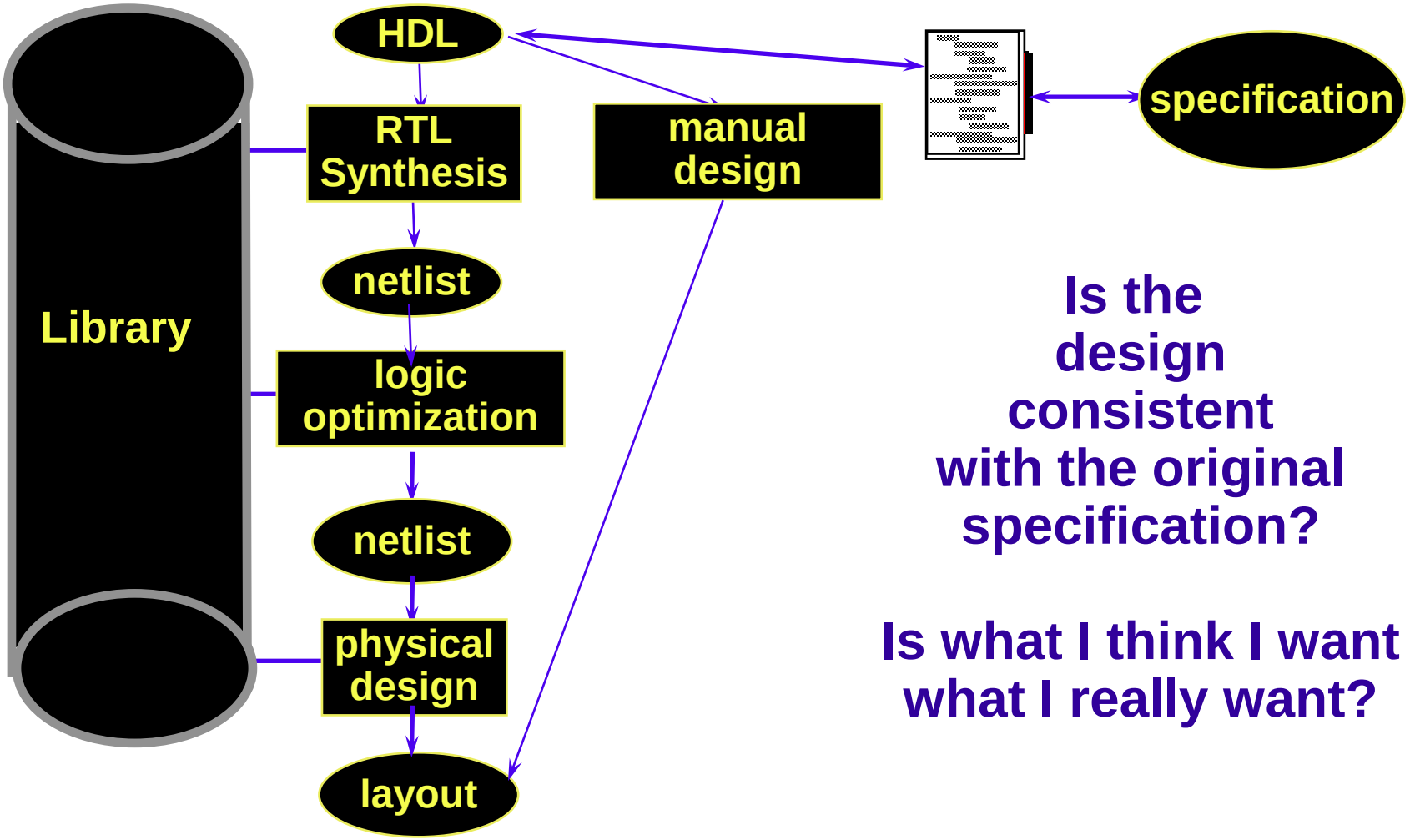
Manufacture Testing of Digital Circuits

Alexander Gnusin

TUG

Design Verification

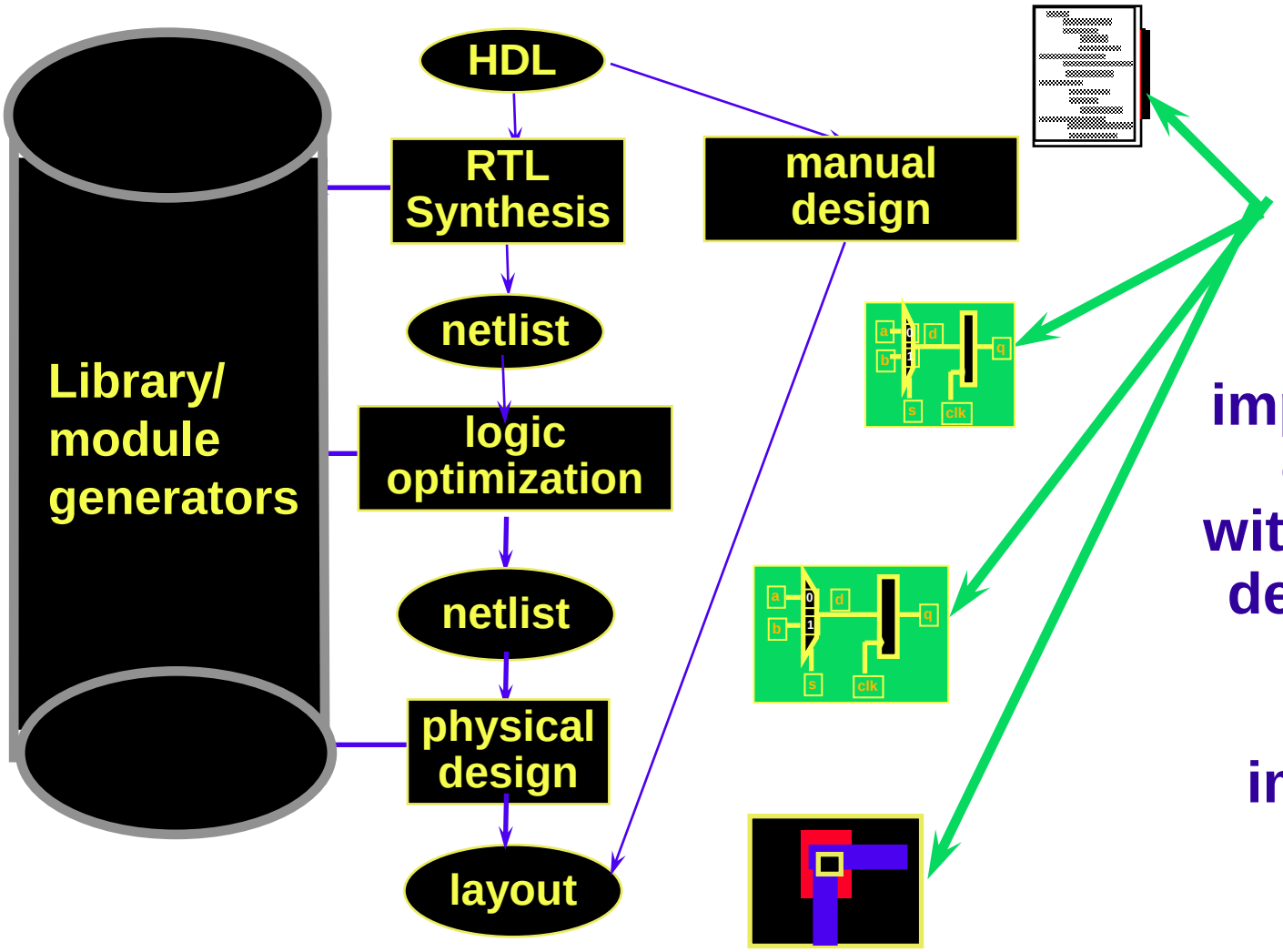


Is the design consistent with the original specification?

Is what I think I want what I really want?

TUG

Implementation Verification

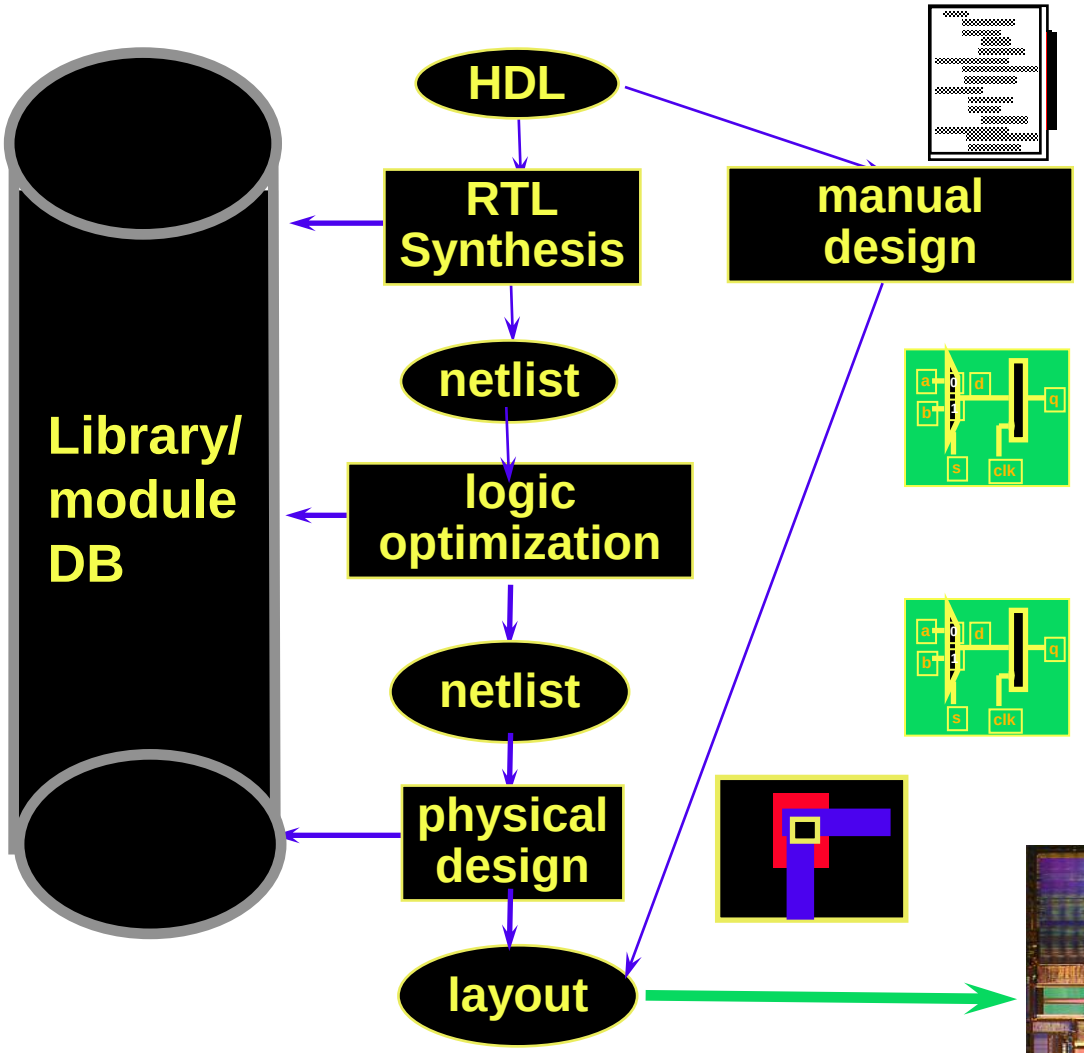


Is the implementation consistent with the original design intent?

Is what I implemented what I wanted?

TUG

Manufacture Verification (Test)



Is the manufactured circuit consistent with the implemented design?

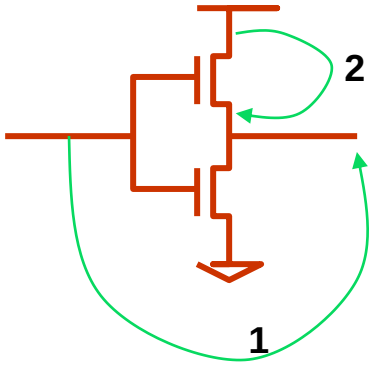
Did they build what I wanted?

TUG

Defects and Fault models

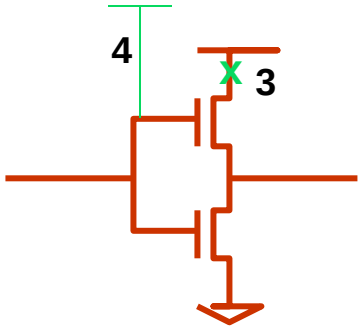
Types of Manufacturing defects:

- Bridging (1)
- Shorts (2)
- Opens (3)
- Transistors stuck-open (4)



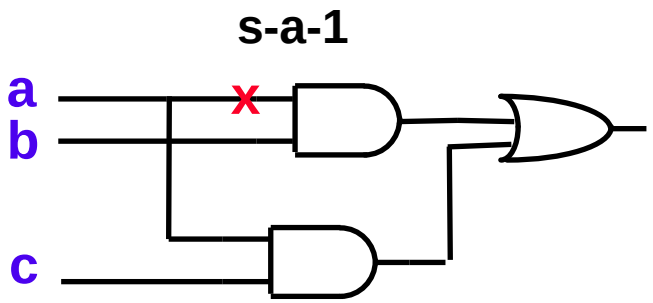
These need to be reduced to models:

- Single stuck-at-1, stuck-at-0
- Multiple stuck-at-1, stuck-at-0
- Delay fault models:
  - Gate
  - Path

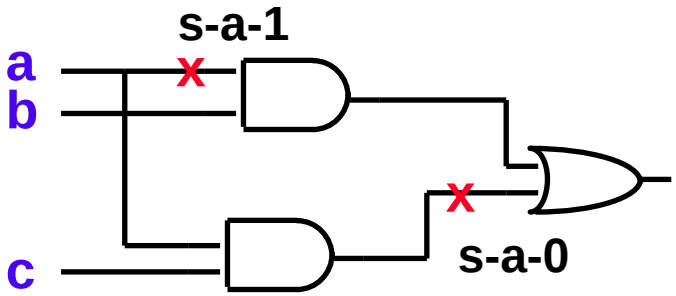


TUG

Fault Models



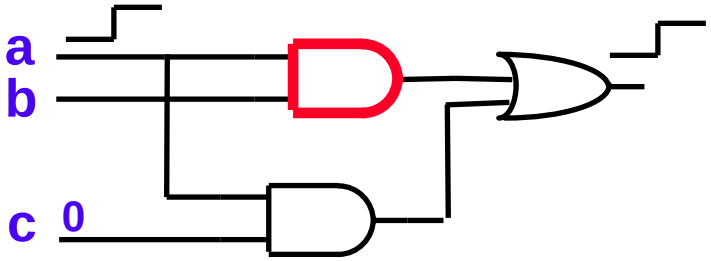
Single stuck-at fault:
In the faulty circuit, a single line/wire is S-a-0 or S-a-1



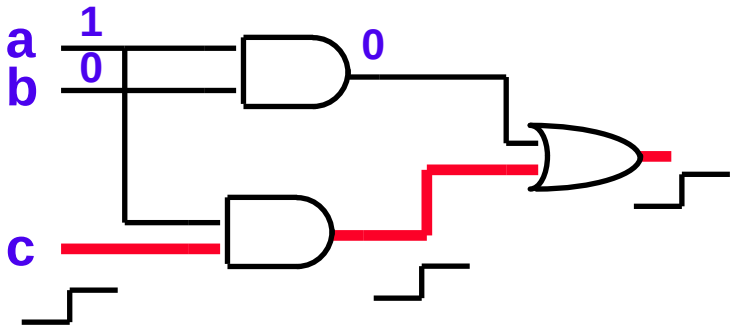
Multiple stuck-at faults:
In the faulty circuit any subset of wires are S-a-0 or S-a-1 (in any combination)

TUG

Fault Models - 2



Gate delay fault



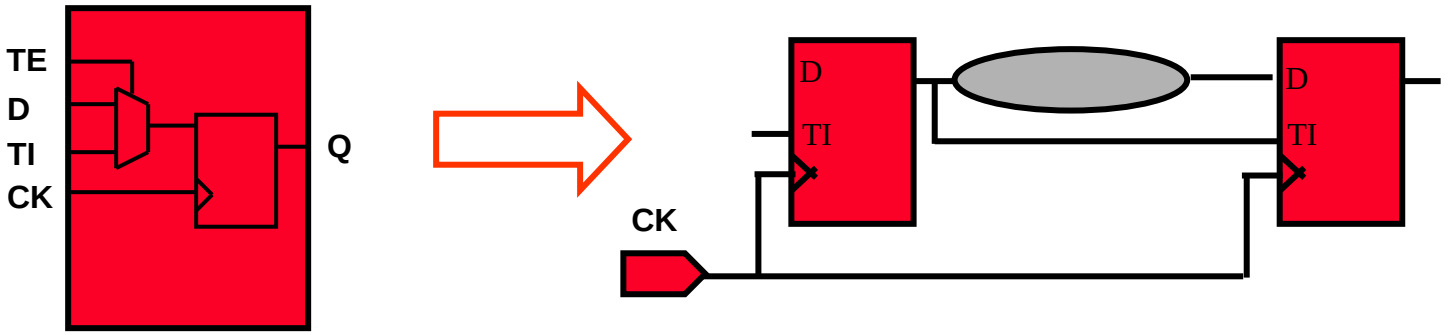
Path delay fault

Sequential to Combinational Logic reduction

Internal scan chains: add additional state to flip-flops (15 - 20% area overhead)

TE = 0 : Functional mode, no influence
TE = 1 : Scan mode, any data can be loaded/extracted

Data Input of FF acts as additional Primary Output
The Output of FF acts as additional Primary Input



Test Generation

Choose a fault model, e.g., single stuck-at fault model

Given a combinational circuit which realizes the function $f(x_1, x_2, \ldots x_n)$, a logical fault changes it to $f_{\propto}(x_1, x_2, \ldots x_n)$

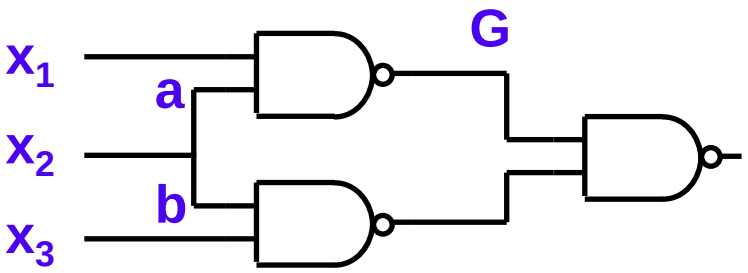
Inputs detecting $\propto$ are $f \oplus f_{\propto}$ (= 1)

Interested in <u>one</u> vector

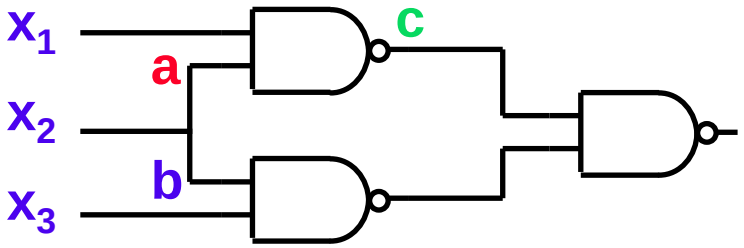
$A = (a_1, a_2, \ldots, a_n) \in f \oplus f_{\propto}$

Single Stuck-At Faults

A fault is assumed to occur only on a single line.



$Z = x_1 x_2 + x_2 x_3$



a s-a-1 $Z = x_1 + x_2 x_3$

c s-a-1 $Z = x_2 x_3$

Definitions

Controlability of internal point x: exist input vectors V0{x} and V1{x} that set x to 0 and 1

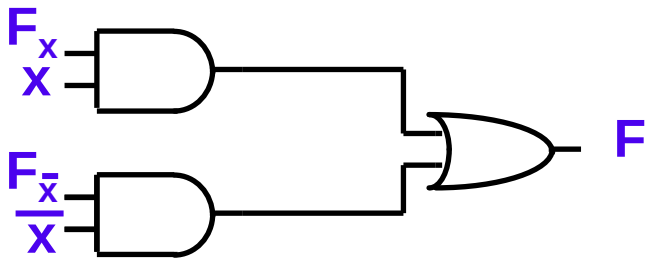
Observability of internal point x: exist input vector V_{obs}{x} that drives the value of x to primary output

Cofactors of function F(x,y,z) w.r.t. variable x:

$F_x = F(1,y,z)$, $F_{\bar{x}} = F(0,y,z)$

Shannon Expansion of function F(x,y,z):

$$F(x,y,z) = x * F_x + \bar{x} * F_{\bar{x}}$$



Boolean Difference

Circuit C has function $f(x_1, x_2, \ldots x_n)$

Input x_i s-a-0 (fault $\propto$)

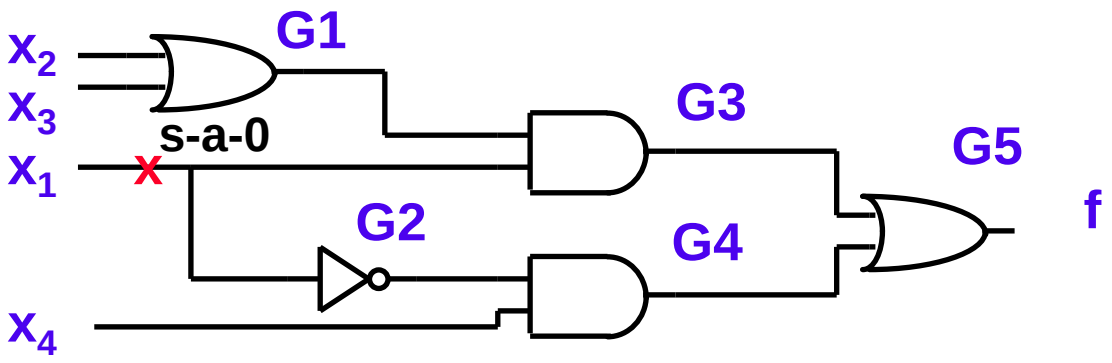
$f_{\propto} = f_{\overline{xi}}$ (cofactor of f w.r.t. $\overline{x_i}$)

Set of tests detecting $\propto$

$$T = f \oplus f_{\propto}$$
$$= (x_i f_{xi} + \overline{x_i} f_{\overline{xi}}) \oplus f_{\overline{xi}}$$
$$= x_i (f_{xi} \oplus f_{\overline{xi}})$$

$$\frac{df}{dx_i} = f_{xi} \oplus f_{\overline{xi}}$$ - Boolean difference of f w.r.t. x_i

Boolean Difference Example



Function: $f = (\overline{x_2} + x_3)x_1 + x_1 x_4$

x_1 s-a-0 - compute $x_1 \frac{df}{dx_1}$

$$\frac{df}{dx_1} = f(0, x_2, x_3, x_4) \oplus f(1, x_2, x_3, x_4)$$

$$= x_4 \oplus (x_2 + x_3)$$

$$T = x_1 \overline{x_2}\ \overline{x_3}\ \overline{x_4} + x_1 x_2 \overline{x_4} + x_1 x_3 \overline{x_4}$$

Internal Faults

Let C be a circuit for f(X) and h be an internal signal in C.

Represent h as a Boolean function h(X)

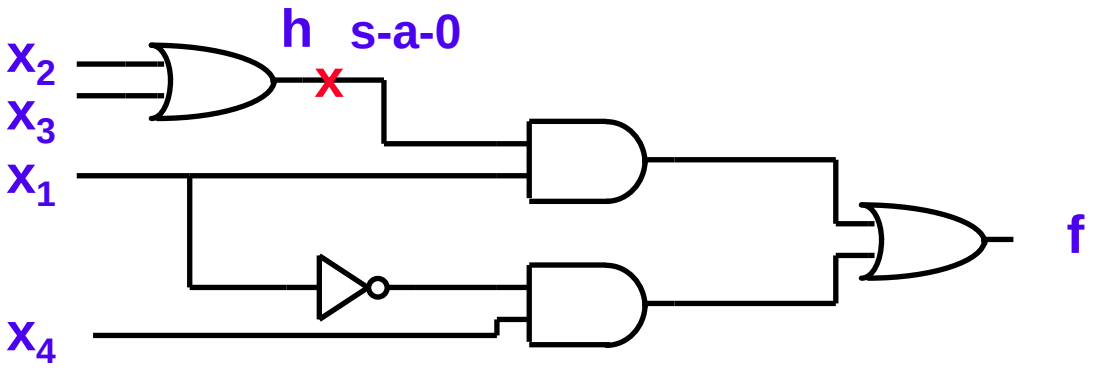
Express f as a function of inputs X and h

$$f^*(X, h) \equiv f(X)$$

Set of all tests for h s-a-0

$$h(X)^* \frac{df^*(X, h)}{dh}$$

Internal Faults Example



$$f = (x_2 + x_3)x_1 + \overline{x}_1 x_4$$

$$f^* = h\,x_1 + \overline{x}_1 x_4 \qquad h = x_2 + x_3$$

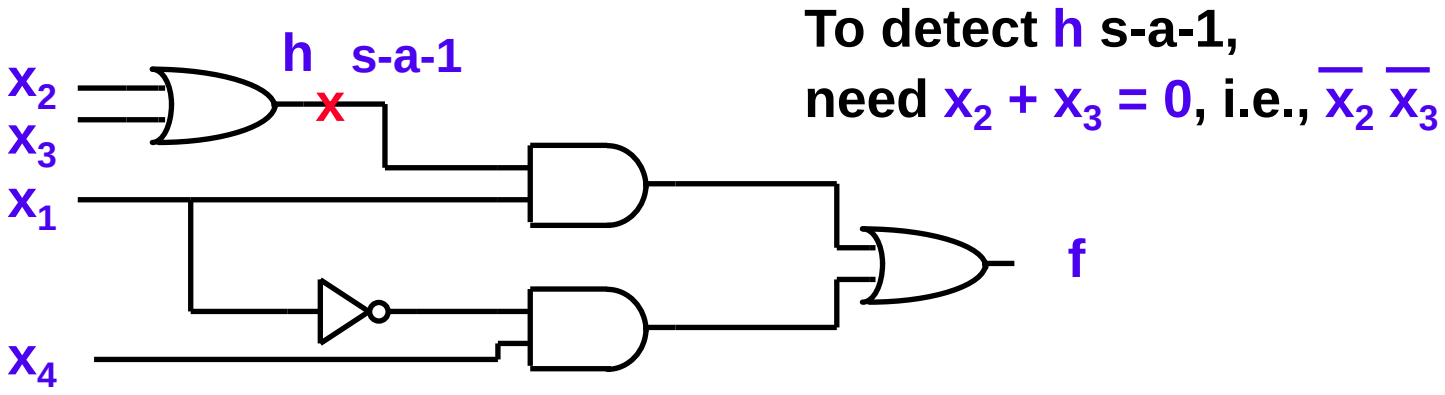
$$\frac{df^*}{dh} = f^*(x_1, x_4, 0) \oplus f^*(x_1, x_4, 1)$$

$$= \overline{x}_1 x_4 \oplus (x_1 + \overline{x}_1 x_4) = x_1$$

Result : $T = x_1 x_2 + x_1 x_3$

Path Sensitization

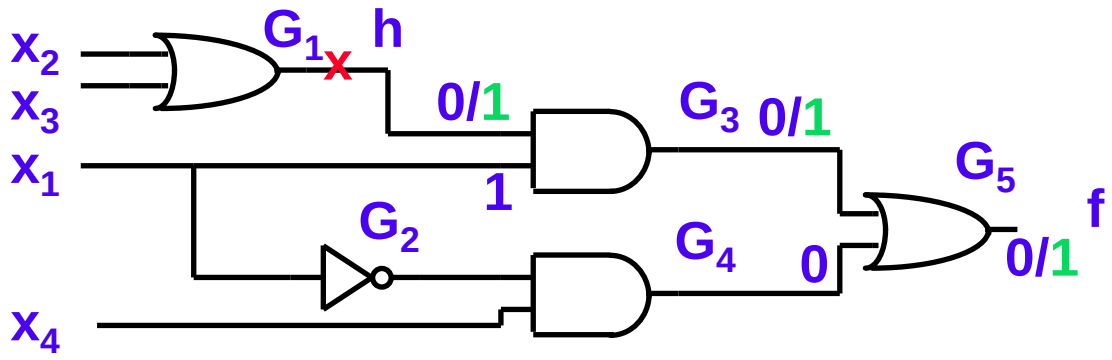
In order for an input vector X to detect a fault a s-a-j, j = 0,1 the input X must cause the signal a in the normal (fault-free) circuit to take the value $\bar{j}$.



The condition is necessary but not sufficient. Error signal must be propagated to output.

Error Propagation

The error signal must be propagated along some path from its origin to an output



h s-a-1, for h to be 0, need $x_2 = x_3 = 0$ $(\overline{x_2}\ \overline{x_3})$

Only one path G_3, G_5

In order to propagate an error through AND gate G_3, other input $x_1 = 1$. To propagate through G_5, need $G_4 = 0$, $x_1 + \overline{x_4}$

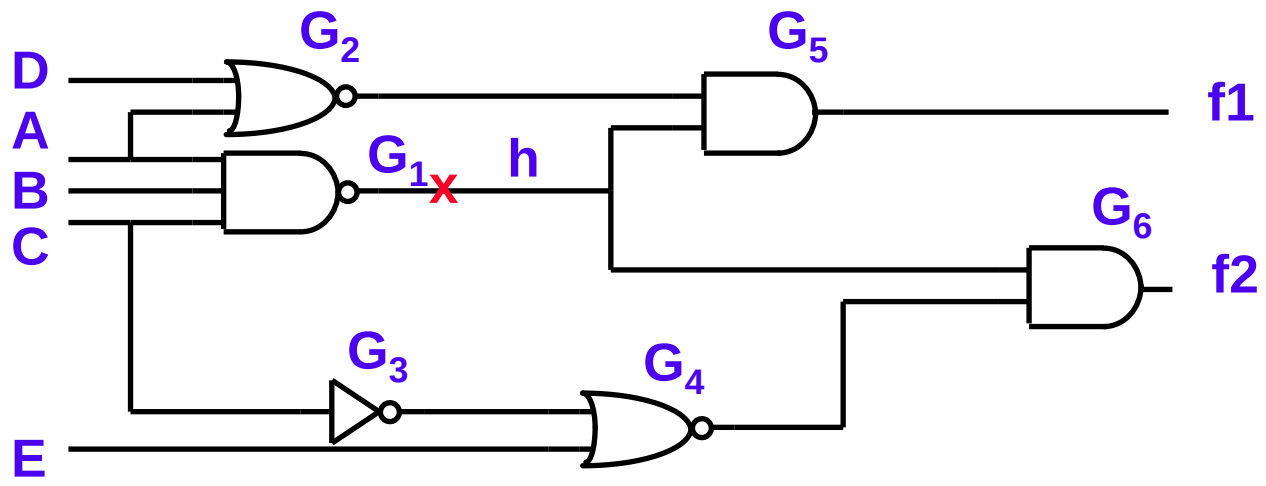
Single Path Sensitization (SPS)

1. Specify inputs so as to generate the appropriate value (0 for s-a-1, 1 for s-a-0) at the site of the fault.

2. Select a path from the site of the fault to an output and specify additional signal values to propagate the fault signal along this path to the output (Error propagation).

3. Specify input values so as to produce the signal values specified in (2) (Line justification).

Sensitization Example



h s-a-1

To generate h = 0, we need A = B = C = 1

Have a choice of propagating through G_5 or via G_6.

Propagating through G_5 requires $G_2 = 1$

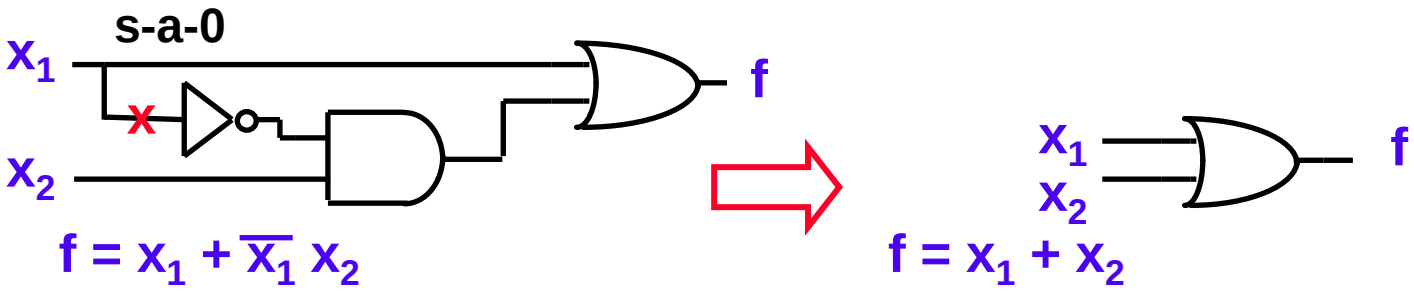
$\Rightarrow$ A = D = 0, Contradiction

Propagating through G_6 requires $G_4 = 1 \Rightarrow$ C = 1, E = 0.

A valid test vector is $ABC\bar{E}$

Redundancy

Existence of a fault does not change the functionality of a circuit ⇒ redundant fault

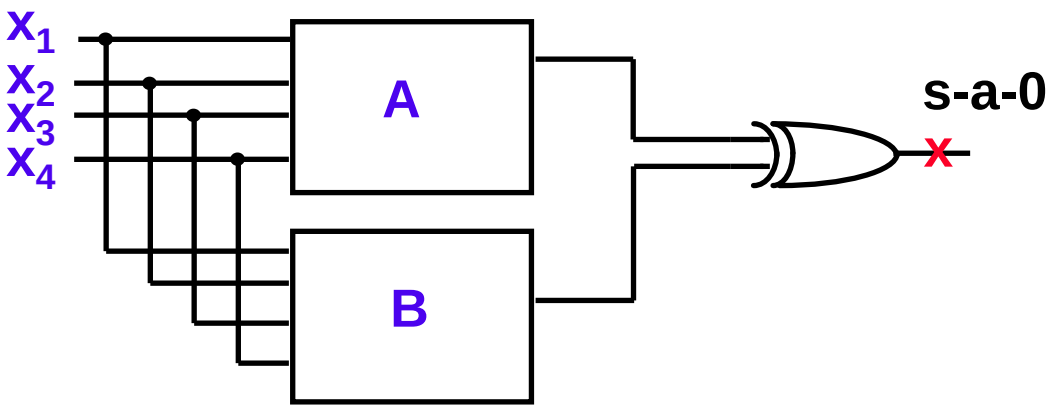


A test generation algorithm is assumed complete if it either finds a test for any fault or proves its redundancy, upon terminating.

Implementation Verification and Testing

Given two single-output circuits A and B

Are A and B equivalent can be posed as: Is there a test for F s-a-0?



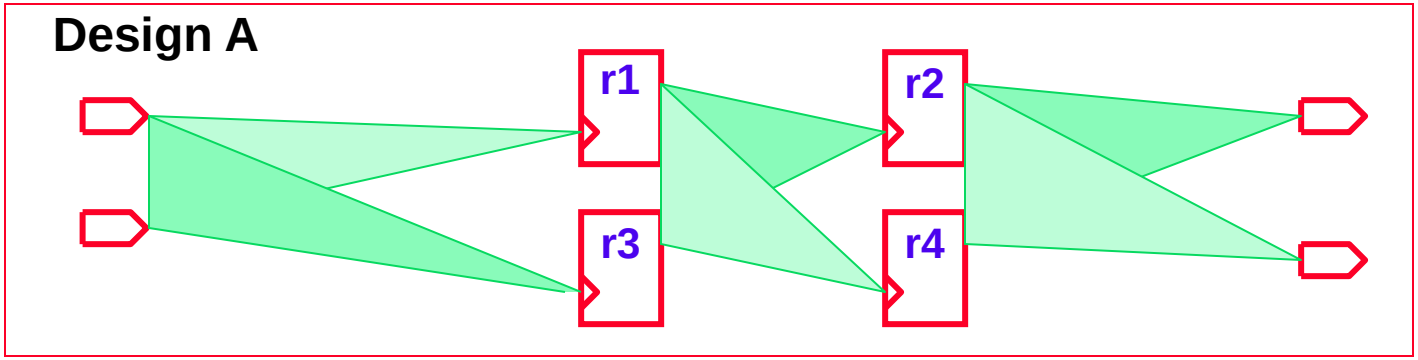
If F s-a-0 is redundant, A ≡ B else test vector produces different outputs for A and B.

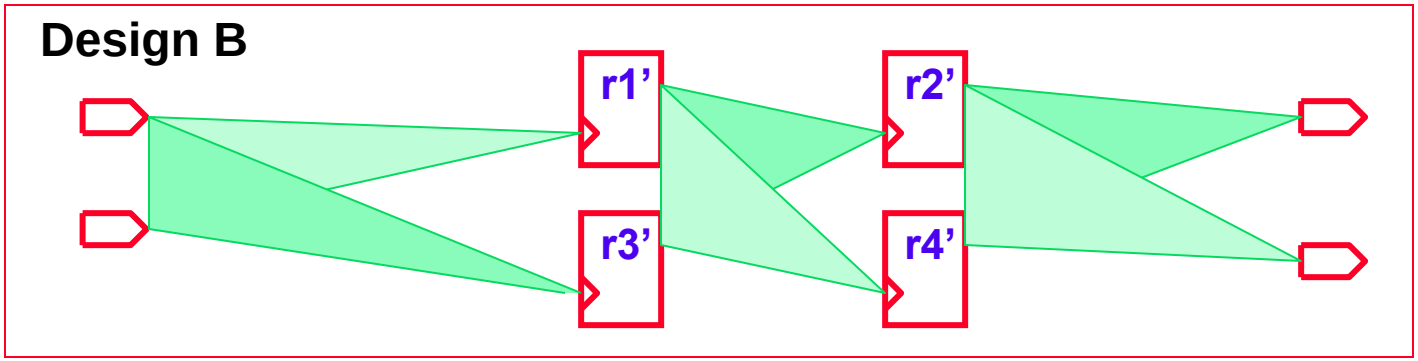
TUG

Implementation Verification (cont)

2 Stages of functional verification:

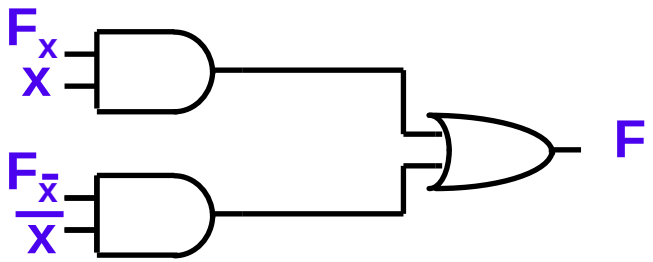
- Match Primary Input, Primary Output and Register names between two designs
- Check Functional Equivalence of each one of matching logic cones





Usage of Shannon expansion for Timing Optimization

$$F(x,y,z) = x * F_x + \overline{x} * F_{\overline{x}}$$



Problem: Long Timing Path from input A to output Y
Solution: Use Shannon expansion of Y by A: