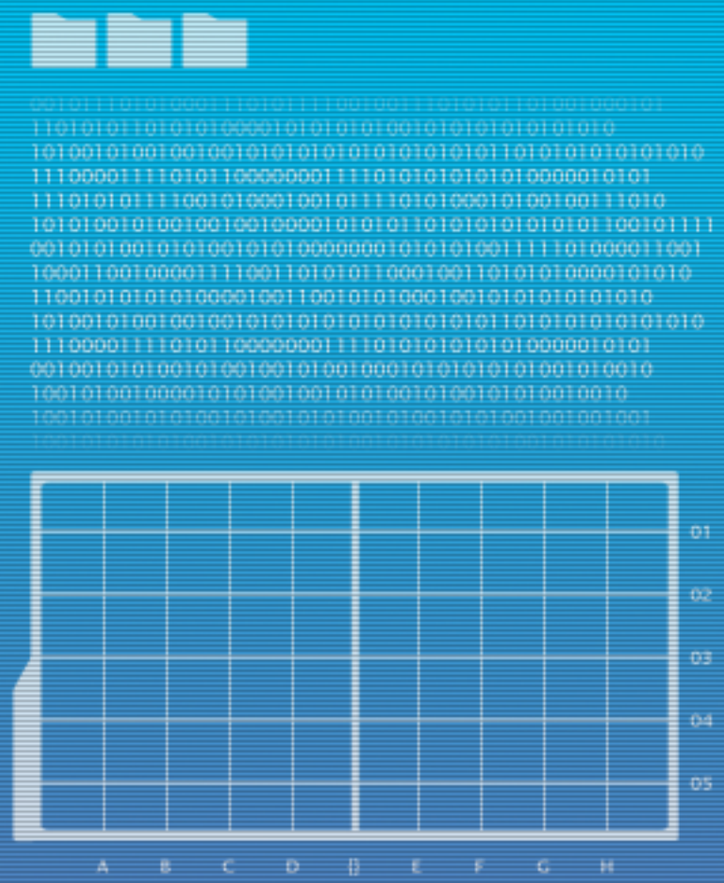
DEPARTMENT OF
INFORMATION SYSTEMS
AND COMPUTER SCIENCE

Timing of Synchronous Circuits

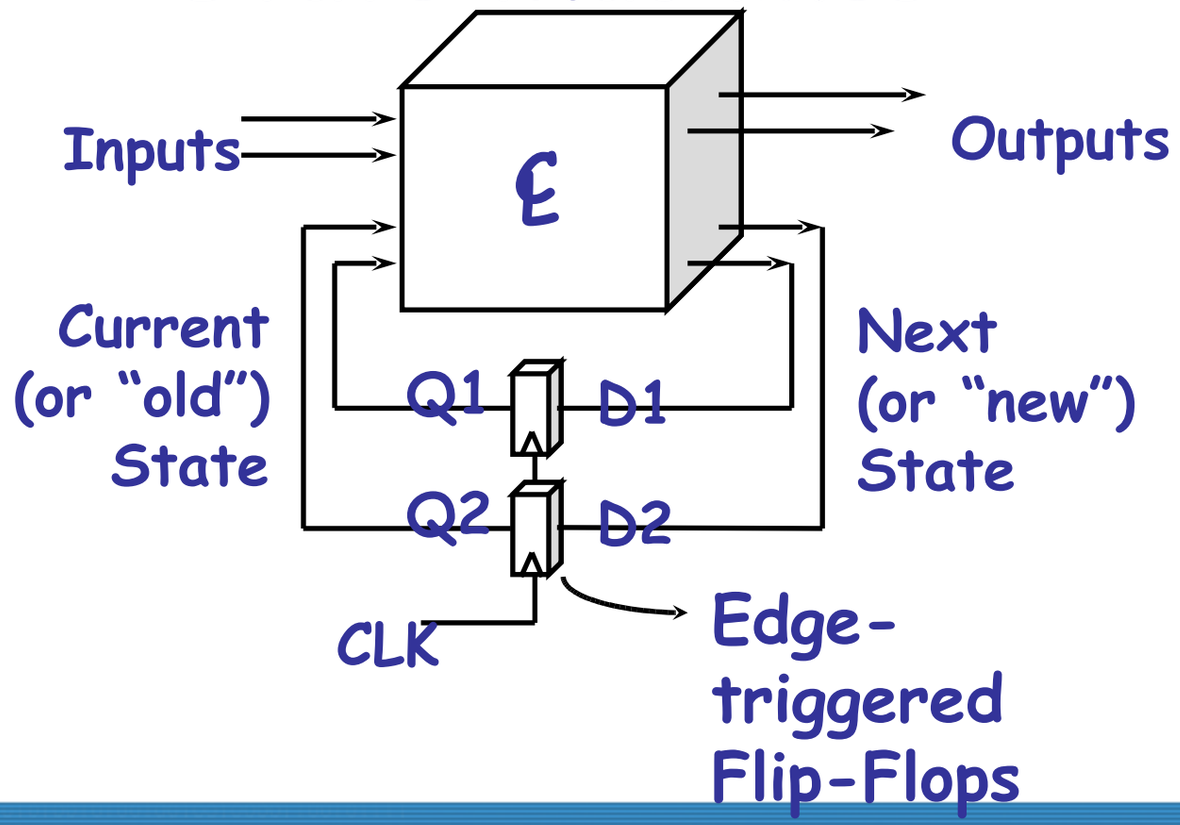
So Things are Ready When They are Ready



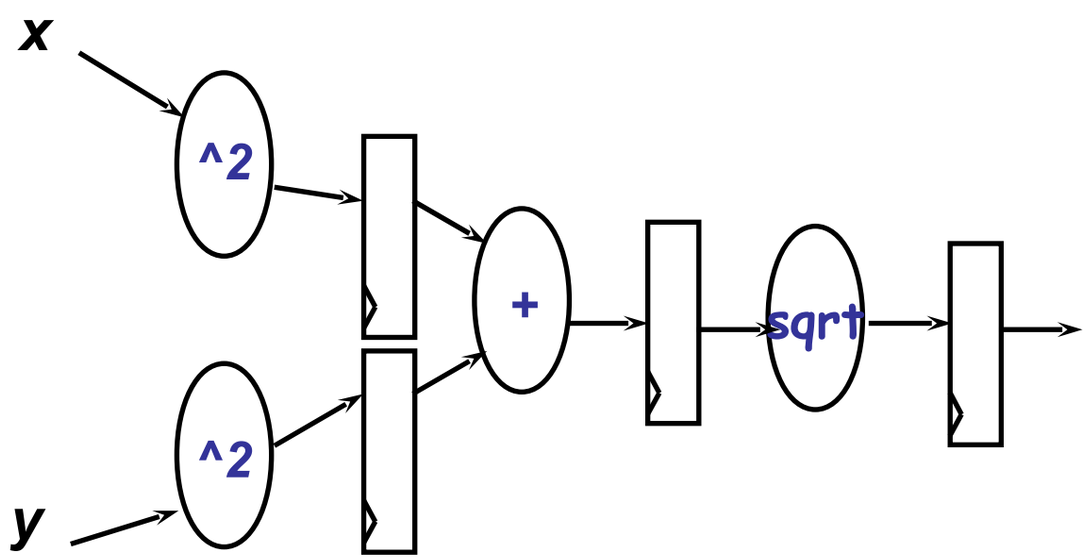
What We Want to Build: Synchronous Circuits

- Circuits where combinational logic paths are broken by edge-triggered flip-flops
- FFs = Almost no transparency = Better control over input transitions

State Machines

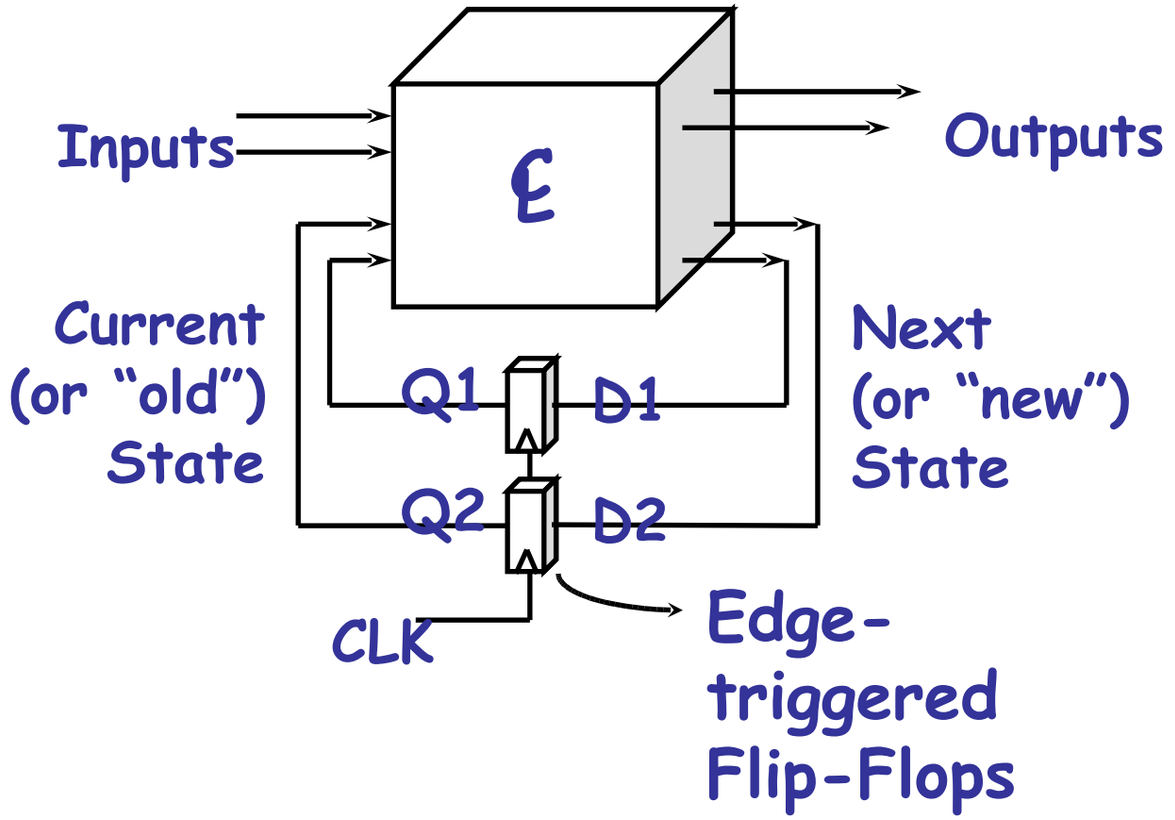


Pipelined Circuits



State Machines

- State is stored in FF's, Next State and Output is computed by Combinational Logic.
- Problem: Can't avoid glitches in complicated Combinational Logic (CL).
  - K-map trick only for SIC.
  - More than one input is changing: next state and incoming input!
- Solution: Break the feedback loop using edge-triggered FF's!
  - At clk edge, D gets copied to Q.
  - At any other time, glitches in D don't matter.
  - Result: No combinational feedback path.

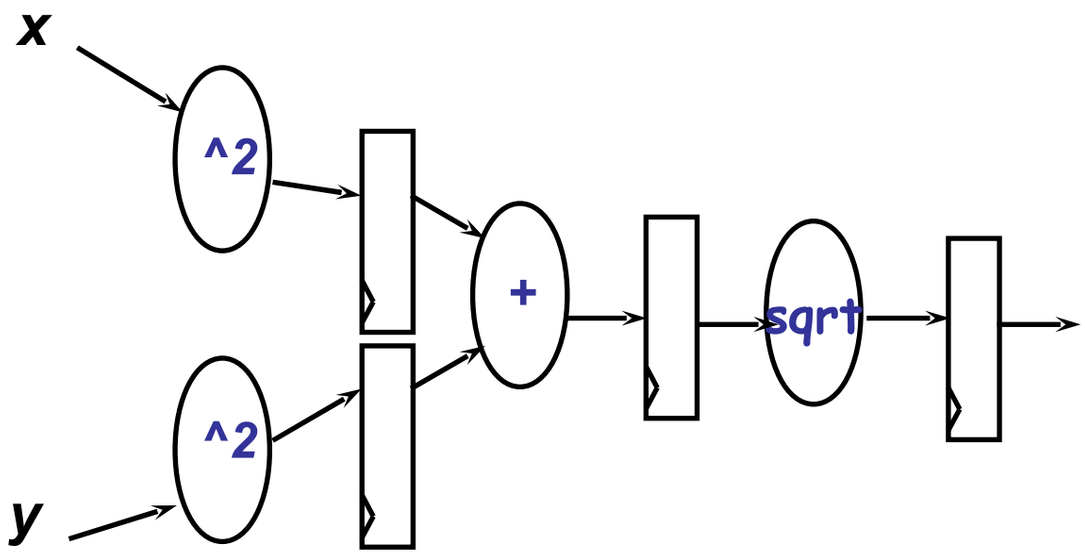


Pipelined Machines

- Goal: Do more computation in less time.
- Solution: Separate the computation into stages.
  - Like an assembly line – different sets of data at different stages at the same time.
  - Use FF's so that values in a stage don't affect the next stage.
    - Data moves with the clock – "synchronous".

Computing Pythagoras's formula:

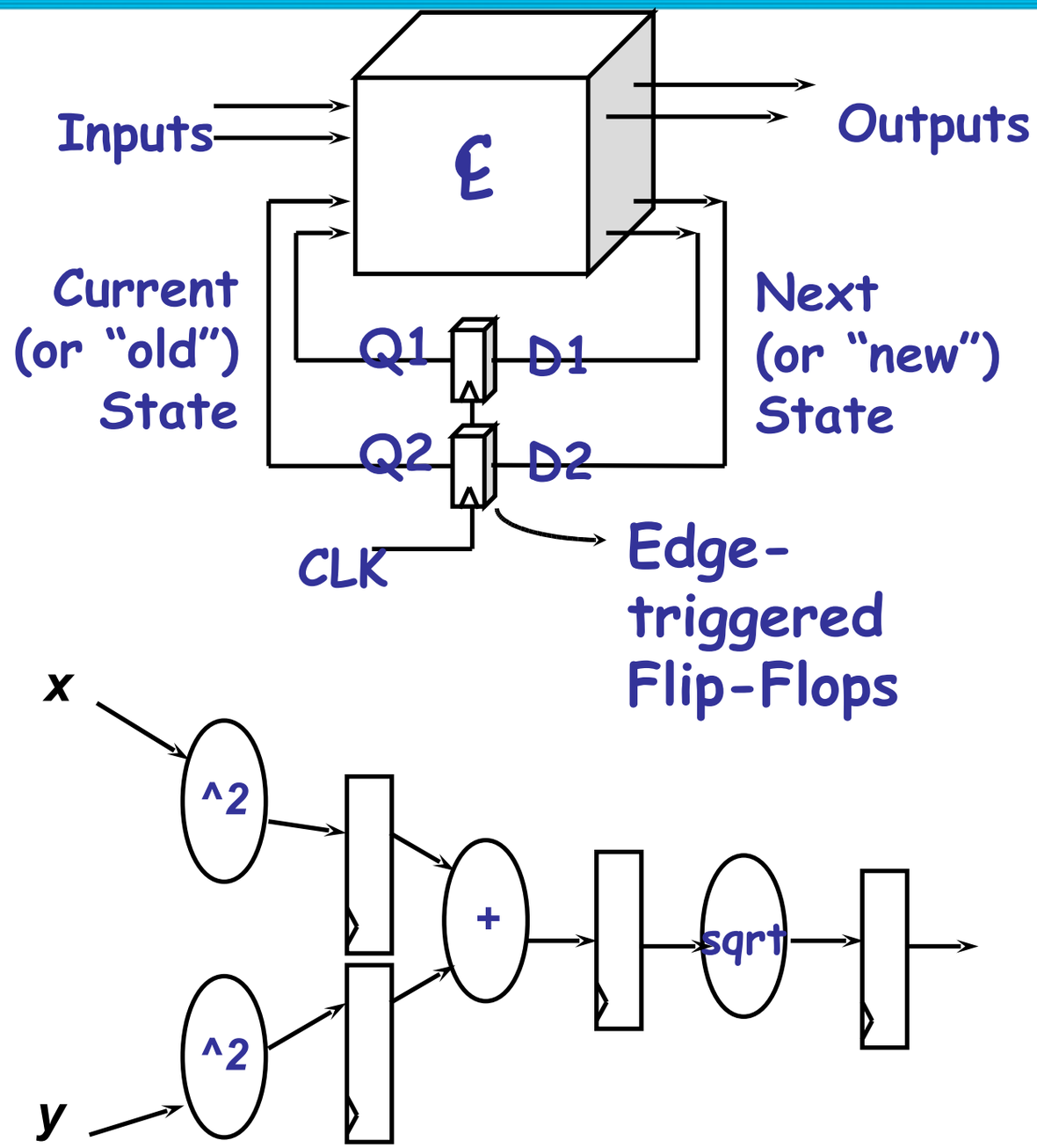
z = sqrt(x^2 + y^2)
(Note: ^ in Java performs XOR.
^ in QBASIC did exponents.)



Timing of Synchronous Circuits

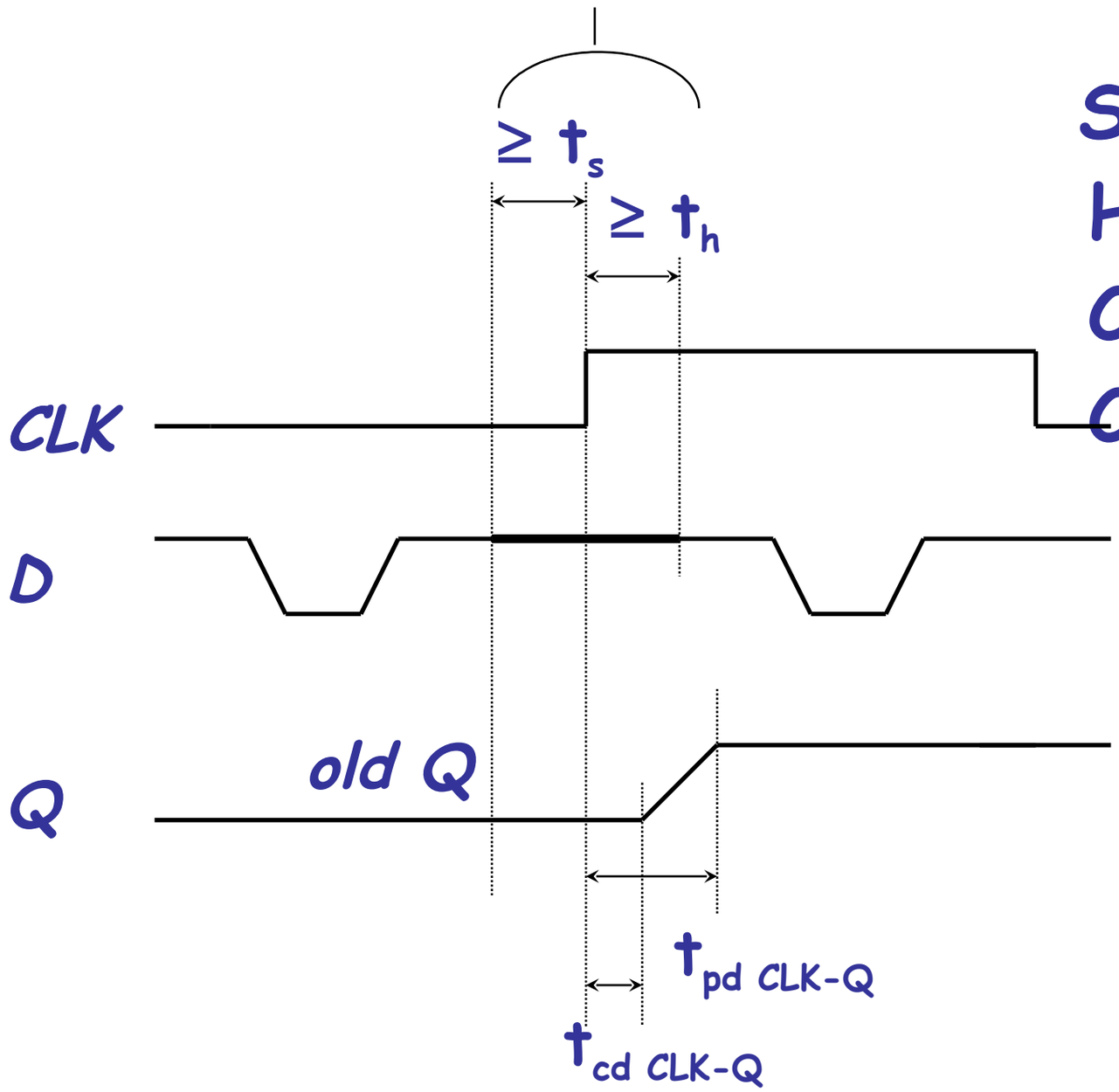
Problem: Need to make sure that setup and hold times are met!

1. ***Data takes time to get from <u>sending</u> FF to <u>receiving</u> FF (Setup).***
   - **Input (D) of receiving FF must be valid and stable for t_s.**
   - **Incoming valid data must not arrive *too late*.**
2. ***FF takes time to <u>latch</u> the data (Hold).***
   - **Make sure D doesn't change until after t_h.**
   - **New/invalid data must not arrive *too early*.**



Edge-Triggered Flip-Flop Timing

D must not change in this region

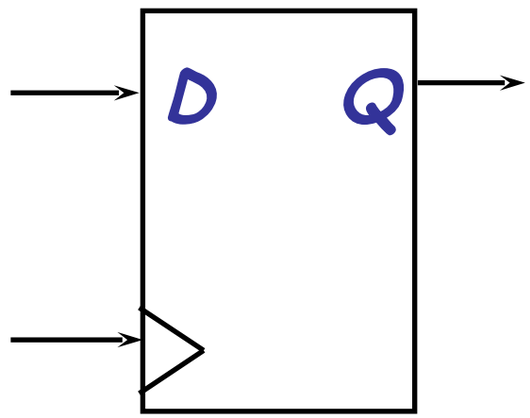


Setup time = t_s

Hold time = t_h

Clk-Q prop. = t_{pdC-Q}

Clk-Q cont. = t_{cdC-Q}

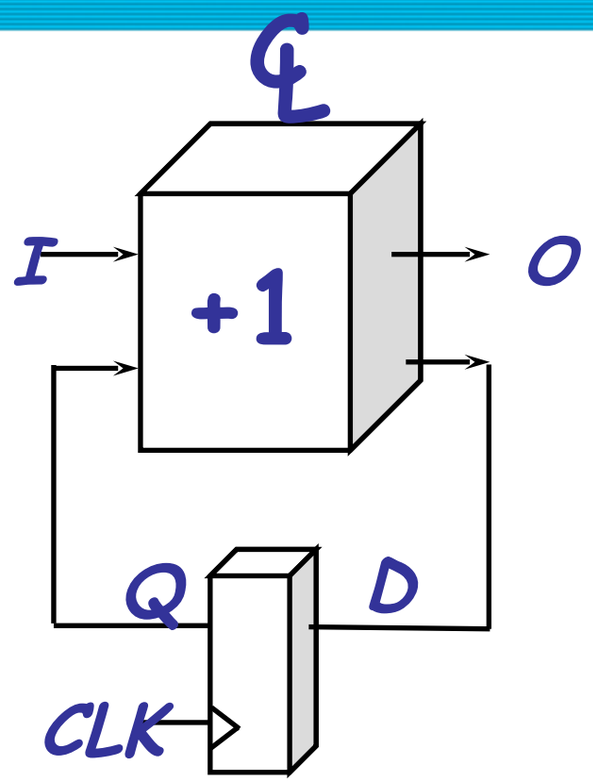


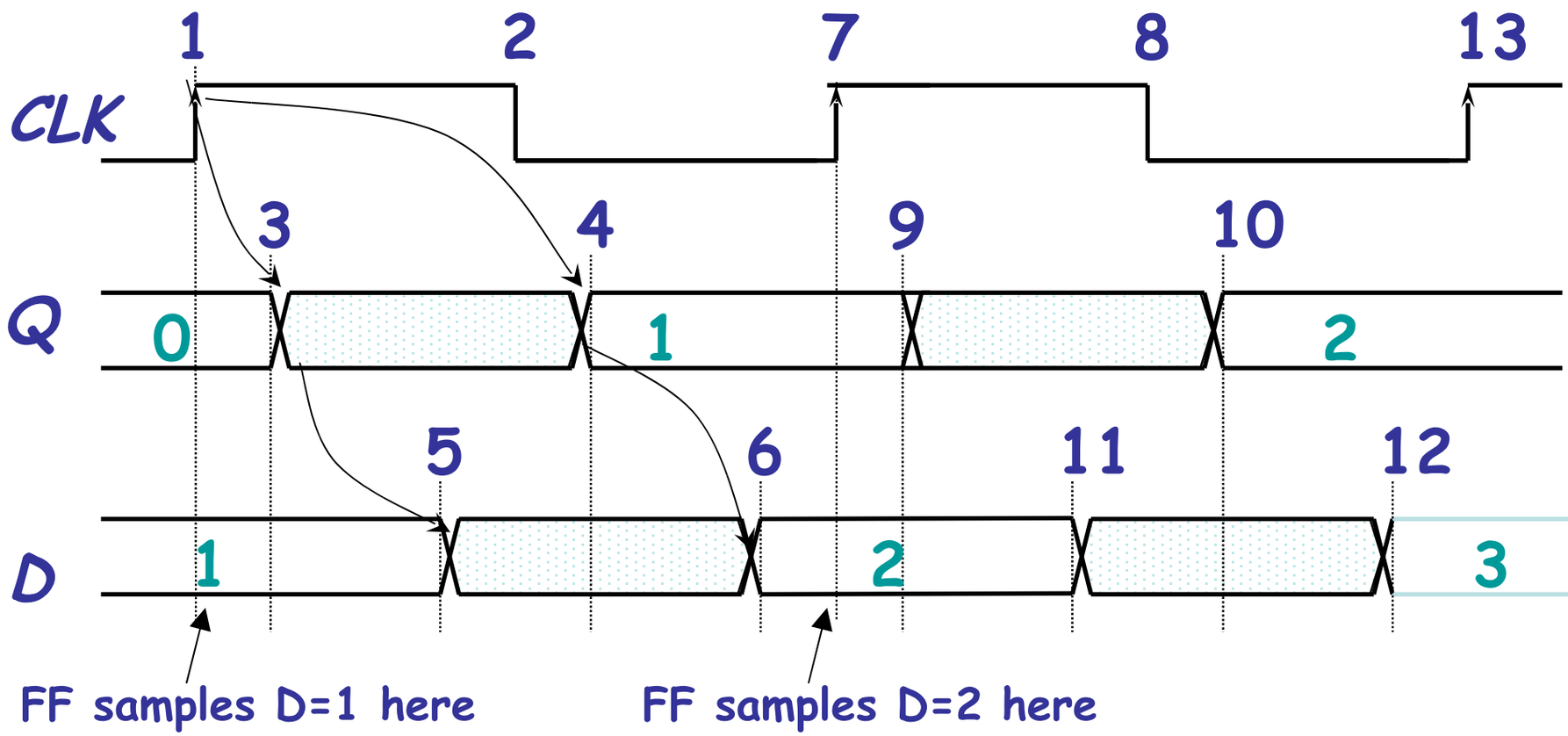
Timing Parameters

- Setup time (t_s)
  - Time <u>before</u> active clock edge at which input D must be valid and stable.
- Hold time (t_h)
  - Time <u>after</u> clock edge until which D must remain valid and stable.
- Clk-to-Q prop. delay ($t_{pdC\text{-}Q}$)
  - Time after Clk edge when new Q is guaranteed to be stable.
- Clk-to-Q cont. delay ($t_{cdC\text{-}Q}$)
  - Time after Clk edge when Q may start changing.

Example:
A counter

Clocked Circuit Timing



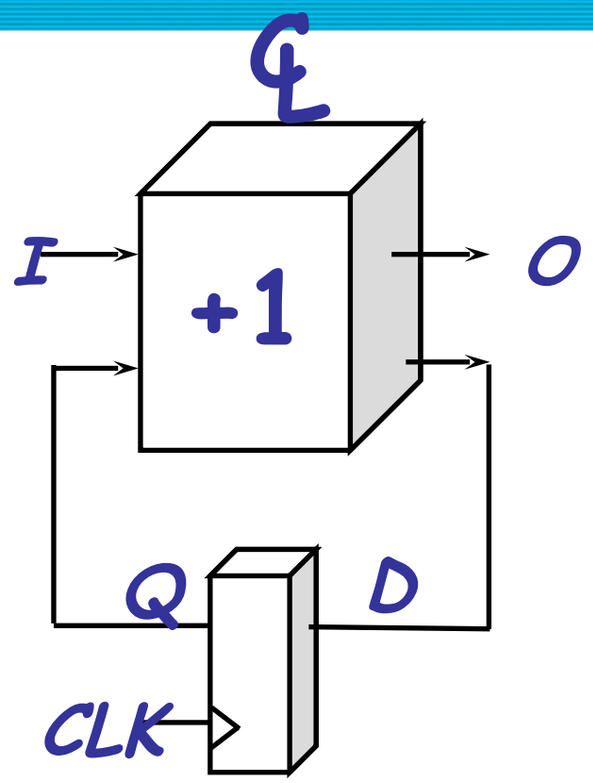


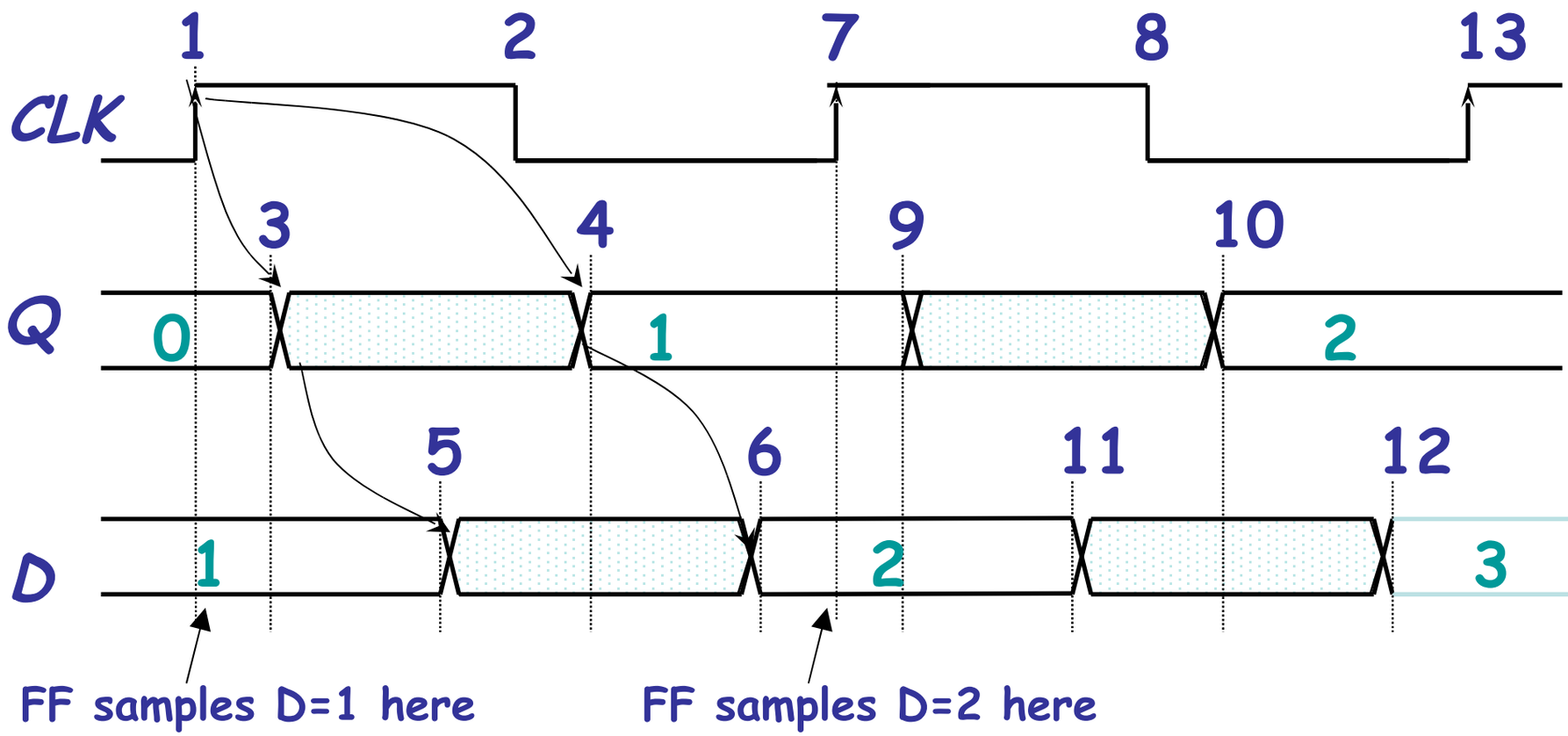
$t_{13} = t_{cdC-Q}$ $t_{14} = t_{pdC-Q}$ $t_{35} = t_{cd-CL}$ $t_{46} = t_{pd-CL}$

Timing rules for flip-flop dictate:

$t_{67} \geq t_s$ t_{15} and $t_{711} \geq t_h$

Timing Rules





Clock period = $t_{clk} = t_{17}$

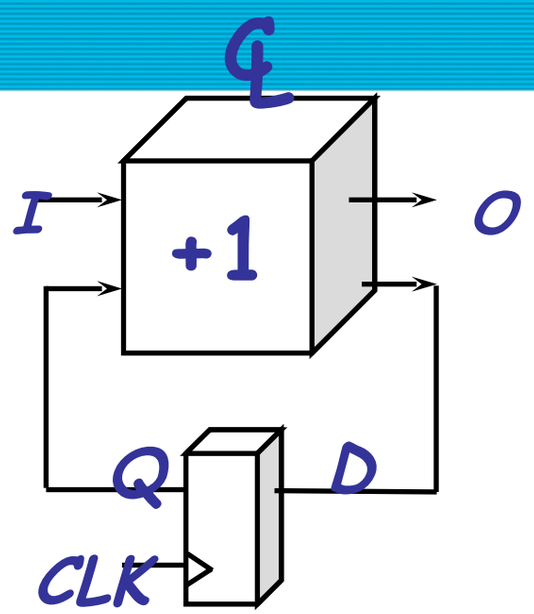
$t_{67} \geq t_s \Rightarrow t_{17} = (t_{14}+t_{46}+t_{67}) \geq t_{pdC-Q}+t_{pd-CL}+t_s$

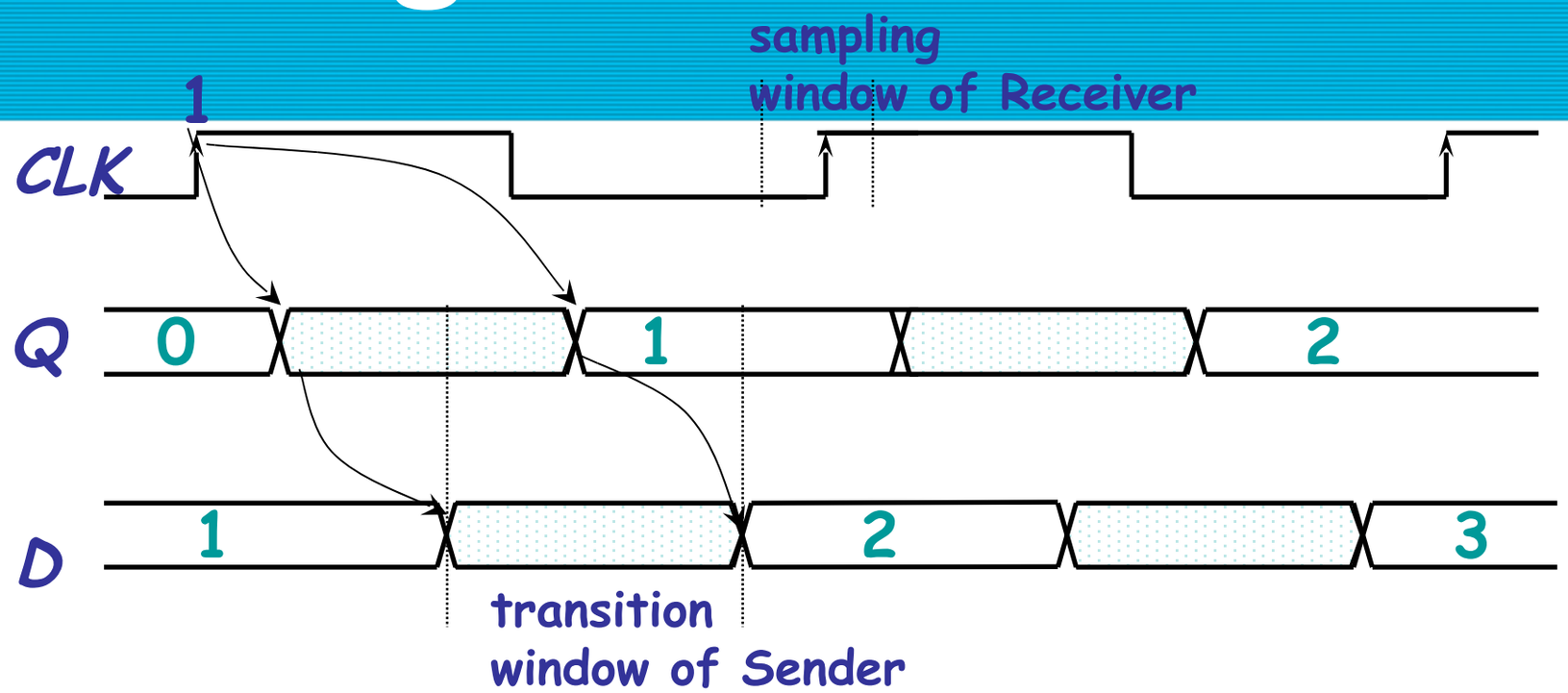
$\Rightarrow t_{clk} \geq t_{pdC-Q}+t_{pd-CL}+t_s$

Hold time constraint:

$t_{15} \geq t_h \Rightarrow (t_{13} + t_{35}) \geq t_h \Rightarrow t_{cd\ C-Q} + t_{cd\ C.L.} \geq t_h$

Timing Rules





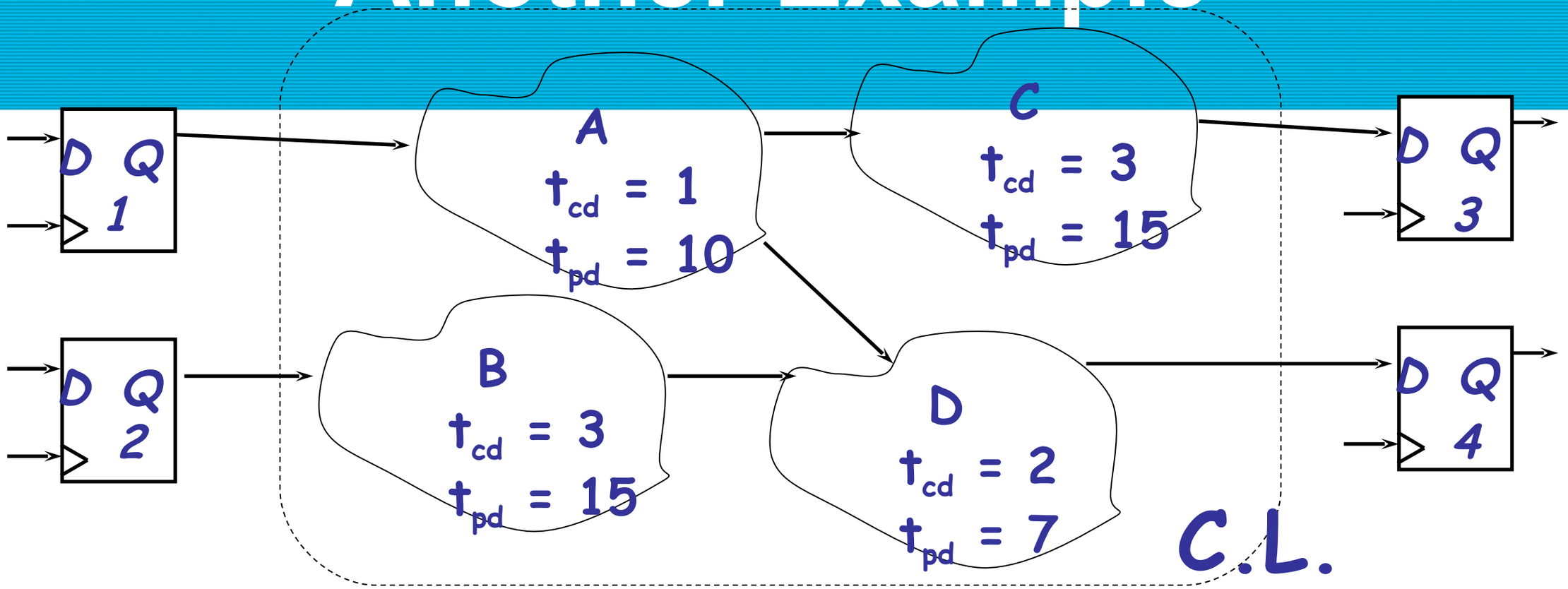
Minimum Clock Period $t_{clk} \geq t_{pdC-Q} + t_{pd-CL} + t_s$

- Clock must be long enough to let new D settle by setup time!

Hold Time Constraint: $t_{cd\ C-Q} + t_{cd\ C.L.} \geq t_h$

- New Q must not contaminate D before hold time
- Usually (but not always), $t_{cdCQ} \geq t_h$, so it works even assuming $t_{cdCL} = 0$.

Another Example



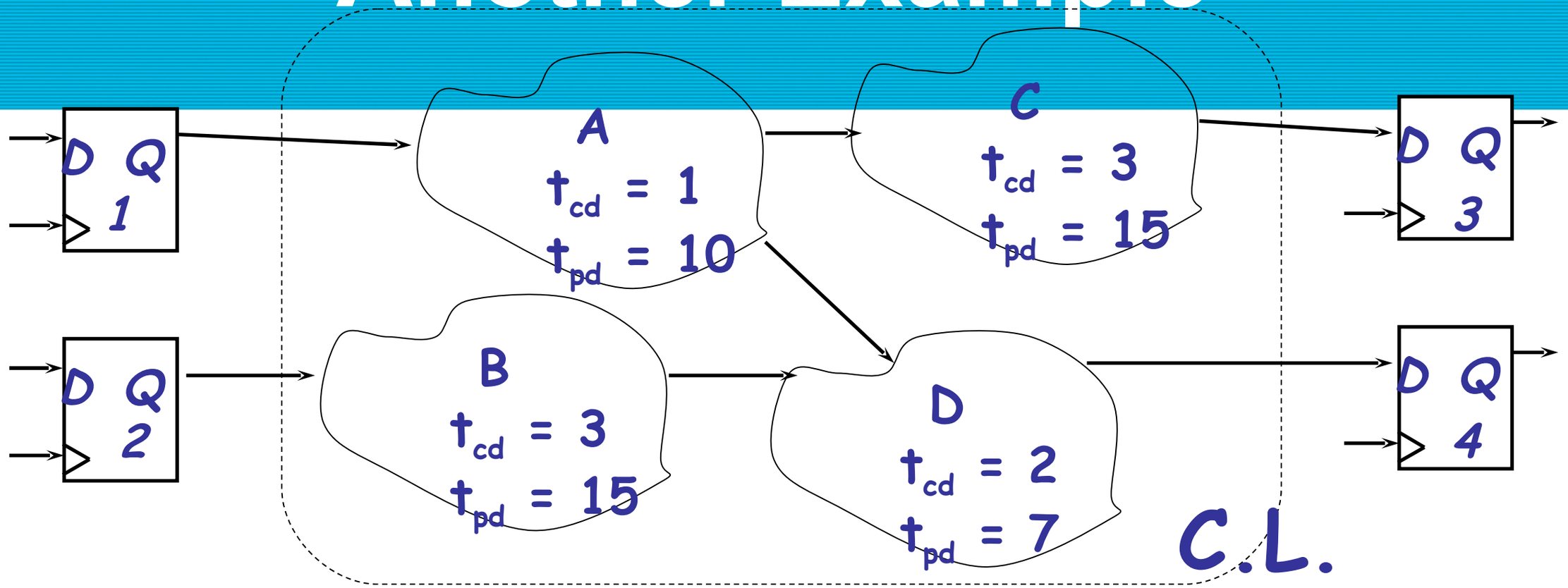
Assume $t_{cdC\text{-}Q}$=2, $t_{pdC\text{-}Q}$=3, t_s=4, t_h=1

$t_{clk} \geq ?$

$t_h \leq ?$

Consider ***worst-case*** (t_{pdCL}) and ***best-case*** (i.e., t_{cdCL}) paths between *any two sending and receiving pairs.*

Another Example



Assume $t_{cdC\text{-}Q}=2$, $t_{pdC\text{-}Q}=3$, $t_s=4$, $t_h=1$

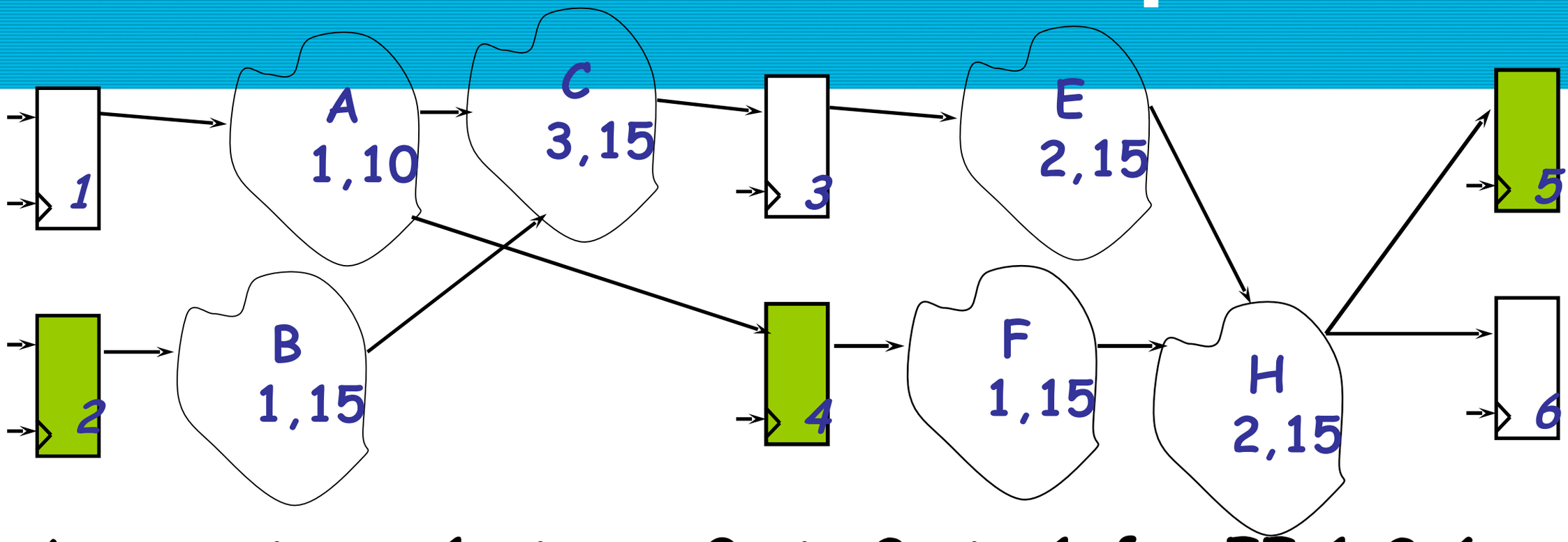
$$t_{clk} \geq t_{pdC\text{-}Q1} + (t_{pdA} + t_{pdC}) + t_{s3} = 32$$

$$t_h \leq t_{cdC\text{-}Q1} + (t_{cdA} + t_{cdD}) = 5$$

Consider ***worst-case*** (t_{pdCL}) and ***best-case*** (i.e., t_{cdCL}) paths between *any two sending and receiving pairs.*

DISCS

Yet Another Example



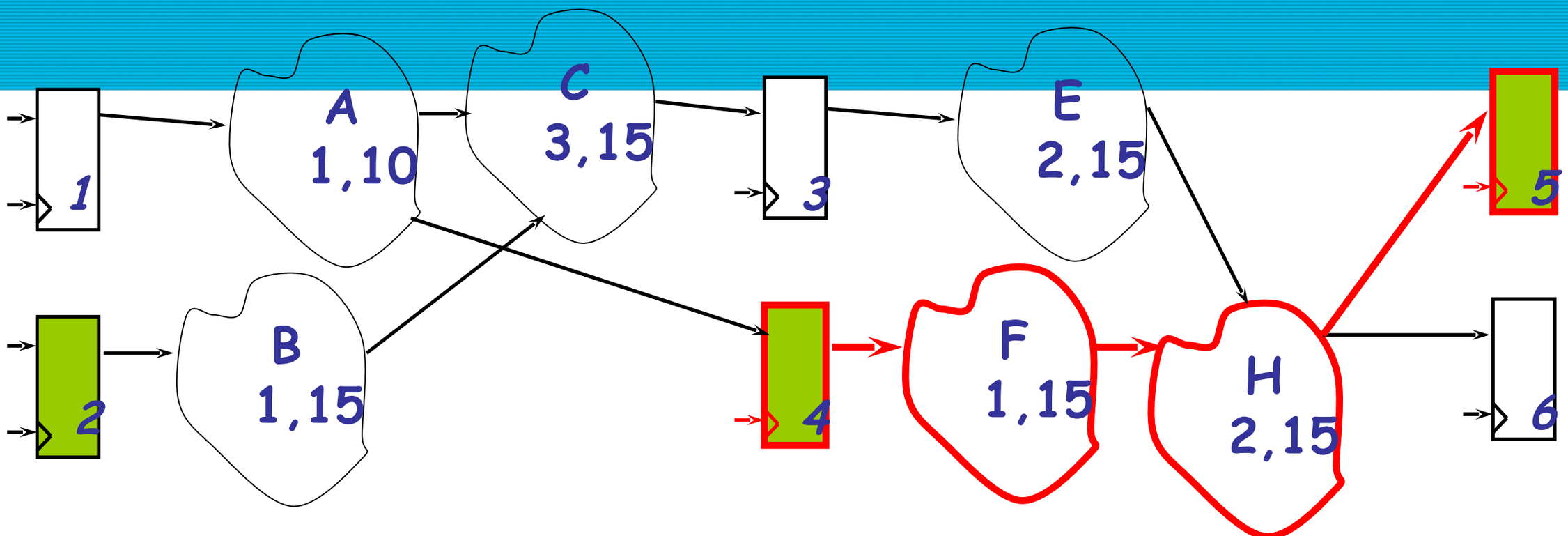
Assume $t_{cdC-Q}=1$, $t_{pdC-Q}=2$, $t_s=3$, $t_h=1$ for FF 1,3,6

$t_{cdC-Q}=3$, $t_{pdC-Q}=4$, $t_s=4$, $t_h=3$ for FF 2,4,5

$t_{clk} \geq ?$

$t_h \leq ?$

Clock Period



Assume $t_{cdC\text{-}Q}=1$, $t_{pdC\text{-}Q}=2$, $t_s=3$, $t_h=1$ for FF 1,3,6

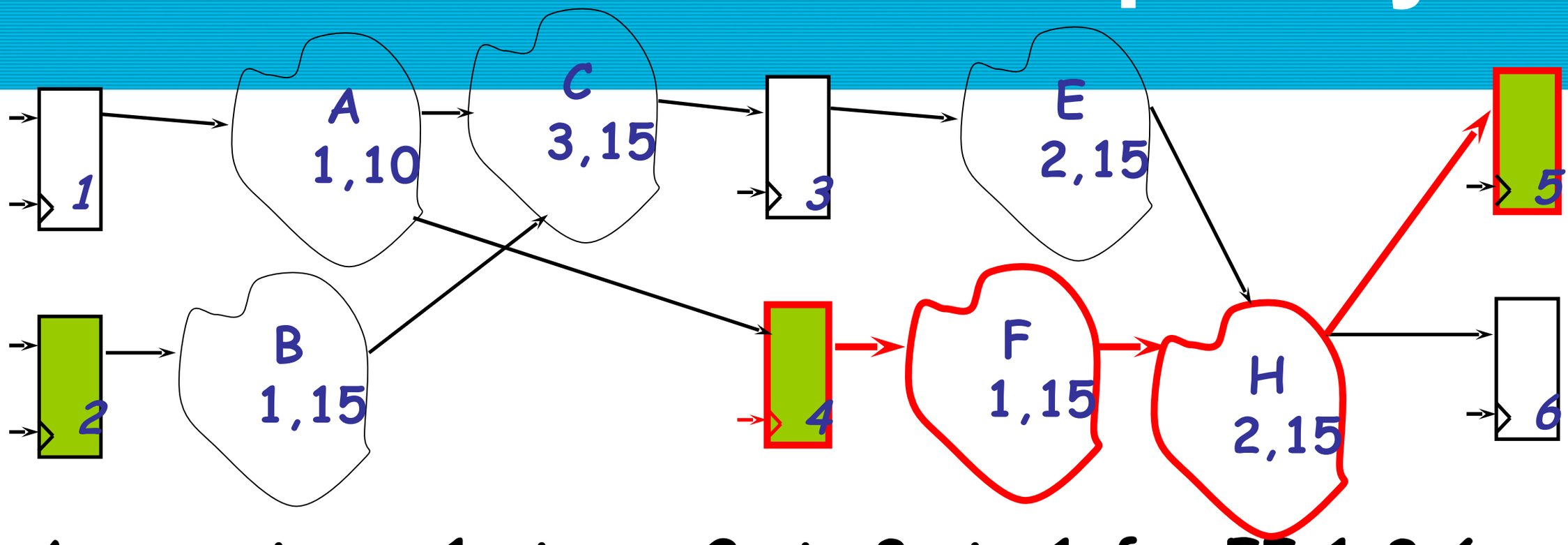
$t_{cdC\text{-}Q}=3$, $t_{pdC\text{-}Q}=4$, $t_s=4$, $t_h=3$ for FF 2,4,5

$$t_{clk} \geq t_{pdC\text{-}Q4} + t_{pdF} + t_{pdH} + t_{s5} =$$

$$4 + (15 + 15) + 4 = 38$$

Many paths with total $t_{pdCL} = 30$, but don't forget $t_{pdC\text{-}Q}$ (start FF) and t_s (end FF)!

Maximum Clock Frequency



Assume $t_{cdC-Q}=1$, $t_{pdC-Q}=2$, $t_s=3$, $t_h=1$ for FF 1,3,6

$t_{cdC-Q}=3$, $t_{pdC-Q}=4$, $t_s=4$, $t_h=3$ for FF 2,4,5

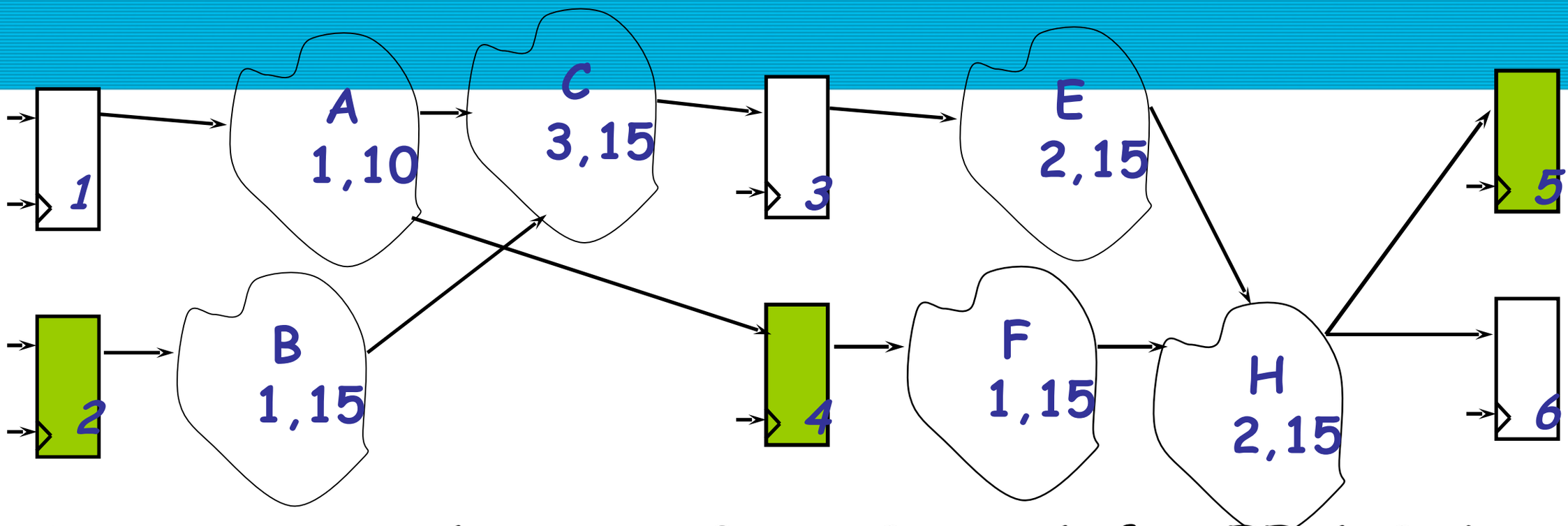
Min Clock Period ⇒ **Max Clock Frequency**

- Suppose time unit is 1 nanosecond (ns)

  $t_{clk} \geq 38$ ns

  $f \leq 1$ cycle / 38ns ⇒ 26 MHz

Hold Times



Assume $t_{cdC-Q}=1$, $t_{pdC-Q}=2$, $t_s=3$, $t_h=1$ for FF 1,3,6

$t_{cdC-Q}=3$, $t_{pdC-Q}=4$, $t_s=4$, $t_h=3$ for FF 2,4,5

these must *all* be true

$$t_{h3} \le t_{cdC-Q1} + t_{cdA} + t_{cdC} = 5$$

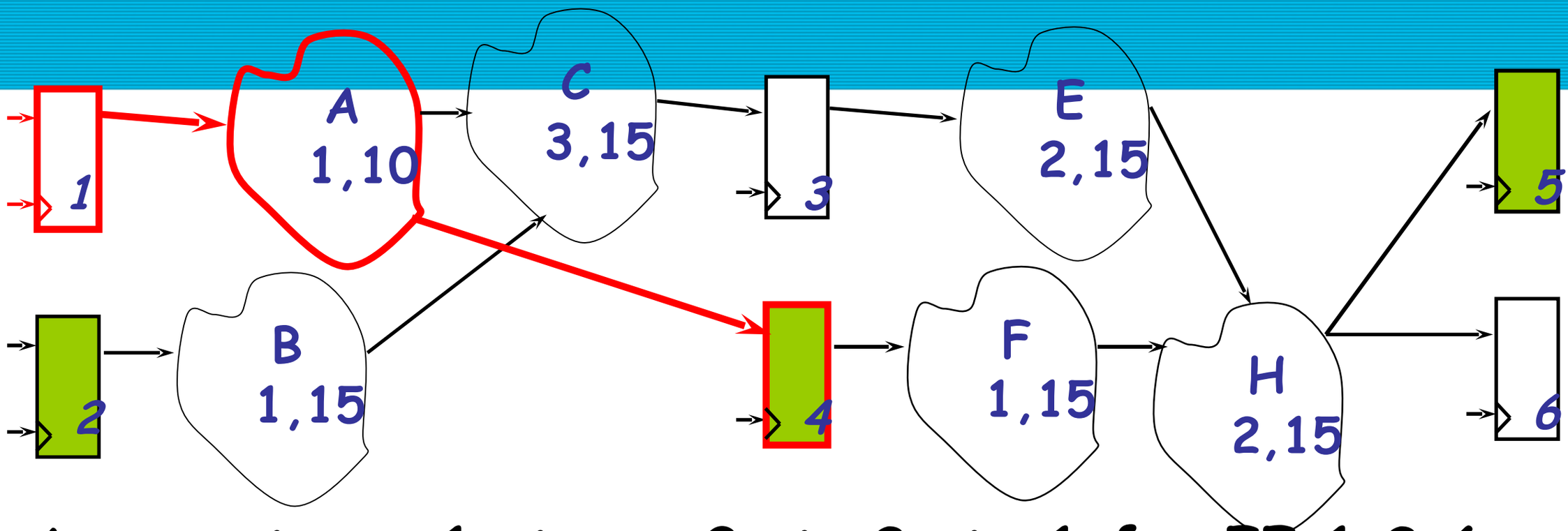
$$t_{h4} \le t_{cdC-Q1} + t_{cdA} = 2$$

$$t_{h5} \le t_{cdC-Q3} + t_{cdE} + t_{cdH} = 5$$

$$t_{h6} \le t_{cdC-Q3} + t_{cdE} + t_{cdH} = 5$$

Does this work?

Hold Times



Assume $t_{cdC-Q}=1$, $t_{pdC-Q}=2$, $t_s=3$, $t_h=1$ for FF 1,3,6

$t_{cdC-Q}=3$, $t_{pdC-Q}=4$, $t_s=4$, $t_h=3$ for FF 2,4,5

these must *all* be true

$$t_{h3} \le t_{cdC-Q1} + t_{cdA} + t_{cdC} = 5$$

$$t_{h4} \le t_{cdC-Q1} + t_{cdA} = 2$$

$$t_{h5} \le t_{cdC-Q3} + t_{cdE} + t_{cdH} = 5$$

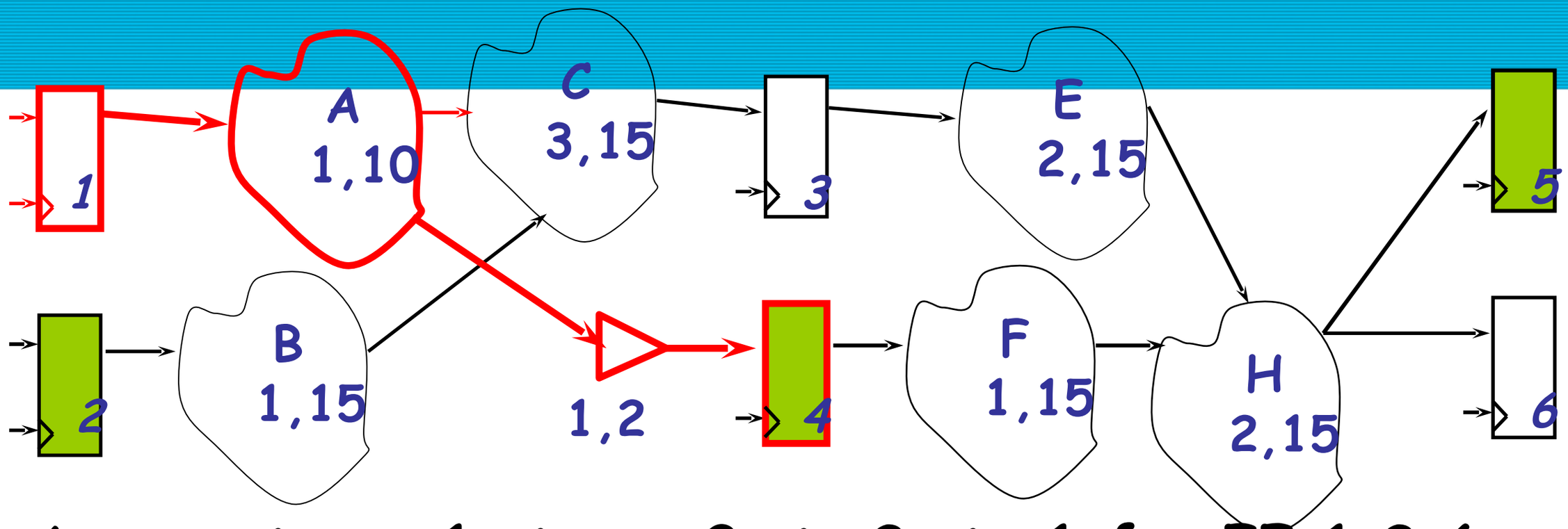
$$t_{h6} \le t_{cdC-Q3} + t_{cdE} + t_{cdH} = 5$$

NO!

$t_{h4} = 3 > 2$

DISCS

How do we fix it?



Assume $t_{cdC-Q}=1$, $t_{pdC-Q}=2$, $t_s=3$, $t_h=1$ for FF 1,3,6

$t_{cdC-Q}=3$, $t_{pdC-Q}=4$, $t_s=4$, $t_h=3$ for FF 2,4,5

these must *all* be true

$t_{h3} \le t_{cdC-Q1} + t_{cdA} + t_{cdC} = 5$

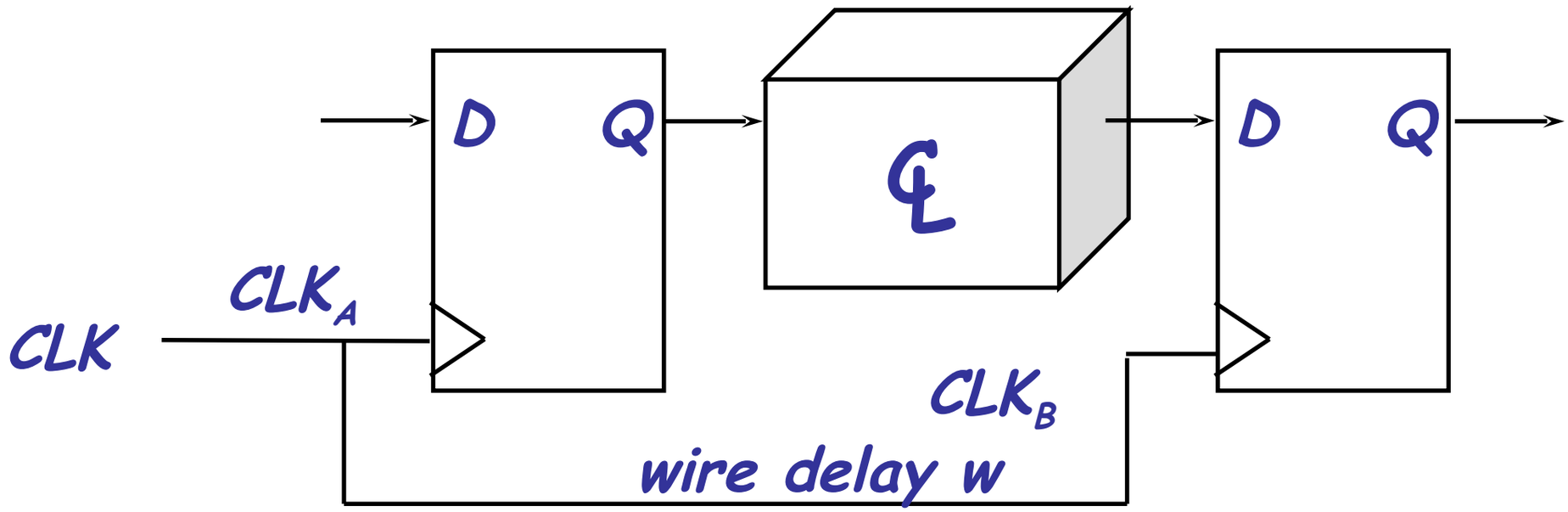
$t_{h4} \le t_{cdC-Q1} + t_{cdA} + t_{cdbuf} = 3$

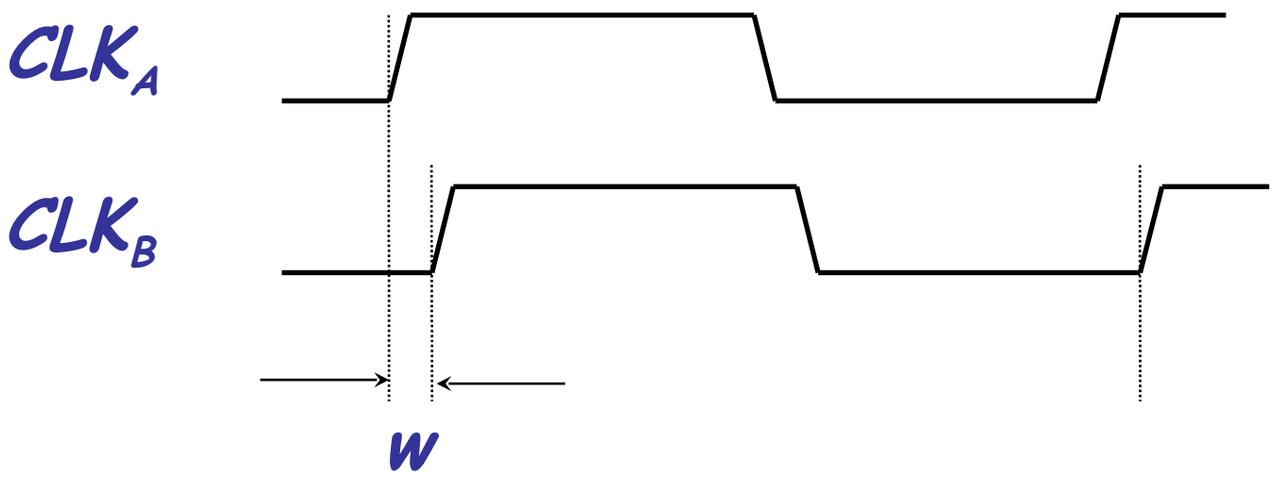
$t_{h5} \le$ [illegible]

Add a tcd delay.
(But be sure to recheck t_{pd} and t_s paths.)

DISCS

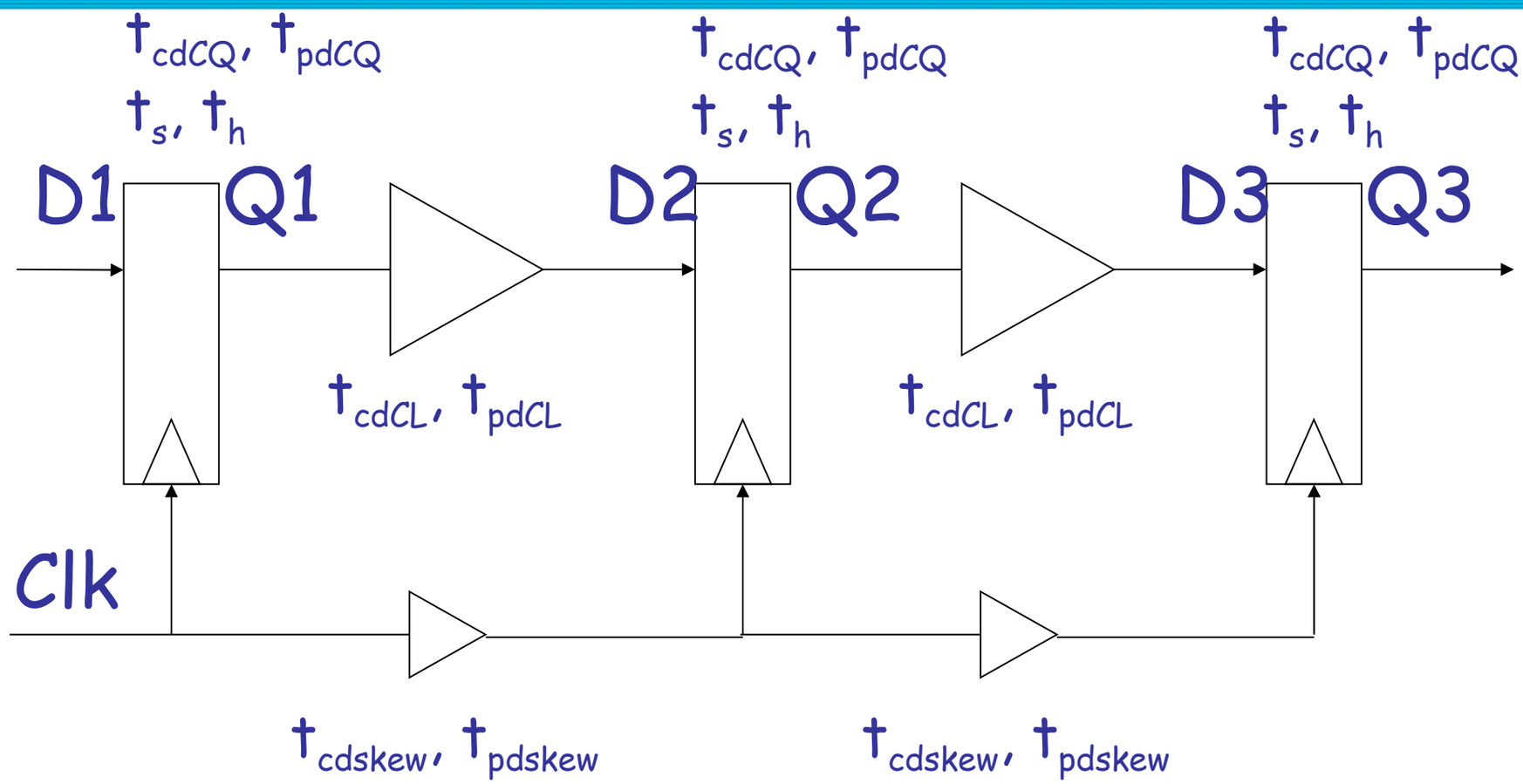
Clock Skew





Skew affects clock period and hold time constraints!

Clock Skew Example



Assume:

$t_{cdCL} = 2$

$t_{pdCL} = 4$

$t_{cdskew} = 1$

$t_{pdskew} = 2$

and

$t_{cdCQ} = 2$

$t_{pdCQ} = 3$

$t_s = 1$

$t_h = ??$

for all FF's.

$t_{clk} \geq ?$

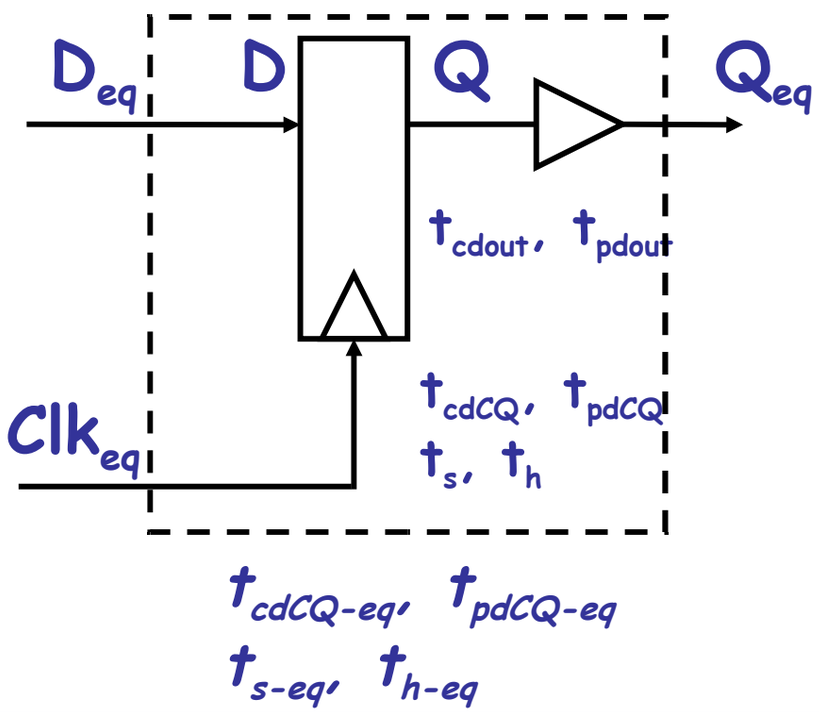
$t_h \leq ?$

How do you solve this??

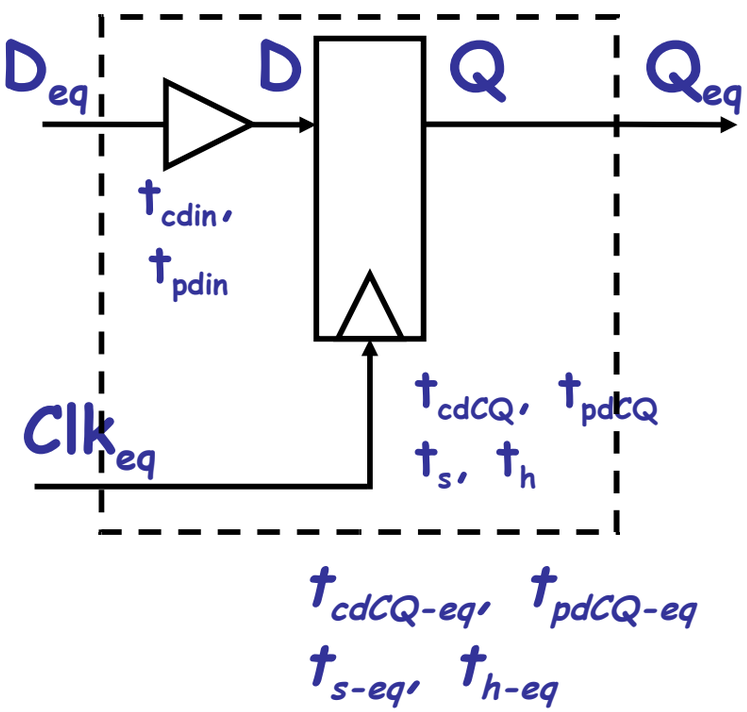
Blackboxing Technique

- Blackboxing:
  - "Hide" a delay element inside a "blackbox".
  - Treat the blackbox as a FF with new "equivalent" timing parameters, as seen from outside.
- 3 cases:

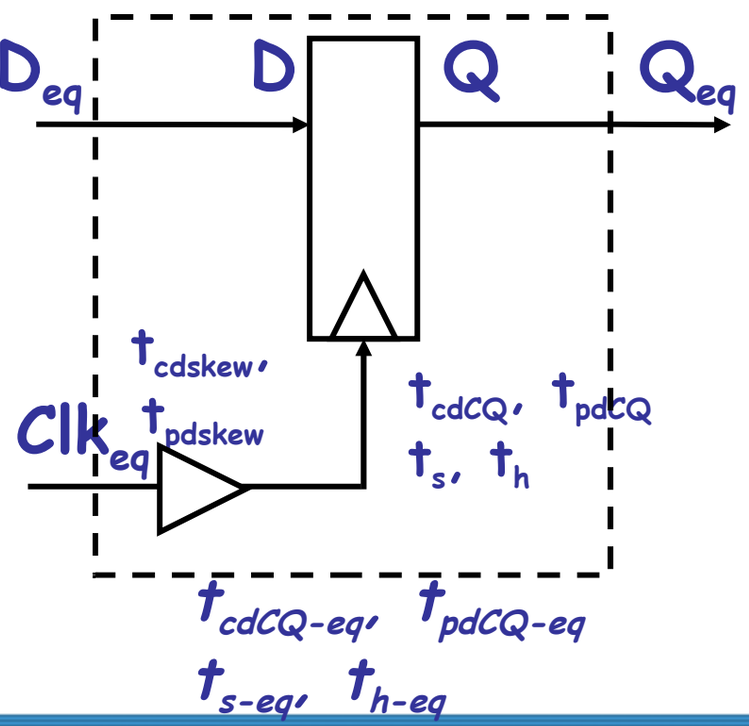
FF with delayed output



FF with delayed input

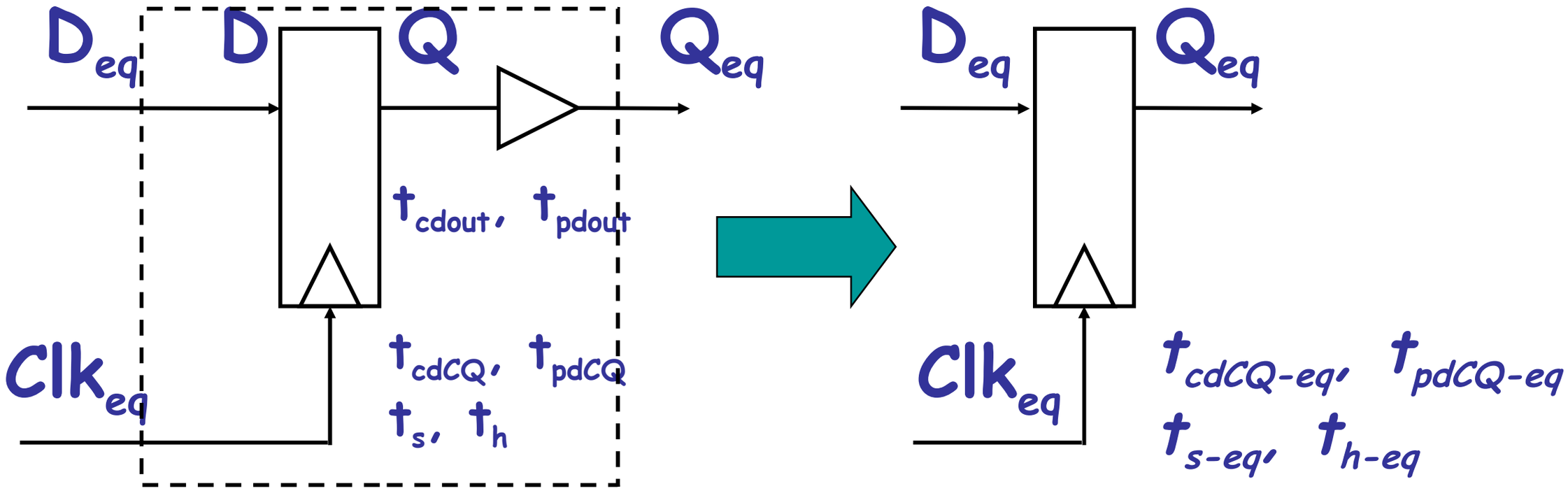


FF with clock skew

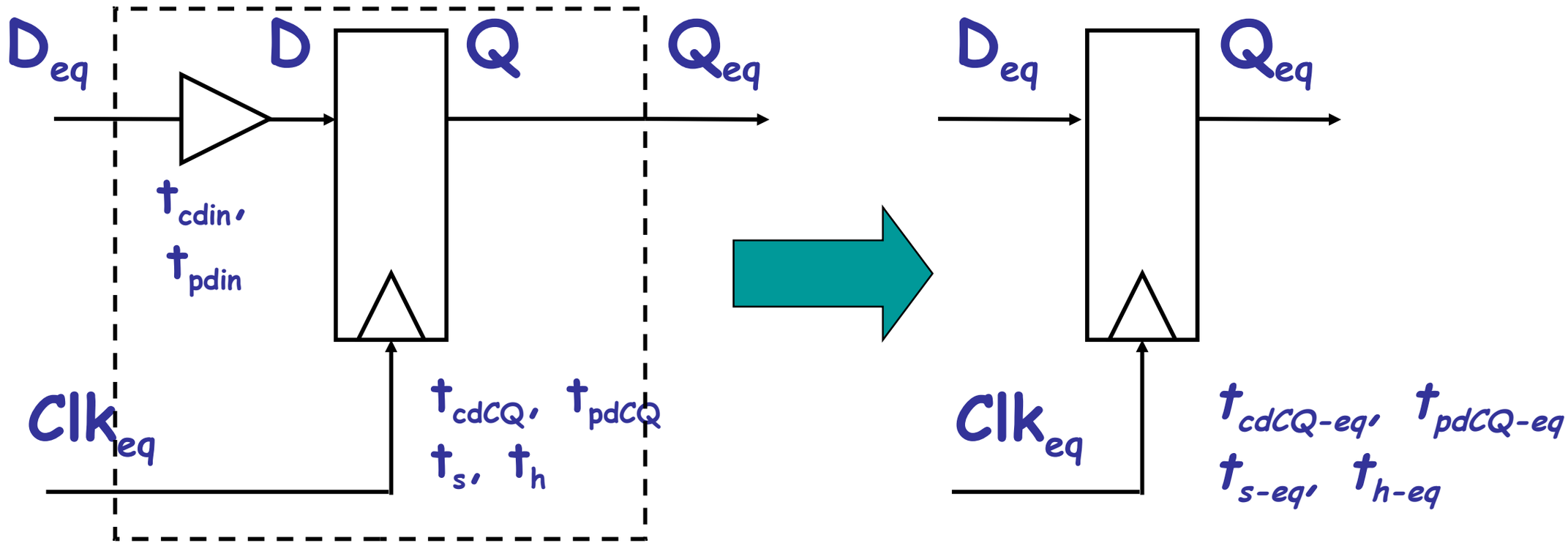


DISCS

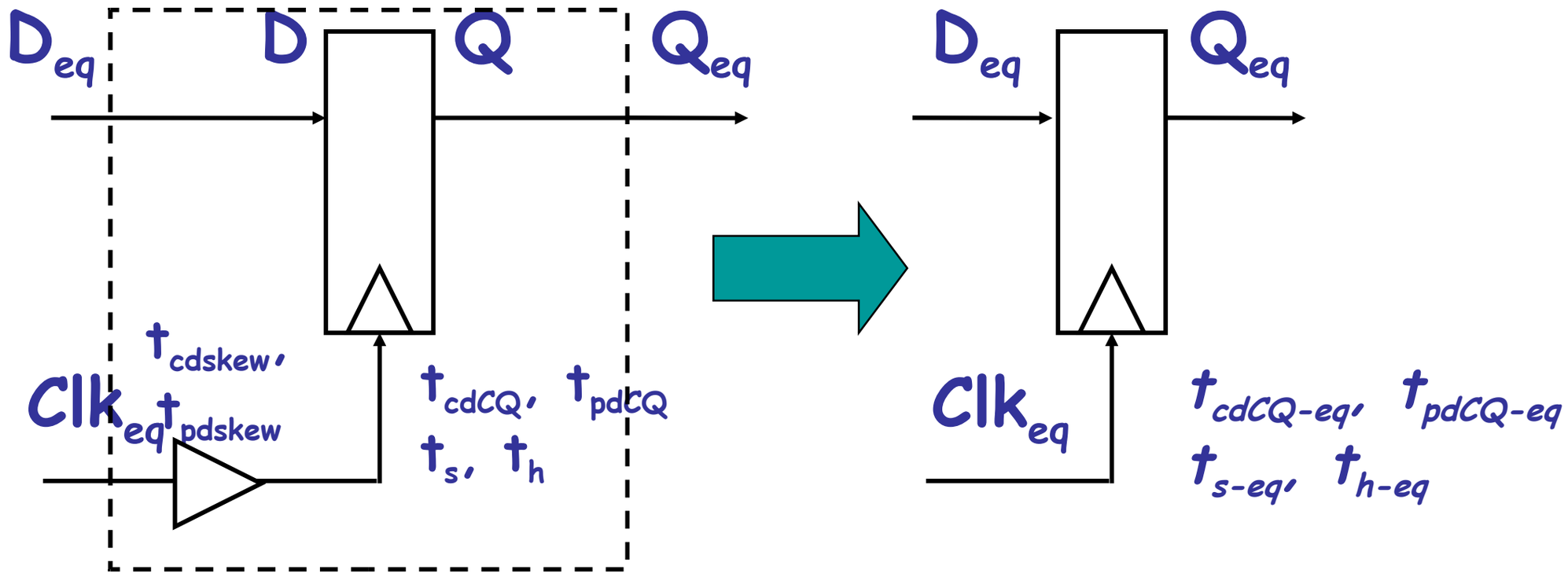
Delayed Output



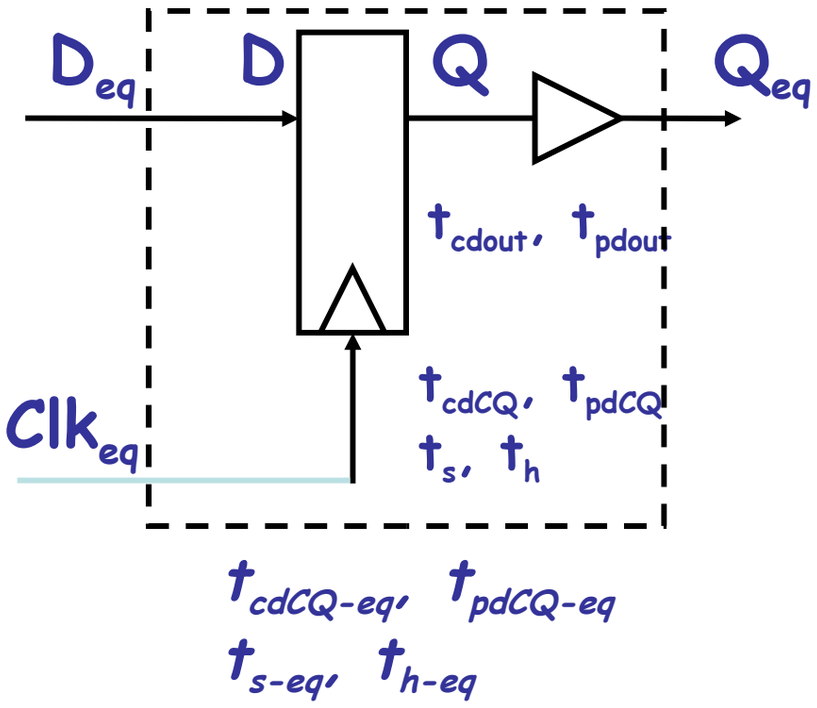
Delayed Input



Clock Skew

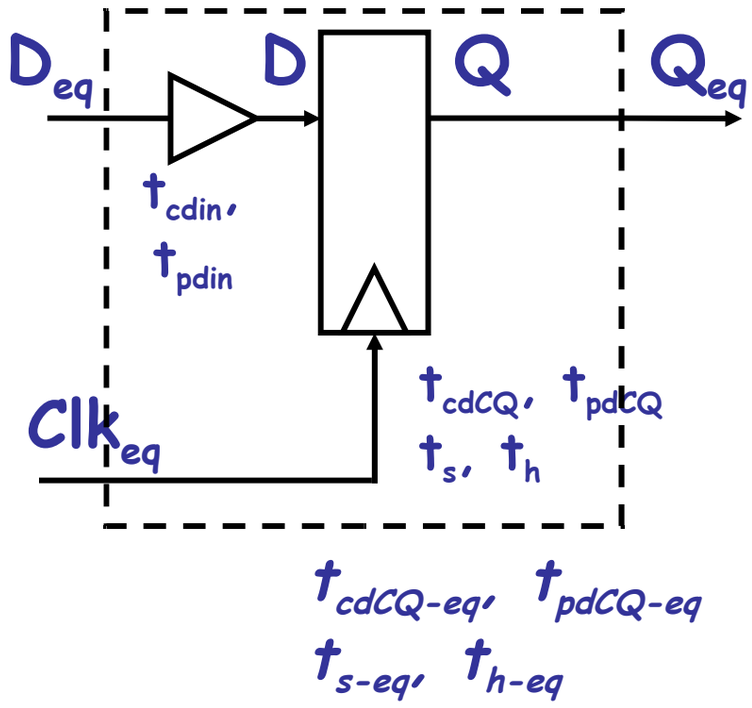

D_{eq}
D
Q
Q_{eq}
t_{cdskew}, t_{pdskew}
Clk_{eq}
t_{cdCQ}, t_{pdCQ}
t_s, t_h
D_{eq}
Q_{eq}
Clk_{eq}
$t_{cdCQ-eq}$, $t_{pdCQ-eq}$
t_{s-eq}, t_{h-eq}

FF with delayed output



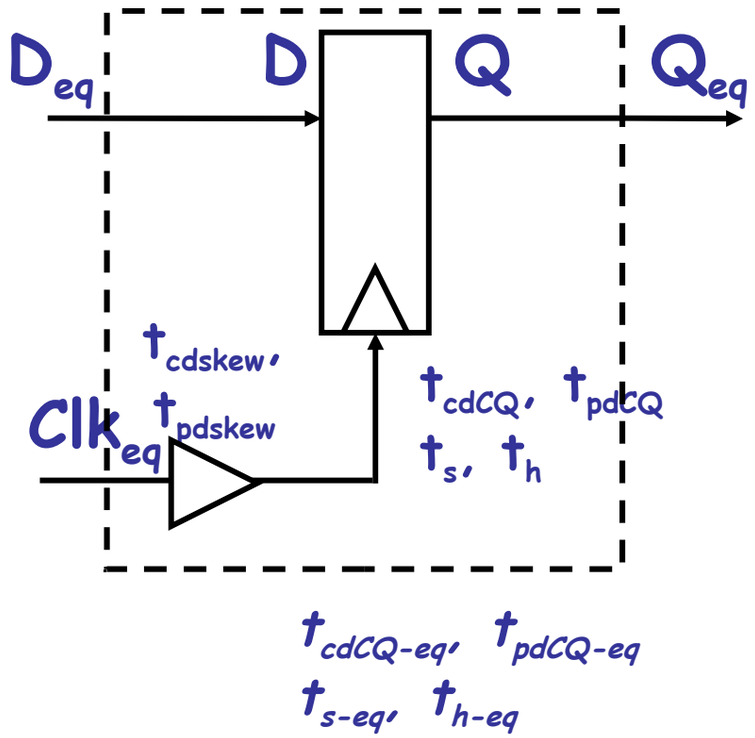
$$t_{cdCQ\text{-}eq} = t_{cdCQ} + t_{cdout}$$
$$t_{pdCQ\text{-}eq} = t_{pdCQ} + t_{pdout}$$
$$t_{s\text{-}eq} = t_s$$
$$t_{h\text{-}eq} = t_h$$

FF with delayed input



$$t_{cdCQ\text{-}eq} = t_{cdCQ}$$
$$t_{pdCQ\text{-}eq} = t_{pdCQ}$$
$$t_{s\text{-}eq} = t_s + t_{pdin}$$
$$t_{h\text{-}eq} = t_h - t_{cdin}$$

FF with clock skew



$$t_{cdCQ\text{-}eq} = t_{cdCQ} + t_{cdskew}$$
$$t_{pdCQ\text{-}eq} = t_{pdCQ} + t_{pdskew}$$
$$t_{s\text{-}eq} = t_s - t_{cdskew}$$
$$t_{h\text{-}eq} = t_h + t_{pdskew}$$

Note: Contamination delay
reduces **$t_{s\text{-}eq}$ (in clock skew case) and $t_{h\text{-}eq}$ (in delayed input)**

- **Gives us more "slack" time.**
- **But it is still offset by the other penalties, limiting its usefulness.**

Clock Skew Example

Assume:

$t_{cdCL} = 2$

$t_{pdCL} = 4$

$t_{cdskew} = 1$

$t_{pdskew} = 2$

and

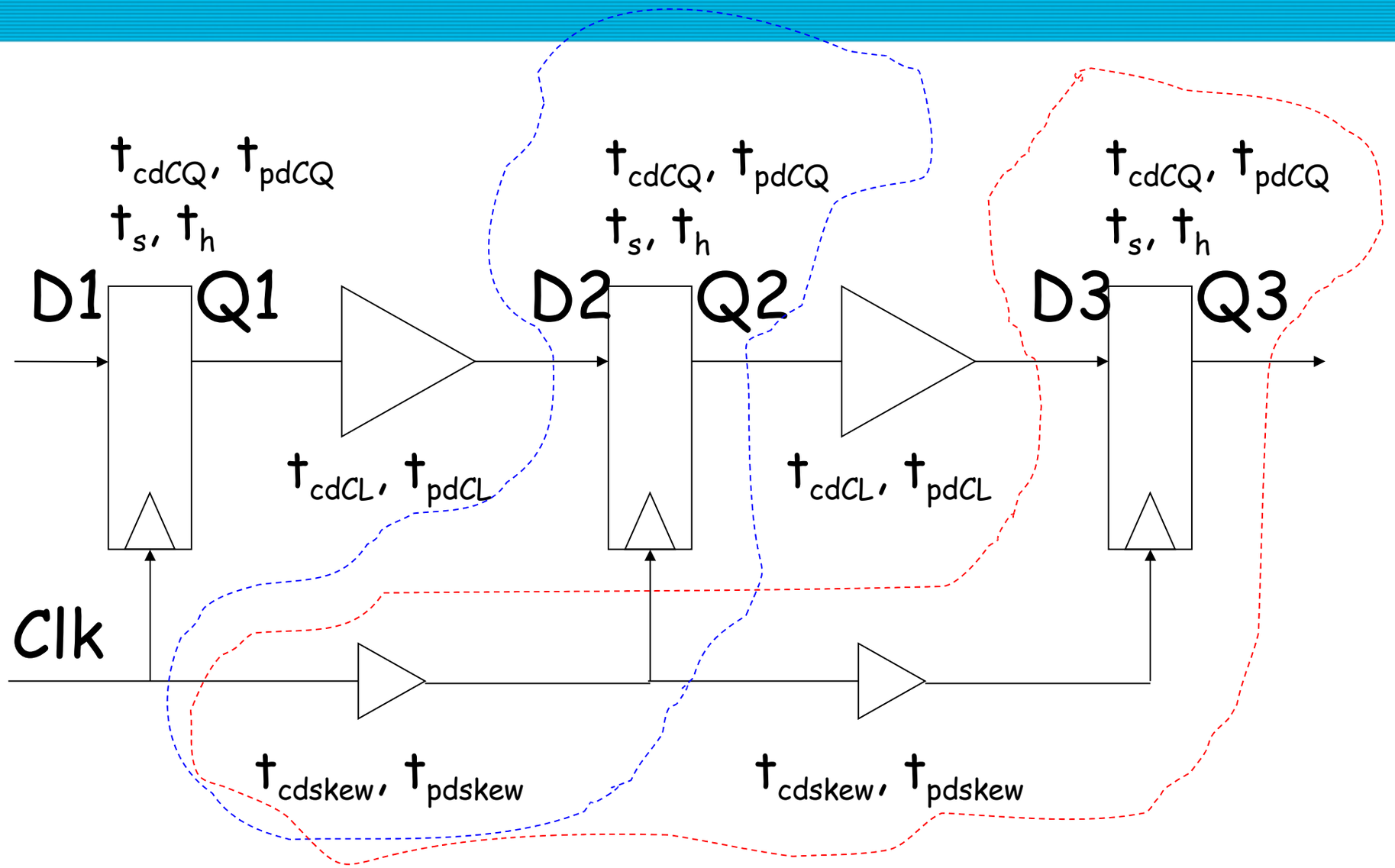
$t_{cdCQ} = 2$

$t_{pdCQ} = 3$

$t_s = 1$

$t_h = ??$

for all FF's.



Clock Skew Example

Assume:

$t_{cdCL} = 2$

$t_{pdCL} = 4$

$t_{cdskew} = 1$

$t_{pdskew} = 2$

and

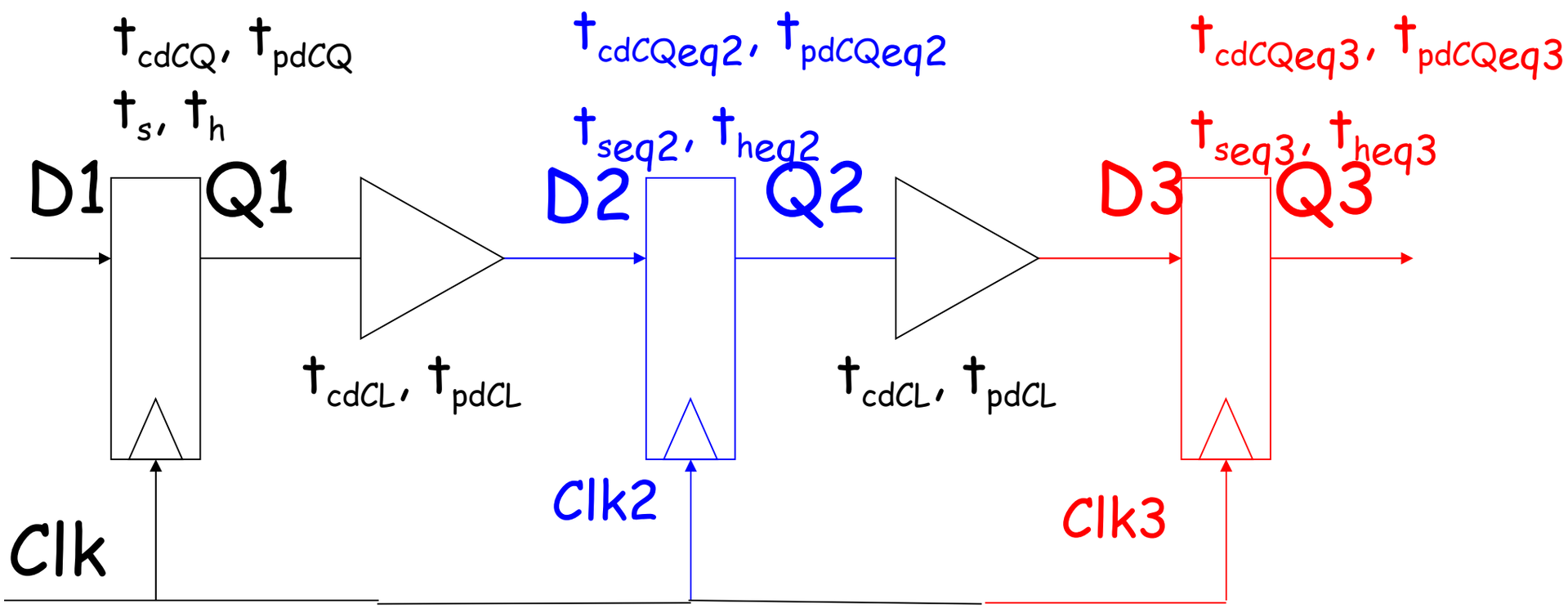
$t_{cdCQ} = 2$

$t_{pdCQ} = 3$

$t_s = 1$

$t_h = ??$

for all FF's.



Equiv. timing params

$t_{cdCQ\text{-}eq2} = t_{cdCQ} + t_{cdskew} = 3,$ $t_{pdCQ\text{-}eq2} = t_{pdCQ} + t_{pdskew} = 5$

$t_{s\text{-}eq2} = t_s - t_{cdskew} = 0,$ $t_{h\text{-}eq2} = t_h + t_{pdskew} = t_h + 2$

$t_{cdCQ\text{-}eq3} = t_{cdCQ} + 2t_{cdskew} = 4,$ $t_{pdCQ\text{-}eq3} = t_{pdCQ} + 2t_{pdskew} = 7$

$t_{s\text{-}eq3} = t_s - 2t_{cdskew} = -1,$ $t_{h\text{-}eq3} = t_h + 2t_{pdskew} = t_h + 4$

DISCS

Clock Skew Example

Assume:

$t_{cdCL} = 2$

$t_{pdCL} = 4$

$t_{cdskew} = 1$

$t_{pdskew} = 2$

and

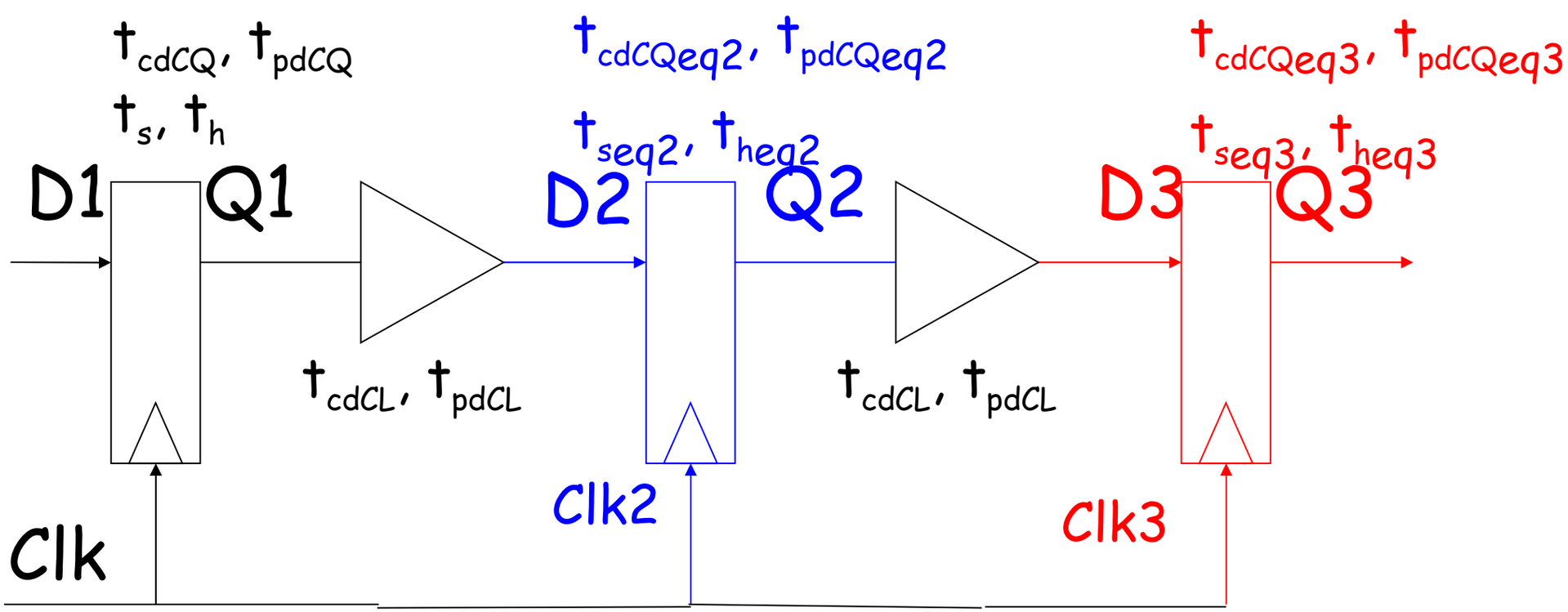
$t_{cdCQ} = 2$

$t_{pdCQ} = 3$

$t_s = 1$

$t_h = ??$

for all FF's.



Constraints

$t_{clk} >= max(t_{pdCQ} + t_{pdCL} + t_{s\text{-}eq2},$
$t_{pdCQeq2} + t_{pdCL} + t_{s\text{-}eq3})$

Equiv. timing params

$t_{cdCQ\text{-}eq2} = 3,$ $t_{pdCQ\text{-}eq2} = 5$

$t_{s\text{-}eq2} = 0,$ $t_{h\text{-}eq2} = t_h + 2$

$t_{cdCQ\text{-}eq3} = 4,$ $t_{pdCQ\text{-}eq3} = 7$

$t_{s\text{-}eq3} = -1,$ $t_{h\text{-}eq3} = t_h + 4$

$t_{heq2} <= t_{cdCQ} + t_{cdCL}$

$t_{heq3} <= t_{cdCQ\text{-}eq2} + t_{cdCL}$

DISCS

Clock Skew Example

Assume:

$t_{cdCL} = 2$

$t_{pdCL} = 4$

$t_{cdskew} = 1$

$t_{pdskew} = 2$

and

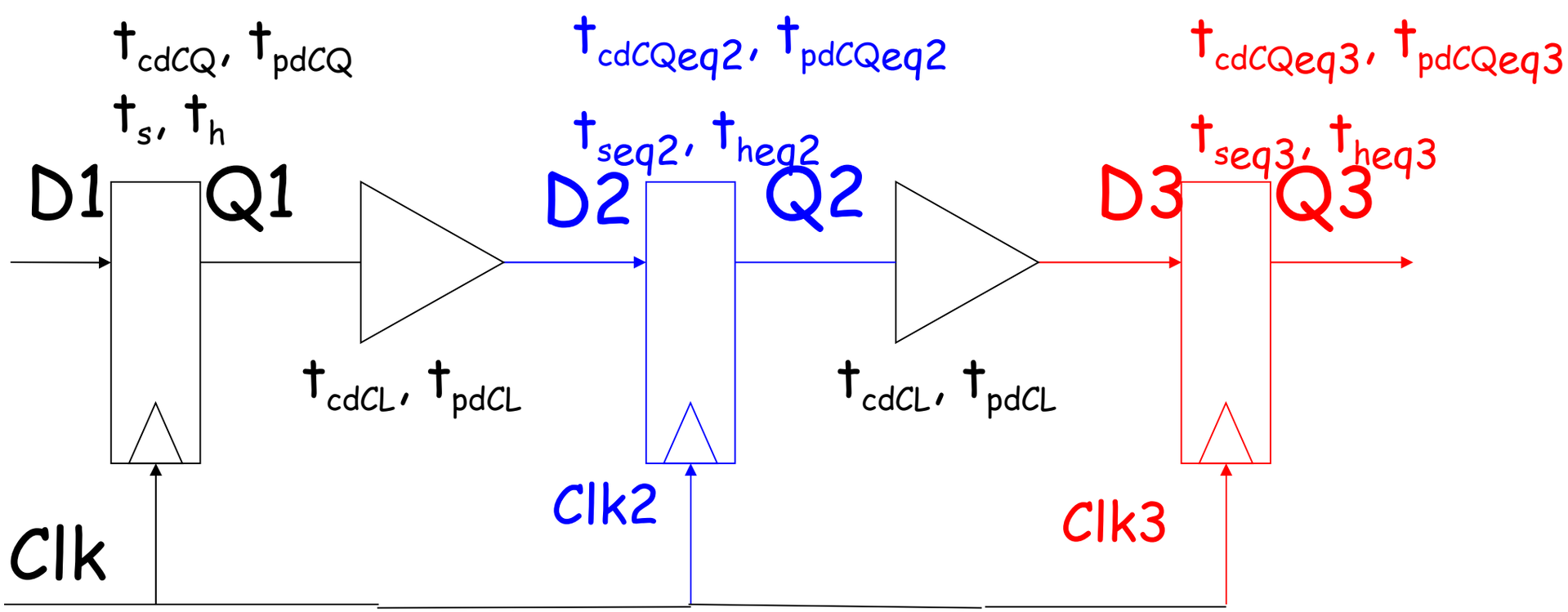
$t_{cdCQ} = 2$

$t_{pdCQ} = 3$

$t_s = 1$

$t_h = ??$

for all FF's.



Equiv. timing params

$t_{cdCQ\text{-}eq2} = 3$, $t_{pdCQ\text{-}eq2} = 5$

$t_{s\text{-}eq2} = 0$, $t_{h\text{-}eq2} = t_h + 2$

$t_{cdCQ\text{-}eq3} = 4$, $t_{pdCQ\text{-}eq3} = 7$

$t_{s\text{-}eq3} = -1$, $t_{h\text{-}eq3} = t_h + 4$

Constraints

$t_{clk} >= max(3 + 4 + 0,$
$5 + 4 + -1)$

$t_h + 2 <= 2 + 2$

$t_h + 4 <= 3 + 2$

DISCS

Clock Skew Example

Assume:

$t_{cdCL} = 2$

$t_{pdCL} = 4$

$t_{cdskew} = 1$

$t_{pdskew} = 2$

and

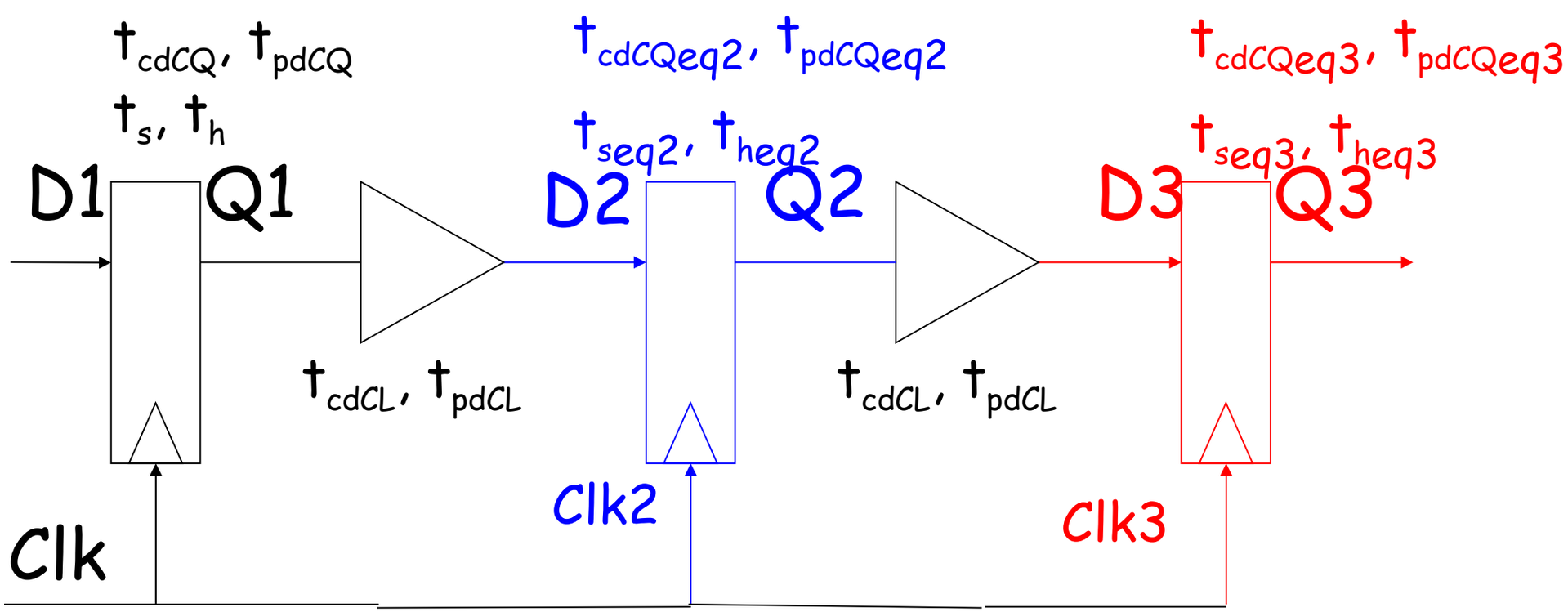
$t_{cdCQ} = 2$

$t_{pdCQ} = 3$

$t_s = 1$

$t_h = ??$

for all FF's.



Equiv. timing params

$t_{cdCQ\text{-}eq2} = 3$, $t_{pdCQ\text{-}eq2} = 5$

$t_{s\text{-}eq2} = 0$, $t_{h\text{-}eq2} = t_h + 2$

$t_{cdCQ\text{-}eq3} = 4$, $t_{pdCQ\text{-}eq3} = 7$

$t_{s\text{-}eq3} = -1$, $t_{h\text{-}eq3} = t_h + 4$

Constraints

$t_{clk} >= max(7, 8) = 8$

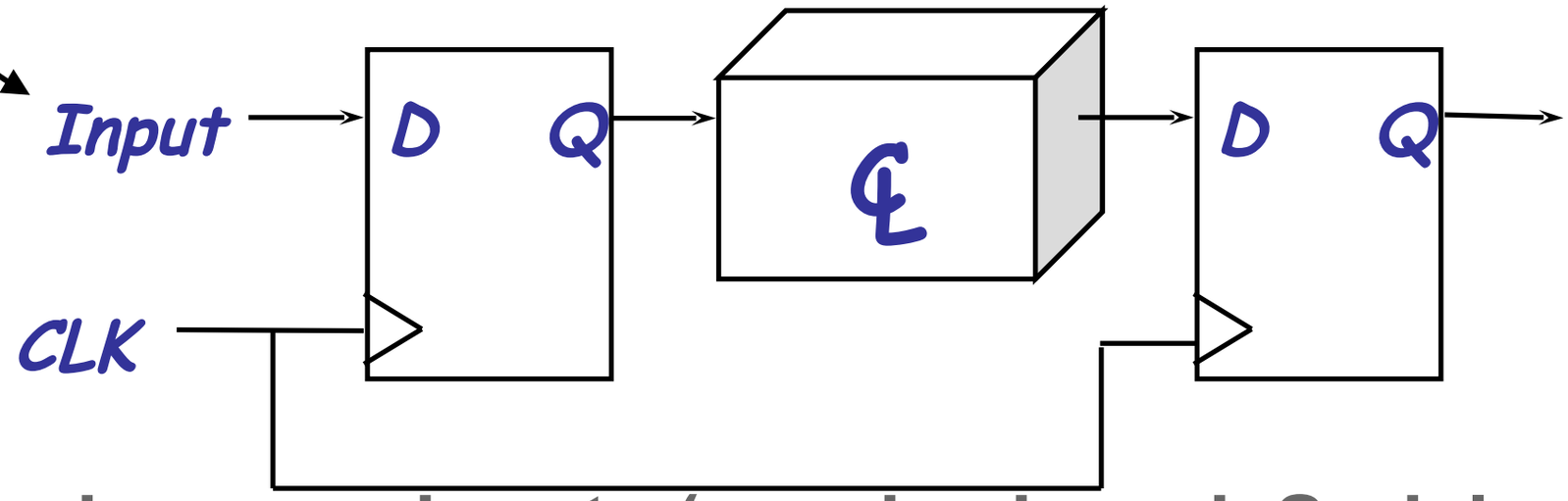
$t_h <= 4 - 2 = 2$

$t_h <= 5 - 4 = 1$

The <u>original</u> FF must have at most this t_h

What About Asynchronous Inputs?

Does input satisfy setup and hold time constraints?
If not, we can still get metastability!



- Need to sample asynchronous inputs (e.g., keyboard, Serial Port, etc.) reliably.
- Usually, feed asynchronous input through a FF with a VERY sharp VTC slope to minimize metastability.
- Or, sample inputs at a slower speed (relative to the actual input device's sampling rate, or perhaps in time with synchronous inputs by using separate registers) to shrink fraction of sampling window.
  - Translation: Input (or a snapshot of it) is fed once in a while, usually in time with CLK, not every time it changes. Video gamers of the past have experienced the drawback of this.