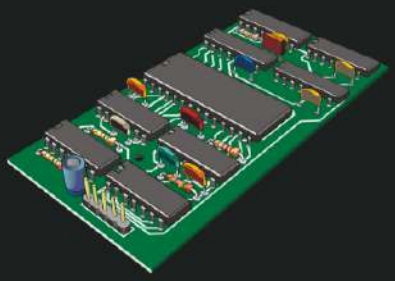


STATIC TIMING ANALYSIS

SHORT NOTES

By

HARSHIT GUPTA

papergrid

Date: / /

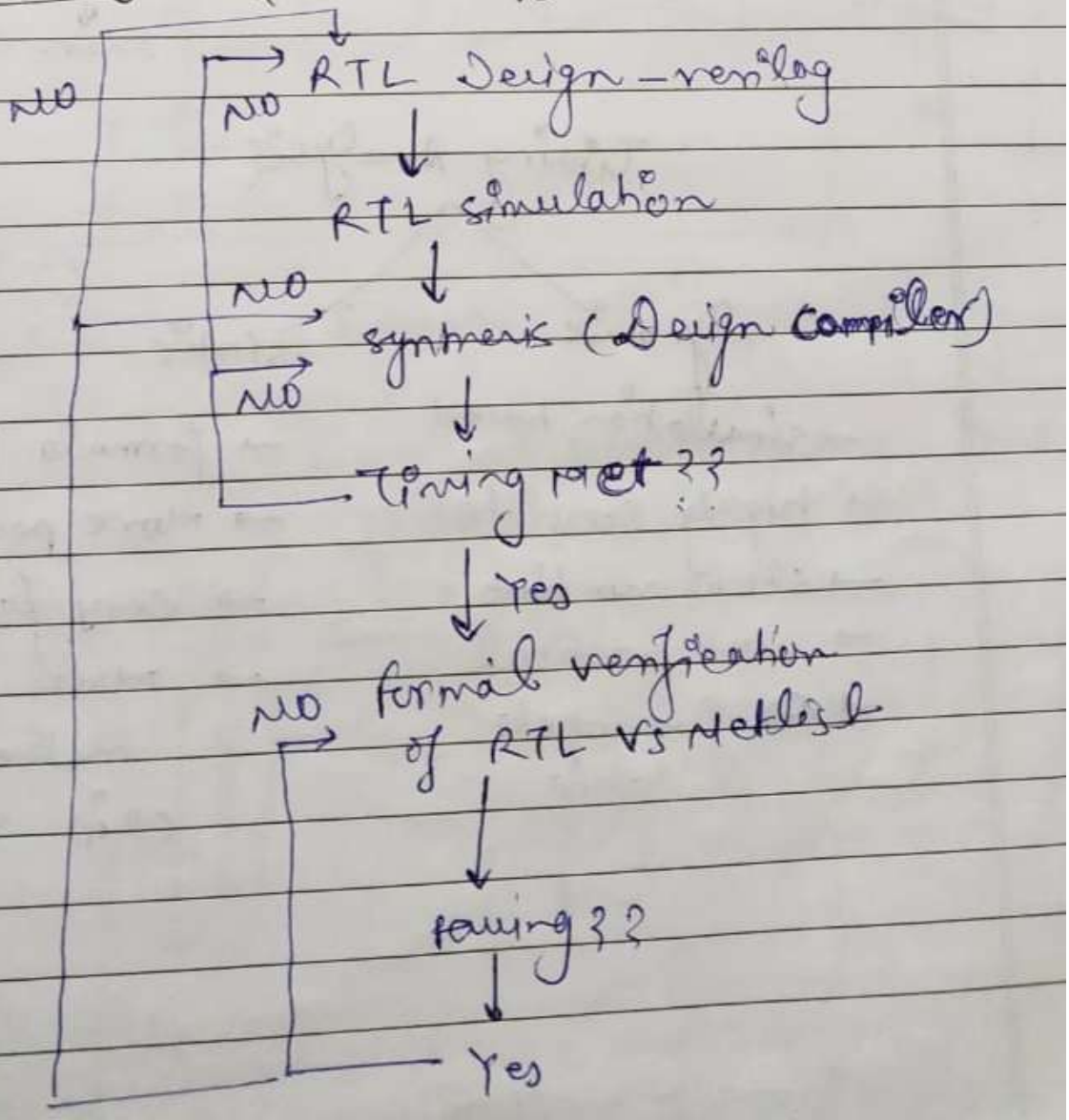
(Static timing Analysis)

Timing Analysis :- methodical analysis of a digital ckt to determine if the timing constraints imposed by components or interfaces are met.

why timing Analysis :-

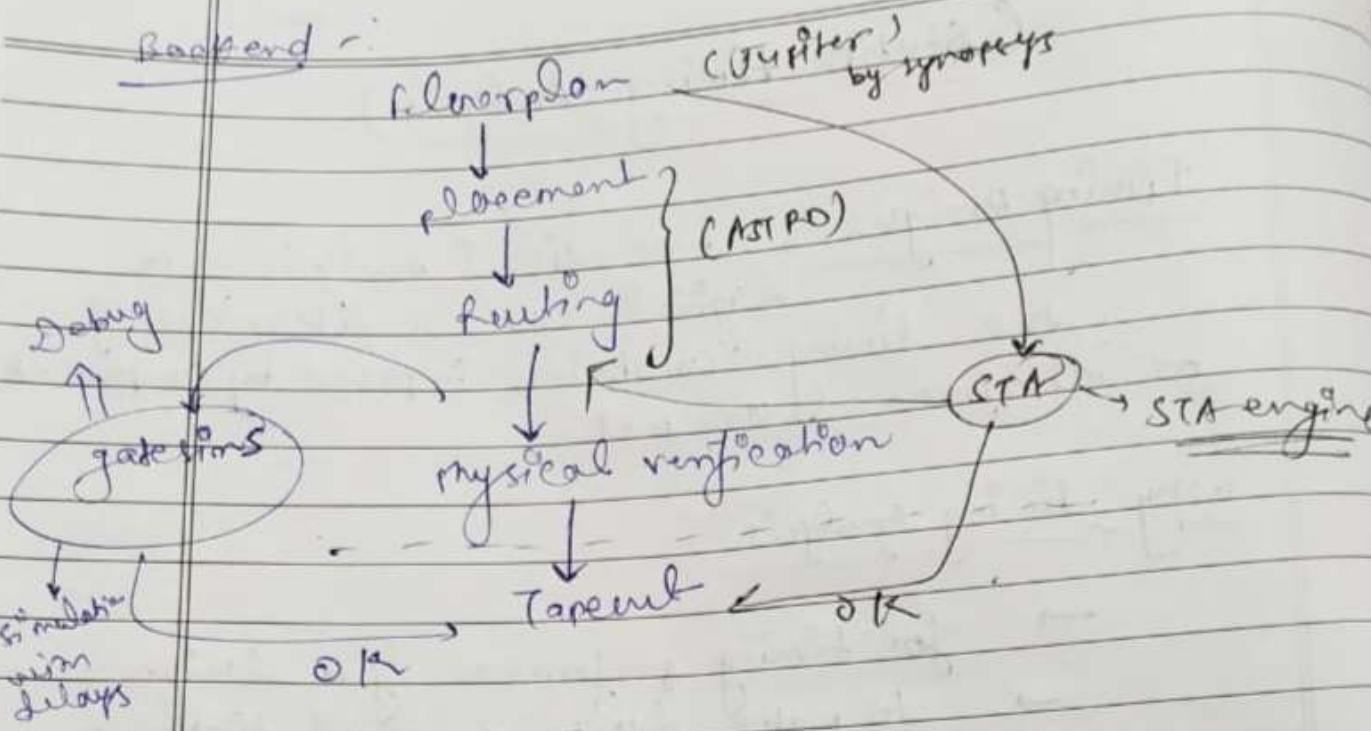
→ for timing performance of a design
→ to make design free of timing violations.
→ to analyse the effect of interconnect traces as technology shrinks.

ASIC flow methodology :-



Date: / /

Backend :-



Priority :-

DRC > Timing > Area/Power

(Design rule check)

↳ checking more than formal verification

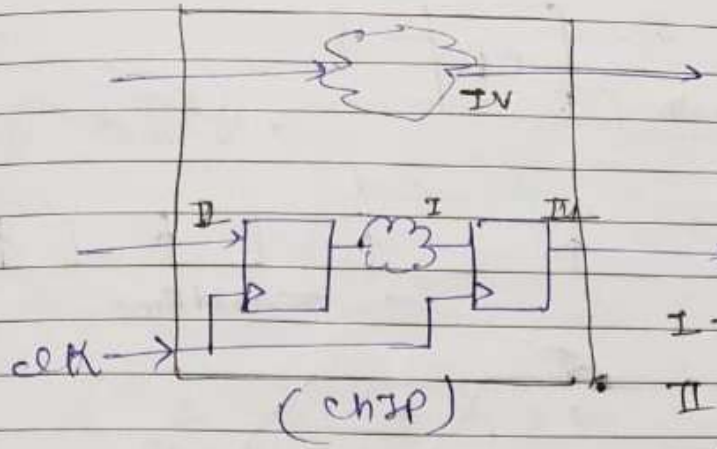
↓

then backend.

Timing Analysis

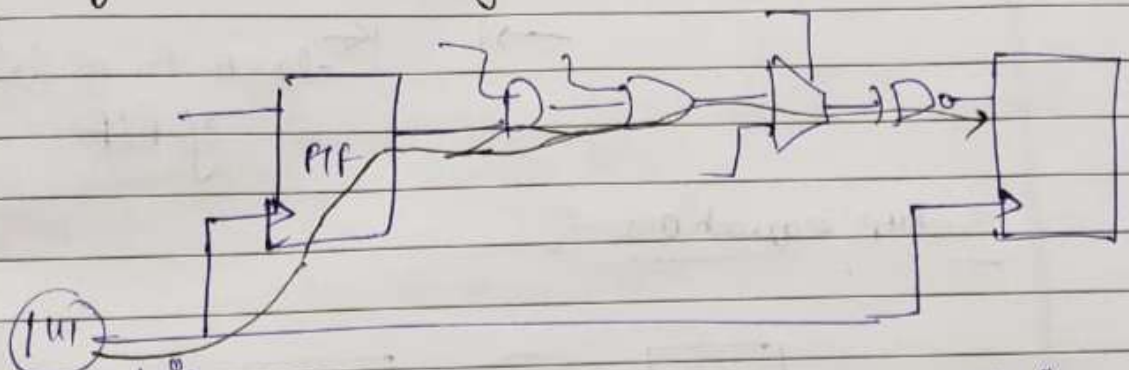
Dynamic	Static
→ simulation based	→ formula based
→ highly accurate	→ more pessimistic
→ slow runtime	→ very fast
→ not practical for whole chip.	→ most popular method for chip sign-off.

4 types of paths in a chip :-



I → Reg to Reg
II → i/p to Reg
III → Reg to o/p
IV → Pure Combinational i/p to o/p

A Typical path (Reg to Reg) :-

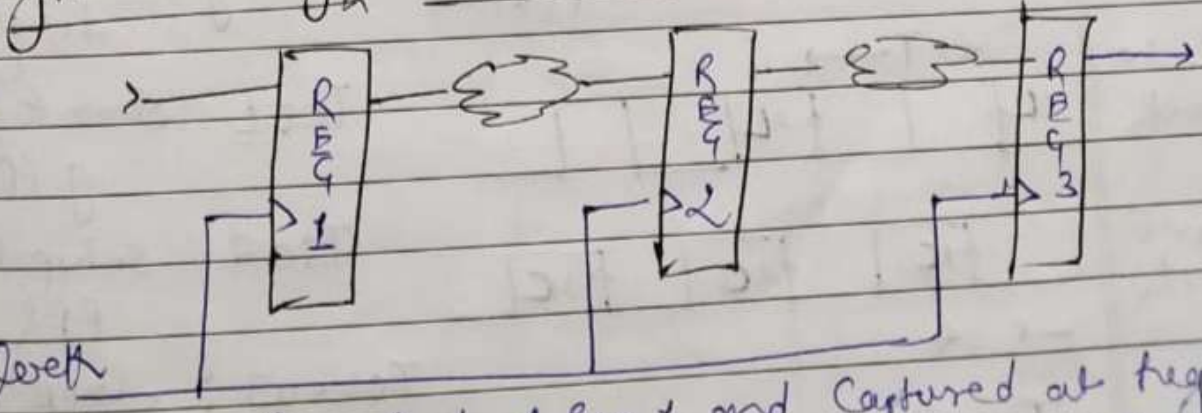


Any path meant for timing -

→ starts at source of the clock.
→ ends at D pin of FF.

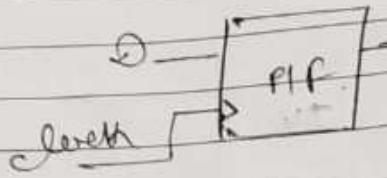
Typical dataflow in a chip :-

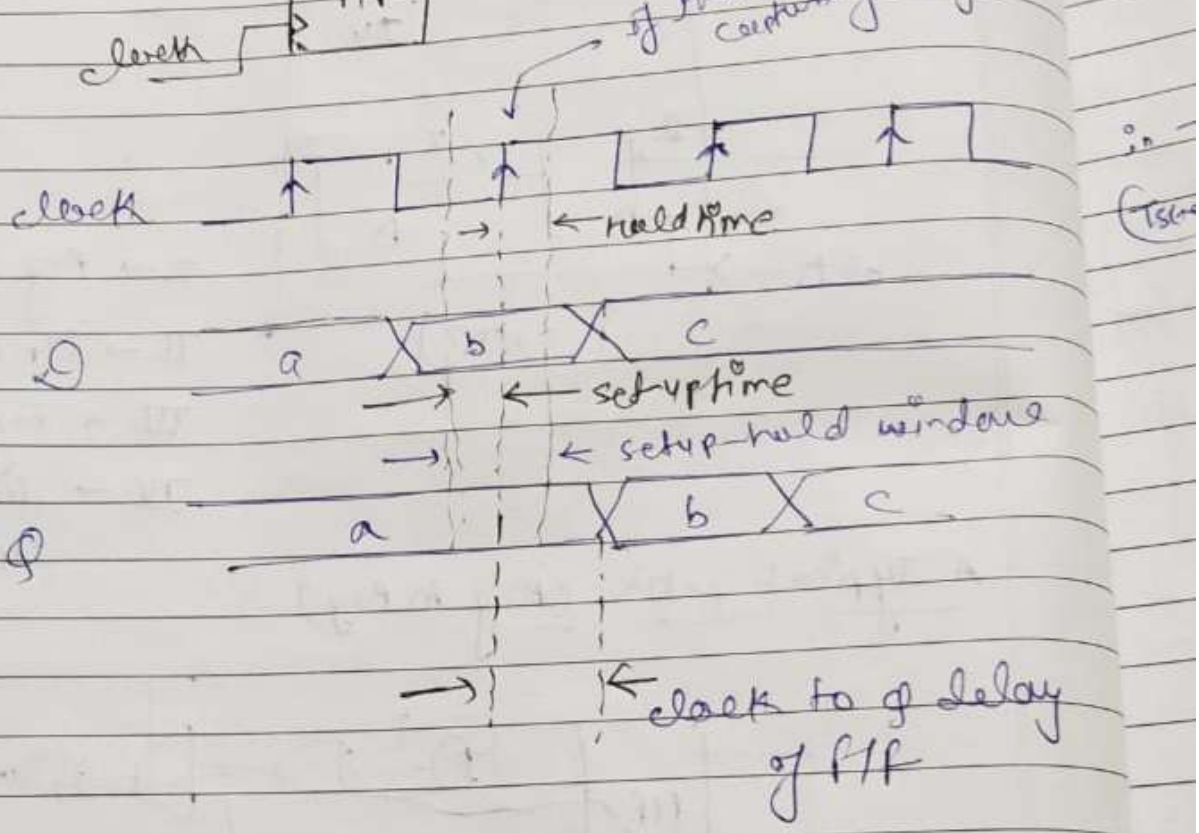
Data Transfer from Reg to another is called Reg Transfer level



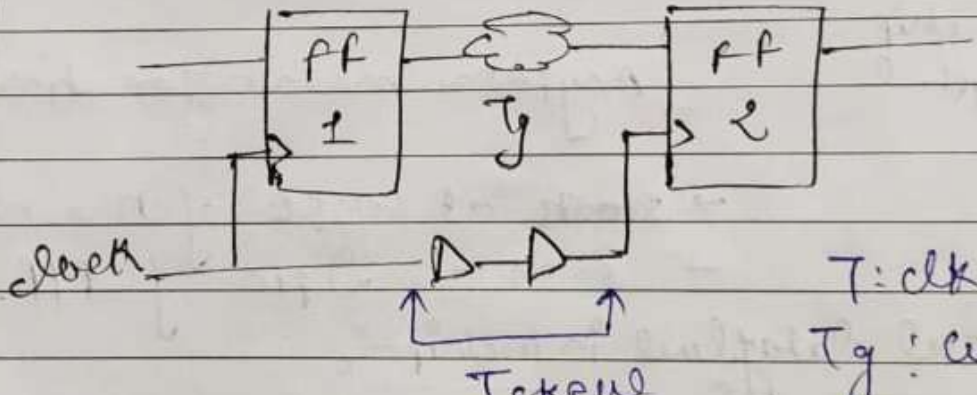
→ Data launched at Reg 1 and Captured at Reg 2
→ Data launched at Reg 2 and Captured at Reg 3

Setup & hold time of a flip flop

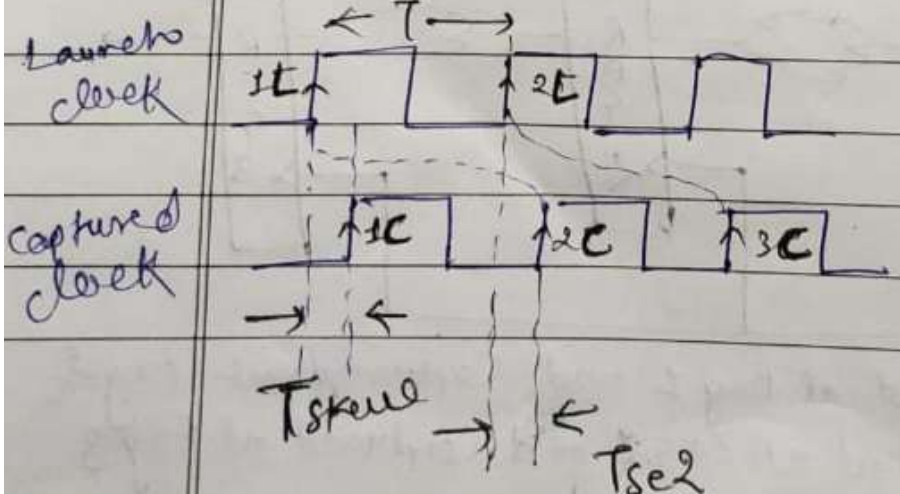




Setup equation



T: clk period
Tg: combinational delay including delay of N/W
Tcq1: clock to Q delay of FF1
Tse2: setup time of FF2
Tskew: clock skew includes the wire delay



critical path
↳ path having max^{m} delay

papergrid
Date: / /

1L launched and captured at 2C
2L " " " " 3C

$$T_{cq1(max)} + T_{g(max)} + T_{seq} \leq T + T_{skew(min)}$$ → say x

$$T \geq (T_{cq1} + T_g + T_{seq} - T_{skew})$$

∴ $-T_{skew}$ ↓ $(T_{skew})_{max}$

This eqn has to be valid for the critical path also

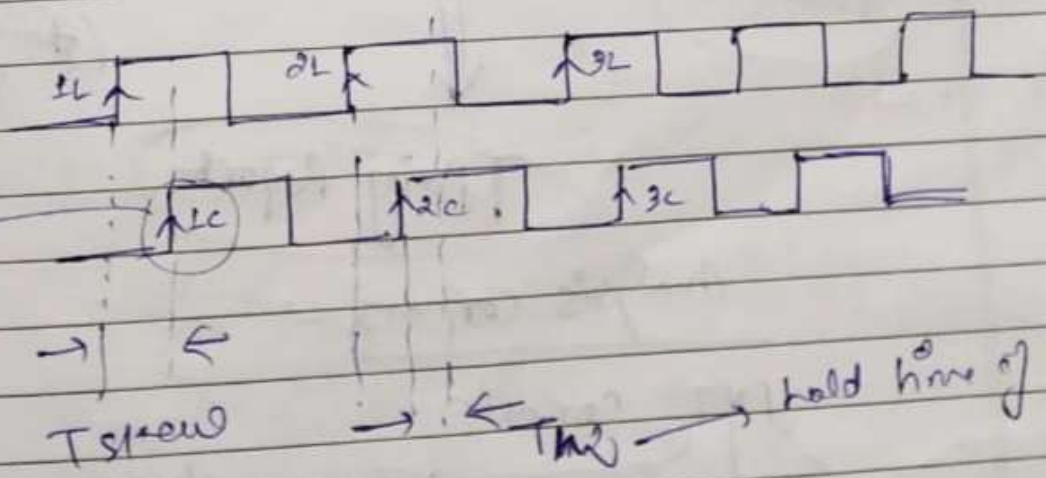
T of system $\geq x_{critical}$

$$f_{system} \leq \frac{1}{(x)_{critical}}$$

$$f_{system(max)} = \frac{1}{(x)_{critical}}$$

⇒ setup time governs the clock freq but hold time (limits) doesn't.

Hold equation



→ During capturing by the edge 2C, the 2nd data is already launched by the edge 2L
This 2nd launched data shouldn't affect the hold time requirement of FF2

→ why hold time path delay shorter than setup time path?
There is always the option of lowering max clk freq
So design may work in lab if you are failing setup
by a small amount but if hold violations, design will most likely

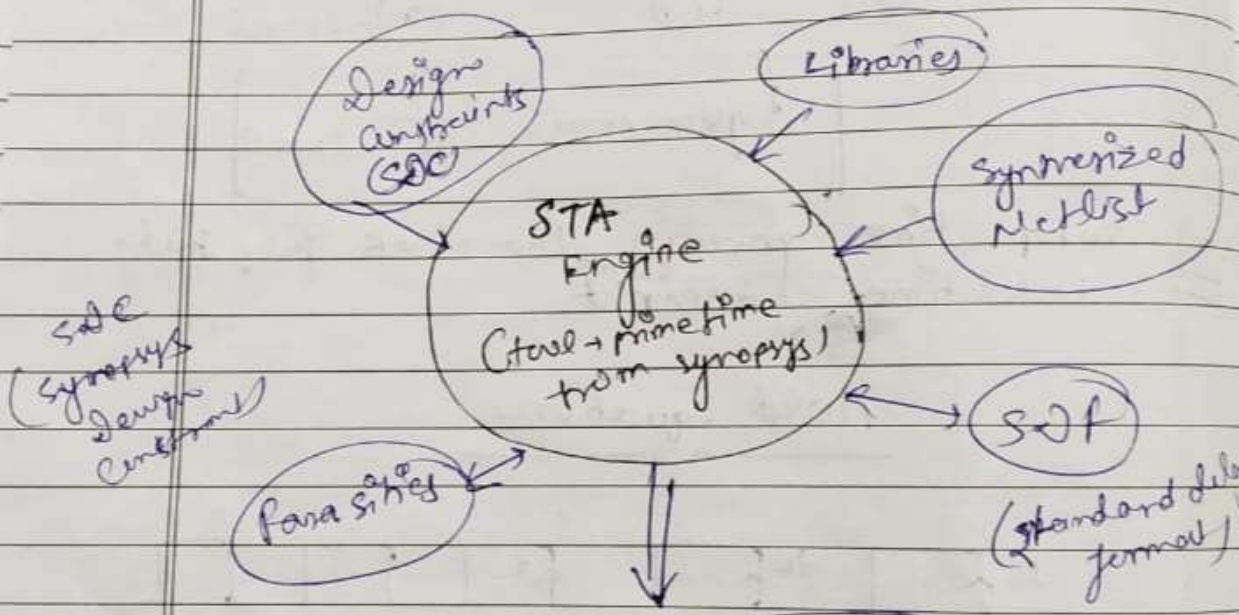
$$T_{cq1(min)} + T_{g(min)} \geq T_{skew(max)} + T_{h2}$$

in register
$-T_{skew(min)}$

→ +ve T_{skew} helps to meet setup time but increases the chance of hold violation.

→ There is no T_{clk} in hold time eqn, so chip can have range of operating freq.
(0 to f_{max})

but if T_{clk} in hold time eqn, particular range to be allowed to chip of freq.



Analysis corners :-

PVT Corner

Process → modelled by means of randomness

Voltage → as voltage ↑, delay ↓

Temperature → as Temp ↑, delay ↑

(But process remains same for the analysis)

→ in order to make our chip to work in all possible conditions, we simulate it at diff corners of process, voltage and temp which is easy type of industrial case.

slow corner	typical corner	fast corner
1.6V, 125°C (for default)	1.8V	2V, 0°C
70°C, (general)	27°C	90°C default
* more prone to have setup violations		* more prone to have HOLD violations

temp increases resistance ↑ low vtg increases delay (both reducing performance)

→ static timing checks are done at the worst combination of PVT (slow corner) and best combinations of PVT (fast corner)

Design constraints → equally valid for synthesis

- clock constraints
- non clock constraints
  - Input-output constraints
  - Exceptions
    - false path
    - multi cycle path etc.

false path :-

→ NOT a valid path for timing.

→ You need not worry to meet setup and Hold constraints for this path.

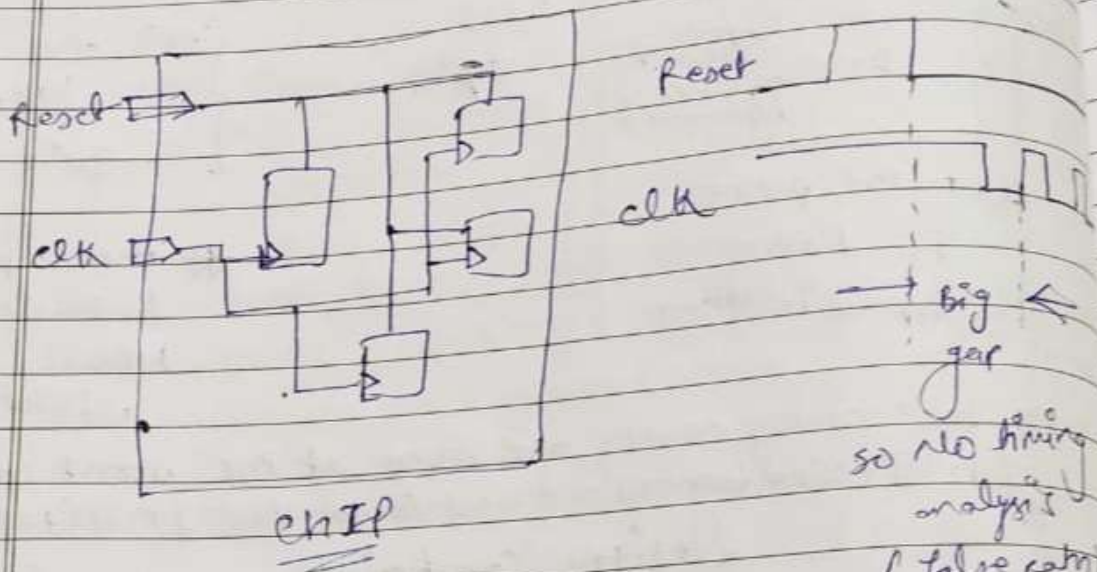
ex :- (i) all asynchronous paths (clk1 - clk2 are independent)

(ii) static paths (you need to decide which static path to take false path)

(iii) non-functional paths.

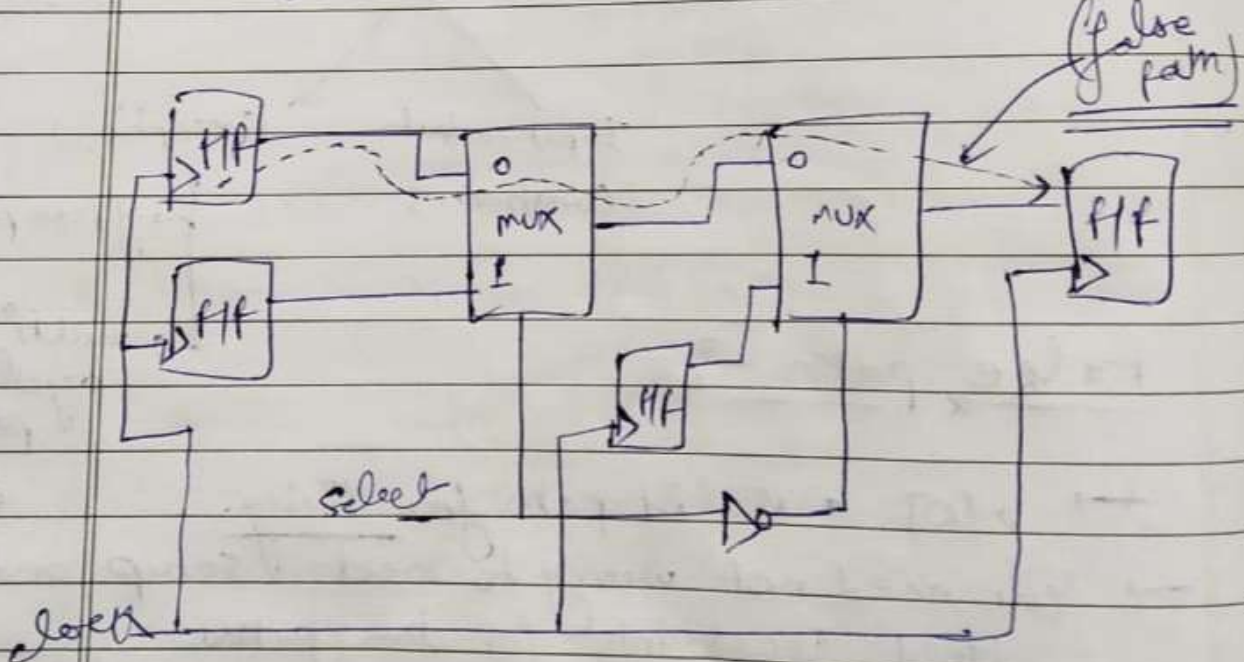
Date: / /

Static path :-



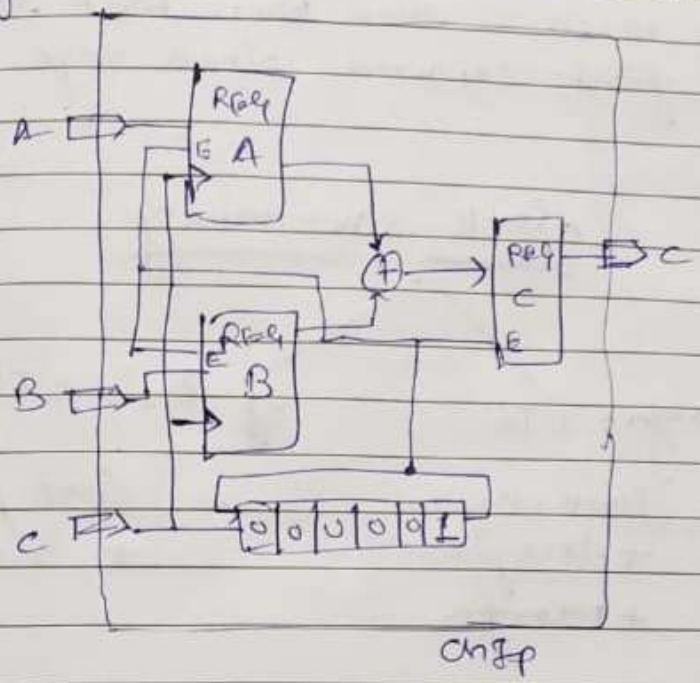
→ So declare all paths starting at Reset pin of the chip as false paths

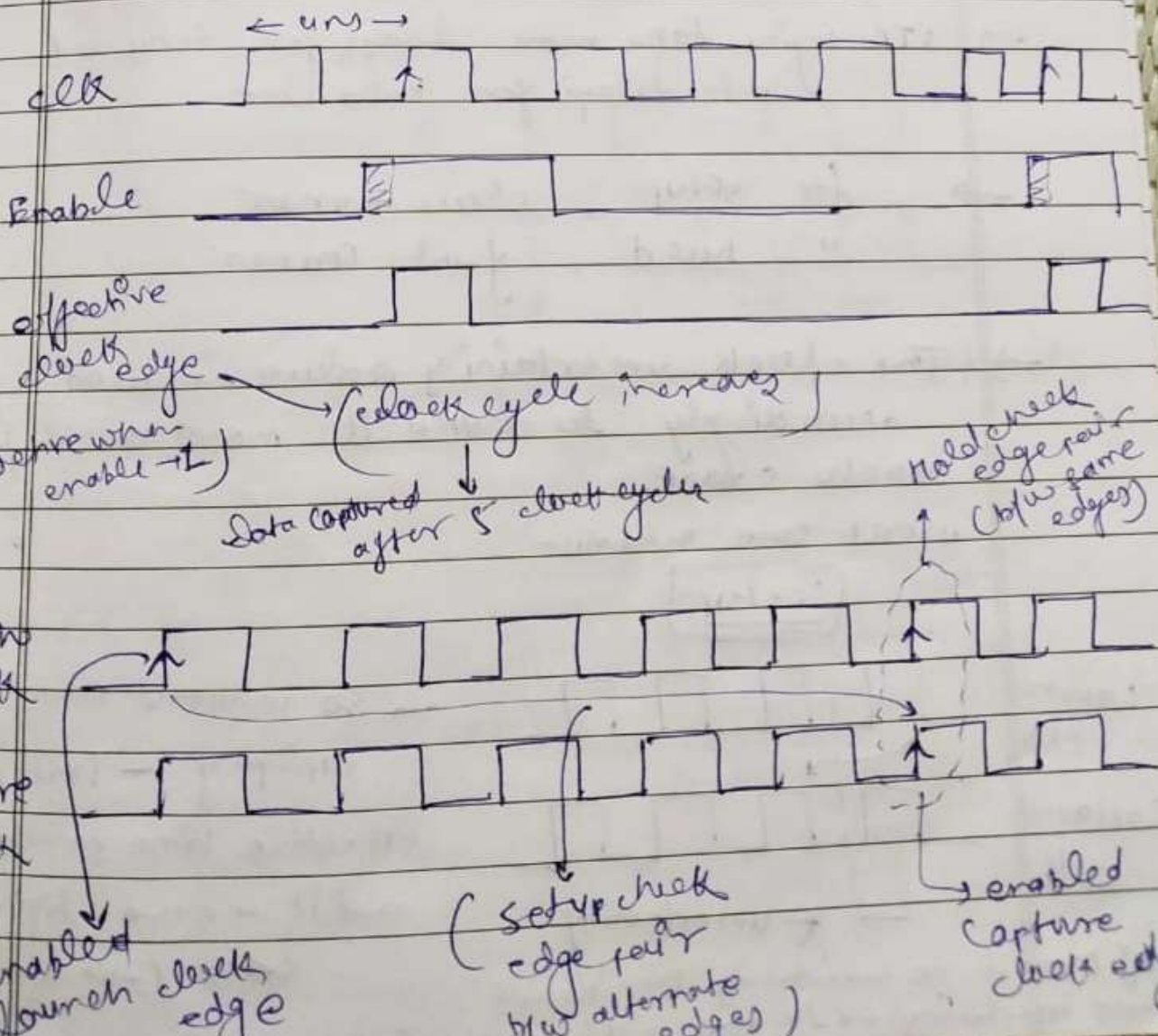
Non-functional path :-



→ Tool does not analyse by seeing the exclusiveness of the select signal, so you need to explicitly declare the false path.

Multi cycle path :- path in which data launched from one flop is allowed to take more than one clock cycle to reach the ~~data~~ destination flop.





→ Hold check is done b/w one first captured data edge and second launched edge data.

→ Setup check is done b/w first launched and first captured clock edge.

CTS → clock tree synthesis

clock uncertainity

before CTS
= clock skew + clock jitter + margin

post CTS
= clock jitter + margin

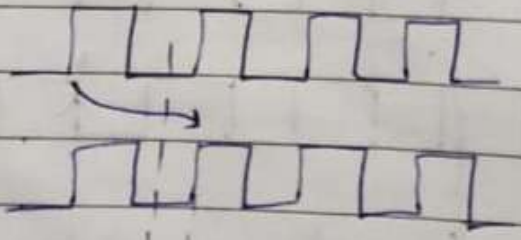
→ STA engine take max delays for setup and min. delays for hold time

→ for setup, slow corner
" hold, fast corner

→ The clock uncertainity value is used accordingly to make it more worse for each case.

worst case scenario-

setup



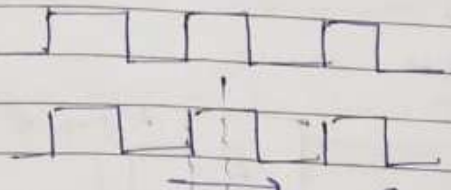
→ So in worst case we applying - This is effective time period and it is more tight for setup (-ve skew)

→ If there is no uncertainity then we will add uncertainity and the edge will be shifted left by some time.

{ latch (level sensitive) → Transparent, when enable → 1
  ff → edge triggered
  non Transparent, en → 0

papergrid
Date: / /

Hold

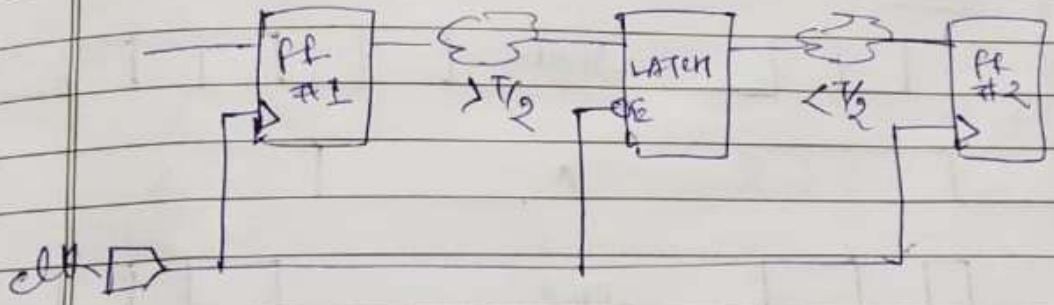


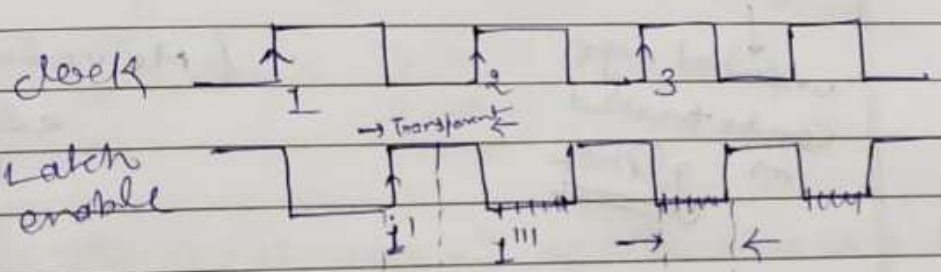
(+ve skew) → moving right

↓

degrades hold analysis

Latch Borrow Concept :-





"1" is captured at 1' (T/2) → 1'' ← Borrowed time

not Transparent

So tool will calculate setup time instead of 1 to 1' check points

1 to 1''

↓ after time Borrow

(for ff #1 and Latch)

↓ Default setup check edge

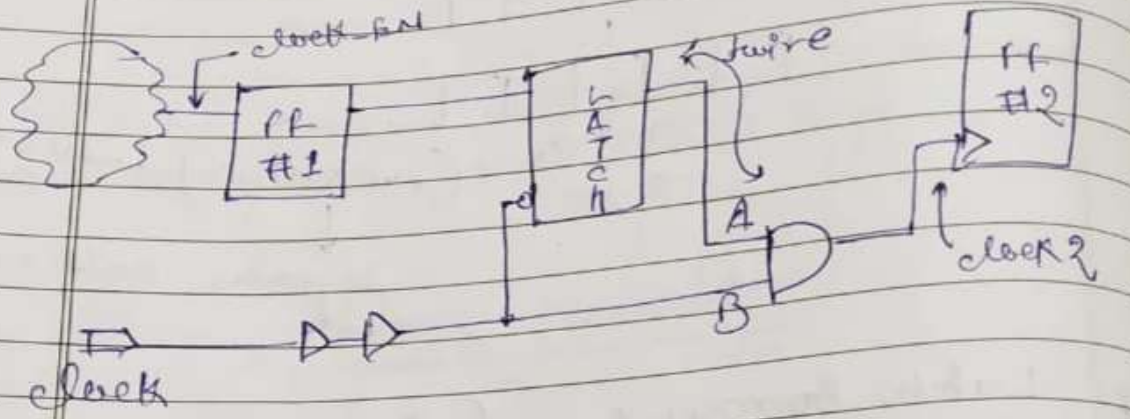
(for Latch to ff #2)

→ 1''-2 are two setup check pairs instead of 1'-2

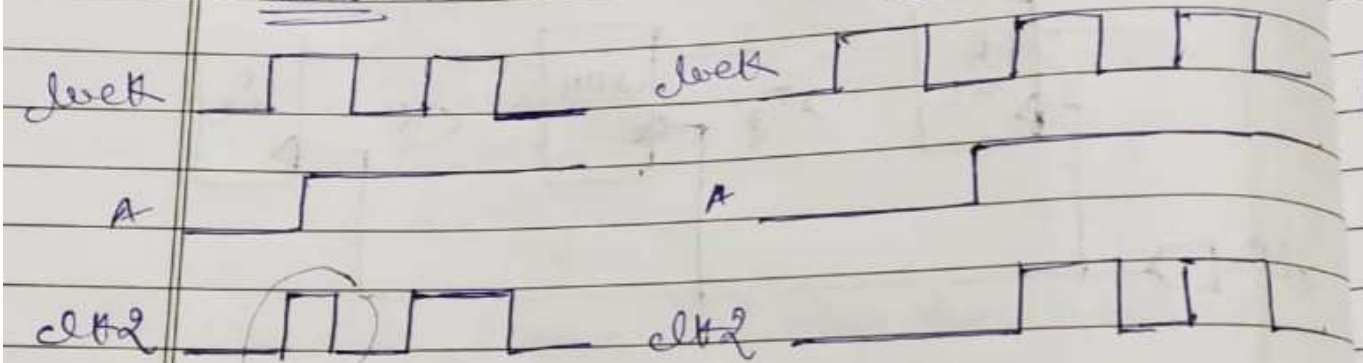
1'''-2 Hold check pairs

Almost zero possibility of hold violation

clock gating :- famous technique for power savings



Case 1 :-



undesired edge can be treated as glitch

Case II

(No undesired edge)

→ in Case II A should come before edge of clock i.e. setup time
so we have to check the setup and hold time of AND gate
AND gate is a combination ckt but we have to check the timing analysis for this because it is coming in path of clock gating.

⇒ A setup required of the path latch-en to A pin of AND gate w.r.t. the clk sig at B

papergrid

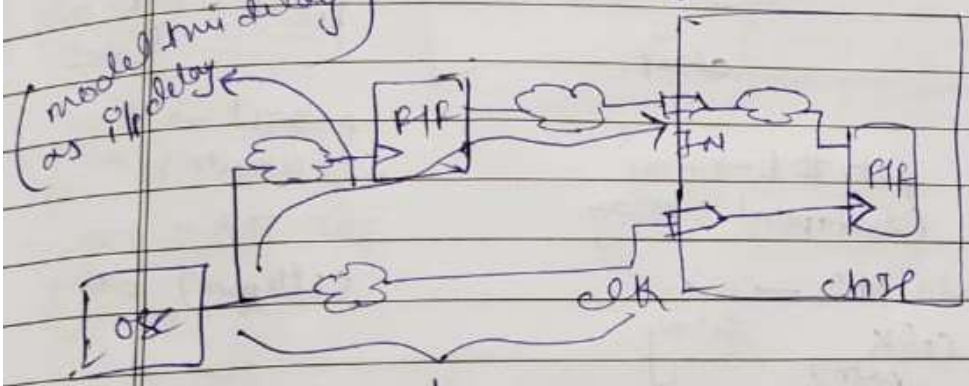
Date: / /

→ we can also use time borrow concept.
if borrow = 0, then my eqn :-

$$T_{cq}(\text{latch}) + T_{wire} + T_{setup} \leq T/2$$

Input pin constraining :-

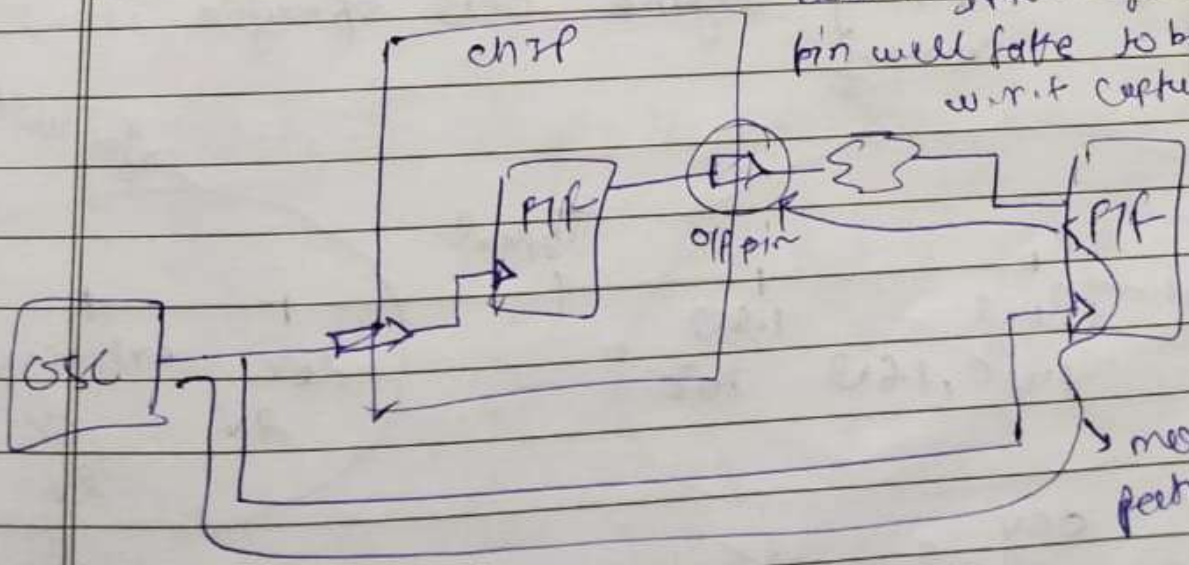
→ specify external time to reach i/p pin w.r.t launching clock



→ specify setup for max delay & Hold for min delay

(model this delay as delay as clk source latency)

output pin constraining :- specify how long the data shown from o/p pin will take to be captured w.r.t captured clk.



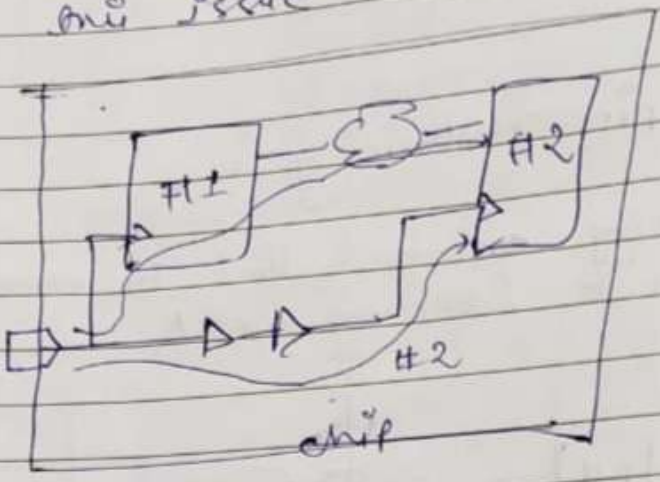
→ model this path as delay

we max, min for setup and Hold time

Date: / /

ON chip variation :-

→ temp across the chip is not same
→ OCV: it is an analysis made to handle this issue



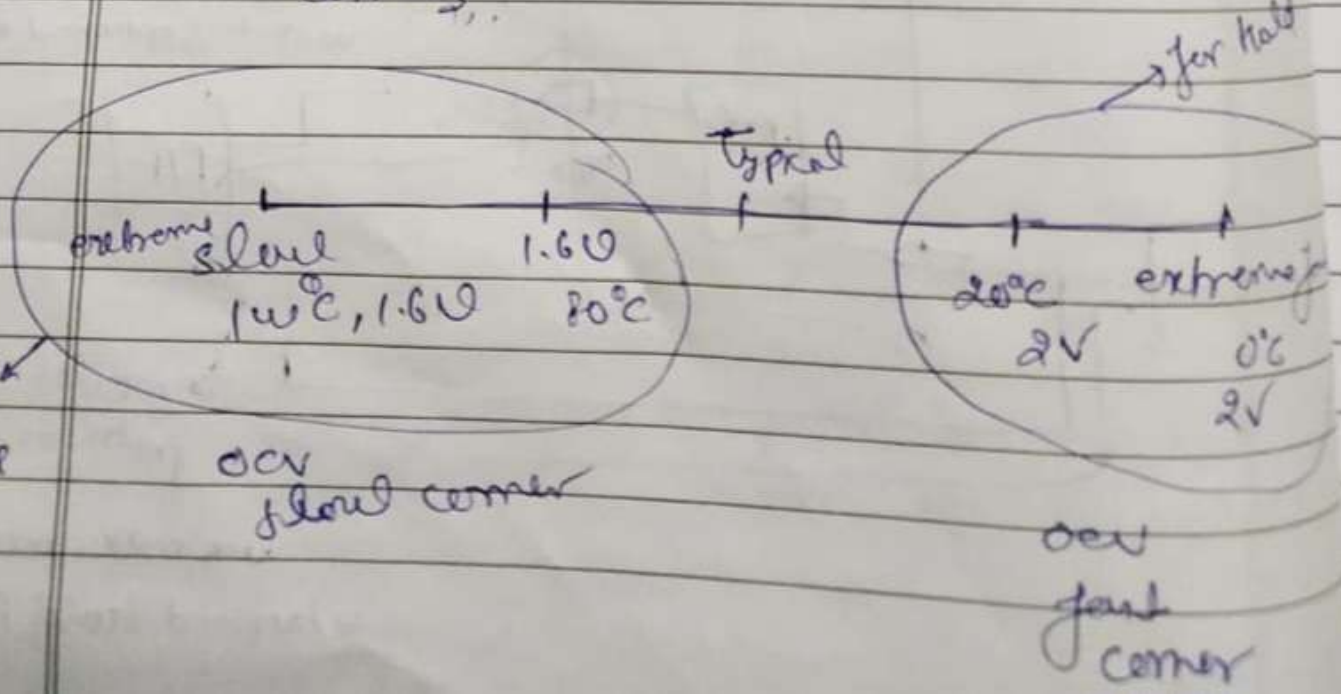
#1 and #2 are placed for easy ...

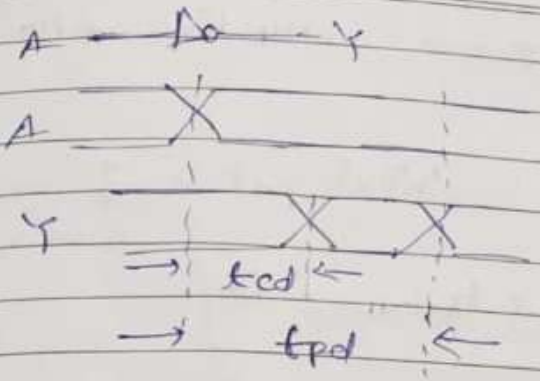
Setup analysis with OCV :- path #1 → max (Datapath) delay
path #2 → min (clk path) delay

(to make setup tighter)

Hold analysis with OCV
path #1 → min (Datapath) delay
path #2 → max (clk path) delay

→ Temp diff across chip can't be around 1w so extreme max and min corners are more pessimistic.
→ So library define OCV specific slow corners.

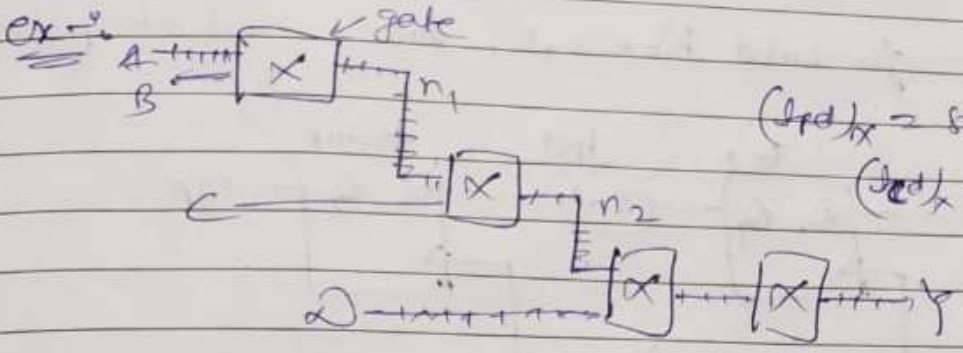




t_{pd} → max^m time after which o/p gets stable

t_{cd} → min^m time after o/p starts to change.

ex:-

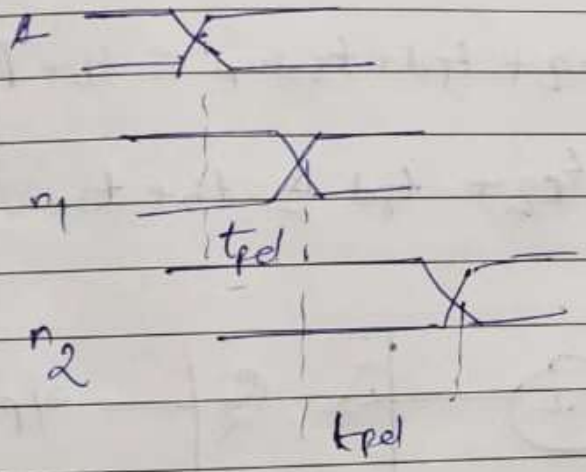


$(t_{pd})_{\text{gate}} = 50\text{ps}$

$(t_{cd})_{\text{gate}} = 30\text{ps}$

$(t_{pd})_{ckt} = ?$

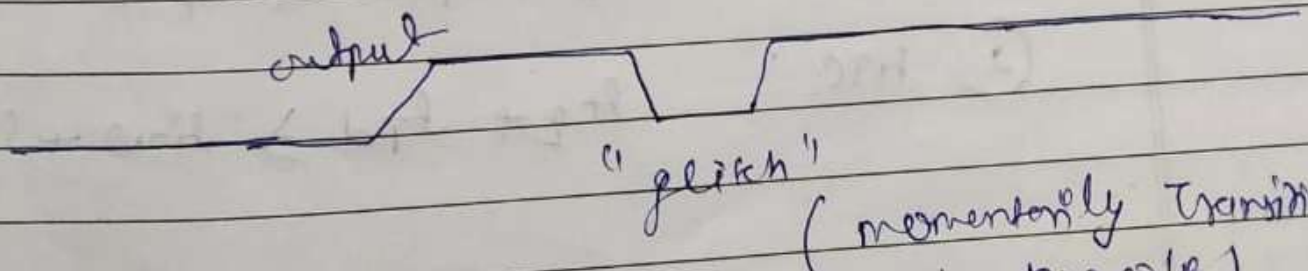
$(t_{cd})_{ckt} = ?$



$(t_{pd})_{ckt} = 4 \times t_{pd}$
$= 200\text{ps}$

$(t_{cd})_{ckt} = 2 \times t_{cd}$
$= 2 \times 30$
$= 60\text{ps}$

Glitch :- single i/p Transitions make multiple o/p Transition called as glitch.



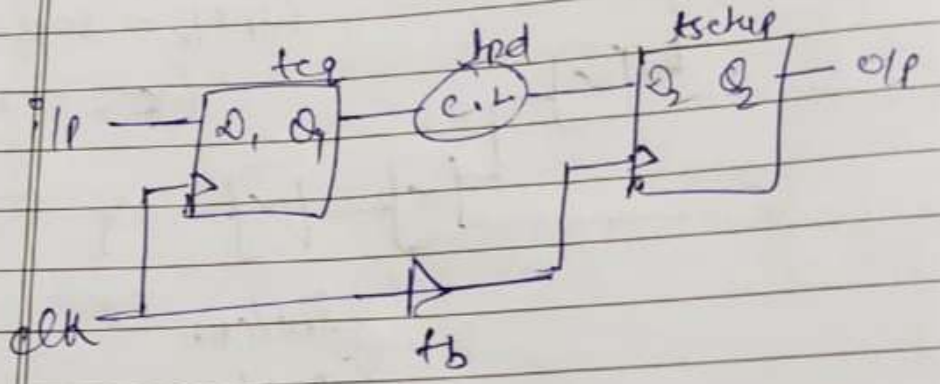
(momentarily Transition in the o/p)

Aperature time → setup time + hold time

→ setup time for critical path ÷ we can get max clock freq.
Calculation

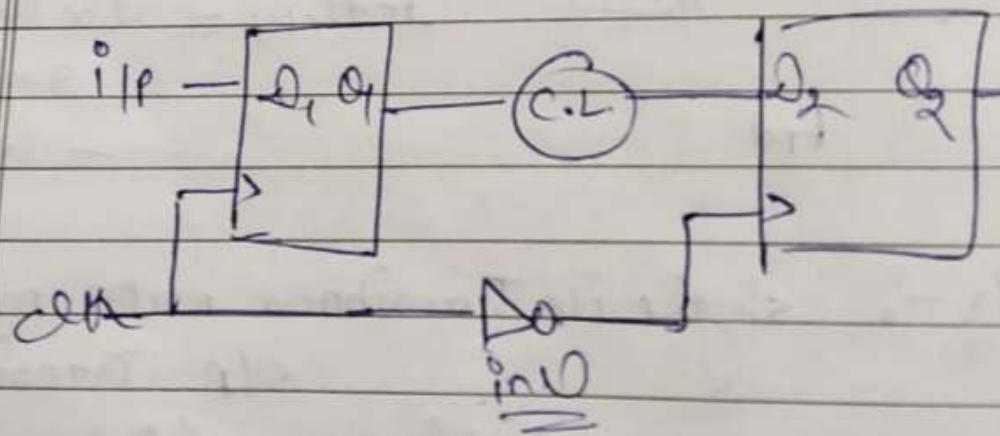
$$t_{pcq} + t_{pd} + t_{setup} \leq T_c$$

for hold time eqn, take shortest path

① 

① STC : $t_{cq} + t_{pd} + t_{setup} \leq t_b + T$

② HTC : $t_{cq} + t_{pd} \geq t_b + t_{hold}$

② 

① STC : $t_{cq} + t_{pd} + t_{setup} \leq$ time

② HTC : $t_{cq} + t_{pd} \geq$ time +

Slack = Required time (Constraint) − Arrival time (Inputs and delays)

+ve slack → timing met
-ve " → not met

① setup time slack = (provided setup time) − (required setup time)

RT − AT

hold time slack ∴ (provided Hold time − required hold time)

(AT − RT)

(min delay in data path) − (max delay in clk path)

(min delay in clk path) − (max delay in data path)

in clk path delay, we have to include clock period also.