200K

- Polar fire FPGA /
  - implement scan ~~de~~ chain / single DES on fpga
  - how 2 design scan chain
  - how 2 attack
  - scan chain for DES ⟷ practical attack
  - complex adder

DR CHEF DID NOT SEEM IMPRESSED W MY PRELIMINARY PRESENTATION :(

78 complex multiplier in DOTP

steps

① FIGURE OUT HOW TO GET COMPLEX MULTIPLIER IMPLEMENTED ON FPGA — FIGURE OUT HOW TO INSTALL LIBERO

② FIGURE OUT SCAN CHAIN INSERTION + ACTIVATION → scan flip flops!

③ FIGURE OUT HOW TO IMPLEMENT A standard DES attack

22/01/2024 AIMS

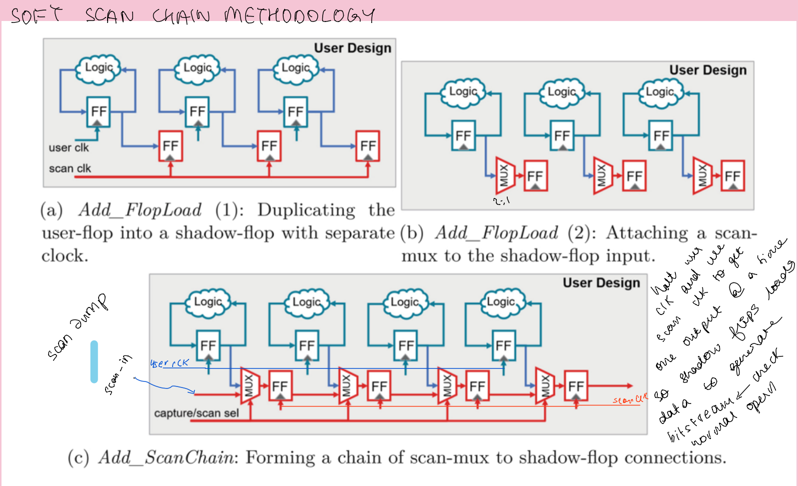
- Finish reading 4 research papers
- ~~setup for Xilinx Kintex 7 series on laptop~~
- start simulating simple verilog / VHDL circuits to learn these :)

Xilinx Kintex 7, 70K

scan chain | use case - cybersecurity - attack

look @ emulation version

SOFT SCAN CHAIN METHODOLOGY



(a) *Add_FlopLoad* (1): Duplicating the user-flop into a shadow-flop with separate clock.

(b) *Add_FlopLoad* (2): Attaching a scan-mux to the shadow-flop input.

(c) *Add_ScanChain*: Forming a chain of scan-mux to shadow-flop connections.

SW:

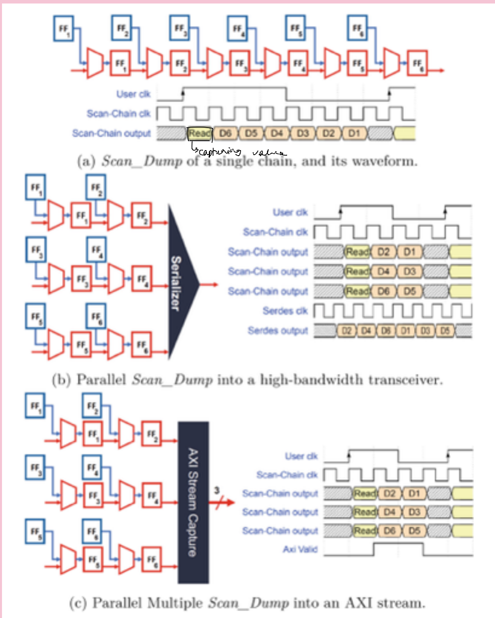
libero } → SPI ba

AMD Xilinx → Vivado limited edition (7-series)

Verilog / VHDL → schematic
↳ simple design

① simple design + latch test
↓
② Replicate on FPGA } → the AMD Xilinx FPGA

③ Develop attack



(a) *Scan_Dump* of a single chain, and its waveform.

(b) Parallel *Scan_Dump* into a high-bandwidth transceiver.

(c) Parallel Multiple *Scan_Dump* into an AXI stream.

COW 2

→ summarize learning pts + presentation

→ ~~getting started~~ → play w simple gates (NOT, NAND, OR) — start building simple circuits on vivado to figure out behaviours

→ define technical Requirement
↳ generate bitstream
↳ record bitstream

→ start emulating simpler scan chain

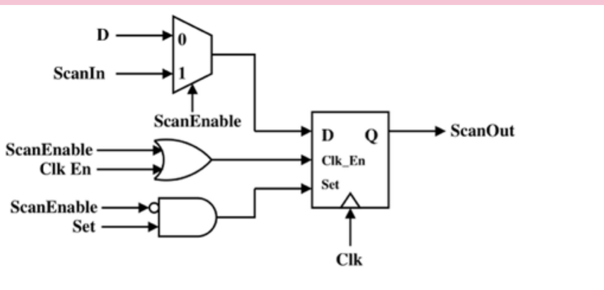


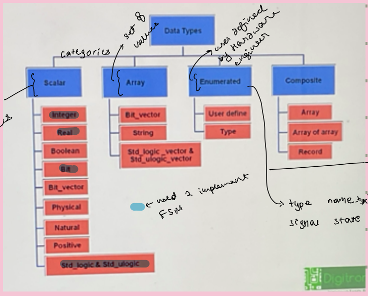
Fig. 1. Instrumenting a Flip-Flop for Scan

VHDL data types and operators

integar, bit, std_logic, std_logic_vector

IEEE 1076 library ← defined in here

use ieee . . . . . ← defines data types



→ type state is : idle, forward, backward, stop

↳ 2 bits will encode data type value

↳ idle is default state

→ type name_type is (A, B, C, D) ;

signal state : name_type := A

bit → 0/1

bit vector → group of multibit signal → bus

SIGNAL x : BIT ; x ⇐ 1

SIGNAL y : BIT_VECTOR (3 DOWNTO 0); y ⇐ 0111

SIGNAL w : BIT_VECTOR (0 TO 7); w ⇐ 01010101

↑ signal assignment operator

Boolean

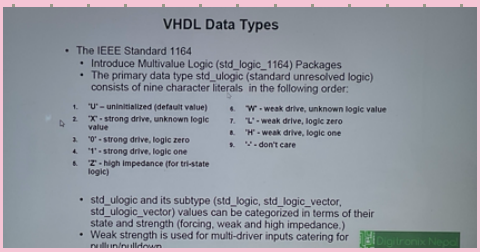
BOOLEAN (TRUE, FALSE)

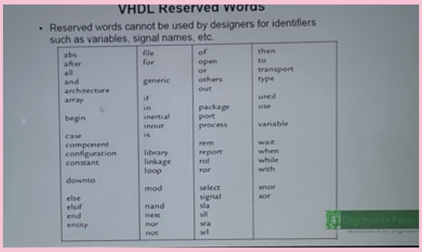
variable VAR1 : boolean := FALSE

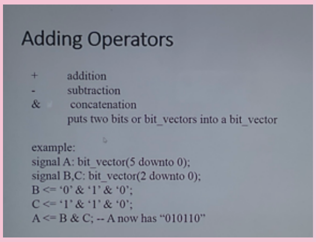
Integar (32 bit, -2147483647 to 2147483647)

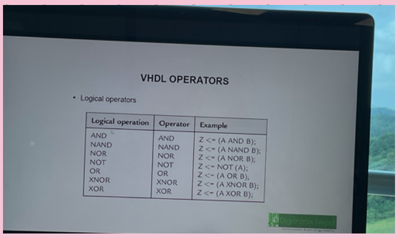
Real (-1.0×10^{38} to 1.0×10^{38})

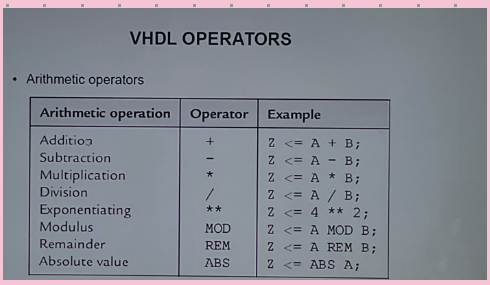
↳ decimal

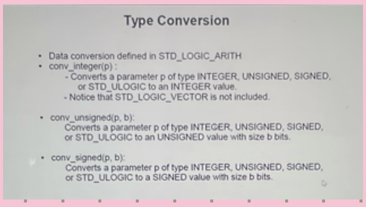












TO DO:-

- summarize learning pts from lit review
- define technical requirements
  - which circuit :
    - AXI slim
    - FPGA PINS
    - how many inputs
- start simulating circuit
- add scan chain