UNIVERSITY OF ARKANSAS

TruLogic
Trustable Logic Circuit Design

Hardware IP Classification through Weighted Characteristics

Brendan McGeehan, Flora Smith, Thao Le, Hunter Nauman, Jia Di
Department of Computer Science and Computer Engineering,
University of Arkansas, Fayetteville, Arkansas, USA

Introduction

- **Today's integrated circuit (IC) industry**
  - **Economically oriented**
  - **Heavily reliant on incorporation of third-party intellectual property (IP)**
  - **Todays ICs are becoming more and more susceptible to attacks**
  - **Increased security threat**
  - **Increased threat of Trojan insertion**

Hardware Trojan

What is a Hardware Trojan?

- Malicious addition or modification of the circuitry of an integrated circuit
- Inserted Trojans can be very small
- Created by human intelligence
- Consequences that can come from Trojan not being detected:
  - Damaging payloads
  - Leaking secret keys
  - Shutting down sections of hardware
- Can end up in locations where security is vital

Previous Work Towards Detecting Trojans

- **Side-Channel Analysis**
  - Look at naturally occurring emissions
    - Power/timing delays
  - Detect modifications in a circuit by analyzing differences in power/delay
- **Path Delay**
  - Measure differences in how long a signal takes to travel through a specific path
  - Detect modifications in circuits by analyzing differences in delays

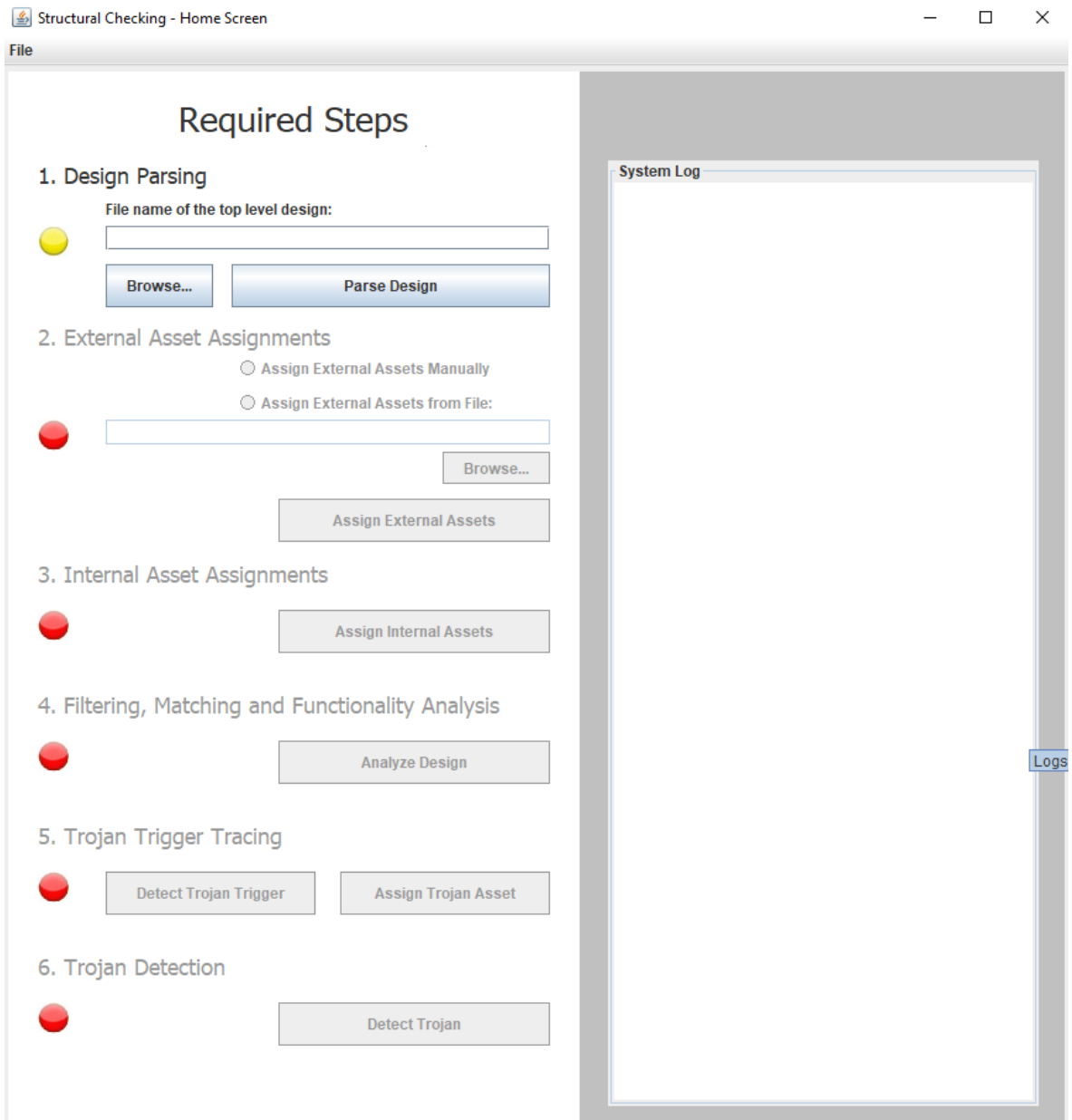
Limitations

- **While both Side-Channel Analysis and Path Delay are viable detection methods, they have their own limitations**
  - **Trojans can be very small**
    - **Do not produce significant emissions**
  - **Mainly focused on detecting Trojans on hard IPs and fabricated chips**

Our Approach

- **Structural Checking (SC)**
  - Analyze Register-Transfer Level (RTL) soft IPs
  - Assign assets to design and create asset patterns
  - Static analysis - no simulation required
  - Designed to be fast and thorough
  - Improved statistical analysis to enhance Trojan detection

Structural Checking Tool GUI



Assets

- **Provide labels to a signal about its purpose/function to the overall IP**
  - **Ex. clock signal would be assigned 'SYSTEM_TIMING' asset**
  - **A signal may have multiple assets**
    - **Help refine how the signal fits within overall design**
  - **There are two main categories of assets within SC tool**
    - **External assets**
    - **Internal assets**

External Assets

- **Used to describe the function/purpose of primary ports of a soft IP**
  - **Must be manually assigned upon first use (only user knows how the IP will be connected to the system)**
  - **Five main categories:**
    - **Data**
    - **Timing**
    - **System Control**
    - **Specific System Control**
    - **Miscellaneous**
  - **The SC tool currently has 58 external assets available**

External Asset Examples

- **Data**
  - DATA_MEMORY: signal that transfers data to or from any type of memory
- **Timing**
  - COUNT: signal that keeps track of a count value
- **System Control**
  - HANDSHAKING: handles any type of handshaking operations
- **Specific System Control**
  - COMMUNICATION_CONTROL: transmission with another component
- **Miscellaneous**
  - ADDRESS_SENSITIVE: connect to memory address of an IP

Internal Assets

- **Intended to describe function of internal signals within a soft IP**
  - **Can also be used for primary port signals**
  - **Most internal assets assigned automatically after SC tool parses the RTL code**
  - **A few assets that deal with scan chains need to be manually assigned**
  - **Examples of internal assets:**
    - **PROCESS_SENSITIVE: signal included in sensitivity list of a process block**
    - **CONDTIONAL_DRIVEN: signal within an “if/case” block**

Asset Filtering

- Allow assets assigned on any primary port signal to propagate through connected signals
  - A set of rules determines whether an asset is copied to its neighbor
  - Create traces for every signal path
- Asset Pattern – compilation of all asset traces of a soft IP
  - Broken down into six characteristics
  - Ex. Characteristic: external assets for primary input port signals

Golden Reference Library (GRL)

- Collection of soft IPs acquired from Trust-Hub, OpenCores, etc., categorized into various functionality groups
- GRL entries are labeled as:
  - Known Trojan-Free (whitelist)
  - Known Trojan-Infested (blacklist)

GRL Example

```
Entity    simple_pic:
    »»    33 port signals
    »»    60 IntraSignals
    »»    4 Port Signal Vectors
    »»    7 Intra-Signal Vectors
    »»    0 SubInstances
    »»    9 Processes
Functionality: INTERRUPT_UNIT
Secondary Func: NON_SEQUENTIAL
Number of Input bits:»  23
Number of Output bits:» 10
>[SYSTEM_TIMING]
>*[PROCESS_SENSITIVE, CONDITIONAL_DRIVING]
>[RESET]
>[INTERRUPT_CONTROL, HANDSHAKING]
>[ADDRESS_SENSITIVE]
>*[CONDITIONAL_DRIVING]
>[INTERRUPT_CONTROL]
>[DATA_SENSITIVE]
>[DATA_SENSITIVE, INTERRUPT]
<[DATA_SENSITIVE, INTERRUPT]
<*[CONDITIONAL_DRIVEN]
<[HANDSHAKING, INTERRUPT_CONTROL]
<*[CONCURRENT_DRIVEN]
<[INTERRUPT, DATA_SENSITIVE]
<*[CONDITIONAL_DRIVEN, PROCESS_OPERATION_SENSITIVE]
>[INTERRUPT, DATA_SENSITIVE]
/[DATA_SENSITIVE]
/*[CONDITIONAL_DRIVEN, PROCESS_OPERATION_SENSITIVE]
/[DATA_SENSITIVE, INTERRUPT]
/*[CONDITIONAL_DRIVEN]
/[HANDSHAKING, INTERRUPT_CONTROL]
/*[CONCURRENT_DRIVEN, CC_OPERATION_AND]
/[INTERRUPT_CONTROL]
/*[CONDITIONAL_DRIVING, CONCURRENT_DRIVEN, CC_OPERATION_AND]
```

Golden Reference Matching

- **Compare unknown soft IP asset pattern against an asset pattern within the GRL**
  - **Algorithm calculates percent match for each GRL entry**
    - **Determine overall functionality of design**
    - **Algorithm chooses best match for soft IP**
- **Basic matching example**

Trace	Unknown IP Assets	GRL Entry Assets	Percent Match
1	DATA_COMMUNICATION	DATA_COMMUNICATION	100%
2	DATA_SENSITIVE, COUNT, STATUS	DATA_SENSITIVE, HANDSHAKING, MEMORY_OP	33%
3	DATA_SENSITIVE	DATA_MEMORY	0%

Statistical Weight

- Overall match calculated by averaging the six percent matches from the six characteristics that make up asset patterns

$$\text{Overall \% Match} = \frac{\sum_{i=A}^{F} \%Match_i}{6}$$

  - Drawback – characteristics do not contribute equal weight
- To improve the algorithm we focus on:
  - Assessing Asset Quantity
  - Assessing Asset Quality

Formulas to Improve Algorithm

- Calculating weight of a given characteristic:

$$P(Asset) = \frac{\sum_{i=1}^{n} GRLEntry_i.contains(Asset)}{Total\ \#\ of\ GRL\ Entires}$$

$$Weight_{Asset} = 1 - P(Asset)$$

$$Average\ Asset\ Weight = \frac{\sum_{i=1}^{n} MatchedAsset_i.weight}{Total\ \#\ Matched\ Assets}$$

- Finally we can calculate the new characteristic weight by combining the formulas from above

$$Weight_{char} = \frac{Characteristic_{char}AverageAssetWeight}{\sum_{i=A}^{F} Characteristic_i AverageAssetWeight} * 100$$

Results

- **Tested IPs include RS232, RSA, AES, and a few microcontrollers**
  - **Statistical algorithm help extract important asset matches**
- **Examples of smaller designs:**
  - **RS232**
    - **Contain denial-of-service attack.**
    - **Both original and improved algorithm correctly match**
  - **AES**
    - **Contain secret key after certain plaintext is read**
    - **Both original and improved algorithm correctly match**

Results (cont.)

- **PIC16F84 – microcontroller obtained from Trust-Hub**
  - **Demonstrated improvement in statistical matching**
  - **Made up of:**
    - **Two types of memory EEPROM and RAM**
    - **Watchdog timer,**
    - **Interrupt ports,**
    - **I/O ports**

Results (cont.)

Asset Assignment

After parsing PIC16F84, assets are assigned to input and output ports

Signal	Assets
clk_i	SYSTEM_TIMING
clk_o	SYSTEM_TIMING
eep_adr_o	ADDRESS_SENSITIVE
eep_dat_i	DATA_MEMORY
eep_dat_o	DATA_MEMORY
existeprom_i	MEMORY_OP
int0_i	INTERRUPT
int4_i	INTERRUPT
int5_i	INTERRUPT
int6_i	INTERRUPT
int7_i	INTERRUPT
mclr_n_i	RESET
pon_rst_n_i	RESET
porta_dir_o	PERIPHERAL_CONTROL
porta_i	DATA_PERIPHERAL
porta_o	DATA_PERIPHERAL
portb_dir_o	PERIPHERAL_CONTROL
portb_i	DATA_PERIPHERAL
portb_o	DATA_PERIPHERAL
powerdown_o	CLOCK_CONTROL
prog_adr_o	ADDRESS_SENSITIVE
prog_dat_i	DATA_MEMORY
ram_adr_o	ADDRESS_SENSITIVE

Results (cont.)

Basic Matching

GRL Entry	Overall Percent Match
Simple_pic	52.553%
Lcd16x2_ctrl	48.233%
Lcd_controller	44.148%
RSACypher_T100	43.414%
Spi_master_1	40.750%

Improved Matching

GRL Entry	Overall Percent Match
Simple_pic	47.149%
Lcd16x2_ctrl	36.591%
Lcd_controller	36.514%
RSACypher_T100	31.785%
Spi_master_1	30.211%

Basic Matching vs. Improved Matching

- After asset assignment, SC tool filters assets to connected signals
- Better matching due to disparity between overall percent match of GRL entries

Results (cont.)

- **MC8051-T500 Core – tested microcontroller known to be Trojan-free**
  - **Also demonstrated improvement in statistical matching**
  - **Made up of:**
    - **Control units for Finite State Machine (FSM) and memory**
    - **ALU**
    - **Serial Interface Unit (SIU)**
    - **Timing Unit**
      - **Also handle interrupt signals**

MC8051-T500 Core Matching Results

	Equal Weight Matching		Statistical Based Matching	
Target IP	Functionality	% Match	Functionality	% Match
MC8051_core	COMMUNICATION	35.321%	INTERRUPT_UNIT	50.899%
MC8051_control	COMPUTATIONAL	44.871%	REGISTER_FILE	54.689%
Control_fsm	COMPUTATIONAL	47.767%	REGISTER_FILE	38.913%
Control_mem	INTERRUPT_UNIT	61.576%	INTERRUPT_UNIT	62.274%
MC8051_alu	COMPUTATIONAL	22.244%	COMPUTATIONAL	29.564%
Alumux	COMPUTATIONAL	55.565%	COMPUTATIONAL	46.519%
Alucore	COMPUTATIONAL	50.297%	COMPUTATIONAL	42.133%
Addsub_core	COMPUTATIONAL	44.250%	COMPUTATIONAL	41.169%
Addsub_cy	COMPUTATIONAL	46.875%	COMPUTATIONAL	44.748%
Addsub_ovcy	COMPUTATIONAL	46.875%	COMPUTATIONAL	44.748%
Comb_mltplr	COMPUTATIONAL	45.833%	COMPUTATIONAL	38.863%
Comb_divider	COMPUTATIONAL	37.500%	COMPUTATIONAL	35.399%
Dcml_adjust	COMPUTATIONAL	31.718%	COMPUTATIONAL	34.492%
MC8051_siu	COMMUNICATION	77.152%	COMMUNICATION	70.793%
MC8051_tmrctr	REGISTER_FILE	52.257%	INTERRUPT_UNIT	48.587%

Conclusion

- **The statistical matching algorithm**
  - **Enhanced matching algorithm for SC tool**
  - **Calculate weights for individual assets**
    - **Tool determines how well an asset matches to a soft IP**
  - **Using weights helps facilitate numerical representation of the six characteristic**
  - **Helps provide a more unique identification for targeted IPs**

Future Work

- In order to improve the SC tool we intend to:
  - Continually grow the GRL to improve matching
  - Add more assets in order to better refine the purpose of each signal within an IP
  - Add more functionalities within GRL to provide more options to classify an unknown IP