Hardware Threat Effect on Parallel CORDIC in IoT Devices

Mahmoud Maher El-Sayed Mohammed, M. Elgazzar

Abstract. Internet of Things (IoT) devices starts to spread all over the world. IoT revolution makes the devices smarter and improves the performance of the devices. The devices can now exchange information between each other and distribute data analysis effort between each other or send it to data analysis center. As a prediction from Cisco, the number of IoT devices will be 50 billion IoT device connected together in 2020. This enormous number will make us think about immunity of these IoT devices against the Hardware attacks. We propose in this paper the effect of inserting Hardware Threat in Coordinate Rotation Digital Computer (CORDIC). Methods are presented in this paper to identify Hardware Trojan and its effect on the CORDIC performance.

Keywords: Internet-of-Things (IoT), Denial of Service, Side-Channel Analysis,Hardware Attack, CORDIC.

I. INTRODUCTION

The Internet of things, or IoTis the new trend in the future of the Electronic Integrated Circuits. IoT Become more important nowadays due to the massive need to connect the different devices together inside our homes, streets, mechanical machines, carsetc. This will help us to access these devices remotely. If we need to change setting in any of these IoT devices, it will be easy to do it through our mobiles, tablet, and computer. This system that we built through these IoT devices is interrelated and these devices can exchange information between each other. The IoT devices help us to facilitate our life and make it easier. We can now convert our life from real life depending on humans to computerized life that exchange data and process it to get the right action .IoT makes the vision to the network and internet become wider. In the Past only the network was consisting of Desktops, Mobiles, and Tablets. Now we can see in the network Smart TV, Smart Air conditioning, sensors, and a lot of Devices like these devices. We can now connect from any place to the device that we need and control its operation. This innovation makes the life easier for us to save time and control these devices remotely and get the best result with the lowest time. This can happen by using our mobiles to control the device remotely.

According to Cisco prediction we expect that the number of IoT devices connected to the internet by 2020 will be 50 billion device [6].

Manuscript published on 30 December 2018.
***** Correspondence Author (s)

Mahmoud Maher El-Sayed Mohammed, Department of Electronics and Communications, Cairo University, Giza, Egypt.

M. Elgazzar, Associate Professor, Higher Institute of Computer Sciences and Information Systems, Fifth Settlement, New Cairo, Egypt.

The different versions of the devices and the limitation in the processing power of the processors which used in the IoT devices, and the spreading of these massive number will make our device vulnerable to hardware or software attacks. These attacks can be in the shape of hardware attacks or software attacks. The hardware security attacks are categorized as side-channel analysis attacks (SCA), hardware Trojan (HT). These two Hardware attacks are the most common attacks, and these attacks take the most of attention in the integrated circuit (IC) design [1].

II. SECURITY ATTACKS

2.1 Side-Channel Analysis (SCA)

This hardware attack aims to get information from the IoT device like power consumption, delays, and electro-magnetic radiated from the device while it performs any cryptographic algorithm. After collecting this information, it will start analyzing it to get the cryptographic algorithm key. The SCA can be categorized in two types, the first category is differential power analysis (DPA) [2], simple power analysis (SPA) [2], differential and correlation power analysis (CPA) [3]. If we compare these attack methods, we will get that the CPA attack can recover the key in fewer power traces than the SPA and DPA [1].

2.2 Denial-of-Service

The Denial of service (DOS) attack is cyber-attack. This attack affects the connection between the IoT device and the network. Hackers or any offenders can launch this attack by flooding the IoT device with excess requests. Also, can happen by sending information which will lead to crash in the target device. This will lead to failure on the device and will not perform its intended behavior [5].

2.3 Hardware Trojan(HT)

The Integrated circuit (IC) fabrication consist of different number of steps so it is complex process. Multiple companies nowadays are working together to design the IC and manufacturing it. The IC at the end should do its task only that is designed for it. If it behaves different than design then it is now under the effect of hardware attack and this attack change its function. The change in the function can be in the shape of bypassing internal logic inside the module or changing its output by changing the existing logic. This attack can happen at any company from the companies that collaborate to make this chip. It may happen in one of the companies that supply part of the Register-transfer level [RTL] of the IC by delivering RTL which contain hardware Trojan inside it.

It may also happen in the IC fabrication factory; the factory can insert HT inside the IC by changing the doping or it can make modifications in the digital or analog circuits [4]. This change in the doping or the circuit will change the behavior of the circuit and will not work probably. It may also happen in the packaging level the packaging can add another circuit when they package the IC.

The Trojan is inserted and still idle till event trigger it. This trigger can be the output value from another module, certain sequence on input pin to the IC, or the Trojan can calculate its event internally. These events can be categorized as internal events or external events. There is another type of Trojan which is simple in debugging and catching it. This simple Trojan don't depend on any event to be fired and it is on all the time.

Hardware Trojan can affect the IC performance in different manners other than the functionality like changing the power cycle time if the chip has sleep and wakeup modes, changing the clock frequency that adjusted internally, Make the processor busy by sending false interrupts to it etc. This type of Trojan will affect the IC and degrade its performance and will be difficult in the debugging. This Trojan can be debugged by comparing the RTL code that we write and the behavior of the IC.Leaking the information from our IC can be the intent of some Trojans, this leaking can happen through any serial interface like JTAG, RS232, or any simple interface.

There is tradeoff between the IC performance and its consumed power especially in IoT field. This tradeoff pushes the companies to use power efficient processors. These processors work in low power but its processing power is weak.We integrate some cores to do any long mathematical operations like cryptographic, or trigonometric functions calculations etc. In order to overcome this weakness.

III. COORDINATE ROTATION DIGITAL COMPUTER (CORDIC)

The first steps to design CORDIC was at the Aero Electronics department by Jack E. Volder. This CORDIC design aim to replace the analog Analyst core that was put in B-58 Bomber' navigation computer. This new design makes the operations more accurate and increase the performance. This Research helps to get the CORDIC algorithm which solves the sine and cosine functions [9]. The CORDIC design can be used to make two different operations. These operations aredescribed in the following two points.

3.1 Rotation Mode

CORDIC can be used to calculate the trigonometric functions like sine and cosine. The vector rotation can help us to calculate the trigonometric functions by rotating the vectors. Figure 1 shows the rotation of vectors fromX_{in},Y_{in} to X_{out}, Y_{out}.The next equations present processing occurs on the input vector to CORDIC and the expected output from output vector.

$$X_{out} = X_{in} \cos \varphi - Y_{in} \sin \varphi \qquad (1)$$

$$Y_{out} = Y_{in} \cos \varphi + X_{in} \sin \varphi \qquad (2)$$

We can take $\cos \varphi$ as common factor from the previous two equations, then these equations will be like:

$$X_{out} = \cos \varphi \, (X_{in} - Y_{in} \tan \varphi) \qquad (3)$$

$$Y_{out} = \cos \varphi \, (Y_{in} + X_{in} \tan \varphi) \qquad (4)$$

We want to calculate the new values X_{out}, Y_{out} then will limit $\tan \varphi = 2^{-j}$, Where j= 0,1,2,3, 4...∞.

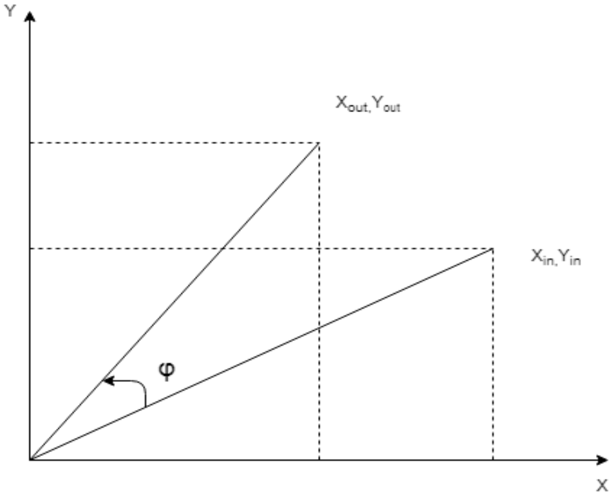


Fig 1. Vector Rotation Between Two Vectors

Using previous equations, if we force the rotation angles to be limited to the list in Table 1, then we can convert $\tan \varphi = 2^{-j}$. The multiplication to the tangent is converted to be simple shift right operation. This will make the Algorithm simpler and reduce the size of hardware, but to get more accuracy we will make the code iterative to deal with different angles.

TABLE 1. Angle Value According To J

J	$\tan \varphi = 2^{-j}$	$\varphi = \tan^{-1} 2^{-1}$
0	1	45
1	0.5	26.56505
2	0.25	14.03624
3	0.125	7.125
4	0.0625	3.57633
5	0.03125	1.7899
6	0.015625	0.89517
7	0.0078125	0.44761
8	0.0039063	0.22381
9	0.0019531	0.1119057

$$X_{out} = C_j \left[X_{in} - Y_{in} S_j 2^{-1} \right] \qquad (5)$$

$$Y_{out} = C_j \left[Y_{in} + X_{in} S_j 2^{-1} \right] \qquad (6)$$

$$C_j = \frac{1}{\sqrt{1+2^{-2j}}} \qquad (7)$$

Where $S_j = \pm 1$ depending on the direction of rotation.

The CORDIC has gain constant which can be calculated from the product ofC_j. This product approaches 0.6073. This happens when the number of CORDIC iterations goes to∞. Therefore, with C_j terms the CORDIC has a gain

$$G = \prod_{j=0}^{n-1} \frac{1}{C_j} = 1.647 \qquad (8)$$

The equations for $X_{out}$and Y_{out}become

$$X_{out} = \cos(Z_{in}) \qquad (9)$$

$$Y_{out} = \sin(Z_{in}) \qquad (10)$$

$$Z_{j+1} = Z_j - S_j \tan^{-1}(2^{-j}) \qquad (11)$$

These equations to be valid it should be in this range

$$-90° \leq \varphi \leq 90° \quad (12)$$

If we want to increase the range to be$|Z_{in}| \leq 2\pi$ for all rotations, then the angles outside the range should be rotated before it is used.

$$X'_{in} = -C\,Y_{in} \ (13)$$

$$Y'_{in} = C\,X_{in} \ (14)$$

$$Z'_{in} = Z_{in} - C\pi/2 \ (15)$$

Where C = +1 for $Y_{in} > 0$ and -1 for all other values.

This Algorithm can be used to calculate the trigonometric sine and cosine functions [7] [8].

3.2 Vectoring Mode

The vector rotation is used to calculate the unknown angle of the input vector. This angle can be calculated by performing rotation. The vector mode rotates the vector using the defined angles in the algorithm, these rotations aim to reduce the y coordinate to be as closely as possible to zero. The rotation direction in every iteration is determined from the sign of residual y coordinate that we obtained in the previous iteration.

IV. HARDWARE TROJAN EFFECT ON PARALLEL CORDIC PROCESSOR

The processors that usually used in the Integrated Circuit is usually have small processing power. If we need to build security algorithm, then the Processor will not be available to execute the security algorithm. It will be better if we build it in hardware Digital signal processing (DSP) security core. In this paper we will use parallel CORDIC which we use in rotation mode to calculate the trigonometric sine and cosine functions. Then this Core is affected by Trojan attack. We will see the effect of this Trojan on the CORDIC performance.

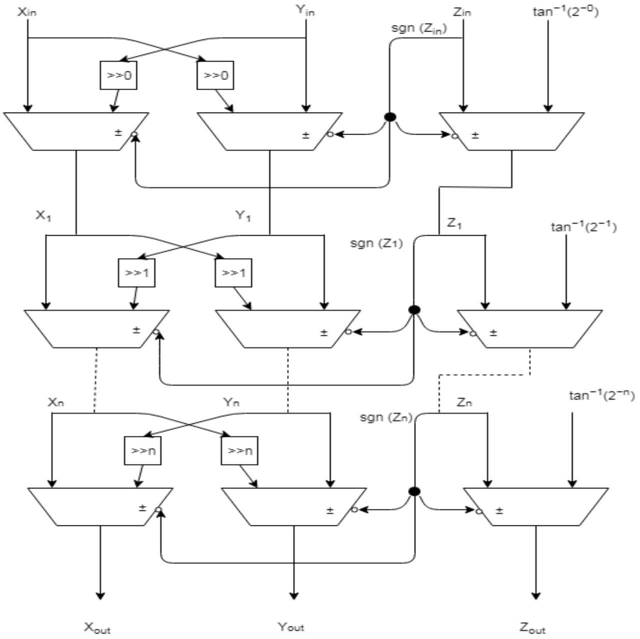


Fig 2. Parallel Architecture of CORDIC

4.1 Parallel CORDIC Architecture

The CORDIC algorithm can be implemented in hardware with different architectures. We will use the combinational or parallel CORDIC implementation. The advantage of the parallel CORDIC is that it has the lowest delay. Figure 2 show the architecture of parallel CORDIC. The parallel CORDIC output will be ready on the next cycle followed the forcing inputs cycle. We can see the critical path for the parallel architecture will be the path from input to output. The CORDIC is designed in Verilog and tested using Modelsim to see the CORDIC performance and simulate the outputs from the CORDIC match the expected outputs. In the next step we will add hardware Trojan to the CORDIC and simulate it and compare simulations versus our reference CORDIC simulations, then this design is synthesized on Xilinx Spartan 3A kit for both the original CORDIC and the CORDIC with hardware Trojan. These simulations will make us see the Trojan effects.

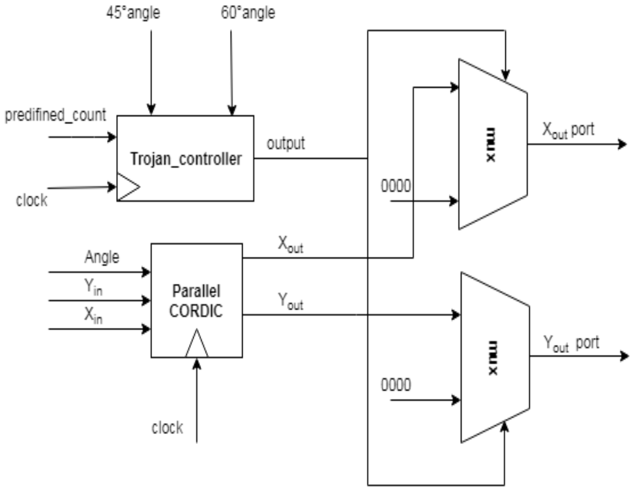


Fig 3. Hardware Trojan Architecture

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

4.2 Hardware Trojan Architecture

The hardware trojan that we design will affect certain range of blocked angles, if the input angle occurs in this range then the Trojan will be active and affect the CORDIC output to make it all zeros. The hardware Trojan controller will consist of sequential circuit such as s simple counter that count the number of times that the input angle is in the blocked range. This counter has predefined number of iterations and it will be fired after the occurrence of blocked angles for number of iterations. The counter firing signal will enable another circuit which will wait until another occurrence of blocked angle. If another angle detected in the blocked range, it will activate the multiplexer to send zeros to the sine and cosine output ports. In the normal case the mux is connected to the actual CORDIC internal signals, but if the Trojan controller is activated then it will send zeros on the CORDIC output ports.

Figure 3 shows the architecture of the hardware Trojan. In this architecture we can get the denial of service (DOS) attack happen when inserting the Trojan. After predefined iterations of detecting angles in blocked range.

If new angle from the blocked range occur on the input then the output will be zeros.

Figure 4 shows the actual output from parallel CORDIC; however, Figure 5 shows the parallel CORDIC while it is infected from the hardware Trojan and outputs go to zeros when angle from blocked angle range occur. In our simulation the angle 45° is the angle that activate Trojan.

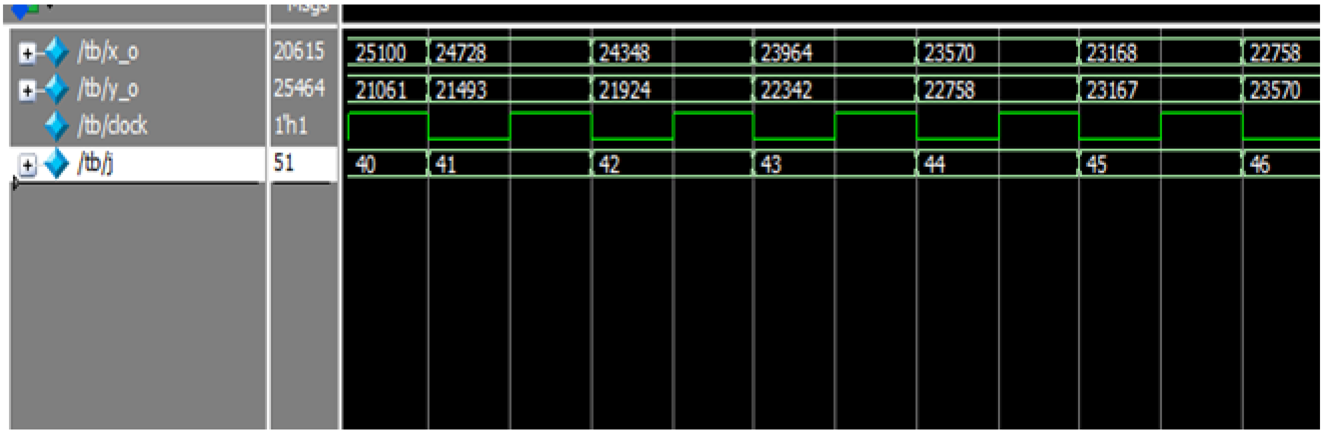


Fig 4. Parallel CORDIC Output Not Affected by Hardware Trojan

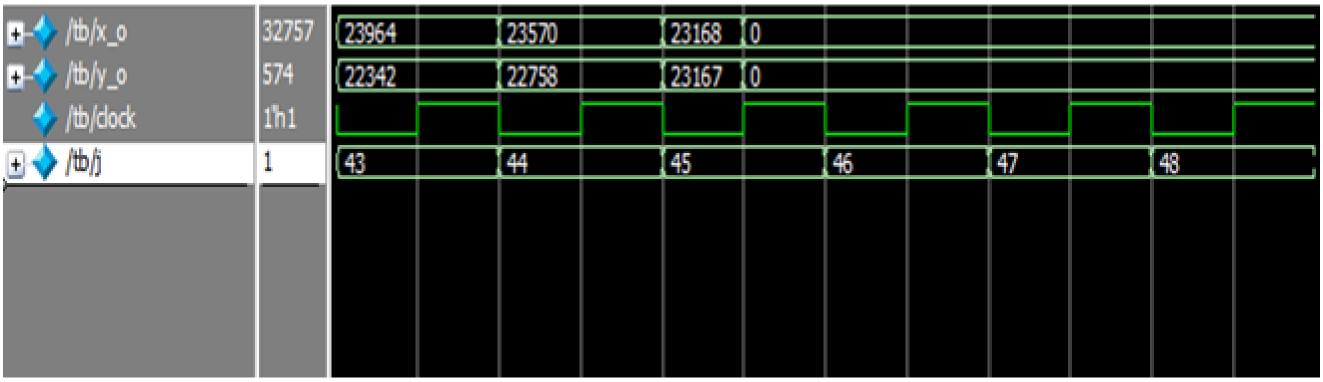
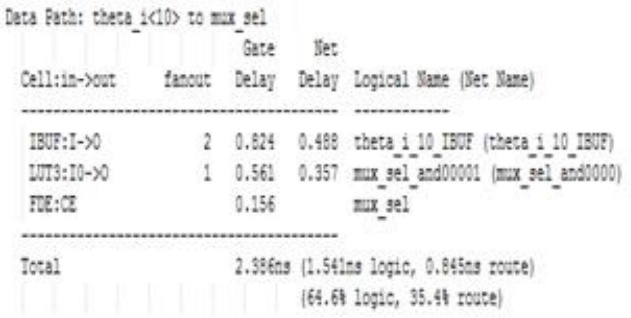


Fig 5. Parallel CORDIC Output Affected by Hardware Trojan

V. DETECTION METHOD

The original parallel CORDIC without any Trojan inserted will be the reference for our comparison. If we compare the RTL schematic of the CORDIC with Trojan attack and original CORDIC, we will find difference between these two schematics. There is another method to detect the appearance of hardware Trojan is to calculate the overall path delay. The additional path delay of 2.386 ns as shown in table 2 is appeared in the CORDIC with hardware Trojan timing report as extra delay while this delay was absent in the original CORDIC. If we compare this method with the previous method then we will find that this method will be faster.

TABLE 2. DATA Path Delay Due to the Trojan Insertion



```
Data Path: theta_i<10> to mux_sel
                              Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   IBUF:I->O             2   0.824   0.488  theta_i_10_IBUF (theta_i_10_IBUF)
   LUT3:I0->O            1   0.561   0.357  mux_sel_and00001 (mux_sel_and0000)
   FDE:CE                    0.156          mux_sel
  ----------------------------------------
  Total                      2.386ns (1.541ns logic, 0.845ns route)
                                     (64.6% logic, 35.4% route)
```

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

VI. RESULTS

When simulating the CORDIC with hardware Trojan,
The hardware Trojan was active if the input angle is in the range of blocked angles, this range is from45° to 60°. This Trojan affects the output of sine and cosine making these outputs zeros. Synthesizing the CORDIC with hardware Trojan shows us the difference in timing between the CORDIC with hardware Trojan and CORDIC without hardware Trojan.

VII. CONCLUSION

In this Paper we tried to present the CORDIC algorithm, and how to implement it using the parallel architecture. We used the parallel CORDIC as a victim of hardware Trojan. We insert hardware Trojan inside it and show how to detect hardware Trojan. The Detection methods are illustrated here. The first method is simulating the CORDIC with hardware Trojan and compare the simulation against the CORDIC without hardware Trojan simulation. The second method is synthesizing both CORDIC modules and see how we can detect the effect of hardware Trojan from the timing report.

REFERENCES

1. J. Dofe, J. Frey, and Q. Yu, “Hardware security assurance in emerging iot applications,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2016, pp. 2050–2053.
2. P. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis,” in Proc. Crypto’99, pp.388-397, 1999. .
3. E. Brier, C. Clavier, and F. Olivier, “Correlation power analysis with a leakage model,” in Proc. Lecture Notes in Computer Science, vol. 3156, pp. 16–29. Springer, Berlin, 2004.
4. 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) 2013, pp. 197-214.
5. Rajendran, J., Gavas, E., Jimenez, J., Padman, V. & Karri, R. (2010) Towards a comprehensive and systematic classification of hardware Trojans, in Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pp. 1871 –1874.
6. Dave Evans, “The Internet of Things How the Next Evolution of the Internet Is Changing Everything,” Cisco White Paper, April 2011.
7. Soumya, V., et al. "Design and Implementation of a Generic CORDIC
8. Processor and its Application as a Waveform Generator." Indian Journal of Science and Technology 8.19 (2015).
9. 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.
10. VOLDER, J. E. (2000). The Birth of CORDIC. Journal of VLSI Signal Processing , 101-105.