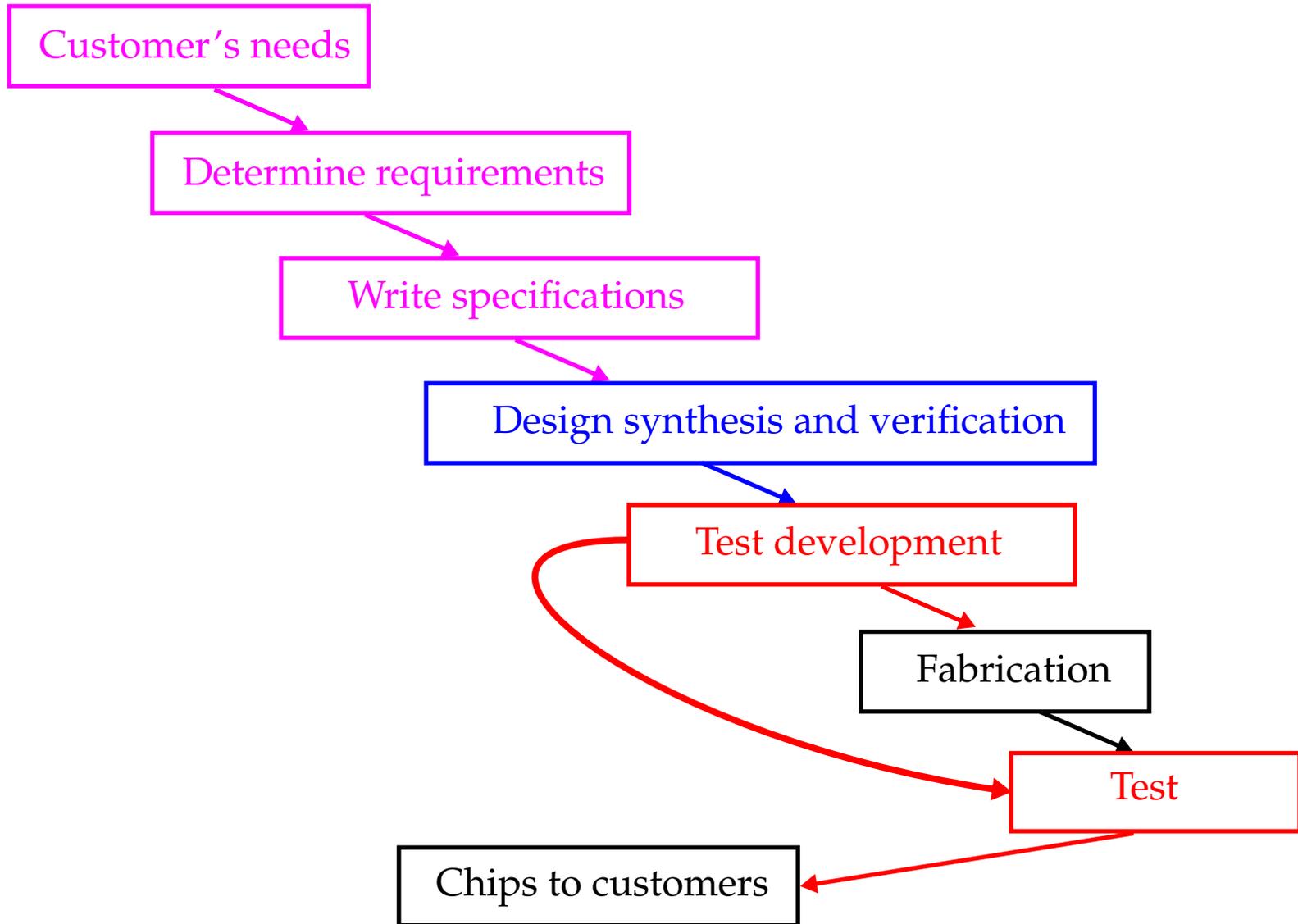


### Big Picture



## Design Verification vs. Manufacturing Test

- **Design Verification:** Predictive analysis to ensure that the synthesized design, when manufactured, will perform the given I/O function
- **Test:** A process that ensures that the physical device, manufactured from the synthesized design, has no manufacturing defects

### Verification

- \* Verifies correctness of design.
- \* Performed by simulation, hardware emulation or formal methods.
- \* Performed "once" prior to manufacturing.

### Test

- \* Verifies correctness of hardware.
- \* Two-parts:
  - Test generation: software process executed "once" during design.
  - Test application: electrical tests applied to hardware.
- \* Test application performed on EVERY manufactured device.

## Ideal vs Real Tests

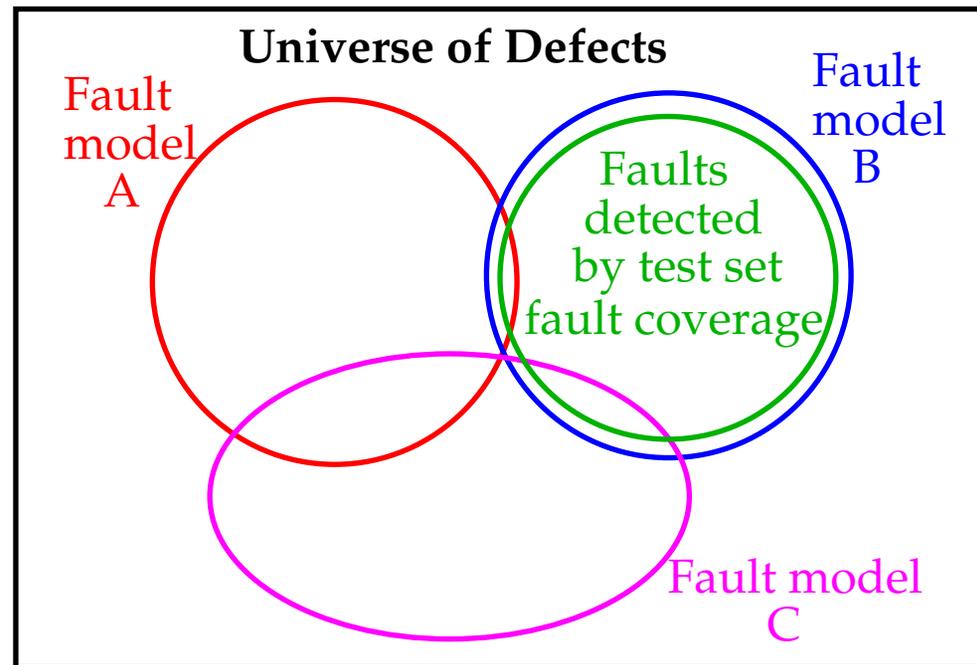
*Ideal tests* detect **all defects** produced in a manufacturing process

Pass all functionally good chips, fail all defective chips

Very large numbers and varieties of possible defects need to be tested

Difficult to generate tests for some real defects -- **defect-based testing** is an active research area

**Ideal tests can detect all defects in this universe**



## Ideal vs Real Tests

Fault models may not map onto real defects

A **fault** is a *logic level abstraction* of a **physical defect** that is used to describe the change in the logic function of a device caused by the defect.

It is difficult to generate tests that detect every possible fault in the chip due to high design complexity

Some good chips are rejected

The fraction of such chips is called **yield loss**

Some bad chips are shipped

The fraction of bad chips among all passing chips is called **defect level** (test escapes)

Benefits of Testing:

Quality and economy: Quality means satisfying the user's need at a minimum cost

**Roles of Testing**

*Detection: Go/no-go, is the chip manufactured properly?*

*Diagnosis: A process to determine where, in the IC, the failure is occurring.*

*Performed on chips that fail go/no-go tests*

*Failure Analysis (FA): A process to determine the specific manufacturing process steps that are producing the defects*

*Performance Characterization: For speed binning parts*

*Process characterization: A process designed to help with yield learning*

*Design for Manufacturability: A process for establishing which design rules and guidelines are best to improve an IC's yield*

*Trust and Security?: Will test "get stuck with" hardware security and trust*

## Components of Test

*Design for Testability (DFT):* On-chip components added to make test easier

Scan-chains

BIST

*Software processes associated with test:*

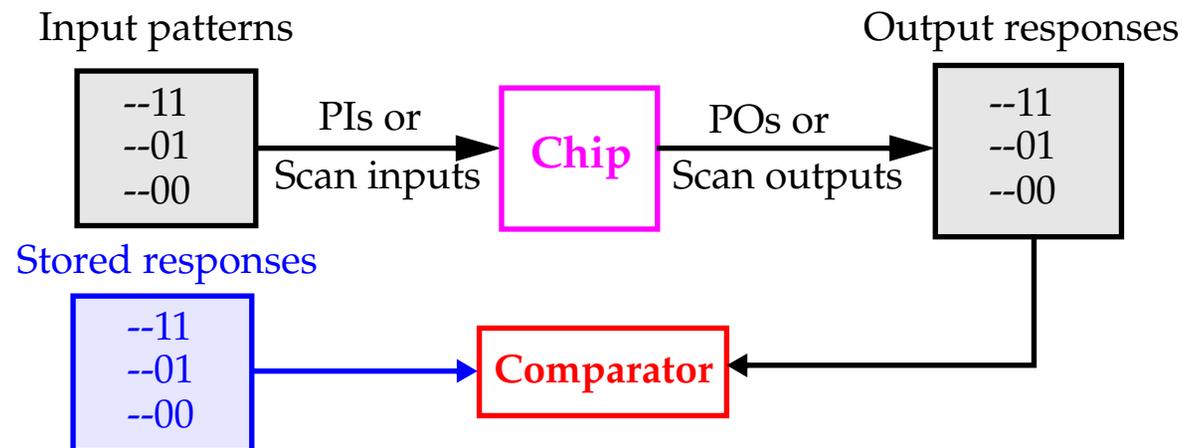
Automatic test pattern generation (ATPG)

Fault simulation

Automatic test equipment (ATE) programming and debug

*Manufacturing test:*

Application of test vectors by ATE

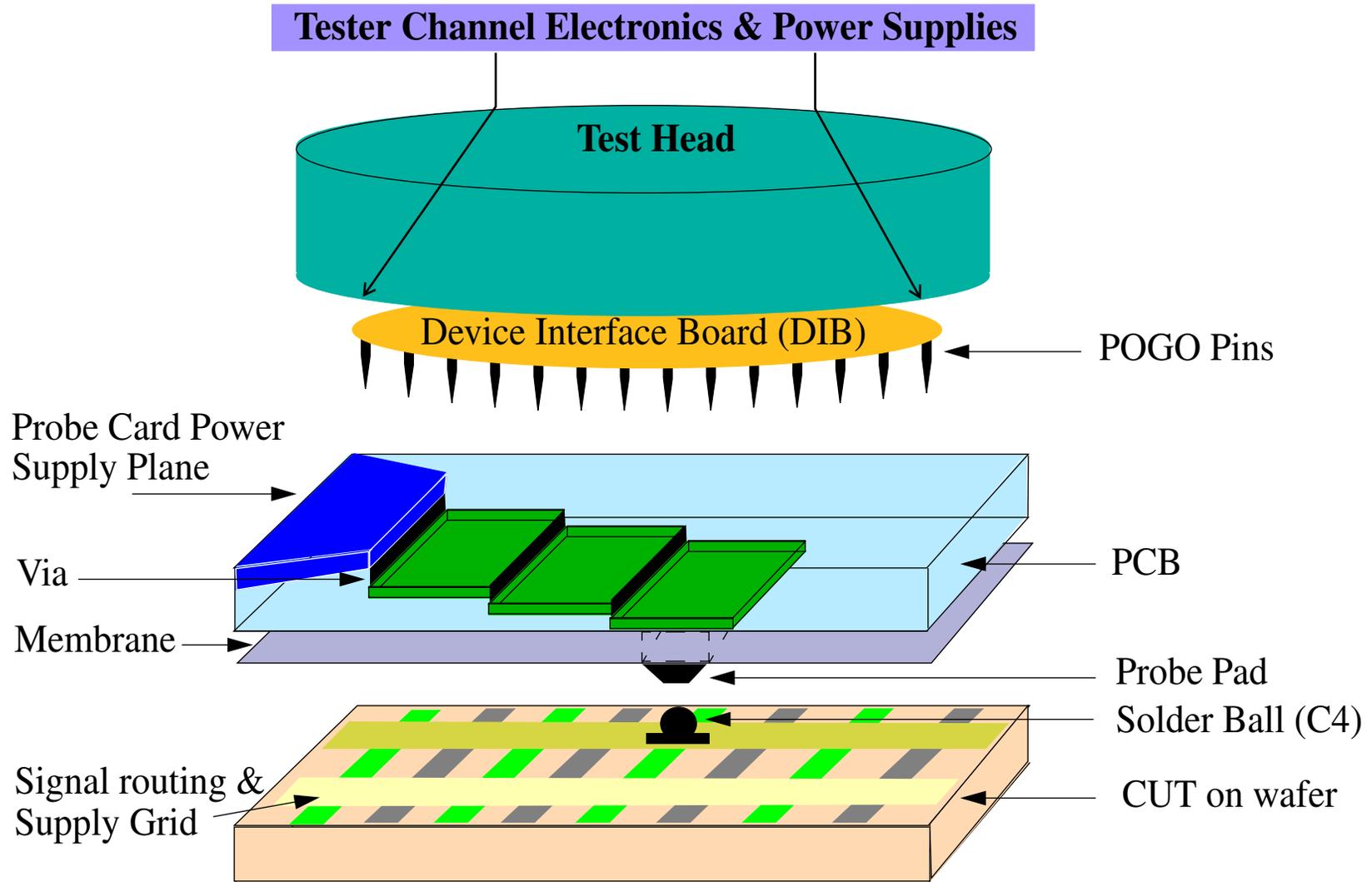


**ATE for Manufacturing Test**

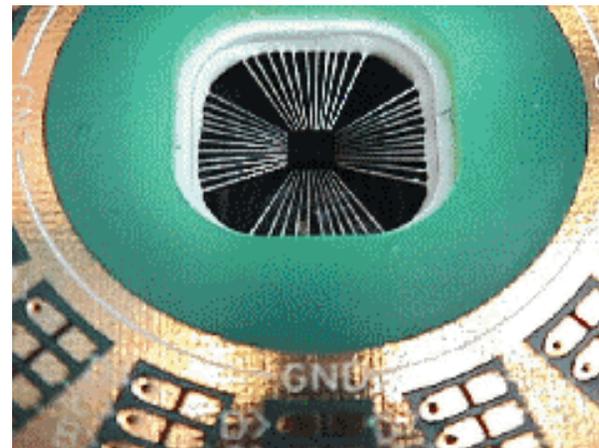
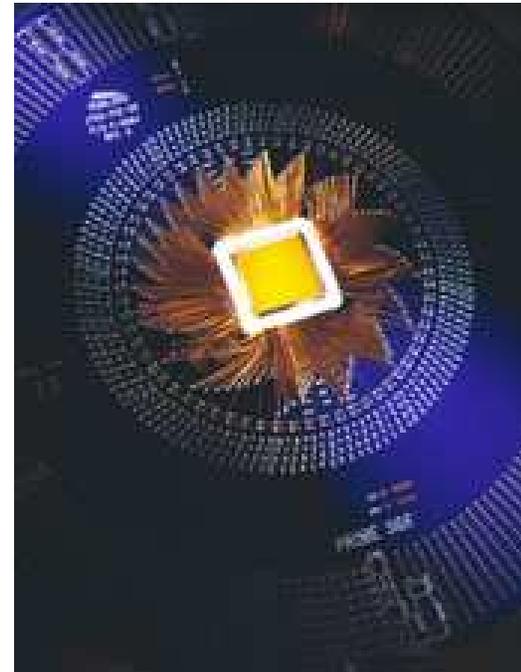
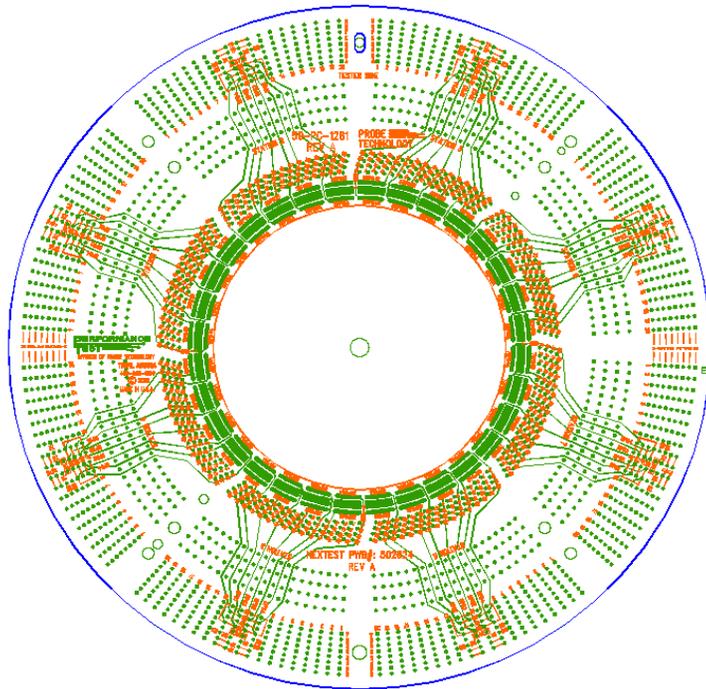


### Wafer Probe Physical Model

Test head and membrane (cobra) probe card for probing C4s



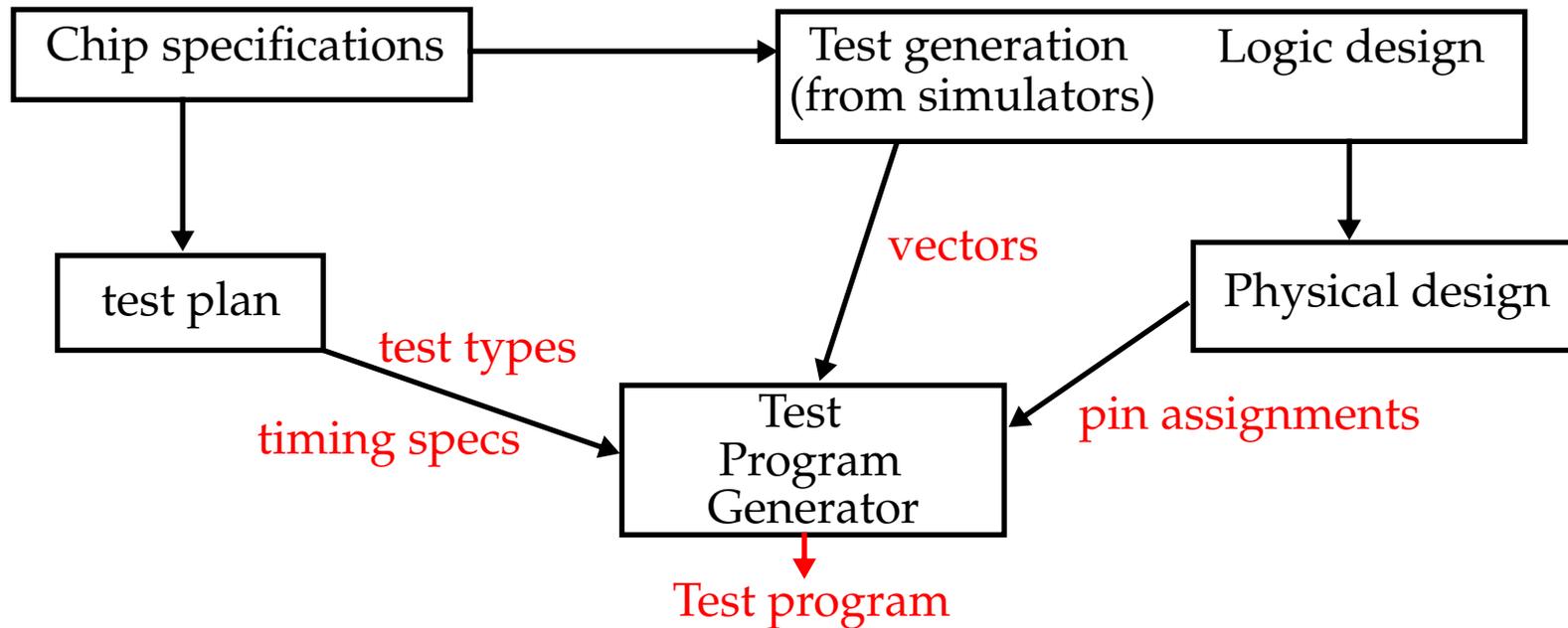
### Cantilever Style Probe Cards



## Test Programming

The test program and test vectors are needed once the chip is contacted

CAD tools are used to automate the generation of the test programs



#### 4 Basic Types of Testing

**Characterization testing**, design debug or verification testing:

Verifies correctness of design and test procedure

Production (**go/no-go test**):

Factory testing of all manufactured chips for parametric faults and for random defects

Shorter and less intensive test performed on every chip

Main driver is cost -- test time **MUST** be minimized

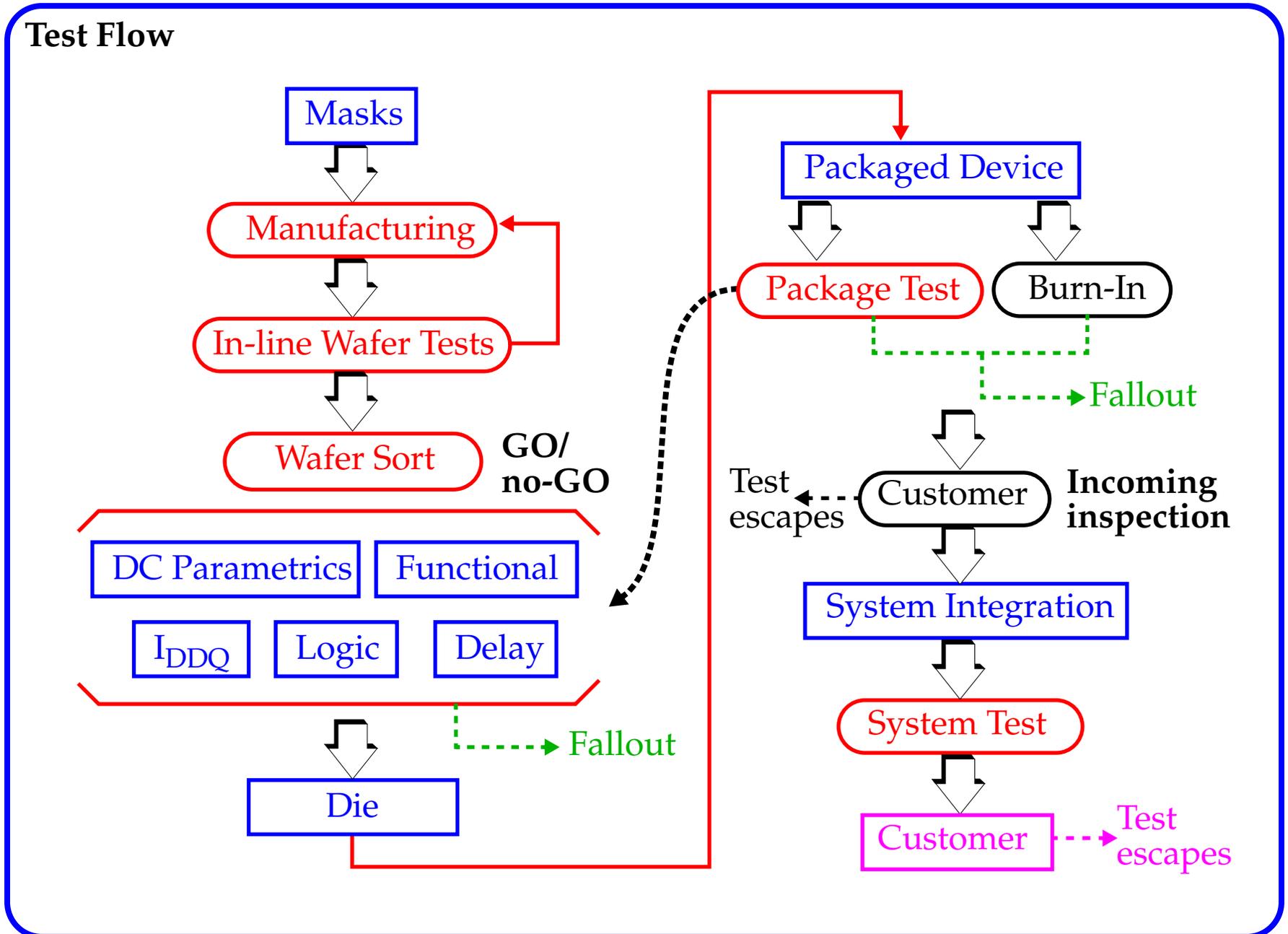
But tests must have high coverage of faults to ensure high quality

**Burn-in** or stress test:

Testing designed to stress the chip and accelerate the mechanisms that cause the chip to fail

**Acceptance testing** or incoming inspection:

Customer performs tests on purchased parts to ensure quality



## Physical Defects

Defects can be caused by *dust particles* on the mask, wafer surface or processing chemicals, e.g. photoresist

During photolithography, these particles lead to *unexposed* photoresist areas, leading to:

- *Unwanted material* or *unwanted etching* of the material
- Causes **shorts** and **opens** in the poly, active or metal layers

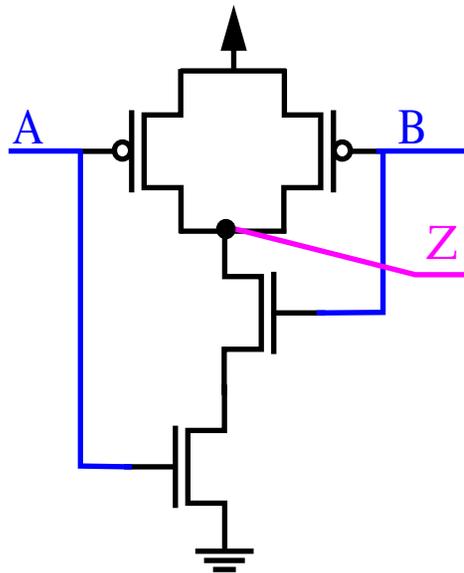
Opens in CMOS circuits are more difficult to detect because fault behavior is dependent on location, resistance and values of parasitic coupling cap, leakage currents, etc.

### Single stuck-at faults (SSF)

Assumes defects cause the signal net or line to remain at a fixed voltage level

Model includes **stuck-at-0** (SA0) or **stuck-at-1** (SA1) faults and assumes only one fault exists

For example, how many *SSF* faults can occur on an  $n$ -input NAND gate?



Inputs AB	Fault-Free Response	Faulty Response					
		A/0	B/0	Z/0	A/1	B/1	Z/1
00	1	1	1	0	1	1	1
01	1	1	1	0	0	1	1
10	1	1	1	0	1	0	1
11	0	1	1	0	0	0	1

What fault(s) does the pattern  $AB = 01$  detect?

What is the *minimum* number of tests needed to "detect" all of them?

## Single stuck-at faults (SSF)

An  $n$ -line circuit can have at most  $2n$  SSF faults.

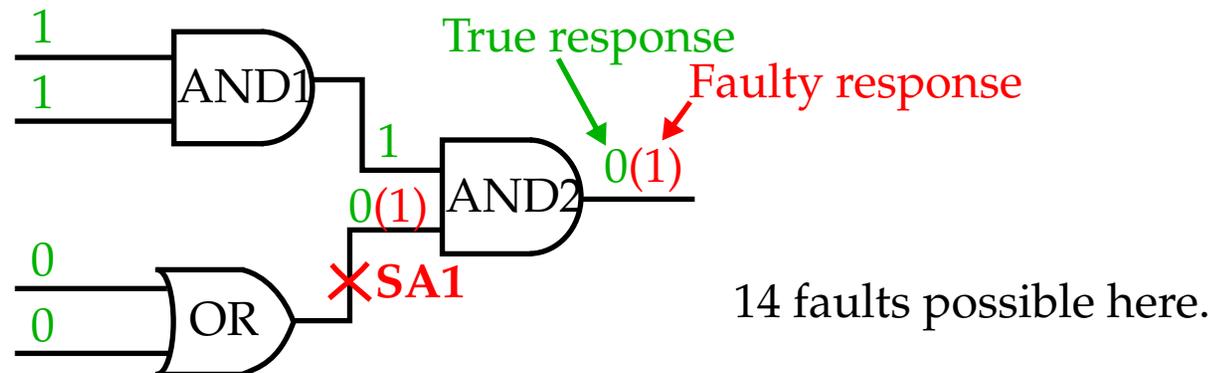
This number can be further reduced through **fault collapsing**.

Fault detection requires:

- A test  $t$  activates or provokes the fault  $f$ .
- $t$  propagates the error to *observation point* (primary output (PO)/scan latch).  
A line that changes with  $f$  is said to be **sensitized** to the fault site.

Fault propagation requires *off-path* inputs be set to *non-dominant* values.

01, 10, and 11  
do **not** provoke  
the fault



### Delay Faults

Delays along every path from PI to PO or between internal latches must be *less than* the operational system clock interval

An SA0 or SA1 can be modeled as a delay fault in which the signal takes an **infinite** amount of time to change to 1 or 0, respectively

Passing stuck fault tests is usually not sufficient however for systems that operate at any appreciable speed

Test Definition:

- At time  $t_1$ , the initializing vector of the two-pattern test,  $V_1$ , is applied through the input latches or PIs and the circuit is allowed to stabilize
- At time  $t_2$ , the second test pattern,  $V_2$ , is applied
- At time  $t_3$ , a logic value measurement (a sample) is made at the output latches or POs

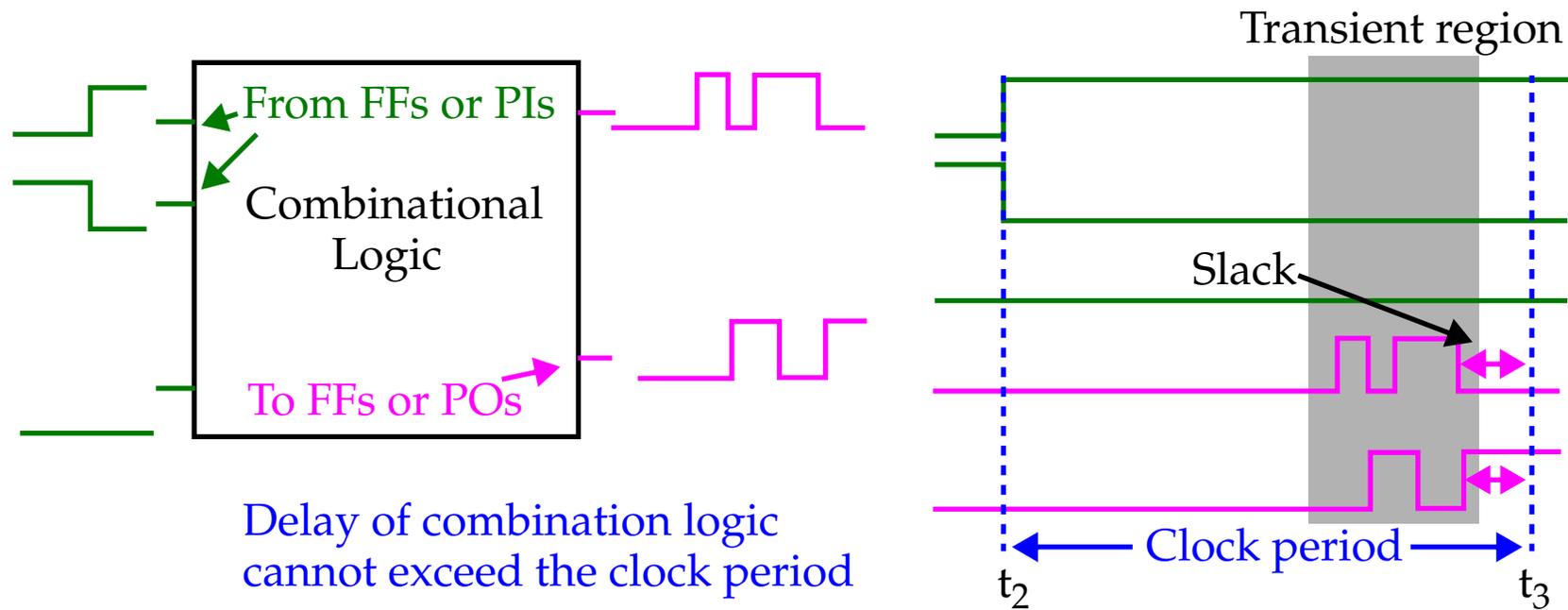
The delay test vectors  $V_1$  and  $V_2$  may sensitize one or more paths,  $p_i$

## Delay Tests

Let:

- $T_C = (t_3 - t_2)$  represent the time interval between the application of vector  $V_2$  at the PIs and the sampling event at the POs
- The *nominal delay* of each of these paths be defined as  $pd_i$
- The **slack** of each path be defined as  $sd_i = T_C - pd_i$

This is the difference between the propagation delay of each of the sensitized paths in the nominal circuit and the test interval



### Delay Fault Test Generation

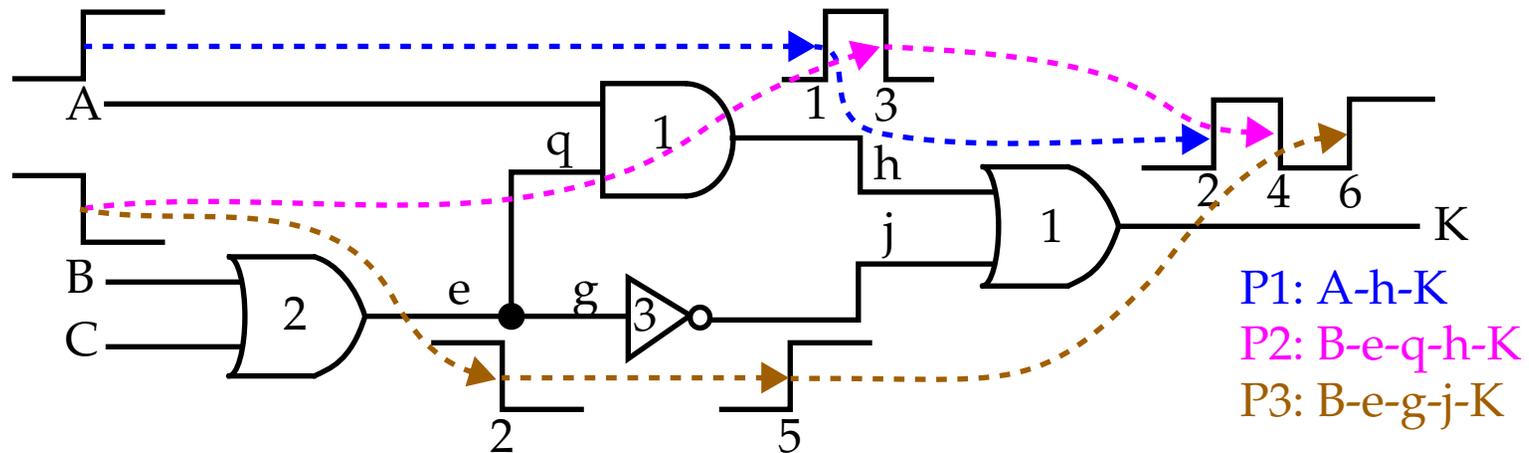
Difficulties with delay fault test generation:

- Test generation requires a sensitized path that extends from a PI to a PO
- Path selection heuristics must be used because the total number of paths is *exponentially* related to the number of inputs and gates in the circuit
- The application of the test set must be performed at the *rated speed* of the device
  - This requires test equipment that is capable of accurately timing two-vector test sequences
- The *detection* of a defect that introduces an additional delay,  $ad_i$ , along a sensitized path is dependent on satisfying the condition:  
$$ad_i > sd_i \text{ (or } pd_i + ad_i > T_C)$$

## Delay Tests

The critical path(s) of this circuit is 6 time units

Let's set the clock period  $T = 7$



Assume only one faulty path

No delay fault is detected if path delay along P3 is less than 7 units

## Scan

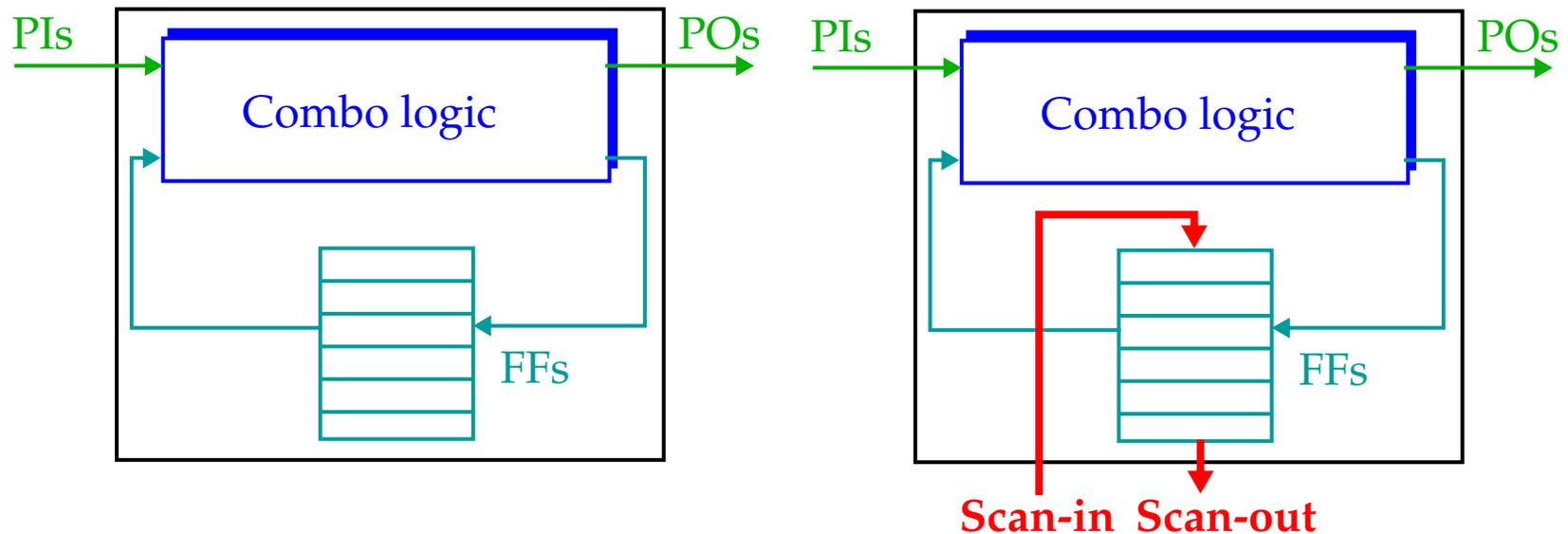
**Scan** proposed in '73 by Williams and Angell

Main idea is to obtain control and observability for FFs

It reduces *sequential* TPG to *combinational* TPG

With Scan, a synchronous sequential circuit works in two modes

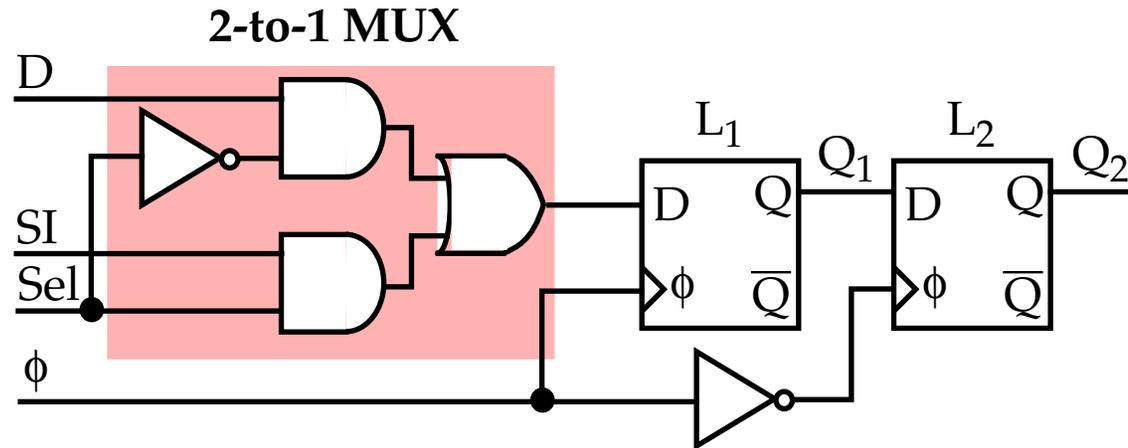
Normal mode and test mode:



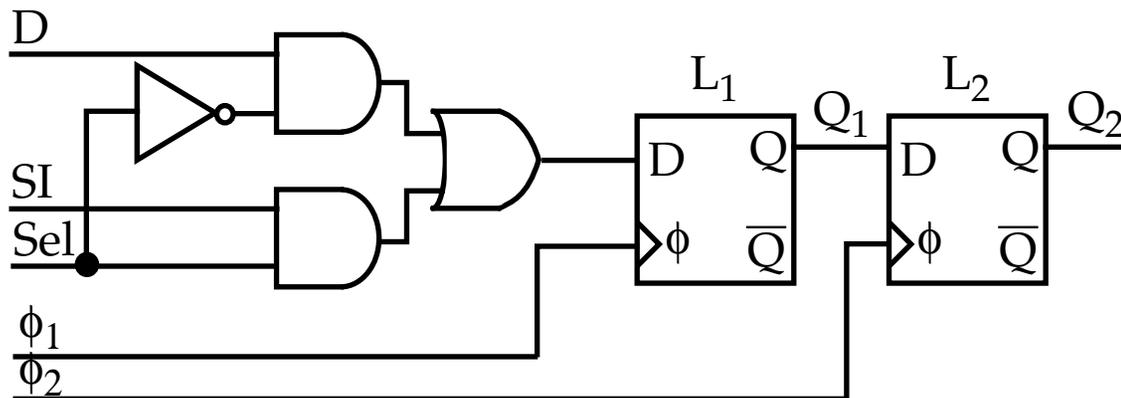
In **test mode**, all FFs are configured as a shift register, with Scan-in and Scan-out routed to a (possibly dedicated) PI and PO

### Storage Cells for Scan Designs

An implementation using two-port master-slave FF with a MUX



To ensure *race-free* operation, use a 2-phase nonoverlapping clk

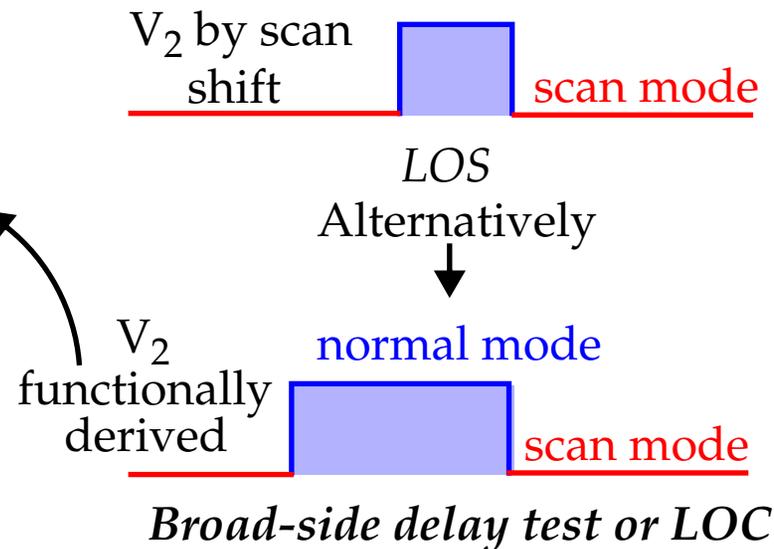
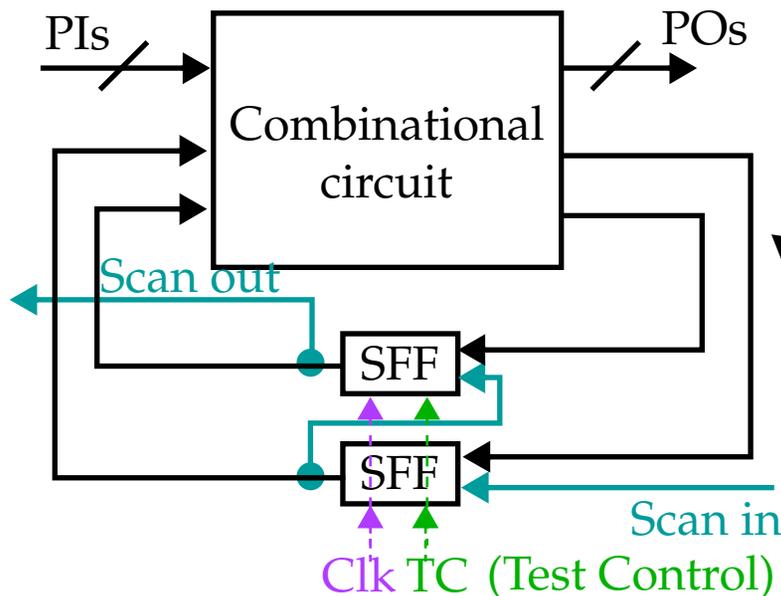


**Delay Test Methodologies**

*Launch-on-shift (LOS):* Scan in of  $V_1$  is followed by one extra cycle of slow clock with the circuit still in scan mode (TC = 0)

The test is designed so that  $V_2$  is obtained from  $V_1$  by a 1 bit translation (PI bits of both vectors are unrestricted)

As soon as  $V_2$  is applied, mode is changed from scan to normal and Clk is controlled at the rated period to latch outputs



## Delay Test Methodologies

### Normal-scan sequential test

*Launch-on-capture (LOC)*, the **state** portion (FF values) of  $V_2$  are functionally generated by the combo logic under  $V_1$

Simultaneous application of  $V_2$  at the PIs and into the FFs via Clk in normal mode generates the  $V_1 \rightarrow V_2$  transitions

The outputs are latched one rated clock period later

### Disadvantages:

For LOS, scan-enable must switch at rated speed of clk

For LOC, correlations between  $V_1$  and  $V_2$  may not allow high fault coverage