Hardware Metering

Farinaz Koushanfar
EECS Dept., UC Berkeley, Berkeley, CA 94720
{farinaz@eecs.berkeley.edu}

Gang Qu
ECE Dept., University of Maryland, College Park, MD 20742
{gangqu@eng.umd.edu}

ABSTRACT

We introduce the first hardware metering scheme that enables reliable low overhead proofs for the number of manufactured parts. The key idea is to make each design slightly different. Therefore, if two identical hardware designs or a design that is not reported by the foundry are detected, the design house has proof of misconduct. We start by establishing the connection between the requirements for hardware and synthesis process. Furthermore, we present mathematical analysis of statistical accuracy of the proposed hardware metering scheme. The effectiveness of the metering scheme is demonstrated on a number of designs.

1. INTRODUCTION

Hardware, design and semiconductor companies have been historically vertically integrated. Companies like IBM, Intel and NEC have both leading edge designs as well as superior foundry facilities. However, in the last five years there have been dramatic changes. The most profitable and fastest growing semiconductor business models have been in horizontally focused companies. On one side, pure contract silicon foundries, such as TSMC, UMC, and Chartered Semiconductor conquered almost 1/3 of all semiconductor world-wide output. On the other side, fabless design houses, such as Xilinx, Altera, Broadcom, and Juniper have been by far the fastest growing companies. There is wide consensus that in the future the horizontally focussed companies will significantly increase their market share.

One of major obstacles in this business model is that design companies do not have control over how many copies of their design are made by silicon foundries. Furthermore, FPGA companies get a significant part of their revenues by selling IPs that can readily be used on any of their chips without paying proper royalties.We propose a new intellectual property (IP) usage metering approach that allows IP providers to control proper collection of their IP royalties. The key idea of the hardware metering scheme is to make a very small part of the design programmable at the configuration time and to consequently configure this part for each manufactured chip in a unique way. Different configurations correspond to implementations that are differently scheduled or have different register assignments. Of course, this principle can be applied to other synthesis steps, including the ones during logic synthesis or physical design.

When each manufactured chip has a unique ID, it is relatively straightforward to enforce proper royalty agreements. For example, if a foundry produces n chips that IDs are not reported to the design house in addition to p chips that are reported and approved, the probability that a randomly selected chip from the field has a non-approved ID is equal to n/n+p. Therefore with relatively few tests one can expect a high probability of detecting unauthorized chips.

An obvious, albeit naive, alternative to the proposed metering scheme is to just add a disconnected extra piece of programmable memories that carries the ID mark of a specific manufactured IC. The first advantage of the proposed distributed and integrated within design hardware metering scheme over this straightforward scheme is that it has lower hardware overhead, since it leverages a part of don't-care signals in the finite state machine of the hardware design. The approach provides some level of protection against reverse engineering. For example in hardware, the presence of programmable control path instead of hard-wired logic makes reverse engineering more difficult since essentially all reverse engineering schemes require multiple chips to be dissected [1, 20]. Since now each chip is slightly different but has the same functionality, the reverse engineering process is more difficult. Furthermore, distributed programmable resources in the control part have a number of potential positive side effects. For example, they can be used to facilitate debugging [22] and engineering change [9].

Finally, it is interesting and important to discuss the relationship between the proposed hardware metering scheme with fingerprinting schemes for IP protection [4]. For example, fingerprinting-based metering solution is to give the manufacturer the number of IPs as stated in the licensing agreement, each IP has a unique fingerprint and implements the same functionality [16]. If the manufacturer uses one piece of IP more than once, then they face risk of being caught by the IP provider from detecting multiple copies of the same fingerprint. However, this challenges the mass foundry production line since each IP requires a unique mask and makes tuning of parameters of the foundry line to design much more difficult. Also, fingerprinting will inevitably introduce a significantly large overhead since it aims at placing hidden information in all parts of the hardware design and follows random signature driven constraints.

2. RELATED WORK

To the best of our knowledge this is the first approach for hardware IP metering. Related work can be traced within broad fields of cryptography and computational security, conceptually related fields of intellectual property protection and licensing and objective-related field of software, content, and WWW access metering. Recently, SiidTech Inc., an Oregon start-up company, has proposed an approach for integrated circuit identification from random threshold mismatches in an array of addressable MOSFETs. The technique leverages on process discrepancies unavoidably formed

DAC 2001, June 18-22, 2001, LasVagas, Navada, USA.
Copyright 2001 ACM 1-58113-297-2/01/0006...$5.00.

during fabrication. This analog technique can be used in tracking semiconductor dies, authentication and intellectual property (IP) tagging. In a recent report of this method's measured performance[19], for a 0.35um poly CMOS, for generating 112 ID bits, 132 blocks area used, each with the area of 252x93um. The IDs proposed by SiidTech are not deterministic and these IDs can not be deterministically compacted. Also, due to the birthday paradox, there is still a small probability that two IDs generated randomly have the same value. Component application enables the user to trace a particular die on a wafer and store this information for future usages. There are several advantages of our scheme over the Siid scheme. We have been able to obtain more than 1.3E12 distinct solutions even in our smallest test cases that have only 15 registers (Our number of solutions will go exponentially high by using a few more registers). Furthermore, our IDs are deterministic and therefore they can be used to contain a defined signature to be used in many cryptographic schemes.

Modern cryptography started with introduction of one-way trapdoor function-based public-key cryptographical communication protocols by Diffie and Hellman [10]. A number of excellent cryptographical textbooks are available [21].

One method to enable design IP protection is based on the constraint manipulation. The basic idea is to impose additional author-specific constraints on the original IP specification during its creation and/or synthesis. The copyright detector checks whether a synthesized IP block satisfies the author-specific constraints. The strength of the proof of authorship is proportional to the likelihood that an arbitrary synthesis tool incidentally satisfies all the added constraints [15,5,23]. Similarly, to protect legal users of the IP, fingerprints are added to the IP as extra constraints [4]. Finally, copy detection techniques for VLSI CAD applications have been developed to find and prove improper use of the design IP [6,16]. These techniques are effective for authentication. However, since they make each design unique, it becomes ill-suited for mass production and cannot be applied for hardware metering. In addition, obfuscation can be used for IP protection [7].

Another research, to some extent related to our work, is forensic engineering technique, that has been explored for detection of authentic Java byte-codes [2] and to perform identity or partial copy detection for digital libraries [3]. Also, forensic analysis principles are used in the VLSI CAD to demonstrate that solutions produced by strategically different algorithms can be associated with their corresponding algorithms with high accuracy [18].

3. PRELIMINARIES

Consider the following scenario that requires hardware metering: a start-up design company *A* builds a system that outperforms all the similar products on the market. *A* gives the VHDL description of the system to manufacturer *B* and makes an agreement with *B* to fabricate 10 million copies of the design. The first 2 million copies sold out almost immediately, then the sale slows down even when company *A* lowers the price. It seems the market has already been saturated. Meanwhile, market survey shows that there are about 12 million similar products in use. *A* suspects that foundry *B* has violated the agreement and fabricated more than 10 million copies without reporting to *A*. However, for a given product, *A* cannot provide convincing evidence to tell whether this copy is legally sold or not.

We observe that the problem comes from the fact that *A* sells identical products on the market. If they can give each product a unique identification number, then when two products with the same identification number are found, the existence of unauthorized becomes obvious. Before the discussion of technical details, we first analyze the requirements and objectives. Four basic questions have to be answered:

P1 How to create many distinct copies with the same functionality?

P2 Once two identical copies are found, how can we prove our ownership?

P3 How many tests do we need to conduct before we gain a certain level of confidence that there are no unauthorized copies on the market?

P4 If there are unauthorized copies, how can we estimate the number of copies that they have made?

The existing watermarking techniques provide solutions to problem **P2**: During the design synthesis, we embed our digital watermarks and later on retrieve such watermarks for authorship [15]. **P3** estimates designer's effort to prove foundry's honesty, while **P4** provides valuable on-court information for the designer. We will build statistical models and address them in the next section. To end this section, we discuss the requirements for solutions to the first question:

- Correct functionality: Although we want the system to be distinct, they must have exactly the same functionality.
- Large number of different copies: The method has to be capable of producing huge amount of distinct copies (from tens of thousands to millions) to accommodate to the market.
- Low overhead: The degradation of system's performance due to the large number of different copies has to be kept at the minimal level, if zero-overhead is not achievable.
- Transparent to design and test: The creation of different copies has to be transparent to the manufacturing and testing. Otherwise, it will make the mass production impossible. For this reason, we suggest post-processing, i.e. keep most components of the chip the same and make small changes at the final stage of the fabrication.
- Resilient against attacks: Attempts to making distinct extra copies or duplicated copies without being caught will be difficult, costly, and time-consuming.

4. HARDWARE METERING TECHNIQUES

In this section, we propose and analyze a number of ways for hardware metering. There are several alternatives for implementing the identification logic within the control path logic for hardware protection. Our focus is on control logic, because in modern design it is usually a very small percentage of the design area, often less than 1%. Each of the proposed techniques that have certain advantages and disadvantages.

PROM-based approach: In this approach the required data is stored in a family of PROMs (preferably non-reconfigurable e.g. OTP EPROMs). This data is then read out of the registers sequentially to form a control path. The fast improving memory technology is rapidly reducing on-board programming time and the required extra manufacturing processing steps. The advantages of this approach includes on-board programmability and small area overhead. However, the additional required mask steps and erasure of UV light for programming the PROM, limits attractiveness of this approach.

Disconnection approach: In this approach, an additional finite state machine (FSM) is designed to facilitate design identification.

Checking the ID of the design, requires an unused state of the other FSMs that are part of the design. Modern designs have a large number of FSM with numerous unused states/input combinations (don't cares). The added FSM, is the same for all the designs in the mask level. In the postprocessing step, lasers burn some of the connections of this added FSM in each design and thus generates different states and functions of it. This added FSM is different in each design since we laser burn different connections in each design to achieve a slightly different control path. The algorithms to decide exactly where to burn the interconnect in each chip, can be derived from a computer simulation of the state machine to derive unique ID for each of them. This solution does not need any extra processing steps and is much faster and more robust than the previous approaches.

BISR approach: BISR designs are designs that have built in self repair fault tolerance that can function properly even if some parts of the design are faulty [14]. The idea here is to intentionally induce variety of faults in BISR designs in such a way that each design has different faulty parts. This solution uses the same methodology as the disconnection approach mentioned in the last section. The difference is that the added FSM is now reading out the unique fingerprint proposed by the SiidTech Corporation [19].

SiidTech approach, which identifies each chip by detecting imbalances in threshold voltages-discrepancies unavoidably formed during fabrication. The advantage of this approach is also that no external programming or special processing steps are needed. The disadvantages of this approach are the same as for the generic Siid technology that was elaborated in Section 2.

5. DETECTION

In this section, we will address problems **P3** and **P4** proposed in Section 3. Suppose the design house asks the foundry to fabricate n copies and N Š n is the number that the foundry really makes. **P3** asks the expected number of tests to find a duplicate if $N > n$ or the number of tests to convince designer that $N = n$. **P4** requires an estimation of N once the first unauthorized is found. We take the dishonest foundry's best strategy in that he makes $k - 1$ duplicates for each original copy. It is proven that for a fixed $N = k \cdot n$, the dishonest foundry has the best chance to survive in this equiprobable case.

Theorem 5.1. Draw l from $N = k \cdot n$ objects which consist of k copies of n distinct ones, the probability that there is no duplicate, denoted by *Prob[n,k,l]*, is

$$\left[1-\frac{k-1}{N-1}\right]\cdot\left[1-\frac{2(k-1)}{N-2}\right]\ldots\left[1-\frac{(l-1)(k-1)}{N-(l-1)}\right] \qquad (1)$$

that has an upper bound

$$\left[1-\frac{p}{n}\right]\cdot\left[1-\frac{2\cdot p}{n}\right]\ldots\left[1-\frac{(l-1)\cdot p}{n}\right] \qquad (2)$$

where $p = 1 - 1/k$.

Prob[n,k,l] is the probability that there are no unauthorized parts found after l random tests (without replacement), provided that there are k copies for each of the n originals. It decreases as k increases, since when the population (N) grows, it becomes more difficult to find duplicates; it also decreases as l, the number of tests, increases.

The quantity 1-*Prob[n,k,l]* is the confidence that the designer can achieve from l consecutive successful tests. Success means that no duplicate is found. Table 1 shows some examples for the case n=1000. For instance, after checking 50 products and not finding any duplicates, the designer believes that there does not exist another copy of 1000 chips with a 46.64% confidence. With the same methodology, the probability that the foundry makes 10000 instead of 1000 is less than 33%. The designer's confidence goes up quickly as more tests are conducted. After 100 successful tests, the designer will be 92.62% convinced of the foundry's honesty.

Table 1: Designer's confidence after l consecutive successful tests

l	k=2	k=3	k=4	k=5	k=10
10	2.24%	2.97%	3.33%	3.55%	3.98%
20	9.15%	12.00%	13.38%	14.20%	15.82%
50	46.64%	56.62%	60.87%	63.21%	67.47%
75	76.34%	85.25%	88.33%	89.86%	92.33%
100	92.62%	96.84%	97.73%	98.39%	99.02%

Theorem 5.1 not only gives formula on the designer's confidence about foundry's honesty, it also answers problem **P3**. As we mentioned, 1-*Prob[n,k,l]* measures the foundry's honesty and it increases as l increases. For a designer to gain a desired level of confidence α, we need to find the smallest l such that $1 - Prob([n, k, l] \geq \alpha)$. Unfortunately, there is no exact closed form for formula (1). However, the solution can be always found numerically and there exist good approximation formulas when n is large [12].

We assume that k is equally distributed and derive Theorem 5..2 that answers problem **P4** immediately.

Theorem 5.2. The probability that the first unauthorized is found at the l+1st test is

$$Pr[n, k, l+1] = Prob[n, k, l]\cdot\frac{l\cdot(l+1)\cdot(k-1)}{N-l} \qquad (3)$$

Corollary 5.3. The expected number of tests to find the first unauthorized copy is

$$\sum_{k=1}^{\infty}\sum_{l=1}^{n(k-1)+1} l\cdot Pr[n, k, l] \qquad (4)$$

Corollary 5.4. If the first failure occurs at l, then the expectation for k is

$$\sum_{k=1}^{\infty} k\cdot Pr[n, k, l] \qquad (5)$$

6. DESIGN FLOW

In this section, we address how to create many different copies of the systems that have the same functionality. We illustrate our approach using the graph coloring problem.

The NP-hard graph vertex coloring (GC) optimization seeks to color a given graph with as few colors as possible, such that no two adjacent vertices receive the same color. Given a graph, our objective is to create as many as possible high quality solutions that are relatively close [11,13]. By high quality, we mean that if the optimal solution is known, then all the solutions that we generate will not use any extra color [17]. Therefore, the fingerprinting techniques for GC cannot be used in this case, because they usually introduce overhead although they are very effective in creating new solutions.

The following steps illustrate our algorithm.

1. Apply a graph coloring heuristic to color the given graph $G(V, E)$ and obtain a k-color scheme as the seed solution.
2. For each node $v \in V$, calculate $c(v)$, the number of different colors that v's neighbors get.
3. Sort the nodes V in the increasing order of $c(v)$.
4. For each node $v \in V$ with $c(v) < k - 1$, change v's color and report $k - 1 - c(v)$ different solutions.
5. For all pairs of nodes (u,v) with $c(u) < k - 1$ and $c(v) < k - 1$, try different coloring schemes for nodes u and v and report the new found solutions if any.

In next section, we will demonstrate the performance of this algorithm by experimental results. It turns out that this simple strategy works very well in real-life graphs. Notice that no extra colors will be used in our approach, i.e., all the derived solutions will have the same quality as the seed solution. And these solutions differ from the seed solution only at the colors of one or two nodes.

7. EXPERIMENTAL RESULTS

In this section we analyze the ability of the proposed metering scheme to generate a large pool of designs with unique ID.

Table 2 shows the results of the application of the scheme on generating numerous graph coloring (register assignment). The first column indicates the name of design from the Hyper benchmark suite [8,24]. The second and third column indicate the number of variables and registers in the designs. Two final columns indicate the number of the unique solutions that can be obtained using the following two methods. The first one (column 4) is the assignment of exactly the same subset of variables to different registers in their physical instances. The last column indicates the number of different solutions produced using the technique presented in Section 5. In both cases, even for the smallest design, the number of solutions is very high. The key reason for this situation is that it is well known that the interval graphs for all known designs are very sparse and it is very easy to color them in many different ways using the minimal number of colors.

Table 2: Generated number of distinct solutions for the register assignment-based metering scheme.

Design	Variables	Registers	#of solutions	
8th CD IIR	35	19	1.2E17	1.1E21
Linear GE Ctrl	48	23	2.6E22	5.0E36
Wavelet	31	20	2.4E18	9.4E17
Modem Filter	33	15	1.3E12	5.9E18
2nd Volterra	28	15	1.3E12	9.0E16
D/A Converter	354	171	> 1E200	5E123
Echo Canceler	1082	1061	> 1E200	6E202

8. CONCLUSION

We have developed the first hardware usage metering scheme. The scheme enables design companies to securely control licensing rights for their IP. The scheme utilizes a small percentage of a design implemented using configurable technology to embed a unique ID to each manufactured design instance. We also presented mathematical analysis for detection accuracy of the proposed scheme. We demonstrated the ability of the scheme to implement large number of chips with different IDs. The main result of the paper is that we established generic connection between the scheme and synthesis and compilation tasks.

9. REFERENCES

[1] R. Anderson, M. Kuhn, "Tamper resistance-a cautionary note." USENIX Workshop on Electronic Commerce, pp. 1-11, 1996.

[2] B.S. Baker, U. Manber, "Deducing similarities in Java sources from bytecodes." USENIX Technical Conference, pp.179-90, 1998.

[3] S. Brin, J. Davis, H. Garcia-Molina, "Copy detection mechanisms for digital documents." SIGMOD Record, vol. 24, no. 2, pp. 398-409, 1995.

[4] A.E. Caldwell, H. Choi, A.B. Kahng, S. Mantik, M. Potkonjak, G. Qu, J.L. Wong, "effective iterative techniques for fingerprinting design IP." ACM/IEEE Design Automation Conference, pp. 843-848, 1999.

[5] G.J. Chaitin, "Register allocation and spilling via graph coloring." SIGPLAN '82 Symposium on Compiler Construction, pp. 98-105, 1982.

[6] E. Charbon, I. Torunoglu, "Copyright protection of designs based on multi source IPs." 35th IEEE/ACM International Conference on Computer Aided Design, pp. 591-595, June 1998.

[7] C.S. Collberg, "Reverse interpretation + mutation analysis = automatic retargeting." Proceedings of the ACM SIGPLAN 1997, pp. 15-18, June 1997

[8] R.E.Crochiere, A.V. Oppenheim, "Analysis of linear digital networks." Proceedings of the IEEE, vol.63, no.4, pp. 581-95, April 1975.

[9] S. Dey, V. Gangaram, M. Potkonjak, "A controller redesign technique to enhance testability of controller-data path circuits." IEEE Transaction on Computer-Aided Design, vol.17, no.2, 157-68. February 1998.

[10] W. Diffie, M. Hellman, "New directions in cryptography." IEEE Transactions on Information Theory, vol.IT-22, no.6, pp.644-54, November 1976.

[11] http://www.dimacs.rutgers.edu

[12] P. Flajolet, D. Gardy, L. Thimonier, "Birthday paradox, coupon collectors, caching algorithms and self-organizing search." Discrete Applied Mathematics, vol. 39, no. 3, pp. 207-229, November 1992.

[13] M.R. Garey, D.S. Johnson, "Computers and intractability. A guide to the theory of NP-completeness." Oxford, UK: Freeman, 1979.

[14] L.M. Guerra, M. Potkonjak, J.M. Rabaey, "Behavioral-level synthesis of heterogeneous BISR reconfigurable ASIC's." IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.6, no.1, pp.158-67, March 1998.

[15] A.B. Kahng, J. Lach, W.H. Mangione-Smith, S. Mantik, I.L. Markov, M. Potkonjak, P. Tucker, H. Wang, G. Wolfe, "Watermarking techniques for intellectual property protection." 35th ACM/IEEE Design Automation Conference, pp. 776-781, June 1998.

[16] A.B. Kahng, D. Kirovski, S. Mantik, M. Potkonjak, J.L. Wong, "copy detection for intellectual property protection of VLSI design." 36th IEEE/ACM International Conference on Computer Aided Design, pp. 600-604, November 1999.

[17] D. Kirovski, M. Potkonjak, "Efficient coloring of a large spectrum of graphs." 35th IEEE/ACM Design Automation Conference, pp.427-32, 1998.

[18] D. Kirovski, D. Liu, J.L. Wong, M. Potkonjak, "Forensic engineering techniques for VLSI CAD tools." 37th ACM/IEEE Design Automation Conference, pp.581-6, June 2000.

[19] K. Lofstrom, W.R. Daasch, D. Taylor, "IC identification circuits using device mismatch." Proceedings of the International Solid-State Circuits Conference, pp. 372-3, 2000.

[20] D. P. Maher, "Fault induction attacks, tamper resistance, and hostile reverse engineering in perspective." Financial Cryptography First International Conference, pp. 109-21, 1997.

[21] A. J. Menezes, P.C. van Oorschot, S.A. Vanstone, "Handbook of applied cryptography." Boca Raton, FL:CRC Press, 1997.

[22] M. Potkonjak, S. Dey, K. Wakabayashi, "Design-for-Debugging of application specific designs." International Conference on Computer-Aided Design, pp. 295-301, November 1995.

[23] G. Qu, M. Potkonjak, "Analysis of watermarking techniques for graph coloring problem." IEEE/ACM International Conference on Computer Aided Design, pp. 190-193, November 1998.

[24] J.M. Rabaey, C. Chu, P. Hoang, M. Potkonjak, "Fast prototyping of datapath-intensive architectures." IEEE Design & Test of Computers, vol.8, no.2, pp. 40-51, June 1991.