Memory fault models and testing

Abhilash Kaushal - June 29, 2015

A different set of fault models and testing techniques is required for memory blocks vs. logic. MBIST algorithms that are used to detect faults inside memory are based upon these fault models. This article discusses different types of memory fault models.

Memory fault models - Single cell faults

**Stuck at (SAFs):** Stuck at faults in memory is the one in which the logic value of a cell (or line in the sense amplifier or driver) is always 0 or 1.

![State diagram for single cell faults](image)

**Transition Faults (TFs):** In transition faults a cell fails to make a (0 to 1) transition or a (1 to 0) transition when it is written; up transition fault is denoted as <0w1/0/- > and a down transition fault is denoted as < 1w0/1/- >

![State diagram for transition faults](image)

**Write destructive faults (WDFs):** A non transition write operation in a memory cell causes the cell to flip. There are two types of Write destructive faults:
1) Memory cell in state 0, write 0 on it. Cell becomes 1. Denoted as <0w0/1/>.

2) Memory cell in state 1, write 1 on it. Cell becomes 0. Denoted as <1w1/0/>.

State diagram for write destructive faults

**Read destructive Faults (RDFs):** A read operation is performed to the cell causes inversion of the value in the cell and returns the incorrect value. There are two types of read destructive faults:

1) Memory cell in state 0, read 0 on it. Cell becomes 1. Denoted as <0r0/1/1>.

2) Memory cell in state 1, read 1 on it. Cell becomes 0. Denoted as <1r1/0/0>.

State diagram for read destructive faults

**Incorrect read Faults (IRFs):** A read operation is performed to the cell returns an incorrect value, while the state of the memory cell is not changed. There are two types of incorrect read faults:

1) Memory cell in state 0, read 0 on it. Cell remains 0 but read operation returns 1. Denoted as <0r0/0/1>.

2) Memory cell in state 1, read 1 on it. Cell remains 1 but read operation returns 0. Denoted as <1r1/1/0>.
State diagram for incorrect read faults

**Deceptive read destructive Faults (DRDFs):** A read operation is performed to the cell causes the inversion of the value in the cell and returns the correct value. There are two types of deceptive read destructive faults:

1) Memory cell in state 0, read 0 on it. Cell becomes 1 but read operation returns 0. Denoted as <0r0/1/0>

2) Memory cell in state 1, read 1 on it. Cell becomes 0 but read operation returns 1. Denoted as <1r1/0/1>

**State diagram for Deceptive read destructive faults**

**Stuck open Faults (SOFs):** In this Fault memory word cannot be accessed. When the sense amplifier contains a latch then during a read operation the previously read value may be produced. If differential amplifier behaves as a buffer it can be modeled as stuck at fault.

**Address decoder Faults (ADFs):** Row and column decoder comprises the address decoder of a memory. From the context of memory testing four types of faults are considered in address decoder.

- AF1: With certain address no word can be accessed
- AF2: No address from which a particular word can be accessed.
- AF3: With certain address multiple words can be accessed simultaneously.
- AF4: A certain word can be accessed with multiple addresses.

**Memory fault models - Two cell faults**

**Coupling Faults (CFs):** Coupling faults are faults in which fault occurs in a cell because of coupling with other cells.

- There can be exponential number of combinations in which a cell can be coupled with others cells.
In the widely used coupling fault model it is assumed that any “two” cells can be coupled together leading to irregular behavior in these two cells; it is called 2-cell coupling fault model. So if there are \( n \) cells in a memory then there can be \( ^nC_2 \) number of 2-cell coupling faults.

**Inversion Coupling Faults (CFin):** An upper (0 to 1) or lower (1 to 0) transition write operation in aggressor word causes an inversion in the cell of a victim word. These faults are never being observed in a faulty memory cells and are just defined due to historical reasons [1]. Therefore, they are not included in the linked faults list.

1) Rising : < ? | ? > (Implying 0 to 1 change in cell \( a \) complements the content of cell \( v \))
2) Falling : < ? | ? > (Implying 1 to 0 change in cell \( a \) complements the content of cell \( v \))

**Idempotent coupling Faults (CFid):** An upper (0 to 1) or lower (1 to 0) transition write operation in cell of aggressor word forces a certain value (0 or 1) in a cell of the victim word. These faults are subset of CFds fault (explained later) as a write operation causes the content of the \( v \) cell to flip from its previous state.
1) Rising 0: < ? | 0 > 0 to 1 change in a cell $a$ sets the content of cell $v$ to be 0.
2) Rising 1: < ? | 1 > 0 to 1 change in a cell $a$ sets the content of cell $v$ to be 1.
3) Falling 0: < ? | 0 > 1 to 0 change in a cell $a$ sets the content of cell $v$ to be 0.
4) Falling 1: < ? | 1 > 1 to 0 change in a cell $a$ sets the content of cell $v$ to be 1.

State diagram of Idempotent coupling faults

**Static coupling Faults (CFst):** A given value 0 or 1 of the cell in the aggressor word forces a certain value 0 or 1 in a cell of the victim word.

1) (0 in cell $a$ sets the content of the cell $v$ to be 0)
2) (0 in cell $a$ sets the content of the cell $v$ to be 1)
3) (1 in cell $a$ sets the content of the cell $v$ to be 0)
4) (1 in cell $a$ sets the content of the cell $v$ to be 1)
State diagram for static coupling Faults

Disturb cell coupling Faults (CFds): A cell in the victim word is disturbed (i.e., makes an upper or a lower transition) due to a write or read operation over the aggressor word.

1) \( < 0r0, ? > \), 0 read operation on cell a cause’s upper (0 to 1) transition in cell v.
2) \( < 0r0, ? > \), 0 read operation on cell a cause’s lower (1 to 0) transition in cell v.
3) \( < 1r1, ? > \), 1 read operation on cell a cause’s upper (0 to 1) transition in cell v.
4) \( < 1r1, ? > \), 1 read operation on cell a cause’s lower (1 to 0) transition in cell v.
5) \( < 0w1, ? > \), (0 to 1) write operation on cell a cause’s upper (0 to 1) transition in cell v.
6) \( < 0w1, ? > \), (0 to 1) write operation on cell a cause’s lower (1 to 0) transition in cell v.
7) \( < 0w0, ? > \), (0 to 0) write operation on cell a cause’s upper (0 to 1) transition in cell v.
8) \( < 0w0, ? > \), (0 to 0) write operation on cell a cause’s lower (1 to 0) transition in cell v.
9) \( < 1w0, ? > \), (1 to 0) write operation on cell a cause’s upper (0 to 1) transition in cell v.
10) \( < 1w0, ? > \), (1 to 0) write operation on cell a cause’s lower (1 to 0) transition in cell v.
11) \( < 1w1, ? > \), (1 to 1) write operation on cell a cause’s upper (0 to 1) transition in cell v.
12) \( < 1w1, ? > \), (1 to 1) write operation on cell a cause’s lower (1 to 0) transition in cell v.
Transition coupling faults (CFtr): A write transition operation applied to a cell of the victim word does not cause a transition if the aggressor word is in a given state.

1) (0w1/0/-; 0) 0 to 1 transition write operation in cell $v$ does not cause transition when content of cell $a$ is 0

2) (0w1/0/-; 1) 0 to 1 transition write operation in cell $v$ does not cause transition when content of cell $a$ is 1

3) (1w0/1/-; 0) 1 to 0 transition write operation in cell $v$ does not cause transition when content of cell $a$ is 0

4) (1w0/1/-; 1) 1 to 0 transition write operation in cell $v$ does not cause transition when content of cell $a$ is 1
State diagram of transition coupling faults

Write destructive coupling faults (CFwd): A non-transition write operation applied to a cell of the victim word flips the cell if the aggressor word is in a given state.

1) \((0w0/1/-; 0)\) 0 to 0 non-transition write operation in cell \(v\) causes transition when content of cell \(a\) is 0

2) \((0w0/1/-; 1)\) 0 to 0 non-transition write operation in cell \(v\) causes transition when content of cell \(a\) is 1

3) \((1w1/0/-; 0)\) 1 to 1 non-transition write operation in cell \(v\) causes transition when content of cell \(a\) is 0

4) \((1w1/0/-; 1)\) 1 to 1 non-transition write operation in cell \(v\) causes transition when content of cell \(a\) is 1
Read destructive coupling faults (CFrd): A read operation applied to a cell of the victim word causes a transition and returns an incorrect value, if the aggressor word is in a given state.

1) $(0r0/1/1; 0)$ 0 read operation in cell v causes transition in the cell and returns the incorrect value when content of cell a is 0

2) $(0r0/1/1; 1)$ 0 read operation in cell v causes transition in the cell and returns the incorrect value when content of cell a is 1

3) $(1r1/0/0; 0)$ 1 read operation in cell v causes transition in the cell and returns the incorrect value when content of cell a is 0

4) $(1r1/0/0; 1)$ 1 read operation in cell v causes transition in the cell and returns the incorrect value when content of cell a is 1
State diagram of read destructive coupling Faults

Incorrect read coupling faults (CFir): A read operation applied to a cell of the victim word returns an incorrect value if the aggressor word is in a given state. The state of the victim word is not changed.

1) (0r0/0/1; 0) 0 read operation in cell v returns the incorrect value when content of cell a is 0
2) (0r0/0/1; 1) 0 read operation in cell v returns the incorrect value when content of cell a is 1
3) (1r1/1/0; 0) 1 read operation in cell v returns the incorrect value when content of cell a is 0
4) (1r1/1/0; 1) 1 read operation in cell v returns the incorrect value when content of cell a is 1
More faults

Deceptive read destructive coupling faults (CFdrd): A read operation applied to a cell of the victim word causes a transition and returns the correct value, if the aggressor word is in a given state.

1) \((0r0/1/0; 0)\) 0 read operation in cell \(v\) causes the transition in the cell & returns the correct value when content of cell \(a\) is 0

2) \((0r0/1/0; 1)\) 0 read operation in cell \(v\) causes the transition in the cell & returns the correct value when content of cell \(a\) is 1

3) \((1r1/0/1; 0)\) 1 read operation in cell \(v\) causes the transition in the cell & returns the correct value when content of cell \(a\) is 0

4) \((1r1/0/1; 1)\) 1 read operation in cell \(v\) causes the transition in the cell & returns the correct value when content of cell \(a\) is 1
Bridging Faults: A bridging fault is a short circuit between two or more cells. Similar to consideration for coupling faults, it is assumed that only two cells can be involved in a bridging fault to keep the faults within practical number. There are two types of bridging faults:

AND Bridging Fault: Both the aggressor as well as victim cell would have the value equal to logical AND of both the cell.

\[ <0,0 | 0,0 >, <0,1 | 0,0 >, <1,0 | 0,0 >, <1,1 | 1,1 > \]

OR Bridging Fault: Both the aggressor as well as victim cell would have the value equal to logical OR of both the cell.

\[ <0,0 | 0,0 >, <0,1 | 1,1 >, <1,0 | 1,1 >, <1,1 | 1,1 > \]

Neighborhood pattern sensitive coupling Faults: Fault that occurs due to coupling of the cell under test with the pattern formed by neighborhood cells. Victim cell behaves incorrectly depending upon the value on adjacent cells.
Active NPSF (ANPSF): The victim cell makes a transition to a particular value depending upon the pattern formed by nearby cells (either Type – 1 or Type -2).

For ex- 1 < 0, 0, 1, 0 | 0 > represents the Active NPSF where the cell under test initially has value of 1, the pattern made by neighboring cells is 0000 (values at cell no. 0,1,3,4 respectively) and fault effect at cell under test is 0 when a 1 to 0 transition in made in cell 3.

Passive NPSF (PNPSF): This victim cell fails to make a transition due to neighborhood pattern formed by nearby cells.

1)  <↑, 0 >: cell under test doesn’t change from 0 to 1 due to presence of neighborhood pattern (initial value of cell under test is 0).
2)  <↓, 1 >: cell under test doesn’t change from 1 to 0 due to presence of neighborhood pattern (initial value of cell under test is 1).
3)  <?, x >: cell under test doesn’t change regardless of content of the cell under test due to presence of neighborhood pattern.

Static NPSF (SNPSF):
Due to Neighborhood pattern around the victim cell value gets stuck to either 0 or 1.

**Linked Faults (LFs):** Linked Faults can be defined as the combination of single/two-cell faults in which faulty behavior of the cell due to a fault can be masked by faulty behavior due to another fault in the same cell. Linked Faults takes place when more than one (Fault primitive) FP is sensitized in defective cell of a memory.

The definition of an LF is as follows: LF1 = FP1 -> FP2. This means that linked fault consists of FP1 linked to FP2. If the sensitizing operation sequence (S1) of FP1 is applied first, it sensitizes a fault in the victim-cell, and when the (S2) of FP2 is applied next, it also sensitizes a fault in the same victim-cell, but fault effect is opposite to that produced by FP1. The net result would be that the fault effect of FP2 masks the fault effect of the FP1.

If we assume that the faulty behavior of a memory contains two FPs that share the same v-cell, then FP1 = <S1/F1/R1> is said to be linked to FP2 = <S2/F2/R2> (denoted as FP1 -> FP2) if the following three conditions are satisfied.

1) **Read operations of FP1 and FP2 do not detect a fault [1]**

   This condition guarantees that both FP1 and FP2 are not detectable by read operations that S1 or S2 may contain as if the FPs is detected through read operations there would be no point talking about the Linked Faults. For ex- RDF1 <0r0, 1, 1 > cannot be linked with any other FPs as fault is detected at the read operation at the end only.

2) **FP2 masks FP1 [1].**

   This means that F2=! F1. This condition ensures that the faulty behavior of FP2 hides the faulty behavior sensitized by FP1 by masking it.

3) **FP2 is compatible with FP1 [1].**

   This condition applies only in the case S2 of the FP2 is applied immediately after S1 to the same cell as the a-cell or the v-cell of FP1. In that case, the final state of the a-cell (or of the v-cell) after performing S1 should be the same as the initial state required by S2 of FP2.

**LF1:** This type of faults is based on a combination of two static single-cell faults. Both faults have the same victim cell. Different possible combinations for Linked faults are enumerated in the table below: [1]
FP2 is not compatible with FP1
FP2 does not mask FP1
Fault is detected through the read operation of FP1 or FP2
FP1 can be linked to FP2 since it satisfies all conditions

There are total 12 instances of single cell linked faults (LF1s),

| L1  | TF0 -> WDF0 | L7  | WDF1 -> WDF0 |
| L2  | TF0 -> RDF1 | L8  | WDF1 -> RDF0 |
| L3  | TF1 -> WDF0 | L9  | DRDF0 -> WDF1 |
| L4  | TF1 -> RDF0 | L10 | DRDF0 -> RDF1 |
| L5  | WDF0 -> WDF1 | L11 | DRDF1 -> WDF0 |
| L6  | WDF0 -> RDF1 | L12 | DRDF1 -> RDF0 |

Instances of LF1 faults[1]

LF2: These are based on a combination of two two-cell FPs, or on a combination of a single-cell FP and a two-cell FP. They are therefore divided into three subgroups:

a) The LF2aa: This LF is based on a combination of two two-cell FPs; both FPs have the same a-cell as well as the same v-cell.

b) The LF2av: This LF is based on a combination of one two-cell and one single-cell FPs; whereby the two-cell FP is sensitized first.

c) The LF2va: This LF is also a combination of one single-cell and one two-cell FPs. However, in this case, first the single-cell FP should be sensitized; thereafter the two-cell FP is sensitized.

Similarly a list of other Linked faults can be made
Instances of LF2aa Faults [1]

| L13  | CFwd < x:0w0/1/> -> CFwd < x:1w1/0/> | L9   | CFds < xOy:0/1/> -> RDF < 1r1/0/> |
| L14  | CFwd < x:1w1/0/> -> CFwd < x:0w0/1/> | L10  | CFds < xOy:1/0/> -> RDF < 0r0/1/> |
| L15  | CFDrd < x:0r0/1/> -> CFwd < x:1w1/0/> | L11  | CFtr < x:1w0/1/> -> RDF < 1r1/0/> |
| L16  | CFDrd < x:1r1/0/> -> CFwd < x:0w0/1/> | L12  | CFtr < x:0w0/1/> -> RDF < 0r0/1/> |
|      | LF2aa (FP1 -> CFrd)                  | L13  | CFwd < x:0w0/1/> -> RDF < 1r1/0/> |
| L17  | CFds < xOy:0/1/> -> CFrd < y:1r1/0/> | L14  | CFwd < x:1w1/0/> -> RDF < 0r0/1/> |
| L18  | CFds < xOy:1/0/> -> CFrd < y:0r0/1/> | L15  | CFdrd < x:0r0/1/> -> RDF < 1r1/0/> |
| L19  | CFtr < x:1w0/1/> -> CFrd < x:1r1/0/> | L16  | CFdrd < x:1r1/0/> -> RDF < 0r0/1/> |

Instances of LF2av Faults [1]

| L1   | TF <1w0/1/-> -> CFds < xOy:1/0/> | L4   | TF <0w1/0/-> -> CFds < xOy:0/1/> |
| L2   | TF <1w0/1/-> -> CFwd < x:1w1/0/> | L5   | TF <0w1/0/-> -> CFwd < x:0w0/1/> |
| L3   | TF <1w0/1/-> -> CFrd < x:1r1/0/0> | L6   | TF <0w1/0/-> -> CFrd < x:0r0/1/1> |
| L7   | WDF <0w0/1/-> -> CFds < xOy:1/0/> | L10  | WDF <1w1/0/-> -> CFds < xOy:0/1/> |
| L8   | WDF <0w0/1/-> -> CFwd < x:1w1/0/> | L11  | WDF <1w1/0/-> -> CFwd < x:0w0/1/> |
| L9   | WDF <0w0/1/-> -> CFrd < x:1r1/0/0> | L12  | WDF <1w1/0/-> -> CFrd < x:0r0/1/1> |
| L13  | DRDF <0r0/1/0> -> CFds < xOy:1/0/> | L16  | DRDF <1r1/0/1> -> CFds < xOy:0/1/> |
| L14  | DRDF <0r0/1/0> -> CFwd < x:1w1/0/> | L17  | DRDF <1r1/0/1> -> CFwd < x:0w0/1/> |
| L15  | DRDF <0r0/1/0> -> CFrd < x:1r1/0/0> | L18  | DRDF <1r1/0/1> -> CFrd < x:0r0/1/1> |

Instances of LF2av Faults[1]

**LF3:** LF3s describe linking two two-cell FPs with different aggressor-cells and the same victim-cells. Following are the instances possible of LF3 faults.
Instances of LF3 faults

Instance of both LF3 and LF2aa are same. Only condition that differs is the compatibility rule doesn’t apply for LF3 whereas it does apply for LF2aa. For CFds <1w0:0/1/-> -> CFds <1w0:1/0/-> is linked fault if LF3 is considered but not if LF2aa is considered.

Two operation dynamic single cell faults (DF1):

A two-operation single-cell dynamic fault is sensitized by applying two-operations sequentially to a single cell. Dynamic single-cell faults are the following:

- Dynamic transition faults
- Dynamic write destructive faults
- Dynamic read destructive faults
- Dynamic incorrect read faults
- Dynamic deceptive read destructive faults

Two-operation dynamic coupling faults (DF2):

A two-operation dynamic coupling fault is sensitized by applying two operations sequentially to the aggressor cell or to the victim cell. Two-operation dynamic coupling faults are the following:

- Dynamic disturb coupling faults
- Dynamic transition coupling faults
- Dynamic write destructive coupling faults
- Dynamic read destructive coupling faults
- Dynamic incorrect read coupling faults
- Dynamic deceptive read destructive coupling faults

If we consider only 2 operations sensitizing in both the above mentioned dynamic faults, then the sensitizing operations could be enumerated as:

**xwywz**: ex: ‘0w0w1’ operation denotes write 0 operation applied to the cell whose state is 0; the write operation is immediately followed by another write 1 operation which sensitizes the fault in the aggressor or the victim cell.
**xrxr**: ex: `1r1r1r1` operation denotes read 1 operation applied to the cell whose state is 1; the read operation is immediately followed by another read 1 operation which sensitizes the fault in the aggressor or victim cell.

**xrxwy**: ex: `1r1w0` operation denotes read 1 operation applied to the cell whose state is 1; the read operation is immediately followed by a write 0 operation which sensitizes the fault in the aggressor or the victim cell.

**xwyry**: ex: `0w1r1` operation denotes write 1 operation applied to the cell whose state is 0; the write operation is immediately followed by a read operation which sensitizes the fault in the aggressor or the victim cell.

Apart from the sensitizing operations, dynamic faults are similar to those of single/two-cell faults.

Almost all the faults models that could be present inside the memory have been discussed above. Knowledge of these fault models will help in establishing skills to develop march-algorithms to target specific set of faults.

**Write Mask Faults (WMF):**

Faults that occur in the write mask logic are called write mask faults. Write mask faults lead to incorrect masking of the data word that is written.

**Inter-port Faults (IPF):**

Faults that occur between bit-lines and word-lines of the same/different ports are under inter-port faults. Inter-ports can be enumerated as:

- Shorts between bit-lines
- Shorts between word-lines
- Shorts between a bit-line and a word-line

**REFERENCES:**