A Multi-Objective Approach for Multi-Application NoC Mapping

Johanna Sepúlveda, Marius Strum, Wang Jiang Chau
Microelectronics Laboratory LME
University of São Paulo
São Paulo, Brazil
jsepulveda,strum,jcwang@lme.usp.br

Guy Gogniat
Lab-STICC Laboratory
Université de Bretagne Sud
Lorient, France
Guy.gogniat@univ-ubs.fr

***Abstract*—Current SoC design trends are characterized by the integration of larger amount of IPs targeting a wide range of application fields. Such multi-application systems are constrained by a set of requirements. In such scenario network-on-chips (NoC) are becoming more important as the on-chip communication structure. Designing an optimal NoC for satisfying the requirements of each individual application requires the specification of a large set of configuration parameters leading to a wide solution space. It has been shown that IP mapping is one of the most critical parameters in NoC design, strongly influencing the SoC performance. IP mapping has been solved for single application systems. In this paper we propose the use of a multi-objective adaptive immune algorithm *(M^2AIA)*, an evolutionary approach to solve the multi-application NoC mapping problem. Latency and power consumption were adopted as the target multi-objective functions. To compare the efficiency of our approach, our results are compared with those of the genetic and branch-and-bound multi-objective mapping algorithms. We tested algorithm on several SoC applications.**

Keywords-Network-on-Chip; mapping; multi-application; multi-objective.

I. INTRODUCTION

Electronics system design is being revolutionized by the widespread adoption of the system-on-chip (*SoC*) paradigm. A SoC can integrate hundreds of cores on a single die. In such a scenario, *SoC* designers are faced with the task of meeting the design requirements in a reduced time-to-market. To be cost effective, SoCs are often programmable and integrate several different applications on the same chip (i.e cell-phone, personal digital assistant) [1]. Although sharing many of the hardware components on the SoC, the different applications executed on the same die may present very different communication requirements and design constraints. Such type of system is called multi-application [1]. A communication centric paradigm, Network-on-chip (*NoC*), has been adopted to address the interconnection issues of current SoCs. NoC has become the heart of the SoC [2]. A *NoC* is an integrated network that uses routers to allow the communication among the computation structure components. The NoC configuration has a great impact on the cost and on the performance of the system-on-chip [2]. A *NoC* may be configured by a set of global parameters (topology, size and mapping) and local parameters (link width, buffer configuration, flow control, routing technique, arbitration mechanism) leading to a very large NoC design space to be explored. The final configuration of the NoC must support the requirements of all the applications of the SoC. NoCs designed for a particular application does not necessarily meet the requirements of the remaining applications. Finding an optimal global solution is not an easy task [2]. This paper addresses the *mapping problem*. It deals with the allocation of HW cores onto the network routers such that the all the applications requirements of the SoC are met and a set of performance metrics are optimized.

According to [1], NoC mapping is one of the most critical parameters in NoC design. Previous works showed that an optimal mapping may enhance the NoC performance up to 60% [2]. Mapping is a quadratic assignment problem that is known to be NP-hard [3]. The search space of the problem increases factorially with the system size [4]. Furthermore, the mapping solution must satisfy all the system requirements consisting of multiple desired objectives that are frequently in contrast with each other [4]. To the best of our knowledge, only [1] addresses the multi-application NoC mapping with the aim of minimizing communication delay by exploiting the possibility of splitting traffic among various paths. However, previous works show that mapping strategies that search for single performance index optimization may lead to unacceptable values for other performance indexes [4]. The best mapping solutions have been obtained using a multi-objective strategy [4-6]. As a result, the designer obtains a set of best mapping alternatives (*Pareto optimal set, nondominated solutions*) featuring different trade-offs among the performance indexes [4-6]. Pareto dominance is used to compare and rank the mapping solutions. A mapping belongs to the *Pareto optimal set* if there is no other mapping that can improve at least one of the objectives without degrading any other objective (*nondominance*) [4-6,8]. Three multi-objective mapping strategies, PBBB, MGAP and MAIA have been proposed to solve single application NoC mapping [4-5]. Their goal was the optimization of latency and power consumption for a mesh-based NoC. PBBB uses a branch and bound algorithm [4], MGAP uses a genetics algorithm [5] and MAIA uses an adaptive immune algorithm (AIA) [6].

This paper is an evolution of the work presented at [6]. In this work we propose M^2AIA an improved version of our Multi-objective Adaptive Immune Algorithm (MAIA), to solve the multi-application NoC mapping problem. M^2AIA

978-1-4244-9485-9/11/$26.00 ©2011 IEEE

explores the mapping space producing a set of best mapping alternatives. We compared our solution with modified versions that we implemented for the PBBB (MA_PBBB) and MGAP (MA_MGAP) algorithms. The Pareto optimal set of all 3 algorithms were then evaluated and compared using a NoC-based TLM *(SystemC)* simulation environment.

The remaining text is divided in five sections. Section 2 presents an overview of the previous multi-objective mapping works. Section 3 presents the M^{2}AIA mapping algorithm. Section 4 shows our experimental results and the comparison among M^{2}AIA, MA_PBBB and MA_MGAP. Finally we present our conclusions in Section 5.

II. Related Work

NoC Mapping has been widely explored [1,4-8]. The purpose of these previous works is to find a NoC configuration that satisfies the requirements of the SoC. According to the number of the optimization objectives and the number of application supported by the SoC, previous works can be divided into 3 categories: 1- *Single objective and single application* [7-8]; 2- *Multiple objective and single application* [4-6]; and 3- *Single objective and multiple applications* [1]. All works [1, 4-8] used an application characterization graph (APCG) that describes the communication requirements. The works that belong to the first category [7-8] presented a heuristic algorithm that selects the first NoC configuration that satisfies the single application SoC latency requirement. However, the works of [4-6] show that for many applications a single objective optimization is not enough. Moreover, the requirements of the set of applications may be contrasting. The works of the second category [4-6] employed multi-objective algorithms to solve the mapping problem for mesh-based NoCs while optimizing latency and power indexes. In [4] PBBB, a branch-and-bound algorithm is proposed. PBBB maps the cores according to their communication traffic, creating a tree of mapping alternatives. At the bound phase, each mapping alternative is evaluated according to both optimization objectives through event-driven trace-based simulation (dynamical model). The best mapping alternatives are kept while the others are pruned. The branch and bound phases are repeated on the survivors. In [5] MGAP, a genetic mapping algorithm is presented. MGAP codifies different mapping alternatives in chromosomes. The mapping alternatives are evaluated through an analytical model (static model). Crossover and mutation operators are used in order to explore the mapping space (create new mapping alternatives). However, using only a static or dynamic model can generate suboptimal solutions [6]. Our previous work of [6] combines a static-dynamic model approach to find the mapping alternatives that optimizes the performance metrics. It uses an artificial immune algorithm to explore the efficiently the huge NoC design space. MAIA integrate a wide set of features that improve local search while preventing the premature convergence by preserving the diversity of solutions in the population. The pareto optimal set is then simulated through a TLM SoC model. Despite their good results, these strategies must be modified to support the new systems requirements, characterized by supporting different applications that may have different performance requirements and design constraints. The work of [1] is the only previous work that belongs to the third category. [1] proposes a heuristic capable of select the NoC configuration that satisfies the latency requirements of all the applications. However, for the best of our knowledge our work is the first attempt that addresses the multi-application NoC mapping while optimizing multiple performance indexes.

III. M^{2}AIA

An immune system protects the organism by producing antibodies capable of identify attackers (antigens). It constantly monitors the defence process through the evaluation of the *affinity* and *avidity*. They quantify the match between antigen-antibody pairs (for recognition) and between a single antibody and the whole antibody population (for diversity). The survival of a specific antibody depends on these values. The immune system integrates a wide set of mechanisms: pattern recognition (affinity), clonal selection (cloning the antibodies that best match the antigen), clonal suppression (killing the worst antibodies), mutation (modifying a set of antibodies), affinity maturation (creating a new set of antibodies), learning and memory (storing the successful antibodies). M^{2}AIA uses the MAIA adaptive immune algorithm to solve the mapping problem. Table 1 shows the metaphors employed by MAIA. The algorithm is depicted on Figure 1.

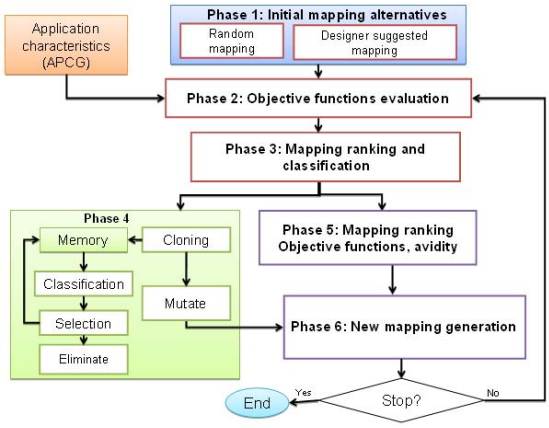

Figure 1. MAIA algorithm

Table 1. Immune system metaphors

Immune system feature	MAIA
Antigen	Application characterization graph (APCG)
Antibody	Mapping alternative
Pattern recognition	Multi-objective quantification
Clonal selection	Top mapping alternatives selection
Clonal suppression	Mapping alternatives elimination
Mutation	Mapping alternatives modification
Maturation	Mapping alternatives creation
Learning and memory	Mapping solutions

A. Mapping algorithm for multiple-objectives and a single application

MAIA uses the APCG of the application and the size of the NoC to find the set of mapping alternatives that optimizes the objective functions. MAIA is composed of six phases.

In the *first phase* of the mapping algorithm, a set of random and designer suggested mapping alternatives is generated. Each mapping consists of an IP core-NoC router pair. At the *second phase*, the power consumption and latency objective functions of each mapping alternative is evaluated. Note that MAIA can support any objective functions as well. Then, the *dominance value* [4-5] of the *objective functions* results are used to tank the mapping alternatives at the *third phase.* Each *m* mapping is assigned a fitness value *r(m,i)* based on its rank *d(m,i)* at *i* iteration, as in (1).

$$r(m,i) = 1 + d(m,i) \quad (1)$$

At the *fourth phase* the *Pareto optimal set* is refined. Only the non-dominant solutions are kept in memory and the remaining are erased. The best solutions are then copied and modified according two mutation operators: *shift* (random shift of IPs) and *somatic point* (random swap of two IP). The latter mappings will be employed to generate the next-iteration mapping alternatives. At the *fifth phase*, the dominated mappings, out of the memory, are ranked according to two parameters: 1) the objective functions and 2) avidity (normalized sum of Euclidean distances between every solution pair). The purpose is to identify and penalize mapping solutions in densely populated areas. Finally, at the *sixth phase*, The new set of mapping alternatives coming from the combination (crossover) of the mutated mappings (phase 4) and the mapping alternatives of the (phase 5). MAIA stops when no more significant improvement can be expected.

B. Mapping algorithm for multiple-application SoC

In order to solve the multi-application SoC mapping, M^{2}AIA adopts the combination of the APCGs of each application of the SoC in order to generate a synthetic APCG, used as an entry of the optimization process. The new APCG is called of Worst-Case APCG (WC-APCG). It includes all the IP cores integrated at the SoC. Figure 2 shows the mapping technique M^{2}AIA. For the communication flow between every pair of IP cores, the tightest communication requirements across all the applications are selected as the requirements of the WC-APCG. Thus the design constraints of all the individual applications are subsumed in the WC-APCG and any NoC mapping that satisfies the constraints in the WC-APCG will satisfy the constraints of each individual application. The WC-APCG is then used for the mapping process. Figure 3. shows an example of the generation of a WC-APCG from a SoC that executes 2 applications. Each application is described by an APCG. The applications share 5 of the 6 cores of the SoC. Each IP pair is characterized by the latency (cycles) and power (mW) requirements. The WC-APCG is composed of 6 cores, whose IP-pairs are characterized by the tightest requirements for both characteristics: latency and power.

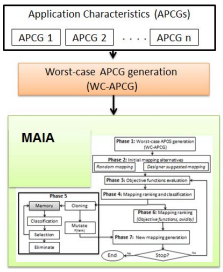

Figure 2. Proposed approach M^{2}AIA

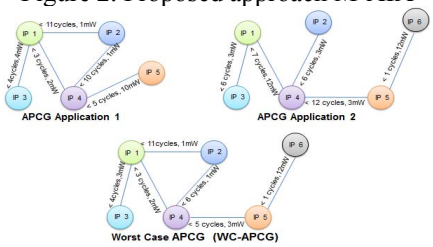

Figure 3. Example of WC-APCG generation

Once M^{2}AIA obtains the *Pareto optimal set* from the WC-APCG, a dynamical evaluation is performed using the SystemC-TLM SoC simulation and evaluation framework.

IV. EXPERIMENTAL RESULTS

M^{2}AIA was tested and compared with the MA_PBBB (branch and bound) [4] and MA_MGAP (genetic-based) [5] multi-objective algorithms. MA_PBBB and MA_MGAP were modified in order to support multi-application mapping. We implemented M^{2}AIA, MA_PBBB and MA_MGAP in C++. All were executed on a single Pentium IV – 1,73 GHz personal computer. For comparison purposes, all 3 algorithms perform the mapping exploration using the same analytical model. The purpose of our experiments was to minimize the latency L_{NoC} and the power consumption P_{NoC}. The objective functions are given by (2) and (3) respectively.

$$L_{NoC} = \frac{\sum t_{acc} + (h_{s,d} - 1)t_C + t_{lea}}{\#\ flits} \quad (2)$$

$$P_{NoC} = \frac{\sum (h_{s,d} + 1)P_R + h_{s,d}P_L}{\#\ flits} \quad (3)$$

L_{NoC} is determined by three components: i) the time t_{acc} to access the NoC and insert the flit; ii) the commutation time t_c, spent by the intermediate h routers from s source to d destination and; iii) the time t_{lea} required to leave the NoC. P_{NoC} is determined by P_R and P_L, the power consumed in the routers and links respectively. P_R and P_L are proportional to the *channel utilization rate* and *router utilization rate* respectively. All the tests were performed on a homogeneous wormhole 2D mesh-based NoC, a XY routing algorithm, 4-flits sized buffers and a round-robin arbitration technique. Table 2 shows the adopted M^{2}AIA parameters. The multi-objective mapping algorithms were used to solve the mapping problem of 11 multi-application benchmarks. Table 3 shows the characteristics of the experimental work. The APCG values of each benchmark were randomly selected. Benchmarks T1-T7 combine different well-known embedded communication patterns [8].

Table 2. M²AIA parameters

Parameter	Value
Initial population *M* (Phase 2)	100
Latency objective function	L_{NoC}
Power objective function	P_{NoC}
Mutation probablity (Phase 5)	0,1
Crossover (Phase 7)	40%
Stop criterion	0.1

Table 3. Characteristics of the experimental work

Benchmarck	# IP´s	# APCGs	Type
T1	9	5	Hot Spot/ Transpose
T2	16		Hot Spot/ Transpose/Random
T3	25		Hot Spot/ Transpose/Random
T4	36		Hot Spot/ Transpose/Random
T5	49		Hot Spot/ Transpose/Random
T6	100		Hot Spot/ Transpose/Random
T7	12		3-Node rooted (1,2,3) forest
T8	7	3	Networking
T9	16	8	Automotive
T10		5	Consumer devices
T11		4	Security

Benchmarks T8-T11 are part of MiBench tool suite. Each benchmark is composed of a set of applications targeting a specific area of the embedded market: automotive, consumer devices and security [10]. T8 represents embedded processors in network devices like switches and routers. It involves the shortest path calculation, tree and table lookups and data input/output. T9 represents typical applications of the automotive systems like air bag controllers, engine performance monitors and sensor systems. The processors require performance in basic math abilities, bit manipulation, data input/output and simple data organization. T10 focuses mainly on multimedia applications. It includes encoding/decoding algorithms, image color format conversion, image dithering and color palette reduction. T11 includes several common algorithms for data encryption, decryption and hashing. Figures 4 and 5 show the TLM-based simulation results for the NoC power and NoC latency respectively of the Pareto optimal set of the 3 algorithms. The M²AIA algorithm produced better results for all benchmarks. M²AIA found NoC mappings that satisfy the requirements of the set of applications and also achieves higher optimizations in power and latency when compared with the MA_PBBB and MA_MGAP. Our algorithm decreases the power consumption in average 27.3% and 42.1% and the latency 29.3% and 36.1% over the MA_PBBB and MA_MGAP, respectively. The best improvement over MA_PBBB was for T3 (52% and 51% for power and NoC latency respectively). The best improvement over MA_MGAP was for T10 (34% and 50% respectively). We evaluated the performance of the 3 algorithms. We use three performance metrics: 1-Spacing *P* (distance between the mapping solutions); 2-Spread *A* (distance among all the mapping alternatives); and 3-Execution time *T* (time spent to reach the stop criterion). The results show that M²AIA achieves a lower spacing and spread values, so that it performs a uniform exploration. Moreover, M²AIA speedups the mapping search almost 100000 times when compared to the MA_PBBB and MA_MGAP techniques. These are desirable characteristics for a search algorithm [6]. The results show the

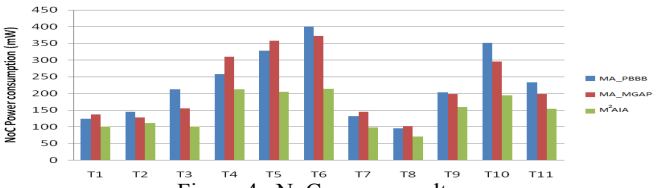

Figure 4. NoC power results

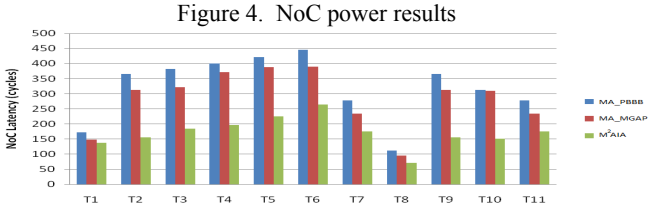

Figure 5. NoC latency results

M²AIA independence of the number of IP cores. Also M²AIA obtained better results than all other reported algorithms in a shorter time.

V. CONCLUSIONS

In this paper we propose the use of a multi-objective adaptive immune algorithm (M²AIA), an evolutionary approach to solve the multi-application NoC mapping problem. The contributions of our work include the adoption of an adaptive immune algorithm combined with the use of both, static and dynamic mapping evaluation techniques in order to improve the efficiency of the exploration of the multi-application mapping space. Previous multi-objective algorithms were modified in order to support the multi-application mapping. M²AIA was tested for a mesh-based NoC targeting the minimization of the total amount of power consumption and latency. M²AIA may use others objective functions and NoC topology. As future work, we plan to use M²AIA to define the NoC sizing parameter.

REFERENCES

[1] S. Murali, M. Coenen, A. Radulescu, K. Goosens, G. De Micheli. "Mapping and Configuration Methods for Multi-Use-Case NoCs". In Proc. Asia and South Pacific DAC conference. 2006.

[2] L. Benini, D. Bertozzi. *"Network-on-Chip Architectures and Design Methods"*. In *Proc*. Comput. Digit. Tech, 2005.

[3] M. R. Garey, D.S. Johnson. *"Computers and Intractability: a Guide to the Theory of NP-Completeness"*. Freeman, 1979.

[4] G. Ascia, V. Catania. *"An Evolutionary Approach to Network-on-chip Mapping Problem"*. In Proc. IEEE Evolutionary computation, 2005.

[5] R. Jena, G. Sharma. "*A Multi-objective Evolutionary Algorithm Based Optimization Model for Network-on-Chip Synthesis"*. In Proc. International Conference on Information Technology, 2007.

[6] J. Sepulveda, R. Pires, M. Strum, W.J. Chau. *"An evolutive approach for multi-objective NoC mapping"*. In Proc. 17th IFIP/IEEE International Conference on very Large Scale Integration. VLSI-SoC 2009.

[7] J. Hu, R. Marculescu. *"Energy and Performance Aware Mapping for Regular NoC Architectures"*. In *Proc*. Computer Aided Design Integration Circuits and Systems, 2005.

[8] E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny. *"QNoC: QoS architecture and design process for network on chip"*. In Journal of system architecture. 2004.

[9] S. Pasricha, N. Dutt, F. Kurdahi. *"Dynamically Reconfigurable On-Chip Communication Architecture"*. In Proc. Asia and South Pacific DAC conference. 2009.

[10] M. Guthaus, J. Ringenberg, D. Ernst. *"MiBench: A free, commercially representative embedded benchmark suite"*. In Proc. International Workshop on Workload Characterization. 2001.