Assignment 2 :

Arjun Srivastava : 160001007, Biplab Kumar Sahoo:160001015

1. Explain the design flow of SoC.

Ans: The design flow of SoC consists of third party IP vendor, System designer or Design integrator and Foundry or Fabrication.

- 3PIP : Provides the relevant IP
- System designer: Then combines the IP to create new functionality. If new things are needed they are created in RTL.
- Last fabrication: Design sent to the factory for manufacturing.

2. Explain the attack models of Trojan in IC design flow.

Ans: There are three places where a Trojan could be embedded in IC design.

- 3pip vendor (provides schematic , vhdl, verilog files): A 3pip vendor could give us schematics which contain Trojans embedded in them.
- Integrator: Usually trusted, so this threat is not modelled.
- Foundry: A factory could embed Trojans in design we ask them to manufacture.

3. What is code coverage analysis?

Ans: It refers to the study of how well the code is tested.
It contains but is not limited to:

- How many times an if condition was checked?
- How many times an particular routine was called ?
- How much of the input space was checked ?

A higher code coverage means the code is well tested and hence is likely to contain fewer bugs. Code coverage analysis also helps remove test cases which don't help increase coverage

4. What is DoS attack.

Ans: A denial of service attack when it comes to SOCs means, a chip's functionality (partial or complete) becoming unavailable due to a Trojan, present in the device, leading to the real consumer not being able to use the device for their uses.

5. Explain H/W Trojan attacks affecting performance.

Ans: A Trojan could just delay the output of a chip instead of giving the wrong output, this would lead to reduction in the performance of SOCs, these kinds of attacks would not be caught under functional verification and are therefore much harder to detect and correct.

6. What is rare combination Trojan?

Ans: A Trojan which only activates upon getting a rare triggering signal to a combinational circuit, ie. A signal which is a very small part of the chip's all possible triggers.

This leads to the chip behaving in a normal way most of the time and only breakdown after receiving the input trigger, it is hard to mitigate against this kind of Trojan because when relying on third party

ip it is difficult to tell useful signals from the Trojan triggering ones and also because of the fact that the triggering signal makes up a very small part of the input space.

7. Explain register allocation step.

Ans: Accessing data from registers is much faster than accessing data from memory, but number of registers is limited, to get high performance we would like to use the minimum number of registers needed to run a program.

Register allocation works as follows.

- Get data of which variables are used together or need to be in memory at the same time to perform a computation
- Create a graph where each node is a variable and there is an edge between two nodes if both variables need to be in the registers simultaneously for an operation.
- Problem reduces to colouring the nodes in such a way such that no two nodes which are connected by an edge are the same colour. Here each colour represents a register and the minimum number of colours needed are the number of registers needed

8. What is list scheduling and how it is performed on a DFG?

Ans: List scheduling is done in following steps:

- Selection of the highest priority process from the list,
- Selection of resource to accommodate this process
- If no resource then proceed to next step.

In case of DFG, the scheduled graph contains timestamps assigned to the function and information about storage variables. Data dependency is taken into account when scheduling of all operations based on on resource constraint is specified.

9. Explain watermarked IP chipset design flow.

Ans:

Design Flow :

1. Architectural synthesis
    a. Scheduling
    b. Hardware allocation and binding
    c. Register allocation
    d. Signature selection and ***embedding of watermark***
2. RTL
3. Gate level
4. Layout level

Watermarking guidelines:

1. The earlier you put your watermark in the better, a watermark deep in in the design makes it harder to reverse engineer.
2. The higher level of abstraction you embed the watermark the better the constraints because of this would automatically be incorporated into lower levels of the design

Reference:

1: https://en.wikipedia.org/wiki/System_on_a_chip#Design_flow