Circuits : → synchronous → 1) change of clk 2) memory elements → ff;

asynchronous → 1) change in. var. 2) unclocked ff's , time delay.

Stable : when out. is the same as in. , feedback → seq. circuit.

in: secondary var. present state y → □ → out. : excitation var. next state. Y

Fundemental mode → only one in. can bechanged at a time.

$Y_1 = xy_1 + x'y_2$ $Y_2 = xy_1' + x'y_2$

y_1y_2 \ x	0	1
00	0	0
01	1	0
11	1	1
10	0	1

y_1y_2 \ x	0	1
00	0	1
01	1	1
11	1	0
10	0	0

y_1y_2 \ x	0	1
00	00	01
01	11	01
11	11	10
10	00	10

transition table

- Flow table: letters or symbols

	0	1
a	a	b
b	c	b
c	c	d
d	a	d

→ primitive flow table → it has only one steady state table in each row.

- race → when we have 2 or more state var.

types: 1 non critical race: final state that circuit reaches doesn't depend on the order in which the state var. change.

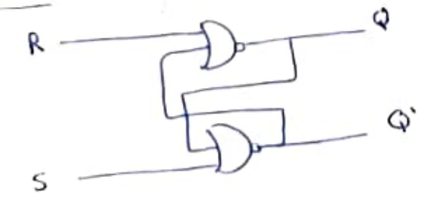
00 → 11 , 00 → 10 → 11 , 00 → 01 → 11

2 critical race : 00 → 01 → 11 , 00 → 11 , 00 → 10 ↳ stable state

final stable state depends on the order in which state var. change.

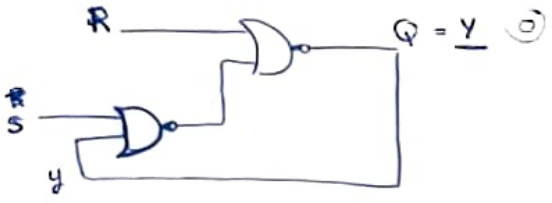
3 cycle : goes through a seq. of unstable states.

- SR Latch:



	S	R	Q	Q'
set ←	1	0	1	0
no change ←	0	0	1	0
reset ←	0	1	0	1
no change ←	0	0	0	1
set ←	1	0	1	0
invalid ?? ←	1	1	0	0 ?

$$\boxed{Y = SR' + R'y}$$



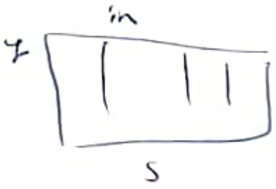
y \ SR	00	01	11	10
0	0	0	0	1
1	1	0	0	1

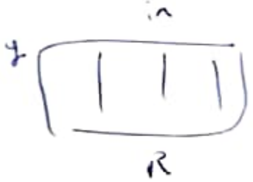
$$\boxed{Y = yR' + SR'}$$

$$\boxed{Y = yR' + S \quad \text{when } SR=0}$$

y	Y	S	R
0	0	0	X
0	1	1	0
1	1	X	0
1	0	0	1

excitation table





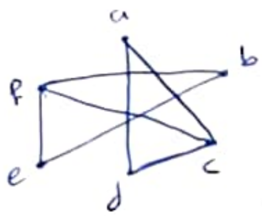
	D	G	Q	after b, e
a	0	0	0	~~a~~, ~~k~~
b	0	1	0 ✓	~~d~~, f,
c	0	0	1	α
d	0	1	1	
e	1	0	0	
f	1	0	1	α
g	1	1	0	
h	1	1	1	

	a	b	c	d	e
b	X				
c	✓	b,c X			
d	✓	a,b X	✓		
e	a,b d,c X	✓	X	X	
f	d,c X	✓	✓	X	✓
	a	b	c	d	e ~~f~~

(c,~~a~~), (~~d~~,a) (~~f~~,d) (c,~~f~~)
(c/b), (~~f~~,~~b~~) (e,~~f~~) ✓

	D	G	Q	notes
a	0	0	0	~~c~~, ~~d~~
b	0	0	1	~~e~~
c	0	1	0	~~a~~, ~~b~~, ~~f~~
d	1	0	0	~~a~~, e
e	1	0	1	~~b~~, ~~f~~
f	1	1	1	c, d, e

	00	01	11	10
a	ⓐ, 0	c, -	-, -	d, -
b	ⓑ, 1	c, -	-, -	e, -
c	a, -	ⓒ, 0	f, -	-, -
d	a, -	-, -	f, -	ⓓ, 0
e	b, -	-, -	f, -	ⓔ, 1
f	-, -	c, -	ⓕ, 1	f, -
	✓	✓	✓	



A (a, d, c)
B (f, b, e)

Testing: after fabrication:

$\boxed{1}$ Functional testing:

ex. CPU 1GHz, 32 inputs → # test vectors $= 2^{32} = 4 \times 10^9$ vectors

time $= \frac{4 \times 10^9}{1 \times 10^9} = \underline{4 \text{ sec.}}$

64 input → # test vectors $= 2^{64} = 2 \times 10^{19}$ vectors

time $= \frac{2 \times 10^{19}}{1 \times 10^9} =$ 585 years.

$\boxed{2}$ structural testing:

ex. adder:

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

→ if we do it by $\boxed{1}$ → 4 vectors

→ $\boxed{2}$ → 2 vectors

① 01	② 00	③ 00	④ 10
10	01	11	11

① you can test, S → stuck at 0, C stuck at 1 50%

② S → stuck at 1, C → stuck at 1, S → stuck at 0 75%

③ S → sa1, C → sa1, C → sa0 75%

④ S → sa1, C → sa1, C → sa0, S → sa0 ✓✓ the best 100%

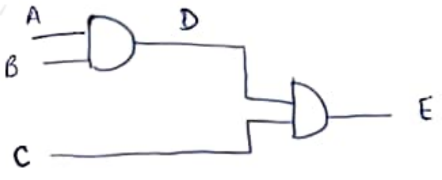
* Fault modeling: ① node stuck at 0 ② node stuck at 1

* Path sensitization method (2-value logic): assume each node in the circuit has either (sa1, sa0)

1) Backtrace phase: send the non-fault value under consideration.

2) propagation phase: steer the value of the node under considerations to one of the outputs.

input → control
output → observe.



. <u>test node D:</u>

① assume D <u>sa1</u>

① Backtrace phase ; send 0 to D ⇒ AB = 0 ⇒ $\overrightarrow{AB} = \begin{matrix}01\\00\\10\end{matrix}$

② propagation phase : C = 1 → E = 0 non-faulty.

E = 1 <u>sa1</u> (Fault)

$\overrightarrow{ABC}$
0 0 1

$\overrightarrow{ABC}$
1 0 1

$\overrightarrow{ABC}$
0 1 1

3- TV

② assume D <u>sa0</u>

☐1 send 1 to D : AB = 1 ⇒ $\overrightarrow{AB}$ = 11

☐2 C = 1 → E = 1 non-faulty

E = 0 (fault) <u>sa0</u>

$\overrightarrow{ABC}$
1 1 1

1 - TV

. test <u>A sa0</u>

☐1 send 1 to A

☐2 B = 1 ⇒ D = A = 1 non-faulty

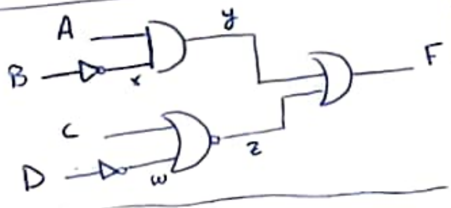
D = 0 fault (sa0)

$\overrightarrow{ABC}$
1 1 1

⇒ C = 1 → E = 1 non-faulty

E = 0 fault (sa0)

<u>ex.</u>



<u>• y sa0</u>

☐1 send 1 to y ⇒ Ax = 1 ⇒ $\overrightarrow{AB}$ = 10

☐2 z must be 0

F = 1 → non-faulty

F = 0 → fault

C w = 1

C	w
1	1
1	0
0	1

C	D
1	0
1	1
0	0

test vectors

A	B	C	D
1	0	1	0
1	0	1	1
1	0	0	0

• D algorithm (5- value logic)

<u>1) logic 1</u>

<u>2) logic 0</u>

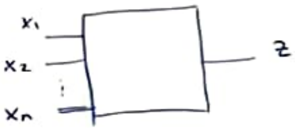
3) <u>D : 1 (no-fault), 0 (faulty)</u>

4) <u>D' = (0 = no-fault)</u>

<u>1 (fault)</u>

5) <u>X : unknown</u>

- Boolean difference:



$$z = f(x_1, x_2, x_3, \dots, x_n)$$

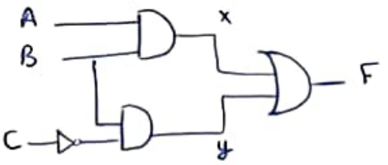
$$\frac{df}{di} = f(x_i == 0) \oplus f(x_i == 1)$$

[1] z is sensitive to node x_i, $\frac{df}{dx_i} = 1$

[2] to test x_i sa0 $\Rightarrow$ $x_i \frac{dz}{dx_i} \overset{??}{=} 1$

[3] to test x_i sa1 $\Rightarrow$ $(x_i)' \frac{dz}{dx_i} \overset{??}{=} 1$

Ex.



- test node A

$$F(A,B,C) = AB + BC'$$

$$\frac{dF}{dA} = F(A=0) \oplus F(A=1)$$

$$= (BC') \oplus (B + BC')$$

$$= (BC') \oplus B$$

$$= BC'B' + (B'+C)B$$

$$= 0 + CB$$

$$= \boxed{BC}$$

- [A] F is sensitive to A when $BC = 1 \rightarrow \overrightarrow{BC} = 11$
- [B] A sa0 when; $ABC = 1 \Rightarrow \overrightarrow{ABC} = 111$
- [C] A sa1 when; $A'BC = 1 \Rightarrow \overrightarrow{ABC} = 011$

- to test node B:

$$\frac{dF}{dB} = 0 \oplus (A+C') = A+C'$$

- F is sensitive to B when $A+C' = 1$

```
1 0          AC
0 1   ->     1 1
1 1          0 0
             1 0
```

- B sa1 $\rightarrow$ $B'(A+C') = 1$ ↳

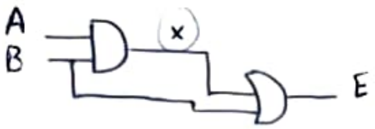
```
ABC
0 0 0
1 0 1
1 0 0
```

- B sa0 $\rightarrow$ $B(A+C') = 1$ ↳

```
ABC
1 1 1
0 1 0
1 ϕ 0
```

untestable faults: partially utestable / completely untestable.

ex. in booolean.



$E = AB + B = (A+1)B = \boxed{B}$

[1] $\frac{dE}{dx} = E(x=0) \oplus E(x=1)$

$= B \oplus 1 = \boxed{B'}$

$E = x + B$

• E is sensitive to x when $B' = 1 \rightarrow \boxed{B=0}$

[2] x sa0 → $(x^{1})\, B' = 1$

$(1 \cdot 0) = 1$ → B = 0, x = 1

must be ~~zero~~ one

$x = AB$ → zero

contradiction → x is untestable

$(A.B)' \cdot B' = ?1$

$A' + B' \cdot B'$

$A'B' + B'B'$

$(A'+1)B'$

$\boxed{B' = 1}$

→ $\boxed{B=0}$, A is don't care → either 1 or zero

[3] x sa 1 → $x' . B' = 1$

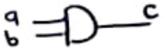
$\boxed{B=0}$

$\boxed{x=0}$

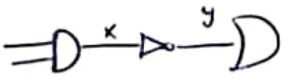
$x = AB$

don't care ← $\boxed{A=0,\ B=0}$ ✓ → partially testable.

• Fault collapsing: 1) fault equivalance: 2 or more faults s said to be eq. if they have the same test vectors.

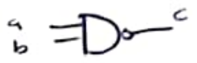


a sa0 : $\boxed{\begin{array}{cc} a & b \\ 1 & 1 \end{array}}$ ≡ b sa0 $\boxed{\begin{array}{cc} a & b \\ 1 & 1 \end{array}}$ ≡ c sa0 $\boxed{\begin{array}{cc} a & b \\ 1 & 1 \end{array}}$



x sa0 ≡ y sa1

y sa0 ≡ x sa1



a sa0 ≡ b sa0 ≡ c sa1

Fault dominance : fault f_1 is said to dominate fault f_2 if the test vectors of f_2 are subset of test vectors of f_1

f_1

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

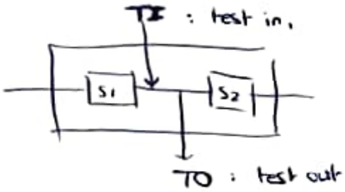
f_2

a	b	c
0	0	0
1	1	1

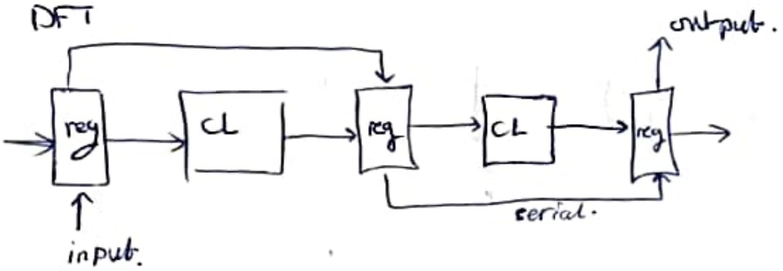
f_1 dominates f_2 → for the minimum test vectors → we remove f_1

- Design for testability (DFT) :

1) ADHOC DFT : minor changes on the circuit, for controlability & observability

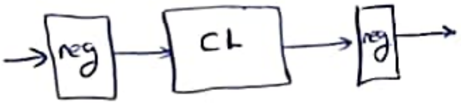


2) structured DFT



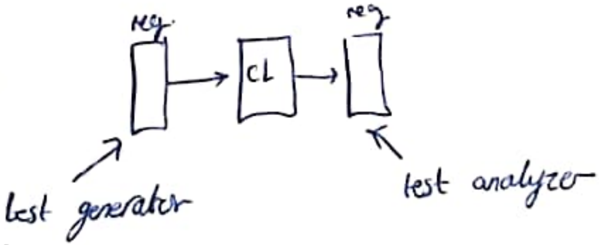
reg → for scan design (shifting. → reg. either scan reg. or shift reg. to insert data) in scan mode. (serially)

3) Built-in self test (BIST) : auto. do the test



in normal mode → work in parallel. (registers.)

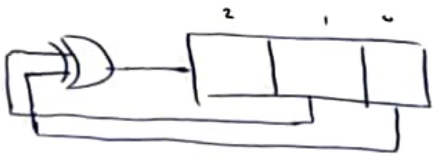
in test mode



L we don't use as a counter, we use linear feedback shift reg. (LFSR)

1. LFSR :

• 3-bit LFSR :

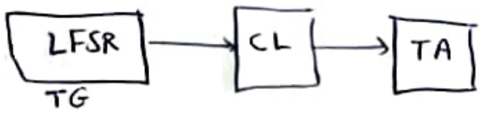


1	1	1
0	1	1
0	0	1
1	0	D
0	1	0
1	0	1
1	1	0
1	1	1

تکرار

$x^3 + x^2 + 1$ → prime polynomial.

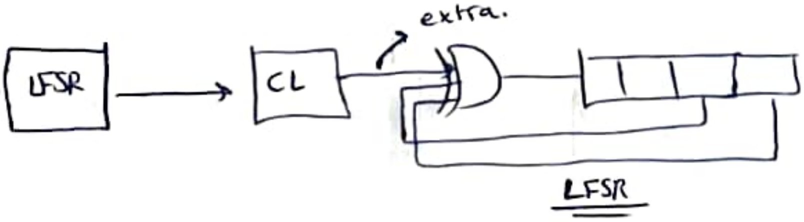
comp $\boxed{x^3 + x + 1}$



• TA : 1) single signature input reg. (SISR)

2) multiple input signature reg. (MISR)

• SISR :



LFSR with extra input to the LFSR coming from the circuit under Test (CUT)