IP Based Design 2002

Session : 4A. IP Business Models

Cryptographically Enforced Pay-Per-Use Licensing of FPGA Design Intellectual Property

Tom Kean, Algotronix Ltd., Edinburgh UK
(tom@algotronix.com)

Abstract :

At present FPGA IP vendors cannot determine how many times their design has been programmed into an FPGA by a customer. This has forced them to adopt an up-front charging model where a customer obtains unlimited use of an IP block for a single relatively large payment. This pricing model is fundamentally unsuited to the FPGA market since the basic motivation for using an FPGA rather than an ASIC is to trade off a higher per-unit cost to avoid a large up front NRE payment. This paper suggests upgrading the cryptographic circuitry presently included on FPGAs to prevent bitstream piracy so that it can also support pay-per-use royalty collection on FPGA IP cores and entire FPGA designs.

INTRODUCTION

Even though 'platform' FPGAs such as Xilinx's Virtex II Pro family have now reached densities as high as 8 million 'system gates' and FPGAs continue to take market share from conventional ASICs the Intellectual Property industry supporting FPGAs is commercially insignificant. FPGA Intellectual Property is a sideline for a few Semiconductor Intellectual Property (SIP) vendors and some design services companies. The FPGA vendors themselves also have internal IP development functions which generate little revenue and are subsidised by their chip sales.

At first sight FPGA IP vendors appear to have considerable technical advantages over Silicon IP vendors. Not only are FPGAs taking an ever increasing share of the total ASIC market but there are no testing issues, no manufacturing yield issues and no electrical compatibility issues for FPGA IP providers. Moreover, while a System on Chip designer working at 0.13um may be risking $700K in NRE and a six month delay on their project if they choose the wrong IP vendor the risks for an FPGA designer are much lower. Nevertheless, the FPGA IP industry has been even less successful than the Silicon IP industry. There is a simple reason for this: the up-front license fee business model the industry has adopted does not fit the low volume applications and constrained development budgets of FPGA designers. After all, a designer turns to FPGAs precisely because they prefer higher per-unit costs to up-front Non-Recurring Engineering (NRE) payments. Therefore, it is not surprising that they also resist paying NREs for intellectual property.

This poor match between business model and customer base has deterred companies from entering the FPGA IP core market and, at present, a high percentage of the available cores are supplied by the FPGA vendors – either free of charge or for nominal fees – in order to stimulate chip sales. As FPGA densities continue to increase it will become impossible for FPGA vendors to provide all the necessary cores free of charge. It is in everyone's interest: FPGA customers, FPGA vendors and third party IP suppliers to find a sustainable and enforceable 'pay-per-use' business model in order to create a viable market for FPGA IP cores.

The pay-per-use model also enables companies to offer complete FPGA design bitstreams as 'virtual ASSPs' to address applications which due to small volume or low gate count cannot be addressed economically by custom chips. Such virtual ASSPs, just as conventional ASSPs, can be used by board level designers without any FPGA design expertise or FPGA design tools.

Xilinx has recently implemented cryptographic circuitry on its Virtex-II Pro family of FPGAs in order to prevent 'cloning' and reverse engineering of bitstreams containing user designs by pirates [2]. Several patent applications [3,4,5,6] and technical papers on alternative schemes for protecting FPGA bitstreams have also been published [1,7].

Extensions of these schemes allow the cryptographic element not only to protect against reverse engineering and piracy but also to protect bitstreams during in-the-field downloads of new configurations.

This paper will show how on-chip cryptographic circuitry, in conjunction with an e-commerce server computer operated by the FPGA vendor or a trusted third party can also enforce a pay-per-use licensing model for FPGA intellectual property without inconveniencing the user of the IP [7].

SECURITY PROTOCOL

The proposed IP licensing protocol involves a Trusted External Party (TEP) which operates the e-commerce servers which implement the licensing scheme and is trusted by the FPGA vendor, FPGA customers and IP vendors to operate fairly. Although the TEP is logically an external organisation for the purposes of the protocol in practice the FPGA vendor may choose to offer this service itself.

The protocol has several phases starting at the time when the FPGA is manufactured.

1. Each FPGA is manufactured with a secret 'chip' key and a non-secret but difficult to alter unique chip identifier embedded on the chip. Various technologies are available to encode these numbers including anti-fuse and Flash EPROM – only a few hundred bits of non-volatile memory are required. The secret key need not be stored anywhere outside the FPGA and all record of it can be destroyed as soon as it is programmed into the chip. This key does not even need to be unique.
2. During product testing the FPGA manufacturer allows a computer owned by the TEP to connect to each FPGA's JTAG interface. This computer creates a random 'trusted external party' (TEP) key and presents it to the FPGA via JTAG which then returns its chip identifier and the TEP key encrypted using its on-chip secret key. The TEP key and the encrypted TEP key are stored in a database indexed by the chip identifier.

The encrypted TEP key is called a 'token': it can be freely stored in insecure memories (e.g. Flash memories in user equipment) and communicated over insecure channels but when presented to the FPGA it allows the FPGA to calculate its secret TEP key. The TEP key is a shared secret between the TEP and the FPGA which can be used to encrypt communications between them. The TEP's key database allows it to establish secure communications with any FPGA at a later date using this shared secret.

The structure of the 'token' is shown in figure 1. If the triple DES cipher is used the IV and checksum fields are 8 bytes long and the TEP key is 24 bytes. The field labelled IV is a random 'Initial Value' used in the industry standard Cipher Block Chaining (CBC) mode of encryption. The purpose of the IV is to ensure that if two identical pieces of data are encrypted there is no correlation between the corresponding encrypted data. This increases the ciphers ability to resist chosen plaintext attacks.

Anyone who has the FPGA in their possession can present it with a new key and obtain a token and the tokens can be used for other purposes than IP licensing. Creating a token is a general purpose mechanism which allows someone in possession of the FPGA to initiate secure communications with it in the future. For example, an equipment manufacturer might create tokens to support secure download of bitstreams to upgrade a product in the field. The only difference between this and the TEP is that the TEP's relationship with the FPGA manufacturer allows it to obtain a token for every FPGA manufactured not just those it purchases.

As well as a database of tokens allowing it to communicate securely with every FPGA chip the TEP also needs a database of information about IP cores.

3. IP vendors communicate design identification, security information and pricing information on their cores to the TEP.

The next phase of the protocol concerns what happens when a customer makes use of a licensed core.

4. When a customer wishes to use an IP the core vendor supplies the IP design information in an encrypted format. FPGA implementation tools supplied by the FPGA vendor process a design containing licensed IP as normal except that bitstream files (and other files containing low level design information like netlist files) are created in an encrypted format which cannot be used directly to program an FPGA. This encrypted bitstream also

contains copyright information on all licensable cores within the design. This copyright information is secured against tampering using cryptographic checksums but need not be encrypted.

The next phase of the protocol concerns making use of the bitstream to configure individual FPGAs. This is the point at which pay-per-use licensing fees are collected.

5. During manufacturing of equipment containing an FPGA trusted programming software supplied by the FPGA vendor reads in the encrypted design bitstream from the CAD tools and extracts the copyright information identifying the licensed cores. It also communicates with the FPGA via JTAG and obtains its unique identifier.
6. The trusted programming software communicates via the internet using a standard secure protocol such as SSL with a server owned by the TEP and supplies the name of the customer using the programming software, the copyright information from the bitstream and the FPGA unique identifier. The TEP server indexes its database using the FPGA's unique identifier to obtain the corresponding TEP key and token which it supplies to the trusted programming software.
7. The TEP server bills the account of the customer requesting that the FPGA be programmed and credits the accounts of the IP core suppliers.
8. The trusted programming software decrypts the encrypted bitstream and re-encrypts it with the TEP key for this particular FPGA. It then prepends the token obtained from the TEP server to the beginning of the bitstream. This complete encrypted bitstream is stored in non-volatile memory within the product containing the FPGA.

   The next phase of the protocol takes place whenever the FPGA chip loads its encrypted bitstream from local memory.

9. When the FPGA finally reads in the bitstream it uses its on chip secret key to re-calculate its own TEP key by decrypting the token at the beginning of the bitstream. Using this TEP key it can successfully decrypt the bitstream information and program its configuration memory. No other FPGA can successfully decrypt the token to determine the correct key to decrypt the bitstream. Therefore the customer must make a request to the TEP server each time they wish to program an FPGA chip with licensed IP.

OVERHEAD OF THE PROTOCOL

When considering a scheme such as this it is important that the additional costs and inconvenience it causes are outbalanced by its benefits.

This scheme requires encryption circuitry on each chip and that each chip has an individually programmed secret key and identifier. The largest vendor of FPGAs, Xilinx Inc., already includes a suitable triple-DES encryption circuit on its chips to prevent bitstream piracy. Therefore, it would be a small step to provide the additional on-chip functionality required by this scheme. The main additional capability required is a small non-volatile on-chip memory since the battery-backed on-chip key register provided by Xilinx is not suitable for this application. The extension to the on-chip circuitry has the additional benefit of supporting secure download of bitstream information to FPGAs in the field.

The scheme requires a method of supplying encrypted IP core design information to FPGA designers. Several methods have been proposed and some are in common use to allow evaluation of IP cores prior to purchase. Most such schemes, for example Altera's OpenCore [9], prevent generation of bitstream information from encrypted IP, where this scheme allows bitstream information to be generated but encrypts it. No information on the user design beyond a list of the licensed IP cores it contains is provided to the TEP.

The scheme requires a secure e-commerce server computer. The database of tokens, unique identifiers and keys would likely require only a few gigabytes of disk space.

The scheme requires secure programming software be used to deal with encrypted bitstreams containing IP cores. This is a relatively simple and straightforward application which could easily run on a personal computer. All recent FPGA families use JTAG interfaces for programming and so the requirement that the FPGA report its unique identifier over JTAG to the programming software is easily met.

Naturally, should a user design not contain any licensed IP the FPGA software would create a normal un-encrypted bitstream file so customers who do not use licensed IP can program their

FPGAs in the normal manner without any contact with the TEP.

The scheme as described above relies on secure software running on the users computer to protect design information. A potential drawback of this method is that a hacker may try to reverse engineer and circumvent the protection methods in the software. One way of reducing this risk this is to complement the software with tamper resistant hardware (such as a dongle) which implements cryptographic functions. Another approach is to carry out operations involving the licensed IP files on the TEP server rather than the user's computer. This would provide better security for the IP but is likely to be less convenient for the FPGA customers.

CUSTOMER BENEFITS

The cryptographic circuitry on the FPGA protects bitstreams from reverse engineering and piracy and provides a mechanism for secure download of bitstreams to devices in the field. These benefits, rather than the ability to enforce a pay-per use IP licensing scheme, will provide the initial motivation to add the necessary hardware to FPGA chips.

An enforceable pay-per-use IP licensing scheme will allow IP vendors to make their IP available without large up-front charges. This will make using IP much more attractive to the majority of FPGA customers who have designs with small or medium volumes. Further, there is no longer a need for special 'evaluation' arrangements prior to purchase since no substantial costs are incurred until a large number of chips are programmed.

Pay-per-use licensing ties IP costs directly to sales and reduces business risk for all IP users. It is to be expected that pay-per-use licensing will allow the FPGA IP industry to achieve significantly higher revenues compared to up-front licensing fees. This will benefit FPGA users by increasing the amount of high-value proprietary IP being made available on FPGAs. It will also potentially create a market for complete FPGA designs sold as 'Virtual' Application Specific Standard Products to system companies with no in-house FPGA design capability.

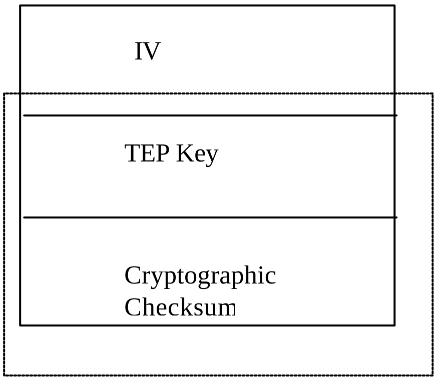
FPGA companies will benefit by increased sales of FPGA chips and a reduced requirement to develop IP in house to supply free of charge to their customers. If they choose to take on the role of TEP themselves they would also create a revenue stream from transaction fees.

SUMMARY

By making use of cryptographic circuits within the FPGA originally created to protect bitstreams from piracy coupled with a secure e-commerce server and small changes to the design flow FPGA vendors can enforce pay-per use licensing of IP cores. This will support the development of a viable IP industry to serve their customers while creating a worthwhile additional revenue stream for themselves. Many extensions and variations of the protocol outlined here are discussed in [6].

REFERENCES

1. Dipert, B., "Cunning Circuits Confound Crooks", EDN Magazine, October 12, 2000.
2. Xilinx Inc., "Using Bitstream Encryption", in Chapter 2 of the Virtex II Platform FPGA Handbook available from www.xilinx.com.
3. Austin, K., "Data Security Arrangements for Semiconductor Programmable Devices", US Patent 5,388,157
4. Algotronix Ltd., "Method and Apparatus for Secure Configuration of a Field Programmable Gate Array", US Patent Application 21015919A1.
5. Algotronix Ltd., "Method of using a Mask Programmed Key to Securely Configure a Field Programmable Gate Array", European Patent Application, EP01244330A2
6. Algotronix Ltd. "Method of Protecting Intellectual Property Cores on Field Programmable Gate Array", Unpublished Pending US Patent Application.
7. Kean, T. "Cryptographic Rights Management of FPGA Intellectual Property Cores", Proceedings ACM Conference on FPGAs, FPGA 2002 Monterey CA, Feb 2002.
8. Kean T. " Secure Configuration of Field Programmable Gate Arrays", Proceedings FPL 2001, Belfast UK, September 2001.
9. Altera Inc., "Evaluating AMPP and MegaCore Functions", Application Note 125, Version 2.0, April 2001.



Encrypted with Chip Key

FIGURE 1. Format of shared secret ‘Token’