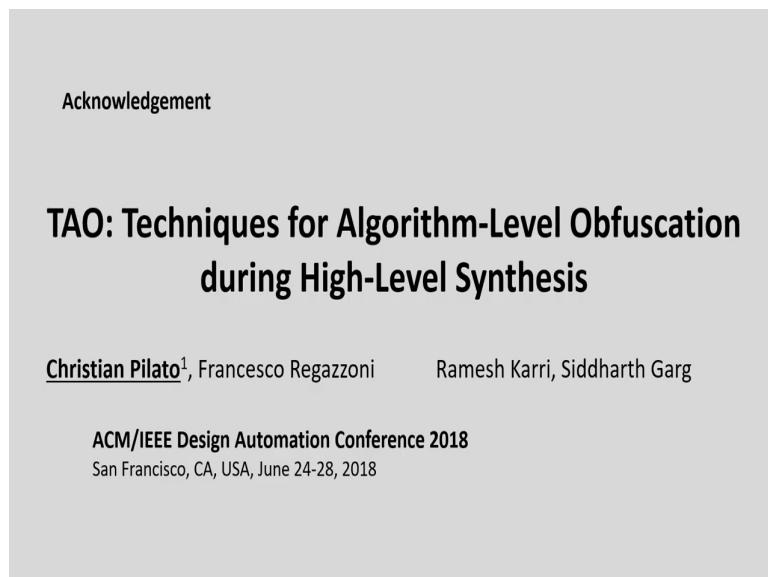


C-Based VLSI Design
Dr. Chandan Karfa
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Module - 11
Securing Design with High level Synthesis
Lecture - 36
Introduction to Hardware Security

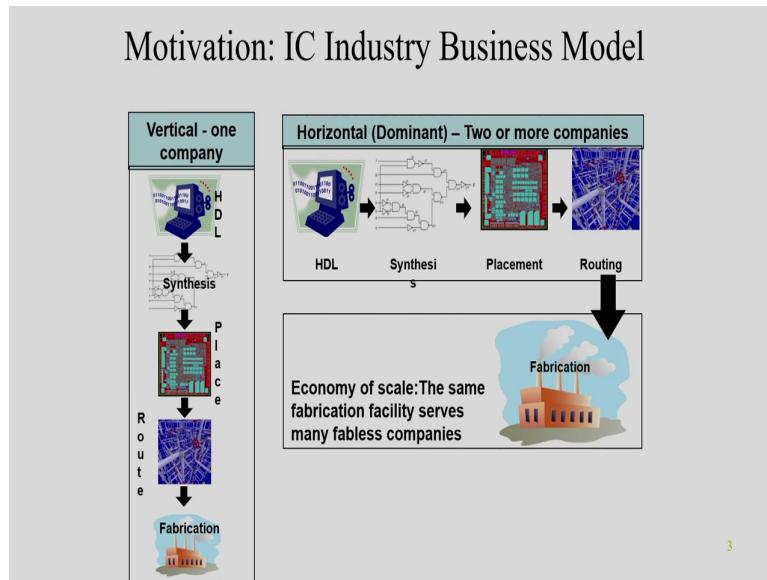
Welcome to the course C-Based VLSI Design, in this week we are going to discuss about Hardware Security and how high-level synthesis can help in making secure hardware. So, in this particular course, we try to mean by hardware security is the security concern that is coming because of this globalization of the IC design flow and whatever the risk is coming and how we can actually solve those security issues in high level synthesis.

(Refer Slide Time: 01:28)



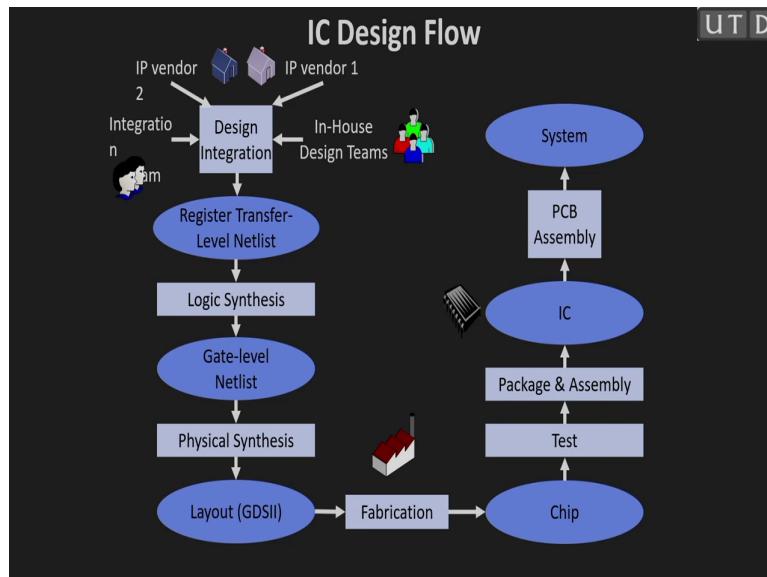
So, I want to acknowledge the authors of this paper because this particular lecture is primarily adapted from the slides of this particular paper given by these authors

(Refer Slide Time: 01:38)



So, let us try to understand the IC design flow. So, the primary motivation is that if you have a chip, you have a specification. So, you have a specification and then you have to do the synthesis which part of this high-level synthesis or logic synthesis physical synthesis and then finally, you will get the layout and then you have to fabricate it. So, the IC is something after fabrication only you will get the IC.

(Refer Slide Time: 02:06)



So, that is the overall idea of the IC design flow. So, if I just go into with details so as I mentioned that nowadays the chip is not a single thing it is a SoC, system on chip. So, you need multiple IPs so you bring the IPs from different vendor and those vendors might use, they develop that IP they give RTL IP and this IP may be developed directly by writing the HDL or you write C code and then you do use high level synthesis to generate those IPs.

So, you have different component you brought from different vendors and then you integrate them and then after integrating, you reintegrate the things in the register transfer level. So, you have the RTL and then you do this logic synthesis which will convert the RTL into gate level.

And then you do the physical synthesis which will convert this gate level into transistors and also you all it also places the transistors in a particular specified area it also place the interconnections of between the transistors. So, it gives you the layout rights which is very common format is GDS2 format. And that layout is given to of the fabrication lab. So, the fabrication lab is something where actual this layout is physically realized in a IC and then you will get a chip.

Once the chip come you have to test whether is there any malfunction and then you once you found that your chip is working fine then you do the packaging assembling you get the IC, integrated circuits. And then you combine into a bigger system PCB and then you have the system you port it to the customers. So, that is the overall the IC design flow.

(Refer Slide Time: 03:44)



So, before globalization so what usually happen the whole flow was having inside a particular single company. So, you have a single company which have all this facility it has a team to develop IPs, it has teams to develop them into integrations, it has software's to

convert these things into logic synthesis physical synthesis all the software they have and then finally, one they have the layout they actually have the fabrication lab also.

So, they have the complete flow and then they have the test engineer, they have the assembly and packaging engineer, then they integrate everything and they have the complete system developing in house and it was shipped to the customers. So, it is basically there is no risk of having their intellectual properties leaked to somebody else. So, everything inside a single company and there is no risk.

(Refer Slide Time: 04:40)



But because of this globalization and it is a tremendous growth of this IC industry, we have lot of companies coming up and having this facility in a single company is become impossible because this specifically this fabrication is lab is way expensive.

So, that kind of model does not survive. So, what the model comes up that we have a dedicated fabrication lab so that will just do the fabrication there are dedicated companies who just develop the IPs. So, they have the experts ah these RTL engineer of a engineers they just only create the efficient IPs.

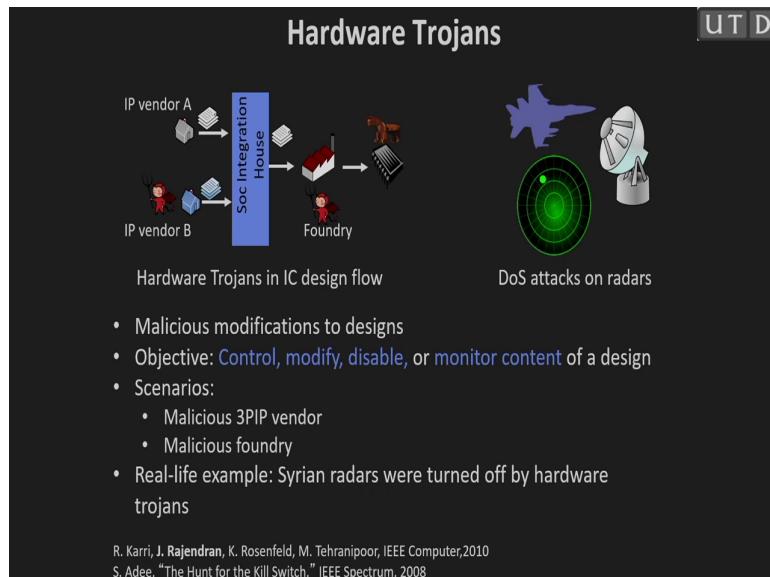
So, and then there are many such companies. So, there are companies exist who actually develop this efficient EDA software for the logic synthesis physical synthesis high level synthesis. So, it basically gets globalized so there are experts there are domain experts for EDA tools there are expert for IP developer.

And then the EDA companies this this chip design companies what they do, they buy the IPs from the vendors they buy this EDA software from the EDA software tool provider and they just integrate the overall things into one and then then give the complete thing to the fabrication lab because they do not have the fabrication facility.

And then the fabrication lab actually fabricates the IC and then it gives back and then the packaging everything happens and then it comes to the customer. So, this is the feasible options and that is what is happening nowadays. So, but what is the problem with this?

So, you can see here that a IP engineer is they are giving the IP to this design house and this whole thing this though after layout everything the layout is given to the fabrication lab so, that means, you are actually revealing your intellectual property to third party.

(Refer Slide Time: 06:43)



So, and that might be costly right because there might be somebody in that fabrication house who are not say trusted employee and they might do something to your intellectual property. And then we will see because what are the security concerns come because of this globalization of the IC design flow.

So, the first one is the hardware trojan it is very well known that hardware trojan is something a malicious modification of the design. So, so you as I mentioned that you have the SoC integration house who actually integrate the IP taken from the viewer vendor and different SoC and that was given to a foundry.

So, now this foundry might have some malicious employees who might ah actually insert hardware trojan in your design. So, then the hardware trojan what it does. There are two type of hardware trojan one is basically leaking some sensitive information through something or it might modify the circuits also.

So, there are two type of hardware trojan available and basically this hardware trojan gets activated in a very corner scenario. So, like this if the possible input is a specific pattern then only this hardware trojan will be activated or say it will activate after one year.

So, this kind of trigger condition is something is very corner cases, as a result even if this malicious modification happened during testing or any other process it will be hard to detect those hardware trojan. So, they will actually bypass the whole checking process and then it will go into the customer side and that they might do the damage.

So, we already have many such incidents and the interested reader can actually google it and they will get lot of studies. So, since I am giving this my layout to the foundry, who would do the fabrications so they might insert hardware trojan in your system. So, this is the one of the security risks.

(Refer Slide Time: 08:48)

Counterfeiting

Forbes
The Serious Risks From Counterfeit Electronic Parts

abcNEWS
Counterfeit Chinese Parts Slipping Into U.S. Military Aircraft: Report

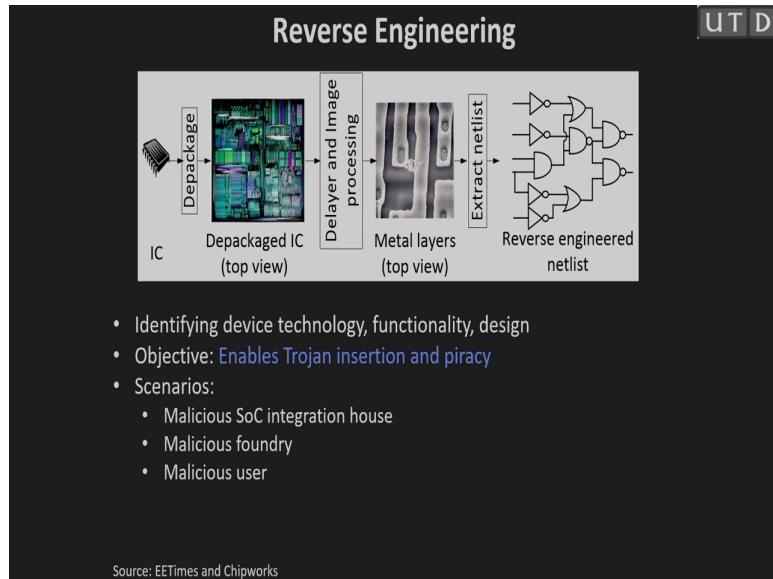
- Forgery or imitation of original components
- Objective: [Commercial benefit and subvert mission-critical systems](#)
- Scenarios:
 - Malicious test facility
 - Malicious assembly unit
 - Malicious sellers
- Real-life example: Counterfeit components in US military systems

Source: Forbes and ABCNews

The next one is the counterfeiting is something you have the ICs you remove the packaging and everything and then you repackage yourself, so this way you creating a counterfeit model. So, it has commercial benefit and also it might actually impact some mission critical system.

So, you can actually create a duplicated/ counterfeited copies and you can sell it into market and you actually have a financial loss to the actual IP provider. So, this is another possible risk.

(Refer Slide Time: 09:19)



And then the other important thing is the reverse engineering. So, this is one of the most important threat is that you have the IC ,you buy something from market you de-package, you remove all these assemblies other things you have the IC and then you actually delayer and you do some image processing. So, this is a bit involved process, but then you try to see the actual layout structure.

So, you just give the see the metal layers you see the axial structure and then if you are very expert on that then you can actually extract the netlist from this. So, then what is going to happen that this is something you have your intellectual property it will go to somebody else he understands what is the structure of the circuit and then what it does, you can develop his own IC from that because it has the netlist which you can assume it in the gate level.

And then he can generate his IP and it can sell in the market and it can actually can have a huge loss for the original companies. So, this is another threat.

(Refer Slide Time: 10:17)

Reverse Engineering

The slide features a news clipping from EE Times about reverse engineering. The headline reads: "System and IC teardowns become critical 'business intelligence'". Below the headline, there's a small image showing a circuit board and some text. To the right, there's a diagram illustrating the process of reverse engineering an IC, showing a flow from a physical IC to a "Reverse engineered netlist".

- Identifying device technology, functionality, design
- Objective: Enables Trojan insertion and piracy
- Scenarios:
 - Malicious SoC integration house
 - Malicious foundry
 - Malicious user
- Real-life example: Chipworks reverse engineers ICs to detect piracy

Source: EETimes and Chipworks

(Refer Slide Time: 10:19)

IC/IP Piracy and Overbuilding

The slide features news clippings from SEMI and EETimes. SEMI states: "AS SEMICONDUCTOR EQUIPMENT A LOSES UP TO \$4 BILLION ANNUALLY". EETimes reports: "ARM files patent infringement suit against IP startup picoTurbo". Below the news, there's a diagram showing a flow from a "3PIP vendor" through an "SoC Integration House" to a "Foundry", resulting in "Orders 100K IC" and "Makes 300K ICs".

- Steal & claim ownership of an IC and/or illegal use
- Scenarios:
 - Malicious SoC integration house
 - Malicious foundry
- Real-life examples:
 - \$4,000,000,000 loss per year to IC industry
 - ARM detected IP piracy in 2000

Source: SEMI and EETimes

The other threat is this IP piracy and overbuilding that is something as I mentioned that by doing this reverse engineering. First of all is this reverse engineering can do this IP piracy that they actually delay and they will get the netlist and they can actually have this own design from that.

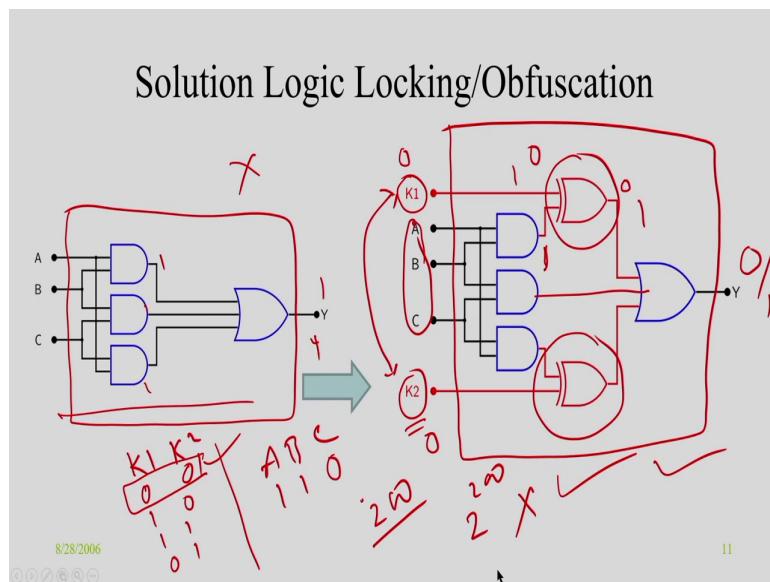
So, that something is always possible and, but even this foundry if it is malicious what can happen if you say you order 100K number of ICs and they produce 300K. So, there is no way you can actually detect whether they are produced 300K or 200K. So, then 200K they can

actually sell it in the market in half price and because they do not have to invest in doing this buying this IPs or invest into integrating things and all the test verification everything.

So, they do not have to pay anything and they have the IC with them. So, that something is really a concern specifically this over building an IP piracy. And if you Google it there are many such news reported specifically in this EDA domain and that electronic domain that there are lot of such incident actually identified. And so, this is something is a real concern.

So, in this particular lecture we try to see how we can actually stop this reverse engineering and specifically reverse engineering enable the IP piracy, so IP piracy and overbuilding. So, how to stop this IP piracy and overbuilding? So, that something is the hardware security concern that we try to address in this particular class and, in this particular week.

(Refer Slide Time: 12:05)



So, one of the important or interesting solution comes to is the logic locking or logic obfuscation. So, because what I when explained so far that you have to give your IC or the layout to the fabrication lab or the fab-lab. Because you do not have your own this foundry or you have no your own fabrication lab so you have to give it, there is no way alternative, you have to give it.

And if there is some malicious either the whole fabrication lab is malicious or there are some employees who are not trusted. So, you cannot control that part. So, then what you have to do? The solution that something this logic locking talks about that I will give my IC to you, I will give my inter-layout to you, you can do the reverse engineering you can extract the netlist you can produce multiple copies you can duplicate the things and you can sell it market, but that particular IP will not work.

So, even if you have the whole netlist, you produce IC you cannot use it. So, that means, I give you my intellectual property to you, you can do this IP piracy you can do this overbuilding, but that will not be useful. So, that was the motivation of this logic locking.

The idea is very simple, the idea is something like this that you have your original circuit say suppose this is your original circuit what I am going to do is that. So, this circuit you can see there are 3 inputs and 1 output. So, what I am going to do? I am going to add some additional input which is called key to the circuit.

And I am going to add certain part of the logic into my circuits using this. As a result, what is going to happen, this particular circuit has now this A B C the original input and few keys and this particular circuit only will work for a specific value of keys.

For example, you see here. So, if you the original circuit you have the A B C so it is basically you can see there are 3 AND gate and there is a OR gate. So, if at least 2 of the input is 1 then the output will be 1 you can understand that, because if A and B is 1 then this will be 1 and then it will be output 1. B and C is 1 then this output will be 1 and if A and C is 1 this output will be 1.

So, that's not so important so this is the circuit. Now, what I am going to do I just added it to XOR gate here. So, and then I just added a key. So, now, you see if I give my key equal to 1 here whatever the output is coming here it will get toggled. So, that is the behaviour of the XOR gate.

So, now suppose you are given the input A, B and C as 1 1 0. So, then this since this two are equal to 1 so this will be 1 and since this is 1 this will become 0. So, in this circuit it is 1 means output is 1 and here it is 0. So, it is 0 the output will be 0 because the other components are 0.

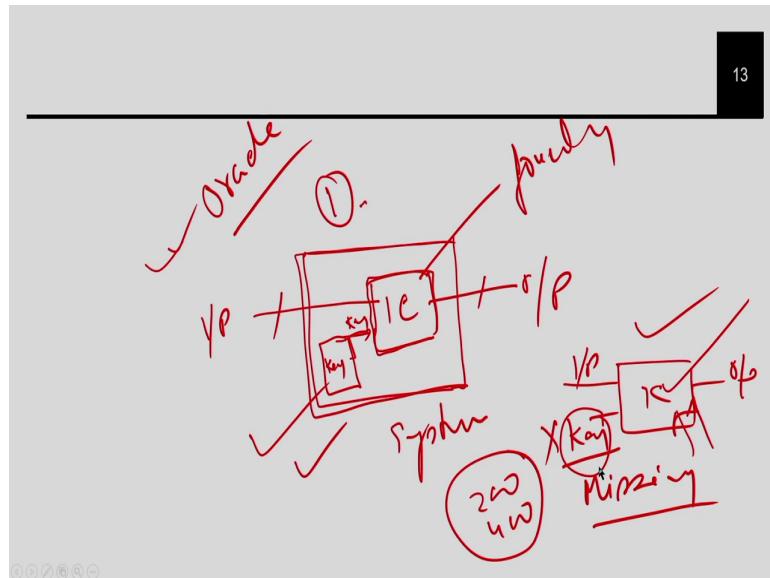
So, what you can understand here is if I give a wrong key K1 then my output will be different, but if I give 0 here this 1 will pass and it will come 1 and output will be 1. So, that means, the correct value of this K1 is actually 0 and similarly you can understand that K2 value is also 0, correct value. But this K1 and K2 have four possible values right 1 0, 1 1, 0 1 and 0 0. So, there are 4 possibilities of these keys and among them only one of the values is correct.

So, what I am going to do I am not going to give this circuit to the foundry I am going to give this circuit to the foundry as the layout and what I am going to do I am not going to reveal the correct value of this key because that is only, I know.

Now think about if there are two keys so how he can do that he can extract suppose by reverse engineering he is able to extract this circuit and then he has to try 4 such combinations and if he applies 4 such combinations then only probably, he can identify this is

something the correct key, say by some way. But, now think about if the key size is 200. So, how many possibilities you have to try, 2 to the power 200 which is impossible.

(Refer Slide Time: 16:45)



So, that means, the basic idea is that I am going to create my circuit like this that I have the actual IC which will come from the foundry, so this is only I am going to give. So, and this is my input which is the actual input of the circuit and this is the output of the circuit. And what I am going to do I am actually going to have an extra input which is called key and this key I am not going to give to the foundry.

And then once I get the IC from the foundry after fabrication, I will create my system like this. So, I will put a memory which is something tamper proof memory where I am going to store the key. So, and then I am going to package like this and this is my system that I am going to give to my customer.

So, once you buy it from me the key is already embedded your system it is a tamper proof. So, tamper memory that means, you cannot go into and check what is the correct value of the key. So, if you buy it from me then circuit will work because the, I have already embedded my key into the actual circuit that I am selling to you.

But if this IC which is something if its overbuilt or after reverse engineering it is produced the pirated IC so this IC has 2 inputs and the key input and the key is missing. So, this key is missing for the fab lab because they do not have this access and you it is very difficult to identify if the key size is a 200 400 or at that range, so you cannot use it.

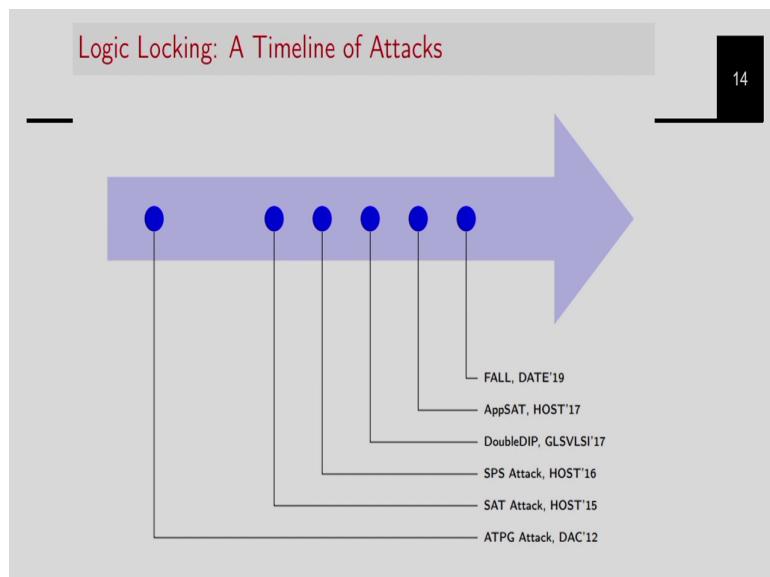
So, what I did is actually this is the difference if you buy from the original vendor and if you buy from the pirated because this is missing and you cannot use it, but if this key is already embedded, the things I am selling to you, you can just use and the customer has no difference

he does not know whether there is a key at all. So, because he does not know what is the key and what is the value of the key, he doesn't need to know it is actually abstracted from the customer.

He will just see that his IC is working, but if you try to sell this because you do not know the key your IC will not work. So, so this is a very simple, but very powerful technique because here with this technique I will give my design to you, you can reverse engineer you get the netlist you produce ICs, but you cannot use it.

So, the all this problem of IP piracy and IP overbuilding can be stopped using this hardware trojan it cannot be completely done by this, but the other two things can be easily done. And that is something is really important and this actually become very important in the context of this IP piracy. And overproducing as a result there are a lot of attraction in this in the last 5 or 10 years.

(Refer Slide Time: 19:37)



So, you have to what is the important point here is that it is something, how to now the question comes into here that yeah, I want to lock it how to lock it. So, where which are the places, I am going to lock? What is the would be my locking logic? So, you and what is my key size? So, so many questions come.

And specifically on the logic technique this locking technique there are many such things come and if you look into this in the last 5 to 10 years history there are so many important techniques come here like, random logic logging that was the initial logic locking where you just add this kind of XOR gates. Then there are many other techniques like just anti-SAT and then you have this SFLL, delay locking there are many techniques.

I am not going to detail of that because that is a completely different subject altogether, but what I can just tell you that there are a huge set of work happen in this particular area how to lock your circuit efficiently. So, that was something very active research area for last few years.

And obviously, once you have this locking, attacker will not sit idle. So, they try to see that if you have this key, can I extract the key because I know the internal circuit because the netlist of the circuit and I can buy one copy of this so from the market the genuine IC. So, that is called oracle.

So, I will buy one genuine IC from the market so that if I give an input, I know what is the correct output only do not know I cannot extract the key, but the actual correct input output functionality I can get it from the oracle. So, now I have the internal of this by doing analysis checking the actual output and analysing this I can probably extract some of the keys. So, that was the attack technique.

So, this is kind of a wire between this attack versus defence. So, you come up with some defence technique and then the attacker will try to say that even if you lock this way, I can actually extract this using this technique. So, this is something a very interesting domain where this work is happening.

So, with this I try to conclude this class and then what I am going to show that this all this locking actually happening in the logic level so the gate level circuits. So, the question that is important in this course context is that, you can do it in the gate level can you do the whole same thing at the higher level of abstraction, at the RTL level or even high-level synthesis.

So, can my high-level synthesis generate an RTL which is already locked. So, that will be great right because you do not have to do anything you do not have to apply any new technique to lock your circuit. And specifically, once you do these things in higher abstraction level your circuit size will be smaller.

And as a result, you can actually have a more control or the semantic information in the circuit and you can your locking technique can be very interesting which cannot be done at the gate level design ok. So, so that with this I conclude this class and in the next class I am going to see how this logic locking can be done during high level synthesis.

Thank you.