exhaustive search nature, the cycle-based simulation takes more than 30 hours and at system-level it takes only 10 seconds. Monchiero *et al.*(2006) also used exhaustive search technique to explore the design space exploration between number of cores, processor issue width and L2 cache size on the basis of throughput, power consumption and thermal effects. In this work, more evaluations are made to select the minimum size of chip floor plans and conclude the temperature effects on these chips.

Palermo *et al.* (2003) select design space exploration configuration parameters on Pareto-optimal curves by using method of random search algorithm. A Pareto-optimal point in design space is an objective or configuration for which, there is no other inferior or equal objective. But this simulation takes 3 days in order to find a solution. Orsilia (2006) represented the modified form of simulated annealing, which can reduce the exploration time. Optimization is achieved by adopting selective schedule in annealing and fixed transition probabilities (Phillips, 2008). Another paper by Kreutz *et al.* (2005) used a Tabu search model to explore the design space of Network-on-Chip (NoC) architectures. The architecture of routers' are selected using the Pareto-curves related to energy and latency models. Trade off among homogeneous and heterogeneous NoC architectures is elaborated using latency and energy consumption.

If the design exploration techniques based on Monte Carlo, Simulated annealing and Tabu search are efficient, then full search methods are robust. But these methodologies require statistical data in off-line mode, and then predicted design model is launched. These strategies propagate the latency problems and performance hindrances. So, design exploration researchers proposed heuristic search schemes and models in contrast to exhaustive and random search techniques.

Beltrame *et al.* (2010) have presented a model for design space exploration, which needs a minimum number of simulations required to point-out a Pareto-curve having a matrix of energy and delay. For non-optimal solutions, an estimation model (moment vectors) is used to further consistently reduce the number of simulations. However, this technique needs the availability of an estimation methodology, which is mostly not present in arbitrary Chip multiprocessor (CMP) architectures (Ishebabi, 2010).

Kang & Kumar (2008) proposed a framework called Magellan which isused to accelerate the design space exploration and optimization. Featured search algorithms (like exhaustive, random, heuristic, genetic, evolutionary etc.) are compared against the machine learning based search algorithms. Simulations are executed on single-objective design perspective and by fixing many parameters, searched space is reduced. This methodology lacks in criteria formulti-objective design space exploration (Calborean, 2011). Erbas *et al.*(2006) investigates the trade off strategies between the mappings of matrices such as architecture cost, processing time and power consumption using multi-objective evolutionary algorithm (MOEAs) schemes. This technique generates set of solutions accurately and in minimum amount of time, however for design space exploration we need a unique solution. There are two major limitations in MOEAs, one they are not good if the solution space for search is small and second after reaching its objective solution it generates the already known solutions (Lukasiewycz, 2008). Mishra *et al.* (2014) have presented Adaptive multi-objective particle swarm optimization based design space exploration in architectural synthesis for application specific processor design.The Qadri *et al.* (2014) paper presents that the design space can be reconfigured in requirements of workload for optimum balance by using matrices of throughput and energy. Fuzzy logic and domain knowledge technique is employed in order to remove the discrepancies of above mentioned approaches in the field of design space exploration (i.e. minimize the number of simulations, target can be configured online, etc).

3. Fuzzy logic based design space exploration engine

Fuzzy logic is a technique to overcome the unknown non-linear real-time constraints of the system. The solution for problem space is devised in more pervasive analytical domain, rather than solving it in abstract domain. Fuzzy based knowledge and defuzzification keeps the output of problem space (DSE) on track for unknown occurring phenomena of a solution space (throughput, energy consumption, etc.), which helps to improve the overall efficiency of nonlinear problem.The presented model of MPSoC architecture is a closed loop system which can be reconfigured using Fuzzy logic technique as shown in Figure 1.

As we know that the ONLINE reconfiguration of the system architecture is not tweakable for every trace of solution space, a approach has been employed to systematically change the parameters of problem space. For this, the block snoops the number of iterations and mentioned output parameters of Figure 1 for the designs of architecture and predicts the threshold for the designed space. It regulates the criteria for designed configuration, under which it should continue to retain the architecture of a compiled configuration.

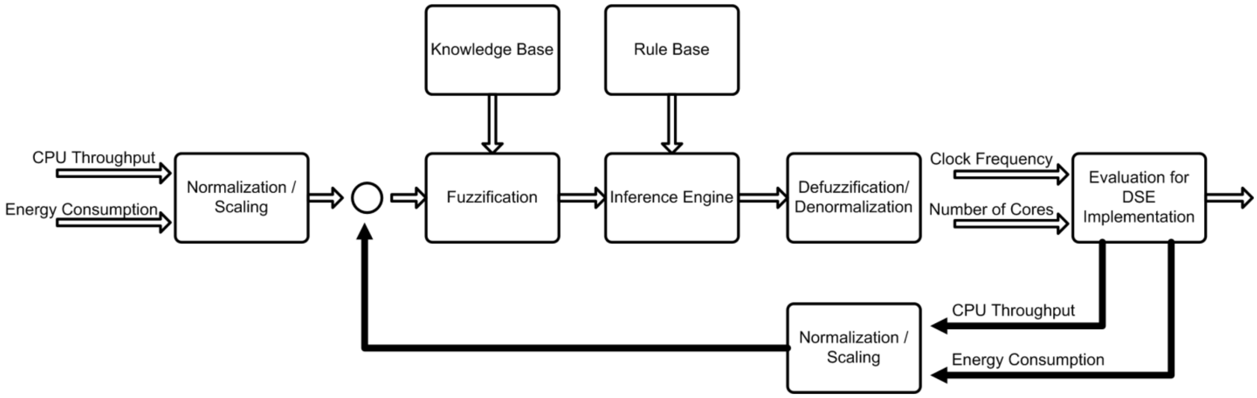


Fig. 1. Block diagram of fuzzy logic based design space exploration engine

4. Implementation steps

The overall implementation of a technique and design configuration of system involved the following steps:

4.1. Design space input variables

In MPSoC architecture, a number of configurable points (parameters) are identified and are controlled by fuzzy logic based design space exploration engine (FLDSEE) i.e. Number of cores and CPU Frequency. Certain other parameters such as CPU throughput and energy consumption are also identified, which have great impact on the input parameters. For these parameters the design space architecture has been explored, evaluated and configured. Mandal et al. presents suitable comparison among fuzzy membership functions (Mandal et al., 2012), and a gaussian membership function (Hameed, 2011; Et, 2007) is selected.

4.2. Mapping variables onto fuzzy membership functions

Input and Output variables are fuzzified and defuzzified by using membership functions. Each variable is further subcategorized into subsets of fuzzy input or output set, and assigned a symbol L, M and H which represents low, high and medium respectively. For example in Figure 2, throughput can be between 0 −100%. So the membership function µL is set from 0 −27%, membership function µM is set from 20 − 85% and membership function µH is set from 68 − 100%. The same methodology is adopted for other input and output variables as shown in Figure 2 and Figure 3.

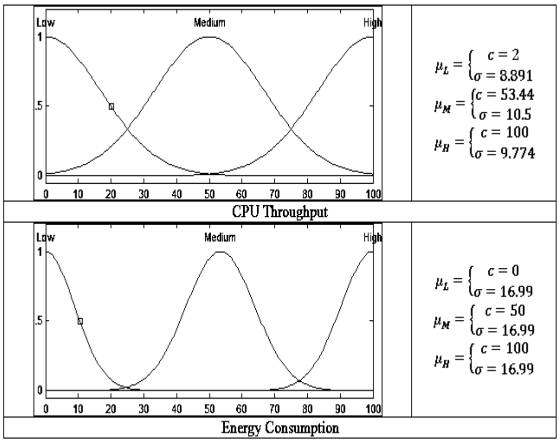


Fig. 2. Input Fuzzy membership function

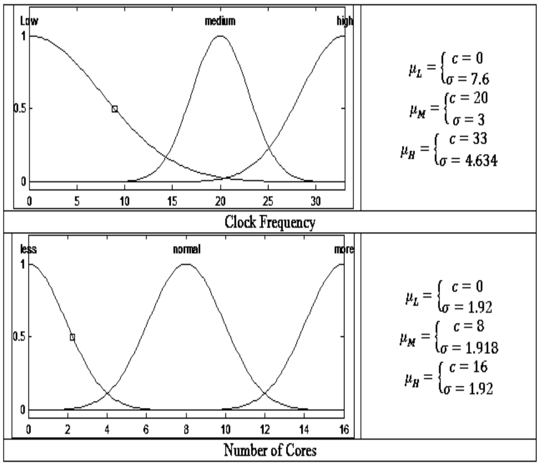


Fig. 3. Output fuzzy membership function

4.3. Knowledge based fuzzy rules

Input and output parameters of MPSoC are related to using the knowledge based fuzzy rules and are shown in Table 1. These rules are considered to balance the optimum relationship between CPU throughput and Energy consumption. Under these rules efficient selection of clock frequency and number of processors are regulated.

4.4. Fuzzy inference engine and defuzzification

Both Mamdani and Takagi-Sugeno-Kang (TSK) or Sugeno inference engines are implemented in design space exploration of MPSoC and their results are comapred in section 7. Mamdani inference engine takes fuzzy inputs, interrelate it with fuzzy rules, depicts the fuzzy outputs, then these fuzzy outputs are aggregated and finally fuzzy value is defuzzified (in this paper center of gravity is used). In Sugeno inference engine, first two parts are same as Mamdani's inference engine. However, the difference is that the Sugeno output membership functions are either constant or linear, but weused constantsand the final output value is calculated using weighted average method.

5. Simulation environment, benchmarks and performance metric

Simulators are installed on Linux Ubuntu 12.04.3 LTS, which inherently supports multicore architecture. Linux advanced configuration and power interface (ACPI) provides a system level control of CPU cores through hot-plugging (online turning on/off) and by using this key feature we can reconfigure an MPSoC.

Table 1. Knowledge based fuzzy rules for MPSoC

Energy Consumption	CPU Throughput	Clock Frequency	Number of cores
L	L	H	L
L	M	M	M
L	H	M	H
M	L	H	L
M	M	M	M
M	H	M	M
H	L	M	L
H	M	L	M
H	H	L	L

The cache miss/hit data and instruction execution cycle information is gathered from MARSSx86 (Patel, 2011). On the other hand, Interconnects network energy consumption is collected from ORION (Kahng, 2012), cache memory access timings and energy consumption generated from CATCTI (Shivakumar & Jouppi, 2001) and also from mathematical models (Qadri & McDonald-Maier, 2010) for cache energy and throughput is accumulated; Intel 486 GX embedded processor energy is accounted from its datasheet. All the applications were sampled for the whole execution cycle of the application and MPSoC is reconfigured under the guided process of FLDSEE. The application is re-executed again for different iterations, in order to investigate the effects of reconfiguration of number of cores, cache size and operating frequency on the energy consumption and throughput of the MPSoC.

Different types of simulators are being used to; model different configurations of MPSoC, acquire the desired data and testify research objectives. These tools are *QEMU*, *MARSSx86* (primarily used for simulation) and *CACTI*.

Stanford parallel applications for shared memory 2 (SPLASH2) is used as a benchmark. It contains 8 applications and 4 kernels benchmarks suite. We have used 4 applications (BARNES, FMM, OCEAN and WATER-SPATIAL) and 1 kernel (LU) benchmarks (Woo, 1995).

Commonly used metric for energy in processing is Joules/instruction or its reciprocal SPEC/W (Gonzalez & Horowitz, 1996). The metric for energy-delay product used is Joules/SPEC or its reciprocal SPEC2/W. In order to compare the given processors, power is not a good metric, because it varies directly with the clock frequency. By decreasing the clock frequency, power dissipation also decreases and cores performance also suffers. Energy is another candidate for metric, measured in Joules/Ins or its reciprocal SPEC/W (Gonzalez & Horowitz, 1996). The energy per instruction can be decreased by decreasing the capacitance or voltage, or by using smaller transistors. With these changes, delay of the path also increases, so one would expect low performance from the lowest energy processor. As both priorities are needed (power and performance) to be correlated relatively, Energy-Delay^{2}Product (ED2P) is a better option in comparison of inference engines performance.

6. Results and comparison

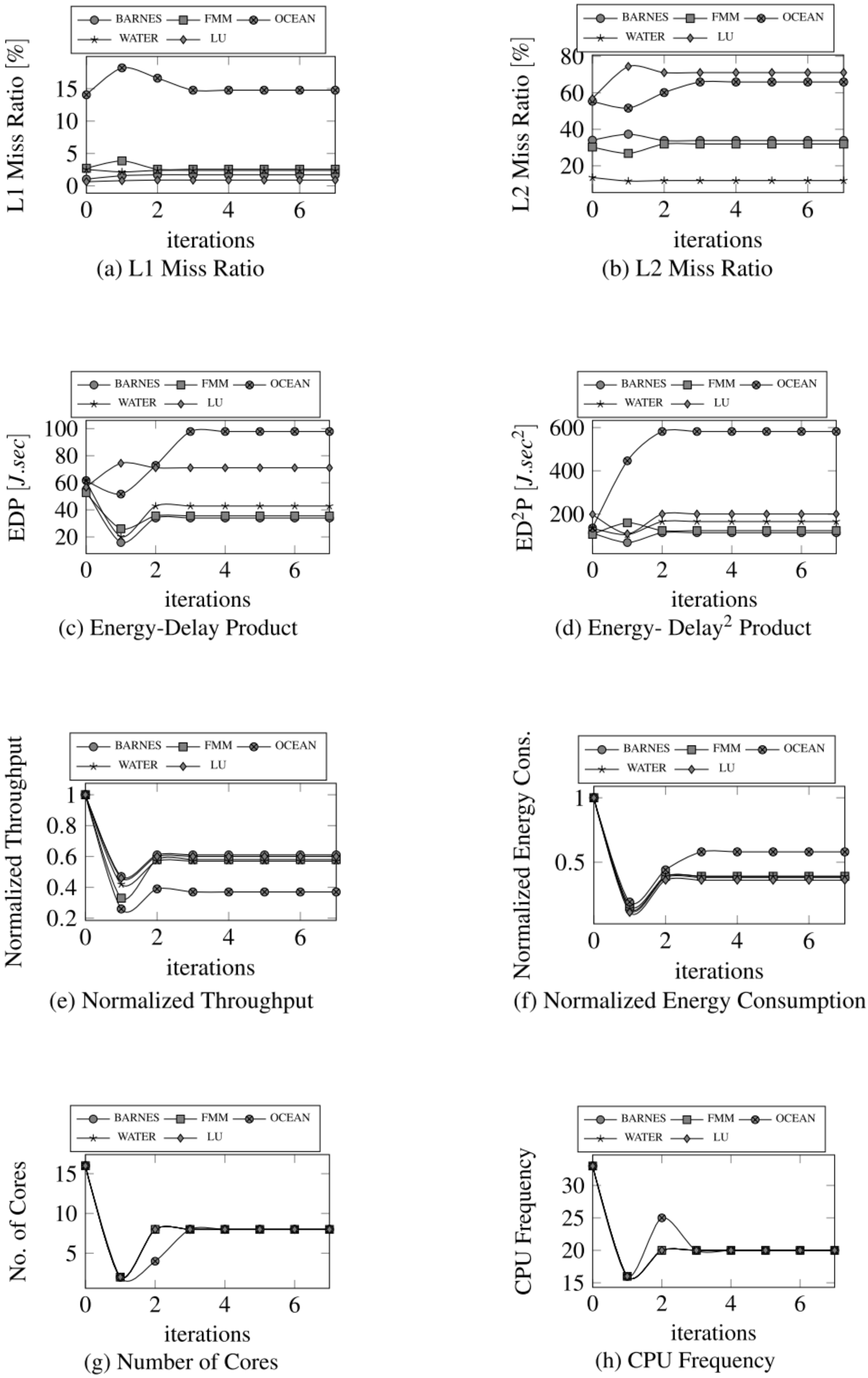
As discussed above, the main objective of this paper is to devise a strategy, which can incorporate an optimum balance between energy and throughput of MPSoC. Two different approaches of fuzzy inference engines have been applied on FLDSEE (engine). Gaussian membership function is used in both procedures. First results of both approaches have been shown and then there comparison has been analyzed.

6.1. Results using Sugeno inference model

Figure 4(a) shows L1 Miss Ratio at Iteration-0 all benchmarks have less miss ratio except OCEAN which reflects high miss ratio. They show 0.6-2.5% miss ratio, while in OCEAN, it is 14% respectively. Overall miss ratio is increased in iteration-1. At iteration-3 and so on, it is observed that miss ratio constant. Figure4(b) exhibits the organizational level miss ratio of L2 cache. At iteration-1, L2 cache miss ratio is increased but ceased at iteration-2. Figure 4(c) shows the energy-delay product of simulated workloads (benchmarks). Figure 4(d) shows the ED^{2}P, it is observable that ED^{2}P become constant at iteration-3. Normalized throughput is shown in Figure 4(e), which fluctuates in first three iterations and stabled at iteration-4. Also in Figure 4(f), which is a normalized energy, is varying in first three iterations and becomes smoothed at iteration-4. Cores are selected by taking average of all the iteration for number of cores, and for this scenario its 8 cores. Figure4(h) shows the CPU clock frequency, for which the system can be effectively configured to 25MHz.

6.2. Results using Mamdani inference model

In case of Mamdani inference engine, Figure 5(a) and Figure 5(b) shows the L1 and L2 miss ratio. L1 and L2 misss ratio is low at iteration-0. It increased at iteration-1 and again decreased at iteration-2. L1 cache miss ratio reduced from 2.4% to 0.8% in all benchmarks except BARNES, where it is 18.2% to 14.8%. In L2 case OCEAN and LU miss ratio is increased to 13-15%, but reduced by 2-6% in BARNES, FMM and WATER.Figure 5(c) reveals the energy-delay product of various simulated workloads and it settles at iteration-2. Figure 5(d) shows the ED^{2}P, it observable that ED^{2}P become constant at iteration-2. Normalized throughput is shown in Figure 5(e), there is a fluctuating effect observed in only first two iterations and become stable at iteration-2. Normalized energy consumption is displayed in Figure 5(f); energy consumption is highest in first iteration, reduced in second iteration and balanced after iteration-3. Number of cores are displayed in Figure 5(g). Cores are selected by taking average of all the iterations for number of cores, and for this scenario its 9 cores. And Figure 5(h) shows the CPU clock frequency, for which the system can effectively be configured to 20MHz.



(a) L1 Miss Ratio

(b) L2 Miss Ratio

(c) Energy-Delay Product

(d) Energy- Delay^2 Product

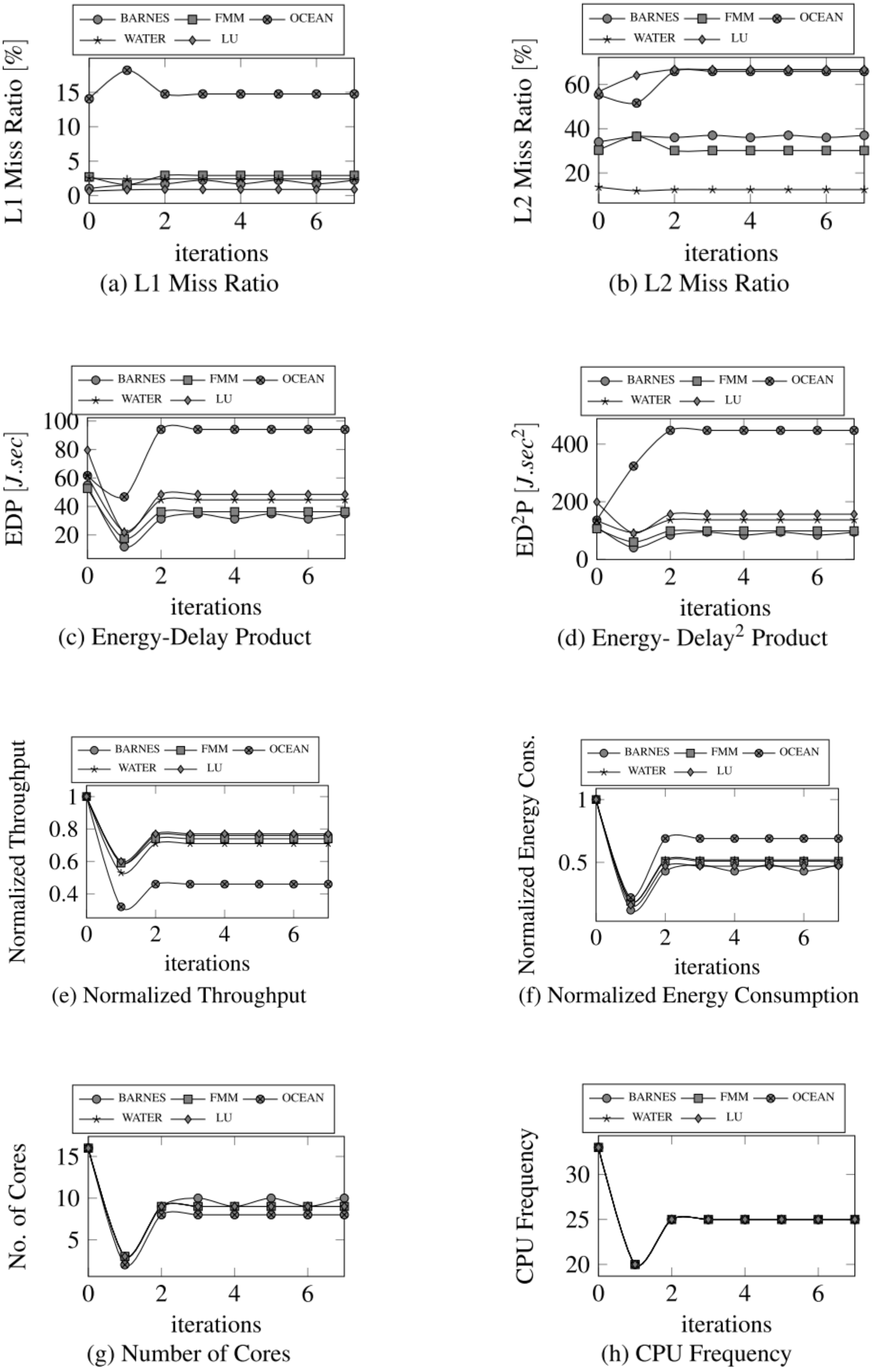
(e) Normalized Throughput

(f) Normalized Energy Consumption

(g) Number of Cores

(h) CPU Frequency

Fig. 4. Block results using Sugeno inference model



(a) L1 Miss Ratio

(b) L2 Miss Ratio

(c) Energy-Delay Product

(d) Energy- Delay^2 Product

(e) Normalized Throughput

(f) Normalized Energy Consumption

(g) Number of Cores

(h) CPU Frequency

Fig. 5. Block results using Mamdani inference model

6.3. Comparative Analysis

Comparison is made for various problem space and solution space parameters. So for convenience, different names are assigned to these comparisoned design space models (DSM) as; Sugeno inference based model — DSM1 and Mamdani inference based model — DSM2 and their analysis is as follows:

6.4. L1 Cache miss ratio

The L1 cache memory size and associativity is fixed in DSM1 and DSM2. By comparing DSM1 and DSM2, Mamdani model has less miss ratio then Sugeno model, which is 0.07% in FMM and 0.3% in OCEAN. There are no big variations in other benchmarks. L1 cache miss ratio is balanced in iteration-2 for DSM1 and DSM2, except for DSM1 OCEAN which settles in iteration-3.

6.5. L2 Cache miss ratio

In these DSMs, L2 cache has shared architecture among cores. DSM1 and DSM2 memory size and associativity are fixed throughout the design reconfiguration process. There is a variation among miss ratios in these DSMs. All benchmarks show there spatial and temporal effect on L2 cache as shown in Figure. 4(b), and Figure 5(b). By comparing DSM1 and DSM2, Sugeno model has less miss ratio then Mamdani model, which is 1.7% in BARNES, 0.1% in FMM, 0.8% in OCEAN and 0.4% in WATER, except 4.6% in LU. L2 cache miss ratio is balanced in iteration-2 for both DSM1 and DSM2, excluding DSM1 BARNES, which settles at iteration-3. However, reducing miss ratio is not the perspective of this paper, and ED^{2}P is the overall design challenge for reconfiguration of MPSoC architecture. This miss ratio effect can be improved by using a decision block within this frame reference. That decides which configuration has less miss ratio, and then that design strategy would be adopted by reconfiguring the cache size and associativity.

6.6. Energy-delay^2 product of MPSoP architecture

In Figure 4(d) and Figure 5(d), energy-delay^2 product is an aggregated energy of MPSoC times the square of execution time. DSM1 and DSM2, ED^{2}P can be improved by reconfiguring the caches. However it is beyond focus of this paper as discussed above. But, if configuration is made on its maximum strength means 16 cores & 33MHz frequency and then at lower strength 2 cores & 16MHz frequency as shown in Figure 4 (g,h) and Figure 5 (g,h) at iteration-2 and 3, ED^{2}P reduction phenomena is observed. DSM2 (Mamdani's model) ED^{2}P is decreased as 21% in BARNES, 25% in FMM, 18.7% in OCEAN, 14.6% in WATER and 18.1% in LU with respect to DSM1(Sugeno's model).

It can be noticed that the number of cores has a huge impact on throughput, while clock frequency has on MPSoC energy. In DSM1 the number of cores are finally decided as 7 for OCEAN and 8 for other benchmarks for optimum balance and same is the case for clock frequency, which is 20MHz, while in DSM2 the number of cores are varying with respect to the benchmarks requirements, but still can be adjusted at 8 cores for OCEAN and 9 cores for other benchmarks. However, clock frequency is same for all these benchmarks and selected as 25MHz at iteration-2. Due to these design selections;normalized throughput of DSM1 case is improved from 0.25 to 0.37 in OCEAN and 0.32 to 0.60 in others. Infact in DSM2 case, it is enhanced from 0.32 to 0.46 in OCEAN and 0.53 to 0.77 in other benchmarks. However, normalized energy consumption of DSM1 is increased from 0.13 to 0.38 in BARNES, 0.16 to 0.39 in FMM, 0.19 to 0.58 in OCEAN, 0.14 to 0.39 in WATER and 0.11-0.36 in LU, while in DSM2 it is 0.12-0.46 in BARNES, 0.19 to 0.51 in FMM, 0.22 to 0.52 in OCEAN, 0.19 to 0.52 in WATER and 0.16 to 0.47 in LU.

7. Conclusions and future work

In this paper, fuzzy logic is used to implement design space exploration of reconfigurable architectures for multiprocessors system on chip. Fuzzy logic is robust and adoptive in highly unpredictable nonlinear systems, although its solution is not precise, it works well in systems whose input is unpredictable with the workloads (applications). So by comparative analysis, it can be concluded that the DSM2 (Mamdani's model) is the optimum option for balancing a throughput and energy consumption in design space reconfiguration of MPSoC.

The proposed models (DSM1 and DSM2) are so established that, they are well suited in scenario of massive parallel architectures (cores) as well as in 2-4 cores based architecture schemes. It shows that by using fuzzy logic in DSE of MPSoC architecture, the performance of system can be raised, even if there is no optimal solution in-hand. Fuzzy logic is robust in the design space exploration process. Architecture reconfiguration can be built in ONLINE mode, and strategy needs no pre-evaluation simulations. This paper may help researchers, industry and market of multi-architectures systems, in implementing the balancing methodology for energy and throughput of highly parallel, battery and resource dependent problem space in an effective way.

As for future work, this work can be implemented with more design spaces. Analysis of all membership functions on design space to explore new balance points devise the fuzzy logic based design space exploration engine (FLDSEE) on heterogeneous architectures and NoC.

Acknowledgement

The work in this paper was fully supported by National ICT R & D Fund, Ministry of IT, Government of Pakistan, under the grant number: ICTRDF/TR&D/2012/65.

References

Baghdadi, A., Zergainoh, N., Cesario, W., Roudier, T. & Jerraya, A. A.(2000). Design spaceexploration for hardware/software co-design of multiprocessor systems. In Rapid System Prototyping, 2000. RSP 2000. Proceedings. 11th International Workshop on, pp 8–13. IEEE.

Bruni, D., Bogliolo, A. & Benini, L.(2001). Statistical design space exploration for application-specific unit synthesis. In Design Automation Conference, 2001. Proceedings, pp 641–646. IEEE.

Beltrame, G., Fossati, L. & Sciuto, D. (2010). Decision-theoretic design space exploration of multiprocessor platforms. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, **29**(7):1083–1095.

Blythe, S.A. & Walker, R.A. (2000). Efficient optimal design space characterization methodologies. ACM Transactions on Design Automation of Electronic Systems (TODAES), **5**(3):322–336.

Cant´o, E., Fons, M., Fons, F., L´opez, M. & Ramos-Lara, R. (2013). Fast self-reconfigurable embedded system on spartan-3. J. UCS, **19**(3):301–324.

Calborean, H., Jahr, R., Ungerer, T. & Vintan, L. (2011). Optimizing a superscalar systemusing multi-objective design space exploration. In Proceedings of the 18th International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, volume 1, pp 339–346.

Erbas, C., Cerav-Erbas, S. & Pimentel, A.D. (2006). Multi-objective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design. Evolutionary Computation, IEEE Transactions on, **10**(3):358–374.

Et, M., YAM & Basarir, M. (2007). On some generalized difference sequences of fuzzy numbers. Kuwait Journal of Science and Engineering, Volume, **34**(3):1–14

Gajski, D.D., Vahid, F., Narayan, S. & Gong, J. (1998). Specsyn: An environment supporting the specify-explore-refine paradigm for hardware/software system design. Very Large Scale Integration (VLSI) Systems, IEEE Transactions, **6**(1):84–100.

Gonzalez, R. & Horowitz, M. (1996). Energy dissipation in general purpose microprocessors. IEEE Journal of Solid-State Circuits, **31**(9):1277–1284.

Hameed, I.A. (2011). Using gaussian membership functions for improving the reliability and robustness of students evaluation systems. Expert Systems with Applications, **38**(6):7135–7142.

Ishebabi, H. (2010). Architecture synthesis for adaptive multiprocessor systems on chip. Ph.D. thesis, Hauptgebäude, Universität Wien, Universitätsring 1, 1010 Wien, Austria.

Kahng, A.B., Lin, B. & Nath, S. (2012). Comprehensive modeling methodologies for No Crouter estimation. Department of Computer Science and Engineering, University of California, San Diego.

Kreutz, M., Marcon, C. A., Carro, L., Wagner, F. & Susin, A. A. (2005). Designspace exploration comparing homogeneous and heterogeneous network-on-chip architectures. In Proceedings of the 18th annual symposium on Integrated circuits and system design, pp 190–195. ACM.

Kang, S. & Kumar, R. (2008). Magellan: a search and machine learning-based framework for fast multi-core design space exploration and optimization. In Proceedings of the conference on Design, automation and test in Europe, pp 1432–1437. ACM.

Lahiri, K., Raghunathan, A. & Dey, S. (2001). System-level performance analysis for designing on-chip communication architectures. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions, **20**(6):768–783.

Lukasiewycz, M., Glaß, M., Haubelt, C. & Teich, J. (2008). Efficient symbolic multi-objective design space exploration. In Proceedings of the 2008 Asia and South Pacific Design Automation Conference, pp 691–696. IEEE Computer Society Press.

Monchiero, M., Canal, R. & Gonz´alez, A. (2006). Design space exploration for multicore architectures: a power/performance/thermal view. In Proceedings of the 20th annual international conference on Supercomputing, pp 177–186. ACM.

Mishra, Kumar, V. & Anirban, S. (2014). MO-PSE: Adaptive multi-objective particle swarm optimization based design space exploration in architectural synthesis for application specific processor design. Advances in Engineering Software. Elsevier, **67**(12):111-124.

Mandal, S.N., Choudhury, J.P. & Chaudhuri, S.B. (2012). In search of suitable fuzzy membership function in prediction of time series data. International Journal of Computer Science, **9**(3):293-302

Orsila, H., Kangas, T., Salminen, E. & Hamalainen, T.D. (2006). Parameter zing simulated annealing for distributing task graphs on multiprocessor SoCs. In System-on-Chip, 2006. International Symposium on, pp 1–4. IEEE.

Patel, A., Afram, F., Chen S. & Ghose, K. (2011). MARSS: A full system simulator formulticore x86 CPUs. In Proceedings of the 48th Design Automation Conference, pp 1050–1055. ACM.

Palermo, G., Silvano, G., Valsecchi, S. & Zaccaria, V. (2003). A system-level methodology for fast multi-objective design space exploration. In Proceedings of the 13th ACM Great Lakes symposium on VLSI, pp 92–95. ACM.

Phillips, J.D. (2008). AC to register transfer level algorithm using structured circuit templates: A case study with simulated annealing. Ph.D. thesis, Utah State University, Logan, Utah, United States.

Qadri, M.Y., McDonald Maier, K.D. & Qadri, N.N. (2014). Energy and throughput aware fuzzy logic based reconfiguration for MPSoCs. Journal of Intelligent & Fuzzy Systems, **26**(1):101-113.

Qadri, M.Y. & McDonald Maier, K.D. (2009). Data cache-energy and throughput models: design exploration for embedded processors. EURASIP journal on embedded systems, **1**(2009):1-7.

Shivakumar, P. & Jouppi, N.P. (2001). CACTI 3.0: An integrated cache timing, power, and area model. Technical report, Technical Report 2001/2, Compaq Computer Corporation.

Sedighpour, M., VAHID, A., Yousefikhoshbakht, M., Didehvar, F. & Rahmati, F. (2014). Solving the open vehicle routing problem by a hybrid ant colony optimization. Kuwait Journal of Science, **41**(3):139-162

Woo, S.C., Ohara, M., Torrie, E., Singh, J.P. & Gupta, A. (1995). The SPLASH-2 programs: Characterization and methodological considerations. In ACM SIGARCH Computer Architecture News, volume 23, pp 24–36. ACM.

Submitted **:** 21-01-2015
Revised **:** 19-09-2015
Accepted **:** 18-10-2015

موازنة الطاقة والإنتاجية في المعالجات المتعددة على رقاقة باستخدام المنطق الضبابي

[1]عديل احمد، [2]محمد ياسر قادري، [3]نادية ن. قادري، [1]جميل أحمد

[1]قسم الهندسة الكهربائية، جامعة HITEC، تاكسيلاكنت، باكستان.
[2]مدرسة علوم الحاسب والهندسة الإلكترونية، جامعة إسكس، كولشستر، CO4 3SQ، المملكة المتحدة.
[3]معهد COMSATS لتقنية المعلومات، واهكنت، باكستان.
المؤلف:yasirqadri@acm.org

خلاصة

في هذه الورقة تم اقتراح وحدة تحكم قائمة على المنطق الضبابي تهدف إلى إعادة تشكيل نظام متعدد المعالجات على رقاقة (MPSoC) وفقا لمتطلبات العمل. تتم مقارنة محركات الاستدلال Mamdani و Sugeno و تحليلها لتعزيز التنظيم السلس لإعادة التشكيل في فضاء التصميم. تتكون منصة MPSoC المقترحة من 16 معالج، حيث يحتوي كل معالج على ذاكرة كاش خاصة L1 وذاكرة كاشمشتركة L2. والمتغيرات التي تم تقييمها هي تردد تشغيل وحدات المعالجة المركزية، وعدد المعالجات، والإنتاجية واستهلاك الطاقة. يستخدممحاكاة Marssx86 لبناء الجهاز الهدف ويتم تنفيذ إختبارات SPLASH-2 عليه. وقد أظهرمحرك Mamdani إنخفاض يصل إلى 14%-23% في حاصل ضرب الطاقة والتأخيرلمختلف الإختبارات القياسية. ومن ثم تم إستنتاج أن أداء محرك الاستدلال Mamdani أفضل من بالنسبة Sugeno لهذا التطبيق.

الكلمات المفتاحية:

استكشاف فضاء التصميم، المنطق الضبابي، MPSoC، عمارة قابلة لإعادة التشكيل.