ECEN 426: Security of Embedded Systems **Fall 2021**

Lecture 2: Hardware Trojans (Part 2)

Lecturer: JV Rajendran *Scribe: Sarah Wauson*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

2.1 Combinational Trojans

1. A Combinational Trojan can only be triggered when a specific gate gives a specific output. The odds of triggering the trojan will be dependent on the gate being used and if a 0 or a 1 triggers the Trojan. In Figure 2.1, the Trojan is triggered when A and B are 0. Therefore, the probability of triggering this Trojan is 1/4.

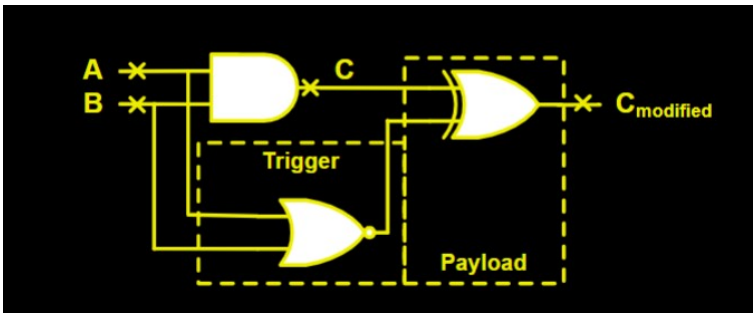

Figure 2.1: Combinational Trojan

2. However, it is desired that trigger is rare. This is due to the fact that it will be harder to notice a Trojan during chip testing if it isn't easily triggered. This can be done by increasing the inputs.

3. Instead of increasing the inputs, it may be better to instantiate a Sequential Trojan instead.

2.2 Sequential Trojans

1. A Sequential Trojan is more rare to trigger than a Combinational Trojan. In Figure 2.2, every time inputs p and q are both one, it add one to the counter. When all the values of the counter become 1, the Troan is triggered. The probability of triggering the Trojan is now depending on how many bits the counter takes in.

2. The goal with Trojans is to make the triggering of it a rare event.

3. Another reason the Sequential Trojan is more sophisticated than the Combinational Trojan is because a counter will be easier to hide opposed to many additional gates.

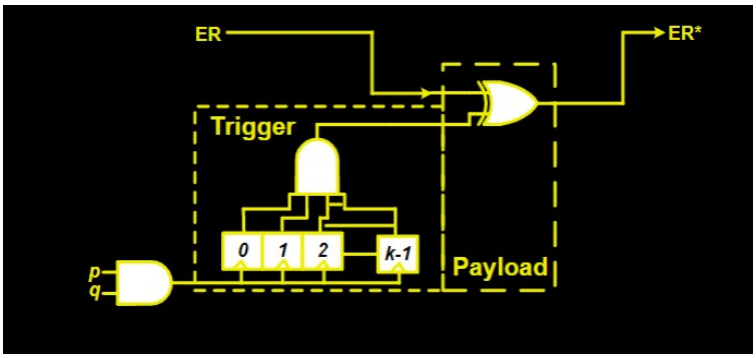

Figure 2.2: Sequential Trojan

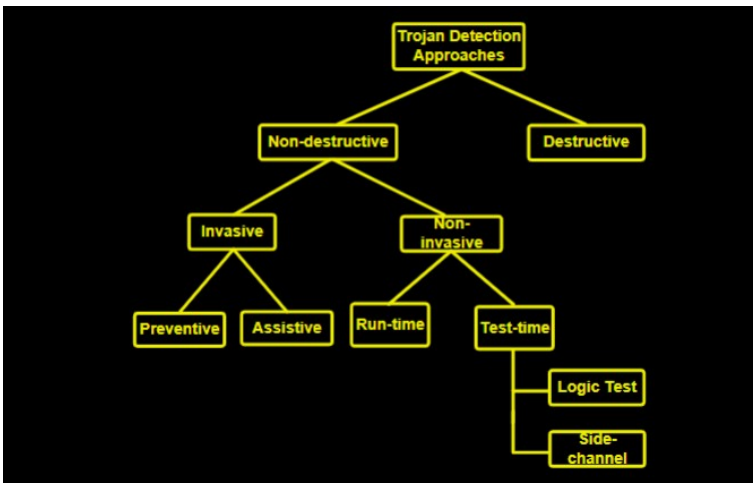

Figure 2.3: Sequential Trojan

2.3 How Do We Detect Trojans?

1. Destructive
   (a) Reverse-engineer the chip and you may be able to tell where the Trojan was inserted.
   (b) However, reverse-engineering techniques may not be advanced enough in order to completely find all Trojans.
   (c) Reverse-engineering techniques are not feasible for a chip that is mass produced. It would be extremely difficult to reverse-engineer every chip that was manufactured.
2. Non-Destructive
   (a) Invasive: the design is modified in some way in order to detect Trojans.
      i. Preventative
         A. Certain techniques are used to prevent an attacker from placing a Trojan in the first place.
         B. An example of this would be Logic Locking. An attacker cannot place an attack if they do not know the key to the locked logic.

ii. Assistive
   A. Room is left for the attacker to place a Trojan, but points are placed in the design that would notify the designer of a Trojan. Essentially, the design would ambush the attacker's Trojan.
   B. An example of this would be placing several monitored outputs within the design to ensure a Trojan was not inserted.
   C. Another example would be to activate the Trojan in order to see the effect it has on the chip.

(b) Non-Invasive: the design is not modified. Other techniques are used to detect Trojans.
   i. Run-Tine
      A. Run-time can be monitored in order to detect errors that are in a chip. The run-time may change if a Trojan is added and the system should be able to detect if it varies from the normal run-time.
      B. However, some Trojans can run within the original run-time so, they cannot be detected this way. A technique that may also used to detect these Trojans is Concurrent Error Detection. Concurrent Error Detection is cyclic redundancy checks. It checks the sums and outputs of parts of the circuit in order to detect an error and correct it.
   ii. Test-Time
      A. Logic Test: This test is used to make sure that a chip is properly working. It can also be a method that is used to search the chip for Trojans. It is just a matter of observing inputs and outputs. If the output varies from desired, a Trojan may have been found.
      B. Side-Channel: This method monitors the power and timing of the chip. Generally, power consumption can be used to pinpoint which processes are currently undergone in a chip. By applying certain inputs into the chip, you can measure the power consumption and use that to help ensure a Trojan was not placed.

2.4 Statistical Analysis

In favor of saving money, running extensive tests on a chip isn't feasible. This is why a much faster technique is needed in order to defend against Trojans: Statistical Analysis.

Trojans are assumed to be placed at points in the circuit that make them rare. The solution to this would be to take the rarest scenarios in the circuit and place more activity there to fight off an attacker. The key in this type of defense is to use Statistical Analysis in order to define the rare points of a circuit. Depending on the circuit, the probability of each input and output nodes can be calculated in order to figure out which values are rare.