ISSN: 2395-1680 (ONLINE)
DOI: 10.21917/ijme.2023.0257

OPTIMISING SYSTEM-ON-CHIP ARCHITECTURE USING ASYNCHRONOUS REGRESSION MODEL

P. Kannan[1], P.M. Sithar Selvam[2], T. Priya[3] and S. Murali[4]

[1]*Department of Electronics and Communication Engineering, Francis Xavier Engineering College, India*
[2]*Department of Mathematics, RVS School of Engineering and Technology, India*
[3]*Department of Mathematics, NPR College of Engineering and Technology, India*
[4]*Department of Mathematics, Coimbatore Institute of Technology, India*

Abstract

This paper presents the results of a study that utilises genetic algorithms to optimise the scheduling of tests for a core-based system-on-chip (SoC) system. The research aims at reducing the amount of time that is required to test a SoC system, as well as ensuring that a system meets the requirements of the design. The findings show that it is possible to locate networks that have been carefully tailored to meet the requirements outlined by the RT packets with relation to latency, area, and deadline. The results show that the irregular networks with heterogeneous routers fared much better than the mesh networks when it comes to performance and compliance with the design requirements.

Keywords:

System-On-Chip, Asynchronous Regression Model, Quality of Service, Scalability

1. INTRODUCTION

One capacitor, one transistor, and three resistors were all that were needed to construct the very first integrated circuit ever created. Integrated circuits (ICs) now have architectures known as System-on-Chip (SoC), which comprise billions of transistors. These advancements came about because of breakthroughs in the technology used to fabricate electronic components. The SoC consists of more than a million gates, as well as cores, memory, and both digital and analogue blocks, all of which are packed onto a single chip.

When designing and manufacturing a large-scale integrated circuit, one of the most important challenges that must be surmounted is to find a solution to the issue of temperature control, as well as the problem of power consumption. Because of the high switching activities that are inherent to testing, it is recommended that more time and energy be spent in test mode as opposed to regular mode [1]. This is because testing requires more switching activities than normal mode.

Utilising modular testing of embedded cores is one solution to the testing challenge posed by SoCs. Another solution is to test embedded cores individually. These cores are complex building pieces that have previously been established, which makes design reuse simpler and, therefore, minimises the amount of time that is required for the development of the product. Nevertheless, the production tests and the debugging of those designs are the components of creating SoCs that present the greatest amount of difficulty [2].

The design of the test architecture is required because of this to gain access to the test resources and cores. Thanks to modular testing, SoCs that are outfitted with an Automatic Test Equipment (ATE) Test Access Mechanism (TAM) and embedded cores can communicate with external ATE [3].

Isolating embedded cores is a required step in modular testing, as is getting access to the individual SoCs that will be utilised for the transmission of test data to complete the testing procedure. This method entails the pushing of pre-computed test sets of embedded cores and is performed as part of a divide and conquer strategy.

The modular testing process is driven by these fundamental components [4]. If adequate effort is invested in advance constructing the test infrastructure and organising the tests themselves, it is possible to significantly reduce the cost of the overall testing process. The objective function for this examination is shown by the Eq.(1) that is presented below.

$$T(W_i) = [1 + \max(S_i,S_o)].tp_i + \min(S_i,S_o) \quad (1)$$

The equation is solved by entering the output and input lengths of the scan chain, which are denoted by S_o and S_i, respectively, as well as the test pattern, which is denoted by tp_i. Presentations of benchmark circuits for evaluating SoCs [5].

The scheduling of tests is a fundamental component of system-on-chip (SoC) test automation. A well-thought-out testing strategy will cut expenses and quicken the rate at which the chip will be made available to customers [5]. The proposed approaches offer the additional advantage of lowering test expenses, which is accomplished by cutting down on the amount of time spent on testing, which in turn cuts down on the amount of time spent on testing.

In this article, we discuss regression strategy for scheduling tests with the intention of cutting down on the amount of time as well as the amount of money that is necessary.

2. RELATED WORKS

An Undefined Topology Network on Chip (UTNoC) is described in [6], which may be found here. Each router in this network can interact with any other topology in the system, and each topology can communicate with a single processor. Additionally, each processor can communicate with any other router in the network. Tables are utilised in the process of routing, and the values contained in such tables are distributed to various locations. Every router in the UTNoC network can communicate with every other router in the topology. Every router in the network can have a sizable number of ports.

Experiments were performed on the UTNoC network utilising a simulator constructed with SystemC. to optimise the network and get results that are comparable to those of the application graph, the tool makes use of a genetic algorithm to cut ties between the routers. This allows the tool to accomplish the desired outcomes. Because of this, the technology can generate findings that are more precise [7].

Even though this will result in a minor increase in latency, the findings indicate that a reduction in the number of connections can provide an irregular topology with performance that is comparable to that of the application graph. This is the case even though this will cause a slight increase in the amount of time it takes for connections to be established.

As a potential component of their solution, the authors of [8] recommend applying irregular topologies in the context of a specific setting. After the conclusion of this study, the target of developing a fault-tolerant irregular topology generator will have been successfully attained. If a communication link between two routers fails, the Tabu search algorithm is the one responsible for determining alternate paths for a packet to take.

We tested a variety of topologies about both their latency and their dependability. Each answer was produced in an automatic method using the data from a task graph as the source information. During the testing, both simulated and real-world programmes (MPEG and VOPD) were utilised, with a total of four instances of each being used for the respective programmes.

It is possible to construct architectural designs that are fault tolerant as well as reliable. When there was a higher degree of performance achieved by the number of connections, it was found that the estimated network latency went down. As more bugs were added, there was an increase in the total number of connections that existed between the various issues. Solutions that had more connections than the minimum necessary for a ring topology were able to anticipate delay while also being able to endure up to thirty percent of random link fault injection. This was possible because the solutions had more connections than the minimum necessary.

In [9] proposes an innovative strategy for the design of non-standard topologies that are flexible enough to meet the requirements of various real-time applications. The purpose of the idea is to eliminate barriers in network construction while maintaining reasonable prices.

In embedded real-time systems that are built on Networks-on-Chips, there is a lot of support for implementing an end-to-end strategy for co-design. This is something that has been advocated for. Following the presentation of a task graph as the first step in the process of designing the system, the next step is co-synthesis, which is followed by cost minimization as the final step in the process.

A task graph that contains annotations has been constructed as the final product. The scheduling of communications allows for both the successful removal of contention at ports and the resolution of path-based conflicts. Both goals can be achieved simultaneously. to do this, non-overlapping pathways have been designated for competing communications. The architecture of the NoC was developed with the goal of avoiding the temporal restrictions imposed by the system while also retaining as much of the available electricity and other resources as is reasonably possible [10].

To consider the time constraints that are imposed by the system, the a priori knowledge of the communication pattern is utilised in the process of computing and transmitting schedules, as well as carrying out routing. This is done to ensure that data is transmitted in a timely manner. The operation of the system did not suffer in any way, and as a direct result of this, many resources were saved.

The purpose of the study cited in the number [11] is to build non-typical topologies for NoCs that are resilient enough to withstand faults. to fulfil the requirements of real-time applications, the current challenge entails the creation of irregular topologies that are intended to reduce the likelihood of temporary system failures. This is done to meet the requirements for apps that run in real time. to find a topology that is best for a wide variety of fault injection scenarios, an evolutionary technique is utilised. This allows for the discovery of an irregular topology. The subsequent phase, which occurs after the topology design of the network has been finished, is to link the routers together.

A fault injector can randomly disperse up to 15% of faults across the links when an irregular topology is developed. This prevents routers from communicating with one another. to carry out the task mapping while simultaneously working to decrease latency, a second genetic algorithm is utilised.

The shortest path algorithm is used to determine the connection between two routers that will take the least amount of time and demand the least amount of effort to complete. If there is a problem along the path that has the shortest distance between two points, the algorithm will go on to the subsequent option that is the best alternative. Because they were able to meet the deadline with a delay of less than 1%, it was proved that the developed topologies are suitable for real-time packet delivery. There was up to 13% greater delay than was first anticipated, on average.

In this section, an optimisation strategy is covered, with a particular emphasis placed on hard real-time application concerns. The purpose of this method is to raise the percentage of packets whose due dates are met while simultaneously reducing the average amount of delay that is caused by the network. An algorithm is used to alter the router mapping to identify an asymmetric topology that improves the rates at which packets are delivered and reduces the amount of time that is typically spent waiting in the NoC.

To expedite the delivery of packets, a comparison of the created topologies was performed using a binary tree structure. This was done to facilitate the delivery of packets. The research indicates that any irregular network, except for the tree topology, has the potential to be more time-efficient than the tree topology. The research concluded that irregular network topologies, as opposed to tree topologies, are more receptive to optimisation in terms of the time restrictions that are imposed on the network. Only one-half of the topologies that were built were able to have their latency reduced, and only one of those was successful [12].

3. SYSTEM SETUP

To ensure that messages are delivered in the interest of achieving the highest possible degree of application performance, the primary goal of a NoC is to ensure that the NoC meets its objectives. Because even a brief disruption in communication has the potential to damage the integrity of a real-time system, those who work in the field are continuously required to keep this concern in the back of their minds. Before performing an accurate analysis of performance, the network must first be inspected to see whether it satisfies the required levels of flow and latency.

To examine the network in terms of its usual delay, the on-time delivery of RT packets, and the reduced geographical footprint that has occurred as a direct result of there being fewer

routers is the purpose of this study. We have a hypothesis that states that if we cut the projected number of routers in the network, the network operations centre (NoC) will have a smaller overall footprint as a result.

A count is kept at each regional gateway to determine what percentage of RT packets arrive at the time that was designated for their arrival. This can be done by determining what percentage of RT packets arrive on time. At the conclusion of the simulation, we computed the % by conducting a comparison between the total number of RT packets issued and the number of packets that arrived before the deadline.

To determine the average latency, first all the different delays that individual packets encounter while travelling through the network are tallied up and then the sum is divided by the entire number of packets that have travelled through the network. A more in-depth description of how to perform this computation is provided by the Eq.(1):

$$Latency = \sum_{i=1}^{packets} \frac{Latency\ of\ \ packet}{Total\ Packets} \quad (1)$$

By leveraging heterogeneous routers, pre-emptive virtual channels, and algorithms that prioritise packets, the research that is being presented here increases the performance of networks. The objective of the heuristic is to improve the operational capabilities of the network through the consolidation of routers and the reorganisation of the connections that exist between them.

The relationship that exists between the topology of the network and the number of routers in the network is demonstrated by the inequality.

$$0.5*\text{Cores Number} \leq \text{Cores Number} \leq \text{Routers Number} \quad (2)$$

In this scenario, the number of routers will be larger than or equal to the bare minimum required by the application in terms of cores, but it will never be more than the maximum number of cores plus two. The GA begins by assigning a core to each router at the beginning of the process. Subsequently, as routers are taken offline, their cores are transferred to routers that are still operating. Due to the likelihood of a deterioration in latency and deadline performance in RT packets if there are too few routers, the final topology may consist of as few as half of the initial number of routers.

This is because the number of routers in the initial topology was too high. Since the number of ports that a router can have been limited to a maximum of four, we are able to assume that each router in our network can connect with a maximum of four other routers. This is because the maximum number of ports that a router can have been four. The capabilities of several brands and types of routers will be evaluated by the GA within the context of a network that initially consists of only one router.

4. PROPOSED ASYNCHRONOUS ML

This issue of idle time was addressed by the paradigm that was provided, which made use of asynchronous parallel computing in setups that were not GP. Consequently, this issue was resolved. These works do not investigate the influence that asynchronous parallelism has on the functional complexity of variable-length structures in GP; instead, they concentrate on real parameter optimisation using fixed-length chromosomes. These research, for instance, do not analyse a bias in evaluation time that favours shorter evaluation lengths.

The APGP technique, which utilises an asynchronous evaluation process, makes it possible for less complicated individuals to enter the breeding population ahead of their more complicated counterparts in the regression model. This is made possible by the fact that the technique makes use of an asynchronous assessment process. The utilisation of asynchronous assessment is what enables this to become a reality. This strategy, in addition to being a natural process, is also compatible with the behaviour of populations that already exist in the real world, which do not cease reproducing even while one of its members is being tested for exposure to environmental factors. However, this is exactly what happens in the standard regression model: the evolution comes to a standstill whenever a person with any level of complexity is being evaluated. This is because the model assumes that all levels of complexity are equivalent.

The conventional regression model does not result in an improvement in the speed at which evaluations are carried out, and hence does not have the potential to be simplified. On the other hand, the purpose of the regression model is to make efficient use of the speed with which assessments may be made to create models that are not only straightforward but also reliable. This can be accomplished through the utilisation of the speed with which assessments may be performed.

The following formula, which is based on the logistic function f(z), can be used to calculate the probability of a landslip occurring at a given location.

$$f(z) = e^{z_1}+e^{z} = 1/(1+e^{-z}) \quad (3)$$

and might be anywhere from 0 to 1 depending on the circumstances at any one time. The coefficient z is represented as a linear combination of the independent elements that are the predictors of landslides. These components are responsible for the occurrence of landslides. This is one possible way that the model could be written.

$$z = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots\ldots + b_nX_n \quad (4)$$

where b_0 is the model coefficient, which is also known as the intercept or constant; b_1 through bn are coefficients that represent the relative importance of the predictor variables X_1, X_2,…, X_n; and m is the number of predictor variables.

Failures in slope stability are currently being caused by X_n. The maximum likelihood method, which is a derivation of the probability distribution of landslides, needs to be applied to estimate the unknown components b0 through bn in the landslide probability model. These words are defined in turn by the link that exists between the independent variables and the conditions that are present in the environment of the landslip.

On the graph, each independent variable that has a role in the occurrence of a landslip is shown as a separate theme layer. This is done so that the data may be more easily interpreted. An index known as z is utilised, the values of which can range anywhere from a minus one to a plus one to consider the several distinct factors that may be responsible for the occurrence of landslides. The information from the sample is utilised in the process of making modifications to a multiple logistic regression model. The computation of the event likelihood, using estimated values for the coefficients b_0, b_1, b_2, and b_3, is what needed to assess the probability of a landslip happening.

We make use of a logistic regression (LR) model to model the likelihood of a landslide occurring. This model is comprised of the following steps: (i) selecting independent variables based on their association with landslide occurrence; (ii) checking the statistical significance of the selected variables using the p-value significance test; (iii) verifying the lack of inter-dependence between the selected independent variables using collinearity statistics.

5. RESULTS AND DISCUSSION

The format of the d695 benchmark circuit displays information regarding the number of inputs, the number of outputs, the number of pins, the number of wrapper inputs, the number of scan chains, and the length of the scan chain. The results are generated using C# code that was written in the Visual Studio IDE. This reference circuit is utilised as an input by the programme, and several strategies of optimisation are employed to determine the method that is going to be the most effective when it comes to carrying out the test.

When considering the number of pins that are available on a SoC, it is possible to generate an estimate for time that will be required to test each core on its own. Given a SoC with K Test Access Mechanism (TAM) wires, a set of N cores, and a set of wrapper designs for each core, we anticipate setting the assignment of TAM wires to each core test and choosing the test start time for each test to minimise the overall test length.

Table.1. Latency for 10 cores

Long Deadline					Average Deadline					Short Deadline				
Core	**Latency**	**Packets**	**T1**	**T2**	**Core**	**Latency**	**Packets**	**T1**	**T2**	**Core**	**Latency**	**Packets**	**T1**	**T2**
2	275.456	0.986	4.930	0.986	2	3076.513	0.986	5.916	1.972	2	1508.675	0.986	5.916	1.972
3	185.478	0.986	2.958	2.958	3	9012.607	0.370	0.000	9.861	3	1043.254	0.986	2.958	2.958
4	175.420	0.986	2.958	2.958	4	1804.493	0.986	2.958	3.944	4	522.613	0.197	0.986	8.875
8	690.391	0.986	3.944	1.972	8	791.591	0.339	0.000	9.861	8	171.328	0.214	9.861	0.000
10	325.154	0.986	0.000	5.916	10	207.004	0.986	9.861	0.000	10	128.188	0.887	3.944	5.916

Table.2. Latency for 15 cores

Long Deadline			Average Deadline			Short Deadline		
Core	**Latency**	**Packets**	**Core**	**Latency**	**Packets**	**Core**	**Latency**	**Packets**
2	412.864	0.444	2	138.739	0.148	2	976.201	0.010
3	302.455	0.309	3	844.069	0.118	3	749.407	0.013
4	413.594	0.296	4	140.810	0.118	4	769.128	0.020
8	195.861	0.179	8	710.408	0.097	8	515.415	0.007
10	170.746	0.170	10	200.378	0.046	10	187.657	0.016

Table.3. Latency for 20 cores

Long Deadline					Average Deadline					Short Deadline				
Core	**Latency**	**Packets**	**T1**	**T2**	**Core**	**Latency**	**Packets**	**T1**	**T2**	**Core**	**Latency**	**Packets**	**T1**	**T2**
2	649.677	0.616	0.000	9.861	2	181.889	0.986	5.916	1.972	2	443.728	0.986	5.916	1.972
3	883.186	0.986	4.930	2.958	3	3668.151	0.189	0.000	9.861	3	558.772	0.986	5.916	1.972
4	740.523	0.986	2.958	4.930	4	861.660	0.986	1.972	5.916	4	731.421	0.986	4.930	2.958
8	481.583	0.610	0.000	9.861	8	319.809	0.348	0.000	9.861	8	181.889	0.875	0.000	9.861
10	688.742	0.800	1.972	6.902	10	1257.877	0.627	1.972	6.902	10	478.953	0.958	4.191	4.191

Table.4. Latency for 25 cores

Long Deadline			Average Deadline			Short Deadline		
Core	**Latency**	**Packets**	**Core**	**Latency**	**Packets**	**Core**	**Latency**	**Packets**
2	258.467	0.191	2	868.967	0.092	2	5.916	0.611
3	161.37	61.25	3	587.5	6.25	3	412.5	0.004
4	192.47	20.58	4	831.18	9.41	4	547.1	0.006
8	204.56	36.87	8	873.75	9.37	8	618.8	0.620
10	204.217	0.345	10	790.349	0.086	10	396.079	0.310

This will allow us to accomplish our goal of shortening the duration of the entire test. We will be able to shorten the total duration of the test to the greatest extent possible. Since a particular TAM wire can be allocated to just one core for the test at any given point in time throughout the test, the converted rectangles for the test scheduling problem are unable to overlap with one another. This eliminates the possibility of any overlap taking place as

The input parameters for DA and ALO are presented in Table 2 for a selection of TAM widths that range from very narrow to very wide. The selection of these values indicates a trade-off between the amount of time that can be spent to testing the SoC and the amount of TAM width that can be dedicated to testing the cores. Altering the parameters of the inputs can drastically reduce the amount of time needed to test a SoC.

An estimate of the total amount of testing time has been derived. Methods of test scheduling are utilised to determine when the optimal time is to put a core-based SOC system through its paces to achieve the best possible results. Throughout the course of this procedure, a variety of factors are evaluated and weighed against one another.

In the IrNoC simulator, the simulation is regarded as having successfully completed once the permitted period of time has gone, as was stated earlier on in this paragraph. This method could result in a significant delay even when only a limited number of packets are being sent or received back and forth. Because of this, the topology that was created at the time was suboptimal, which led to a congestion of packets, missed deadlines, and increased delay times.

When the topology of the network is optimised, a second type of router is implemented to give improved support for real-time multi-packet traffic by utilising its dedicated channels. This occurs after the first type of router has been used. The installation of the second kind of router makes this goal attainable. If it can construct a connectivity configuration for the router that works in favour of the application graph, then it will be possible to send every packet on time. The optimisation procedure for the chip area results in the employment of fewer routers, which results in reduced latency as a direct effect of the optimisation.

6. CONCLUSION

When it comes to solving the issue of latency, the mesh architecture has proven to produce superior outcomes time and time again. On the other hand, as was said earlier, since it was necessary to put an end to the simulations, these statistics are only able to reflect a small fraction of the overall influence that the delay had. This is because the irregular method uses fewer resources than the other two methods. If the network is not detected, there is only a 33% reduction in the area, whereas if it is observed, there is a 46% reduction. We were able to lower the amount of space that was required for each of our applications by leveraging mesh topologies, which helped us achieve this goal.

The findings show that it is possible to locate networks that have been carefully tailored to meet the requirements of a variety of use cases and the demands placed on packet latency. When testing ad hoc networks, the evolutionary algorithm was employed to look for topologies that satisfied the constraints outlined by the RT packets with relation to latency, area, and deadline. These conditions were set by the RT packets. The irregular networks with heterogeneous routers fared much better than the mesh networks when it came to performance and compliance to the requirements of the design.

REFERENCES

[1] K. Boikos and C.S. Bouganis, "A High-Performance System-on-Chip Architecture for Direct Tracking for SLAM", *Proceedings of International Conference on Field Programmable Logic and Applications*, pp. 1-7, 2017.

[2] A.J. Martin and M. Nystrom, "Asynchronous Techniques for System-on-Chip Design", *Proceedings of the IEEE*, Vol. 94, No. 6, pp. 1089-1120, 2006.

[3] L. Torres and D. Puschini, "An Introduction to Multi-Core System on Chip Trends and Challenges", *Proceedings of International Conference on Multiprocessor System-on-Chip: Hardware Design and Tool Integration*, pp. 1-21, 2010.

[4] K. Ilamathi and P. Rangarajan, "Intelligent Computation Techniques for Optimization of the Shortest Path in an Asynchronous Network-on-Chip", *Cluster Computing*, Vol. 22, pp. 335-346, 2019.

[5] X. Meng and S. Ray, "SoCCAR: Detecting System-on-Chip Security Violations under Asynchronous Resets", *Proceedings of International Conference on ACM/IEEE Design Automation*, pp. 625-630, 2021.

[6] Akash Punhani, NeetuFaujdar and Sunil Kumar, "Design and Evaluation of Cubic Torus Network-on-Chip Architecture", *International Journal of Innovative Technology and Exploring Engineering*, Vol. 8, No. 6, pp. 1672-1676, 2019.

[7] Sonal Yadav and Hemangee K. Kapoor, "Lightweight Message Encoding of Power-Gating Controller for On-Time Wakeup of Gated Router in Network-on-Chip", *Proceedings of International Symposium on Embedded Computing and System Design*, pp. 1-6, 2019.

[8] Fawaz Alazemi, Arash Azizimazreah, Bella Bose and Lizhong Chen, "Routerless Network-on-Chip", *Proceedings of IEEE International Symposium on High Performance Computer Architecture*, pp. 492-503, 2018.

[9] N. Viswanathan, K. Paramasivam and K. Somasundaram, "An Optimized 3D Topology for On-Chip Communications", *International Journal of Parallel, Emergent and Distributed Systems*, Vol. 29, No. 4, pp. 346-362, 2014.

[10] L. Benini and G. De Micheli, "Networks on chip: A new SoC paradigm", *IEEE Computers*, Vol. 31, No. 1, pp. 70-78, 2002.

[11] Daniel U Becker, "Efficient Micro-architecture for Network-on-Chip Routers", Ph.D Thesis Dissertation, Stanford University, 2012.

[12] A. Agarwal, C. Iskander and Ravi Shankar, "Survey of Network on Chip (NoC) Architectures & Contributions", *Scientific Journals International*, Vol. 3, No. 1, pp. 1-15, 2009.