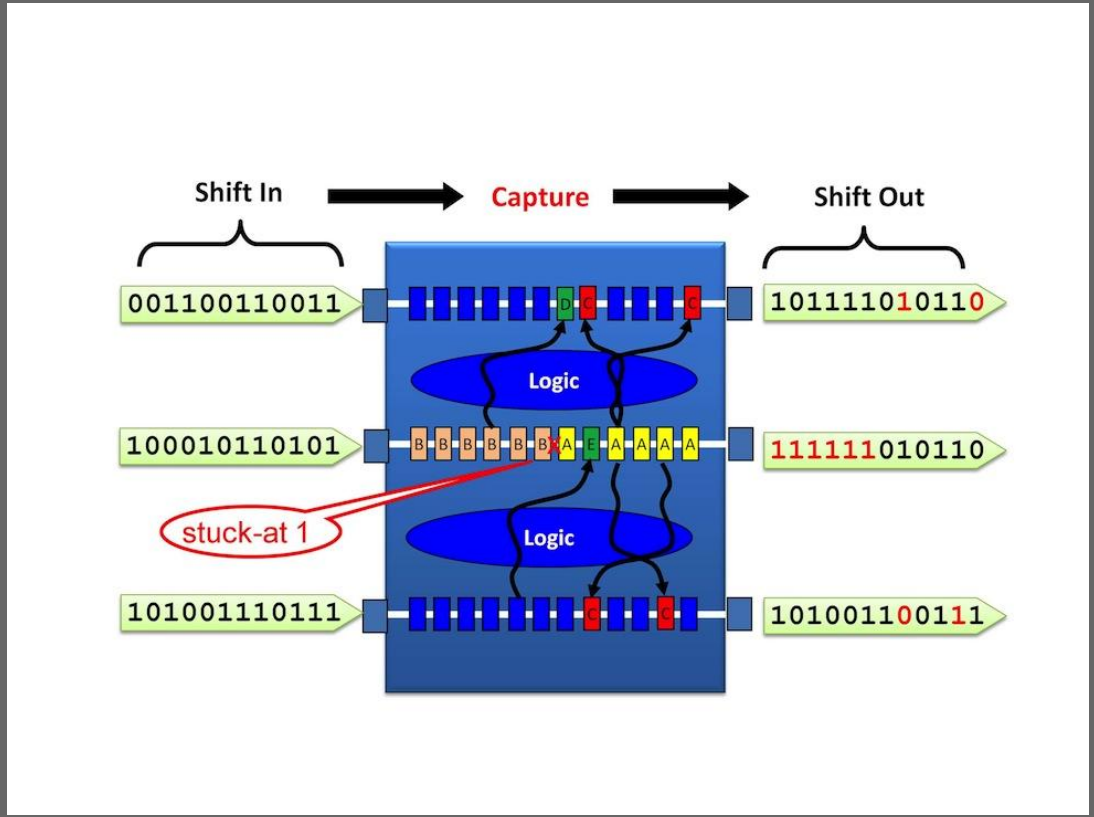
CPE 470 - Design For Test



Glossary

DFT: Design for Test

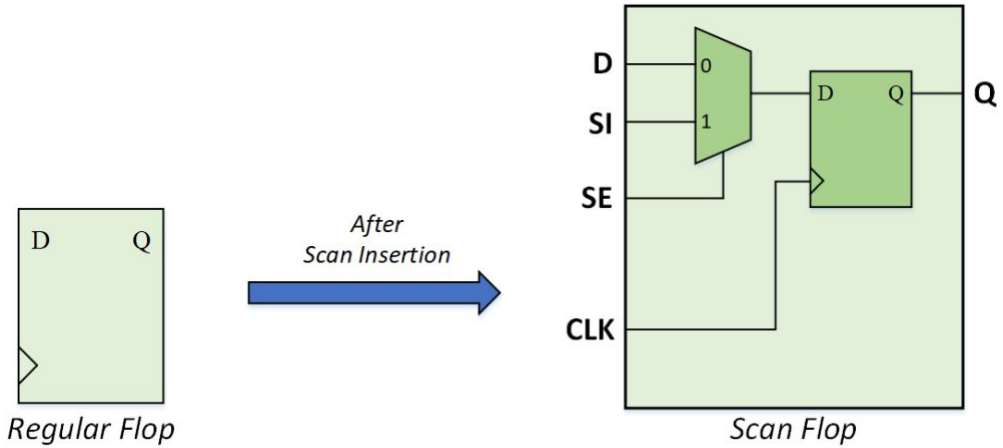
DFT

- Bringing Up Chips after Fabrication is hard
  - Billions of transistors each will failure risk
- Can be really hard to diagnose issues without view into the system

- **DFT**: methodology or design principles to ensure design is testable at the post fabrication stage
  - Same acronym, but not a Discrete Fourier Transform
- Build system with ways of debugging its internal pieces
  - Trade off logic area for higher ability to diagnose issues
- Build configuration features in
  - Ability to disable or power off certains parts of chip if they don't work

Scan Chain Topology

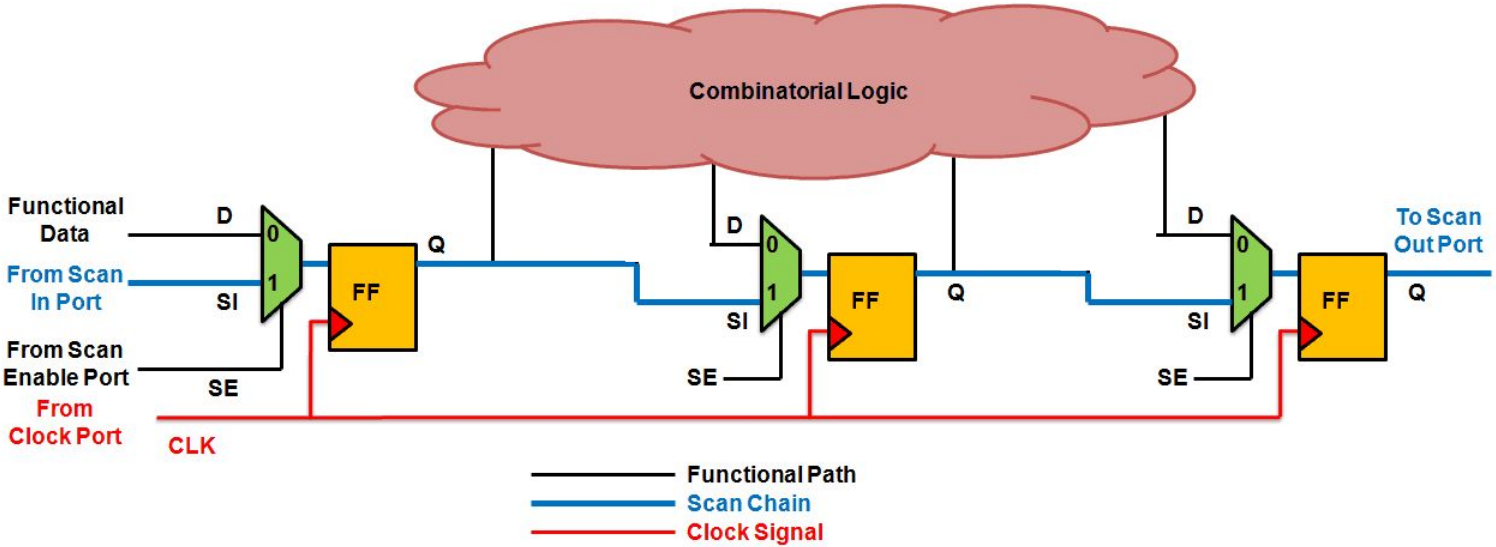
- Problem: How can we most easily debug every element in our design?
- Solution: Conditionally connect EVERY flip flop in a serial shift register
  - How do we do this?
- Replace every Register with a Scan Chain Register
  - Register + Mux that selects between normal operation and scan mode
  - When Scan Enable is active, every flip flop in design becomes a shift register



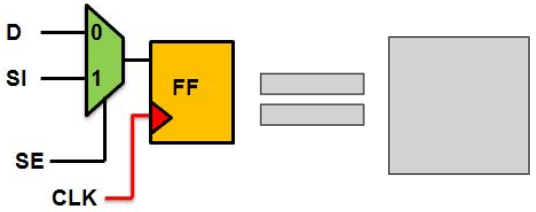
- Use Scan Chain to read or write values to each flip flop

Scan Chain Topology

- All Flip-Flop Outputs feed into combinational logic AND next Flip Flop in chain
- Combinational logic could be many gates!
  - Can’t expose the output of every logic gate, only ends of combination paths
- Data shifted in on Scan In and out on Scan Out
  - Usually Scan Chain also has some way to single-step the clock

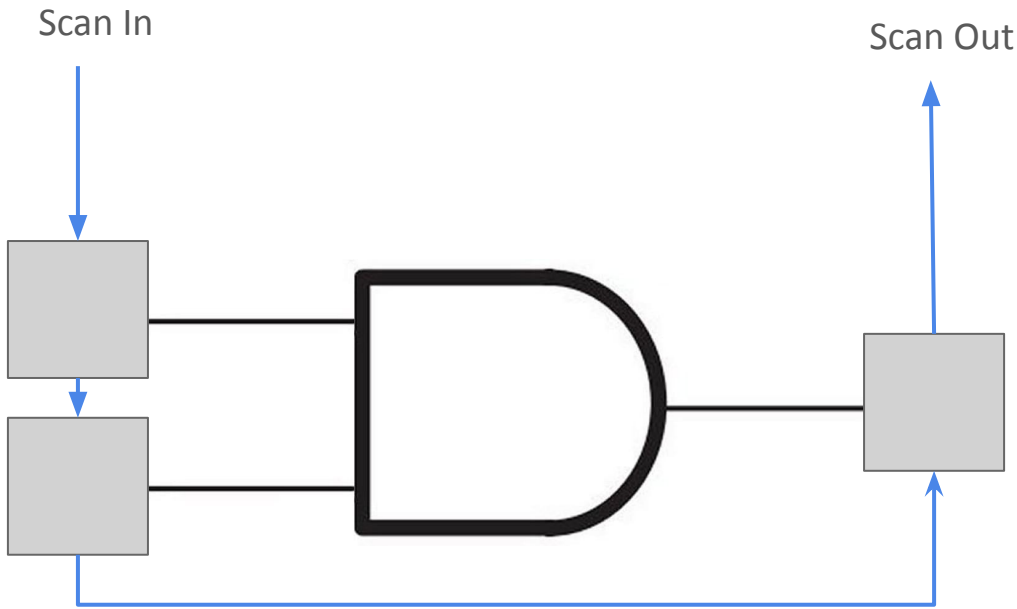


Scan Chain Example

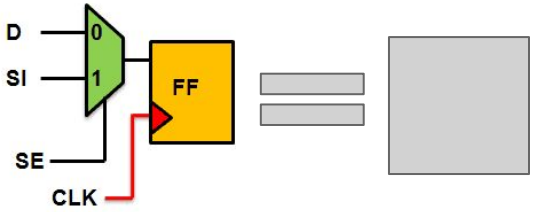


Example: Test that all flip flops work

- Set Scan Enable
- **Shift in 000**
- Shift Out and check equal
- **Shift in 111**
- Shift Out and check equal
- Lower Scan Enable

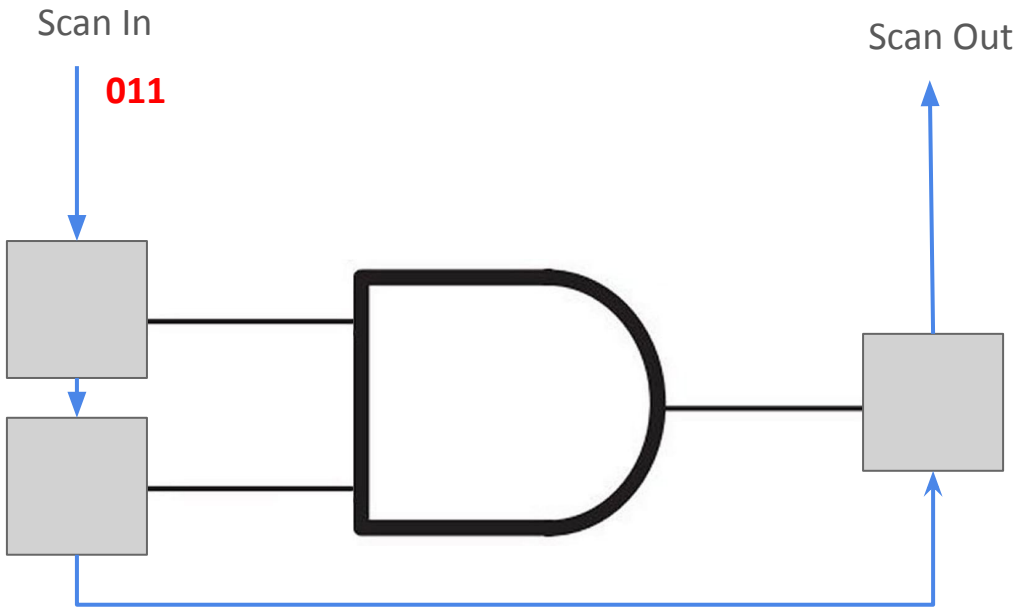


Scan Chain Example

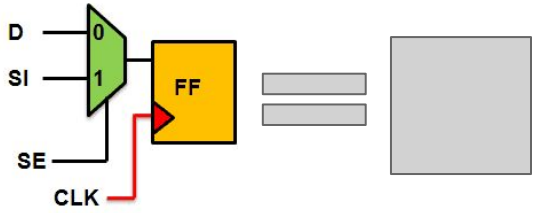


Example: Test And Gate

- Enable Scan
- **Shift in 011 (MSB)**
- Disable Scan
- Step the clock
- Enable Scan
- Check for **1** on scan out

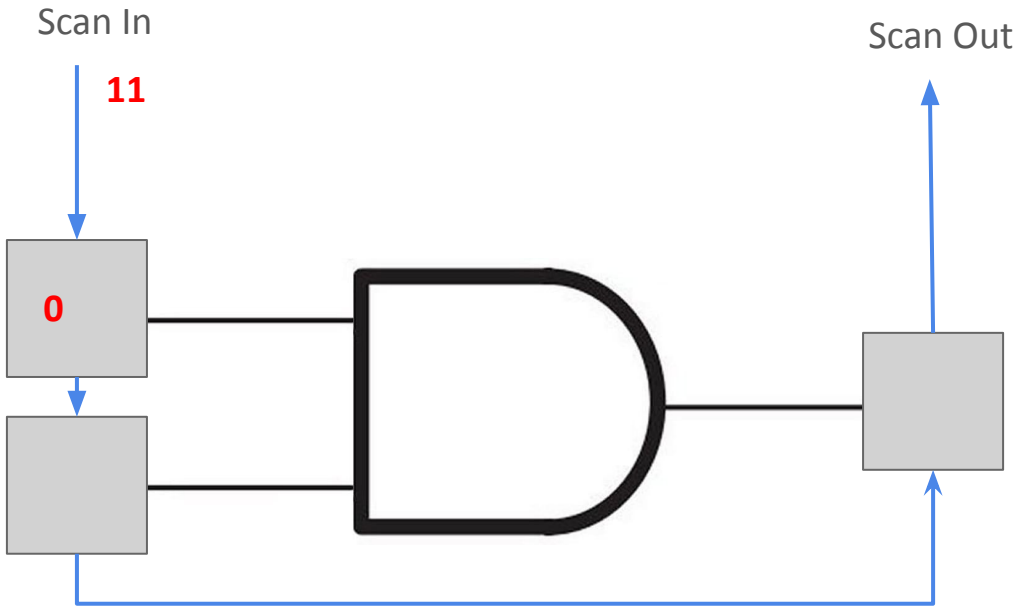


Scan Chain Example

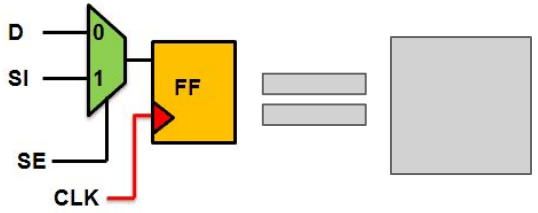


Example: Test And Gate

- Enable Scan
- **Shift in 0**11 **(MSB)**
- Disable Scan
- Step the clock
- Enable Scan
- Check for **1** on scan out

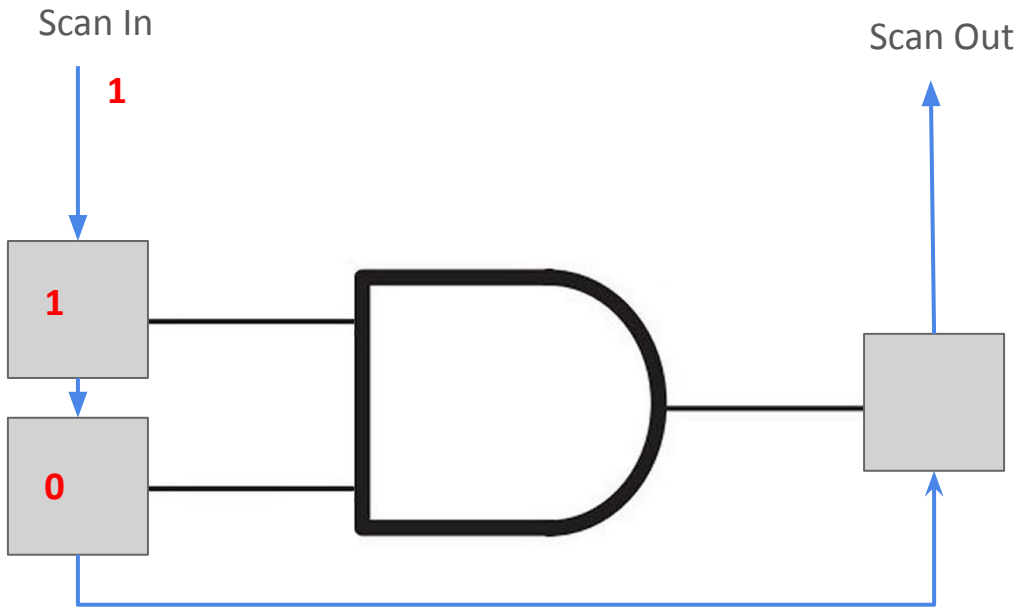


Scan Chain Example

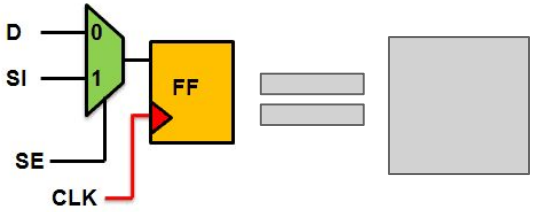


Example: Test And Gate

- Enable Scan
- **Shift in** 011 **(MSB)**
- Disable Scan
- Step the clock
- Enable Scan
- Check for **1** on scan out

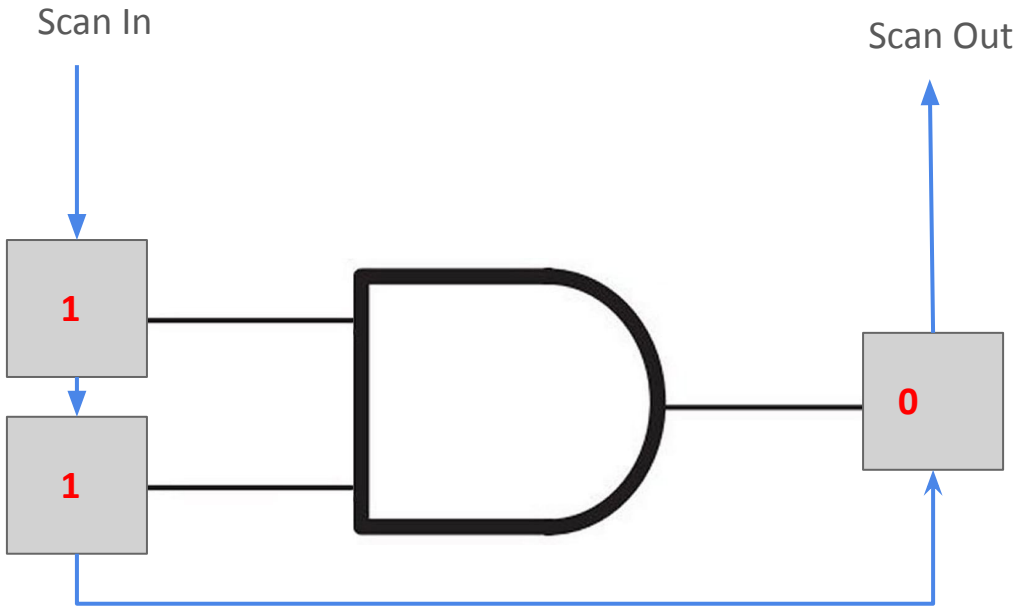


Scan Chain Example

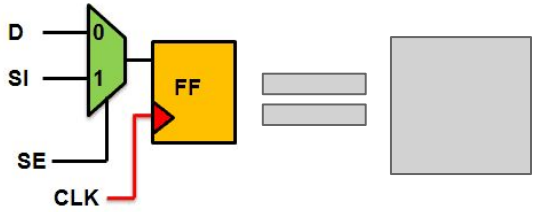


Example: Test And Gate

- Enable Scan
- **Shift in** 011 **(MSB)**
- Disable Scan
- Step the clock
- Enable Scan
- Check for **1** on scan out

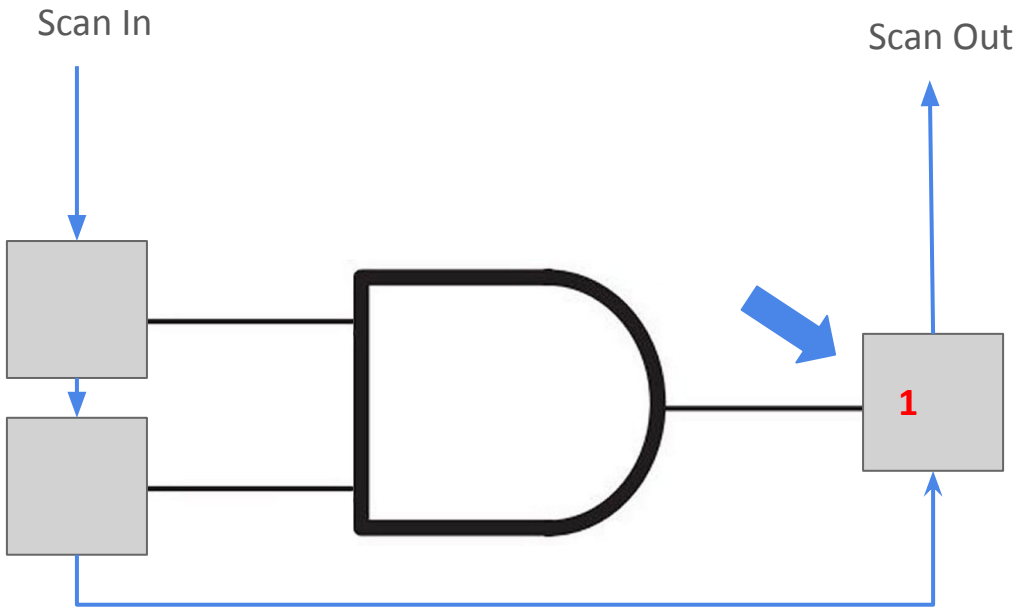


Scan Chain Example

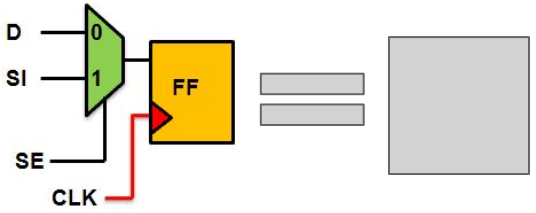


Example: Test And Gate

- Enable Scan
- Shift in 011 (MSB)
- Disable Scan
- **Step the clock**
- Enable Scan
- Check for 1 on scan out

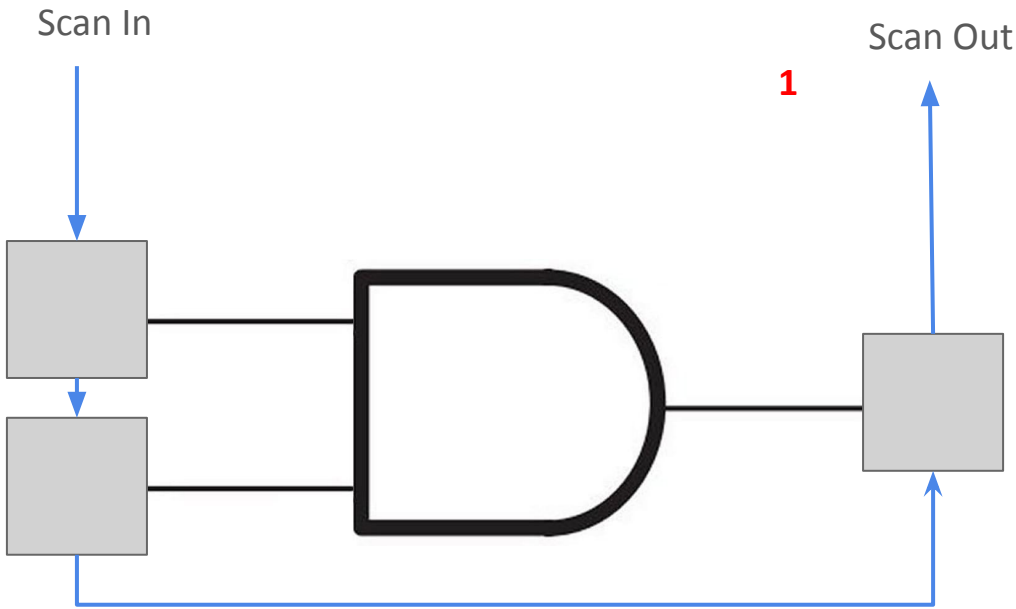


Scan Chain Example

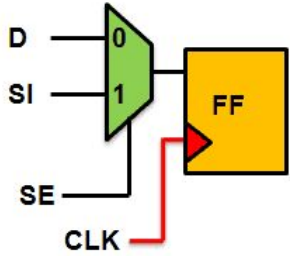


Example: Test And Gate

- Enable Scan
- Shift in 011 (MSB)
- Disable Scan
- Step the clock
- Enable Scan
- **Check for 1 on scan out**



Scan Chain Overhead



- Overhead: Add 1 mux and 2 wires (SI, SE) to every single flip flop
  - This is not cheap!
  - Shift registers are prone to hold violations → Need extra buffers to fix
- Usually adds between 1 and 10% maximum area
- Makes routing more congested

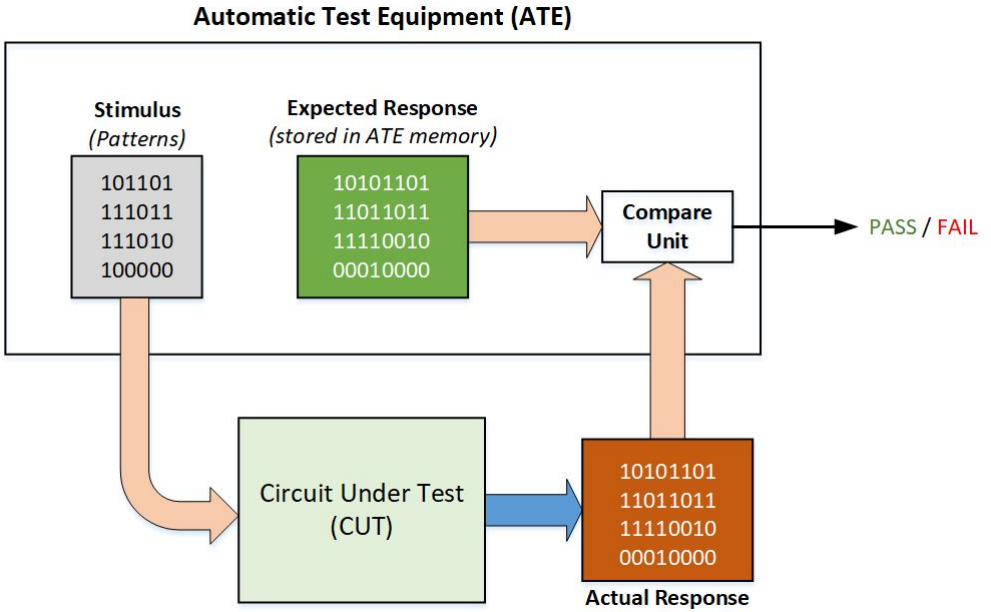
Circuit	Original RTL (before scan insertion)	Total Area		Area Overhead	
		RTL scan	*GL scan*	*RTL scan*	*GL scan*
FIR	3219.67	3341.72	3506.59	5.12 %	8.91 %
IIR	8038.32	8321.10	8442.74	3.52 %	5.03 %

ATPG

Glossary

ATPG: Automatic Test Pattern Generation
ATE: Automatic Test Equipment

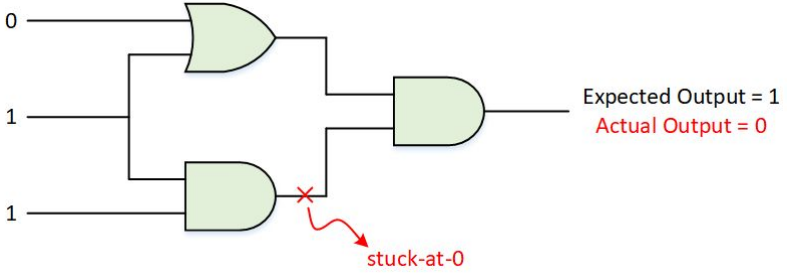
- **ATPG** algorithmically generates test patterns to find faults
  - Hundreds Millions of flip flops
  - 2 states each
    - cannot just test every possible state
  - Must make smart decisions about what to test
  - SAT solvers used to find optimal tests



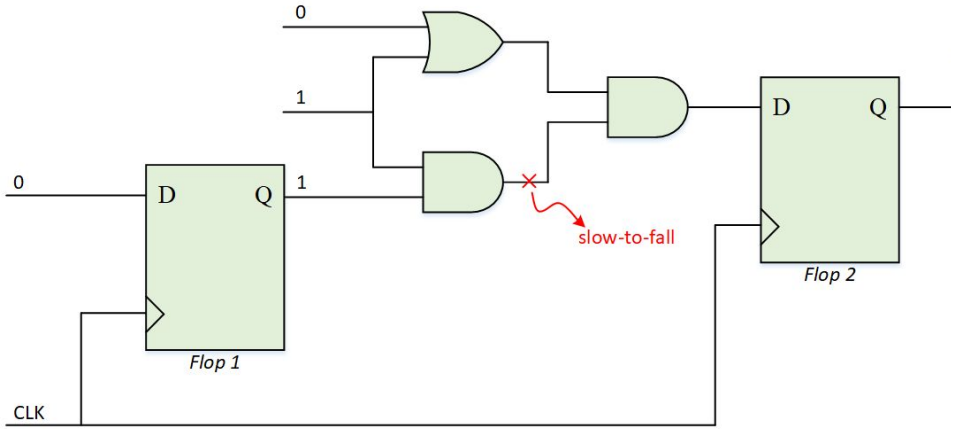
Bring-Up Testing:

- Use **ATE** to inject patterns at high speed
  - Physical hardware connected to physical chip

Fault Types



- **Stuck-At** Faults: gate is shorted high or low
  - Gate always outputs 1 or 0
  - Will appear correct sometimes
  - Need to test cases such that both outputs are tested
    - Ex: Stuck-At-1 Nand Gate appears correct ¾ of the time
- **At-Speed** Faults: only show up in certain situations, frequency dependent
  - Slow-To-Rise: only shows up on transition from 0 to 1
  - Slow-To-Fall: only on 1 to 0
- At-Speed faults require tests spanning multiple clock cycles, to force a transition

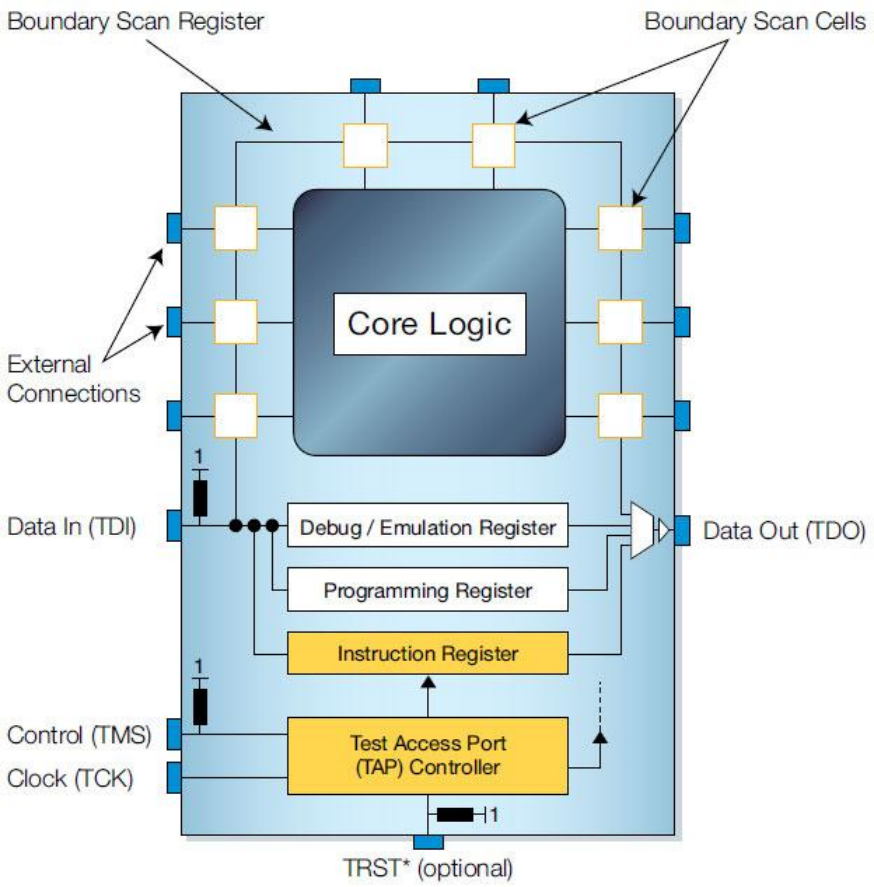


JTAG

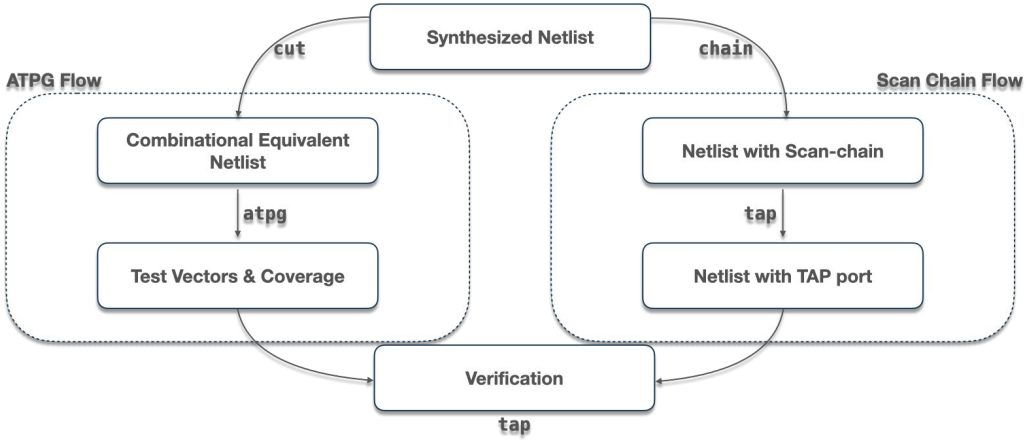
Glossary

JTAG: Joint Test Action Group
TAP: Test Access Port

- **JTAG** introduces industry standard way of verifying ICs and PCBs after manufacture
  - Find issues with PCB to IC connections
  - Defines standardized interface
- **Boundary Scan**: scan chain only connected to each IO Pins, exposed over JTAG
  - Enables reading each IO pin automatically
- **TAP** has a small FSM that:
  - takes instructions
  - exposes the boundary scan
  - runs tests



Fault DFT



- Open Source DFT toolchain
  - Scanchain and ATPG
- Written in Swift
  - Same language as IOS apps
- Developed by American University in Cairo's Open Source Hardware Lab
  - Same group behind DFFRAM

- Cuts each flip flop out of netlist, adds mux and scan enable
- Integration with Openlane still in progress

References

- https://anysilicon.com/overview-and-dynamics-of-scan-testing/
- https://www.researchgate.net/publication/375688367_ASIC_Implementation_for_Multiple_Scan_at_Register_Transfer_Level
- https://ieeexplore.ieee.org/document/9324799
- https://www.xjtag.com/about-jtag/what-is-jtag/
- https://vlsitutorials.com/dft-scan-and-atpg/