UNIT 5:
Low – Power CMOS Logic Circuits
Ad Hoc Testable Design Techniques

• 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.

Ad hoc testable design techniques:
• *Partition-and-Mux Technique*
• *Initialize Sequential Circuit*
• *Disable Internal Oscillators and Clocks*
• *Avoid Asynchronous Logic and Redundant Logic*
• *Avoid Delay-Dependent Logic*
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.

• 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.
Partition-and-mux method for large circuits.
**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.
**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.
Avoid synchronization problems via disabling of the oscillator.

Diagram:

- OSC
- Disable
- Test Clock
- AND gate
- Circuit
Avoid Asynchronous Logic and Redundant Logic

• 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 shows that the bottom NAND2 gate is redundant and the stuck-at- 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.
\[ F = AB + BC + \bar{AC} \]

\[ = AB + \bar{AC} \]

(a) A redundant logic gate example. (b) Equivalent gate with redundancy removed
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.
• 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.
A pulse-generation circuit using a delay chain of three inverters