

# Module 5: Memory

Objectives: At the end of this unit we will be able to understand

- System timing consideration
- Storage / Memory Elements
  - dynamic shift register
    - 1T and 3T dynamic memory
    - 4T dynamic and 6T static CMOS memory
- Array of memory cells

## System timing considerations:

- Two phase non-overlapping clock
- $\phi_1$  leads  $\phi_2$
- Bits to be stored are written to register and subsystems on  $\phi_1$
- Bits or data written are assumed to be settled before  $\phi_2$
- $\phi_2$  signal used to refresh data
- Delays assumed to be less than the intervals between the leading edge of  $\phi_1$  &  $\phi_2$
- Bits or data may be read on the next  $\phi_1$
- There must be atleast one clocked storage element in series with every closed loop signal path

## Storage / Memory Elements:

The elements that we will be studying are:

- Dynamic shift register
- 3T dynamic RAM cell
- 1T dynamic memory cell
- Pseudo static RAM / register cell
- 4T dynamic & 6T static memory cell
- JK FF circuit
- D FF circuit

## Dynamic shift register:

**Circuit diagram: Refer to unit 4(ch 6.5.4)**

**Power dissipation**

- static dissipation is very small
- dynamic power is significant
- dissipation can be reduced by alternate geometry

**Volatility**

- data storage time is limited to 1msec or less

## 3T dynamic RAM cell:

### Circuit diagram

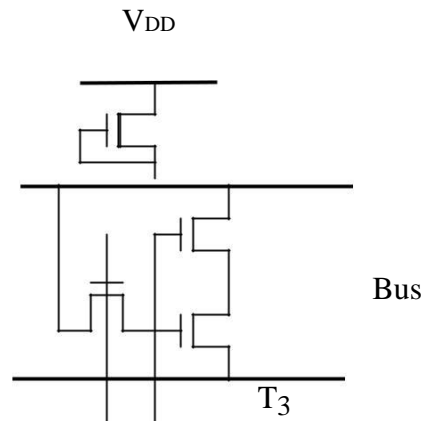


Figure 7.1: 3T Dynamic RAM Cell

### Working

- RD = low, bit read from bus through T1, WR = high, logic level on bus sent to Cg of T2, WR = low again
- Bit level is stored in Cg of T2, RD=WR=low
- Stored bit is read by RD = high, bus will be pulled to ground if a 1 was stored else 0 if T2 non-conducting, bus will remain high.

### Dissipation

- Static dissipation is nil
- Depends on bus pull-up & on duration of RD signal & switching frequency

### Volatility

- Cell is dynamic, data will be there as long as charge remains on Cg of T2

## 1T dynamic memory cell:

### Circuit diagram

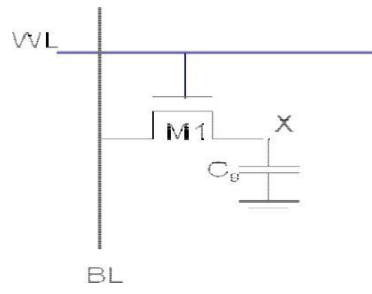


Figure 7.2: 1T Dynamic RAM Cell

## Working

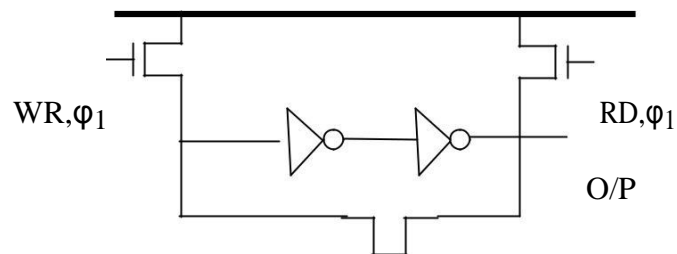
- Row select (RS) = high, during write from R/W line  $C_m$  is charged
- data is read from  $C_m$  by detecting the charge on  $C_m$  with RS = high
- cell arrangement is bit complex.
- solution: extend the diffusion area comprising source of pass transistor, but  $C_d \lll C_{gchannel}$
- another solution : create significant capacitor using poly plate over diffusion area.
- $C_m$  is formed as a 3-plate structure
- with all this careful design is necessary to achieve consistent readability

## Dissipation

- no static power, but there must be an allowance for switching energy during read/write

## Pseudo static RAM / register cell:

### Circuit diagram



T

$\phi_2$

Figure 7.3: nMOS pseudo-static memory Cell

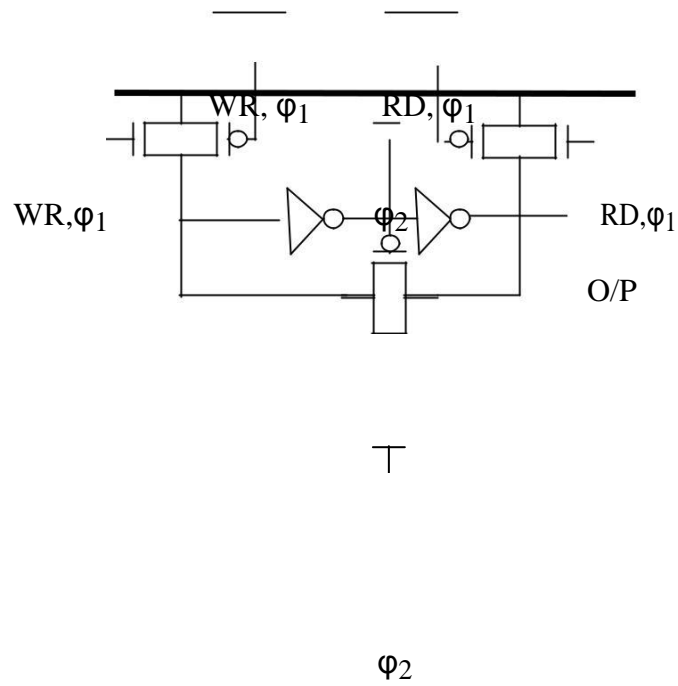


Figure 7.4: CMOS pseudo-static memory Cell

## Working

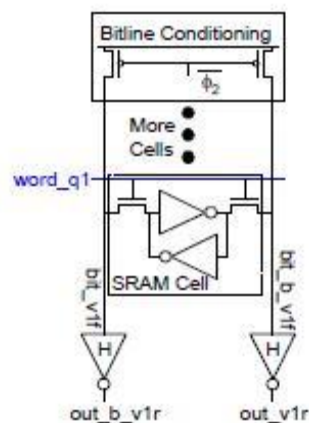
- dynamic RAM need to be refreshed periodically and hence not convenient
- static RAM needs to be designed to hold data indefinitely
- One way is connect 2 inverter stages with a feedback.
- say  $\phi_2$  to refresh the data every clock cycle
- bit is written on activating the WR line which occurs with  $\phi_1$  of the clock
- bit on Cg of inverter 1 will produce complemented output at inverter 1 and true at output of inverter 2
- at every  $\phi_2$ , stored bit is refreshed through the gated feedback path
- stored bit is held till  $\phi_2$  of clock occurs at time less than the decay time of stored bit
- to read RD along with  $\phi_1$  is activated

Note:

- WR and RD must be mutually exclusive
- $\phi_2$  is used for refreshing, hence no data to be read, if so charge sharing effect, leading to destruction of stored bit
- cells must be stackable, both side-by-side & top to bottom
- allow for other bus lines to run through the cell

## 4T dynamic & 6T static memory cell:

### Circuit diagram



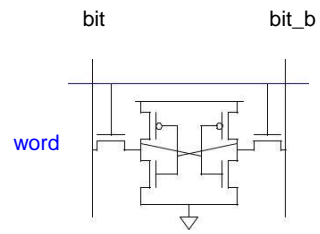


Figure 7.4: Dynamic and static memory cells



## Working

- uses 2 buses per bit to store bit and bit'
- both buses are precharged to logic 1 before read or write operation.
- write operation
- read operation

## Write operation

- both bit & bit' buses are precharged to VDD with clock  $\phi_1$  via transistor T5 & T6
- column select line is activated along with  $\phi_2$
- either bit or bit' line is discharged along the I/O line when carrying a logic 0
- row & column select signals are activated at the same time => bit line states are written in via T3 & T4, stored by T1 & T2 as charge

## Read operation

- bit and bit' lines are again precharged to VDD via T5 & T6 during  $\phi_1$
- if 1 has been stored, T2 ON & T1 OFF
- bit' line will be discharged to VSS via T2
- each cell of RAM array be of minimum size & hence will be the transistors
- implies incapable of sinking large charges quickly
- RAM arrays usually employ some form of sense amplifier
  - T1, T2, T3 & T4 form as flip-flop circuit
  - if sense line to be inactive, state of the bit line reflects the charge present on gate capacitance of T1 & T3
  - current flowing from VDD through an on transistor helps to maintain the state of bit lines

## Testability

Objective: At the end of this unit we will be able to understand

- Design for testability (DFT)
- DFT methods for digital circuits:
  - Ad-hoc methods
  - Structured methods:
    - Scan
    - Level Sensitive Scan Design
    - Boundary scan

- Other Scan Techniques

**Definition:**

*Design for testability* (DFT) refers to those design techniques that make test generation and test application cost-effective.

**Some terminologies:**

**Input / output (I/O) pads**

- Protection of circuitry on chip from damage
- Care to be taken in handling all MOS circuits
- Provide necessary buffering between the environments On & OFF chip
- Provide for the connections of power supply
- Pads must be always placed around the peripheral

Minimum set of pads include:

- VDD connection pad
- GND(VSS) connection pad
- Input pad
- Output pad
- Bidirectional I/O pad

Designer must be aware of:

- nature of circuitry
- ratio/size of inverters/buffers on which output lines are connected
- how input lines pass through the pad circuit (pass transistor/transmission gate)

**System delays**

Buses:

- convenient concept in distributing data & control through a system
- bidirectional buses are convenient
- in design of datapath
- problems: capacitive load present

- largest capacitance
- sufficient time must be allowed to charge the total bus
- clock  $\phi_1$  &  $\phi_2$

Control paths, selectors & decoders

1. select registers and open pass transistors to connect cells to bus
2. Data propagation delay bus
3. Carry chain delay

## Faults and Fault Modeling

A fault model is a model of how a physical or parametric fault manifests itself in the circuit Operation. Fault tests are derived based on these models

Physical Faults are caused due to the following reasons:

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

Physical faults cause Electrical and Logical faults

Logical Faults are:

- Single/multiple stuck-at (*most used*)
- CMOS stuck-open
- CMOS stuck-on
- AND / OR Bridging faults

Electrical faults are due to short, opens, transistor stuck on, stuck open, excessive steady state currents, resistive shorts and open.

## Design for Testability

Two key concepts

- Observability
- Controllability

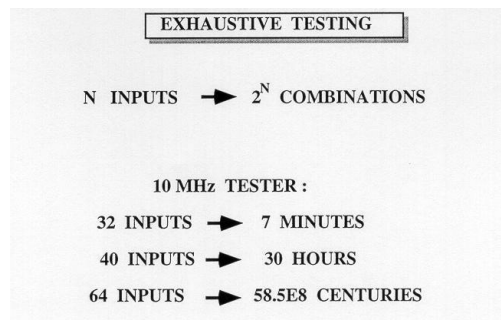
DFT often is associated with design modifications that provide improved access to internal circuit elements such that the local internal state can be controlled (controllability) and/or observed (observability) more easily. The design modifications can be strictly physical in nature (e.g., adding a physical probe point to a net) and/or add active circuit elements to facilitate controllability/observability (e.g., inserting a multiplexer into a net). While controllability and observability improvements for internal circuit elements definitely are important for test, they are not the only type of DFT

## Testing combinational logic

The solution to the problem of testing a purely combinational logic block is a good set of patterns detecting "all" the possible faults.

The first idea to test an N input circuit would be to apply an N-bit counter to the inputs (controllability), then generate all the  $2^N$  combinations, and observe the outputs

for checking (observability). This is called "exhaustive testing", and it is very efficient... but only for few- input circuits. When the input number increase, this technique becomes very time consuming.

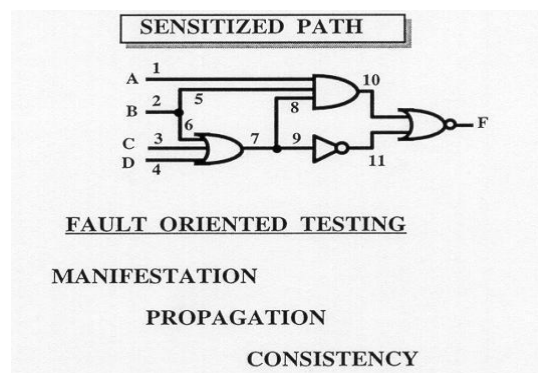


### Sensitized Path Testing

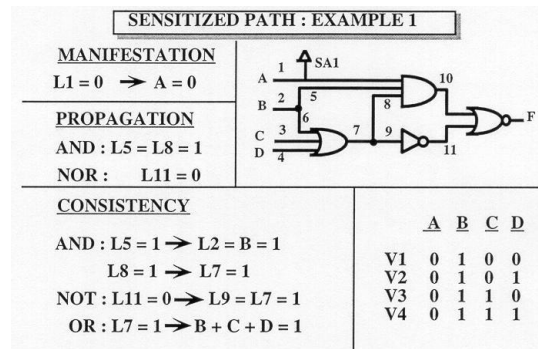
Most of the time, in exhaustive testing, many patterns do not occur during the application of the circuit. So instead of spending a huge amount of time searching for faults everywhere, the possible faults are first enumerated and a set of appropriate vectors are then generated. This is called "single-path sensitization" and it is based on "fault oriented testing".

The basic idea is to select a path from the site of a fault, through a sequence of gates leading to an output of the combinational logic under test. The process is composed of three steps :

- **Manifestation** : gate inputs, at the site of the fault, are specified as to generate the opposite value of the faulty value (0 for SA1, 1 for SA0).
- **Propagation** : inputs of the other gates are determined so as to propagate the fault signal along the specified path to the primary output of the circuit. This is done by setting these inputs to "1" for AND/NAND gates and "0" for OR/NOR gates.
- **Consistency** : or justification. This final step helps finding the primary input pattern that will realize all the necessary input values. This is done by tracing backward from the gate inputs to the primary inputs of the logic in order to receive the test patterns.



**Example1** - SA1 of line1 (L1) : the aim is to find the vector(s) able to detect this fault.



- **Manifestation:** L1 = 0, then input A = 0. In a fault-free situation, the output F changes with A if B,C and D are fixed : for B,C and D fixed, L1 is SA1 gives F = 0, for instance, even if A = 0 (F = 1 for fault-free).
- **Propagation:** Through the AND-gate : L5 = L8 = 1, this condition is necessary for the propagation of the " L1 = 0 ". This leads to L10 = 0. Through the NOR-gate, and since L10 = 0, then L11 = 0, so the propagated manifestation can reach the primary output F. F is then read and compared with the fault-free value: F = 1.

### Improve Controllability and Observability

All "design for test" methods ensure that a design has enough observability and controllability to provide for a complete and efficient testing. When a node has difficult access from primary inputs or outputs (pads of the circuit), a very efficient method is to add internal pads according to this kind of node in order, for instance, to control block B2 and observe block B1 with a probe.

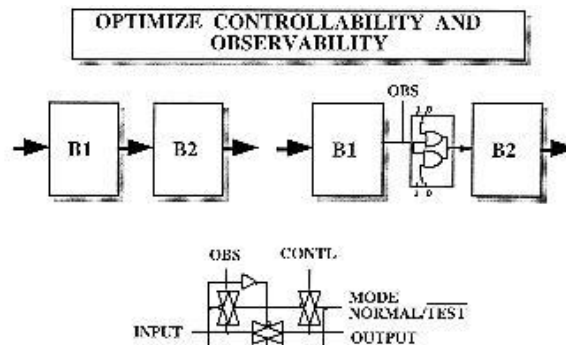


Figure 8.1 Improve Controllability and Observability

It is easy to observe block B1 by adding a pad just on its output, without breaking the link between the two blocks. The control of the block B2 means to set a 0 or a 1 to its input, and also to be transparent to the link B1-B2. The logic functions of this purpose are a NOR-gate, transparent to a zero, and a NAND-gate, transparent to a one. By this way the control of B2 is possible across these two gates.

Another implementation of this cell is based on pass-gates multiplexers performing the same function, but with less transistors than with the NAND and NOR gates (8 instead of 12).

The simple optimization of observation and control is not enough to guarantee a full testability of the blocks B1 and B2. This technique has to be completed with some other techniques of testing depending on the internal structures of blocks B1 and B2.

DRAM's includes the incorporation onto the chip of additional circuits for pattern generation, timing, mode selection, and go-/no-go diagnostic tests.

Advantages of implementing BIST include:

- 1) Lower cost of test, since the need for external electrical testing using an ATE will be reduced, if not eliminated
- 2) Better fault coverage, since special test structures can be incorporated onto the chips
- 3) Shorter test times if the BIST can be designed to test more structures in parallel
- 4) Easier customer support and
- 5) Capability to perform tests outside the production electrical testing environment. The last advantage mentioned can actually allow the consumers themselves to test the chips prior to mounting or even after these are in the application boards.

Disadvantages of implementing BIST include:

- 1) Additional silicon area and fab processing requirements for the BIST circuits
- 2) Reduced access times
- 3) Additional pin (and possibly bigger package size) requirements, since the BIST circuitry need a way to interface with the outside world to be effective and
- 4) Possible issues with the correctness of BIST results, since the on-chip testing hardware itself can fail.

Techniques are:

- compact test: signature analysis
- linear feedback shift register
- BILBO
- self checking technique

### **Compact Test: Signature analysis**

Signature analysis performs polynomial division that is, division of the data out of the device under test (DUT). This data is represented as a polynomial  $P(x)$  which is divided by a characteristic polynomial  $C(x)$  to give the signature  $R(x)$ , so that

$$R(x) = P(x)/C(x)$$

This is summarized as in figure 8.16.

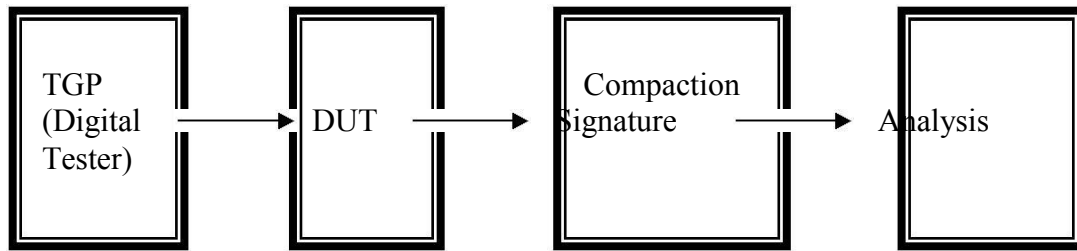


Figure 8.16: BIST – signature analysis

**Linear feedback shift register (LFSR):**

An LFSR is a shift register that, when clocked, advances the signal through the register from one bit to the next most-significant bit. Some of the outputs are combined in exclusive-OR configuration to form a feedback mechanism. A linear feedback shift register can be formed by performing exclusive-OR (Figure 8.16) on the outputs of two or more of the flip-flops together and feeding those outputs back into the input of one of the flip-flops.

LFSR technique can be applied in a number of ways, including random number generation, polynomial division for signature analysis, and n-bit counting. LFSR can be series or parallel, the differences being in the operating speed and in the area of silicon occupied; Parallel LFSR being faster but larger than serial LFSR.

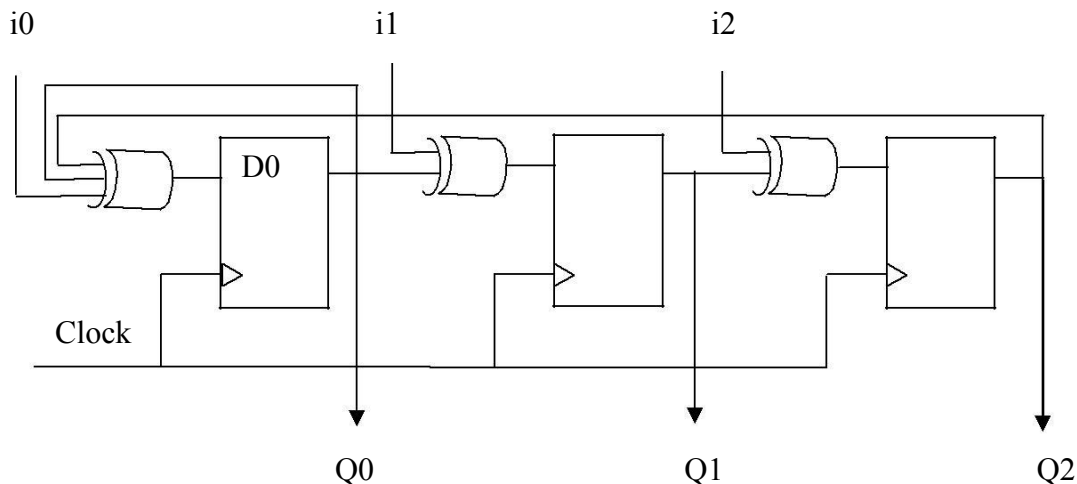


Figure 8.16: Linear feedback shift register

**Built-in logic block observer (BILBO):**

BILBO is a built-in test generation scheme which uses signature analysis in conjunction with a scan path. The major component of a BILBO is an LFSR with a few gates (Figure 8.17).

A **BILBO** register (**built-in logic block observer**) combines normal flipflops with a few additional gates to provide four different functions. The example circuit shown in the applet realizes a four-bit register. However, the generalization to larger bit-widths should be obvious, with the XOR gates in the LFSR feedback path chosen to implement a good polynomial for the given bit-width.

When the **A** and **B** control inputs are both 1, the circuit functions as a normal parallel D-type register.

When both **A** and **B** inputs are 0, the D-inputs are ignored (due to the AND gate connected to A), but the flipflops are connected as a shift-register via the NOR and XOR gates. The input to the first flipflop is then selected via the multiplexer controlled by the **S** input. If the **S** input is 1, the multiplexer transmits the value of the external **SIN** shift-in input to the first flipflop, so that the BILBO register works as a normal shift-register. This allows to initialize the register contents using a single signal wire, e.g. from an external test controller.

If all of the **A**, **B**, and **S** inputs are 0, the flipflops are configured as a shift-register, again, but the input bit to the first flipflop is computed by the XOR gates in the LFSR feedback path. This means that the register works as a standard LFSR pseudorandom pattern generator, useful to drive the logic connected to the Q outputs. Note that the start value of the LFSR sequence can be set by shifting it in via the **SIN** input.

Finally, if **B** and **S** are 0 but **A** is 1, the flipflops are configured as a shift-register, but the input value of each flipflop is the XOR of the D-input and the Q-output of the previous flipflop. This is exactly the configuration of a standard LFSR signature analysis register.

Because a BILBO register can be used as a pattern generator for the block it drives, as well provide signature-analysis for the block it is driven by, a whole circuit can be made self-testable with very low overhead and with only minimal performance degradation (two extra gates before the D inputs of the flipflops).

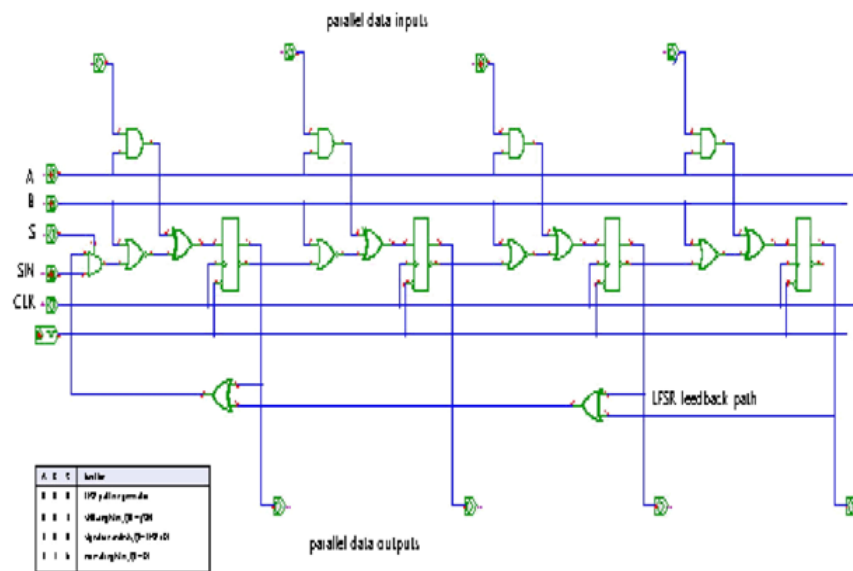


Figure 8.17: BIST – BILBO



**Self-checking techniques:**

It consists of logic block and checkers should then obey a set of rules in which the logic block is 'strongly fault secure' and the checker 'strongly code disjoint'. The code use in data encoding depends on the type of errors that may occur at the logic block output. In general three types are possible:

- Simple error: one bit only affected at a time.
- Unidirectional error: multiple bits at 1 instead of 0 (or 0 instead of 1)
- Multiple errors: multiple bits affected in any order.

Self-checking techniques are applied to circuits in which security is important so that fault tolerance is of major interest. Such technique will occupy more area in silicon than classical techniques such as functional testing but provide very high test coverage.