EE434
ASIC & Digital Systems

Fault Modeling

Spring 2016

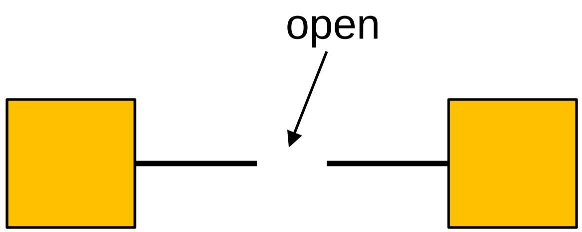
Dae Hyun Kim
daehyun@eecs.wsu.edu

Logical Fault Models

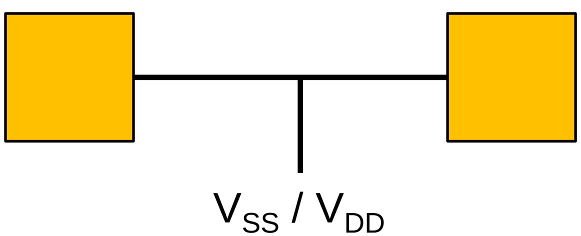
- We will discuss permanent faults.
- Single-fault assumption
  - There is at most one logical fault.
- Structural fault models
  - Components are fault-free.
  - Only interconnections are affected.

Faults

- Open



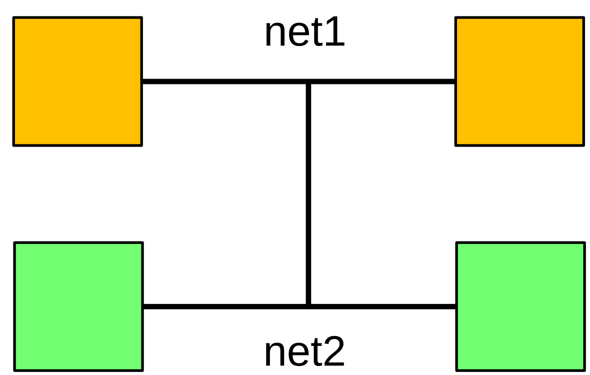
- Short



stuck-at-0

stuck-at-1

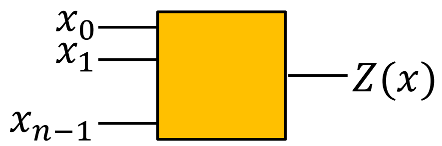
stuck-at-v (s-a-v)



bridging fault

Fault Detection (Combinational Logic)

- Input: $x = x_{n-1}x_{n-2} \dots x_1x_0$
- Output: $Z(x)$

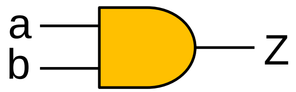


- A test vector t detects a fault f iff $Z_f(t) \neq Z(t)$.
  - t: test input
  - $Z(t)$: expected (correct) output
  - $Z_f(t)$: faulty output

- $Z_f(t) \neq Z(t) \Leftrightarrow Z_f(t) \oplus Z(t) = 1$

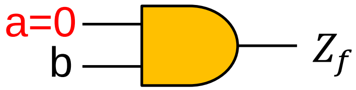
Fault Detection (Combinational Logic)

- Example



- $Z = a \cdot b$

- Stuck-at-0 fault

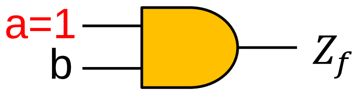


- $Z_f = 0 \cdot b = 0$

- $Z \oplus Z_f = 1 \Leftrightarrow (a \cdot b) \oplus 0 = 1 \Leftrightarrow a \cdot b = 1 \Leftrightarrow a = 1, b = 1$
  - If we apply (a, b) = (1, 1), we can detect the stuck-at-0 fault.

Fault Detection (Combinational Logic)

- Stuck-at-1 fault



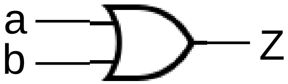
- $Z_f = 1 \cdot b = b$

- $Z \oplus Z_f = 1 \Leftrightarrow (a \cdot b) \oplus b = 1 \Leftrightarrow a = 0, b = 1$
  - If we apply (a, b) = (0, 1), we can detect the stuck-at-1 fault at a.

- What input vector can detect a stuck-at-1 fault at input b?
  - (a, b) = (1, 0)

Fault Detection (Combinational Logic)

- Example

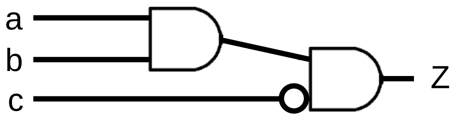


- Find input vectors that can detect stuck-at-0 faults at a and b.

- Find input vectors that can detect stuck-at-1 faults at a and b.

Fault Detection (Combinational Logic)

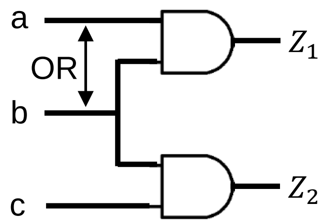
- Example



- Find input vectors that can detect stuck-at-0 faults at a and b.
- Find input vectors that can detect stuck-at-1 faults at a and b.

Fault Detection (Combinational Logic)

- Example
  - Find input vectors that can detect the following OR bridging fault.

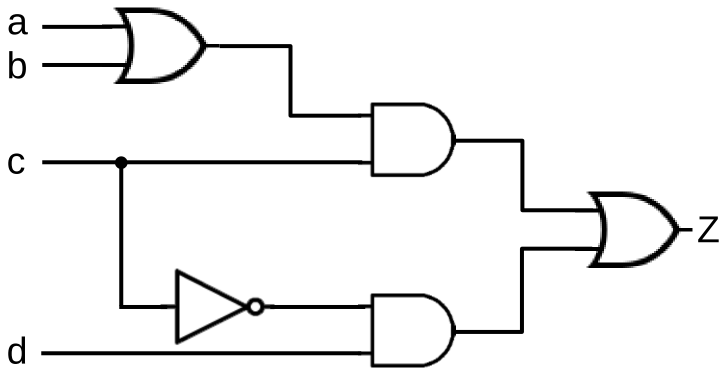


- $Z_1 = a \cdot b, Z_{1f} = a + b$
- $Z_2 = b \cdot c, Z_{2f} = (a + b) \cdot c$

- $Z_1 \oplus Z_{1f} = (a \cdot b) \oplus (a + b) = 1 \Leftrightarrow abc = 01*, 10*$
- $Z_2 \oplus Z_{2f} = (b \cdot c) \oplus \{(a + b) \cdot c\} = 1 \Leftrightarrow abc = 101$

Fault Detection (Combinational Logic)

- Example



Fault Detection (Combinational Logic)

- Detectability
  - A fault f is *undetectable* if there is no test vector t that detects f.
    - $Z_f(x) = Z(x)$

- A combinational circuit that contains an undetectable stuck fault is *redundant*.

- Example
  - n-input AND gate: $Z(x) = x_0 \cdot \cdots \cdot x_{n-1}$
  - stuck-at-1 at x_0
    - $Z(x) \oplus Z_f(x) = (x_0 \cdot \cdots \cdot x_{n-1}) \oplus (x_1 \cdot \cdots \cdot x_{n-1}) = 1 \Leftrightarrow x = 011..1$
  - If the stuck-at-1 at x_0 is undetectable, an n-input AND gate with a constant 1 value on x_0 is logically equivalent to an (n-1)-input AND gate with input $x_1, \ldots, x_{n-1}$.

Fault Detection (Combinational Logic)

Undetectable fault	Simplification
AND (NAND) input s-a-1	Remove the input
AND (NAND) input s-a-0	Remove the gate, replace by 0(1)
OR (NOR) input s-a-0	Remove the input
OR (NOR) input s-a-1	Remove the gate, replace by 1(0)

Fault Detection (Sequential Logic)

- Needs a sequence of inputs.

- A test sequence T *strongly detects* fault f iff the output sequences $R(q,T)$ and $R_f(q_f,T)$ are different for every possible pair of initial states q and q_f.

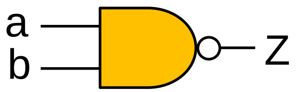
- A test sequence T *detects* fault f iff the output sequences $R(q,T)$ and $R_f(q_f,T)$ are different for every possible pair of initial states q and q_f and for some specified vector $t_i \in T$.

Fault Equivalence

- Two faults f and g are *functionally equivalent* iff $Z_f(x) = Z_g(x)$.
- A test t is said to *distinguish* between two faults f and g if $Z_f(t) \neq Z_g(t)$. Those faults are distinguishable.
- No test can distinguish between two functionally equivalent faults.
- If a test x distinguishes between f and g, $Z_f(x) \oplus Z_g(x) = 1$.

Equivalence Fault Collapsing

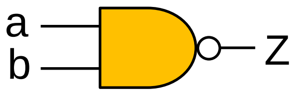
- Before collapsing
  - a
    - s-a-0, s-a-1
  - b
    - s-a-0, s-a-1
  - Z
    - s-a-0, s-a-1



- After collapsing
  - a
    - s-a-1
  - b
    - s-a-1
  - Z
    - s-a-0, s-a-1

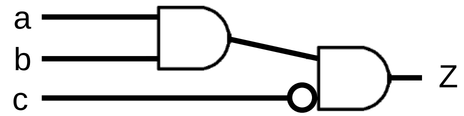
Dominance Fault Collapsing

- Before collapsing
  - a
    - s-a-0, s-a-1
  - b
    - s-a-0, s-a-1
  - Z
    - s-a-0, s-a-1
- After collapsing
  - a
    - s-a-1
  - b
    - s-a-1
  - Z
    - s-a-1

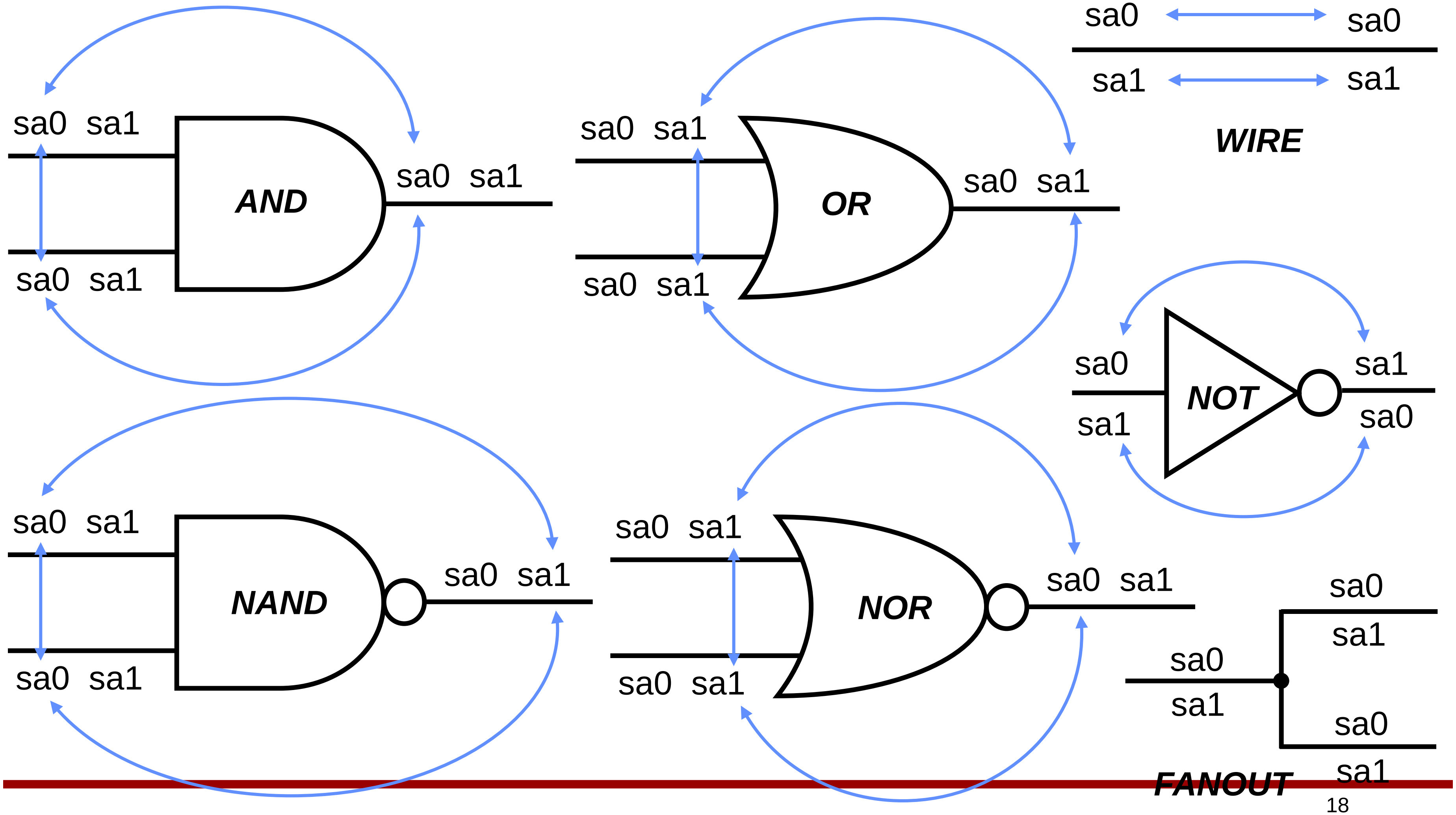


Equivalence Fault Collapsing & Dominance Fault Collapsing

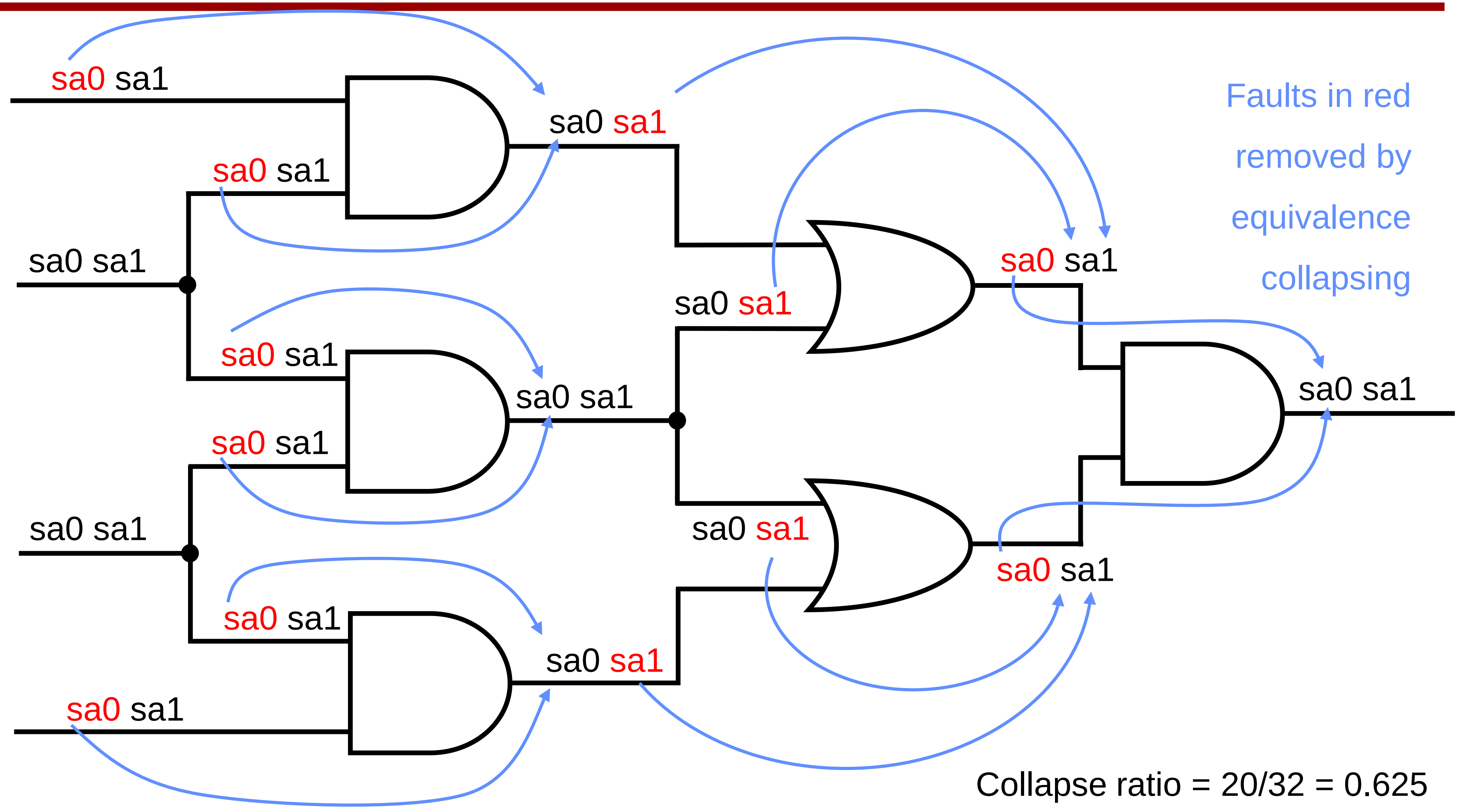
- Example



Equivalence Rules



Equivalence Example

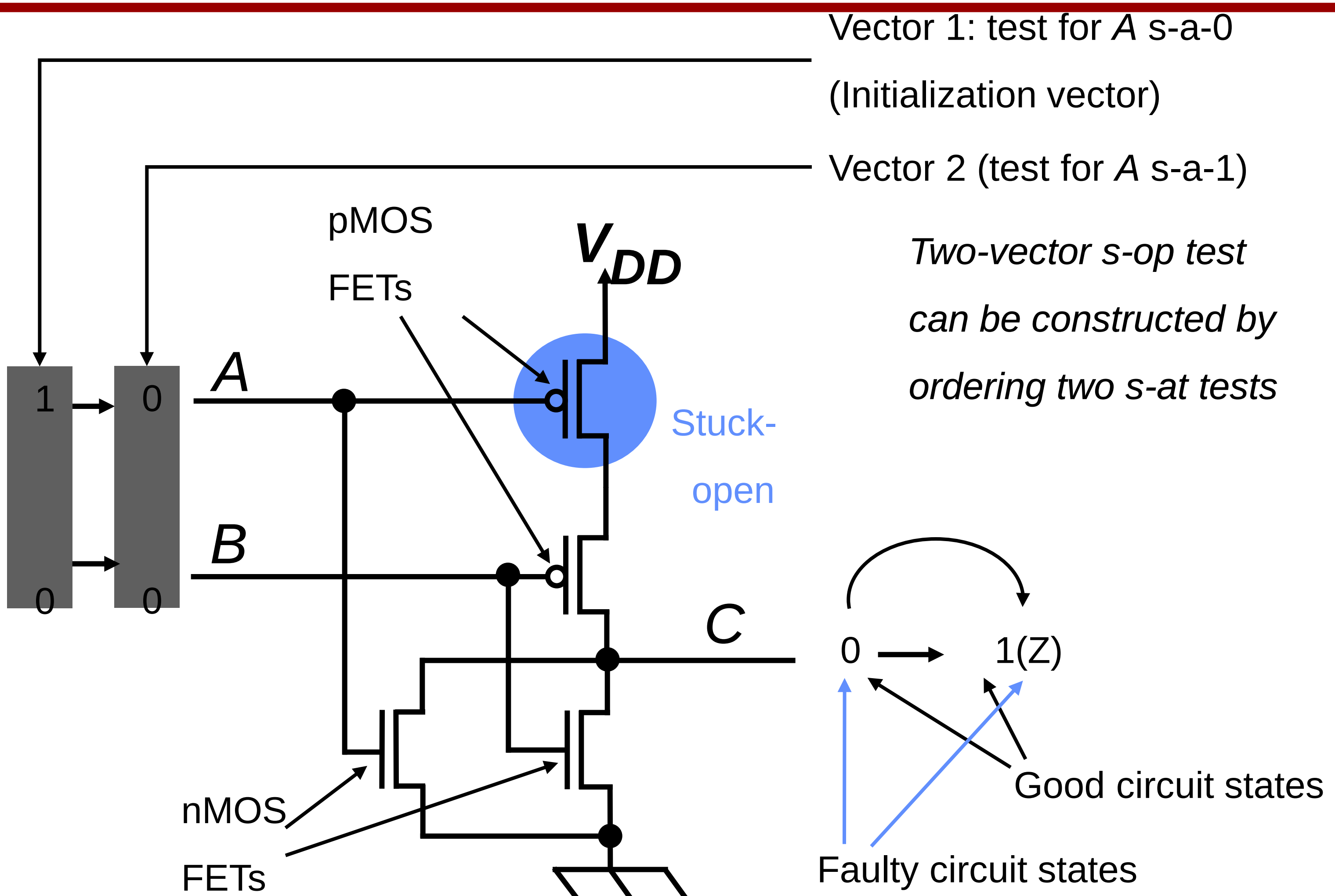


Collapse ratio = 20/32 = 0.625

Transistor (Switch) Faults

- MOS transistor is considered an ideal switch and two types of faults are modeled:
  - Stuck-open: a single transistor is permanently stuck in the open state.
  - Stuck-short: a single transistor is permanently shorted irrespective of its gate voltage.
- Detection of a stuck-open fault requires two vectors.
- Detection of a stuck-short fault requires the measurement of quiescent current (I_{DDQ}).

Stuck-Open Example



Two-vector s-op test can be constructed by ordering two s-at tests

Stuck-Short Example