

# Lecture 2 - Fault Modeling

- Defects, Errors, and Faults
- Some real defects in VLSI and PCB
- Why model faults?
- Common fault models
- Stuck-at faults
  - Single stuck-at faults
  - Fault equivalence
  - Fault dominance
- Other Common Faults
- Faults in FPGAs

# Defects, Faults, and Errors

- **Defect:** unintended difference between the implemented HW and its intended design
  - May or may not cause a system failure
- **Fault:** representation of a defect at the abstracted function level



# Defects, Faults, and Errors

- **Error:** Manifestation of a fault that results in incorrect circuit (system) outputs or states
  - Caused by faults
- **Failure:** Deviation of a circuit or system from its specified behavior
  - Fails to do what it should do
  - Caused by an error
- Defect --> Fault ---> Error ---> Failure

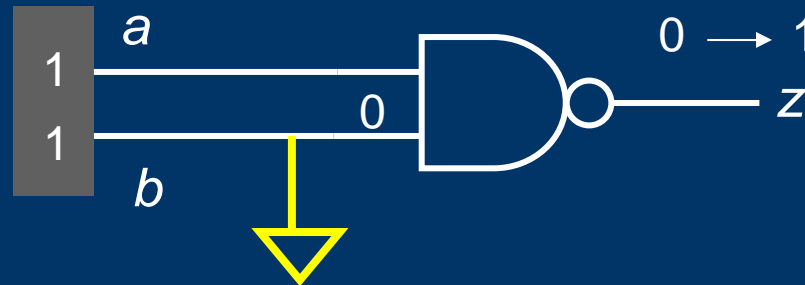
# Some Real Defects in Chips

- Processing defects
  - Missing contact windows
  - Parasitic transistors
  - Oxide breakdown
  - . . .
- Material defects
  - Bulk defects (cracks, crystal imperfections)
  - Surface impurities (ion migration)
  - . . .
- Time-dependent defects
  - Dielectric breakdown
  - Electromigration
  - . . .
- Packaging defects
  - Contact degradation
  - Seal leaks. . .

Ref.: M. J. Howes and D. V. Morgan, *Reliability and Degradation - Semiconductor Devices and Circuits*, Wiley, 1981.

# Example

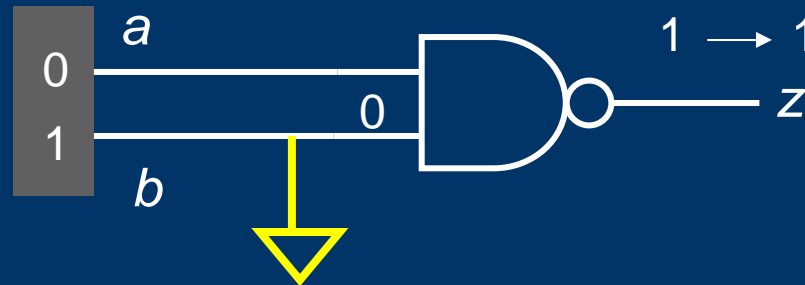
- Processing defect: a short of b to ground
- Fault: signal b stuck-at logic 0



- Error: *z* has the wrong value if  $a = b = 1$

# Example

- Defect: a short to ground
- Fault: signal b stuck at logic 0



- Error: *z* has the wrong value if  $a = b = 1$
- But, if  $a = 0$ , fault exists, but no error!

# Why Model Faults?

- Real defects too numerous and often not analyzable
- A fault model identifies targets for testing
  - Model faults most likely to occur
- Fault model limits the scope of test generation
  - Create tests only for the modeled faults
- A fault model makes analysis possible
  - Associate specific defects with specific test patterns
- Effectiveness measurable by experiments
  - Fault coverage can be computed for specific test patterns to reflect its effectiveness

# Common Fault Models

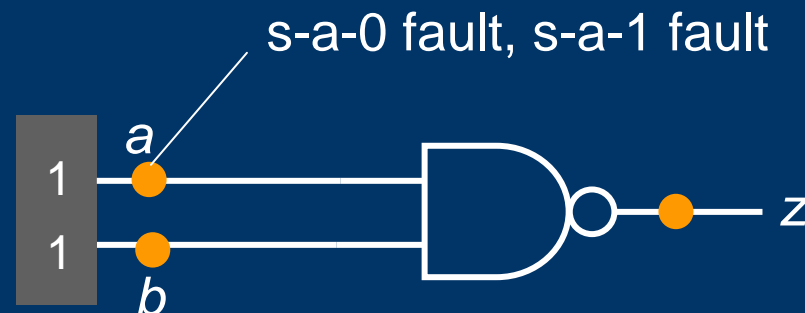
- Single stuck-at faults
- Transistor open and short faults
- Memory faults
- Functional faults (processors)
- Delay faults (transition, path)
- Analog faults (parametric deviations)

# Single Stuck-at Fault

- Three properties define a single stuck-at fault
  - Only one line is faulty
  - The faulty line is permanently set to 0 or 1
  - The fault can be at an input or output of a gate

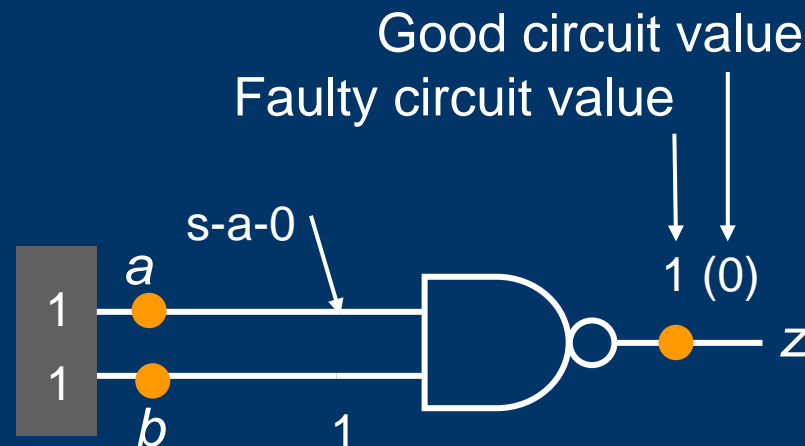
# Single Stuck-at Fault

- Three properties define a single stuck-at fault
  - Only one line is faulty
  - The faulty line is permanently set to 0 or 1
  - The fault can be at an input or output of a gate
- Example: NAND gate has 3 fault sites (●) and 6 single stuck-at faults



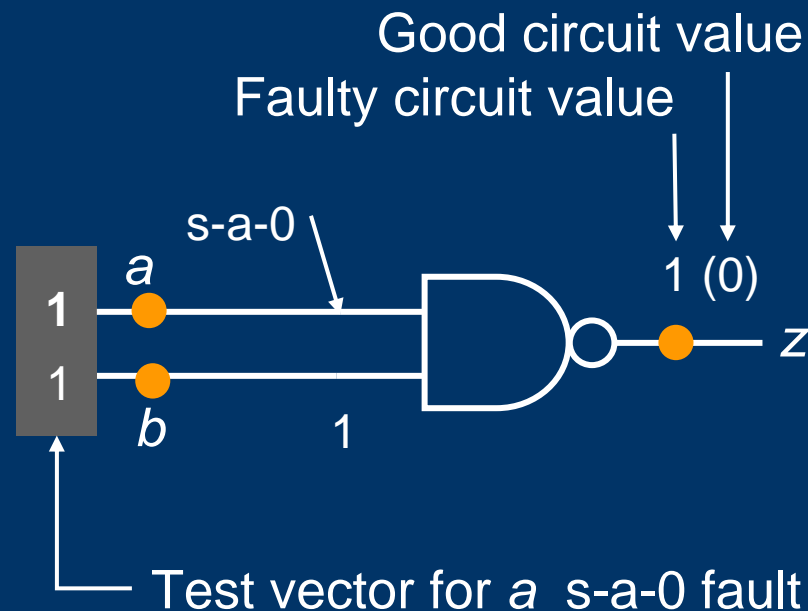
# Single Stuck-at Fault

- Three properties define a single stuck-at fault
  - Only one line is faulty
  - The faulty line is permanently set to 0 or 1
  - The fault can be at an input or output of a gate
- Example: NAND gate has 3 fault sites (●) and 6 single stuck-at faults



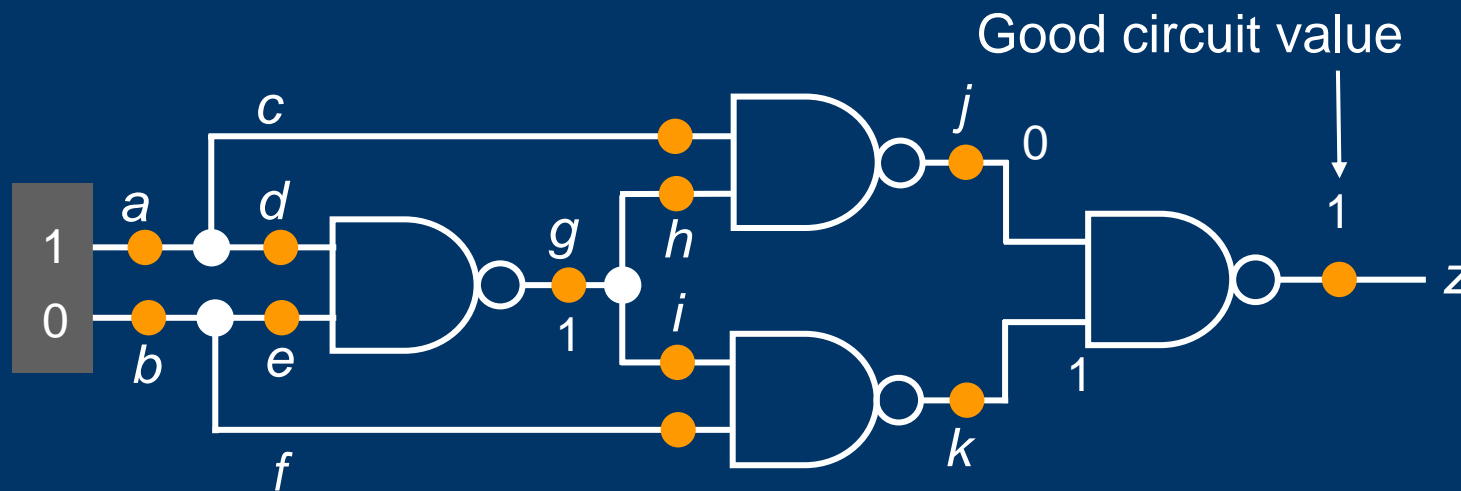
# Single Stuck-at Fault

- Three properties define a single stuck-at fault
  - Only one line is faulty
  - The faulty line is permanently set to 0 or 1
  - The fault can be at an input or output of a gate
- Example: NAND gate has 3 fault sites (●) and 6 single stuck-at faults



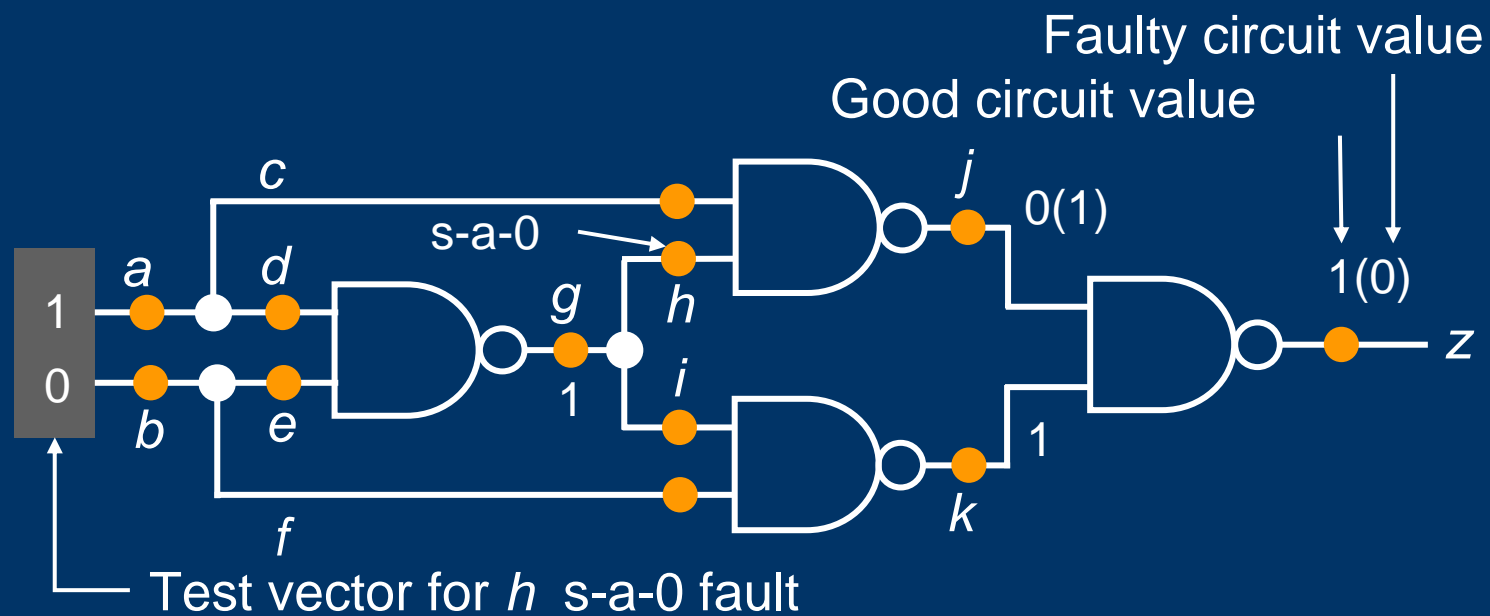
# Single Stuck-at Fault

Example: XOR circuit has 12 fault sites and 24 single stuck-at faults



# Single Stuck-at Fault

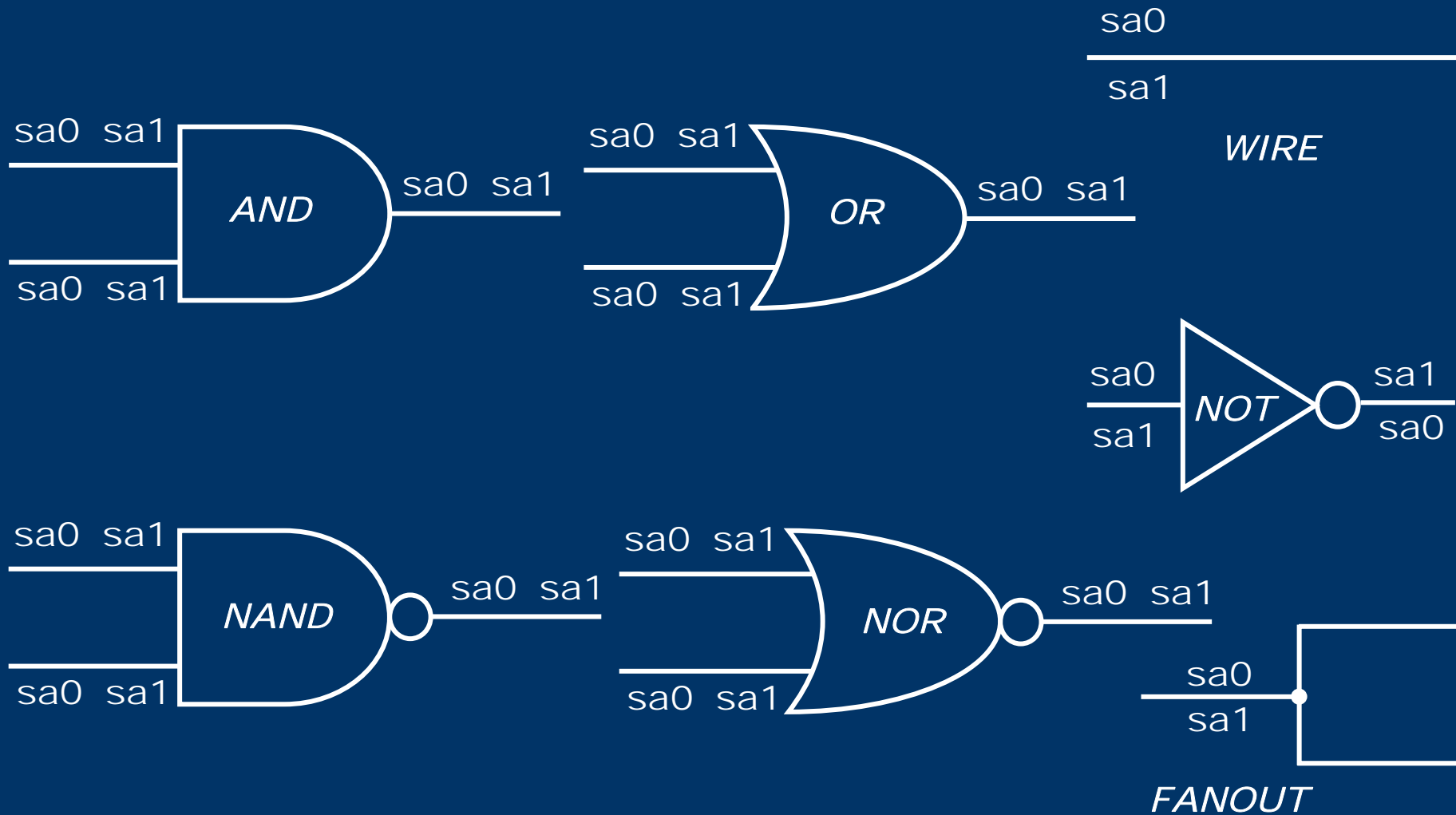
Example: XOR circuit has 12 fault sites and 24 single stuck-at faults



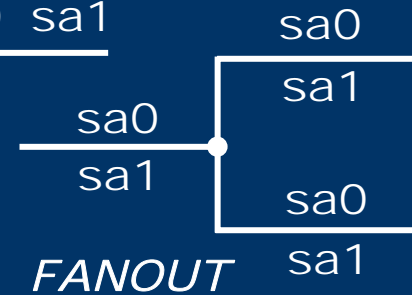
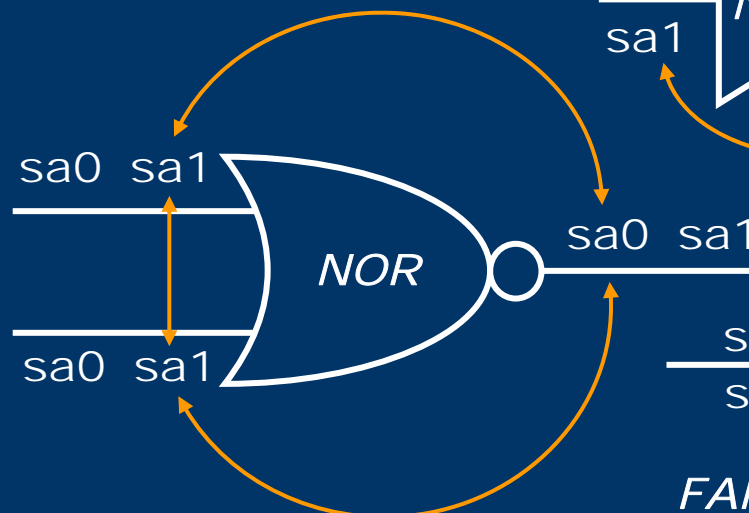
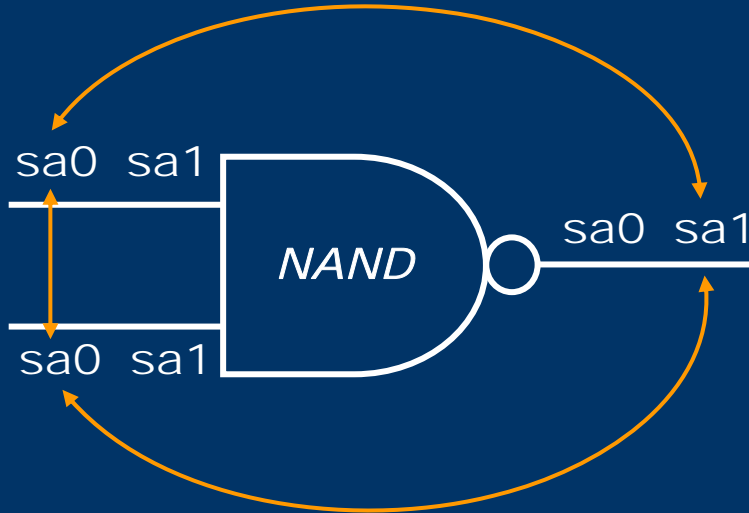
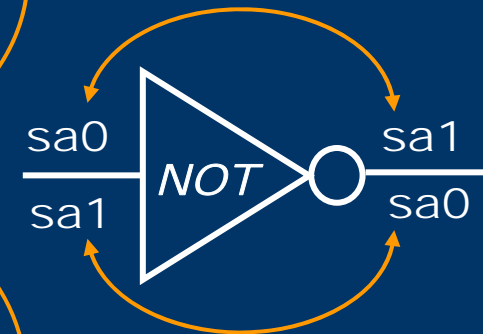
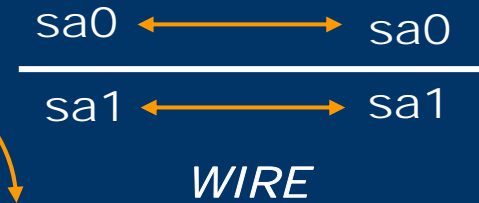
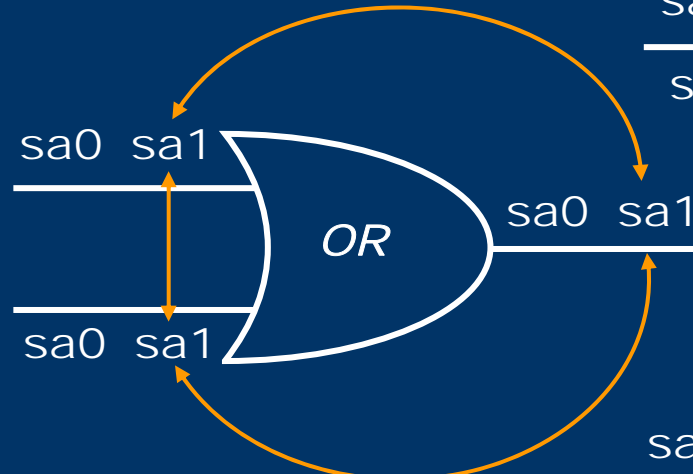
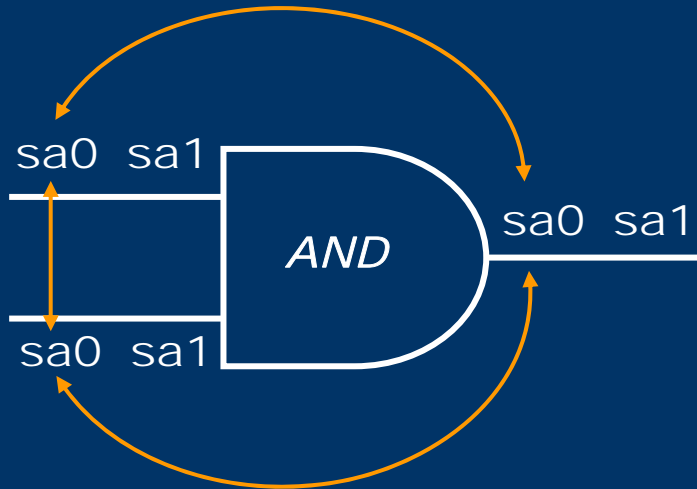
# Fault Equivalence

- Fault equivalence: Two faults  $f_1$  and  $f_2$  are equivalent if all tests that detect  $f_1$  also detect  $f_2$ .
- If faults  $f_1$  and  $f_2$  are equivalent then the corresponding faulty functions are identical.
- Fault collapsing: All single faults of a logic circuit can be divided into disjoint equivalence subsets, where all faults in a subset are mutually equivalent. A collapsed fault set contains one fault from each equivalence subset.

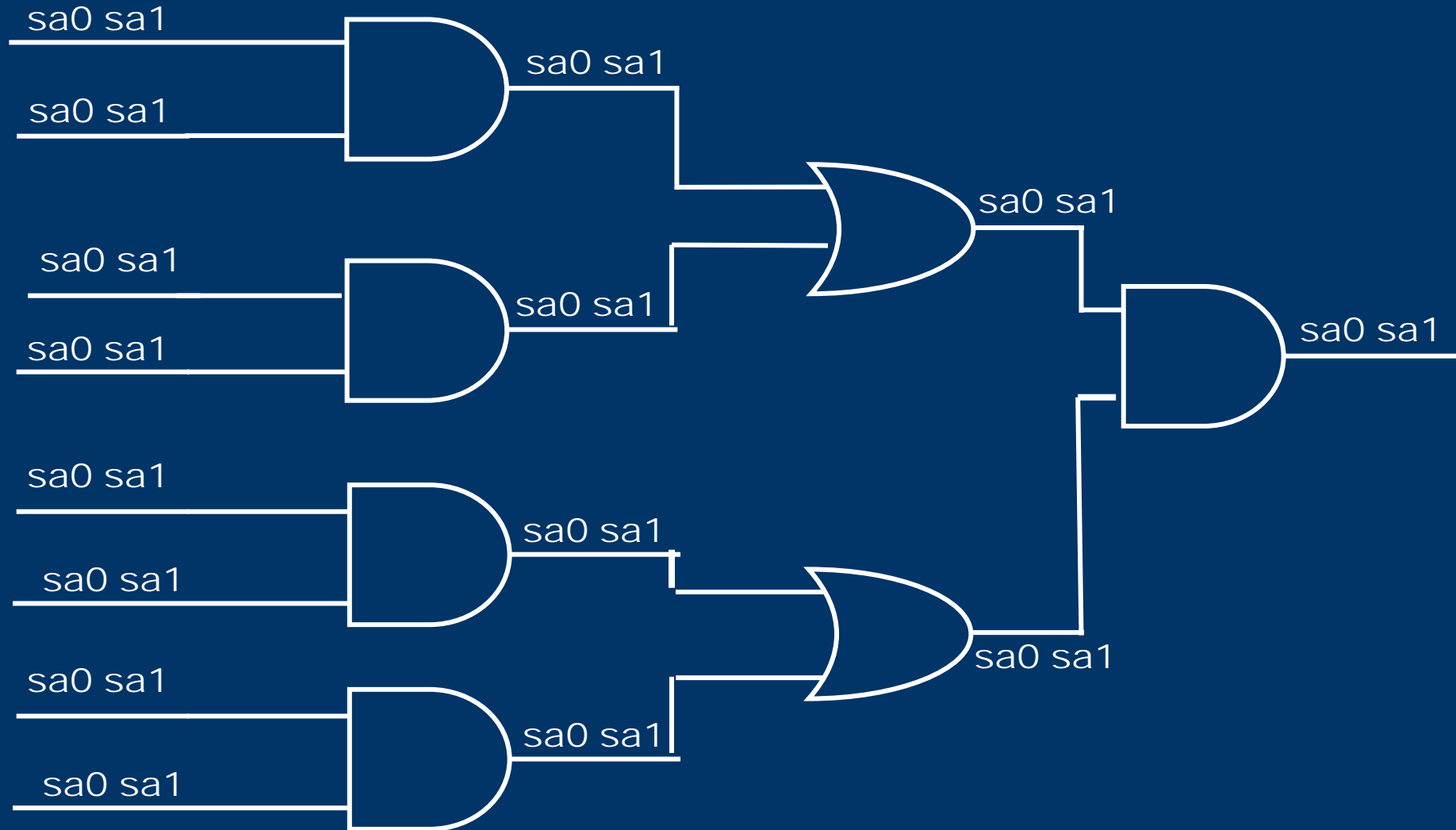
# Equivalence Rules



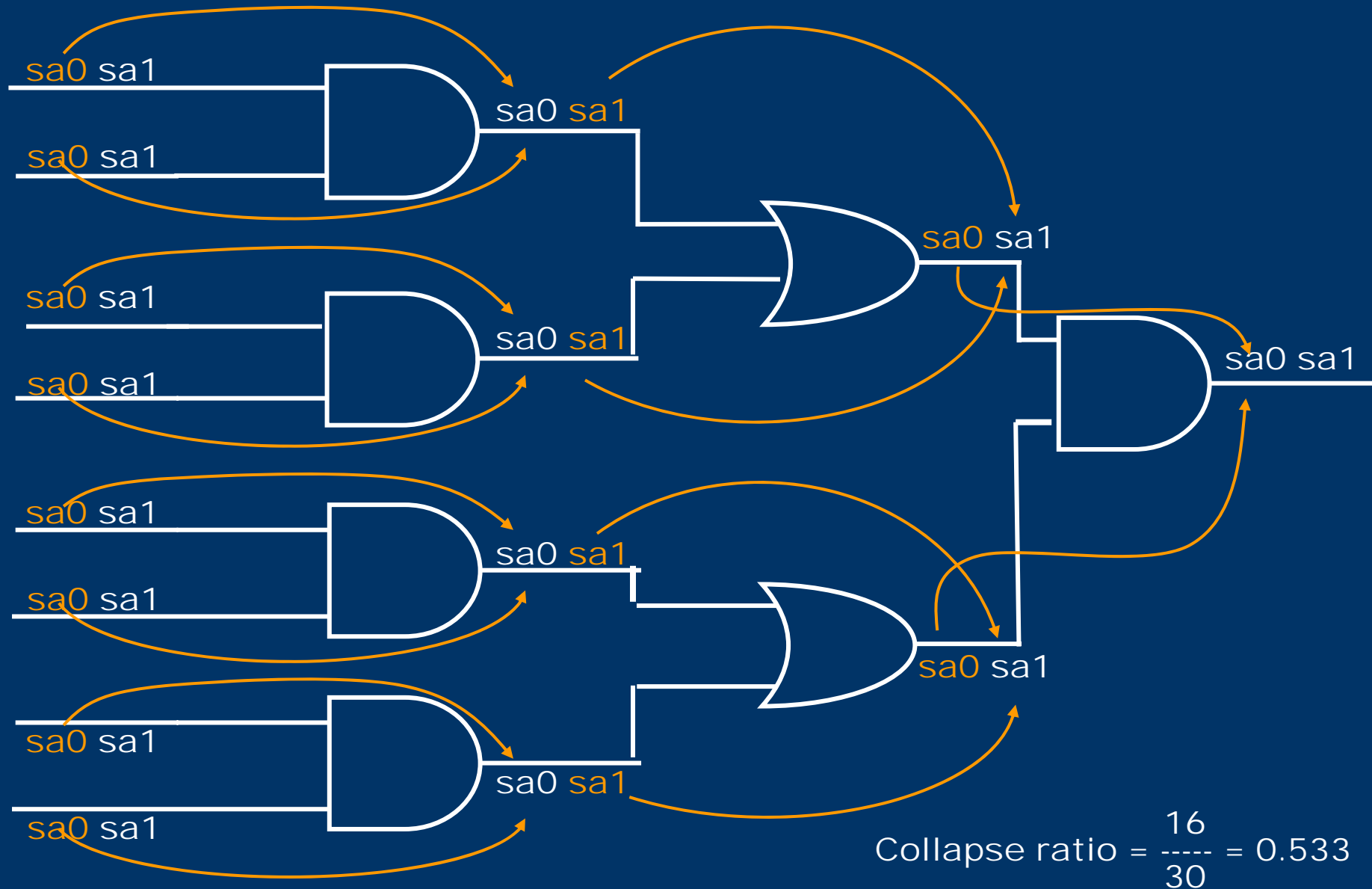
# Equivalence Rules



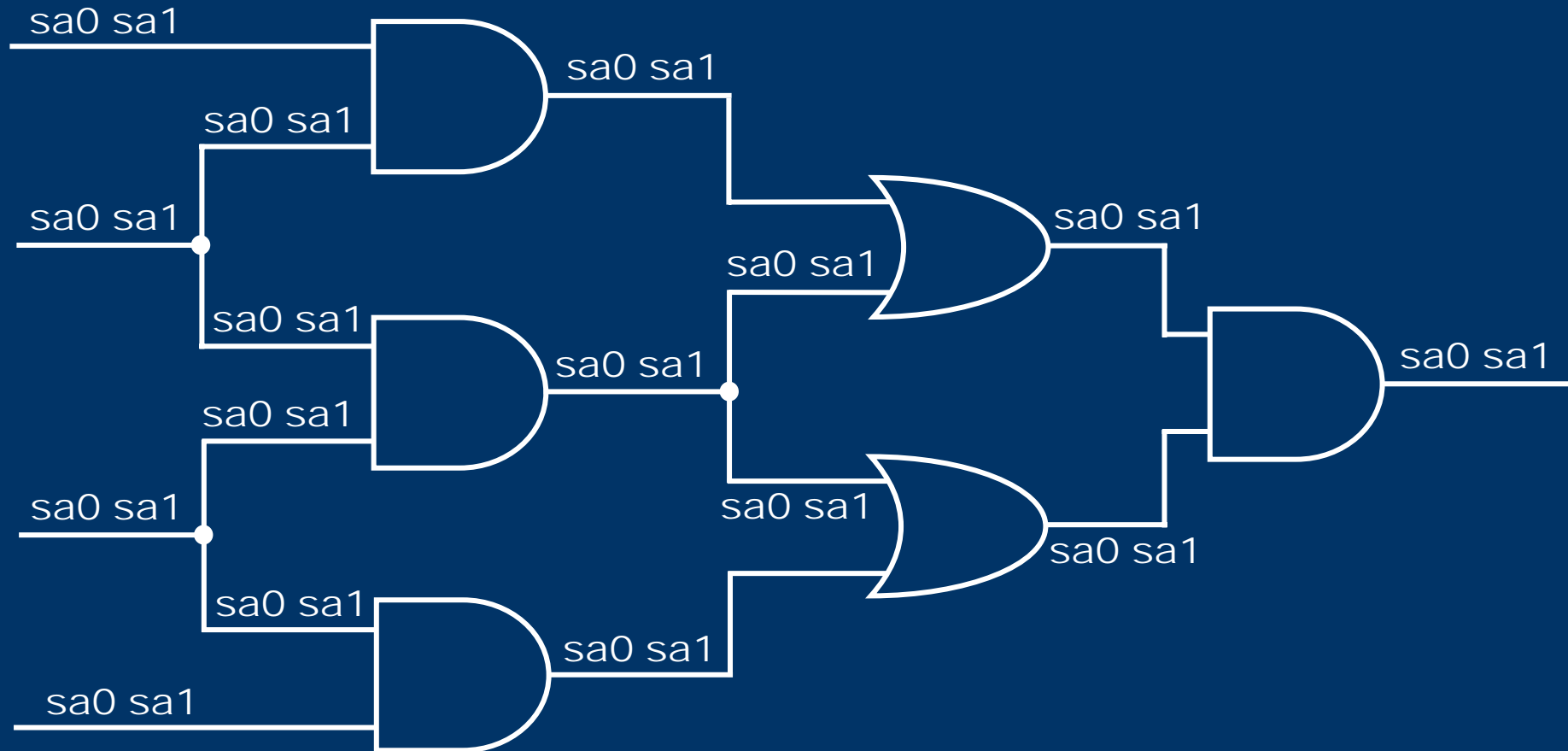
# Equivalence Example



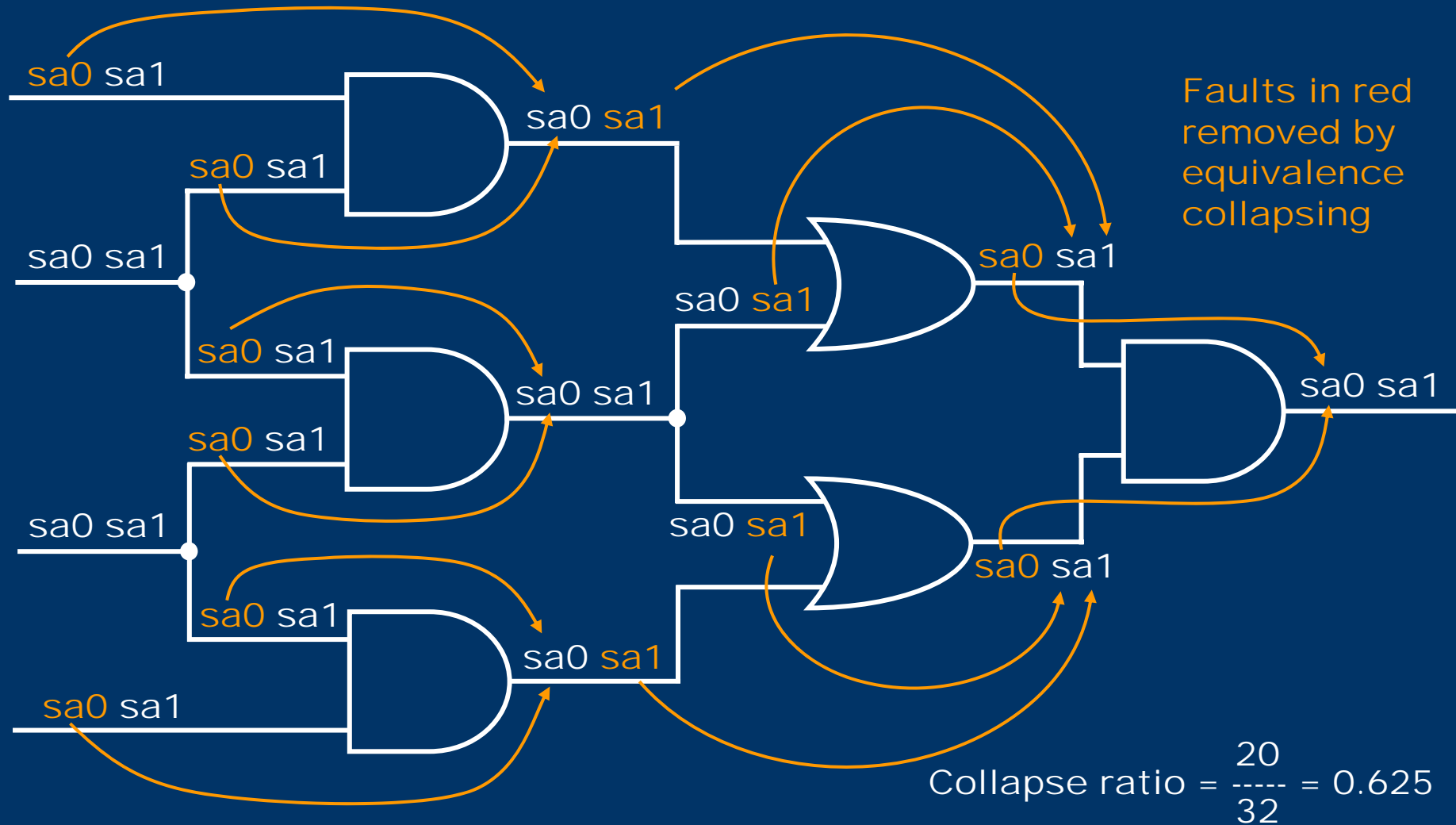
# Equivalence Example



# Equivalence Example



# Equivalence Example



# Fault Dominance

- If all tests of some fault  $F1$  detect another fault  $F2$ , then  $F2$  is said to dominate  $F1$ .
- Dominance fault collapsing: If fault  $F2$  dominates  $F1$ , then  $F1$  is removed from the fault list.

# Dominance Example



# Dominance Example

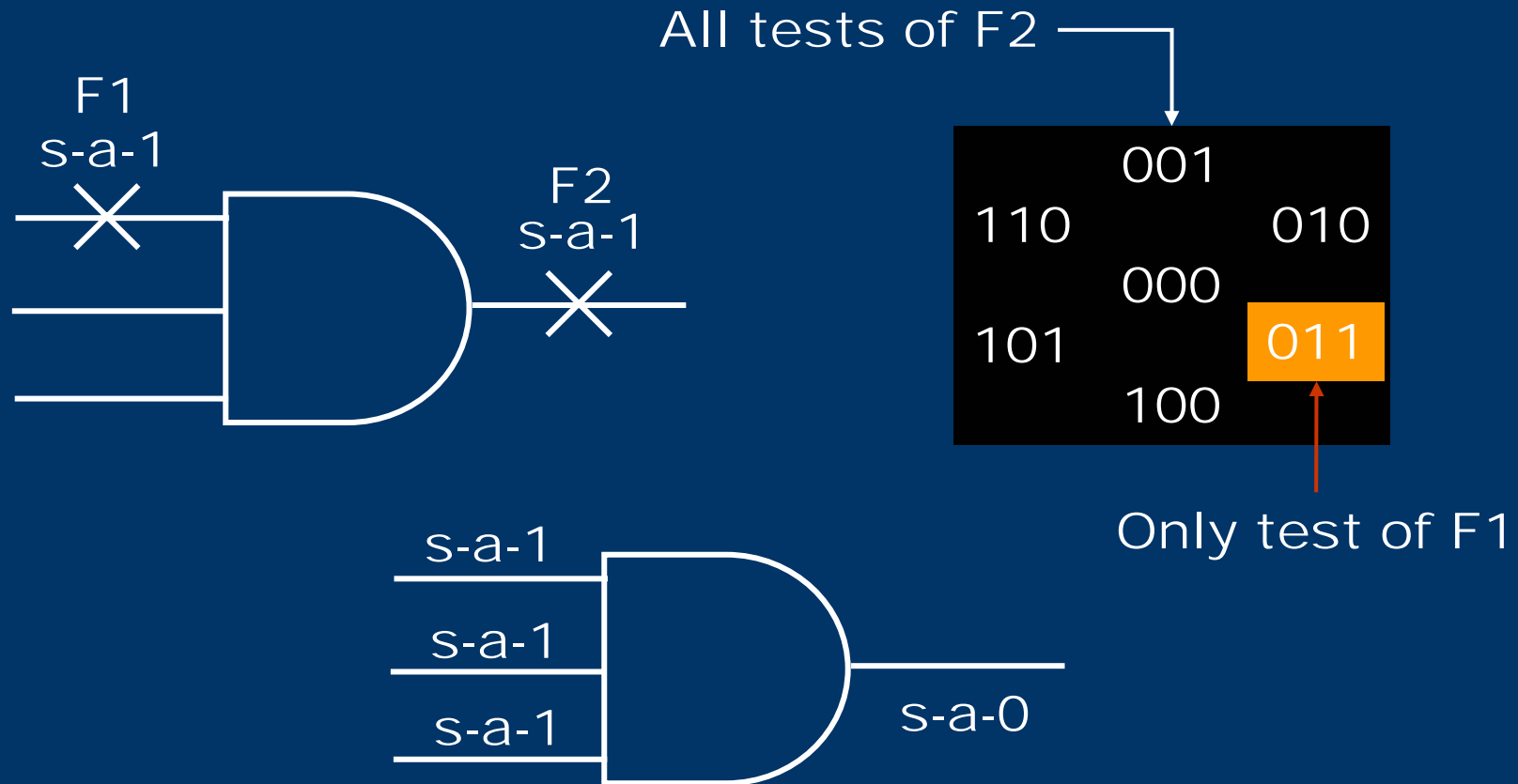


All tests of F2

	001	
110		010
	000	011
101		
	100	

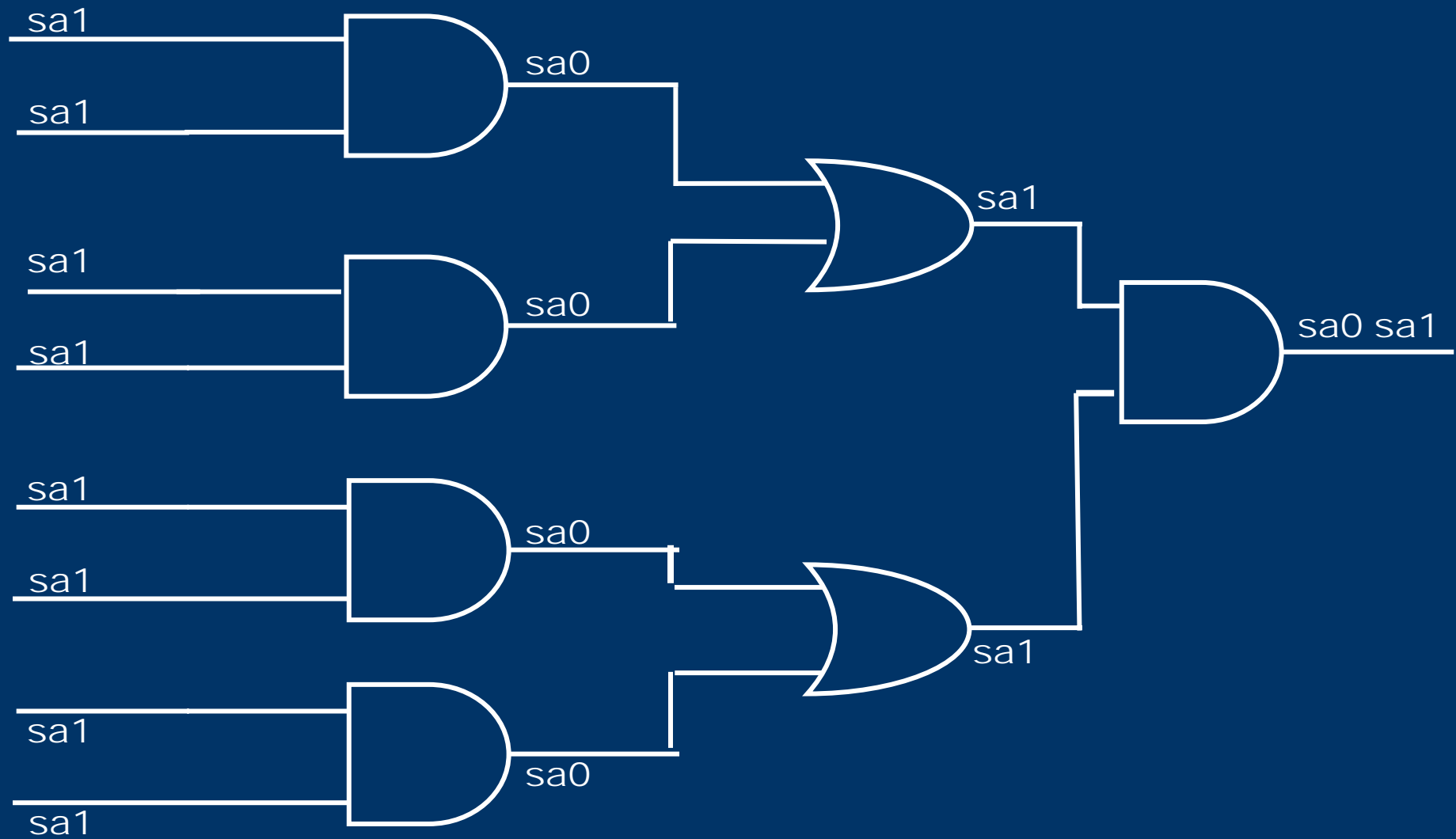
Only test of F1

# Dominance Example

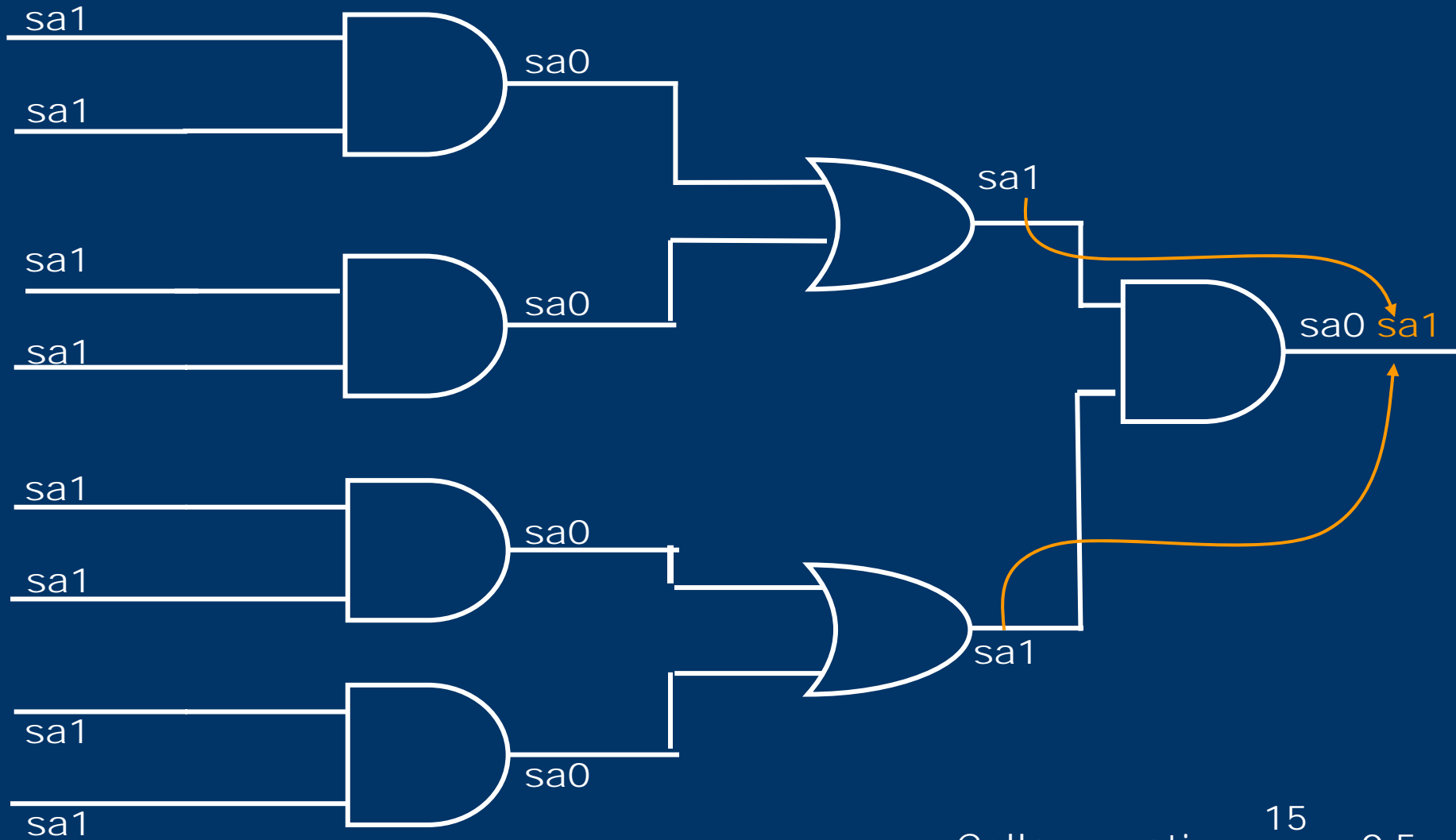


A dominance collapsed fault set

# Dominance Example

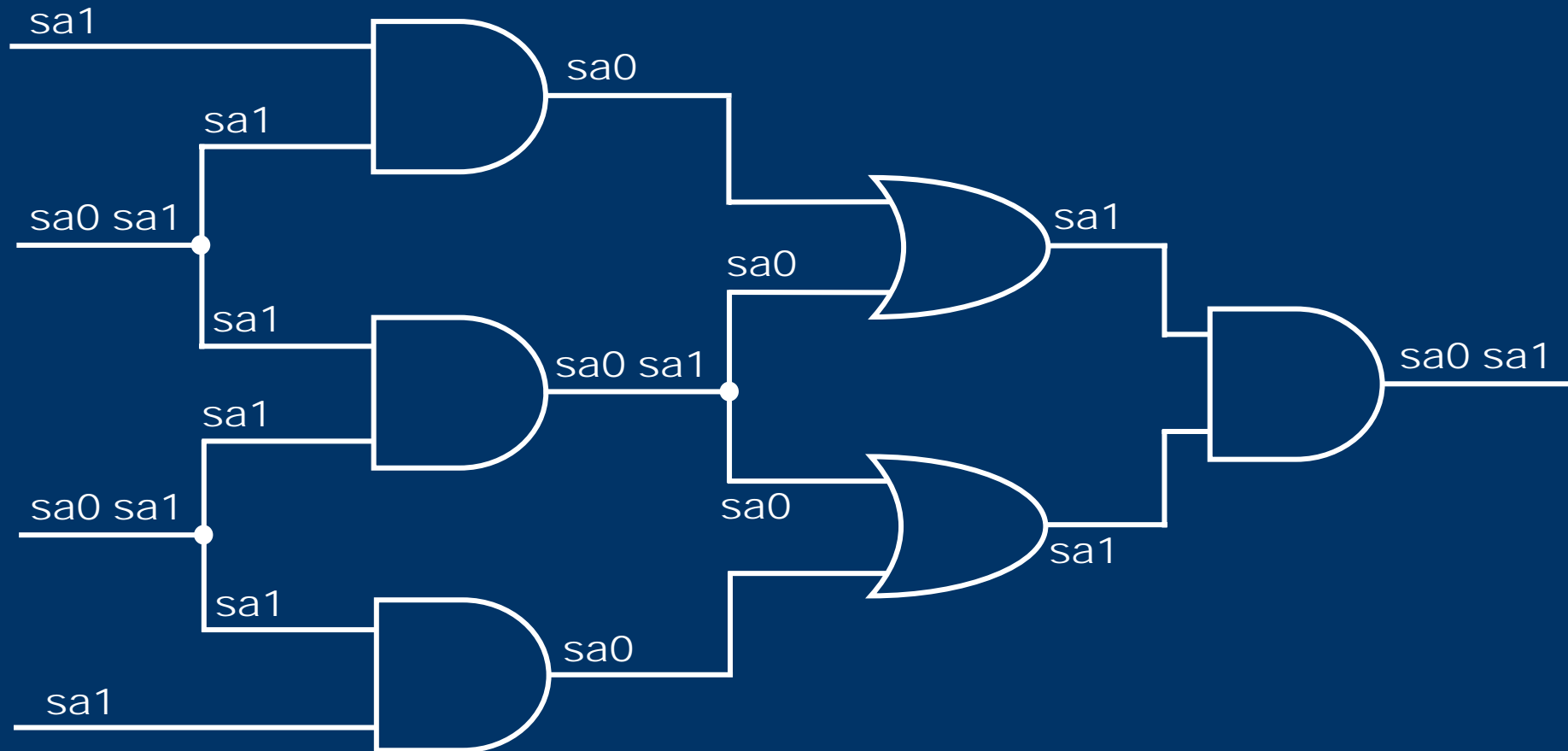


# Dominance Example

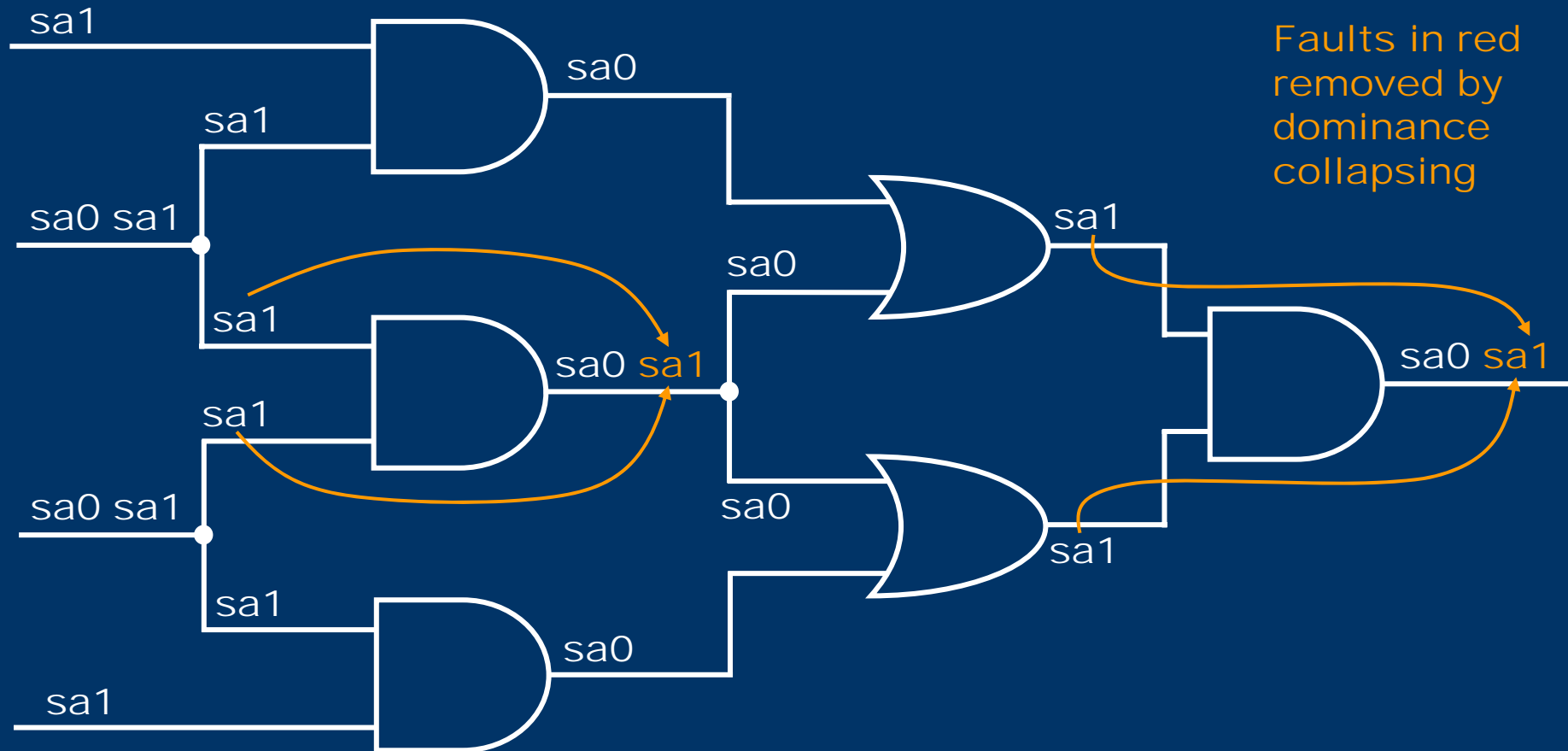


$$\text{Collapse ratio} = \frac{15}{30} = 0.5$$

# Dominance Example



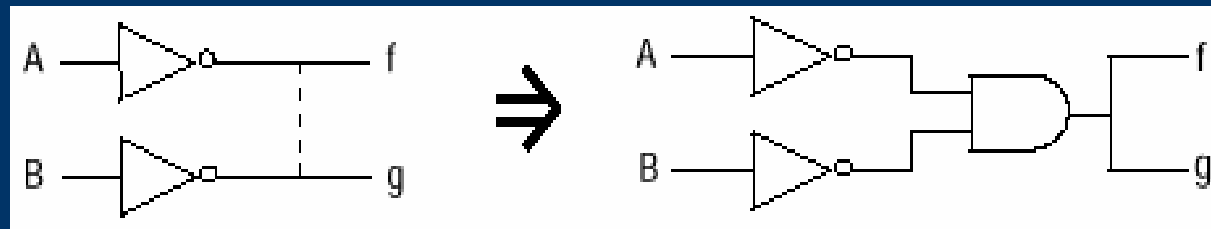
# Dominance Example



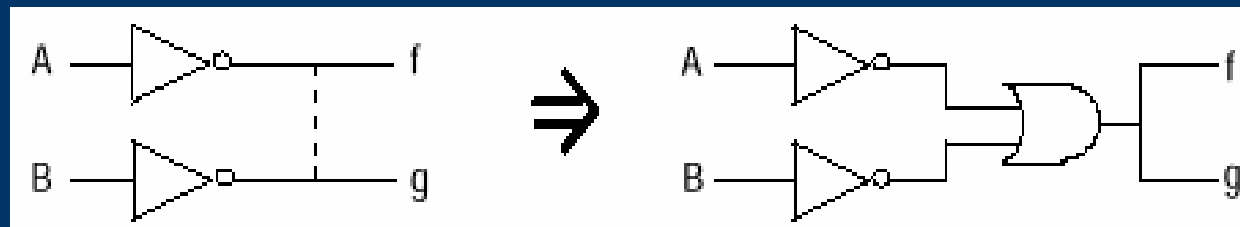
$$\text{Collapse ratio} = \frac{18}{32} = 0.5625$$

# Bridging Faults

- Two or more normally distinct points (lines) are shorted together
  - Logic effect depends on technology
  - Wired- AND for TTL

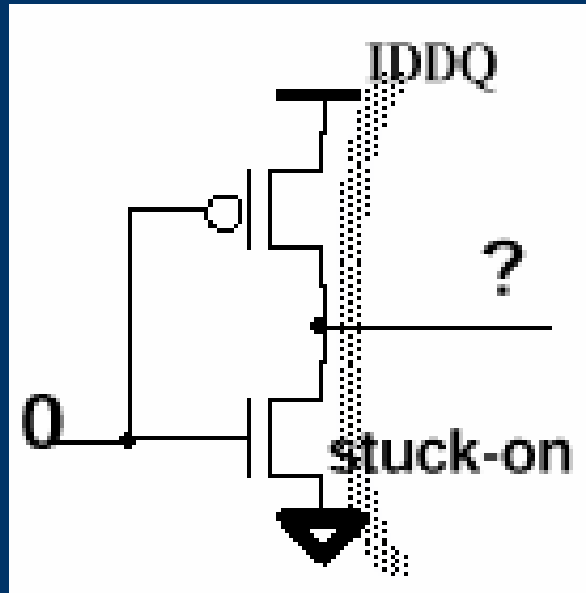


- Wired- OR for ECL



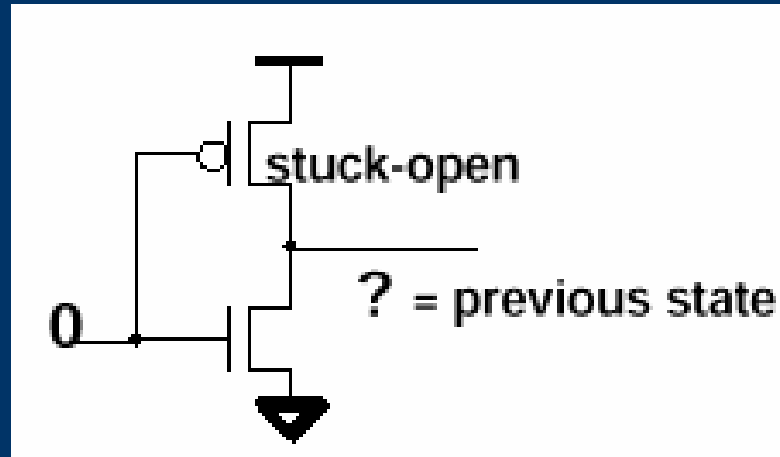
- CMOS?

# CMOS Transistor Stuck- Short



- Transistor stuck- on may cause ambiguous logic level
  - depends on the relative impedances of the pull- up & pull- down networks
- When input is low, both P and N transistors are conducting causing increased quiescent current, called  $I_{DDQ}$  fault.

# CMOS Transistor Stuck- OPEN



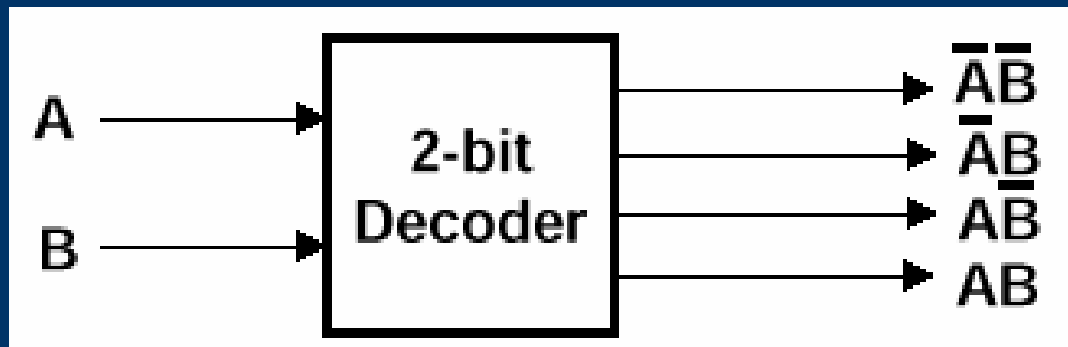
- Transistor stuck- open may cause output floating.

# Functional Faults

- Fault effects modeled at a higher level than logic for function modules, such as
  - Decoders
  - Multiplexers
  - Adders
  - Counters
  - RAMs
  - ROMs

# Functional Faults of Decoder

- $f(L_i / L_j)$ : Instead of line  $L_i$ , Line  $L_j$  is selected
- $f(L_i / L_i + L_j)$ : In addition to  $L_i$ ,  $L_j$  is selected
- $f(L_i / 0)$ : None of the lines are selected



# Memory Faults

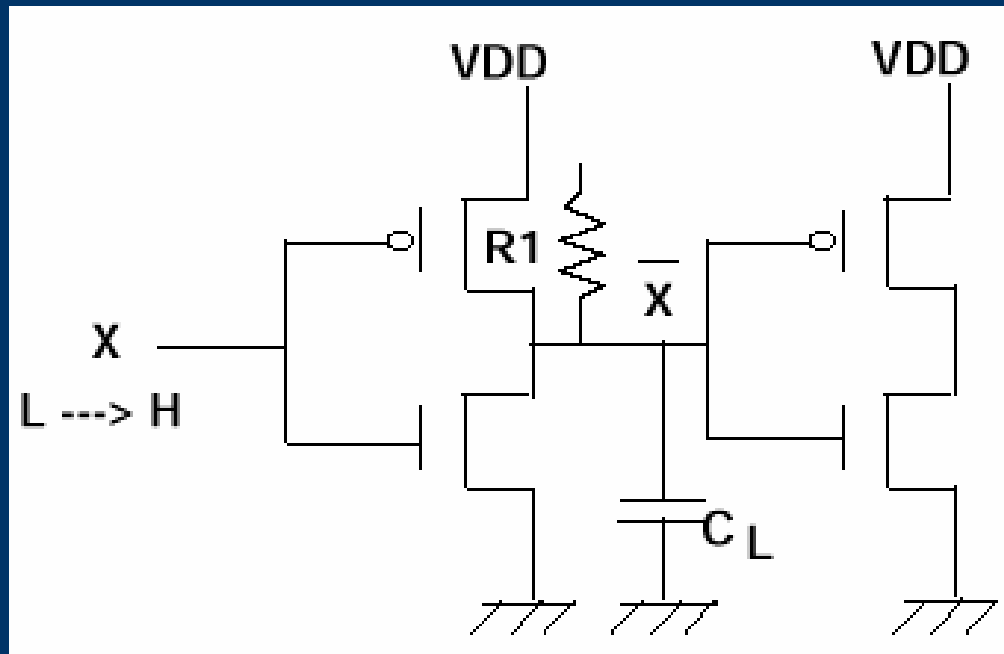
- Pattern- sensitive faults: the presence of a faulty signal depends on the signal values of the nearby points
  - Most common in DRAMs



- Adjacent cell coupling faults
  - Pattern sensitivity between a pair of cells

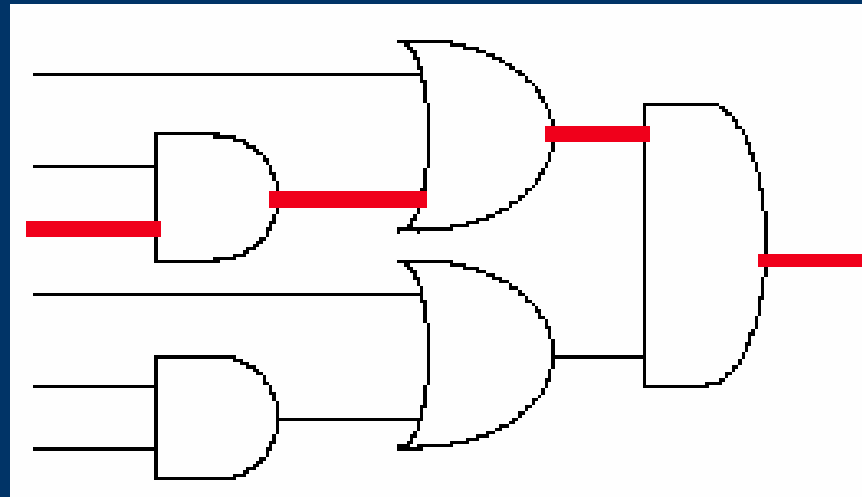
# Gate- Delay- Fault

- Slow to rise, slow to fall
  - x is slow to rise when channel resistance R1 is abnormally high



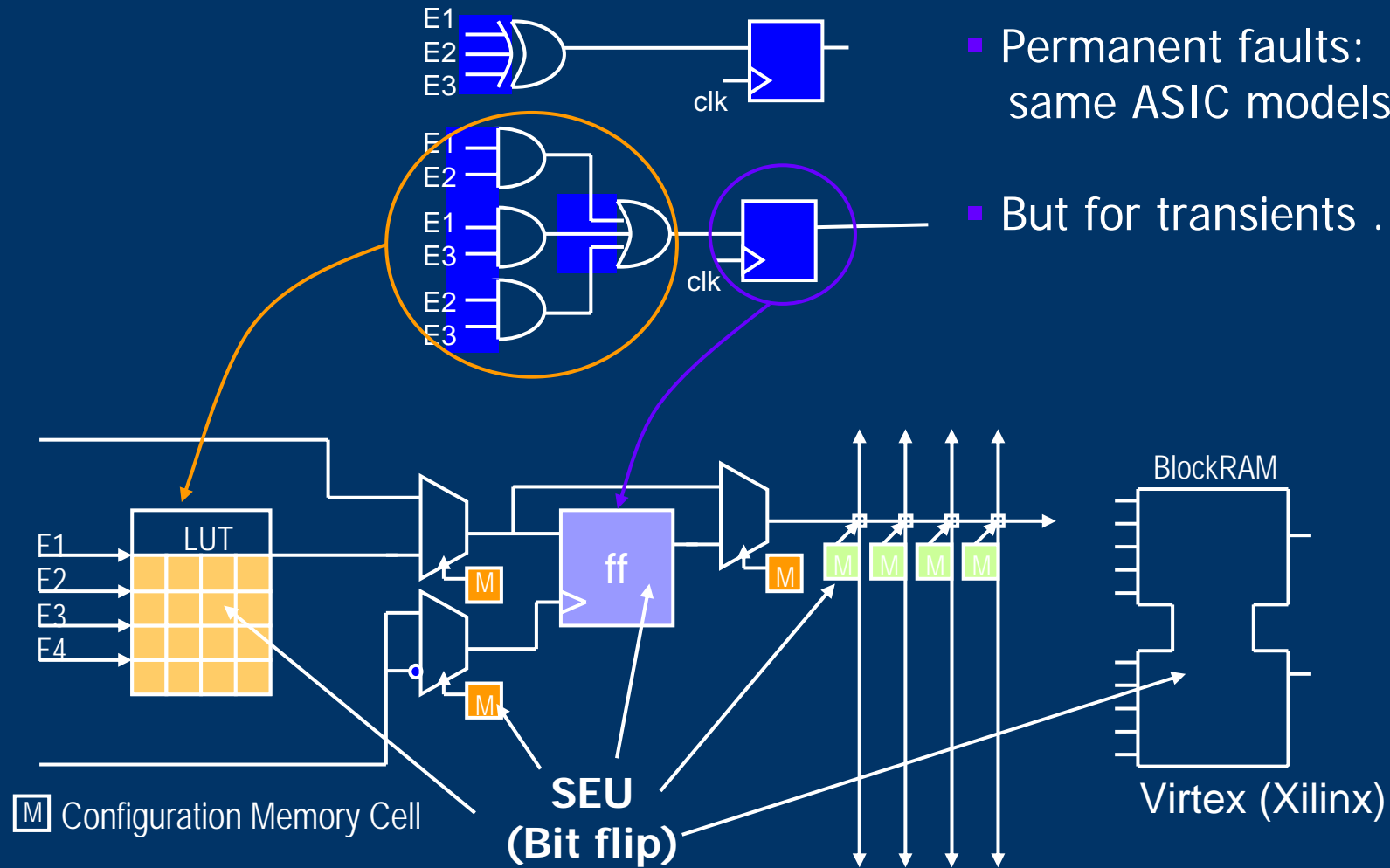
# Path- Delay- Fault

- Propagation delay of the path exceeds the clock interval.
- The number of paths grows exponentially with the number of gates.



# Faults in FPGAs

## FPGA building blocks:

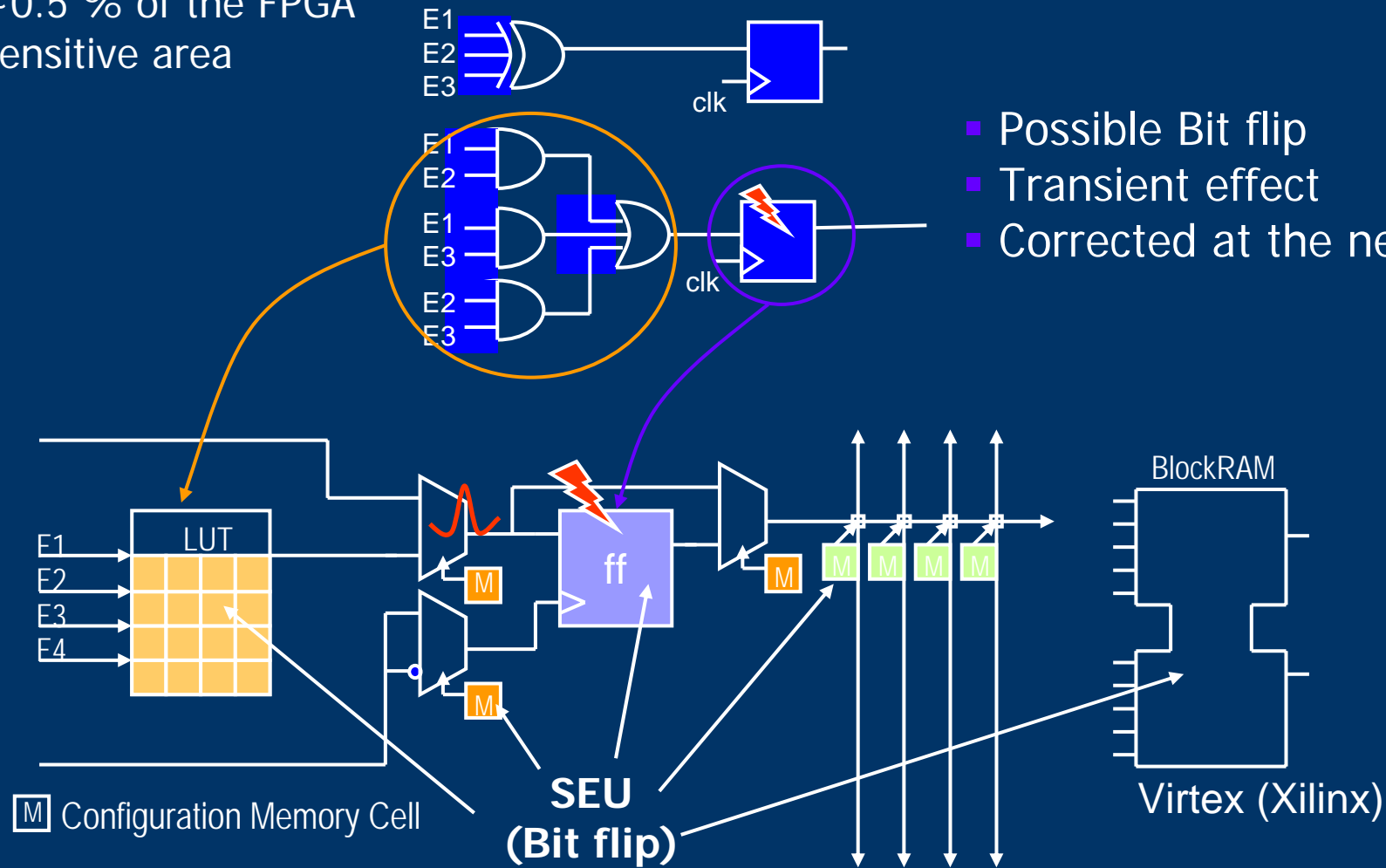


- Permanent faults: same ASIC models apply
- But for transients ...

# Effect of Transients in SRAM-based FPGAs

## CLB Comb. Logic:

~0.5 % of the FPGA sensitive area

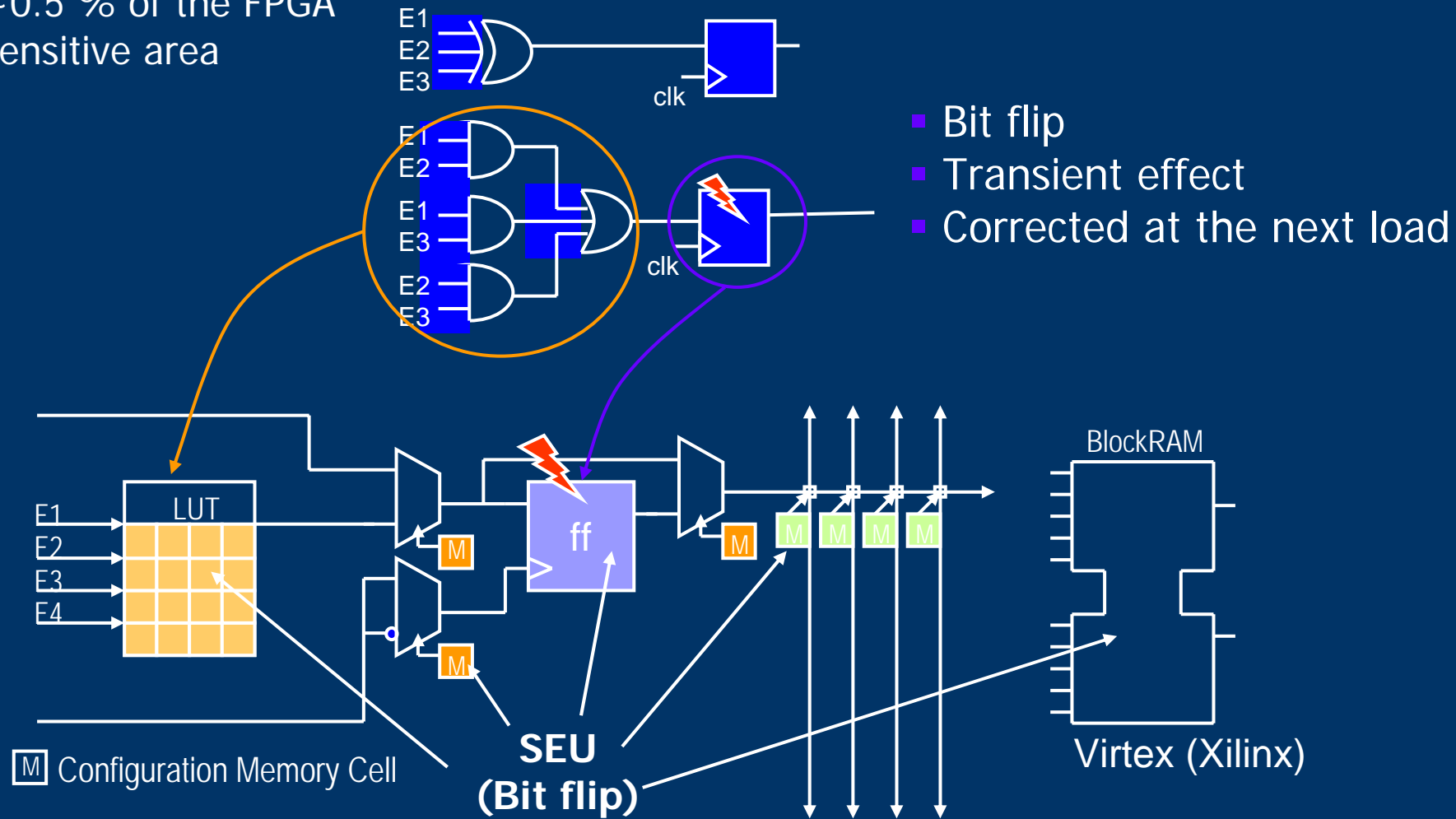


- Possible Bit flip
- Transient effect
- Corrected at the next load

# Effect of Transients in SRAM-based FPGAs

## CLB Flip-flops:

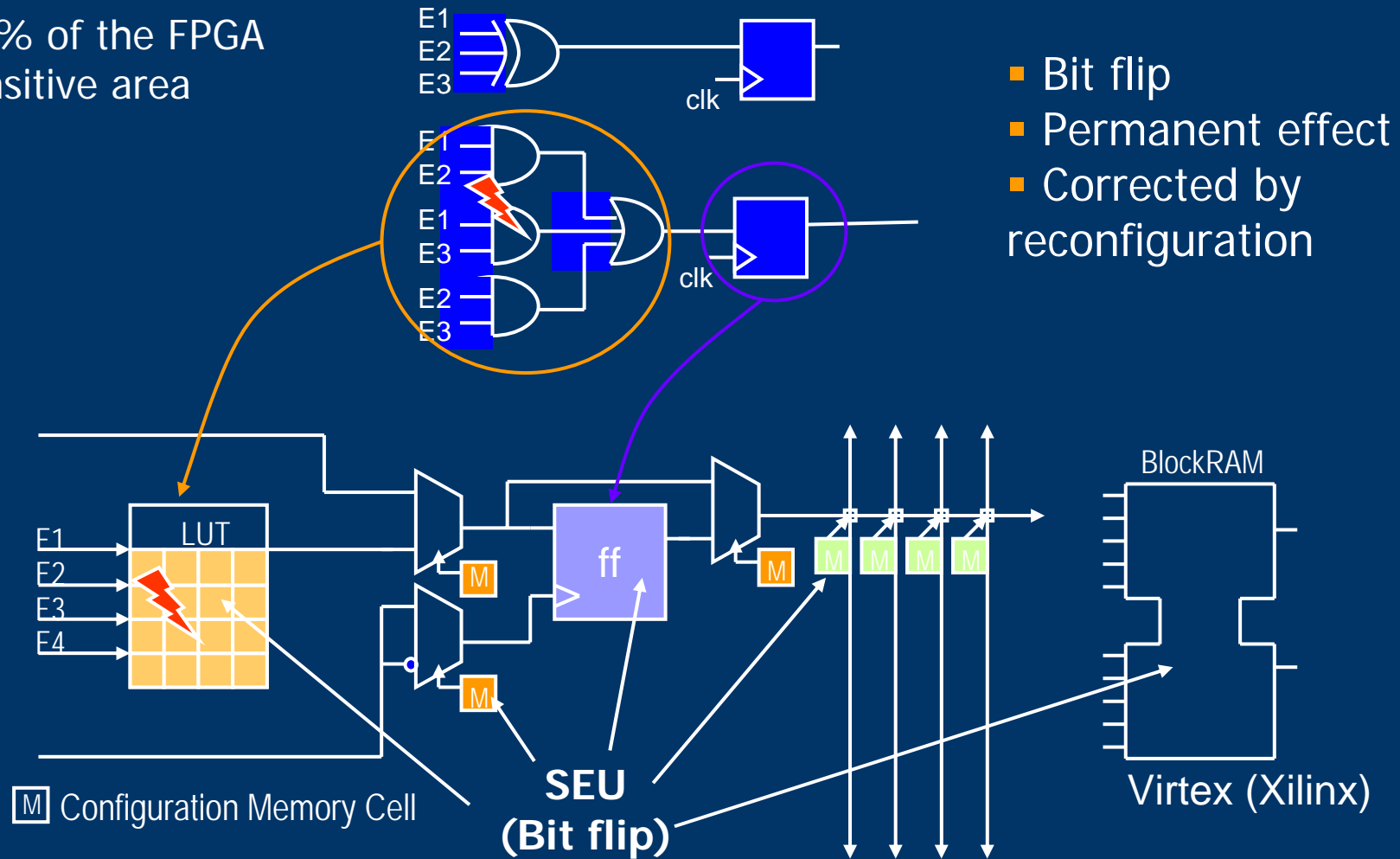
~0.5 % of the FPGA sensitive area



# Effect of Transients in SRAM-based FPGAs

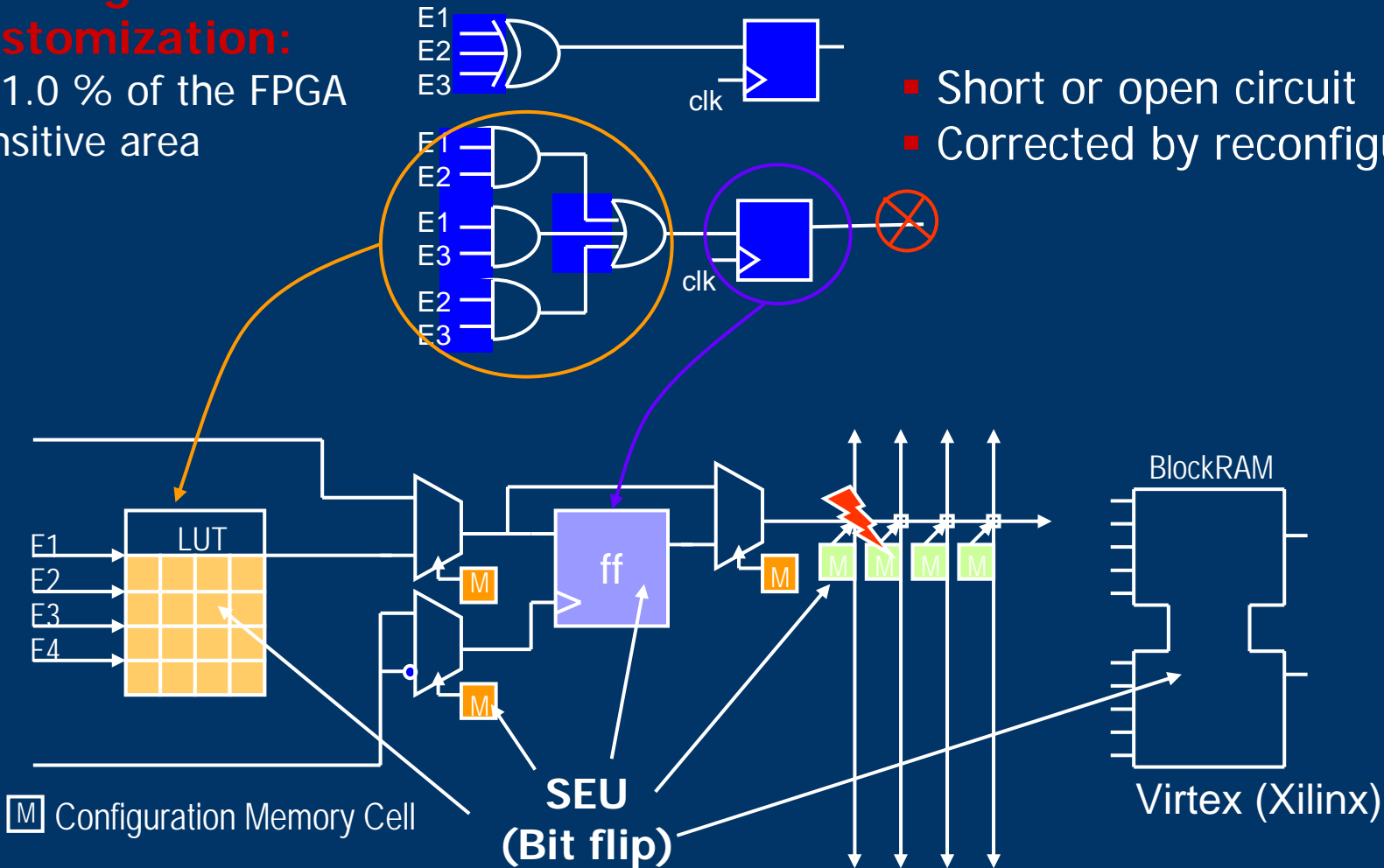
## CLB LUTs:

~8% of the FPGA sensitive area



# Effect of Transients in SRAM-based FPGAs

**Routing and CLB customization:**  
~91.0 % of the FPGA sensitive area



# Summary

- Fault models are analyzable approximations of defects and are essential for a test methodology.
- For digital logic single stuck-at fault model offers best advantage of tools and experience.
- Many other faults (bridging, stuck-open and multiple stuck-at) are largely covered by stuck-at fault tests.
- Memory and analog circuits need other specialized fault models and tests.
- Transient faults may have permanent effects in FPGAs