International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Design and Implementation New Techniques to Prevent Trojan Thread in Parallel CORDIC

Mohamed. M. Elgazzar[1], Mahmoud Maher El-Sayed Mohammed[2]
[1]Ass.Prof., Higher Institute of Computer Science and Information System, Cairo, Egypt
[2]Engineering College, Cairo University, Giza, Egypt
Email address: [1]m_elgazzar1961@yahoo.com, [2]mahmoudmaher2011@gmail.com

Abstract*—Trojan thread affects the Integrated Circuit (IC) and changes its operation. Trojan thread is injected to the designs that have no security or weak security. After inserting Trojan thread, the IC operates in different behavior other than the intended behavior. The aim of the Trojan thread is to defect the IC operation or leak information from the IC to another IC through established link wired or wireless to the attacker. The presence of Trojan thread in IC lead to high losses and increase the defective products. There for it is very important to decrease the effect of the Trojan thread. In this paper we insert Trojan thread inside Coordinate Rotation Digital Computer (CORDIC) and see its effect and represent techniques to increase the design immunity against the Trojan thread.*

***Keywords*—** *Trojan Thread, Voting Algorithm, Lockup Table Technique.*

I. INTRODUCTION

The IC security is very important nowadays due to the spread of the Computer devices in our life applications. Without security we will be under the mercy of attackers. The IC with design weakness in security or implemented with hardware defects is subjected to hardware attacks. Nowadays, multiple companies are collaborating in designing and manufacturing the IC, so this collaboration with no security embedded in the IC will make the IC subjected to the hardware attacks. The security issue appears clearly in the Internet of Things (IoT) chips due to the cost and power restrictions. This power and cost limitations will decrease the security level. Cisco expect that the number of IoT ICs connected to the internet by 2020 will be 50 billion ICs [1]. The less power and low-cost IC are the targets for all the IoT companies. These limitations in power and cost prevent us from using software security stack to run on weak processor in the IC. The old ICs that already manufactured in the beginning of IoT revolution miss the concept of security in its design. These old ICs can be the open door to the attackers to attack out IC remotely.

The IC consists of multiple intellectual property (IP) connected together. These IPs are delivered from third party companies. These third party companies may depend on outsourced engineers for design or testing. Tools from different companies are used also in designing or manufacturing. The Trojan thread maybe inserted in the Register transfer level (RTL) of the IP, also the simulation tool or manufacturing tool can insert the Trojan thread, or the testing engineers can inject Trojan thread inside the IC [2], [3].

If the IC didn't perform its task as described in the design documents after fabrication then we are under hardware attack. This hardware attack can be simple that occur every time with the same pattern, or it can be randomly appeared. This randomly appearance will make the debugging more difficult. The activation of hardware attack can depend on internal pattern appearance or from external input. The internal pattern like observing certain sequence of input data or defecting range of input data. The external trigger can come through pad input from another IC. This Trojan thread need to be isolated during the design of the chip. In the next section we will discuss the different techniques to isolate or detract effect of the Trojan thread.

II. HARDWARE ATTACKS ELIMINATION

A. Dynamic Permutation

This solution changes the order of the data from the input and feed permutated data to the core as shown in figure 1. If the Trojan thread use certain pattern to start the attack, this permutation will change this pattern. This will give the core immunity against the hardware attack. The permutation core can choose from different permutation patterns. These patterns will make it harder for the Trojan thread to attack the core. We can use random number generator to control this dynamic permutation patterns. If the permutation core detects hardware failure then it will request new permutation pattern with random number generated. This method will make the data encrypted inside the IC, and make the IC immune against the hardware attacks [4].

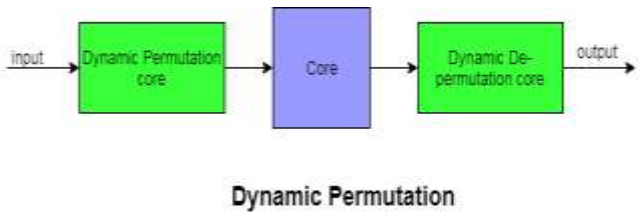

Fig. 1. Dynamic permutation technique.

B. Network on Chip

The network on chip (NoC) is network-based architecture. NoC connect modules together on the IC. We Can use the concept of the NoC to remove the effect of the trojan thread. NOC is based on routers that is implemented in the IC, so we can treat NoC as network between the modules in the IC connected together through routers. The function of the router is to understand the data packet header information, then forwarding this packet to the next routing router. If there is

Mohamed. M. Elgazzar and Mahmoud Maher El-Sayed Mohammed, "Design and Implementation New Techniques to Prevent Trojan Thread in Parallel CORDIC," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 1, pp. 21-25, 2018.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

change in the module due to hardware attack, then the header of the packet will be affected from this hardware attack. The router will see errors in packet header come from infected module. At this case, the router will stop the data packet come from the infected module and isolate it, so this infected module will not affect rest of the IC [5].

C. Voting Algorithm

This technique depends on adding multiple cores which make the same task on the integrated circuit. These cores are implemented in different companies. This design technique will decrease the effect of hardware attack on the IC. These different cores are connected to controller which choose the agreed outputs from the three cores and connect it to the global outputs. If there is attack happen on one of these cores then the controller will see difference between the outputs and will make voting between different outputs and choose the majority outputs from the cores' outputs, then connect it to the global outputs.

D. Lockup Table

The lockup table contains few expected outputs from model of the core, or from design calculations. The controller will be connected to two cores with the same function from different companies. The controller will compare the outputs from two cores according to the expected values in the lockup table. If there is difference between the values in the lockup table and the outputs from any of the cores then the controller will ignore all the outputs from the infected core and connect the main outputs to the Core which its outputs match the lockup table values.

III. TROJAN THREAD ATTACK

We will use the Rotation Digital Computer (CORDIC) as an example of module that will be affected by Trojan thread. The CORDIC is used to calculate the trigonometric functions for the input angle [6], [7]. The Trojan thread inserted will affect certain range of input angles. This range in our hardware attack design is from **45° to 60°**. If the input angle locates in the attacked range then the Trojan thread will affect the CORDIC outputs by applying zeros to the outputs. The Trojan controller randomize number which will calculate the appearances of blocked input angles. These number will make the Trojan thread wait before starting attack. This randomization will make the debugging more complex.

The Trojan thread architecture as shown in figure 2 is simple counter that counts the number of appearances of blocked input angles, this counter is followed by controller which compare the count with the random count feed from random generator. If the counter value equals the random count, then it will activate multiplexer which will connect the outputs to zeros.

Figure 3 shows the outputs of CORDIC after inserting the Trojan thread inside it, when there is input angle in the range of blocked angles, the CORDIC outputs become zeros.

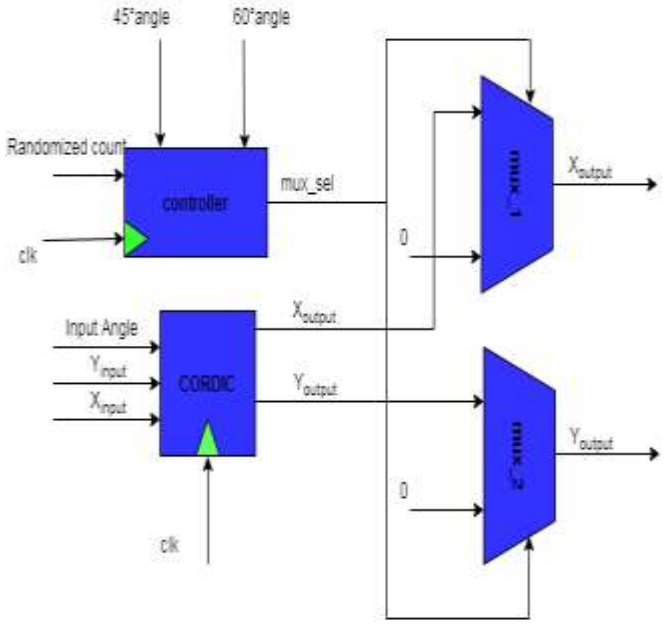

Fig. 2. Trojan thread architecture.

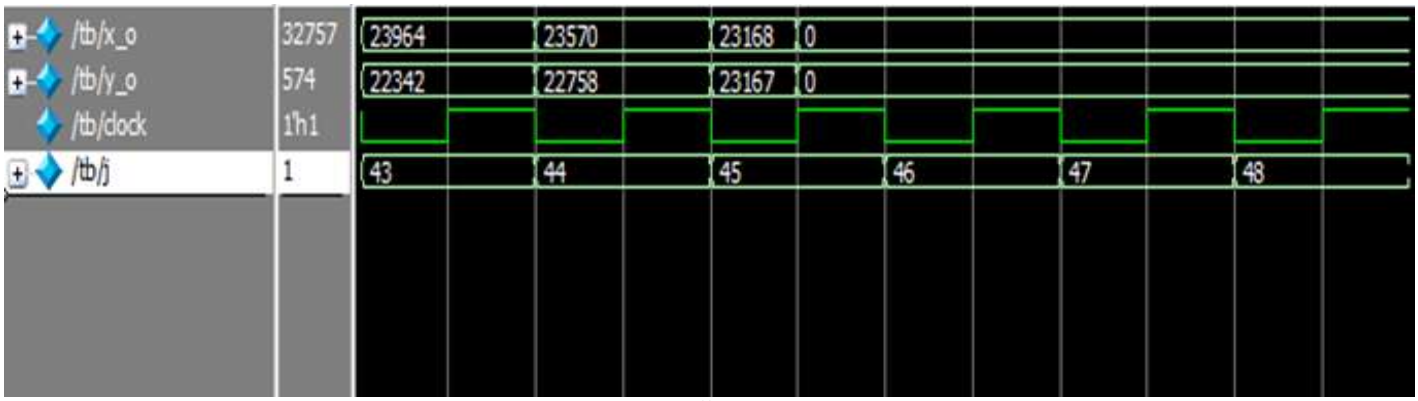

Fig. 3. Trojan thread affects CORDIC output.

Mohamed. M. Elgazzar and Mahmoud Maher El-Sayed Mohammed, "Design and Implementation New Techniques to Prevent Trojan Thread in Parallel CORDIC," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 1, pp. 21-25, 2018.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

IV. TROJAN THREAD EXCLUSION

A. *Voting Algorithm*

Using the voting technique, we will make simulation using three different CORDIC cores, the first and the third cores are normal cores without any hardware attacks. The second core will suffer from Trojan thread inserted in it. The outputs of the infected CORDIC will be zeros if the input angle in the range of blocked angles [$\mathbf{45^\circ}$ **to** $\mathbf{60^\circ}$]. The voting controller will compare between the three outputs and if there is one CORDIC core outputs different to the other two cores outputs then it will choose one of the agreed CORDIC cores outputs and connect these outputs to the global outputs. Figure 4 shows the voting algorithm architecture. In our case First CORDIC and the third CORDIC will have matched outputs, and second CORDIC will have zero outputs. The controller will connect one of the first or third CORDIC core outputs to the global outputs and ignore the second CORDIC core outputs as its outputs is defected from the Trojan thread.

The simulation results are represented in figure 5. In the simulation we can see that the global outputs [$\mathbf{X_o}, \mathbf{Y_o}$] is not affected by the zeros come from CORDIC_2.

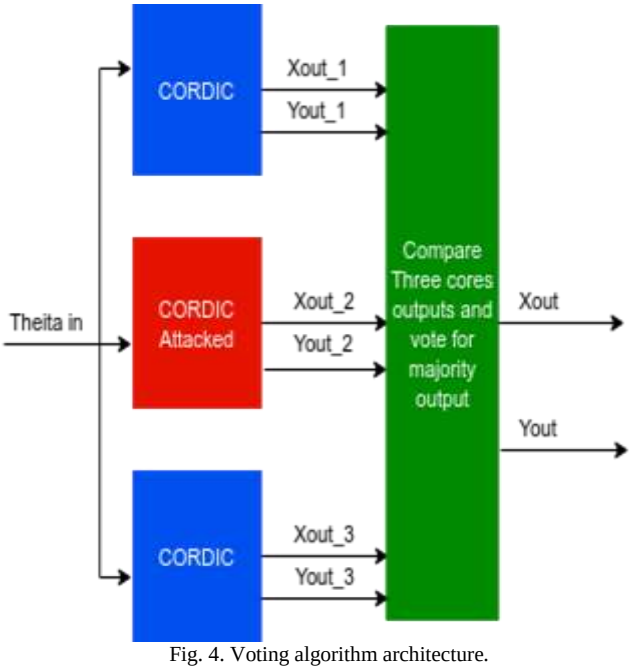

Fig. 4. Voting algorithm architecture.

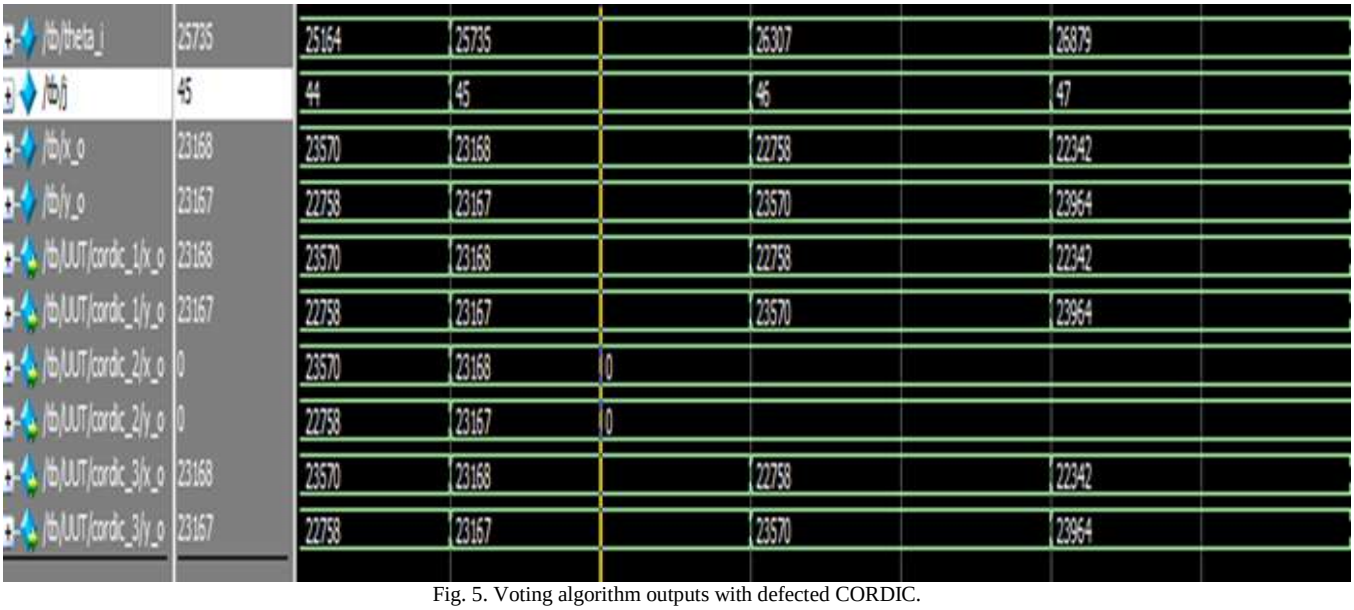

Fig. 5. Voting algorithm outputs with defected CORDIC.

B. *Lockup Table Technique*

In this technique we will use lockup table which contains some of the expected outputs from CORDIC [$\mathrm{X_o}, \mathrm{Y_o}$]. These outputs are limited to predefined input angles, so the lockup table contains few samples of the outputs due to the size limitations. As shown in figure 6 the controller will compare the outputs from First CORDIC core and the Second CORDIC core according to the values in the lockup table. This comparison will happen if the input angle exists in the lockup table. If the outputs are matched then it will connect one of the CORDIC cores outputs to the global outputs. If there is difference between the expected values in the lockup table and the outputs values from any of the CORDIC cores then it will ignore all the outputs from the infected core and connect the global outputs to the core which its outputs match the lockup table values.

The simulation results are represented in figure 7. In the simulation we can see that the global outputs [$\mathbf{X_o}, \mathbf{Y_o}$] is affected by the zeros come from CORDIC_1 core for a while. Then when there is input angle exist in the lockup table, the lockup table outputs did not match the CORDIC_1 core outputs, and match CORDIC_2 core outputs. The controller will change mux_sel to choose CORDIC_2 core outputs to be connected to the global outputs.

Mohamed. M. Elgazzar and Mahmoud Maher El-Sayed Mohammed, "Design and Implementation New Techniques to Prevent Trojan Thread in Parallel CORDIC," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 1, pp. 21-25, 2018.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

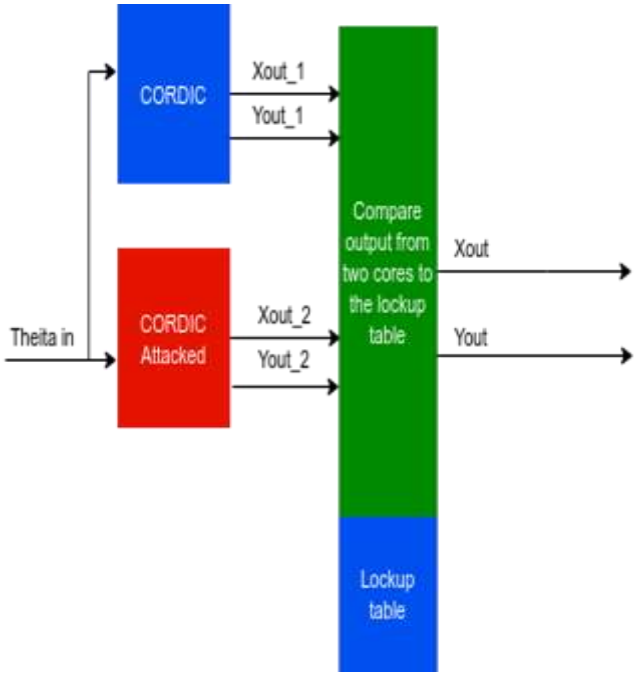

Fig. 6. Lockup table architecture.

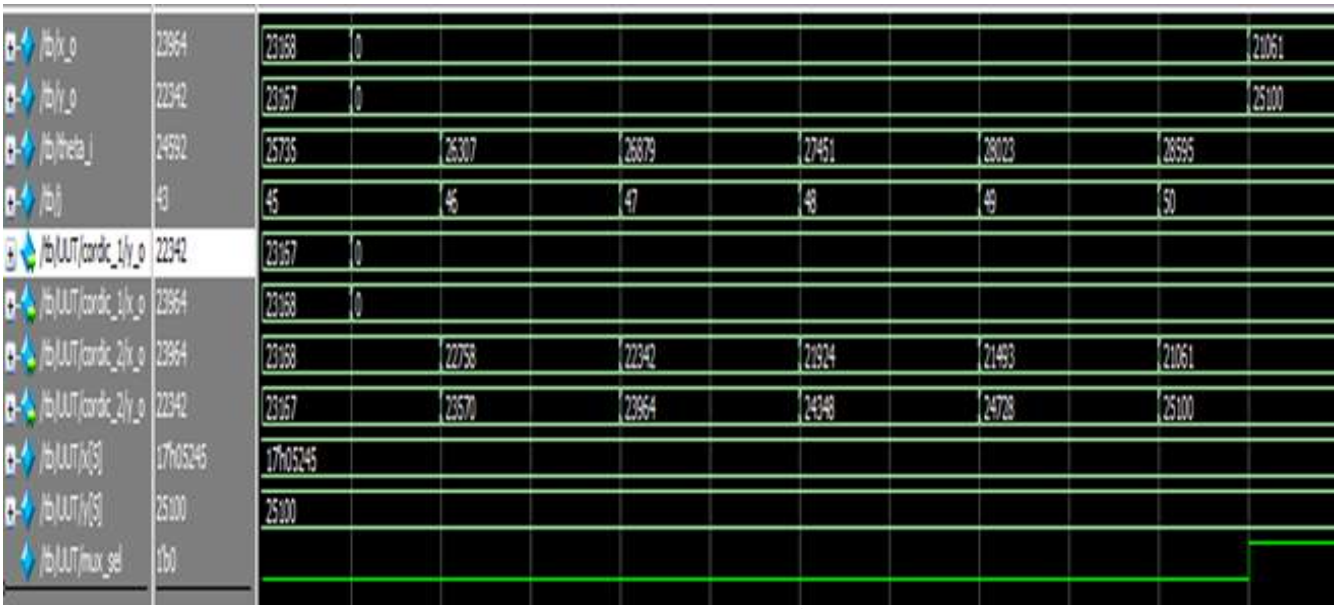

Fig. 7. Lockup table algorithm outputs with defected CORDIC.

V. Results

Simulation of the CORDIC with hardware attack can show us that the CORDIC is infected by Trojan thread and all outputs become zero at range 45° to 60°. The voting algorithm is fast in detecting the hardware attack and can be used to be sure that if one of the three CORDIC cores in the IC is infected, there will not be any effect on the IC performance, but it will take large area and power as three cores will work in parallel to choose the correct outputs. For the Lockup table we will use two CORDIC cores and small lockup table which contains some expected outputs from this core. This technique detects hardware attack most likely slower than voting algorithm but it is less area than voting algorithm due to it depends on two CORDIC cores.

VI. Conclusion

In this paper represent the Trojan thread attack that inserted inside CORDIC core as an example. We saw the effect of this Trojan thread attack and how it will affect the outputs of the CORDIC core and make it zeros with random behavior. Voting Algorithm is used in the design to make the IC immune against the Trojan thread attack. We used three CORDIC cores in this Algorithm. If one CORDIC is infected, the controller will choose one of the two CORDIC cores outputs and connect it to the global outputs. Lockup table technique help also in solving

Mohamed. M. Elgazzar and Mahmoud Maher El-Sayed Mohammed, "Design and Implementation New Techniques to Prevent Trojan Thread in Parallel CORDIC," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 1, pp. 21-25, 2018.

Trojan thread attack. It compares the outputs from two CORDIC cores to the outputs of the lockup table. If one of the CORDIC cores is infected the controller will detect it and move selection to the other CORDIC core which its outputs match lockup table outputs.

REFERENCES

[1] Dave Evans, "The internet of things how the next evolution of the internet is changing everything," Cisco White Paper, April 2011.
[2] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware Trojans," *Proceedings of the 15th Internatinal Conference on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 197-214, 2013.
[3] Beaumont, Mark, Bradley Hopkins, and Tristan Newby. HardwareTrojans-prevention, detection, countermeasures (a literature review). No. DSTO-TN-1012. Defence Science and Technology Organisation Edinburgh (Australia) Command Control Communications and Intelligence Div., 2011.
[4] J. Dofe, J. Frey, and Q. Yu, "Hardware security assurance in emerging iot applications," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, pp. 2050–2053, May 2016.
[5] L. Fiorin, G. Palermo, S. Lukovic, V. Catalano, and C. Silvano, "Secure memory accesses on networks-on-chip," *IEEE Transactions on Computers*, vol 57, no. 9, pp. 1216-1229, Sep. 2008.
[6] V. Soumya, Raghavendra Shirodkar, A. Prathiba, V. S. Kanchana Bhaaskaran, "Design and implementation of a generic CORDIC processor and its application as a waveform generator," *Indian Journal of Science and Technology*, vol. 8, issue 19, pp. 2015.
[7] J. E. Volder, "The birth of CORDIC," *Journal of VLSI Signal Processing*, vol. 25, pp. 101-105, 2000.