Review on Logic Encryption Technique for Secure Hardware Design

Deeksha Sharma[1] and Shweta Agrawal[2]

[1]Research scholar,[2]Assistant Professor,

Department of Electronics and Comm., SRCEM Banmore, Morena, India

Abstract—**Secure hardware is the prime requirement for the modern applications such as medical, finance and defense. The economic constraint and time criticality results in use of untrusted intellectual property (IP) and foundry which may cause IP theft, IC overproduce counterfeit and or insert malicious functionality called hardware Trojan. The logic encryption technique hides the functionality using additional circuitry such that the design functions correctly with valid key only. Several encryption techniques are proposed in the literature. This work reviews existing insertion based techniques and then presents comparative analysis. Designs are implemented on Tanner and simulated with 45nm technology file. Simulation results provide comparative analysis of different design metrics.**

Keywords—**Hardware security, Energy Efficiency, Integrated Circuits, VLSI.**

I. Introduction

Due to the used of untrusted foundry and intellectual property (IP), the integrated circuits (ICs) are becoming more vulnerable to various kinds of attacks. Since these IC are used in several critical applications such as health, banking, and military, secure data processing is required. However, an application may lose significant information due the use of untrusted IC which utilizes untrusted third-party within the IC design flow. An untrusted foundry has access to the design which may cause IP theft, IC overproduce, counterfeit and or insert malicious functionality called hardware Trojan (HT) [1]. Therefore, the security of the IC has been a prime concern due to the emergence of recent attacks called HT. In the HT, malicious editions are done during design or fabrications to corrupt the IC. The HT may cause the design malfunction or steal useful information to defame the designer reputation. Therefore, secure IC design requires costly trusted foundry and IPs which may fail to meet the economic or timing constraints.

Quantum of work has been given in the literature to achieve secure IC [2], [3]. The techniques which prevent HT and also make them easy to detect if inserted are known as design for trust techniques. These techniques include design modification such that HT can be detected easily. The techniques are based on hiding the original functionality by inserting additional logic and the IC works correctly when correct key is applied. The IC provides wrong functionality when the invalid key is applied. The key is stored in the temper proof registers. Since the trusted costumer gets the access to the key, it can access the correct functionality. Therefore, to achieve reliable IC design, logic insertion based techniques have emerged as new paradigm.

In the literature, several approaches are presented that insert additional logic to hide the functionality as shown in Fig. 1. A random XOR insertion based technique [4] is presented in, where additional XOR/XNOR gates are inserted at the output of rare triggered nets. It introduces significant area overhead. Apart from large area overhead, this technique exhibits fixed key which is easy to detect. Similar to XOR based insertion, a multiplexor based, look up table (LUT) based logic encryption techniques are also proposed in the literature. The multiplexor based design introduces an 2x1 mux to hide the functionality whereas the LUT based technique replaces the current gate with the LUT. Although, the encryption using LUT provide various functionality thus exhibits higher security over the other existing, it exhibits very large area overhead.

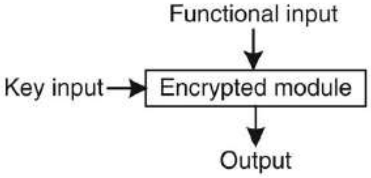


Fig. 1: System with logic encryption.

This paper presents the limitations of existing logic encryption techniques and provides a comparative analysis on the high level and then from simulation results basis. The rest of the paper is organized as follows. Section II provides an extensive literature review on various encryption techniques. The simulation results analysis of the proposed encryption over existing is discussed in Section III. Finally, Section IV concludes the paper.

II. Logic encryption techniques

This section presents review on different techniques to achieve efficient encryption.

2.1 EPIC: Random insertion based technique [4]

In this paper random insertion based logic encryption is presented. The author either insert XOR or XNOR gate to hide the functionality as shown in Fig. 2.

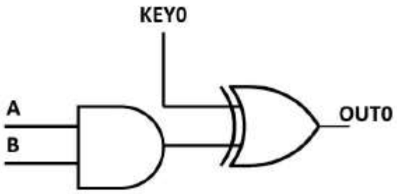


Fig. 2: XOR based logic encryption of AND gate

This technique stops the piracy of the IC automatically by hiding the functionality. The externally added XOR gate exhibits key input which decides whether the circuit will provide correct or incorrect output. The technique is easy to implement and provide incorrect output when invalid key is applied as shown in Fig. 3.

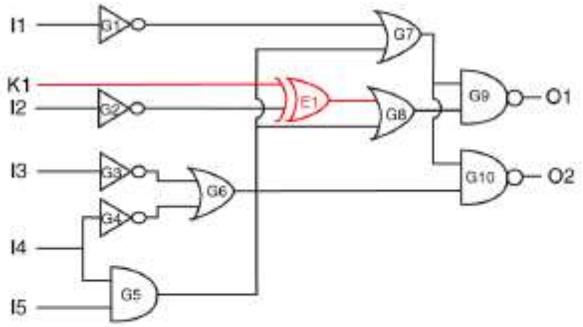
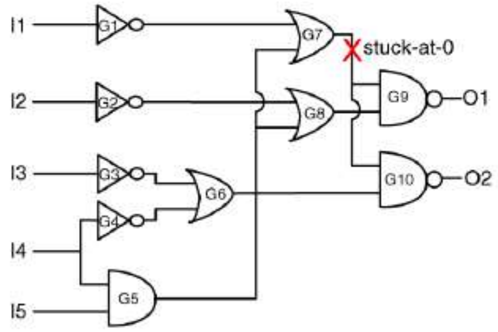


Fig. 3: Circuit with random inserted XOR gate.

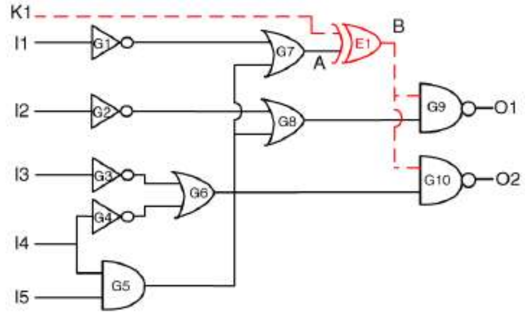
It can be observed from the Fig. 3 that the circuit works different with different key. The valid key (K1 = 0) provides the correct functionality while the non-valid key provides the wrong output.

2.2 Logic encryption using fault analysis [5]

This paper presents an encryption technique which exploit fault analysis to improve the security of the encryption design. This method inserts the gate (XOR/XNOR) at point which affect the output which is similar to the fault that corrupt the output. A fault analysis based logic encryption which provides better compression over the random generation based logic encryption. It is shown that logic encryption using random insertion does not guarantee to provide wrong output for the invalid key. The showed that this scenario is similar to the IC testing where the fault is not propagating to the output. In order to ensure the wrong key produces the wrong output, two conditions should be met which are 1) applications of the wrong key should generate a fault and 2) the fault must be propagated to the output. In order to meet the first condition, a trigger logic must be inserted into the design such that activation of this produces a fault, whereas to propagate the fault to the output, fault must be generated at the point which gets sensitize to output.



(a) A faulty circuit



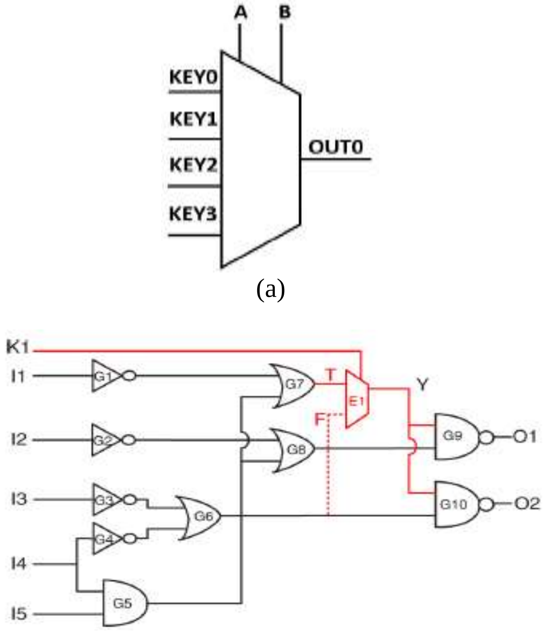
(b) An encrypted circuit

Fig.4: XOR insertion at faulty point.

The analogy between the fault analysis and the logic encryption can be understood with the example shown in Fig. 4. Let us consider a stuck-at-0 fault inserted at the output of gate G7. This fault will propagate to the output of both gates when activated. This fault propagate is similar to the circuit shown in Fig.4(b) where applications of wrong key will invert the input and will produce inverted logic at the output. Even though the activation of the fault, for some input, the fault may not propagate to the output. For example, application of 01110 at the input of Fig. 4(a) will activate the fault but will not propagate i.e. it will provide 00 as output which is similar to the functional output. It is nothing but fault masking. It is also possible that the fault may be inserted at multiple places where activation and propagation of one fault may mask the other.

2.3 Multiplexor based Logic Encryption [6]

In this paper, author presented multiplexor based logic encryption techniques where an addition inverter is added to corrupt the output when invalid key is applied as shown in Fig. 5(a).



(a)

(b)

Fig.5: Mux based logic encryption.

In this technique based on the value of key, either true/correct value of the signal is passed or the wrong value of the signal is pass to the output. Although, the topology as shown in Fig. 5(b) provides wrong output for each invalid input, the prime limitation of this method is difficult to achieve complemented output. In other words, the method demands complemented signal of the true value which is very difficult to achieve. The metric called, transition probability is used to find wire with opposite signal. Further, the multiplexor insertion provides significant overhead to the existing design. Therefore, the method is still not good due to large overhead, complex to achieve inverted signal and poor security due random insertion.

2.4 Logic Cone Analysis Based Encryption [7]

A logic cone analysis based attack strategy is presented by Lee et al. where an attacker can logic cone information to reduce the number of brute force attack to decipher the key and then presented a new technique that can be used to further increase the efforts of an attacker. It is shown that that not all the key gates affect the output i.e. only few key affects an output pin. Therefore, brute force is applied to crack these keys. Once the attacker finds these keys, the attacker uses this information to reduce the number of brute force attempts. For example, consider a circuit shown in Fig. 6. It can be observed from the Fig. 6 that the circuit exhibits six key gates. Therefore, it requires 2^6 i.e. 64 brute force attempt to crack the key. It can also be seen that not all output pin depends on all key i.e. it depends on small number of keys.

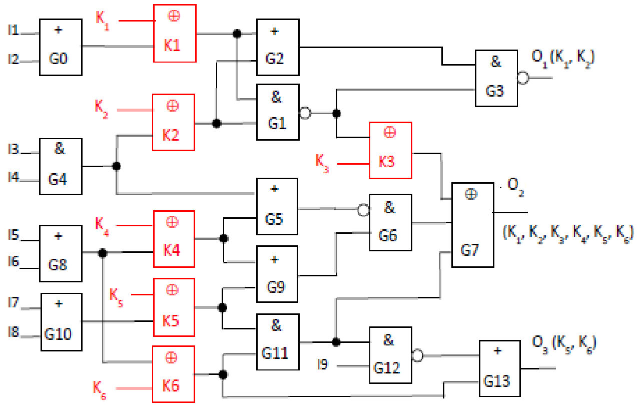


Fig. 6: Logic cones with variant key sensitivity.

For example, the output O_1, O_2 and O_3 depends on two, six and two keys, respectively. Therefore, an attacker can reduce the number of attempts by first applying brute force on the output pin which exhibits less dependence on key and then to the next lower one. For example, for the above circuit, attacker will first decipher the keys K1 and K2 by applying 4 brute force attempts and then extracts keys K5 and K6 by applying 4 more brute force attempts. Now only two key left which can be detected by applying only 4 more attempts which results in overall 12 brute force attempts compared to the 64 attempts required ideally. To improve the security of the design, designer should increase the dependency of each pin to large number of keys. Therefore, to improve the key dependency additional MUXes are inserted as shown in Fig. 7.

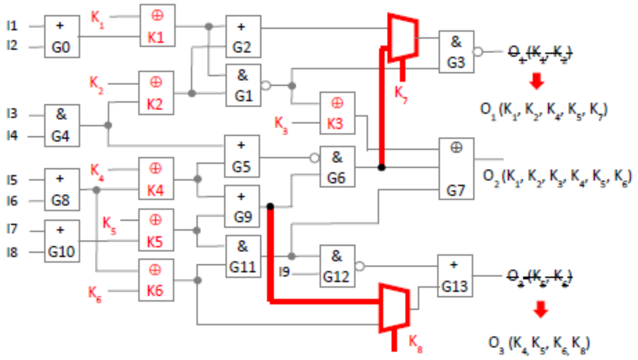


Fig. 7: Mux insertion to increase attacker effort.

The MUX insertion is done in a non-deterministic manner so that even if an attacker knows the key insertion algorithm, the attacker cannot exploit this knowledge to decipher the logic obfuscation.

2.3 Sensitization Attack and Encryption Technique

An attacker can decipher the key in a linear time, by sensitizing the key to the output. The author demonstrated a methodology that can fix this problem by introducing a small logic encryption technique. In the logic encryption, additional gates are inserted in the design with key input. In order to achieve the correct functionality valid key must be inserted [8].

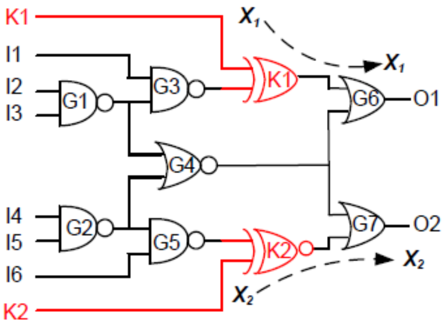


Fig. 8: Logic encrypted circuit using two key gates.

To understand the sensitization of the fault to the output to detect the original key let us consider an example shown in Fig. 8. The application of the 100000 to the input will sensitise the key to the output. Therefore, an attacker can identify the key using this sensitization approach. To avoid this sensitization based key extraction, logic encryption technique must insert the gate such that sensitization depends on the multiple keys. Therefore, smart logic encryption technique cannot propagate the key to output as shown in Fig. 9. If the effect of key can be muted, then the sensitization attack can determine the key. Therefore, to avoid this sensitization attack attacker should introduce non-mutable key gates.

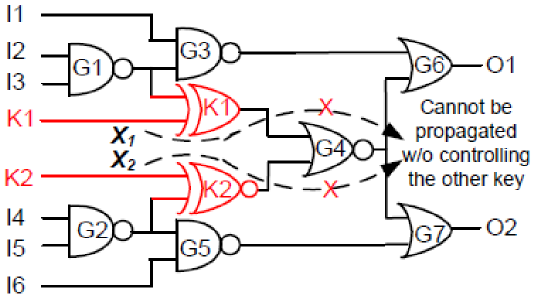


Fig. 9: Smart logic encryption.

If the inserted key gates are non-mutable and the number of key gates is more than 100, then it will take more than several years to extract the key. The author further presented the concept of key dominance as shown in Fig. 10. In this figure key K2 is dominating over key K1 i.e. key K2 can mute the effect of key K1 but opposite is not true [9].

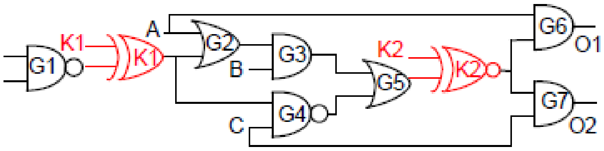


Fig. 10: Key dominance.

The author implemented the proposed techniques on the various benchmark inputs and showed that proposed techniques improves the security without much area overhead.

2.4 Logic gate replacement based encryption

A replacement based logic encryption where they present new reconfigurable designs [10]. Since the insertion based logic encryption provides large overhead, the author showed that replacement based technique provides better security with very small overhead. In this paper, stacked based topology and transmission gate based topology to design reconfigurable logic. For example, a circuit shown in Fig. 11 below can work as either NAND or NOR gate based on key value. It can be seen from the figure that when key value is logic ‘0’, it provides the functionality of the NAND gate whereas for key equal to logic ‘1’ it works as NOR gate. The advantage of the circuit is its lower implementation complexity which is achieved due to the shared function. Further, the based on the value of key either upper PMOS stack is gated off or the lower NMOS stack is gated off which effectively reduces the capacitance. Thus, this provides improved performance over the existing designs. Although the design exhibits small overhead, it is only applicable to the applications where invalid key does not necessarily produce wrong output.

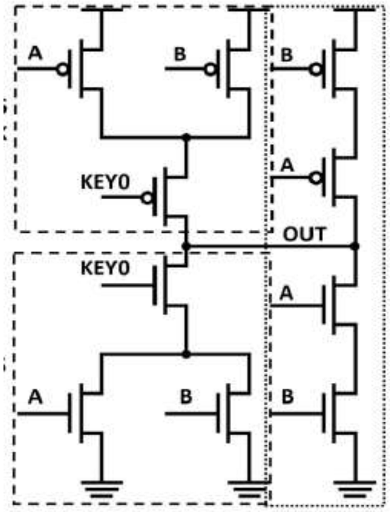


Fig. 11: Reconfigurable circuit for NAND/NOR.

In addition to the above mentioned gate stacked topology, the paper presented a transmission based topology [8] as shown in Fig. 12.

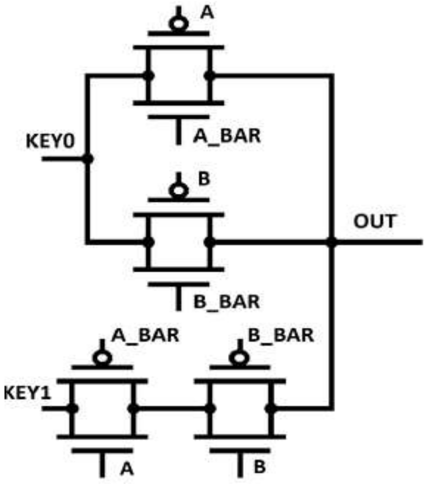


Fig. 12: Transmission gate based encryption.

Based on the value of the key, the circuit can either work as AND or OR gate, for example, the circuit provides the functionality of the AND for key combination of 01 while OR gate for the key combination of 10. The additional advantage of the transmission based topology is that it provides constant logic 0 and logic 1 for the key combination of 00 and 11 respectively.

III. Simulation results and analysis

In order to compare different existing encryption techniques, all the techniques are implemented on Tanner to compute design metrics. The net-lists is extracted for all the implemented designs which are then simulated with 45nm predictive technology model fileon spice simulator to achieve metrics.

4.1 Simulation results on Tanner

The existing encryption techniques are implemented on Tanner to evaluate the design metrics. The schematic of the NAND gate is shown in Fig 13. Similarly, other gates are also implemented and design metrics are evaluated as shown in Table 1.

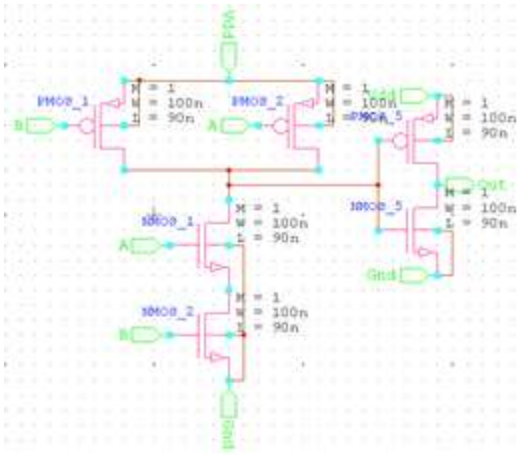


Fig. 13: Schematic of NAND gate on Tanner

Table 1: Design metrics of the basic gates.

Cell	Area (#Tran)	Power (μw)	Delay (ps)
NOT	2	0.072	6.9
AND2	6	0.174	64.37
OR2	6	0.237	57.9
XOR2	12	0.31	31.1
XNOR2	12	0.251	51.1
Mux2_1	12	0.411	67.7
Mux4_1	34	1.11	141.6

Fig. 14 compares the area required by the different gate which further decides the implementation complexity.

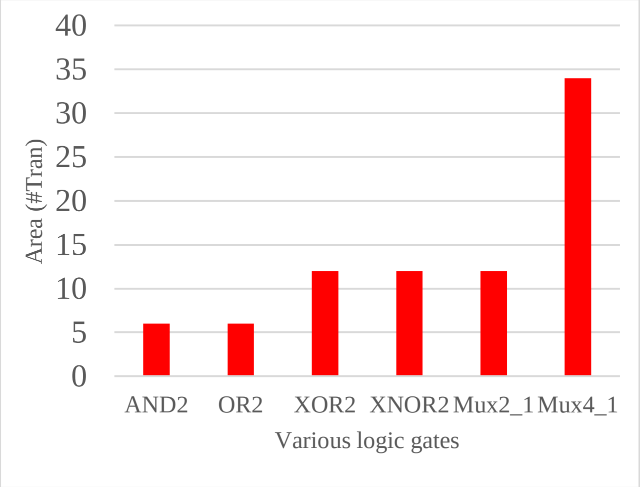
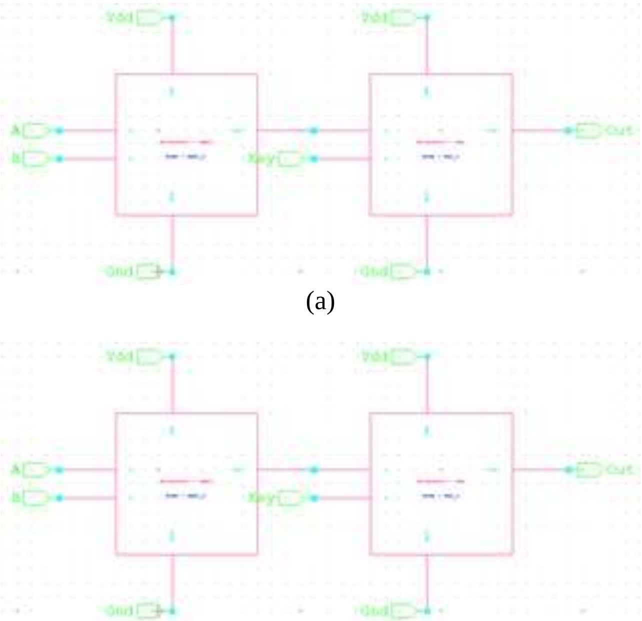


Fig. 14: Area comparison.

Further, an AND gate is encrypted with existing encrypting techniques as shown in Fig. 15.



(a)

(b)

Fig. 15: Schematic of encrypted AND gate with XOR and XNOR.

These designs are then simulated to achieve the design metrics. The area is computed in term of number of transistor while the power and delay are computed in watt and second respectively. Fig. 16 compares the are required by the various encryption techniques.

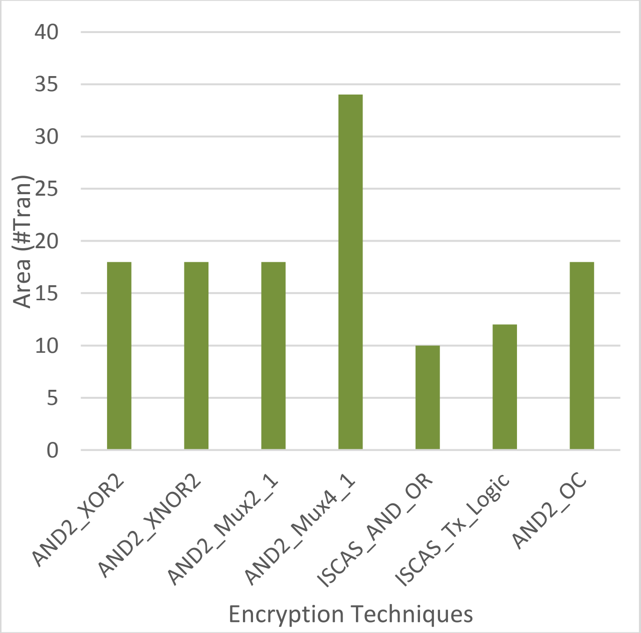


Fig. 16: Area of encrypted AND gate.

The simulation results of the AND gate encrypted with different techniques are shown in Table 2.

Table 2: Metrics of variant encrypted AND gate.

Encryption Technique	Area (#Tran)	Power (µw)	Delay (ps)
AND2_XOR2 [3]	18	0.576	223.6
AND2_XNOR2 [3]	18	0.606	130.7
AND2_Mux2_1	18	0.534	141.6
AND2_Mux4_1	34	1.04	327.5
ISCAS_AND_OR [7]	10	0.302	73.7
ISCAS_Tx_Logic [8]	12	0.0183	28.6
AND2_OC	18	0.515	200

Finally, half adder with different adders is implemented and simulated. Fig. 17 shows a comparison of different encryption on the basis of power consumption. It can be observed from the figure that transmission logic requires small area over the most of the existing techniques. Although, it consumes small power it exhibits poor value of the logic. Finally, the delay of the various encrypted half adders is compared in Fig. 18 where it can be seen that mux 4:1 requires largest power while the stack based, transmission gate technique and the OC cell requires small power consumption. The design metrics are summarized in Table 3.

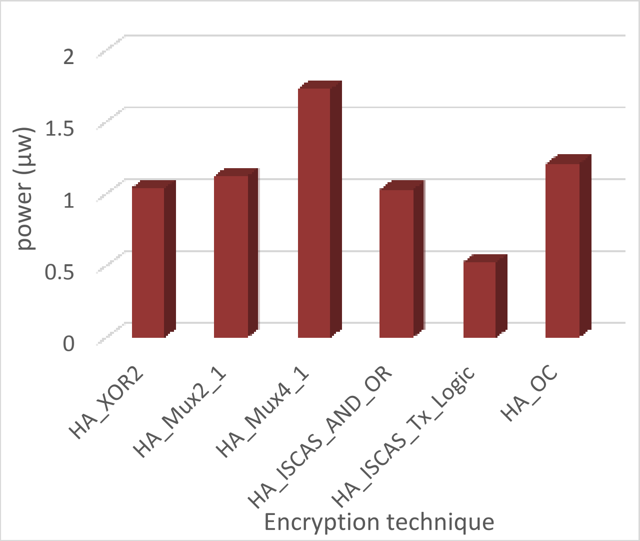


Fig. 17: Power consumption of variant encrypted half adders.

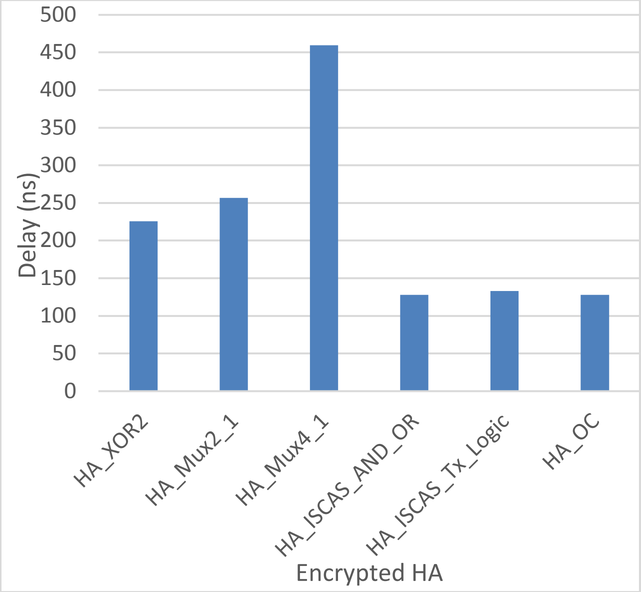


Fig. 18: Delay of variant encrypted half adders.

Table 3: Metrics of variant encrypted half adders.

Encryption Technique	Area (#Tran)	Power (µw)	Delay (ps)
HA_XOR2	32	1.04	225.6
HA_Mux2_1	32	1.124	256.6
HA_Mux4_1	48	1.73	459.3
HA_ISCAS_AND_OR	26	1.03	128
HA_ISCAS_Tx_Logic	26	0.525	133
HA_OC	32	1.21	128

IV. CONCLUSION

Several techniques to encrypt the design have been proposed in the literature to hide the functionality by either inserting the addition logic/circuit or replacing circuit with new reconfigurable logic.Based on key value design provides correct functionality with valid key while provide incorrect functionality with invalid key. This work compare different insertion base logic encryption technique The existing encryption techniques are implemented on Tanner and simulated to compute the design metrics. The simulation results show that different encryption technique provides different improved design metrics.

REFERENCES

[1]. S. Bhunia, M. S. Hsiao, M. Banga and S. Narasimhan, "Hardware Trojan Attacks: Threat Analysis and Countermeasures," in Proc. of the IEEE, vol. 102, no. 8, pp. 1229-1247, Aug. 2014.

[2]. S. Bhunia, M. S. Hsiao, M. Banga and S. Narasimhan, "Hardware Trojan Attacks: Threat Analysis and Countermeasures," in Proc. of the IEEE, vol. 102, no. 8, pp. 1229-1247, Aug. 2014.

[3]. M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," IEEE Design and Test of Computers, Vol. 27, No. 1, pp. 10 – 25, Feb. 2010.

[4]. J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy of integrated circuits," in *Proc. Design, Auto. Test Europe*, 2008, pp. 1069–1074.

[5]. J. Roy, F. Koushanfar, and I. Markov, ‘‘EPIC: Ending piracy of integrated circuits,’’ IEEE Computer, vol. 43, no. 10, pp. 30–38, Oct. 2010.

[6]. K. Juretus and I. Savidis, "Reduced overhead gate level logic encryption," 2016 International Great Lakes Symposium on VLSI, 2016, pp. 15-20.

[7]. Y. W. Lee and N. A. Touba, "Improving logic obfuscation via logic cone analysis," 2015 16th Latin-American Test Symposium (LATS), Puerto Vallarta, 2015, pp. 1-6.

[8]. J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, “Logic Encryption: A Fault Analysis Perspective,” Proceedings of the IEEE/ACM Design, Auto. and Test in Europe, pp. 953 – 958, Oct. 2012.

[9]. Rajendran, J. and Zhang, H. and Zhang, C. and Rose, G. and Pino, Y. and Sinanoglu, O. and Karri, R., “Fault Analysis-Based Logic Encryption,” *IEEE Transaction on Computers*, Vol. 64, No. 2, pp. 410 – 424, Feb. 2015.

[10]. J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, “Security analysis of logic obfuscation,” in *Proc. 49th IEEE Design Automation Conference*, Jun. 2012, pp. 83–89.