CHAPTER 5

IP CORE PROTECTION OF FPGA BITSTREAM USING SECURE START HARDWARE

This chapter explains protection of FPGA bitstreams using Secure Start Hardware (SSH) in FPGA. This research work focuses on establishing an effective method for embedding IP designer information starting from abstraction level. High security is provided to IP cores by using a novel hardware in the FPGA called Secure Start Hardware. Information given by IP core developer is passed to the FPGA via the synthesis tool. It is assumed that before selling or deploying the FPGA platform in the field, the IP core developer initializes it by storing the keys and data’s in SSH. The proposed SSH approach for protection of FPGA IP core uses a technique that ensures FPGA design security beyond bitstream encryption.

5.1 INTRODUCTION

In recent decades, FPGA have been widely used in various critical applications. By analyzing various researches, FPGA based system is developed and development behavior should be made secure in all phases. IP core private update from remote area is a benefit of FPGA-based devices. Whereby the system developer and user can modify or restore a part of the equipment as mentioned in Tingyuan et al (2010). Many cores are used and reused, which have been written for other projects or were obtained from other sources. They present a modular concept, fast development-cycles,
tremendous cores commercially exist today from specialized core developers. IP cores are licensed and disseminated like software.

System developer purchases the IP core from various IP core developers and integrates it to form SoC. IP core developer receives the FPGA from FPGA manufacturer and supply it to the customer along with supporting device and software. It is mentioned in Figure 5.1.

![Figure 5.1 FPGA based IP Core Protection](image)

The main serious problem in FPGA based IP core is the deficient of security against unlicensed usage. The flexible and impulsive nature of an FPGA configuration is the key to many of the advantages related with FPGAs, but also opens the door to IP abuse; e.g., copying an FPGA design is easy because an FPGA’s configuration is more efficient in digital data. Another drawback is that common tools are cannot frequently handle encrypted cores and that the shipped tools can be splintered Sridhar & Prabha (2013). The FPGA vendors’ main answer to this is to maintain configuration encryption by given that an on-chip cryptographic decryption phase and secure key storage. This technique is more efficient adjacent to direct cloning of FPGA configurations in business end products, however it make use of is relatively firm and sacrifices part of the FPGAs flexibility. The lack of a key management service makes implementing the decryption keys in the devices a tedious and security-sensitive task for a system developer as discussed in Zhang & Chang (2012). Furthermore, the protection given is not suitable to
the modular design approach discussed earlier, since only a single “monolithic” FPGA design can be encrypted. In a system-level FPGA design platform, integrating many IP cores from different parties, the need for IP protection at a modular level, additionally to the system level, is evident.

FPGA based IP core protection has been an attractive research area owing to the amount of cost and time spent for protection the IP cores. Alderighi et al (2007) introduced SRAM based reprogrammable FPGAs for protecting FPGA based IP core. The SRAM based reprogrammable FPGA are responsive to radiation-induced Single Event Upsets (SEU), not only in their respective user flip-flops and memory, but as well in the configuration memory. FLIPPER fault injection platform has been presented which consent to testing the efficiency of the SEU mitigation design. FLIPPER imitates SEU-like faults by processing partial reconfiguration and then implements stimulus resultant from HDL simulation, whereas comparing the outputs with the golden pattern, also resulting from simulation. FLIPPER has its Device-Under-Test (DUT) FPGA on a mezzanine board, allowing an simple exchange of the DUT device.

Cryptographic techniques are also used for the IP protection in FPGA. Jorge et al (2007) introduced a new protocol for the IP protection problem on FPGAs based on public-key cryptography and analyzed the advantages and costs of such approach. The author observed that a major advantage in using private key based protocols is that it allows for an implementation in which the private key stored in the FPGA never has to leave the device, thus increasing security. Finally, notice that this comes at the cost of extra hardware resources but not at significant performance degradation.

Kumar et al (2008) presented a new construction of a PUF which can be used for all the various types of FPGAs. The Butterfly PUF uses the
internal matrix of the FPGA to uniquely identify it based on the intrinsic physical characteristics of the integrated circuits. Experimental results show that it is very stable to environmental and other FPGA operating parameter variations. Hence, the Butterfly PUF promises to be a significantly secure way to protect IP with no additional costs in manufacturing. Other secure features like volatile key generation for cryptographic applications are also shown to be feasible due to the low noise levels.

Roy et al (2008) introduced the first bus-based IC locking and activation scheme, that works by uniquely locking each chip at the manufacturing site. The locking is performed by unique random IDs on each chip and Diffie Hellman key sharing between the IP rights owner and the chip. The author demonstrates the flow of the new scheme, discuss its wide range of applications, devise an implementation based on permutations and one-way functions, and present the attacks and countermeasures. Evaluation results confirm that the locking scheme has a very low overhead while it is highly resilient against attacks.

Although the traditional FPGA bitstream protection algorithms gives secure results, some major issues araised during the real time implementation of FPGA. There is lack of throughput, latency as well as area are one of the major issues found in the traditional FPGA IP core protection techniques. Since the traditional algorithms are lack of security, throughput, latency, the overall performance of the FPGA protection is also affected.

Various cryptographic algorithms are proposed by several researchers including DES, Triple DES and AES which provides security at the costs of increased hardware complexity. In case of control word based FPGA system, bitstream are not encrypted and an attacker who copies the bit stream may download this bitstreams into this FPGA to extract
original design. But, the system asks for the proper control word before proceeding further. As the intruder is not aware of the control word, attack is failed.

FPGA uses bitstream encryption method to protect IP cores once it is loaded onto the FPGA. Static Random Access Memory based FPGA are volatile and the requirement of configuring on each power up results in attacks such as cloning, reverse engineering or tampering of the bitstream as discussed in Jerraya & Wolf (2005). To overcome these issues, a novel scheme is proposed which supports unique identification of IP core developer and system developer. Because of the security feature incorporated into the IP core unit the system developer can be sure that the IP cores they have received are not substituted ones. The IP core developer is also assured because in this scheme the usage of the IP cores is restricted to the authorized person.

Key storage in FPGA plays an essential role. The attacker tries to identify the key. Battery backed key techniques which prevent non-encrypted bitstreams may not be effective if the battery is removed or the battery temporarily is shortened until the battery is dead.

As the FPGA designs become larger and more integrated illegally copying them becomes easier and more profitable. Static random-access memory, or SRAM, is a type of semiconductor memory which uses bistable latching circuitry in order to store each bit. SRAM is different from dynamic RAM, which must be refreshed periodically.

5.2 PROPOSED SECURE START HARDWARE

This research work focuses on establishing an effective method for embedding IP designer information starting from synthesis tool level. High
security is also provided to IP cores by using a novel hardware in the FPGA called Secure Start Hardware. A minimal cryptographic protocol is introduced to achieve an authenticated channel between the system developer and SSH. The proposed methodology flow is explained in the Figure 5.2. Secure start hardware is designed inside the FPGA.

![Figure 5.2 Proposed Methodology Flow](image)

Authentication is the mechanism in which one entity proves its identity to another entity. 128 bit AES symmetric cryptography has been employed in this proposed approach because of standardization case of implementation and better performance when compared to other alternatives as discussed in Rivest et al (1978). Security of AES relies on the secure storage of the key and the inability of an adversary to compute the key given both plain text and ciphertext as described in Chakraborty & Henriquez (2009). It is assumed that the cryptographic and security parameters chosen are computationally secure such that no attacker will be able to compromise the system by any attacks. The technology employed in proposed SSH approach provides an effective authentication feature for IP cores in addition to encryption techniques are widely used for IP protection.

The architecture of SSH hardware designed in FPGA is shown in Figure 5.3. A cryptographic module may use Pseudo Random Number Generator (PRNG) as discussed in Dorren’dorf et al (2009) to produce cryptographic keys and other PRNG\texttt{CHALLENGE} internally. Moreover, in this chapter, $||$ is used as the concatenation operator and information given by SSH validation unit is passed to the FPGA via the synthesis tool.
Figure 5.3 Architecture of secure start hardware in FPGA

Figure 5.4 shows the block diagram of synthesis flow showing device information and SSH header. The main advantage of inclusion of device information is unauthorized person cannot able to access the IP core implemented in FPGA. On receiving the device information, the synthesis tool appends this information with user constraint. The MD5 hash output is given to PRNG. The resultant stream of pseudo random bits is used to generate a unique set of design constraints. These constraints are superimposed on original design.
The watermarked design is synthesized by synthesis tool. After synthesis, watermarked output specification is generated. It is encrypted using 128 bit AES algorithm. SSH header and file header is appended along with encrypted bitstream. The SSH hardware has two zones namely processing zone and secure zone.

5.2.1 SSH Secure Zone

It is assumed that before selling or deploying the FPGA platform in the field, the system developer initializes it by storing the following keys and data’s in SSH as shown in Table 5.1. The secure zone interface and control
logic act as an interfacing unit between SSH processing zone and SSH secure zone. It does not allow unauthorized person to access data stored in SSH secure zone.

**Table 5.1 Content of Flash / Battery Backed Memory, RAM and ROM Memory**

<table>
<thead>
<tr>
<th>Name</th>
<th>Slot address</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEY_MASTER_FPGA</td>
<td>0x01</td>
<td>Master FPGA key</td>
<td>Flash / Battery Backed Memory</td>
</tr>
<tr>
<td>PRNG_CHALLENGE</td>
<td>0x02</td>
<td>Pseudo random number</td>
<td></td>
</tr>
<tr>
<td>KEY_SSH_MAC_GEN</td>
<td>0x03</td>
<td>SSH MAC generation key</td>
<td></td>
</tr>
<tr>
<td>KEY_CONFIG_AES</td>
<td>0x04</td>
<td>AES configuration key</td>
<td></td>
</tr>
<tr>
<td>CMAC_SSH</td>
<td>0x05</td>
<td>Stored CMAC</td>
<td></td>
</tr>
<tr>
<td>INFO_SYN_TOOL_SPEC</td>
<td>0x06</td>
<td>Synthesis tool specification</td>
<td></td>
</tr>
<tr>
<td>DEVICE_TYPE</td>
<td>0x07</td>
<td>FPGA family Type</td>
<td>ROM</td>
</tr>
<tr>
<td>DEVICE_DNA</td>
<td>0x08</td>
<td>Unique number to each FPGA</td>
<td>RAM</td>
</tr>
<tr>
<td>KEY_DESIGN_0</td>
<td>0x09</td>
<td>Design key</td>
<td></td>
</tr>
<tr>
<td>KEY_DESIGN_1</td>
<td>0x0A</td>
<td>Design key</td>
<td></td>
</tr>
</tbody>
</table>

Generally KEY denotes the key of the particular memory slot, the security flag of that memory slot. COUNT denotes how many times KEY is updated. SSH secure zone consists of four memories. They are Battery backed secure memory, Flash memory, Key slot RAM, Device information ROM. SSH processing zone consists of AES Decryption module, Key update module, MAC generation, Configuration protocol, Device information module and SSH process.
The unique IDs provided to reconfigurable device stops the unauthorized entity to generate bitstream for FPGA. Customer is provided along with FPGA the SSH device and supporting device from FPGA manufacturer. SSH validation unit is outside the FPGA. It generates the commands and sends it to SSH through the EDA tool. The IP core developer purchases the FPGA device from the FPGA manufacturer.

PRBS is generated with a cryptographically secure variation on a Linear Feedback Shift Register (LFSR). The LFSR generates a sequence of random bits and the sequence repeats only once in a very long time. The LFSR is designed using a large shift register and some simple XOR gates that form the feedback circuit when n is the length of the shift register then PRBS repeats once every $2^{n-1}$ bits. LFSRs are initialized with a key similar to a key used in cryptography. This is simply the initial value of a shift register on reset. The security lies in initial value of the key.

Flash memory is an electronic non volatile computer storage that can be electrically removed and reprogrammed as mentioned in Xu et al (2014). There are two major classification of flash memory, according to the type of logic gates used. The NAND type is mainly employed in main memory, memory cards, USB flash drives and solid-state drives, for common storage and transfer of data. The NOR type, which permits true random access and thus direct code execution, is used as a substitute for the older EPROM. PRNG\textsubscript{CHALLENGE}, SSH MAC generation key, AES configuration key and Cipher based MAC (CMAC) values are stored in flash memory. Advantages of flash memory are as follows
- As memory space increases and size decreases, the portability benefits of flash memory devices will only become more evident.

- Flash memory cards are in addition forbearing to magnetism, shock and heat or cold and it is durable.

- Flash memory cards also consume very less power

Unique FPGA device number, master FPGA key, FPGA family type is stored in battery backed memory. The advantage of this type of memory is that, data can be constantly stored for several years exclusively on battery power. Table 5.2 shows the size of different registers.

**Table 5.2 Bit size of Memory contents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit-size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device_Type</td>
<td>128 bit</td>
</tr>
<tr>
<td>Device_DNA</td>
<td>128 bit</td>
</tr>
<tr>
<td>Sec_Flag_Status</td>
<td>8 bit</td>
</tr>
<tr>
<td>PRNG Challenge</td>
<td>32 bit</td>
</tr>
<tr>
<td>INF_Tool_Specific_Information</td>
<td>128 bit</td>
</tr>
<tr>
<td>User_defined_Information</td>
<td>48 bit</td>
</tr>
<tr>
<td>Key*</td>
<td>128 bit</td>
</tr>
<tr>
<td>Count*</td>
<td>32 bit</td>
</tr>
<tr>
<td>Slot_Address</td>
<td>32 bit</td>
</tr>
<tr>
<td>Key</td>
<td>128 bit</td>
</tr>
<tr>
<td>Key SSH_CMAC_GEN</td>
<td>128 bit</td>
</tr>
<tr>
<td>Key Master_FPGA</td>
<td>128 bit</td>
</tr>
<tr>
<td>Key Conf_AES</td>
<td>128 bit</td>
</tr>
</tbody>
</table>
Battery backed memory has an internal lithium power source and a self-contained control circuitry which continually monitors power supply for any out-of-tolerance conditions. If such a condition took plus, the lithium battery will involuntarily switch on and the memory’s write protection is unreservedly enabled in order to prevent data corruption. The numbers of write cycles that can be carried out have no limit and there is no extra support circuitry that is necessary for microprocessor interfacing.

Device DNA is stored in ROM since it is a permanent storage device. Read-only memory (ROM) is a class of storage medium used in computers and other electronic devices. ROM is referred to memory that it uses diode matrix. The additional design keys can be stored in RAM if necessary. A random-access memory device allows data to be read and written roughly in the same amount of time not considering the order in which data are accessed.

5.2.2 SSH Processing Zone

It consist modules for generation of CMAC. Decryption of encrypted AES bitstream, module for generating key update message, FPGA bitstream configuration and SSH process unit.

5.3 SSH OPERATION

The protocol for sending the information from SSH validation unit to SSH in FPGA and also to receive information from SSH to SSH validation unit through synthesis tool is discussed below.
5.3.1 Device Information Module

The module that is used to feed the device information about FPGA to synthesis tool is known as device info protocol. Since all the keys have been empty (i.e. reset) in the target FPGA during the initial stage, there is a necessity for the key update process. The device info protocol sends the device information $\text{DEVICE\_TYPE} || \text{DEVICE\_DNA} || \text{SEC\_FLAG\_STATUS} || \text{PRNGCHALLENGE}$ whenever the synthesis tool asks for device information. On receiving it the synthesis tool append this information with user constraint. $\text{PRNGCHALLENGE}$ is a random number that is created by the pseudo random number generator in SSH process.

5.3.2 AES Decryption

Configuration bitstream for FPGA is encrypted using AES then this block decrypts the information. Advanced encryption standard (AES), is the arrangement to encrypt the electronic data which has fixed block size and a key size of 128 bits. Algorithm utilizes four steps to get the cipher text from the plaintext. It may contain sub bytes, shift rows, mix columns and add round key. AES method is secure against the brute force attack Arora (2012) because the AES key length will determine the feasibility of such attack. The brute force attack involves, charitable all probable combinations until the correct key is found.

5.3.3 Key Update Module

Flowchart of key update process is shown in the Figure 5.5. Subsequent to a regular interval of time IP designer changes the keys which are stored in SSH secure zone. If system developer wants to change the key,
SSH validation unit sends the following “KEY_UPDATE_MESSAGE” to the SSH through Synthesis tool.

![Flowchart](image)

**Figure 5.5 Flowchart for key update process**
KEY_UPDATE_MESSAGE=ENC_KEY|DEVICE_TYPE||DEVICE_DNA||PRNG_CHALLENGE||KEY*||COUNT||INFO_SYN_TOOL_SPEC ||SLOT_ADDRESS

KEY* - New key which is to be stored in SSH secure zone

On receiving KEY_UPDATE_MESSAGE, SSH decrypt it with the unique key. It checks initially the device type. If it is equal then it checks the device DNA. If it is equal then goes for checking security flag status. If it is also equal than it checks PRNG challenge value. If it is equal then checks synthesis tool specification. Next checks the quality of COUNT value. If all the values are equal then key updating process succeeds. If all conditions are satisfied particular key is modified or else, it displays Key update failure message. If a key update message succeeds then the original key is replaced by key*.

5.3.4 FPGA Configuration

Normally the FPGA configuration bit is encrypted by means of 128bit AES algorithm. By AES key in a secure zone of SSH, the input configuration file might be decrypted. The FPGA configuration is explained in Figure 5.6 CMAC generation protocol and configuration protocol play a significant role to organize the bitstream file in a secure manner. When the output from the synthesis tool is loaded into the FPGA, SSH Process reads the bitstream and SSH header. SSH header contains Header ID, Start position and CRC header. The SSH process completes CRC verification. Header ID is checked to initiate FPGA in a secure manner or in an insecure manner. Two different values are stored in header ID to distinguish secure start and unsecured start. If header ID denotes secure initiated then using KEY_SSH_MAC_GEN, start position and End position in SSH header, the CMAC
generation unit generates a CMAC. The SSH process compares \( \text{CMAC}_{\text{SSH}} \) which is stored in a secure zone with calculated CMAC. If both CMACs are equal then secure start is allowed.

**Figure 5.6 FPGA Configuration**
In cryptography, CMAC is a block cipher-based message authentication code algorithm. It is used to provide assurance of the authenticity and, hence, the integrity of binary data. Cyclic Redundancy Check (CRC) ensures the validity of place and routed bitstream.

5.4 RESULTS AND DISCUSSION

The hardware support required by the obtained solution should be implemented in the static logic in FPGA by FPGA vendors to permit system developer to implement it. Therefore, in this research work, AES, HMAC-SHA-512 has been considered for the performance evaluation. FPGA Vendors have to slightly change their bitstream generation tools to add SSH header in addition to necessary support to create the update command.

Recent FPGA boards in market uses AES algorithm for IP core design’s bitstream downloaded into FPGA. In this section, results of Virtex 2 FPGA implementation of all the designs are presented in Table 5.3. The parameters used to evaluate the performance of the proposed FPGA implantation is shown below

- Clock Frequency
- Resource Utilization
- Throughput
Table 5.3 Virtex 2 FPGA Implementation Results Comparison

<table>
<thead>
<tr>
<th>Resources</th>
<th>HMAC-SHA-512</th>
<th>AES</th>
<th>AES_SSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occupied CLB Slices</td>
<td>3000</td>
<td>3250</td>
<td>3500</td>
</tr>
<tr>
<td>Occupied LUTs</td>
<td>4250</td>
<td>4500</td>
<td>4750</td>
</tr>
<tr>
<td>Occupied FFs</td>
<td>3600</td>
<td>3800</td>
<td>4200</td>
</tr>
<tr>
<td>Maximum Clock Frequency (MHz)</td>
<td>69.65</td>
<td>71</td>
<td>75.36</td>
</tr>
</tbody>
</table>

Clock Frequency

The clock frequency is calculated by following formula

\[
\text{Maximum Clock Frequency} = \frac{1}{\text{Min Clock Period}}
\]

Figure 5.7 shows the maximum clock frequency comparison between the different FPGA implemented techniques. It is observed from the figure that the proposed AES_SSH approach performs better than the other two approaches taken for consideration.

![Figure 5.7 Comparison of Maximum Clock Frequency for FPGA Implementations](image-url)
**Throughput analysis**

Throughput is the amount of data processed per clock cycle. The units of throughput are Gbps.

Throughput = (Block Size/No. of Clock Cycles) * Clock Frequency

Figure 5.8 shows the throughput comparison between the different FPGA implementation techniques. It is observed from the figure that the proposed AES_SSH approach provides high throughput of about 0.9 Gbps where the other two techniques namely HMAC-SHA-512 and AES provides 0.4 Gbps and 0.6 Gbps respectively.

![Figure 5.8 Comparison of Throughput for FPGA Implementations](image)

**Resource Utilization**

Figure 5.9 shows that the comparison of the resource utilization capability of the proposed approach with the existing approach, it is observed
that the proposed approach performs better than the existing approach in terms of resources utilization.

![Comparison of Resource Utilization for FPGA Implementations](image)

**Figure 5.9 Comparison of Resource Utilization for FPGA Implementations**

### 5.5 CONCLUSION

In this chapter, the IP core protection of FPGA bitstream is focused mainly. As we have discussed earlier the traditional protection algorithms have problems such as lack of security, throughput, latency and poor performance. To overcome these issues, a novel hardware based FPGA IP core protection is proposed in this chapter. The key updation module and encryption algorithms in the proposed algorithm are used to improve the throughput, latency of the protection technique. Since AES is used in proposed technique the proposed technique is not subjected to attacks such as non brute force. AES used in proposed hardware is not only assures security but also improves the performance of FPGA based IP cores. Without the knowledge of IP core developer the unauthorized person cannot able to download the bitstream into their FPGA hardware.