Trusted and Secure Integrated Circuits and Systems

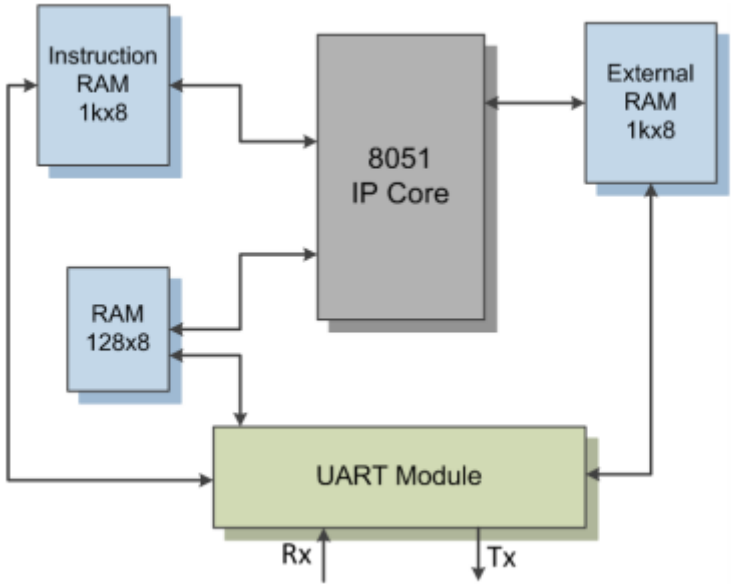
(EEDG7V81)

Hardware Trojan Design in 8051 Microprocessor

Introduction

In this project, we present three process-level hardware Trojan designs. Certain malicious modifications in the micro-processor, with some overhead, is exploited by the software codes which make an impact on the whole system.

The Block Diagram of the 8051 micro-processor with UART channel.



The 8051 IP core is given in VHDL language and comes with test benches, simulation models and sample code.

The implementation of 8051 on a FPGA, the Xilinx CORE generator system is used to generate three memory modules:

- Internal RAM- 8x128 bits data width
- External RAM - 8x1024 bits data with
- Program RAM – 8x1024 bits data width

A DCM module is inserted to the design to decrease the speed from 100MHz to 10MHz.

Description:

Three process-level hardware Trojan design is introduced. The instructions set plays an important role to control the functionality of the processor. Two hardware Trojans targeting instruction memory to demonstrate that any malicious modifications in the instruction memory cause an impact to the processor and the third Trojan is execution of malicious code.

TROJAN 1:

Malicious Instruction Insertion.

This Trojan contains a malicious ADDC instruction. The PSW (Program Status Word) contains status bits that reflect the current CPU state. The carry bit (CY) of the Program Status Word (PSW) is set to 1 in the program. Hence when ADDC operation is executed the result adds the carry bit by default, even when carry is not generated by actual addition of the two numbers. This will tend to an erroneous result.

	7	6	5	4	3	2	1	0
PSW (0D0)H	CY	AC	F0	RS1	RS0	OV		P

TROJAN 2:

Unused instruction used as LCALL.

The operation code of LCALL is '00010010'. This operation code is modified to '11110010' which is an unused operation code. Hence when malicious LCALL is executed, the LCALL operation does not perform its actual operation and the result is not what is expected.

TROJAN 3:

Modification of instruction in PUSH:

In the PUSH instruction, which is used for stacking, there is an instruction s_regs_wr_en <= "001". The function of this instruction is increasing the stack pointer when the processor stacks data into next memory position. This Trojan changes the instruction to s_regs_wr_en<="010", which is write accumulator (ACC). Thus, break the stacking process.

Strength:

TROJAN 1:

Malicious Instruction Insertion:

The carry bit will be cleaned up when the ADDC operation is done. So it is hard to find what is wrong by just checking memory or any register or the instruction dump, after the code has been executed.

TROJAN 2:

Unused instruction used as LCALL:

The modification of the operation code of the instruction cannot be detected just by looking at the malicious output. Hence it is difficult to find if it is a malicious LCALL instruction.

TROJAN 3:

Modification of instruction in PUSH:

The file containing the malicious modification also includes many other instructions. Moreover, the modification looks very similar to the original format. Thus, the Trojan is considered covert. Also the Trojan cannot be detected just by looking at the instruction dump.

Limitation:

TROJAN 1:

Malicious Instruction Insertion:

The result is erroneous. The Trojan can be detected by seeing the assemble program along with the simulation results.

TROJAN 2:

Unused instruction used as LCALL:

The operation code of the malicious LCALL instruction can be detected by looking at the instruction dump and Trojan can be found.

TROJAN 3:

Modification of instruction in PUSH:

The stacking process will be broken down and data in memory will be changed. The Trojan will be found if look up every step in the PUSH instruction.

Usage Evaluation:

The area overhead and performance impact resulting after the introducing the hardware Trojans is tabulated below.

Trojan Type	Slice Registers	Slice LUTs	Average Fan-out
Genuine	**551**	**2689**	**5.19**
Trojan 1	549	2661	5.42
Trojan 2	551	2667	5.19
Trojan 3	552	2670	5.3

References:

[1] Yier Jin, Michail Maniatakos and Yiorgos Makris, “Exposing Vulnerabilities of Untrusted Computing Platforms”, *Computer Design (ICCD), 2012 IEEE 30th International Conference.*