review

Exam contents

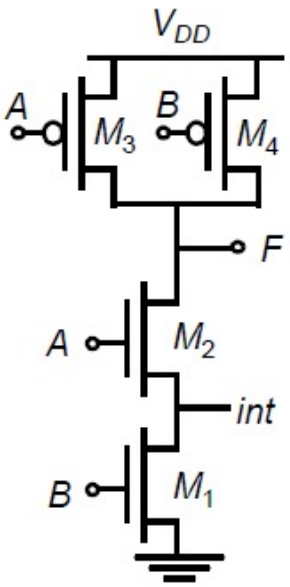
- Combination circuit logic
- wire
- Datapath
- Sequential circuit Logic

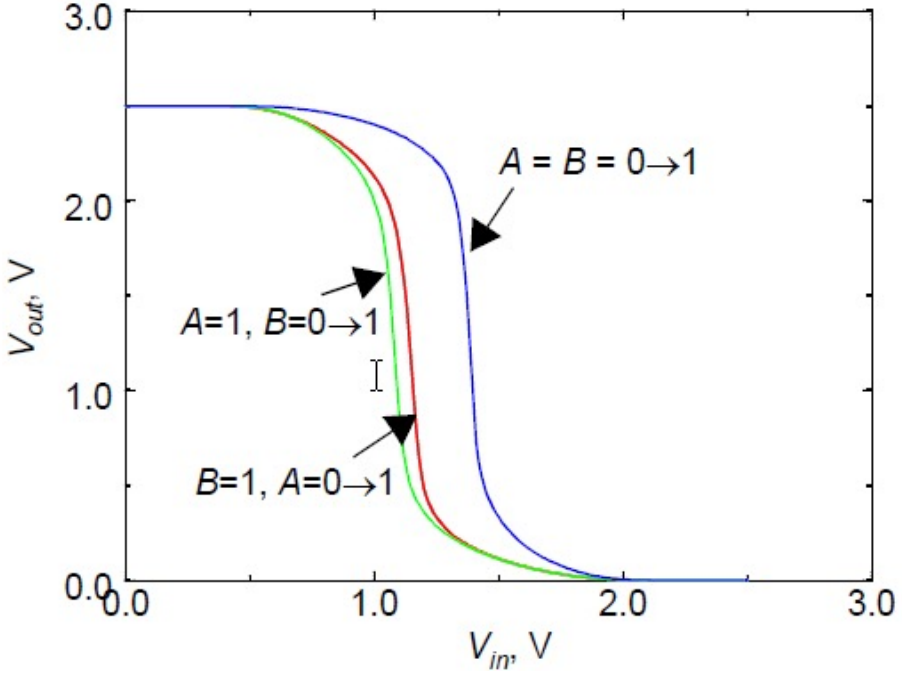
Exam contents

- **Combination circuit logic**
- wire
- Datapath
- Sequential circuit Logic

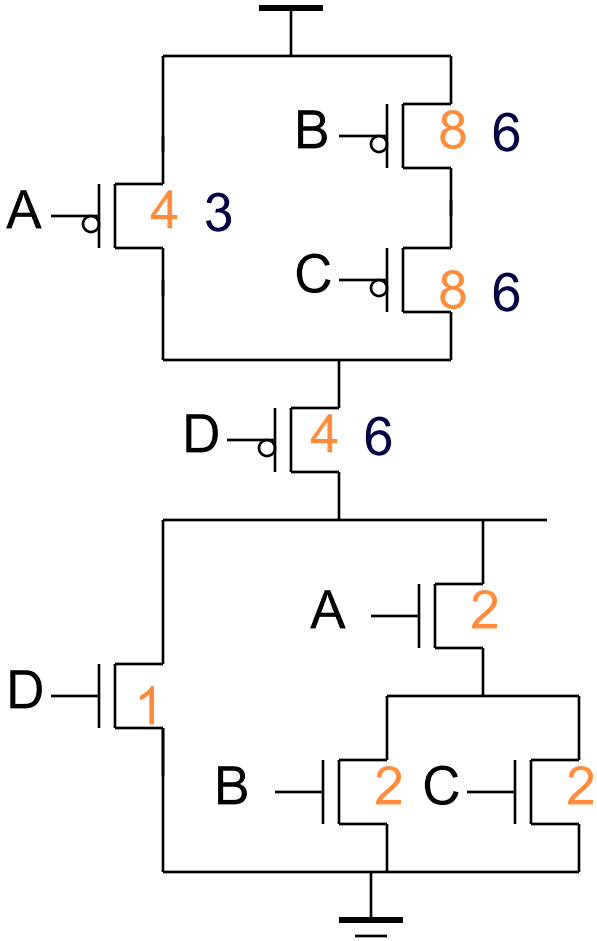
Combination logic.

- CMOS logic Properties





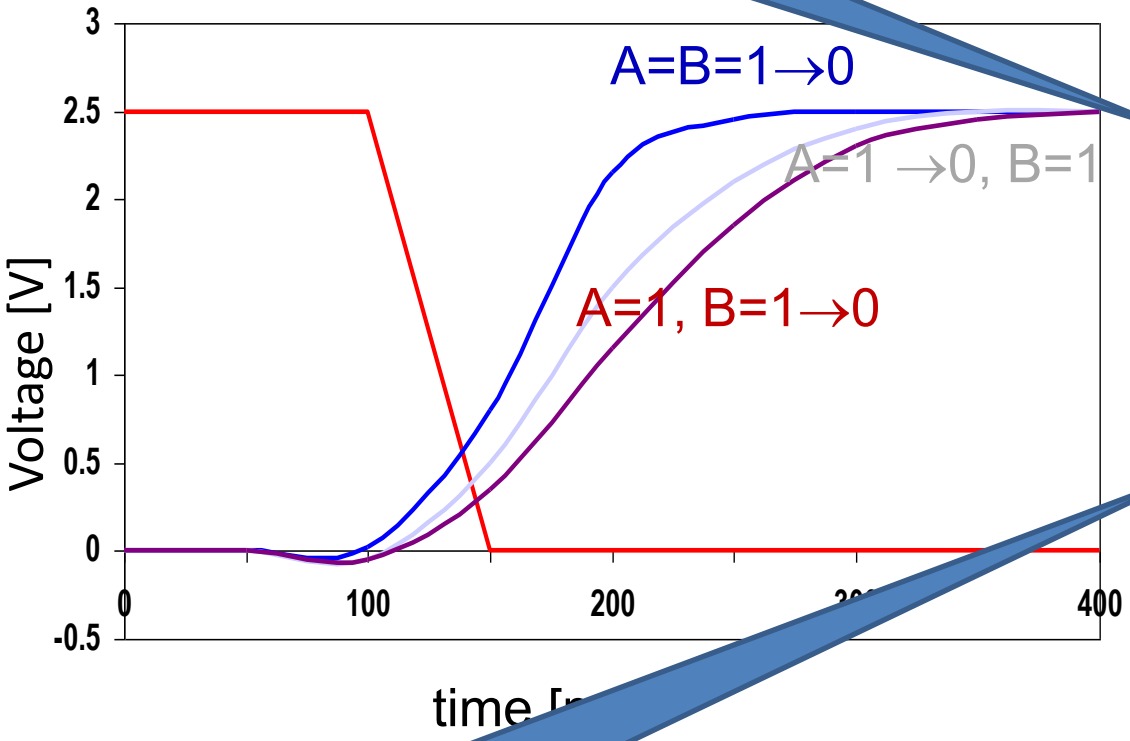
Sizing has different options



$$OUT = \overline{D + A \bullet (B + C)}$$

Delay Dependence on Input Patterns

Shared cap. **discharging**



Shared cap. **charging**

Input Data Pattern	Delay (psec)
A=B=0→1	69(max)
A=1, B=0→1	62
A= 0→1, B=1	50
A=B=1→0	35(min)
A=1, B=1→0	76
A= 1→0, B=1	57

NMOS = 0.5μm/0.25 μm
PMOS = 0.75μm/0.25 μm
C_L = 100 fF

Modified formula

$$t_p = t_{par} + t_{ext} = \ln 2.(C_{par} + C_{ext})R_{par}$$

$$= \ln 2.R_{inv}C_{int}\left(\frac{C_{par}}{C_{int}} + \frac{C_g}{C_{int}}\frac{C_{ext}}{C_g}\right)$$

$$= t_{p0}\left(\frac{C_{par}}{C_{int}} + \frac{C_g}{C_{int}}\frac{C_{ext}}{C_g}\right)$$

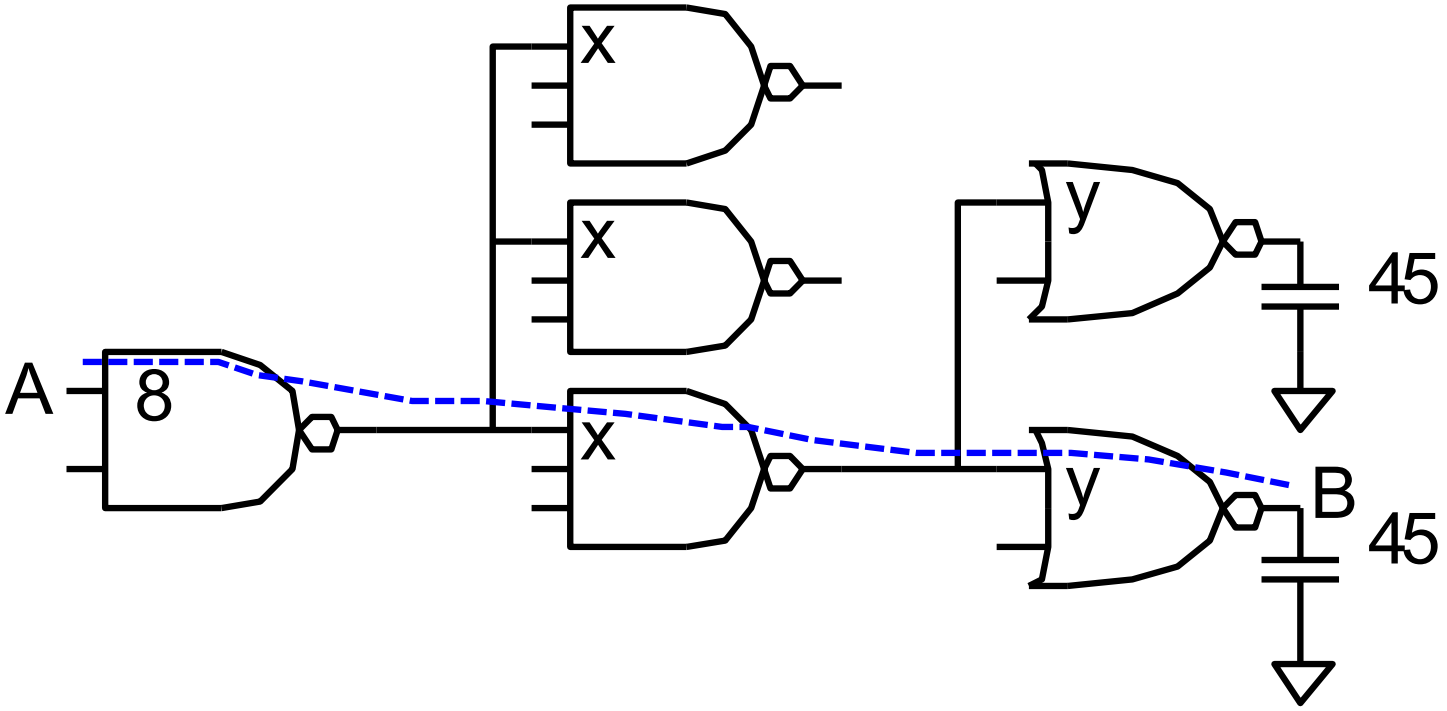
$$= t_{p0}\left(\frac{C_{par}}{C_{int}} + \frac{C_g}{\gamma C_{ginv}}\frac{C_{ext}}{C_g}\right)$$

$$= t_{p0}\left(p + \frac{gf}{\gamma}\right)$$

$$C_{ginv}\Big|_{R_{par}=R_{inv}}$$

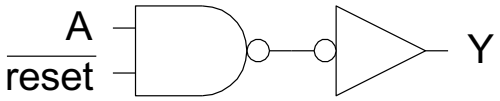
Example: 3-stage path

- Select gate sizes x and y for least delay from A to B

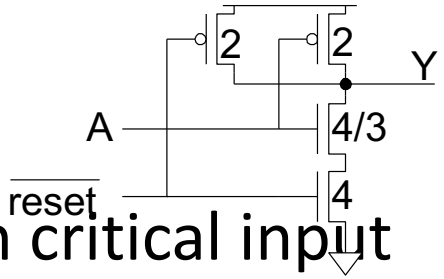


Asymmetric Gates

- Asymmetric gates favor one input over another
- Ex: suppose input A of a NAND gate is most critical
  - Use smaller transistor on A (less capacitance)
  - Boost size of noncritical input
  - So total resistance is same



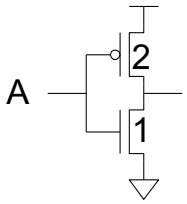
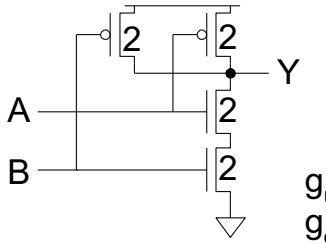
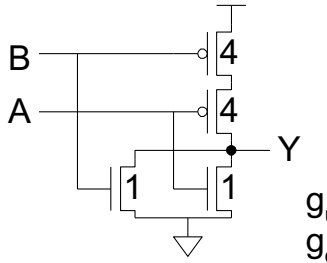
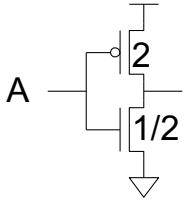
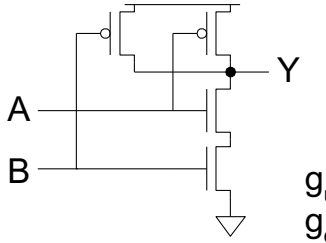
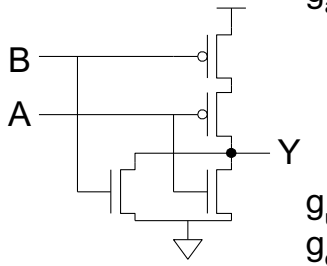
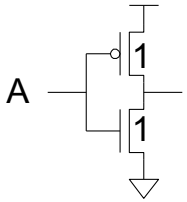
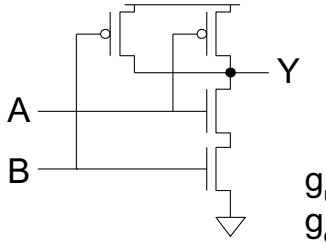
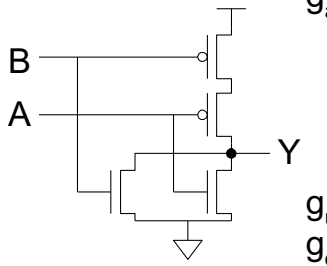
- **$g_A = 10/9$**
- $g_B = 2$
- $g_{total} = g_A + g_B = 28/9$



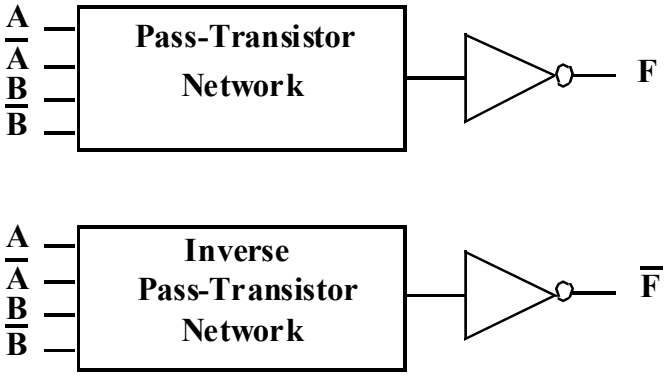
- Asymmetric gate approaches g = 1 on critical input
- But total logical effort goes up

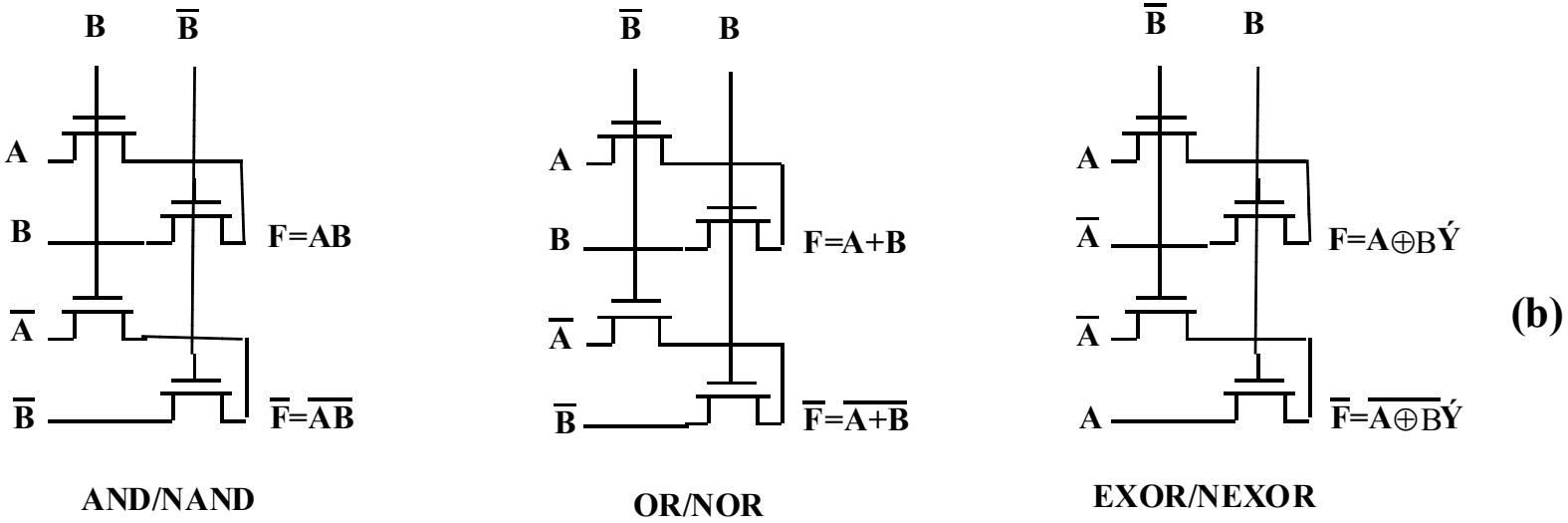
the special Path Optimization

Catalog of Skewed Gates

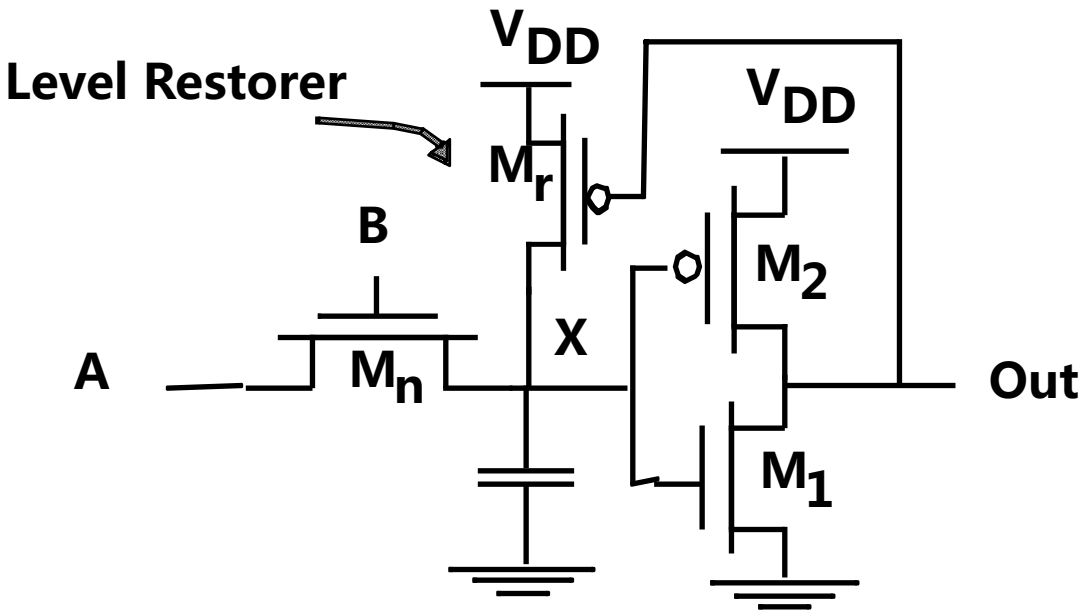
	Inverter	NAND2	NOR2
unskewed	 $g_u = 1$ $g_d = 1$ $g_{avg} = 1$	 $g_u = 4/3$ $g_d = 4/3$ $g_{avg} = 4/3$	 $g_u = 5/3$ $g_d = 5/3$ $g_{avg} = 5/3$
HI-skew	 $g_u = 5/6$ $g_d = 5/3$ $g_{avg} = 5/4$	 $g_u =$ $g_d =$ $g_{avg} =$	 $g_u =$ $g_d =$ $g_{avg} =$
LO-skew	 $g_u = 4/3$ $g_d = 2/3$ $g_{avg} = 1$	 $g_u =$ $g_d =$ $g_{avg} =$	 $g_u =$ $g_d =$ $g_{avg} =$

Complementary Pass Transistor Logic



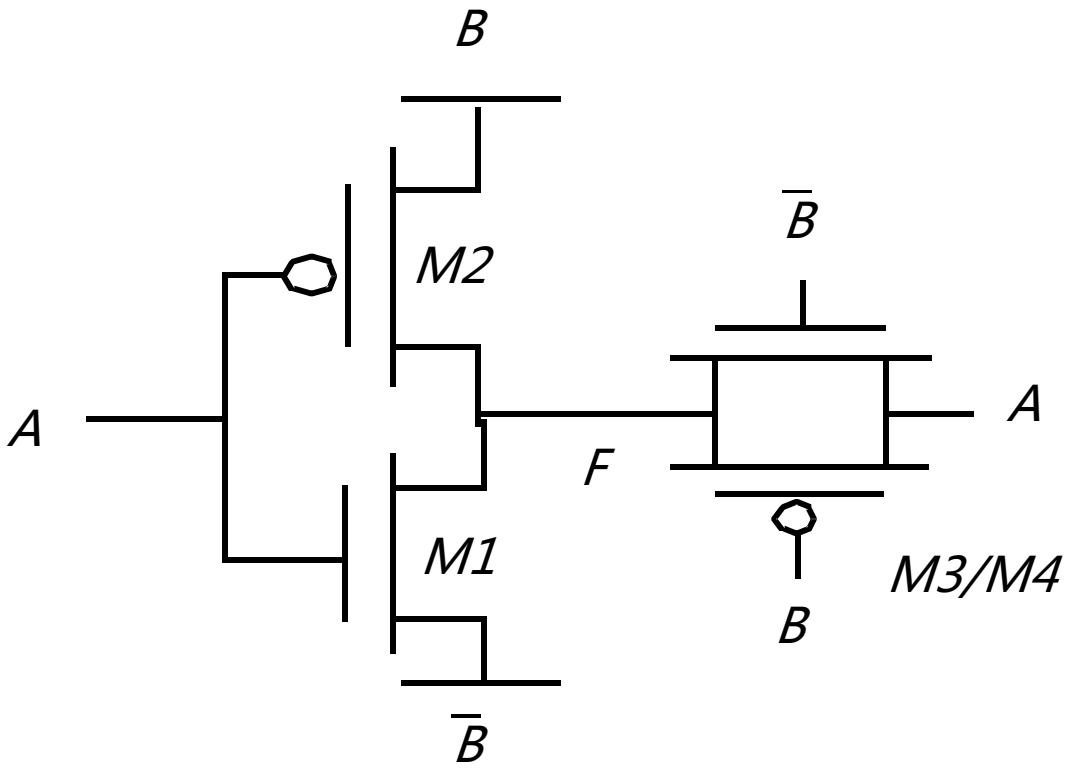


Solution 1:Level Restoring Transistor

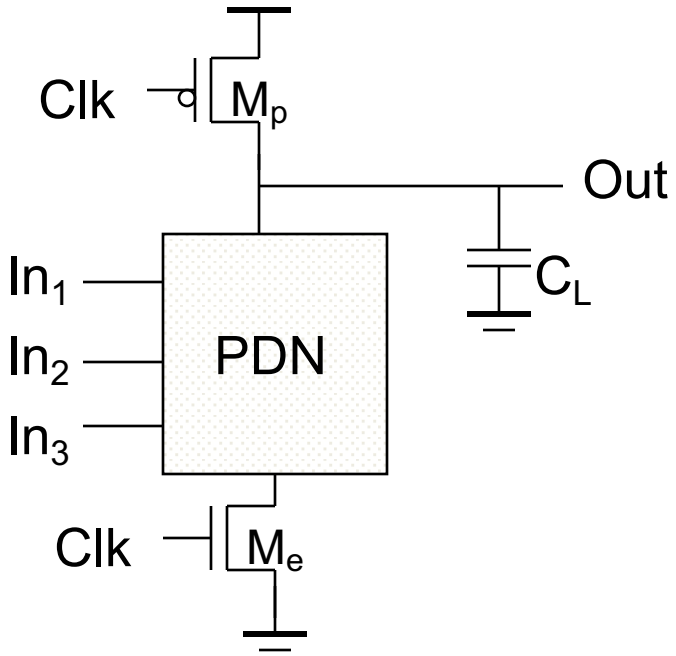


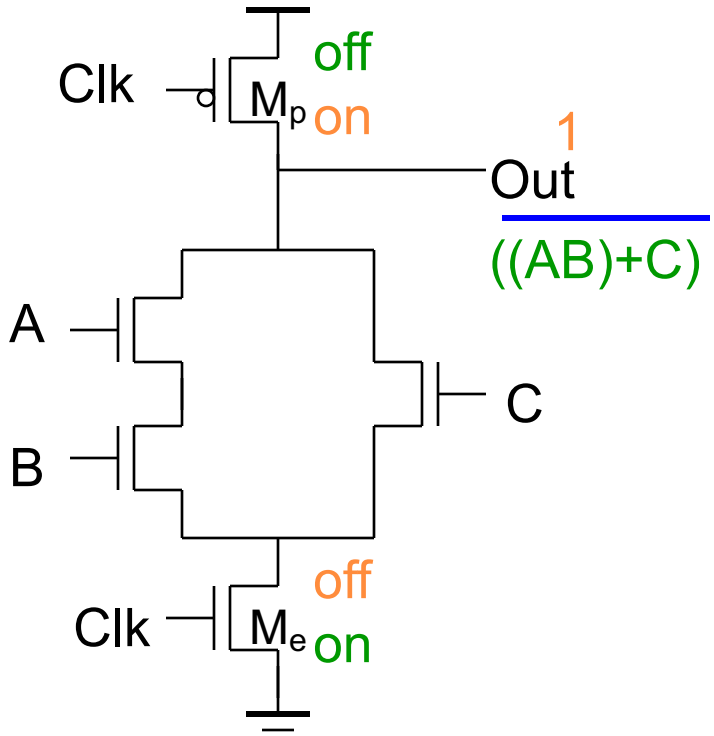
- Advantage: Full Swing
- Restorer adds capacitance, takes away pull down current at X
- Ratio problem

Transmission Gate XOR



Dynamic Gate



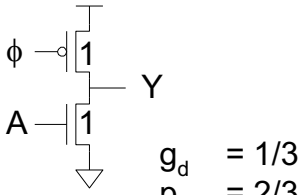
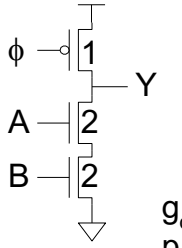
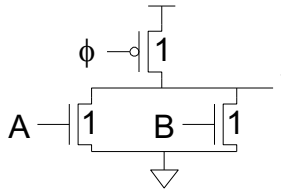
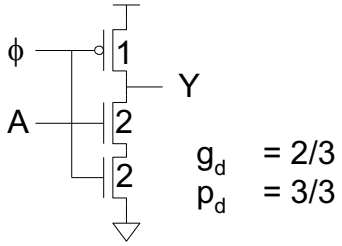
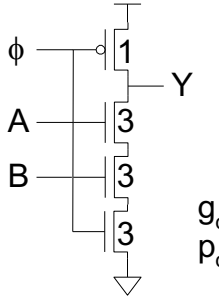
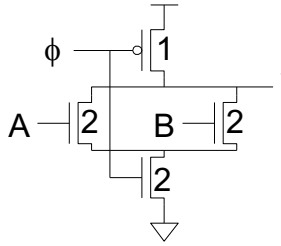


Two phase operation

Precharge (Clk = 0)

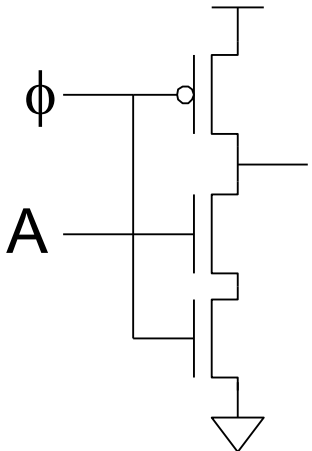
Evaluate(Clk = 1)

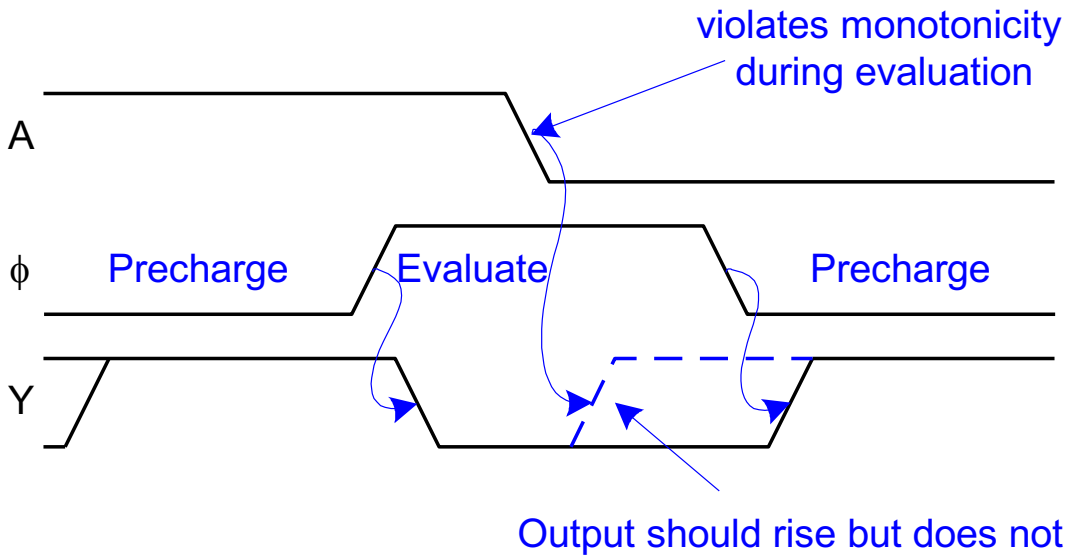
Logical Effort

	Inverter	NAND2	NOR2
unfooted	 $g_d = 1/3$ $p_d = 2/3$	 $g_d = 2/3$ $p_d = 3/3$	 $g_d = 1/3$ $p_d = 3/3$
footed	 $g_d = 2/3$ $p_d = 3/3$	 $g_d = 3/3$ $p_d = 4/3$	 $g_d = 2/3$ $p_d = 5/3$

Monotonicity

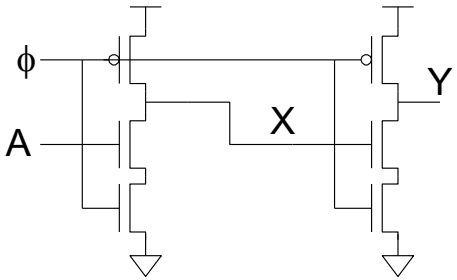
- Dynamic gates require ***monotonically rising*** inputs during evaluation
  - 0 -> 0
  - 0 -> 1
  - 1 -> 1
  - ***But not 1 -> 0***

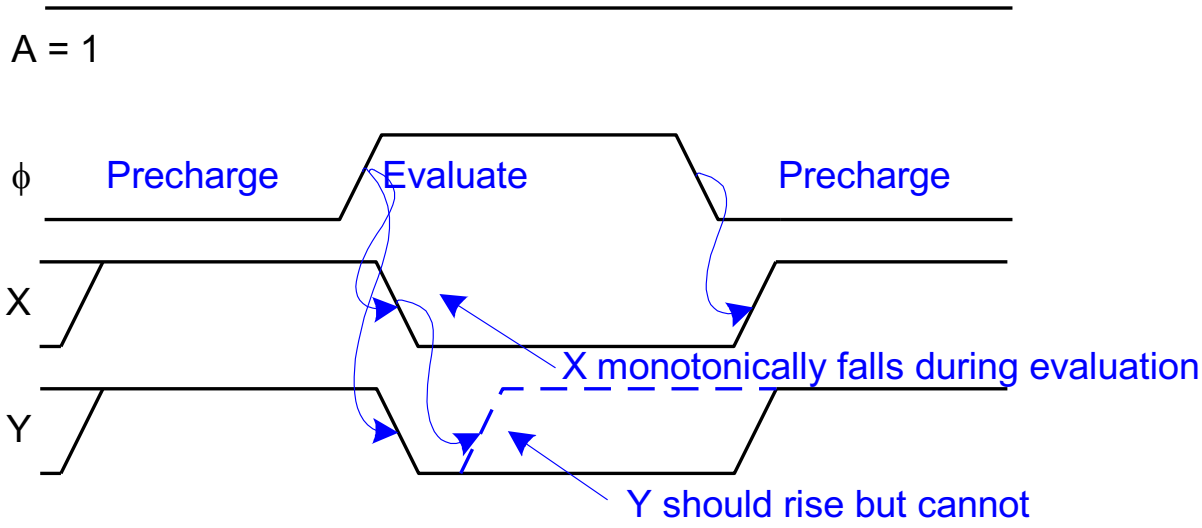




Monotonicity Woes

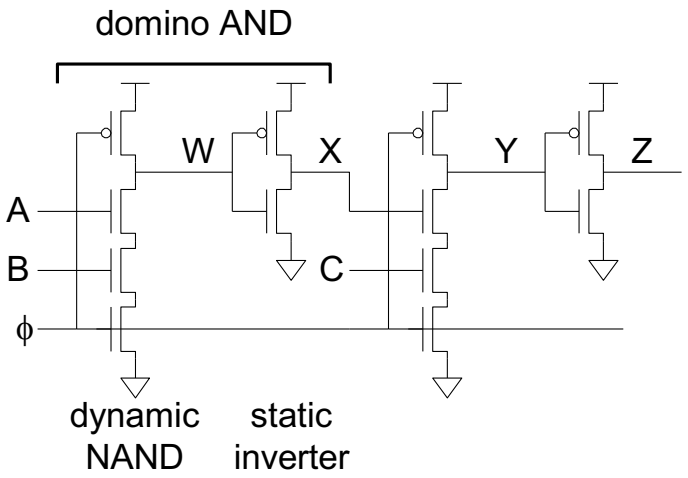
- But dynamic gates produce monotonically falling outputs during evaluation
- Illegal for one dynamic gate to drive another!

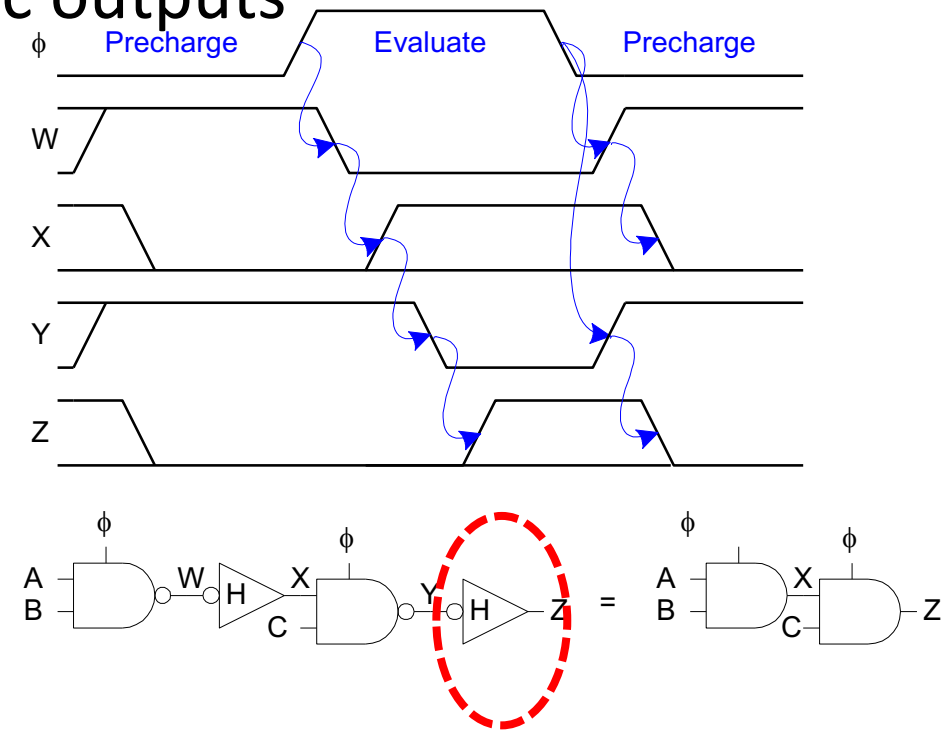




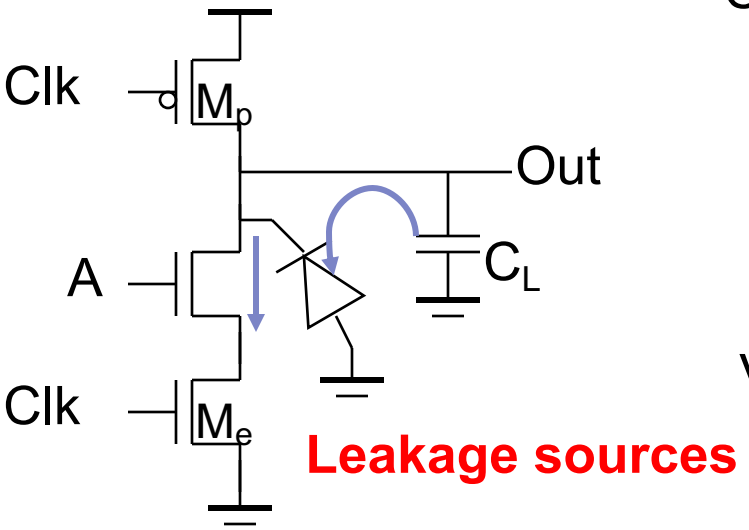
Domino Gates

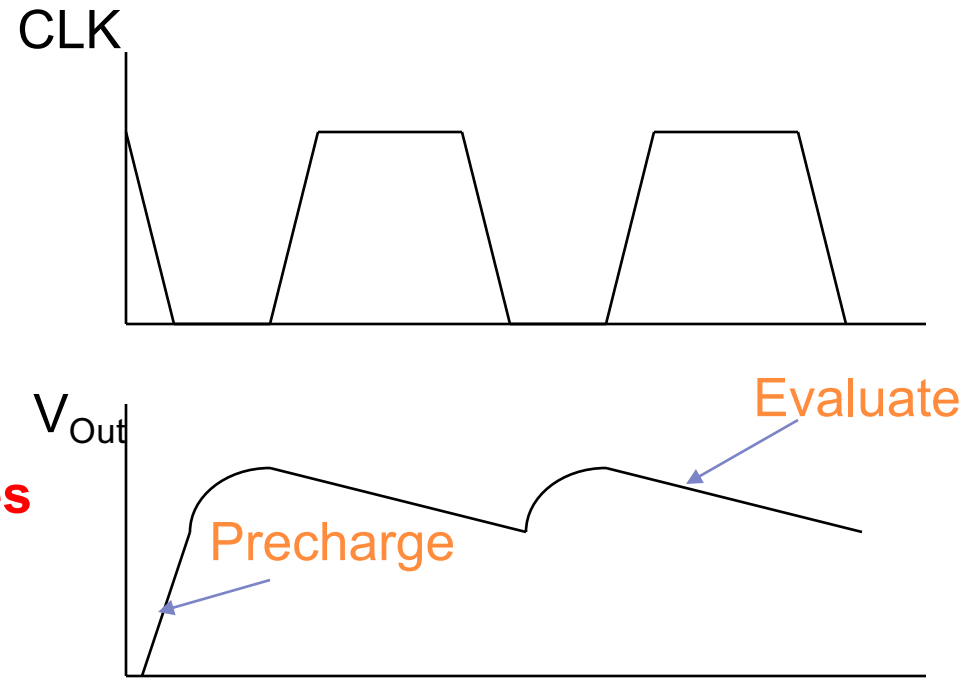
- Follow dynamic stage with inverting static gate
  - Dynamic / static pair is called domino gate
  - Produces monotonic outputs





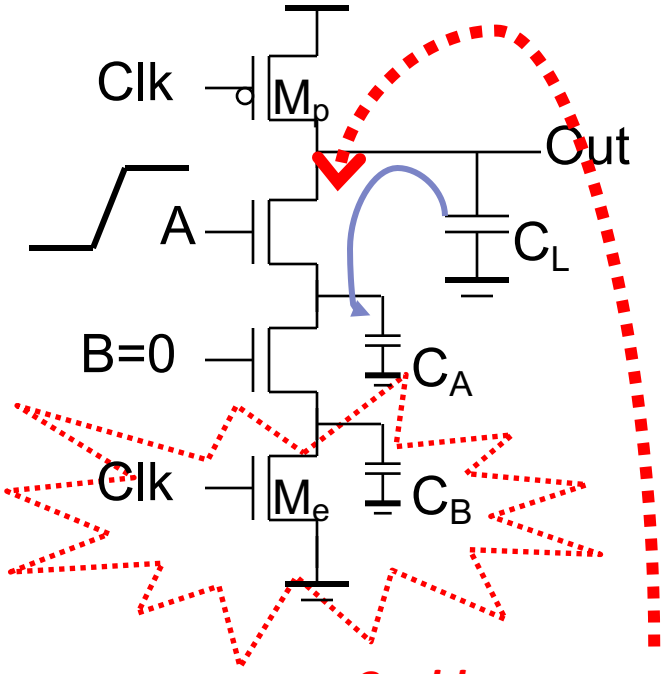
Issues in Dynamic Design 1: Charge Leakage





Dominant component is subthreshold current

Issues in Dynamic Design 2: Charge Sharing



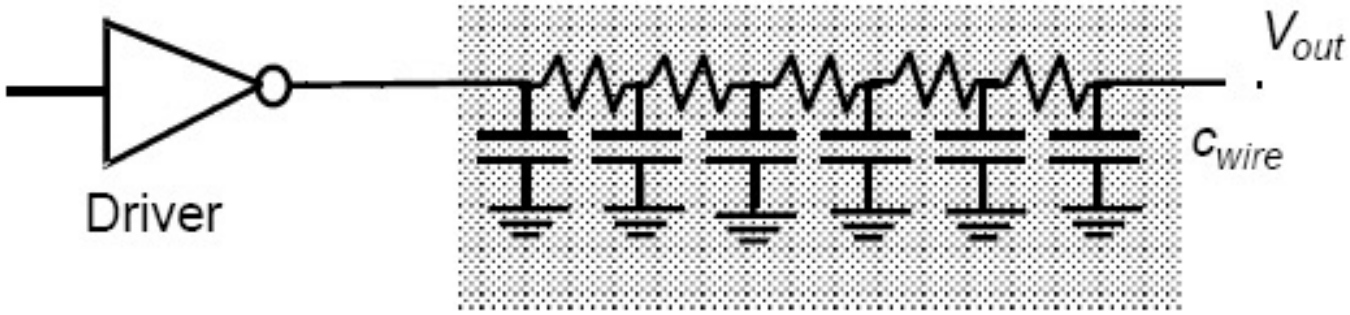
Charge stored originally on C_L is redistributed (shared) over C_L and C_A leading to reduced robustness

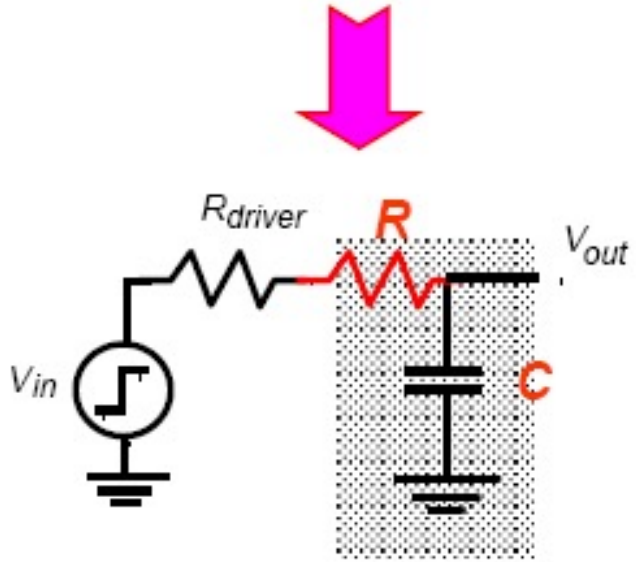
Could we move it to there?

Exam contents

- Combination circuit logic
- **wire**
- Datapath
- Sequential circuit Logic

Lumped RC Model



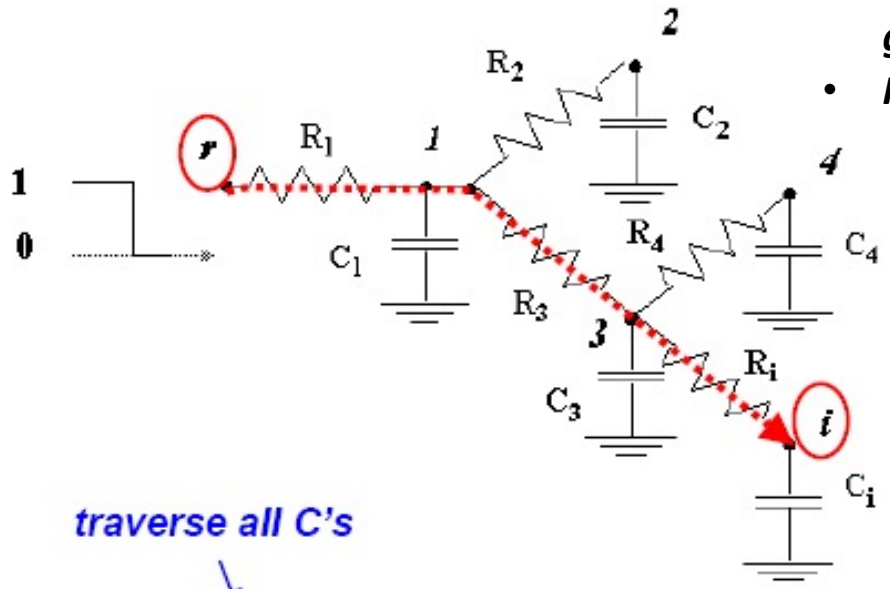


Pessimistic approximation

time const = RC

Elmore Delay of RC-Network

- ***Does not have any resistive loops***
- ***All capacitances are between a node and a ground***
- ***Have a single input node***



$$\tau_{Di} = R_1C_1 + R_1C_2 + (R_1 + R_3)C_3 + (R_1 + R_3)C_4 + (R_1 + R_3 + R_i)C_i$$

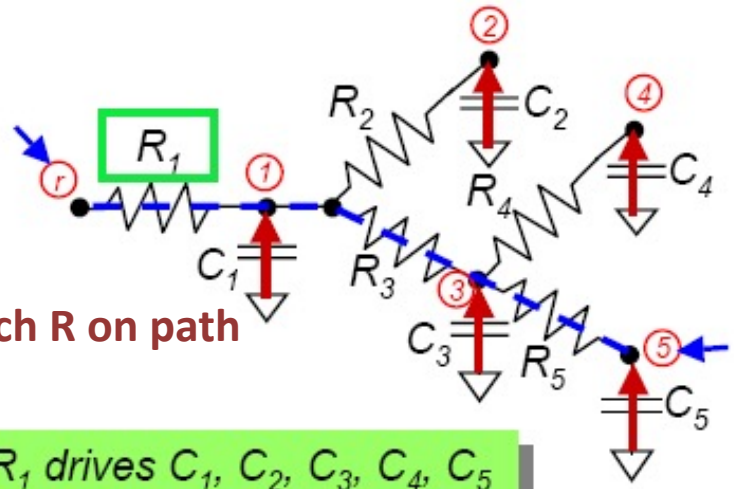
traverse all C's

$$\tau_{Di} = \sum_{k=1}^{N} C_k R_{ik}$$

shared path

$$R_{ik} = \sum R_j \quad \text{for } R_j \in path(r \to i) \cap path(r \to k)$$

Delay Elmore Delay – Another Solution Search all C



Search R on path

$$\tau_{D5} = R_1C_1 + R_1C_2 + (R_1 + R_3)C_3 + (R_1 + R_3)C_4 + (R_1 + R_3 + R_5)C_5$$

R_1 drives C_1, C_2, C_3, C_4, C_5

R_3 drives C_3, C_4, C_5

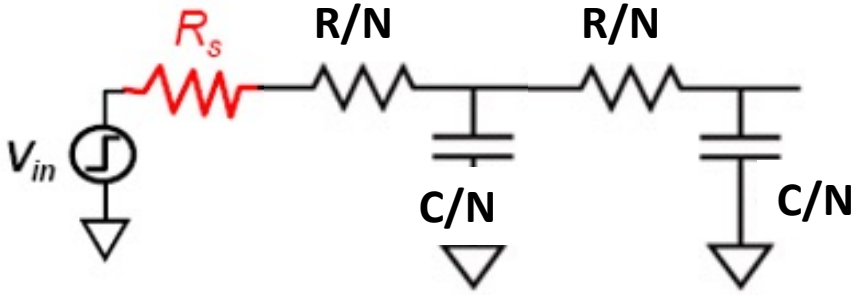
R_5 drives C_5

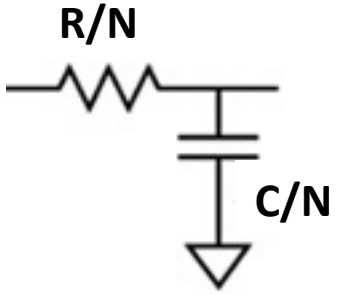
$$\tau_{D5} = R_1(C_1 + C_2 + C_3 + C_4 + C_5) + R_3(C_3 + C_4 + C_5) + R_5C_5$$

downstream from the root

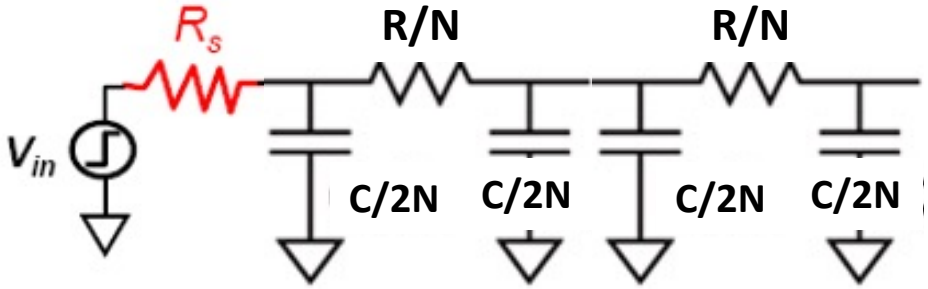
$$\tau_{s\to t} = \sum_{R_k \text{ along path } s\to t} R_k \text{ (sum of } C_i \text{ driven by } R_k)$$

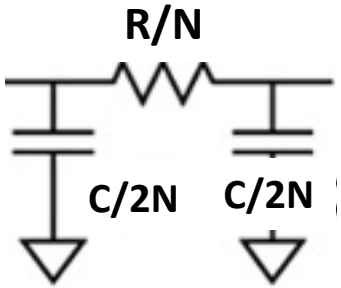
Which model is the fastest one





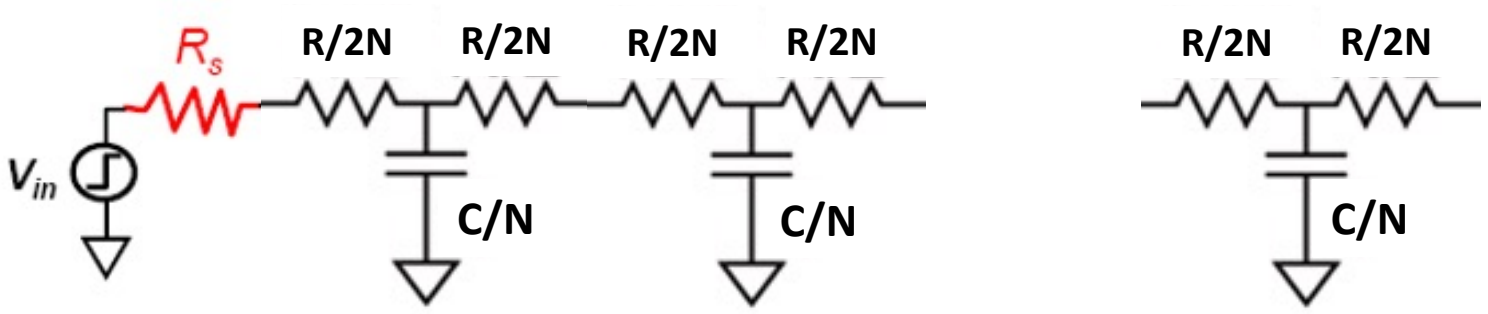
$$\sum_{i=1}^{N}\left(R_s+\frac{i}{N}R\right)\frac{C}{N}=R_sC+\frac{N(N+1)}{2N}R\,\frac{C}{N}=R_s\,C+\frac{(N+1)}{2N}RC$$





$$R_s\frac{C}{2N}+\sum_{i=1}^{N-1}\left(R_s+\frac{i}{N}R\right)\frac{C}{N}+\left(R_s+\frac{N}{N}R\right)\frac{C}{2N}=R_sC+\frac{N}{2N}RC$$

Which model is the fastest one

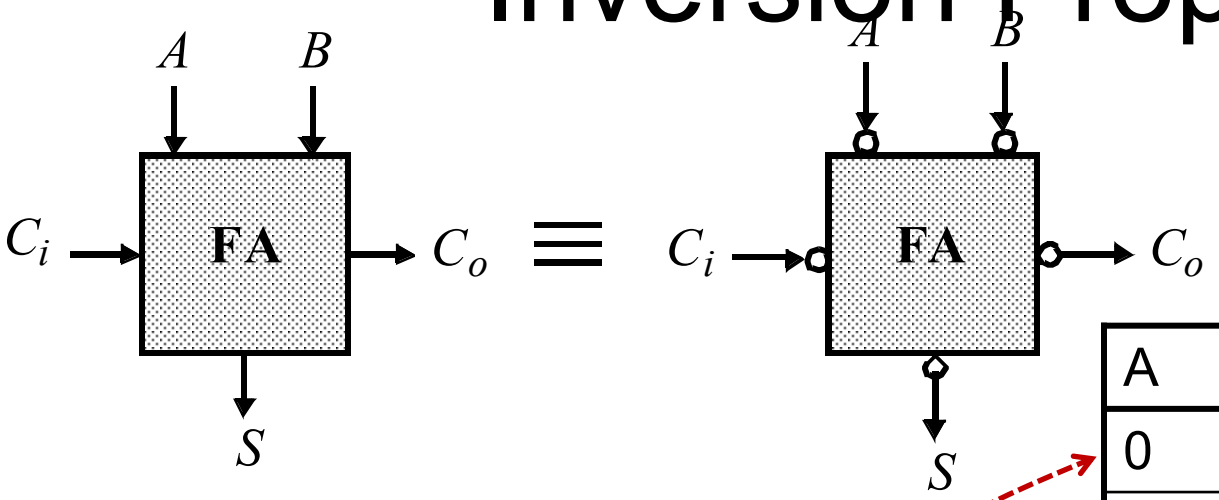


$$\sum_{i=1}^{N}\left(R_s+\frac{1}{2N}R+\frac{i-1}{N}R\right)\frac{C}{N}=R_sC+\frac{RC}{2N}+\frac{N(N-1)}{2N}R\,\frac{C}{N}=R_s\,C+\frac{N}{2N}RC$$

Exam contents

- Combination circuit logic
- wire
- **Datapath**
- Sequential circuit Logic

Inversion Property



$$\bar{S}(A,B,C_i) = S(\bar{A},\bar{B},\bar{C_i})$$

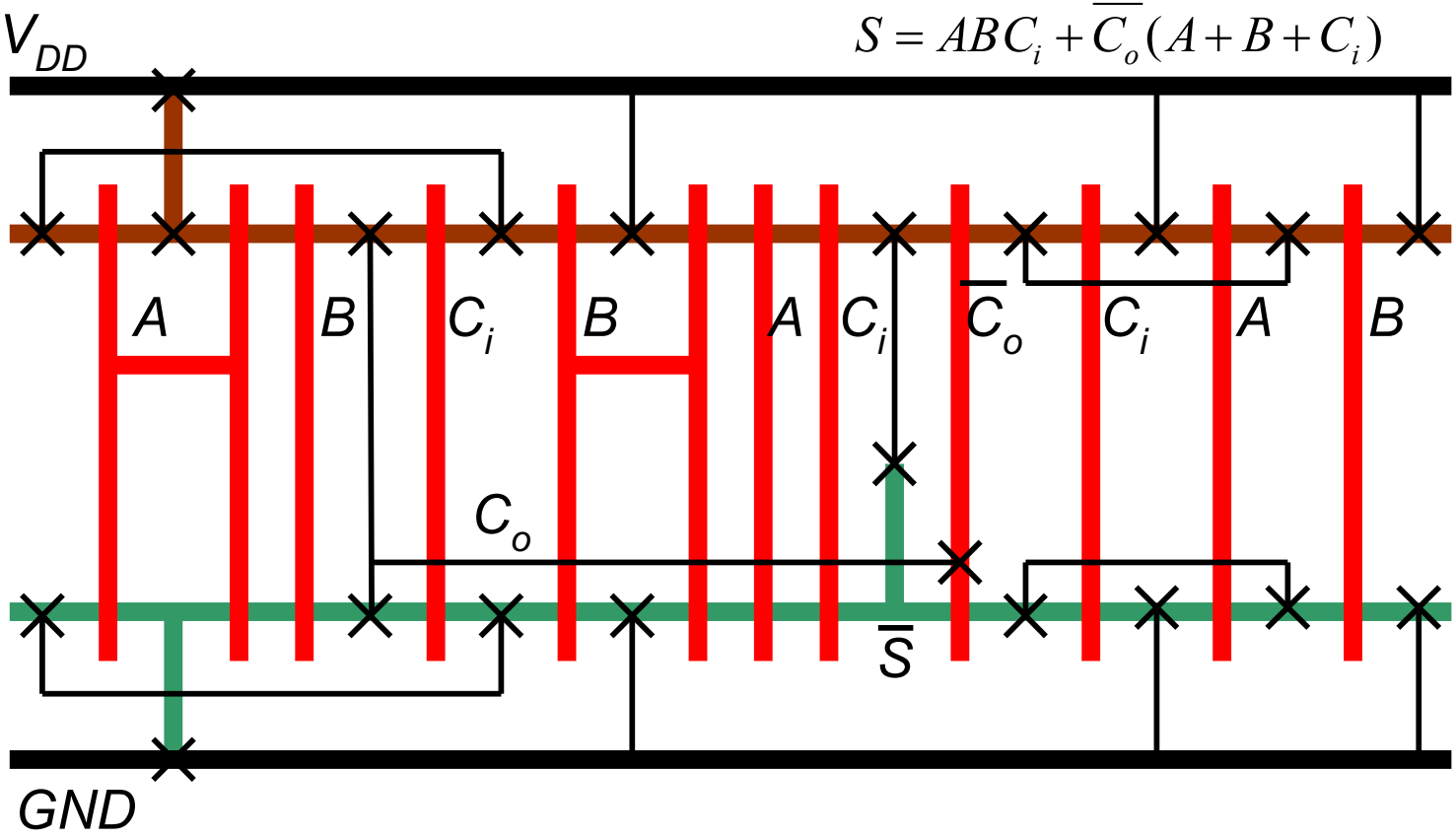
$$\overline{C_o}(A,B,C_i) = C_o(\bar{A},\bar{B},\bar{C_i})$$

A	B	C	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Mirror Adder-Stick Diagram

$$C_o = AB + BC_i + AC_i$$

$$S = ABC_i + \overline{C_o}(A + B + C_i)$$



Single-Bit Addition

Half Adder

$$S = A \oplus B$$

$$C_{out} = AB$$

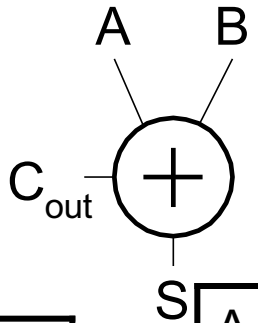
A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = A\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}\,\overline{B}C + ABC$$

$$= A \oplus B \oplus C = P \oplus C$$

$$C_{out} = AB + (A+B)C$$

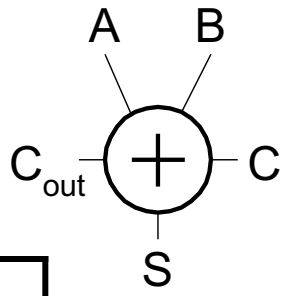
$$= \overline{\overline{A}\,\overline{B} + (\overline{A} + \overline{B})\overline{C}} = MAJ(A,B,C)$$



Full Adder

$$S = A \oplus B \oplus C$$

$$C_{out} = MAJ(A,B,C)$$



A	B	C	C_{out}	S	P	G
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	1	1	0
0	1	1	1	0	1	0
1	0	0	0	1	1	0
1	0	1	1	0	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	1

Delete/Kill

propagate

generate

Full Adder Design IV

$$S = A \oplus B \oplus C = P \oplus \mathrm{C}$$

$$C_{out} = AB + (A \oplus B)C = AAB + A\bar{A}\bar{B} + (A \oplus B)C = A\bar{P} + PC$$

$$\overline{C_{out}} = (\bar{A} + P)(\bar{P} + \bar{C}) = \bar{A}\bar{P} + \bar{A}\bar{C} + P\bar{C} = \bar{A}\bar{P} + \bar{A}\bar{C}\bar{P} + \bar{A}\bar{C}P + P\bar{C} = \bar{A}\bar{P} + P\bar{C}$$

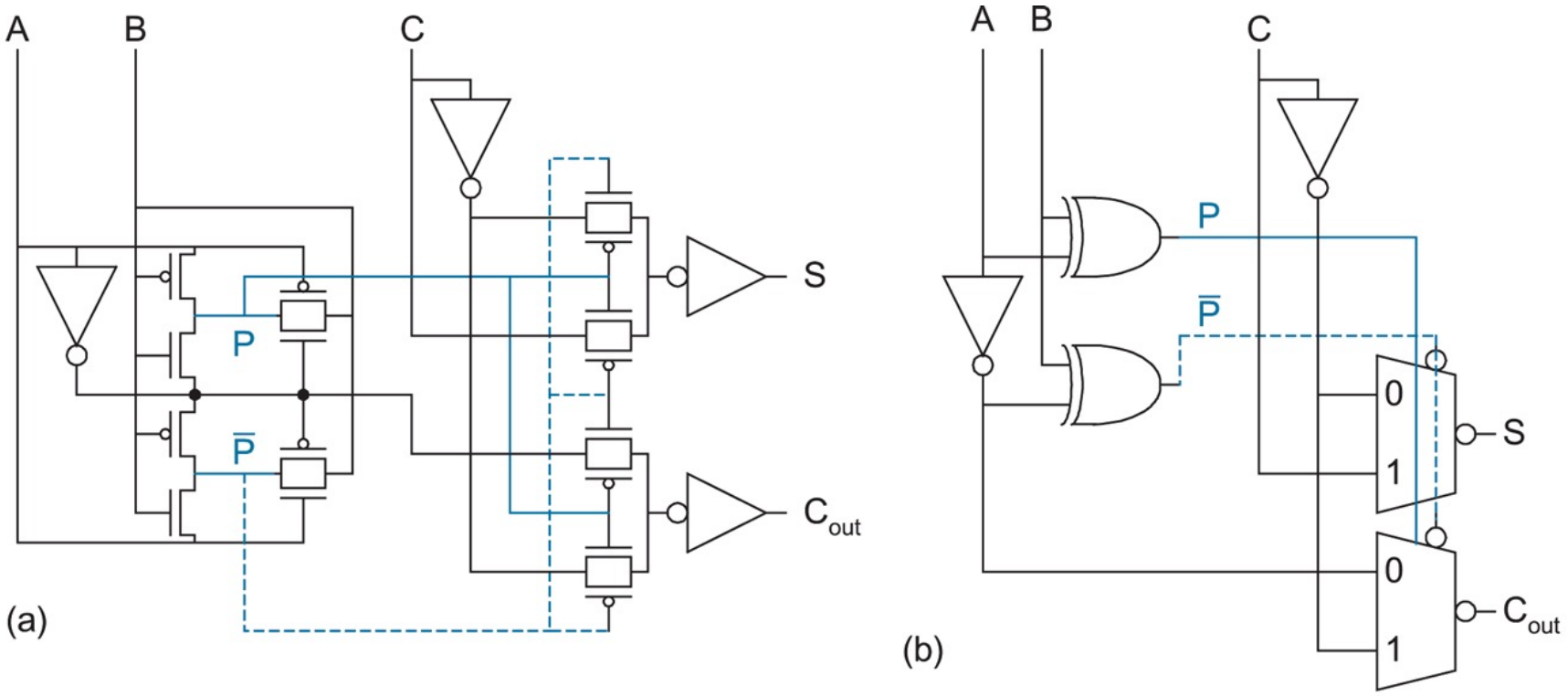
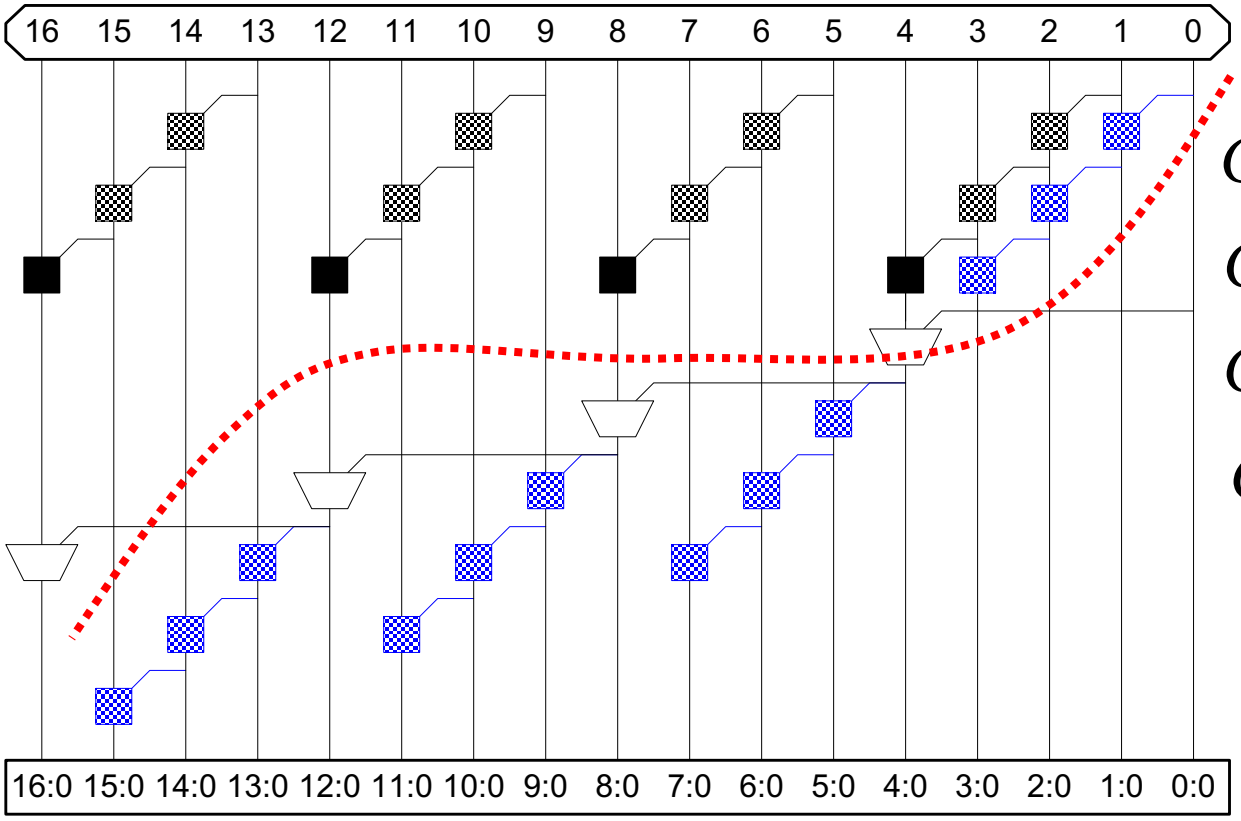


FIG 10.6 Transmission gate full adder

Carry-Skip PG Diagram



$$G_{4:0} = G_{4:1} + P_{4:1}G_{0:0}$$

$$G_{8:0} = G_{8:5} + P_{8:5}G_{4:0}$$

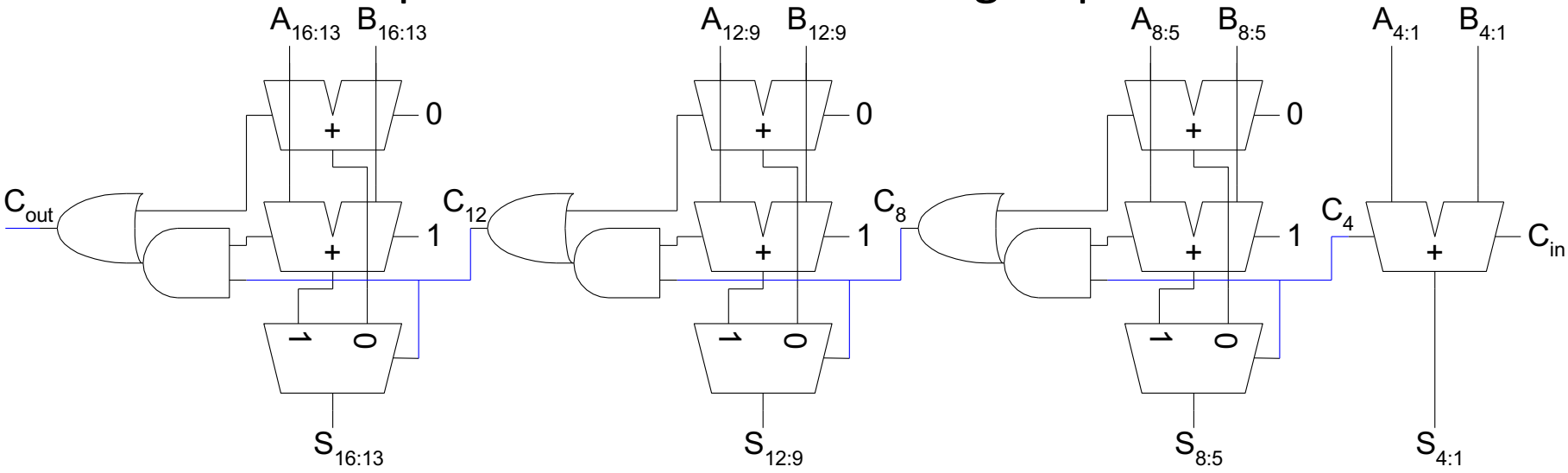
$$G_{12:0} = G_{12:0} + P_{12:9}G_{8:0}$$

$$G_{16:0} = G_{16:13} + P_{16:13}G_{12:0}$$

$$t_{\text{skip}} = t_{pg} + \left[2(n-1) + (k-1)\right] t_{AO} + t_{\text{xor}}$$

Carry-Select Adder

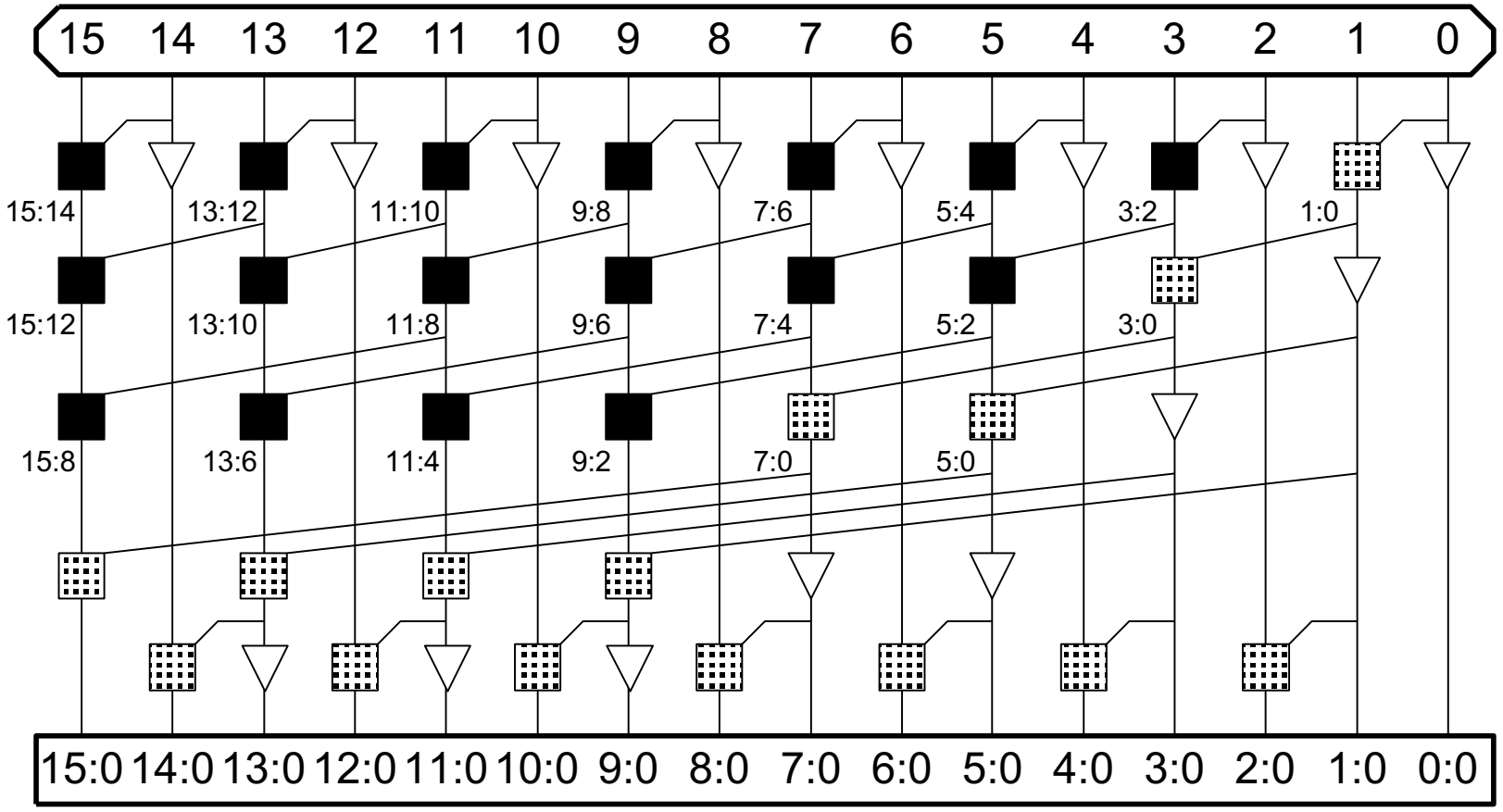
- Trick for critical paths dependent on late input X
  - Precompute two possible outputs for X = 0, 1
  - Select proper output when X arrives
- Carry-select adder precomputes n-bit sums
  - For both possible carries into n-bit group



$$t_{select} = t_{pg} + [n + (k-2)]t_{AO} + t_{mux}$$

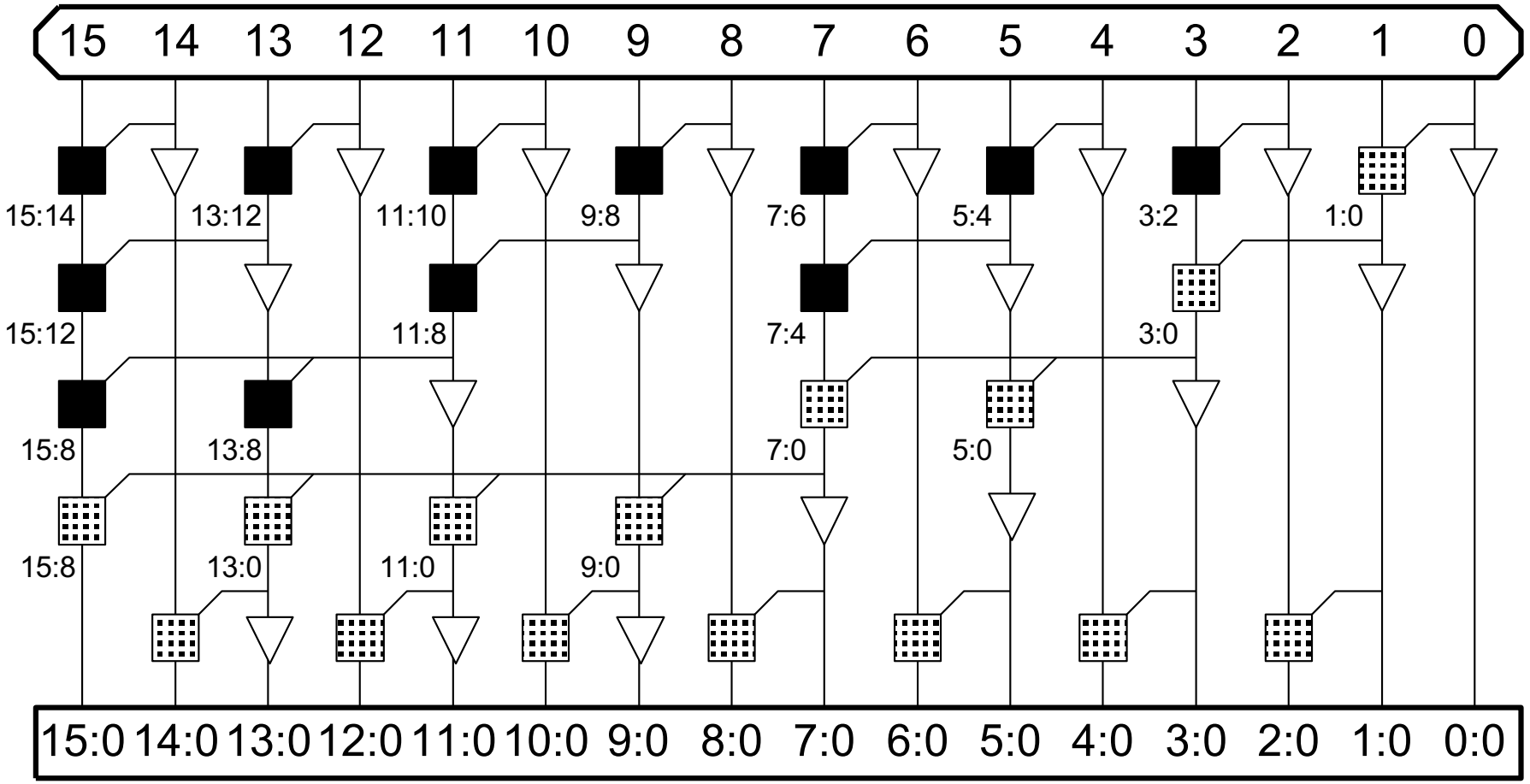
Han-Carlson(B+K tree)

Kogge Stone————Brent Kung



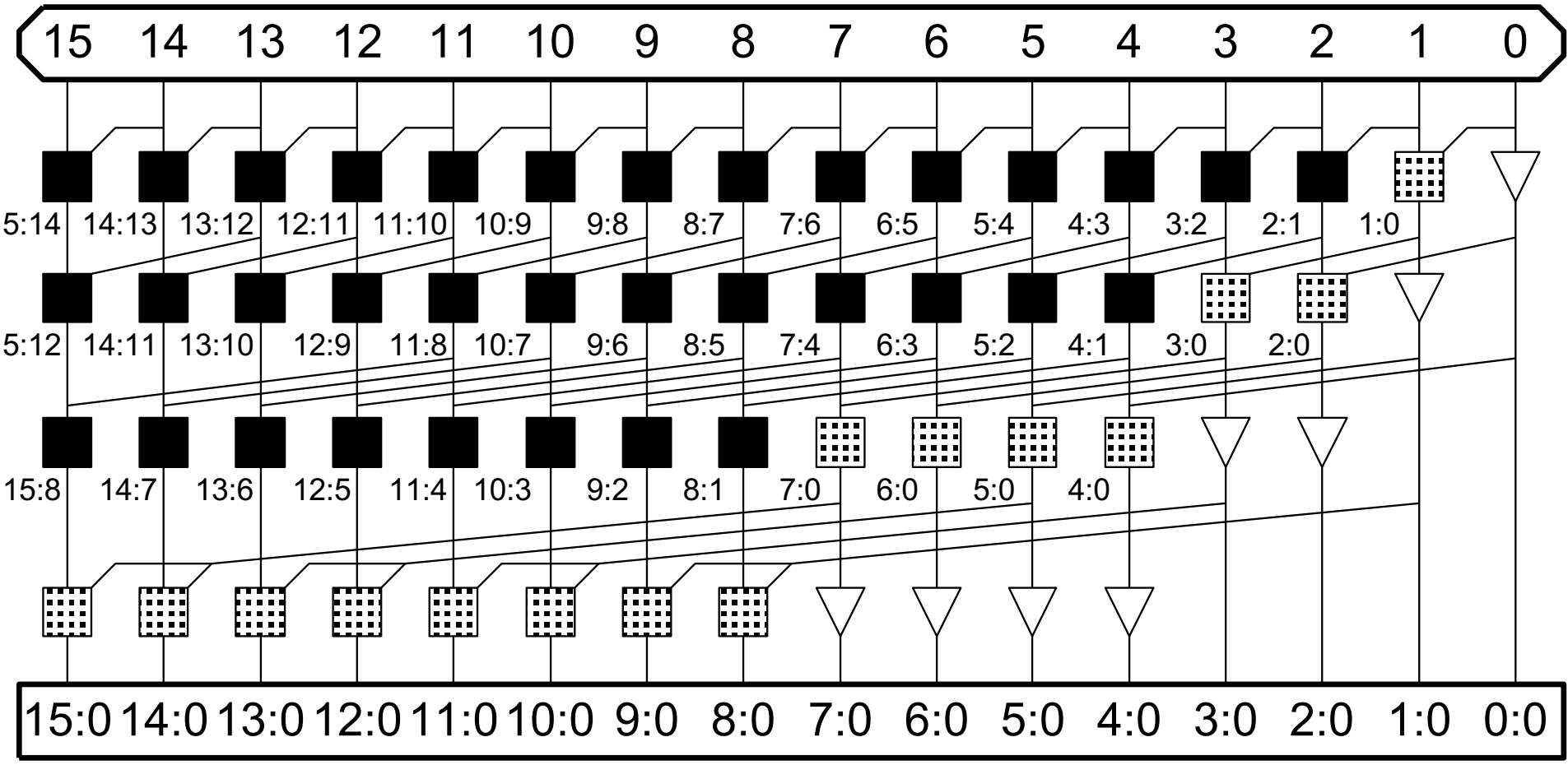
Ladner-Fischer(S+B)

Sklansky————Brent Kung



Knowles [1, 1, 1,2](S+K)

Kogge Stone————Sklansky

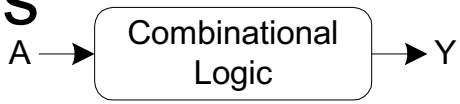


Exam contents

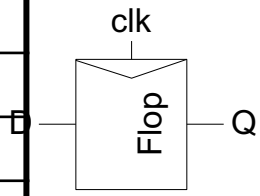
- Combination circuit logic
- wire
- Datapath
- **Sequential circuit Logic**

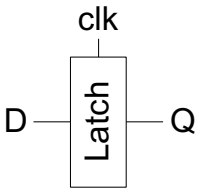
Timing Diagrams

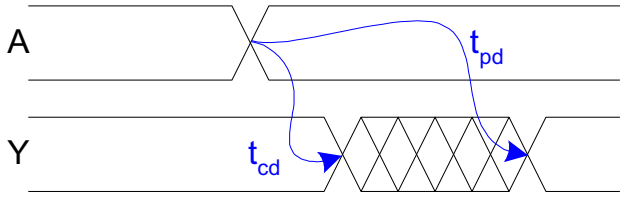
Contamination and **Propagation** Delays

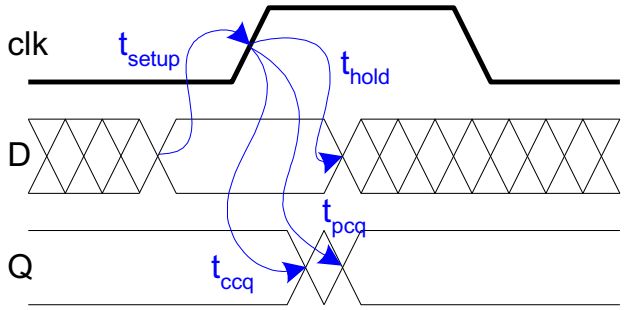


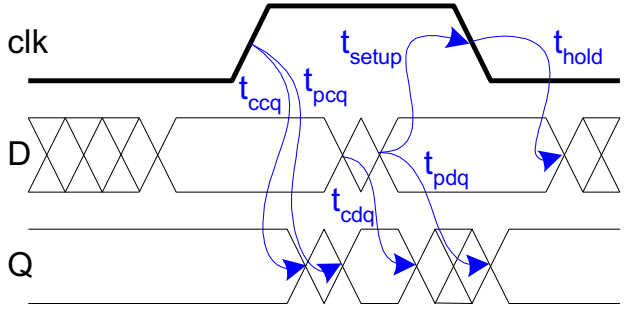
t_{pd}	Logic Prop. Delay
t_{cd}	Logic Cont. Delay
t_{pcq}	Latch/Flop Clk-Q Prop Delay
t_{ccq}	Latch/Flop Clk-Q Cont. Delay
t_{pdq}	Latch D-Q Prop Delay
t_{pcq}	Latch D-Q Cont. Delay
t_{setup}	Latch/Flop Setup Time
t_{hold}	Latch/Flop Hold Time











Clock skew about FF *negative skew*

$$t_{pdq} \leq T_c - (t_{pcq} + t_{setup} + t_{skew})$$

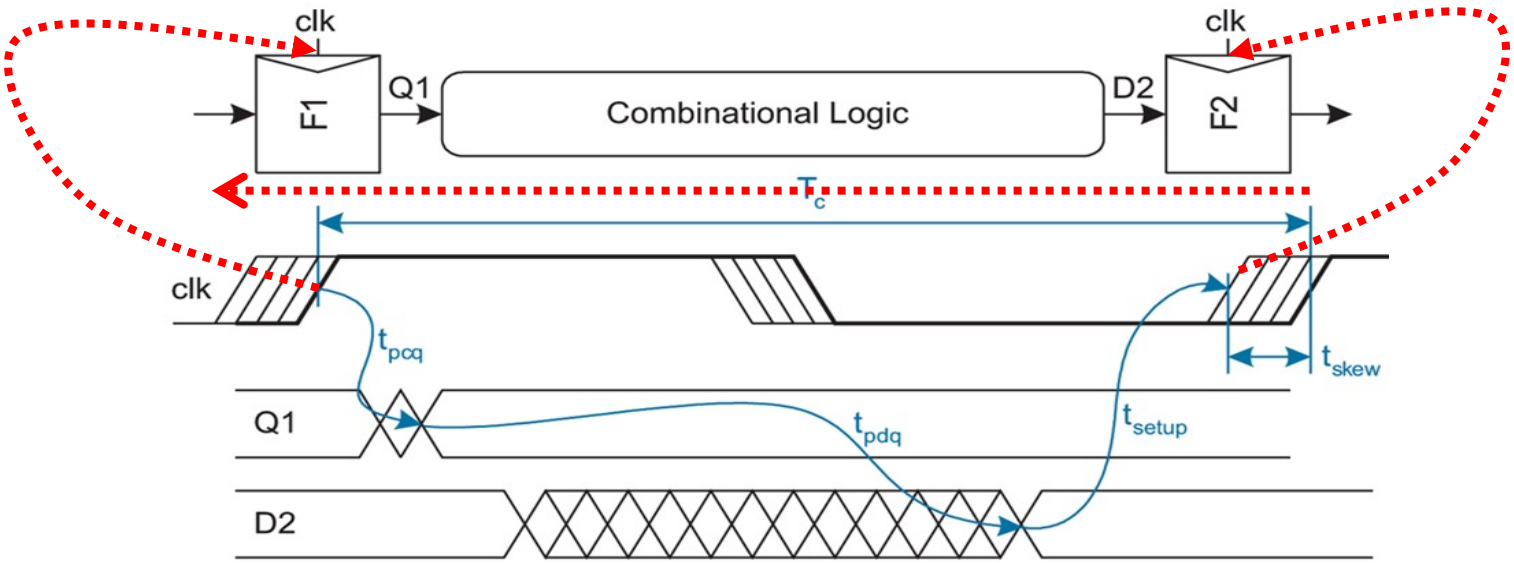
Large is better !

Positive skew for better condition

Ex. t_{pcq}=20ps,t_{setup}=30ps, T_c=300ps,t_{skew}=50ps

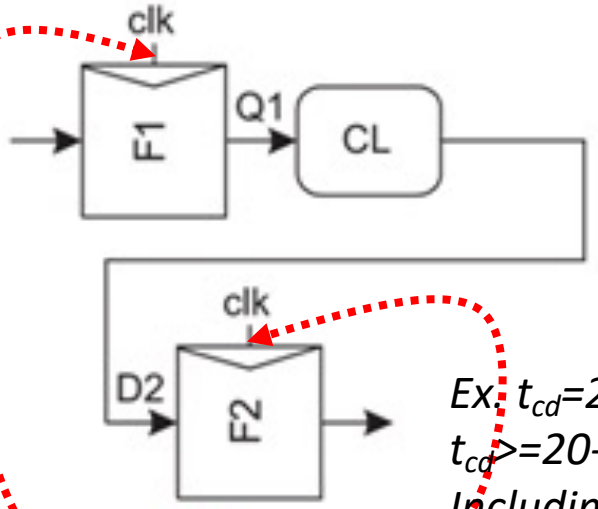
t_{pdq}<=300-20-30=250 , t_{pdq} =240 is OK

Including skew delay, t_{pdq} <=300-20-30-50=200ps, t_{pdq} =240 is not OK!



Clock skew about FF

Small is better !



$$t_{cd} \geq t_{hold} - t_{ccq} + t_{skew}$$

Positive skew for worse condition

Ex. t_{cd}=20ps,t_{hold}=20ps, t_{ccq}=2ps,t_{skew}=5ps
t_{cd}>=20-2=18, t_{cd}=20 is OK!
Including skew delay, t_{cd} !>=20-2+5=23,20 is Not OK