

An optimized Mapping of IP Core onto NoC using Multi-Objective Evolutionary Algorithms

N.Janakiraman ^{#1} P.Mano Arunika ^{#2}, P.Nirmal Kumar ^{#3}

[#] Department of ECE, K.L.N college of Engineering, Guindy, Anna University, Chennai, Tamilnadu, India

Abstract— In a System on a Chip (SoC), the communication between the components using network switches is termed as Network on a Chip (NoC). Each interconnection or communication between the components is carried out by the Intellectual Property (IP) block of each component. Mapping of IP core onto NoC is the major NP-hard problem while designing the platform based NoC. To solve this problem, we are using Multi-Objective Evolutionary Algorithms (MOEAs). The IP should be selected to exploit re-usability and to optimize the execution of a given application. Mapping an IP deals with the accurate map of selected IP onto the network platform. The resultant NoC platform will be minimized in the required hardware area, execution time and in power consumption, and the hotspots can also be avoided. Thus providing a custom-cut NoC platform for the application at hand.

Keywords—Network-on-chip, IP assignment, IP mapping, Multi-Objective design.

I. INTRODUCTION

Due to the increased integration rate of semiconductor devices, the complex System-on Chips are launched more. Basically a simple SoC can be formed by homogeneous or heterogeneous independent components. The communication between these components by means of communication architecture tends to Network-on-chip and each component of a NoC is designed as an Intellectual Property blocks. It can be of general or special purpose with processors, memories and DSPs. Only limited resources such as area, bandwidth and power can be used in a NoC [1, 2]. We can design a NoC in-order to run a specific application which contains a limited number of tasks that can be implemented by set of IP blocks. A single IP block can be assigned to a number of task in an application.

A predefined minimal architectural design called platform based design is a standardized process to speedup the designing process. In-order to make it more effective, we have to choose an adequate minimal set of IP blocks. Besides that it is very necessary to map these IP blocks onto NoC infrastructure [3]. Poor mapping and application will degrade the performance of the application. Therefore a perfect mapping is very necessary for NoC designing.

We can describe an application in a higher level by means of task graph (TG). Mapping and assigning a perfect IP is main NP-hard problem while designing. By using Multi-Objective Evolutionary Algorithms (MOEAs) we can easily solve these problems [4, 5]. Here two MOEAs were used and they are Non-Sorted Genetic Algorithm-II (NSGA-II) and micro Genetic Algorithm (micro GA). These algorithms can be modified according to the NoC design. The optimization can be done by the minimization of required hardware area, execution time and the total power consumption.

Traditionally, ICs have been designed with dedicated point-to-point connections, with one wire dedicated to each signal. For large designs, in particular, this has several limitations from a physical design viewpoint. These wires occupies much of the area of the chip, and in nanometer CMOS technology, interconnects dominate both performance and dynamic power dissipation, as signal propagation in wires across the chip requires multiple clock cycles.

An approach to exceed such a limitation of communication and overcome such an enormous wiring delay in future technology is to adopt network-like interconnections which is called Network-on-Chip (NoC) architecture. Basic concept of such kind of interconnections is from the modern computer network evolution. By applying network-like communication which inserts some routers in-between each communication object, the required wiring can be shortened. Therefore, the switch-based interconnection

mechanism provides a lot of scalability and freedom from the limitation of complex wiring. Replacement of SoC busses by NoCs will follow the same path of data communications when the economics prove that the NoC either reduces SoC manufacturing cost, SoC time to market, SoC time to volume, and SoC design risk or increases SoC performance. The NoC approach has a clear advantage over traditional busses and most notably system throughput. The hierarchies of crossbars or multilayered busses have characteristics somewhere in between traditional busses and NoC; however they still fall far short of the NoC with respect to performance and complexity.

The success of the NoC design depends on the research of the interfaces between processing elements of NoC and interconnection fabric. The interconnection of a SoC established procedures has some weak points in those respects of slow bus response time, energy limitation, scalability problem and bandwidth limitation. Bus interconnection composed of a large number of components in a network interface can cause slow interface time. In addition the interconnection has a defect that power consumption is high on the score of connecting all objects in the communication. Moreover it is impossible to increase the number of connection of the elements infinitely by reason of the limitation of bandwidth in a bus. As a consequence, the performance of the NoC design relies greatly on the interconnection paradigm.

Though the network technology in computer network is already well developed, it is almost impossible to apply to a chip-level intercommunication environment without any modification or reduction. For that reason, many researchers are trying to develop appropriate network architectures for on-chip communication. To be eligible for NoC architecture, the basic functionality should be simple and light-weighted because the implemented component of NoC architecture should be small enough to be a basic component constructing a SoC. Even though the basic functionality should be simple, it also satisfies the basic requirement in general communication. On the other hand, to apply the prevailing mobile environment, it should be low-powered. In order to be low powered one has to consider many parameters such as clock rate, operating voltage, and power management scheme.

II. NOC INTERNAL STRUCTURE

A network-on-chip is composed of three main building blocks. The first and most important one is the links that physically connect the nodes and actually implement the communication. The second block is the router, which implements the communication protocol (the decentralized logic behind the communication protocol). One can see the NoC as an evolution of the segmented busses where the router plays the role of a “much smarter buffer”. The router basically receives packets from the shared links and, according to the address informed in each packet, it forwards the packet to the core attached to it or to another shared link. The

protocol itself consists of a set of policies defined during the design (and implemented within the router) to handle common situations during the transmission of a packet, such as, having two or more packets arriving at the same time or disputing the same channel, avoiding deadlock situations, reducing the communication latency, increasing the throughput, etc. The last building block is the network adapter (NA) or network interface (NI). This block makes the logic connection between the IP cores and the network, since each IP may have a distinct interface protocol with respect to the network.

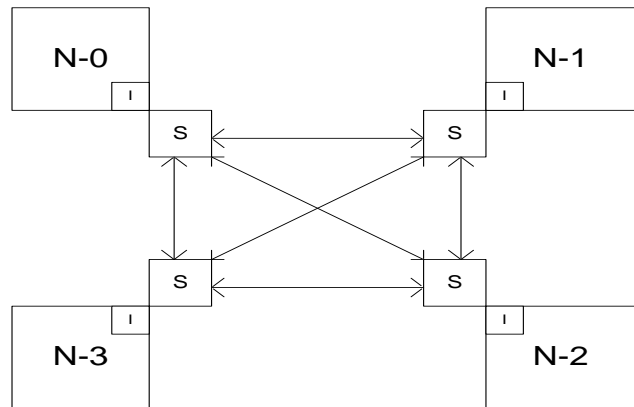


Fig 1. 2x2 NoC mesh architecture

A. NoC Links

A communication link may consist of one or more logical or physical channels and each channel is composed of a set of wires. In the remaining chapters, unless stated otherwise, the words net, wire, and line mean a single wire interconnecting two entities (IP cores). The words channel and link mean a group of wires connecting two entities. Typically, a NoC link has two physical channels making a full-duplex connection between the routers (two uni-direction channels in opposite directions). The number of wires per channel is uniform throughout the network and is known as the channel bandwidth.

B. Network interfacing unit

The third NoC building block is the network adapter (NA) or network interface (NI). This block makes the logic connection between the IP cores and the network, since each IP may have a distinct interface protocol with respect to the network. This block is important because it allows the separation between computation and communication. This allows the reuse of both, core and communication infrastructure independent of each other.

C. Routers

The design and implementation of a router requires the definition of a set of policies to deal with packet collision, the routing itself, and so on.

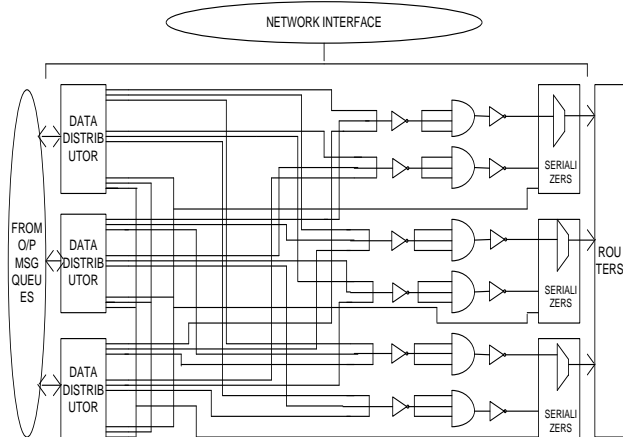


Fig 2. An interfacing unit

A NoC router is composed of a number of input ports (connected to shared NoC channels), a number of output ports (connected to possibly other shared channels), a switching matrix connecting the input ports to the output ports, and a local port to access the IP core connected to this router.

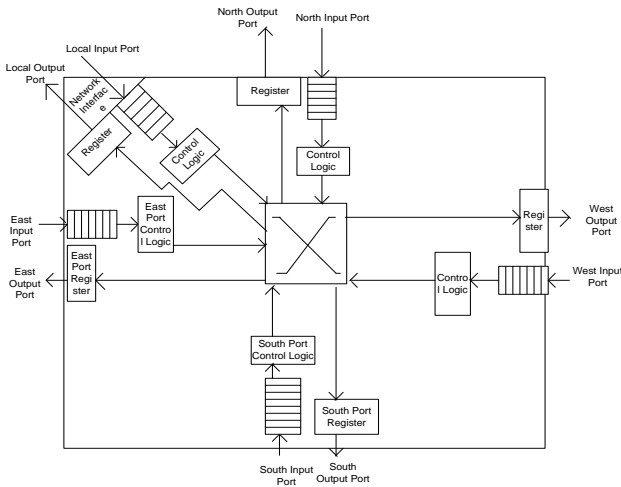


Fig 3. NoC router architecture

III. APPLICATION MAPPING RELATED PROBLEM

The platform based methodology encourages the reuse of components in-order to reduce the cost, and the time-to-market of new design in a SoC. The two main problems we are facing while design: exact selection of adequate set of IPs and finding the best way to map this IPs into the NoC structure. The selection of IPs is called IP assignment stage and the physical mapping is called IP mapping stage.

A. IP assignment problem

To assign, we have to select the adoptable IP from the repository IPs. They are the set of reusable IP formats which can be assignable to execute the given

application. At this stage, no information about physical location of IPs is available so optimization must be done based on application’s description (TG) and IP features only.

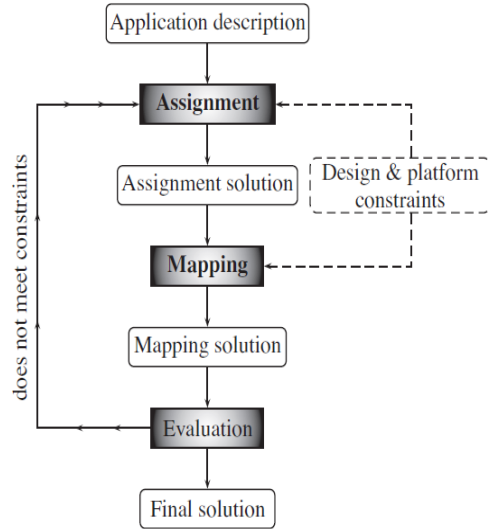


Fig 4. Typical platform based NoC design work flow

So, the result of this stage is the set of IPs that maximizes NoC performances due to IPs feature. The number of possible assignments is defined as

$$A = n_0 * n_1 * \dots * n_{m-2} * n_i \tag{1}$$

Where n_i represents the number of IPs, A represents the number of possible assignments m represents number of tasks in the application t_0, t_1, \dots, t_{m-1} .

The TG is then annotated and an application characterization graph (ACG) is produced, wherein each node (task) has an IP assigned to it.

B. IP mapping problem

It uses the result obtained from the assignment stage, which consists of many non-dominated solutions. Here we can consider s as the number of distinct assignments evolved and pi as the number of processors used in the assignment i and ni be the minimal number of resources in the NoC to be utilized in the implementation of the application with assignment solution i. In this stage, the total number of possible mappings is defined by the sum of all mappings corresponding to each one of the evolved assignments as described as

$$M_s = \sum_{i=1}^s \frac{n}{(n_i - 1)} \tag{2}$$

As a result of this stage, an optimal allocation of one of the prescribed IP assignments to execute a desired application on a NoC platform can be done.

C. MULTIOBJECTIVE EVOLUTIONARY ALGORITHM

In the computer science field of artificial intelligence, a genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate

solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover.

Evolutionary algorithms (EA's) are often more suited for optimization problems which involving several often conflicting objectives. Since 1985, various evolutionary approaches to multi objective optimization have been developed that are capable of searching for multiple solutions concurrently in a single run. However, few comparative studies of different methods presented up to now remain mostly qualitative and are often restricted to a few approaches.

Many real world problems involve simultaneous optimization of several incommensurable and often competing objectives. Often there is no single optimal solution, but rather a set of alternative solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all objectives are considered, which are known as pareto-optimal solutions. An optimal design might be an architecture that minimizes cost and power consumption while maximizing the overall performance. But these goals are generally conflicting: one architecture may achieve high performance at high cost, an alternative low cost architecture might considerably increase power consumption. All these solutions can be superior if we include preference information. In this paper we are using two evolutionary algorithms NSGA-II and microGA.

There are three different operators (reproduction, crossover, and mutation); similar to natural operations are applied to create a better population. Both the algorithms vary from simple GA only in the way the reproduction operator works. The crossover and mutation operators remain as usual. Before the selection is performed, the non-dominated individuals present in the population are assumed to constitute the first identified from the current population. Then all these individuals present in the population are assumed to constitute the first non-dominated front in the population and assigned a large dummy fitness value. The same fitness value is assigned to give an equal reproductive potential to all these non-dominated individuals.

In order to maintain the diversity in the population, the classified individuals are then shared with their dummy fitness values. After sharing, these non-dominated individuals are ignored temporarily to process the reset of population in the same way to identify individuals for the second non-dominated front. These new set of points are then assigned a new dummy fitness value which is kept smaller than the minimum shared dummy fitness of the previous front. This process is continued until the entire population is classified into several fronts. The population is then reproduced according to the dummy fitness values.

D. MATHEMATICAL MODELLING

To implement a higher level description of design stage the model used to evaluate the solution is quite crucial. Such an accurate mode will leads to a feasible implementations according o the goals and constraints of the design. The fitness of the individuals with respect to each one of the selected objectives like area, time and power must be efficiently computed based on appropriate models. So here we are presenting our first model is to

evaluate the assignment solutions and the second one is to be evaluate the mapping solutions.

E. Assignment Fitness Evaluation

For an assignment solution so, the fitness can be measured. On silicon area, the approximate execution time and the power consumption required, when the application is executed. In this part of evaluation only computation time and power due to computation can only be considered when the actual allocation when the IPs are mapped within the NoC resource nodes are known. Some of the characteristics are quantified below.

1) Hardware Area

To compute the required area, it is necessary to add up the area of each processor used in a given solution. A simpler method to identify the processor of solution is by grouping the nodes of same processor and identifying the nodes of dedicated processors. The following equation-3 shows how to compute the area of solution, wherein the function provides a set of non-dedicated processor used in this project.

$$Area(S_a) = \sum_{t \in TG} area(S_a[t])_{TF} * S_a[t]_{dedicated} \quad (3)$$

2) Execution Time

It is necessary to find the critical path of ACG, when we are in need to compute the execution time required by a solution. It can be found by visiting all nodes of all paths and recording the execution time of the slowest path. Sequential scheduling of the tasks must be done for the tasks that should be executed in parallel which are allocated in the same processor.

The following equation-4 shows the details of this computation. Wherein, the function returns all possible paths of the task graph 'G', function returns the set of all tasks in the ACG that may be executed in the parallel with task 't' and are associated with the same processor in the solution, function informs all the tasks that depends on the execution of and that are also allocated to the same processor in

$$Time(S_a) = \max_{M \in C(GT)} (\sum_{t \in M} time(S_a[t])_{TF} + T(S_a)) \quad (4)$$

3) Power Consumption

The power consumption of each IP must be added in-order to compute the power consumption of application. Equation-5 shows which represents the power consumption when a task is executed by its assigned IP in a solution.

$$Power(S_a) = \sum_{t \in TG} P_t \quad (5)$$

4) Mapping fitness evaluation

The fitness of the mapping solution can be measured in terms of the silicon area that would be used to implement the NoC-based application, the execution time and power consumption. Some of the characteristics are quantified below.

In order to compute the area required by a given mapping, we need to know about the area required for the processors used and by the channels and switches. The following equation-6 describes about the total area of a given mapping solution.

$$Area_m(S_m) = Area_A A_S + area_c * \sum_{d \in e(TG)} CH(S_m[d_{src}],_{res}, S_m[d_{tgt}],_{res}) + area_s * \sum_{d \in e(TG)} SW(S_m[d_{src}],_{res}, S_m[d_{tgt}],_{res}) \quad (6)$$

Here, the communication channel and switch areas are the represented as constants.

In the considered solution, the execution time is evaluated based on the prescribed mapping. The execution time of the application with respect to a given mapping solution will not be returned. There are some situations which will increase the execution time of the application while analyzing the mapping problem.

- (i) The communication channels which shares common source with parallel tasks.
- (ii) The communication channels which shares common target with parallel tasks.

The following equation-7 describes about the execution time considering the computation and communication for a given mapping solution.

$$Time(S_m) = \max_{Q \in C(TG)} (Time_p(Q) + Time_c(Q) + T'(Q)) \quad (7)$$

The power consumed due to processing and a communication should be necessarily considered while compute the power consumption of a given mapping is explained in equation-8.

$$Power(S_m) = Power_P(S_m) + Power_C(S_m) \quad (8)$$

Where,

$$Power_P(S_m) = \sum_{t \in TG} Power_{S_m[t]TF}$$

and

$$Power_C(S_m) = \sum_{t \in TG} Vol_{d(t,t')} * E_{bit}^{S_m[t]_{res}, S_m[t']_{res}}$$

IV SIMULATED OUTPUT

In this paper we have simulated the output of the router, which plays a key role in NoC interconnection. Thereby we can get a clear-cut knowledge about the process of a router. Here we are using Xilinx version 12.2 to simulate its output. From the output the input and the

output parameters can be analyzed easily. The functionality of the internal components in the buffer element can be briefly understood from the data types used in the stimulated output. The data types of the received output from the buffer element are logics, arrays and integer types. The width, depth and the threshold values are of integer type which can be a fixed value. Clock (clk), reset, data input (din), read and write values are given as inputs to the buffer element. The data output (dout) and the output of full adder and half adder will be arrayed data types. The full adder and half adder values are single logical datatype whereas din and dout are dual arrayed one.

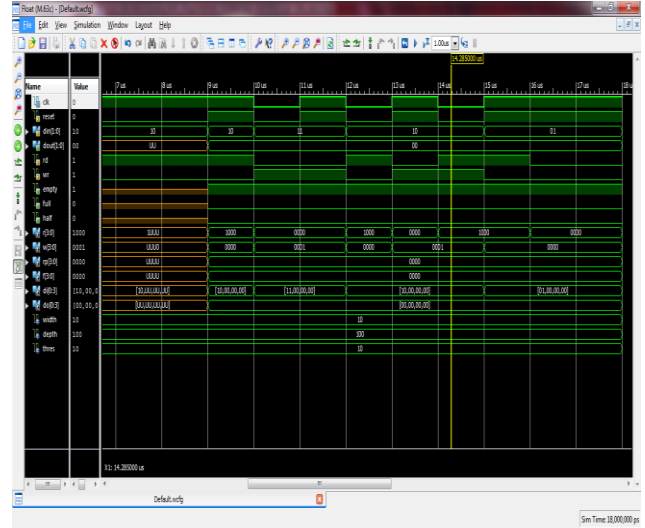


Fig 5. Simulated output of a buffer element

V CONCLUSION

The problem of assigning and mapping IPs are NP-hard problems and key research problems in NoC design field. The approaches have been then evaluated and compared, in terms of both accuracy and efficiency, on a platform based on an event-driven trace based simulator which makes it possible to take account of important dynamic effects that have a great impact on mapping. The network connectivity in our NoC is guaranteed by partitioning the entire links into two sets and keeping the connection of the links of one of the sets fixed. Simulation results showed the mechanism of how a buffer element plays its role in NoC communication. And we are going to design a NoC architecture which consumes less power and reduced average communication latency when compared to a conventional NoC. The work described in this paper can be used as a basis of any automatic synthesis tool of NoC-based embedded system. Of course, besides the assignment and mapping stages, this tool will be benefiting for the systematization of other stages of the synthesis process.

REFERENCES

- [1] Nadia Nedjah, Vinicius Carvalho da Silva, Luiza de Macedo Mourelle, Customized computer-aided application mapping on NoC infrastructure using multi-objective optimization, in: Journal of system architecture 57(2011) 79-94.
- [2] Tang Lei, Shashi Kumar, A two-step genetic algorithm for mapping task graphs to a network on chip architecture, in: DSD, IEEE Computer Society, 2003, pp.180-189.

- [3] Srinivasan Murali, Giovanni De Micheli, SUNMAP: a tool for automatic topology selection and generation for NoCs, in: Proceedings of the 41st Annual Conference on Design Automation (DAC-04), ACM Press, New York, 2004, pp. 914-919.
- [4] Srinivasan Murali, Giovanni De Micheli, Bandwidth-constrained Mapping of cores onto NoC Architectures, in: DATE, IEEE Computer Society, 2004, pp. 896-903.
- [5] Wenbiao Zhou, Yan Zhang, Zhigang Mao, Pareto based multi-objective mapping IP cores onto NoC architecture, in: Asia Pacific Conference on Circuits and Systems, IEEE Press, Los Alamitos, CA, 2006, pp. 331-334.
- [6] Rabindra Ku. Jena, Gopal Ku. Sharma, A multi-objective evolutionary algorithm based optimization model for network-on-chip synthesis, in: ITNG, IEEE Computer Society, 2007, pp. 977-982.
- [7] G. Ascia, V. Catania, M. Palesi, Multi-objective mapping for mesh-based NoC architectures, in: Second IEEE/ACM/IFIP international Conference on Hardware/Software Codesign and System Synthesis, ACM Press, New York, NY, 2004, pp. 182-187.
- [8] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6(2002) 182-197.
- [9] Jingcao Hu, Radu Marculescu, Energy-aware mapping for tile-based NoC architectures under performance constraints, in: Proceedings of the 2003 conference on Asia South Pacific design automation ACM, New York, NY, 2003, pp. 233-239.
- [10] N. Srinivas, Kalyanmoy Deb, Multi-objective optimization using non-dominated sorting in genetic algorithms, Evolutionary Computation 2 (1994) 221-248.
- [11] Carlos A. Coello Coello, Gregorio Toscano Pulido, A micro-genetic algorithm for multi-objective optimization, Lecture Notes in Computer Science 1993 (2001) 126-138.
- [12] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the theory of NP-completeness, W.H. Freeman, USA, 1979.
- [13] Shashi Kumar et. Al., "A Network on Chip Architecture and Design Methodology", IEEE Computer Society Annual Symposium on VLSI, Pittsburgh, Pennsylvania, USA, April 2002, pp. 117-124.