Network-on-Chip based Systems

End-semester Project Report

Submitted in partial fulfillment of the requirements of
CS F266 Study Project

By

Akanksha Chaudhari
ID No. 2017AAPS0994G

Under the supervision of:

Dr. Kanchan Manna

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, GOA CAMPUS
September 2021

“ The view that machines cannot give rise to surprises is due, I believe, to a fallacy to which philosophers and mathematicians are particularly subject. This is the assumption that as soon as a fact is presented to a mind all consequences of that fact spring into the mind simultaneously with it. It is a very useful assumption under many circumstances, but one too easily forgets that it is false. A natural consequence of doing so is that one then assumes that there is no virtue in the mere working out of consequences from data and general principles. ”

- Alan Turing
Computing Machinery and Intelligence (1950)

Abstract

Massive advancements in silicon photonic technologies have now enabled the manufacturing of hybrid devices such as Photonic Networks-on-Chips (NoCs) by allowing us to integrate both the optical and electronic components of a system onto a single microchip. Photonic NoC based systems demonstrate a significant improvement over their conventionally used electronic counterparts in terms of bandwidth, latency and power consumption, and thus appear to be a promising solution to handle future communication needs. However, there are several other issues that need to be addressed when it comes to the design and testing of Photonic NoCs. The major design challenges include insertion power losses and crosstalk noise. High values of power loss and noise may lead to a significant performance degradation of Photonic NoCs in terms of their Signal-to-Noise Ratio (SNR), preventing them from communicating properly. These losses also pose a serious problem when it comes to testing of Photonic NoCs. Since simultaneous testing of multiple cores in NoC-based systems has become a general trend in order to reduce the test time, the traffic generated at the time of testing is quite high. This may lead to very high values of crosstalk noise, and hence very low SNR in Photonic NoCs. Therefore, if the tests are not scheduled properly, the low values of SNR may result in incorrect test outputs - thereby, resulting in the rejection of fully-functioning Photonic NoCs and reducing the overall manufacturing yield.

This project seeks to address the above-mentioned issues by considering the problem of preemptive test scheduling techniques in Photonic NoCs and exploring the Particle Swarm Optimisation (PSO) strategy in order to devise a crosstalk-aware mapping of the cores being tested. An optimized mapping of the test cores, their corresponding I/O pairs, test pattern frequencies and preemption times onto a generic mesh-based photonic NoC architecture is devised in such a way that the overall test time is minimized and the worst-case SNR is maximized.

Contents

Chapter 1

Introduction

1.1 Theory and motivation

Motivated by the ever-increasing computational demands and the rapidly scaling microchip technologies, the trend of integrating multiple parallel computing systems with multi-cores on a single chip is gaining traction. With such an escalating rise in the processing power and the number of components on a single chip, the design of the communication architecture plays a defining role in determining the system costs induced by the area, performance, and energy consumption of the overall system [7]. In order to drive down these costs, most of the on-chip communication interconnects are now evolving into Network-on-Chip (NoC) architectures.

Aside from better predictability and lower power consumption, the NoC approach offers greater scalability compared to previous solutions for on-chip communication [7]. The NoC architectures have further evolved with the advent of silicon photonic technologies, giving rise to the Photonic NoCs. Photonic NoCs show a significant improvement over their electronic counterparts because of their ultra-high bandwidth, low latency and extremely low power consumption and thus, appear to be a promising solution for high performance on-chip communication [2]. However, the design and testing of an optical on-chip network poses several challenges.

One of the major design challenges posed by optical on-chip networks is that they suffer from *insertion power loss* and *crosstalk noise*. Insertion loss is the signal power loss induced by a photonic element when it is inserted in an optical path. Whereas, crosstalk noise is the noise introduced in an optical signal due to its unfavorable coupling with the other optical signals. For instance, two optical signals can induce crosstalk noise to each other when they simultaneously reach a waveguide crossing or a photonic switch. In an ideal scenario, these signals would have propagated entirely at each waveguide crossing, with no reflection and no

crosstalk. Ideal crossing is however unfeasible and hence a small amount of optical power almost always switches into the coupled waveguide [9].

For reliable communication in Photonic NoCs, the optical signal power arriving at the receiver should be higher than the sensitivity of the photo-detector. High values of power loss and/or crosstalk noise may easily result in a significant performance degradation of the Photonic NoCs in terms of their Signal-to-Noise Ratio (SNR) resulting in a very low signal power. This may result in preventing the Photonic NoC from communicating properly [2]. Therefore, it is extremely crucial to factor in these power losses for Photonic NoC-based systems at design time and test time to ensure proper functionality of the entire system.

Another challenge faced by any (Electronic or Photonic) NoC-based system is the high failure rate of designed systems which leads to a loss in the yield of the manufacturing process. Therefore, guaranteeing the reliability and functionality of a designed system and all of its components by testing at manufacturing time is essential [1]. In NoC based systems, testing a core involves transporting the test patterns from system input to the inputs of individual IP cores, collecting their responses and transferring them to the system output. So, the testing process for a multi-core system for its entire functionality can be complex and time-consuming. Since test time also becomes a crucial factor in deciding the time-to-market, test engineers adopt extensive parallel testing strategies in order to reduce this time.

Parallel testing of cores in Photonic NoCs would result in multiple test patterns (optical signals) being transported over the on-chip network at any given time. So, there is a high probability of two or more of these signals interfering with each other, thereby inducing a significant amount crosstalk noise in the links. The amount of crosstalk noise generated in this case clearly depends on the testing strategy employed. So, if the testing strategy is not designed properly by factoring in all these losses, the tests may result in very low worst-case SNR values causing even the good chips to fail the test resulting in manufacturing yield loss. Therefore, designing a proper test scheduling strategy becomes extremely crucial in order to improve the overall test time without compromising the yield of the manufacturing process. This is achieved by reducing the usage of resource conflict [6] thereby maximizing worst-case SNR of the Photonic NoCs at the time of testing.

This work: This work seeks to address the above-mentioned problems by designing a preemptive test scheduling strategy that optimizes the test time and worst-case SNR for a generic mesh-based Photonic NoC architecture. In order to achieve this, the Particle Swarm Optimisation (PSO) strategy is explored for formulating a crosstalk-aware mapping of - the cores being tested, their corresponding I/O pairs, test frequencies and the traffic patterns associated with the test schedule - onto an M x N mesh-based photonic NoC architecture such that the overall test application time is minimized and the worst-case SNR for the given configuration is maximized.

1.2 Report outline

The remainder part of this report is organised as follows:

- Chapter 2 consists of the Architectural Description of the Photonic NoC considered, and the Crosstalk Propagation model for the same.
- Chapter 3 presents the Crosstalk Noise and Power Loss calculations, the worst-case SNR analyses equations at both Network and Router levels for the Photonic NoC configuration considered and the testtime calculations.
- Chapter 4 elaborates on the problem statement of this project and explains the solution proposed for the same.
- Chapter 5 includes the experimental setup - description of the implemented code and tools used, along with sample outputs to demonstrate the working.
- Chapter 6 concludes the report by reiterating the parts of the project that have been completed, along with the future work that needs to be done in order to further this work.

Chapter 2

Photonic NoC: Architectural Description

2.1 Mesh-based Photonic NoC Architecture

This work considers a mesh-based Photonic NoC with M x N tiles. Each tile in the network consists of an Intellectual Property (IP) core and an optical router as shown in FIGURE 2.1. All the tiles are connected to each other via channels through mesh topology. Each optical router is connected to four neighboring routers and to an IP core via two unidirectional silicon waveguides which constitute the input and output channels of the router.

Mesh-based Photonic NoCs consist of a hybrid network structure with two overlapped networks of the same topology. First, an electronic control network that is used to route the control packets using the packet switching technique and control the optical network. Second, an optical data network that is used to route the payload packets using the circuit switching technique. The metallic and optical interconnects in both the networks are bidirectional in nature, and the two overlapped networks can be implemented in different layers of the on-chip network [9].

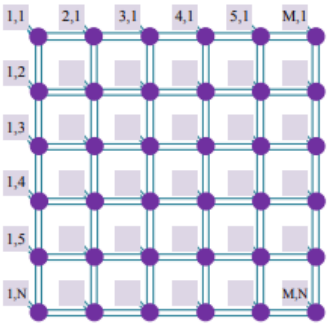

FIGURE 2.1: Mesh-based NoC with m x n tiles [8]

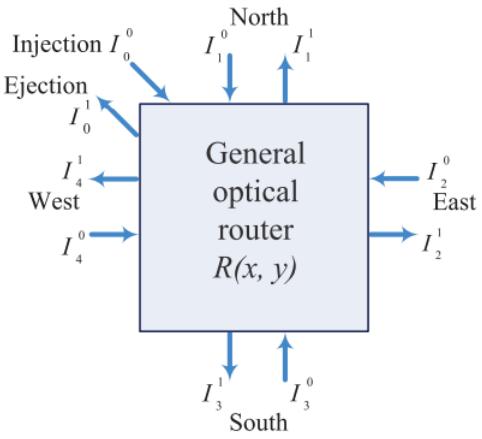

FIGURE 2.2: Generic 5 × 5 optical router model [9]

In mesh-based Photonic NoCs, the optical router is responsible for establishing and maintaining optical paths from the source to the destination for an optical signal carrying the payload data. For the purpose of the work, we consider the general 5 x 5 optical router model described in [9], which follows the dimension-order routing algorithm. The said model consists of five input and five output ports including: Injection/Ejection port, North port, East port, South port, and West port.

Each of the ports can be described by the parameter I_i^j where the subscript i defines the port number and j defines the port type. The i value for different ports varies from 0 to 4 according to the following table

i = 0	Injection/Ejection Port
i = 1	North Port
i = 2	East Port
i = 3	South Port
i = 4	West Port

Whereas, the superscript j takes only two values 0 and 1, j = 0 denoting that the port is an input one and j = 1 denoting that the port is an output one.

2.2 Switching Elements and Crosstalk Propagation Model

As discussed in CHAPTER 1, Photonic NoCs suffer from crosstalk noise whenever two or more optical signals reach a waveguide crossing or a photonic switch simultaneously. We now present

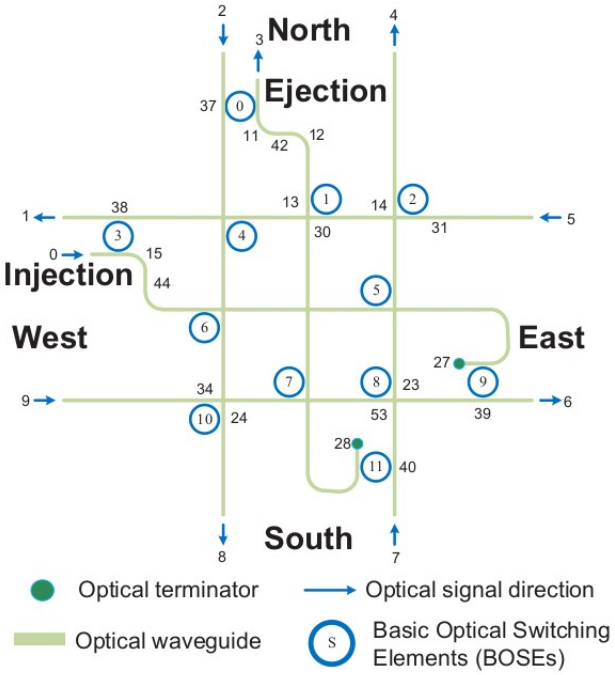

FIGURE 2.3: Optical router structure [8]

the architectural description of the switching elements in the optical router considered for this work, and describe how the optical signals and the induced crosstalk noise propagate through these elements.

Optical routers consist of Basic Optical Switching Elements (BOSEs), waveguides, waveguide crossings and optical terminators (as shown in FIGURE 2.3). Waveguides and microresonators form two different types of BOSEs [9]:

- **Parallel Photonic Switching Element (PPSE)**: built from a microresonator located between two parallel waveguides.
- **Crossing Photonic Switching Element (CPSE)**: built from a microresonator adjacently positioned next to a waveguide intersection.

The microresonators used in the circuit have a certain resonance frequency, depending on the material and structural properties. Based on whether or not this resonance frequency matches the frequency of the Basic Optical Switching Elements, the switches can be powered on (ON state) or off (OFF state) [9]:

- **ON state**: The switch is powered ON by injecting an electrical current into the p-n contacts surrounding the ring, or changing the temperature using the metal-plate based thermal heating. This causes the resonance frequency (λ_{on}) of the microresonator to shift

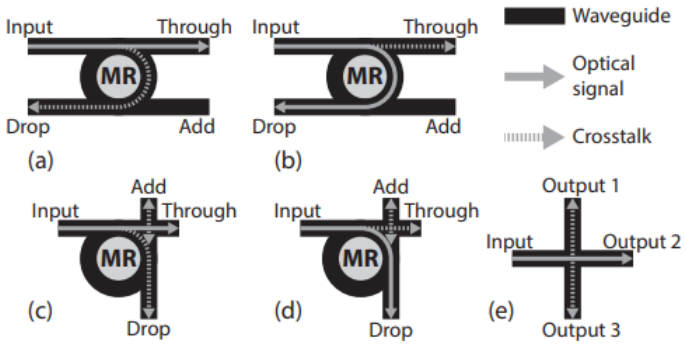

FIGURE 2.4: Optical signal and crosstalk noise propagation through: (a) PPSE in OFF state; (b) PPSE in ON state; (c) CPSE in OFF state; (d) CPSE in ON state; (e) Waveguide Crossing [2]

such that the wavelength of the optical signal now equals λ_{on}. This results in a resonance and the optical signal gets coupled into the ring which directs it to the drop port.

- **OFF state**: The switch is said to be in OFF state whenever the wavelength of the optical signal is different from the resonant wavelength of the ring (λ_{off}). The input optical signal, thus, propagates from the input port straight to the through port.

As shown in FIGURE 2.4 (e), a waveguide crossing has one Input port and three possible output ports - Output 1, Output 2, and Output 3. When two optical signals pass through a waveguide crossing simultaneously, a crosstalk noise will be generated at the crossing, as shown in FIGURE 2.4 (a) - (d). A small portion of these optical signals will also reflect back on the input ports. In the next section, an analytical model is presented for characterizing the crosstalk noise and power losses at both router and network levels.

Chapter 3

Signal power loss, Worst-case SNR and Testtime analysis

3.1 Crosstalk Noise and Power Loss calculations

In order to analyse the crosstalk noise and insertion power loss associated with an optical signal, the analytical model presented in [9] is considered. The output power at each port of both the PPSE and CPSE in both the ON and OFF resonance state is evaluated as a function of the input power and the loss, crosstalk and reflectance coefficients listed in FIGURE 3.1.

Parameter	Notation
Crossing loss	L_c
Propagation loss per cm	L_p
Power loss per CSE in OFF state	$L_{c,\text{off}}$
Power loss per CSE in ON state	$L_{c,\text{on}}$
Bending loss	L_b
Power loss per PSE in OFF state	$L_{p,\text{off}}$
Power loss per PSE in ON state	$L_{p,\text{on}}$
Crossing's crosstalk coefficient	K_c
Crosstalk coefficient per PSE in OFF state	$K_{p,\text{off}}$
Crosstalk coefficient per PSE in ON state	$K_{p,\text{on}}$
Crossing's back-reflection coefficient	K_r
Optical terminator's reflectance coefficient	K_t

FIGURE 3.1: Notation for insertion loss, crosstalk and reflectance coefficients [9]

Let P_{in} be the input optical power. Then, the output powers at ports Output 1, Output 2, and Output 3 denoted P_{out1}, P_{out2}, and P_{out3} can be calculated as follows:

$$P_{out1} = L_c P_{in} \ ... \ (1a)$$
$$P_{out2} = P_{out3} = K_c P_{in} \ ... \ (1b)$$

The fraction of the output power reflected back on the input port, P_{Rc} is calculated as:

$$P_{Rc} = K_r P_{in} \ ... \ (1c)$$

The output powers at the through port (P_T) and drop port (P_D) [1] as a function of input optical power (P_{in}) can be calculated as follows:

$$P_{Tpse,off} = L_{p,off} P_{in} \ ... \ (2a)$$
$$P_{Dpse,off} = K_{p,off} P_{in} \ ... \ (2b)$$

$$P_{Dpse,on} = L_{p,on} P_{in} \ ... \ (3a)$$
$$P_{Tpse,on} = K_{p,on} P_{in} \ ... \ (3b)$$

Optical routers also consist of a component called the optical terminator. This component is responsible for avoiding the optical signal from reflecting back on the input port. The reflected power at the optical terminator (P_{Rt}) is given by:

$$P_{Rt} = K_t P_{in} \ ... \ (4)$$

Compared with the PPSE, the CPSE has a waveguide crossing that introduces a non-negligible crossing insertion loss to the signal power. These power losses for OFF state of CPSE, at all the ports can be calculated as follows:

$$P_{Tcse,off} = L_{c,off} P_{in} \ ... \ (5a)$$
$$P_{Dcse,off} = (K_{p,off} + L^2_{p,offKc}) P_{in} \ ... \ (5b)$$
$$P_{Acse,off} = K_c L_{p,off} P_{in} \ ... \ (5c)$$
$$P_{Rcse,off} = K_r L^2_{p,off} P_{in} \ ... \ (5d)$$

Note that, when CPSE is in the OFF state, the power loss $L_{c,off}$, can be calculated based on the models of waveguide crossing and the PSE in OFF state which results in a total power loss of $L_c L_{p,off}$.

Similarly, the power losses for ON state of CPSE, at all the ports can be calculated as follows:

[1] The crosstalk noise on the add port of the BOSE is negligible and therefore, not considered.

$$P_{Dcse,on} = L_{c,on}P_{in} \ ... \ (6a)$$
$$P_{Tcse,on} = K_{p,on}P_{in}(L_c(1 + K_cL_{p,on}) + K_rL_{p,on}K_c) \ ... \ (6b)$$
$$P_{Acse,on} = K_{p,on}P_{in}(K_c(1 + K_cL_{p,on}) + K_rL_{p,on}L_c) \ ... \ (6c)$$
$$P_{Rcse,on} = K_{p,on}K_rP_{in} \ ... \ (6d)$$

Note that, when the CPSE is in ON state, the power loss, $L_{c,on}$, can be calculated by using the models of PPSE in ON state and the waveguide crossing which results in $L_{p,on}(1 + K_{p,off}K_c^2) + K_{p,on}^2K_c$. But, since the crosstalk coefficients have very small values ($K_iK_j \approx 0$), $L_{c,on}$ can be approximated by $L_{p,on}$.

3.2 Worst-case SNR analysis

This section describes the router-level and the network-level worst-case SNR analysis for a generic a M x N mesh-based Photonic NoC. Each node in the M x N consists of the 5 x 5 optical router (shown in FIGURE 2.2).

Consider a router at position (x,y) where $x \in \{1, ..., M\}$ *and* $y \in \{1, ..., N\}$ in the given mesh network, say R(x, y). Then, the **insertion loss ($L_{i,j}(x, y)$)** from the i^{th} port to the j^{th} port in the router R(x, y) is given by:

$$L_{i,j}(x,y) = \begin{cases} \mathrm{L}_{i,0}^{OR}(x,y), & \text{if j} = 0 \\ \mathrm{L}_{i,j}^{OR}(x,y)L_p^D, & \text{if j} \neq 0 \end{cases} \ ... \ (7a)$$

where, $L_{i,j}^{OR}(x, y)$ denotes the power loss introduced by the optical router R(x, y) and the D in L_p^D denotes the hop-length [9].

The optical router power loss $L_{i,j}^{OR}(x, y)$ includes the switching power loss, $S_{Li,j}(x, y)$, caused by the switching elements and waveguide crossings as well as the the propagation loss inside the optical router. The optical router power loss is thus calculated by considering the waveguide length between the i^{th} input port and the j^{th} output port, $W_{li,j}(x, y)$, and the propagation loss, L_p:

$$L_{i,j}^{OR}(x,y) = S_{Li,j}(x,y)\ L_p^{W_{li,j}(x,y)} \ ... \ (7b)$$

The hop-length D for homogeneous symmetrical Photonic NoCs which chip size $S(cm^2)$ and network size M x N is given by:

$$D \cong \sqrt{\frac{S}{M \times N}} \ ... \ (7c)$$

Therefore, we can calculate the optical power $P_{i,j}(x,y)$ outputted by the j^{th} port caused by optical power $P_i^0(x,y)$ injected into the i^{th} port in the router R(x, y). This is given by [9]:

$$P_{i,j}(x,y) = P_i^0(x,y)\, L_{i,j}(x,y) \ ... \ (8)$$

Next, to calculate the crosstalk noise for R(x,y), we first need to define the router status $s_{i,j}(x,y)$. The router status $s_{i,j}(x,y)$ of the router R(x, y) characterizes the propagation paths of optical signals between different router ports and therefore, obviously depends on the routing algorithm selected. For the purpose of this work, we consider the dimension-order routing or the XY routing algorithm. Accordingly, the router status $s_{i,j}(x,y)$ is given by [9]:

$$\begin{gathered} s_{i,j}(x,y) = (I_a^0, I_b^0, I_c^0, I_d^0, I_e^0) \text{ where, } ... \ (9) \\ a \in \{1,1,2,3,4\}, \\ b \in \{1,0,2,3,4\}, \\ c \in \{1,0,4\}, \\ d \in \{1,0,1,2,4\}, \\ e \in \{1,0,2\} \end{gathered}$$

The **crosstalk noise** ($N_{i,j}(x,y,s_{i,j}(x,y))$) added to the optical signal traveling from the i^{th} port to the j^{th} for router R(x,y) under the status $s_{i,j}(x,y)$ is given by [9]:

$$\begin{aligned} N_{i,j}(x,y,s_{i,j}(x,y)) = {} & P_0^0(x,y)K_{i,j,0}(s_{i,j}(x,y)) + ... \ (10) \\ & P_1^0(x,y)K_{i,j,1}(s_{i,j}(x,y)) + \\ & P_2^0(x,y)K_{i,j,2}(s_{i,j}(x,y)) + \\ & P_3^0(x,y)K_{i,j,3}(s_{i,j}(x,y)) + \\ & P_4^0(x,y)K_{i,j,4}(s_{i,j}(x,y)) \end{aligned}$$

where $K_{i,j,m}(s_{i,j}(x,y))$ is defined as the crosstalk noise coefficient introduced by optical power, $P_m^0(x,y)$ injected into the m^{th} port in the router R(x, y) under the status $s_{i,j}(x,y)$.

The SNR is the ratio of the signal power to the power of the noise corrupting the signal and is defined as follows:

$$SNR = 10\, log(\tfrac{P_S}{P_N}) \ ... \ (11)$$

Here P_S is the optical signal power and P_N is the power of noise.

Using these equations, the crosstalk noise and worst-case SNR can be obtained for all the routers R(x, y) in the given network; and thus, we can perform a network-level analysis to obtain the

worst-case SNR in the given (M x N) network. This analysis determines the feasibility of the given mesh-based Photonic NoC in terms of its communicability.

To find the worst-case SNR, two major criteria should be taken into consideration:

- The minimum SNR link should have high power loss.
- The minimum SNR link should suffer from high crosstalk noise introduced by other links.

Hence, it is not necessary for the longest link with maximum power loss to be the minimum SNR link, since it may not have as much crosstalk noise as shorter links, which suffer from higher crosstalk noise. Hence, all the links in the network must be analyzed to find the link which has minimum SNR [9].

3.3 Testtime calculation for Mesh-based NoCs

In this work, we consider the problem of offline preemptive test scheduling of Photonic NoCs. In preemptive test scheduling, the test administered to each core is distributed over a number of chunks, and each of these chunks is scheduled independent of the others.

The testtime for one chunk can be calculated using the method mentioned in [6]. Each chunk is composed of multiple flits (flow control unit or flow control digit)[2]. These flits move through the scan-chain in a pipelined fashion. As each flit is unpacked in one clock cycle, the time to send a test packet from source to the core plus the time to collect a response in the sink can be calculated as:

$$h_{s\to c} + (l-1) + 1 + h_{c\to k} + (l-1) \ \ldots \ (12)$$

where, p denotes the number of test patterns in the chunk, l denotes the maximum scan-chain length (number of flits per packet) of the core, $h_{s\to c}$ denotes the distance (in terms of hops) from source to core, and $h_{c\to k}$ denotes the distance (in terms of hops) from core to sink.

During testing, consecutive flits are applied to the core. So, the minimum inter-arrival time between consecutive flits is given by [6]:

Maxtime required to shift-in a flit into the core + time required to shift-out a response from the core + unit cycle to unpack the flit + time taken to generate the response

[2] A flit is a link-level atomic piece that forms a network packet or stream.

Therefore, the entire testtime ($\mathbf{T}$) for a given chunk can be calculated as follows:

$$\begin{aligned}\mathbf{T} &= h_{s\to c} + (l-1) + 1 + [1 + Max\,\{h_{s\to c} + (l-1), h_{c\to k} + (l-1)\}] \times (p-1) + h_{c\to k} + (l-1) \\ &= [1 + Max\,\{h_{s\to c}, h_{c\to k}\} + (l-1)] \times p + [Min\,\{h_{s\to c}, h_{c\to k}\} + (l-1)] \quad ... \quad (13)\end{aligned}$$

Chapter 4

Problem Statement and Proposed Solution

Problem statement: *To design a preemptive test scheduling strategy that (i) minimizes the test application time by minimizing the resource usage conflicts while (ii) maximizing worst-case SNR generated by the schedule for a given M* x *N mesh-based Photonic NoC.*

Testing a core involves transporting the test patterns from system input via an *input core* in the mesh network to the inputs of individual IP cores, collecting their responses and transferring them to the system output via an *output core* in the mesh network). So, there exist one or more pre-designated I/O pairs in the network that are responsible for feeding the test patterns to the test cores and collecting the respective test responses. Each test core in the network will be assigned to one of these I/O pairs, and will be tested by that I/O pair at the given test frequency[1].

Since the testtime for a given test core and the associated traffic pattern (which in turn determines the worst-case SNR) depends on the aforementioned factors - a naive, straightforward solution to the test scheduling problem would be to formulate a mapping of the test cores, their corresponding I/O pairs, test pattern frequencies and the preemption points in their respective test schedules (as shown in FIGURE 5.1) for each possible scenario and determine the mapping that optimizes the testtime as well as the SNR of the NoC at test application time.

The testtime (**T**) resulting from the scheduling of all cores of the mapping can be calculated using the procedure described in Section 3.3 (Expression (13)), and the Signal-to-Noise Ratio (SNR) in a given link for the associated test traffic pattern can be calculated using the procedure described in Section 3.2 (Expression (11)).

[1] A core may support multiple frequencies for a test. [5]

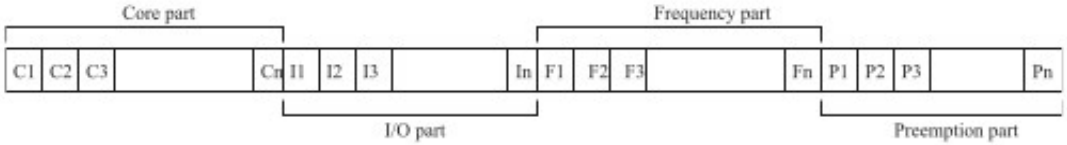

FIGURE 4.1: Representation of the formulated mapping of all the test cores, their I/O pairs, test pattern frequencies and the preemption points[6].

Drawing a parallel between the crosstalk-aware preemptive test scheduling problem considered in this work and the thermal-aware test scheduling problem considered in [6], we can infer that this problem is convex in nature [3]. Therefore, for a quality solution, the objective function must be formulated using a linear combination of testtime ($\mathbf{T}$) and worst-case SNR (SNR_{wc}). Thus, we define the following expression as the objective function:

$$\textbf{Objective funcion: } W \times \frac{\mathbf{T}}{\alpha} + (1 - W) \times \frac{SNR_{wc}}{\beta} \quad ... \quad (14)$$

where W $\in$ [0,1] is a weighting coefficient meant to balance the testtime and the worst-case SNR of the Photonic NoC. When W = 1, only the testtime is minimized, and when W = 0, only the worst-case SNR is maximized. The symbols α and β represent the normalizing coefficients for the testtime and worst-case SNR and can be calculated as the maximum testtime and the maximum worst-case SNR among all the configurations (respectively).

However, the solution discussed would involve a huge number of computations and comparisons in order to find the optimal solution as the network size increases. In order to reduce this computational load, the Particle Swarm Optimization (PSO) technique can be explored. Particle Swarm Optimization is a computational method that optimizes a problem by iteratively trying to improve a solution vector (in this case, the mapping considered) with regard to a given measure of quality (in this case, objective function specified by Equation 14) [4].

Chapter 5

Experimental Setup and Results

5.1 Experimental Setup

The experiment designed (insofar) includes a NoC-Map-Generator tool, implemented in C. The NoC-Map-Generator accepts an input file containing the mesh network dimensions, the designated input and output cores in the network, the number of test patterns to be administered to each test core and the scan-chain lengths corresponding to each test core. Based on the inputs provided, the NoC-Map-Generator generates n random mappings of the test cores and I/O pairs using the rand() function. For every mapping, this tool calculates the value of the objective function defined in Equation (14). The testtime is calculated directly by the NoC-Map-Generator, but the SNR calculation is done using the CLAP tool [8].

CLAP or the Crosstalk Noise Anlayses Platform (implemented in C++), is used to perform the analysis of crosstalk noise, insertion power loss and the resulting SNR for each of the mappings at the network as well as the router levels. CLAP accepts the following files as input:

Input file name	Application
input.txt	Determines the architecture type and the considered optical link
Device_Parameters.txt	Contains the basic device parameters' initiations
Router_Structure_Definition.txt	Defines the optical router structure
Router_Configuration.txt	Used to configure the optical router
Network_Configuration.txt	Defines network's properties and the communication pattern

FIGURE 5.1: CLAP Input Files[8]

The files - Device_Parameters.txt, Router_Structure_Definition.txt and Router_Configuration.txt already contain the device parameter initiations and the structure and configuration of 5 x 5 generic optical router (FIGURE 2.2) considered for this work, hence the default versions of these

files are used directly for the calculations. The parameters defined in these files consider the following technological profile.

Crossing insertion loss (dB)	Lc = 0.04
Crossing crosstalk coefficient (dB)	Kc = 40
Crossing back-reflectance coefficient (dB)	Kr = 0
Parallel switching passing loss (dB)	L_pse_off = 0.2
Parallel switching drop loss (dB)	L_pse_on = 0.33
Parallel switching crosstalk coefficient in OFF state (dB)	K_pse_off = 20.87
Parallel switching crosstalk coefficient in ON state (dB)	K_pse_on = 8.86
Optical detectors' loss - PASSING state (percentage)	L_det_off = 0.911908898
Optical detector insertion loss (DETECTING mode)	L_det_on = 0.52512987
Optical detectors' crosstalk noise (DETECTING mode)	K_det_on = 0.023186142
Optical terminator back-reflectance coefficient (dB)	Kt = 50
Waveguide bending loss (dB)	Lb = 0.005
Propagation loss (dB/cm)	Lp = 0.247
Wavelength polarization loss (dB)	Lpol = 0
Coupling loss (dB)	Lcpl = 1
Micro-resonator dimension (micrometer)	MR_Dimension = 10
Waveguide Width (micrometer)	WGD_width = 2
Input optical power (Injection power, dBm)	Pin = 0

The remaining two input files for CLAP - input.txt and Network_Configuration.txt are created by the NoC-Map-Generator depending on the mapping generated. The input.txt file defines the architecture type as mesh for this experiment and the list of considered optical links in input.txt consists of two entries per test core for this experiment - (i) assigned input core to test core (ii) test core to assigned output core[1]. The Network_Configuration.txt file defines the network dimensions M x N for the experiment and defines the test communication pattern which again consists of two entries per test core - (i) assigned input core to test core (ii) test core to assigned output core (same as the list of considered optical links in input.txt).

Once the NoC-Map-Generator creates all the required input files for the CLAP tool, it invokes the clap_driver function to run the CLAP tool and obtain the SNR values corresponding to all the test cores in an output file. Since the default CLAP output file format makes it cumbersome to extract the output values required for our calculations, the CLAP tool function in network_structure.cpp file is modified in order to create another output file map-snr-output.txt

[1]**Note that** - The insertion power loss, crosstalk noise and SNR values will be calculated for each of these links. However, to obtain the total SNR for testing each test core the SNR values of both the links (i) and (ii) corresponding to the respective test cores must be added added.

which only records the power values (Signal power, Crosstalk noise power) required for the objective function calculations.

The testtime values obtained from NoC-Map-Generator and the SNR values obtained from CLAP can now be used to calculate the objective function value for the given mapping.

5.2 Results

In this section we present a sample output of the code in order to demonstrate the working of the tool. The experiment is carried out for 5 x 4 mesh-based Photonic NoC with the technological profile mentioned in the previous section. The Photonic Noc considered has three I/O pairs listed as follows:

IO pair number	**Input and Output core numbers**
1	(Core 1, Core 20)
2	(Core 2, Core 15)
3	(Core 3, Core 19)

The remaining 14 cores (5 x 4 - 3 x 2 = 14) in the network are test cores. Each test core will have some scan-chain length associated with it, and will be tested using a given number of test patterns. The values considered for this experiment are as follows:

Number of test patterns	**Scan-chain length**
100	30
150	15
220	25
200	30
180	20
100	10
175	25
150	35
100	15
200	10
200	25
180	20
190	20
120	15

We run the NoC-Map-Generator tool to generate a single randomized mapping for the given inputs. The mapping generated by the tool is:

Test Core number	**IO pair number**	**Number of test patterns**
4	2	100
5	2	150
6	1	220
7	2	200
8	3	180
9	2	100
10	2	175
11	1	150
12	1	100
13	2	200
14	3	200
16	2	180
17	3	190
18	2	120

Testing of each of the test cores in the given mapping involves an optical signal propagating via two links (i) input core to test core (ii) test core to output core. The power values of the crosstalk noise and signal in both these links are added in order to calculate the final SNR, and the testtimes of the two links are also added in order to get the final testtime corresponding to each test core. The results obtained for the given mapping are as follows:

Test Core number	**IO pair number**	**Testtime**	**SNR**
4	2	3402	0.711312
5	2	2852	1.917018
6	1	6821	0.113682
7	2	6601	0.073126
8	3	3961	1.866631
9	2	1303	1.252244
10	2	4901	1.313218
11	1	6002	0.495707
12	1	1903	0.799232
13	2	2802	0.773891
14	3	5401	3.575023
16	2	4503	0.056813

Test Core number	**IO pair number**	**Testtime**	**IO pair number**
17	3	4563	1.161640
18	2	2161	0.570326

It is obvious that the results obtained for the given mapping are not optimal, as the SNR value drops below 1 for some test cores. Searching through all possible combinations of test cores, i/o pairs, test frequencies and preemption will allow us to find the most optimal mapping for each Photonic NoC configuration. However, the complexity of traversing through all possible mappings is very high. Thus, for this computation to be of some practical applicability other methods like Partical Swarm Optimization must be explored in order to reduce the complexity and the search space for obtaining the optimal mapping and test schedule.

Chapter 6

Conclusion and future work

Photonic Network on Chips (NoCs) appear today a promising solution to handle future communication needs. However, their design of presents us with several challenges. Crosstalk noise, an intrinsic characteristic of such NoCs, could lead to severe performance degradation and prevent the photonic NoC from communicating properly. At the time of testing, a high level crosstalk noise due to test patterns administered can cause good chips to fail the reliability test, and result in a manufacturing yield loss.

This project attempted to address the above issues by proposing a solution that involves devising randomized mappings of - the cores under test, their corresponding I/O pairs, test frequencies and the associated test traffic patterns - onto a generic M x N mesh-based Photonic NoC. Each of the mappings can then be evaluated in terms of the test application time and worst-case SNR corresponding to them. An objective function as a linear combination of the testtime and worst-case SNR is proposed in order to determine the quality of the obtained mapping. A NoC-Map-Generator tool is also designed and implemented for calculating the testtime and SNR per link for a given mapping in Photonic NoCs. This tool can be used to perform the calculations for all possible mappings. However, the resulting complexity in this case would be very high. Thus, alternative ways of finding the optimized mapping must be explored.

This work can thus be furthered by including a more sophisticated algorithm such as Particle Swarm Optimization in order to reduce the complexity and search time required. This problem can also be extended to include analysis for Photonic NoCs supporting Wavelength Division Multiplexing (WDM).

Bibliography

[1] Benton H Calhoun et al. “Digital circuit design challenges and opportunities in the era of nanoscale CMOS”. In: *Proceedings of the IEEE* 96.2 (2008), pp. 343–365.

[2] E. Fusella and A. Cilardo. “PhoNoCMap: An application mapping tool for photonic networks-on-chip”. In: *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2016, pp. 289–292.

[3] V. Iyengar and K. Chakrabarty. “Precedence-based, preemptive, and power-constrained test scheduling for system-on-a-chip”. In: *Proceedings 19th IEEE VLSI Test Symposium. VTS 2001* (2001), pp. 368–374.

[4] J. Kennedy and R. Eberhart. “Particle swarm optimization”. In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4. 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[5] Chunsheng Liu, V. Iyengar, and D.K. Pradhan. “Thermal-aware testing of network-on-chip using multiple-frequency clocking”. In: *24th IEEE VLSI Test Symposium*. 2006, 6 pp.–51. DOI: 10.1109/VTS.2006.88.

[6] Kanchan Manna et al. “Thermal-Aware Test Scheduling Strategy for Network-on-Chip Based Systems”. In: *J. Emerg. Technol. Comput. Syst.* 15.1 (2019).

[7] R. Marculescu et al. “Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28.1 (2009), pp. 3–21. DOI: 10.1109/TCAD.2008.2010691.

[8] M. Nikdast et al. “CLAP: a crosstalk and loss analysis platform for optical interconnects”. In: *2014 Eighth IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*. 2014, pp. 172–173. DOI: 10.1109/NOCS.2014.7008779.

[9] Y. Xie et al. “Formal Worst-Case Analysis of Crosstalk Noise in Mesh-Based Optical Networks-on-Chip”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 21.10 (2013), pp. 1823–1836. DOI: 10.1109/TVLSI.2012.2220573.