LECTURE 11 Timing of sequential circuits.

- In order to make progress, we need to understand the timing of memory elements & sequential circuits:

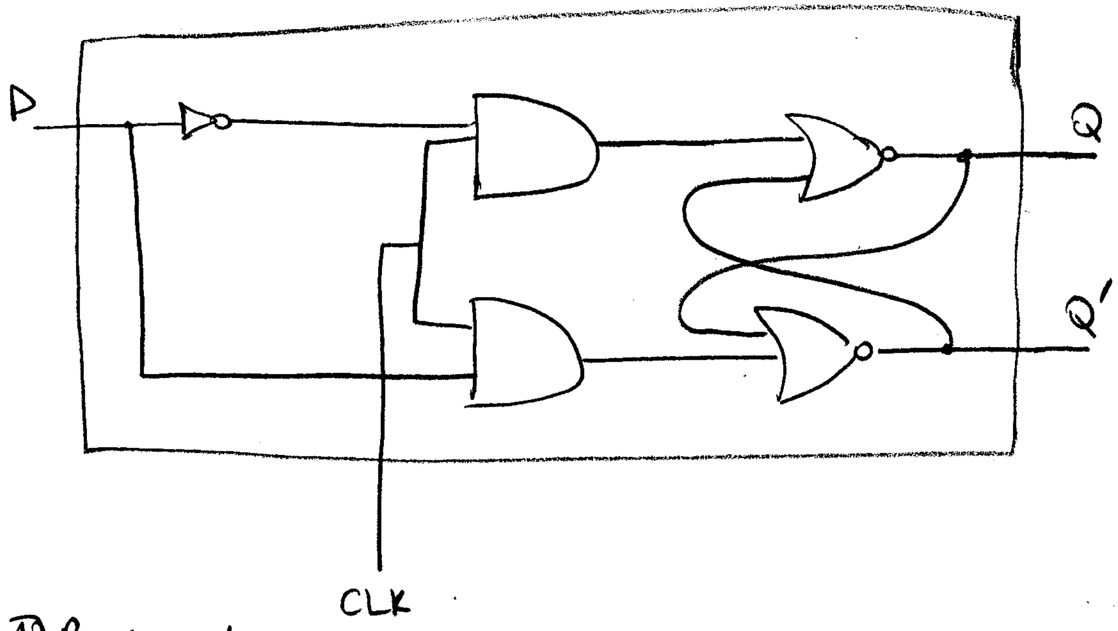
- So far, we used an ideal timing behavior of latches & ffps. In reality, each latch or ff has:

  1) propagation delay(s),

  2) set-up time,

  3) hold time.

I (clocked D-latch):



A) Propagation delay:

- Clearly, the clocked D-latch has a propagation delay.

Two types of prop, delay.

① D-to-Q delay:

≡ maximum delay incurred from a change in D to a ~~change~~ stable output → Q. (and Q'). [Assuming that CLK is constant & at CLK = 1]

② <u>CLK-to-Q delay</u>:

Assuming a constant D, ~~that~~ this is the maximum delay from a change in CLK to a <u>stable output</u> (Q and Q').

- clearly, CLK ↓ (going low): will not make a change in Q: the basic latch will hold the value it already had.
- So, the critical transition here is CLK ↑ (going high).

Ⓑ <u>Set-up time</u>:

≡ the minimum duration <u>before the clock falls</u>, for which the input D has to be <u>constant at the correct value</u>, for the correct operation of the latch.

- Why is there such a duration? [Ask class]

i.e what if I wanted to allow D to change, say right before clock falls?

<u>Answer</u>: Then, that value of D will not get a chance to be propagated → so the latch (Q) will not hold the right value.

(CLK ↓ 0 shuts the latch – and prevents the value to be propagated.)

– So, given our simple gate-delay model, let's figure out what the setup time is for the clocked D-latch.

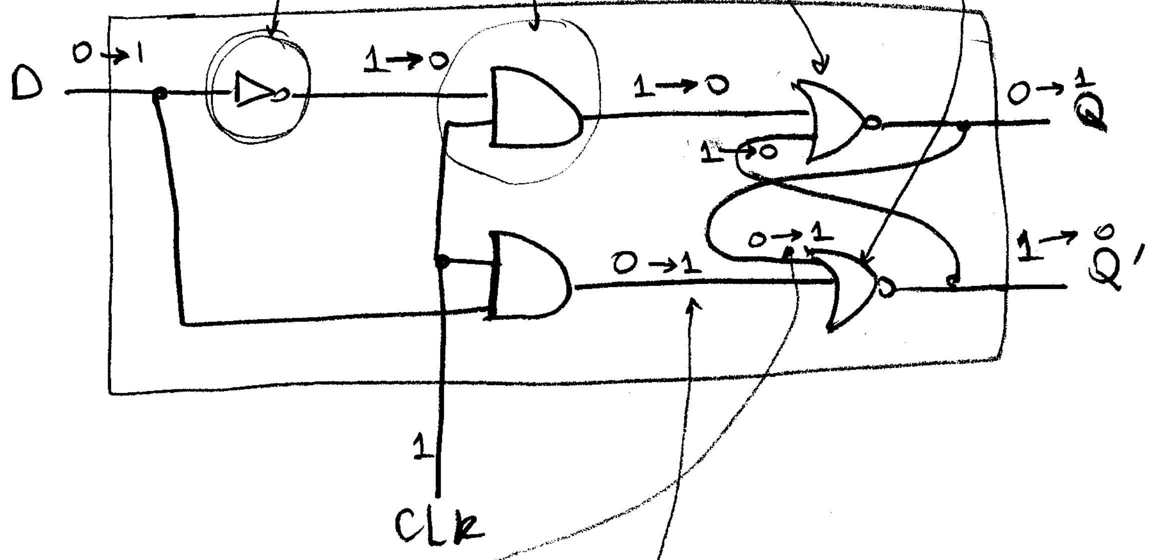
Critical transition:

Initially: $Q = 0$, $(Q' = 1)$, and $D = 0$. , and $CLK = 1$.

Now, assume $D \uparrow 1$.

Total Delay: $t_{inv} + t_{AND} + t_{NOR} + \overset{(\text{to } Q')}{t_{NOR}} \equiv \Delta$

Q' becomes stable



An ~~conservative~~ estimate is that CLK can fall Δ seconds after D : 0→1.

Note that 0→1 on this wire: this wire has to stay at 1 until this change arrives.

So, the minimum time is a little bit less

But can say $t_{setup} \approx \Delta$. for this latch.

Hence, D has to be stable t_{setup} seconds before CLK falls, for it to work correctly.

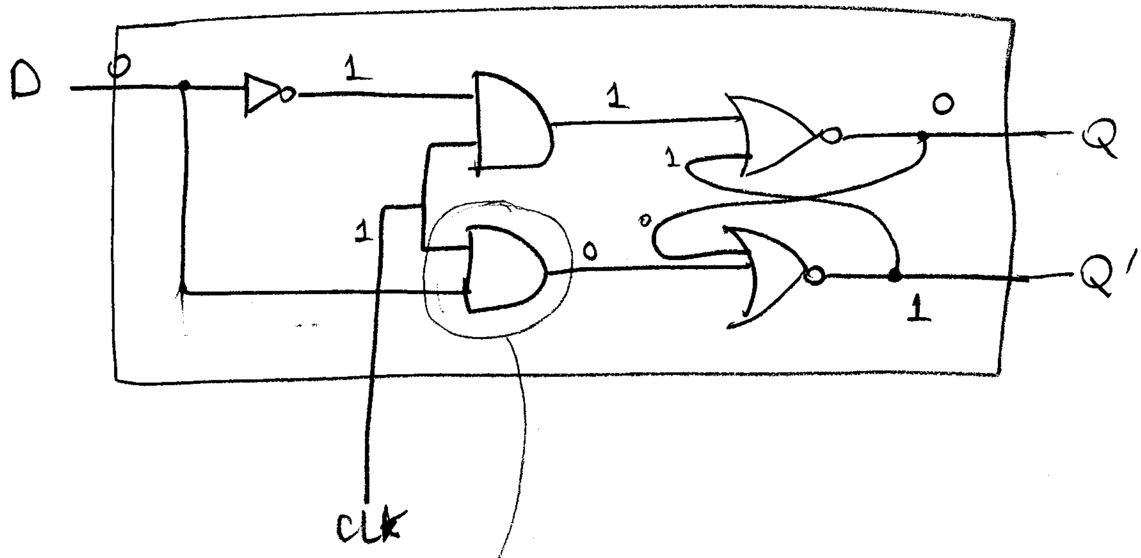
Ⓒ <u>Hold time.</u>

≡ minimum duration for which the D signal has to remain <u>stable</u> (≡ constant at correct value) <u>after</u> the clock <u>has fallen.</u>

- Q: Why would there be such a duration?

<u>Answer:</u>

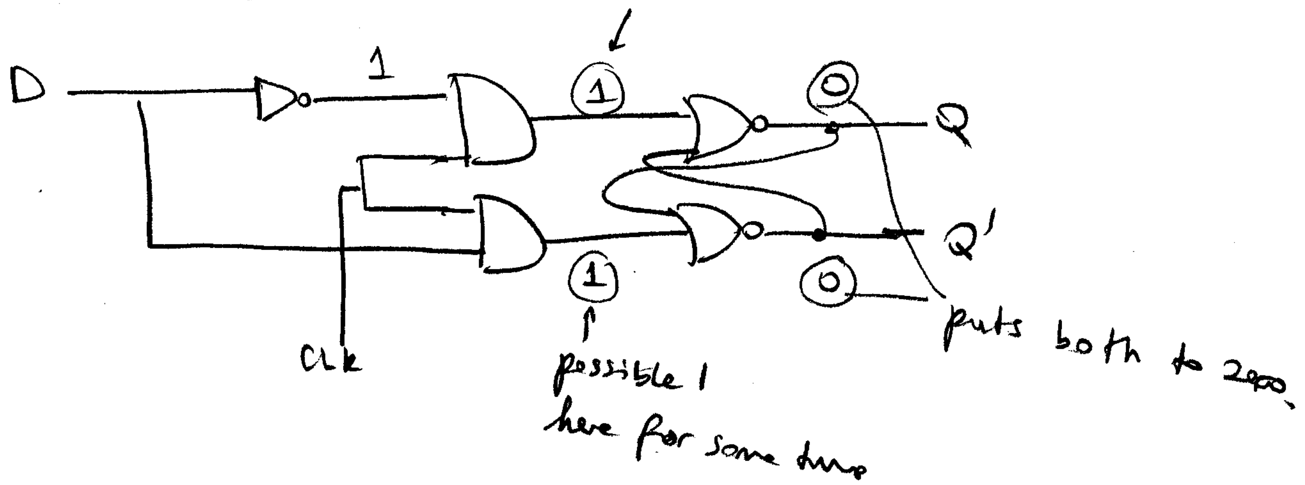
	NOR
0 0	1
0 1	0
1 0	0
1 1	0



Imagine that D = 0, Q = 0, (Q' = 1), CLK = 1 initially.
Now, CLK: 1 → 0, and "D: 0 → 1" at the same time.

We know that this AND gate will settle to a value of 0, but it may oscillate to 1 and then come back to 0. (we are guaranteed that after t_{AND} seconds, it will stabilize to the correct value, but no guarantees change 2 inputs simultaneously.

~~Now~~ then you get:



To avoid this, you must hold the D steady at 0 at least t_{AND} seconds, in the above example

$\therefore$ $t_{hold} = t_{AND}$ for above example.

SUMMARY:

Each latch has:

(A) Propagation delay:

1.a) D-to-Q delay: (assumes clock is steady.

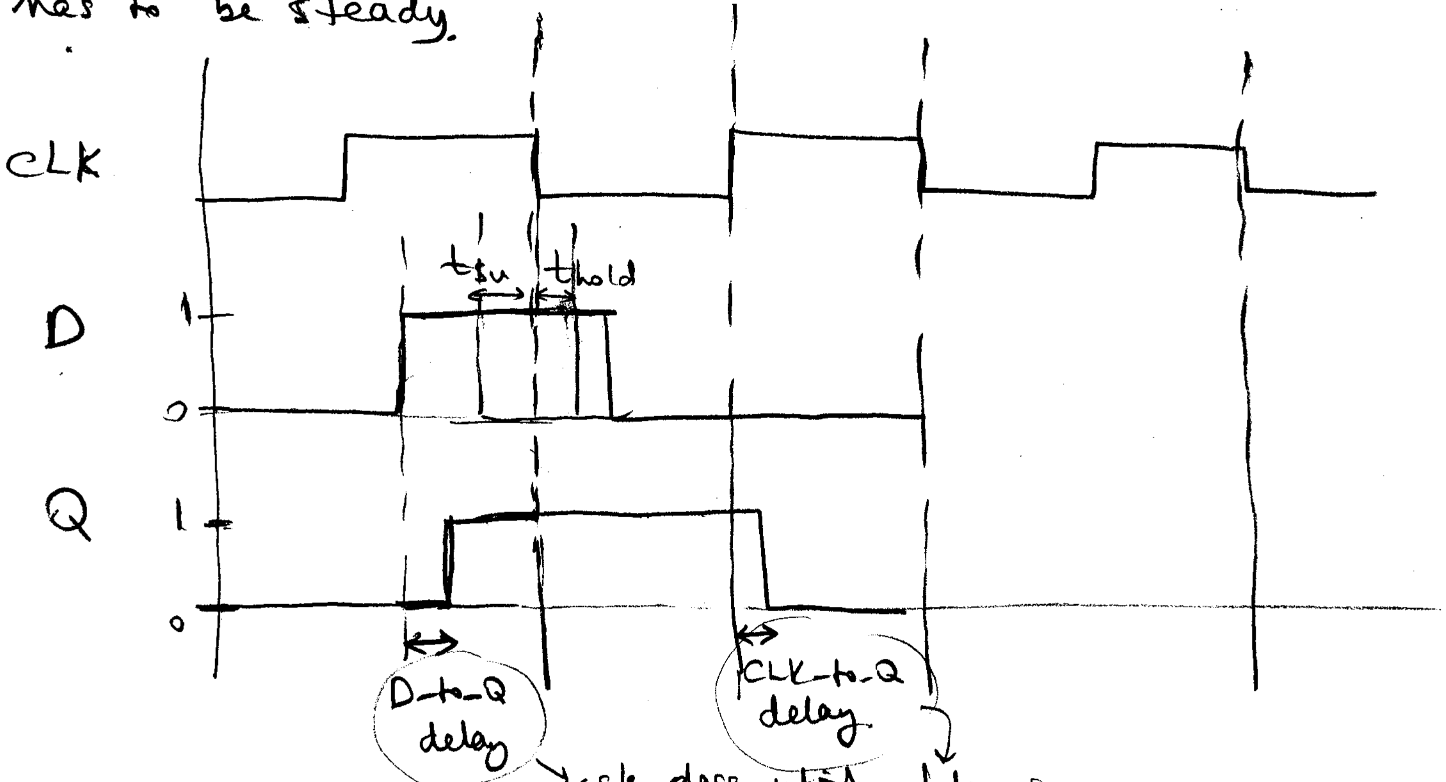
1.b) CLK-to-Q: delay. (" D is steady)

- for each situation you need to examine which one is relevant.

(B) set-up time: the minimum duration before clock falls for which the D input has to be stable.

(C) hold time: minimum time after clock

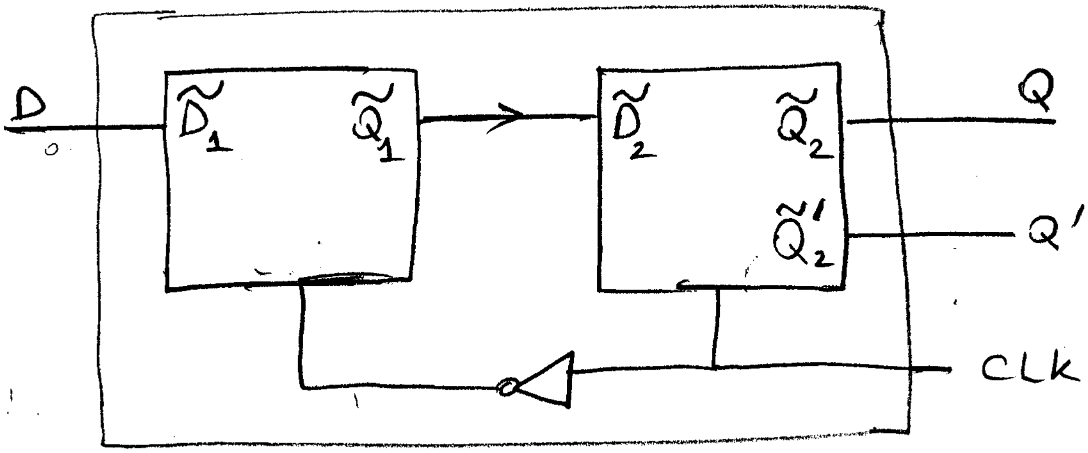
has to be steady.



ask class which delay is applicable in each situation

[II] D flip-flop:

Assume 1) PET FF
2) Master-slave implementation.



(A) <u>Propagation delay</u>:

(- D-to-Q delay not relevant, since change takes place at rising clock edge.)

<u>CLK-to-Q delay</u>:

≡ maximum delay from CLK: $0 \rightarrow 1$, assuming that D is constant at correct value.

In above implementation:

$$t_{CLK-to-Q}^{(ff)} = t_{CLK-to-Q}^{(L2)}$$ → second latch.

(B) <u>Set-up time</u>:

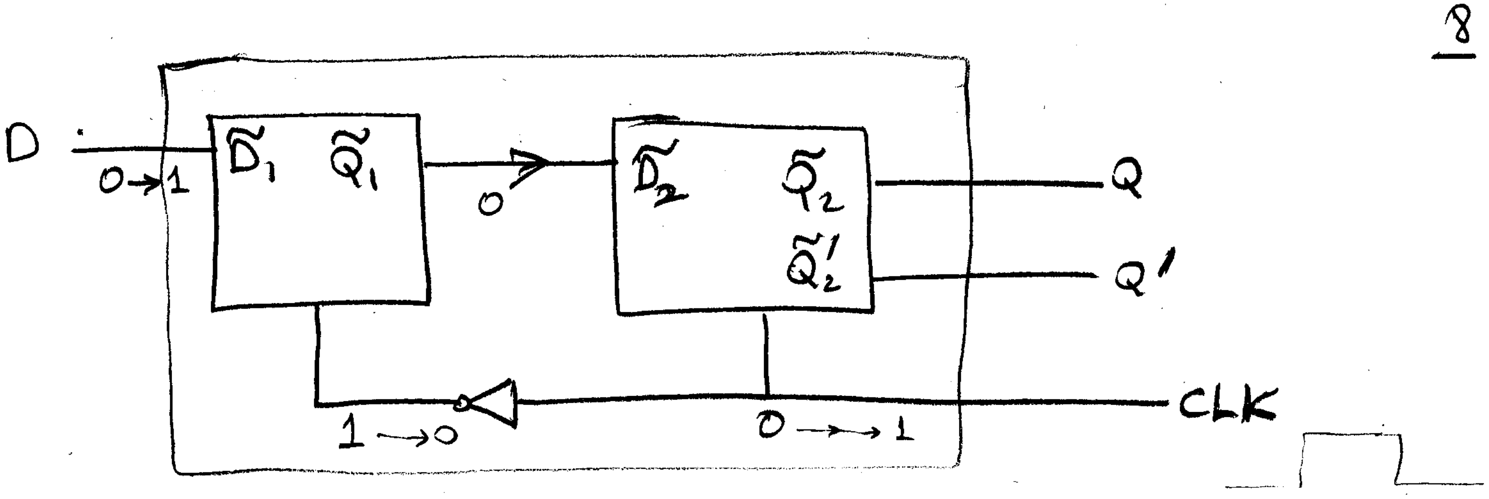
≡ minimum duration for which the D input has to be stable <u>before</u> the active (ex: rising) edge of the clock.

- Why is there such a duration?

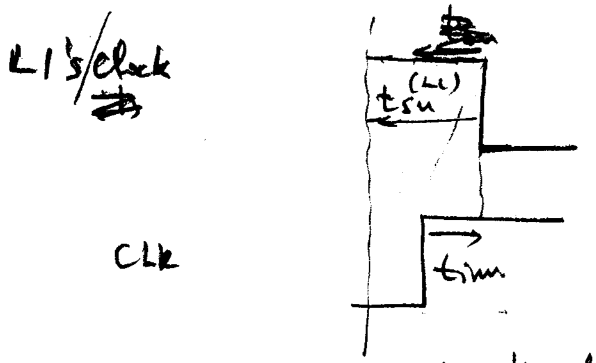
Imagine that $D=0$, $\tilde{Q}_1=0 \Rightarrow \tilde{D}_2=0$ initially and $CLK=0$.

But $D: 0 \rightarrow 1 \rightarrow 0$ before the rising edge of the clock. (see picture below)

Note that CLK on L1 is falling, so, for the L1 to work correctly, D must be stable for a $t_{su}^{(L1)}$ seconds before its clock falls.



Its clock falls an inverter delay after the actual CLK falls.



∴ The D signal to the ff. has to be stable:

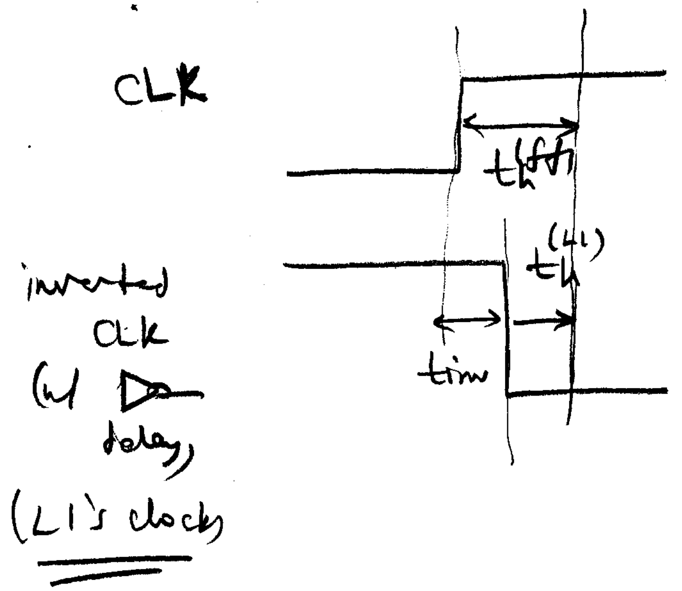
$$\boxed{t_{su}^{(ff)} = t_{su}^{(L1)} - t_{inv}}$$ seconds before CLK rises.

(c) Hold time:

≡ minimum duration for which the D input has to be stable (≡ constant at correct value) after the active (ex: rising) edge.

- why is there such a required duration?
  Imagine D changed to a new (undesired) value right after rising CLK edge.

Recall that L1 has a hold time ($t_h^{(L1)}$).



$$\therefore \boxed{t_h^{(ff)} = t_{inv} + t_h^{(L1)}}$$

in above example. ↑ D must be stable ~~[illegible]~~ $t_h^{(ff)}$ seconds after clock ~~active~~ rising edge.

SUMMARY. (Ffps)

(A) Prop. delay.

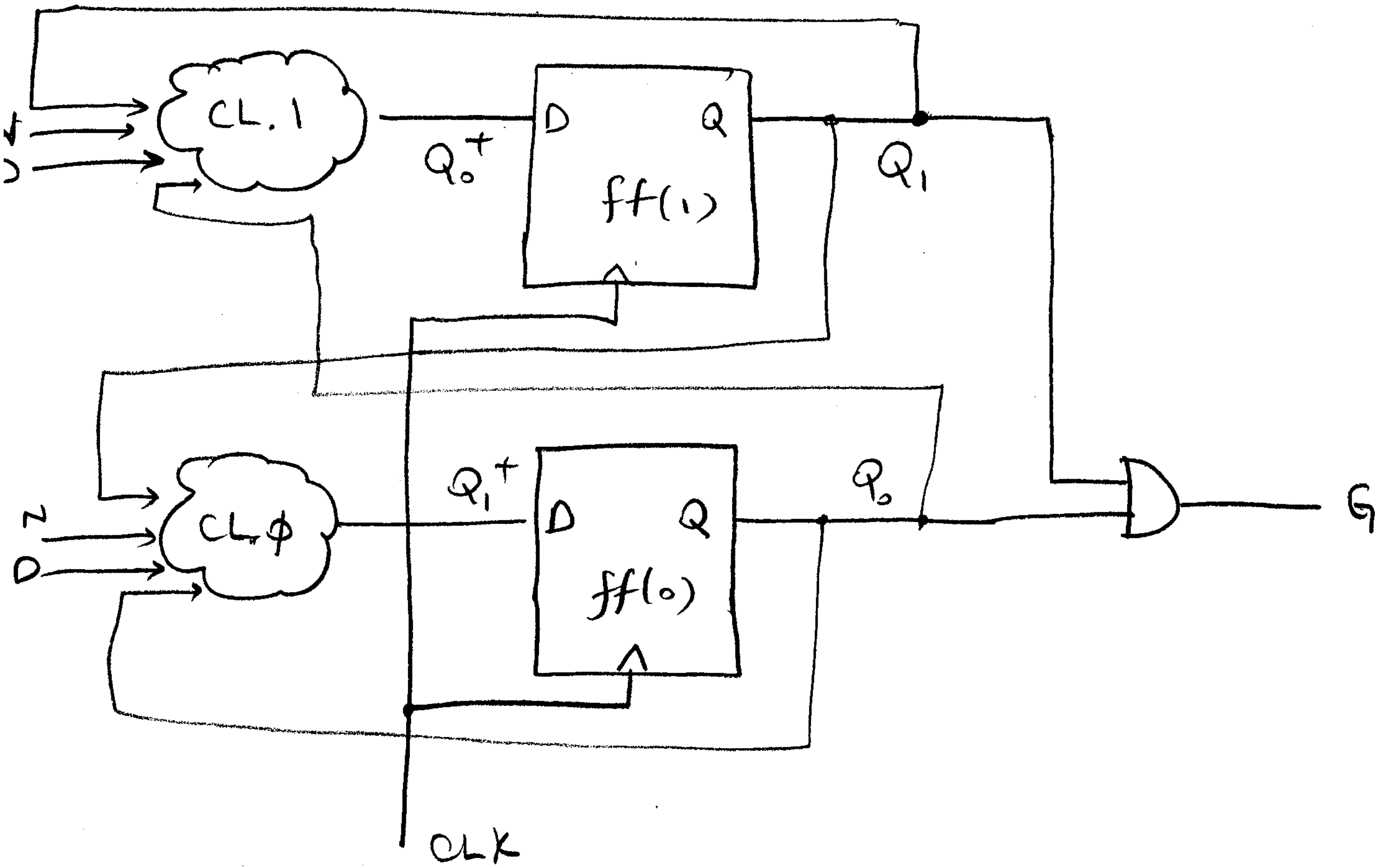
- CLK – ~~to~~ Q prop. delay ≡ max. delay incurred (assuming D is stable) from active clock edge to Q(N

(B) Set-up time, ≡ min. duration for which D has to be stable before active clock edge.

(C) hold time. ≡ min. duration for which D has to be

~~Ex~~ [using setup and hold times and prop delays].
for sequential circuits (e.g. FSMs):

Go back to our simple Vending Machine controller - implementation (end of Lecture 8B, p.16.).



Assume given: →

$t_h^{(i)} \equiv$ hold time of ff i

$t_{su}^{(i)} \equiv$ setup " - - - i

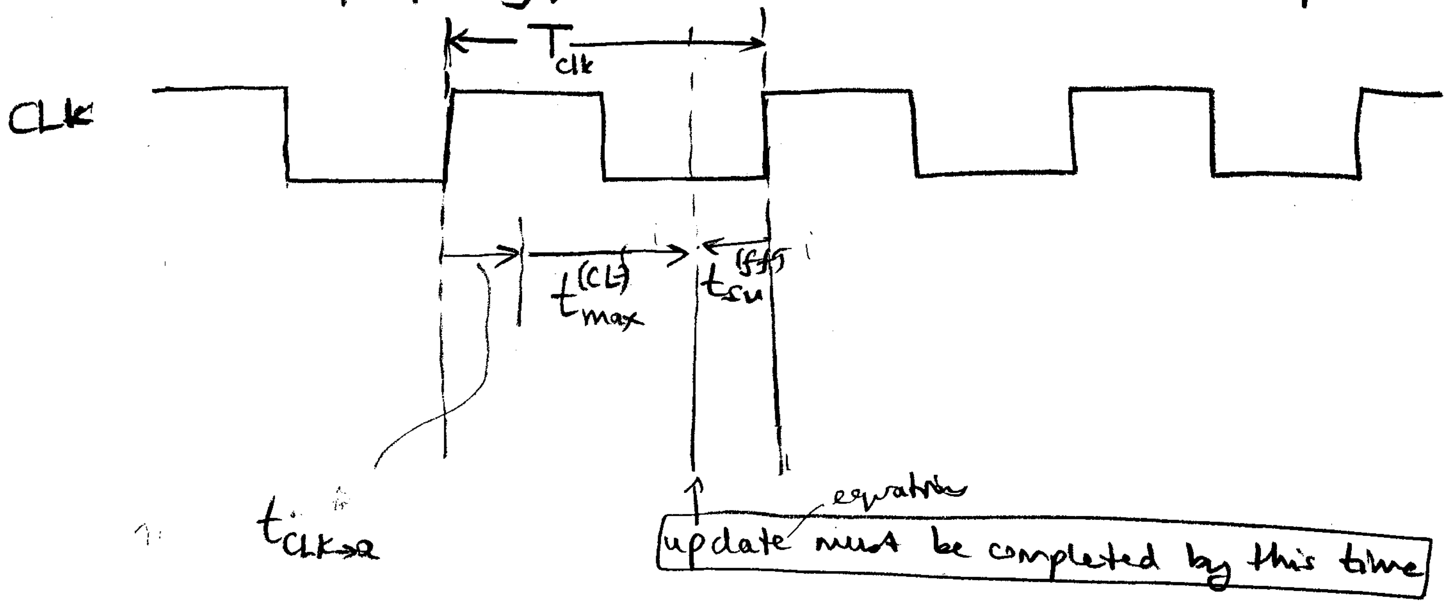
$t_{clk\text{-}to\text{-}Q}^{(i)} \equiv$ CLK-to-Q prop delay of ff i.

$t_{max}^{(CL_i)} :\equiv$ max delay of CL_i.

Questions

① What is the maximum clock frequency such that the circuit will still work correctly?

— Why would the circuit not work correctly beyond a certain frequency? (OR i.e. below a certain clock period?)



update equation must be completed by this time

Let $t_{CLK\to Q} \equiv \max\left\{ t_{CLK\to Q}^{(0)}, t_{CLK\to Q}^{(1)} \right\}$

Then

$$\boxed{T_{clk} \geqslant \underbrace{t_{CLK\to Q} + \max\left\{ t_{su}^{(0)} + t_{max}^{(CL\,0)},\ t_{su}^{(1)} + t_{max}^{(CL\,1)} \right\}}_{\text{minimum clock period, } T_{clk}^{(min)}}}$$

- If you try to run the CLK at a freq $> \frac{1}{T_{clk}^{(min)}}$, then, you will not have to do the update & the ff will sample wrong values

The above is an example of why synchronous digital circuits are limited in their speed. In order to speed up the circuit, you need to reduce the $t_{max}^{(CL)}$:

- there are ways to design architectures so that you will distribute this CL over different cycles: you will see this ~~[illegible]~~ in ECE 154 and ECE 152B.