



Journal of Artificial Intelligence

ISSN 1994-5450

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

A Novel Custom Topology Generation for Application Specific Network-on-chip Using Genetic Algorithm Optimization Technique

M. Maheswari

Department of ECE, J.J. College of Engineering and Technology, Trichy-620009, Tamil Nadu, India

Corresponding Author: M. Maheswari, G2-Rohini Grand, AA-26, Sivaprakasam Salai, Anna Nagar, Thennur, Trichy-620 017, Tamil Nadu, India Tel: 91-8940573868

ABSTRACT

In Networks-on-chips (NoC), the main sources of power consumption are global interconnection links and routers. In Application Specific NoC (ASNoC) power can be minimized by mapping the cores on the application specific topology (custom topology) rather than mapping on the standard topologies. In ASNoC, the design of the topology plays an important role in minimizing the power consumption and hop count. In this study, we propose a novel topology generation algorithm using genetic algorithm optimization technique to generate a custom topology for ASNoC architectures. We applied the proposed algorithm to six benchmark video applications MPEG 4 decoder, VOPD, MWD, mp3 audio encoder, mp3 audio decoder and DSP. The proposed topology generation algorithm achieves significant amount of power saving and decrease in the average number of hop count compared to the existing custom topology generation algorithms.

Key words: NoC, custom topology, genetic algorithm, low power, FPGA, ASIC, ASNoC

INTRODUCTION

In the last years, with the technology scaling down, the number of processors, DSPs, memory elements and Intellectual Property (IP) cores on a System on Chip (SoC) is increasing more (Srinivasan *et al.*, 2005). This rapid increase in the number of components on a single chip has resulted in increased computation and communication complexity. Hence, today's SoC requires new on-chip communication medium that are energy efficient and have high performance (Benini and De Micheli, 2002). Networks-On-Chip (NoC) has come out as a feasible solution for designing energy efficient and high performance communication architectures, for Multi Processor SoC (MPSoC) (Lai and Lin, 2012). In NoC, the design of the topology affects the performance, power consumption and overall area of the on chip. Hence to minimize the power consumption, application specific topology is to be designed rather than mapping on the standard topologies like mesh, star. In modern Application Specific NoC (ASNoC), different communication requirement exists between different cores. For these ASNoCs, standard topologies would result in poor performance and large overhead of power consumption and area. This requires application specific topology which requires few resources like routers than standard topologies to minimize the area and power consumption (Lai and Lin, 2012).

In NoC, routers are placed to reduce the length of interconnect. IP cores communicate through routers. The number of routers the data bits travel to reach the destination determines the power consumption. The more the number of routers the data bits travel, the higher is the total power consumption. Hence, we must construct application specific topology in such a way that the number

of hops or routers the data bits travel to reach the destination is reduced (Lai and Lin, 2012). This is achieved by having high communicating cores placed close to each other. This creates shorter communication distance between high communicative cores. This reduces power consumption and increases the performance, as the message bits travel fewer routers to reach the destination (Deepak *et al.*, 2009). In this study, we propose a novel topology generation algorithm using genetic algorithm based optimization technique that is capable of generating custom topology to a specific application for reduction in the power consumption compared to regular topologies.

In recent years, many research works have been carried out in the design of the application specific custom topology. Lai and Lin (2012) used genetic algorithm for the generation of custom topology. The authors consider core to router mapping instead of core to port mapping considered Leary *et al.* (2009) in the initial population. This saves the search path. The authors (Lai *et al.*, 2010) use Dijkstra's shortest path algorithm to find the route in the final topology generation. Throughput oriented custom topology is generated by using two algorithms (Dumitriu and Khan, 2009) and their topology is compared with standard topologies like Mesh and Fat tree and another custom topology. Four various heuristics based algorithms are proposed (Deepak *et al.*, 2009) for generating the optimal tree based topology for multimedia applications. Power analysis and the energy consumption are performed using Orion power modeling. Murali and de Micheli (2004a) proposed mapping and routing of the cores onto NoC standard topology using unified approach. Pande *et al.* (2005) proposed an evaluation methodology to compare the performance and characteristics of a variety of standard NoC architectures like SPIN, CLICHÉ, Torus, Folded torus, Octagon, BFT. A prototype implementation of the NoC architecture for multimedia applications in FPGA and ASIC is carried out (Ogras *et al.*, 2007). The authors also evaluated the power and the area, both in FPGA and ASIC implementation. They do not attempt to evaluate and compare the performance of their custom NoC architecture with standard architectures. The SCOTCH partitioning tool is used to map complex application task graph onto four architecture (Bononi *et al.*, 2007) and analysis is performed by OMNET++ simulator. Chang and Chen (2008) proposed the power aware topology construction method which is used to construct Application Specific low power interconnection topologies based on the traffic characteristics of SoC. The authors analyze their low power algorithm using C++ based simulator and have port constrain for router and use only the four port router and their algorithm is compared with mesh topology only. Atienza *et al.* (2008) proposed a complete NoC Synthesis flow. They compare several multimedia applications with the standard mesh topology and their custom topology, in terms of area and power for nanometer regime. Elmiligi *et al.* (2009) proposed a partitioning algorithm to generate hybrid topology and apply for the MPEG 4 decoder. The authors analyze and compare the performances of the NoC like power, hop count and the number of links with the standard topologies and hybrid topologies. Ar *et al.* (2009) generate custom topology using port constraint router for benchmark video applications and they compare their work with the existing algorithm. The authors generate (Srinivasan *et al.*, 2005) custom topology for ASNoC using polynomial algorithms and applied for several benchmark video applications. Number of clusters is determined using Genetic Algorithm based optimization Technique (Vijendra *et al.*, 2012). An efficient and powerful design method for calculating optimal Proportional-integral-derivative (PID) controllers for AVR systems is proposed using genetic algorithm (Mohammadi *et al.*, 2009). Wei *et al.* (2007) proposed a hybrid routing protocol to reduce the delay in mesh network. A new GA based task scheduling technique is proposed (Akhtar, 2007) and is tested on a multi-node grid environment and the results show that this method can lead to significant performance gain in various applications network load balancing application.

Chiu (2010) proposed an efficient path selection algorithm using genetic algorithm. A cloud based manufacturing resources service problem is proposed (Ding *et al.*, 2012). Four ports routers are used by Chang and Chen (2008) and Ar *et al.* (2009) for topology generations of ASNoC, only 2 or 3 cores are connected to a single router. This constrain forces them to use more number of routers. In contrast to these works, in this study, we propose a three phase novel topology generation algorithm that uses a router in which the numbers of ports are parameterizable. In our proposed work the number of ports can be parameterized up to seven. The increase in the number of ports in a router from 4 ports to seven ports increases the router crossbar size which in turn increases the power consumption of the single router. Although the power consumed by the single router is increased the overall router power consumption in the topology is reduced. This is because, the number of routers used for the given applications is reduced compared to the works proposed (Chang and Chen, 2008; Ar *et al.*, 2009). To assess the efficiency of the proposed algorithm, we applied the proposed algorithm to few benchmark video applications namely MPEG 4 decoder, VOPD, MWD, mp3 encoder, mp3 decoder and DSP. The custom topology generated by the proposed algorithm uses lesser number of routers than the other custom topologies generated by the existing topology generation algorithms. This reduces area, power consumption compared to the previous works.

POWER ANALYSIS IN NoC

Power consumption is the main restriction factor in NoCs. In NoC, the power consumption is due to two components: (1) router and (2) the interconnect wires. These interconnect wires consume noteworthy amount of power (Vitkovski *et al.*, 2008). A brief power analysis is given for the NoC router and the interconnection links.

Power analysis of NoC router: In NoC router, power dissipation takes place on three components: (1) the internal node switches, (2) the internal buffers used to store the flits temporarily. (3) the interconnect wires inside the switch (Ye *et al.*, 2002).

The design of the network router (Murali and Gopalakrishnan, 2011) consists of four parts: (1) input port (2) arbitration unit (3) crossbar switch and (4) output port. The basic router unit is given in the Fig. 1. The input port has buffers to store the data and a header decoder unit to decode the

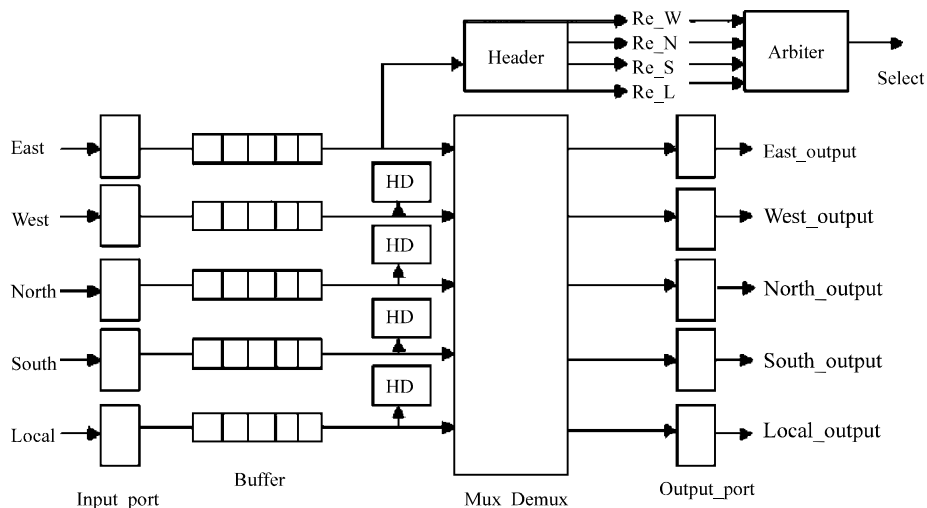


Fig. 1: Router architecture

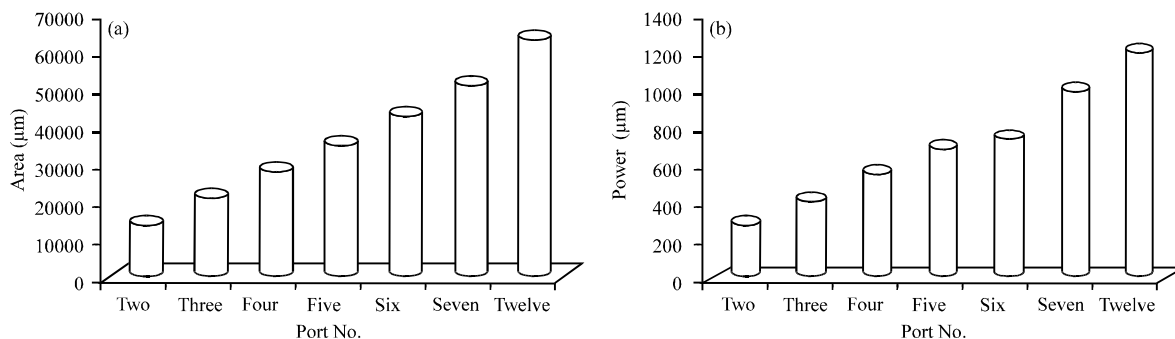


Fig. 2(a-b): Comparison of (a) Area and (b) Power of the router on TSMC 0.18 μm technology for different ports numbers

destination address. We employ wormhole flow control to reduce the switch buffering and to reduce the latency. We apply round robin arbitration. Multiplexer and De-multiplexer were used as crossbar switch. When the flits arrive at the input port, the flits are stored in the buffer and the header unit decodes the destination address from the header flit. It sends the request to the output port. When more than one input ports request the same output port, the arbiter in the output port, gives access to one input port at a time in round robin fashion. The input port that receives the access sends the flits through the crossbar switch to the output port. The packet size is 256 bits (16 flits each of 16 bit width). The input buffer size is 8 flits depth and wormhole switching is used.

We examined the area and the power consumption of the NoC router, by designing the five port router in Virolog HDL and implemented on TSMC 0.18 μm technology using cadence tool. Then we customized the router architecture, by changing the no of ports in the routers as 2, 3, 4, 5, 6, 7 and 12 based on the requirement of the topology and examined the area and the power consumption for all the router architectures.

The results of area occupancy and power consumption are shown in Fig. 2. When the number of ports in the router is increased, the area occupied and the power consumed by the router increases proportionally. This is because, increase in more ports to a single router results in larger decoding logic. The crossbar must handle increased number of interconnect which increases crossbar size. For example the crossbar switch for the router having number of ports up to four can be implemented in an efficient manner using a 4 to 1 MUX. However, for designing crossbar for the router having more than four ports, the crossbar must be designed using 8 to 1 MUX or combination of large number of smaller MUXs. This is the reason for the increase in the crossbar size of the router when number of ports per router is increased.

Power analysis of NoC links: The most important sources of power dissipation in on chip interconnections are: (1) the coupling capacitance between the two neighboring wires and (2) the capacitance between wire and ground (Vitkovski *et al.*, 2008). In NoC Platform, two distinct links are considered. They are: (1) The link between routers to router (Global link) and (2) The link between PE to router (semi global link). The power consumption on interconnection wires between two neighbor routers is given as in Eq. 1:

$$P_{\text{dyn}} = \frac{1}{2} \alpha C f V_{\text{dd}}^2 \quad (1)$$

where, C is the interconnection wire capacitance, α is the average switching activity in the wires, f is the frequency, V_{dd} is the supply voltage (Vitkovski *et al.*, 2008). The switching activity α is defined as the ratio of the number of bits changing from 0>1 or 1>0 to the total number of bits, passed from one end to another end of a interconnection wire over a period of time. The values of C , V_{dd} and f depend on the process technology. But, the value of switching activity α depends on the data passed through the wires. Hence, power saving is achieved by reducing the switching activity on the wires (Vitkovski *et al.*, 2008).

PROPOSED METHODOLOGY

The overall objective of this novel topology generation algorithm is to generate a custom topology for ASNoC to reduce the power consumption and to reduce the hop count. The proposed algorithm can be used at the early stage of the topology design. The idea for generating low power custom topology is to form clusters of the cores. To do this, the cores are clustered based on the communication volume (traffic) between them. The cores that have larger communication volume (traffic) are grouped to the same cluster to form localization. This localization creates shorter communication distance between the cores that have larger communication volume and reduces the average number of hops between sources to destination. We connect one router for one cluster. The number of ports in the router is selected based on the number of cores in the cluster. When a core is connected to a router, the data enters through one of the input ports of the router and leaves one of the output ports of the routers. The two ports are considered as one hop. When larger communication cores are connected to a single router it minimizes the hop count, this in turn minimizes the total energy consumption. Hu and Marculescu (2003a) proposed the average energy dissipation to send one bit of data from tile v_i to v_j is:

$$E_{\text{bit}}^{i, v_j} = n_{\text{hops}} E_{\text{sbit}} + E_{\text{lbit}} (n_{\text{hops}} - 1) \quad (2)$$

where, E_{sbit} is the Energy consumption of a router, E_{lbit} is the energy consumption of interconnecting wires, n_{hops} is the number of routers the bit passes.

To formulate this problem more formally, we propose the following definitions as proposed by Murali *et al.* (2006).

Definition 1: The Communication Task Graph (CTG) is an undirected Graph, $G(V, E)$, where each vertex $v_i \in V$ represents a core and the directed edge $w_{ij} \in E$ represents the communication volume between two cores v_i and v_j .

Definition 2: The Router Communication Graph (RCG) is a fully connected graph with m vertices, where m is the number of clusters. The edge weights are set to zero.

The proposed topology generation algorithm can generate topology for ASNoC with heterogeneous routers in which the number of input/output ports is parameterizable. In the proposed topology generation algorithm, the router ports can be parameterizable up to seven. The number of input/output ports in a particular router in the final topology depends on the number of cores in the clusters. The bandwidth constrain of the router Br is assumed to be 3200 MB sec^{-1}

(assuming the router is operated at 200 MHz and the port width is 16 bit). We prefer the cluster to have number of cores less than or equal to six. Sethuraman (2007) used multi local port router to reduce the number of routers used in the topology. The use of multi local port router increases the router performance and reduces the number of routers used in the topology which in turn reduces the overall power consumption in the topology. However, when the number of ports in the router is increased largely, the router size will be bulkier as it has to handle large amount of data and in turn the power consumption of a single router will be increased. Hence, to get better performance of the router and at the same time only small increase in the router size, we select the routers to have number of ports less than or equal to seven. Leaving one port for interconnection, we select the cluster size is less than or equal to 6 cores. For example, if the number of cores in a cluster is 3, we use 4 ports router for that cluster and if the number of cores in a cluster is 6, we use 7 ports router for that cluster. Therefore the proposed algorithm can support heterogeneous router architectures. Table 1 compares the overall router power consumption for the proposed algorithm with the previous works. The power consumption values are obtained from TSMC 0.18 μm technology. The proposed algorithm uses routers up to seven ports. But the works (Chang and Chen, 2008; Ar *et al.*, 2009) use only four ports routers. As shown in the table, a single seven port router has more power consumption than a single four port router. But the overall router power consumption is reduced in the topology as the number of routers required is reduced when seven ports router is used. In the following sub section, we present the proposed three phase topology generation algorithm.

The proposed algorithm: The proposed topology generation algorithm has three phases. They are: (1) Construction of clusters (2) optimization of clusters using GA (3) topology generation. The proposed algorithm can be used in the early stage of the topology design.

Topology generation algorithm:

Input : Communication Task Graph (CTG), $G(V,E)$

Output : Custom topology

Phase I: Construction of clusters:

Step 1: Arrange the communication volume (or) traffic between the cores (edge weights) in descending order. Find out the number of clusters $m \left\lceil \frac{n}{6} \right\rceil$

Step 2: Assign:

E_{\max} = Maximum communication volume

E_{\min} = Minimum communication volume

Table 1: Comparison of the total router power consumption for MPEG 4 decoder

References	7 port	6 port	5 port	4 port	3 port	2 port	Total router power (μW)
Chang and Chen (2008)	-	-	-	5	-	-	2857.9
Ar <i>et al.</i> (2009)	-	-	-	5	-	-	2857.9
Srinivasan <i>et al.</i> (2005)	2	--	--	-	-	3	2859.6
Proposed algorithm	2	--	--	-	-	-	1998.9

Step 3: Find upper limit communication volume for each cluster

- (i) for $x = 1; x \leq m; x++$ $U_limit[x] = \frac{E_{max, x}}{m}$
- (ii) Assign the cores that have communication volume in the range $(U_limit[x-1]+1:U_limit[x])$ to cluster m to cluster 2 as shown below:
for $x = m; x \leq 2; x--$
cluster $[x] = (U_limit[x-1]+1:U_limit[x])$
- (iii) Assign the cores that have communication volume in the range $(E_{min}:U_limit[1])$ for cluster 1 as shown Below. If the cores are already assigned in one cluster neglect it.

$$cluster[1] = (E_{min}:U_limit[1])$$

Phase II: Cluster optimization using GA and generation of RCG:

Step 4: The initial clusters formed creates unbalanced traffic in the clusters. Hence, the clusters are optimized using GA such that

$$S^x \leq B_r \tag{3}$$

Where:

$$S^x = \sum_{ij \in x} W_{ij}^x, x = 1, \dots, m$$

W_{ij}^x is the communication volume between the cores v_i and v_j in the x th cluster. B_r is the bandwidth constraint of the router and m is the number of clusters:

$$\text{minimize} \sum_{x=1}^m d_x \tag{4}$$

Such that unbalanced traffic is avoided among the routers.

Where:

$$d_x = S^x - S^{x+1}$$

If $x+1 > m$, then $x+1 = 1$:

$$S^x = \sum_{ij \in x} W_{ij}^x$$

$$S^{x+1} = \sum_{ij \in x+1} W_{ij}^{x+1}$$

Where:

W_{ij}^x = The communication volume between the cores v_i and v_j in the x th cluster

$\sum_{ij \in x+1} W_{ij}^{x+1}$ = The communication volume between the cores v_i and v_j in the $x+1$ th cluster

$$x = 1 \dots m$$

where, m is the number of clusters.

Then to generate RCG, the routers with different number of ports are selected based on the number of cores present in the clusters and the number of clusters. Finally the routers are interconnected to form RCG.

Phase III: Topology generation:

Step 5: The final topology is generated by connecting the cores in the clusters to the ports of the routers in the RCG such that each core is connected to only one port of the router

Phase I: Construction of clusters: In the first phase of the proposed algorithm, clustering of cores is completed. The number of clusters is decided based on the number of cores in the application. In the proposed algorithm, n defines the number of cores in the CTG, m defines the number of clusters. The main idea behind the construction of the cluster is to group the cores that have larger communication volume (traffic) in the same cluster. This is done by arranging the communication volume in descending order and grouping them into the clusters. Construction of the clusters is shown in step 1 to step 3 of the proposed algorithm.

Phase II: Optimization of the clusters: In the second phase of the proposed algorithm, optimization of cores is done and RCG is built. The clusters are optimized using Genetic Algorithm (GA) based optimization technique.

GA based optimization: A GA is based on the biological fact of the genetic evolution. It keeps a set of solutions known as populations. GA operates in an iterative manner and develops to new solutions from the current solutions by the application of genetic operators (Leary *et al.*, 2009). Our GA applies crossover and mutation to produce new solutions. In crossover two solutions are combined to generate a new solution and in mutations an existing solution is modified to generate a new solution. Application of two point crossover in cluster array is shown in Fig. 3.

Population representation: GA based optimization technique requires a representation of the population for the application of genetic operators (Leary *et al.*, 2009). We set the population size is equal to the number of cores ‘n’ multiplied by the number of clusters ‘m’. The total population size is divided equally among the clusters. In each cluster the population is represented as strings of

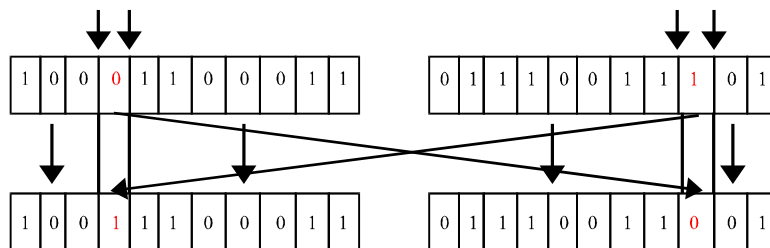


Fig. 3: Crossover in cluster array

chromosomes in an array and is called as cluster array. We form ‘m’ cluster arrays. The cluster array is a binary array where a ‘1’ in a location denotes the presence of the core in that cluster and a ‘0’ denotes that the core is not present in that cluster. For example in a cluster array a ‘1’ in 5th location represents the core 5 is present in that cluster and a ‘0’ in 9th location denotes core 9 is not in that cluster.

Criteria for solution selection: Since a cluster has number of cores less than or equal to six, the criteria to be satisfied in the cluster array is that the number of 1 sec in the cluster array is less than or equal to six. We apply two point cross over to produce the solutions. Only the solutions that satisfy the above criteria are considered for the fitness calculation. The fitness constraints are given by Eq. 3 and 4.

Once the clusters are optimized, the RCG is built by selecting the routers based on the number of cores in the clusters and the number of clusters. The router ports are determined based on the number of cores in the cluster and the number of routers is determined based on the number of clusters. The selected routers are interconnected to build RCG.

Phase III (topology generation): In the third phase, final topology is generated by connecting the cores in the clusters to router ports in RCG and then interconnecting the routers. The main goal of the proposed algorithm is to minimize the power consumption and the hop count.

The custom topology generated by the proposed algorithm minimizes the number of hops the bit travels to reach the destination core and the energy consumed by a single bit which in turn minimizes overall energy consumption as given in Eq. 2. Overall energy consumption is minimized because most of the bits travel within the router.

The proposed algorithm also reduces the number of routers used in the topology. This in turn reduces area and the power consumption of the overall topology design. We apply the proposed algorithm to six benchmark video applications MPEG 4 decoder, VOPD, MWD, mp3 audio encoder, mp3 audio decoder and DSP. We will explain the steps of the proposed algorithm for benchmark video application MPEG 4 decoder. Figure 4 shows the CTG for MPEG 4 decoder.

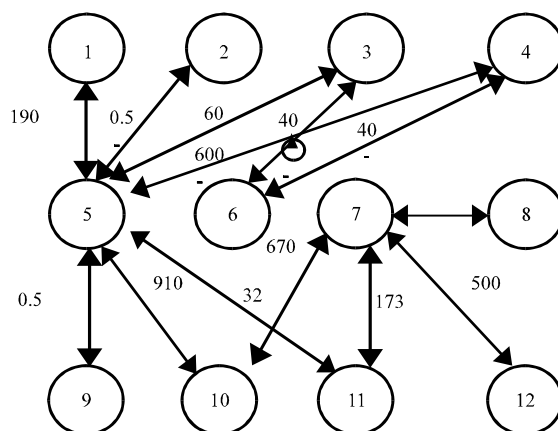


Fig. 4: CTG for MPEG 4 video decoder. Edge values are communication requirement given in Mb sec⁻¹

Table 2: Arrangement of communication volume in descending order

Cores	Communication volume (MB sec ⁻¹)
5-10	910
7-10	670
5-4	600
7-12	500
8-7	250
1-5	190
7-11	173
5-3	60
5-11	32
3-6	40
4-6	40
2-5	0.5
5-9	0.5

Table 3: Initial cluster formation

Clusters	Cores in clusters
Cluster 1	8, 1, 11, 3, 6, 2,9
Cluster 2	5, 10, 7, 4, 12

Table 4: Optimized clusters

Clusters	Cores in clusters
Cluster 1	1, 2, 3, 7, 10, 12
Cluster 2	4, 5, 6, 8, 9, 11

Step 1-2: The first step of the proposed methodology is to arrange the communication volumes (or) traffic between the cores of the CTG in descending order as shown in Table 2. For MPEG 4 decoder, the number of cores n is 12 and the number of cluster m is $\left\lceil \frac{12}{6} \right\rceil = 2$ (since we have maximum of 6 cores in a cluster). From the Table 1 we assign $E_{max} = 910 \text{ MB sec}^{-1}$ and $E_{min} = 0.5 \text{ MB sec}^{-1}$

Step 3: In step 3, we find out the values for the array $U_limit[x]$ using $U_limit[x] = \frac{E_{max} \cdot x}{m}$. Here, $x = 1, 2$. Hence, $U_limit[1] = 455 \text{ MB sec}^{-1}$ and $U_limit[2] = 910 \text{ MB sec}^{-1}$. The cores that communicate with communication volume in the range $U_limit[1]+1$ to $U_limit[2]$ are grouped in cluster 2 and cores that communicate with communication volume in the range E_{min} to $U_limit[1]$ are grouped in cluster 1. Table 3 shows the initial cluster formation

Step 4: In step 4, we optimize the clusters using genetic algorithm optimization technique. Crossover and mutation are applied to the cluster array to produce the new solutions. The cluster array is formed based on the presence of the core in the cluster. The solution that satisfies the above said fitness function is selected as the optimized cluster. In MPEG 4 decoder application, there are 12 cores present. Two clusters are formed and each cluster has 6 cores. Table 4 shows the optimized cluster. We use seven ports router for both the clusters. The RCG is shown in Fig. 5

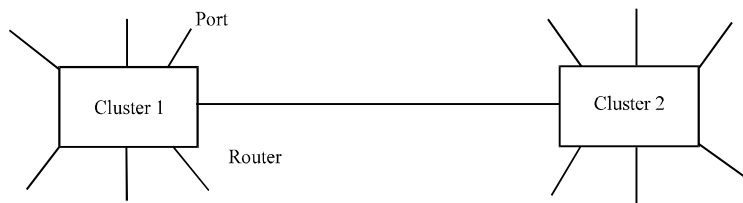


Fig. 5: Router communication graph (RCG) for MPEG 4 decoder

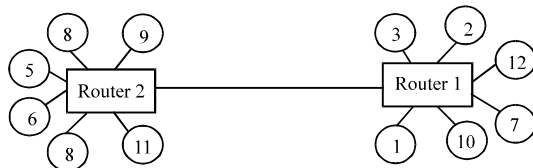


Fig. 6: Custom topology generated for MPEG 4 decoder by the proposed algorithm

Step 5: The cores in the clusters are connected to the ports of the routers in RCG and final topology is generated. The generated topology for MPEG 4 decoder application is shown in Fig. 6.

RESULTS

We present the results obtained by the application of the proposed algorithm on benchmark video applications. We first give details about the benchmark video applications, the experimental setup and finally we present the results.

Benchmark applications: We have applied the proposed custom topology generation algorithm for two multimedia applications namely mp3 audio encoder, mp3 audio decoder. We obtained the CTG for these applications from Hu and Marculescu (2003b). In addition to that we have also applied the proposed algorithm to four benchmark video applications namely MPEG 4 decoder, VOPD, MWD and DSP. The CTG for these applications were obtained from (Jalabert *et al.*, 2004) and the CTG for DSP is obtained from (Murali and de Micheli, 2004b).

Experimental setup: We used the power consumption model for 100 nm technology given (Srinivasan *et al.*, 2005). The power consumption for the input port, output port and for the link are estimated to be 328, 65.5 and 79.9 nW/Mbps, respectively. We estimated the power consumption for the custom topology generated by applying the proposed algorithm for the above said applications using the model given (Srinivasan *et al.*, 2005).

We first applied the proposed algorithm for the benchmark applications to generate the custom topologies. We compare the power consumption for the custom topologies generated by applying the proposed algorithm with the existing topology generation algorithms. The results of the proposed algorithm and the existing topology generation algorithms are given in Table 5. The proposed algorithm achieves significant power saving compared to the existing algorithms. We also compare the average hop counts of the topology generated by the proposed algorithm with the average hop

Table 5: Comparison of power consumption for benchmark applications'

Applications (1)	Nodes (2)	Power consumption (μ W)					Average number of hops	
		Proposed algorithm (3)	Other studies (4)	Ratio (4/3)	Other studies (5)	Ratio (5/3)	Mesh	Proposed algorithm
VOPD	12	1753.56	2611.1 (Srinivasan <i>et al.</i> , 2005)	1.48	3001.9 (Ar <i>et al.</i> , 2009)	1.71	1.33	0.25
MWD	12	610.63	993.6 (Srinivasan <i>et al.</i> , 2005)	1.62	3001.9 (Ar <i>et al.</i> , 2009)	2.18	1.38	0.31
MPEG 4	12	1626.32	7392.18 (Srinivasan <i>et al.</i> , 2005)	4.54	1853.86 (Dumitriu and Khan, 2009)	1.13	1.38	0.53
Mp3 encoder	8	4.16	6.48 (Srinivasan <i>et al.</i> , 2005)	1.55	5.7 (Srinivasan <i>et al.</i> , 2006)	1.37	1.33	0.223
Mp3 decoder	5	2.59	2.622 (Srinivasan <i>et al.</i> , 2006)	1.01	-	1	0.25	
DSP	6	1036	1686 (Srinivasan <i>et al.</i> , 2005)	1.62		1.5	0.625	
Average						2.07	0.36	

count of the standard mesh topology for all the benchmark applications. We achieve 5.75x improvement in the average hop count compared to standard mesh topology.

CONCLUSION

In this study, we presented a novel custom topology generation algorithm using genetic algorithm optimization technique to generate the custom topology for application specific Network on chips. There are three phases in the proposed topology generation algorithm. They are: (1) Construction of clusters (2) Optimization of clusters and (3) Topology generation. In the first phase, we group the cores that communicate with larger communication volume (traffic) in the same cluster. In the second phase, the clusters are optimized using genetic algorithm optimization technique such that each cluster has number of cores less than or equal to six, bandwidth constraint of the routers are satisfied and the traffic among the clusters are balanced. Then RCG is built. In the final phase, custom topology is generated by connecting the cores in the cluster to the router ports and interconnecting the routers. The main goal of the proposed algorithm is to minimize the power consumption and the average hop count. We evaluated the performance of the proposed algorithm for six benchmark applications namely MPEG 4 decoder, VOPD, MWD, mp3 audio encoder, mp3 audio decoder, DSP by comparing with the existing topology generation algorithm. The proposed algorithm achieves significant amount of power reduction compared to the existing algorithm and average hop count compared to the standard mesh topology.

REFERENCES

- Akhtar, Z., 2007. Genetic load and time prediction technique for dynamic load balancing in grid computing. *Inform. Technol. J.*, 6: 978-986.
- Ar, Y., S. Tosun and H. Kaplan, 2009. TopGen: A new algorithm for automatic topology generation for network on chip architectures to reduce power consumption. *Proceedings of the International Conference on Application of Information and Communication Technologies*, October 14-16, 2009, Baku, Azerbaijan, pp: 1-4.

- Atienza, D., F. Angiolini, S. Murali, A. Pullini, L. Benini and G. de Micheli, 2008. Network-on-chip design and synthesis outlook. *Integr. VLSI J.*, 41: 340-359.
- Benini, L. and G. De Micheli, 2002. Networks on chips: A new SoC paradigm. *IEEE Comput.*, 35: 70-78.
- Bononi, L., N. Concer, M. Grammatikakis, M. Coppola and R. Locatelli, 2007. NoC topologies exploration based on mapping and simulation models. *Proceedings of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools*, August 29-31, 2007, Lubeck, Germany, pp: 543-546.
- Chang, K.C. and T.F. Chen, 2008. Low-power algorithm for automatic topology generation for application-specific networks on chips. *IET Comput. Digital Tech.*, 2: 239-249.
- Chiu, M.C., 2010. Numerical assessment of path planning for an autonomous robot passing through multi-layer barrier systems using a genetic algorithm. *Inform. Technol. J.*, 9: 1483-1489.
- Deepak, M., A. Pasalapudi and K. Yalamanchili, 2009. Low energy tree based network on chip architectures using homogeneous routers for bandwidth and latency constrained multimedia applications. *Proceedings of the 2nd International Conference on Emerging Trends in Engineering and Technology*, December 16-18, 2009, Nagpur, India, pp: 358-363.
- Ding, B., X.Y. Yu and L.J. Sun, 2012. A cloud-based collaborative manufacturing resource sharing services. *Inform. Technol. J.*, 11: 1258-1264.
- Dumitriu, V. and G.N. Khan, 2009. Throughput-oriented NoC topology generation and analysis for high performance SoCs. *IEEE Trans. Very Large Scale Integr. Syst.*, 17: 1433-1446.
- Elmiligi, H., A.A. Morgan, M.W. El-Kharashi and F. Gebali, 2009. Power optimization for application-specific networks-on-chips: A topology-based approach. *Microprocess. Microsyst.*, 33: 343-355.
- Hu, J. and R. Marculescu, 2003a. Energy-aware mapping for tile-based NoC architectures under performance constraints. *Proceedings of the ASP-DAC Asia and South Pacific Design Automation Conference*, January 21-24, 2003, Kitakyushu, Japan, pp: 233-239.
- Hu, J. and R. Marculescu, 2003b. Exploiting the routing flexibility for energy/performance aware mapping of regular NOC architectures. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, March 3-7, 2003, Munich, Germany, pp: 688-693.
- Jalabert, A., S. Murali, L. Benini and G. de Micheli, 2004. *xpipesCompiler*: A tool for instantiating application specific networks on chip. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, Volume 2, February 16-20, 2004, Paris, France, pp: 884-889.
- Lai, G. and X. Lin, 2012. Floorplan-aware application-specific network-on-chip topology synthesis using genetic algorithm technique. *J. Supercomput.*, 61: 418-437.
- Lai, G., X. Lin and S. Lai, 2010. GA-based floorplan-aware topology synthesis of application-specific network-on-chip. *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems*, Volume 2, October 29-31, 2010, Xiamen, China, pp: 554-558.
- Leary, G., K. Srinivasan, K. Mehta and K.S. Chatha, 2009. Design of network-on-chip architectures with a genetic algorithm-based technique. *IEEE Trans. Very Large Scale Integr. Syst.*, 17: 674-687.
- Mohammadi, S.M.A., A.A. Gharaveisi and M. Mashinchi, 2009. A novel fast and efficient evolutionary method for optimal design of proportional integral derivative controllers for automatic voltage regulator systems. *Asian J. Applied Sci.*, 2: 275-295.

- Murali, M. and S. Gopalakrishnan, 2011. Design and implementation of low complexity router for 2D mesh network on chip using FPGA. Proceedings of International Conference on Embedded System Application, July 18-21, 2011, Las Vegas, NV., USA., pp: 1-6.
- Murali, S. and G. de Micheli, 2004a. Bandwidth-constrained mapping of cores onto NoC architectures. Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Volume 2, February 16-20, 2004, Paris, France, pp: 896-901.
- Murali, S. and G. de Micheli, 2004b. SUNMAP: A tool for Automatic Topology Selection and Generation for NOCs. Proceedings of the 41st Design Automation Conference, July 7-11, 2004, San Diego, CA., USA., pp: 914-919.
- Murali, S., P. Meloni, F. Angiolini, D. Atienza and S. Carta *et al.*, 2006. Designing application-specific networks on chips with floorplan information. Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, November 5-9, 2006, San Jose, CA., USA., pp: 355-362.
- Ogras, U.Y., R. Marcillescu, H.G. Lee, P. Choudhary, D. Marculescu, M. Kaufman and P. Nelson, 2007. Challenges and promising results in NOC prototyping using FPGAs. IEEE Micro, 27: 86-95.
- Pande, P.P., C. Grecu, M. Jones, A. Ivanov and R. Saleh, 2005. Performance evaluation and design trade-offs for network-on-chip interconnect architectures. IEEE Trans. Comput., 54: 1025-1040.
- Sethuraman, B., 2007. Novel methodologies for efficient networks-on-chip implementation on reconfigurable devices. Ph.D. Thesis, Computer Science and Engineering, University of Cincinnati, Cincinnati, USA.
- Srinivasan, K., K.S. Chatha and G. Konjevod, 2005. An automated technique for topology and route generation of application specific on-chip interconnection networks. Proceeding of IEEE/ACM International Conference on Computer-Aided Design, November 6-10, 2005, San Jose, CA., USA., pp: 231-237.
- Srinivasan, K., K.S. Chatha and G. Konjevod, 2006. Linear-programming-based techniques for synthesis of network-on-chip architectures. IEEE Trans. Very Large Scale Integr. Syst., 14: 407-420.
- Vijendra, S., K. Ashiwini and S. Laxman, 2012. A fast evolutionary algorithm for automatic evolution of clusters. Inform. Technol. J., 11: 1409-1417.
- Vitkovski, A., A. Jantsch, R. Lauter, R. Haukilahti and E. Nilsson, 2008. Low-power and error protection coding for network-on-chip traffic. IET Comput. Digital Tech., 2: 483-492.
- Wei, D., H.A. Chan and M.B. Rawoot, 2007. Hybrid routing protocol to decrease delay and to extend lifetime for mesh networks. Inform. Technol. J., 6: 518-525.
- Ye, T.T., L. Benini and G. de Micheli, 2002. Analysis of power consumption on switch fabrics in network routers. Proceedings of the 39th Design Automation Conference, June 10-14, 2002, New Orleans, LA., USA., pp: 524-529.