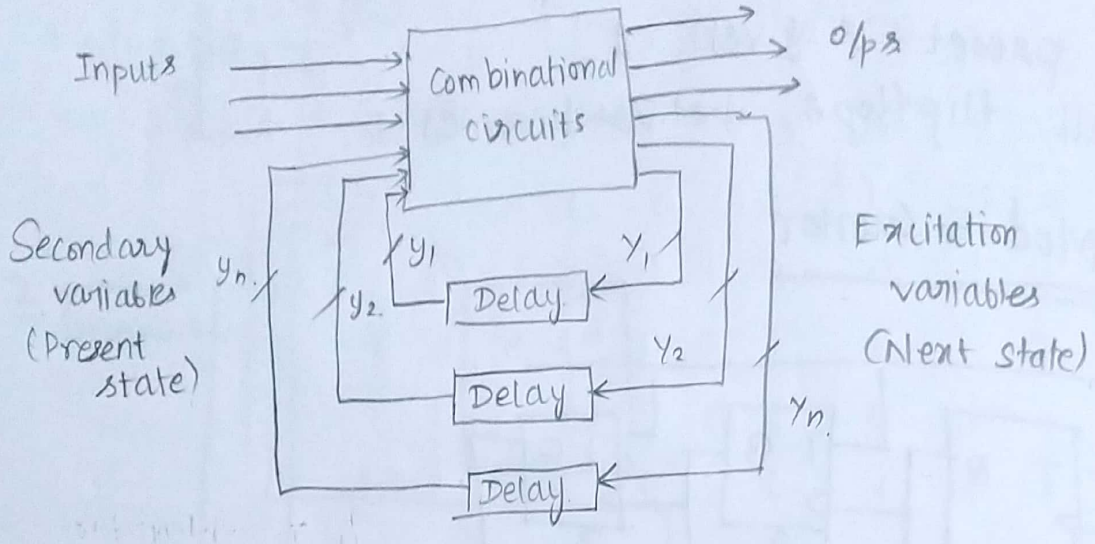
4) Asynchronous Sequential Circuits



No clock signal is applied

The o/p is obtained after a delay in the ~~outp~~ input

Operated in Two mode

1). Fundamental mode of operation

2) Pulse mode of operation

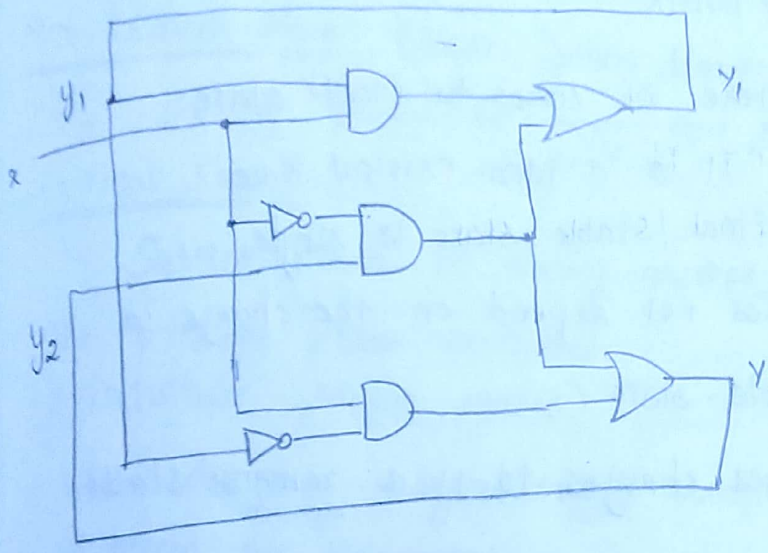
[At a time any one of the input variable may change. The next change is not allowed until the change made goes to a steady state).

$Y_i = y_i \longrightarrow$ Steady state condition.

After steady state, the input may be changed.

Time change between two input change should be greater than the time taken to have the steady state.

Stable state
Unstable state
Transistion table
Flow table
Race — Non-critical race
Race — Critical race.
Hazards — Static
Hazards — Dynamic
Hazards — Essential.



Y_1, Y_2 - Next states
y_1, y_2 - Present states.

$Y_1 = x y_1 + \bar{x} y_2$

$Y_2 = \bar{x} y_2 + x \bar{y_1}$

Transistion Table:

→ input is at top.

y_1, y_2 - two present state variables

Y_1 / $y_1 y_2$	x = 0	x = 1
00	0	0
01	1	0
11	1	1
10	0	1

①

Y_2 / $y_1 y_2$	x = 0	x = 1
00	0	1
01	1	1
11	1	0
10	0	0

②

By using transistion table, find Y_1, Y_2

①, ② ⇒ Transistion tables.

$Y_1 Y_2$ / $y_1 y_2$	x = 0	x = 1
00	(00)	01
01	11	(01)
11	(11)	10
10	00	(10)

③

③ → Flow table
(By combining the transistion tables ① & ②)

↳ Circle if P.S = N.S i.e stable state
$Y = y$

y_1y_2 \ x	0	1
a	(a)	b
b	d	(b)
d	(d)	c
c	a	(c)

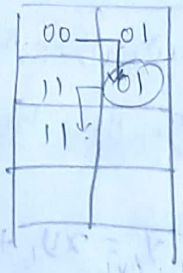
Flow table

(But can have multiple stable states)
Unstable

When $x \to 0 \to 1$

y_1y_2 changes from $00 \to 01 \to 01$

i.e



Here 01 comes to stable state.

It is a Non-critical Race.
(Final stable state is single)

The final state does not depend on the change in the present variable state.

Here $01 \to$ either changes to 11 & remains stable

Critical Race:

Final state depends on the change in the present variable state

Cycle

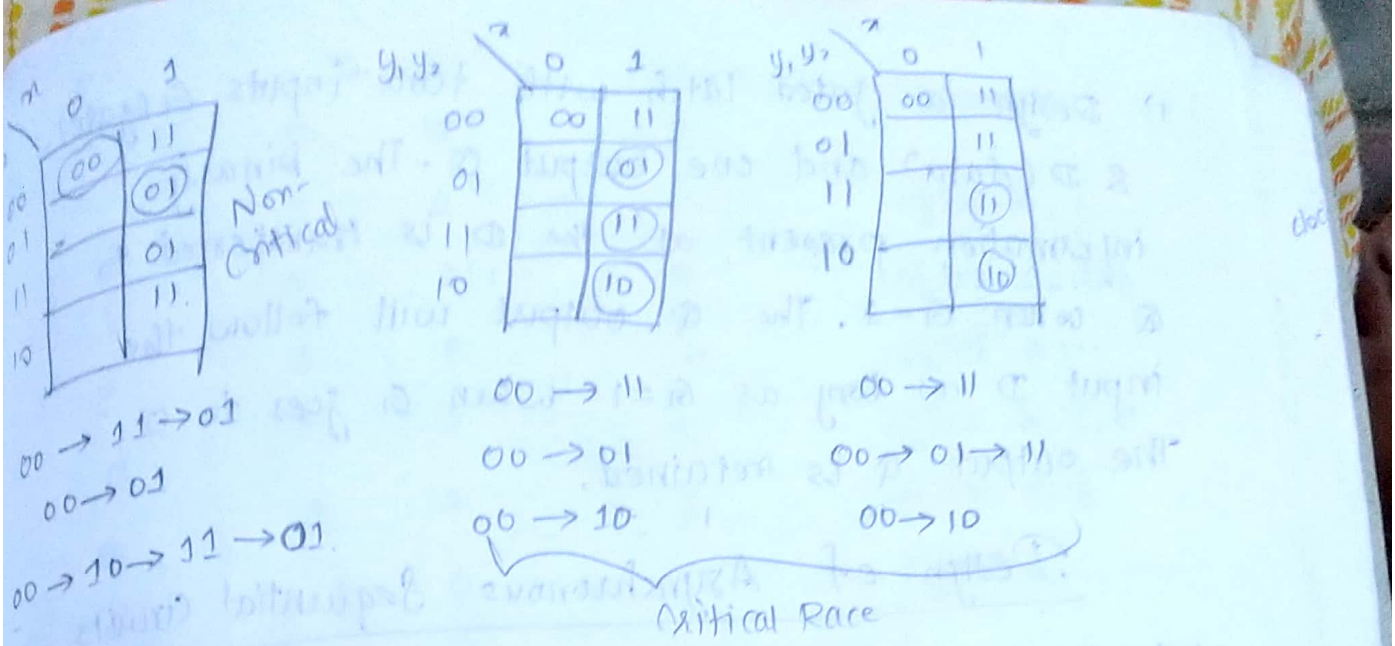
$00 \to 01 \to 11 \ldots$ does not settle down in a stable state.

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

$00 \to 11$
$00 \to 01 \to 11$
$00 \to 10 \to 11$

} Any one of these possibilities occurs depending on the delay (feedback)

Non-Critical Race



Non-critical Race: (Best Option than critical race)

Settles down to only one state.

Critical Race:

Settles down in many states depending on the present state variables.
ie (Multiple stable states)

∴ While designing asynchronous circuit, avoid critical race.
& Avoid the Hazards)

Primitive flow table:

In flow table, only single stable state
But in primitive flow table multiple stable states.
∴ Reduce primitive flow table to flow table
which is called as state Reduction.

State Reduction is done with the help of
Implication table → Merger diagram. . . .

1) Design a gated latch with two inputs G (gate) & D (data) and one output Q. The binary information present at the D is transferred to Q when G = 1. The Q output will follow the input D as long as G = 1 when G goes to 0 the output Q is retained.

Design of Asynchronous Sequential Circuits

When G = 1 Q = D

↓

0 → Q = No change. → because previous o/p is known.

State	input D, G		Output Q	Comments
	D	G	(assume)	
a	0	0	0	after b, c.
b	0	1	0	after a, ~~d state~~
c	1	0	0	after a.
	0	0	0	already available
d	1	1	1	after b, c.
	0	0	0	

→ at a time only one present state is allowed to change

c is from a (not from b)

e 1 1 1

1 0 1

It is changed from b

0 0 11

1 0

00 11

Again

11

01 10

State	Input D	Input G	Output Q	Comments
a	0	0	0	
b	0	1	0	after b, c
c	1	0	0	after a, d, f
d	1	1	0	after a
e	1	0	1	after b, c, e
f	0	0	1	after d, f
			1	after e

From every state two posibilities a,a b,b c,c ..... f,f

Primitive flow table:

no need to mention.

	DG 00	01	11	10
a	(a),0	b,(-)	-,-	c,-
b	a,-	(b),0	d,-	-,-
c	a,-	-,-	d,-	(c),0
d	-,-	b,-	(d),1	e,-
e	f,-	-,-	d,-	(e),1
f	(f),1	b,-	-,-	e,-

① Plot the stable states.

(In primitive flow table only one stable state in each row)

From 0,0 → 1,1 is not possible (in I^st row)

(-,- represents don't cares in asynchronous seq ckts)

These 3 are equivalent

Flow table

	DG 00	01	11	10
a, b, c	(a),0	(b),0	d,-	(c),0
d, e, f	(f),1	b,-	(d),1	(e),1

represent with d

Check each column

If any stable state write the stable state.

or represent unstable state.

New Flow Table

	DG 00	01	11	10
a	(a), 0	(a), 0	d, -	(a), 0
d	(d), 1	a, -	(d), 1	(d), 1

b is replaced with a

Assign binary values to a as in Synchronous sequential circuits.

Y; Z ⟶ output

y \ DG	00	01	11	10
0	0, 0	0, 0	1, -	0, 0
1	1, 1	0, -	1, 1	1, 1

Input Y

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

$Y = DG + y\bar{G}$

Output Z

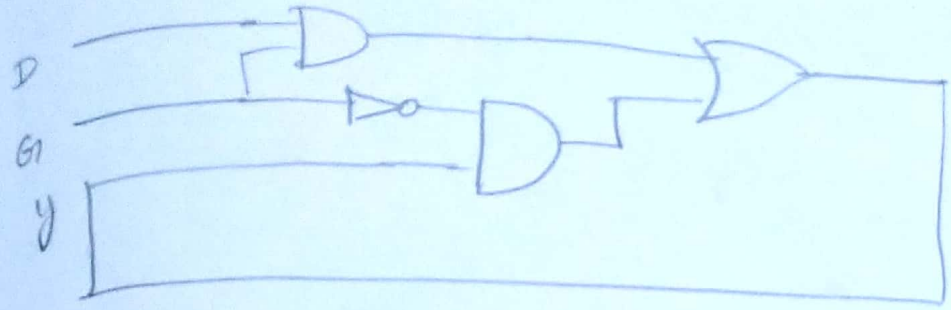
y \ DG	00	01	11	10
0	0	0	-	0
1	1	-	1	1

Assign ~~to~~ o/p values to unstable states "-"

z, y \ DG	00	01	11	10
0	0	0	1	0
1	1	0	1	1

$Z = DG + y\bar{G}$

$\therefore Y = DG + y\bar{G} = Z$



$Y = z$

$DG + y\bar{G}$

y - present state, Y = Next state.

When $G = 1$ $\quad Q = D$

$\downarrow$

0 $\quad Q =$ no change.

State	Input D	Input G	Output Q	Comments
a	1	1	1	after b, c.
b	0	1	0	after a., d.
c.	1	0	1	after a, d.
d	0	0.	0	after b, c.