Implementation of a Bio-Inspired Framework for Secure Network on Chip

Sridhar M
Department of Electrical and Computer Engineering
Texas A&M University
sridharm@neo.tamu.edu

Megha Dey
Department of Electrical and Computer Engineering
Texas A&M University
megha_dey@neo.tamu.edu

ABSTRACT

Network-on-Chip (NoC) is a way to communication among the variety of intellectual property (IP) blocks required in complex System-on-Chips (SoC). However the security aspect of such inter-connect network is still unexplored. In this project we have implemented a bio-Inspired security model that detects and provides security against Denial of Service and hijack attacks in SoCs as well as security against attacks that tries to extract secret information. We then compare the performance of our implemented model against the existing 4X4 NoC.

General Terms

Algorithms, Design, Security, Embedded Systems, NoC, encryption, denial of service

Keywords

Embedded Systems, System on Chip (SoC), Network on Chip (NoC), Security, Communication Architecture, Bus, Security-Aware Design, Intrusion Detection, Anomaly Detection, Denial of Service (DoS) Attacks, Extraction of secret information, AXI Bus.

1. INTRODUCTION

Traditional SOC designs consists of processing elements, memories, DSP cores etc. communicating over bus-based interconnect templates. But with migration towards smaller technologies and increasing number of on-chip components, scalability, timing and power issues make these templates infeasible. The main difficulties with SoC designs are electrical noises due to large number of connecting wires, electromagnetic interference, increased energy consumption and decreased chip reliability.

To overcome these issues, a new design methodology has been proposed. This new method, popularly known as NoC is an on

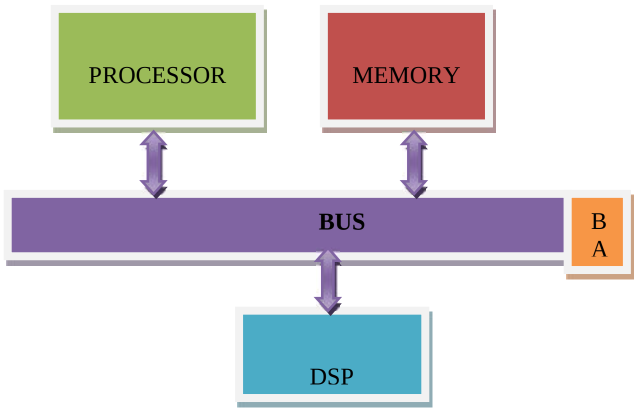


Fig 1: Traditional SoC architecture

chip-packet interconnection network which provide designers with a systematic and flexible framework to connect and manage the SoC's, besides addressing the timing, power and scalability issues.

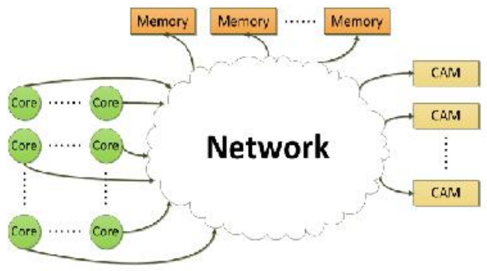


Fig 2: A typical NoC

A multiplication of communication links within SoCs also means an increase of doors to intrusions and hence access to sensitive data. While many papers have proposed various NoC models, the security aspect in such systems remains so far mainly unexplored Hence there is a need for a security model that detects such intrusions. The security model should effectively react to such security attacks by disallowing the offending communication transactions, or by notifying appropriate IP of a security violation.

In this paper, we propose a framework implementation for Bio-Inspired Framework for Secure System on Chip [1].

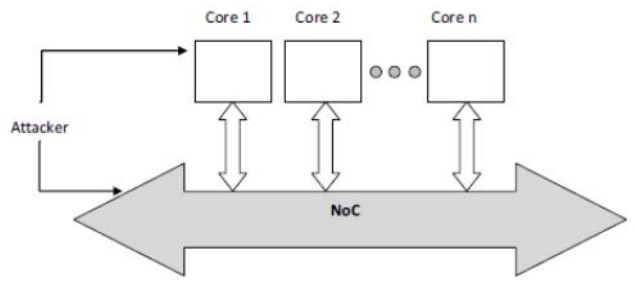


Fig 3 : An example of a general NoC and security attacks

The major security issues involved in NOC can be classified as denial of service attack, extraction of secret information and hijacking. End users may try to extract, from the firmware of their own device, information for hacking licenses or network accesses. Hackers aim to modify behaviors of remote systems for different reasons and operators may be interested by reading personal data for marketing or licensing purposes. All these attacks can be initiated through abnormal communications.

This paper is organized as follows: section 2 discusses the existing security issues in NoC. In section 3 we discuss about the bio-inspired model and how it can be applied to this problem. In section 4, we discuss the simulation results. Finally in section 5 we conclude and discuss future work in this field.

2. Security issues in NOC

2.1 Types of attacks

Figure II-A shows a general NoC and attacks over it. Here the attacker can be either from outside or inside NoC. S.Evainet al. [3] addressed the possible attacks with respect to NoC's. The main possible attacks are as follows:

• **Denial of service attack**: These attacks mainly aim to waste the system resources in several ways. The possible attacks under this category are replay, incorrect path, deadlock, and live lock.

–Replay: Replay attacks mainly attempt to waste the bandwidth of the system. For this purpose the attacker (either from inside or outside) will send some specific packets repeatedly.

– Incorrect path: Here, the attacker will send some packets with erroneous path with the aim to trap it into a dead-end. With this approach, some of the valid packets cannot use certain channels as they are already occupied by the attacker's packets.

– Deadlock: In this case, the attacker will send some packets with the intention to break the deadlock free rules so that these packets will create deadlock in the network.

– **Livelock:** Here, the attacker will introduce some packets with some erroneous path so that these packets would not reach their destination and stay forever inside the network.

• **Extraction of secret information:** These attacks are aimed at reading secure data stored in some secure areas such as memory core or other IP cores. These attacks can affect the configuration of IP cores and programs inside it.

• **Hijacking:** These attacks will try to get the permission to write some data in a secure memory area so that it can modify the behavior of the system. These kinds of attacks can be made by using buffer overflow and reconfiguring the internal registers.

3. Implementation

3.1 BIO-Inspired model

A security layer could be manifested into the SoC taking inspiration from the biological immune response. The figure below demonstrates this. In biological systems, the activation of an immune response is mediated by the white blood cells or the, macrophages. The macrophage presents antigen fragments on the surface of the cell. The T-Helper cell, a specialized macrophage interacts with the macrophage, recognizes the threat and gets activated. This activation causes an avalanche of cell destroyers. This influx of cell destroyers is an indication for the antibody producers to destroy an infected cell. This model has been used in various domains from computer security to decision support system to fault diagnosis to multi optimization problems as it is highly parallel and distributed.

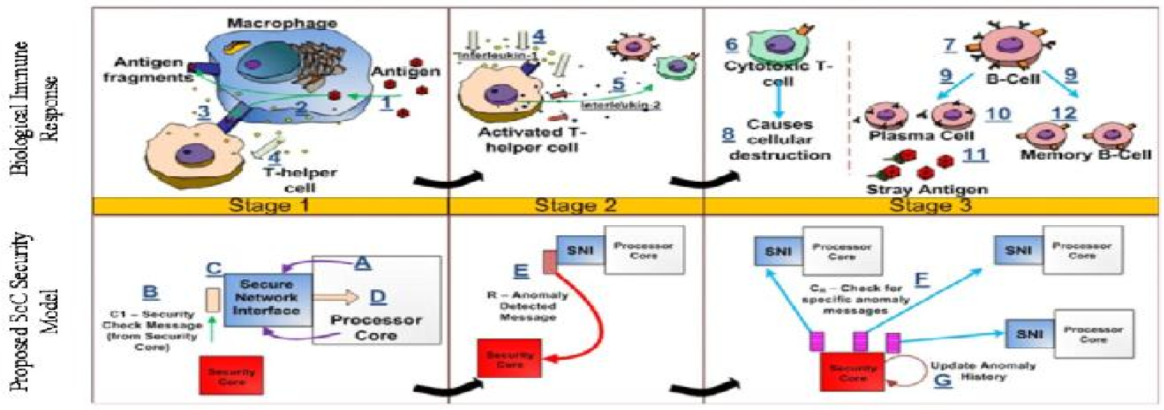


Fig 4: Bio – inspired model for SoC

In our model, the SNI knows the kind of traffic which it should expect. The SNI monitors the current behavior of the core (traffic generator) and compares the incoming traffic to the expected traffic. This monitoring of the SNI is analogous to the monitoring done by the macrophages. If the anomalous behavior continues beyond a threshold, the SNI goes to security threat activated stage (activation of T helper cell) an anomaly detected signal is sent to the security core which then stores this information for future uses as well as takes appropriate measures to ward off the attack. This is similar to the production of antibodies to kill the infected cell. The generation of the anomaly message corresponds to the generation of cell destroyers.

3.2 Existing NoC Framework

Our Security Model is built on top of the existing AXI compatible 4x4 mesh NoC based interconnect IP core. The 4x4 mesh NOC architecture is an m × n mesh of routers and resources are placed on the slots formed by the routers. Each router is connected to one resource and four neighboring routers. A resource can be a memory or Processing Element (PE) or any other intellectual property (IP) block, which fits into the available slot and complies with the interface of the NOC.

The router's job is to efficiently route packets through the network. A router mainly consists of the input channels to receive the packets, output channels for sending, a virtual circuit network for switching and a routing logic for doing the routing. Each router has four input and output ports for communicating with the other routers and one input output port for communication with the core.

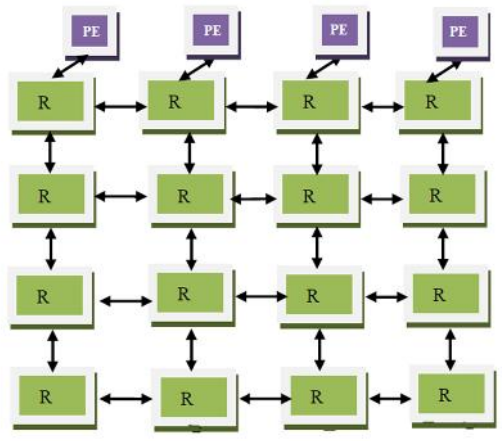


Fig 5 :Existing NoC

In our Project, the resource is an AXI stub that generates predefined packets (traffic) injected into the network at predefined intervals. These packets are generated at the source and carried to the destination by intermediate routers. Each packet has fields for source address(x and y coordinates), destination address(x and y coordinates) and the data. The NoC employed uses a classic XY routing algorithm i.e., a packet's output port at each router is computed from its destination address and the address of the router. The principle is that go as far as possible in the x-direction before taking a turn i.e., take a turn only when necessary. At the destination, the packet is decomposed into data and processed by the receiver.

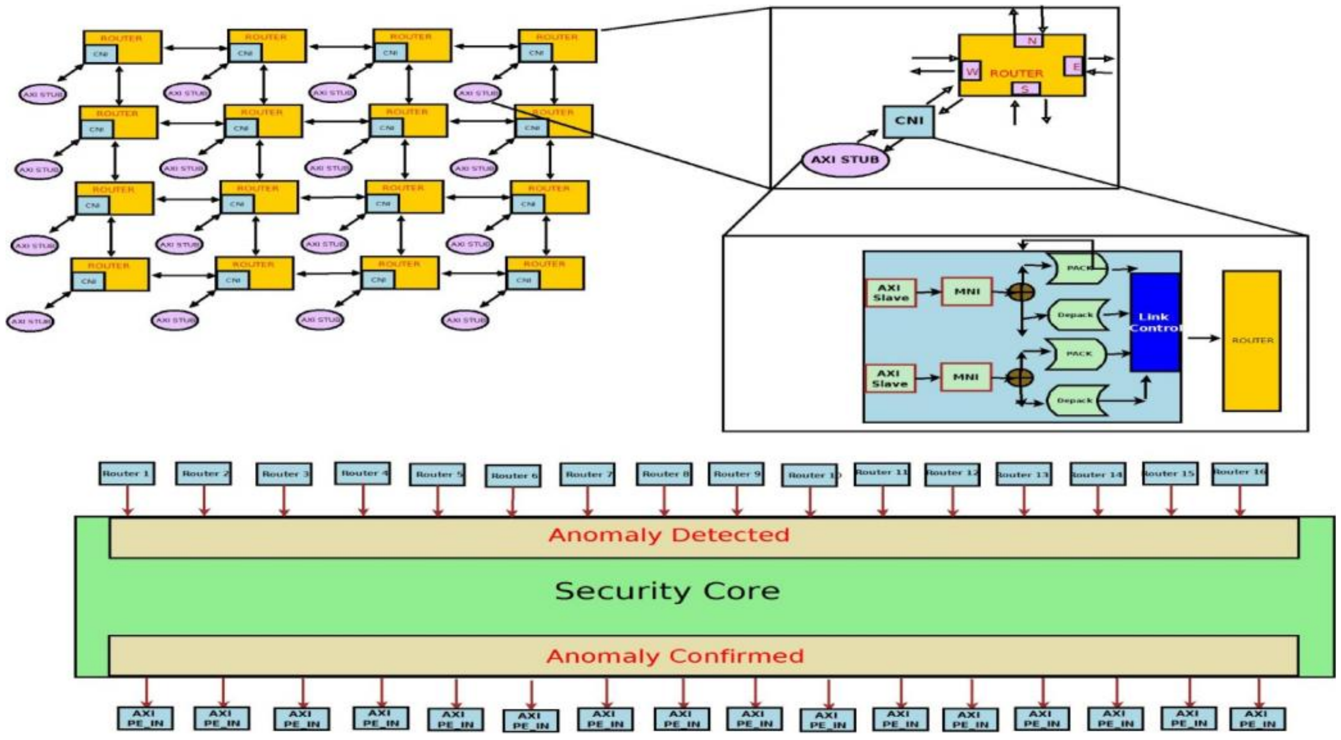


Fig 6: Implemented model

3.3 Implementation of security Framework

The implemented Bio-Inspired framework for a secured NoC includes the following:

3.3.1 Secured Core-Network Interface (SCNI)

The Security Core-Network Interface (SCNI) monitors three types of attacks: extraction of secret information, denial of Service (DoS) and hijacking. Once detected, it notifies the security core.

Extraction Of Secret Information:

In this type of attack, some external intruder could gain insight of the data which is being transferred in the network. Hence, once the CNI receives the data packet to be transferred from the traffic generator core, an encryption scheme is used such that even if an intruder gets hold of the data, it would not be compromised, as the intruder would not have the key to decrypt the message being sent. This acts as the first level of security in our design. Since all the routers used in our NoC have the decryption key available, they would be able to decrypt the message once received. The encrypt module is placed in the PACK module. Similarly, the decrypt module is present in the DEPACK module.

Denial Of Service Attack:

In this type of attack, the intruder tries to congest the network by continuously sending data packets of large sizes, thus hogging the bandwidth. This attack is only applicable when the type of request is a write.

In order to simulate this type of attack, we have considered the length of the incoming data packets as a means to detect an intrusion. The test script is able to generate traffic varying from BURST_1 to BURST_16. The S_AWLEN parameter of the AXI protocol signifies the number of bursts in a packet. So if the SW_LEN of 5 consecutive packets is greater than the threshold (in our case if S_AWLEN greater than 7), an anomaly detected signal is generated by the security core. The security core then takes the necessary action.

Hijacking

Each router has associated with it a bank of memory of size 0X10000000 bytes .This memory is divided into separate memory regions as follows:

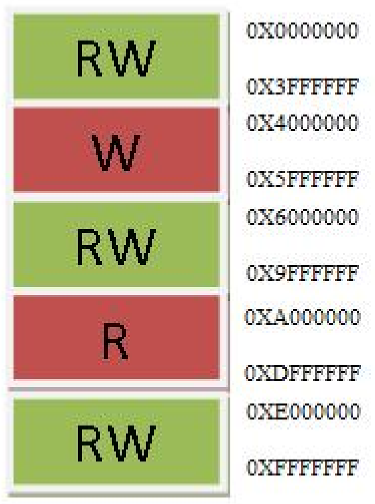


Fig 7: Memory Protection Regions of each router

Hence, whenever any traffic generator core tries to access any memory location in the restriction section of memory, an access violation anomaly detection signal is generated.

In the case of both reads and writes, whenever the CNI gets a request, it would first check if the address to written to or read from lies in the forbidden range. If so, an anomaly detected signal is sent to the security core and the write (or read) is aborted, else normal passage of data continues. However, if there are more than 5 attempts (need not be continuous) from a particular traffic generator core to access the write/read protected region, the SC generates an anomaly detected signal.

3.3.2 Security core

The security core has 16 inputs, one each from the 16 routers and 16 outputs, each connected to the traffic generators. The interconnections among the various components are as shown in figure 6.

Once the anomaly detected signal is received 5 times continuously in the DoS attack or any 5 times in the hijack attack, the security core generates an anomaly confirmed signal.

The Security core then prevents the corresponding traffic generator from injecting traffic to the network. Hence, all routers do not receive traffic from the buggy traffic generator. However, they are able to manage requests from other cores.

Flow chart in figure 8 indicates the data flow in the secure NoC.

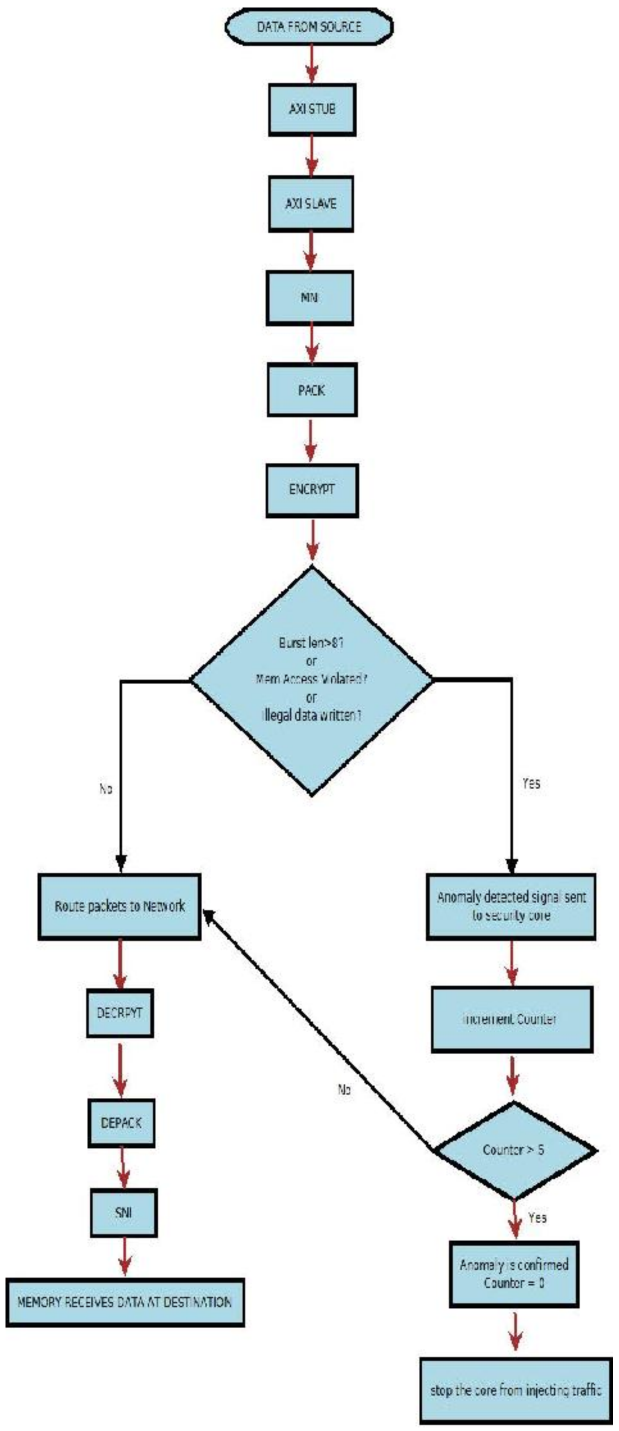

DATA FROM SOURCE
AXI STUB
AXI SLAVE
NNI
PACK
ENCRYPT
Burst len>8? or Mem Access Violated? or Illegal data written?
No
Yes
Route packets to Network
Anomaly detected signal sent to security core
DECRYPT
Increment Counter
DEPACK
No
Counter > 5
SNI
Yes
Anomaly is confirmed Counter = 0
MEMORY RECEIVES DATA AT DESTINATION
stop the core from injecting traffic

Fig 8: Data flow in secure NoC

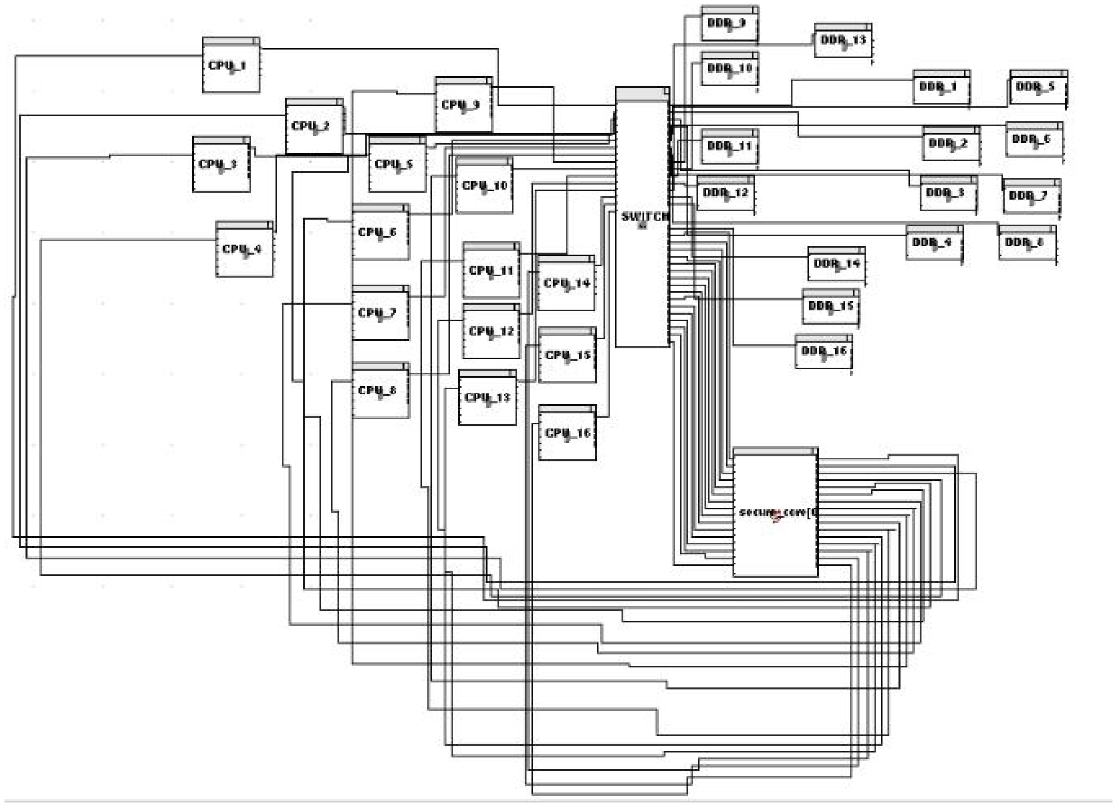


Fig 9: Schematic of NoC with security core in carbon SoC Designer

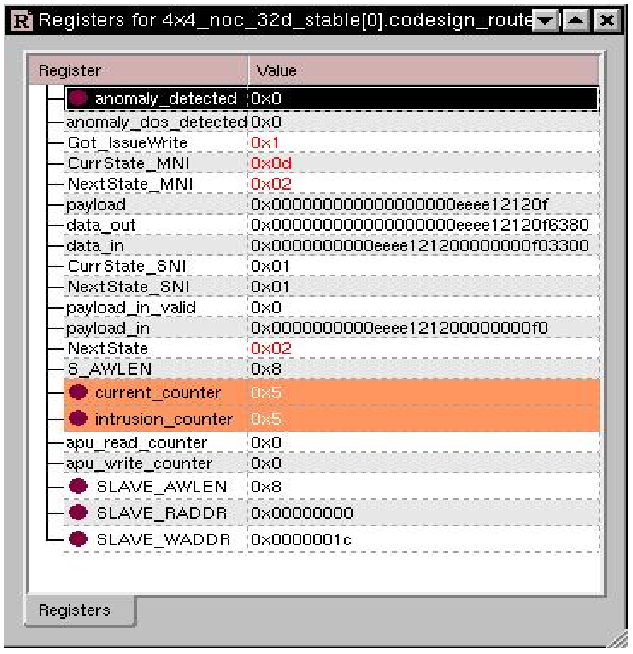
4. VALIDATION

The simulation is executed in Carbon SoC Designer, an industrial cycle-accurate instruction set simulator. Carbon Model Studio (creates system components) and Carbon Compiler enables the creation of a high-performance linkable software object that contains a cycle and register accurate model of RTL. We created our Secure Core Module in cycle accurate Carbon Model Studio and incorporated this model to the existing NoC. The schematic of the NoC with the security core is as shown in fig 9.

4.1 Denial Of Service:

Figure 10 illustrates the state when 5 successive data packets are sent with a burst length greater than 7. As we can see, the intrusion_counter signal is set to 5, while the anomaly signals are still zero.

Figure 11 shows the detection of this attack and correspondingly updates anomaly_dos_detected and anomaly detected signal to 1 while resetting the intrusion counter to 0.


Registers for 4x4_noc_32d_stable[0].codesign_route

Register	Value
anomaly_detected	0x0
anomaly_dos_detected	0x0
Got_IssueWrite	0x1
CurrState_MNI	0x0d
NextState_MNI	0x02
payload	0x00000000000000000000eeee12120f
data_out	0x00000000000000000000eeee12120f6380
data_in	0x0000000000eeee121200000000f03300
CurrState_SNI	0x01
NextState_SNI	0x01
payload_in_valid	0x0
payload_in	0x0000000000eeee121200000000f0
NextState	0x02
S_AWLEN	0x8
current_counter	0x5
intrusion_counter	0x5
apu_read_counter	0x0
apu_write_counter	0x0
SLAVE_AWLEN	0x8
SLAVE_RADDR	0x00000000
SLAVE_WADDR	0x0000001c

Registers

Fig 10: NoC registers before DoS attack is identified

Figure 12 shows the corresponding waveform when a DoS attack is identified.

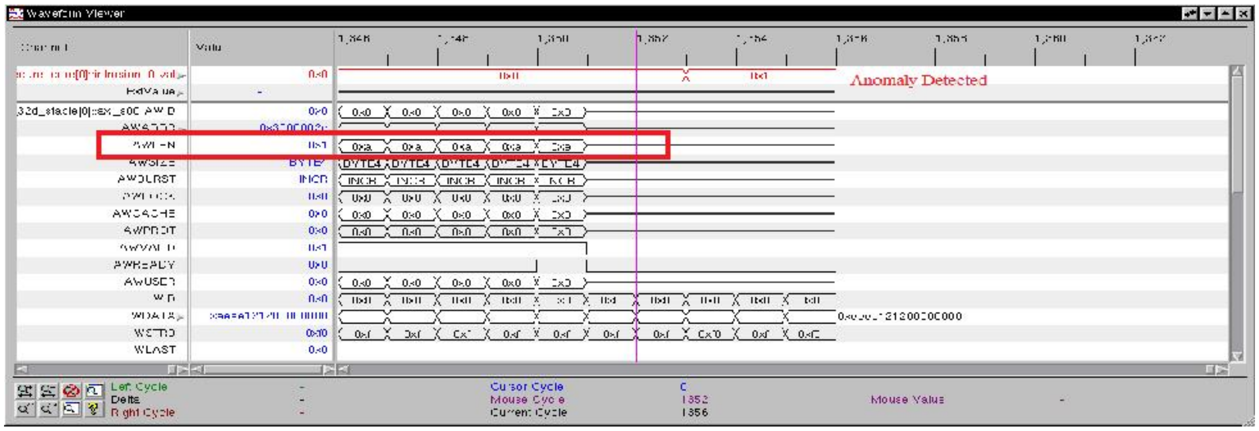
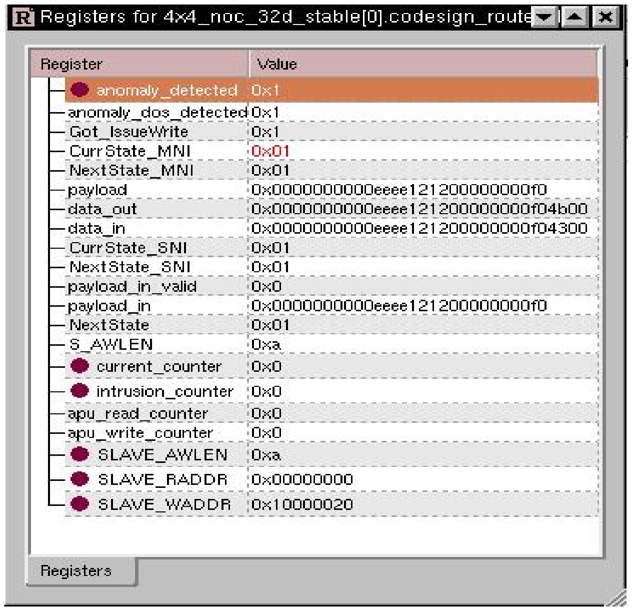


Figure 12: waveform showing Dos attack detection



Registers for 4x4_noc_32d_stable[0].codesign_route

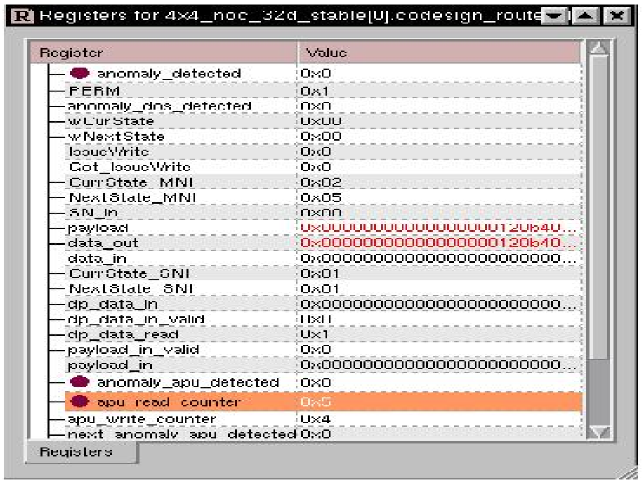
Register	Value
● anomaly_detected	0x1
anomaly_dos_detected	0x1
Got_IssueWrite	0x1
CurrState_MNI	0x01
NextState_MNI	0x01
payload	0x0000000000eeee121200000000f0
data_out	0x0000000000eeee121200000000f04b00
data_in	0x0000000000eeee121200000000f04300
CurrState_SNI	0x01
NextState_SNI	0x01
payload_in_valid	0x0
payload_in	0x0000000000eeee121200000000f0
NextState	0x01
S_AWLEN	0xa
● current_counter	0x0
● intrusion_counter	0x0
apu_read_counter	0x0
apu_write_counter	0x0
● SLAVE_AWLEN	0xa
● SLAVE_RADDR	0x00000000
● SLAVE_WADDR	0x10000020

Registers

Fig 11: NoC registers before DoS attack is identified

4.2 Hijacking

Figure 13 illustrates the state when 5 illegal read memory accesses have occurred. As we can see, the apu_read_counter is now at 5 while the anomaly_apu_detected signal is 1and anomaly detected signal is 0. Figure 14 shows the detection of this attack and accordingly updates anomaly detected to 1 while resetting the apu_read_counter to 0.

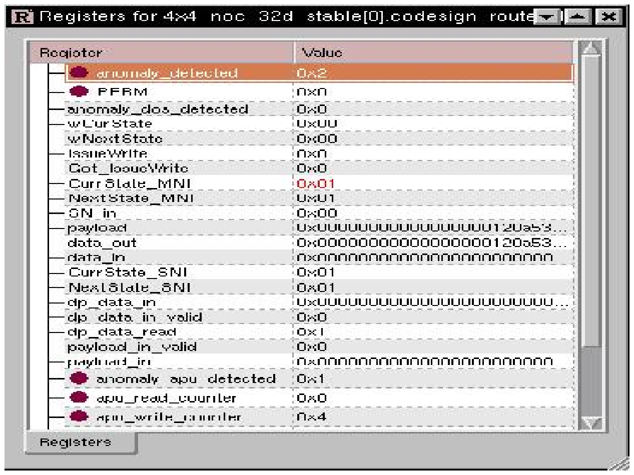


Registers for 4x4_noc_32d_stable[0].codesign_route

Register	Value
● anomaly_detected	0x0
PERM	0x1
anomaly_dos_detected	0x0
wCurState	0x00
wNextState	0x00
IssueWrite	0x0
Got_IssueWrite	0x0
CurrState_MNI	0x02
NextState_MNI	0x05
SN_in	0x00
payload	0x000000000000000000120b40...
data_out	0x000000000000000000120b40...
data_in	0x000000000000000000000000...
CurrState_SNI	0x01
NextState_SNI	0x01
dp_data_in	0x000000000000000000000000...
dp_data_in_valid	0x0
dp_data_read	0x1
payload_in_valid	0x0
payload_in	0x000000000000000000000000...
● anomaly_apu_detected	0x0
● apu_read_counter	0x5
apu_write_counter	0x4
next_anomaly_apu_detected	0x0

Registers

Fig 13: NoC registers before Hijack attack is identified



Registers for 4x4 noc 32d stable[0].codesign route

Register	Value
● anomaly_detected	0x2
● PERM	0x0
anomaly_dos_detected	0x0
wCurState	0x00
wNextState	0x00
IssueWrite	0x0
Got_IssueWrite	0x0
CurrState_MNI	0x01
NextState_MNI	0x01
SN_in	0x00
payload	0x000000000000000000120a53...
data_out	0x000000000000000000120a53...
data_in	0x000000000000000000000000
CurrState_SNI	0x01
NextState_SNI	0x01
dp_data_in	0x000000000000000000000000...
dp_data_in_valid	0x0
dp_data_read	0x1
payload_in_valid	0x0
payload_in	0x000000000000000000000000
● anomaly_apu_detected	0x1
● apu_read_counter	0x0
● apu_write_counter	0x4

Registers

Fig14: NoC registers after Hijack attack is identified

4.3 Latency

We have analyzed average packet latency in the implemented Secure NoC and compared the results with the existing results for NoC without the security model. Figure 15 clearly indicates, our model can be incorporated into the existing design without considerable overhead to the latency of the network

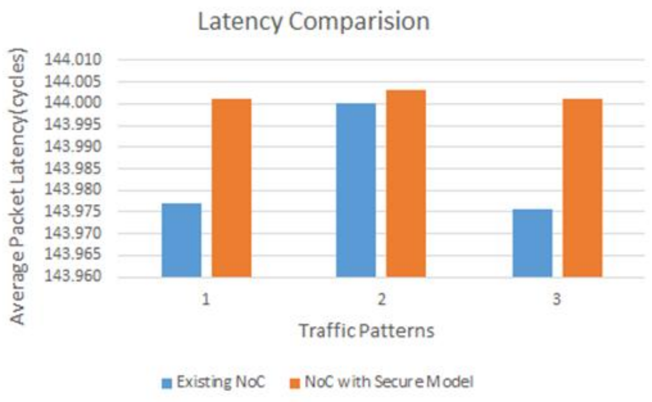


Figure 15

5. Conclusion and future work

In this project, we implemented a Bio-Inspired framework that provides security to the NoC architecture. We implemented a security model by adding Secure Core to the existing 4x4 Network that continuously monitors for the Denial of Service and Hijack attacks in NoC and inhibits the faulty core from further injecting bad traffic into the network.

The model presented does not take into account scalability aspects. Also, the protocols for identifying the attacks could be made stronger. Future research could also include optimizing the response time of the security core to any intrusions.

6. REFERENCES

*[1] A. Mandal, **S. K. Mandal**, A. Tripathy, N. Gupta and R. Mahapatra, "A Bio-Inspired Framework for Secure System on Chip", Workshop on SoC Architecture, Accelerators and Workloads, 2010*

[2] Jean-Philippe Diguet, Samuel Evain, Romain Vaslin,. Proceedings of the First International Symposium on Networks-on-Chip (NOCS'07)

[3] K. Sajeesh and Hemangee K. Kapoor, An Authenticated Encryption based Security Framework for NoC Architectures , 2011 International Symposium on Electronic System Design

[4]] J. P. Diguet, S. Evain, R. Vaslin, G. Gogniat, and E. Juin. NoC-centric security of reconfigurable soc. In Proceedings of the First International Symposium on Networks-on-Chip(NOCS'07), May 7-9 2007.

[5] Leandro Fiorin, Gianluca Palermo, Slobodan Lukovic, SecureMemory Accesses on Networks-on-Chip, IEEE TRANSACTIONS ON COMPUTERS, VOL. 57, NO. 9, SEPTEMBER 2008