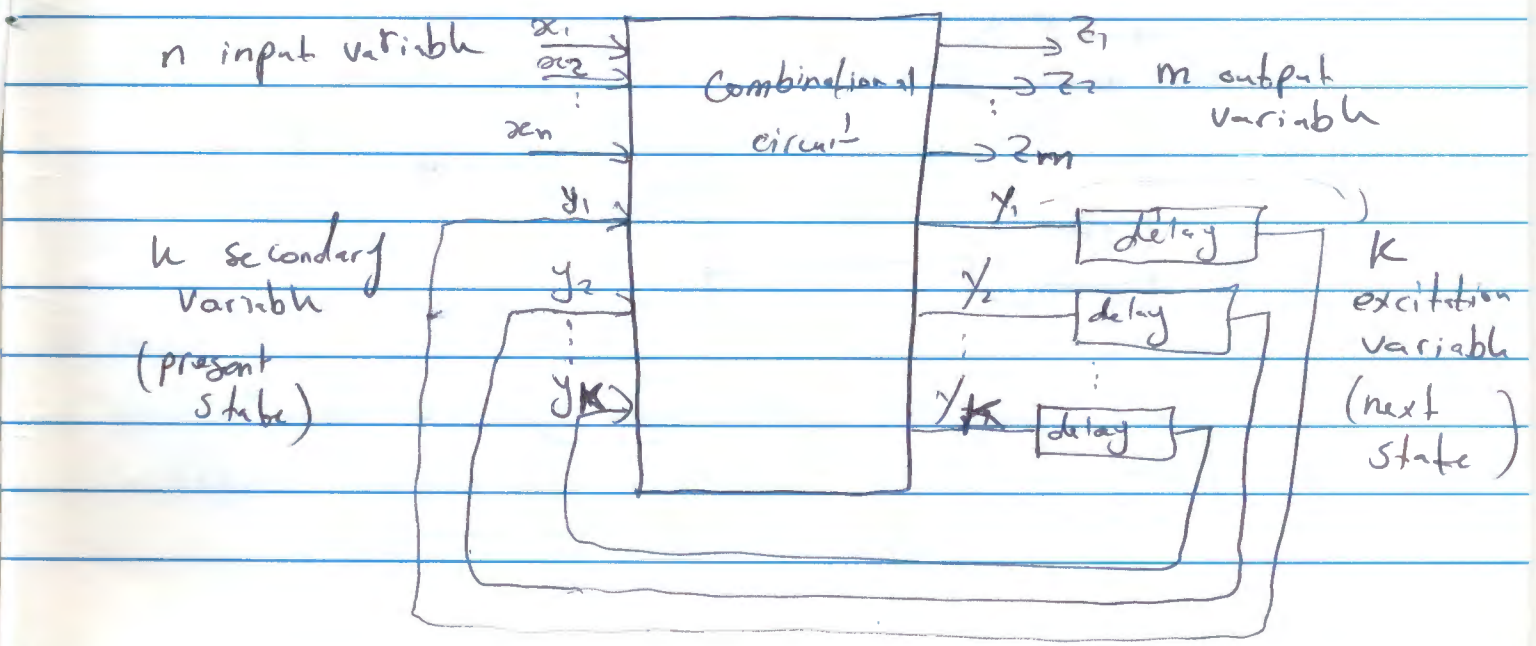
⑧ Asynchronous Sequential Logic

- In synchronous sequential circuits, the change of internal state occurs in response to the synchronized clock pulses. Also the memory elements in synchronous sequential circuits are clocked flip-flops

- In asynchronous sequential circuits, The change of internal state occurs when there is a change in the input variable because ~~Asynchronous sequential cir~~ these circuits don't use clock pulses. Also the memory elements are either unclocked flip-flops (latches) or time delay element.

- One of the applications of asynchronous circuits is when the speed of operation is important such that the digital system must respond quickly without having to wait for a clock pulse.

⑧ The following is a general block diagram of Asynchronous sequential circuit



- When an input variable changes in value, the y secondary variables do not change instantaneously.

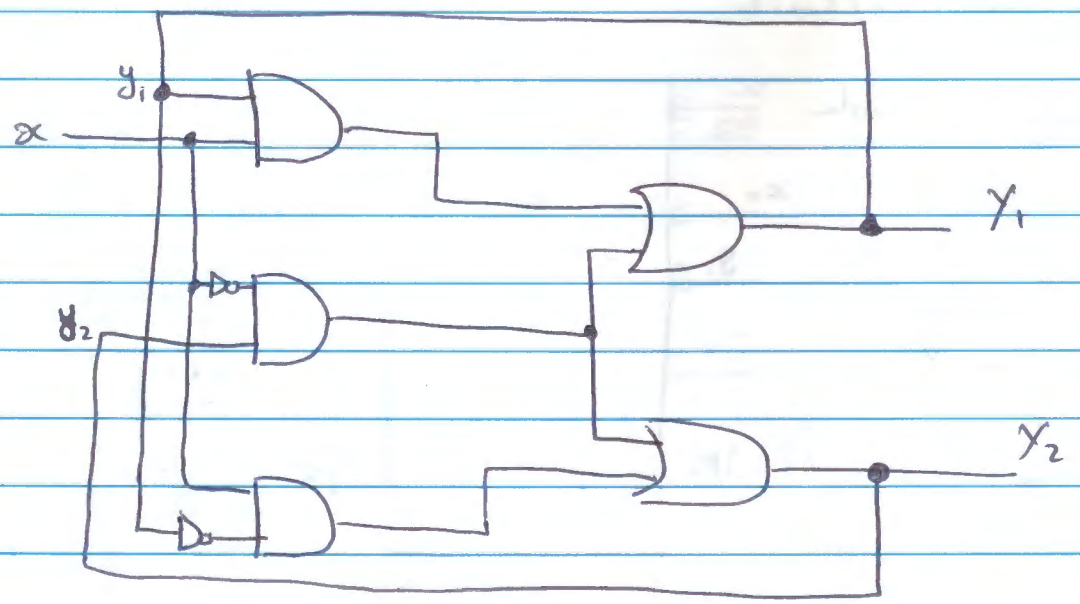
- In the steady-state condition, Y's and y's are the same, but during transition they are not. The system is said to be stable when it reaches the steady state condition where $y_i = Y_i$

- For proper operation, the circuit must reach stable state before the input is changed to a new value.

- fundamental-mode operation assumes that the input signals change one at a time and only when the circuit is in a stable condition.

⊛ Analysis of circuits with feedback

Ex.



(*) The first step is to get the excitation variables as a function of the input x and secondary variable $\Rightarrow$

(*) $Y_1 = x y_1 + x' y_2$

$Y_2 = x y_1' + x' y_2$

(*) The next step in the analysis is to plot Y_1 and Y_2 functions in a map

$y_1 y_2$ \ x	0	1
00	0	0
01	1	0
11	1	1
10	0	1

map for $Y_1 = x y_1 + x' y_2$

$y_1 y_2$ \ x	0	1
00	0	1
01	1	1
11	1	0
10	0	0

map for $Y_2 = x y_1' + x' y_2$

(*) The next step is to get the transition table. The transition table shows the value of $Y = Y_1 Y_2$ inside each square

$y_1 y_2$ \ x	0	1
00	(00)	01
01	11	(01)
11	(11)	10
10	00	(10)

$y = y_1 y_2$; $Y = Y_1 Y_2$

(*) Then, the entries in the transition table where $Y = y$ are circled to indicate stable condition. Uncircled entries represent unstable state

* The transition table can be presented in the form of state table in synchronous circuits

Present state	next state $x=0$	next state $x=1$
0 0	0 0	0 1
0 1	1 1	0 1
1 0	0 0	1 0
1 1	1 1	1 0

- note that each row contains a next state which is similar to present state, otherwise the circuit will be unstable for that state

- total state is a combination of internal states and the input x $\Rightarrow$

the circuit above has 4 stable total states

$y_1 y_2 x = 000, 011, 110, 101$

Analysis with flow table

- A flow table is similar to a transition table except that the internal states are symbolized with letters rather than binary numbers.

	x = 0	x = 1
a	(a)	b
b	c	(b)
c	(c)	d
d	a	(d)

flow table
4 states
1 input
no output

Ex:

	x_1x_2 = 00	01	11	10
a	(a), 0	(a), 0	(a), 0	b, 0
b	a, 0	a, 0	(b), 1	(b), 0

This is a flow table for circuit with 2 states, 2 inputs and one output

primitive flow table: is the flow table which has only one stable state in each row

- If $x_1 = 0 \Rightarrow$ circuit is in state a (output = 0)
- if $x_1 = 1 \quad x_2 = 0 \Rightarrow$ circuit is in state b (output = 0)
- if $x_1 = 1 \quad x_2 = 1 \Rightarrow$ circuit may be in state a (output = 0) or in state b (output = 1)

* If the input changes from 01 to 11, then the circuit will stay in state a.

if the input changes from 10 to 11, then the circuit will stay in state b.

(as we assume that in fundamental mode only one input can change).

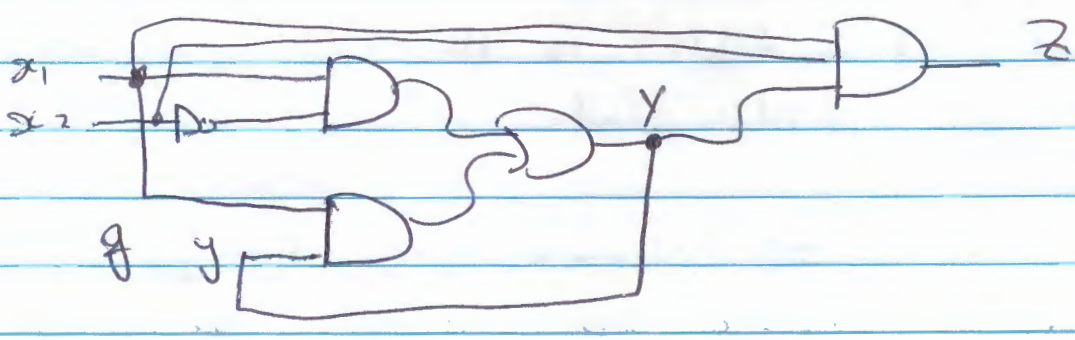
* In order to obtain the circuit described by a flow table, then we must assign the states a binary value. This will convert the flow table to transition table

y \ x_1x_2	00	01	11	10
0	(0)	(0)	(0)	1
1	0	0	(1)	(1)

$Y = x_1x_2' + x_1y$

y \ x_1x_2	00	01	11	10
0	0	0	0	0
1	0	0	1	0

$Z = x_1x_2y$



⊛ Race Conditions

- A race condition is said to exist in an asynchronous sequential circuit when two or more binary state variables change value in response to a change in an input variable.

- for example, if state variable must change from 00 to 11 when changing the input ⟹:

  00 - 11 equal delay

  00 - 01 - 11 second variable is faster

  00 - 10 - 11 1st " " " "

- If the final stable state that the circuit reaches doesn't depend on the order in which the state variable change ⟹ the race is called noncritical. otherwise, the race is critical.

- For proper operation, critical races must be avoided.

Ex:

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

Possible transitions:

00 → 11

00 → 01 → 11

00 → 10 → 11

⟹ total stable state $y_1y_2x = 111$ ⟹ non critical race.

Ex

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

possible transition

00 - 11 - 01

00 - 01

00 - 10 - 11 - 01

⇒ total stable state = y_1y_2x = 011

⇒ non-critical race

Ex.

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

00 - 11

00 - 01

00 - 10

⇒ 3 different stable states

⇒ critical race

Exc.

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

00 - 11

00 - 01 - 11

00 - 10

⟹ 2 different stable states

⟹ critical race

- Race may be avoided by making a proper binary assignment to state variables in such a way that only one state variable can change at any one time.

- Races can be avoided through the use of cycle of intermediate unstable states with a unique state variable change

Exc.

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

00 - 01 - 11 - 10

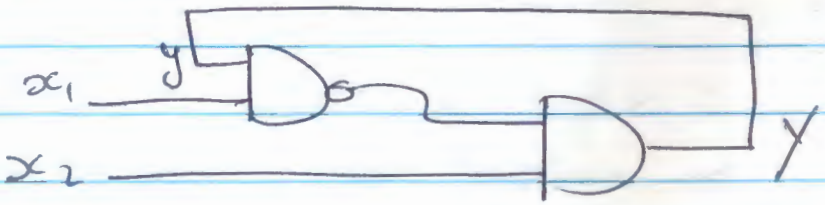
- If a cycle doesn't terminate with a stable state, then the circuit will keep going from one unstable state to another making the entire circuit unstable

Ex:

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

$00 \to 01 \to 11 \to 10 \to$ [ ]

Example of unstable asynchronous circuit



$$Y = (x_1 y)' x_2 = x_1' x_2 + x_2 y'$$

(Excitation function)

y \ x_1x_2	00	01	11	10
0	(0)	1	1	(0)
1	0	(1)	0	0

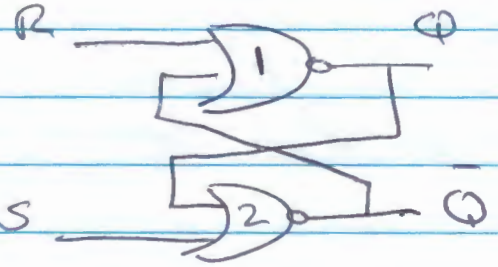
now if we fix x_1x_2 on 11 $\Rightarrow$ if $y=0$ then $Y=1$, then $y=1$, then $Y=0$

$\Rightarrow$ oscillation (not stable)

④ circuits with latches

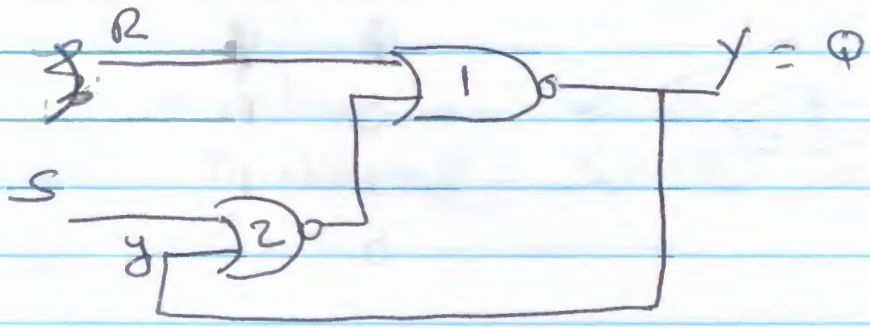
⑧ SR latch

- ~~using~~ SR using 2 nor gates



S	R	Q	$\bar{Q}$
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

undesirable



$$Y = \left((S+y)' + R\right)' = (S+y)R' = SR' + R'y$$

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

transition table

- In normal operation, we must make sure that S and R are not 1s simultaneously. This condition can be expressed by the Boolean function $SR = 0$

$\Rightarrow$

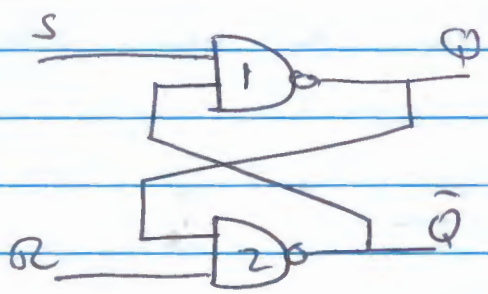
$$Y = SR' + R'Y$$

$$\Rightarrow Y = SR' + SR + R'Y$$

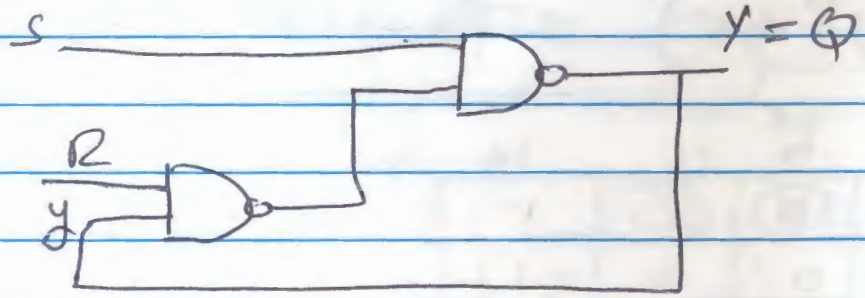
$$= S(R' + R) + R'Y$$

$$= S + R'Y \quad (\text{when } SR = 0)$$ next excitation function

SR latch using 2 NAND gates



S	R	Q	$\bar{Q}$
1	0	0	1
1	0	0	1
0	1	1	0
1	1	1	0
0	0	1	1



$$X = [(Ry)'S]' = Ry + S'$$

$(S'R')'$

y_0 \ SR	00	01	11	10
0	1	1	(0)	(0)
1	(1)	(1)	(1)	0

② Analysis Example (Fig 9-12)

$S_1 = x_1 y_2$

$S_2 = x_1 y_2$

$R_1 = x_1' x_2'$

$R_2 = x_2' y_1$

- let us check the condition $SR = 0$??

$S_1 R_1 = (x_1 y_2) \cdot (x_1' x_2') = 0$

$S_2 R_2 = (x_1 x_2) \cdot (x_2' y_1) = 0$

- because the conditions are valid, then we can use the function $Y = S + R'y$ to derive excitation functions

$$Y_1 = S_1 + R_1' y_1 = x_1 y_2 + (x_1 + x_2) y_1$$
$$= x_1 y_2 + x_1 y_1 + x_2 y_1$$

$$Y_2 = S_2 + R_2' y_2 = x_1 x_2 + x_2 y_2 + y_1' y_2$$

- now we can get the transition table

$y_1 y_2$ \ $x_1 x_2$	00	01	11	10
00	(00)	(00)	01	(00)
01	(01)	(01)	11	11
11	00	(11)	(11)	10
10	00	(10)	11	(10)

- this circuit has 1 critical race

Latch Excitation Table

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

Implementation Example

flow table

	00	01	11	10
a	(a)	(a)	(a)	b
b	a	a	(b)	(b)

($x_1 x_2$)

assign a=0 and b=1 ⇒ transition table

y	00	01	11	10
0	0	0	0	1
1	0	0	1	1

($x_1 x_2$)

$\Rightarrow Y = x_1 x_2' + x_1 y$

Map for S

y	00	01	11	10
0	0	0	0	1
1	0	0	X	X

($x_1 x_2$)

$S = x_1 x_2'$

Map for R

y	00	01	11	10
0	X	X	X	0
1	1	1	0	0

($x_1 x_2$)

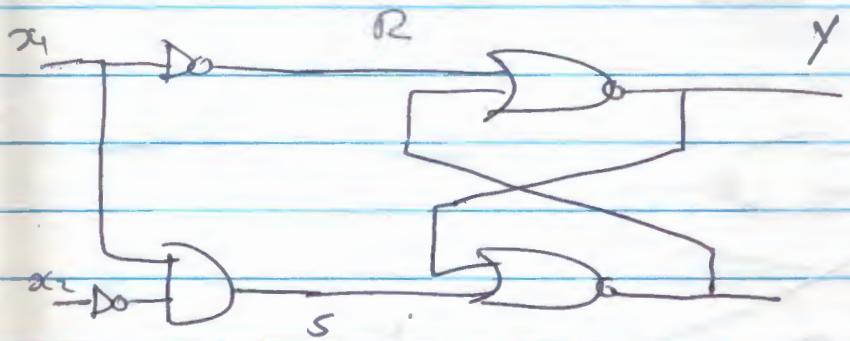
$R = x_1'$

or $Y = x_1 x_2' + x_1 y$

$Y = S + R' y$

$\Rightarrow S = x_1 x_2'$

$R = x_1'$



Debounce circuit

- Normally, the input signals of digital systems are controlled by the use of mechanical switches.
- mechanical switches vibrates many times before reaching their final states, This will cause to oscilate between 0 and 1.

- debounce circuit is one that removes the series of pulses that result from the use of the switch.