UNIT- VIII Testing

→ Tests fall into 3 mean categories :-

① Functionality tests or logic verification :.

The first set of tests verify that the chip perfoms its intended function. These tests are run before tapeout to verify the functionality of the circuit.

② silicon debug :-

The second set of ~~sets~~ tests are run on the first batch of chips that return from fabrication.

→ They can be much more extensive than the logic verification tests because the chip can be tested at full speed in a system.

For example, a new microprocessor can be placed in a prototype motherboard to try to boot the operating system.

→ This silicon debug require creative detective work to locate the cause of failures because the designer has much less visibility into chip compared to during design verification.

(3) Manufacturing Tests :-

These set of Tests verify that every Transistor, gate and storage element in the chip functions correctly.

- These Tests are conducted on each manufactured chip before shipping to the customer to verify that the silicon is completely intact.

$$\text{Yield of an IC} = \frac{\text{no. of good die}}{\text{Total number of die per wafer}}$$

- Because of the complexity of the manufacturing process, not all die on the wafer function correctly.
- Dust particles and small imperfections in starting material or photomasking can result in bridged connections or missing feature.
- These imperfections result in what is termed as a Fault.
- The goal of the manufacturing Test procedure is ~~to~~ determine which die are good and should be supplied to customers.

Testing a chip(die) can occur at diffelent levels

1. wafer level
2. packaged chip level
3. board level
4. system level
5. Field level.

→ By detecting a malfunctioning chip early, the manufacturing cost can be kept low.

→ For instance, the approximate cost to a company of detecting a fault at various levels is

→ wafer	\$0.01 – \$0.10
packaged chip	\$0.10 – \$1
board	\$1 – \$10
system	\$10 – \$100
field	\$100 – \$1000

Need For Testing :-

From above example, it is clear that the manufacturing cost is kept low, if the faults are detected at wafer level.

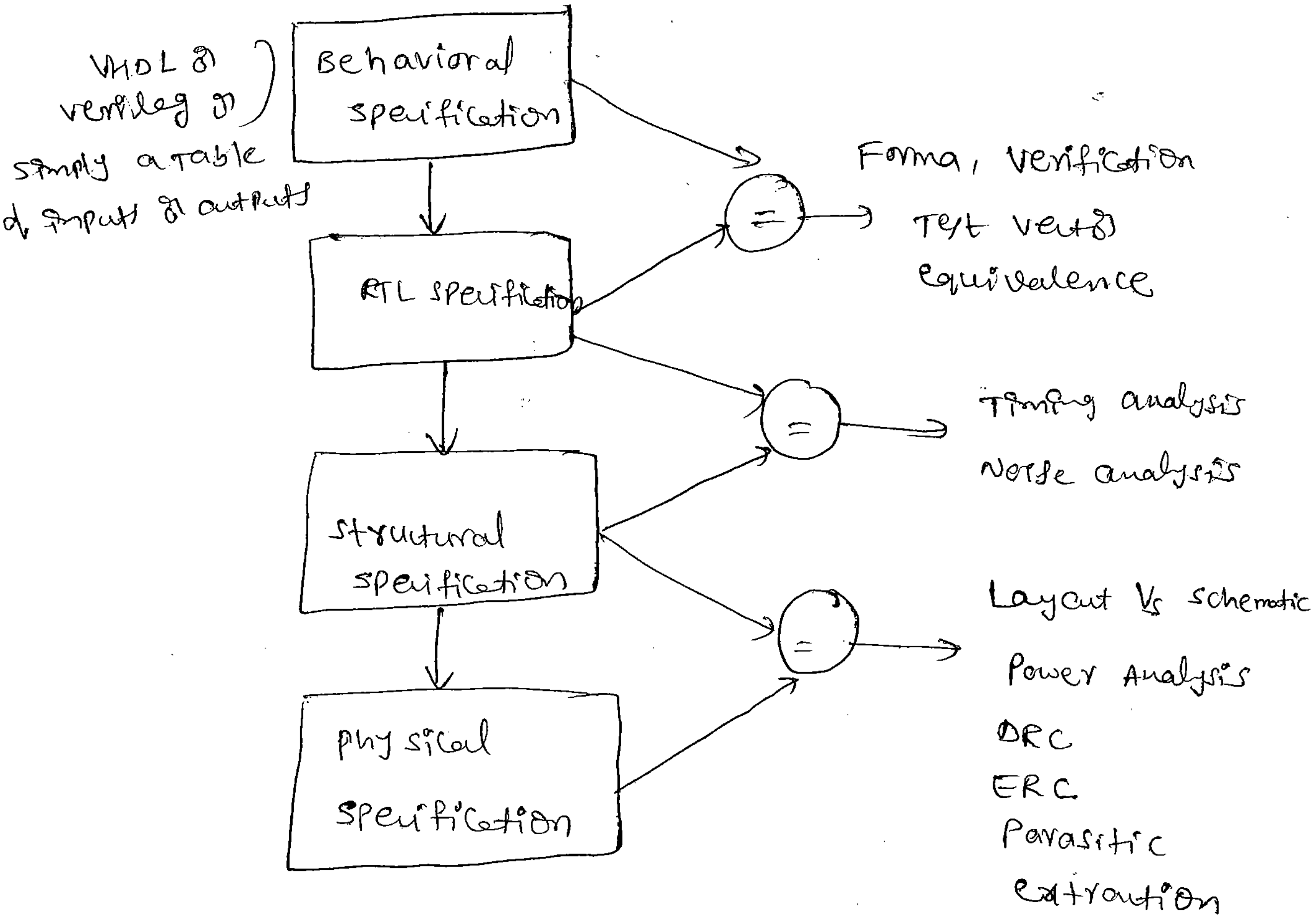
→ If yield high (good devices more) & package cost low (ie plastic package) then part can be tested only after packaging.

→ If wafer yield was lower and package cost high (ceramic package) it is better to screen bad dice at wafer level.

Logic & Funtional Verification:-

→ verification tests are usually the first ones a designer might construct as a part of the design process. like adder adding or not, counter counting & not.

→ In Functional Tests it is required to prove that a synthesized gate description was functionally equivalent to the source RTL.



→ Functional equivalence involves running a simulator at some level on the 2 descriptions of the chip (one at the gate level and one at functional level) and ensure that the outputs are equivalent at some convenient check points in time for all inputs applied. This is mostly convenient in HDL by employing a Test bench.

→ We have to simulate as closely as possible the way in which the chip or system will be used in the real world.

Manufacturing Tests :—

~~They~~ These tests are used to verify that every gate operates as expected. The need to do this arises from a number of manufacturing defects that might occur during either chip fabrication or accelerated life testing (where the chip is stressed by over voltage or over temperature).

Typical defects include:

- Layer to Layer shorts (e.g. metal-to-metal)
- discontinuous wire (eg: metal thins when crossing vertical topology jumps)
- missing or damaged via
- shorts through the thin gate oxide to the substrate or well).

These cause circuit maladies like,

① nodes shorted to power or ground

② nodes shorted to each other

③ inputs floating / outputs disconnected.

Apart from the verification of internal gates, I/o integrity is also tested with the following test being completed.

① I/o levels (ie, checking for noise margin for TTL, ECL & Cmos I/o pads).

② speed test.

→ In general, manufacturing test generation assumes the function of the circuit/chip is correct. It require ways of exercising all gate inputs and monitoring all gate outputs.

Functional verification principles:-

① Test Benches and Harnesses.

② Regression testing

③ version control

④ Bug Tracking.

① Test benches and Harnesses:-

a verification test bench or harness is a piece of HDL code that is placed as a wrapper around a core piece of HDL.

→ In Test bench, inputs are applied to the module under ~~test~~ test and at each cycle, the outputs are examined to determine wheather they comply with a predefined expected data set.

→ The data set can be derived from another model and available of a file or the value can be computed on the fly.

Regression Testing :-

(C, FORTRAN, Pascal & LISP)

High level language scripts are frequently used when running large test benches, especially for "regression testing".

→ Regression testing involves performing a suite of simulations to automatically verify that no functionality has inadvertently changed in a module or set of modules.

Version control :-

Combined with regression testing is the use of versioning, ie, the orderly management of different design iterations.

Unix / Linux Tools such as CVS (Circuit vs Schematic) are useful for this.

Bug Tracking :-

Important tool to use during verification is a "bug-Tracking" system.

→ Bug Tracking systems such as unix / Linux based GNATS allow the management of a wide variety of bugs.

→ Each bug is entered & noted the location, nature and severity of the bug.

Manufaturing Test principley :-

① observability :-

→ The obseevability of a Particular circuit node is the degree to which you can obseeve that node at the outputs of an Integrated circuit (ie, the pins).

→ This metric is relevant when you want to measure the output of a gate within a larger circuit to check that it operates correctly.

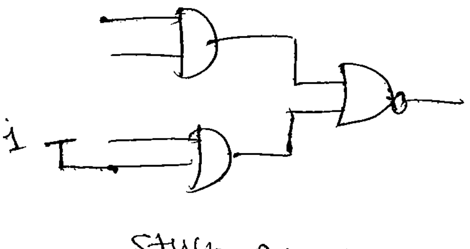
② ~~8~~ Fault modely :-

This is a model for how faults occur and their impact on circuits. The most popular model is called "stuck-At" model.

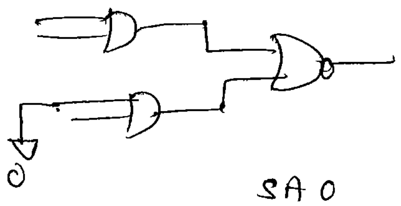
→ The short circuit / open circuit model can be closer fit to reality, but is harder to incorporate into logic simulation tooly.

⇒ Stuck-At faults :-

In this model, a faulty gate input is modeled as a stuck at zero (stuck-At-0, S-A-0) & stuck At one (S-A-1).



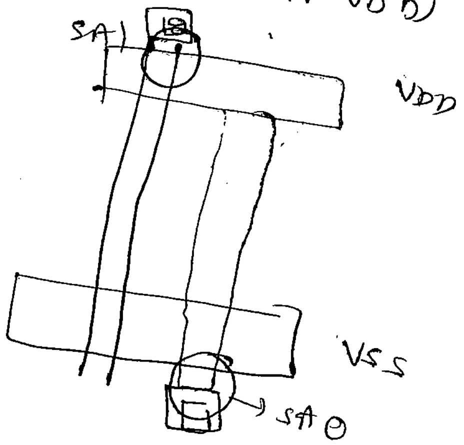
Stuck At 1



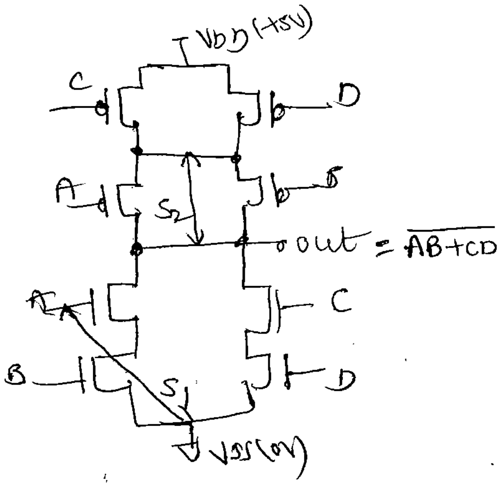
SA 0

Stuck At 0

These faults most frequently occur due to gate oxide shorts (nmos gate to GND , Pmos gate to VDD) & metal-metal shorts.



⇒ short circuit and open-circuit Faults:-

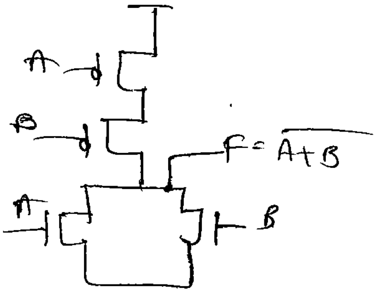


S2 modifie function of gate

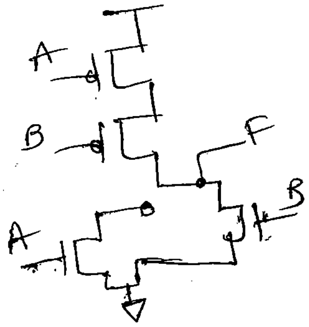
S1 → modeled by an SA0 fault at input A , while short S2 modifie the function of the gate .

→ A problem that arises with Cmos is that it is possible for a fault to convert a combinational circuit into a sequential circuit.

Consider 2-input NOR gate



Normal Circu...



0	0	1
0	1	0
1	1	0
1	0	0 (Fn).

$$F = (\overline{A+B}) + A\bar{B}.Fn$$

→ Fn is the Previous state of the gate.

1ly If B ~~B~~ n-Transistor drain connection is missing

the function is,

$$F = ((\overline{A+B})) + \bar{A}.B.Fn.$$

Controllability :-

The Controllability of an internal circuit node within a chip is a measure of the eaye of setting the node to a 0 & 1 state.

→ An easily controllable node would be directly settable via an input pad.

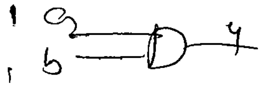
EX: global reset to ~~one~~ ~~to~~ get initial positions of all inputs.

Fault Coverage :-

A measure of goodness of a Test program is the amount of fault coverage it achieves.

ie, for the vectors applied, what percentage of the chip's internal nodes were checked.

→ Each circuit node is taken in sequence and held to SA0, and the circuit is simulated, comparing the chip outputs with a known "good machine".



a	b	good machine	SA0-a	SA0-b	SA1-a	SA1-b
0	0	0	0	0	0	
0	1	0	0	0	1	
1	0	0	0	0		
1	1	1	0	0		

a	b	y	SA0-a	SA0-b	SA1-a	SA1-b
0	0	0	0	0	0	0
0	1	0	0	0	$\boxed{1}$	0
1	0	0	0	0	0	$\boxed{1}$
1	1	1	$\boxed{0}$	$\boxed{0}$	1	1

11, 01, 10 Test vectors.

→ The total number of nodes that, when set to 0 or 1, do result in the detection of fault, divided by the total number of nodes in the circuit is called "percentage fault coverage").

Fault-sampling :-

An Approach to Fault analysis is "Fault-sampling". This is used in circuits where it is impossible to fault every node in the circuit.

→ node are randomly selected and faulted.

→ The resulting fault-detection rate may be statistically inferred from the no. of faults that are detected in the fault set and the size of the set.

Statistical Fault Analysis:-

→ This method of fault analysis relies on estimating the probability that a fault will be detected.

→ Extra statistics are gathered by a modified simulator on a per-input vector basis.

① Zero counter – The 0 count on each gate input when a 1→0 change of output is detected.

② One-counter – The 1 count on each gate input when a 0→1 change of output is detected.

③ sensitization counter – incremented If the input change causes the output to be sensitized.

④ Loop-counter – used to detect & deal with feedback.

The one-controllability of line l is given by,

$$C1(l) = \text{one-count}/N;$$

N→ no. of vectors.

zero-controllability

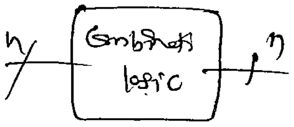
$$C0(l) = \text{zero-count}/N;$$

One-level sensitization probability is,

manufacturing Test principle:

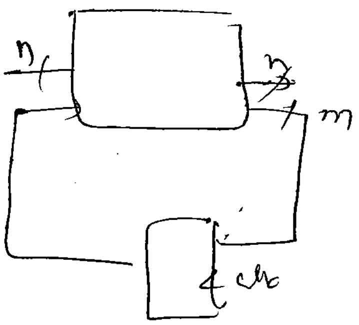
Consider a combinational circuit with n-inputs.

→ To test this circuit exhaustively, a sequence of 2^n inputs (Test vectors) must be applied & observed to fully exercise the circuit.



$n=10 \Rightarrow 2^{10} = 1024$ Test vectors.

→ The combinational circuit is converted to sequential circuit with additional of m-storage registers.



→ The state of the circuit is determined by the inputs & previous state.

A minimum of 2^{n+m} Test vectors must be applied to exhaustively Test the circuit.

Ex: A n/w with $n=25$, $m=50$

$2^{25+50} = 2^{75}$ Test vectors

$\cong 3.8 \times 10^{22}$.

If 1 µs per pattern, Then total test time would be over a billion years (10^9).

```
library IEEE;
Use IEEE. STD - LOGIC-1164. All;

 entity and1 is
    port ( a : in std-logic;
           b : in std-logic;
           y : out std-logic);
   end and1;
   architecture and12 of and1 is
     begin
     ~~y<= a and b;~~        process (a,b)
                              begin
                              c<= a and b;
                             end process;
                             end and12;
```

Test bench:

```
     library IEEE;
     Use -

      entity and1-tb is
        end and1-tb;
      architecture and1-tb of and1-tb is
          ~~be~~ signal  a-s, b-s;
            ~~begin~~
           component  and1 is
               ~~entity~~ ~~po~~port ( a : in std-logic,
                              b : in std-logic,
                               y : out std-logic);
              end component;
          begin
       U1 : component and1 portmap (a => a-s,
b <= 0;
a <= 0 ;
wait for 1 ns;

a <= 1;
 b <= 0;
wait for 1 ns;
a <= 0
 b <= 1
 wait for 1ns;
  a <= 1
  b <= 1
  wait;
 end and1_tb;
```

21/09

1, 2, 3, 6, 8, 9, 10, 13, 16, 19, 20, 22, 23, 31, 35, 38

Chip-level Test Technique :-

In this we will examine practical method of incorporating test requirements into a design. This discussion is structured around the main types of circuit structure that will be encountered in a digital cmos chip.

1. Regular Logic Arrays
2. memories
3. Random Logic

Regular Logic Arrays :-

Partial serial scan & parallel scan is probably the best approach for structures such as datapaths.

One approach that has been used in a Lisp microprocessor is,

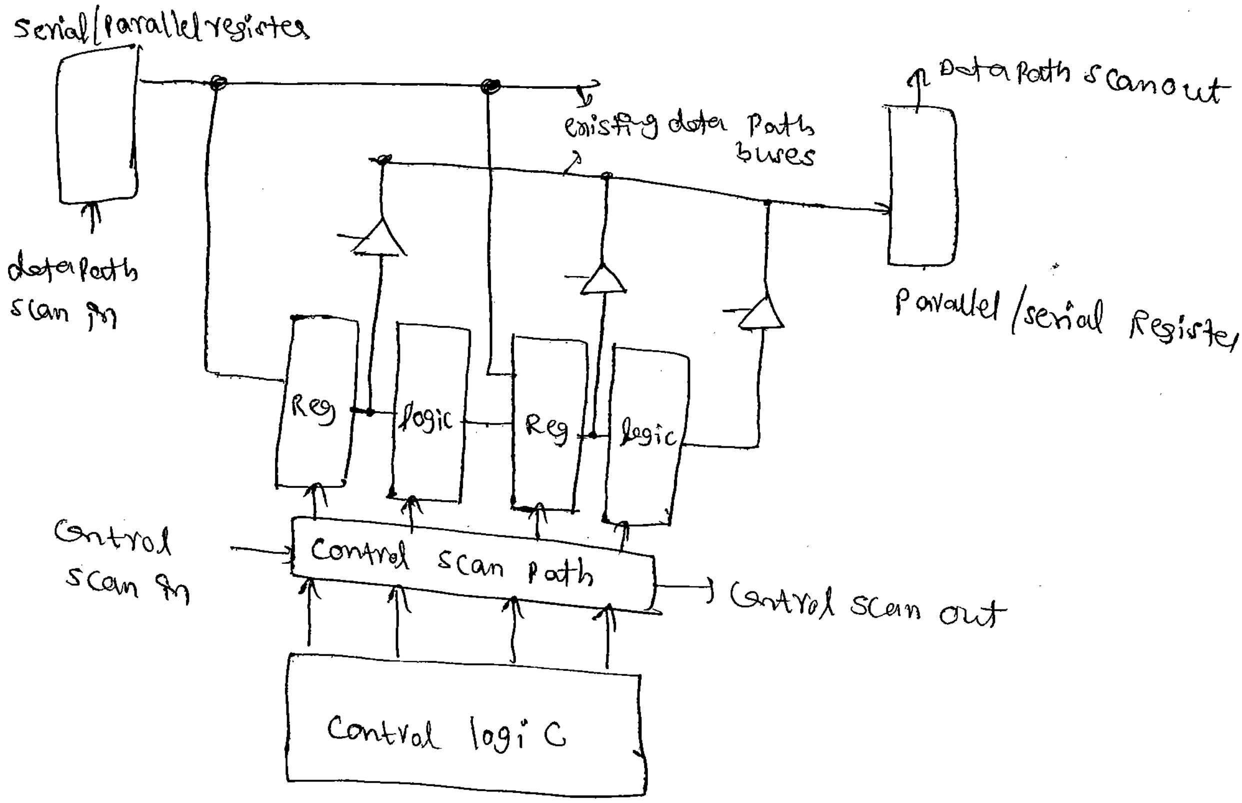


Fig:

→ The input buses may be driven by a serially loaded register.

→ They e in turn may be used to load internal data path Registers.

→ The data path Registers may be sourced onto a bus & this bus may be loaded into a Register that may be serially accessed. All of the control signals to the Datapath are also made scannable.

Memories:—

memory may use the self-testing techniques by embedding self Test circuits for memories in higher-speed circuits.

→ Anotherway, the provision of multiplexers on data inputs and addresses and convenient external access to data outputs enables the testing of embedded memories.

→ It is a mistake to have memories indirectly accessible.
ie, data is written by passing through logic.
data is observed after passing through logic.
addresses cannot be conveniently sequenced.

→ Because memories have to be tested exhaustively, any overhead on writing and reading the memories can substantially increase the test time.

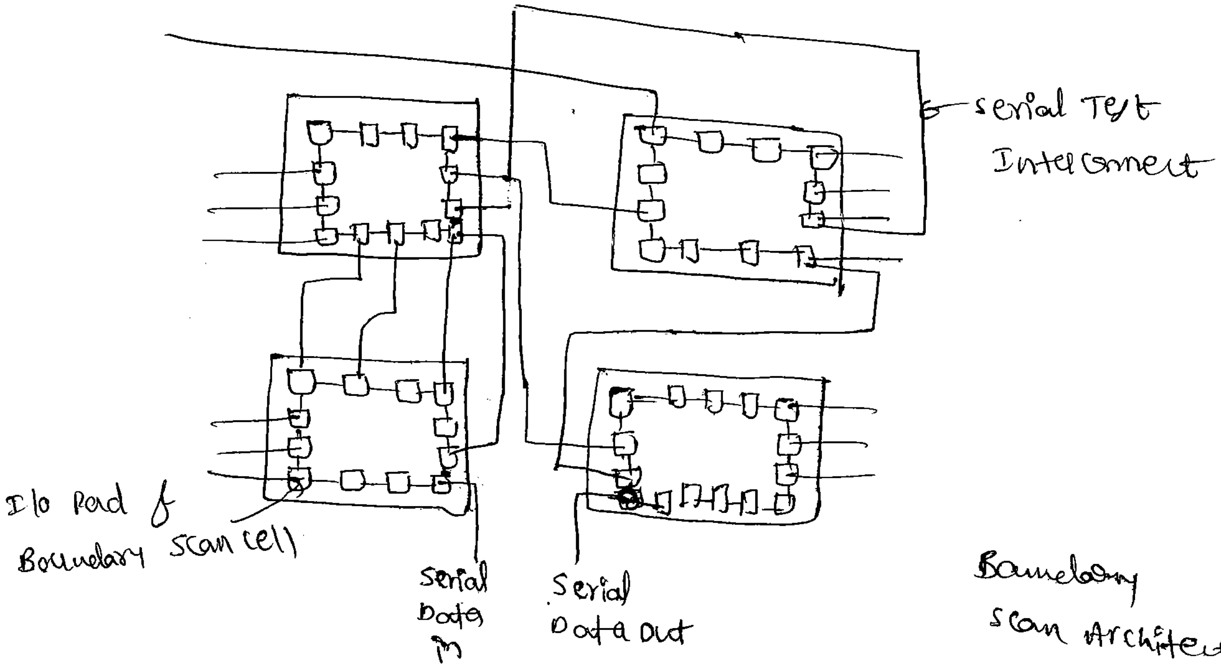
Random logic:—

Random logic is probably best tested via full serial scan &

System level Test Techniques:-

The increasing complexity of boards and the movement to different technologies like MCM (multi chip modules) resulted in system designers agreeing on a unified scan-based methodology for testing chips at the board and system level. This is called "Boundary Scan".

Boundary scan:-



Boundary Scan Architecture

→ It provides a standardized serial scan path through the I/o Pins of an IC.

TAP: (Test Access Port)

The TAP is a definition of the interface that need to be included in an IC to make it capable of being [illegible]

In a Boundary - scan architecture.

→ The port has 4 or 5 single bit connections.

① TCK (Test clock input)- used to clock tests into & out of chips.

② TMS (Test mode select) - used to control test operations.

③ TDI (The test Data input)— Used to input test data to a chip

④ TDO (Test Data output)- used to output Test Data from a chip.

optional pin:

* TRST (Test Reset signal) used to asynchronously reset the TAP controller.

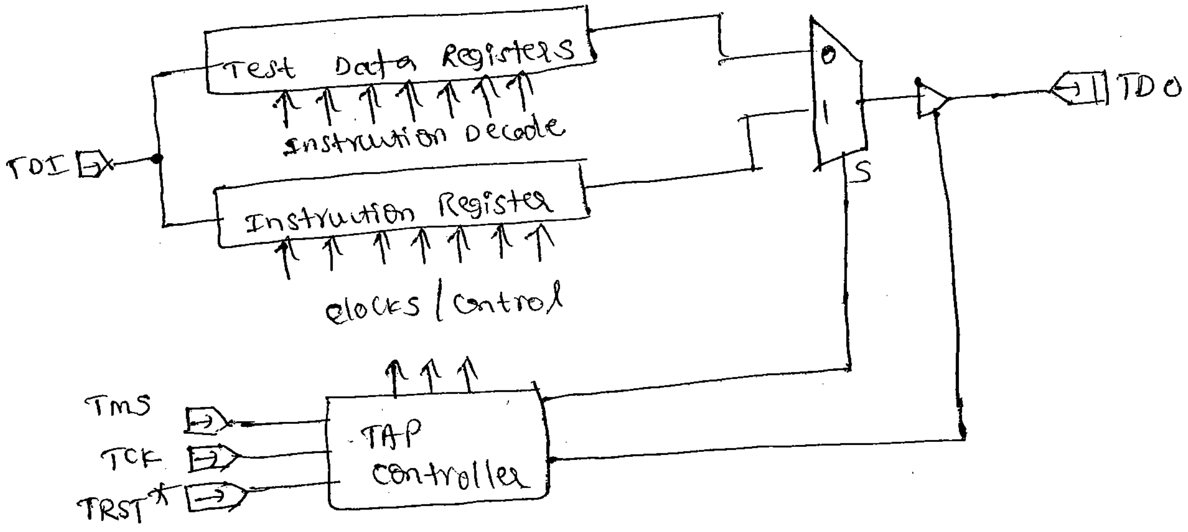
The Test Architecture:-

It consists of ① The TAP Interface pins

② Set of test-data Registers to collect data from chip.

③ an Instruction Register to enable test inputs to be applied to the chip.

④ A TAP controller, which interprets test instructions & controls the flow of data into & out of the TAP.



...ntroller :-

The TAP controller is a 16-state FSM that proceeds from state to state based on the TCK and TMS signals. It provides signals that control the test data Registers, and Instruction Register. These include serial-shift clocks & update clocks.

The Instruction Register (IR):

The IR has to be atleast 2 bits long & logic detecting the state of the IR has to decode at least 3 Instructions. They are,

① BYPASS — This Instruction is represented by an IR having all zero's in it.

→ It is used to bypass any serial-data Registers in a chip with a 1-bit Register.

② EXTEST — This allows for the testing of off-chip circuitry and is represented by all ones in the IR.

③ SAMPLE/PRELOAD — This places the boundary scan registers (ie at chip's I/O Pins) in the DR chain & samples or preloads the chips I/O's.

Test Data Registers (DRs):

These are used to set the inputs of modules to be tested and to collect the results of running Tests.

→ The simplest data-Register configuration would be a boundary scan register and a bypass Register (1-bit long).

→ A multiplexer under the control of ~~TAP~~ TAP controller selects which particular data register is routed to the TDO pin

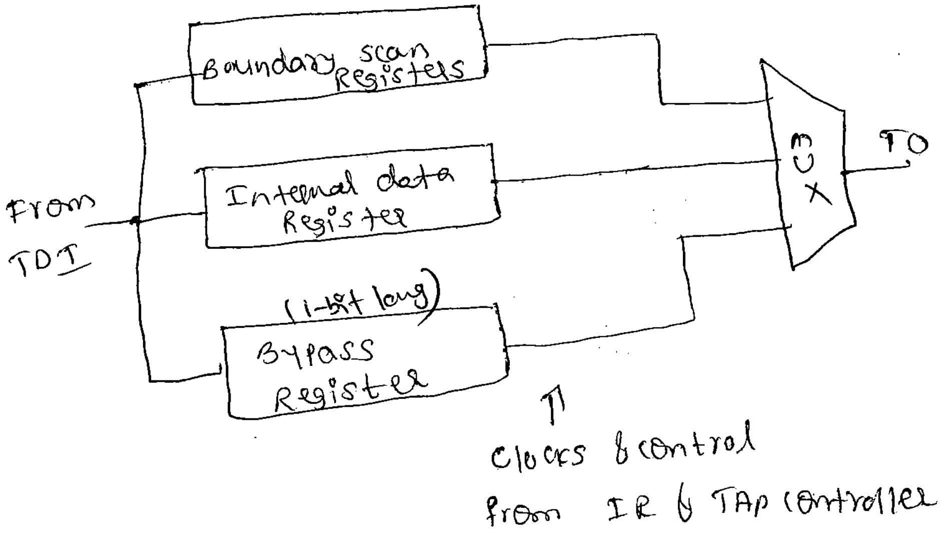


Fig: TAP data Register.

Boundary scan Registers :-

This is a special case of data Register. It allows circuit-board interconnections to be tested, external components tested & the state of the chip digital I/o to be sampled.

→ It consists of 2 multiplexers & 2 edge-Triggered Registers.

Design strategies for Test :-

Design For Testability :-

The 3 main approaches to what is commonly called "Design for Testability". These may be Categorized as

1. Ad-hoc Testing
2. Scan-based approaches
3. Self-Test & Built-in Testing

Ad-Hoc Testing :-

Ad-hoc Test Technique are collections of ideas aimed at Reducing the combinational explosion of Testing. Common Techniques involve:

1. Partitioning large sequential circuits.
2. adding Test points.
3. adding multiplexers
4. Providing for easy state reset.

Ex: - Long Counters.

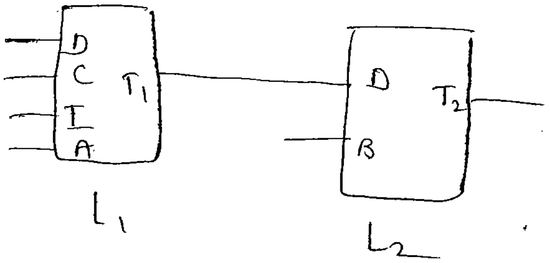
Scan-based Test Techniques:-

1. Level sensitive scan Design (LSSD)

LSSD is based on 2 tenets.

1. The circuit is Level sensitive
2. each register may be converted to a serial shift Register.

The basic building block of LSSD is shift Register Latch (SRL).



SRL

SRL consist of 2 latches L_1 & L_2.

L_1 has a serial data port I,
enable A
Data port D
enable C.

when A is high ⇒ value of $L_1(T_1)$ is set by value of I.

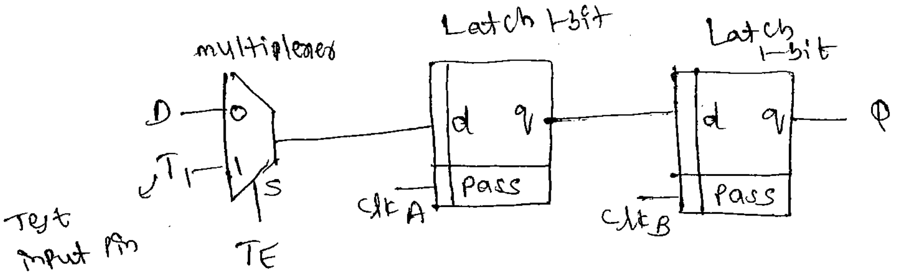
when C is high ⇒ value of $L_1(T_1)$ is set by D.

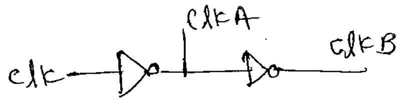
A & C can not be simultaneously high.

→ when signal B in L_2 is high ⇒ ~~TP~~ $T_2 \Leftarrow T_1$ (T_1 passed to T_2)

serial scan:-

Schematic for a cmos edge-sensitive scan-Register is





→ A mux is added before the master latch in a conventional D-Register.

→ TE is Test enable pin, T_1 is Test Input pin

when TE enabled ⇒ T_1 is clocked into Register by Rising edge of clk.

Partial serial scan :-

In Dataflow section of the chip, pipeline Registers removed. In this case only input & output Registers need be made scannable. This Technique of Testing is known as "Partial Scan", & depends on the designer making decisions about which registers need to be made scannable.

→ In full scan Test, all Registers would have to be scannable.

→ The part of the circuit that is being tested and monitored by the scan Registers known as "kernel".

Parallel scan :-

→ serial scan Chains can become quite long & loading, unloading sequence can dominate Test time. An extension of serial scan is called "Random access & Parallel scan".

→ In parallel scan,

each Register in the design is arranged on an imaginary or real grid where Registers on common rows receive common data lines

Registers in common columns Receive common Read & write control signals.

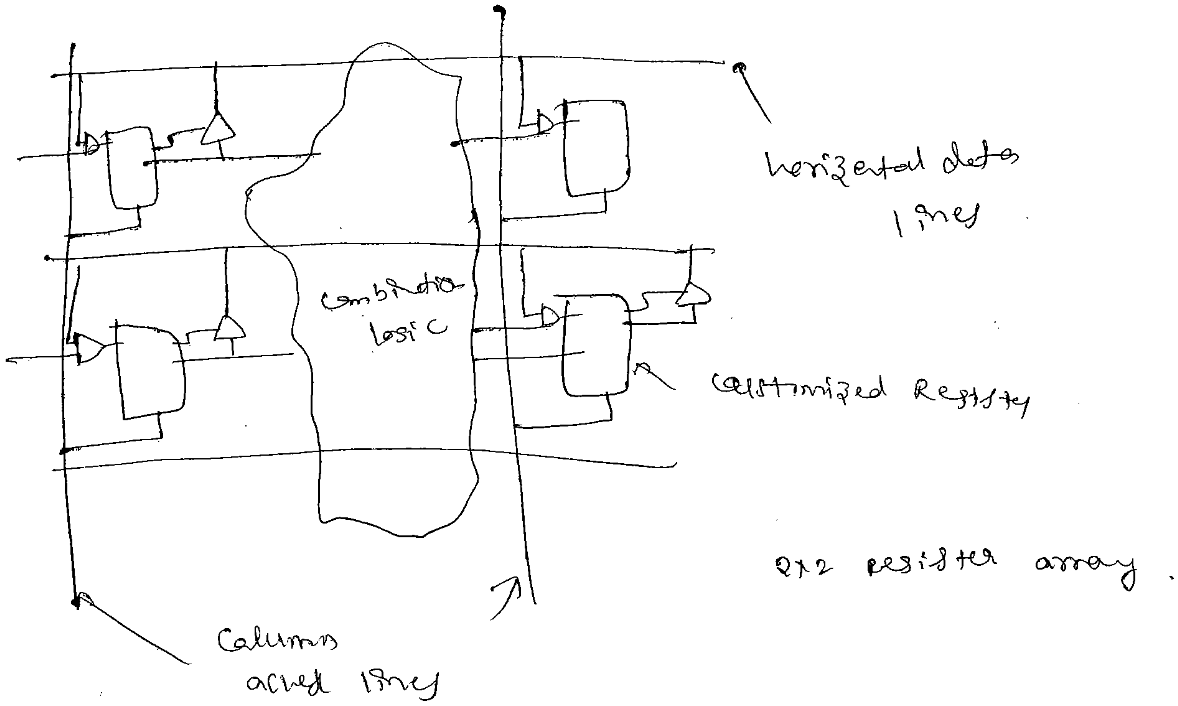


fig: parallel scan- basic structure.

→ Any register output may be observed by enabling the appropriate column Read line & setting the appropriate address on an output data multiplexer.

→ similarly, data may be written to any Register.

Layout Design for Improved Testability :-

→ In order to predict layout styles that improve testability, a designer has to have some idea of nature and frequency of defects for a particular process.

→ The types of defects divided into ① shorts circuits
② open circuits.

→ shorts occurs in all layers used for connections ie, diffusion, polysilicon, metal1, metal2 · & oxide short to substrate & source or drain,

→ All conducting layers might have open circuits. In addition ... layer of a badly etched lead to interlayer

Memory self-Test :-

Embedding self-Test circuits for memorie in higher speed circuits not only may be the way of testing the structures at speed but can save on the number of external Test vectors that have to be run.

EX: 4-mbit RAM with self-Test.

The self-Test consists of 256k cycle that input a checkboard pattern to test for cell to cell interference.

→ This is followed by 256k cycle in which the data is readout. Then a complemented checkboard is written and read.

→ A total of 1 million cycle provide a test sufficient for system maintenance.

→ The advantage of self-Test methods is that "Testing may be completed. when the part is in the field."

→ we can perform self-Test even during normal system operation with care.

Iterative Logic Array Testing: (ILA Testing)

→ Arrays of logic present is the problem to the Test architect because the replication can be used to advantage in reducing number of Tests.

→ An ILA is a collection of identical logic modules (EX: n-bit adder).

→ An ILA is C-Testable if it can be tested with a constant

An ILA is I-Testable if a particular fault that occur in any module as a result of an applied input vector is identical for all modules in ILA.

IDDQ Testing :-

Popular method of testing for bridging faults is called "IDD Q" (VDD supply current Quiescent) or current - supply monitoring.

→ When CMOS logic gates not switching it draws no DC current (except for leakage).

→ When bridging fault occur, for some combination of inputs a measurable DC IDD will flow.

→ Applying normal vectors, allowing signals to settle & then measuring IDD. This is IDDQ Testing.

→ Because current measuring is slow, test must be run slower than normal , so increasing Test Time.

↳ But gives a form of indirect massive observability at little circuit overhead.