Complete Tests, Defects, Fault Models

Complete Tests

Try all possible input combinations

(Circuits must be small)

4 inputs

$2^4 = 16$ tests

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	0	0	1	1
10	1	1	1	0

64 inputs

$2^{64} = 10^{19}$ tests

10^{11} sec @ 100MHz

32.15 centuries

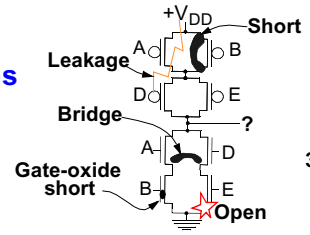
combinational only each FF doubles tests

Defect Based Tests

Look at what happens

Design test to find

(Still too many tests)



8 open
8 short
3 bridge
8 gate-oxide shorts

$3N_Q$ open/short/gate-oxide tests

N_Q transistors

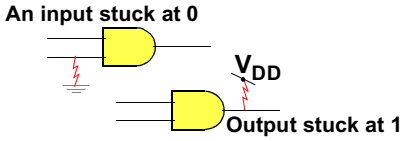
Hard to characterize

Fault Models

Assume most defects act like simple problems

Common model

(What is used)

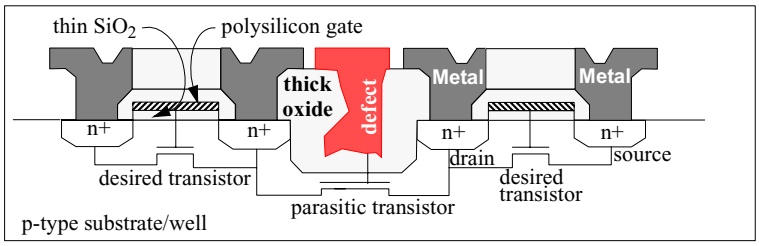


Physical Defects

CMOS Failure Mechanisms

Types of Failures

- Gate-Oxide Shorts
- Shorts
- Opens
- Punch-through: When the source-drain voltage gets too high for the channel length. The high field accelerates electrons so fast they ionize atoms in the channel by impact. This generates more high speed electrons which trigger more ionization. The effect sustains itself and the gate loses control.
- Parasitic transistor



The figure shows two desired transistors constructed between two n+ regions underneath the polysilicon gate. The channel is along the surface of the silicon under the thin SiO2, under the polysilicon gate. The transistors are shown schematically under the drawing showing their construction.

A defect in the thick oxide allows metal to reach almost through the thick oxide. Then part of the thick oxide becomes thin oxide, and may make a weak undesired transistor.

Physical Defects

CMOS Failure Mechanisms

Types of Failures

Dominant

1. Gate-oxide shorts
2. Shorts
3. Opens

Others

Incorrect $V_{THRESHOLD}$
Parasitic transistor leaks
Punch-through

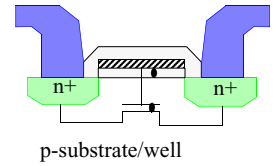
1. Gate-oxide shorts

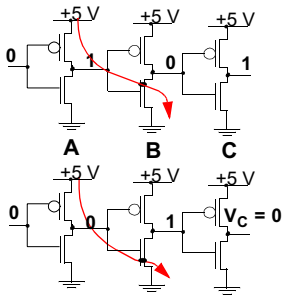
Weak Shorts

- High Resistance
- Close to drain

Strong Shorts

- Low resistance
- Close to source





Weak short
I_{DD} increases
Does not cause a logic fault

Strong short
I_{DD} greatly increases
V_C either slow or stays at 0.
Gate A output stuck at 0.
(Wrong to say gate B input stuck at 0)

CMOS Failure Mechanisms (Cont.)

2. Shorts

Lines Shorted

Transistors Shorted

- Large I_{DD}
- V_{OUT} depends on R of short

Bridge Fault

- Large I_{DD}
- V_{OUT} depends on R of short

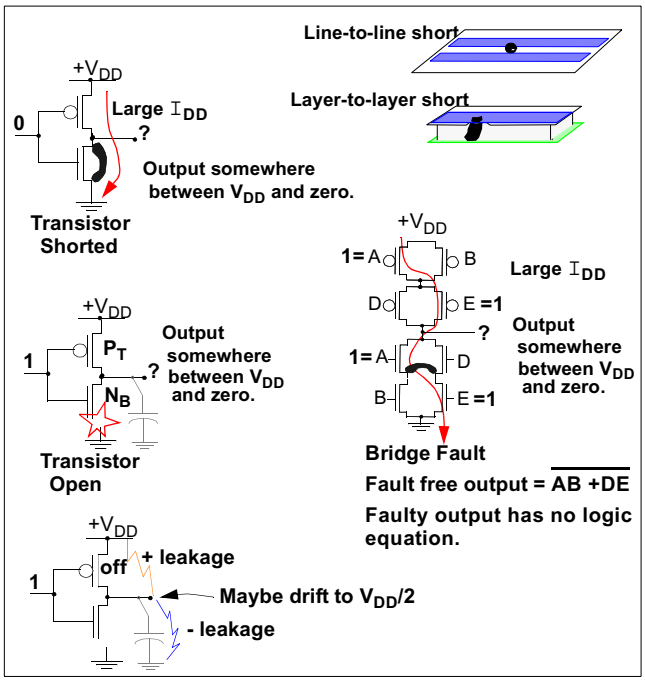
3. Opens

With N_B open, and input=1 Output floats at last value.

Combinational circuit now has storage due to fault.

Eventually C will float to some voltage, perhaps $V_{DD}/2$

That would cause high I_{DD} on the next stage.



Some Fault Models

Some bridge faults may oscillate

Fault Models of Defects

- ***Defects*** **are physical things like gate-oxide shorts.**
- ***Fault models*** **try to model the effects of defects as simple circuit changes.**
- ***Fault models*** **make it easier to see how to test for the defect.**

Some Fault Models

"Stuck At" Models

- **Assumes defects can be modelled as a gate input or output shorted to ground or V_{DD}.**
- **Very widely used.**
  Often the only model used.

Bridge Fault Models

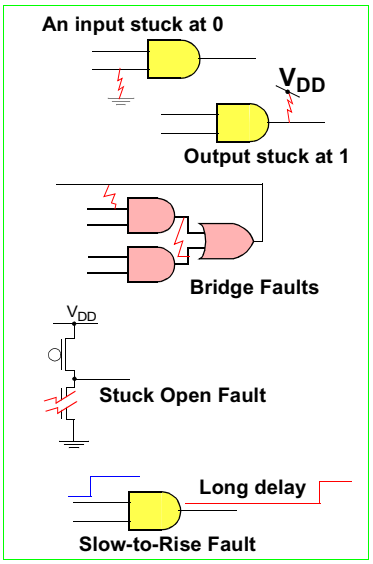
- **Assumes defect can be modelled as a wire between gate inputs/outputs.**
- **Hard to generate tests.**

Stuck-Open Faults

- **Gate can only pull one way.**
- **Usually found by I_{DDQ} tests**

Slow-To-Rise/Fall Faults

- **Slow gates**
- **Hard to test.**



Testing Using Single Stuck-At Fault models

One can formally prove that every K input black box needs no more than K+1 tests for single stuck at faults on the input leads, or ouput lead. However one needs more tests if one looks at gates inside the box.

Numb of tests = $\Theta(N)$ means the number of tests increases linearly with N.

This is important for large circuits. For example, if the number of tests increased as $\Theta(N^2)$ the number of tests for a million gate circuit would increase a million times. It would be not be possible to do such tests!

With an odd number of inputs 2 tests will find any single stuck at fault in an XOR. With an even number of inputs one extra test is needed to check the output for S@0.

Testing Using Single Stuck-At Fault Models

Properties

- Every K-input/one-output gate has 2(K+1) stuck at faults.
  K+1 S@1 faults
  K+1 S@0 faults.

- K+1 tests, or less, can find all 2K+2 faults

  K+1 tests for AND, NAND, OR, NOR.

  Only 2 tests for XOR (K Odd)
  111...1 and 000...0

- Number of tests needed increases linearly with the number of gates.

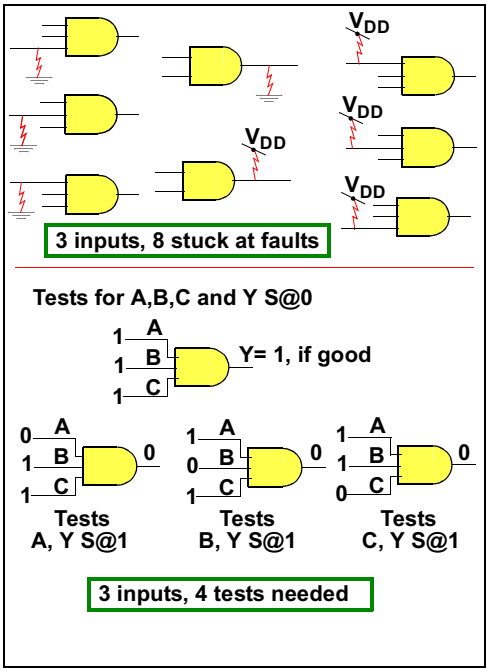
  Numb of tests = $\mathcal{O}$(N)

  N = number of gates,

  $\hat{K}$ = average inputs per gate

  need N($\hat{K}$+1) tests or less.

  In practice far fewer are needed.



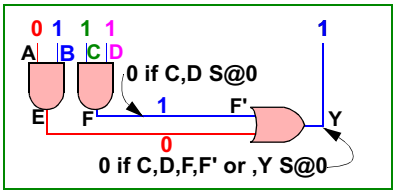
Philosophy of Stuck at Testing

- Test only for single stuck at faults
- Pretend all defects show up as single stuck at faults
  (even if they don't)
- Keep the test vector set small

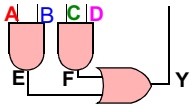
Example

4 vectors find all 14 single s@ faults

How the last vector works.



Test Vectors A B C D	Y	What they test for
1 1 1 0	1	A,B,E,Y S@0
0 1 1 0	0	A,D,E,F,Y S@1
1 0 0 1	0	B,C,E,F,Y S@1
0 1 1 1	1	C,D,F,Y S@0



4 test vectors find all single s@ faults

There are N•2(K+1) =18 s@ faults.

E and F are both gate inputs and outputs.
Since fanout is one, count each only once

gives 14 different s@ faults

K=2....avg inputs/gate
N=3...number of gates

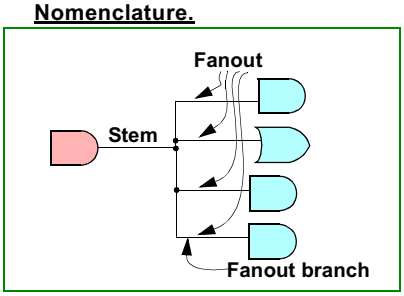
How Stuck @ Tests are Used

Each lead has two faults. Because F and F' are the same lead, removing F' removes two of the stuck-at faults.

Fanout and Stuck at Tests

- Test only for single stuck at faults

Nomenclature.

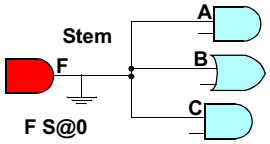


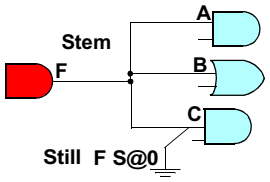
A stuck-at fault on any lead is taken as an output fault.

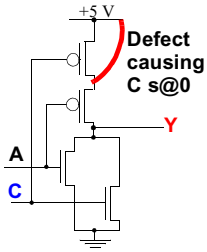
An input fault acts only on a single branch, hence it acts like the branch lead was open

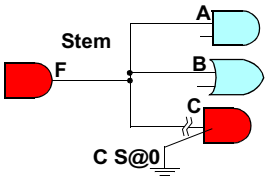
The internal fault shown is:

C S@ 0









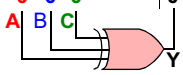
Fanout and Stuck@ Models

Other Fault Models

Multiple Stuck-at Faults

- One consider several faults at once.
- A second fault can mask the tests for single faults, so the gate tests good.
- Such conditions are rare.
- Test vector generation for multiple faults is very difficult, $\mathcal{O}(N^2)$ for double faults. It is not usually done.

Test Vectors A B C	Y	What they test for	
1 1 1	1	A,B,C,Y	S@0
0 0 0	0	A,B,C,Y	S@1



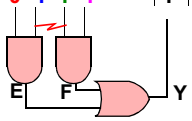
2 test vectors find all single s@ faults

But if both A S@1 and B S@1, the tests will say the gate is good.

Bridge Faults

- Modelled as a short between leads.
- Often not be found by stuck-at tests.
- In the figure, a complete set of S@ test vectors does not find the bridge fault.
- Test vector generation for bridge faults is very hard, and is not done.
- Bridge faults are found by:
  a) luck,
  b) I_{DDQ} tests.

Test Vectors A B C D	Y	What they test for	
1 1 1 0	1	A,B,E,Y	S@0
0 1 1 0	0	A,D,E,F,Y	S@1
1 0 0 1	0	B,C,E,F,Y	S@1
0 1 1 1	1	C,D,F,Y	S@0



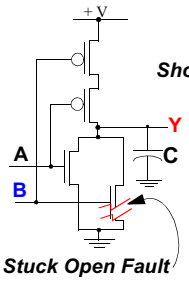
Multiple Stuck@ and Bridge Faults

Because it is so time consuming, tests for even double faults are not specifically done. However masking examples like the XOR are very rare. Almost any third test vectors would expose the double XOR fault.

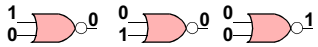
Fault Models (cont.)

CMOS Stuck Open Faults.

- B cannot pull Y ↓
  but cap C will hold Y down
  if A pulls it down first
- Apply tests in right order:
  1. Have A pulls Y ↑
  2. Then check if B can pull ↓
- In the wrong order
  1. Have A pull Y ↓
  2. This discharges C
  3. Then no one knows if
     B can pull Y ↓.



Stuck Open Fault



Complete set of S@ 0,1 tests for NOR

Measured
Should be

Tests applied in wrong order

A	B	Y	explanation
1	0	0 0	A pulls Y ↓
0	1	0 0	B can't pull ↓, but Y floats.
0	0	1 1	A&B pull Y ↑

Tests applied in correct order

A	B	Y	explanation
0	0	1 1	A&B pull Y ↑
0	1	0 ~~1~~	B can't pull ↓, but Y floats.
1	0	0 0	A pulls Y ↓

Automatic Test Generators

- Are good at S@ faults.
- Are poor at getting the order right.
- Stuck open faults in production testing are mostly found by:
  1. luck
  2. I_{DDQ} testing

Another Viewpoint

By introducing storage on a capacitance,
S@ 𝒪 faults make combinational circuits into sequential ones (ones with storage).

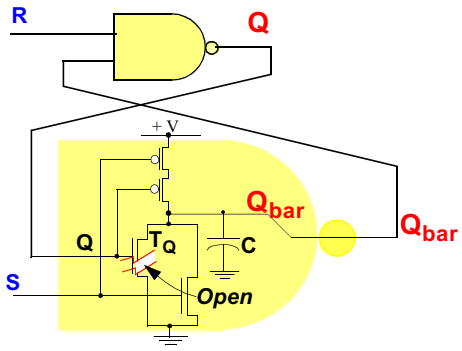
CMOS Stuck-Open Faults

CMOS Stuck Open Faults (cont)

Untestable Faults

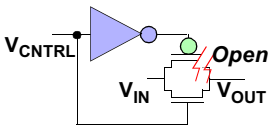
R-S Latch

- S can pull Q_{bar} ↓
  T_Q cannot hold Q_{bar} down
- Cap C will hold Q_{bar} down,
  for a while.
  Eventually charge will leak off C. ↓
- Latch has become a dynamic latch.
  Will work in circuit if no long waits..



Transmission Gate

- The circuit will work with an open transistor.
- Must pull up through NMOS:>
  - Slow.
  - Cannot pull V_{out} up to V_{DD}.



Untestable Stuck-Open Faults

R-S latch

The latch will work at high and medium speeds because the capacitance will hold $\overline{Q}$ low for a while. After some time, perhaps 0.1 to 10 ms., the capacitance may discharge and the latch will forget.

This fault is called untestable reasonable production testing methods, because the tester would have to check how long the latch would hold a stored value. The tester must not, while waiting, send out a test vector that would recharge the capacitance. Also it must not send out signals on nearby lines that might capacitively couple energy into the capacitance. It is very very difficult to tell how nearby lines will affect the charge.

The transmission gate

A transmission gate with an open transistor will have a high resistance when passing one polarity of signal. In this case, with the PMOS transistor open, $V_{OUT} < V_{DD} - V_{Threshold.}$

Also V_{OUT} will rise only slowly through the PMOS transistor and may give a slow-to-rise fault. Scan tests are usually run at less than full speed, and may not find the fault.

At speed testing may find such faults, but these tests are still far less commonly done than scan testing for stuck@ faults.

Faults for At Speed Testing

Transition Delay Faults (Gate Delay Faults)

Faults are *slow to rise* and
slow to fall

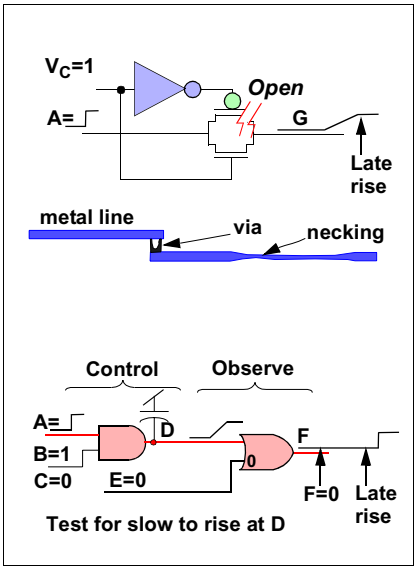
Causes might be:

- Open pass transistor.
- Thin metal line or resistive via.
- Part of gate polysilicon missing.

To test for *slow to rise* at D.

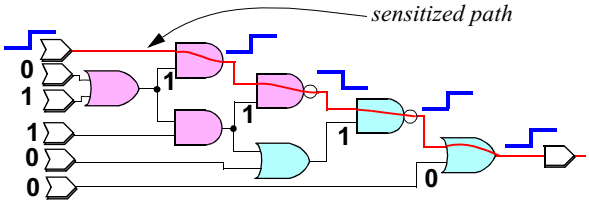
Force D to 0, same pattern as for a S@1 fault.
Wait until the output receives the good value.
Change pattern, force D to 1
Keep the path D to F sensitized.

Needs two-step test.

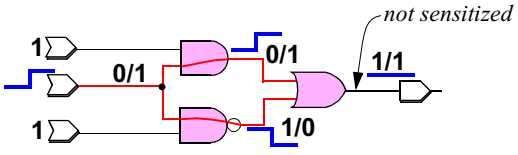


Transition Delay Faults (Gate Delay Faults)

Sensitized Path



In a sensitized path through a circuit,
a single level change at the input to the path causes a change at the path output.
Placing different values on the other inputs can desensitize the path.



Test Generation

Finding Test Vectors

Using Single Stuck at Fault Model

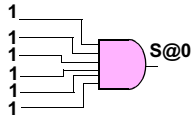
Controllability and Observability

Controllability: Ease with which which we can force a node to a value which exposes a fault.

Observability: Ease with which the faulty value can be transmitted to an output

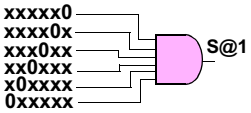
Poor Controllability

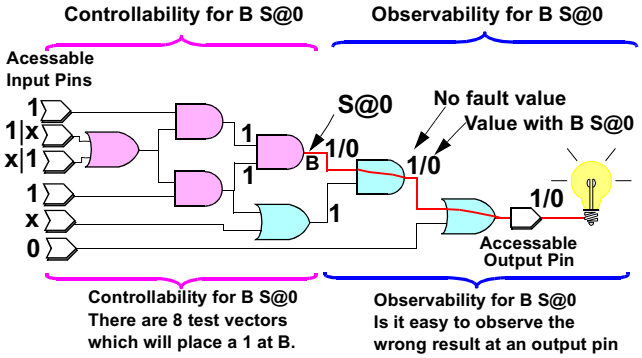
one test vector



Good Controllability

many test vectors

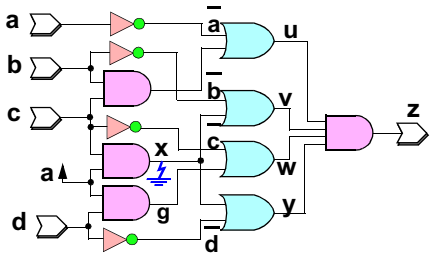




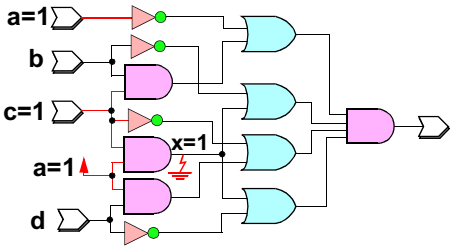
There are 4 test vectors of the form 11x1x0
and 4 of the form 1x11x0

Finding Test Vectors

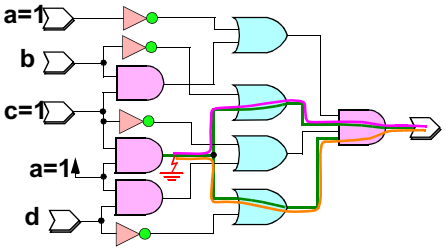
Find a test for X S@0



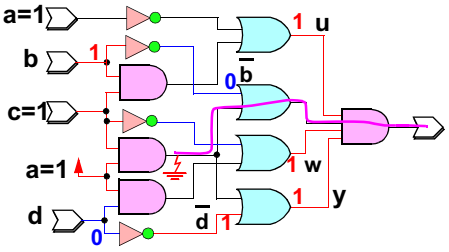
Control point X, force it to 1.



Three possible paths to observe fault

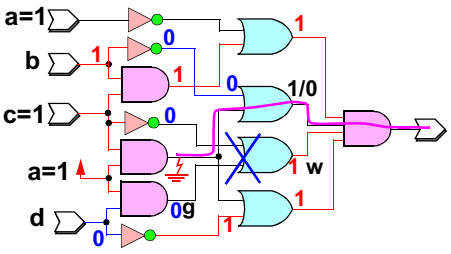


Try the upper path. To sensitize path must have $\bar{b}$=0, u, w, y=0, ⇒ d=1.

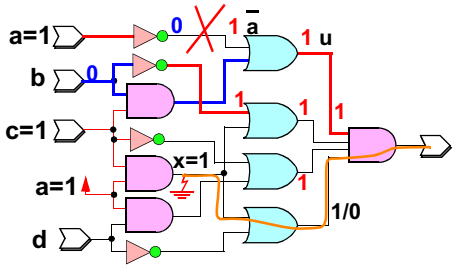


Finding Test Vectors

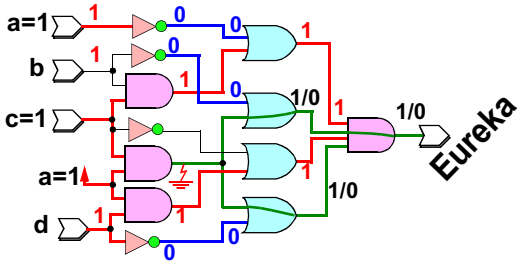
Continue trying to sensitize the upper path. g=0, c=0 ⇒ W≠1



Try sensitizing the lower path.
$\bar{a}$=0 ⇒ u≠1



Try sensitizing the reconvergent path make sure the polarities are not opposite where they reconverge.



Moral
It is a lot of work to find test vectors even for a Pentium.

Tests for At Speed testing

Tests for At Speed Testing

Transition Delay Faults (Gate Delay Faults)

Faults are *slow to rise* and
slow to fall

To test for *slow to rise* at D.

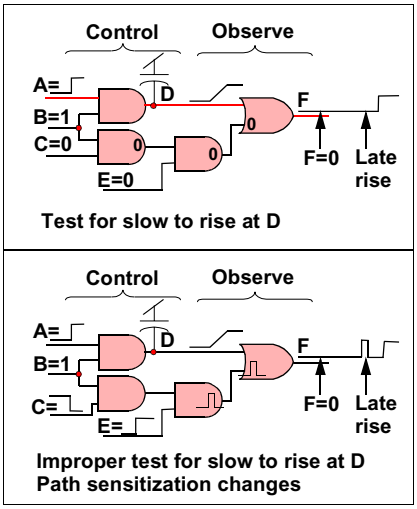
Force D to 0, same pattern as for a S@1 fault.
Wait until the output receives the good value.
Change pattern, force D to 1
Keep the path D to F sensitized.

Do not cause glitches in the path D to F
by changing the way it is sensitized.

Delay fault testing always finds s@ faults.
These effectively have infinite delay.

ATPG tests are observed on the closest output.

Path delay tests want the longest path.
The one most likely to fail by say slightly thin metal.



In *transition delay faults* need to sensitise a path from the input to the output.
ATPG for this is almost the same as for a stuck at fault test.