UNIT – IV CMOS TESTING

1. **Mention the levels at which testing of a chip can be done?**
   - At the wafer level
   - At the packaged-chip level
   - At the board level
   - At the system level
   - In the field
2. **What is meant by observability?**

   The observability of a particular internal circuit node is the degree to which one can observe that node at the outputs of an integrated circuit.
3. **What is meant by controllability?**

   The controllability of an internal circuit node within a chip is a measure of the ease of setting the node to a 1 or 0 state.
4. **List the basic types of CMOS testing. [AUC JUNE 2013]**
   - Logical verification
   - Functional verification
   - Fault testing and coverage
   - Physical and pre-layout verification
   - Post layout verification
5. **What is meant by logic verification ?[AUC JUNE 2013]**
   - Test benches
   - Regression testing
   - Version control
   - Bug ttracking
6. **What are the test fixtures required to test a chip ?[AUC NOV 2011]**

   A Socket for transmitting electrical signals from the IC tester to an IC device under test.
7. **What are the categories of testing?**
   - Functionality tests
   - Manufacturing tests

8. **Write notes on functionality tests?[AUC NOV 2011]**

Functionality tests verify that the chip performs its intended function. These tests assert that all the gates in the chip, acting in concert, achieve a desired function. These tests are usually used early in the design cycle to verify the functionality of the circuit.

9. **Write notes on manufacturing tests?**

Manufacturing tests verify that every gate and register in the chip functions correctly. These tests are used after the chip is manufactured to verify that the silicon is intact.

10. **Mention the defects that occur in a chip?[AUC MAY 2011]**

- layer-to-layer shorts
- discontinous wires
- thin-oxide shorts to substrate or well
- Give some circuit maladies to overcome the defects?
- nodes shorted to power or ground
- nodes shorted to each other
- inputs floating/outputs disconnected

11. **What are the tests for I/O integrity?**

- I/O level test
- Speed test
- IDDQ test

12. **What is meant by fault models?**

Fault model is a model for how faults occur and their impact on circuits.

13. **Give some examples of fault models?**

- Stuck-At Faults
- Short-Circuit and Open-Circuit Faults

14. **What is stuck – at fault?**

With this model, a faulty gate input is modeled as a “stuck at zero” or “stuck at one”. These faults most frequently occur due to thin-oxide shorts or metal-to-metal shorts.

15. **What is known as percentage-fault coverage?**

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

16. **What is fault grading?**

Fault grading consists of two steps. First, the node to be faulted is selected. A simulation is run with no faults inserted, and the results of this simulation are saved. Each node or line to be faulted is set to 0 and then 1 and the test vector set is applied. If and when a

discrepancy is detected between the faulted circuit response and the good circuit response, the fault is said to be detected and the simulation is stopped.

17. Mention the ideas to increase the speed of fault simulation?

- parallel simulation
- concurrent simulation

18. What is fault sampling?

An approach to fault analysis is known as fault sampling. This is used in circuits where it is impossible to fault every node in the circuit. Nodes are randomly selected and faulted. The resulting fault detection rate may be statistically inferred from the number of faults that are detected in the fault set and the size of the set. The randomly selected faults are unbiased. It will determine whether the fault coverage exceeds a desired level.

19. What are the approaches in design for testability?

- ad hoc testing
- scan-based approaches
- self-test and built-in testing

20. Mention the common techniques involved in ad hoc testing?

- partitioning large sequential circuits
- adding test points
- adding multiplexers
- providing for easy state reset
- What are the scan-based test techniques?
- Level sensitive scan design
- Serial scan
- Partial serial scan
- Parallel scan

21. What are the self-test techniques?

- Signature analysis and BILBO
- Memory self-test
- Iterative logic array testing

22. What is known as BILBO?

Signature analysis can be merged with the scan technique to create a structure known as BILBO- for Built In Logic Block Observation.

23. What is known as IDDQ testing? [AUC APR 2011]

A popular method of testing for bridging faults is called IDDQ or current supply monitoring. This relies on the fact that when a complementary CMOS logic gate is not

switching, it draws no DC current. When a bridging fault occurs,for some combination of input conditions a measurable DC IDD will flow.

24. What are the applications of chip level test techniques?

- Regular logic arrays
- Memories
- Random logic

25. What is boundary scan?[AUC NOV 2009]

The increasing complexity of boards and the movement to technologies like multichip modules and surface-mount technologies resulted in system designers agreeing on a unified scan-based methodology for testing chips at the board. This is called boundary scan.

26. What is the test access port?

The Test Access Port (TAP) is a definition of the interface that needs to be included in an IC to make it capable of being included in a boundary-scan architecture. The port has four or five single bit connections, as follows:

- TCK(The Test Clock Input)
- TMS(The Test Mode Select)
- TDI(The Test Data Input)
- TDO(The Test Data Output)
- It also has an optional signal
- TRST*(The Test Reset Signal)

27. What are the contents of the test architecture?

- The test architecture consists of:
- The TAP interface pins
- A set of test-data registers
- An instruction register
- A TAP controller

28. What is known as boundary scan register?[AUC NOV 2009]

The boundary scan register is a special case of a data register. It allows circuit-board interconnections to be tested, external components tested, and the state of chip digital I/Os to be sampled.

PART –B (16 MARKS)

1. **Explain briefly about different faults take place in CMOS design. [AUC MAY 2011]**
Examples of *physical defects* include:

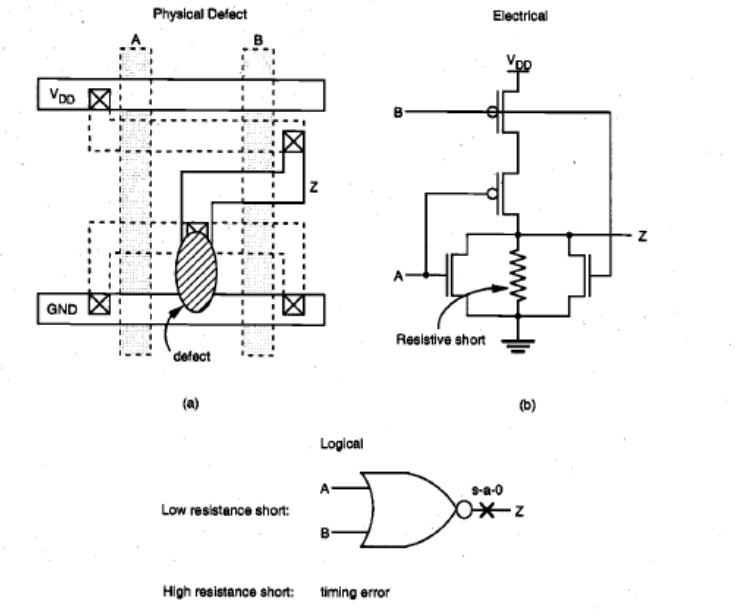
- Defects in silicon substrate
- Photolithographic defects
- Mask contamination and scratches
- Process variations and abnormalities
- Oxide defects

The physical defects can cause electrical faults and logical faults. The *electrical faults* include:

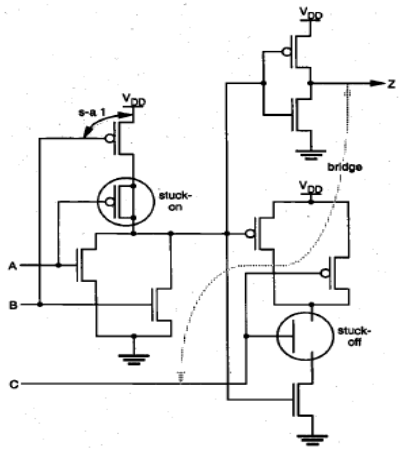
- Shorts (bridging faults)
- Opens
- Transistor stuck-on, stuck-open
- Resistive shorts and opens
- Excessive change in threshold voltage
- Excessive steady-state currents

The electrical faults in turn can be translated into logical faults. The *logicalfaults* include:

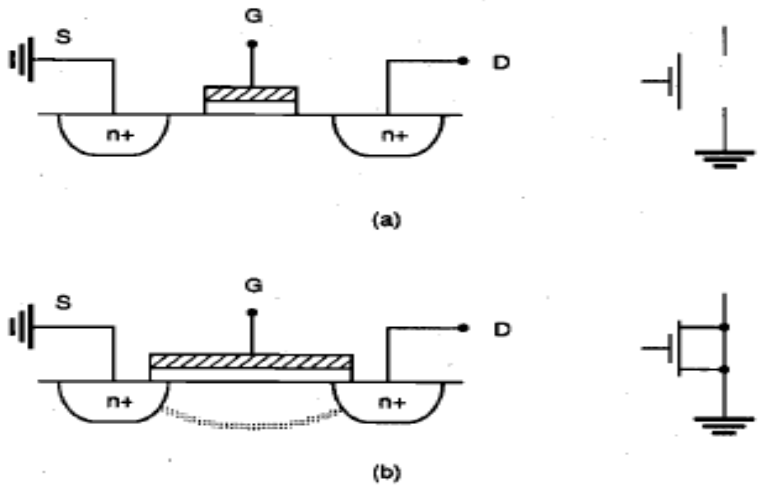
- Logical stuck-at-0 or stuck-at-l
- Slower transition (delay fault)
- AND-bridging, OR-bridging



NAND2, and inverter gates. In this circuit, the input line B can be stuck-at- 1 (s-a- 1), since some part of the input line is shorted to the power rail. The pMOS transistor of the first stage NOR2 gate is stuck-on due to a process problem that causes a short between its source and drain terminals. The top nMOS transistor in the NAND2 gate, on the other hand, is stuck-open due to either an incomplete contact (open) of the source or drain node or due to a large separation of drain or source diffusion from the gate, which causes permanent turn-off of the transistor regardless of the input C value. The stuck-on and stuck-open faults are elaborated on in Fig.The bridging fault between the output line of the inverter and the input line C can be due to a fabrication defect which causes a short between any two parts of the two lines. Although in the circuit diagram, these two lines are seemingly far apart, in the actual layout, some parts of these two lines can be close to each other. In such a layout, these two lines can be shorted due to underetching in the line patterning process.



- Improper estimation of on –chip interconnect delays and other timing considerations.
- Excessive variations in the fabrication process which cause significant variations in circuit delays and clock skews.
- Opens in metal lines connecting parallel transistors which make the effective transistor size much smaller.
- Ageing effects such as hot –carrier induced delay increase.



The task of detecting delay faults is even more subtle than detecting functional faults in steady state. The functional test is usually done at speeds lower than the target speed due to the limitations of the testers. Special clocking is used to apply delay tests on a slow tester. The fault models mentioned above are used in fault simulation aimed at

- Test generation,
- Construction of fault dictionaries, or
- Circuit analysis in the presence of faults.

Each fault dictionary stores the expected output response of every faulty circuit to a particular test vector corresponding to a particular simulated fault.

2. **Explain briefly about Controllability and observability. [AUC NOV 2011]**

Controllability : The ability to set nets, nodes, gate inputs or outputs or sequential elements to a known logic state.

Observability : The ability to observe nets, nodes, gate inputs or outputs or sequential elements to a known logic state.

The controllability of an internal circuit node within a chip is a measure of the ease of setting the node to a 1 or 0 state. This measure is of importance when assessing the degree of difficulty of testing a particular signal within a circuit. An easily controllable node would be directly settable via an input pad.

Controllability (C1) : Controllability C1 is the probability of a signal value on line l being set to 1 by a random vector .

Controllability (C0) : Controllability C0 is the probability of a signal value on line l being set to 0 by a random vector .

SCOAP (Sandia Controllability / Observability analysis program)

Combinational 0-controllability, CC0(n)

Combinational 1-controllability, CC1(n)

Combinational observability, C0(n)

Sequential 0-controllability, SC0(n)

Sequential 1-controllability, SC1(n)

Sequential observability, S0(n)

If a logic gate output is produced by setting only one input to a controlling value then

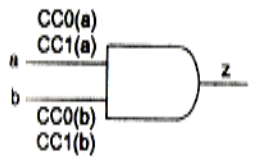
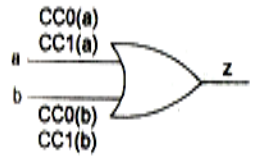
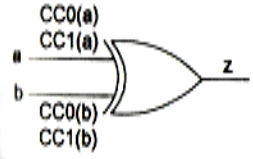
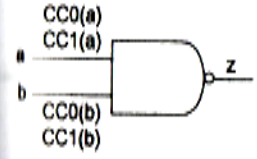
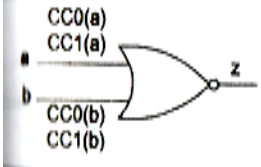
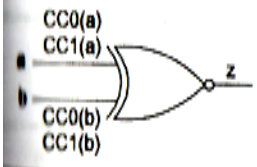
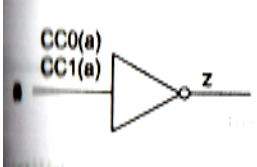
Output controllability =min(input controllabilities) +1

If a logic gate output can only be produced by setting all inputs to a non -controlling value then

Output controllability =∑(input controllabilities) +1

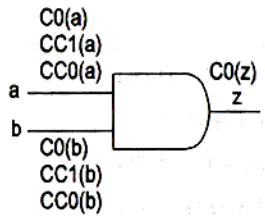
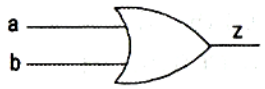
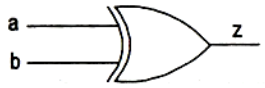
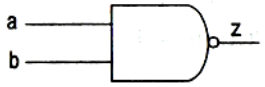
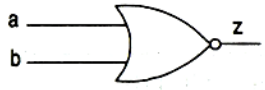
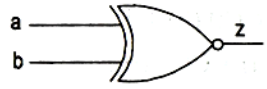
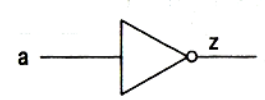
If an output can be controlled by a multiple input sets then

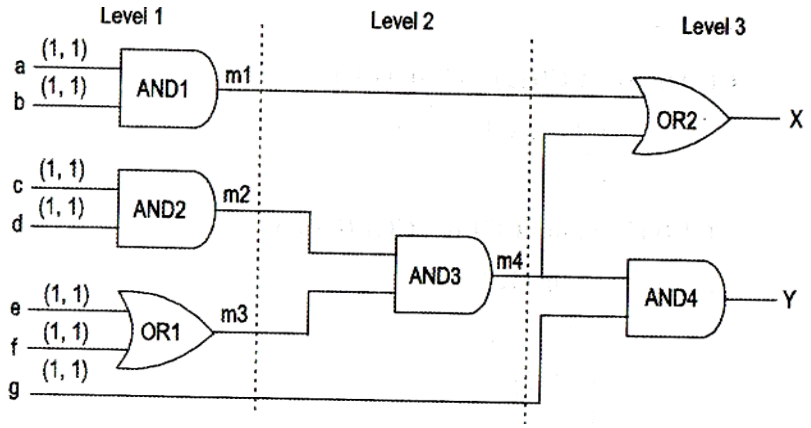
Output controllability =min(controllabilities of input sets) +1

Logic Gate	Truth Table	SCOAP Controllability Calculation
	$a\ b\ z$ 0 0 0 0 1 0 1 0 0 1 1 1	CC0 (z) = min (CC0 (a), CC0 (b)) + 1 CC1 (z) = CC1 (a) + CC1 (b) + 1
	$a\ b\ z$ 0 0 0 0 1 1 1 0 1 1 1 1	CC0 (z) = CC0 (a) + CC0 (b) + 1 CC1 (z) = min (CC1 (a), CC1 (b)) + 1
	$a\ b\ z$ 0 0 0 0 1 1 1 0 1 1 1 0	CC0 (z) = min (CC0 (a) + CC0 (b), CC1 (a) + CC1 (b)) + 1 CC1 (z) = min (CC0 (a) + CC1 (b), CC1 (a) + CC0 (b)) + 1
	$a\ b\ z$ 0 0 1 0 1 1 1 0 1 1 1 0	CC0 (z) = CC1 (a) + CC1 (b) + 1 CC1 (z) = min (CC0 (a), CC0 (b)) + 1
	$a\ b\ z$ 0 0 1 0 1 0 1 0 0 1 1 0	CC0 (z) = min (CC1 (a), CC1 (b)) + 1 CC1 (z) = CC0 (a) + CC0 (b) + 1
	$a\ b\ z$ 0 0 1 0 1 0 1 0 0 1 1 1	CC0 (z) = min (CC0 (a) + CC1 (b), CC1 (a) + CC0 (b) + 1) CC1 (z) = min (CC0 (a) + CC0 (b), CC1 (a) + CC1 (b) + 1)
	$a\ z$ 0 1 1 0	CC0 (z) = CC1 (a) + 1 CC1 (z) = CC0 (a) + 1

Observability

Observability =observability of the output + non controlling value +1

Logic Gates	Observability Calculation
	C0 (a) = C0 (z) + CC1 (b) + 1 C0 (b) = C0 (z) + CC1 (a) + 1
	C0 (a) = C0 (z) + CC0 (b) + 1 C0 (b) = C0 (z) + CC0 (a) + 1
	C0 (a) = C0 (z) + min (CC0(b), CC1 (b)) + 1 C0 (b) = C0 (z) + min (CC0 (a), CC1 (a)) + 1
	C0 (a) = C0 (z) + CC1 (b) + 1 C0 (b) = C0 (z) + CC1 (a) + 1
	C0 (a) = C0 (z) + CC0 (b) + 1 C0 (b) = C0 (z) + CC0 (a) + 1
	C0 (a) = C0 (z) + min (CC0 (b), CC1 (b)) + 1 C0 (b) = C0 (z) + min (CC0 (a), CC1 (a)) + 1
	C0 (a) = C0 (z) + 1



3. Explain briefly about Ad Hoc Testable Design Techniques [AUC NOV 2011]

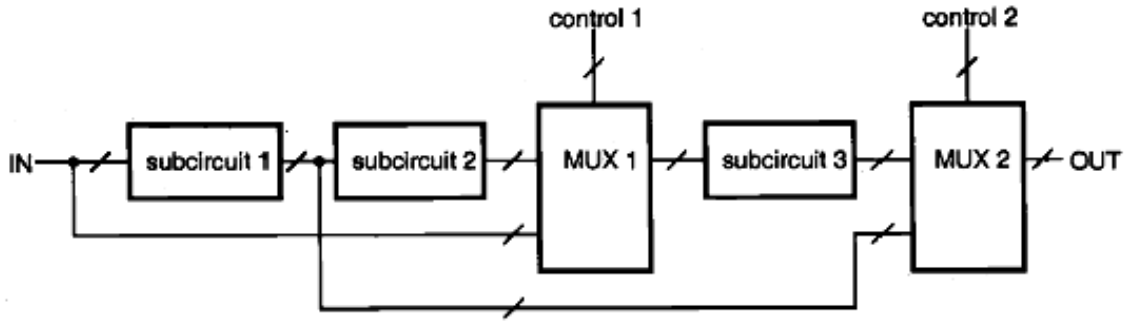
One way to increase the testability is to make nodes more accessible at some cost by physically inserting more access circuits to the original design. Listed below are some of the ad hoc testable design techniques.

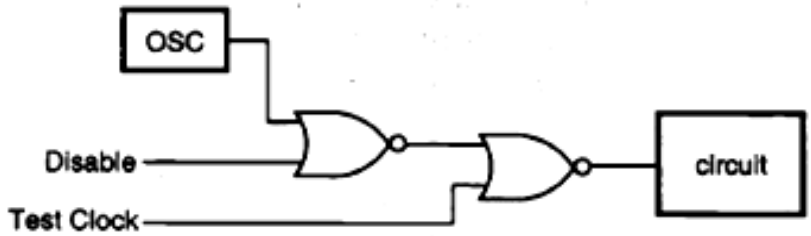
Partition-and-Mux Technique

Since the sequence of many serial gates, functional blocks, or large circuits are difficult to test, such circuits can be partitioned and multiplexors (muxes) can be inserted such that some of the primary inputs can be fed to partitioned parts through multiplexers with accessible control signals. With this design technique, the number of accessible nodes can be increased and the number of test patterns can be reduced. A case in point would be the 32-bit counter. Dividing this counter into two 16-bit parts would reduce the testing time in principle by a factor of 2^{15}. However, circuit partitioning and addition of multiplexers may increase the chip area and circuit delay. This practice is not unique and is similar to the divide-and-conquer approach to large, complex problems. Figure 16.5 illustrates this method.

Initialize Sequential Circuit

When the sequential circuit is powered up, its initial state can be a random, unknown state. In this case, it is not possible to start the test sequence correctly. The state of a sequential circuit can be brought to a known state through *initialization*. In many designs, the initialization can be easily done by connecting asynchronous preset or clear-input signals from primary or controllable inputs to flip-flops or latches.



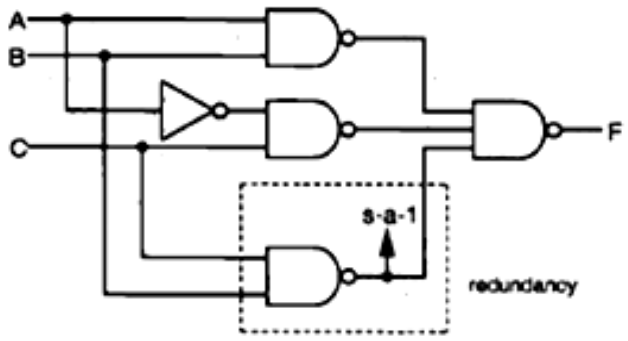


Avoid Asynchronous Logic and Redundant Logic

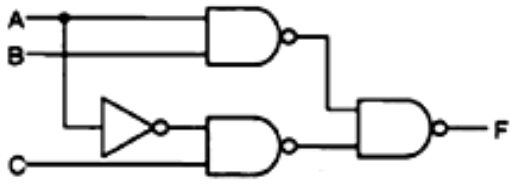
The enhancement of testability requires serious tradeoffs. The speed of an asynchronous logic circuit can be faster than that of the synchronous logic circuit counterpart. However, the design and test of an asynchronous logic circuit are more difficult than for a synchronous logic circuit, and its state transition times are difficult to predict. Also, the operation of an asynchronous logic circuit is sensitive to input test patterns, often causing race problems and hazards of having momentary signal values opposite to the expected values. Sometimes, designed-in logic redundancy is used to mask a static hazard condition for reliability. However, the redundant node cannot be observed since the primary output value cannot be made dependent on the value of the redundant node. Hence, certain faults on the redundant node cannot be tested or detected. Figure 16.7 shows that the bottom NAND2 gate is redundant and the stuck-at-1 fault on its output line cannot be detected. If a fault is undetectable, the associated line or gate can be removed without changing the logic function.

$F = AB + BC + \bar{A}C$

$= AB + \bar{A}C$



(a)



(b)

Disable Internal Oscillators and Clocks

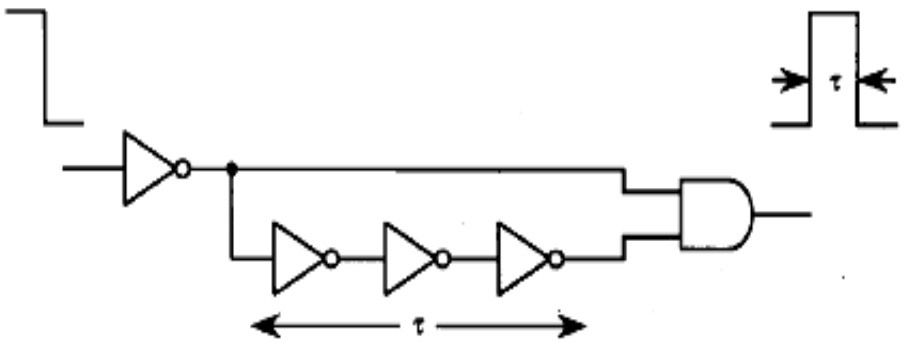
To avoid synchronization problems during testing, internal oscillators and clocks should be disabled. For example, rather than connecting the circuit directly to the on-chip oscillator, the clock signal can be ORed with a disabling signal followed by an insertion of a testing signal as shown in Fig. 16.6.

Although it is unessential to test redundant nodes when they are designed in as backup parts, either to enhance the circuit reliability or to increase the fabrication yield, the use of redundant circuits can make test generation much more complex and difficult. In fact, test generators, especially the random or deterministic test generators, would not be able to recognize such design intent. Some redundancy in circuits may be unintentional due to lack of design efficiency.

Avoid Delay-Dependent Logic

Chains of inverters can be used to design in delay times and use AND operation of their outputs along with inputs to generate pulses, as shown in Fig. 16.8.

Most automatic test pattern generation (ATPG) programs do not include logic delays to minimize the complexity of the program. As a result, such delay-dependent logic is viewed as redundant combinational logic, and the output of the reconvergent gate is always set to logic 0, which is not correct. Thus, the use of delay-dependent logic should be avoided in design for testability.

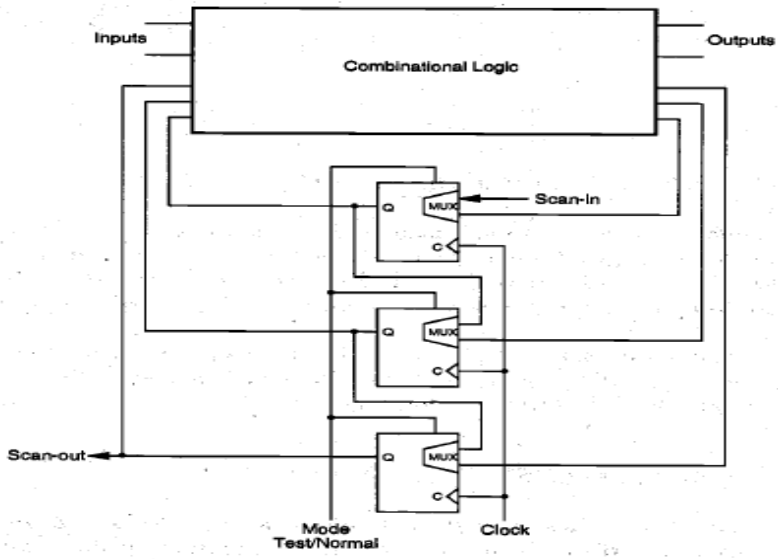


As discussed earlier, the controllability and observability can be enhanced by providing more accessible logic nodes with use of additional primary input lines and multiplexors. However, the use of additional I/O pins can be costly not only for chip fabrication but also for packaging. A popular alternative is to use scan registers with both shift and parallel load capabilities. The scan design technique is a structured approach to design sequential circuits for testability. The storage cells in registers are used as observation points, control points, or both. By using the scan design techniques, the testing of a sequential circuit is reduced to the problem of testing a combinational circuit.

In general, a sequential circuit consists of a combinational circuit and some storage elements. In the scan-based design, the storage elements are connected to form a long serial shift register, the so-called *scan path*, by using multiplexors and a mode (test/ normal) control signal, as shown in Fig. 16.9.

In the test mode, the scan-in signal is clocked into the scan path, and the output of the last stage latch is scanned out. In the normal mode, the scan-in path is disabled and the circuit functions as a sequential circuit. The testing sequence is as follows:

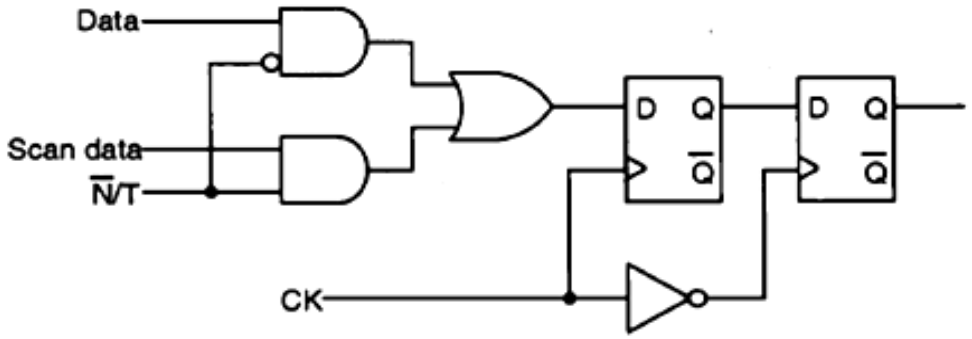
Step 1: Set the mode to *test* and let latches accept data from scan-in input.
Step 2: Verify the scan path by shifting in and out the test data.
Step 3: Scan in (shift in) the desired state vector into the shift register.
Step 4: Apply the test pattern to the primary input pins.
Step 5: Set the mode to *normal* and observe the primary outputs of the circuit after sufficient time for propagation.
Step 6: Assert the circuit clock for one machine cycle to capture the outputs of the combinational logic into the registers.
Step 7: Return to *test* mode; scan out the contents of the registers, and at the same time scan in the next pattern.
Step 8: Repeat steps 3-7 until all test patterns are applied.



The storage cells in scan design can be implemented using edge-triggered D flip-flops, master-slave flip-flops, or level-sensitive latches controlled by complementary clock signals to ensure race-free operation.

two separate clocks, one for normal operation and one for shift operation, are used. Since the shift operation does not have to be performed at the target speed, its clock is much less constrained.

An important approach among scan-based designs is the level sensitive scan design (LSSD), which incorporates both the level sensitivity and the scan path approach using shift registers. The level sensitivity is to ensure that the sequential circuit response is independent of the transient characteristics of the circuit, such as the component and wire delays. Thus, LSSD removes hazards and races. Its ATPG is also simplified since tests have to be generated only for the combinational part of the circuit.



The boundary scan test method is also used for testing printed circuit boards (PCBs) and multichip modules (MCMs) carrying multiple chips. Shift registers are placed in each chip close to I/O pins in order to form a chain around the board for testing. With successful implementation of the boundary scan method, a simpler tester can be used for PCB testing.

On the negative side, scan design uses more complex latches, flip-flops, I/O pins, and interconnect wires and, thus, requires more chip area. The testing time per test pattern is also increased due to shift time in long registers.

4. **Explain briefly about Built –In Self Test (BIST) Techniques.[AUC May 2011]**

Pseudo Random Pattern Generator

To test the circuit, test patterns first have to be generated either by using a pseudo random pattern generator, a weighted test generator, an adaptive test generator, or other means. A pseudo random test generator circuit can use an LFSR, as shown in Fig. 16.12.

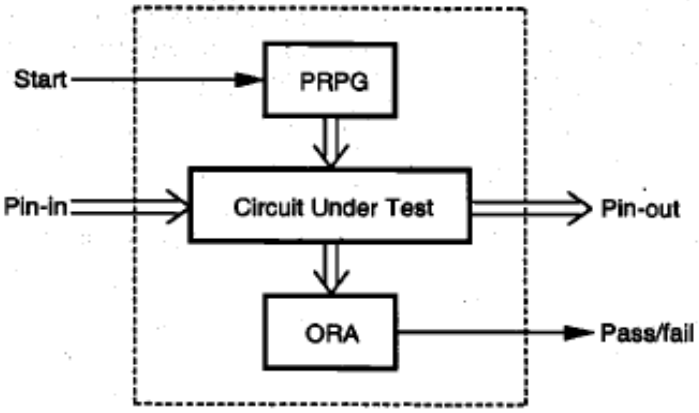


Figure 16.11. A procedure for BIST.

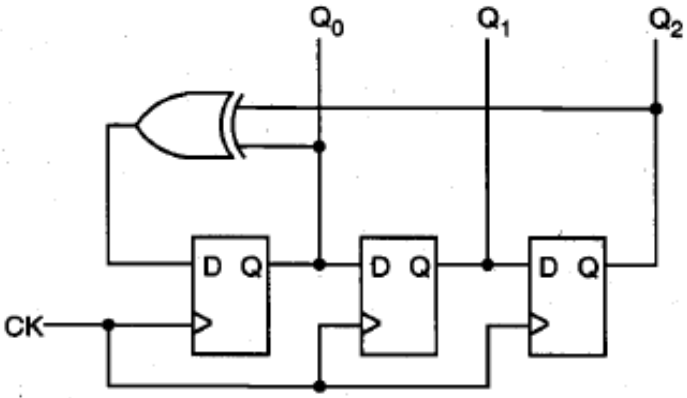


Figure 16.12. A pseudo-random sequence generator using LFSR.

Linear Feedback Shift Register as an ORA

To reduce the chip area penalty, data compression schemes are used to compare the compacted test responses instead of the entire raw test data. One of the popular data compression schemes is the *signature analysis*, which is based on the concept of cyclic redundancy checking. It uses polynomial division, which divides the polynomial representation of the test output data by a characteristic polynomial and then finds the remainder as the signature. The signature is then compared with the expected signature to determine whether the device under test is faulty. It is known that compression can

cause some loss of fault coverage. It is possible that the output of a faulty circuit can match the output of the fault-free circuit; thus, the fault can go undetected in the signature analysis. Such a phenomenon is called *aliasing*.

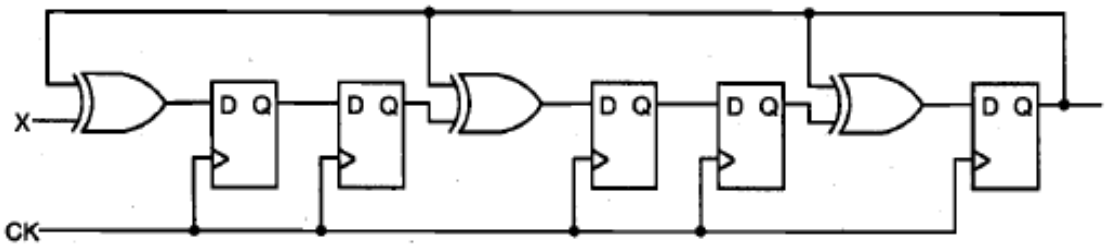
In its simplest form, the signature generator consists of a single-input linear feedback shift register (LFSR), as shown in Fig. 16.13, in which all the latches are edge-triggered. In this case, the signature is the content of this register after the last input bit has been sampled. The input sequence $\{a_n\}$ is represented by polynomial $G(x)$ and the output sequence by $Q(x)$. It can be shown that $G(x) = Q(x)\ P(x) + R(x)$, where $P(x)$ is the characteristic polynomial of LFSR and $R(x)$ is the remainder, the degree of which is lower than that of $P(x)$. For the simple case in Fig. 16.13, the characteristic polynomial is

$$P(x) = 1 + x^2 + x^4 + x^5$$

For the 8-bit input sequence {1 1 1 1 0 1 0 1}, the corresponding input polynomial is

$$G(x) = x^7 + x^6 + x^5 + x^4 + x^2 + 1$$

and the remainder term becomes $R(x) = x^4 + x^2$, which corresponds to the register contents of {0 0 1 0 1}.

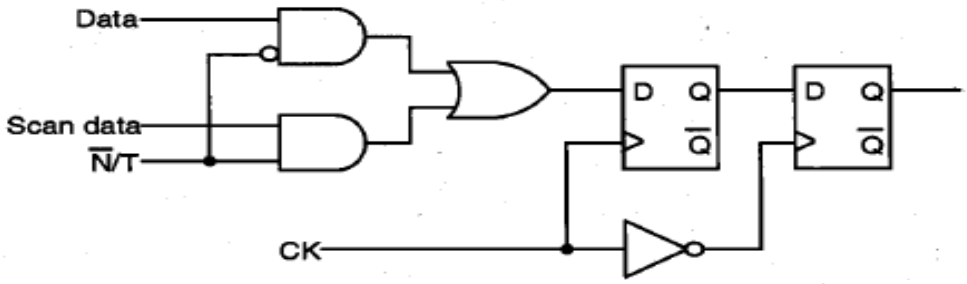


Current Monitoring IDDQ Test

An often-used technique for testing fabrication defects is the I_{DDQ} test. Under a bridging fault, the static currents drawn from the power supply in CMOS circuits can be noticeably high, well beyond the expected range of leakage currents. For example, if the drain node

two separate clocks, one for normal operation and one for shift operation, are used. Since the shift operation does not have to be performed at the target speed, its clock is much less constrained.

An important approach among scan-based designs is the level sensitive scan design (LSSD), which incorporates both the level sensitivity and the scan path approach using shift registers. The level sensitivity is to ensure that the sequential circuit response is independent of the transient characteristics of the circuit, such as the component and wire delays. Thus, LSSD removes hazards and races. Its ATPG is also simplified since tests have to be generated only for the combinational part of the circuit.

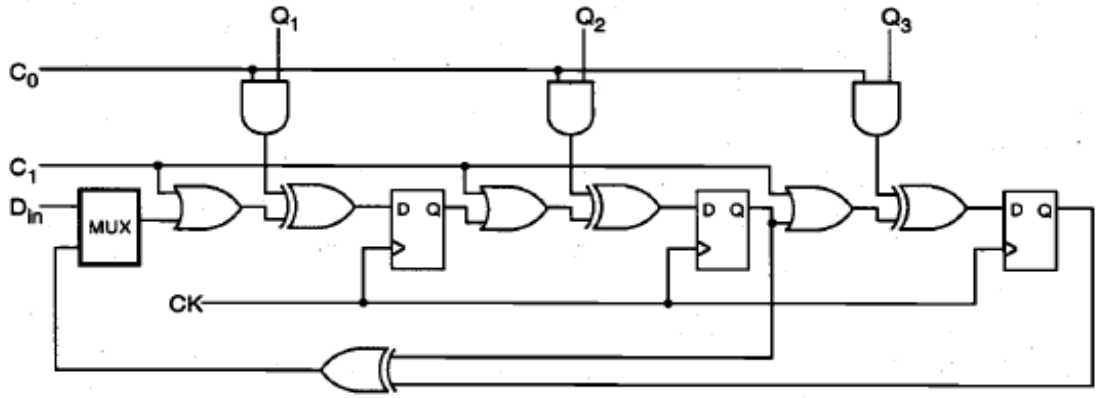


The boundary scan test method is also used for testing printed circuit boards (PCBs) and multichip modules (MCMs) carrying multiple chips. Shift registers are placed in each chip close to I/O pins in order to form a chain around the board for testing. With successful implementation of the boundary scan method, a simpler tester can be used for PCB testing.

On the negative side, scan design uses more complex latches, flip-flops, I/O pins, and interconnect wires and, thus, requires more chip area. The testing time per test pattern is also increased due to shift time in long registers.

Built-In Logic Block Observer

The built-in logic block observer (BILBO) register is a form of ORA which can be used in each cluster of partitioned registers. A basic BILBO circuit is shown in Fig. 16.14, which allows four different modes controlled by C_0 and C_1 signals.



C_0	C_1	Mode
0	0	linear shift
1	0	signature analysis
1	1	data (complemented) latch
0	1	reset

The BILBO operation allows monitoring of circuit operation through exclusive-ORing into LFSR at multiple points, which corresponds to the signature analyzer with multiple inputs.

5. Write a note on boundary scan test . [AUC APR 2008 ,NOV 2011]

Boundary Scan Test (BST)

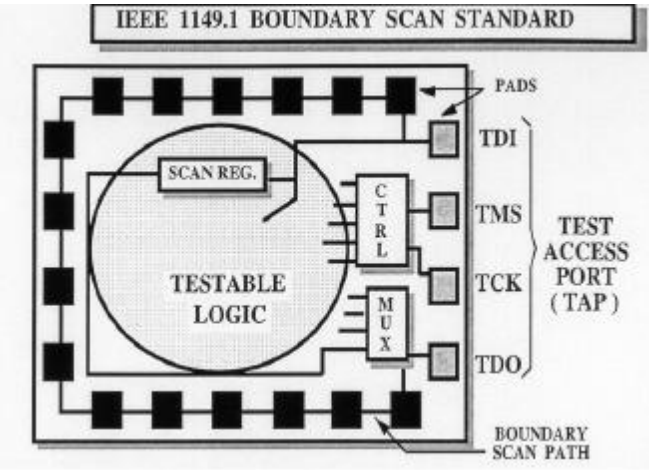
Boundary Scan Test (BST) is a technique involving scan path and self-testing techniques to resolve the problem of testing boards carrying VLSI integrated circuits and/or surface mounted devices (SMD).

Printed circuit boards (PCB) are becoming very dense and complex, especially with SMD circuits, that most test equipment cannot guarantee good fault coverage.

BST (figure 8.15) consists in placing a scan path (shift register) adjacent to each component pin and to interconnect the cells in order to form a chain around the border of the circuit. The BST circuits contained on one board are then connected together to form a single path through the board.

The boundary scan path is provided with serial input and output pads and appropriate clock pads which make it possible to:

Test the interconnections between the various chip
Deliver test data to the chips on board for self-testing
Test the chips themselves with internal self-test

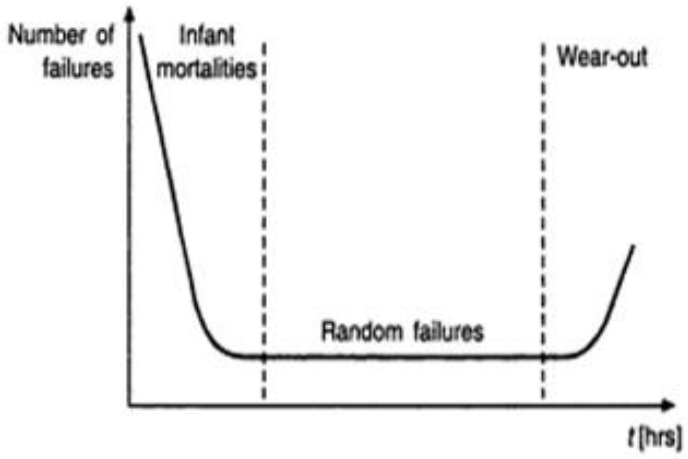


The advantages of Boundary scan techniques are as follows :

- No need for complex testers in PCB testing
- Test engineers work is simplified and more efficient
- Time to spend on test pattern generation and application is reduced
- Fault coverage is greatly increased.

6. Explain briefly about the reliability in CMOS testing .

Reliability is concerned with projecting the lifetime of a component once it is placed into operation. The projected lifetime is the number of hours of operation that can be expected before a failure occurs. This is described graphically using a bathtub curve as shown in Figure which plots the number of failures of a given system as a function of time.



Bathtub curves are semilog plots with time t in hours, and points plotted as $\log_{10}(t)$. There are three general regions shown in the curve. Infant mortalities are failures that occur after a very short period of time, i.e., early in the life of the system. These tend to arise from manufacturing defects that manifest themselves after a few hours of operation. The central portion of the curve represents random failures during normal operation, while wear-out describes the end of life.

Reliability Modelling

Reliability data can be used to construct mathematical models that are useful in predigesting failure rates. Suppose that we test a group of devices starting at time $t = 0$. As time progresses, a device fails at time t_1, then end at time t_2, and so on. At the end of the test period, we find that N devices fail. The total number of operational hours for the group is the sum

$$T = \sum_{i=1}^{N} t_i$$

where T is measured in hours.

The average failure rate of the test group is defined as the number of failures divided by the total number of operational hours.

$$\lambda_{av} = \frac{N}{T}$$

The mean time to failure (MTTF) is given by

$$\text{MTTF} = \frac{T}{N} = \frac{1}{\lambda_{av}}$$

MTTF represents an average lifetime for the test group.

One of the most commonly used metric to describe failure rates is the failure in time (FIT). It is defined as

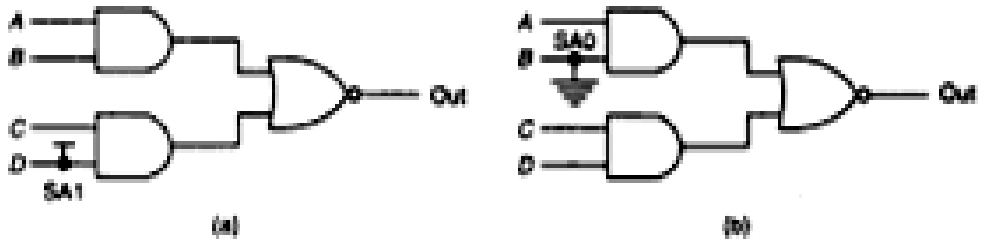
1 FIT = 1 failure in 1 million parts over 1000 hours

or 1 FIT = 1 ppm/k

7. Explain briefly about manufacturing test principles.

STUCK AT Faults

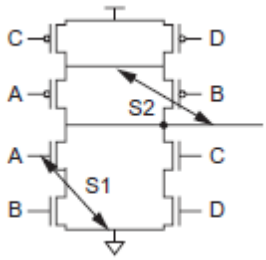
These faults occur when a node is accidently connected to the power supply(SA1) or ground (SA0)



Stuck at 1 fault Stuck at 0 fault

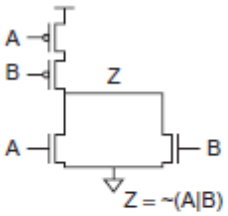
Short circuit and open circuit fault

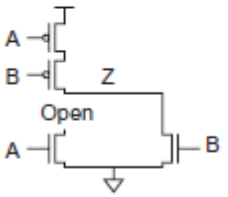
The short *S*1 results in an S-A-0 fault at input A, while short *S*2 modifies the function of the gate.



2-input NOR gate in which one of the transistors is rendered ineffective. If nMOS transistor *A* is stuck open, then the function displayed by the gate will be

$$Z = \overline{A + B} + \overline{B}Z'$$





Z = ~(A|B) | (~B&Z')

Controllability : The ability to set nets, nodes, gate inputs or outputs or sequential elements to a known logic state.

Observability : The ability to observe nets, nodes, gate inputs or outputs or sequential elements to a known logic state.

The controllability of an internal circuit node within a chip is a measure of the ease of setting the node to a 1 or 0 state. This measure is of importance when assessing the degree of difficulty of testing a particular signal within a circuit. An easily controllable node would be directly settable via an input pad.

Controllability (C1) : Controllability C1 is the probability of a signal value on line l being set to 1 by a random vector .

Controllability (C0) : Controllability C0 is the probability of a signal value on line l being set to 0 by a random vector .

SCOAP (Sandia Controllability / Observability analysis program)

Combinational 0-controllability, CC0(n)

Combinational 1-controllability, CC1(n)

Combinational observability, C0(n)

Sequential 0-controllability, SC0(n)

Sequential 1-controllability, SC1(n)

Sequential observability, S0(n)

If a logic gate output is produced by setting only one input to a controlling value then

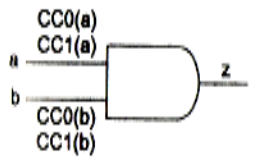
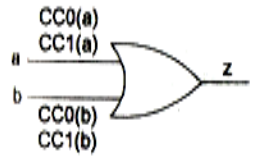
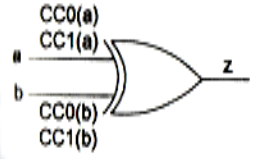
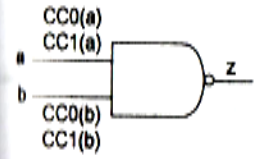
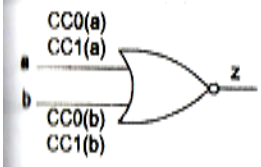
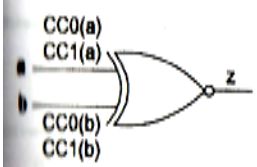
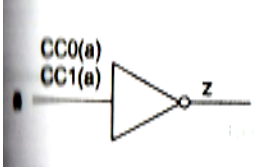
Output controllability =min(input controllabilities) +1

If a logic gate output can only be produced by setting all inputs to a non -controlling value then

Output controllability =∑(input controllabilities) +1

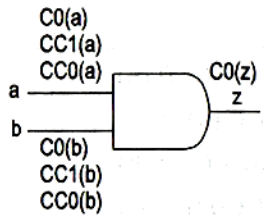
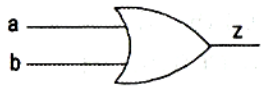
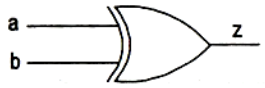
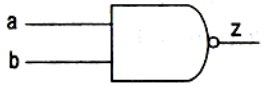
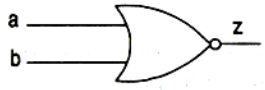
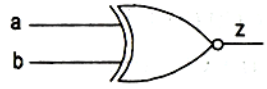
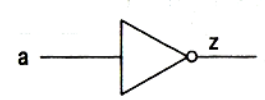
If an output can be controlled by a multiple input sets then

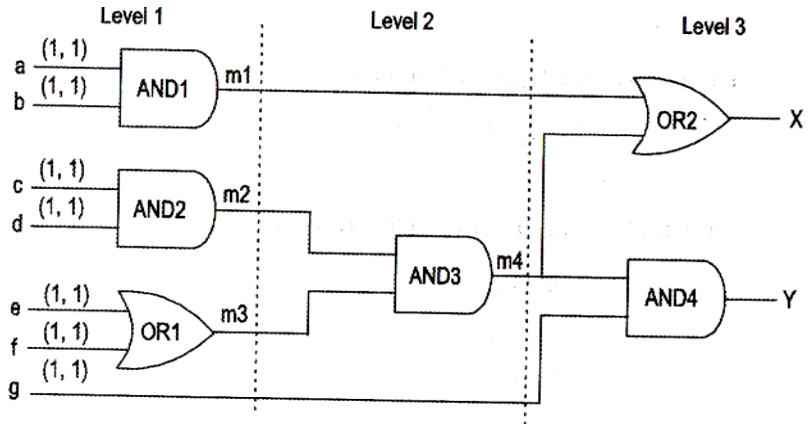
Output controllability =min(controllabilities of input sets) +1

Logic Gate	Truth Table	SCOAP Controllability Calculation
	$a\ b\ z$ 0 0 0 0 1 0 1 0 0 1 1 1	CC0 (z) = min (CCO (a), CC0 (b)) + 1 CC1 (z) = CC1 (a) + CC1 (b) + 1
	$a\ b\ z$ 0 0 0 0 1 1 1 0 1 1 1 1	CC0 (z) = CC0 (a) + CC0 (b) + 1 CC1 (z) = min (CC1 (a), CC1 (b)) + 1
	$a\ b\ z$ 0 0 0 0 1 1 1 0 1 1 1 0	CC0 (z) = min (CC0 (a) + CC0 (b), CC1 (a) + CC1 (b)) + 1 CC1 (z) = min (CC0 (a) + CC1 (b), CC1 (a) + CC0 (b)) + 1
	$a\ b\ z$ 0 0 1 0 1 1 1 0 1 1 1 0	CC0 (z) = CC1 (a) + CC1 (b) + 1 CC1 (z) = min (CC0 (a), CC0 (b)) + 1
	$a\ b\ z$ 0 0 1 0 1 0 1 0 0 1 1 0	CC0 (z) = min (CC1 (a), CC1 (b)) + 1 CC1 (z) = CC0 (a) + CC0 (b) + 1
	$a\ b\ z$ 0 0 1 0 1 0 1 0 0 1 1 1	CC0 (z) = min (CC0 (a) + CC1 (b), CC1 (a) + CC0 (b) + 1) CC1 (z) = min (CC0 (a) + CC0 (b), CC1 (a) + CC1 (b) + 1)
	$a\ z$ 0 1 1 0	CC0 (z) = CC1 (a) + 1 CC1 (z) = CC0 (a) + 1

Observability

Observability =observability of the output + non controlling value +1

Logic Gates	Observability Calculation
	C0 (a) = C0 (z) + CC1 (b) + 1 C0 (b) = C0 (z) + CC1 (a) + 1
	C0 (a) = C0 (z) + CC0 (b) + 1 C0 (b) = C0 (z) + CC0 (a) + 1
	C0 (a) = C0 (z) + min (CC0(b), CC1 (b)) + 1 C0 (b) = C0 (z) + min (CC0 (a), CC1 (a)) + 1
	C0 (a) = C0 (z) + CC1 (b) + 1 C0 (b) = C0 (z) + CC1 (a) + 1
	C0 (a) = C0 (z) + CC0 (b) + 1 C0 (b) = C0 (z) + CC0 (a) + 1
	C0 (a) = C0 (z) + min (CC0 (b), CC1 (b)) + 1 C0 (b) = C0 (z) + min (CC0 (a), CC1 (a)) + 1
	C0 (a) = C0 (z) + 1



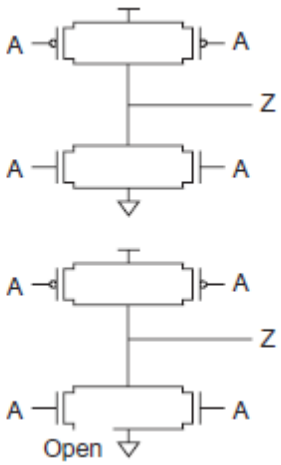
Fault Coverage

A measure of goodness of a set of test vectors is the amount of *fault coverage* it achieves. Each circuit node is taken in sequence and held to 0 (S-A-0), and the circuit is simulated with the test vectors comparing the chip outputs with a *known good machine*—a circuit with no nodes artificially set to 0 (or 1). When a discrepancy is detected between the *faulty machine* and the good machine, the fault is marked as detected and the simulation is stopped. This is repeated for setting the node to 1 (S-A-1). In turn, every node is stuck (artificially) at 1 and 0 sequentially. The fault coverage of a set of test vectors is the percentage of the total nodes that can be detected as faulty when the vectors are applied.

Automatic Test Pattern Generation (ATPG)

Improves the observability of a system and can reduce the number of test vectors required to achieve a desired fault coverage.

Delay Fault Testing



An inverter gate composed of paralleled nMOS and pMOS transistors. If an open circuit occurs in one of the nMOS transistor source connections to GND, then the gate would still function but with increased *tpdf*. In addition, the fault now becomes sequential as the detection of the fault depends on the previous state of the gate.
Delay faults may be caused by crosstalk Delay faults can also occur more often in SOI logic through the history effect. Software has been developed to model the effect of delay faults and is becoming more important as a failure mode as processes scale.

Digital VLSI Design

Lecture 9: I/O and Pad Ring

Semester A, 2016-17

Lecturer: Dr. Adam Teman

January 22, 2017

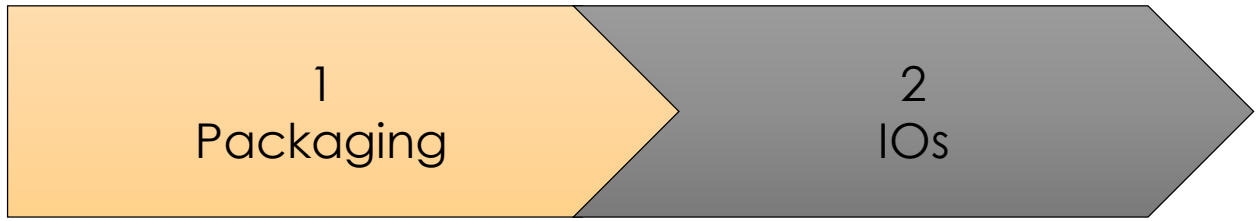
Emerging Nanoscaled Integrated Circuits and Systems Labs

Tradition of Excellence

Bar-Ilan University
אוניברסיטת בר-אילן

Disclaimer: This course was prepared, in its entirety, by Adam Teman. Many materials were copied from sources freely available on the internet. When possible, these sources have been cited; however, some references may have been cited incorrectly or overlooked. If you feel that a picture, graph, or code example has been copied from you and either needs to be cited or removed, please feel free to email adam.teman@biu.ac.il and I will address this as soon as possible.

www.Jntufastupdates.com

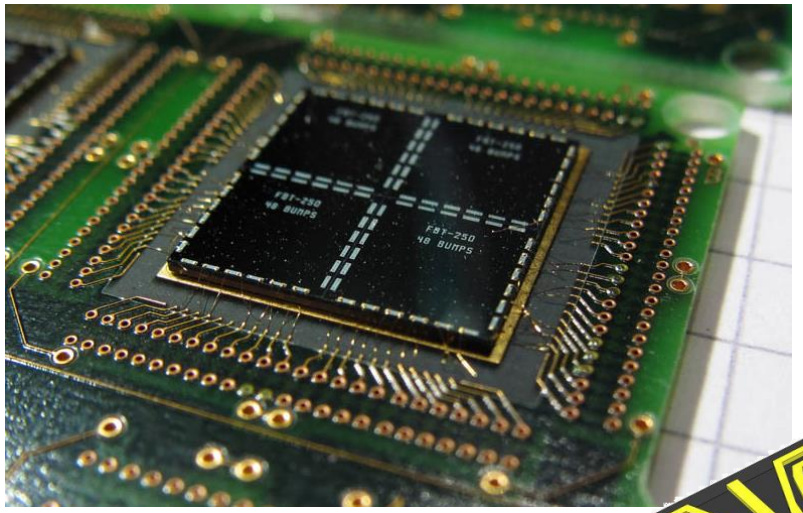


A bit about Packaging

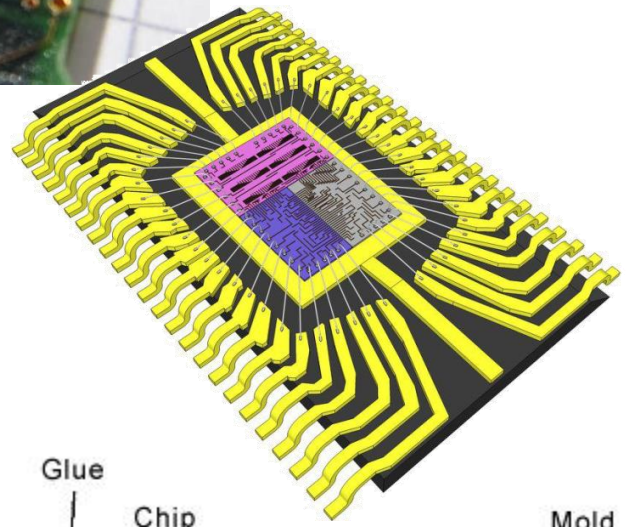
www.Jntufastupdates.com

How do we get outside the chip?

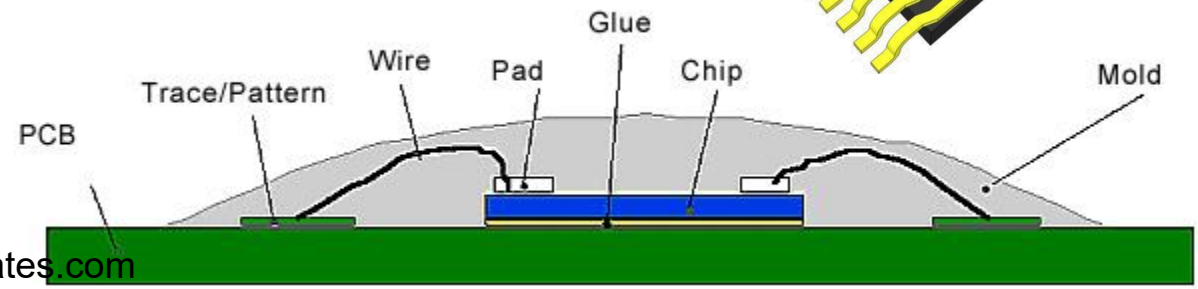
- **It's actually a pretty long road…**
  - I/O Circuits
  - Bonding
  - Package
  - Board



- **Once we get out of the chip**
  - Long wires mean a lot of delay, capacitance, inductance.
  - We can use fat wires for low resistance.
  - But we have a lot more room to play around.



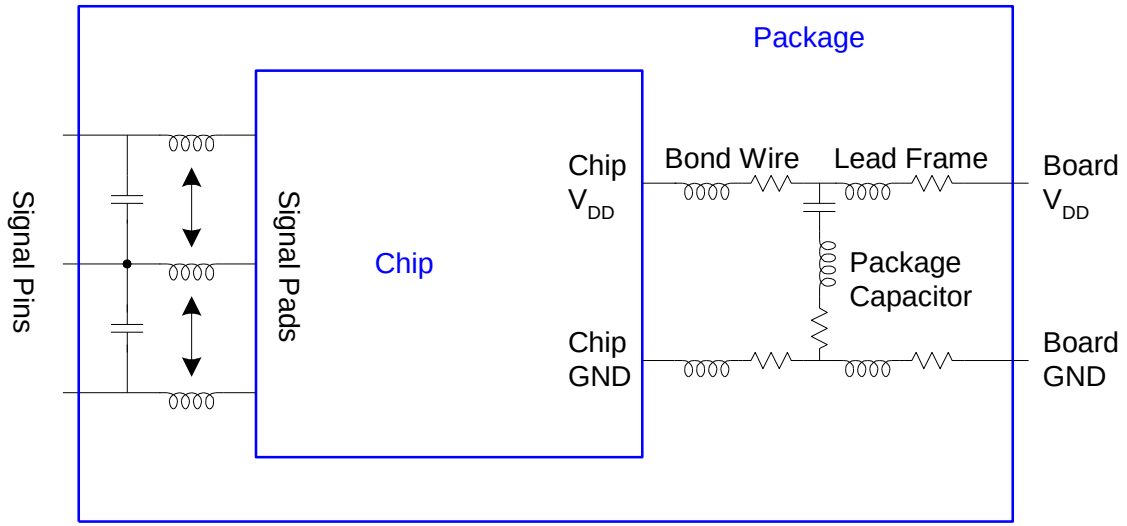
- **The interface between the chip and the outer world is the IC package.**



www.Jntufastupdates.com

Main Properties of Package

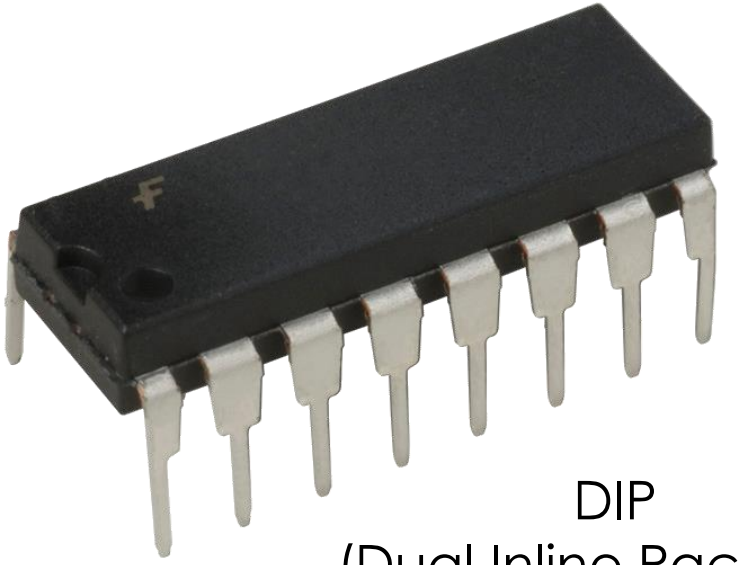
- **The package provides the physical, temperature and electrical protection.**
  - Electrical connection from chip to board
  - Physical connection from chip to board
  - Protection from high voltages (outside)
  - Physical protection
  - Thermo isolation



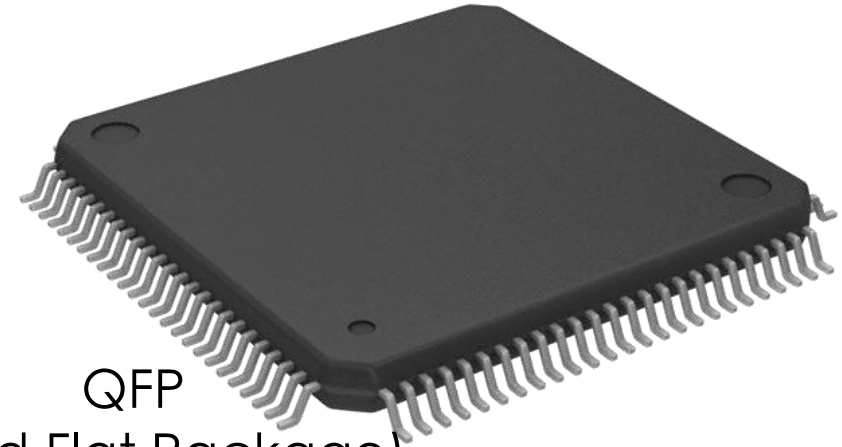
- **Requirements of a package are:**
  - **Electrical**: Capacitance, Resistance, Inductance, Impedance Tuning
  - **Interface**: A large number of I/O pins
  - **Mechanical**: Die/Bond protection, Compatibility with PCB
  - **Thermal**: Heat Removal
  - **Cost**: As low cost as possible (without fan, heat sink, etc.)

www.Jntufastupdates.com

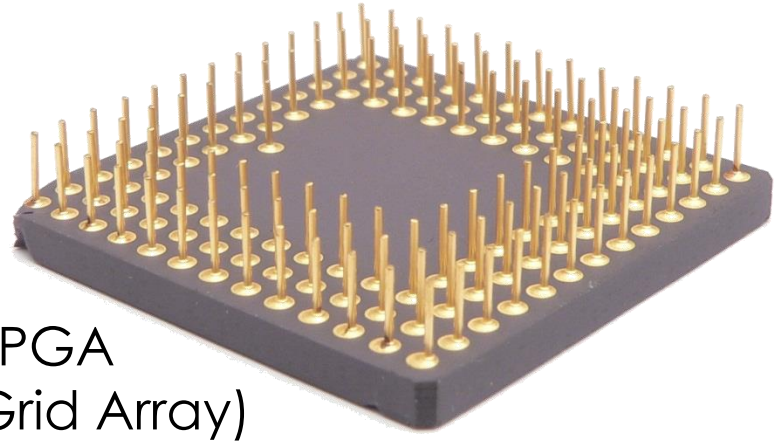
Package to Board Connection



DIP
(Dual Inline Package)



QFP
(Quad Flat Package)



PGA
(Pin Grid Array)



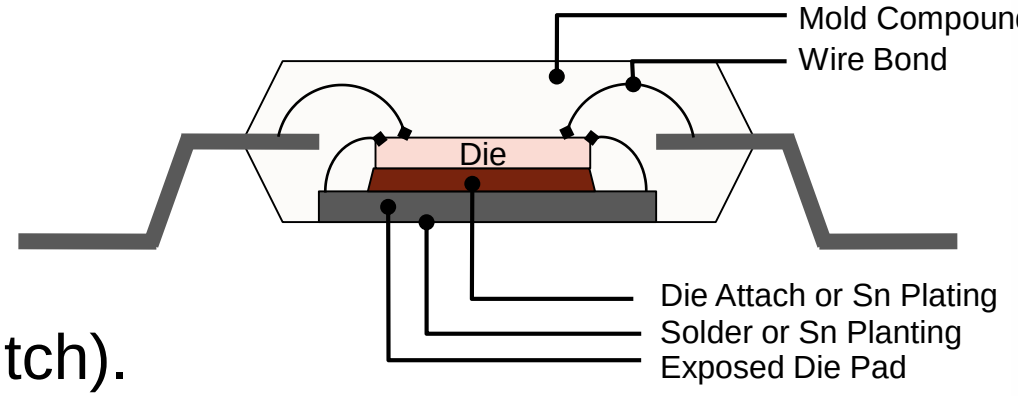
BGA (Ball Grid Array)

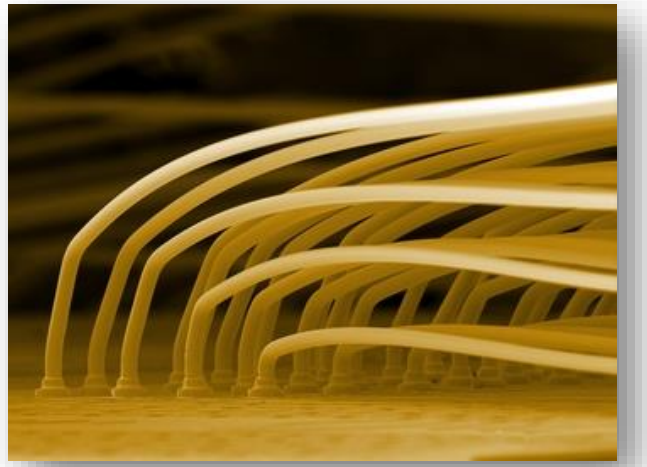
www.Jntufastupdates.com

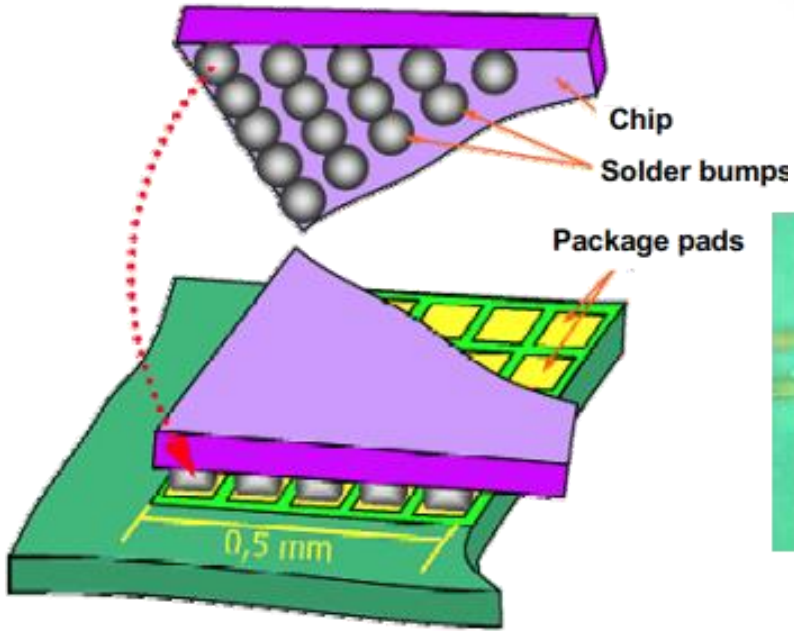
IC to Package Connection

Two main approaches:

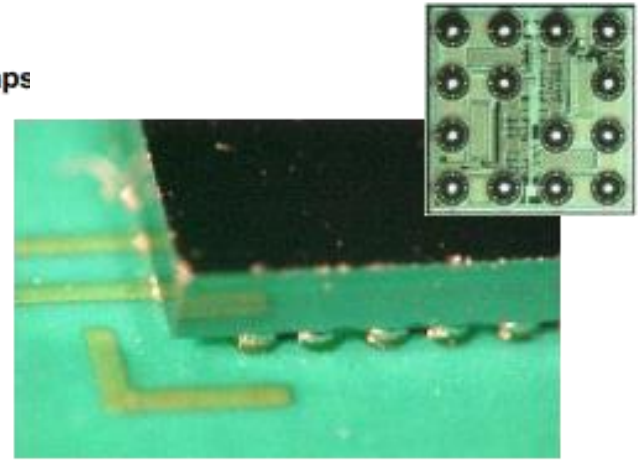
- **Wire bonding**
  - All pads are around chip edges (~100um pitch).
  - Slow, serial bonding process.
  - Long, high RLC wires (~5nH, 1pF per wire).
- **Flip Chip**
  - Pads on top of IC core.
  - High pin count.
  - Short, low RLC bonds (0.1nH)
  - Fast parallel bonding process.
  - But… Expensive!



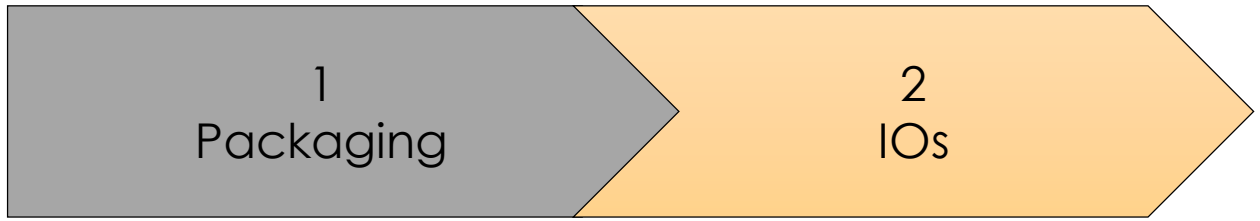




Controlled Collapse Chip Connection, C4



www.ultrafastupdates.com

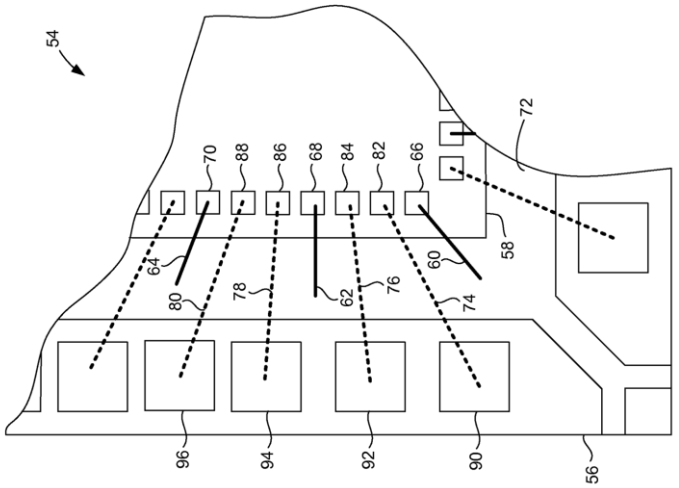


Input/Output Circuits (I/Os)

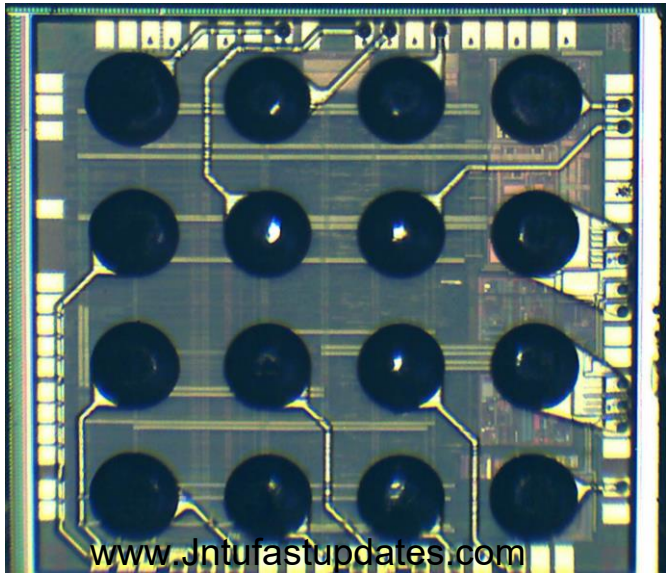
www.Jntufastupdates.com

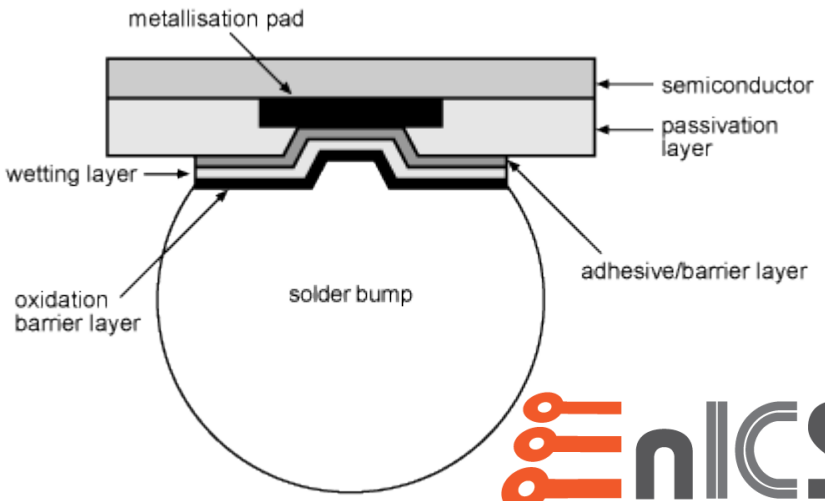
So how do we interface to the package?

- **We need to create a physical connection to the bonding wire.**
- **For wire bond packaging:**
  - Use a landing pad.
  - Basically a big (100um X 100um) piece of metal.
  - Many stacking layers for physical robustness.



- **For flip chip packaging:**
  - Use solder bumps.
  - Route to bumps with Redistribution Layer (RDL)





www.Jntufastupdates.com

But what connects to the bonding pads?

I/O Circuits!

- **Requirements of I/O Circuits:**
  - Availability to drive big loads
    - Due to package and transmission lines
  - Voltage Consistency
    - Due to different supply voltages on the board
  - Low switching noise
    - Due to package and transmission line inductance
  - ESD protection
    - Due to high potential difference of external devices

Goals of I/O Design:

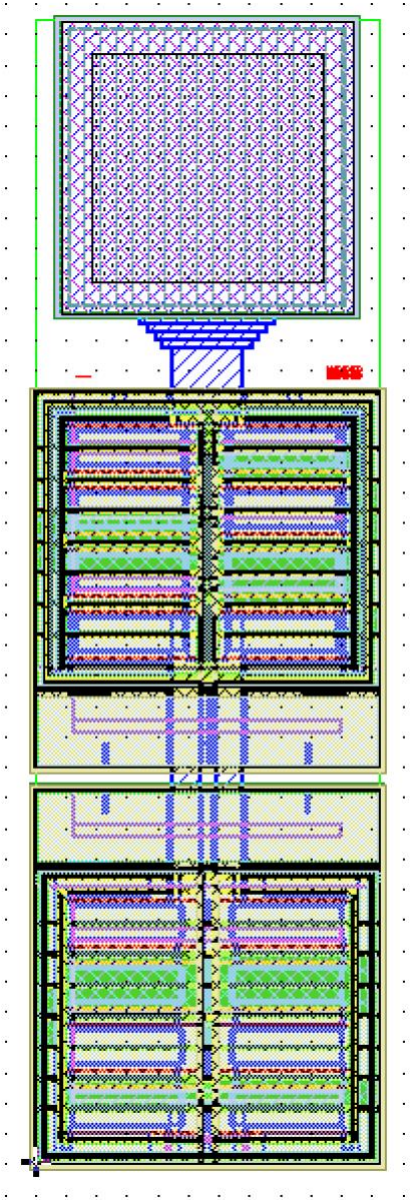
- Reduce delay to and from outside world (PCB)
- High drive current capability
- Match impedance to load
- ESD Protection
- Level shifting of voltages (i.e.1.2V inside/3.3V outside)
- Meet specifications of Interfaces
- Reduce power (short circuit current through output buffers)
- High voltage tolerance

www.Jntufastupdates.com

Types of I/O Cells

There are several types of basic I/O cells:

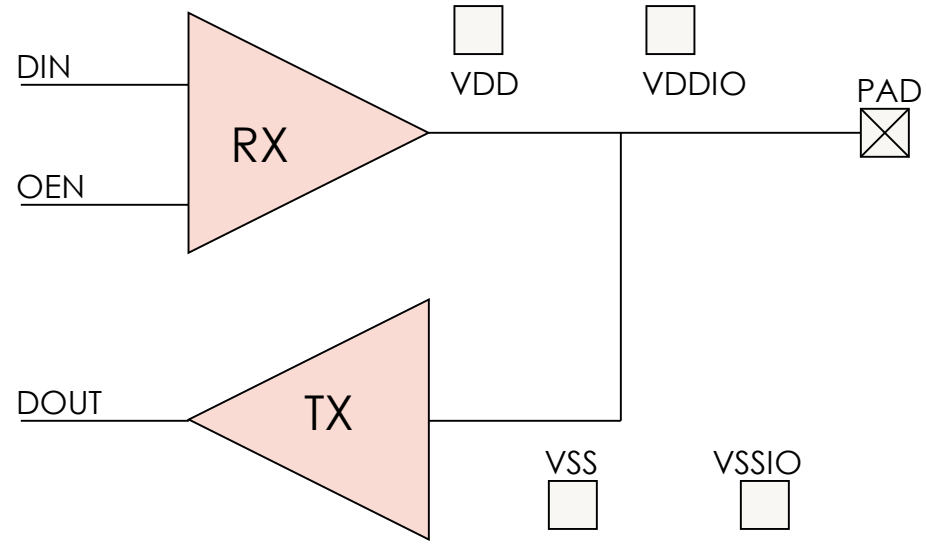
- **Digital I/O Buffers**
  - Provide high drive up-level shifting output
  - Provide down-level shifting and ESD protection for inputs
- **Analog I/O Cells**
  - Provide ESD protected analog inputs/outputs
- **Power supplies**
  - Provide power to the I/O and Core supplies
  - Provide the basis for ESD protection

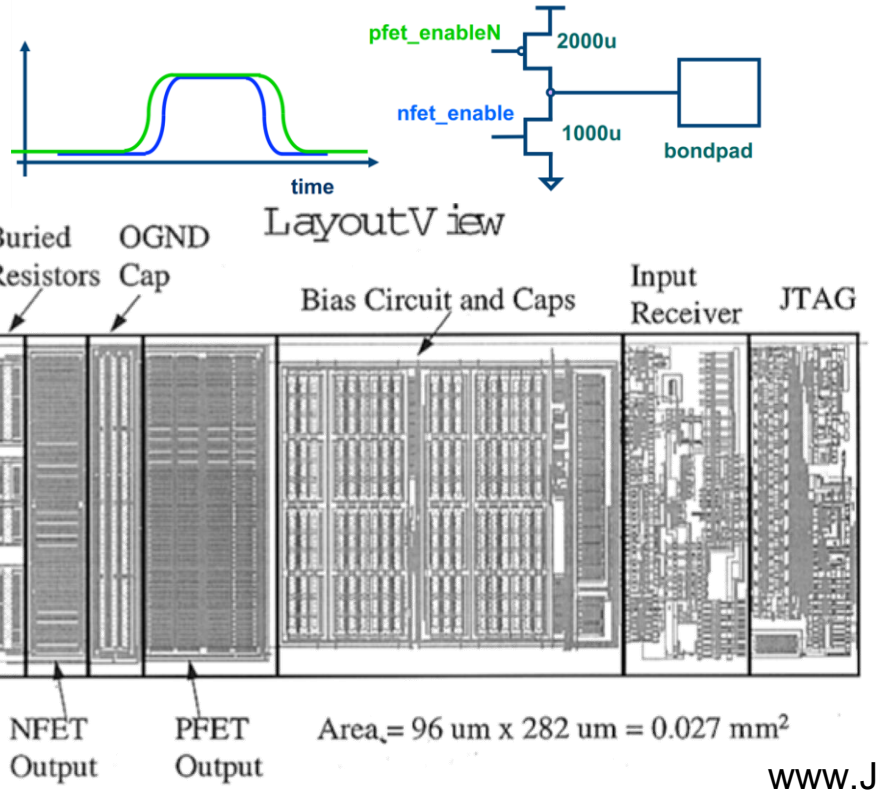


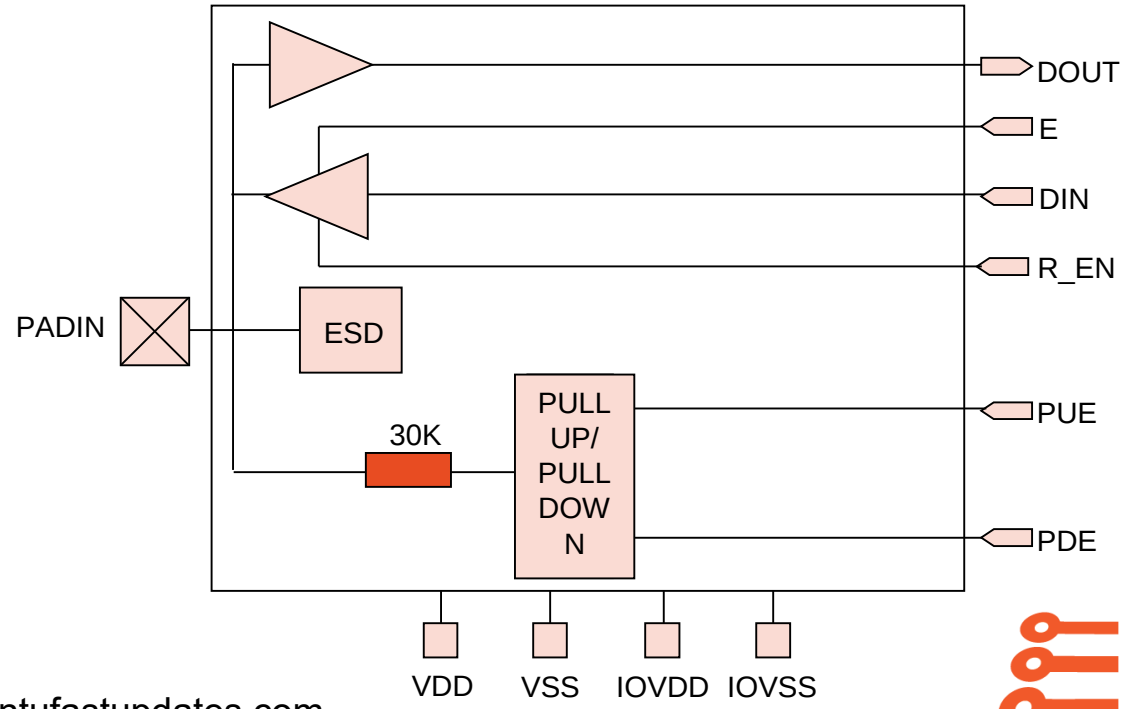
www.Jntufastupdates.com

Digital I/O Buffer

- **Digital I/O**
  - Output buffer needs to drive pF, not fF
  - Requires increasing fanout inverter chain
  - Short circuit current is unacceptable!

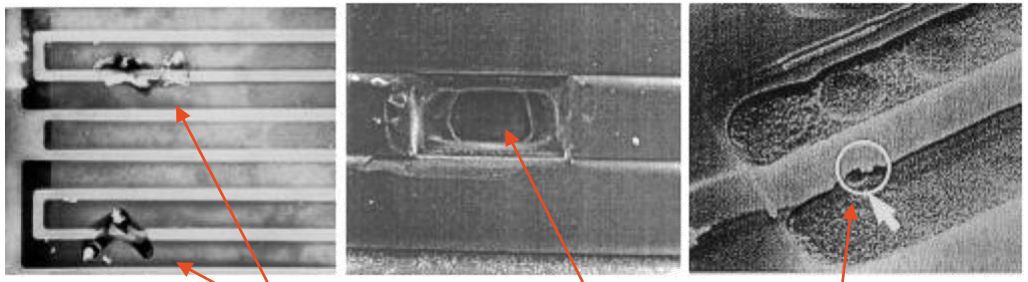




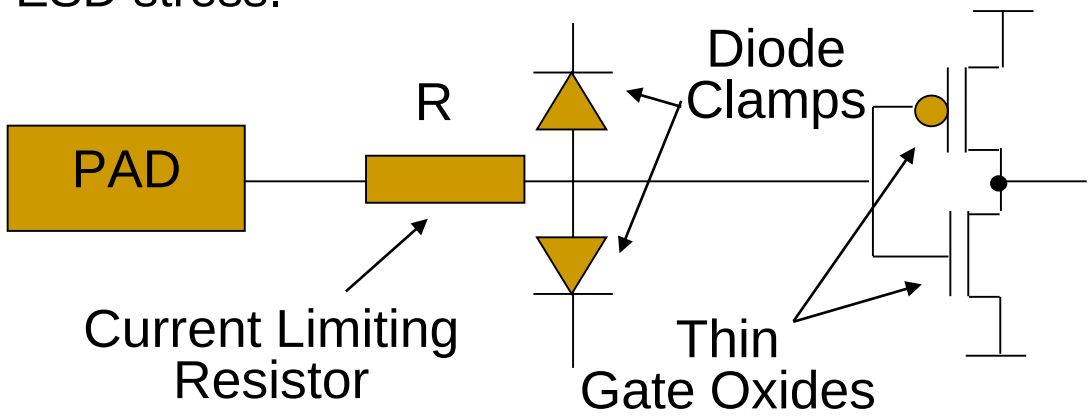


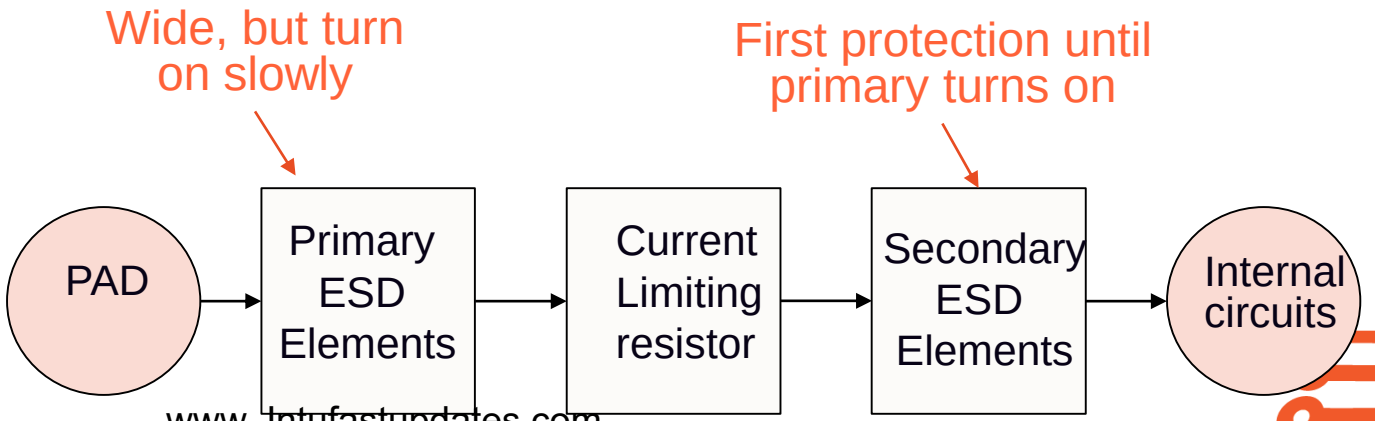
www.Jntufastupdates.com

ESD Protection



- **Electrostatic discharge (ESD)**
  - One of the most important reliability problems in the IC industry.
  - ESD protection circuits divert high currents away from the internal circuitry and clamp high voltages during an ESD stress.
- **Diode clamps**
  - Diodes turn on if pad voltage:
    - Exceeds VDD +0.7V
    - Drops below VDD -0.7V
- **Resistor**
  - Limits the current
  - Protects secondary protection
- **Diode formation**
  - P+ diffusion in n-well
  - N+ diffusion in p-substrate
- **Resistors**
  - Diffusion
  - Polysilicon

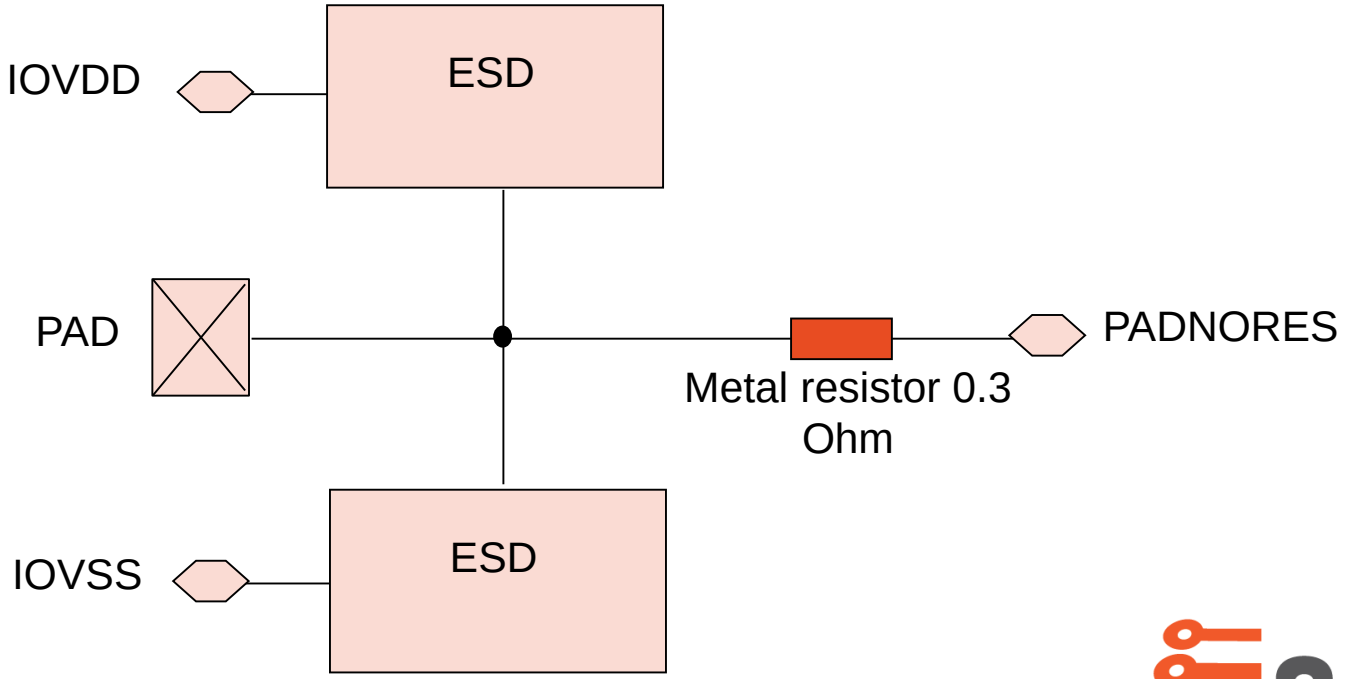




www.Jntufastupdates.com

Analog I/O Cell

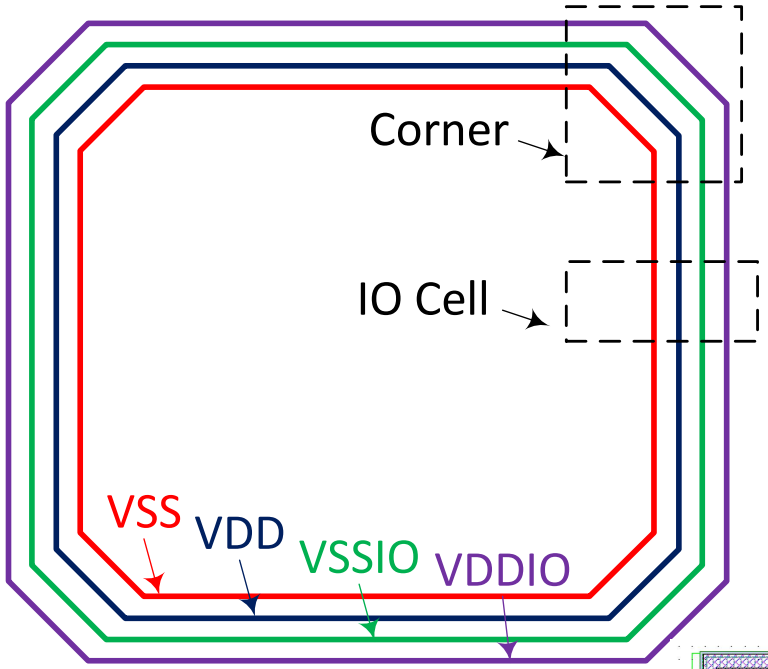
- **Analog I/O**
  - Used for passing “analog” signals to/from the chip.
  - Basically, “a wire”, but should have some degree of ESD protection.

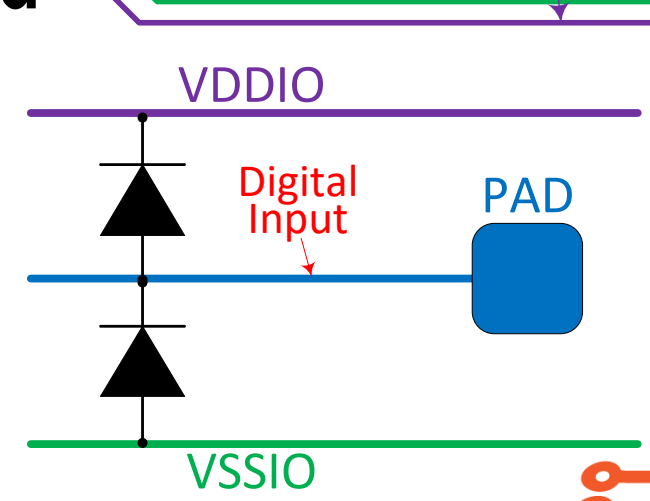


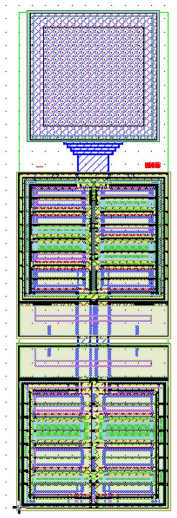
www.Jntufastupdates.com

Power Supply Cells and ESD Protection

- **Power supply cells are analog cells (i.e., just a wire).**
- **But these cells supply the I/O rings for:**
  - Power distribution
  - ESD Protection
- **Generally, digital (core) and I/O power/ground supplies are separate:**
  - I/Os sink a lot of current → Power supply noise
  - I/Os usually run at a higher voltage level (i.e., 2.5V vs. 1.2V)
  - All (four) types of supplies connect to rings under the I/O circuits.



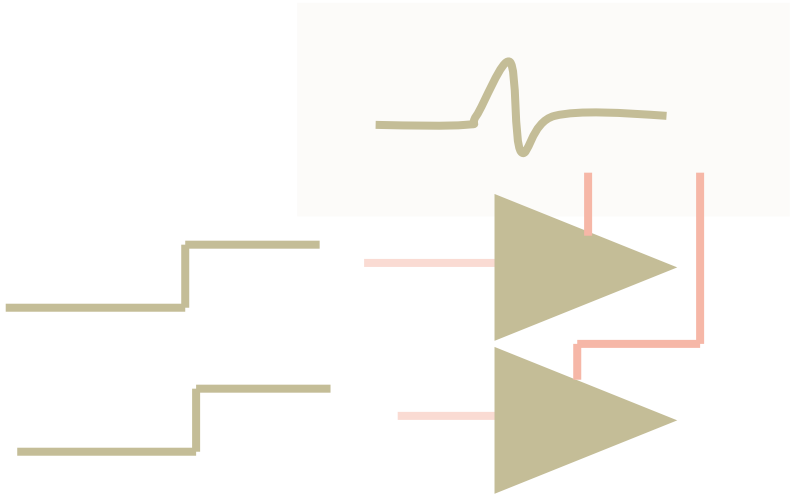




www.Jntufastupdates.com

Simultaneously Switching Outputs

- **Simultaneously Switching Outputs (SSO) is a metric describing the period of time during which the switching starts and finishes.**
  - Consider a 64 bit output bus.
  - If all transition from high to low,
    lots of current must driven/sunk leading to extensive voltage drop.
  - Problem is independent of frequency



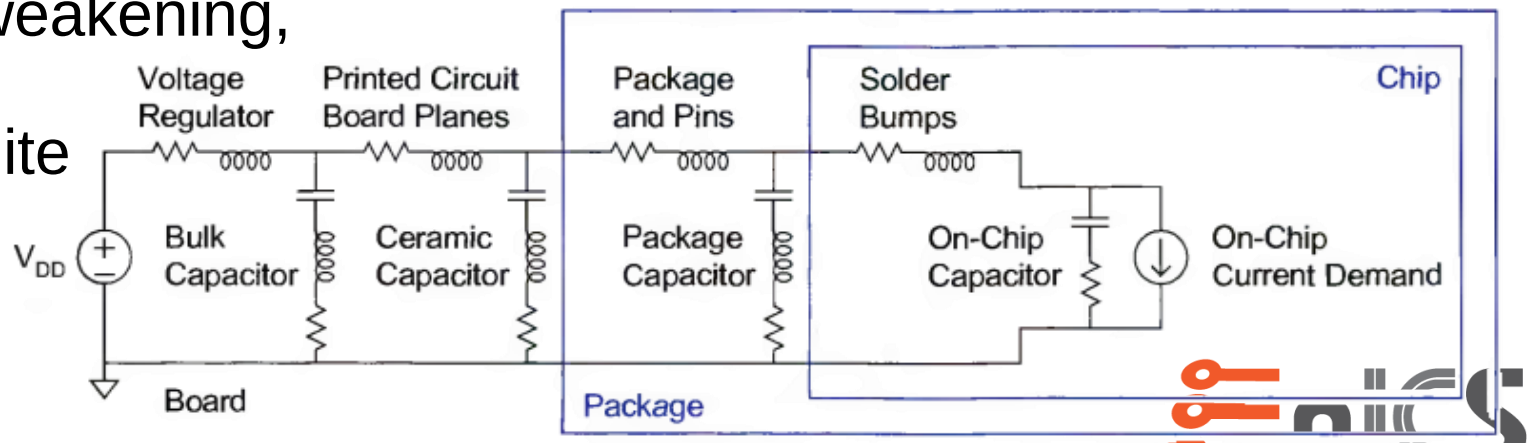
SSO-2

$$V_{drop}=L_{package}di/dt$$

www.Jntufastupdates.com

Design Guidelines for Power

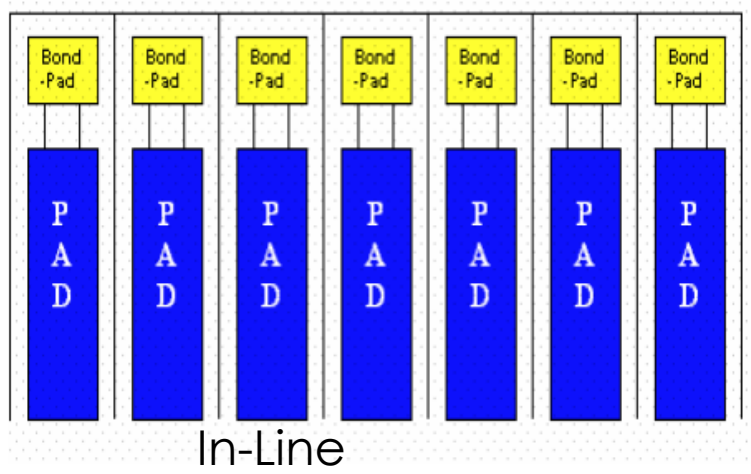
- **Follow these guidelines during I/O design:**
  - Put as many mutual capacitances as possible between IC supply voltages.
  - Put as many supply voltage pins as possible.
    Put supply and ground supply voltages as close to each other as possible.
  - Provide separate supply voltages for the core and I/Os.
  - Reduce inductances as much as possible by using as short transmission lines as possible.
  - Reduce signal rate as much as possible. But be careful as the reduction of signal rate leads to signal weakening, and experiments show that those noises can have definite affect on the given I/O cells.



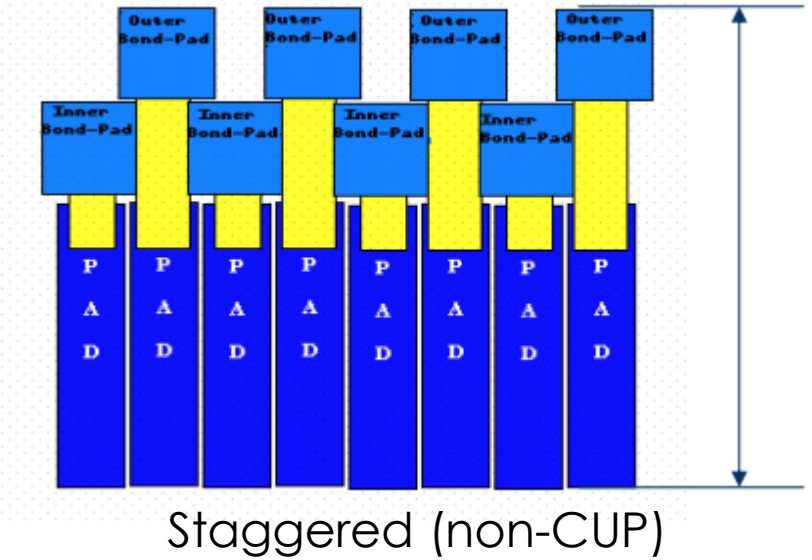
www.Jntufastupdates.com

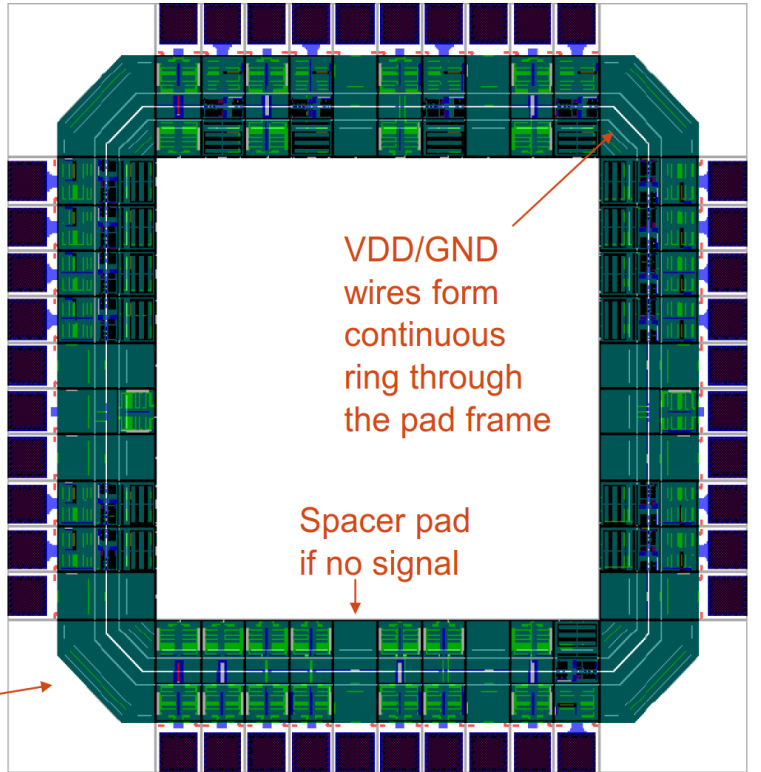
Pad Configurations

- **In-line:**
  - Pads are placed next to each other, with the corresponding bond pads lined up against each other having a small gap in between.



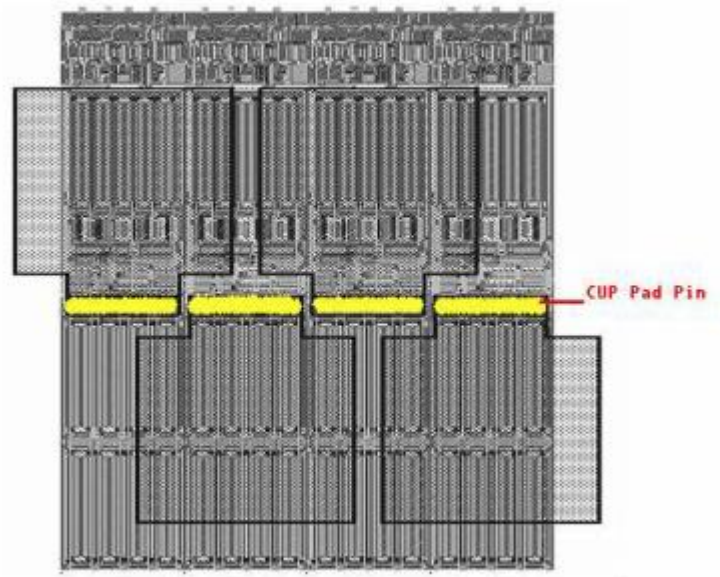
- **Staggered:**
  - Useful technique if design “Pad Limited”.
  - A larger number of pads can be accommodated in the design, but the overall height of the pad structure increases significantly



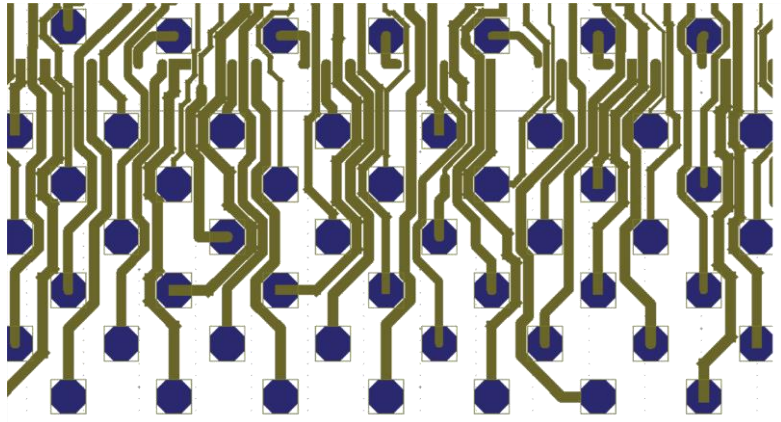


www.Jntufastupdates.com

Pad Configurations



- **Circuit Under Pad (CUP):**
  - CUP I/O is has the bonding pad over the I/O body itself.
  - Bonding pad has to be placed over the I/O body and is connected to the PAD pin of the I/O.
  - CUP I/O can substantially reduce the die size since the bonding pad does not take any extra space in addition to the I/O body itself.



- **Flip Chip with RDL:**
  - In the Flip chip methodology, I/O bumps and driver cells may be placed in the peripheral or in the core area.
  - Signals and power are connected to the bumps through a top aluminum layer called the Redistribution Layer (RDL).

www.Jntufastupdates.com

Main References

- AMMOS – CDNLive 2007
- IDESA
- CMOS VLSI Design

www.Jntufastupdates.com