DETECTION OF HARDWARE TROJAN IN NOC BASED SOC

PROJECT SYNOPSIS OF MINOR PROJECT

BACHELOR OF TECHNOLOGY

INFORMATION TECHNOLOGY

SUBMITTED BY :-

Lokesh Dhingra (1905359)

Krishan (1905355)

Branch : D3-IT-A2

Guru Nanak Dev Engineering College

Ludhiana-141006

Table Of Contents

INTRODUCTION TO PROJECT

Hardware Trojan is a malicious modification of the circuitry of an integrated circuit. A hardware Trojan is completely characterized by its physical representation and its behavior. The payload of an Hardware Trojans is the entire activity that the Trojan executes when it is triggered. In general, malicious Trojans try to bypass or disable the security fence of a system It can leak confidential information by radio emission. Hardware Trojans also could disable, derange or destroy the entire chip or components of it.

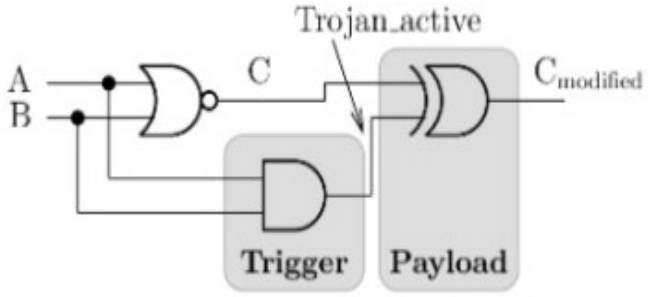

Figure 1: Hardware Trojan

Hardware Trojans can be implanted in security-weak parts of a chip with various means to steal the internal sensitive data or modify original functionality, which may lead to huge economic losses and great harm to society. Therefore, it is very important to analyze the specific Hardware Trojan threats existing in the whole life cycle of integrated circuits, and perform protection against hardware Trojans

System On Chip (SOC) :
A system on a chip, also known as an SoC, is essentially an integrated circuit or an IC that takes a single platform and integrates an entire electronic or computer system onto it. It is, exactly as its name suggests, an entire system on a single chip. The components that an SoC generally looks to incorporate within itself include a central processing unit, input and output ports, internal memory, as well as analog input and output blocks among other things. Depending on the kind of system that has been reduced to the size of a chip, it can perform a variety of functions including signal processing, wireless communication, artificial intelligence and more

Network On chip (NOC) :
A typical NoC architecture consists of multiple segments of wires and routers. n a tiled, city-block style of NoC layout, the wires and routers are configured much like street grids of a city, while the clients (e.g., logic processor cores) are placed on city blocks separated by wires. A network interface (NI) module transforms data packets generated from the client logic (processor cores) into fixed-length flow-control digits (flits). The flits associated with a data packet consist of a header (or head) flit, a tail flit, and a number of body flits in between.

OBJECTIVES OF PROJECT

1. To Detect Hardware Trojan in System on chip
2. To Study and analyze the effect of Hardware Trojan on SOC

FEASIBILITY STUDY

1. **Technical Feasibility -** The project is evaluated to be technically feasible as the technology to be used is easily available and open source. The technology and related dependencies are up to date with the current technical requirements and provide the best possible way for implementation of the project.

2. **Market Feasibility -** The project is market feasible in the sense that this topic is in demand and many researches are currently undergoing. The project being open source provides a feasible and easy way for them to customise the system according to their needs and use it to boost their target space.

3. **Economic Feasibility -** The project is economically feasible in the sense the project is simulation based and there is no extra cost in running the simulation. The cost will only be applicable when the simulation will be fabricated on actual hardware. And moreover the technologies used in the project are easily available.

4. **Operational Feasibility -** The project is operationally Feasible as the recommendation system is easily customizable to meet the requirements of a particular product. The project being open source provides horizons to continuous improvements which are beneficial for the product to get the best accurate/optimised recommendations. Moreover as it is a simulation, it can be easily fabricated on actual hardware with some modifications.

METHODOLOGY

The Execution of project can be divided into different phases. The Major phases of the project are:

Installation of gem5 :

The gem5 simulator is an open-source system-level and processor simulator. It is utilized in academic research and in industry by companies. To use it First the source code of the gem5 repository should be clones from `https://gitlab.com/abhijitcse/gem5.git`.
Now, clone the gem5 repository using command ‘ `$ git clone <git link>` ‘. After this gem5 system is installed in your system.

Building of gem5 :

After cloning the source code, you can build gem5 by using `SCcons`. The Building of gem5 can take anywhere from a few minutes on a large server to 45 minutes on a laptop. Gem5 must be built on a Unix platform. Linux is tested on every commit, and some people have been able to use MacOS as well, though it is not regularly tested. It is suggested to not try to compile gem5 when running on a virtual machine for high specification. When running with a VM on a laptop gem5 can take over an hour just to compile.
First go to the clone gem5 repository `$ cd gem5` and then the build the gem5 using build command `$ scons build/X86/gem5.opt -j <NUMBER OF CPUs ON YOUR PLATFORM>`.

Running scripts with parameters :

When performing experiments with gem5, you don’t want to edit your configuration script every time you want to test the system with different parameters. To get around this, you can add command-line parameters to your gem5 configuration script. Again, because the configuration script is just Python, you can use the Python libraries that support argument parsing.
Python interpreter is compiled into gem5 which takes care of library of simulation objects described in python- Now that you have a gem5 binary, you can run your simulation. gem5’s interface is Python scripts. The gem5 binary reads in and executes the provided Python script which creates the system under test and executes the simulator.

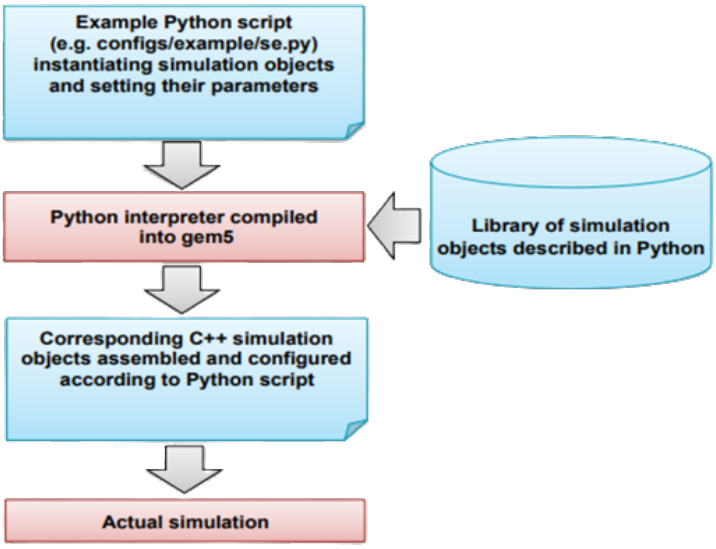

Figure 2: Methodology

Analysis Phase :

After the execution phase when the simulation finishes the results can be analyzed easily by opening scripts file present in the m5out folder. The changes in parameters for different cases can be analyzed and further conclusions can be made

FACILITIES REQUIRED FOR THE PROPOSED WORK

To develop this project, We have to get some system requirements to develop this project and below here we mentioned the minimum requirements to get access of gem5 and ubuntu in system.

SYSTEM REQUIREMENT :

- Recommended Processor : Intel core i3 or above
- Recommended Processor Speed : 2 GHz CPU
- Recommended RAM : 4 GB or Above
- Recommended OS : Ubuntu or Mac

SOFTWARE REQUIREMENT :

- Gem5
- Garnet 2.0
- C++
- Python

REFERENCES

[1] Gem5 Simulator System [Online] Available: https://www.gem5.org/ [Accessed: 10-Mar-2022].

[2] Tushar Krishna, Learning Gem5 [Online]
Available: https://tusharkrishna.ece.gatech.edu/teaching/garnet_gt/ [Accessed 15-Mar-2022]

[3] R. Chakraborty, S. Narasimhan and S. Bhunia, "Hardware Trojan: Threats and Emerging Solutions", 2021. [Accessed 2-Apr-2022].

[4] M. R, A. Das, J. Jose and P. Mishra, "SECTAR: Secure NoC using Trojan Aware Routing", 2022. [Accessed 1-Apr-2022].

[5] "Abhijit Das / gem5", GitLab, 2022. [Online].
Available: https://gitlab.com/abhijitcse/gem5.git. [Accessed: 5-Mar-2022].