Hardware Trojans in Incompletely Specified On-chip Bus Systems

Nicole Fern, Ismail San, Çetin Kaya Koç, and Kwang-Ting (Tim) Cheng

ECE/CS Department – UC Santa Barbara

Problem Statement

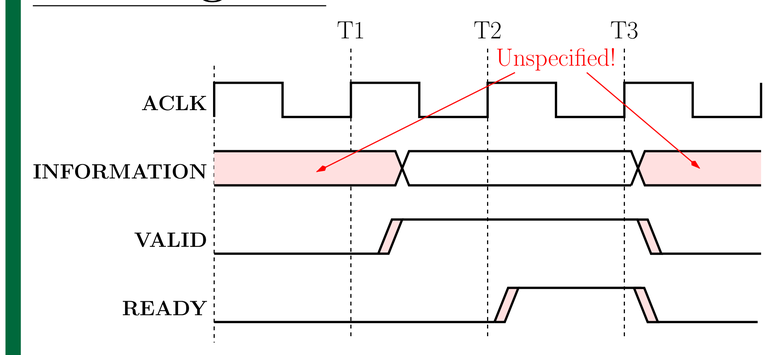
- **Functional** and **secure** on-chip bus networks are critical in modern SoCs
- Most bus protocols only **partially specify signal behavior** meaning some scenarios are never tested/verified
- Extra malicious circuitry can use existing bus signals when they are **unspecified** to create a **covert communication channel**
- **Information leakage** and Trojan communication across an SoC are use cases for the covert channel

Threat Model

- Attacker capable of inserting a **Hardware Trojan** (malicious design modification) in the RTL code
- Attacker creates a **Trojan channel** which does not suppress, alter, or create valid bus transactions, but instead re-uses existing bus protocol signals to define a new "Trojan" bus protocol
- A bus component either leaks data or sends valid data otherwise not visible to the receiver over the Trojan channel

Unspecified Functionality in AMBA AXI Bus Protocol

Data Signals:



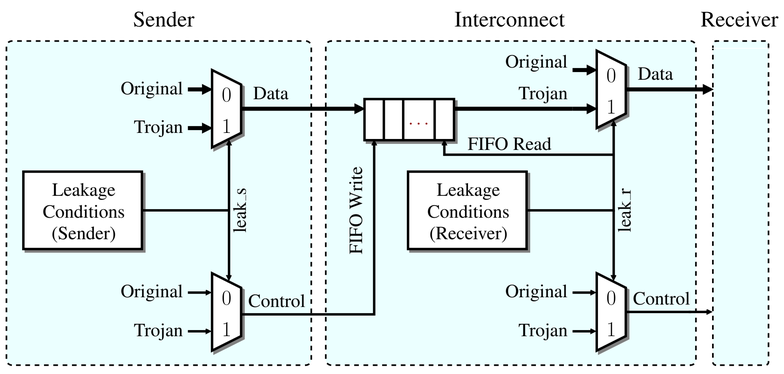
AXI Bus Protocol VALID/READY Handshake [2]. Bus data can be anything (**including Trojan communications**) when VALID is LOW!

Write Strobes: "When WVALID is LOW, **the write strobes can take any value**... " [2]

Trojan Communication Channel

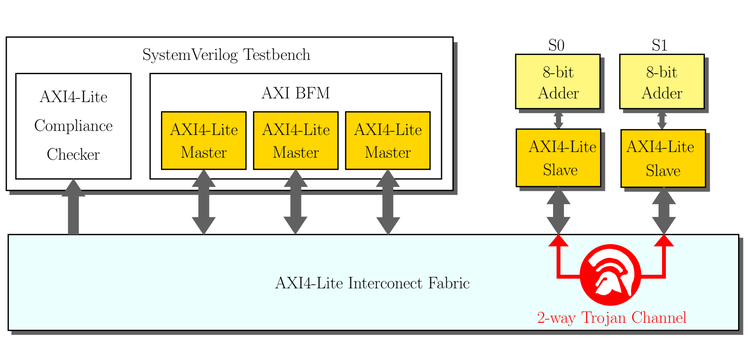
Components necessary to create a Trojan channel:

- **Data:** Existing bus signal(s) chosen to transmit k-bits of Trojan data
- **Control:** Signal(s) chosen to mark Trojan transactions (distinguish from an idle bus)
- **Leakage Conditions Logic:** Monitors bus interface to determine when Trojan data can be sent/received (bus is idle)



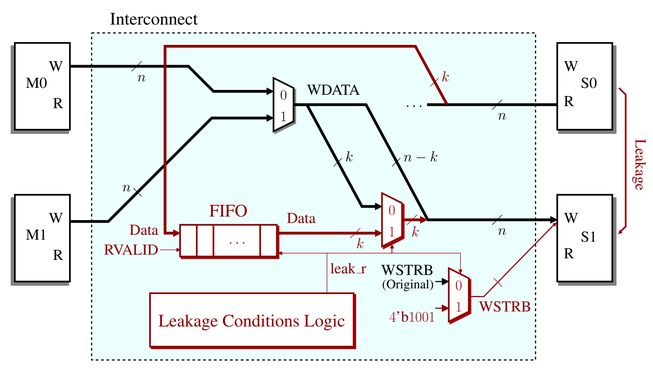
- **FIFO:** Used to buffer Trojan channel data if the receiver bus interface is busy

Case Study: AXI4-Lite System



AXI4-Lite Case Study Verification Infrastructure

- S0 and S1 are adder coprocessors that receive operands from an AXI4-Lite bus
- 2-way Trojan channel inserted to allow S0 to view S1's operands and vice versa
- Assigning WSTRB to 4'b1001 marks valid Trojan data
- Over 50 AXI4-Lite assertions packaged by ARM [1] active during simulation, **none are violated** even when data is flowing through the Trojan channel!
- Interconnect and adders implemented on a Virtex-7 FPGA (7vx330t-3)



Trojan Channel Logic (×2 for 2-way Leakage)

- Trojan channel stays within 3% of the original FF and LUT utilization

Configuration	# FF	# LUT	# BRAM	Frequency [MHz]
3 Masters 2 Slaves	1814	2474	2	250
4 Masters 6 Slaves	3071	4247	3	250

Trojan-Free Design Results (After Place and Route)

Data Width	FIFO Depth	% Increase in FF		% Increase in LUT	
		3M2S	4M6S	3M2S	4M6S
2	2	0.8	0.5	0.9	0.4
	4	1.1	0.7	1.5	0.6
	8	1.4	0.8	1.8	1.1
4	2	1.0	0.6	1.4	0.7
	4	1.3	0.8	2.0	0.8
	8	1.7	1.0	2.0	1.5
8	2	1.4	0.8	1.8	1.0
	4	1.8	1.0	2.4	1.2
	8	2.1	1.2	3.0	1.7

Area Overhead of 2-way HW-Trojan Channel

Key Contributions

1. **General model** for creating a covert channel between Trojan components
2. Information transmitted/leaked by **altering existing bus signals only when they are unspecified** meaning the channel is undetectable by most verification and Trojan detection techniques
3. Demonstrated feasibility and overhead of Trojan channel insertion by creating a Trojan-infested AXI4-Lite System

References

[1] ARM, *AMBA 4 AXI4, AXI4-Lite and AXI4-Stream Protocol Assertions BP063 Release Note (r0p1-00rel0)*, 2012.

[2] ARM, *AMBA AXI and ACE Protocol Specification*, 2013.

Acknowledgements

This work was supported by NSF/SRC STARSS (1526695). We would also like to thank the Xilinx University Program for their generous donation of several FPGA boards.

Contact Information

Web: http://cadlab.ece.ucsb.edu
http://www.nicolefern.com

Email: nicole@ece.ucsb.edu