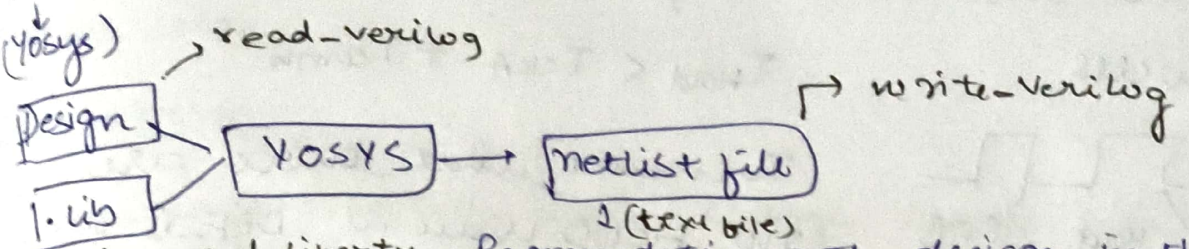
Synthesizer

Tool used for converting RTL to Netlist

(Yosys)



1 (text file)

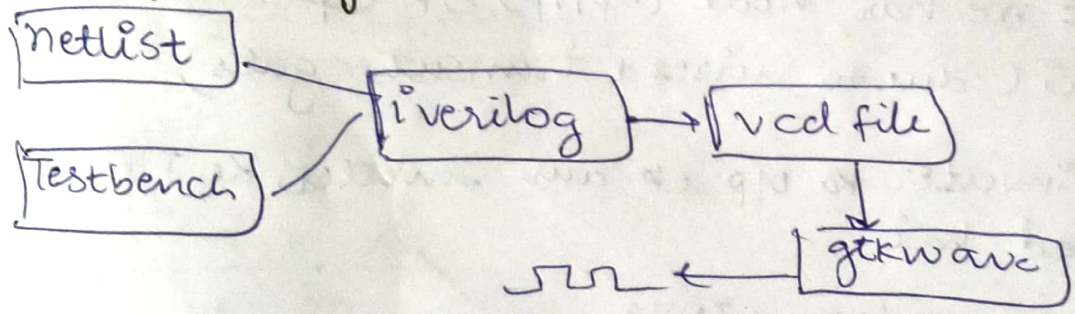
Representation of the design in the form of standard cell in .lib.
(logic gate)

Netlist: Mera ckt kis gate se Bana hai aur connections kaise hai.

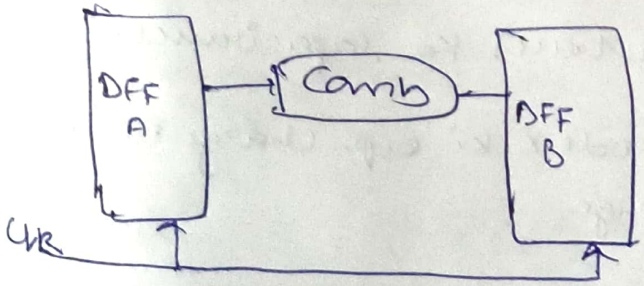
.lib = Un gates ki specification delay aur power info kya hai speed size etc.. (AND or) ...

- lib has slow med fast AND ... or etc (2i/p 3i/p 4i/p)

Verify the Synthesis



note primary i/p & o/p will remain same b/w RTL design & synthesized netlist → Same (TB) can be used



$T_{clk} > T_{clkQ} + T_{comb} + T_{setup\,B}$

$f_{clk\,max} = \frac{1}{T_{clk\,min}}$

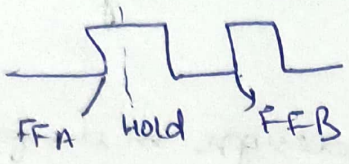
Comb. delay in logic path determines the Max. Speed of operation of digital logic ckt.

If we need the cells to work fast | Tcombi small |

Why do we need slow cells?

To meet the hold Requirement we need slow cells

$T_{Hold} < T_{CLKA} + T_{comb}$



To ensure that there are no HOLD issue at DFF_B we need cells that work slowly.

→ we need fast cells to meet the req'd performance

→ we need slow cells to meet Hold Requirements

This condition is present in (.lib)

Fast cell v/s slower cell

Digital ckt me har node (o/p/I/p) ek R capacitance banata hai (due to wires + transistor gates)

Maan lo ek inverter ka o/p ek aur Inverter ke I/p se connected hai

INV1 → Node x → INV2

↳ Node x pe do cap. hai

1) INV2 ka I/p gate cap.
2) wire (Node x ka Metal trace) ka capacitance

· Jab INV1 o/p ko 0 → 1, usko Node x ki cap. charge karni padti hai

· jab 1 → 0 Node x ki cap discharge

Ye charging and discharg. time he delay banata hai

NOTE: Har ek node load cap. banata hai aur jitna zyada load cap. hoga utna zyada delay hoga.

(0/1) Jo node hota hai vo node banata hai.

2) Cell delay kispe depend karta hai
→ Agar cap jaldi charge/dischg Hogi → signal fast change → Less delay
→ vice versa.

3) Kaise Capacitance fast charge/dischg hoti hai?
→ Iske liye transistor Mei zyada current flow hona chaiye
→ Matlab transistor (wider) mota hona chaiye → tab vo zyada current supply kar skta hai.

Note current zyada → cap fast charge Hogi → delay kam but [area + power zyada] as transistor Bada Hai
So, it is used if want fast Design.

Selection of cells

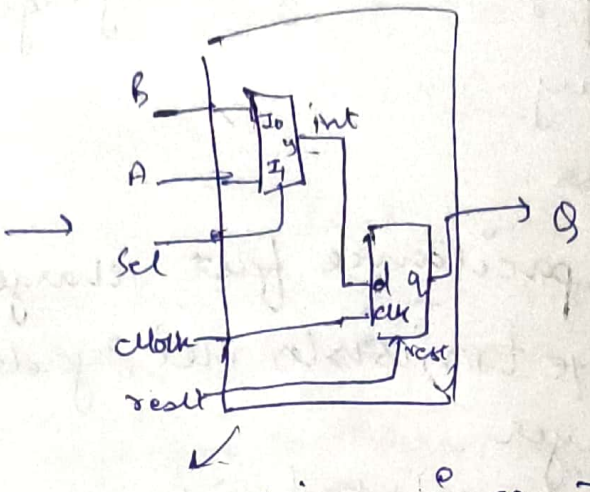
- Cell selection is very critical
- Agar sirf fast cells use karoge / Power Area Prob
- Sirf slow cells → timing problem
- Isliy Synthesizer ko correct CONSTRAINTS dena Zarui hai

Balance = CONSTRAINTS b/w speed, Ar, Powr

Synthesizer ko app Constraints dete Ho.
Jaise
→ Clock freq kitni chaiye
→ Max power kitna allow
→ Max area Allow hai

Synthesizer in cond^n ko dekh ke decide karta Konsa cell fast ya slow use karna hai.

Code
module ( )
O/P
Wire
assign
always@
begin
if
else
end
endmodu



The CKt on the right is created from RTL using using gates available in .lib and given out as Netlist.

DAY 2

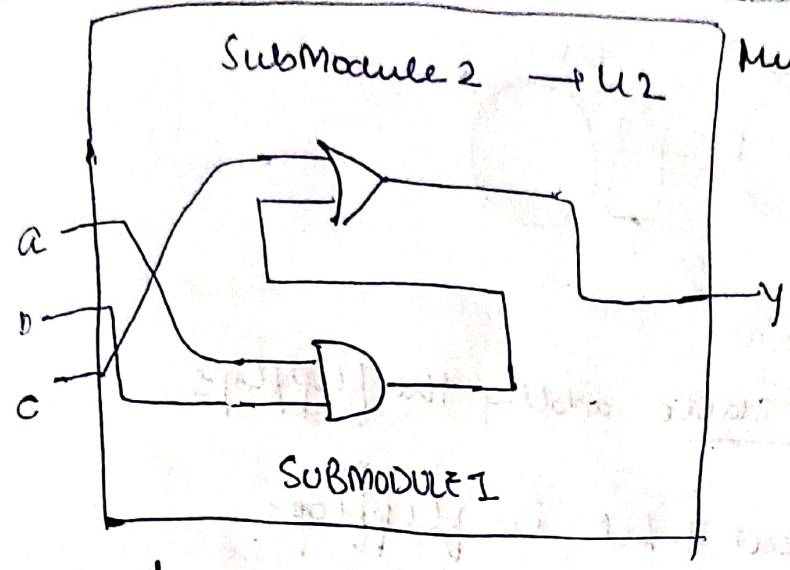
Library ("sky130_fd_sc_hd__tt_025C_1v80")
typical process
temprature
voltage

PVT → temprature
Process
voltage

variations due to fabrication →
- .lib giving leakage powers
- Area no.
- Delay
- Power Information
- Timing Information

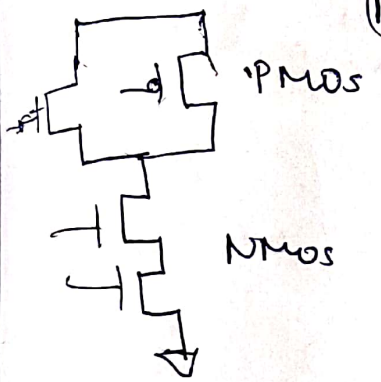
going from ("sky130_fd_sc_hd__and2_0") to ("and2_2")
→ Slower
area = .6.256...
NW delay

faster
area = 7.50720...
Then
means this have
Wider Transistors
less Delay

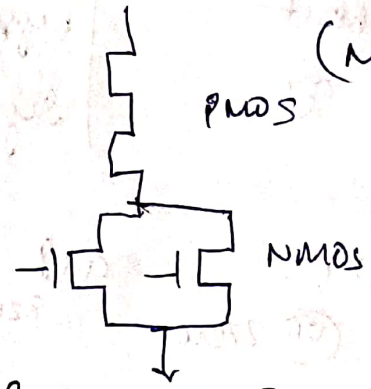


Multiple Module

Stacked NMOS. In Netlist good (NAND)



Stacked PMOS is Bad (NOR)



Q why Sub Module level Synthesis ?

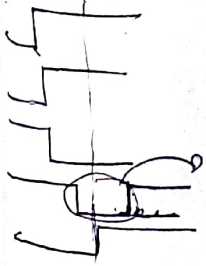
1) when we have multiple instance of same Module

2) Divide & Conquer
(massive) design portion by portion.

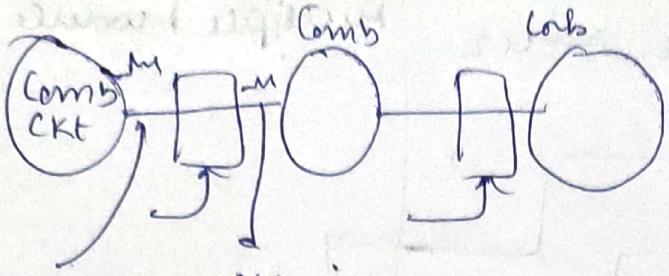
Command Synth -top (Module Name)

Why flip flop used

as when we use Continuous Combinational ckts.
we at some point get glitch.
hence to avoid this we us flip flops.



→ acts like storing element



means
y the i/p is
glitching

o/p will
be stable adding the flipflops

Clk has Reset & Set in flipflops

Diff b/w with Dff

Async Reset	Syn Reset
Async Set	Syn Set

eg ax4 ax8

(50) 2zeros 3zeros

ax9=y

ax [8+1]

ax8 + ax1

a000 ← ax8 + a

a + a + 9

↓

aa ax9

I/p [2:0] a

o/p [5:0] y