Trusted Design in FPGAs

Mark Tehranipoor

Introduction to Hardware Security & Trust
University of Florida
Outline

- Intro to FPGA Architecture
- FPGA Overview
- Manufacturing Flow
- FPGA Security
  - Attacks
  - Defenses
  - Current Research
- Conclusion
FPGA Architectures

- Field Programmable Gate Array
  - Configurability
    - Configurable logic block (CLB)
  - Reconfigurable interconnects
  - I/O
    - Similar to ASIC
  - HDL
CLB

4-input Look-Up Table

e.g., SRAM

D Flip-Flop

Inputs

Clock

Output

in1 in2 in3 in4

out

out
CLB Wiring

Wire Segment

Programmable Switch
Why FPGAs?

- Time to Market
- Cost Reduction
- Reliability
- Programmability
- High performance designs
  - Speed
  - Power consumption
  - Package size
Common FPGA Applications

- High Performance Computing
- Medical Equipment
- Data Servers
- Consumer Electronics
- Computer Networking
- Aerospace and Defense
- Etc.
Traditional Manufacturing Flow

- Design exposed throughout process
  - Vendor
  - Foundry
  - Assembly
FPGA Manufacturing Flow

1. Non-Secure Manufacturing
   - Generic FPGA Base Array

2. Secure Design Facility
   - Design
   - Add Secret Bitstream

3. Non-Secure Environment
Production Flow

- System Designer
Programming

- Software
Attacks

- Cloning, Overproducing, Mislabeling
- Reverse Engineering the Bitstream
- Readback
- Side Channels
  - Power Analysis
  - EM Analysis
  - Timing Analysis
  - Ionizing Radiation
- Invasive and Semi-Invasive
- Brute Force, Crippling, Fault Injection
- Relay and Replay
Cloning, Overproducing, Mislabling

- FPGA’s are generic
  - A generated bitstream will work on any device within the respective device family and size
  - Attackers can clone bitstreams
    - Recording in transmission to FPGA
    - Use them in other systems
    - Cheaper clones
Reverse Engineering the Bitstream

- **Bitstream Reversal**: transformation of an encoded bitstream into functionally equivalent description of the original design
Bitstream Reversal

- Partial reversal
  - Extraction of data from bitstream without full functionality
    - BRAM/LUT
    - Memory cell states
    - Keys could be compromised

- Full reversal would divulge the entire design
Readback

- Readback: Process of reading back data from the FPGA device to verify that the design was downloaded properly.
- Retrieving a snapshot of the FPGA's current state while still in operation
  - Configuration
  - LUT
  - Memory contents
- Useful for vendors to verify correct operation
- If enabled, an attacker can add missing header/footer info
  - Use in another device
  - Reprogram FPGA with modified version, Tamper with a Trojan
  - Reverse engineering
  - “Readback Difference Attack”
Defensive usage
- Providing evidence of tampering
  - Ionizing radiation attack

Xilinx provides a bitstream bit to disable readback, but is easily found

Altera's devices do not provide readback capabilities
Challenge: isolate internal operations of IC from the environment

- Power Analysis
- EM Analysis
- Timing Analysis
- Ionizing Radiation
Power Analysis

- SPA on Xilinx Virtex FPGA
  - Not practical for most paralleled cryptographic operations

- DPA possible
  - Statistical correlation techniques against AES and DES

- Power analysis attacks could be made harder
  - Equivalent power signatures
Electromagnetic Field Analysis

- Movement of charge
- Used to efficiently inject signal/noise in attacks
- Successful side channel attack to be exploited
Timing Analysis

- Timing attacks are difficult on FPGA
- Off chip for functionality
- Observable via device pins
Ionizing Radiation

- **Single event upsets (SEU, Soft Errors)**
  - Radiation induced errors caused when charged particles lose energy by ionizing the medium through which they pass
  - May cause transient pulse resulting in delay faults
  - Cause memory bit to change state

- Exhaustively irradiating device until desired results are obtained

- Given the number of transistors & devices, this may not be practical
Ionizing Radiation Detection

- FPGA vendors introduced measures to ensure high-reliability
  - CRC or Hamming

- Triple Modular Redundancy

- Chip “scrubbing” to remove block faults from SEU
Flip Chip Packaging
Side Channel: Conclusion

Some challenges an attacker faces with most side channel attacks:

- Familiarity with implementation details
- Isolation of target function
- Obtaining high signal to noise ratio
- Probing BGA packages
- Devices manufactured at 90/65/45nm technologies
Crippling & Fault Injection

- Subvert a system to perform malicious functions or take it off-line

- Reprogramming with or without encryption can take the system down
  - Authentication can solve this issue

- Attempt to force the device to execute an incorrect operation, or be left in a compromising state
  - Altering input clock or voltage
Relay Attack

• Loaded bitstream uses an authentication protocol to communicate to a chip nearby in which case they share a key. This is meant to prevent the bitstream from being used on another system.

• **Relay attacks** allow an adversary to impersonate a participant during an authentication protocol
Replay

- Attacker resends recorded protocol transaction data
  - ex. impersonation of a participant in authentication protocol

- Cloning of bitstreams is the simplest form
A **replay attack** is a form of attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary who intercepts the data and re-transmits it.
Replay

Authenticate Yourself!

Password
Defenses

- Bitstream Encryption
  - Key Storage
  - Key Management

- Theft Deterrents
  - PUFs
  - DRM
Bitstream Encryption

- Encrypt bitstream at end of design flow
- Decrypt it on the FPGA
  - Cloning
  - Reverse Engineering
  - Tampering
- Bitstream produced
  - Software requests key
  - Encryption
- User 'programs' same key into FPGA
- Bitstream is downloaded, directed through decryption circuitry
Key Storage

- Keys must be present inside the device to decrypt

- Two storage devices
  - Volatile
    - SRAM
  - Non-volatile
    - Fuses
    - Flash
    - EEPROM
  - PUF
Key Management

- Encryption
  - Xilinx: Triple DES, AES 256
  - Altera:
    - Stratus II : AES 128
    - Stratus III: volatile & non volatile, AES 256

- If encryption is used:
  - Disable readback & partial configuration
Key Management

Establishing Value

- Simple: One key
  - Catastrophe if compromised
- More secure: One key per device
  - Very costly
  - If compromised, single stream is affected
  - Database of keys is threat
Design Theft Deterrents

- Vendors offer a few cloning deterrents that rely on secrecy of bitstream encoding
  - Xilinx Spartan 3A “Device DNA”
  - Challenge-response schemes
Watermarking and Fingerprinting

- Passive
- Proves ownership
- Fingerprinting is a watermark used to identify specific end users
- Can be inserted:
  - HDL
  - Netlist
  - Bitstream
- Do not prevent theft, but can provide proof in court of fraud
Ongoing Research

- Physically Unclonable Functions
- Bitstream Authentication
- FPGA Digital Rights Management
PUFs

- One-way functions
- Unique identities from physical properties
- PUFs cannot be reversed
- Very active research area

Arbiter PUF
- Uses delay variations within paths
PUFs

- Ring Oscillator
  - Manufacturing creates different oscillation frequencies

- Initial State of SRAM
  - Upon power on, an SRAM cell is more prone to settle at 0, or 1
    - Butterfly PUF
Bitstream Authentication

- Allows two major items:
  - Sender verification
  - Message integrity
- Sometimes considered more important than encryption
- Very complex methods have been devised
- Restrictions for bitstreams and cores from being used in unauthorized devices
  - Pay-per-use
`protect begin_protected
  protect directives and encoded encrypted information
`protect end_protected

Example:

```vhdl
architecture RTL of accelerator is
  `protect begin_protected
    `protect encrypt_agent = "Encryptomatic"
    `protect encrypt_agent_info = "2.3.4a"
    `protect data_keyowner = "ACME IP User"
    `protect data_keyname = "ACME Sim Key"
    `protect data_method = "aes192-cbc"
    `protect encoding=(enctype="base64", line_length=40,
      bytes=4006)
    `protect data_block
      encoded cipher-text
      ...
  `protect end_protected
end architecture RTL;
```
Conclusion

- Security is ongoing. What is secure today, may be trivial to circumvent tomorrow.

- FPGA security (hardware in general) is a relatively new research area which is advancing rapidly.
References

- Steve Trimberger, Trusted design in FPGAs, Proceedings of the 44th annual Design Automation Conference, June 04-08, 2007, San Diego, California
