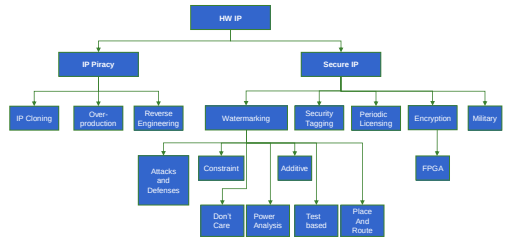
Watermarking of HW IPs/ Secure IP

Mohammad Tehranipoor

ECE6095: Hardware Security & Trust
University of Connecticut
ECE Department

Contents



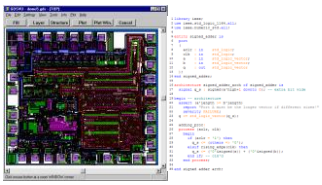
HardWare IntellectualProperty (HW IP)

- Why HW IP?
  - Design reuse
  - System-on-Chip (SoC)
  - FPGA
- Players
  - IP designer
  - System integrator
  - Foundry
- What qualifies as an HW IP?
  - HDL code, GDSII, netlist, layout, technique, ...

Figure from http://pdf.directindustry.com/pdf/cypress-semiconductor/cypress-psoc-programmable-system-on-chip-brochure/34220-70363.html

HW IP Lexicon

- Soft IPs
  - HDL codes
- Firm IPs
  - Placed RTL design
  - Fully placed Netlist
- Hard IPs
  - GDSII file



GDSII file HDL code

Figure from http://www.artwork.com/gdsii/gdsii/tiff/apd_note/apd_note.htm

IP Piracy: Vulnerabilities

- Hard IPs
  - Overproduction
  - Reverse engineering
  - Cloning of FPGA bitstream
- Firm and soft IPs
  - Cloning
  - Reverse engineering

IP Piracy: Threats

- IP Cloning
  - By system integrator
  - Use of HW IPs without license
  - Production of compatible IPs are not IP cloning
- Overproduction
  - By foundry or system integrator

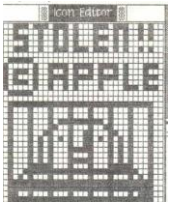
IP Piracy: Threats, cont'

- Reverse Engineering
  - By 3rd party system integrator
  - On hardware: is often legal
    - Often used to determine infringement
    - Often banned in software EULA
  - Types
    - In order to make cloning possible
    - Also is crucial in attacking secure IP designs
  - Methods
    - IO analysis
    - Decap
    - Thermal imaging

7

Approaches to Secure IP

- Prevention/Protection
  - Chemical (Military)
  - Obfuscation
  - Expiration of Service
    - Periodic licensing
    - Trojan insertion
  - Encryption
    - FPGA bitstream
- Detection/Identification
  - "Stolen from Apple"
  - Watermarking
  - Security tagging



"Stolen from Apple" icon in MacIntosh® ROM, inserted after *Apple v. Franklin Computers* lawsuit, intended as identification method

8

Secure IP Design Goals

- Robustness
  - Higher predictable performance
  - More design flexibility
- Overhead
  - Less risks
  - Acceptable constraint
- Tradeoff
  - Pre-processing: more robustness, more overhead
  - Post-processing: less overhead, less robustness

9

Obfuscation

- Denies unlicensed usage
- normal functional behavior is enabled only after application of a specific input sequence
  - Inserting finite state machine to obfuscate IO

Secure IP: Periodic Licensing

- System composition
  - License token
  - License controller
  - Timer
  - Non-volatile memory
- Goal
  - Disable the IP core when expire/fail to access
- Method
  - Encrypted token (Encryption)

11

Encryption: Secure FPGA

- Why is FPGA vulnerable?
  - Configuration bitstream stored in EPROM or Flash
    - When power on -> download
    - When power off -> removed
- Encryption
  - used to prevent interception (Prevention)
  - Limitations:
    - Key is exposed to a lot of people.
    - Battery is needed to “power on” the Encryption key.
    - Vulnerable to side channel attacks.

12

Encryption: Secure FPGA, cont'

- One Time Programmable FPGA
  - No encryption and on-site battery is needed
  - Configuration data is permanently burned in chip.
  - Based on Antifuse technology.
  - Limitations:
    - Expensive
    - Inflexible
    - Difficult to test
- Non-volatile Reprogrammable FPGA
  - Configuration data stored in on-chip non-volatile memory.
  - Robust security scheme prevents readback of configuration data.
  - Present technology even prevents direct probing of the memory cell.
  - Limitations:
    - Extra non-standard voltages are needed to reprogramming.

13

Secure IP: Security Tagging

- System composition
  - Tag to transmit label of the IP core
  - Wand to detect that label
- Goal
  - Identify the existence of the IP core and determining its version
- Method
  - Covert transmission channel (obfuscation)

14

Secure IP: Watermarking

- Why watermarking?
  - Uniquely identify IP Cores
  - Ease detection of piracy and counterfeiting
  - Trace counterfeit parts back to the source
  - Deter piracy

15

Digital vs. IP Watermarking

- Digital Watermarking
  - Performed on data: Image, audio, video, ...
  - Insert unique signature into the data
  - Data now carries a unique identifier
  - Invasive: the data is altered by the watermark
- IP Watermarking
  - Performed on IP
  - Noninvasive: data can't be altered lest function be altered

16

Digital vs. IP Watermarking Example

- Cartographer's watermark
  - "Trap streets" can be used on maps
  - However, this cannot enter navigational data (e.g. GPS) lest hurt the functionality

17

IP Core Watermarking: Principles

- Must not alter the functionality of the IP Core
- Performance degradation should be unnoticeable
- Should be hard to detect or remove
- Overall goal: high probability of authorship

18

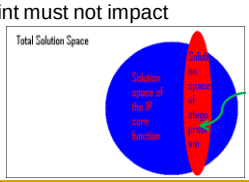
Topics in IP Watermarking

- Major approaches
  - Constraint Based watermarking
  - Additive watermarking
- Evaluation of watermarking technique
- Method of application
  - Test based watermarking
  - Don't Care Condition watermarking
  - Power Analysis watermarking
  - Placement and Route based watermarking
- Attacking and defending watermarks

19

Constraint Based Watermarking

- Author's signature → a set of constraints
  - Watermarking constraints in addition to functional constraints
  - More constraints, higher probability of authorship
- Choice of constraint must not impact performance



20

Boolean Satisfiability Problem (SAT)

- Set of Variables
  - U={u_1, u_2, ..., u_n}
  - u_i = 1 or 0, i∈[1,n]
- Clauses
  - Means logic OR; for example {u_1,u_2} means $u_1|u_2$
- Satisfiability
  - Is there an assignment of U that satisfy all clauses?
- Example

$U = \{u_1, u_2\}; C = \{\{u_1, u_2\}, \{\overline{u_1}\}, \{\overline{u_1}, \overline{u_2}\}\}$

$U = \{u_1, u_2\}; C = \{\{\overline{u_1}, u_2\}, \{u_1\}, \{\overline{u_1}, \overline{u_2}\}\}$

Example from ref[1]

21

Method to Add Constraint

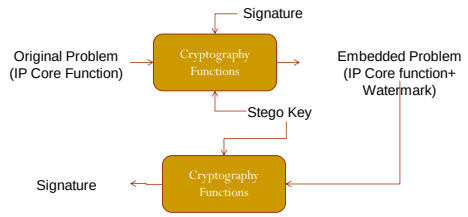
- Assuming function of the IP is described by example problem to the right
- Task: To modify this SAT problem so that
  - Any solution to modified problem satisfies old problem
  - Both modified problem and solution contains information uniquely identifies author

$$U = \{u_1, u_2, \ldots, u_{14}\}$$
$$C = \{\{\bar{u}_1\bar{u}_2u_9\}, \{\bar{u}_1\bar{u}_3\bar{u}_4\}, \{\bar{u}_1u_2\bar{u}_5\}$$
$$\{u_1\bar{u}_2u_{10}\}, \{\bar{u}_1\bar{u}_3u_8\}, \{\bar{u}_1\bar{u}_3u_7\}$$
$$\{u_1\bar{u}_5u_7\}, \{\bar{u}_1\bar{u}_6\bar{u}_{12}\}, \{\bar{u}_1u_{10}u_{12}\}$$
$$\{\bar{u}_1u_6u_9\}, \{\bar{u}_2\bar{u}_3\bar{u}_{10}\}, \{u_2\bar{u}_5\bar{u}_{14}\}$$
$$\{\bar{u}_2u_7u_8\}, \{u_2\bar{u}_8u_9\}, \{u_3u_4u_8\}$$
$$\{u_3u_5\bar{u}_7\}, \{\bar{u}_3u_8u_{13}\}, \{u_3\bar{u}_9\bar{u}_{11}\}$$
$$\{u_3u_{10}\bar{u}_{12}\}, \{\bar{u}_4\bar{u}_7\bar{u}_8\}, \{\bar{u}_5\bar{u}_8\bar{u}_{12}\}$$
$$\{u_4\bar{u}_7u_{13}\}, \{\bar{u}_5\bar{u}_9\bar{u}_{11}\}, \{\bar{u}_5u_7u_9\}$$
$$\{u_6u_{10}u_{11}\}, \{u_6\bar{u}_8\bar{u}_{12}\}, \{u_7u_9\bar{u}_{12}\}$$
$$\{u_7u_9\bar{u}_{13}\}, \{u_9u_{11}\bar{u}_{14}\}, \{u_{10}u_{11}\bar{u}_{12}\}\}.$$

Figure from ref[1]

Constraint Based Watermarking, cont'

- Exemplary methods of application
  - Unused CLB (Combinational Logic Block) blocks
  - Path timing constraints



23

Advantages and Disadvantages

- By constraining CLB block
  - Easy to discover through reverse engineer
  - Easy to remove from bitstream
  - No performance degradation and minimal area overhead
- By constraining path timing
  - Sub path can also be used
  - Hard to detect or remove
  - No performance degradation, minimal area overhead

24

Additive Watermarking

- Method
  - Done at source code level, independent from layout constraints
  - Implanted into functional logic so as to prevent removal
- Advantage
  - Strength easily adjustable
  - Hard to discover or remove
  - Easy to read in stego key
- Weaknesses
  - Incurs area overhead
  - Degrades performance

25

Don't Care Condition Based Watermarking

- Method formulation
  - Function blocks that have unneeded input combinations
  - Therefore, outputs to these input combinations can be forced
- Application
  - To assert a '1', add one such input combination
  - To assert a '0', remove this input combination
- Example
  - Original function $f(a,b,c,d)=\bar{a}\bar{b}c+\bar{a}bd+b\bar{c}d$
  - To assert $\bar{a}\bar{b}\bar{c}\bar{d}=1$' , change it to $f(a,b,c,d)=\bar{a}\bar{b}c+\bar{a}bd+b\bar{c}d+\bar{a}\bar{b}\bar{c}\bar{d}$

26

Testing based Watermarking

- Method
  - Add watermark to the test circuitry at functional level
- Extraction
  - Scan-out
- Advantage
  - Very tough to remove
- Weaknesses
  - Large area overhead
  - Power consumption overhead

27

Other Testing based Watermarking

- Methods
  - Apply watermark as a header
    - Easy to implement, but easy to remove
  - Pseudorandom bit insertion
    - Harder to implement, added obscurity
  - XOR the watermark and test bits
    - Easy to implement, secure, but prone to errors

28

Power Analysis Watermarking

- Method
  - Add components that draw power at certain frequencies (not multiples of clock)
- Extraction
  - Monitor the power supply pin on the IC
  - Dynamic power consumption magnitude will be higher at certain frequencies
- Advantage
  - Very tough to remove
- Weaknesses
  - Large area overhead
  - Power consumption overhead

29

Place-and-Route based Watermarking

- Method
  - Region and grouping constraints
    - Constrain the physical placement of standard cells
    - Can cause severe performance degradation when not used properly
  - Row placement constraints
    - Deliberately place standard cells in even or odd rows of the layout with a grid abstraction

30

Place-and-Route based Watermarking, cont'

- Advantage
  - No area overhead
  - Strong proof of authorship easily attainable
    - Coincidence chance = 1 over 2 to the power of number of blocks placed
  - Removal or circumvention will most likely render the IP Core useless
    - IP core thus protected are hard/firm IP, therefore tempering requires reverse engineering its soft IP
- Weakness
  - Not applicable to soft IP

31

Evaluation of Watermarking Techniques

- Proof of Authorship
  - As low as possible
- Err on overestimation when exact value is hard to calculate

$$P_c \equiv P(X \le b) = \sum_{i=0}^{b} \left[(C!/(C-i)!*i!) * (p)^{C-i} * (1-p)^{i} \right]$$

'p' - probability of satisfying one random constraint by coincidence.
'C' - number of imposed constraints.
'b' - number of constraints unsatisfied.
'x' - random variable, represents how many of the 'c' constraints were not satisfied.

32

Attacking and Defending Watermarks

- Attacks
  - Ghost Signatures
  - Tampering
  - Forging
- Defenses
  - Watermark Obfuscation
  - Multiple Small Watermarks
  - Parity in Watermarks

33

Ghost Signatures

- Intention: To announce a watermark when there is none
  - So that you may announce it contains your watermark as well
- Methods
  - Starting from solution characteristics, try to figure out the input pattern from current solution
  - Try different signatures, hope for a collision
    - Unlikely
  - Addition of a new signature
    - Easy to disprove

34

Tampering

- Alter, damage, or remove the watermark
  - Prohibitively large amount of effort required
- Move backwards through design phase
  - Keep going back until before the watermark was added, then remove or replace it at will
- Depend heavily on reverse engineering previous design steps

35

Forging

- Objective: to subvert proprietor's watermark by inappropriately watermarking other solutions with proprietor's watermark
- Need to Steal the Private Key of an IP Author
- Usually prevented by encryption

36

Defense Against Attacks on Watermark

- Watermark Obfuscation
  - Against tampering
  - Make watermark harder to detect
- Multiple Small Watermarks
  - Against tampering
  - Make watermark harder to alter
- Parity in Watermarks
  - Against tampering
  - Detect and repair tampering
  - Often use XOR for parity check (whether sum is odd or even)

37

References

- A. Kahng and J. Lach, "Constraint-Based Watermarking Techniques for Design IP Protection" in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 20, October 2001
- A. Kahng and W. Mangione-Smith, "Watermarking Techniques for Intellectual Property Protection" in Proceedings of the 35th Design Automation Conference (DAC98), June 15-19 1998
- D. Ziener and S. Aßmus, "Identifying FPGA IP-Cores Based on Lookup Table Content Analysis", 2006 IEEE Xplore
- J. Lach and W. Magione-Smith, "Signature Hiding Techniques for FPGA Intellectual Property Protection", 1998 ACM
- J. Lach and W. Magione-Smith, "Robust FPGA Intellectual Property Protection Through Multiple Small Watermarks", 1999 ACM
- D. Zeiner and Jurgen Teich, "FPGA Core Watermarking Based on Power Signature Analysis", 2006 IEEE
- Y. Fan, "Testing-Based Watermarking Techniques for Intellectual-Property Identification in SOC Design", March 2008 IEEE Transactions
- Narasimhan, S.; Chakraborty, R.; Bhunia, S.; , "Hardware IP Protection During Evaluation Using Embedded Sequential Trojan," Design & Test of Computers, IEEE , vol.PP, no.99, pp.1, 0
- Chakraborty, R.S.; Bhunia, S.; , "RTL Hardware IP Protection Using Key-Based Control and Data Flow Obfuscation," VLSI Design, 2010. VLSID '10. 23rd International Conference on , vol., no., pp.405-410, 3-7 Jan. 2010

38