Computation Structures 2014 yuzly

S5
- ☐ moving to sequential logic.
- ☐ D-latch. memory device and dynamic discipline
- ☐ clocks and edge-triggered flip-flops

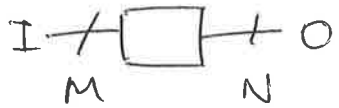
· Moving to sequential logic

- one can build an ALU using combinational logic, and any Boolean function could be done (discipline. | F(1,0) VIH VOL ···
- but there are functions that cannot be computed:

  summing up a sequence of numbers

  ∵ the sequence can be arbitrarily long ∴ no fixed input

- statemachines can.



t = [0, 1, 2, 3, 4 ···]

X = [5, 6, 1, 2, 3 ···]

Y = [5, 11, 12, 14, 17 ···]

how is a state machine implemented?

t = [0, 1, 2, 3, 4 ···]

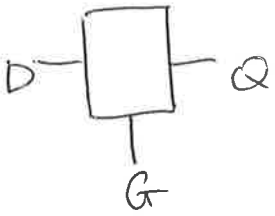
X = [5, 6, 1, 2, 3 ···]

S = [0 → 5 → 11 ···]

Y = [5 11 ···]

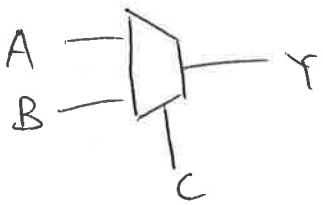
- Designing a state is the key to building a state mac.
- Two things to consider in a hardware
  - how to represent time series?
  - how to make a device remember its state?
- Talk about the latter first.

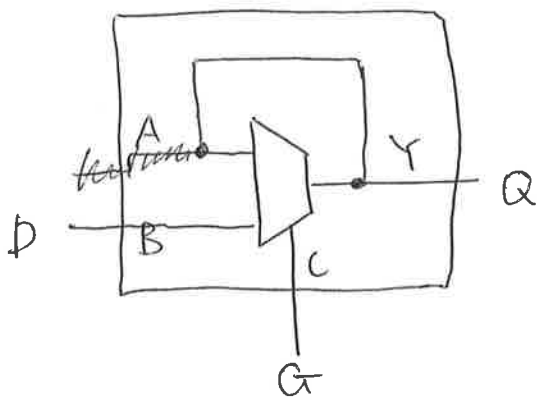
• D-latch: memory device and dynamic discipline.



G=0 : Select previous output (remember it)
G=1 : select current D.

- Data select (S3)

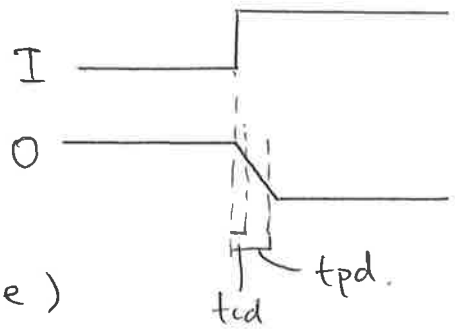




- Consider time constraints.

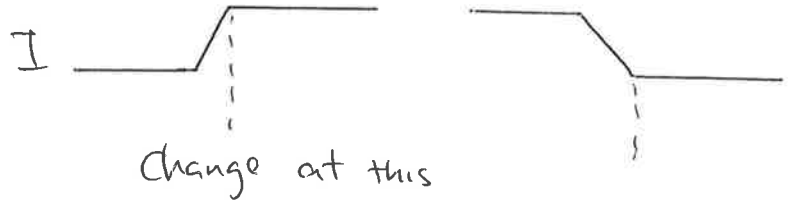
drawing wave forms

- we learned

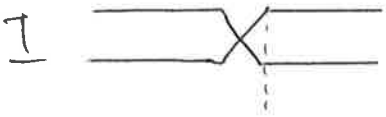


(static discipline)

- I is real also

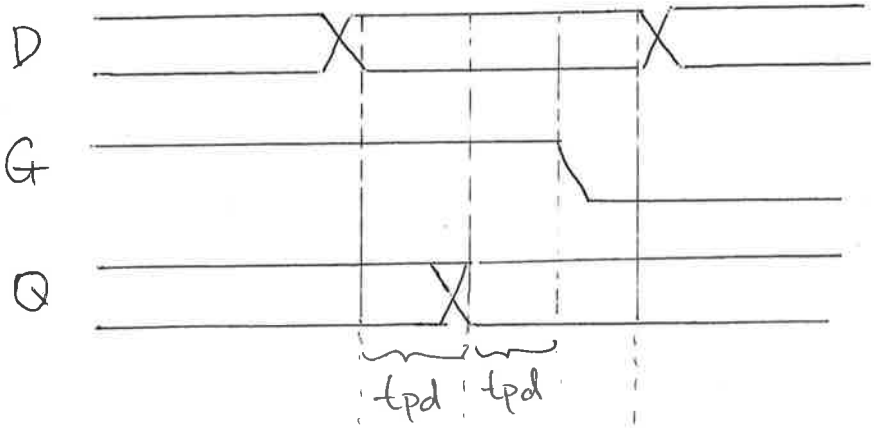


to write together

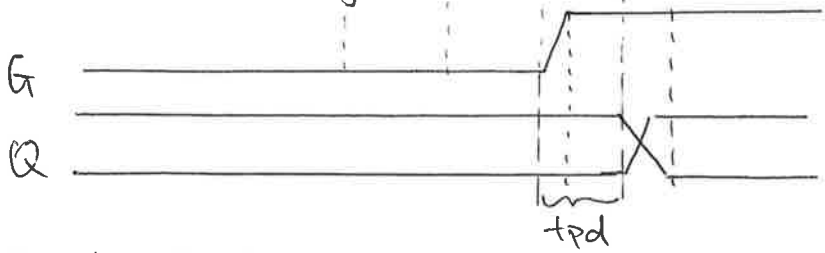


- Now back to D-latch

what if not? meta stability!



lock G 1→0 D change before G D→Q→ A = 2tpd.



Unlock G 0→1 Q change 1 tpd D must remain.

Dynamic discipline include

- Setup time t_s data before clock
- hold time t_h data after clock.

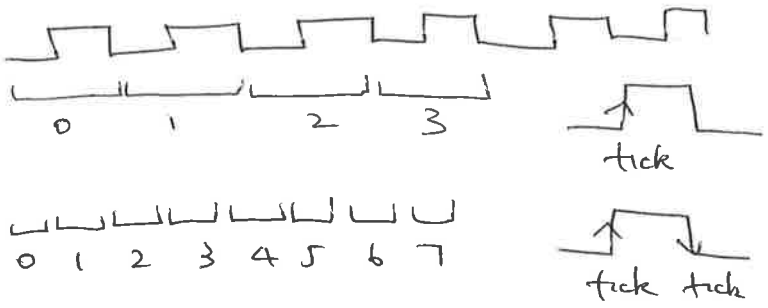
Breaky that lead to logic error

- Clocks and edge-triggered flip-flops.

- what is a clock?

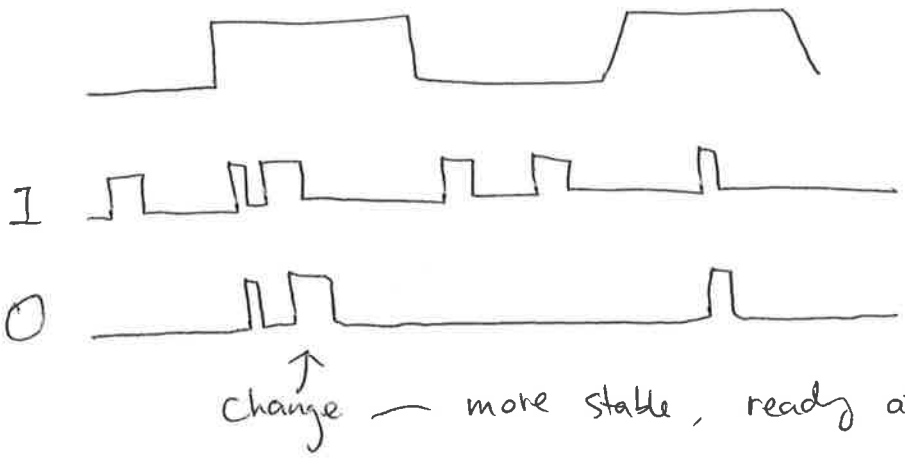
$t = [0, 1, 2, 3, 4, \cdots]$
$x = [1, 3, 5, 7, \cdots]$
$y = [1, 4, 9, 16, 25]$

periodical →



we use single-edge.

- D-latch does not work with the clock neatly



- edge-triggered-flip-flops (ETFF)

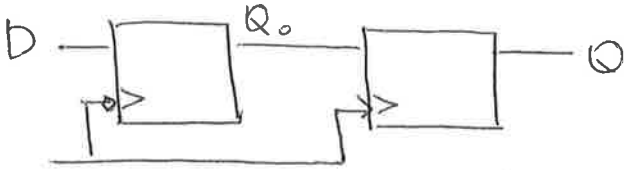
  stable

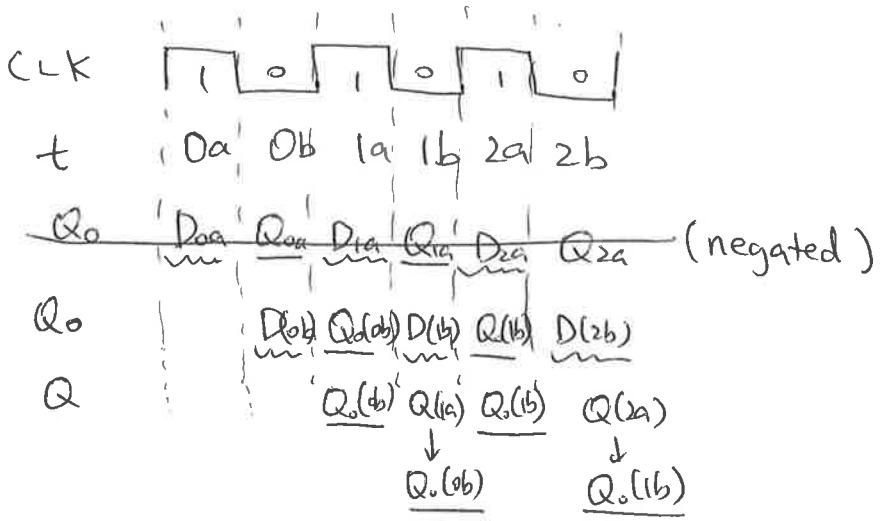
  act as delay

  $t = [0, 1, 2, 3, 4 \cdots]$

  $X = [0, 0, 1, 0, 1 \cdots]$

  $Y = [0, 0, 0, 1, 0 \cdots]$

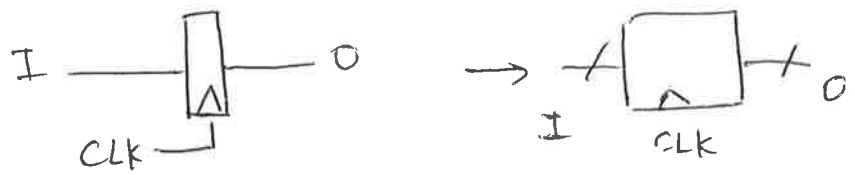


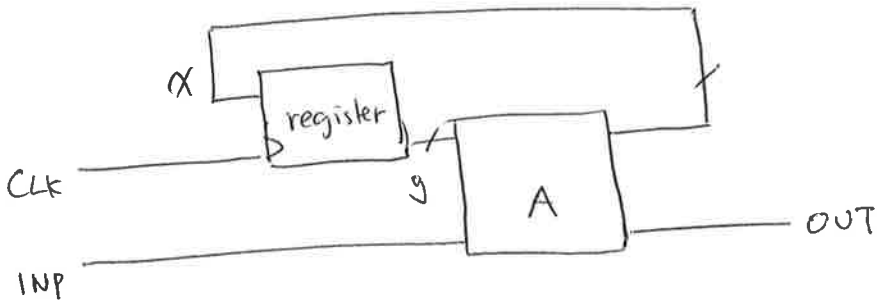


- dynamic discipline

t_{setup}	for $Q_0 \rightarrow t_{setup}$ (D-latch)	for $Q \rightarrow t_{setup}$ (D-latch) $\therefore$ $\sqcup\sqcap\sqcup - t_{pd}^{Q_0} > t_{setup}$
t_{hold}	for $Q_0 \rightarrow t_{hold}$ (D-latch) ~~$\therefore t_{cd} > t_{hold}$~~	for $Q \rightarrow t_{hold}$ (D latch) $\therefore t_{cd}^{Q_0} > t_{hold}$
t_{pd}	after clock change	$\rightarrow t_{pd}$ (D-latch)
t_{cd}		$\rightarrow t_{cd}$ (D-latch)

- register = multi-bit ETFF





register $t_{cd}=1ns$ $t_{pd}=3ns$ $t_s=2ns$ $t_h=2hs$

clock period = 10ns

A $t_{pd} \leq$ ___ $t_{cd} \geq$ ___ ?

after clock, $x \geq 2ns \rightarrow$ ~~INP~~ $\rightarrow$ A $\therefore g \geq 1ns \therefore t_{cd} \geq 2-1=1ns$

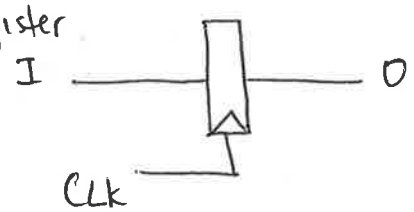
before clock, $x \geq 2ns \rightarrow \therefore x \rightarrow g$ at least 3ns $\quad g \rightarrow x + x \rightarrow g + t_{hold} \leq 10ns$

$\underline{g \rightarrow x}$ 3ns $\quad \underline{x \rightarrow g}$ t_{pd} $\quad$ 2ns

$\therefore t_{pd} \leq 5ns$

In summary, key:

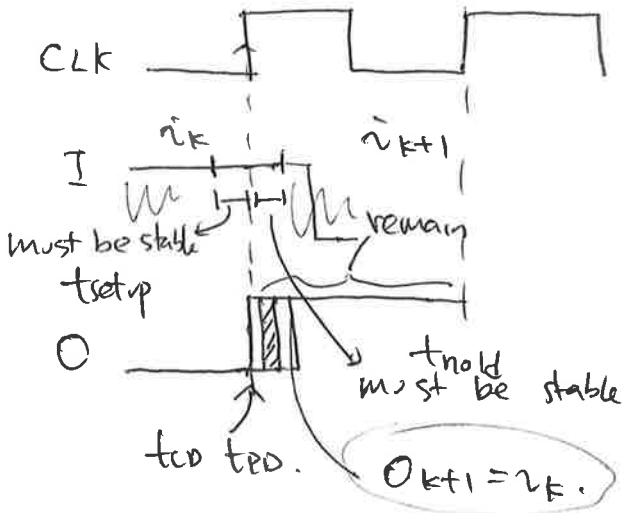
Register



abstraction (level 1)

t=	0	1	2	...
x=	1	0	1	...
y=	0	1	0	...

(level 3)



(level 2)