Module 5

Memory, Registers and Aspects of System Timing

System Timing Considerations

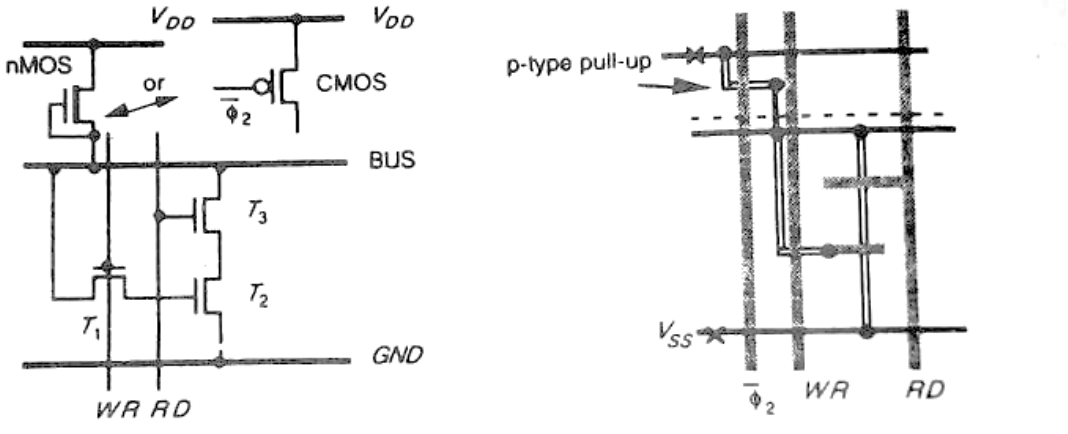
1. Two phase non-overlapping clock is assumed to be available and this clock will be used throughout the system.
2. Clock phases are assumed to be ϕ_1 and ϕ_2 and ϕ_1 is assumed to lead ϕ_2.
3. Bits (data) to be stored are written to registers, storage elements and subsystems **on ϕ_1** of the clock i.e., WR (write) signal is ANDed with ϕ_1.
4. Bits written into storage elements may be assumed to have settled before ϕ_2 signal (which follows immediately) and ϕ_2 signal may be used for refreshing the stored data.
5. Delay through data paths, combinational logic etc., are assumed to be less than the interval between leading edge of ϕ_1 of the clock and leading edge of following ϕ_2 signal.
6. Bits or data may be read from storage elements **on the next of ϕ_1**. RD (read) signal is ANDed with ϕ_1. Thus RD and WR signals are mutually exclusive.
7. General requirement for system stability is that there must be at least one clocked storage element in series with every clocked loop signal path.

Some commonly used storage/memory elements:

The storage elements are compared based on three parameters

1. Area requirement
2. Estimated dissipation per bit stored.
3. Volatility

A three-transistor Dynamic RAM Cell (3T DRAM Cell)



(a) Circuit

(b) CMOS stick diagram

Note: *WR* and *RD* are coincident with ϕ_1.

FIGURE 9.1 Three-transistor dynamic memory cell.

- In this cell arrangement it uses a single transistor for storing data and 2 transistors for each RD and WR access switch.
- It has a pull-up network with either CMOS or nMOS technology and RD/WR circuit as pull-down network.
- The binary data is stored at gate capacitance of transistor in the form of charge; RD and WR are the control lines.
- T1 with T2 is used for writing the data and T3 with T2 is used for reading the data. At point I data is written and read.
- Here T2 is the storage transistor and T1 & T3 are pass transistors which acts as access switches for control lines RD and WR and also for read and write operations.

Write Operation

- WR and RD signals are mutually exclusive i.e., compliment to each other.
- ✓ When WR = 1, RD will be 0
- ✓ Because of WR = 1, T1 is ON but T3 and T2 are OFF.
- ✓ If data bit on bus is 1, as T1 pass transistor is ON it will pass the signal ($V_{DD} – V_{th}$) towards T2. The capacitor is charged to this potential at I
- ✓ If data bit is 0, as T1 is ON it will pass the signal and charge stored at I is 0.
- ✓ After the data is stored at I or capacitor WR signal is made to 0

Read Operation

- For this RD = 1, WR = 0
  - ✓ As WR = 0, T1 is OFF and T3 is ON as RD = 1
  - ✓ T2 will be ON/OFF depending on the voltage/charge stored at I (gate capacitance of T2)
  - ✓ If logic 1 is stored at I, then T2 will be ON. Thus T3 and T2 is ON and path for discharge and the bus is pulled down to ground.
  - ✓ If logic 0 is stored at I, the T2 is OFF and charge does not any path for discharge and retains logic at logic1

Note: The compliment of stored bit is read on the data bus

- In DRAM sensing amplifiers will be connected and as the output begins to decrease from 1 to 0 and this makes the sensing amplifier output as logic 1. If the output does not change then sensing amplifier will make the output as logic 0.

Static Power:

- Static power dissipation is nil since current flows only when i)RD signal is high and ii)logic 1 is stored
- Thus actual dissipation associated with each stored bit depends on the bus pull-up and the duration RD and on switching frequency

Area:

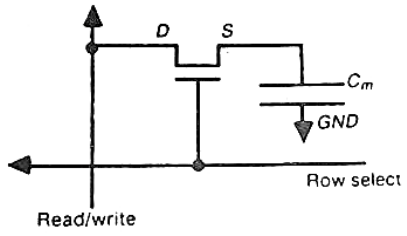
- In 4mm× 4mm, silicon chip area data storage can be >4.8kbits.

Volatility:

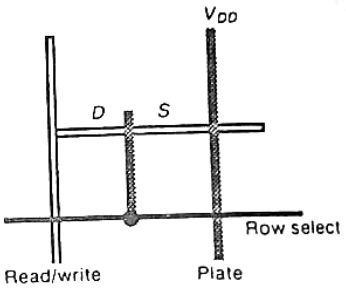
- Cell is dynamic and will hold data only for as long as sufficient charge remains on gate capacitance of T2

One-Transistor Dynamic Memory Cell:

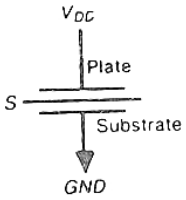
- This is an approach which reduces area/bit
- It consists of capacitor Cm and pass transistor. The circuit arrangement and stick diagram is shown in Fig a & b.



(a) Circuit arrangement



(b) Stick diagram



(c) Equivalent circuit

Write operation

- The capacitor Cm will be charged when Read/Write = 1 and Row Select = 1
- If the Read/Write line is provided with logic 1, Cm will be charged to logic 1 and if the line is provided with logic 0 charge stored will be logic 0.

Read operation

- If logic 0 is stored in Cm and when Row select line is high M1 is ON. Then the sense amplifier at the bit line will sense and give the output as logic 0
- If logic 1 is stored in Cm and when Row select line is high M1 is ON, the logic 1 stored will begin to discharge as the path exists. The sense amplifier senses this and this gives the output as logic 1

- The area occupied in 1T DRAM cell configuration is of a single transistor and a capacitor
- Larger the value of Cm longer is the duration of storage of charge. Thus Cm should be large. But this in turn consumes more space.
- However the transistor and capacitor can be built in a single transistor.
- Cm can be fabricated by extending and enlarging the diffusion area forming the source of process transistor. For this capacitance between n-diffusion and p-substrate is considered.
- But this value is very small compared to gate capacitance
- Thus in order to get higher Cm larger area is required.
- An alternate solution to this by using a polysilicon plate used over diffusion area. This results in the formation of a 3 plate capacitor structure, where polysilicon plate is connected to V_{DD}. This is shown in the Fig c.

Area:

- In 4mm× 4mm, silicon chip area data storage is about 12 Kbits.

Dissipation:

- With the cell there is no static dissipation but switching energy while reading and writing must be considered.

Volatility:

- The data in Cm will be held only up-to 1msec or less. Thus periodic refreshing must be provided

Pseudo-static RAM/register cell:

- This is a memory cell which combines high storage capability of DRAM and ease of use of SRAM
- It can be used as SRAM as no external refreshing circuit is required and also used as a DRAM having built-in refresh logic.
- This is a static storage cell which will hold data indefinitely. This is achieved by storing bit in 2 inverters with feedback. This feedback is used to refresh the data in every clock cycle.
- But care to be taken by not allowing read/write operation during internal refreshing.
- Circuit arrangement is as shown in the Fig.
- Φ_1 and ϕ_2 are mutually exclusive clock signals, WR and RD signal coincides with ϕ_1 signals
- When ϕ_1 is high and WR = 1, transistor T1 is ON and data is charged/stored on Cg (gate capacitance) of inverter. This is write operation
- When ϕ_1 is high and RD = 1, transistor &the data stored at inverter stage is made available at the output and also the compliment. Thus data is read at the output.
- When Φ_2 = 1, T3 is ON. The output is read and feedback i.e., refreshed (reading and storing back the data). The gated feedback path from output of T2 is fed to the input of T1.
- The bit will be held as long as ϕ_2 rescues and this time is less than decay time of stored charged bit.

Note:

- WR and RD must be mutually exclusive but both should coincide with ϕ_1
- During refreshing of memory cell i.e., at ϕ_2 the cell must not be read. If an attempt is made to read the cell data onto the bus, the charge sharing effect between bus and Cg (input gate capacitance) may cause destruction of stored bit.
- Other bus lines should be allowed to run through the cells so that register and memory arrays can be easily configured.

- The Pseudo-static memory cell can also be implemented using transmission gate (TG). This is seen the Fig. [replace nMOS transistors with TG]

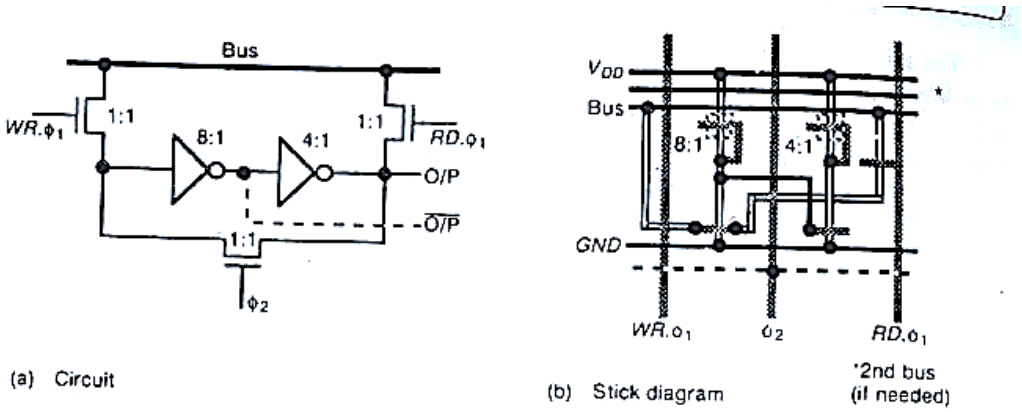


FIGURE 9.4 nMOS pseudo-static memory cell.

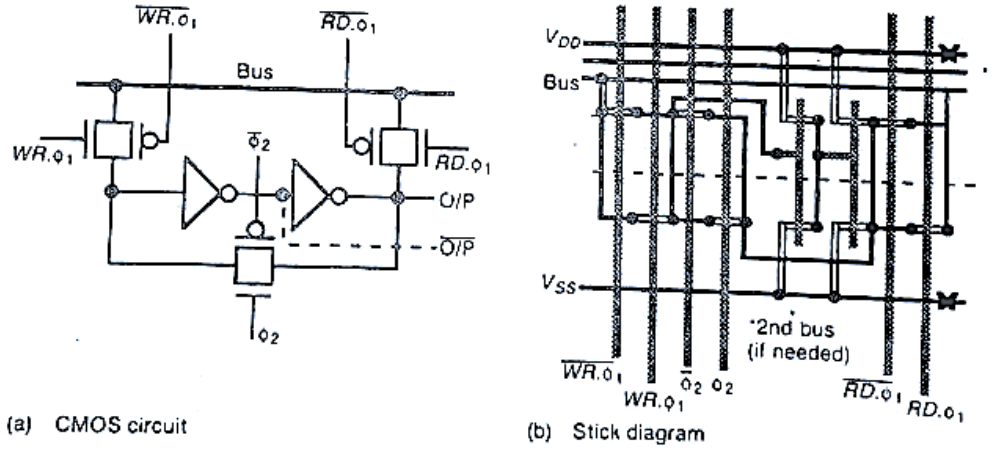


FIGURE 9.5 CMOS pseudo-static memory cell.

Area:

- In 4mm× 4mm, silicon chip area data storage is about 1.4 Kbits.

Dissipation:

- The nMOS cell uses 2 inverters, one with 8:1and other with 4:1 ratio. Thus power dissipation depends on the current drawn and actual geometry of the inverters.

Volatility:

- The cell is non-volatile until unless ϕ_2 is present.

Four Transistor Dynamic and Six-Transistor CMOS memory cell:

- The cells here include both n-type and p-type transistors and are intended for CMOS systems.
- Both the dynamic and static elements uses 2 bus per bit arrangement so that the bit is available in both normal and compliment form on bit and bit' bus
- Prior to reading and writing operation of the data, the buses are pre-charged to V_{DD} or logic 1.

4 Transistor Dynamic memory cell:

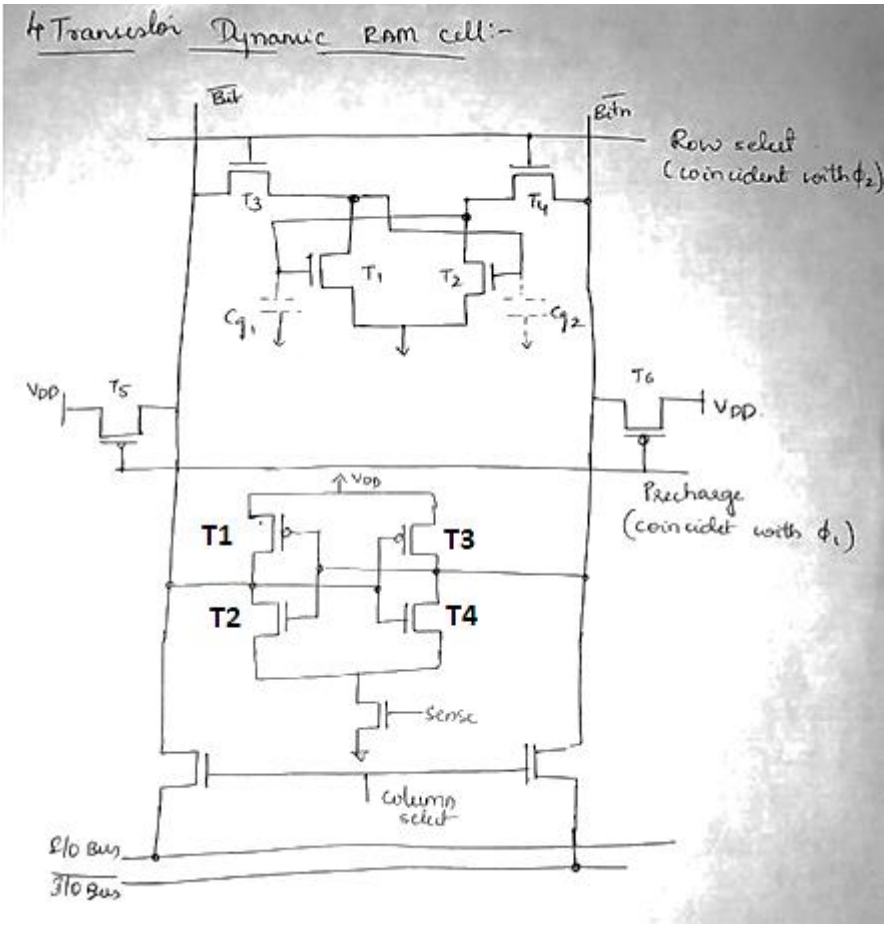


Fig. four transistor dynamic RAM with sense amplifier

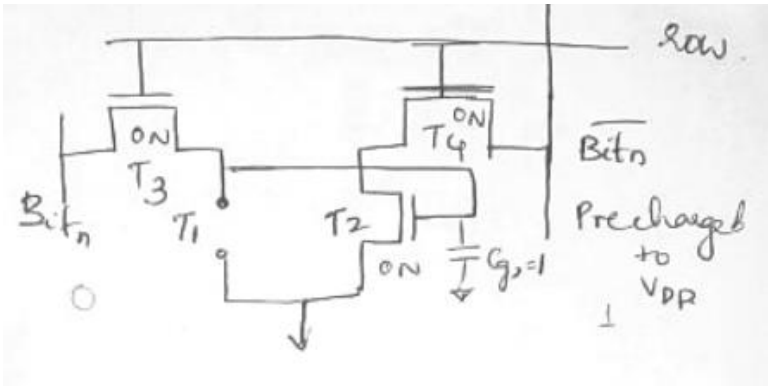
Write operation:

- Before writing onto memory the bit and bit' line is pre-charged to logic 1 using pMOS transistor T5 and T6 in coincidence with clock signal ϕ_1
- Next appropriate column is selected in coincidence with the clock signal ϕ_2.
- Depending on the data on the bus either bit or bit' is discharged.
- At the same clock signal ϕ_2 the row select line is activated, turning on transistors T3 and T4.
- Thus value on bit and bit' are written via T3 and T4 stored at T2 and T1 as gate capacitances C_{g2} and C_{g1} respectively.
- The way in which T2 and T1 are connected always gives the complimentary states when row select line is activated. When row line is deactivated the data stored will remain until the gate capacitance can hold the value.
- For refreshing sense amplifier is provided which will permanently hold the data.

Read operation:

- Before reading again bit and bit' lines are pre-charged to V_{DD} using T5 and T6 transistors.
- Suppose in the memory element if logic 1 is stored i.e., at gate of T2 and at gate of T4 logic 1 is stored.

- When column and row lines are selected i.e., T3 and T4 will be in ON state.
- As logic 1 is available at T2, T2 will be in ON state and T1 will be in OFF state. Thus T3 = ON, T1 = OFF, T4 = ON, T2 = ON. With this condition bit' which was pre-charged to V_{DD} has now a path to discharge to V_{SS}. Hence bit' = 0 and bit = 1 as shown in the Fig.
- When sense amplifier senses this voltage variation on bit' line and outputs the data on bus line. The bit = 1 and bit' = 0, which represents the data in the memory.
- The sense amplifier formed from the arrangement of T1, T2, T3 and T4, which forms a flip flop circuit.
- If the "sense" de-active/ inactive, then the bit line state is reflected in the gate capacitances of T1 and T3 and this is w.r.t V_{DD}. This will cause one of the transistor to turn ON and other to turn OFF.
- When sense = enabled, current flows from V_{DD} through ON transistor and helps to maintain the state of the bit line.
- Sense amplifier performs 2 function
  1. Rewriting the data after reading i.e., refreshing the memory cell so that it holds the data without signal degradation
  2. It predetermines the state of the data lines.



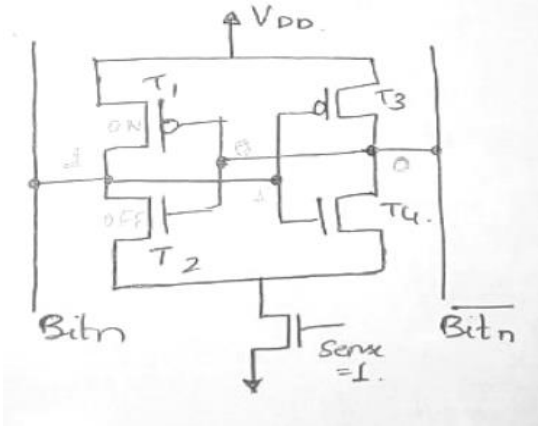


Fig. shows Read operation in the memory cell and in the sense amplifier.

Six Transistor Static RAM cell:

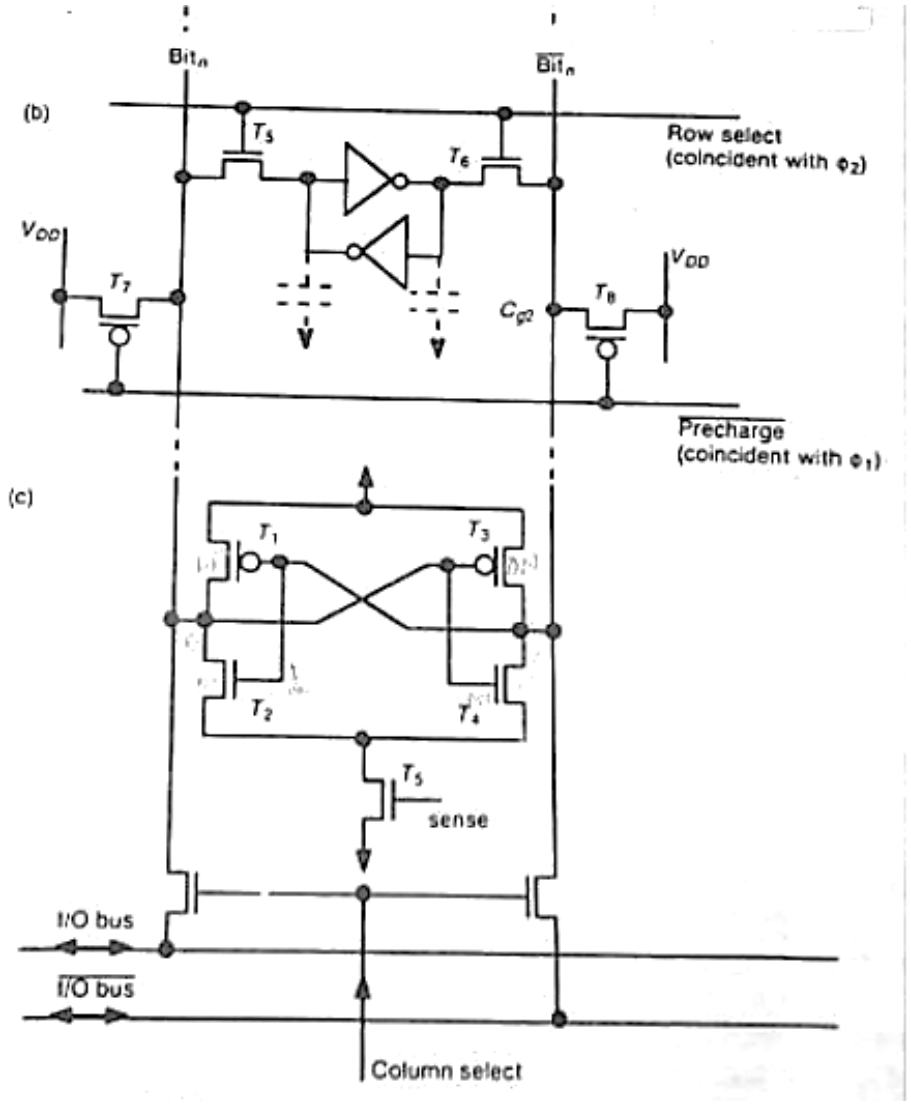


Fig. six transistor static RAM cell with sense amplifier

- Figure shows 6 T SRAM with the adaption of dynamic cell and modifying it to form a static memory cell.
- It includes 2 additional transistor per store bit thus it is called 6 transistor. The transistor T5 and T6 acts as the access switch for memory element which is formed by connecting two inverters back to back (i.e., output of one is connected as the input of the other)
- Similar to 4T Dynamic RAM the information is stored in memory cell. The memory cell is connected in such a way that it gives the complimentary states when row select line is activated. When row line is deactivated the data stored will remain in the memory cell.

Below Fig. shows dynamic and static RAM cell together as the sense amplifier is same in both the memory cell.

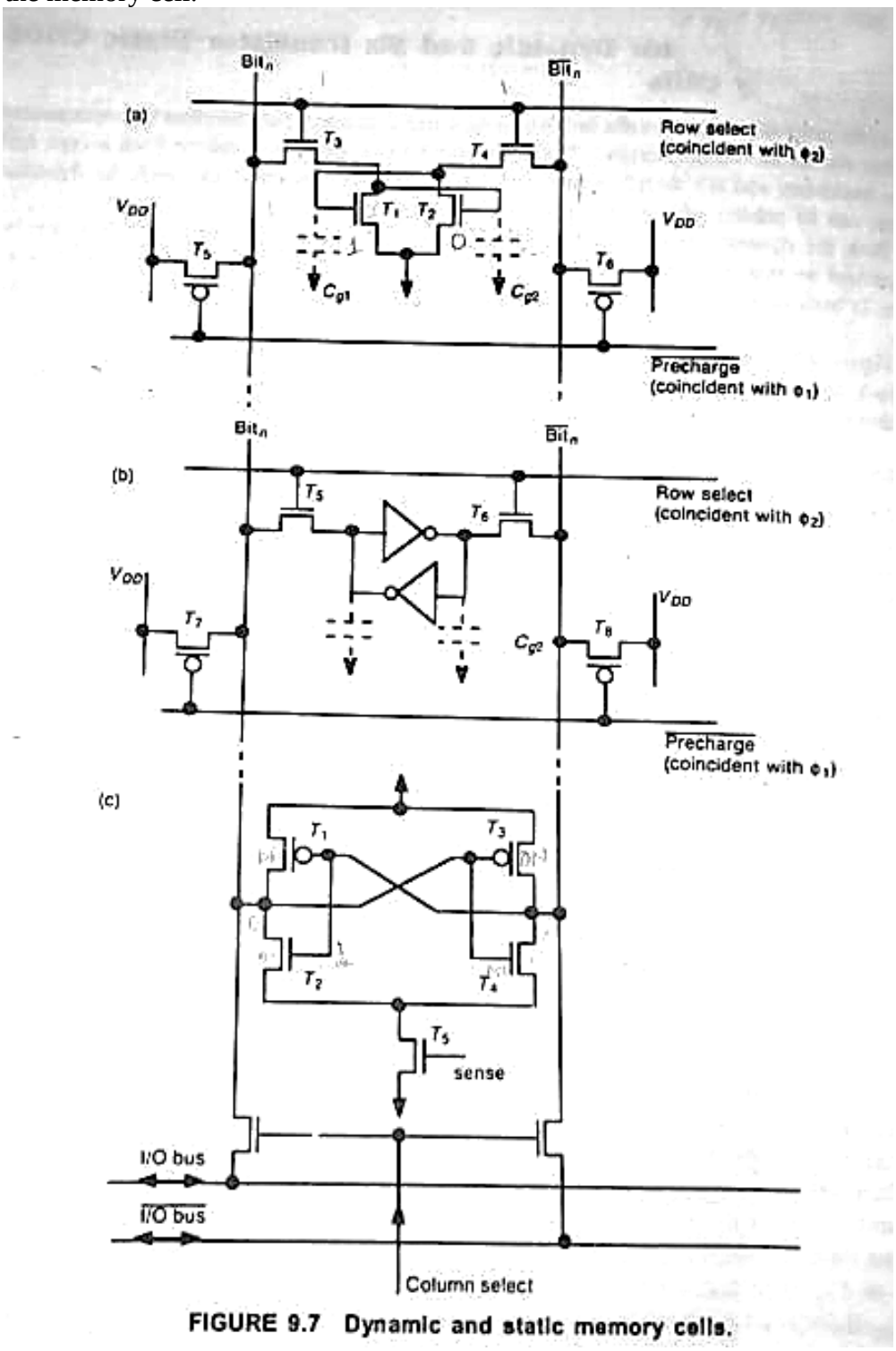


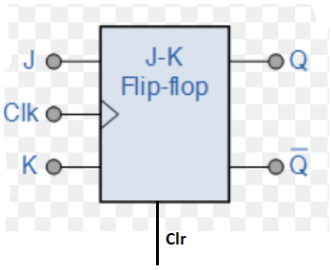
FIGURE 9.7 Dynamic and static memory cells.

JK flip flop:

- It is a memory element. It is the widely used arrangement for static memory element.
- Also with JK other flip-flop arrangements can be obtained such as T and D flip-flop.

- The flip-flop has inputs clocked J and K along with asynchronous clear and has the output as Q and Q'
- The inputs J and K are read for the rising edge of clock signal and data is passed to the output for the falling edge of clock.

Note: here JK is implemented in master slave configuration in order to solve the race around condition



Gate Implementation of flip-flops:

- The expressions for the flip-flops can be implemented using either NAND or NOR logic
- If NAND arrangement is used (in NAND inputs have to be connected in series) then it would take large area. Also it is seen that when connected in series the overall delay increases and in practice not more than 4 transistors should be connected in series. However the number can be increased by including buffers in between and next four transistors. Also it is seen that performance of NAND is slower than NOR.
- In NOR the implementation can be done easily (as the inputs have to be connected in parallel)

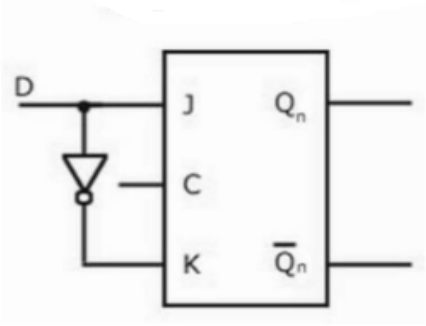
Switch logic and inverter implementation:

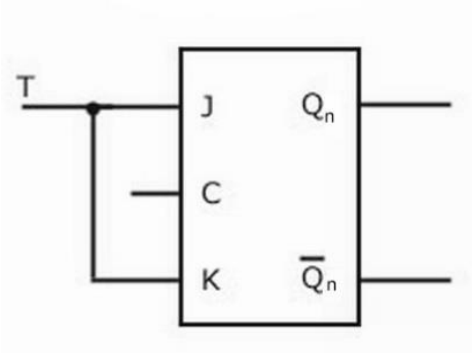
- If n pass transistors are used to realize the logical requirements, it must be kept in mind that
  1. There should be more than 4 pass transistors connected in series.
  2. One pass transistor should not drive the gate of other pass transistor.

D and T flip-flop circuit:

D flip-flop can be formed from JK by connecting an inverter between both inputs.

T flip-flop can be readily formed from JK by connecting JK to form T input. This is shown in the Fig.





Testing and Verification

- In VLSI testing relates to the procedure that takes place after chip is fabricated in order to find any defects.
- There are 2 benefits from testing – Quality and economy
    - Quality – is satisfying user's need at minimum cost and testing weeds/removes all bad products before they reach the user.
    - If more number of products are bad then automatically cost increases. Thus bad products will heavily effect the price of good products.
- Testing is classified into 3 groups.
    1. Logical Verification: The first set of tests verifies that the chip performs its intended function. These tests, called functionality tests or logic verification, are run before tape-out to verify the functionality of the circuit.
    2. The second set of tests called silicon debug are run on the first batch of chips that return from fabrication. These tests confirm that the chip operates as it was intended and help debug any discrepancies. They can be much more extensive than the logic verification tests because the designer has much less visibility into the fabricated chip compared to during design verification.
    3. The third 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. These are called manufacturing tests.
- As manufacturing process is complex, not all die on a wafer may function correctly. Dust particles and small imperfections in starting material or photo-masking can result in bridged connections or missing features. These imperfections result in what is termed a **fault**.
- The goal of a manufacturing test procedure is to determine die (chip) that are good and should be shipped to customers. Testing a die (chip) can occur at the following levels:
    - ✓ Wafer level
    - ✓ Packaged chip level
    - ✓ Board level
    - ✓ System level
    - ✓ Field level

Logic Verification:

- Verification tests are usually the designer first choice that is constructed as part of the design process
- Verification tests is necessary to prove that a synthesized gate description was functionally equivalent to the source RTL. This proves that RTL is equivalent to the design specification at a higher behavioral or specification level of abstraction.
- The behavioral specification might be a verbal description, a plain language textual specification, a description in some high level computer language such as C, a program in a system-modeling language such as System C, or a hardware description language such as VHDL or Verilog, or simply a table of inputs and required outputs.

- Often, designers will have a golden model in one of the previously mentioned formats and this becomes the reference against which all other representations are checked.
- Fig. shows functional equivalence at various levels of abstraction.

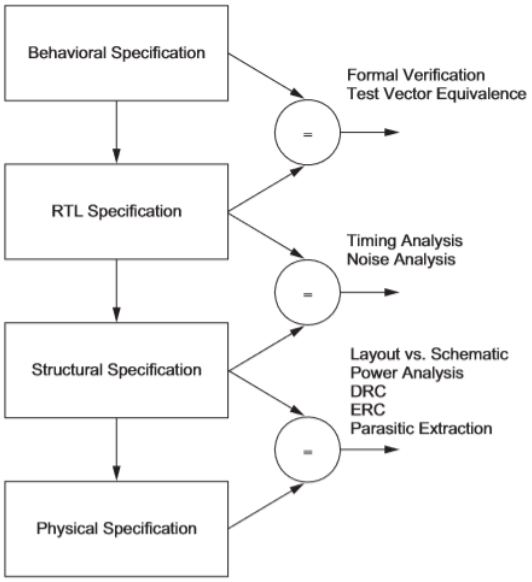


Fig. Functional equivalence at various levels of abstraction

Logic verification principles:

Test vectors: Test vectors are a set of patterns applied to inputs and a set of expected outputs. Both logic verification and manufacturing test require a good set of test vectors. These must be large enough to catch all the logic errors and manufacturing defects, yet small enough to keep test time (and cost) reasonable.

Test bench and Harnesses: A test bench or harness is a piece of HDL code that is placed as a wrapper around a core piece of HDL (stimuli) to apply and check test vectors. In the simplest test bench, input vectors are applied to the module under test and at each cycle, the outputs are examined to determine whether they are same as predefined expected data set. The expected outputs can be derived from the golden model and saved as a file or the value.

Regression Test: 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. During a design, it is common practice to run a regression script every night after design activities have concluded to check that bug fixes or feature enhancements have not broken completed modules

Bug Tracking: Another important tool to use during verification is a bug-tracking system. Bug-tracking systems such as the Unix/Linux based GNATS (is a set of tools for tracking tools) allow the management of a wide variety of bugs. In these systems, each bug is entered and the location, nature, and severity of the bug is noted.

Manufacturing Test Principles:

A critical factor in VLSI design is the necessity to incorporate methods of testing circuits. This task should proceed concurrently with architectural considerations and not be left until fabricated parts are available.

Fig a. below shows a combinational circuit with N inputs. To test this circuit exhaustively, a sequence of 2^N inputs (or test vectors) must be applied and observed to fully exercise the circuit.

If this combinational circuit is converted to a sequential circuit with addition of M registers, as shown in **Fig b**. The state of the circuit is determined by the inputs and the previous state. A minimum of 2^{N+M} test vectors must be applied to exhaustively test the circuit. This would take a long time.

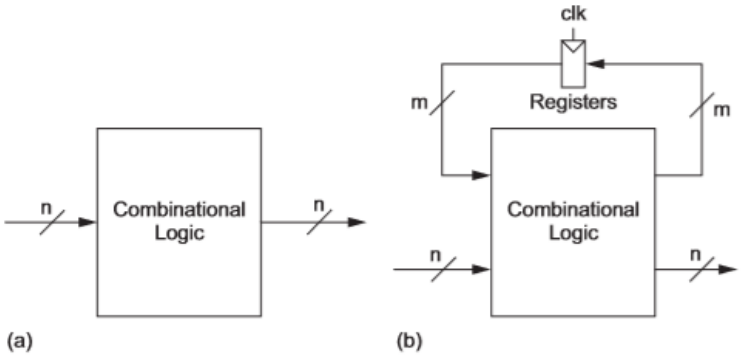


Fig. The combinational explosion in test vectors

Hence, exhaustive testing is infeasible for most systems. Thus the verification engineer must cleverly devise test vectors that detect any (or nearly any) defective node without requiring so many patterns.

Fault Models

- In order to determine good and bad parts in a chip, it is necessary to propose a fault model. This model will help to know where and how faults occur, what is their impact on circuits. The most popular model is called the Stuck-At model. The Short Circuit/ Open Circuit is another model can be a closer fit to reality, but this is difficult to implement in logic simulation tools.

Stuck-At Faults: In the Stuck-At model, a faulty gate input is modeled as a stuck at zero (Stuck-At-0, S-A0) or stuck at one (Stuck-At-l, S-A-l). This model dates from board-level designs, where it was determined to be adequate for modeling faults. Fig illustrates how an S-A-0 or S-A-1fault might occur in basic gates. These faults most frequently occur due to gate oxide shorts (the nMOS gate to GND or the pMOS gate to V_{DD}) or metal-to-metal shorts.

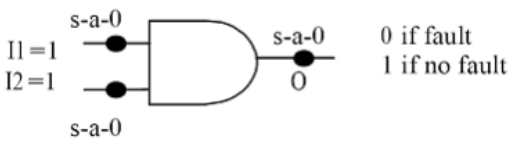


Fig Stuck-at-0 faults in an AND gate

Fig. Stuck-at-1 fault in an OR gate

To test the fault at I1, input pattern is I1=1, I2=1; if the output is 0, s-a-0 fault in I1 is present, else it is absent. Now, also for the s-a-0 fault in net I2, the pattern is I1=1, I2=1.

Short-Circuit and Open-Circuit Faults: Other models include stuck-open or shorted models. Two bridging or shorted faults are shown in Figure. The short S1 results in an S-A-0 fault at input A, while short S2 modifies the function of the gate.

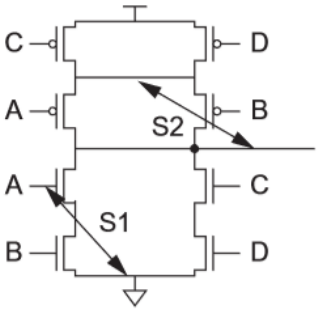
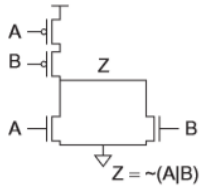
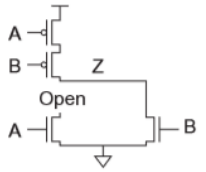


Fig. CMOS Bridging Faults





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

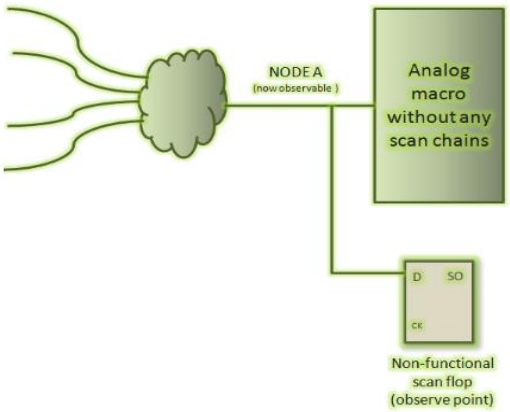
Fig. CMOS open Fault that causes sequential faults

- A particular problem that arises with CMOS is that it is possible for a fault to convert a combinational circuit into a sequential circuit. This is illustrated in the Fig.
- Considering the case of a 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 = (A + B)' + BZ', where Z' is the previous state of the gate.
- Stuck - closed states can be detected by observing the static V_{DD} current (I_{DD}) while applying test vectors.

Observability:

The observability of a particular circuit node is the degree to which you can observe that node at the outputs of an integrated circuit (i.e., the pins). OR It is the ability to observe the response of the circuit through primary outputs or at some other output points.

This is relevant when you want to measure the output of a gate within a larger circuit to check if it operates correctly. Given the limited number of nodes that can be directly observed, it is the aim of good chip designers to have easily observed gate outputs

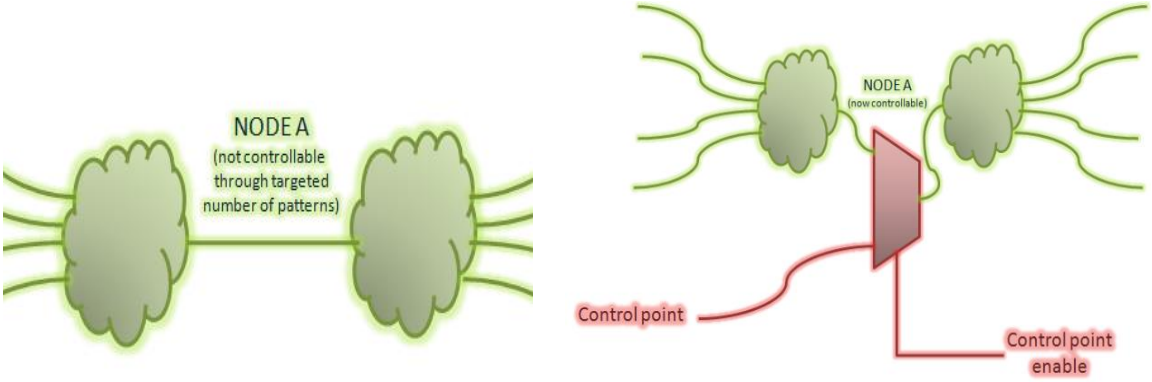


Using some basic design for test techniques can help tremendously in this respect. It should be able to observe every gate output in an circuit directly or with moderate indirection (where have to wait for few cycles). In order to enhance observability the outputs must be observed seperately and this may be done with expense of extra test circuit.

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. OR It is the ability to apply test patterns to the inputs of the circuit through primary inputs of the circuit.

This is of important 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.



If a node is little controllability, such as the MSB bit of a counter may need hundreds or thousands of cycles to get it to the right state. And it is highly difficult to generate a test sequence to set a number of poorly controllable nodes.

It should be the aim of good chip designer to make all nodes easily controllable. In common with observability, the adoption of some simple design for test techniques can help in this respect tremendously. Example making all flip-flops resettable via a global reset signal is one step toward good controllability.

Fault Coverage:

- This determines what percent of the chip's internal nodes are checked when the test vectors are applied. 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.
- The way in which the fault coverage is calculated is as follows:
- Each circuit node is taken in sequence and held to 0 (S-A-0), and the circuit is simulated with the test vectors and then comparing the chip outputs with a known good machine—a circuit with no nodes artificially set to 0 (or 1).
- If any 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.
- To achieve world-class quality levels, circuits are required to have in excess of 98.5% fault coverage.

Automatic Test Pattern Generation:

In the IC industry, logic and circuit designers implements the functions at the RTL or schematic level, mask designers completes the layout, and test engineers write the tests.

The test engineers took the assistance of designers to include extra circuitry to ease the burden of test generation. With increased complexity and density, the inclusion of test circuitry has become less of an overhead for both the designer.

In addition, as tools have improved, more of the burden for generating tests has fallen on the designer. To deal with this burden, Automatic Test Pattern Generation (ATPG) methods have been invented.

Commercial ATPG tools can achieve excellent fault coverage. However, they are computation-intensive and often must be run on servers or compute farms with many parallel processors.

Some tools use statistical algorithms to predict the fault coverage of a set of vectors without performing as much simulation. Adding scan and built-in self-test improves the observability of a system and can reduce the number of test vectors required to achieve a desired fault coverage.

Delay Fault Coverage

The fault models seen till now point have neglected timing. Failures that might have occured in CMOS would leave the functionality of the circuit untouched, but may affect the timing. For example considering an inverter gate with 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 t_{pdf} (rising propagation delay). In addition, the fault now becomes sequential as the detection of the fault depends on the previous state of the gate. Delay faults may also 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.

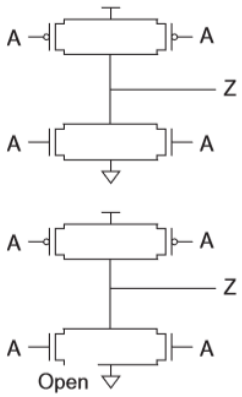


Fig. An example of Delay Fault

Design for Testability:

The keys to designing circuits that are testable are controllability and observability. Controllability is the ability to set (to 1) and reset (to 0) every node internal to the circuit. Observability is the ability to observe, either directly or indirectly, the state of any node in the circuit.

Good observability and controllability reduce the cost of manufacturing testing because they allow high fault coverage with relatively few test vectors.

There are three main approaches to what is commonly called as Design for Testability (DFT). These may be categorized as follows:

- Ad hoc testing
- Scan-based approaches
- Built-in self-test (BIST)

Ad hoc Testing:

Ad hoc test techniques, as their name suggests, are collections of ideas aimed at reducing the combinational explosion of testing. They are only useful for small designs where scan, ATPG, and BIST are not available.

Some of the common techniques for ad hoc testing are:

- ✓ Partitioning large sequential circuits
- ✓ Adding test points
- ✓ Adding multiplexers
- ✓ Providing for easy state reset

Some of the examples are: multiplexers can be used to provide alternative signal paths during testing. In CMOS, transmission gate multiplexers provide low area and delay overhead. Use of the bus in a bus-oriented system for test purposes. Here each register is made loadable from the bus and capable of being driven onto the bus. Here, the internal logic values that exist on a data bus are enabled onto the bus for testing purposes.

Any design should always have a method of resetting the internal state of the chip within a single cycle or at most a few cycles. Apart from making testing easier, this also makes simulation faster as a few cycles are required to initialize the chip. In general Ad hoc testing techniques represent a bag of tricks.

Scan Design:

- The scan-design strategy for testing has evolved to provide observability and controllability at each register.
- In designs with scan, the registers operate in one of two modes.
- In normal mode: they behave as expected
- In scan mode: they are connected to form a giant shift register called a scan chain spanning the whole chip.
- By applying N clock pulses in scan mode, all N bits of state in the system can be shifted out and new N bits of state can be shifted in. Thus scan mode gives easy observability and controllability of every register in the system.
- Modern scan is based on the use of scan registers, as shown in Fig. The scan register is a D flip-flop preceded by a multiplexer. When the SCAN signal is deasserted (made to 0), the register behaves as a conventional register, storing data on the D input. When SCAN is asserted (made to 1), the data is loaded from the SI pin, which is connected in shift register fashion to the previous register Q output in the scan chain.
- To load the scan chain, SCAN is asserted and 8 CLK pulses are given to load the first two ranks of 4-bit registers with data. Then SCAN is deasserted and CLK is asserted for one cycle to operate the circuit normally with predefined inputs. SCAN is then reasserted and CLK asserted eight times to read the stored data out. At the same time, the new register contents can be shifted in for the next test.
- . Testing proceeds in this manner of serially clocking the data through the scan register to the right point in the circuit, running a single system clock cycle and serially clocking the data out for observation. In this scheme, every input to the combinational block can be controlled and every output can be observed.

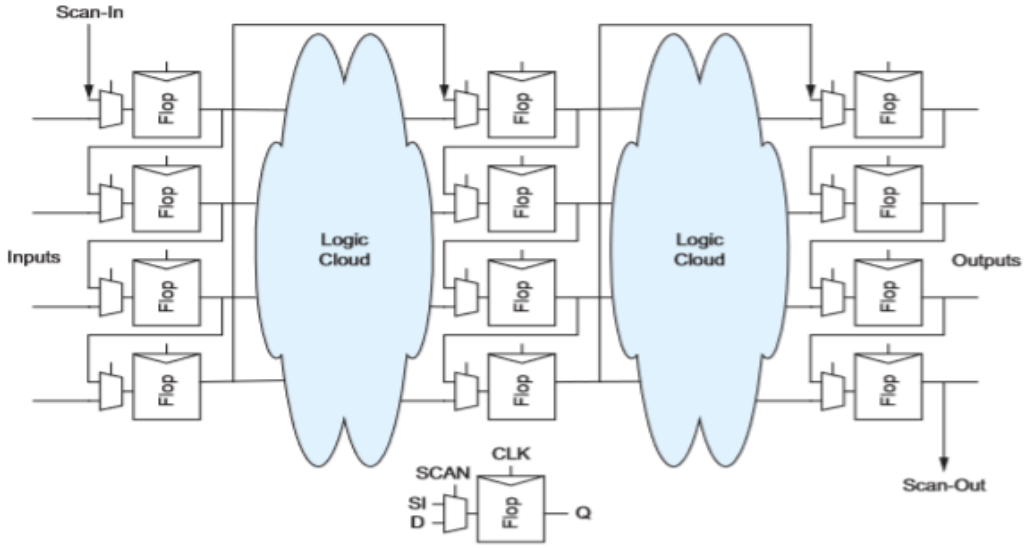


Fig. Scan based testing

- Test generation for this type of test architecture can be highly automated.
- The prime disadvantage is the area and delay impact of the extra multiplexer in the scan register.

Parallel Scan:

Serial scan chains can become quite long, and the loading and unloading can dominate testing time. A simple method/solution is to split the chains into smaller segments. This can be done on a module-by-module basis or completed automatically to some specified scan length. This method is called 'Random Access Scan'.

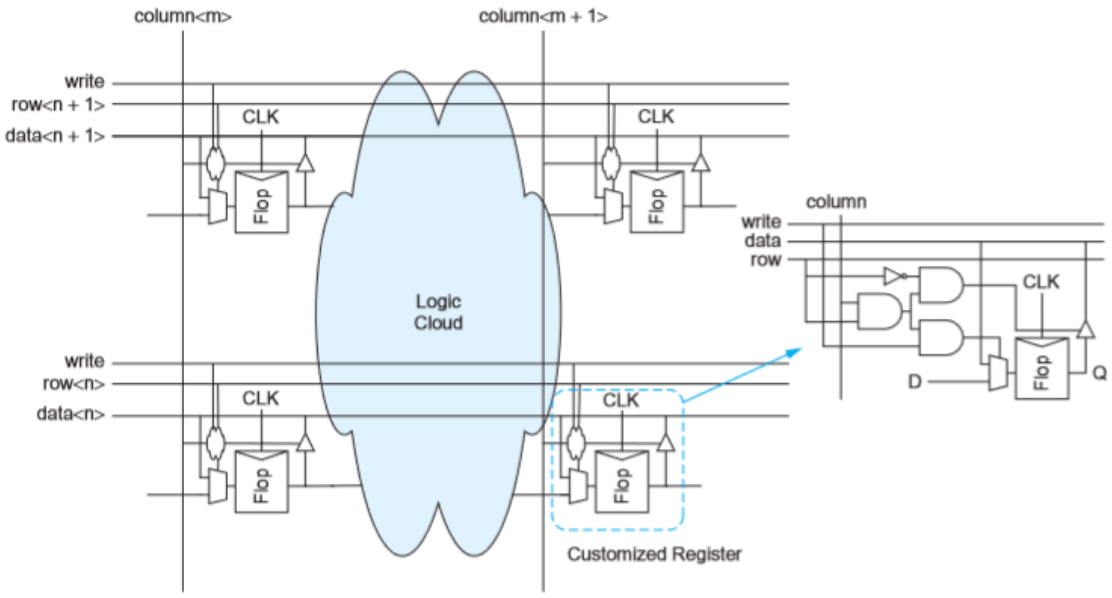


Fig. Parallel Scan - basic architecture

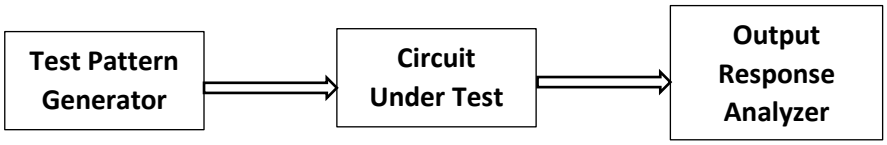
Fig shows a two-by-two register section. Each register receives a column (column<m>) and row (row<n>) access signal along with a row data line (data<n>). A global write signal (write)

is connected to all registers. By asserting the row and column access signals in conjunction with the write signal, any register can be read or written.

Built–In Self-Test (BIST):

Built-in test techniques, as their names suggest, rely on augmenting (additional) circuits to allow them to perform operations upon themselves that prove correct operation. These techniques add area to the chip for the test logic, but reduce the test time required and thus can lower the overall system cost.

The structure of BIST is shown below.



- One method of testing a module is to use 'signature analysis' or 'cyclic redundancy checking'. This involves using a pseudo-random sequence generator to produce the input signals for a section of combinational circuitry and a signature analyzer to observe the output signals.
- A PRSG of length n is constructed from a linear feedback shift register (LFSR), which in turn is made of n flip-flops connected in a serial fashion.
- The XOR of particular outputs are fed back to the input of the LFSR. An n-bit LFSR will cycle through 2^{n-1} states before repeating the sequence. One problem seen is that it is not possible to generate pattern with all 0's.

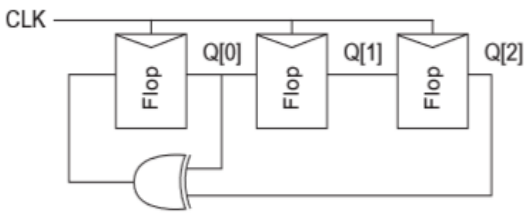


Fig. Linear Feed Back Shift Register (LFSR)

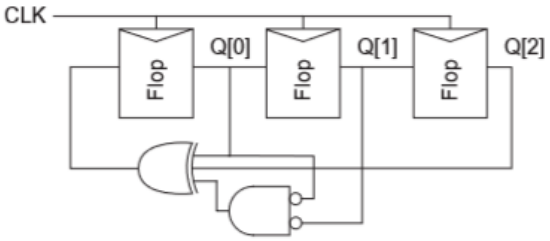


Fig. Complete Feedback Shift Register (CFSR)

- A complete feedback shift register (CFSR), shown in Fig, includes the zero state that may be required in some test situations. An n-bit LFSR is converted to an n-bit CFSR by adding an n – 1 input NOR gate connected to all but the last bit. When in state 0…01, the next state is 0…00.
- A signature analyzer receives successive outputs of a combinational logic block and produces a syndrome that is a function of these outputs. The syndrome is reset to 0, and then XORed with the output on each cycle.
- The syndrome is present in each cycle so that a fault in one bit is unlikely to cancel itself out. At the end of a test sequence, the LFSR contains the syndrome that is a function of all previous outputs. This can be compared with the correct syndrome (derived by running a test program on the good logic) to determine whether the circuit is good or bad.

BILBO – Built-In Logic Block Observation:

- The combination of signature analysis and the scan technique is the formation of BILBO

- The 3-bit BIST register shown in Fig is a scannable, resettable register that also can serve as a pattern generator and signature analyzer.
- This structure can operate in different mode as shown in table below

C[1]	C[0]	Mode
0	0	Scan
0	1	Test
1	0	Reset
1	1	Normal

- In the reset mode (10), all the flip-flops are synchronously initialized to 0. In normal mode (11), the flip-flops behave normally with their D input and Q output. In scan mode (00), the flip-flops are configured as a 3-bit shift register between SI and SO. In test mode (01), the register behaves as a pseudo-random sequence generator or signature analyzer.

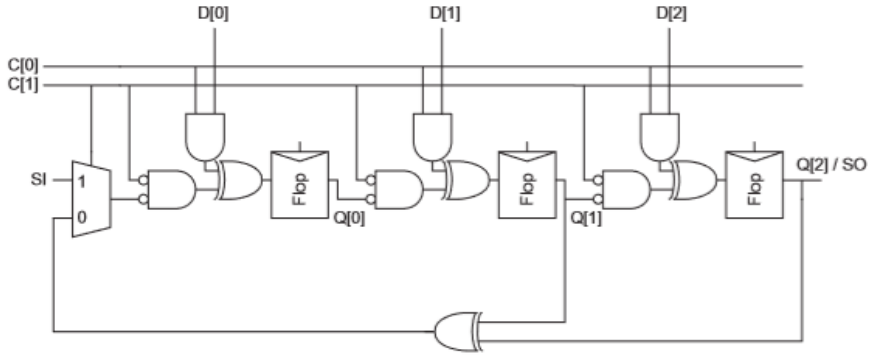


Fig. 3 bit - Register with BILBO

- In summary, BIST is performed by first resetting the syndrome in the output register. Then both registers are placed in the test mode to produce the pseudo-random inputs and calculate the syndrome. Finally, the syndrome is shifted out through the scan chain.

Memory BIST:

On many chips, memories involves with majority of the transistors. A robust testing methodology must be applied to provide reliable parts. In a typical MBIST scheme, multiplexers are placed on the address, data, and control inputs for the memory to allow direct access during test. During testing, a state machine uses these multiplexers to directly write a checkerboard pattern of alternating 1s and 0s. The data is read back, checked, then the inverse pattern is also applied and checked. ROM testing is even simpler: The contents are read out to a signature analyzer to produce a syndrome.