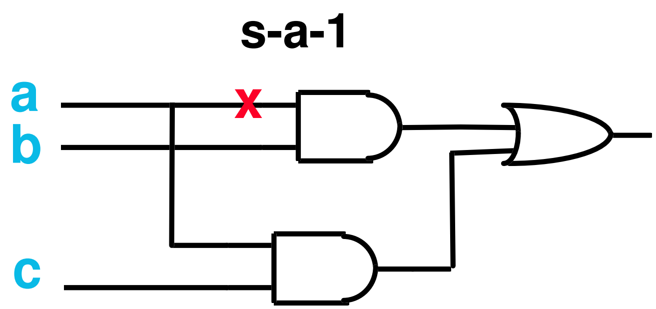
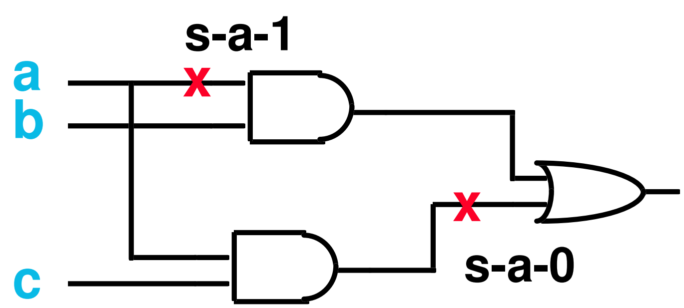
Fault Models

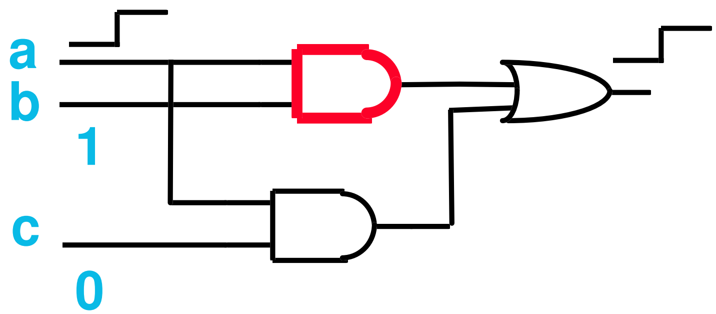


Single stuck-at fault

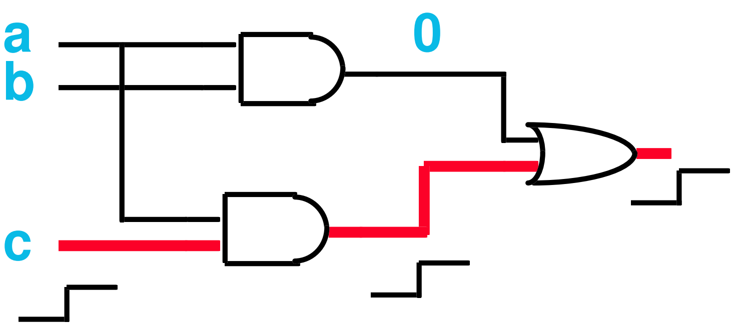


Multiple stuck-at faults

Fault Models - 2



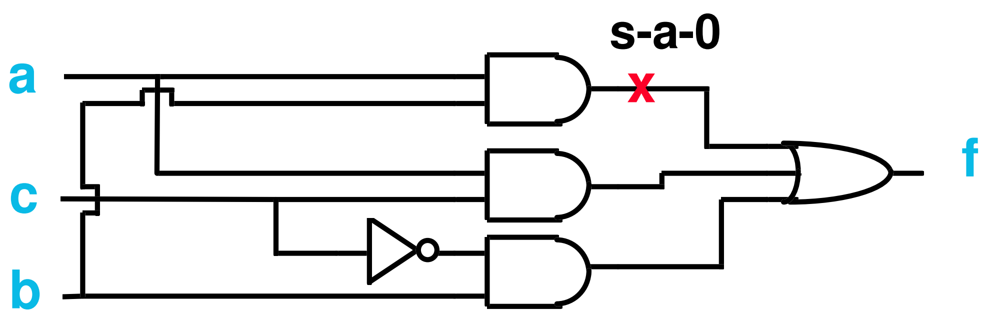
Gate delay fault



Path delay fault

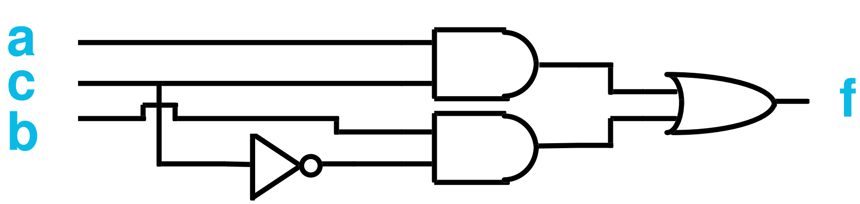
Redundancy and Testability

If a fault in a circuit is redundant, i.e., there is no test for it



a	b	c	f
1	1	-	1
1	-	1	1
-	1	0	1

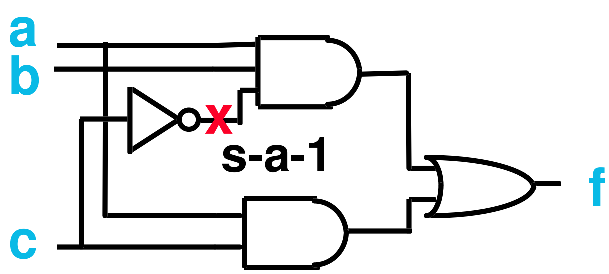
Replace line on which fault resides with a constant.

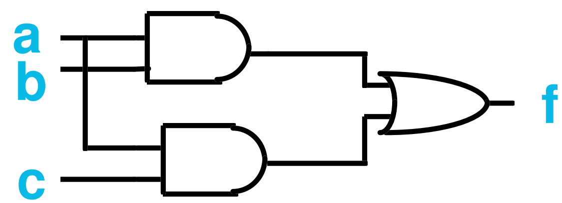


a	b	c	f
1	-	1	1
-	1	0	1

Prime + Irredundant = Testable

A prime and irredundant cover for a single-output function represents a two-level circuit that is fully testable for all single stuck-at faults.





1 1 0 1
1 - 1 1

non-prime irredundant cover

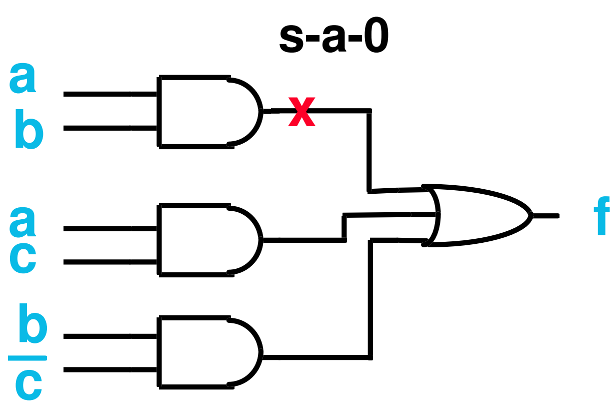
1 1 - 1
1 - 1 1

prime & irredundant cover

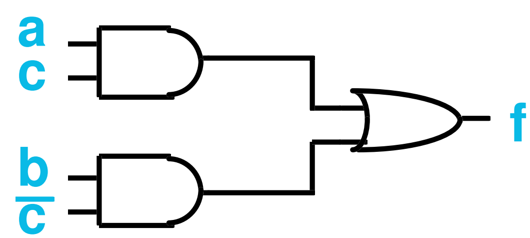
Correspondence

Primality ⇔ s-a-1 faults on and gate inputs

Irredundancy ⇔ s-a-0 faults on or gate inputs



prime but redundant cover



prime & irredundant cover

Multiple-Output Functions

Given the two-output function below

000	01
010	01
100	01
101	01
110	11
111	11

3 primes

1 1 -	1 1
1 - -	0 1
- - 0	0 1

They form a cover

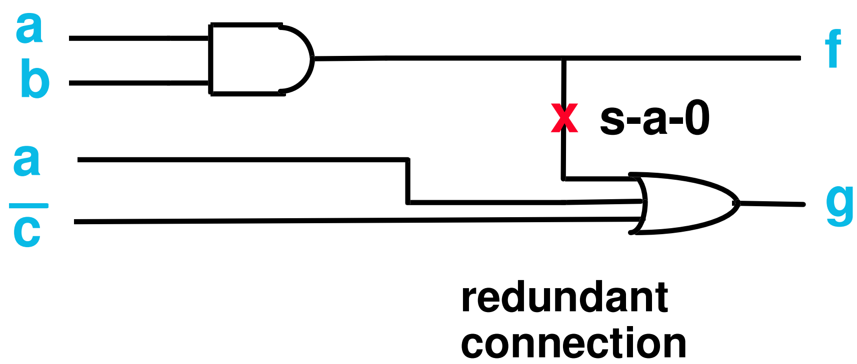
1 1 -	1 1
1 - -	0 1
- - 0	0 1

Prime & irredundant cover

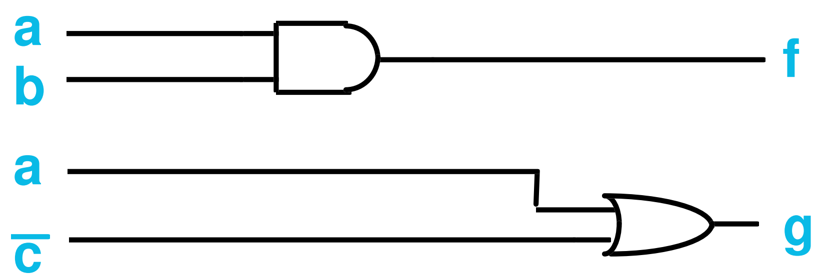
Is it fully testable for single stuck-at faults?

Multiple-Output Functions - 2

a b c	f g
1 1 -	1 1
1 - -	0 1
- - 0	0 1



a b c	f g
1 1 -	1 0
1 - 1	0 1
- - 0	0 1



Don't really want a prime cover

Testable Multiple-Output Covers

– Modify Quine-Mcluskey method

– Generate primes as usual

– During branch & bound covering check selected prime for unnecessary 1's in output part

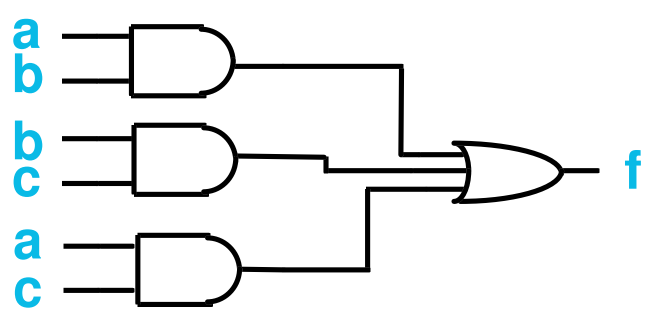
⇒ If there are unnecessary 1's, replace prime in current solution with maximally (output) reduced cube.

– Any solution will be fully testable

Multiple Stuck-At Faults

3^K Multifaults

Theorem: The set of tests detecting all single faults in a prime and irredundant single-output two-level circuit detect all multifaults.



$f = ab + bc + ac$

s-a-1 on a in ab

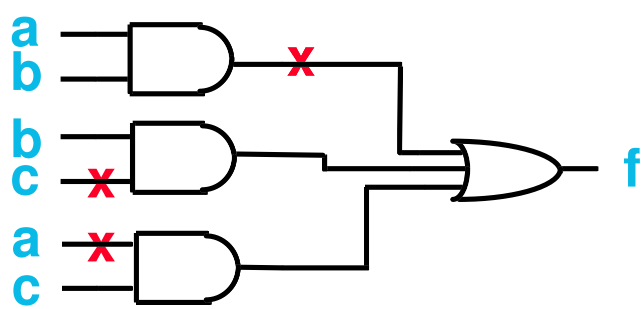
$f^* = b + bc + ac$

s-a-0 on bc

$f^* = b + ac$

Effects: removing literals in cubes (raising cube)
removing cubes from cover

Multiple Stuck-At Faults - 2

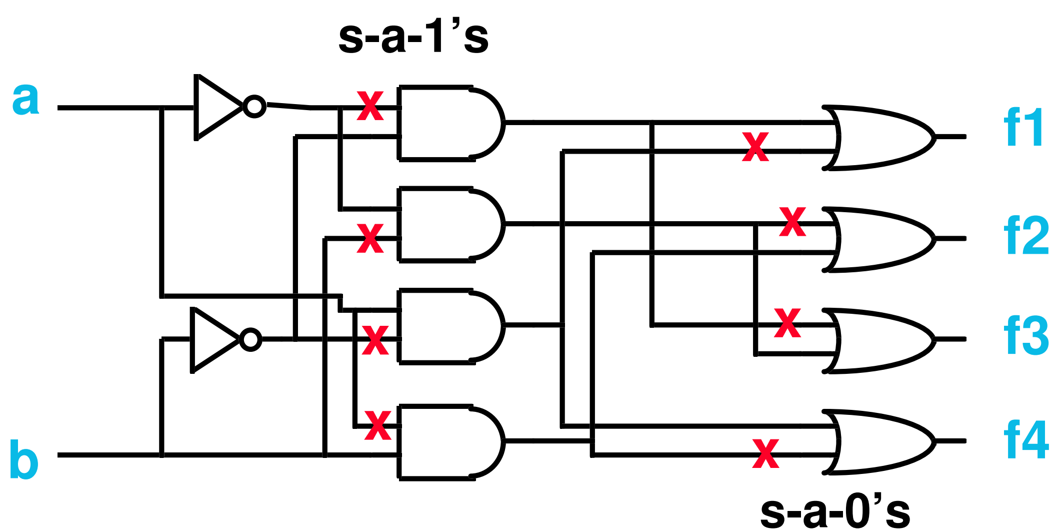


Three cases:

1. **Cubes uniformly removed from f: s-a-0 test for some removed cube will detect multifault**
2. **Cubes uniformly raised in f: s-a-1 test for some removed literal in cube will detect multifault**
3. **Some cubes removed, some raised: s-a-1 test for some removed literal in uremoved cube will detect multifault.**

Multiple-Output Circuits

Theorem does not generalize



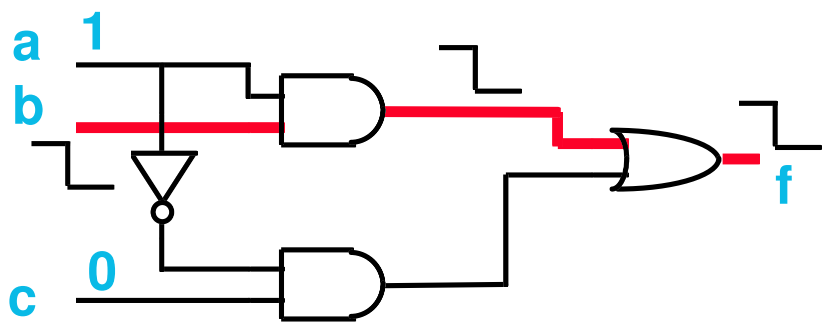
fully single-fault testable

8-fault is redundant

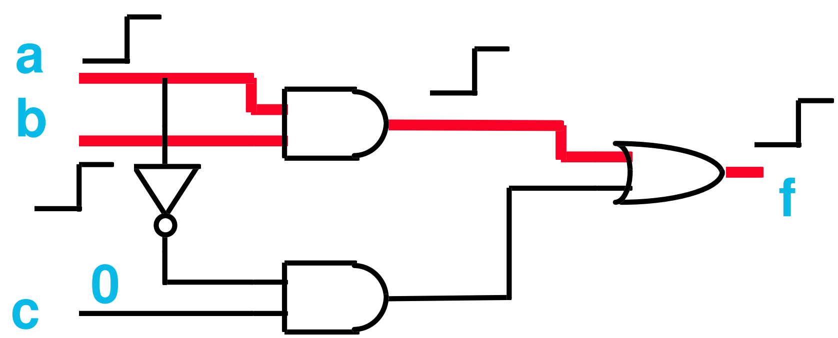
Need to implement each single-output "cone" as prime and irredundant circuit for full multifault testability

Path Delay Faults

Need to propagate transition down the path that is to be tested

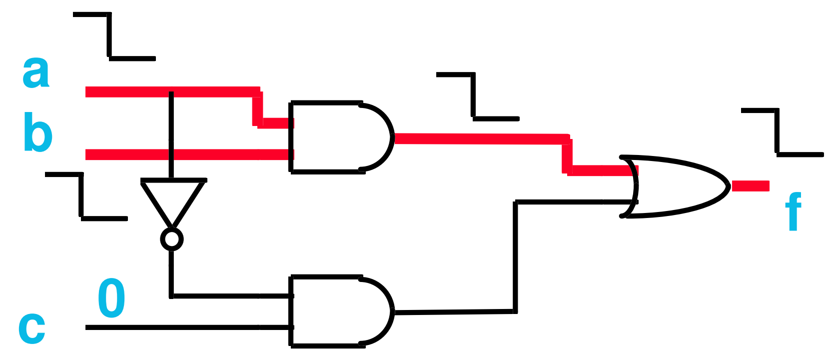


path from b to f is tested

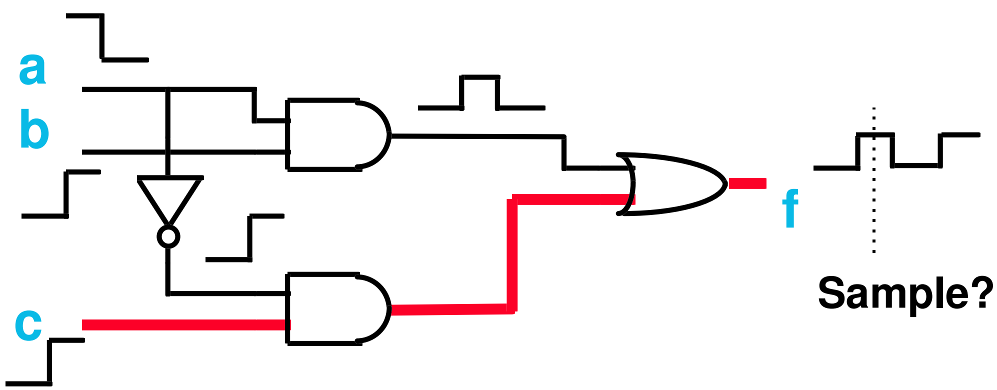


paths from a, b to f are tested

Robust Testing



paths from a, b race to set f to 0

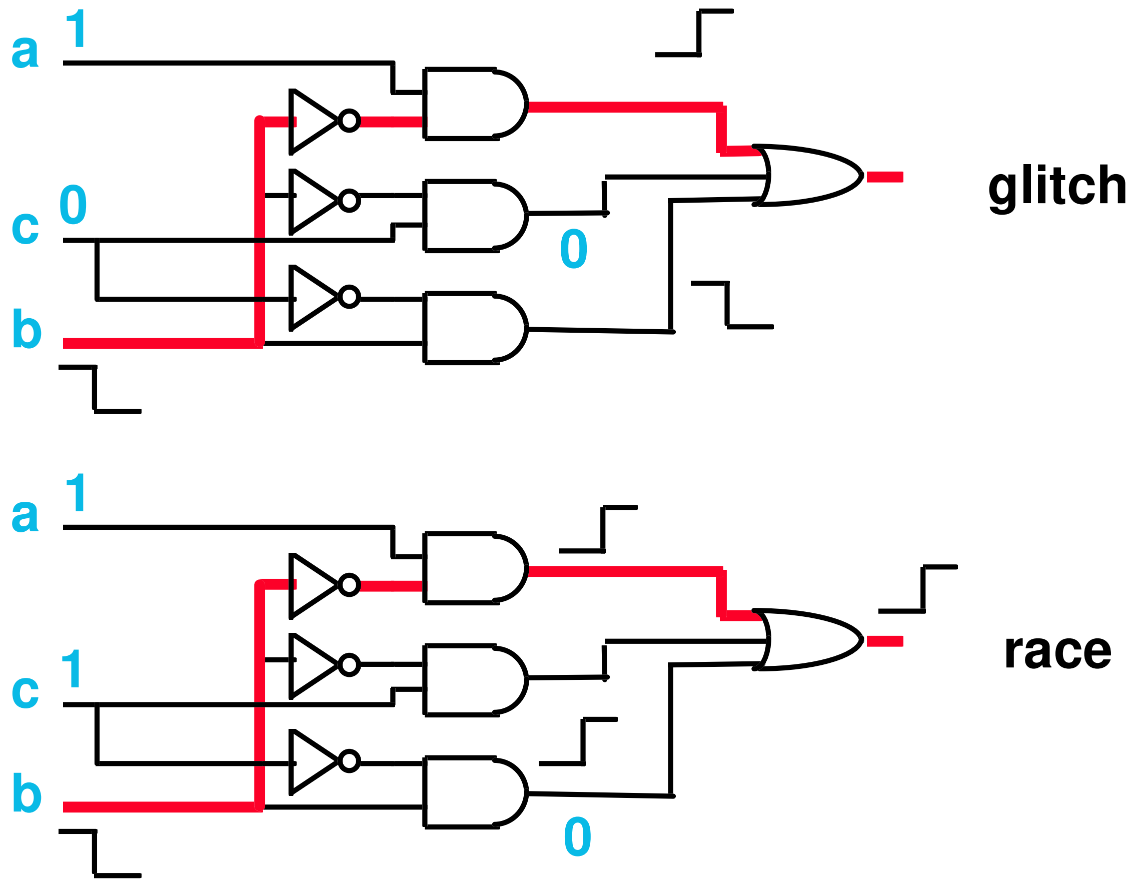


output glitches before transition from c propagate to f

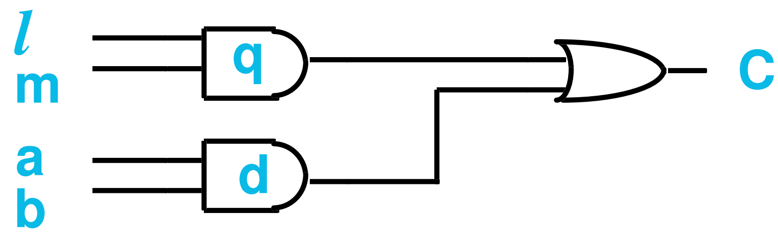
Have to avoid races and hazards ⇒ robust testing

Path Delay Fault Testability

Not all paths in a prime and irredundant two-level circuit are robustly testable



Definitions



A path Π in circuit C is associated with a literal l in cube **q**

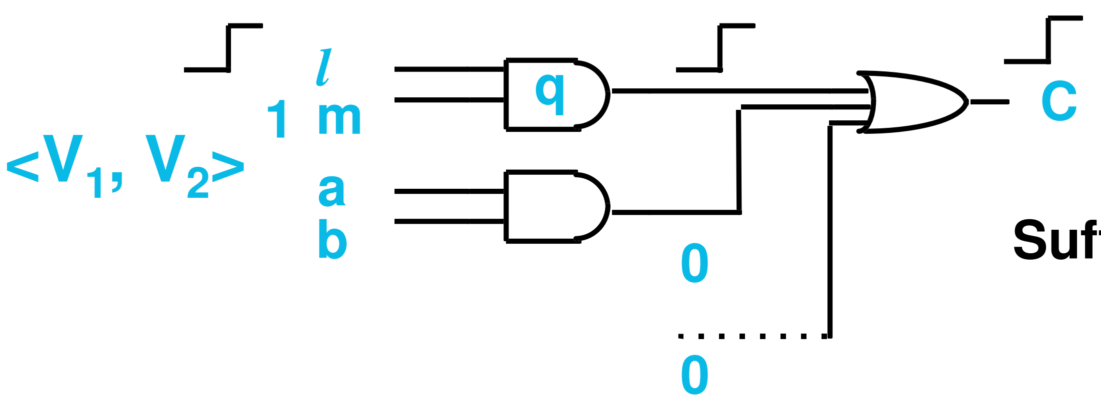
A relatively essential vertex of a cube **q** is a minterm that is not in any cube of **C** but is in **q**

$l\,m\bar{a}b$ is a relatively essential vertex of q above

Testability Conditions

Theorem: Let C be a single-output circuit. Let Π be a path in C that starts with l in cube q. There exists a hazard-free robust delay fault test for Π if and only if:

1) There exists a vertex V_2 that is a relatively essential vertex of q and
2) Vertex V_1 distance-1 from V_2 in l is in the OFF-set of C.

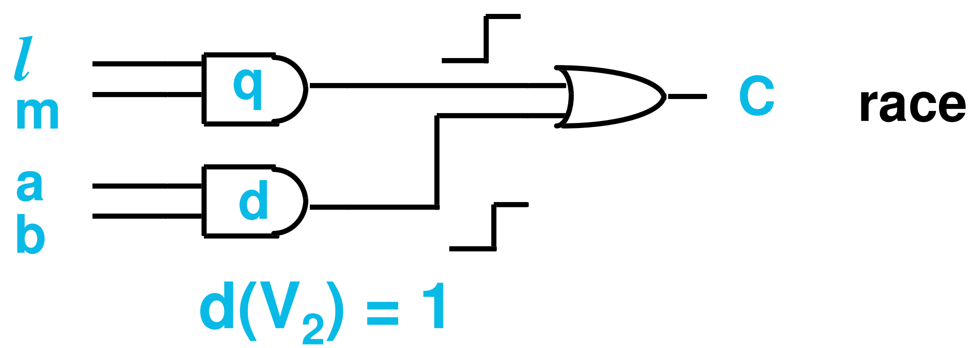


Sufficiency is trivial

Necessity

Suppose $<W_1, V_2>$ is a delay-fault test for Π.

Suppose V_2 is not a relatively essential vertex of q.



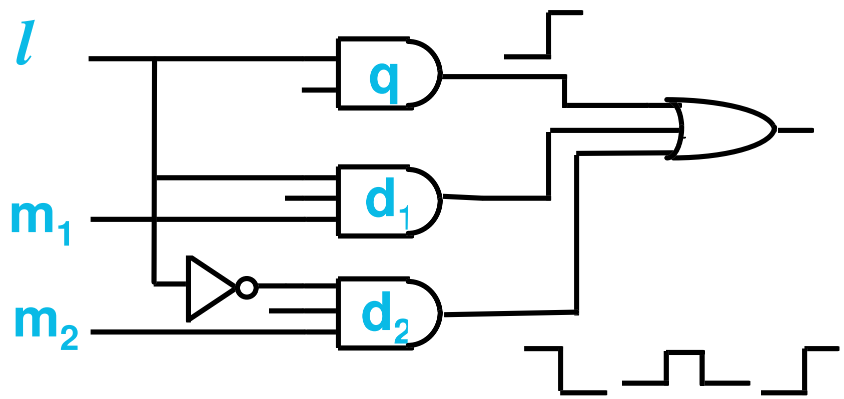
So V_2 has to be a relatively essential vertex of q.

Clearly W_1 has to be in the OFF-set of C.

But do W_1 and V_2 have to differ only in l ?

Necessity - 2

If W_1 and V_2 are not distance-1 in l we can construct a V_1 and V_2 distance-1 in l that are a delay fault test for Π.

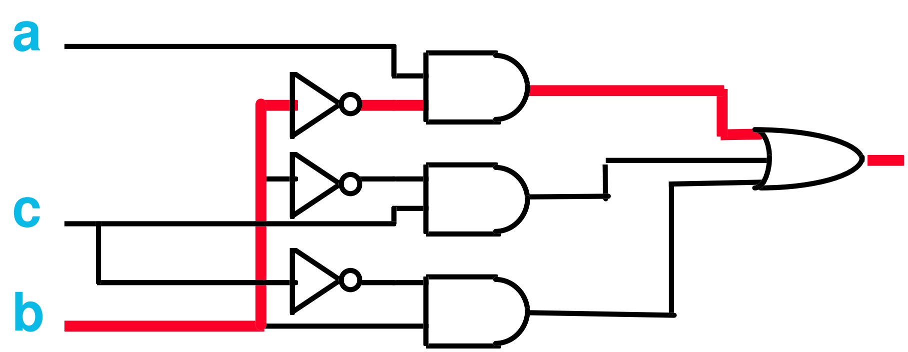


not allowed

Need some literal $m_i \in d_i$ such that $m_i = 0$ for <u>both</u> W_1 and V_2. Else glitch would invalidate test.

Just arbitrarily set remaining literals in W_1 other than l to values in V_2.

Example - I



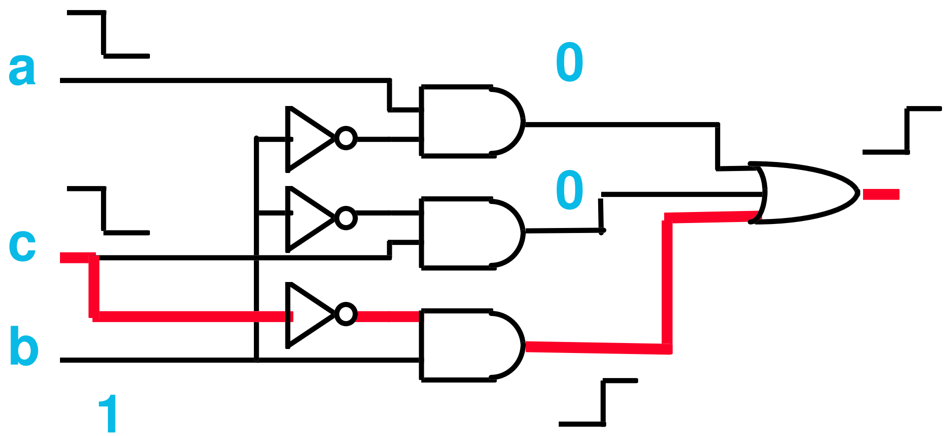
a	b	c
1	0	-
-	0	1
-	1	0

Relatively essential vertex of 10- is 100

But 110 (distance-1 from 100 in b) is in the ON-set

Therefore, there is no robust path delay fault test

Example - II



a	b	c
1	0	-
-	0	1
-	1	0

<111, 010> is a robust path delay fault test

Can construct distance-1 test by setting literals in V_1 other than c to values in V_2

Obtain <011, 010>, which is also a robust test