Low Overhead Gate Level Logic Encryption

Kyle Juretus and Ioannis Savidis
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, Pennsylvania 19104
kjj39@drexel.edu, isavidis@coe.drexel.edu

Abstract—*Integrated circuits (ICs) used in both commercial and government applications require the utmost reliability and security. As IC design firms continue to shift towards third-party foundries, serious concerns regarding the safety and reliability of ICs have emerged. Concerns include IC counterfeiting, intellectual property (IP) theft, IC overproduction, and the insertion of hardware Trojans. An area of research aimed at ensuring secure and reliable ICs for critical applications is logic encryption, which requires a key to enable the correct functionality of an IC. While the security of an IC is increased when using logic encryption, current implementations, such as the XOR or the look-up table (LUT) methods, have high per gate overheads. A reduction in the per-gate overhead is required to implement logic encryption in a wide variety of applications. Novel gate level designs for logic encryption are described in this paper, resulting in reduced overhead in power, area, and performance as compared to the XOR or LUT based techniques. With the proposed gate level technique, encrypting an AND gate results in a power reduction of 43.2%, an estimated area reduction of 19.8%, and a performance increase of 46.9% in comparison to the XOR based implementation.*

I. Introduction

Applications that utilize integrated circuits (ICs), whether commercial or government in nature, require the IC to function as intended. An emerging threat is the increased reliance on third parties throughout the IC design flow, which introduces security and reliability concerns. A specific threat that is gaining significant traction is the increasing reliance on third-party foundries, as in-house fabrication facilities in advanced technologies cost in excess of $5 billion US dollars [1]. The third party foundry has access to the IC design, which further increases the threat as the foundry has the required tools and knowledge to reverse engineer a design from the GDS-II file alone [2].

A variety of threats including intellectual property (IP) theft, IC counterfeiting and overproduction, and the insertion of malicious circuitry (hardware Trojans) are possible when IC design information is provided to an untrusted third-party foundry. The monetary costs alone are reason for concern with counterfeiting and piracy expected to cause losses of $1.7 trillion dollars in 2015 [3]. In addition, a 2008 analysis conducted by SEMI estimated an IP revenue loss of $4 billion dollars to the IC industry alone [4]. Overshadowing the monetary concerns is the risk of utilizing ICs that do not function as intended, leading to increased failure rates, production of logical errors, and/or the inclusion of malicious hardware Trojans embedded within the ICs. Hardware Trojans utilize many different attack vectors as well, that aim to deny service, steal information, and/or cause incorrect functionality [5]. The wide variety of possible threats presents a challenge to IC security.

One area of research that has emerged to reduce the wide variety of security risks including IP theft, IC counterfeiting and overproduction, and hardware Trojan insertion is logic encryption[1]. Logic encryption increases security by the insertion of additional circuitry (key gates) to a design to hide the functionality of the IC from an adversary. The IC functions incorrectly until the correct input key combination is applied. Requiring a key reduces the information a third party has access to, and results in an increased challenge for the malicious third party foundry to reverse engineer the entire circuit. Without the ability to reverse engineer the design, the malicious third party foundry now has a more challenging task to steal the IP, counterfeit the IC, produce extra ICs, and even insert hardware Trojans as the adversary is not certain what effects the correct IC functionality has on the Trojan.

The current logic encryption methodologies predominantly use XOR gates or replace the original gate with a look-up table (LUT) utilizing a 4x1 MUX. While XOR or LUT based implementations offer a means to increase security, the per-gate area, power, and performance encryption overhead is high. A reduction in the per-gate encryption overhead is therefore necessary to permit the use of logic encryption in a large portion of IC applications. Two novel gate level implementations of logic encryption with reduced circuit overhead are presented in this paper that allow for increased use of IC encryption.

The structure of the paper begins with an introduction to combinational logic encryption in Section II. A per-gate overhead analysis of current logic encryption techniques is provided in Section III, followed by an explanation of the proposed gate topologies in Section IV. A comparison between the proposed topologies and current techniques is described in Section V. Finally, conclusions are offered in Section VI.

II. Combinational Logic Encryption

Combinational logic encryption alters the structure of the circuit to require a key that enables the correct operation of the IC. The two prominent logic encryption techniques are: 1) the utilization of XOR/XNOR gates [6], [8], [9], and 2) the insertion of a LUT as a gate replacement [10]. In addition to

[1] Researchers have previously referred to logic encryption as logic obfuscation [6], but [7] distinguishes between the two as logic encryption prevents black-box use of the design.

the XOR and LUT methodologies, a technique that utilizes 2x1 MUXes was also introduced in [9]. The technique connects two nets to the inputs of the 2x1 MUX, one being the correct input, and the other, a net that has a high probability of carrying the negated value of the correct input. The select signal of the 2x1 MUX is then utilized as the key input to choose between the two nets. The limitation of the approach proposed in [9] is finding a net that consistently negates the desired output, which is challenging in practice. Therefore, an overview of the XOR and LUT based logic encryption techniques is provided in this section.

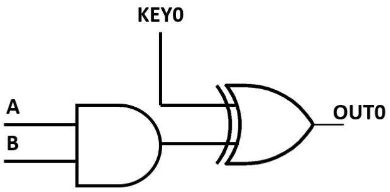

Fig. 1: Logic encryption with the use of an XOR gate.

A. XOR Based Encryption

A simple example of XOR based logic encryption is shown in Fig. 1, where an XOR gate is added after an AND gate. The functionality of XOR encryption is based on the applied key value, where a 1 on KEY0 results in the XOR gate behaving as an inverter, and a 0 results in the XOR behaving as a buffer. Therefore, when KEY0 is 1, an incorrect value is obtained on OUT0. In addition, it is important that an adversary is not able to discern the correct key value of 0. In order to increase the difficulty for adversaries to determine the key based on the implementation of the key gate (i.e. an XOR/XNOR gate), inverters are added in select locations on the data path [6]. The insertion of additional inverters increases the security of XOR based logic encryption, but comes at the cost of larger circuit overhead.

TABLE I: Analysis of the propagation delay, power, and area of standard cells from a 180 nm IBM technology.

Standard Cell	Prop. Delay (ps)	Power (nW)	Area (μm^2)
AND	69.79	70.40	30.73
NAND	36.71	72.67	23.25
OR	92.9	64.52	30.73
NOR	42.09	96.85	23.25
XOR	91.23	148.0	45.69
XNOR	106.7	204.6	41.62

B. LUT Based Encryption

The encryption of a standard gate is achieved through the insertion of a look-up table in place of the original gate [10]. The structure of a 4x1 MUX that encrypts the functionality of a single gate is shown in Fig. 2. The gate encryption is completed by passing the key values to the inputs of a 4x1 MUX, resulting in the key inputs implementing the gate functionality. Such a structure allows for the realization of any 2-input function as opposed to the simple inversion provided by the XOR gate. The expanded number of potential gate functions increases the number of key combinations an adversary must search at the expense of increased circuit area. In addition, as shown in Fig. 2, memory elements are not included to store the key inputs of the 4x1 MUX. If memory was included, the per-gate area and power overhead to encrypt logic increases significantly.

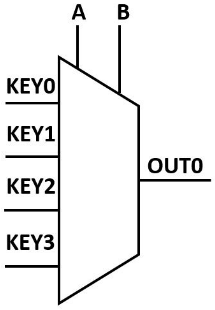

Fig. 2: Logic encryption with the use of a 4x1 MUX.

While the 4x1 MUX increases the area penalty of encrypting a gate, replacing the original gate within the design provides an additional benefit over the XOR based method. Through gate replacement, information of the original gate is removed, whereas the XOR based logic encryption method leaves the original AND gate within the design, as shown in Fig. 1.

TABLE II: Analysis of the propagation delay, power, and area of XOR based logic encryption. Per-gate overheads are provided as percent increases over the standard cell values listed in Table I.

Standard Cell	Prop. Delay (ps)	Power (nW)	Area (μm^2)
AND	151.3 (116.8%)	150.4 (113.6%)	63.73 (107.4%)
NAND	127.6 (247.6%)	142.2 (95.68%)	63.73 (174.1%)
OR	157.5 (69.54%)	174.6 (170.6%)	63.73 (107.4%)
NOR	134.3 (219.1%)	168.5 (73.98%)	63.73 (174.1%)
XOR	181.8 (99.27%)	219.3 (48.18%)	84.50 (84.94%)
XNOR	201.3 (88.66%)	226.3 (10.61%)	84.50 (101.4%)
Average	**140.2%**	**85.45%**	**124.9%**

III. Overhead of Logic Encryption

The XOR and LUT based logic encryption techniques both hinder the ability of an adversary to maliciously use an IC at the cost of increased per-gate overheads. The performance, power, and area penalties associated with encrypting a standard logic gate were analyzed and are provided in this section. Note that no inverters were added after the XOR encryption gates, and no memory elements were used for the LUT based approach for the analysis of the encryption overhead. The values provided for the per-gate overhead of each method are therefore highly optimistic as compared to implementations that include the inverters and memory elements.

The results presented in this paper are based on a 180 nm IBM process. The drive strengths of each implementation were matched to ensure a fair comparison, and all simulations drove a load capacitance of 5 fF. The per-gate overhead in power, performance, and area when implementing the XOR and LUT

TABLE III: Analysis of the propagation delay, power, and area of 4x1 MUX based logic encryption. Per-gate overheads are provided as percent increases over the standard cell values listed in Table I.

Standard Cell	**Prop. Delay** (ps)	**Power** (nW)	**Area** (μm^2)
AND	122.5 (75.53%)	146.0 (107.4%)	90.58 (194.8%)
NAND	124.6 (239.4%)	157.0 (116.0%)	90.58 (289.6%)
OR	120.8 (30.03%)	142.2 (120.4%)	90.58 (194.8%)
NOR	126.8 (201.3%)	158.8 (63.96%)	90.58 (289.6%)
XOR	124.0 (35.92%)	191.8 (29.59%)	90.58 (98.25%)
XNOR	124.6 (16.78%)	191.7 (6.310%)	90.58 (115.9%)
Average	**99.82%**	**73.95%**	**197.2%**

based logic encryption techniques are listed, respectively, in Tables II and III.

The significant per-gate overheads in power, area, and performance observed in Tables II and III prohibit the utilization of logic encryption in a wide variety of applications. The large overhead limits both the total number of gates that are encrypted in a circuit and the locations the encrypted gates are placed to meet timing constraints. It is therefore necessary to reduce the per-gate encryption overhead.

IV. Encrypted Gate Topologies

A method to reduce the per-gate encryption overheads of current logic encryption methodologies is proposed by incorporating logic encryption into the gate design. Reductions in overhead are possible due to inefficiencies in the circuits depicted in Figs. 1 and 2. For example, to encrypt an AND gate, an additional XOR gate is added to the output of the AND, as shown in Fig. 1. The addition of the XOR gate increases the delay, area, and power consumption of the circuit. Similarly, the logic encryption that uses 4x1 MUXes adds a large area overhead and creates additional levels of logic, which decreases performance, as listed in Table III. Embedding security into novel gate level designs reduces the cost of implementing encryption as compared to prior logic encryption methods.

A. Stack-based Topology

The first topology, shown in Fig. 3, is termed *stack-based*, as it relies on the ability of a key input to turn on/off the PMOS/NMOS logic stacks. When KEY0 is 0, the PMOS stack is activated allowing the gate to behave as a NAND. When KEY0 is set to 1, the NMOS stack is activated allowing the gate to behave as a NOR. The ability to essentially disconnect one of the logic stacks depending on the value of the input key reduces power consumption and limits the degradation in performance.

The NAND/NOR stack-based topology has two important characteristics: 1) The ability to share common input-output combinations between implemented logical functions, and 2) implementations that do not require negated inputs. For example, when inputs A and B are both 0 or 1 the NAND and NOR gates produce the same logical output, permitting shared functionality as indicated by the fine dashed box (*shared func.*) shown in Fig. 3. The ability to eliminate the key transistor reduces the area, power, and performance overheads of utilizing the stack-based approach.

Truth Table

KEY0	A	B	OUT
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

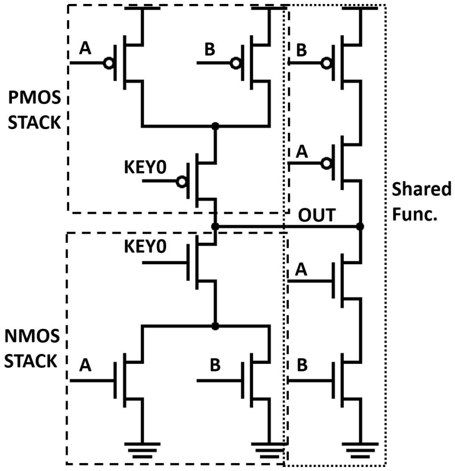

Fig. 3: Stack-based topology implementing a NAND or NOR gate depending on the value of KEY0.

The second characteristic of the NAND/NOR topology, is not requiring negated inputs of A and B, which removes two additional inverters from the circuit and further reduces the overhead of the topology. If a NAND/AND stack-based topology was implemented instead, then the negated inputs are required, as the same input combinations must either turn on a PMOS or NMOS stack depending on the key value.

Truth Table

KEY0	KEY1	A	B	OUT
0	0	X	X	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	X	X	1

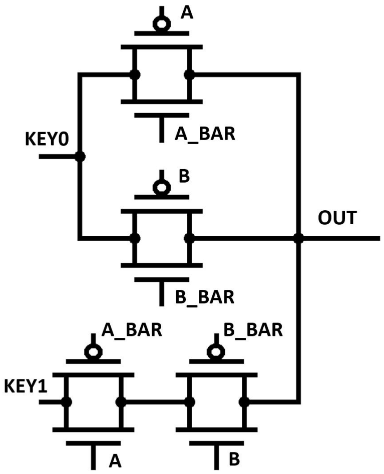

Fig. 4: Transmission gate based topology implementing an AND or NAND depending on the values of KEY0 and KEY1.

B. Transmission Gate Topology

The transmission gate topology passes the key value through transmission gates, eliminating the need to utilize a key transistor. The removal of the key transistor from the encrypted gate increases the performance and reduces the area of the replicated logic. The topology shown in Fig. 4 is therefore better suited to replace the XOR logic encryption method. The functionality of the AND/NAND transmission gate topology is as follows: If logic low is applied to either input A or input B, the value of KEY0 is passed to OUT. Similarly, when inputs A

TABLE IV: Propagation delay, power, and area analysis of the AND/OR implemented with stack-based encryption. Percent improvements over XOR and LUT logic encryption are listed.

Comp.	Standard Cell	Prop. Delay (ps)	Power (nW)	Area (μm^2)
XOR	AND	119.8 (20.82%)	80.72 (46.33%)	41.62 (34.77%)
	OR	116.7 (25.90%)	79.87 (54.26%)	
LUT	AND	119.8 (2.204%)	80.72 (44.71%)	41.62 (54.05%)
	OR	116.7 (3.394%)	79.87 (43.83%)	

and B are both logic high the value of KEY1 is passed to OUT. The truth table shown in Fig. 4 describes the functionality of the encrypted gate for the four different key combinations.

An OR/NOR encryption is achieved by inverting the A and B inputs to the transmission gates. The modification results in KEY0 being passed to OUT when input A or B is logic high, and KEY1 being passed when inputs A and B are both low. The encryption of an XOR or XNOR gate does not offer the same logic minimization possible with an AND, OR, NAND, or NOR gate. The only simplification possible for the XOR or XNOR gates is to eliminate one of the key inputs by tying KEY1 and KEY2 together to form a single key input, as shown in Fig. 2. The number of keys is therefore reduced from four to three when encrypting a single gate, which is important if the IC has high wire congestion.

TABLE V: Propagation delay, power, and area analysis of the NAND/AND implemented with transmission gate encryption. Percent improvements over XOR and LUT logic encryption are listed.

Comp.	Standard Cell	Prop. Delay (ps)	Power (nW)	Area (μm^2)
XOR	AND	80.19 (46.99%)	85.41 (43.21%)	51.10 (19.81%)
	NAND	98.54 (22.77%)	101.0 (28.97%)	
LUT	AND	80.19 (34.54%)	85.41 (41.50%)	51.10 (43.58%)
	NAND	98.54 (20.91%)	101.0 (35.67%)	

V. Evaluation of Power, Area, and Performance

The improvements in power, propagation delay, and area for both the stack and transmission gate topologies are provided in this section. The drive strengths of the encrypted cells were matched with those of the XOR and LUT based logic encryption techniques, similar to the simulations described in Section III. In order to match the drive strength, an inverter was added to the output of both the stack and transmission gate topologies. The functionality of the stack-based design is therefore an AND/OR gate as opposed to a NAND/NOR. The improvement in delay, power, and area over the XOR and LUT based encryption techniques for the AND/OR, AND/NAND, and OR/NOR is listed, respectively, in Tables IV, V, and VI. The improvement in power is based on the average power usage of each gate. The area is estimated from layouts of the proposed, XOR, and LUT encrypted gates. The performance is described as a reduction in propagation delay. The results show significant improvement in performance, power, and area. Aside from the 3% increase in performance as compared to the LUT for an encrypted AND/OR gate, the performance is improved by greater than 20% for all other comparisons. However, the area is reduced the most (54.05%) for the AND/OR as compared to the LUT. In addition, the lowest reduction in power is 28.97% and the smallest reduction in area is 19.8%, both demonstrating substantial improvements with the proposed encryption techniques.

TABLE VI: Propagation delay, power, and area analysis of the NOR/OR implemented with transmission gate encryption. Percent improvements over XOR and LUT logic encryption are listed.

Comp.	Standard Cell	Prop. Delay (ps)	Power (nW)	Area (μm^2)
XOR	OR	86.99 (44.77%)	91.44 (47.63%)	51.10 (19.81%)
	NOR	96.77 (27.94%)	88.00 (47.78%)	
LUT	OR	86.99 (27.99%)	91.44 (35.70%)	51.10 (43.58%)
	NOR	96.77 (23.68%)	88.00 (44.58%)	

VI. Conclusions

A new methodology for logic encryption, called gate level logic encryption, was described in this paper. Two different gate level logic encryption topologies were described, resulting in substantial reductions in the per-gate encryption overhead. Encrypting an AND gate, for example, results in a power reduction of 43.2%, an estimated area reduction of 19.8%, and a performance increase of 46.9% as compared to an XOR based implementation. The stack-based topology also demonstrated similar improvements, indicating that gate level logic encryption is an effective technique to mask circuit functionality while lowering the implementation cost of securing an IC from IP theft and attack.

References

[1] DigiTimes, "Trends in the global IC design service market," http://www.digitimes.com/news/a20120313RS400.html?chid=2, March 2012.

[2] R. Torrance and D. James, "The State-of-the-Art in Semiconductor Reverse Engineering," *Proceedings of the IEEE Design Automation Conference*, pp. 333 – 338, June 2011.

[3] International Chamber of Commerce, "Impacts of counterfeiting and piracy to reach US $1.7 trillion by 2015," http://www.iccwbo.org/News/Articles/2011/Impacts-of-counterfeiting-and-piracy-to-reach-US$1-7-trillion-by-2015/, Feb. 2011.

[4] Semiconductor Equipment and Materials Industry, "Intellectual Property (IP) Challenges and Concerns of the Semiconductor Equipment and Materials Industry," http://www.semi.org/cms/groups/public/documents/web_content/p043701.pdf, April 2008.

[5] 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.

[6] J. Roy, F. Koushanfar, and I. Markov, "EPIC: Ending Piracy of Integrated Circuits," *Proceedings of the IEEE/ACM Design, Automation and Test in Europe*, pp. 1069 – 1074, Oct. 2008.

[7] J. Rajendran, O. Sinanoglu, and R. Karri, "Regaining Trust in VLSI Design: Design-for-Trust Techniques," *Proceedings of the IEEE*, Vol. 102, No. 8, pp. 1266 – 1282, July 2014.

[8] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Logic encryption: A fault analysis perspective," *Design, Automation Test in Europe Conference Exhibition*, pp. 953–958, March 2012.

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

[10] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," *IEEE Design and Test of Computers*, Vol. 27, No. 1, pp. 66 – 75, Feb. 2010.