On Reliability Trojan Injection and Detection

Abstract

Hardware design houses are increasingly outsourcing designs to be manufactured by cheaper fabrication facilities due to economic factors and market forces. This raises the question of trustable manufactured products for highly sensitive applications. One such type of trust issue is the possible incorporation of Trojan circuits into the IC with the goal of tampering with IC reliability and hastening the aging of the chip. In this paper we present examples of such reliability Trojans and describe testing approaches for detecting these reliability tampering attempts.

Keywords

Reliability Trojans, Trust, Mean time to Failure, Electromigration, Rare Events

1. Introduction

With the increasing number of fab-less companies, more designs are being shipped to off-shore foundries for cheaper manufacturing. As most of these off-shore foundries are located in foreign countries, trust issues such as security and integrity of the product in semiconductor manufacturing have become an important concern. A non-trustworthy fabrication facility can re-engineer the original design and incorporate malicious hardware without affecting the normal behavior and with almost no increase in the area of the IC. Such types of malicious hardware circuits, that are called “Trojans,” can be triggered by certain special events and potentially cause the IC to fail or operate in an undesirable manner. ICs that intended to be used in sensitive commercial products and in defense related devices are of the highest concern [1]

Malicious hardware can be a small combinational or sequential circuit. Combinational malware circuits can be triggered by a set of special logic signal combinations or by inputs through a hidden port. Sequential malware circuits can be triggered by a series of input patterns that may be generated as a result of special events. Dynamic logic Trojans can also be incorporated where the trigger changes with the clock. An example of a Trojan circuit discussed in this paper is shown in Figure 1. The original Verilog code would not include the multiplexer (MUX), or the other extra logic. A malicious circuit is added as shown, where the MUX selects the good value during normal circuit operation. When a trigger pattern appears, this circuit is activated and a faulty output is produced every clock cycle.

Trojans can also be used to tamper with circuit reliability. Hardware Reliability Trojans modify the operation of the circuit and accelerate device failure using the hardware properties of the circuit such as reliability and temperature dependent aging. This causes chips to fail well before their mean time to failure (MTTF). Long term device reliability problems primarily result from (i) Electro-Migration (EM), (ii) Negative Bias Temperature Instability (NBTI), (iii) Gate Oxide short, and (iv) Mechanical Stress effects.

It is possible to tamper with device reliability using simple layout modifications. A layout modification to a metal line in a standard cell, for example, is typically not thoroughly tested during normal manufacturing tests and can be detected only by applying special trigger patterns. It has been proven to be very difficult, if not impossible to expose such reliability tampering. In the next section we show how an aluminum line can be made to experience a higher level of Electromigration (EM). The reduction in the lifetime of aluminum lines due to EM has been analyzed in [2] where the increase in Mean Time to Failure (MTTF) and Deviation of Time to Failure with decreasing line length has been studied. The dependency of the EM process on grain size and line dimensions has been presented in [3][4][5]. In this paper we study the aging process due to EM effect, Hot-electron effect, NBTI and Gate-oxide failures [8][9].

![Figure 1: A Mealy machine with a Trojan circuit](image)

Figure 1: A Mealy machine with a Trojan circuit

Hardware Trojans are far more difficult to detect (than software viruses) due to limited controllability of the nodes in a design [30]. Typically such malicious circuitry is designed so that it is not detected by conventional scan-based testing. This can be achieved by designing the Trojan circuit so that it is activated by rare events that would normally not be triggered by ordinary test procedures. Identifying such rare events and being able to mimic the exact conditions needed to trigger the malicious circuitry with no information of its operation is very difficult. In the context of test pattern generation, detecting the infrequent occurrence of Trojan activity is akin to hard-to-detect faults [10]. Faults that are detectable by random test vectors are considered easy to detect; the rest are targeted by deterministic test pattern generators. When a deterministic test pattern can detect a fault with few backtracks, it falls into the easy to detect category. The remaining ones are considered to be hard to
detect. The rare events must be identified and targeted for pattern generation to increase excitation of these rare events. In Section 2 deals with test strategies dealing with detecting trojans in combinational and sequential circuit. In Section 3 we present several circuit techniques to create reliability Trojans that affect different reliability concerns. We also present manufacturing test conditions to excite reliability Trojans. In Section 4 we present rare event identification, unmasking of device reliability tampering and acceleration techniques

2. Related work

There have been a few recent works devising test strategies to detect generic combinational and sequential Trojan circuits. In [11][12] the authors propose a partitioning technique to isolate regions to target Trojans but they only consider malicious inputs to flip-flops. The use of side-channels for building IC fingerprints for Trojan detection was proposed in [13] which has demonstrated the feasibility of the basic approach using simulations performed for small circuits with Trojans that are approximately 1% of the size of the circuit. The effectiveness and the limits of this approach for larger circuits are still not clear. Even though the proposed fingerprinting is novel, it is based on combining well-established techniques [14][15][16] such as side-channel cryptanalysis and side-channel based template attacks, IC defect detection and localization using IDDQ and IDDT analysis as well as signal detection and estimation theory. In [17] the author uses a path delay fingerprint to detect Trojans in the circuits, but this can be defeated through a suitable transistor sizing. Various other mechanisms have been suggested that target Trojan detection [32][33][34][30][35]. None of the above mentioned techniques have dealt with reliability Trojans that are not only triggered by rare patterns but also depend on how other parameters, such as temperature, can accelerate the aging.

3. Circuit Aging Techniques & Trojan circuits

Circuit aging refers to the degradation of a circuit over time. The degradation time is typically a few years, but can be reduced to a few months under worst-case conditions. Circuit aging has become a dominant design factor with technology scaling. In today’s ultra-deep sub-micron technology, circuits are highly vulnerable to hard failures due to EM or Time-dependent dielectric breakdown (TDDB) and timing failures due to NBTI and Hot-Carrier Injection. Hence, hardware Trojans that target such circuit vulnerabilities have to be analyzed and suitable manufacturing test strategies devised.

3.1. Electromigration Effects

EM is a term defining the transport of mass through metal under stress due to high current density. It has been shown that EM is an important contributor to interconnects’ wear out leading to electrical opens [18]. The flow of current through a metal conductor creates a wind of electrons in the opposite direction with momentum proportional to the amplitude of the current. With high momentum, the electrons tend to dislodge the metal ions creating vacancies. These vacancies form voids over time leading to high interconnect resistance or interconnect opens [19][20]. Due to its dependence on mass transport, current flow and temperature, the mean time to failure of an interconnect is given by [18]:

$$MTTF = A J^n \exp \left( \frac{Q}{k_B T_{\text{metal}}} \right)$$  

where $MTTF$ is the mean time to failure, $A$ is a constant dependent on metal interconnect geometry, $J$ is the current density in Ampere/m², $Q$ is the activation energy of the interconnect material ($Q_{\text{Cu}} \approx 1.2 \text{eV}$), $k_B$ is the Boltzmann transport constant and $T_{\text{metal}}$ is the metal conductor temperature. $T_{\text{metal}}$ is dependent only on the reference silicon junction temperature ($T_{\text{ref}}$) when self-heating is assumed to be very small. $T_{\text{ref}}$ is usually between 100°C to 120°C.

In a circuit, the interconnect current density depends on the interconnect capacitance, resistance, length and the driving buffer size. The temperature dependence of such aging is also an important factor. As interconnect current density increases, the temperature increases with it and this contributes to aging due to the failures rate’s exponential dependence on the temperature. Any circuit that has a big enough buffer driving a very thin and long interconnect line will produce a high current density in the line leading to faster aging.

An example of a Reliability Trojan relying on the EM effect to cause fast circuit aging in any node in the circuit is shown in Figure 2.

Figure 2: Reliability Trojan circuit targeting Electromigration

In the above example, for normal operation the trigger input $T$ is 0. Hence a 1 is present at input A of G4 and a 0 at the second input B. Since G4 is an OR gate, a 1 at any of its inputs will produce a 1 at the select input of the MUX providing a correct value to DFF. Net $n1$ does not change with the clock. Hence no current flows through the interconnect line and thus, there is no Electromigration effect.

When the Trigger input $T$ is 1, input A of G4 is 0. Input B of G4 changes with the system clock. With the positive edge of the clock a correct value is selected at the MUX. Current is pumped into $n1$ every clock cycle leading to aging in the presence of high temperature. It must be noted that the select input to the MUX does not change immediately with $T$ going to 1. Even after $T$ is 1, the correct value is registered at DFF, but with time, electro migration causes an open in $n1$ leading to a faulty output.
Figures 3 and 4 show plots for MTTF as a function of the current flowing through the wire (e.g., net $nJ$) and the temperature. It can be seen that the MTTF has an inverse dependence on the current density and exponential dependence on temperature. Also, the MTTF reduces with an increase in the buffer size when all other parameters remain constant.

The MTTF effect in digital circuits has been analyzed in [23][24][25][26]. Several models have been proposed to explain the mechanism of NBTI based on the Reaction-Diffusion model [27][28][29]. It has been shown that the traps generated due to the applied negative bias threshold voltage degradation can be explained by the following equation:

$$\Delta V_{TP} = -\frac{q^2t}{C_{ox}} \quad (2)$$

where $t$ is the time required for the NBTI stress to cause the corresponding change in $V_{TP}$. Figure 5 shows the increase in $\Delta V_{TP}$ with stress time.

3.2. Negative Bias Temperature Instability (NBTI) Effects

NBTI is a circuit aging condition under which the P-type transistor of a gate is under high stress inducing an increase in the threshold voltage of the device. Interface traps are generated under negative bias conditions at the gate (i.e., $V_{GS} = -V_{DD}$) that at a high temperature cause reliability problems. Interface traps are formed due to crystal mismatches at the Si-SiO$_2$ interface between the gate oxide and the substrate. Interface traps are physical defects with energy distributed between their valence and conduction bands [22]. They manifest as an increase in absolute threshold voltage and reduce the ON current of the PMOS. An increase in $|V_{TP}|$ makes the device slower and threatens the reliability of the circuit. NBTI predominantly affects PMOS transistors degrading drive currents and noise margins. With the scaling of gate oxide below a thickness of 4nm for future generations, NBTI effect becomes more pronounced.

The concept of stress time is well explained for the cases of static and dynamic NBTI. A NOR gate has two P-type transistors stacked (two transistors are said to be stacked if they are in series). In Figure 6(b), the transistor $M_{p1}$ is under stress as its $V_{GS}$ is $-V_{DD}$. This causes the $|V_{TP}|$ of the $M_{p1}$ to
increase. A NOR gate like the one shown in Figure (c) has the worst case timing when the PMOS closest to V_{DD} in a stack (M_{p1}) is switching. In the presence of NBTI, this switching speed is further reduced if the transistor is under stress for a period of time. The impact of NBTI on stacked P-type transistors can be used to create a Trojan circuit that will quicken the aging process of the circuit.

Figure 6(a) shows the Trojan circuit that we have developed. During normal operation Tbar is 1 and hence the correct value is registered. When the trigger input T is 1, the k-bit counter starts counting and its value is compared to a set value that determines the static stress time. Once the counter reaches the set value, the switching delay of NOR gate G8 increases affecting the reliability of the circuit.

3.3. Gate Oxide Breakdown Effects

Gate Oxide breakdown refers to the destruction of gate oxide of a transistor when a conduction path is formed between the gate and the source/drain region [6][7]. A conduction path is formed due to the formation of interface traps within the gate oxide that overlap each other to connect the gate terminal to the substrate. Once a conduction path is formed, it heats up the device leading to thermal damage and generation of more interface traps. This forms a positive feedback loop thereby breaking down the gate dielectric. This effect is exacerbated with a thin gate dielectric.

Figure 7: $T_{BD}$ variation with a spike voltage at the gate

Interface traps that cause dielectric breakdown are formed when a high electric field is applied at the gate terminal. A high electric field creates a large tunneling current through the gate oxide. Electrons with high kinetic energy in the substrate transfer this energy to the holes that tunnel into the gate oxide, thus creating traps. This is seen even when there is no potential difference between the source and the drain regions. When such a potential difference exists, the effect is more pronounced leading to increased generation of traps. Through experimentation, it has been found that for a transistor in 45nm technology with gate dielectric thickness of 4nm, dielectric breakdown happens at a field of 5MV/cm². This positive cycle inducing mechanism is called time dependent dielectric breakdown (TDDB)[7].

Figure 8: Reliability Trojan circuit using TDDB effect

Figure 8 shows an example of a Trojan circuit that causes device aging based on the TDDB effect explained above. During normal operation, Tbar is 1 and hence a correct value is registered at PPI. When the trigger value T is 1, the circuit is made to fail through dielectric breakdown of the second inverter. The transistors in the second inverter are designed to have very thin gate dielectrics and are hence susceptible to breakdown. The first inverter is carefully designed such that the extra capacitance $C_{trig} >> C_{load}$. When T is 1, $C_{trig}$ forms a charge pump and induces a higher than V_{DD} voltage on node N1 causing $M_{p2}$ and $M_{n2}$ to breakdown. Figure 10 shows the amount of voltage spike that can be injected into a gate by forming a charge pump using $C_{trig}$. For the purposes of this evaluation, we used 45nm predictive technology models [36]. Circuit simulations were performed using HSPICE. The area was estimated using Synopsys Design Compiler and Cadence Encounter tool suite.

Table 1: Trojan Circuit Area and Power Consumption

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Number of gates</th>
<th>Original Circuit</th>
<th>Trojan circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Area (μm²)</td>
<td>Power (nW)</td>
<td>Area (%)</td>
</tr>
<tr>
<td>c2670</td>
<td>1269</td>
<td>1472.26</td>
<td>0.06</td>
</tr>
<tr>
<td>c3540</td>
<td>1669</td>
<td>2740.03</td>
<td>0.047</td>
</tr>
<tr>
<td>c5315</td>
<td>2307</td>
<td>3369.69</td>
<td>0.034</td>
</tr>
<tr>
<td>c6288</td>
<td>2416</td>
<td>7683.24</td>
<td>0.033</td>
</tr>
<tr>
<td>c7552</td>
<td>3513</td>
<td>4038.76</td>
<td>0.012</td>
</tr>
</tbody>
</table>

Table 1 shows the added area and power due to the Reliability Trojan circuits discussed above as a percentage of the area and power of the original circuit. The area increase is less than 0.05% of the original area and it decreases with the size of the design. The increase in power consumption is even smaller. It can be concluded that it is easy to tamper with a design to shorten its lifetime. Today, burn-in test systems use a simple toggle coverage metric that only considers whether a node has toggled [31]. It does not keep track of how many times a particular node has toggled. With such a simple-minded metric, it is very difficult to expose reliability
tampering. In the next section we will develop node toggle coverage targets and test patterns to achieve them.

4. Trojan Identification and acceleration

The effectiveness of Trojan detection relies on the following factors: (a) detection resolution - i.e., the number of gates and activity level of Trojan, (b) detection rate - the time to detect identifiable Trojans, (c) acceleration rate - the fraction of latent Trojans activated during manufacturing test, (d) false detect rate - the number of false detections over a set of identifiable Trojans, and finally, (e) the implementation overhead - design, fabrication and test overhead.

If certain nodes are rarely set to a specific value, they are candidates for rare events. This may frequently be the case for control signals in a control data-flow circuit. The nodes that have rare single events are shortlisted.

Once the rare events have been identified, test patterns for exciting them are generated. The pattern generation problem differs in goal from automatic test pattern generation for stuck at faults in that (i) only excitation is needed, error propagation is not required and, (ii) multiple rare events may be targeted at once.

It is unlikely that a single or even a limited number of excitations of rare events will precipitate a failure during stress testing in burn-in chambers where the chips will be subjected to high voltage and high temperature stress conditions. To that end, we propose, the use of n-detect test sets that have been employed in chip test. By setting a reasonable value of n, the number of times a rare event must be excited, and repeating the resulting patterns in a loop we can accelerate the unmasking of Reliability Trojans.

5. Results

We validated our approach using the Table II shows the number of rare events in the given circuits and the number of test patterns that can excite them. An event is said to be rare if the number of times a node registers a value is less than certain X% of the total number of patterns applied to the circuit. 100,000 random patterns were applied to check for rare event nodes in this experiment. Table II shows the number of rare event nodes that are triggered by less than 1% and 0.1% of the total number of patterns applied. The bigger circuits have one or more rare events nodes that have a very low probability of being activated, thus making any Reliability Trojan that targets one of them, hard to detect. In some of these circuits, e.g., C6288, all the rare events listed in the 0.1% column of Table II were found to be part of redundant logic that will be removed during circuit synthesis. For such circuits a single internal signal would not be an ideal trigger for a Trojan, instead, a combination of two (or more) internal signals could provide a suitable trigger for a hard to detect Trojan. For the c6288 circuit we found a combination of two somewhat rare events (i.e., activated in less than 5% of the 100,000 patterns applied) such that only one of the 100,000 patterns has triggered this combination. Similar analysis was performed on other circuits where combinations of two rare events were found to be triggered by fewer then 5 patterns out of 100,000; thus allowing us to introduce into the circuit a hard to detect Trojan. Such rare events were also found to be present in Cryptography ASICs such as AES. With the presence of these rare events, triggered Trojan circuits can enable mechanisms that not only fail during encrypting/decrypting but can also be used to modify the circuitry to use the key itself as the trigger.

Figure 9: Rare event identification and test pattern generation technique

Figure 9 shows our methodology involving rare event identification and test pattern generation. Each step in our methodology is explained in detail below. Rare event detection has been proposed previously in [35]. However, the attack model they use is different from ours. They are concerned about payload delivery; whereas our primary concern is trigger.

Identification of rare events is initially based on logic simulation on RTL description. Both random and architectural verification patterns are simulated.

The RTL nodes are instrumented with simple counters to count the frequency of the nodes being set at specific value or a pair of values. The counts are termed as single event frequency \( O_{\text{freq}} \) and \( I_{\text{freq}} \). As a first step we are looking into single events. When combination of internal signals become part of the candidate set, the identification problem becomes exponential in nature.
### Table 2:
**PATTERN GENERATION FOR TRIGGERING RARE EVENTS**

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Number of Gates</th>
<th>1% of total patterns</th>
<th>Rare event Nodes</th>
<th>No of Patterns</th>
<th>0.1% of total patterns</th>
<th>Rare even Nodes</th>
<th>No of Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1355</td>
<td>546</td>
<td>96</td>
<td>792</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c1908</td>
<td>880</td>
<td>98</td>
<td>806</td>
<td>18</td>
<td>140</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>c2670</td>
<td>1269</td>
<td>19</td>
<td>51</td>
<td>15</td>
<td>20</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>c3540</td>
<td>1669</td>
<td>45</td>
<td>220</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c5315</td>
<td>2307</td>
<td>4</td>
<td>40</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c6288</td>
<td>2416</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c7552</td>
<td>3513</td>
<td>49</td>
<td>432</td>
<td>28</td>
<td>240</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>s13207</td>
<td>8989</td>
<td>229</td>
<td>646</td>
<td>57</td>
<td>296</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s15850</td>
<td>8206</td>
<td>179</td>
<td>654</td>
<td>46</td>
<td>293</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AES</td>
<td>9292</td>
<td>188</td>
<td>847</td>
<td>4</td>
<td>302</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

6. References


