AUTOMATED KEY BASED LOCKING OF GATE LEVEL DESIGN & VERIFICATION

Created by Md. Alvir Islam Nobel

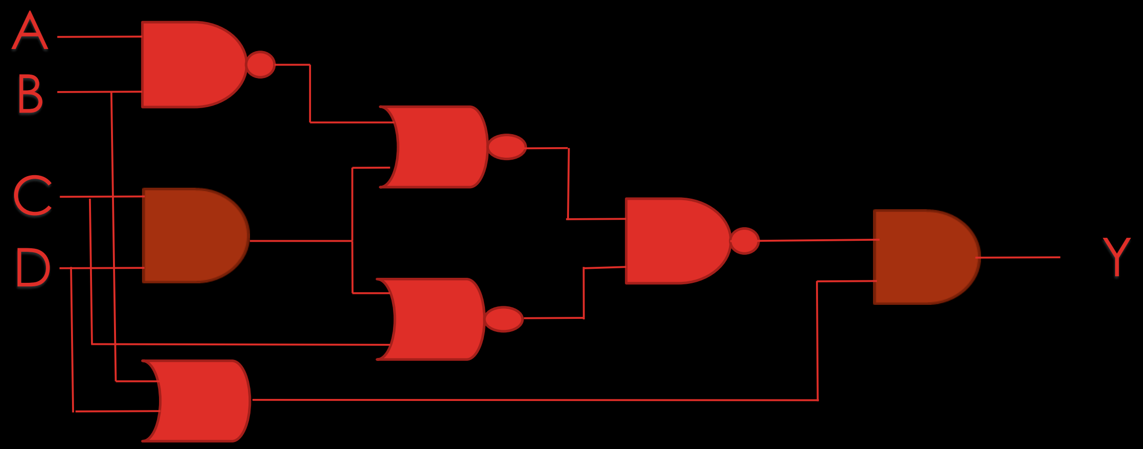
HARDWARE OBFUSCATION

- It is the technique of hiding the functional behavior of the hardware
- Introduces ambiguity in structural and functional behavior to protect hardware IP form piracy

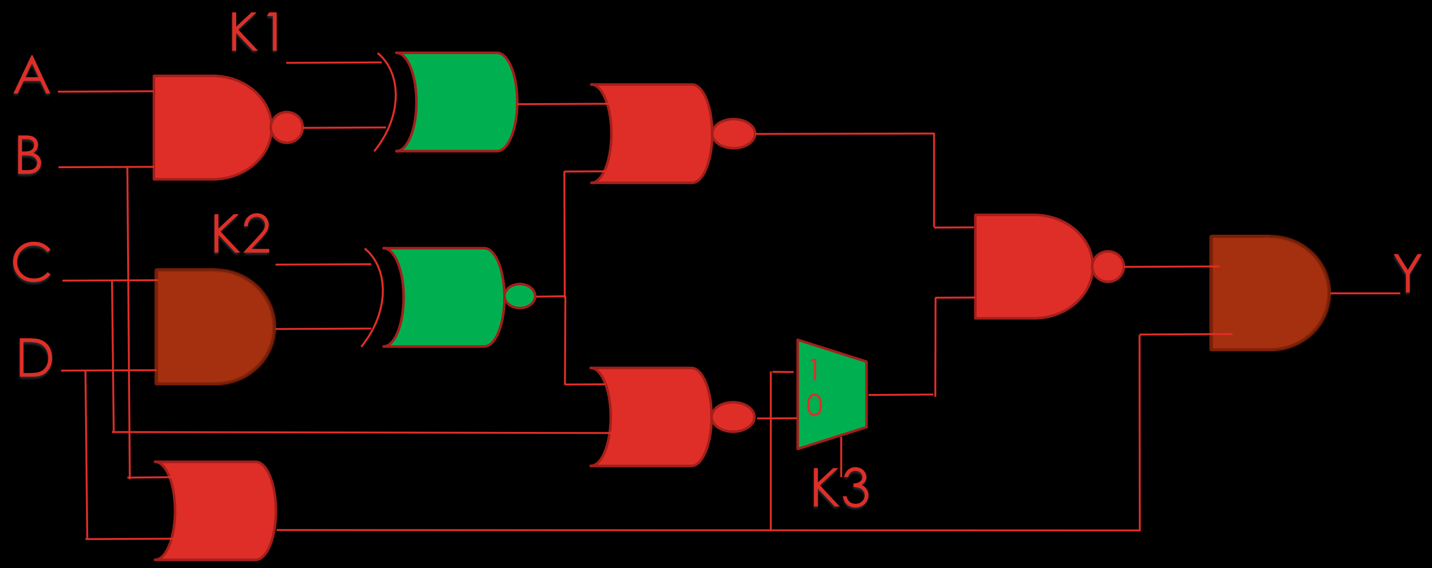
GATE-LEVEL DESIGN OBFUSCATION

- The process of including additional gates for functional locking & structural modification of the design
- Also known as "Logic Locking"
- Combinational locking is a good locking process & requires "Key Gates" which are XOR, XNOR, MUX
- The design functionality can be fully achieved only if the Key Combination of the Key Gates are known

ORIGINAL GATE LEVEL DESIGN



DESIGN AFTER KEY GATE LOCKING



The gates in green color are Key Gates. The Key Gates with K1,K2,K3 inputs are XOR, XNOR & MUX respectively

SCRIPTING

- Python Script has been used to execute the locking process from the Verilog Code
- The script can analyze the design & initiate the indicated locks

SIMULATION

- The Simulation of the Gate-Level Design has been executed in Xilinx Vivado
- The Truth Table of the design is provided in the next page
- The Key Gate Unlock Combination is (K1,K2,K3)=(0,1,0)
- Simulation results for the wrong Key combinations are also available

A	B	C	D	Y	K1 K2 K3 0 0 0	K1 K2 K3 0 0 1	K1 K2 K3 0 1 0	K1 K2 K3 0 1 1	K1 K2 K3 1 0 0	K1 K2 K3 1 0 1	K1 K2 K3 1 1 0	K1 K2 K3 1 1 1
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	0	1	1
0	1	0	0	1	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	1	1	1	0	1
0	1	1	0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1	1	0	0
1	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	1	1	0	1	1	1	1	1
1	1	0	1	0	1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	0	1	1	1	1	1	1

REFERENCE

- “ Hardware Obfuscation and Logic Locking : A Tutorial Introduction “ by Tamzidul Hoque, University of Florida & Rajat Subhra Chakraborty ; IEEE Design & Test, Volume 37, Issue:3