A Protection Mechanism for Intellectual Property Rights (IPR) in FPGA Design Environment

Wael Adi, Bassel Soudan,Nizar Kassab
wadi@ieee.org, bsoudan@sharjah.ac.ae, kassabnzr@netscape.net
Etisalat College of Engineering, University of Sharjah, Ajman University of Science and Technology
UAE

ABSTRACT

One of the major difficulties in offering new VLSI designs is protecting the designer's Intellectual Property Rights IRP. It often requires limited field deployment and testing before a novel implementation may be accepted for general use. The difficulty arises in the need to deploy the design for testing while disabling the tester from deciphering the design details. A similar requirement applies when the designer is interested in limiting the number of deployments as part of a business agreement. This work leverages the similarities between the issues of IPR protection in the hardware and software arenas and presents a novel solution to protect the use of designs in FPGA hardware environment. The mechanisms used are based on hardware-supported design encryption and secured authentication protocols.

Keywords: secured IPR, FPGA device uniqueness, provable identification, global authentication, VLSI identity.

1. INTRODUCTION

The major problem in disclosing VLSI designs for evaluation is the difficulty to protect the Intellectual Property Rights IRP of the design originator. New design ideas can only be accepted after they have been tested in a realistic environment. This leads hardware manufacturers to depend on their customers (who may eventually become their competitors) for evaluating these designs. A problem arises in protecting the designer's Intellectual Property Rights – IPR – in the process. How can a designer deliver a new system to an evaluator/competitor for testing and ensure that they won't be able to extract the design information from the system and violate the designer's IPR?

There are other situations that might be quite different from the above but pose similar difficulties and dangers. One of these would be the designer's interest in limiting the number of system deployments as part of a business agreement. This is an issue especially for small design houses who have to depend on out-of-plant chip manufacturers/programmers for the production of the final system. How can the designer limit the number of systems that can be manufactured from their design? This scenario is very similar to software copy-protection requirements. Here, the issue may not be protecting the actual design data. Rather the issue here is controlling the number of copies that can be made of a particular design. Therefore, limiting IPR violations in the manufacturing process as well as limiting IPR violations by possible production of illegal copies after the design has been introduced into the market.

Of particular interest in this area are FPGA-based designs. Field Programmable Gate Areas (FPGAs) have gained a lot of importance in the recent past as they present a useful vehicle for introducing new design concepts to the evaluators (and the market) quickly and economically. FPGAs have become the tool of choice for a large number of fab-less design houses. Especially those who deal with designs that don't require cutting edge speed or complicated implementations requiring customized layout. FPGAs have an advantage in the IPR protection area, as they appear to have some particular structures, which would allow implementing the necessary IPR security.

There is a very important difference between IPR protection for FPGA-based designs and software copyright protection methods; the FPGA device programmer has access to the internal structure of the FPGA device. The device programmer is an intermediate partner, and the real producer, between the end-user and the device manufacturer. This fact allows different IPR protection scenarios.

Our proposed solution allows the designer to sell a limited number of copies, say m, of his design to run only in m different FPGA devices. This assumes that the FPGA device manufacturer would offer a provable unique identity for every delivered FPGA device.

The proposed system offers mutual, provable and traceable IPR transfer with limited number of uses. All parties are securely traceable in the entire commercial and technical transaction. The system is breakable only if the manufacturer cheats; this can be easily traced by using a globally authenticated mechanism to prove the device uniqueness. The mechanisms and algorithms required for implementing the scenario are presented in

this paper. The mechanisms employed are based on trustable secret-key low-complexity functions similar to those described in [1], [2] and [3].

The rest of this paper is organized as follows: Section 2 describes the details of the IPR protection scenario, Section 3 describes the IPR protection mechanisms for architectures, Section 4 details the security threats and possible attacks and Section 5 presents a summary and conclusion.

2. "IPR" PROTECTION SCENARIO

The scenario for implementing our proposed solution to the IPR protection issue can be summarized in the following required steps:

1. The FPGA manufacturer produces devices with secured unique identity and takes the responsibility of delivering only such devices. A violation of this responsibility is easily traceable as the device uniqueness must be global.

2. The IPR carrier offers his VLSI design – such as a core function performing a particular task in a certain FPGA technology. The binary stream representing this design is to be securely downloaded into the pre-defined devices to run this function.

3. The end-customer orders from the IPR carrier a certain number of copies of the offered design – say m copies. In the order the end-customer assigns the unique FPGA device identities DI's in which the design should run. For more security, the device owner (end-customer) can prove that he owns the devices by delivering a challenge response proof using his own random challenges together with eventually random challenges from the IPR carrier.

4. After receiving the order, the IPR carrier forwards the device identities to the manufacturer (eventually with the proof responses if it was requested during the ordering process by the IPR carrier) asking for authentication keys for these particular devices. The manufacturer delivers the requested IPR protection keys PK's and certifies that the devices are owned by the customer by checking his responses (if available). In addition to that the manufacturer can verify the identity of the IPR carrier and certify that he received the IPR protection keys.

5. Using the IPR protection keys PK, the IPR carrier then generates for every device its own ciphered binary design stream C-BDS together with a set of random challenges and delivers m ciphered binary design streams to the end-customer.

6. The end-customer downloads the m ciphered binary design streams to the particular m FPGA devices designated in the original purchase order. Every design stream runs only on the device uniquely identified in the order. The stream gives no information about the design structure itself.

The proposed system offers mutual, provable and traceable security transaction, which can be performed using open Internet media without loss of security. All parties are securely traceable in the entire commercial and technical transaction. The mechanisms employed are based on trustable secret-key low complexity functions such as those used in mobile systems [1], [2] and [3]. A public key mechanism is in preparation.

3. IPR PROTECTION MECHANISM FOR FPGA ARCHITECTURES

The proposed system implementation includes hardware and software components. The hardware component is the *device identity module* DIM that should reside in the FPGA in a secured area. This module will guarantee the essential physical uniqueness of each device. The location of the module is to be selected such that no attack would be possible on the module in any operation mode. Figure 1 represents a simplified functional block diagram for the proposed device identification module.

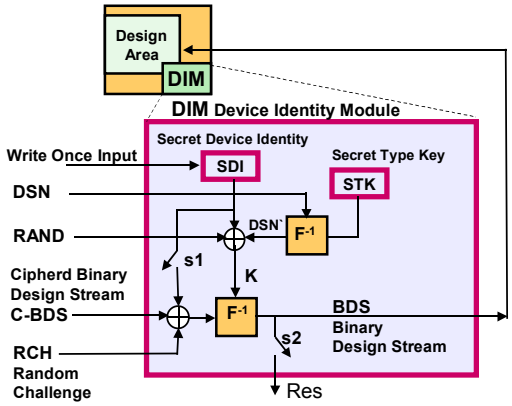

Figure 1. Architecture of a Possible Device Identity Module

The module includes mainly a non-volatile *write once memory* register, which accommodates the *secret device identity* SDI. The manufacturer selects a unique open *device identity* DI, which can be branded on the device itself and/or stored in a readable area in the device. The secret device identity SDI is mapped from DI by some keyed hash function in a secret way selected by the manufacturer such that no repetition is possible. A strong cipher with a size of 128 bit – such as AES – can be used as a hash function. A mapping cipher function F^{-1} – again such as AES – in deciphering mode should be implemented as a hardware block in the module with SDI as a secret key input and the *ciphered binary design stream* C-BDS as a cipher text input. The output is the clear text representing the clear *binary design stream* BDS that represents the design layout in the FPGA. The

whole structure should be floor planned such that no way is possible to reach BDS and SDI in any direct or indirect method.

To check the response of the device to a challenged random value RCH the output is switched to an output "Res". The cipher text input is XORed automatically with the secret value SDI by closing the switches s1 and s2 in that case.

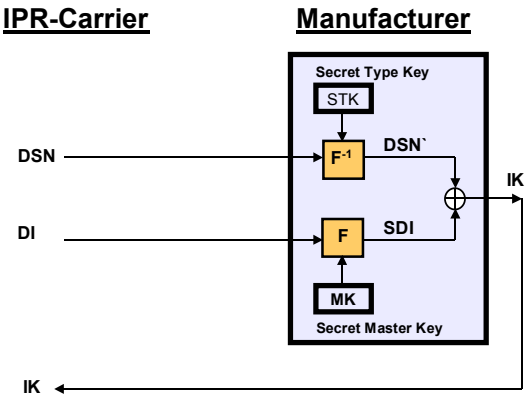

Figure 2. IPR Carrier-Manufacturer Transactions

Figure 2 shows the required transactions and operations between the IPR-carrier and device manufacturer. The IPR-carrier generates his own unique design serial number DSN for each design copy he wants to sell. For every device to be licensed, the IPR-carrier then sends its DI number and the particular DSN. As shown in Figure 2, the device manufacturer generates the intermediate key IK where:

$$IK = F^{-1}(STK, DSN) \oplus SDI = DSN' \oplus SDI \quad (1)$$

IK is generated individually for every device and sent back to the IPR-Carrier. The IPR-Carrier then generates the following license token LT for every device from the customer as shown in Figure 3.

$$LT = C\text{-}BDS \mid RAND \quad (2)$$

Where C-BDS is the ciphered binary design stream BDS enciphered by using the key K where

$$K = IK + RAND \quad (3)$$

RAND is a random number generated by the IPR-Carrier uniquely for each license.

The customer uses the received license token LT to feed the corresponding device with the device identity DI.

The binary vectors C-BDS, DSN and RAND are sufficient to generate the clear binary design stream BDS internally in the corresponding device. Notice that a correct BDS would result only if the device has the right unique identity. The customer would not be able to see the design stream nor replicate it. The whole communication transactions need not be communicated secretly as no one can make use of the communicated information.

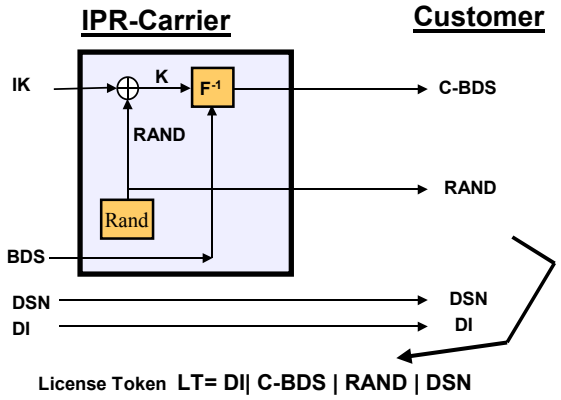

Figure 3. IPR-Carrier with Customer Transaction for One Design Copy

4. SECURITY THREATS AND POSSIBLE ATTACKS

The system is breakable only if the manufacturer cheats. However, this can be easily proven, as the device uniqueness should be a globally authenticated mechanism. The system includes a hardware identification module, which should reside in every FPGA device to achieve physical uniqueness. The device manufacturer is the only one who can attack the presented mechanisms. In that case the manufacturer needs to cooperate with the customer. He can either generate devices with the same device identity DI or take the individual random number of the device to generate the clear binary design stream BDS. The fist case can be traced, as duplicated-identity can be detected by challenging existing devices and checking for uniqueness. The second case can also be traced as the manufacturer would not be able to find BDS if the customer did not violate the license agreement and release the secret random number to the manufacturer. In such a case the cooperation can be also traced. These two attacks can be prohibited if the IPR-carrier would personalize the devices himself and insert secretly the secret type key STK. This requires however that the devices should be shipped to the IPR-carrier for that purpose. In that case the manufacturer can no longer attack the system other than by building hardware backdoors in the manufactured FPGA architecture.

5. SUMMARY AND CONCLUSION

The proposed system offers new mutual, provable and traceable security transactions to protect IPR in FPGA design environment. The secured design distribution can be established by using simple Internet media without security loss. All parties are securely traceable in the whole commercial and technical transaction. The system is breakable only if the device manufacturer cheats which can be easily proven as the device uniqueness incorporates global authentication. The technical and

procedural requirements as well as all security mechanisms are described for the proposed system.

6. REFERENCES

[1] Adi, W., "Secured Mobile Device Identification with Multi-Verifier", Proceedings of the International Conference on Telecommunications (ICT2001), pp. 289 – 292, 2001

[2] Technical Specification 3G Security, Security Architecture 3G TS 33.102 V. 3.2.0 from 10.1999

[3] Specification of the MILENAGE Algorithm Set. 3GPP TS 35.206 V5.0.0 ETSI, http://www.3gpp.org, 2002.